

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra matematiky

DIPLOMOVÁ PRÁCE

Řešení kontaktu pružných těles ve 2D
metodou konečných prvků
Solving of contact problem of elastic bodies in 2D
using by finite element method

PLZEŇ, 2014

Zdeňka Baxová

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem diplomovou práci vypracovala samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

PLZEŇ, 2014

Zdeňka Baxová

Poděkování

Ráda bych poděkovala Doc. Ing. Josefu Daňkovi, Ph.D. za odborné vedení mojí práce a rady během jejího zpracovávání.

Abstrakt

Práce se zabývá kontaktním problémem pružných těles, řešeném pomocí metody konečných prvků (MKP). Popisuje matematický základ problému, metodu konečných prvků a její aplikaci na problém a následnou implementaci v programovém prostředí MATLAB. Práce se zaměřuje na vhodnou realizaci úlohy ve formě skriptu a ověření jeho funkčnosti. Na závěr jsou uvedeny výsledky získané ze skriptu a také jeho možné zkvalitnění a další úpravy vedoucí k zlepšení chování metody.

Klíčová slova : pružnost, metoda konečných prvků, MKP, Galerkinova metoda, MATLAB

Abstract

This thesis deals with contact problem of elastic bodies, two or more, solved by finite element method (FEM). It describes mathematical principles of contact problem, finite element method algorithm and its implementation in MathWorks Matlab. The paper focuses on creating the application, which would be able to solve the problem efficiently and correctly. In conclusion paper presents outcomes of several tests and contains some improvements as a suggestion for the future research.

Keywords : elasticity, finite element method, FEM, Galerkin method, MATLAB

Obsah

Úvod	6
1 Formulace problému pružnosti	8
1.1 Klasická formulace	8
1.2 Slabá formulace	9
1.3 Kontaktní problém	10
2 Metoda konečných prvků	11
2.1 Galerkinova metoda	11
2.2 Metoda konečných prvků	12
2.2.1 Diskretizace oblasti	12
2.2.2 Bázové funkce	13
2.2.3 Sestavení matic	14
2.3 Výsledná soustava	16
3 Realizace	17
3.1 Příprava dat	17
3.1.1 Geometrie tělesa	17
3.1.2 Okrajové podmínky	18
3.1.3 Definování působících sil	19
3.1.4 Tvorba zadání úlohy	20
3.2 Sestavení matice a vektoru pro úlohu	21
3.3 Aplikace okrajových podmínek	23
3.4 Matice kontaktu	24
3.4.1 Konformní případ	24
3.4.2 Nekonformní případ	24
3.5 Řešení problému	25
3.6 Zpracování výsledků	25
3.7 Shrnutí	25
4 Zhodnocení výsledků	27
4.1 Experiment 1 - čtvercová síť	27
4.2 Experiment 2 - trojúhelníková síť - různé silové zatížení	29

4.2.1	Silové zatížení ve směru osy x	29
4.2.2	Silové zatížení ve směru y	31
4.2.3	Silové zatížení v obecném směru 1	33
4.2.4	Silové zatížení v obecném směru 2	34
4.3	Experiment 3 - trojúhelníková síť - různá hustota sítě	34
4.3.1	Zhušťování jedné sítě	35
4.3.2	Zhušťování obou sítí	36
4.4	Experiment 4 - trojúhelníková síť - jiné umístění Neumannovy okrajové podmínky	38
4.5	Experiment 5 - trojúhelníková síť - jiné umístění Dirichletovy okrajové podmínky	38
4.6	Experiment 6 - trojúhelníková síť - ukázka tří oblastí	40
4.6.1	Bez nenulové Dirichletovy okrajové podmínky	40
4.6.2	S nenulovou Dirichletovou okrajovou podmínkou	42
4.7	Experiment 7 - trojúhelníková síť - ukázka jiného tvaru oblastí	43
5	Závěr	45
	Příloha	47
	Obsah přiloženého CD	47
	Literatura	48

Seznam obrázků

2.1	Diskretizace oblasti pomocí čtyřúhelníků	13
2.2	Diskretizace oblasti pomocí trojúhelníků - příklad konformní vlevo a nekonformní síť vpravo	13
4.1	Experiment 1 - vstupní síť	28
4.2	Experiment 1 - výsledná deformace	28
4.3	Experiment 2 - vstupní síť	29
4.4	Experiment 2 - schéma	30
4.5	Experiment 2 - silové zatížení ve směru osy x	30
4.6	Experiment 2 - schéma 2	31
4.7	Experiment 2 - silové zatížení ve směru osy y - záporné	32
4.8	Experiment 2 - silové zatížení ve směru osy y - kladné	32
4.9	Experiment 2 - silové působení záporné konstantní na ose y a kladné na ose x	33
4.10	Experiment 2 - silové působení záporné na ose y a záporné konstantní na ose x	34
4.11	Experiment 3 - výsledek výpočtu pro maximální velikost trojúhelníku menší sítě 0.1	35
4.12	Experiment 3 - měření času potřebných k výpočtu - Zmenšení jedné sítě	36
4.13	Experiment 3 - výsledek výpočtu pro maximální velikost trojúhelníku obou sítí 0.1	37
4.14	Experiment 3 - měření času potřebných k výpočtu - Zmenšení obou sítí	37
4.15	Experiment 4 - výsledek při silovém působení v jiné části horního tělesa	38
4.16	Experiment 5 - schéma	39
4.17	Experiment 5 - výsledek při změně umístění Dirichletovy okrajové podmínky v dolním tělese	39
4.18	Experiment 6 - Vstupní data	40
4.19	Experiment 6 - schéma	41
4.20	Experiment 6 - výsledek výpočtu pro tři stejné obdélníky	42
4.21	Experiment 6 - výsledek výpočtu pro tři stejné obdélníky - nenulová Dirichletova okrajová podmínka	43
4.22	Experiment 7 - Vstupní data	44
4.23	Experiment 7 - výsledek úlohy s klínem a upevněným nosníkem.	44

Úvod

V dnešní době, ve které se technologie každým dnem vyvíjejí leží na firmách po celém světě velice těžký úkol. Pokud chtějí udržet krok s konkurencí, musí stále zlepšovat své produkty a nabízet nové a lepší funkce. Zásadní problém však nastává, pokud je postup výroby onoho produktu nákladný. Jako demonstrativní příklad může být uveden například automobilový průmysl. Zdálo by se, že v testování určité součástky do nového automobilu nebude příliš nákladné, ovšem opak je pravdou. Na zmíněnou součástku se musí aplikovat mnoho testů, jakým je například teplotní a deformační analýza. Pokud součástka selže, vyvstává nutnost vytvořit novou a znovu jí otestovat. Takto může firma lehce přijít o značné množství financí. Naštěstí však existuje levnější varianta. Díky počítačům je možné všechny potřebné testy provést ještě předtím, než je vyrobena první součástka. Takové testování je nazváno simulační.

Základem takových simulací je často **metoda konečných prvků**, která tvoří základní stavební kámen této práce. Konkrétně se text této práce bude věnovat řešení kontaktního problému pružných těles ve 2D prostoru. Pružným tělesem je v tomto případě myšlena obecná polygonální oblast s určitými fyzikálními vlastnostmi. Kontaktní problém pak spočívá ve zjišťování chování několika oblastí, které na sebe při zatížení vnějšími silami působí, čímž dochází k posuvům a deformacím v obou tělesech.

Text této práce je členěn do kapitol tak, aby postihl všechny úhly pohledu na řešenou úlohu. Nejdříve bude v Kapitole 1 podrobně popsána matematická teorie, která se skrývá za tématem tohoto textu. Součástí této části bude zejména popis všech vztahů a jejich souvislostí. V Kapitole 2 bude představena **metoda konečných prvků** a s tím spjatá i diskretizace úlohy na tvar, ze kterého je možné přejít k implementaci, které bude věnována Kapitola 3. S diskretizací úlohy jde ruku v ruce také diskretizace oblastí, přičemž v této práci se tím myslí téměř výhradně triangulace na trojúhelníkové elementy. V rámci implementace v Kapitole 3 budou popsány nejdůležitější části algoritmu tak, aby byl čtenář schopný na základě těchto popisů zmíněné části sám implementovat. Na konci kapitoly budou všechny části dány do souvislostí tak, aby bylo možné pochopit celou implementaci kontaktního problému. S tím samozřejmě souvisí i fakt, že kompletní implementace je součástí přílohy k této práci.

Kapitola 4 bude obsahovat nejdůležitější část celé práce a to výsledky několika

navržených a podrobně popsáných experimentů. Většina výsledků bude mít podobu grafů zobrazujících tělesa před a po vyřešení úlohy. Součástí výsledků však bude i graf závislosti časové náročnosti na hustotě sítě oblasti, respektive oblastí.

Nakonec bude celá práce shrnuta v závěru, který popíše splnění cílů práce, které jsou postaveny na podrobném prozkoumání a vyřešení kontaktního problému. Součástí tohoto shrnutí bude i nástin směrů, kterými by bylo možné celou práci rozšířit.

Kapitola 1

Formulace problému pružnosti

Úlohou pružnosti je myšleno řešení problému zatížení tělesa s danou tuhostí. Řešení úlohy vede na soustavu parciálních diferenciálních rovnic. Ideální by bylo řešení analytické, které by poskytlo přesné výsledky. Jak však bude ukázáno, tak v praxi analytické řešení většinou není možné kvůli složitosti problému vypočítat. Proto se k řešení využívá numerické řešení metodou konečných prvků, jejíž výsledky jsou aproximací analytického řešení. Než však bude popsána metoda konečných prvků je vhodné formulovat úlohu pružnosti.

1.1 Klasická formulace

Nejprve je potřeba nadefinovat oblast, na které bude úloha řešena. Mějme oblast Ω , která má lipschitzovskou hranici $\partial\Omega$. Tedy hranici, která je lokálně popsateľná lipschitzovskou funkcí. Pojmeme oblast se rozumí souvislá a omezená množina bodů prostoru \mathbb{R}^2 . Hranici $\partial\Omega$ můžeme rozdělit na dvě disjunktní části: $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$, kde $\partial\Omega_D$ je část hranice se zadanými posuvy a $\partial\Omega_N$ je část hranice se zadaným napětím.

Úloha pružnosti řešená na oblasti Ω je popsána pomocí základní podmínky rovnováhy

$$\frac{\partial\tau_{ij}}{\partial x_j}(u_i) + f_i = 0 \quad x \in \Omega, \quad (1.1)$$

a okrajových podmínek

$$u_i = \hat{u}_i, \quad x \in \partial\Omega_D \quad (1.2)$$

$$\tau_{ij}n_j = \hat{\sigma}_i, \quad x \in \partial\Omega_N, \quad (1.3)$$

kde u_i jsou neznámé posuvy, n_j je vnější normála k příslušné části hranice, τ_{ij} je tenzor napjatosti a f_i je síla působící na danou oblast.

Dále je potřeba uvést několik důležitých vztahů. Vztah pro tenzor napjatosti udává Hookeův zákon ve tvaru

$$\tau_{ij}(u_i) = E_{ijkl}e_{kl}(u_i) \quad x \in \Omega, \quad (1.4)$$

kde E_{ijkl} je elastický tenzor obsahující pouze konstanty. Dalším vztahem je definice Cauchyova tenzoru přetvoření

$$e_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad x \in \Omega. \quad (1.5)$$

Klasickým řešením této úlohy se rozumí hledání funkce $u \in C^2(\Omega)$ takové, aby vyhovovala výše uvedeným vztahům, tedy i okrajovým podmínkám. Toto řešení se v jednodušších úlohách dá určit přímo. Bohužel u většiny úloh, které jsou rozsáhlejšího charakteru nelze toto řešení najít a proto se častěji používá řešení numerické. Pro určení řešení této úlohy použijeme tzv. slabé formulace (resp. slabé řešení).

1.2 Slabá formulace

Pro odvození slabého řešení potřebujeme variační formulaci naší úlohy. Vezměme oblast Ω popsanou výše a na ní prostor $U = U(\Omega)$ což je prostor po částech hladkých funkcí definovaných na Ω takový, který je podprostorem prostoru spojitě diferencovatelných funkcí $C^1(\Omega)$. Do tohoto prostoru patří slabé řešení úlohy pružnosti.

Pro převedení výše uvedené úlohy na variační formulaci je třeba definovat prostor tzv. testovacích funkcí:

$$V = \{v \in U : v(x) = 0 \quad \forall x \in \partial\Omega_D\}. \quad (1.6)$$

Testovacím funkcím se také říká *virtuální* nebo *možná posunutí*. U těchto funkcí platí následující: jestliže u je řešením okrajové úlohy, potom $u+v$ splňuje zadanou okrajovou podmínku Dirichletova typu. Pokud $v \in V$, pak platí integrální identita ve tvaru:

$$\int_{\Omega} \left(\frac{\partial \tau_{ij}}{\partial x_j} (u_i) + f_i \right) v \, dx + \int_{\partial\Omega_N} (\hat{\sigma}_i - \tau_{ij} n_j) v \, dx = 0 \quad \forall v \in V. \quad (1.7)$$

Na první část prvního integrálu rovnosti aplikujeme Greenovu větu, rozepíšeme jednotlivé integrály a dále zjednodušíme. Dostaneme:

$$- \int_{\Omega} \frac{\partial v}{\partial x_j} \tau_{ij} (u_i) \, dx + \int_{\Omega} f_i v \, dx + \int_{\partial\Omega_N} \hat{\sigma}_i v \, dx = 0 \quad \forall v \in V. \quad (1.8)$$

Tato rovnost je splněna pro funkci u , na kterou klademe menší nároky, tedy je jednodušší ji najít. Do rovnosti je třeba ještě dosadit Hookeův zákon a vztah pro Cauchyův tenzor deformace, čímž dostaneme výslednou integrální identitu, která již zahrnuje okrajové podmínky Neumannova typu a je tedy nutné předepsat pouze podmínky Dirichletova typu.

Definujme tedy prostor funkcí U_D , což je prostor funkcí z U takový, že tyto funkce splňují Dirichletovu okrajovou podmínku. Potom slabé řešení problému lze popsat funkcí u takovou, že patří do prostoru U_D a zároveň splňuje integrální identitu (1.8) doplněnou o Hookeův zákon a vztah pro Cauchyův tenzor.

1.3 Kontaktní problém

Úloha kontaktu dvou či více těles [2] má stejný základ jako úloha pružnosti, pouze je definováno více částí hranice a tím i více okrajových podmínek. Označme části hranice $\partial\Omega_D$, $\partial\Omega_N$ a $\partial\Omega_C$, což jsou postupně části hranice s podmínkou Dirichletova typu, podmínkou Neumannova typu a poslední část hranice je hranice na kontaktu, tedy ta, kde se oblasti dotýkají. Řešíme úlohu v následujícím tvaru:

$$\begin{aligned} \frac{\partial}{\partial x_j} \tau_{ij}(u_i^l) + f_i^l &= 0 \quad x \in \Omega, \\ u_i &= \hat{u}_i, \quad x \in \partial\Omega_D, \\ \tau_{ij} n_j &= \hat{\sigma}_i, \quad x \in \partial\Omega_N, \\ u_n^k - u_n^l &\leq 0, \quad \tau_{ij}^{kl} \leq 0, \quad (u_i^k - u_i^l) \tau_{ij}^{kl} = 0, \quad x \in \partial\Omega_C, \end{aligned} \tag{1.9}$$

kde $u_n^k = u_i^k n_i^k$, tedy průmět posuvu do směru normály k oblasti, a $u_n^l = u_i^l n_i^l = -u_i^l n_i^l$, indexy k a l označují jednotlivé oblasti kontaktu.

Slabé řešení této úlohy odvodíme stejným způsobem jako řešení úlohy na pružnost, kde jsme dostali integrální identitu ve tvaru:

$$a(u, v) = b(v) \quad \forall v \in V, \tag{1.10}$$

kde $a(u, v) = \int_{\Omega} E_{ijkl} e_{kl}(u^l) e_{ij}(v^l) dx$ $u, v \in V$ a $b(v) = \int_{\Omega} f_i v dx + \int_{\partial\Omega_N} \hat{\sigma}_i v dx$ $u, v \in V$.

Mějme prostor $K = \{v \in V : v^k(x) - v^l(x) \leq 0 \quad \forall x \in \partial\Omega_C\}$, kde V je prostor testovacích funkcí. Pak můžeme definovat slabé řešení kontaktního problému následovně: Řekněme, že funkce $u \in K$ je řešením kontaktního problému, jestliže platí:

$$a(u, v - u) \geq b(v - u) \quad \forall v \in K, \tag{1.11}$$

kde tvary bilineární formy a a lineární formy b jsou uvedeny výše.

Problém (1.11) je ekvivalentní problému hledání funkce $u \in K$ tak, aby

$$L(u) = \min_{v \in K} L(v), \tag{1.12}$$

kde $L(v)$ je kvadratický funkcionál ve tvaru $L(v) = \frac{1}{2}a(v, v) - b(v)$.

Kapitola 2

Metoda konečných prvků

Metoda konečných prvků (MKP) je numerická metoda založená na diskretizaci kontinuálního tělesa a jemu příslušejícího slabého řešení. Výsledkem je tedy numerické řešení původně spojitého problému.

2.1 Galerkinova metoda

Celý řešený problém je třeba převést z ∞ -dimenzionálního prostoru do prostoru konečně-dimenzionálního, protože jinak by nebylo možné řešit problém numericky. Pro tento převod se používá několik metod, nejobecnější je Galerkinova metoda [13]. Princip Galerkinovy metody spočívá v nalezení přibližného řešení $u_N \in V_N$ takového, že splňuje

$$a(u_N, v_N) = b(v_N), \quad (2.1)$$

kde v_N a u_N jsou z prostoru V_N , který je podprostorem prostoru testovacích funkcí V a platí, že bázi prostoru V_N tvoří prvních N prvků báze prostoru V . Díky vlastnostem prostoru můžeme každou funkci $v_N \in V_N$ psát jako lineární kombinaci bázových prvků, tedy $v_N = \sum_{n=1}^N \alpha_n v_n$, kde α_n jsou čísla, a dále také platí, že každá funkce v_N leží i v prostoru V . Můžeme proto psát, že prostor V_N je aproximací prostoru V a funkce u_N je Galerkinovou aproximací hledaného řešení u . Jelikož řešení u_N je z prostoru V_N lze jej vyjádřit jako lineární kombinaci bázových funkcí v_N , tedy ve tvaru:

$$u_N = \sum_{i=1}^N x_i v_i. \quad (2.2)$$

Dále pokud dosadíme do naší rovnosti za u_N a zároveň za v_N postupně volíme jednotlivé bázové funkce prostoru V_N dostaneme soustavu lineárních algebraických rovnic ve tvaru

$$\sum_{i=1}^N x_i a(v_i, v_j) = b(v_j) \quad j = 1, 2, \dots, N. \quad (2.3)$$

Pokud dále také postupně volíme místo libovolných funkcí v_j přímo jednotlivé bázové funkce prostoru V_N a tuto rovnost zapíšeme maticově, dostaneme výslednou soustavu

$$\mathbb{A}\mathbf{x} = \mathbf{b}, \quad (2.4)$$

kde $x = [x_1, x_2, \dots, x_N]$ jsou hledané neznámé, matice \mathbb{A} má prvky $a_{ij} = a(v_i, v_j)$ a vektor \mathbf{b} je vektor pravých stran.

2.2 Metoda konečných prvků

Metoda konečných prvků je metoda se speciální konstrukcí bázových funkcí v metodě Galerkinova typu. Tyto bázové funkce definujeme na dané nasít'ované oblasti. Nejprve je tedy třeba vytvořit síť'.

2.2.1 Diskretizace oblasti

Definujeme dělení naší oblasti Ω na podoblasti, jejichž velikost je omezena parametrem h . Pro 2D se jako podoblasti používají obecně mnohoúhelníky, nejčastějšími jsou trojúhelníky nebo čtyřúhelníky. Parametr h určuje maximální délku hrany. Tyto podoblasti budeme nazývat *prvky*. Zavedme dělení S pomocí trojúhelníků T_i následovně

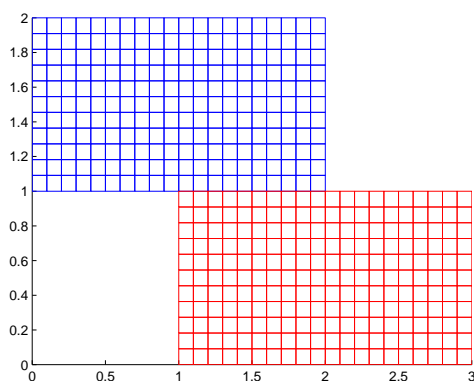
$$S = \{T_i\}, \quad (2.5)$$

kde $T_i = \{X_j, j = 1, 2, 3\}$ jsou trojúhelníky tvořené 3 body X_j , které leží v dané oblasti a pro které platí:

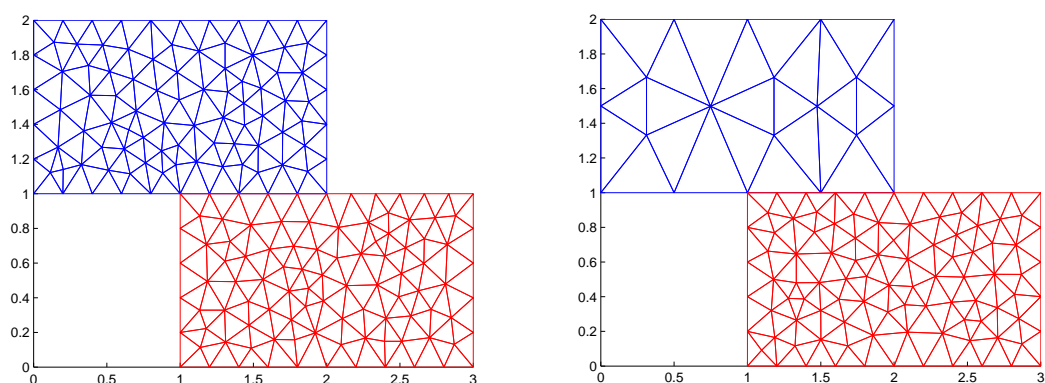
- dva trojúhelníky T_i a T_j pro $i \neq j$ mají společnou nejvýše jednu hranu
- sjednocením všech T_i dostaneme původní oblast s polygonální hranicí (dostaneme pouze aproximaci hranice)
- vnitřky dvou libovolných trojúhelníků mají prázdný průnik

Platí, že hranice oblasti je aproximována po částech stranami jednotlivých trojúhelníků resp. čtyřúhelníků. Pro celou úlohu platí, že čím jemnější dělení, tedy menší parametr h , tím přesněji je aproximováno klasické řešení. V této práci byla využita diskretizace pomocí trojúhelníků (viz. obr. 2.2) a pro nejjednodušší oblast ve tvaru obdélníků i pomocí čtyřúhelníků (viz. obr. 2.1).

Jelikož je problém řešen na kontaktu dvou či více těles je třeba definovat podobu diskretizace na kontaktu těchto oblastí. Pro tuto společnou část hranice mohou nastat dvě možnosti. Buď platí, že uzly obou sítí x_i na hranici kontaktu mají společné souřadnice, nebo některé uzly padnou do hrany trojúhelníku z druhé sítě. První možnosti se říká konformní a druhé nekonformní síť. Oba druhy sítí jsou zobrazeny na obrázku 2.2.



Obrázek 2.1: Diskretizace oblasti pomocí čtyřúhelníků



Obrázek 2.2: Diskretizace oblasti pomocí trojúhelníků - příklad konformní vlevo a nekonformní sítě vpravo

2.2.2 Bázové funkce

Bázové funkce pro úlohu volíme tak, aby splňovaly následující:

- byly na každém prvku polynomem zvoleného stupně
- byly nenulové na části oblasti (pouze několik málo prvků), tedy měli malý nosič
- byly lineárně nezávislé
- měli na celém prvku určitý stupeň hladkosti (potřeba kvůli derivaci)
- splňovali interpolační podmínky.

Volba bázových funkcí závisí také na řešené úloze, protože pro úlohu vyššího řádu, tedy úlohu s rovnicí obsahující vyšší derivace, musíme volit stupeň polynomu, tvořícího bázové funkce tak, aby odpovídal rovnici úlohy. Tedy například pro úlohu s rovnicí druhého řádu se obvykle volí lineární bázové funkce, resp. po částech lineární, jelikož tyto funkce musí splňovat, že mají malý nosič.

V této práci byly použity po částech lineární bázové funkce, neboť máme řešit rovnici druhého řádu. Tyto funkce splňují následující vlastnosti:

- jsou lineárně nezávislé
- na každém prvku obsahujícím uzel sítě X_n jsou ve tvaru lineárního polynomu $N^k(x, y) = a^k x + b^k y + c^k$
- splňují interpolační podmínky na trojúhelníku T_i (viz. 2.6)
- jsou nenulové pouze na trojúhelnících příslušejících uzlu vůči kterému jsou definované.

Interpolační podmínky pro bázovou funkci v_n definované vůči uzlu X_n jsou takové, že

$$v_n(X_n) = 1, \quad v_n(X_m) = 0 \quad \forall m \neq n. \quad (2.6)$$

2.2.3 Sestavení matic

Při sestavování matice tuhosti a vektoru zatížení tohoto problému vycházíme z tvaru bilineární formy a lineární formy popsané výše. Jednotlivé prvky matice tuhosti jsou určeny bilineární formou $a(v_i, v_j)$. Díky speciální volbě bázových funkcí popsaných výše můžeme říci, že prvek matice je nenulový pouze tehdy, když bázové funkce v_i a v_j přísluší prvkům i a j , které mají společný uzel. Matice tuhosti je čtvercová matice řádu N , kde N je počet uzlů sítě. Tato matice je také symetrická, což vyplývá z vlastností bilineární formy $a(v_i, v_j)$, která je symetrická. Dále je možné matici uvažovat jako řídkou, jelikož každý uzel sousedí jenom s konečným počtem dalších uzlů.

Celkovou matici tuhosti pro úlohu kontaktu poskládáme z matic tuhosti pro jednotlivá tělesa. Tato matice vypadá následovně:

$$\mathbb{A}_{12} = \begin{bmatrix} \mathbb{A}_1 & \mathbf{0}_{(2N, 2M)} \\ \mathbf{0}_{(2M, 2N)} & \mathbb{A}_2 \end{bmatrix},$$

kde \mathbb{A}_1 a \mathbb{A}_2 jsou matice tuhosti pro tělesa a 0 znamená nulová matice odpovídající velikosti. Jednotlivé matice tuhosti poskládáme z matic spočtených vždy pro příslušný prvek a následně přičtených do globální matice. Prvky matice tuhosti pro element vypočteme následovně:

$$\mathbb{A}_e = \mathbb{B}^T \mathbb{E} \mathbb{B}, \quad (2.7)$$

kde matice \mathbb{B} je matice poskládaná z derivací bázových funkcí a matice \mathbb{E} je matice koeficientů. Těmito koeficienty jsou Lamého koeficienty λ a μ , které jsou určeny z Youngova modulu pružnosti E a Poissonovy konstanty ν :

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \mu = \frac{E}{2(1+\nu)}.$$

Následně je třeba matice \mathbb{A}_e umístit do globální matice tuhosti \mathbb{A} . Pro tuto operaci (jde o přičítání) je třeba pouze kontrolovat, aby se jednotlivé prvky matice \mathbb{A}_e přičetli na správné pozice v globální matici. Toto zajistíme pomocí očíslování uzlů sítě. Uzly máme očíslovány a matice tuhosti má jednotlivé řádky resp. sloupce seřazeny podle tohoto číslování. Tím získáme kontrolu nad přičítáním \mathbb{A}_e . Využitím číslování uzlů přiřadíme jednotlivé prvky matice \mathbb{A}_e do celkové matice tuhosti tělesa \mathbb{A} tak, aby se hodnoty přičetly do všech uzlů, se kterými se operuje v matici \mathbb{A}_e . Matici \mathbb{A}_e je také možné si představit jako řídkou matici stejné velikosti jako \mathbb{A} , která má nenulové prvky na místech ovlivněných uzlů a na zbylých pozicích 0. Pak se dá přičtení této matice zapsat jako součet matic tedy:

$$\mathbb{A} = \mathbb{A} + \mathbb{A}_e.$$

Prvky vektoru pravých stran \mathbf{b} jsou vyjádřeny pomocí lineární formy $b(v) = \int_{\Omega} f_i v \, dx + \int_{\partial\Omega_N} \hat{\sigma}_i v \, dx$. Jelikož v této úloze předpokládáme, že síly nepůsobí uvnitř tělesa, ale pouze na povrchu, v lineární formě první integrál nebude hrát roli (bude nulový). Další částí lineární formy je integrál přes Neumannovskou část hranice. Pokud nám působí na těleso osamělá síla pouze v bodu, tak jen dosadíme na příslušnou pozici v pravé straně hodnotu zadanou v Neumannovské okrajové podmínce. Pokud nám na těleso působí spojitá síla, musíme určit hodnoty vektoru pravých stran, příslušných uzlům ležícím na Neumannovské hranici. Nejjednodušším způsobem jak tohoto docílit je integrovat numericky vždy přes danou hranu prvku, která leží na hranici s Neumannovskou podmínkou. Dalším způsobem je aproximovat hodnotu pomocí následujícího vzorce:

$$\mathbf{b}_e = \frac{1}{2} f \|h\|,$$

kde f je hodnota síly ve středu hrany a $\|h\|$ je velikost příslušné hrany.

Poslední maticí, kterou je třeba sestavit pro řešení úlohy je matice kontaktních podmínek, zkráceně matice kontaktu. Tato matice je obdélníková a rozměry jsou dány počtem okrajových podmínek na kontaktu a počtem stupňů volnosti, tedy počtem uzlů, pro které není dána Dirichletovská okrajová podmínka. Podmínka na kontaktu je dána ve tvaru:

$$u_n^k - u_n^l \leq 0,$$

kde $u_n^k = u_i^k n_i^k$, tedy průmět posuvu do směru normály k oblasti, a $u_n^l = u_i^l n_i^k = -u_i^l n_i^l$. Pokud tedy dosadíme za u_n^k a u_n^l dostaneme

$$u_i^k n_i^k - u_i^l n_i^k \leq 0,$$

pokud tedy zapíšeme tuto podmínku maticově máme

$$\mathbb{K}\mathbf{x} \leq \mathbf{0},$$

kde \mathbf{x} je vektor neznámých u_i^k a u_i^l a matice \mathbb{K} je řídká matice, která obsahuje na příslušných pozicích složky normály n_i^k (je třeba pouze kontrolovat znaménka). Pro konformní sítě se tato matice poskládá jednoduše tak, že se na příslušné pozice umístí složky normály. Pro nekonformní sítě je třeba určit, kolik máme podmínek na kontaktním okraji. Toto závisí na počtu uzlů sítě na kontaktní hranici a volí se vedoucí síť, která má více uzlů. Každý uzel z vedoucí sítě se dá vyjádřit jako lineární kombinace uzlů z druhé sítě mezi nimiž leží. Tedy označíme-li uzly vedoucí sítě X_i a uzly druhé sítě Y_i můžeme lineární kombinaci zapsat následovně

$$X_i = \alpha Y_k + (1 - \alpha) Y_l,$$

kde $Y_k Y_l$ je hrana trojúhelníka z hrubší sítě ležící na kontaktní hranici. V tomto případě je třeba do matice kontaktu zahrnout i koeficient α , abychom dostali vztah mezi jednotlivými uzly. Matice kontaktu má tedy rozměry počet uzlů vedoucí sítě krát počet stupňů volnosti. Opět doplňujeme do této matice hodnoty pro normálu ovšem pro uzly z hrubší sítě tuto hodnotu přenásobíme koeficientem α pro uzel ležící vlevo od uzlu určující podmínku nebo $(1 - \alpha)$ pro uzel ležící vpravo.

2.3 Výsledná soustava

Shrneme-li všechny výše popsané úkony, dostaneme výslednou soustavu na řešení úlohy pružnosti na kontaktu. Máme tedy diskretizovanou úlohu ve tvaru

$$\mathbb{A}\mathbf{x} = \mathbf{b}, \quad \mathbb{K}\mathbf{x} \leq \mathbf{0}, \quad (2.8)$$

kde nerovnice vyjadřuje kontaktní okrajovou podmínku.

Řešení výše uvedené úlohy je získáno převedením na úlohu kvadratického programování, které minimalizuje kvadratickou formu

$$\min_x \frac{1}{2} \mathbf{x}^T \mathbb{A} \mathbf{x} + \mathbf{b}^T \mathbf{x}, \quad kde \begin{cases} \mathbb{K}\mathbf{x} \leq \mathbf{h} \\ \mathbb{K}_{eq}\mathbf{x} = \mathbf{h}_{eq} \\ \mathbf{l}_h \leq \mathbf{x} \leq \mathbf{u}_h \end{cases}$$

kde v případě této úlohy budou využity pouze nerovnostní podmínky ve tvaru

$$\mathbb{K}\mathbf{x} \leq \mathbf{0}.$$

Výsledkem získaným z aplikace kvadratického programování jsou posuvy v jednotlivých uzlech všech oblastí, se kterými je úloha prováděna. Pro porovnání původního stavu a stavu po vyřešení kontaktní úlohy je potřeba tyto posuvy přičíst k původním souřadnicím jednotlivých uzlů. Podrobněji bude celý postup probrán v Kapitole Realizace a názorně ukázán v Kapitole Zhodnocení výsledků.

Kapitola 3

Realizace

Tato kapitola se podrobně věnuje implementaci řešené úlohy v prostředí MATLAB. V následujících odstavcích budou popsány hlavní části algoritmu a zajímavosti, se kterými se bylo nutné při psaní skriptu vypořádat ([11], [10], [8]).

V některých částech implementace byly využity funkce prostředí MATLAB, které významně ulehčily tvorbu celého algoritmu. Tyto funkce budou v příslušných odstavcích zmíněny.

Celá kapitola je uspořádána tak, aby popsala celý skript od jeho spuštění až po vykreslení výsledků. Nejdříve popíšeme přípravu dat, zejména pak geometrii obou těles, okrajové podmínky a způsob jakým je zadání úlohy vytvářeno. Dále zde bude popsáno sestavování důležitých matic a vektorů, na které se následně aplikují Dirichletovy s Neumannovy okrajové podmínky. V neposlední řadě popíšeme matici kontaktu, jejíž tvar je podmíněn geometrií obou dotýkajících se těles. Nakonec bude popsáno vyřešení úlohy následované krátkým shrnutím.

3.1 Příprava dat

Promyšlení problému přípravy dat je velice důležité, neboť může velmi ovlivnit strukturu dalších částí algoritmu. V tomto případě bylo vše voleno tak, aby bylo zadávání dat co nejjednodušší a umožnilo během krátké doby vytvořit úlohu s odlišnými tělesy i parametry.

3.1.1 Geometrie tělesa

V této práci jsou používány polygonální tvary těles, proto se pro zadání geometrie tělesa používá sada bodů, jejichž poloha je definována v souřadných osách x i y . Například pro definici těles zobrazených na grafech v obrázku 2.2 vypadá definice geometrie následovně:

```
geometry(1).x = [0;2;2;0];  
geometry(1).y = [1;1;2;2];
```

```

geometry(2).x = [1;3;3;1];
geometry(2).y = [0;0;1;1];

```

Jsou tedy postupně zadány x-ové a y-ové souřadnice prvního a druhého tělesa. Dále je zapotřebí definovat společnou hranici, na které se obě tělesa dotýkají. Opět je zadání rozděleno na jednotlivé souřadnice

```

geometry(1).edge_x = [1,2];
geometry(1).edge_y = [0.99,1.01]
geometry(2).edge_x = [1,2];
geometry(2).edge_y = [0.99,1.01].

```

Výše uvedené hodnoty představují vždy počáteční a koncový bod oblasti, na které se nachází společná hranice obou těles. V případě, že je hranice rovnoběžná s některou ze souřadných os (v tomto případě osa y), je vhodné vytvořit tuto oblast jako velice úzkou oblast namísto pouhé úsečky, aby nedošlo k nechtěnému vynechání některého z bodů hranice.

Pro další zpracování je nutné zadat ještě jeden důležitý parametr udávající maximální velikost hrany trojúhelníku. Parametr se samozřejmě udává pro každé těleso zvlášť, aby se mohlo docílit řešení problému s konformní i nekonformní sítí. Výše uvedený příklad se tedy doplní o dva řádky (v tomto případě se bude jedna o úlohu s konformní sítí)

```

geometry(1).tr_size = 0.2;
geometry(2).tr_size = 0.2;.

```

Pokud na začátku požadujeme, aby sítě byly ve výsledku konformní, je třeba zadat při vytváření geometrie mimo vrcholů oblasti i počáteční a koncový vrchol hrany na kontaktu a také mít sítě se stejným parametrem maximální velikosti hrany trojúhelníku. Na toto zadání se již mohou aplikovat algoritmy, které vytvoří dekompozici vstupních polygonů na sítě. V práci je využívána zejména trojúhelníková síť, která je získána využitím funkcí, které nabízí prostředí MATLAB. Konkrétně se jedná o funkci `decsg` z balíku PDE Toolbox. Tato funkce převede geometrii objektu na dekomponovanou geometrii tvořenou sadou nepřekrývajících se oblastí. Tato dekompozice slouží jako vstup do funkce `initmesh`, která dekomponovanou geometrii převede na matice p, e, t , které obsahují informace o trojúhelníkové síti. Konkrétně obsahují po řadě informace o bodech sítě, hranicích tělesa a jednotlivých trojúhelnících uvnitř sítě. Výsledkem procesu jsou dvě trojúhelníkové sítě (viz grafy s trojúhelníkovou sítí na obrázku 2.2).

3.1.2 Okrajové podmínky

Další informací, která je potřebná k vyřešení úlohy kontaktu těles jsou Dirichletovy a Neumannovy okrajové podmínky udávající oblast se zadanými posuvy a oblast, kde na těleso působí vnější síly.

Zadávání Dirichletových podmínek je prováděno podobným způsobem jako zadání geometrie. Opět uvažujme příklad z obrázku 2.2 s tím, že budou zadány dvě nulové a jedna nenulová okrajová podmínka.

```
dirichlet(1).area = [0,0,1,2];
dirichlet(1).value = [0, 0.5];
    dirichlet(1).object = 1;
dirichlet(2).area = [3,3,0,1];
    dirichlet(2).value = 0;
    dirichlet(2).object = 2;
dirichlet(3).area = [0,0,1,1];
    dirichlet(3).value = 0
    dirichlet(3).object = 1;.
```

Pro každou podmínku jsou zadány tři typy údajů. číslo objektu, ke kterému podmínka patří, hodnota podmínky a oblast, na které je podmínka zadána. Oblast je určena vektorem s prvky, jejichž význam je zleva doprava následující

1. x_1 - udávající x-ovou souřadnici počátku obdélníkové oblasti,
2. x_2 - udávající x-ovou souřadnici konce obdélníkové oblasti,
3. y_1 - udávající y-ovou souřadnici počátku obdélníkové oblasti,
4. y_2 - udávající y-ovou souřadnici konce obdélníkové oblasti.

Hodnota Dirichletovy okrajové podmínky je v případě nulové podmínky zadána jednoduše číslem 0. V případě nenulové podmínky se zadává vektor posunu ve směru x a y , jak je patrné na podmínce 1 ve výše uvedeném příkladu.

V případě Neumannovy podmínky je zadán vektor s oblastí¹, na níž bude okrajová podmínka aplikována.

```
neumann = [1,2,2,2];.
```

V tomto však nestačí zadat pouze část hranice, kde působí síly, ale též hodnoty působící síly v bodech oblasti. Jelikož se jedná o zajímavou část algoritmu, bude definicí síly věnována samostatná sekce.

3.1.3 Definování působících sil

Pro modelování sil v této práci byla vybrána lineární funkce dvou proměnných ve tvaru:

$$f(x, y) = ax + by + c,$$

¹Platí stejné uspořádání vektoru jako v případě Dirichletovy podmínky výše.

kde x a y jsou souřadnice bodu, ve kterém je síla počítána a a, b, c jsou koeficienty. Tato funkce tvoří plochu, díky čemuž lze modelovat změny silového působení v obou souřadných osách.

Při zadávání úlohy je potřeba zadat dvě takové funkce. Jednu pro modelování silového působení ve směru osy X a druhou pro modelování silového působení v ose Y. Jelikož by však nemuselo být pro uživatele jednoduché zadávat přímo koeficienty funkce, zadává se namísto nich sada tří bodů a jejich požadovaná hodnota. Tyto body musí definovat plochu, to znamená, že nesmí být kolineární. V případě splnění této podmínky jsou následně koeficienty lineární funkce automaticky dopočítány a je následně možné získat hodnotu této funkce v jakémkoliv bodě na hranici tělesa. Zadání silového působení vypadá následovně

```
force.x_x = [1 , 2, 2];
force.x_y = [2, 2, 1];
force.x_f = [0, 0, 0];
force.y_x = [1 , 2, 2];
force.y_y = [2, 2, 1];
force.y_f = [1e-4, 1e-5, 1-e5]; .
```

V příkladu byly definovány hodnoty sil v bodech $[1, 2]$, $[2, 2]$ a $[2, 1]$. V případě sil působících ve směru osy X jsou hodnoty nulové a proto bude i lineární funkce pro všechny body vracet hodnotu 0. Působení sil ve směru osy Y je nenulové a ze zadání je patrné, že se bude měnit vzhledem k ose X a naopak bude vždy konstantní při změně souřadnice po ose Y.

3.1.4 Tvorba zadání úlohy

K dodefinování celé úlohy zbývá pouze zadat materiálové konstanty pro jednotlivá tělesa. V práci jsou například použity tyto hodnoty

```
young = 1010 * [0.0021;0.0012]; [Pa]
poisson = [0.3; 0.34];,
```

kde konstanta 10^{10} je vytknutá z Youngova modulu pružnosti kvůli omezenosti velikosti čísel v počítači, čímž je nutno počítat s výsledky lišícími se o tuto konstantu a stejně tak upravovat velikost síly. Každý řádek vektoru definuje konstantu pro jedno těleso, je pouze důležité zachovat pořadí, ve kterém je zadána geometrie tělesa, jinak by mohlo dojít k prohození údajů a tedy vlastností.

Výše uvedené hodnoty mohou být zadány přímo v příkazové řádce programu MATLAB, ovšem ideálnější způsob je vytvoření m-skriptu, který bude obsahovat všechny údaje a následně zavolat funkci `createProblem`², která data zpracuje a následně vytvoří dva soubory kompletně definující úlohu.

²V souborech přiložených k této práci je možné u ukázkových zadání nalézt m-skript `newProblem`, který toto zadání demonstruje

První soubor je pojmenován dle parametru předaného funkci `createProblem`. Jedná se o automaticky vytvořený m-skript, ve kterém jsou uvedeny všechny konstanty, okrajové podmínky a příkaz načítající druhý soubor.

Ten je pojmenovaný zvoleným jménem, za které je přidán řetězec `_bodies.mat`. Tento soubor obsahuje sítě jednotlivých oblastí a již vypočtené parametry funkce silového působení.

V hlavním souboru aplikace (pozn. `program.m`) se pak zadání úlohy provede jednoduše. Uvažujme, že ve složce s hlavním souborem máme složku `data`, ve které jsou vygenerovaná zadání. Dále předpokládejme, že chceme spustit skript se zadáním označeným `uloha1`. Pak tedy začátek hlavního souboru musí obsahovat následující řádky

```
cd ./data/;
uloha1;
cd ..;
```

Skript se tedy přesune do složky `data`, kde zavolá skript `uloha1.m`, pak se vrátí zpět do složky s hlavním souborem a provede výpočty nad načteným zadáním.

3.2 Sestavení matice a vektoru pro úlohu

Největší část celého řešení je věnována sestavení matice tuhosti a vektoru pravých stran, tedy soustavy rovnic, jejíž vyřešením bude získán požadovaný výsledek.

Matice tuhosti \mathbb{A} je matice velikosti $2N \times 2N$, kde N je počet bodů v triangulační (resp. čtvercové) síti. Dvojnásobná velikost matice je potřebná proto, že každý bod v síti má dvě souřadnice a je nutné do matice uložit hodnoty pro každou zvlášť.

K získání matice tuhosti je nutné vytvořit nejdříve lokalizační matici, která má rozměry $N \times 6$ pro trojúhelníkovou síť a $N \times 8$ pro síť čtyřúhelníkovou. V každém řádku matice jsou uvedeny čísla uzlů tvořící jeden trojúhelník (resp. čtyřúhelník), které reprezentují polohu řádku (resp. sloupce) x-ové souřadnice tohoto uzlu v matici tuhosti (resp. vektoru pravých stran). Hodnota y-ové souřadnice je získána následovně

$$\text{číslo uzlu} + N,$$

kde N je opět počet uzlů v síti.

Výpočet elementární matice tuhosti \mathbb{A}_e pro trojúhelníkový element obsahuje několik kroků. Nejdříve se sestaví matice \mathbb{B} vyjadřující vztah mezi napětím v elementu a rozložením jeho vrcholů. Matice má následující tvar

$$\mathbb{B} = \frac{1}{2\Delta} \begin{bmatrix} y_{23} & y_{31} & y_{12} & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{32} & x_{13} & x_{21} \\ x_{32} & x_{13} & x_{21} & y_{23} & y_{31} & y_{12} \end{bmatrix},$$

kde $x_{ij} = x_i - x_j$ je rozdíl příslušných x-ových pozic uzlů v trojúhelníku a podobně $y_{ij} = y_i - y_j$ je rozdíl příslušných y-ových pozic uzlů v trojúhelníku. Dále se ve výše uvedeném vzorci vyskytuje hodnota Δ , která představuje plochu trojúhelníku a je spočtena následovně

$$\Delta = \frac{(x_2 \cdot y_3 + x_3 \cdot y_1 + x_1 \cdot y_2 - x_2 \cdot y_1 - x_3 \cdot y_2 - x_1 \cdot y_3)}{2},$$

Tím je matice \mathbb{B} kompletní. Zbývá definovat matici \mathbb{E} obsahující informace o materiálových konstantách

$$\mathbb{E} = \begin{bmatrix} \lambda + 2\mu & \lambda & 0 \\ \lambda & \lambda + 2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix},$$

význam konstant λ a μ je uveden v Kapitole 2.

Lokální matice tuhosti pro element je pak získána vynásobením matic následovně

$$\mathbb{A}_e = \mathbb{B}^T \mathbb{E} \mathbb{B}.$$

Takto získaná lokální matice tuhosti se dále musí asemblovat do celkové matice tuhosti \mathbb{A} na řádky a sloupce příslušející uzlům zpracovávaného trojúhelníku. Konkrétní postup je uveden v příloženém souboru `addToStiffnessMatrix.m`.

Vektor pravých stran \mathbf{b} o rozměrech $2N \times 1$ je vytvořen jako vektor samých nul a je následně dodefinován aplikací Neumannových okrajových podmínek, které budou společně s Dirichletovými okrajovými podmínkami popsány v následující části.

Výše uvedený postup platí v případě, že řešíme úlohu s jednou oblastí. V této práci je však řešen případ, kdy bude oblastí více. Pro jednoduchost bude uveden postup pro dvě oblasti.

Necht' máme vytvořeny matice tuhosti $\mathbb{A}_1, \mathbb{A}_2$ a k nim příslušné vektory pravých stran $\mathbf{b}_1, \mathbf{b}_2$ o rozměrech

$$\begin{aligned} \mathbb{A}_1 &\dots 2N \times 2N, \\ \mathbb{A}_2 &\dots 2M \times 2M, \\ \mathbf{b}_1 &\dots 2N \times 1, \\ \mathbf{b}_2 &\dots 2M \times 1. \end{aligned}$$

Matici tuhosti \mathbb{A}_{12} pro obě tělesa má tvar

$$\mathbb{A}_{12} = \begin{bmatrix} \mathbb{A}_1 & \mathbf{0}_{(2N,2M)} \\ \mathbf{0}_{(2M,2N)} & \mathbb{A}_2 \end{bmatrix},$$

kde $\mathbf{0}_{(I,J)}$ je nulová matice o rozměrech $I \times J$

Podobným způsobem sestavíme i vektor pravých stran, který vznikne spojením vektorů \mathbf{b}_1 a \mathbf{b}_2 do jednoho vektoru \mathbf{b}_{12} o rozměrech $2N + 2M \times 1$.

$$\mathbf{b}_{12} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix},$$

Tento postup lze samozřejmě zobecnit pro řešení úlohy s vyšším počtem oblastí. V dalších odstavcích bude vysvětlena aplikace Dirichletových a Neumannových okrajových podmínek.

3.3 Aplikace okrajových podmínek

Aplikace okrajových podmínek je v této práci rozdělena do několika souborů, jejichž názvy korespondují s názvy okrajových podmínek.

Aplikace Neumannovy okrajové podmínky vnějšího silového působení ovlivní vektor pravých stran, který byl do této chvíle nulový (*viz Sestavení matice a vektoru pro úlohu*). Postup spočívá v nalezení všech bodů, které se vyskytují na hranici uvnitř zadané oblasti.

Ke každé dvojici nalezených sousedících bodů na hranici oblasti se vypočte hodnota příslušného řádku ve vektoru pravých stran \mathbf{b} . Výpočet vypadá následovně

$$\mathbf{f}_e = \frac{1}{2} \mathbf{s} \cdot \|\mathbf{v}_{21}\|,$$

kde \mathbf{s} je síla v bodě nacházejícím se mezi dvojicí právě zpracovávaných bodů a \mathbf{v}_{21} je vektor vytvořený ze dvojice hraničních bodů.

V případě Dirichletových okrajových podmínek je situace složitější, neboť je nutné uvažovat zvláště postup v případě nulové a v případě nenulové podmínky.

Aplikace nulové podmínky spočívá ve snížení dimenze matice tuhosti \mathbf{A} a vektoru pravých stran \mathbf{b} . Tedy odstranění řádků (resp. sloupců) z řešené soustavy. Tím je zajištěno, že se pozice těchto bodů výpočtem nezmění. Problém ovšem nastává po vyřešení úlohy, jelikož je potřeba nulové posuvy vložit do výsledku úlohy na správná místa.

Nenulová okrajová podmínka, na rozdíl od nulové, nemění velikost matice tuhosti \mathbf{A} ani vektoru pravých stran \mathbf{b} . Namísto toho potřebné řádky vhodným způsobem upravuje. Obecně je možné použít více přístupů, ovšem v této práci byl zvolen implementačně jednodušší.

Postup spočívá v nahrazení řádek matice \mathbf{A} , náležících bodům v oblasti působení okrajové podmínky, nulovým řádkem s jedničkou na diagonále. Matematicky zapsáno

$$a_{ij} = 0, \quad \text{pro } i \neq j, \quad i = \text{konst.}, \quad j = 1, \dots, 2N,$$
$$a_{ii} = 1,$$

kde a_{ij} je prvek na pozici (i, j) v matici tuhosti. Dále je do vektoru pravých stran (na stejný řádek jako v matici tuhosti) uložena hodnota posunu zadaná při definici úlohy.

Tím je aplikace okrajových podmínek kompletní na soustavu jednoho tělesa kompletní. Zbývá stejným způsobem provést aplikaci podmínek i na druhé těleso³. V další podkapitole je tak již možné přistoupit k vytvoření matice kontaktu, která zohledňuje vazby mezi oblastmi.

³Obecně na všechny zbylá tělesa respektive oblasti.

3.4 Matice kontaktu

Při sestavování matice kontaktu \mathbb{K} je nutné rozeznat dva případy. Sít' na kontaktu dvou oblastí může být buď konformní, nebo nekonformní. Tedy buď se uzly sítí překrývají a nebo nepřekrývají (viz. Obr 2.2). V případě, že by matice kontaktu při řešení úlohy byla špatně sestavena, nebo nebyla k dispozici vůbec, pak se úloha nedá řešit. Došlo by totiž k tomu, že by nebyl brán zřetel na vztah oblastí na hranici a mohlo by tak dojít k proniknutí jednoho tělesa do druhého.

3.4.1 Konformní případ

Matice \mathbb{K} má v tomto případě velikost $P \times 2N + 2M$, kde P je počet bodů na kontaktu obou oblastí. Do každého řádku matice, náležícího vždy kontaktu dvou uzlů, se do sloupců patřících daným uzlům přiřadí hodnota normály k hranici oblasti. Přiřazení do i -tého řádku může vypadat například takto :

$$\begin{aligned}K(i, u1.x) &= \text{normala.x}; \\K(i, u1.y) &= \text{normala.y}; \\K(i, u2.x) &= -\text{normala.x}; \\K(i, u2.y) &= -\text{normala.y};\end{aligned}$$

Všechny ostatní body matice \mathbb{K} mají hodnotu 0. Tímto je matice připravena k řešení úlohy.

3.4.2 Nekonformní případ

V případě nekonformní sítě je sestavení matice složitější. Jelikož počet bodů na společné hranici nemusí být u obou oblastí stejný, bude i matice mít jinou velikost než v předchozím případě. Konkrétně se bude velikost lišit v počtu řádků. Tedy P již nebude počet společných uzlů na hranici. Namísto toho bude P počet uzlů hustší sítě na společné hranici. Tuto sít' nazveme řídicí. Sestavení matice je pak do jisté míry podobné. Také je potřeba do řádků jednotlivých bodů hustší sítě dostat hodnoty, které popíšou kontakt s hranicí druhé oblasti. Ovšem každý bod hustší sítě v tomto případě sousedí s dvěma uzly sítě druhé oblasti. Takovým uzlům se samozřejmě nedá přiřadit celá hodnota normály. Postup je takový, že se na základě poměru vzdáleností mezi řídicím uzlem a dvěma uzly určí dvě váhy, jejichž součet je jedničkový. Zadání hodnot do i -tého řádku může vypadat takto:

$$\begin{aligned}K(i, u1.x) &= \text{normala.x}; \\K(i, u1.y) &= \text{normala.y}; \\K(i, u21.x) &= -\text{normala.x} * \alpha; \\K(i, u21.y) &= -\text{normala.y} * \alpha; \\K(i, u22.x) &= -\text{normala.x} * (1 - \alpha); \\K(i, u22.y) &= -\text{normala.y} * (1 - \alpha);\end{aligned}$$

Po sestavení celé matice je možné přistoupit k samotnému řešení soustavy.

3.5 Řešení problému

K řešení soustavy byla v případě této práce využita funkce `quadprog`, kterou nabízí přímo prostředí Matlab. Jedná se o řešení problému kvadratického programování. Tedy hledání minima k soustavě zadané následovně⁴

$$\min_x \frac{1}{2} \mathbf{x}^T \mathbb{H} \mathbf{x} + \mathbf{f}^T \mathbf{x}, \quad \text{kde} \begin{cases} \mathbb{A} \mathbf{x} \leq \mathbf{b} \\ \mathbb{A}_{eq} \mathbf{x} = \mathbf{b}_{eq} \\ \mathbf{l}_b \leq \mathbf{x} \leq \mathbf{u}_b \end{cases}$$

Matice \mathbb{H} je v tomto případě matice tuhosti \mathbb{A} a vektor \mathbf{f} je vektor pravých stran \mathbf{b} . Dále jsou uvedeny nerovnostní a rovnostní podmínky pro řešení úlohy, které označují naší podmínku na kontaktu (\mathbb{A} přeznačme jako \mathbb{K} a vektor \mathbf{b} jako $\mathbf{b_podm}$).

Konkrétně je v této úloze zadána nerovnostní podmínka, kde výše uvedená matice \mathbb{A} je matice kontaktu \mathbb{K} a vektor $\mathbf{b_podm}$ je nulový vektor. Splněním této podmínky se v podstatě plní podmínka nepronikání jednoho tělesa do druhého. Volání funkce vypadá následovně

```
x = quadprog(A,b,K,b_podm,[],[],[],[],[],options);
```

Pět prázdných parametrů je ve volání funkce proto, že by jinak nebylo možné zadávat též parametr `options`, kterým se ovlivňuje nastavení algoritmu. Seznam možných nastavení je k dispozici v nápovědě k funkci `quadprog`.

3.6 Zpracování výsledků

V této fázi jsou již k dispozici výsledky vyřešené soustavy. Tedy konkrétně posuny v jednotlivých bodech. Jak již bylo zmíněno výše, je dále potřeba doplnit nulové posuny na místa uzlů s nulovou Dirichletovou okrajovou podmínkou. Tento postup není úplně triviální, jelikož se řádky v řešení soustavy neshodují s původními čísly uzlů v soustavě bez odebraných uzlů. Proces přidávání musí tedy probíhat postupně, jak je patrné na konci přiloženého souboru `program.m`.

Po doplnění je výsledek kompletní. Obsahuje posuny ve všech bodech obou oblastí. Zbývá tedy pouze posuny přičíst k původním umístěním bodů v síti a síť vykreslit⁵. Tím je celá úloha dokončena a je tedy vhodné celý postup krátce shrnout.

3.7 Shrnutí

V předchozích sekcích byly uvedeny zajímavé části implementace kontaktního problému. Nyní tedy zbývá zkompletovat celý postup a ukázat v jakém pořadí jednotlivé úkony v algoritmu vystupují.

⁴Záměrně je použito značení z nápovědy k programu Matlab.

⁵viz níže v kapitole Výsledky

1. **Příprava zadání a vygenerování dat** - Definování tvaru a vlastností jednotlivých oblastí, působících sil a okrajových podmínek. Vygenerování dat pomocí funkce `createProblem`.
2. **Načtení vygenerovaných dat** - Načtení všech dat ze dvou vygenerovaných souborů.
3. **Sestavení matice tuhosti \mathbb{A}_i pro každou oblast** - Výpočet lokálních matic tuhosti a jejich zařazení na správná místa v matici tuhosti oblasti.
4. **Sestavení vektoru pravých stran \mathbf{b}_i pro každou oblast a aplikace Neumannovy okrajové podmínky** - Vytvoření nulového vektoru a přiřazení nenulových hodnot k uzlům, ve kterých působí síla.
5. **Aplikace Dirichletovy nenulové okrajové podmínky** - Úprava matice tuhosti a vektoru pravých stran dle této okrajové podmínky.
6. **Aplikace Dirichletovy nulové okrajové podmínky** - Vymazání řádek a sloupců v uzlech s touto okrajovou podmínkou.
7. **Sestavení celkové matice tuhosti \mathbb{A} a vektoru pravých stran \mathbf{b}** - Sloučení všech matic tuhosti a vektorů pravých stran do jedné soustavy $\mathbb{A}\mathbf{x} = \mathbf{b}$.
8. **Sestavení matice kontaktu \mathbb{K}** - Sestavení matice kontaktů mezi jednotlivými oblastmi.
9. **Vyřešení úlohy** - zavolání funkce `quadprog`, která vyřeší soustavu za podmínek určených maticí kontaktu \mathbb{K} .

Výše uvedený výčet reprezentuje obecný algoritmus pro 2 a více oblastí. Podrobnou implementaci pak lze nalézt v souborech přiložených k práci, zejména pak v souboru `program.m`.

Kapitola 4

Zhodnocení výsledků

V předchozích kapitolách byla úloha kontaktního problému popsána z hlediska teoretického a implementačního. Nyní bude implementace prověřena za pomoci jednoduchých experimentů, které budou klást důraz na schopnost spolehlivě řešit úlohu na základě nastavených parametrů.

Jednotlivé experimenty jsou provedeny tak, aby se výše popsaná implementace otestovala z hlediska správné funkčnosti a časové náročnosti výpočtu. V následujících experimentech bude zkoumáno chování celé úlohy při různém nastavení okrajových podmínek, hustoty sítí a silového působení. V experimentu s různou hustotou sítě bude uvedena i časová náročnost výpočtu. Pro každý experiment bude uveden tvar oblastí, okrajové podmínky a hustota sítě. Součástí dále budou výsledky ve formě grafů, které jsou výstupem skriptu v programovém prostředí MATLAB. Zadané parametry budou uváděny ve tvaru:

- Dirichletova okrajová podmínka: $[x1,x2,y1,y2]$ - hodnota, číslo oblasti¹
- Neumannova okrajová podmínka: $[x1,x2,y1,y2]$ - hodnota, číslo oblasti,

kde $[x1,y1]$ je počáteční bod a $[x2,y2]$ je koncový bod části okraje oblasti, na kterou působí daná okrajová podmínka. Pokud máme zadáno více okrajových podmínek budou vypsány za sebou. Síla, kterou obsahuje Neumannova okrajová podmínka je uvažována vždy spojitá. Pro výpočet bylo třeba vzít v úvahu i velikost čísla v počítači. Proto byla z Youngova modulu pružnosti vytknuta konstanta 10^{10} , čímž bylo nutné upravit i velikost síly působící na těleso. Zobrazené výsledky jsou výsledky experimentů, kdy uvažujeme Youngův modul pružnosti větší a sílu menší.

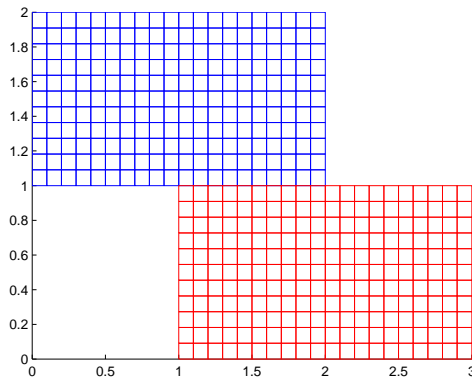
4.1 Experiment 1 - čtvercová síť

Tento experiment je ukázkou použití jiného tvaru elementů v metodě MKP. Jedná se o čtyřúhelníkové elementy, které jsou použitelné pouze v omezené míře, neboť

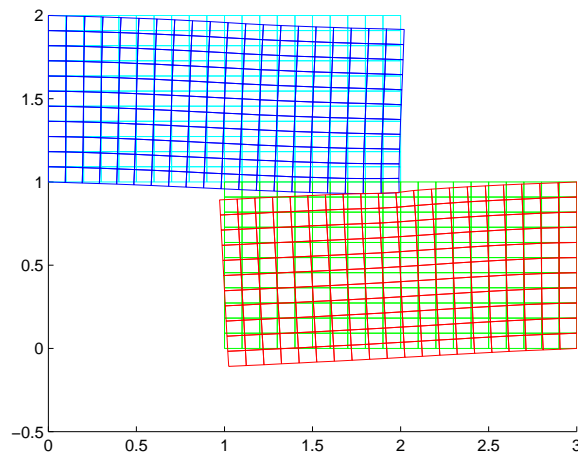
¹Oblasti jsou číslovány vzestupně zleva doprava a shora dolů

složitější tvary oblastí obecně nemůžeme diskretizovat pouze pomocí čtyřúhelníků. Z těchto důvodů bude v dalších experimentech použita výhradně trojúhelníková síť. Na následujících grafech je ukázána původní síť (viz obr. 4.1) a dále graf s výsledky (viz obr. 4.2) po nastavení následujících parametrů:

- Dirichletova okrajová podmínka: $[0,0,1,2]$ - nulový, 1; $[3,3,0,1]$ - nulový, 2
- Neumannova okrajová podmínka: $[1,2,2,2]$ - konstantní ve směru -y s velikostí $1/4 * 10^{-2}$, 1



Obrázek 4.1: Experiment 1 - vstupní síť



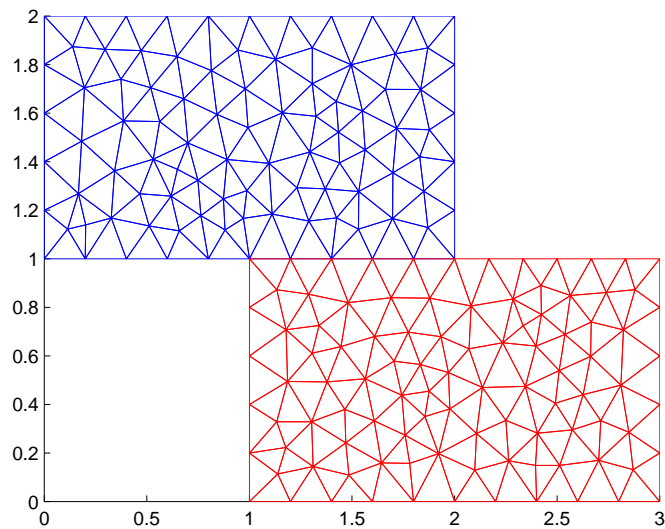
Obrázek 4.2: Experiment 1 - výsledná deformace

4.2 Experiment 2 - trojúhelníková síť - různé silové zatížení

Tento experiment je prověření chování úlohy v závislosti na různém silovém zatížení. Vstupní sítě jsou ukázány na následujícím grafu (obr. 4.3). Síla, pokud nebude uvedeno jinak, bude působit vždy na stejné části hranice modré sítě. Konkrétně bude Neumannova okrajová podmínka definovaná na oblasti $[1, 2, 2, 2]$ s různými hodnotami ve směru souřadných os x a y . Dále pro všechny části tohoto experimentu platí

- Dirichletova okrajová podmínka: $[0,0,1,2]$ - nulový, 1; $[3,3,0,1]$ - nulový, 2
- Nastavení maximální velikosti hrany trojúhelníku na hodnotu 0.2.

Jednotlivá nastavení působících sil a jim příslušné výsledky budou uvedeny v podkapitolách.

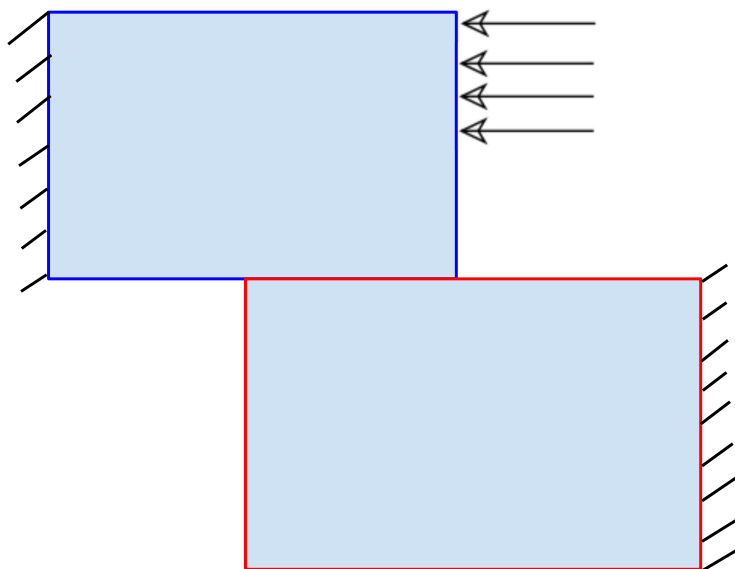


Obrázek 4.3: Experiment 2 - vstupní sítě

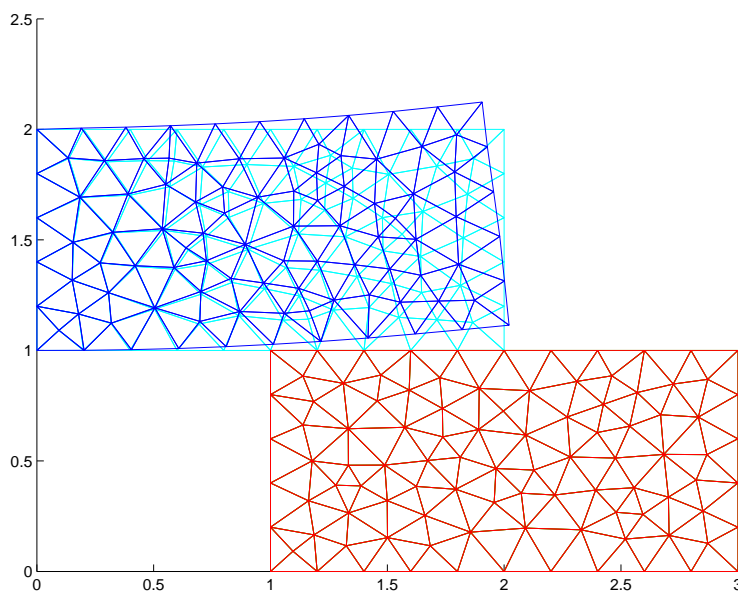
4.2.1 Silové zatížení ve směru osy x

Nejdříve bylo nutné vyzkoušet chování obou těles při silovém zatížení pouze ve směru jedné ze souřadných os. V tomto případě bylo testováno reakce při působení síly o velikosti $f = 10^{-4}$ N v záporném směru (Obr. 4.5) na hranici $[2, 2, 1, 2]$. Na obrázku 4.4 je vidět schéma experimentu.

Obrázek 4.5 ukazuje situaci, kdy je na horní těleso tlačeno v záporném směru osy x . Tím se těleso deformuje tak, že se odkloní od druhého tělesa a nedochází tak k žádnému působení na jejich kontaktu.



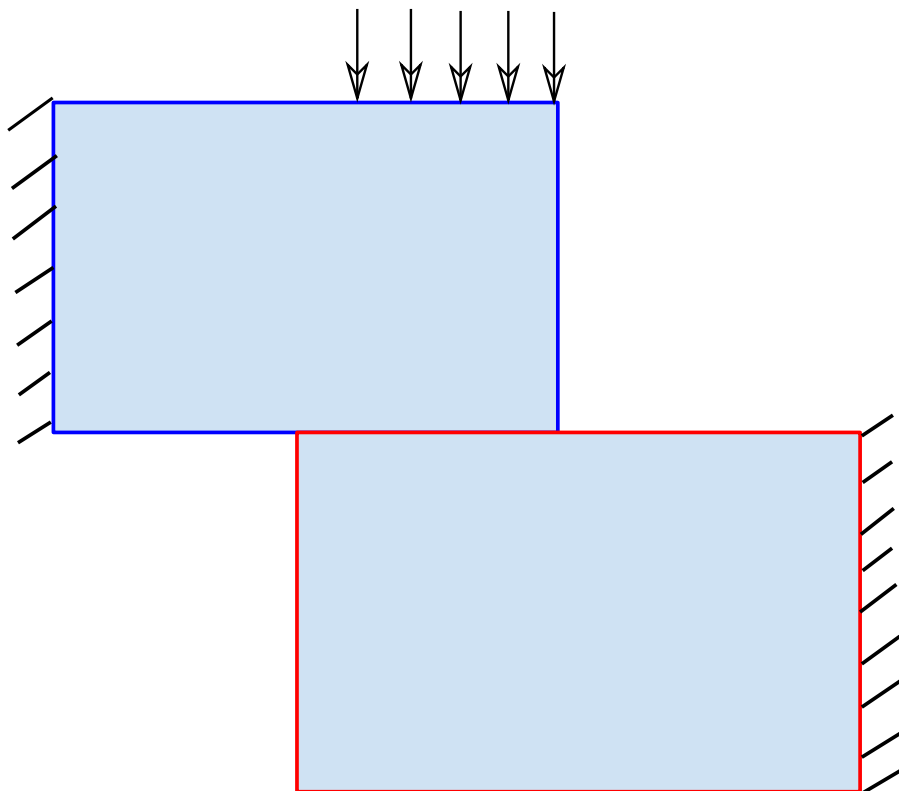
Obrázek 4.4: Experiment 2 - schéma



Obrázek 4.5: Experiment 2 - silové zatížení ve směru osy x

4.2.2 Silové zatížení ve směru y

V tomto případě je situace podobná jako v předchozím odstavci. Je použita stejná síla, pouze bude působit v záporném (Obr. 4.7) a kladném (Obr. 4.8) směru osy y . Na obrázku 4.6 je vidět schéma experimentu.

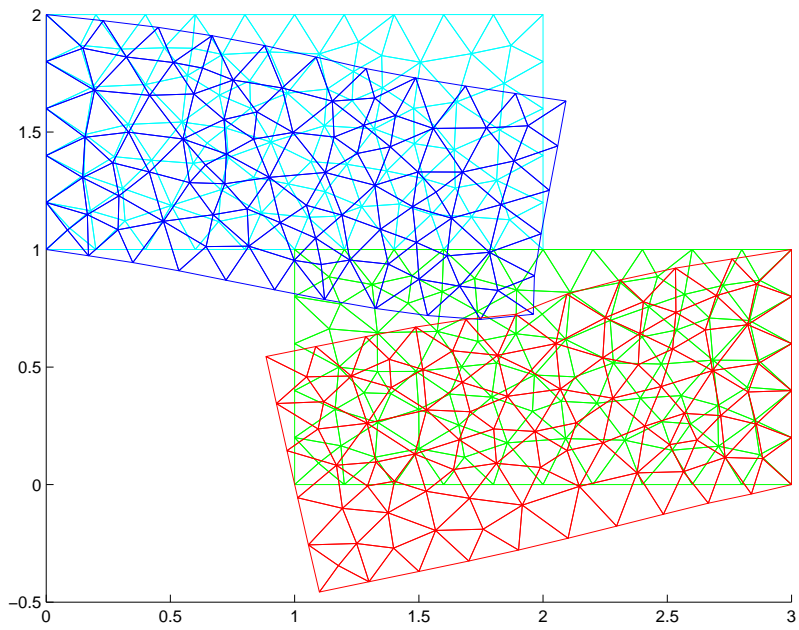


Obrázek 4.6: Experiment 2 - schéma 2

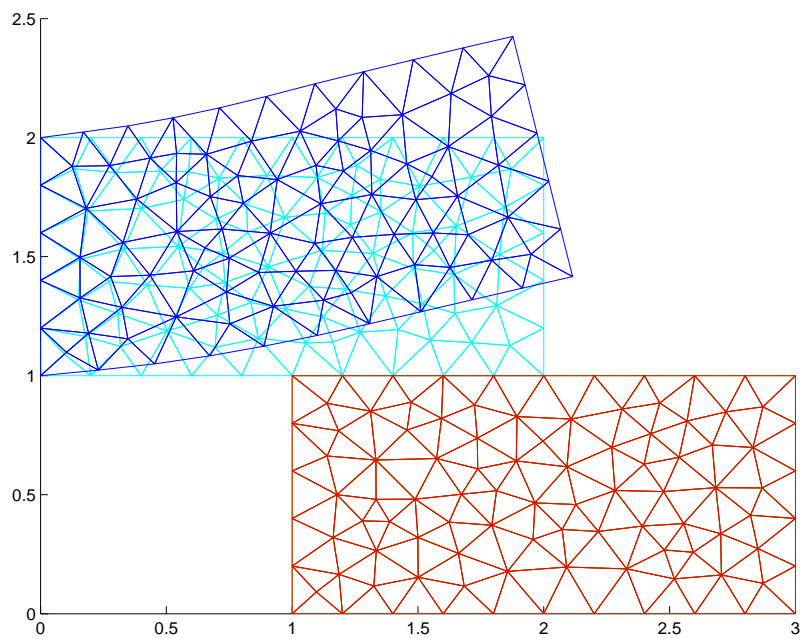
V prvním případě je situace naprosto zřejmá. Na horní těleso je působeno shora silou, díky čemuž se síla přenáší na spodní těleso, které reaguje viditelnou deformací.

Na obrázku 4.8 je vidět výsledek experimentu, při kterém síla působí v kladném směru osy y . Síť horního tělesa je deformována v tomto směru. Na kontaktu obou těles tudíž nedojde k žádnému přenosu sil a spodní těleso zůstává neovlivněno. Výše uvedené výsledky jsou tedy v souladu s představou o chování tělesa zatíženého vnější silou.

V dalších částech tohoto experimentu budou síly působit ve směru obou dvou souřadných os. Silové působení bude ve směru dané osy buď konstantní ve všech bodech nebo se bude určitým způsobem měnit. Takových silových působení se dá jistě vymyslet bezpočet. V dalších odstavcích tak bude uvedeno několik ukázkových příkladů, ze kterých je možné vyvodit fakt, že se algoritmus řešení kontaktní úlohy chová i v tomto případě správně.



Obrázek 4.7: Experiment 2 - silové zatížení ve směru osy y - záporné



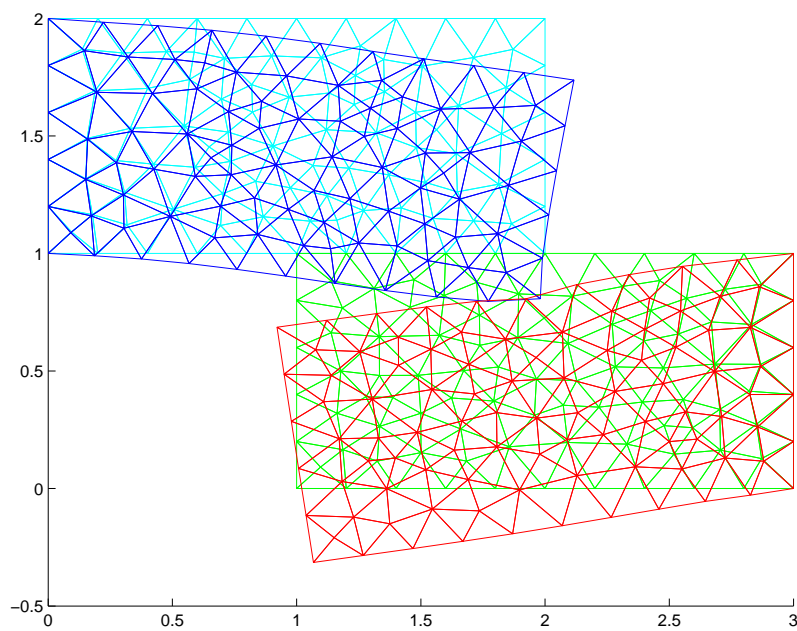
Obrázek 4.8: Experiment 2 - silové zatížení ve směru osy y - kladné

4.2.3 Silové zatížení v obecném směru 1

Nejdříve byla síla působící na těleso zvolena následovně

- Konstantní silové působení o velikosti $5 \cdot 10^{-5}$ v kladném směru osy x
- Nekonstantní síla, která na oblasti působení ve směru osy x klesá z hodnoty 10^{-4} až k hodnotě 0.

Výsledek tohoto nastavení je zobrazen na obrázku 4.9. Samotný obrázek je podobný jako v případě působení síly v záporném směru osy y. Můžeme si zde však povšimnout, že na rozdíl od výsledků na obrázku 4.7 je horní hrana modré sítě jinak tvarovaná, což způsobuje právě působení síly v kladném směru osy x.



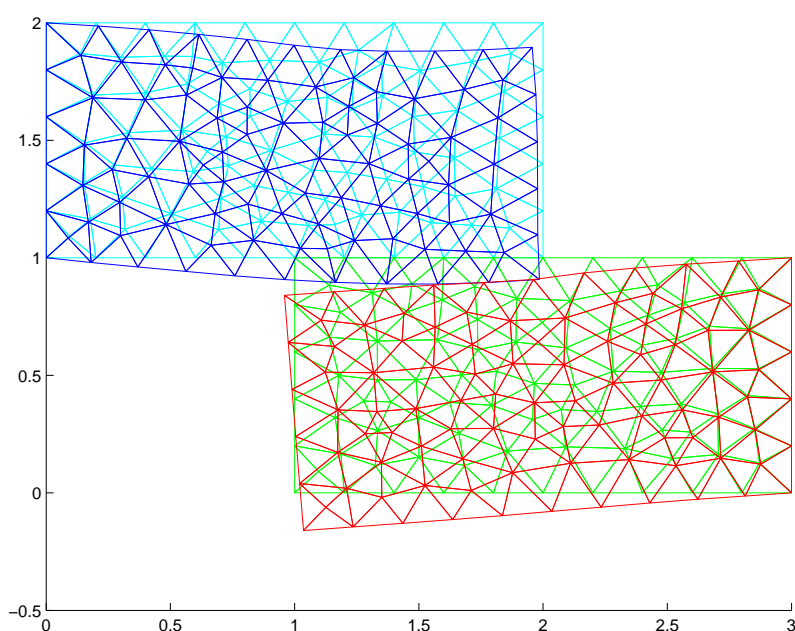
Obrázek 4.9: Experiment 2 - silové působení záporné konstantní na ose y a kladné na ose x

4.2.4 Silové zatížení v obecném směru 2

Poslední ukázka tohoto experimentu je mírně odlišná. Síla nyní v ose x působí v opačném směru. Konkrétní nastavení je následující:

- Konstantní silové působení o velikosti $5 \cdot 10^{-5}$ v záporném směru osy x
- Nekonstantní síla v záporném směru osy y, která na oblasti působení klesá ve směru osy x z hodnoty 10^{-4} až k hodnotě $2 \cdot 10^{-5}$.

Z výsledku je vidět, že se modré těleso deformuje nejen směrem k červenému tělesu, ale také se stlačuje doleva. Výsledná deformace červeného tělesa je díky tomu nižší než při absenci síly v záporném směru osy x, která deformaci tělesa snižuje jeho deformaci v záporném směru osy y.



Obrázek 4.10: Experiment 2 - silové působení záporné na ose y a záporné konstantní na ose x

4.3 Experiment 3 - trojúhelníková síť - různá hustota sítě

V předchozím experimentu byla zkoumána úloha z hlediska měnící se síly působící v úloze. Tentokrát se bude zkoumat změna výsledků při změně hustoty sítě, respektive sítí. Jako vstupní data bude použita stejná síť jako v Experimentu 2 (viz Obr. 4.3) Pro všechny části tohoto experimentu bude platit následující nastavení:

- Dirichletova okrajová podmínka: $[0,0,1,2]$ - nulový, 1; $[3,3,0,1]$ - nulový, 2
- Neumannova okrajová podmínka: $[1,2,2,2]$ - konstantní ve směru -y s velikostí $5 \cdot 10^{-5}$.

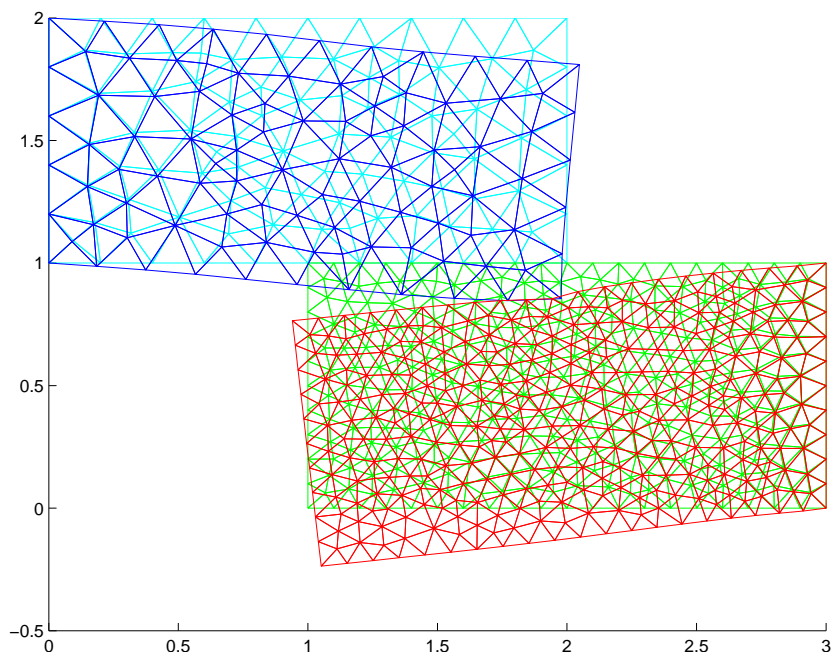
Jediným měnícím se parametrem bude maximální velikost hrany prvku, která se v programovém prostředí Matlab označuje `hmax`.

4.3.1 Zhušťování jedné sítě

Nejdříve byla zhušťována jen síť tělesa číslo 2. Byly použity následující případy hustoty sítě:

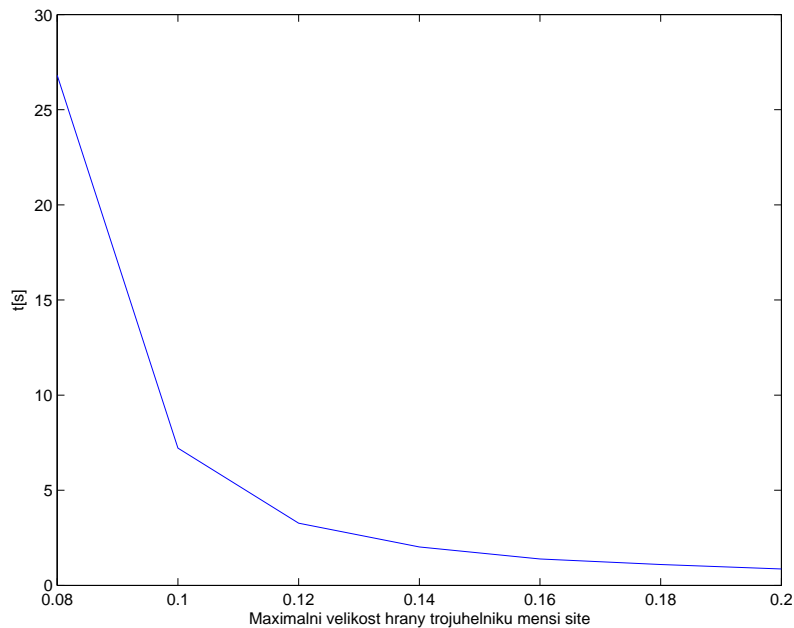
- `hmax` druhého tělesa = 0,2
- `hmax` druhého tělesa = 0,15
- `hmax` druhého tělesa = 0,1 (Obr. 4.11).

Změna hustoty jedné sítě nemá vliv na výsledky úlohy, což bylo vizuálně ověřeno. Jinými slovy nezáleží na hustotě sítě, jelikož výsledek je na pohled shodný. Dá se však předpokládat, že hustější síť bude znamenat též menší chybu způsobenou diskretizací, tedy aproximací spojitého řešení.



Obrázek 4.11: Experiment 3 - výsledek výpočtu pro maximální velikost trojúhelníku menší sítě 0.1

Tento experiment je vhodný na ukázkou závislosti času nutného pro výpočet na hustotě sítě respektive sítí. Výsledkem je pak křivka (Obr. 4.12), která při zhušťování sítě výrazně roste. To znamená, že při zhušťování sítě se čas potřebný k výpočtu výrazně prodlužuje. V tomto případě měla nejhustější síť nastaven parametr h_{max} na hodnotu 0.08. Výsledkem byl čas delší než 26 vteřin². Dá se předpokládat, že pro ještě nižší hodnoty h_{max} by se čas již musel počítat v minutách a stával by se problematický pro reálné nasazení.



Obrázek 4.12: Experiment 3 - měření času potřebných k výpočtu - Zmenšení jedné sítě

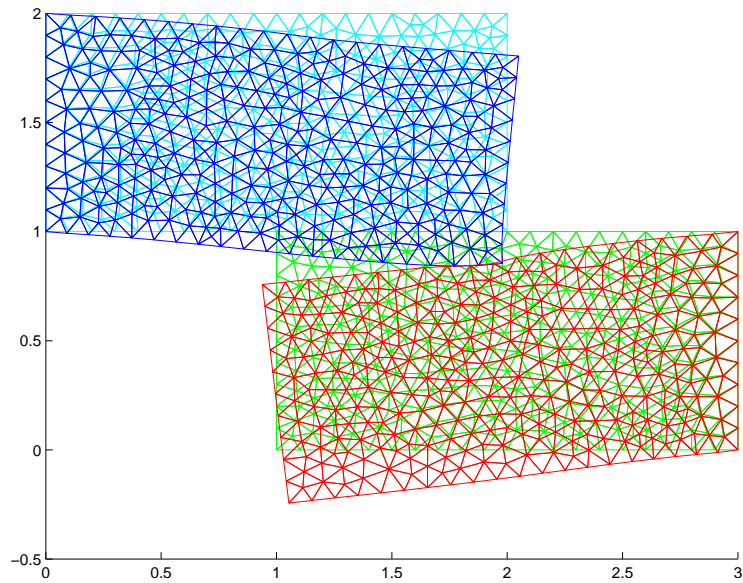
4.3.2 Zhušťování obou sítí

Pro úplnost je vhodné uvést též situaci, kdy budou zmenšovány maximální velikosti prvků obou dvou sítí. Nastavení je stejné jako v předchozím případě, jen jsou zhušťovány obě sítě zároveň.

- $h_{max} = 0,2$
- $h_{max} = 0,15$
- $h_{max} = 0,1$ (Obr. 4.13).

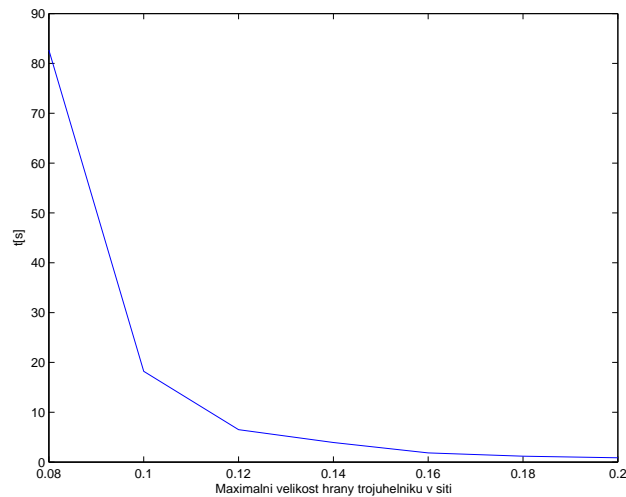
Opět se výsledek úlohy viditelně nemění, lze tedy předpokládat, že implementace funguje a provádí všechny operace správně.

²Časy byly získány na počítači s procesorem Intel Core2Duo 3,33 GHz, 6 GB RAM



Obrázek 4.13: Experiment 3 - výsledek výpočtu pro maximální velikost trojúhelníku obou sítí 0.1

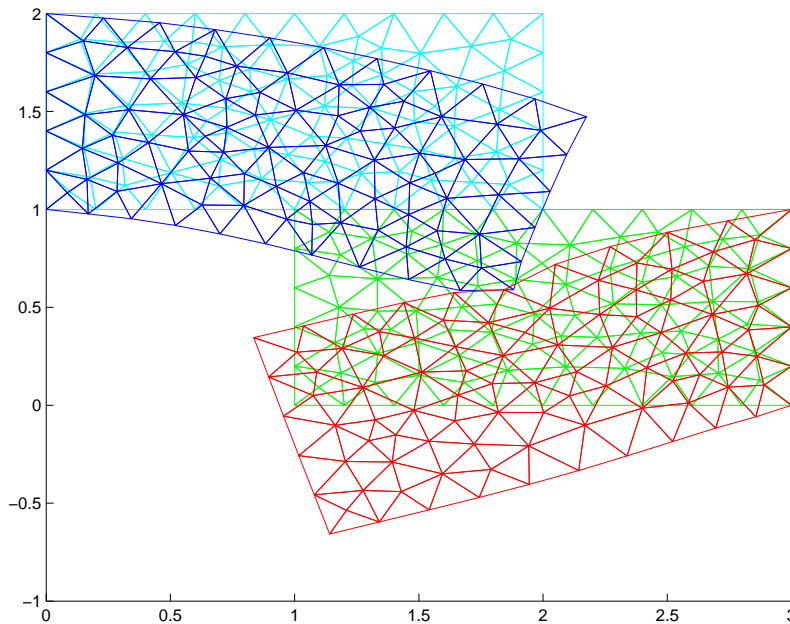
V tomto případě byly také změřeny časové hodnoty při zhušťování sítí. Výsledkem je křivka (Obr. 4.14), která je svým tvarem téměř shodná s křivkou na obrázku 4.12. Jediný rozdíl je v hodnotách, které jsou v případě zhušťování obou sítí vyšší, což je logické, jelikož při zhušťování obou sítí roste rozměr matice tuhosti výrazně rychleji. Nejvyšší naměřenou hodnotou pro $h_{\max} = 0,08$ je čas téměř 83 vteřin.



Obrázek 4.14: Experiment 3 - měření času potřebných k výpočtu - Zmenšení obou sítí

4.4 Experiment 4 - trojúhelníková síť - jiné umístění Neumannovy okrajové podmínky

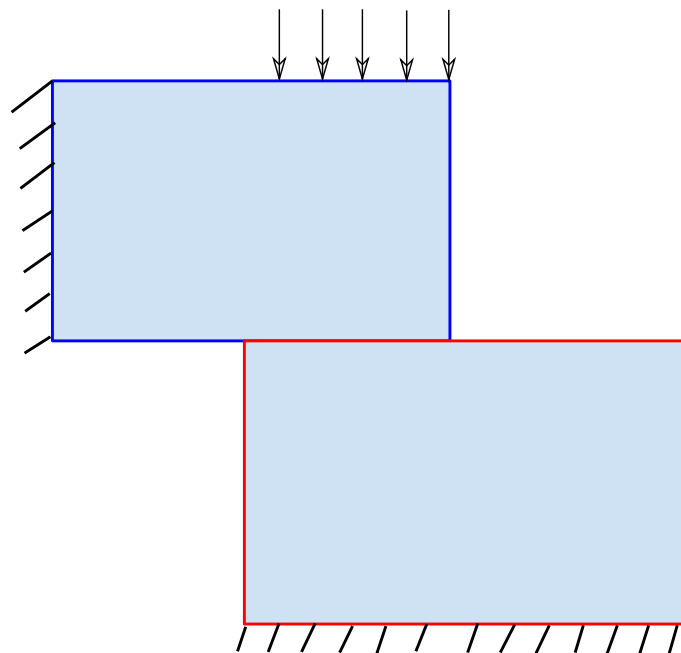
Následující experiment ukazuje změnu výsledku úlohy při silovém působení ve směru y na jiné části tělesa než tou bylo v případě nastavení na obrázku 4.7. Neumannova okrajová podmínka je nyní definována na oblasti $[2, 2, 1, 2]$, se stejnou hodnotou jako v případě výše uvedené úlohy v experimentu 2. Výsledek je zobrazen na obrázku 4.15. Rozdíly oproti 4.7 jsou zřejmé zejména na horní hraně modré sítě.



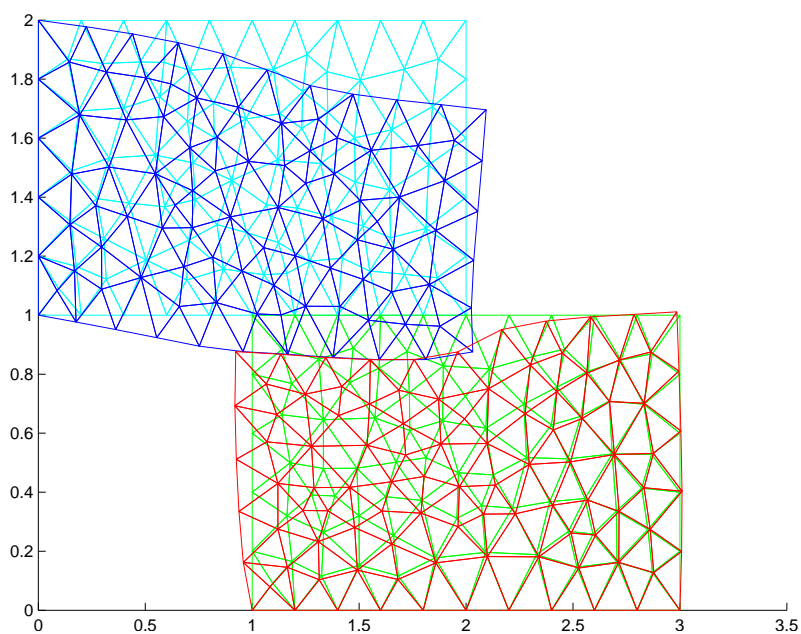
Obrázek 4.15: Experiment 4 - výsledek při silovém působení v jiné části horního tělesa

4.5 Experiment 5 - trojúhelníková síť - jiné umístění Dirichletovy okrajové podmínky

Tento experiment je rovněž připraven tak, aby jej bylo možné porovnat s obrázkem 4.7. Nyní je nastavení síly stejné jako v případě experimentu 2. Rozdíl je v umístění Dirichletovy okrajové podmínky, která je definována na spodní hraně červeného tělesa, tedy $[1, 3, 0, 0]$ (schéma 4.16). Tím je těleso pevně ukotveno proti posunutí ve směru osy y . Při působení horního tělesa, se pak spodní těleso deformuje stlačováním do sebe. Výsledek je zobrazen na obrázku 4.17.



Obrázek 4.16: Experiment 5 - schéma

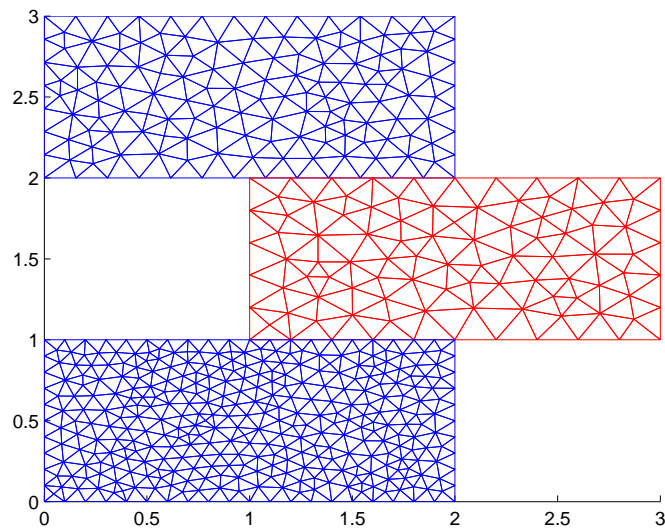


Obrázek 4.17: Experiment 5 - výsledek při změně umístění Dirichletovy okrajové podmínky v dolním tělese

4.6 Experiment 6 - trojúhelníková síť - ukázka tří oblastí

Tento experiment slouží jako ukázka úlohy s více než dvěma oblastmi. Konkrétně se jedná o dvě nastavení tří oblastí, jejichž vstupní nastavení je zobrazeno na obrázku 4.18 a schématu 4.19.

Výsledky budou v tomto případě rozděleny na dva případy. V prvním budou použity pouze nulové Dirichletovy okrajové podmínky a ve druhém bude ukázána i nenulová okrajová podmínka.



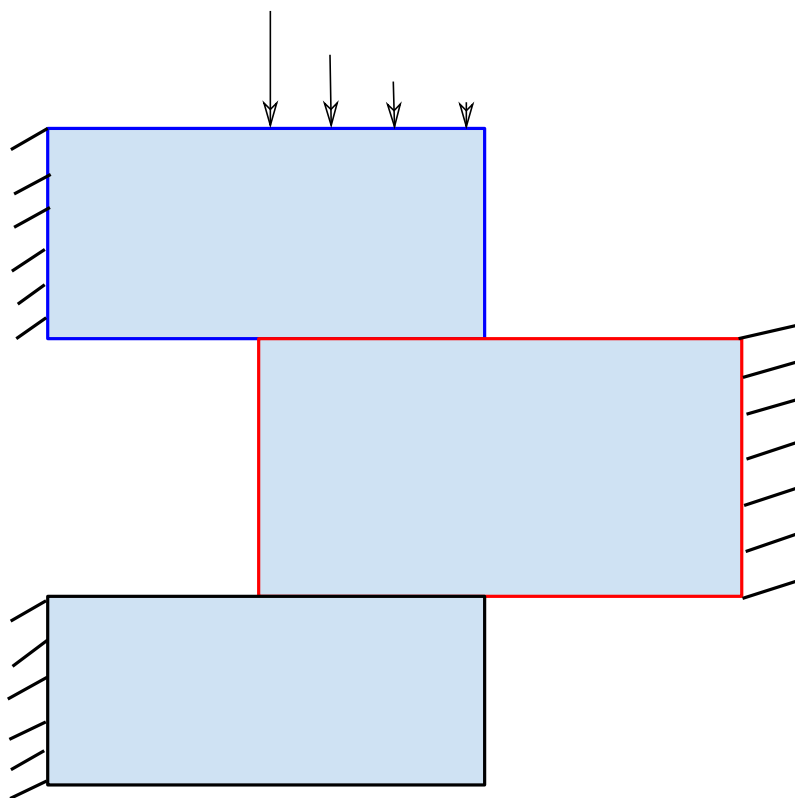
Obrázek 4.18: Experiment 6 - Vstupní data

4.6.1 Bez nenulové Dirichletovy okrajové podmínky

Nastavení podmínek pro uvedené zadání bylo následující

- Dirichletova okrajová podmínka: $[0,0,2,3]$ - nulový, 1; $[3,3,1,2]$ - nulový, 2; $[0,0,0,1]$ - nulový, 3
- Nastavení maximální velikosti hrany trojúhelníku na hodnotu 0.6 , 0.2 a 0,1.
- Síla působí nekonstantně v záporném směru osy y, přičemž se hodnota směrem doprava mění z $3 \cdot 10^{-4}$ na 0.

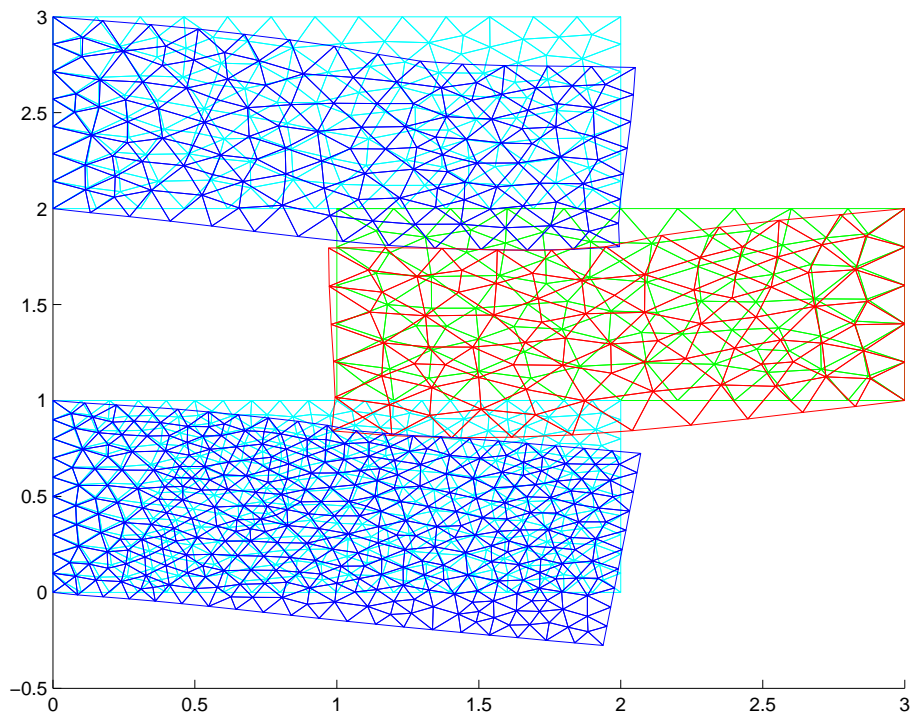
Výsledky tohoto zadání jsou uvedeny na obrázku 4.20. Je patrné, že výsledek odpovídá zadání. Síla se přenáší z prvního tělesa na druhé a z něj dál na třetí.



Obrázek 4.19: Experiment 6 - schéma

Tělesa pak reagují na deformaci první oblasti a jsou také deformovány ve stejném směru.

Tento výsledek zároveň ukazuje obecnost implementace, která je po přidání třetího tělesa schopna úlohu správně vyřešit jako v případě dvou oblastí.



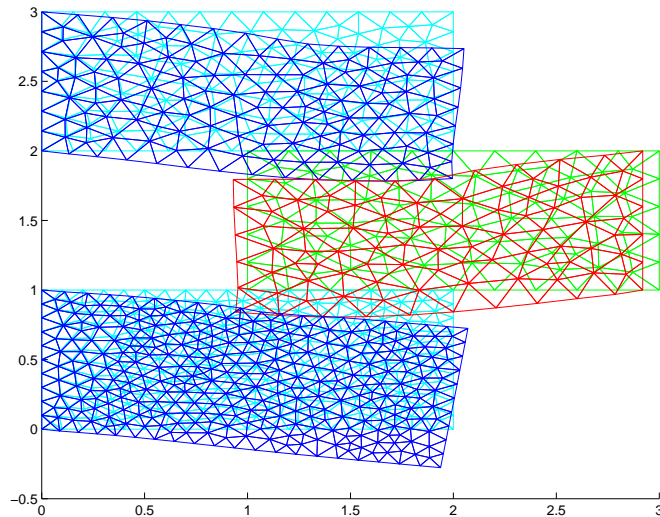
Obrázek 4.20: Experiment 6 - výsledek výpočtu pro tři stejné obdélníky

4.6.2 S nenulovou Dirichletovou okrajovou podmínkou

V této ukázce bude nastavení celé úlohy podobné, jediná změna se bude týkat výměny nulové podmínky na prostřední oblasti za nenulovou. Konkrétní nastavení je následující:

- Dirichletova okrajová podmínka: $[0,0,2,3]$ - nulový, 1; $[3,3,1,2]$: $-0,08$ v x a 0 v y , 2; $[0,0,0,1]$ - nulový, 3
- Nastavení maximální velikosti hrany trojúhelníku na hodnotu $0,6$, $0,2$ a $0,1$.
- Síla působí nekonstantně v záporném směru osy y , přičemž se hodnota směrem doprava mění z $3 \cdot 10^{-4}$ na 0 .

Na výsledcích je vidět efekt, který nenulová Dirichletova okrajová podmínka způsobí. V případě obrázku 4.21 je prostřední oblast posunuta doleva, což je jediná viditelná změna oproti výsledkům na obrázku 4.20.



Obrázek 4.21: Experiment 6 - výsledek výpočtu pro tři stejné obdélníky - nenulová Dirichletova okrajová podmínka

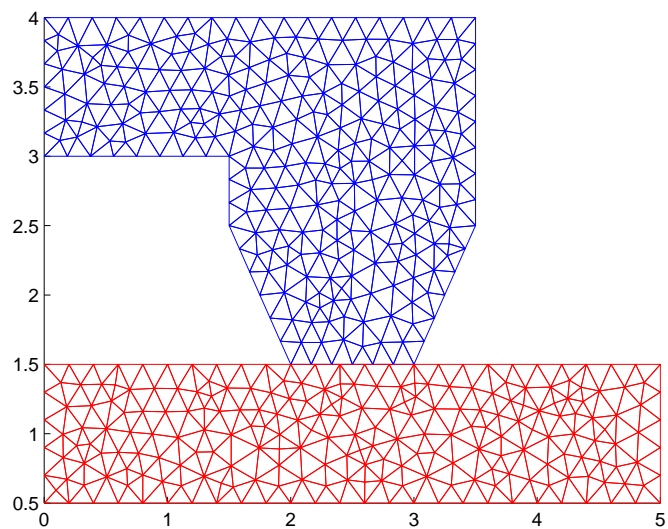
4.7 Experiment 7 - trojúhelníková síť - ukázka jiného tvaru oblastí

Tento experiment slouží jako ukázka řešení úlohy na jiném tvaru oblastí. Pro testy byly zvoleny sada oblastí, která je zobrazeny na obrázku 4.22.

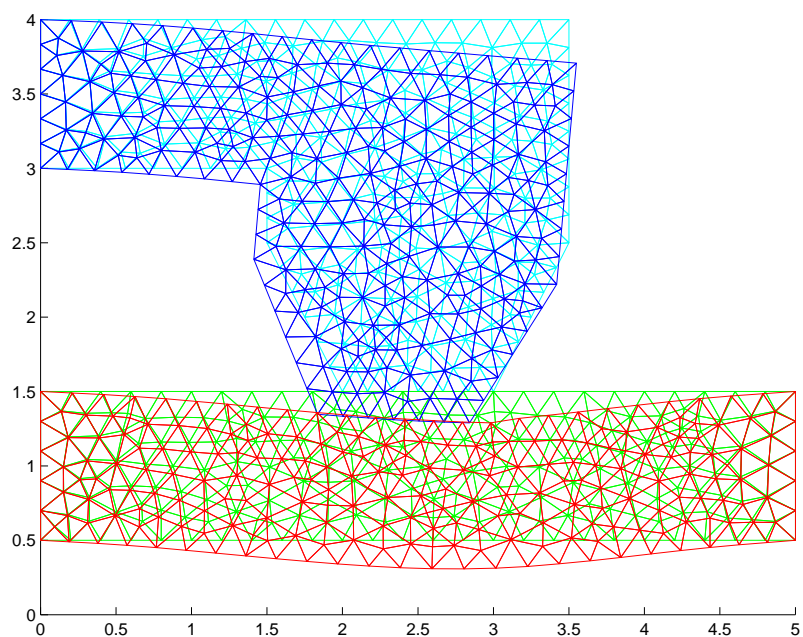
Tato dvojice oblastí byla inspirována tvarem klínu a upevněného nosníku, do kterého jej zarážíme. Pro účely výpočtu byl klín upevněn na nosník který se s ním ohýbá. Byly zvoleny následující okrajové podmínky:

- Dirichletova okrajová podmínka: $[0,0,3,4]$ - nulový, 1; $[0,0,\frac{1}{2},\frac{3}{2}]$ - nulový, 2; $[5,5,\frac{1}{2},\frac{3}{2}]$ - nulový, 2
- Neumannova okrajová podmínka: $[2.5,3.5,4,4]$ - konstantní ve směru -y s velikostí $2 \cdot 10^{-5}$.

Na obrázku 4.23 je vidět výsledek experimentu. Je zřejmé, že pokud bychom působili na klín větší silou mělo by dojít k roztržení nosníku na dva díly, tedy nosník by měl prasknout. Výsledky nám zobrazují pouze prohnutí a to z důvodu velikosti působící síly a faktu, že v této práci není možnost roztržení tělesa brána v úvahu.



Obrázek 4.22: Experiment 7 - Vstupní data



Obrázek 4.23: Experiment 7 - výsledek úlohy s klínem a upevněným nosníkem.

Kapitola 5

Závěr

Tato práce se věnovala řešení kontaktního problému pružných těles ve dvourozměrném prostoru pomocí metody konečných prvků. Problém a metoda byly nejprve po nastudování níže uvedené literatury popsány teoreticky a poté implementovány prakticky v programovém prostředí MathWorks Matlab.

Na výsledcích bylo ukázáno, že implementace úlohy pomocí metody konečných prvků funguje spolehlivě a je schopná se vypořádat s různým zadáním okrajových podmínek, různou geometrií těles i se změnami v hustotách triangulované sítě.

Konkrétně je na základě výsledků možné prohlásit, že změna hustoty některé ze sítí nezmění viditelně výsledky úlohy. Samozřejmě se dá předpokládat, že se zhuštěním sítě redukuje chyba vznikající diskretizací na trojúhelníkové prvky. Na druhou stranu se díky možnosti sítí zředit při zachování dostatečně přesných výsledků mohou významně snížit výpočetní nároky celé úlohy. Právě výpočetní náročnost se může pro husté sítě stát nemalým problémem, což bylo ukázáno v grafech na obrázcích 4.12 a 4.14, které jasně dokazují, že při volbě velmi husté sítě se může čas potřebný na výpočet výrazně prodloužit.

Geometrie těles na výsledky úlohy vliv má, neboť jiný tvar tělesa už sám o sobě ukazuje na fakt, že řešená úloha není stejná. Ovšem schopnost implementace řešit problém pro obecnou geometrii oblastí je jednoznačná výhoda, která umožní řešit ve dvourozměrném prostoru úlohy, které jsou zobecněním skutečných fyzikálních problémů, jako příklad může sloužit výsledek na obrázku 4.23, který je možné chápat jako průhyb nosníku ukotveného na obou koncích při působení síly shora. Jinými slovy se může jednat o zjednodušení úlohy pro simulaci průhybu mostu při určitém zatížení.

Nastavení okrajových podmínek, je v implementaci řešeno zadáním úseku hranice a příslušnou hodnotou. Díky tomu je možné řešit kontaktní problém určitých těles při různých silových zatíženích a při definování různých posuvů. Je tak možné simulovat mnoho rozličných situací, které by v reálném prostředí mohly nastat. Jako příklad můžeme opět uvést obrázek 4.23, kde by se mohlo silou působit na různých místech, přičemž by bylo možné zjišťovat jak se mostní konstrukce zachová. Další možností by bylo definování Dirichletovy okrajové podmínky i na spodní části druhého tělesa.

Tím by se úloha změnila a mohlo by se jednat o simulaci účinku zatížení vozovky nákladním automobilem.

Celá úloha kontaktního problému ve dvourozměrném prostoru je tedy v práci kompletně prostudována a implementována, čímž byly splněny cíle práce. Neznamená to ovšem, že by se na tuto práci nemohlo dále navázat a rozšiřovat její výsledky. Vždy se nabízí například zobecnění problému pro úlohy řešené v trojrozměrném prostoru, použití jiného modelu silového působení namísto zde použité lineární funkce dvou proměnných, nebo například do úlohy zakomponovat jiné druhy prvků, na které jsou tělesa diskretizována. I bez těchto změn se však výsledky této práce dají použít v mnoha zjednodušených úlohách, které pak mohou sloužit jako prvotní modelový příklad pro další výzkum.

Příloha

Obsah přiloženého CD

Na CD přiloženém k práci jsou k dispozici všechny materiály spojené s prací. Od zdrojové kódy, přes zadání jednotlivých experimentů až po elektronickou verzi diplomové práce. Přiložené CD má následující strukturu:

- Elektronická verze Diplomové práce
 - DiplomovaPrace.pdf
 - PDF vygenerovaných obrázků z diplomové práce
- Experimenty
 - Zadání jednotlivých experimentů ve formě Matlab skriptů.
 - Soubor cti_me.txt
- Zdrojové kódy aplikací
 - Zdrojový kód implementace pro 2 tělesa
 - Zdrojový kód implementace pro 3 tělesa
 - Zdrojový kód s ukázkou řešení pomocí obdélníkové sítě

Literatura

- [1] BREZZI F., FORTIN M.: *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, 1991
- [2] DANĚK J.: *Using Matlab for solving contact problem in elasticity*, Sborník konference MATLAB'2013, díl I., pp.94-99, Praha, 2003
- [3] FIALA O.: *Metoda nekonformních konečných prvků v rovinné úloze lineární pružnosti*, diplomová práce, ZČU Plzeň, 2011
- [4] KUTÁKOVÁ H.: *Mortar metoda konečných prvků pro lineární eliptické problémy ve 2D*, diplomová práce, ZČU Plzeň, 2008
- [5] KRÍŽEK M., NEITTAANMÄKI P.: *Finite Element Approximation of Variational Problems and Applications*, Pitman Monographs and Surveys in Pure and Applied Mathematics vol. 50, Longman Scientific & Technical Harlows, copublished in the United States with John Wiley & Sons, New York, 1990, 237 pp., ISBN 0-582-05666-7
- [6] WOHLMUTH B.: *Discretization Techniques and Iterative Solvers Based on Domain Decomposition*, Lectures Notes in Computational Science and Engineering, Vol. 17, 2001
- [7] BLAHETA R.: *Matematické modelování a metoda konečných prvků*, [online] <http://mi21.vsb.cz/modul/matematicke-modelovani-metoda-konecnych-prvku-numericke-metody-2>, 2012
- [8] SAYAS F.-J.: *A gentle introduction to the Finite Element Method*, 2008

- [9] MÍKA S., PŘIKRYL P., BRANDNER M.: *Speciální numerické metody - Numerické metody řešení okrajových úloh pro diferenciální rovnice*, Plzeň, 2006, ISBN 80-86843-13-0
- [10] CHEN L.: *Programming of Finite Element Method in MATLAB*, lecture notes, University of California, [online] <http://www.math.uci.edu/~chenlong/programming.html>
- [11] ALBERTY J., CARSTENSEN C., FUNKEN S. A.: *Remarks around 50 lines of Matlab: short finite element implementation*, Mathematisches Seminar, Christian-Albrechts-Universität zu Kiel, Ludewig-Meyn-Str. 4, D-24098 Kiel, Germany
- [12] PLÁNIČKA F.: *Základy matematické teorie pružnosti a moderní výpočtové metody*, Plzeň, 1991
- [13] OKROUHLÍK M., HÖSCHL C., PLEŠEK J., PTÁK S., NADRCHAL J.: *Mechanika poddajných těles, numerická matematika a superpočítače*, Praha, Ústav termomechaniky AV ČR, 1997
- [14] *Wikipedia.org, otevřená encyklopedie*, [online] <http://en.wikipedia.org/>