



Projekt

Numerická homogenizace mechanických vlastností

Studijní program:

Studijní obor:

Autor práce:

Vedoucí práce:

B0588A110003 – Aplikované vědy v inženýrství

B0588A110003AVI – Aplikované vědy v inženýrství

Ronald Christopher Siddall

doc. Mgr. Jan Stebel, Ph.D.

Liberec 2024

ZADÁNÍ SEMESTRÁLNÍHO PROJEKTU

Jméno a příjmení: **Ronald Christopher Siddall**
Název práce: **Numerická homogenizace mechanických vlastností**
Zadávající katedra: **Ústav nových technologií a aplikované informatiky**
Vedoucí práce: **doc. Mgr. Jan Stebel Ph.D.**
Rozsah práce: **15–20 stran**

Zásady pro vypracování:

1. Rešerše:
 - Nastudování základů mechaniky pružných těles a jejich modelování metodou konečných prvků.
 - Nastudování základů výpočetní mikromechaniky a její využití pro modelování heterogenních materiálů.
2. Numerická homogenizace
 - Seznámení se s výpočetním SW Flow123d.
 - Návrh výpočetního postupu pro určení efektivního elastického tenzoru ve 2D.
 - Implementace zvoleného postupu do programu, který provede výpočetní analýzu detailního (mikrostrukturálního) modelu a určí efektivní elastický tenzor.
3. Aplikace
 - Demonstrace výpočetního postupu na modelových úlohách s heterogenními materiály.

Seznam odborné literatury:

- [1] JOHNSON, C. Numerical solution of partial differential equations by the finite element method. Courier Corporation, 2012.
- [2] SCHÄFER, M. Computational engineering: introduction to numerical methods. Berlin: Springer, 2006.
- [3] ZOHDI, T. I.; WRIGGERS, P. An introduction to computational micromechanics. Springer Science & Business Media, 2004.

V Liberci dne

.....
doc. Mgr. Jan Stebel Ph.D.

Prohlášení

Prohlašuji, že svůj projekt jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mého projektu a konzultantem.

Jsem si vědom toho, že na můj projekt se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mého projektu pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li projekt nebo poskytnu-li licenci k jeho využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Beru na vědomí, že můj projekt bude zveřejněn Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

24. 5. 2024

Ronald Christopher Siddall

Numerická homogenizace mechanických vlastností

Abstrakt

Tato práce se zaměřuje na numerickou homogenizaci mechanických vlastností dvoudimenzionálních fyzikálních problémů. Numerická homogenizace umožňuje modelovat mechanické vlastnosti materiálů s komplikovanou mikroheterogenní strukturou. K řešení byl implementován program, který automaticky prováděl homogenizaci s využitím simulátoru Flow123d.

Klíčová slova: efektivní elastický tenzor, homogenizace, reprezentativní objemový vzorek, mikromechanika, mechanika pružných těles, modelování heterogenních materiálů

Abstract

This work focuses on the numerical homogenization of mechanical properties of two-dimensional physics problems. Numerical homogenization allows to model the mechanical properties of materials with complicated micro heterogeneous structure. A program was developed to automatically perform the homogenization using the Flow123d simulator.

Keywords: effective elastic tensor, homogenization, representative elementary volume, micromechanics, mechanics of elastic bodies, modelling of heterogeneous materials

Poděkování

Chci zvláště poděkovat vedoucímu své práce, doc. Mgr. Janu Stebelovi, Ph.D.

Jeho odborné znalosti, konstruktivní kritika, trpělivost, férový přístup a neustálá ochota poskytovat cenné rady a připomínky během tvorby této práce byly pro mě nenahraditelné.

Kromě toho bych rád poděkoval svým rodičům a rodině, kteří mi též dodávali potřebnou podporu, energii a prostor k práci.

A samozřejmě kolegům z AVI.

Obsah

1	Úvod	8
2	Teorie	9
2.1	Matematický popis mechaniky pružných těles	9
2.1.1	Deformace a posun	9
2.1.2	Pružnost, pevnost, tuhost a Youngův modul pružnosti	10
2.1.3	Poissonova konstanta	11
2.1.4	Zobecněný Hookův zákon a bilance sil	11
2.1.5	Voigtova notace	12
2.2	Numerická homogenizace	13
2.2.1	Definice homogenizace a efektivního elastického tenzoru . . .	13
2.2.2	Okrajové podmínky problému a matice zatížení	13
2.2.3	Efektivní elastický tenzor pro obecné matice zatížení	14
3	Popis mikrostruktur a výpočtu efektivního tenzoru	16
3.1	Šachovnice a sendvič	16
3.2	Popis numerického výpočtu výsledků	18
3.2.1	Efektivní elastický tenzor pro konkrétní matice zatížení	18
3.2.2	Reprezentativní objemový vzorek a tolerance	19
3.2.3	Výpočet relativního residua	20
4	Popis, funkce a dokumentace programu	22
4.1	Předpoklady před použitím programu	22
4.1.1	Spouštění v OS Windows	22
4.2	Obecná struktura programu	23
4.3	Popis jednotlivých tříd a souborů	25
4.3.1	config_file.yaml	25
4.3.2	GenerateMesh.py	26
4.3.3	ConfigManager.py	27
4.3.4	GenerateVtuFiles.py	27
4.3.5	EffectiveElasticTensor.py	27
4.3.6	main.py	29
4.3.7	automatic_simulation.py	30

5	Ukázkové výpočty vzorových úloh	31
5.1	Ukázka jednorázové simulace	31
5.2	Ukázka automatické simulace	36
6	Výsledky provedených výpočtů	39
6.1	Rozbor všech provedených výpočtů	39
6.2	Finální výsledky provedených výpočtů	40
6.2.1	Šachovnice, $Y1 = 100$, $Y2 = 50$	40
6.2.2	Šachovnice, $Y1 = 200$, $Y2 = 50$	41
6.2.3	Šachovnice, $Y1 = 500$, $Y2 = 50$	41
6.2.4	Šachovnice, $Y1 = 2500$, $Y2 = 50$	41
6.2.5	Sendvič, $Y1 = 100$, $Y2 = 50$	42
6.2.6	Sendvič, $Y1 = 200$, $Y2 = 50$	42
6.2.7	Sendvič, $Y1 = 500$, $Y2 = 50$	42
6.2.8	Sendvič, $Y1 = 2500$, $Y2 = 50$	43
6.2.9	Tabulky s přehledem hodnot rozměrů REV	43
6.3	Grafy různých závislostí	43
6.4	Validace výsledků	46
6.4.1	Výsledky pro sudé a liché násobky n	46
6.4.2	Analýza spočtených výsledků pomocí pravidla směsí	46
6.5	Sumarizace výsledků a pozorování	47
7	Závěr	48

1 Úvod

Všichni jsme se již někdy setkali s nějakým materiálem, který má heterogenní tj. prostorově proměnnou mikrostrukturu. Materiálů s touto mikrostrukturou je velké množství, ale mezi ty nejvýznamnější patří např. horniny, biomateriály, dřevo či kompozitní materiály.

Žula je příkladem horniny složené z různých minerálů, především z křemene, živce a slídy. Stejně tak i biomateriály, jako jsou kosti a svalstvo, projevují mikroheterogenitu. Svaly obsahují různé typy svalových vláken, zatímco kosti jsou tvořeny mnohými minerály a organickými látkami. A neposledně i kompozity, mezi které patří třeba kevlar, jsou vhodným příkladem heterogenního materiálu, neboť kompozity bývají obvykle složené z několika od sebe odlišných složek s různými vlastnostmi.

Co mají všechny tyto materiály společné? Kvůli jejich velmi komplexní mikrostruktuře je extrémně náročné obecně předpovídat resp. modelovat jejich mechanické vlastnosti. Mechanické vlastnosti jako tuhost, pevnost či elasticita jsou charakteristiky materiálu, které popisují jeho chování a reakce na různé mechanické zatížení (např. v tahu, tlaku či smyku).

Tyto vlastnosti jsou naprosto stěžejní pro hlubší porozumění chování daného materiálu. Pokud neznáme mechanické vlastnosti materiálu, je obtížné předpovídat jeho chování v různých situacích. Ve chvíli, kdy nedokážeme předpovědět jeho chování, činí to obrovské komplikace při zkoumání, vývoji či zlepšování těchto materiálů v průmyslu, technice, vědě, medicíně apod.

Jinými slovy obecné modelování mikroheterogenních materiálů se složitou mikrostrukturou je velmi náročný až téměř nemožný úkol, který je ale nutné řešit.

Numerická homogenizace mechanických vlastností nabízí jeden ze způsobů, jak tento problém řešit, a právě tím se zabývá tento semestrální projekt.

V tomto projektu tedy ukazují způsob, jak získat efektivní mechanické vlastnosti materiálu s mikroheterogenní strukturou pomocí numerické homogenizace.

Nejprve analyzuji problém homogenizace matematicky. Poté popisuji, jak jsem implementoval program, který automaticky počítá efektivní mechanické vlastnosti pomocí numerické homogenizace pro dva různé typy mikrostruktur. Následně se věnuji tomu, jak optimálně získat mechanické vlastnosti u různých typů materiálů. V závěru podávám ucelené shrnutí o tom, jaké jsem pozoroval výsledky a závislosti.

2 Teorie

Popis obecné numerické homogenizace pro třírozměrný prostor a libovolně velké deformace je velmi komplikovaný. Z toho důvodu se tento projekt bude zabývat zjednodušeným fyzikálním modelem, který uvažuje pouze **dvoudimenzionální** (rovinné) problémy a pouze malé, tj. **lineárně elastické deformace**. Teoretická část je rozdělená na dvě podkapitoly. Nejprve v kapitole 2.1 jsou shrnuty základní poznatky z matematické teorie pružnosti. Následně v kapitole 2.2 je popsána základní myšlenka numerické homogenizace a obecný postup pro získání efektivního elastického tenzoru.

2.1 Matematický popis mechaniky pružných těles

Začneme tím, že si definujeme skalární součiny vektorů a tenzorů. Skalární součin vektorů $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ definujeme následovně:

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i. \quad (2.1)$$

Podobně skalární součin tenzorů druhého řádu $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ se definuje takto:

$$\mathbf{A} : \mathbf{B} = \sum_{i=1}^n \sum_{j=1}^n A_{ij} B_{ij}. \quad (2.2)$$

Dále je potřeba definovat tzv. **Frobeniovu normu** tenzoru. Pokud máme obecný tenzor $\mathbf{A} \in \mathbb{R}^{n \times m}$ s prvky a_{ij} , tak poté Frobeniova norma je:

$$\|\mathbf{A}\| = \sqrt{\sum_{i=1}^n \sum_{j=1}^m a_{ij}^2} \quad (2.3)$$

2.1.1 Deformace a posun

Naším cílem je popsat reakci mikroheterogenního tělesa či materiálu při působení zátěže resp. **při deformaci**.

Pojmem deformace tělesa rozumíme změnu tvaru či objemu tělesa způsobenou působením nějakých sil. Tyto síly, také označované jako deformační, vyvolávají v tělese napětí a existuje jich více typů. Mezi ty nejčastější řadíme deformační síly v tahu,

tlaku, smyku, ohybu či torzi. Deformace jako takové se dělí na dva základní typy, a to **elastické** (pružné) a **plastické** (nepružné). Elastická deformace je reverzibilní (vratný) děj, tj. pokud těleso elasticky deformujeme, tak se po ukončení působení těleso vrátí do původního stavu. Pokud se jedná o plastickou deformaci, mluvíme o ireverzibilních (nevratných) dějích. Jinými slovy při plastických deformacích dojde k permanentní změně tvaru daného materiálu (kap. 5.2. (Brdička et al., 2005)).

Je celkem přirozené usoudit, že k elastickým deformacím dojde pouze pro relativně malé síly (samozřejmě závisí na typu materiálu), zatímco k plastickým deformacím dojde při působení větších sil.

Pro matematický popis elastické deformace nejprve zavedeme tzv. **vektor posunutí**.

Mějme libovolný bod, jehož kartézské souřadnice před deformací označíme x_i . Působením vnějších sil dojde k posunu bodu o vektor \mathbf{u} do jiného bodu se souřadnicemi y_i . Vektor posunutí (či zkráceně posun) \mathbf{u} je poté pro tento bod definován následovně (kap. 4.1. (Brdička et al., 2005)):

$$\mathbf{u} = \mathbf{y} - \mathbf{x}, \quad (2.4)$$

kde vektor \mathbf{x} se složkami $x_i, i = 1, 2$ je poloha bodu před deformací a vektor \mathbf{y} se složkami $y_i, i = 1, 2$ je poloha bodu po deformaci.

Je nutné zdůraznit, že vektor posunutí je **obecný** popis změny polohy bodu. Jinými slovy vektor posunutí sice souvisí s deformací, ale nejsou to ekvivalentní pojmy. Například translace (fyzický posun) nebo rotace (otočení) povedou k netriviálnímu (nenulovému) posunutí, tj. $\mathbf{u} \neq 0$. Nicméně při těchto pohybech nedochází k deformaci, protože nedošlo ke změně tvaru a objemu tělesa.

Změnu tvaru popisuje tzv. **tenzor malých deformací** (Schafer, 2006) ϵ , jehož prvky jsou dány vztahem:

$$\epsilon_{kl} = \left(\frac{\partial u_k}{\partial x_l} + \frac{\partial u_l}{\partial x_k} \right), \quad k, l = 1, 2. \quad (2.5)$$

2.1.2 Pružnost, pevnost, tuhost a Youngův modul pružnosti

Pevnost, pružnost, tuhost a Youngův modul jsou pojmy, které souvisí se vztahem deformace a působících sil.

Pružnost je schopnost materiálu se deformovat vlivem napětí a následně se vrátit do původního stavu.

Pevnost je definovaná tzv. mezí pevnosti. Mez pevnosti je maximální napětí, které materiál dokáže snést před trvalým deformováním nebo porušením.

Tuhost vyjadřuje, jak rychle se materiál deformuje v reakci na vnější sílu resp. odpor proti deformaci pod zatížením.

Youngův modul pružnosti Y , jinými slovy modul pružnosti v tahu, lze interpretovat jako míru tuhosti materiálu v tahu, ale přesněji je definován jako poměr napětí v tahu σ a relativního prodloužení¹ ϵ . Relativní prodloužení je definované jako podíl

¹Existuje více označení např. poměrná deformace, podélná deformace aj.

prodloužení materiálu Δl vůči jeho původní délce l_0 (kap. 5.2. (Brdička et al., 2005)):

$$Y = \frac{\sigma}{\varepsilon}, \quad (2.6)$$

$$\varepsilon = \frac{\Delta l}{l_0} = \frac{l - l_0}{l_0}. \quad (2.7)$$

Rovnice (2.6) je někdy označovaná jako Hookův zákon elasticity, často zapisovaná ve tvaru $\sigma = Y\varepsilon$ (rovnice 5.2.1 kap. 5.2. (Brdička et al., 2005)). Tato lineární rovnice je platná pouze pro malé hodnoty napětí (což je v souladu s předpoklady tohoto projektu, neboť se zabýváme pouze lineárně elastickými deformacemi). Vzhledem k lineární povaze této rovnice lze Y vnímat jako směrnici, tj. konstantu úměrnosti v lineární závislosti σ na ε .

2.1.3 Poissonova konstanta

Poissonovu konstantu ν lze definovat následovně:

$$\nu = -\frac{\varepsilon_x}{\varepsilon_y}, \quad (2.8)$$

kde ε_x je podélné a ε_y je příčné protažení.

Pro většinu materiálů platí $\varepsilon_x \geq 0, \varepsilon_y < 0$, což znamená, že se při natažení daný materiál v příčném směru zúží. Pro tyto nejčastější materiály leží hodnota ν v rozsahu 0 až 0,5.

Existují ale i moderní tzv. auxetické materiály, které se při natažení zvětší i v příčném směru ($\varepsilon_y > 0$) a platí pro ně $\nu < 0$ (Carneiro et al., 2013).

2.1.4 Zobecněný Hookův zákon a bilance sil

Youngův modul pružnosti a Poissonova konstanta charakterizují elastickou odezvu tzv. *izotropních materiálů*, jejichž mechanické vlastnosti jsou **stejné** pro všechny směry. Chování obecných anizotropních materiálů popisuje tzv. **elastický tenzor** prostřednictvím **zobecněného Hookova zákona**.

Mějme pevné mikroheterogenní těleso, které je ohraničené oblastí $\Omega \subset \mathbb{R}^2$. Potom lineární elastickou deformaci daného tělesa na oblasti Ω lze popsat pomocí zobecněného Hookova zákona:

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon}, \quad \text{tj.} \quad \sigma_{ij} = \sum_{k,l=1}^2 C_{ijkl} \varepsilon_{kl}, \quad i, j = 1, 2, \quad (2.9)$$

kde $\boldsymbol{\sigma}, \boldsymbol{\varepsilon} \in \mathbb{R}^{2 \times 2}$, $\mathbf{C} \in \mathbb{R}^{2 \times 2 \times 2 \times 2}$, $\boldsymbol{\sigma}$ je **tenzor napětí** a \mathbf{C} elastický tenzor².

Samotný zobecněný Hookův zákon nestačí pro úplný popis vektorových polí posunutí \mathbf{u} a napětí $\boldsymbol{\sigma}$ v celé oblasti Ω . Pro kompletní popis je nutné uvažovat tzv.

²Také se nazývá Hookův tenzor.

bilanci sil a okrajové podmínky:

$$\begin{aligned} \text{v } \Omega : \quad \nabla \cdot \boldsymbol{\sigma} &= \mathbf{f} \quad (\text{balance sil}), \\ \text{na } \Gamma_1 : \quad \mathbf{u} &= \mathbf{u}_D \quad (\text{Dirichletova podmínka}), \\ \text{na } \Gamma_2 : \quad \boldsymbol{\sigma} \cdot \mathbf{n} &= \mathbf{t}_N \quad (\text{Neumannova podmínka}), \end{aligned} \quad (2.10)$$

kde ∇ je nabla operátor, \mathbf{f} jsou vnější působící síly, Γ_1 a Γ_2 jsou dvě disjunktní části hranice $\partial\Omega$, \mathbf{n} je jednotkový normálový vektor, \mathbf{u}_D je zadaný vektor posunutí a \mathbf{t}_N je zadaná povrchová síla.

2.1.5 Voigtova notace

V matematice je Voigtova notace způsob, jak reprezentovat **symetrický** tenzor pomocí tenzoru nižšího řádu (Povey, 2023).

Uvažujme dvoudimenzionální symetrický tenzor druhého řádu $\mathbf{A} \in \mathbb{R}^{2 \times 2}$ a dvoudimenzionální symetrický tenzor čtvrtého řádu $\mathbf{B} \in \mathbb{R}^{2 \times 2 \times 2 \times 2}$.

Ve Voigtově notaci se tenzor \mathbf{A} zredukuje na tenzor prvního řádu (vektor) a tenzor \mathbf{B} se zredukuje na tenzor druhého řádu (matici):

$$\begin{aligned} \mathbf{A} \in \mathbb{R}^{2 \times 2} &\longrightarrow \tilde{\mathbf{A}} \in \mathbb{R}^3, \\ \mathbf{B} \in \mathbb{R}^{2 \times 2 \times 2 \times 2} &\longrightarrow \tilde{\mathbf{B}} \in \mathbb{R}^{3 \times 3}, \end{aligned} \quad (2.11)$$

kde $\tilde{\mathbf{A}}$ a $\tilde{\mathbf{B}}$ značí zredukované tenzory. Tyto zredukované tenzory obsahují pouze *různé* prvky původních tenzorů:

$$\tilde{\mathbf{A}} = \begin{bmatrix} A_{11} \\ A_{22} \\ A_{12} \end{bmatrix}, \quad \tilde{\mathbf{B}} = \begin{bmatrix} B_{1111} & B_{1122} & B_{1112} \\ B_{2211} & B_{2222} & B_{2212} \\ B_{1211} & B_{1222} & B_{1212} \end{bmatrix}.$$

Vzhledem k tomu, že $\boldsymbol{\sigma}$, $\boldsymbol{\varepsilon}$ i \mathbf{C} jsou symetrické tenzory, je možné na ně aplikovat (2.11). Pro tenzor malých deformací $\boldsymbol{\varepsilon}$ se však zavádí drobná korekce:

$$\tilde{\boldsymbol{\varepsilon}} = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \gamma_{12} \end{bmatrix} \quad (2.12)$$

kde $\gamma_{12} = 2\varepsilon_{12}$. Díky tomu je pak zachován skalární součin $\tilde{\boldsymbol{\sigma}} \cdot \tilde{\boldsymbol{\varepsilon}} = \boldsymbol{\sigma} : \boldsymbol{\varepsilon}$, což je veličina důležitá pro vyjádření mechanické energie. Ve Voigtově notaci přejde rovnice (2.9) na tvar:

$$\tilde{\boldsymbol{\sigma}} = \tilde{\mathbf{C}} \tilde{\boldsymbol{\varepsilon}}, \quad (2.13)$$

kde

$$\tilde{\boldsymbol{\sigma}} = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix}, \quad \tilde{\mathbf{C}} = \begin{bmatrix} C_{1111} & C_{1122} & C_{1112} \\ C_{2211} & C_{2222} & C_{2212} \\ C_{1211} & C_{1222} & C_{1212} \end{bmatrix}, \quad \tilde{\boldsymbol{\varepsilon}} = \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \gamma_{12} \end{bmatrix}, \quad (2.14)$$

což je **zobecněný Hookův zákon** ve Voigtově notaci.

2.2 Numerická homogenizace

Mikroheterogenní materiály mají v různých místech různé vlastnosti, které jsou obecně popsány elastickým tenzorem. Jinými slovy tyto materiály jsou prostorově proměnné, což lze matematicky vnímat jako závislost všech dílčích tenzorů vystupujících v rovnici (2.13) na poloze prostoru. Do rovnice (2.13) lze tedy přidat i závislost na \mathbf{x} resp. $\mathbf{u}(\mathbf{x})$:

$$\tilde{\boldsymbol{\sigma}}(\mathbf{x}) = \tilde{\mathbf{C}}(\mathbf{x})\tilde{\boldsymbol{\varepsilon}}(\mathbf{u}(\mathbf{x})). \quad (2.15)$$

Hlavním cílem práce je pomocí homogenizace nahradit prostorově závislý tenzor $\tilde{\mathbf{C}}(\mathbf{x})$ konstantním efektivním elastickým tenzorem $\tilde{\mathbf{C}}^*$ s co nejpodobnějšími vlastnostmi.

2.2.1 Definice homogenizace a efektivního elastického tenzoru

Homogenizace se provede s využitím operace průměrování $\langle \cdot \rangle$, definované jako:

$$\langle \cdot \rangle = \frac{1}{|\Omega|} \int_{\Omega} \cdot d\Omega. \quad (2.16)$$

Pokud aplikujeme tento operátor na rovnici (2.15), dostáváme:

$$\frac{1}{|\Omega|} \int_{\Omega} \tilde{\boldsymbol{\sigma}}(\mathbf{x}) d\Omega = \frac{1}{|\Omega|} \int_{\Omega} \tilde{\mathbf{C}}(\mathbf{x})\tilde{\boldsymbol{\varepsilon}}(\mathbf{u}(\mathbf{x})) d\Omega. \quad (2.17)$$

Naším cílem je nahradit tenzor $\tilde{\mathbf{C}}(\mathbf{x})$ tenzorem $\tilde{\mathbf{C}}^*$ takovým, že je nezávislý na \mathbf{x} a splňuje následující požadavek:

$$\frac{1}{|\Omega|} \int_{\Omega} \tilde{\boldsymbol{\sigma}}(\mathbf{x}) d\Omega = \tilde{\mathbf{C}}^* \cdot \frac{1}{|\Omega|} \int_{\Omega} \tilde{\boldsymbol{\varepsilon}}(\mathbf{u}(\mathbf{x})) d\Omega$$

resp.

$$\langle \tilde{\boldsymbol{\sigma}} \rangle = \langle \tilde{\mathbf{C}}\tilde{\boldsymbol{\varepsilon}} \rangle = \tilde{\mathbf{C}}^* \langle \tilde{\boldsymbol{\varepsilon}} \rangle. \quad (2.18)$$

Homogenizaci rovnice (2.13) lze tedy symbolicky zapsat následujícím způsobem:

$$\tilde{\boldsymbol{\sigma}} = \tilde{\mathbf{C}}\tilde{\boldsymbol{\varepsilon}} \xrightarrow{\text{homogenizace}} \langle \tilde{\boldsymbol{\sigma}} \rangle = \tilde{\mathbf{C}}^* \langle \tilde{\boldsymbol{\varepsilon}} \rangle \quad (2.19)$$

2.2.2 Okrajové podmínky problému a matice zatížení

Vzhledem k tomu, že $\tilde{\mathbf{C}}^* \in \mathbb{R}^{3 \times 3}$ a $\langle \tilde{\boldsymbol{\sigma}} \rangle, \langle \tilde{\boldsymbol{\varepsilon}} \rangle \in \mathbb{R}^3$, představuje vztah (2.19) soustavu 3 rovnic o 9 neznámých, což není dostatečné množství rovnic pro nalezení $\tilde{\mathbf{C}}^*$.

Pro kompletní určení všech prvků tenzoru $\tilde{\mathbf{C}}^*$ je nutné provést určitý počet **lineárně nezávislých** zatížení prostřednictvím **okrajové podmínky** definované takto:

$$\mathbf{u}_D = \mathbf{E}\mathbf{x}, \quad (2.20)$$

kde \mathbf{u}_D je posun, $\mathbf{E} = \begin{bmatrix} E_{11} & E_{12} \\ E_{12} & E_{22} \end{bmatrix}$ značí **symetrickou matici zatížení** a $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$.

Dosazením těchto okrajových podmínek do problému (2.10) získáme pole posunutí \mathbf{u} a pole napětí $\boldsymbol{\sigma}$ v celé oblasti Ω .

Výsledkem průměrování jsou pak 3 *různé zprůměrované složky napětí*. Takto získáme 3 rovnice pro prvky $\tilde{\mathbf{C}}^*$. Pro určení celého tenzoru $\tilde{\mathbf{C}}^*$ je nutné mít 3 různé případy, abychom měli 9 rovnic pro 9 neznámých. Abychom mohli obecnou matici zatížení aplikovat na rovnici (2.18), je nejprve nutné definovat vztah mezi $\tilde{\mathbf{E}}$ (obecná matice zatížení ve Voigtově notaci) a $\langle \tilde{\boldsymbol{\varepsilon}} \rangle$. Tato rovnost je výsledkem "The Average Strain Theorem", což je popsáno detailně v kapitole 4.1.1 (Zohdi a Wriggers, 2005):

$$\tilde{\mathbf{E}} = \langle \tilde{\boldsymbol{\varepsilon}} \rangle \iff \begin{bmatrix} E_{11} \\ E_{22} \\ 2E_{12} \end{bmatrix} = \begin{bmatrix} \langle \varepsilon_{11} \rangle \\ \langle \varepsilon_{22} \rangle \\ \langle \gamma_{12} \rangle \end{bmatrix}. \quad (2.21)$$

Nyní si ukážeme postup určení efektivního elastického tenzoru $\tilde{\mathbf{C}}^*$.

2.2.3 Efektivní elastický tenzor pro obecné matice zatížení

Aplikací tří různých matic zatížení získáme z rovnice (2.18) podmínky:

$$\begin{aligned} \langle \tilde{\boldsymbol{\sigma}} \rangle^1 &= \tilde{\mathbf{C}}^* \langle \tilde{\boldsymbol{\varepsilon}} \rangle^1, \\ \langle \tilde{\boldsymbol{\sigma}} \rangle^2 &= \tilde{\mathbf{C}}^* \langle \tilde{\boldsymbol{\varepsilon}} \rangle^2, \\ \langle \tilde{\boldsymbol{\sigma}} \rangle^3 &= \tilde{\mathbf{C}}^* \langle \tilde{\boldsymbol{\varepsilon}} \rangle^3, \end{aligned} \quad (2.22)$$

kde $\langle \tilde{\boldsymbol{\sigma}} \rangle^i$ a $\langle \tilde{\boldsymbol{\varepsilon}} \rangle^i, i = 1, 2, 3$, představují získané zprůměrované tenzory pro jednotlivé případy. Tyto soustavy rovnic (2.22) lze přepsat, čímž dostáváme následující kompaktnější zápis:

$$\begin{aligned} \mathbf{K} \mathbf{c}_1 &= \mathbf{s}_1, \\ \mathbf{K} \mathbf{c}_2 &= \mathbf{s}_2, \\ \mathbf{K} \mathbf{c}_3 &= \mathbf{s}_3, \end{aligned} \quad (2.23)$$

nebo také ekvivaletně:

$$\begin{bmatrix} \mathbf{K} & & \\ & \mathbf{K} & \\ & & \mathbf{K} \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \mathbf{s}_3 \end{bmatrix}, \quad (2.24)$$

kde \mathbf{K} , \mathbf{c}_1 , \mathbf{c}_2 , \mathbf{c}_3 , \mathbf{s}_1 , \mathbf{s}_2 a \mathbf{s}_3 jsou definované následovně:

$$\begin{aligned} \mathbf{K} &= \begin{bmatrix} E_{11}^1 & E_{22}^1 & 2E_{12}^1 \\ E_{11}^2 & E_{22}^2 & 2E_{12}^2 \\ E_{11}^3 & E_{22}^3 & 2E_{12}^3 \end{bmatrix}, \\ \mathbf{c}_1 &= \begin{bmatrix} C_{1111}^* \\ C_{1122}^* \\ C_{1112}^* \end{bmatrix}, \quad \mathbf{c}_2 = \begin{bmatrix} C_{2211}^* \\ C_{2222}^* \\ C_{2212}^* \end{bmatrix}, \quad \mathbf{c}_3 = \begin{bmatrix} C_{1211}^* \\ C_{1222}^* \\ C_{1212}^* \end{bmatrix}, \\ \mathbf{s}_1 &= \begin{bmatrix} \langle \sigma_{11} \rangle^1 \\ \langle \sigma_{11} \rangle^2 \\ \langle \sigma_{11} \rangle^3 \end{bmatrix}, \quad \mathbf{s}_2 = \begin{bmatrix} \langle \sigma_{22} \rangle^1 \\ \langle \sigma_{22} \rangle^2 \\ \langle \sigma_{22} \rangle^3 \end{bmatrix}, \quad \mathbf{s}_3 = \begin{bmatrix} \langle \sigma_{12} \rangle^1 \\ \langle \sigma_{12} \rangle^2 \\ \langle \sigma_{12} \rangle^3 \end{bmatrix}. \end{aligned} \quad (2.25)$$

Naším cílem je vyjádřit \mathbf{c}_1 , \mathbf{c}_2 a \mathbf{c}_3 z každé příslušné rovnice, protože tím dostaneme řešení pro prvky efektivního elastického tenzoru $\tilde{\mathbf{C}}^*$, které jsou obsažené v rámci \mathbf{c}_1 , \mathbf{c}_2 a \mathbf{c}_3 . Tuto soustavu rovnic je možné řešit analyticky nebo numericky.

3 Popis mikrostruktur a výpočtu efektivního tenzoru

3.1 Šachovnice a sendvič

Obecně mikroheterogenní materiály mohou mít velmi rozmanité mikrostruktury, které obvykle závisí na daném materiálu. V tomto projektu se řešily dva typy mikrostruktur pracovně nazvané:

- 1) **šachovnice** - prototyp mikrostruktury představující např. příčný řez vláknovým kompozitním materiálem;
- 2) **sendvič** - prototyp podélného řezu laminovaným nebo vláknovým kompozitním materiálem.

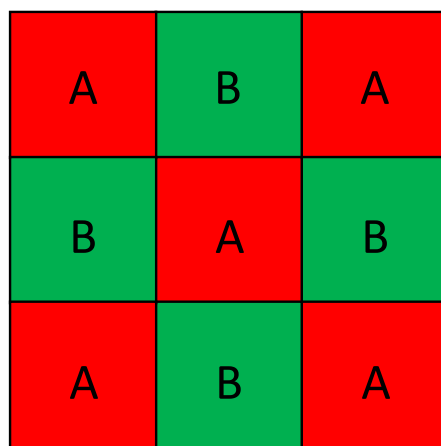
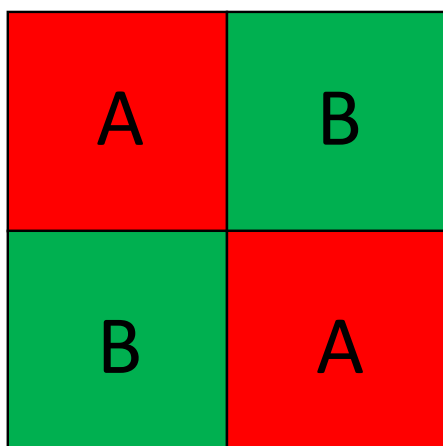
Celou oblast uvažovaného vzorku materiálu Ω lze rozdělit na jasně definovaný periodicky se opakující počet podoblastí resp. podmnožin podle daného typu mikrostruktury. Tyto podmnožiny označíme A a B (viz obrázky 3.1, 3.2 a 3.3). Obě tyto mikrostruktury jsou definované třemi volitelnými parametry:

- n
- Y_1
- Y_2

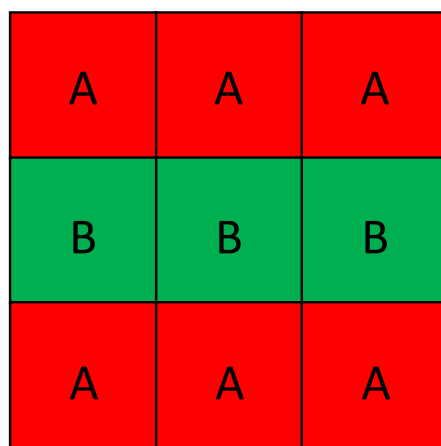
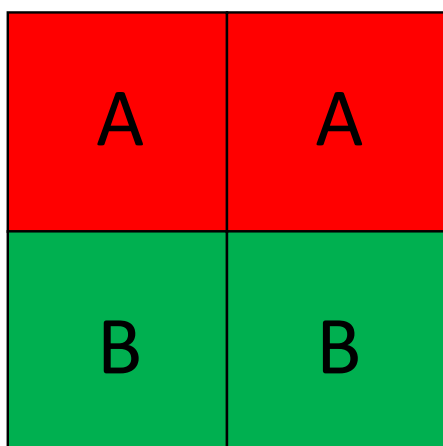
Číslo $n \in \mathbb{N}$, $n > 1$ definuje rozměr dané struktury, tj. počet podoblastí v jednom směru.

Čísla $Y_1, Y_2 \in \mathbb{R}$, $Y_1, Y_2 > 0$ jsou Youngovy moduly pružnosti. Podmnožině A odpovídá Y_1 a podmnožině B odpovídá Y_2 . V rámci projektu budeme předpokládat, že $Y_1 > Y_2$.

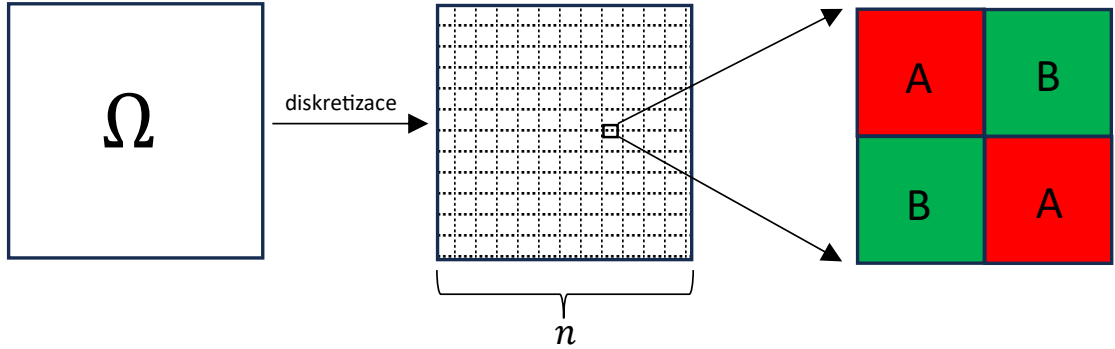
Vzhledem k povaze úloh, které tento projekt řeší, budeme u všech simulací počítat s hodnotou $\nu = 0$, protože jinak bychom dostali v efektivním elastickém tenzoru nenulové prvky v ose z , což je možné pro dvoudimenzionální problémy zanedbat.



Obrázek 3.1: Ukázka struktury šachovnice pro $n = 2$ (vlevo) a $n = 3$ (vpravo)



Obrázek 3.2: Ukázka struktury sendviče pro $n = 2$ (vlevo) a $n = 3$ (vpravo)



Obrázek 3.3: Vizualizace diskretizace oblasti Ω na n^2 podoblastí

3.2 Popis numerického výpočtu výsledků

V této kapitole nejprve uvedeme výpočet efektivního tenzoru $\tilde{\mathbf{C}}^*$ pro konkrétně zvolené matice zatížení. Dále zavedeme pojem „reprezentativní objemový vzorek” a popíšeme postup k nalezení jeho velikosti při zvolené toleranci.

3.2.1 Efektivní elastický tenzor pro konkrétní matice zatížení

V kapitole 2.2.3 byl odvozen postup výpočtu $\tilde{\mathbf{C}}^*$ pro tři obecné matice zatížení. Dále budeme uvažovat pouze následující tři lineárně nezávislé matice zatížení:

$$\mathbf{E}_1 = \begin{bmatrix} \beta & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{E}_2 = \begin{bmatrix} 0 & 0 \\ 0 & \beta \end{bmatrix}, \quad \mathbf{E}_3 = \begin{bmatrix} 0 & \beta \\ \beta & 0 \end{bmatrix}, \quad (3.1)$$

kde $\beta > 0$ je zátěžový parametr¹.

Tyto matice zatížení lze ekvivalentně vyjádřit pomocí Voigtovy notace (2.11):

$$\tilde{\mathbf{E}}_1 = \begin{bmatrix} \beta \\ 0 \\ 0 \end{bmatrix}, \quad \tilde{\mathbf{E}}_2 = \begin{bmatrix} 0 \\ \beta \\ 0 \end{bmatrix}, \quad \tilde{\mathbf{E}}_3 = \begin{bmatrix} 0 \\ 0 \\ 2\beta \end{bmatrix}. \quad (3.2)$$

¹Pro náš případ lineární elasticity stačí uvažovat např. hodnotu $\beta = 1$.

Abychom našli řešení $\tilde{\mathbf{C}}^*$ pro konkrétní matice definované v (3.2), využijeme rovnosti (2.21), kterou dosadíme do (2.22). Ve výsledku dostáváme následující:

$$\begin{aligned} C_{1111}^* &= \frac{\langle \sigma_{11} \rangle^1}{\beta}, & C_{2211}^* &= \frac{\langle \sigma_{22} \rangle^1}{\beta}, & C_{1211}^* &= \frac{\langle \sigma_{12} \rangle^1}{\beta}, \\ C_{1122}^* &= \frac{\langle \sigma_{11} \rangle^2}{\beta}, & C_{2222}^* &= \frac{\langle \sigma_{22} \rangle^2}{\beta}, & C_{1222}^* &= \frac{\langle \sigma_{12} \rangle^2}{\beta}, \\ C_{1112}^* &= \frac{\langle \sigma_{11} \rangle^3}{2\beta}, & C_{2212}^* &= \frac{\langle \sigma_{22} \rangle^3}{2\beta}, & C_{1212}^* &= \frac{\langle \sigma_{12} \rangle^3}{2\beta}. \end{aligned} \quad (3.3)$$

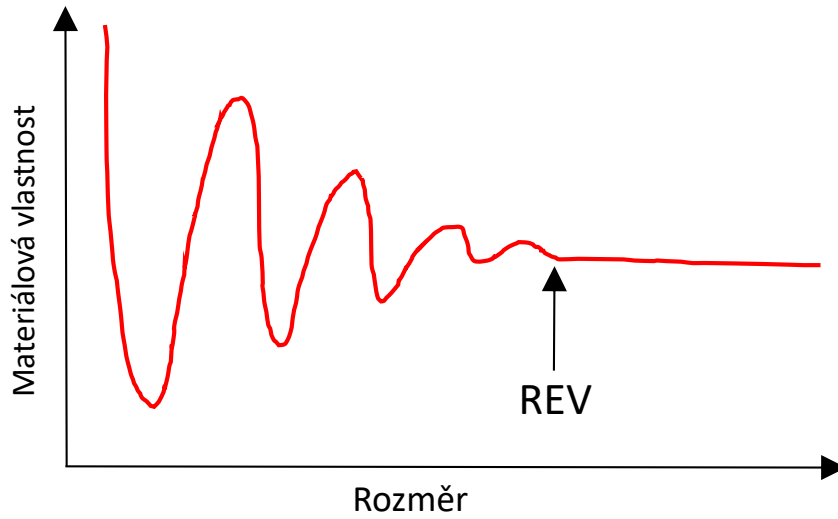
Pokud pevně stanovíme zátěžový parametr $\beta = 1$, každý dílčí homogenizovaný prvek tenzoru napětí je jednoduše roven korespondujícímu prvku efektivního elastického tenzoru (či jeho polovině).

3.2.2 Reprezentativní objemový vzorek a tolerance

Hodnoty v efektivním tenzoru $\tilde{\mathbf{C}}^*$ jsou ovlivněny velikostí uvažované mikrostruktury, tedy parametrem n . Pro dosažení hodnot blízkých nekonečně periodické struktury je třeba zvolit n dostatečně velké, tak aby zároveň zůstala úloha výpočetně přijatelně náročná. Pro tento účel se používá pojem **reprezentativní objemový vzorek** (zkráceně REV - representative elementary volume²). REV je definován jako **minimální objem**³ (nebo rozmezí), za nímž materiálové **charakteristiky zůstávají** v podstatě **konstantní** (Min a Jing, 2003). Jinak řečeno je to co nejmenší objem na dané oblasti, který v průměru může popsat zkoumané mechanické vlastnosti na celé oblasti.

²Používá se i RVE - representative volume element.

³V kontextu projektu je objem vnímán jako plocha.



Obrázek 3.4: Závislost vlastnosti na rozměru a REV (inspirace (Min a Jing, 2003))

Je zřejmé, že REV nemá jasně definovaný rozměr, protože se můžeme stále přibližovat k přesnému řešení, resp. zvětšovat REV, ale za cenu vyšší výpočetní náročnosti i delšího trvání výpočtu. Pro nalezení přibližné velikosti REV bylo zvoleno ukončovací kritérium založené na tzv. relativním residuu, které porovnává \tilde{C}^* pro dvě různé hodnoty n . Pokud toto residuum klesne pod zvolenou toleranci, je výpočet ukončen.

V celém projektu byla pro všechny výpočty uvažována tolerance o hodnotě 0,1%.

3.2.3 Výpočet relativního residua

Residuum je obecně označení pro zbytek či zůstatek, což v kontextu projektu značí zbytek rozdílu dvou po sobě jdoucích iterací. Nejprve je nutné předem definovat, jaké budeme srovnávat rozměry v těchto iteracích. V tomto projektu projektu bylo zvoleno, že se efektivní elastický tenzor získaný pro rozměr n bude srovnávat s efektivním elastickým tenzorem vypočteným pro rozměr $m = n + 2$. Pro přehlednost vypíšeme tyto tenzory v maticové formě:

$$\begin{aligned} \text{pro } n : \quad \tilde{C}_n^* &= \begin{bmatrix} C_{1111,n}^* & C_{2211,n}^* & C_{1211,n}^* \\ C_{1122,n}^* & C_{2222,n}^* & C_{1222,n}^* \\ C_{1112,n}^* & C_{2212,n}^* & C_{1212,n}^* \end{bmatrix}, \\ \text{pro } m : \quad \tilde{C}_m^* &= \begin{bmatrix} C_{1111,m}^* & C_{2211,m}^* & C_{1211,m}^* \\ C_{1122,m}^* & C_{2222,m}^* & C_{1222,m}^* \\ C_{1112,m}^* & C_{2212,m}^* & C_{1212,m}^* \end{bmatrix}. \end{aligned} \tag{3.4}$$

Nyní si definujeme tenzor $\mathbf{R} \in \mathbb{R}^{3 \times 3}$, který reprezentuje rozdíl mezi $\tilde{\mathbf{C}}_n^*$ a $\tilde{\mathbf{C}}_m^*$ tj.:

$$\mathbf{R} = \begin{bmatrix} r_{1111} & r_{2211} & r_{1211} \\ r_{1122} & r_{2222} & r_{1222} \\ r_{1112} & r_{2212} & r_{1212} \end{bmatrix}, \quad \mathbf{R} = \tilde{\mathbf{C}}_m^* - \tilde{\mathbf{C}}_n^*, \quad (3.5)$$

resp.

$$r_{ijkl} = C_{ijkl,m}^* - C_{ijkl,n}^*, \quad i, j, k, l = 1, 2. \quad (3.6)$$

V tuto chvíli máme vše pro definici relativního residua \hat{r} , která vypadá takto:

$$\hat{r} = \frac{\|\mathbf{R}\|}{\|\tilde{\mathbf{C}}_m^*\|}. \quad (3.7)$$

4 Popis, funkce a dokumentace programu

V této kapitole nejprve popíšeme předpoklady a kroky potřebné pro instalaci a použití implementovaného programu pro numerickou homogenizaci mechanických vlastností, a to včetně všech pomocných softwarů (odteď značeno zkráceně SW) a programů, které byly využity při řešení. Nakonec podrobněji rozebereme implementaci programu.

Zejména obě jeho možnosti použití, kterými jsou buď výpočet pro **jedno zvolené** n nebo pro **sekvenci** n , z níž se pak určí i velikost REV.

4.1 Předpoklady před použitím programu

Pro použití implementovaného programu k numerické homogenizaci je zapotřebí nejprve nainstalovat či stáhnout několik SW.

- Flow123d - výpočetní simulátor (Flow123d, [2024](#))
- GitHub repozitář - repozitář s kompletním kódem (Siddall, [2024](#))

Dále zde nebude uvedeno, jak se s zmíněnými programy pracuje resp. jak je používat. Případné návody by měly být k dohledání v jejich dokumentaci. Jak se tyto programy/SW použily k řešení, bude zmíněno až později.

4.1.1 Spouštění v OS Windows

Za předpokladu úspěšné instalace všech zmíněných SW a programů je implementovaný program téměř připravený k používání. Jediná komplikace činí nutnost při každém spuštění terminálu opět instalovat některé moduly resp. instalační balíčky v Pythonu (např. pyvista, sympy, ruamel.yaml). Z toho důvodu je nutné ze zmíněného GitHubu stáhnout i **Dockerfile**, ve kterém jsou právě tyto (či kterékoliv jiné) potřebné Python balíčky. Jakmile je Dockerfile vytvořený, je zapotřebí s ním vytvořit nový Docker image. Celý postup vytvoření nového Docker image je popsán níže:

- Vytvoření nového Docker image:
 - Otevřít příkazovou řádku ve složce, kde se nachází soubor **Dockerfile**

- Spustit následující příkaz pro vytvoření Docker image ("flow123d_siddall" je příklad názvu nového image):

```
docker build -t flow123d_siddall
```

Pokud jsou v Dockerfile sepsané všechny potřebné knihovny, stačí spouštět terminál Flow123d ze souboru `fterm_siddall.bat`.

4.2 Obecná struktura programu

Nejprve se budeme zabývat obecnou strukturou programu v tom nejširším měřítku tj. jaké platí vztahy mezi jednotlivými třídami¹, které v projektu jsou použité a jaké je celkové rozvržení těchto tříd v rámci programu jako takového. Poté se budeme zabývat trochu více dopodrobna každou třídou zvlášť tzn. zobrazit v grafech jaké jsou jejich vstupy, metody a výstupy. Vzhledem k velkému množství metod, které tyto jednotlivé třídy mají, tak se nebude do hloubky rozebírat každá dílčí metoda. Zkrátka cílem je popsat strukturu z nejširšího měřítku a potom rozebrat každou třídu zvlášť pro ucelený přehled funkcionality implementovaného programu na výpočet numerické homogenizace mechanických vlastností.

Program by obecně šlo rozložit na tyto části:

1. Uživatelský vstup:

- `config_file.yaml` (+ související `ConfigManager.py`)
- vstup z příkazové řádky pro `automatic_simulation.py`

2. Logika programu:

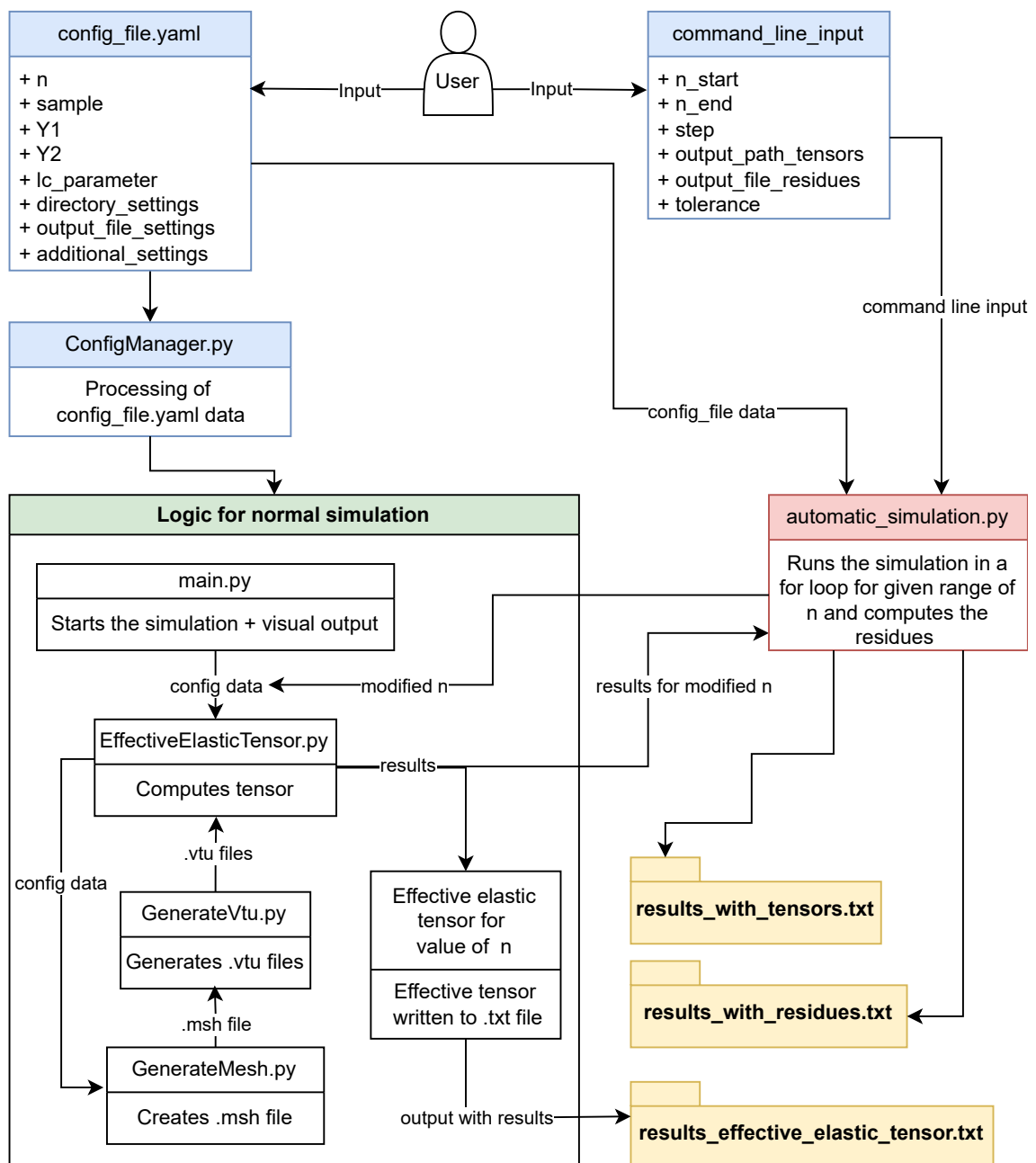
- `automatic_simulation.py`
- `main.py`
- `EffectiveElasticTensor.py`
- `GenerateVtu.py`
- `GenerateMesh.py`

3. Výstup:

- `results_effective_elastic_tensor.txt`
- `results_with_tensors.txt`
- `results_with_residues.txt`

Na následující straně je zobrazen class diagram ukazující vztahy mezi třídami.

¹Pro ujasnění, tak třídy jsou pojmenované s velkým počátečním písmenem, zatímco soubory (konfigurační i výsledné), spouštěcí skripty/programy začínají malým počátečním písmenem.



Obrázek 4.1: Class diagram celého programu

4.3 Popis jednotlivých tříd a souborů

4.3.1 config_file.yaml

V tomto souboru jsou definované veškeré uživatelem volitelné parametry či volby pro simulaci, kterými jsou:

1. `simulation_parameters`:

- `n` - definuje rozměr dané struktury
- `sample` - udává jestli se jedná o šachovnici či sendvič
- `Y1` - Youngův modul pružnosti pro podmnožinu A
- `Y2` - Youngův modul pružnosti pro podmnožinu B
- `lc_parameter` - parametr ovlivňující jemnost generovaných sítí

2. `directories`:

- `absolute_path_to_dir_with_skripts` - adresář, od kterého se odvíjí všechny ostatní cesty níže
- `dir_where_yamls_are_created` - relativní cesta k adresáři, kde vznikají .yaml soubory
- `path_to_yaml_template` - relativní cesta k předloze .yaml souboru, který se nesmí mazat a měnit
- `directory_where_vtus_are_created` - relativní cesta k .vtu souborům
- `dir_where_mesh_and_geo_are_created` - relativní cesta k adresáři, kde se vytváří .msh (případně .geo) soubory

3. `results_file_settings`

- `name_of_file_with_tensor` - název souboru s výsledky simulace
- `output_dir_of_file_with_tensor` - relativní cesta k souboru s výsledky

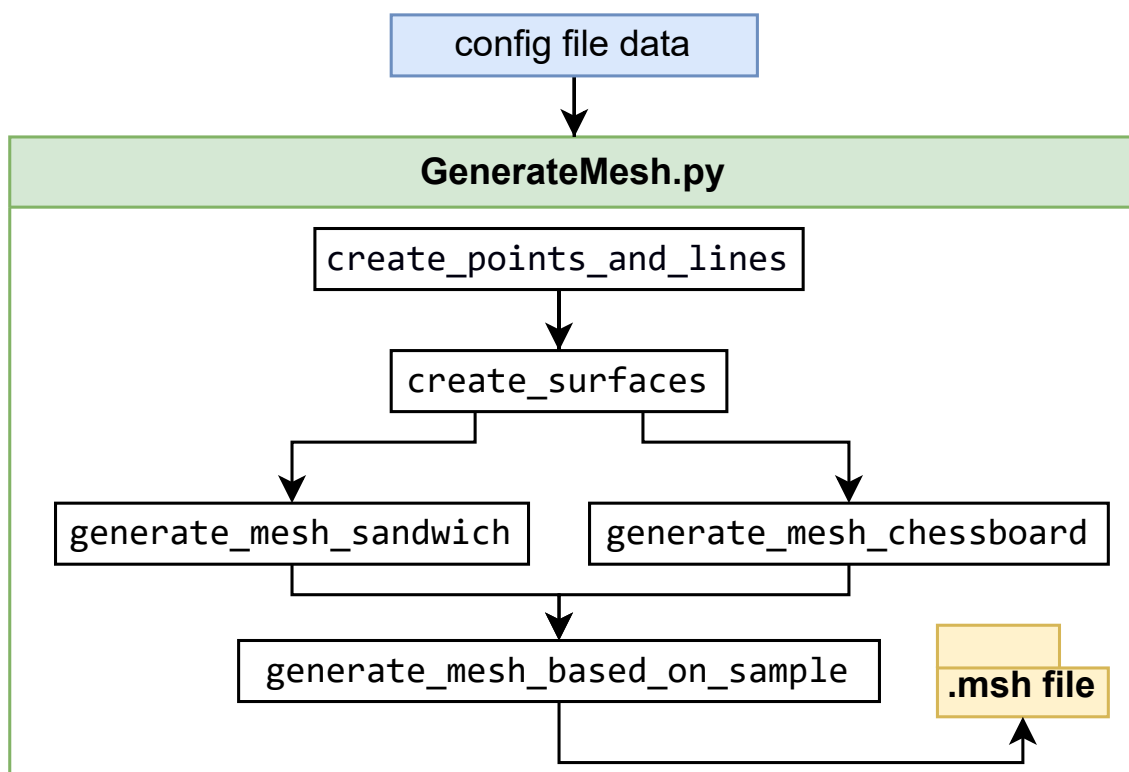
4. `additional_settings`

- `delete_yaml_dir_after_simulation` - přepínač zda smazat adresář s .yaml soubory
- `delete_vtu_dir_after_simulation` - přepínač zda smazat adresář s .vtu soubory
- `change_names_of_computed_yamls` - přepínač zda uživatel si chce zvolit názvy .yaml souborů
- `new_names_of_yamls` - pokud uživatel zvolil, že chce změnit názvy, zde si je definuje

- `change_names_of_computed_output_dirs` - přepínač zda uživatel si chce zvolit názvy adresářů s výslednými .vtu soubory
- `new_names_of_output_dirs` - pokud uživatel zvolil, že chce změnit názvy, zde si je definuje
- `change_name_of_msh_file` - přepínač pro volbu názvu .msh souboru
- `new_name_of_mesh` - případný nový název .msh souboru
- `create_geo_file` - přepínač zda vytvořit .geo soubor
- `change_name_of_geo_file` - přepínač zda pro volbu názvu .geo souboru
- `new_name_of_geo` - pokud uživatel zvolil, že chce změnit název, zde si ho definuje
- `measure_time_of_computation` - přepínač zda uživatel si chce měřit čas

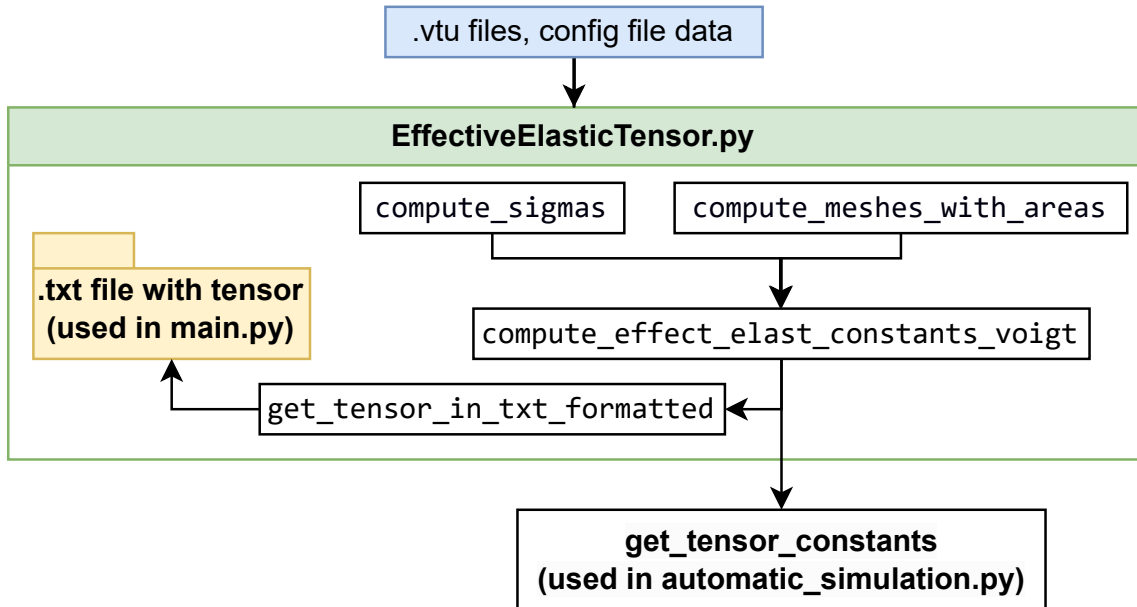
4.3.2 GenerateMesh.py

Tato třída na základě vstupních dat konfiguračního souboru vytváří automaticky geometrii, tj. body, hranice, plochy, podoblasti a v závěru také výpočetní síť ve formátu .msh, kterou následně vrací zpět.



Obrázek 4.2: Vizualizace fungování metod ve třídě GenerateMesh.py

uživatel provádí simulaci pomocí `main.py` nebo `automatic_simulation.py`. Pokud zvolí první variantu, tak se jednorázově zapíše výsledný tenzor do uživatelem zvoleného souboru, ale v druhém případě jsou postupně přidávány jednotlivé tenzory pro celý rozsah rozměrů, které uživatel zadal v příkazové řádce.



Obrázek 4.4: Vizualizace fungování metod ve třídě `EffectiveElasticTensor.py`

Kvůli důležitosti je potřeba se trochu více do detailu zabývat metodou `compute_effect_elast_constants_voigt`. Tato metoda je klíčová, neboť se v ní počítají všechny samotné efektivní elastické konstanty, které se následně převádí do Voigtovy notace.

Metoda začíná tím, že vytvoří prázdný list `constants_voigt_list`, do kterého jsou postupně ukládány vypočtené konstanty.

Poté se pomocí metod `compute_sigmas` a `compute_meshes_with_areas` získají napětí (sigmas) a plochy jednotlivých elementů (areas).

Následně se v cyklu násobí každá hodnota napětí plochou elementu a výsledky se uloží do listu `stress_area`. Jakmile je cyklus dokončen sečtou se všechny hodnoty v seznamu `stress_area` a tato suma se poté vydělí celkovou plochou oblasti resp. sumou všech dílčích elementů. Tyto získané hodnoty konstant se vyberou podle indexů `[0, 4, 1]`. Důvod pro tyto indexy je následující: V seznamu `all_constants` je 27 efektivních elastických konstant. Pro každou matici zatížení dostáváme 9 konstant, z nichž 5 je nulových. Zbývající 4 konstanty tvoří symetrickou matici. Nás tedy zajímají pouze unikátní prvky s indexy `[0, 4, 1]` pro danou matici zatížení, což jsou výsledky tvořící jeden ze sloupců výsledného efektivního elastického tenzoru. Konstanty se poté uloží se do výsledného seznamu `constants_voigt_list`, který je nakonec vrácen jako výstup metody.

Kód metody je k nahlédnutí v GitHub repozitáři, tj. (Siddall, 2024)) případně v náhledu níže:

```

def compute_effect_elast_constants_voigt(self):
    constants_voigt_list = []
    sigmas = self.compute_sigmas()
    areas = self.compute_meshes_with_areas()
    amount_of_meshes = len(self.meshes)

    for i in range(amount_of_meshes):
        sigma = sigmas[i]
        area = areas[i]
        stress_area = []
        for j in range(len(sigma)):
            stress_area.append(sigma[j, :] * area[j])

        all_constants = np.sum(stress_area, axis=0) / np.sum(area)
        needed_indexes = [0, 4, 1]
        constants_voigt = [all_constants[index] for index in needed_indexes]
        constants_voigt_list.append(constants_voigt)
    return constants_voigt_list

```

Obrázek 4.5: Detail metody výpočtu efektivních elastických konstant

Poslední, co je důležité rozebrat, co se funkcionalit programu týče, je rozdíl mezi spouštěcími skripty `main.py` a `automatic_simulation.py`, což bude rozebráno v následujících dvou kapitolách.

4.3.6 main.py

`Main.py` slouží k inicializaci **jednorázové** simulace právě pro **jednu sadu** dat. Těmito daty jsou výhradně data z konfiguračního souboru `config_file.yaml`. V tomto konfiguračním souboru si uživatel může před spuštěním simulaci navolit veškeré potřebné parametry a poté jednoduše spustit skript v terminálu Flow123d, a to následovně:

```
<python3> <main.py> <config_file.yaml>
```

Výstupem tohoto skriptu je pouze **spočtený efektivní tenzor** na základě vstupních dat, který je umístěn v textovém souboru, který je v příslušném adresáři, který si uživatel před simulací nastavil. Drtivá většina volitelných parametrů v `config_file.yaml` jsou určeny **výhradně pro změnu výstupu `main.py`**, protože při použití `automatic_simulation.py` nejsou žádné parametry z konfiguračního souboru používány. Jedinými výjimkami, které je nutné i nastavovat pro `automatic_simulation.py`, jsou `sample`, `Y1`, `Y2` a `lc_parameter`, ale o tom více v další části.

4.3.7 automatic_simulation.py

`Automatic_simulation.py` slouží k inicializaci **několika po sobě jdoucích automaticky spouštěných simulací**. Příkaz se spouští následovně:

```
<python3> <automatic_simulation.py> <config_file.yaml> <n_start> ...  
... <n_end> <step> <output_tensors> <output_residues> <tolerance>
```

Před spuštěním skriptu je ještě nutné si v `config_file.yaml` nastavit `sample`, `Y1`, `Y2` a `lc_parameter`, které budou pro všechny simulace použité. Jakmile je toto nastavené, tak je zapotřebí zadat zbývající parametry:

- `n_start` - udává počáteční n , na kterém se začíná
- `n_end` - udává konečné n , kde skript končí
- `step` - udává krok, se kterým se n zvyšuje
- `output_tensors` - absolutní cesta k textovému souboru s uloženými tenzory
- `output_residues` - absolutní cesta k textovému souboru s uloženými residui
- `tolerance` - tolerance při numerických výpočtech

Způsob fungování `automatic_simulation.py` spočívá v tom, že při každé iteraci se **vytváří nové dočasné konfigurační soubory s daným n** (vlastně se vytvoří dočasný `.yaml` konfigurační soubor pro každou hodnotu n ze zadaného rozsahu), které se následně použije pro novou simulaci. Poté se pro dvě po sobě jdoucí iterace resp. pro dva po sobě jdoucí tenzory spočte relativní residuum. Mimo jiné se průběžně tyto dočasné soubory mažou.

Jinými slovy `automatic_simulation.py` slouží k automatickému řízení několika po sobě jdoucích simulací s nějakým daným rozsahem rozměrů, krokem a tolerancí. Kontroluje se, zda relativní residua klesla pod zadanou toleranci, pokud ano, tak se vypíše, pro který rozměr resp. optimální n se konvergence dosáhlo (to stejné je poté i ve výsledném souboru) a proces se automaticky ukončí. Pokud se nepodařilo klesnout pod hranici tolerance, tak se ještě nedosáhlo dostatečné přesnosti výsledků a tím pádem REV.

Výsledné tenzory pro daný rozměr n jsou po jednotlivých sloupcích zapisovány do textového souboru. Soubory s relativními residui i výslednými tenzory jsou průběžně přepisované v průběhu běhu programu tzn. nepřijedeme o spočtené výsledky pokud dojde k manuálnímu či náhlému přerušení programu.

Lepší ukázka toho, jak vypadají nějaké konkrétní výpočty a výstupy (jak v terminálu, tak v textových souborech), pro zadaná data, je ukázán v následující kapitole, kde si ukážeme kompletní postup výpočtu pro dva specifické příklady.

5 Ukázkové výpočty vzorových úloh

Cílem této kapitoly je kompletně popsat a vizualizovat dva ukázkové příklady simulací. Nejprve si ukážeme, jak vypadá jeden ukázkový výpočet pro jednorázovou simulaci (`main.py`) a v druhé části kapitoly si ukážeme druhý příklad pro automatické simulování (`automatic_simulation.py`).

5.1 Ukázka jednorázové simulace

Obecně lze postup shrnout následovně; nejprve se vytvoří jednoduchá šachovnicová geometrie pro zadaný rozměr. Poté se na dané geometrii vytvoří síť. Proces pokračuje tím, že se vytvoří automaticky tři různé `.yaml` soubory pro každý typ zatížení s odpovídajícími parametry `Y1` a `Y2` pro tah v ose x . Následuje simulace s využitím `Flow123d`, který vrací tři různé `.vtu` soubory, vizualizace tahu v ose x , ze kterých je poté spočten výsledný efektivní elastický tenzor. V závěru je i ukázka výstupu `main.py`.

Úloha, která se zde vyřeší, má tato konkrétní data v konfiguračním souboru (adresáře vypsané v `directories` jsou konkrétní adresáře, ve kterých byl kód spouštěn):

1. `simulation_parameters`:

- `n`: 3
- `sample`: "chessboard"
- `Y1`: 100
- `Y2`: 50
- `lc_parameter`: 0.16

2. `directories`:

- `absolute_path_to_dir_with_skripts`:
"/C/Plocha/Semestrál_project/Python_skripts/"
- `dir_where_yamls_are_created`: "data_vtu/yamls_dir"
- `path_to_yaml_template`:
"data_vtu/TEMPLATE_DO_NOT_DELETE/template.yaml"
- `directory_where_vtus_are_created`: "data_vtu/vtu_files"
- `dir_where_mesh_and_geo_are_created`: "data_vtu/"

3. results_file_settings

- name_of_file_with_tensor: "example_tensor"¹
- output_dir_of_file_with_tensor - "results_elastic_tensor"

4. additional_settings

- delete_yaml_dir_after_simulation: "no"
- delete_vtu_dir_after_simulation: "no"
- change_names_of_computed_yamls: "no"
- new_names_of_yamls: ["name1.yaml", "name2.yaml", "name3.yaml"]
- change_names_of_computed_output_dirs: "no"
- new_names_of_output_dirs: ["name1", "name2", "name3"]
- change_name_of_msh_file: "yes"
- new_name_of_mesh: "example.msh"
- create_geo_file: "yes"
- change_name_of_geo_file: "yes"
- new_name_of_geo: "example.geo_unrolled"
- measure_time_of_computation: "yes"

Ještě pro úplnost uvedeme okrajové podmínky této úlohy. Pokud aplikujeme matice zatížení definované v (3.1), tak pro jednotlivé vektory posunu, viz (2.20), dostáváme:

$$\mathbf{u}_1 = \mathbf{E}_1 \cdot \mathbf{x} \iff \mathbf{u}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ 0 \end{bmatrix},$$

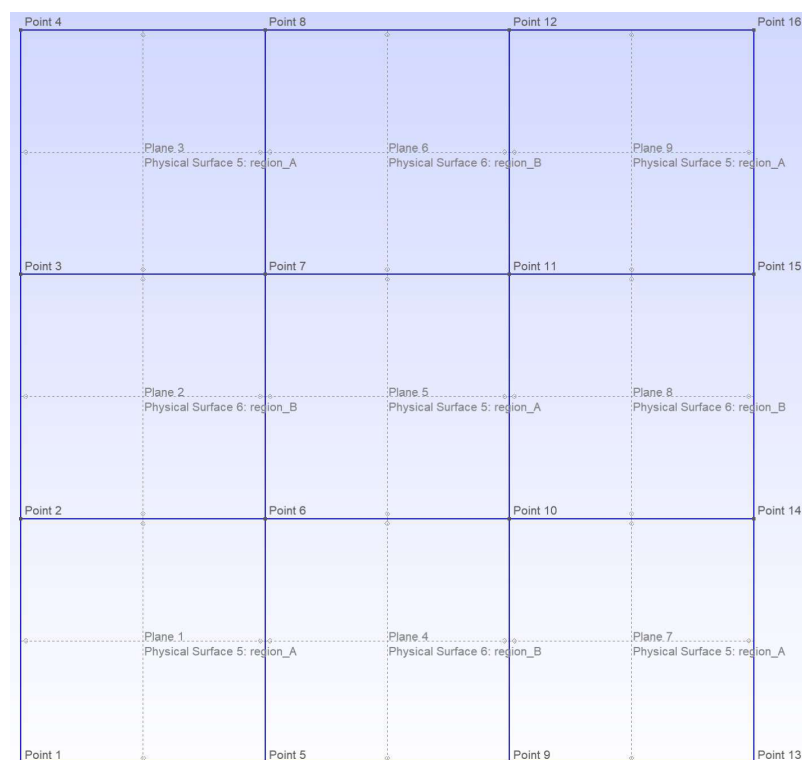
$$\mathbf{u}_2 = \mathbf{E}_2 \cdot \mathbf{x} \iff \mathbf{u}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ x_2 \end{bmatrix},$$

$$\mathbf{u}_3 = \mathbf{E}_3 \cdot \mathbf{x} \iff \mathbf{u}_3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_1 \end{bmatrix}.$$

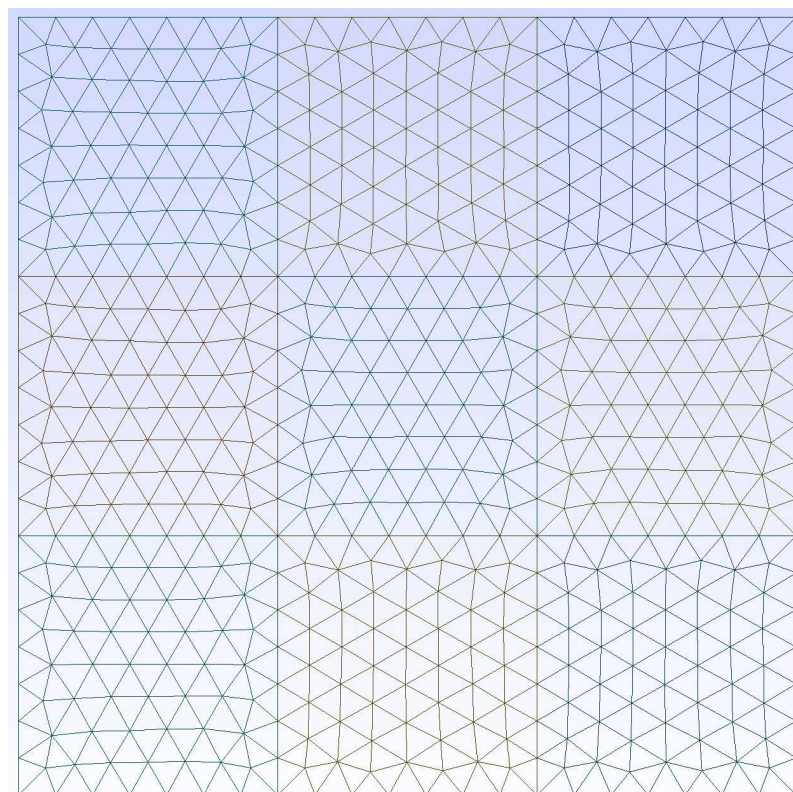
Slovně řečeno první zátěž představuje tah v ose x , druhá zátěž simuluje tah v ose y a třetí zátěž reprezentuje natažení ve smyku v osách x, y . Tyto okrajové podmínky resp. vektory posunutí \mathbf{u} se využijí k definici okrajových podmínek v jednotlivých .yaml souborech.

Analogický postup těchto výpočtů a celkového postupu by se aplikoval pro strukturu „sendvič“. Níže je vícero snímků názorně ukazující některé dílčí kroky.

¹V programu je implicitně zadán vytváření .txt souborů, není potřeba koncovku přidávat.



Obrázek 5.1: Vytvořená geometrie example.geo_unrolled pro ukázkou z kapitoly 5.1



Obrázek 5.2: Vygenerovaná výpočetní síť pro ukázkou z kapitoly 5.1

```

# Author: Ronald Ch. Siddall

flow123d_version: 3.1.0
problem: !Coupling_Sequential
  description: 2D linear elastic Dirichlet BC problem.
  mesh:
    mesh_file: /C/Plocha/Semestral_project/Python_skripts/data_vtu/example.msh
  flow_equation: !Coupling_Iterative
    input_fields:
      time:
        end_time: 3

    flow_equation: !Flow_Darcy_LMH
    input_fields:
    output:
      fields: []

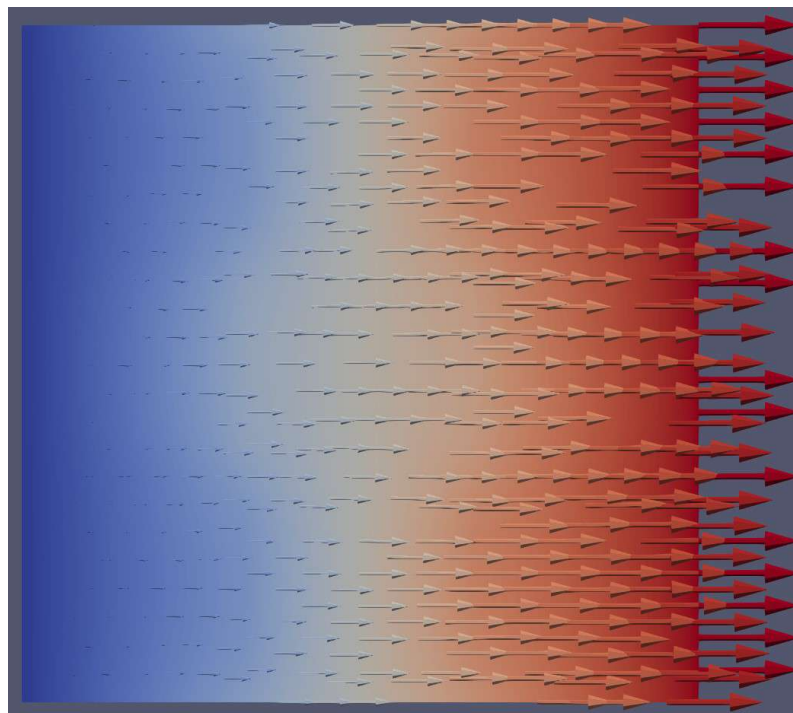
mechanics_equation:
  output_stream:
    file: mechanics.pvd
    format: !vtk
  output:
    fields:
      - displacement
      - stress
      - displacement_divergence
      - region_id
  solver: !Petsc
    a_tol: 1e-20
    r_tol: 1e-15
  input_fields:
    - region: region_A
      young_modulus: 100
      poisson_ratio: 0

    - region: region_B
      young_modulus: 50
      poisson_ratio: 0

    - region: .BOUNDARY
      bc_type: displacement
      bc_displacement: !FieldFormula
      value: [x,0,0]

```

Obrázek 5.3: Vytvořený .yaml soubor pro u_1 pro ukázkou z kapitoly 5.1



Obrázek 5.4: Vizualizace vektorů posunutí pro první zatížení v ParaView pro ukázkou z kapitoly 5.1

```

example_tensor.txt
Soubor  Upravit  Zobrazit

Settings of the simulation:
- n: 3
- type of geometry: chessboard
- Y1: 100
- Y2: 50

=====
Effective elastic tensor in matrix form for 2D problems
=====

C =      73.81537919579347      0.7197680236474644      -0.00012916870045901304
      0.7197680236473514      73.81396535529225      1.9395047606476615e-05
      -0.00012916870025809276      1.9395047286411653e-05      37.093813130071304

-----
This result was computed using these files:
/C/Plocha/Semestral_project/Python_skripts/data_vtu/vtu_files/1_mechanics-00000.vtu
/C/Plocha/Semestral_project/Python_skripts/data_vtu/vtu_files/2_mechanics-00000.vtu
/C/Plocha/Semestral_project/Python_skripts/data_vtu/vtu_files/3_mechanics-00000.vtu

```

Obrázek 5.5: Zobrazení spočteného efektivního elastického tenzoru pro ukázkou z kapitoly 5.1

5.2 Ukázka automatické simulace

Ukázková automatická simulace byla provedena se stejným nastavením konfiguračního souboru, jako v předešlé kapitole a pro následující vstupní data:

- `n_start`: 3
- `n_end`: 7
- `step`: 2
- `output_tensors`: `"/C/Plocha/results_tensors.txt"`
- `output_residues`: `"/C/Plocha/results_residues.txt"`
- `tolerance`: 0.001

```
=====
                        AUTOMATIC SIMULATION STARTED
=====
Settings of the simulation:
- Type of geometry: chessboard
- Y1: 100
- Y2: 50
- n: 3 to 7
- step: 2
- tolerance: 0.001
- output with tensors: /C/Plocha/results_tensors.txt
- output with residues: /C/Plocha/results_residues.txt
-----
Simulation started:2024-05-07 17:44:34
Progress status of the simulation:
-----
2024-05-07 17:44:44: (1/2) - 50.00% completed
2024-05-07 17:44:58: (2/2) - 100.00% completed
-----
Optimal n not found within the given tolerance
Tolerance: 0.001
=====
```

Obrázek 5.6: Ukázka výstupu `automatic_simulation.py` pro ukázkovou úlohu z kapitoly 5.2

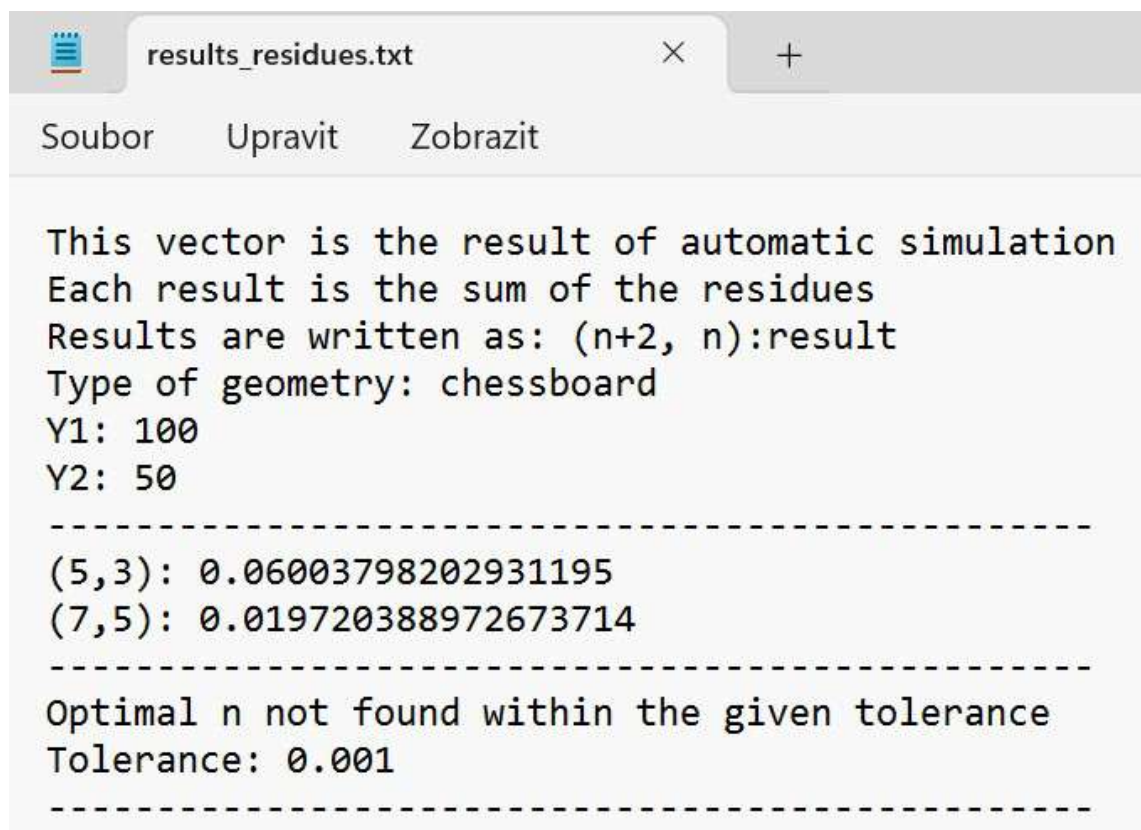
Jak lze z obrázku 5.6, vidět výstup nejdříve vypíše nastavení dané simulace a cesty k souborům s výsledky. Dále průběžně vypisuje informaci o tom, jak samotná simulace probíhá tj. vypisuje se čas, kdy každá dílčí simulace/iterace proběhla. V tomto ukázkovém případě se nedosáhlo konvergence, neboť zadané rozměry jsou příliš malé.

Pro doplnění v následujícím obrázku je zobrazen příklad výstupu `automatic_simulation.py`, pokud došlo ke konvergenci v zadaném rozsahu rozměrů.

```
-----  
Optimal n: 21  
Result for optimal n: 0.00093871256304287  
Tolerance: 0.001  
-----
```

Obrázek 5.7: Ukázka poslední části výstupu pokud se dosáhlo konvergence

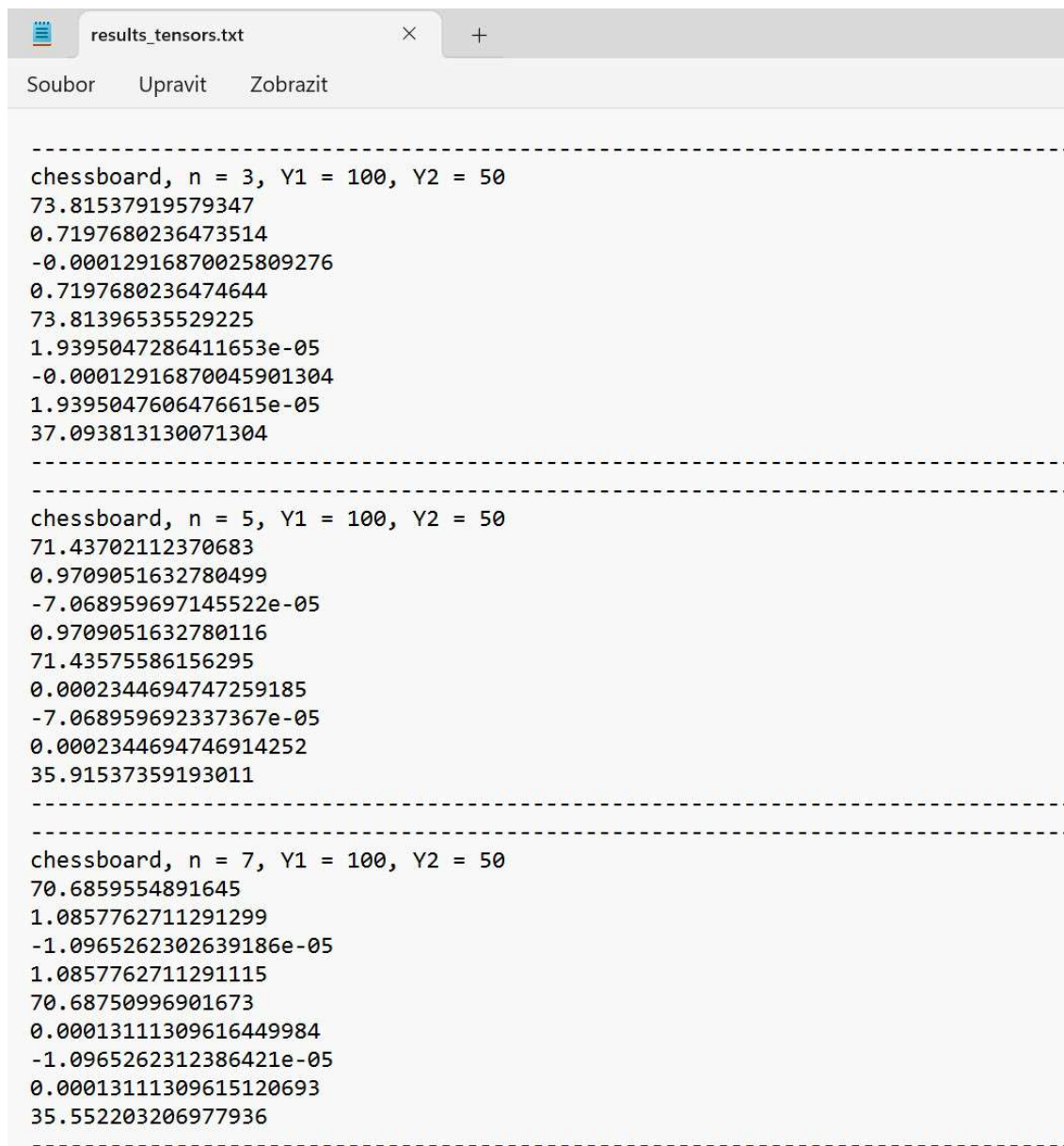
Na obrázku 5.8 je detail souboru obsahující relativní residua.



```
results_residues.txt  
Soubor  Upravit  Zobrazit  
  
This vector is the result of automatic simulation  
Each result is the sum of the residues  
Results are written as: (n+2, n):result  
Type of geometry: chessboard  
Y1: 100  
Y2: 50  
  
-----  
(5,3): 0.06003798202931195  
(7,5): 0.019720388972673714  
-----  
  
Optimal n not found within the given tolerance  
Tolerance: 0.001  
-----
```

Obrázek 5.8: Ukázka souboru s relativními residui ze zadaného rozsahu pro ukázkovou úlohu z kapitoly 5.2

Na obrázku 5.9 je zobrazen soubor s výsledky automatické simulace. Všimněme shodného výsledku pro $n = 3$. Řešení je stejné jako u ukázkové úlohy v kap. 5.1.



```
-----
chessboard, n = 3, Y1 = 100, Y2 = 50
73.81537919579347
0.7197680236473514
-0.00012916870025809276
0.7197680236474644
73.81396535529225
1.9395047286411653e-05
-0.00012916870045901304
1.9395047606476615e-05
37.093813130071304
-----
chessboard, n = 5, Y1 = 100, Y2 = 50
71.43702112370683
0.9709051632780499
-7.068959697145522e-05
0.9709051632780116
71.43575586156295
0.0002344694747259185
-7.068959692337367e-05
0.0002344694746914252
35.91537359193011
-----
chessboard, n = 7, Y1 = 100, Y2 = 50
70.6859554891645
1.0857762711291299
-1.0965262302639186e-05
1.0857762711291115
70.68750996901673
0.00013111309616449984
-1.0965262312386421e-05
0.00013111309615120693
35.552203206977936
-----
```

Obrázek 5.9: Ukázka výsledného souboru s tenzory ze zadaného rozsahu pro ukázkovou úlohu z kapitoly [5.2](#)

6 Výsledky provedených výpočtů

V této kapitole si rozebereme výsledky všech provedených výpočtů. Začneme vymezením toho, jaké případy byly spočítány a jaké byly jejich parametry. Dále si vypíšeme spočtené výsledky každého výpočtu, kterými se rozumí hodnota rozměru, kdy došlo ke konvergenci a s tím související tenzor, který pro daný rozměr vyšel. Poté si ukážeme různé závislosti v grafech, který jsou podložený naměřenými daty. V závěru této kapitoly je shrnutí všech získaných poznatků a pozorování, jak se ve výsledku jednotlivé typy a velikosti geometrií chovají pro různé hodnoty Youngových modulů pružnosti.

6.1 Rozbor všech provedených výpočtů

Během provádění jednotlivých měření bylo zjištěno, že je dost značný rozdíl v konvergenci mezi sudými a lichými násobky n . Totiž obecně platilo, že relativní residua pro liché násobky n vždy vycházela větší v porovnání se sudými násobky n , což je způsobeno odlišným rozložením podoblastí A, B pro liché a sudé násobky n . Tímto rozdělením se zajistil monotónní pokles relativních residuí zvlášť pro sudé a zvlášť pro liché násobky n .

Bylo analyzováno celkem **16 různých případů**, kterými jsou:

a) Šachovnice:

- 1) pro sudé násobky n , $Y1 = 100$, $Y2 = 50$
- 2) pro liché násobky n , $Y1 = 100$, $Y2 = 50$
- 3) pro sudé násobky n , $Y1 = 200$, $Y2 = 50$
- 4) pro liché násobky n , $Y1 = 200$, $Y2 = 50$
- 5) pro sudé násobky n , $Y1 = 500$, $Y2 = 50$
- 6) pro liché násobky n , $Y1 = 500$, $Y2 = 50$
- 7) pro sudé násobky n , $Y1 = 2500$, $Y2 = 50$
- 8) pro liché násobky n , $Y1 = 2500$, $Y2 = 50$

b) Sendvič:

- 9) pro sudé násobky n , $Y1 = 100$, $Y2 = 50$
- 10) pro liché násobky n , $Y1 = 100$, $Y2 = 50$
- 11) pro sudé násobky n , $Y1 = 200$, $Y2 = 50$
- 12) pro liché násobky n , $Y1 = 200$, $Y2 = 50$
- 13) pro sudé násobky n , $Y1 = 500$, $Y2 = 50$
- 14) pro liché násobky n , $Y1 = 500$, $Y2 = 50$
- 15) pro sudé násobky n , $Y1 = 2500$, $Y2 = 50$
- 16) pro liché násobky n , $Y1 = 2500$, $Y2 = 50$

Po zbytek práce se bude odkazovat na jednotlivé případy buď podle těchto číselných označení (1-16) případně zkráceně podle poměru $Y1$ vůči $Y2$, tj. poměr.

6.2 Finální výsledky provedených výpočtů

Zde si pouze jednoduše vypíšeme výsledky pro jednotlivé případy, kterými jsou spočtené tenzory, jejichž změna relativního residua je pro zmíněný rozměr menší než tolerance 0,1%. V následujících kapitolách jsou vypsané jednotlivé efektivní elastické tenzory pro všech 16 případů, jejichž prvky jsou od shora vypsané v následujícím pořadí: C_{1111}^* , C_{1122}^* , C_{1112}^* , C_{2211}^* , C_{2222}^* , C_{2212}^* , C_{1211}^* , C_{1222}^* , C_{1212}^* . V závěru jsou i dvě tabulky shrnující rozměry, kdy došlo ke konvergenci pro všechny případy.

6.2.1 Šachovnice, $Y1 = 100$, $Y2 = 50$

1) REV pro $n = 18$

69.77790412852876
1.268966397406456
0.0015400102086822543
1.2689663974064747
69.77883933121191
0.0016499129583467062
0.0015400102086773626
0.0016499129583499147
35.123542320394286

2) REV pro $n = 21$

69.7974325270019
1.2851838850191901
-3.375389169509585e-05
1.285183885019233
69.79814363461671
2.3699074542096177e-05
-3.3753891691373876e-05
2.3699074524563276e-05
35.13635599256736

6.2.2 Šachovnice, $Y1 = 200$, $Y2 = 50$

3) REV pro $n = 32$

95.4942863782584
6.656104637850853
0.005186057016212931
6.656104637850688
95.49868643195286
0.005215034742229218
0.005186057016260705
0.0052150347422039904
49.206199367922125

4) REV pro $n = 35$

95.49456610791185
6.679170336660807
9.452142842176485e-05
6.6791703366609285
95.49745108473925
-4.876211455186902e-05
9.452142840686483e-05
-4.876211463879784e-05
49.20988234501693

6.2.3 Šachovnice, $Y1 = 500$, $Y2 = 50$

5) REV pro $n = 46$

149.58438163454568
24.686233344953514
0.014817396738947754
24.686233344953838
149.59112513217812
0.014350614316744005
0.014817396738886173
0.014350614316633963
81.55103665042539

6) REV pro $n = 47$

149.64439418609484
24.695220464874005
-0.00013076264218503112
24.695220464873877
149.64674457689802
-0.0001013015961663917
-0.00013076264236725653
-0.00010130159631682866
81.58496382152258

6.2.4 Šachovnice, $Y1 = 2500$, $Y2 = 50$

7) REV pro $n = 56$

448.7538921227705
140.7095943919707
0.07791784541829024
140.70959439197912
448.7620760445914
0.07456287673995386
0.07791784541861534
0.07456287674009215
271.843579476085

8) REV pro $n = 59$

448.7339575220418
140.79816572557928
0.0011317897175975495
140.79816572558238
448.7337631201216
-0.0021087192578492474
0.001131789718711193
-0.002108719257941151
271.8367569322998

6.2.5 Sendvič, $Y1 = 100$, $Y2 = 50$

9) REV pro $n = 14$

74.99847398140356
-1.8652219694723548e-06
9.379221466503778e-09
-1.8652219623201964e-06
66.95726604003637
6.284790911036168e-06
9.37921262326113e-09
6.284790825319649e-06
33.607666763522374

10) REV pro $n = 35$

75.71366861854767
-2.12962485752517e-06
-9.057722796937459e-10
-2.129624893284967e-06
67.41017637236148
-2.0214943893211664e-06
-9.057388262221473e-10
-2.0214943622162866e-06
33.75199014276259

6.2.6 Sendvič, $Y1 = 200$, $Y2 = 50$

11) REV pro $n = 24$

124.9985145125482
-3.4044149069349482e-06
-6.816667895814417e-09
-3.4044148744409204e-06
80.81139931837016
-3.3183225646630386e-05
-6.816549437608952e-09
-3.318322585535557e-05
40.72025010831128

12) REV pro $n = 49$

126.52987545600318
-3.371304213952673e-06
5.8145736437936237e-11
-3.3713043170585957e-06
81.3700253890178
-3.1376314452288377e-06
5.835365329858265e-11
-3.137631512646047e-06
40.83068228679981

6.2.7 Sendvič, $Y1 = 500$, $Y2 = 50$

13) REV pro $n = 32$

274.9975478691609
-7.607953550492294e-06
4.196254145044891e-10
-7.607953767531503e-06
93.18302310854162
-5.9475703427960495e-05
4.198236537246162e-10
-5.947570449568044e-05
47.14168882592856

14) REV pro $n = 57$

278.9459720317214
-6.764279305520123e-06
1.3702033346499766e-09
-6.764279319092109e-06
93.50080575801393
-9.295384282596763e-06
1.3694921969209712e-09
-9.295384394406406e-06
47.04557919993169

6.2.8 Sendvič, $Y1 = 2500$, $Y2 = 50$

15) REV pro $n = 34$

1274.9892989500217
 -4.201983699667877e-05
 -1.2066016661293384e-08
 -4.201983625644647e-05
 110.91815212605339
 0.0001425361974907446
 -1.206623923450982e-08
 0.00014253619142419577
 55.202924641297024

16) REV pro $n = 57$

1296.4847385718072
 -3.2121213682415905e-05
 8.458767170776745e-09
 -3.212121229496166e-05
 107.53471487023648
 -3.9190121007972586e-05
 8.456839269883612e-09
 -3.919012748561843e-05
 53.56243516868323

6.2.9 Tabulky s přehledem hodnot rozměrů REV

Pro přehlednost si zapíšeme hodnoty rozměrů REV pro šachovnici i sendvič do tabulek.

Poměr	Sudé n	Liché n
2	18	21
4	32	35
10	46	47
50	56	59

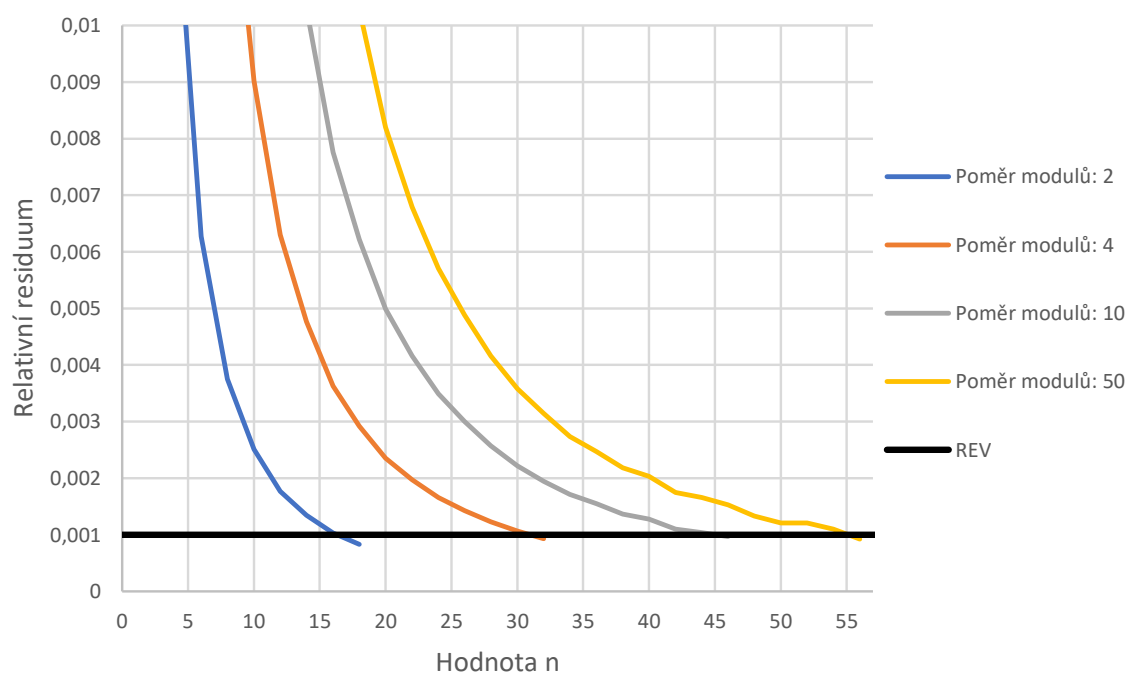
Tabulka 6.1: Shrnutí rozměrů REV pro šachovnici

Poměr	Sudé n	Liché n
2	14	35
4	24	49
10	32	57
50	34	57

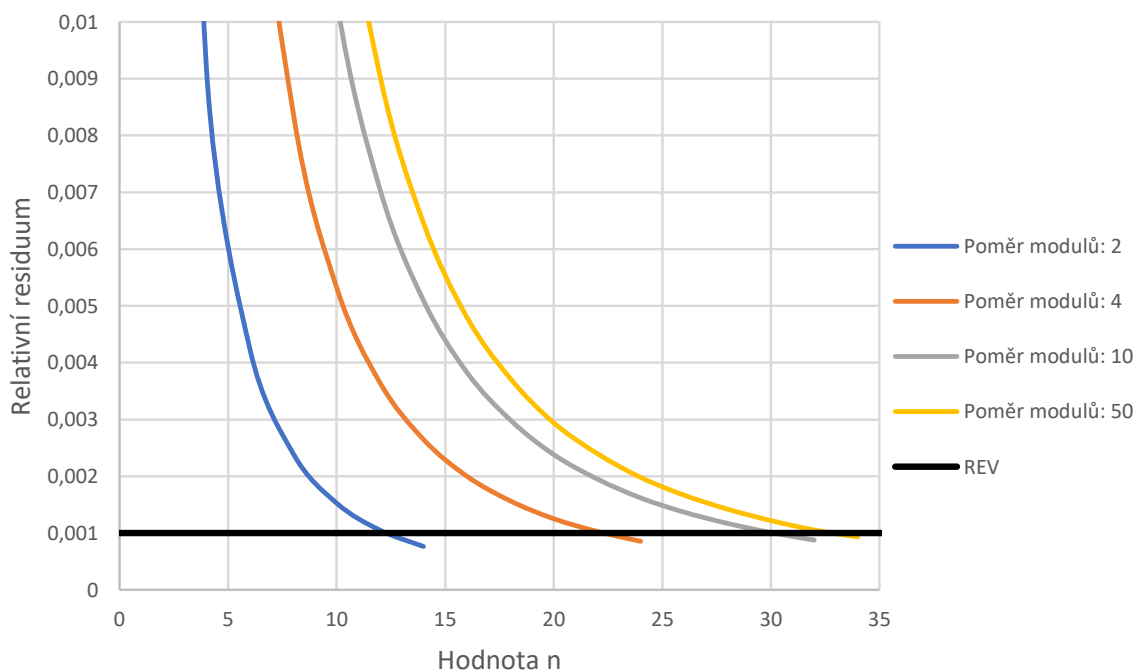
Tabulka 6.2: Shrnutí rozměrů REV pro sendvič

6.3 Grafy různých závislostí

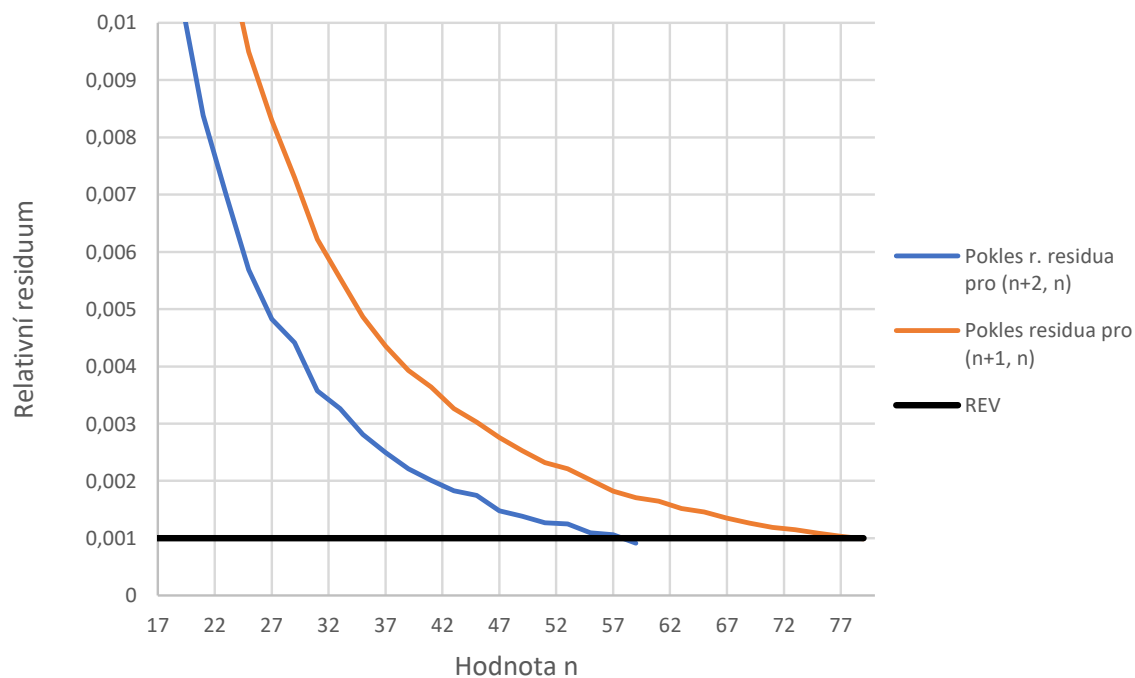
Obrázky 6.1 a 6.2 ukazují závislost relativního residua na rozměru, tzn. je zde zobrazen pokles relativního residua s zvětšujícími se hodnotami n . Tato závislost je ukázána pouze pro sudé násobky n , ale v podstatě ekvivalentní výsledky vychází i pro liché násobky, takže není potřeba přidávat obojí.



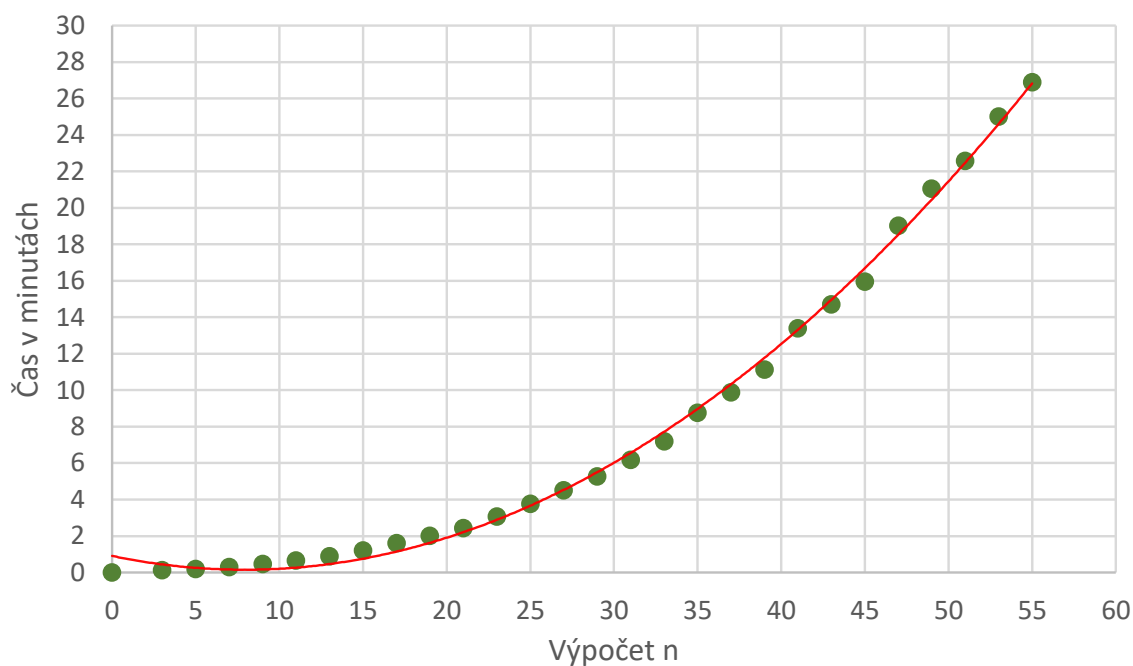
Obrázek 6.1: Šachovnice: závislost rel. residua na hodnotě rozměru REV pro sudé n



Obrázek 6.2: Sendvič: závislost rel. residua na hodnotě rozměru REV pro sudé n



Obrázek 6.3: Porovnání rychlosti poklesu rel. residua vůči hodnotě n při porovnávání (n+2, n) a (n+1, n) pro šachovnici - liché násobky - poměr modulů 50



Obrázek 6.4: Graf závislosti času potřebného ke konvergenci na rozměru - pro šachovnici - poměr modulů 50 - červená tečkovaná křivka je kvadratický polynom

6.4 Validace výsledků

V této části provedeme „validaci“ výsledků, tzn. ověření zda naměřené výsledky odpovídají teoretickému očekávání.

6.4.1 Výsledky pro sudé a liché násobky n

Teoreticky by nemělo záležet na tom, jakým způsobem se k REV blížíme. Proto je nutné ověřit, že pro stejné hodnoty Youngových modulů vychází výsledný tenzor téměř shodný, nehledě na tom, zda se k REV blížíme přes sudé či liché násobky. Pro potvrzení shodnosti výsledků jsou v následující tabulce sepsaná relativní residua mezi tensory vycházející pro sudé a liché násobky n v procentech pro všechny poměry:

Tabulka 6.3: Relativní residua mezi tensory pro sudé a liché násobky n

Poměr	Šachovnice	Sendvič
2	0,071062457%	1,093277233%
4	0,040265626%	1,340800827%
10	0,07103133%	1,431932456%
50	0,05272631%	1,910778411%

V tabulce 6.3 je viditelný rozdíl relativních residuí mezi strukturami. Tento rozdíl je způsobený hodnotami n , pro které došlo ke konvergenci. U šachovnice došlo ke konvergenci pro velmi podobné rozměry, viz tabulka 6.1, ale u sendviče je rozdíl v rozměrech mnohem větší, viz tabulka 6.2 a rozdíl se zvyšuje s vyšším poměrem Youngových modulů.

6.4.2 Analýza spočtených výsledků pomocí pravidla směsí

Pro sendvič odpovídají získané prvky efektivního tenzoru harmonickému průměru Youngových modulů. To je v souladu s empirickým očekáváním, kterým je tzv. **inverzní pravidlo směsí**, jenž se u kompozitů používá k předpovědi různých vlastností. Tímto pravidlem se zde nebudeme podrobněji zabývat, pro více informací viz (Roylance, 2000), případně (Wikipedia contributors, 2024).

Pro šachovnici jsou hodnoty efektivního tenzoru v rozmezí Y_1 a Y_2 , ale spíše se blíží k hodnotě Y_2 , zejména při velkém podílu Y_1 a Y_2 . To je nejspíš způsobené tím, že jednotlivá šachovnicová pole s velkou tuhostí jsou propojena jen v rozích.

6.5 Sumarizace výsledků a pozorování

Z vypočtených výsledků, grafů a pozorování jsou patrné určité trendy či závislosti, které jsou sumarizované v následujícím souhrnu:

Souhrn zjištění a pozorování na základě spočtených výsledků
Vyšší poměr Youngových modulů \rightarrow Větší hodnota rozměru REV
Vyšší poměr Youngových modulů \rightarrow Vyšší hodnoty efektivních koeficientů
Liché násobky n vždy konvergují pomaleji
U šachovnice platí, že první a druhý prvek na hlavní diagonále jsou téměř stejné, zatímco třetí prvek je oproti nim poloviční
U sendviče platí, že prvky na hlavní diagonále postupně klesají
Hodnoty efektivních elastických koeficientů jsou vesměs podobné pro sudé a liché násobky n (maximální odchylka 1,9%)
Relativní residua tenzorů pro sudé a liché násobky n u šachovnice jsou řádově menší než u sendviče
Relativní residua tenzorů pro sudé a liché násobky n u sendviče rostou s větším poměrem Youngových modulů
Efektivní elastické koeficienty jsou vyšší u sendviče než u šachovnice
Získané prvky pro šachovnici i sendvič leží v rozmezí hodnot Youngových modulů
U sendviče odpovídají získané prvky harmonickému průměru
U šachovnice se získané prvky spíše blíží nižší hodnotě Youngova modulu
Pokles residuí se dost liší při porovnávání $(n+2) - (n)$ oproti $(n+1) - (n)$
Čas pro každý výpočet roste přibližně kvadraticky s n

7 Závěr

Cílem této práce bylo implementovat efektivní výpočetní nástroj, umožňující získat mechanické vlastnosti složitě popsatelných mikroheterogenních materiálů pomocí numerické homogenizace.

Nejprve jsme teoreticky rozebrali problém numerické homogenizace, včetně základních fyzikálních zákonů, matematických rovnic a okrajových podmínek nezbytných pro pochopení této problematiky. Dále jsme provedli analýzu různých mikrostruktur a typů materiálů, na které se tento problém vztahuje.

Teoretická část a analýza mikrostruktur poskytly nezbytné poznatky pro návrh a implementaci výpočetního algoritmu řešícího tuto problematiku. Podrobný popis tohoto programu, od instalace po jeho použití k výpočtům požadovaných tenzorů, doplňuje tuto práci.

V poslední části jsme zhodnotili dosažené výsledky a jejich interpretaci. Výsledky odpovídaly empirickému očekávání, což potvrzuje správnost použitých postupů.

Závěrem lze tedy konstatovat, že numerická homogenizace poskytuje efektivní způsob, jak získat mechanické vlastnosti materiálů, což je klíčové pro hlubší porozumění a další výzkum nejen v technických, ale i v dalších odvětvích.

Motivací pro tuto práci bylo pochopení problematiky mikroheterogenních materiálů, jejich charakteristik a získání jejich vlastností ve dvou dimenzích. Jako přirozený následující krok by mohlo být rozšíření této práce do tří dimenzí či případná analýza složitějších mikrostruktur třeba těch obsahujících náhodné pukliny.

Bibliografie

- BRDIČKA, Miroslav, Ladislav SAMEK a Bruno SOPKO, 2005. *Mechanika kontinua*. Academia.
- CARNEIRO, Vitor Hugo, José MEIRELES a Hélder PUGA, 2013. Auxetic materials—A review. *Materials Science-Poland*. Roč. 31, s. 561–571.
- FLOW123D, 2024. *Flow123d*. Dostupné také z: <https://flow123d.github.io/>. [Online; navštíveno 5. 05. 2024].
- MIN, Ki-Bok a Lanru JING, 2003. *Numerical determination of the equivalent elastic compliance tensor for fractured rock masses using the distinct element method*. Sv. 40. Č. 6. ISSN 1365-1609. Dostupné z DOI: [https://doi.org/10.1016/S1365-1609\(03\)00038-8](https://doi.org/10.1016/S1365-1609(03)00038-8).
- POVEY, Rhys G, 2023. Voigt transforms.
- ROYLANCE, David, 2000. *INTRODUCTION TO COMPOSITE MATERIALS*. Dostupné také z: https://ocw.mit.edu/courses/3-91-mechanical-behavior-of-plastics-spring-2007/a4d6750d1f2a3f8834338507b209657b_03_composites.pdf. [Online; navštíveno 19. 05. 2024].
- SCHAFER, M., 2006. *Computational Engineering - Introduction to Numerical Methods*. Springer.
- SIDDALL, R., 2024. *Odkaz na GitHub repozitář s kódem*. Dostupné také z: <https://github.com/RonaldSiddall>. [Online; navštíveno 5. 05. 2024].
- WIKIPEDIA CONTRIBUTORS, 2024. *Rule of mixtures — Wikipedia, The Free Encyclopedia*. Dostupné také z: https://en.wikipedia.org/w/index.php?title=Rule_of_mixtures&oldid=1224132421. [Online; navštíveno 19.05.2024].
- ZOHDI, T. I. a P. WRIGGERS, 2005. *An Introduction to Computational Micromechanics*. Springer.