

**Návrh a implementace rozhraní mezi
COMSOlem a OOSolem pro řešení
kontaktních úloh**

Prohlášení diplomanta

Prohlašuji, že jsem celou diplomovou práci včetně příloh vypracoval samostatně a uvedl jsem všechny podklady a literaturu.

Na tomto místě bych rád poděkoval především svému vedoucímu diplomové práce Doc. Ing. T. Kozubkovi, Ph.D. a také Doc. RNDr. R. Kučerovi, Ph.D., kteří mě vedli při tvorbě této práce.

Souhlasím s tím, že s výsledky mé diplomové práce může být naloženo dle uvážení vedoucího diplomové práce jako jejího spoluautora a doporučení vedoucího katedry. V případě publikace výsledků nebo její významné části budu uveden jako spoluautor.

V Ostravě

.....
podpis diplomanta

Abstrakt

Cílem této práce bylo vytvořit rozhraní, které umožní napojení kontaktních úloh modelovaných v systému COMSOL na řešiče implementované v MATLABu a OOSOLu (objektově orientovaná knihovna vyvíjená Katedrou aplikované matematiky, VŠB-TU Ostrava na bázi C++). Sestavení matice tuhosti, vektoru zatížení a objektů z rovnostních vazeb (Dirichletovy podmínky a podmínky lepení) je provedeno pomocí COMSOLovských funkcí. Doimplementovány byly funkce pro sestavení matic a vektorů z podmínek nepronikání a zákona tření a efektivní násobící procedury pro násobení Hessiánem a zobecněnou inverzí. Výsledné rozhraní bylo aplikováno na řešení Hertzovy úlohy ve 2D a 3D. K řešení algebraické formulace úlohy byl použit algoritmus SMALBE vyvinutý prof. Dostálem z Katedry aplikované matematiky, VŠB-TU Ostrava.

Klíčová slova

Kontaktní úloha, COMSOL, algoritmus MPRGP, algoritmus SMALBE, Hertzova úloha, dané a Coulombovo tření.

Abstrakt

The goal of this work was to create an interface, which enables to connect contact problems modeled in COMSOL to the solvers implemented in MATLAB and OOSOL (object oriented library developed by the Department of Applied Mathematics, VSB-TU Ostrava in C++). Stiffness matrix, load vector and equality constraints objects (Dirichlet conditions and gluing conditions) we construct by COMSOL functions. We implemented functions for construction matrices and vectors from non-penetration condition and friction law and efficient multiplication procedures for multiplying by Hessian and generalized inverse. The resulting interface we apply on the solution of the Hertz problems in 2D and 3D. We use algorithm SMALBE developed by prof. Dostal from Department of Applied Mathematics, VSB-TU to solve the resulting algebraic formulation.

Key words

Contact problem, COMSOL, algorithm MPRGP, algorithm SMALBE, Hertz problem, given and Coulomb friction.

Seznam zkratek

$\mathbf{u}$	vektor
$\mathbf{u}^T$	transponovaný vektor
$\mathbb{A}$	matice
$\mathbb{A}^T$	transponovaná matice
$\mathbb{A}^\dagger$	zobecněná inverze matice $\mathbb{A}$
$\mathbb{I}$	jednotková matice
$\mathbb{O}$	nulová matice
$Im\mathbb{A}$	obor hodnot matice $\mathbb{A}$
$Ker\mathbb{A}$	jádro matice $\mathbb{A}$
$\mathbb{R}^n$	n -rozměrný reálný prostor
H	Hilbertův prostor
L^2	prostor měřitelných funkcí lebesgueovsky integrovatelných s kvadrátem
L^∞	prostor všech lebesgueovsky měřitelných funkcí
Ω	oblast v $\mathbb{R}^2$ nebo $\mathbb{R}^3$
$\partial\Omega$	hranice oblasti Ω
DOFs	Degrees of freedom (stupně volnosti)
PDR	parciální diferenciální rovnice
KKT	Karush-Kuhn-Tuckerovy podmínky
SMALBE	Semimonotonic Augmented Lagrangian algorithm for Bound and Equality constraints
MPRGP	Modified Proportioning with Reduced Gradient Projection

Obsah

Úvod	8
1 Formulace kontaktního problému s daným a Coulombovým třením	9
2 Dekompozice oblastí a diskretizace kontaktní úlohy	11
3 Řešiče kontaktních úloh	14
3.1 Minimalizace s rovnostními a nerovnostními vazbami	14
3.1.1 KKT podmínky a projektovaný gradient	14
3.1.2 Algoritmus SMALBE	15
3.2 Minimalizace kvadratické funkce s jednoduchými lineárními omezeními . .	16
4 Rozhraní COMSOL MATLAB	20
4.1 Co je to COMSOL	20
4.2 Co je to MATLAB	21
4.3 Vytváření modelů v COMSOLu	21
4.4 Struktura FEM	22
4.4.1 Položka version	22
4.4.2 Položka appl	22
4.4.3 Položka sdim	23
4.4.4 Položka shape a sshape	23
4.4.5 Položka border	23
4.4.6 Položka units	23
4.4.7 Položka bnd	23
4.4.8 Položka var	23
4.4.9 Položka draw	23
4.4.10 Položka geom	24
4.4.11 Položka mesh	24
4.4.12 Funkce Meshextend	25
4.4.13 Funkce Xmeshinfo	25
4.4.14 Funkce Assemble	25
4.4.15 Funkce Geominfo	25
4.5 Funkce z rozhraní COMSOL MATLAB	25
4.5.1 Funkce bc_contact_pair2d	25
4.5.2 Funkce bc_contact_pair3d	25
4.5.3 Funkce cpdata	26
4.5.4 Funkce gen_lu	26
5 Příklady	27
5.1 Hertzova úloha ve 2D	27
5.1.1 2D Hertzova úloha bez tření	28
5.1.2 2D Hertzova úloha bez tření, jemnější síť	30

5.1.3	2D Hertzova úloha s Coulombovým třením	32
5.2	Hertzova úloha ve 3D	35
5.2.1	3D Hertzova úloha bez tření	35
5.2.2	3D Hertzova úloha bez tření s rozložením těles na 16 podoblastí . .	36
5.2.3	3D Hertzova úloha s daným třením	38
Závěr		42
Literatura		43

Úvod

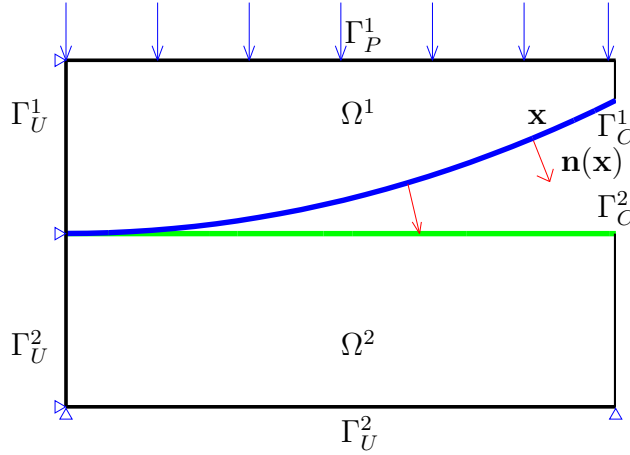
Cílem této práce je vytvořit rozhraní, které umožní napojení kontaktních úloh modelovaných v systému COMSOL na řešiče implementované v MATLABu a OOSOLu (objektově orientovaná knihovna vyvíjená Katedrou aplikované matematiky, VŠB-TU Ostrava na bázi C++). Sestavení matice tuhosti, vektoru zatížení a objektů z rovnostních vazeb (Dirichletovy podmínky a podmínky lepení) bude provedeno pomocí COMSOLovských funkcí. Doimplementovány budou funkce pro sestavení matic a vektorů z podmínek nepronikání a zákona tření a efektivní násobící procedury pro násobení Hessiánem a zobecněnou inverzí. Výsledné rozhraní bude aplikováno na řešení Hertzovy úlohy ve 2D a 3D. K řešení algebraické formulace úlohy bude použit algoritmus SMALBE vyvinutý prof. Dostálem z Katedry aplikované matematiky, VŠB-TU Ostrava.

V první kapitole se věnujeme spojitě formulaci kontaktní úlohy. Druhá kapitola pojednává o diskretizaci kontaktní úlohy a o převodu primární úlohy na duální, jenž je výhodnější z praktického pohledu. Ve třetí kapitole jsou popsány algoritmy SMALBE a MPRGP na řešení kontaktních úloh. Čtvrtá kapitola je věnována popisu systémů MATLAB a COMSOL. Speciálně se budeme zabývat strukturou *fem*, kterou podrobněji popíšeme společně s některými funkcemi užitými v rozhraní. Je zde také uveden popis jednotlivých doimplementovaných funkcí, které jsou hlavní náplní této práce. V páté kapitole je ukázán přínos implementovaného rozhraní na modelových úlohách Hertzova typu ve 2D i 3D.

1 Formulace kontaktního problému s daným a Coulombovým třením

V této kapitole si představíme spojitou formulaci kontaktní úlohy s daným a Coulombovým třením.

Uvažujme systém pružných těles reprezentovaných oblastmi $\Omega^p \in \mathbb{R}^3, p = 1, 2, \dots, s$ s Lipschitzovskými hranicemi Γ^p , které jsou rozděleny na tři disjunktní části Γ_u^p, Γ_t^p a Γ_c^p tak, že $\Gamma^p = \overline{\Gamma_u^p} \cup \overline{\Gamma_t^p} \cup \overline{\Gamma_c^p}$. Nulová posunutí jsou předepsána na hranicích Γ_u^p , zatímco povrchové síly $\mathbf{t}^p \in (L^2(\Gamma_t^p))^3$ působí na Γ_t^p . Hranice Γ_c^p oblasti Ω^p se může dostat do jednostranného kontaktu s nějakou další oblastí, což vnese efekt tření. Dále předpokládáme, že v oblasti Ω^p působí objemové síly o hustotě $\mathbf{f}^p \in (L^2(\Omega^p))^3, p = 1, 2, \dots, s$. Pro ilustraci se podívejme na obrázek 1, na kterém jsou vidět dvě tělesa ve vzájemném kontaktu.



Obrázek 1: Dvě tělesa ve vzájemném kontaktu

K popisu nepronikání těles použijeme linearizovanou podmínku nepronikání, která je definovaná zobrazením $\chi : \Gamma_c \rightarrow \Gamma_c, \Gamma_c = \bigcup_{p=1}^s \Gamma_c^p$, které přiřadí každému $\mathbf{x} \in \Gamma_c^p$ nějaký blízký bod $\chi(\mathbf{x}) \in \Gamma_c^q, p \neq q$. Nechť $\mathbf{v}^p(\mathbf{x}), \mathbf{v}^q(\chi(\mathbf{x}))$ označují vektory posunutí v bodě $\mathbf{x}$, respektive v bodě $\chi(\mathbf{x})$. Za předpokladu malých posunutí, *podmínky nepronikání* vypadají následovně:

$$v_n^p(\mathbf{x}) \equiv (\mathbf{v}^p(\mathbf{x}) - \mathbf{v}^p(\chi(\mathbf{x}))) \cdot \mathbf{n}^p(\mathbf{x}) \leq \delta^p(\mathbf{x}), \quad (1.1)$$

kde $\delta^p(\mathbf{x}) = (\chi(\mathbf{x}) - \mathbf{x}) \cdot \mathbf{n}^p(\mathbf{x})$ je počáteční vzdálenost a $\mathbf{n}^p(\mathbf{x})$ je kritický směr definovaný vztahem $\mathbf{n}^p(\mathbf{x}) = (\chi(\mathbf{x}) - \mathbf{x}) / \|\chi(\mathbf{x}) - \mathbf{x}\|$, nebo pokud $\chi(\mathbf{x}) = \mathbf{x}$, vnějším jednotkovým normálovým vektorem ke Γ_c^p .

Začneme s pomocnou kontaktní úlohou s daným třením. Za tímto účelem zavedeme prostor *virtuálních posunutí* V :

$$V = \left\{ \mathbf{v} = (\mathbf{v}^1, \dots, \mathbf{v}^s) \in \prod_{p=1}^s (H^1(\Omega^p))^3 : \mathbf{v}^p = 0 \text{ na } \Gamma_u^p \right\} \quad (1.2)$$

a jeho uzavřenou podmnožinu *kinematicky přípustných* posunutí K :

$$K = \{ \mathbf{v} \in V : v_n^p(\mathbf{x}) \leq \delta^p(\mathbf{x}) \text{ pro } \mathbf{x} \in \Gamma_c^p \}. \quad (1.3)$$

Dále předpokládejme, že normálové kontaktní napětí $T_n \in L^\infty(\Gamma_c)$, $T_n \geq 0$, takže můžeme vypočítat mez skluzu $g = FT_n$, kde $F > 0$ je koeficient tření na Γ_c . Označme $g^p = g|_{\Gamma_c^p}$.

Variační formulace kontaktního problému s *daným* třením vypadá následovně:

$$\min \mathcal{J}(\mathbf{v}) \quad \text{za podmínky} \quad \mathbf{v} \in K, \quad (1.4)$$

kde

$$\mathcal{J}(\mathbf{v}) = \frac{1}{2}a(\mathbf{v}, \mathbf{v}) - b(\mathbf{v}) + j(\mathbf{v}) \quad (1.5)$$

je funkcionál celkové potenciální energie s bilineární formou a reprezentující vnitřní energii těles a lineární formou b reprezentující práci působících sil $\mathbf{t}^p$ a $\mathbf{f}^p$. Sublineární funkcionál j reprezentuje práci třecích sil

$$j(\mathbf{v}) = \sum_{p=1}^s \int_{\Gamma_c^p} g^p \|\mathbf{v}_t^p\| d\Gamma, \quad (1.6)$$

kde $\mathbf{v}_t^p$ je projekce posunutí $\mathbf{v}^p$ na tečnou rovinu ke kritickému směru $\mathbf{n}^p$. Dále zavedeme jednotkové tečné vektory $\mathbf{t}_1^p, \mathbf{t}_2^p$ tak, že trojice $\mathcal{B} = \{\mathbf{n}^p, \mathbf{t}_1^p, \mathbf{t}_2^p\}$ tvoří ortogonální bázi v $\mathbb{R}^3$ pro téměř všechny $\mathbf{x} \in \Gamma_c^p$ a označíme $v_{t_1}^p = \mathbf{v}^p \cdot \mathbf{t}_1^p$, $v_{t_2}^p = \mathbf{v}^p \cdot \mathbf{t}_2^p$. Potom $\mathbf{v}_t^p = (0, v_{t_1}^p, v_{t_2}^p)$ vzhledem k bázi $\mathcal{B}$, a tedy norma objevující se ve funkcionálu j se redukuje na Euklidovskou normu prostoru $\mathbb{R}^2$. Více detailů o formulaci kontaktních problémů můžeme nalézt v [14].

Zdůrazněme, že řešení $\mathbf{u} \equiv \mathbf{u}(g)$ úlohy (1.4) závisí na konkrétní volbě $g \in L^\infty(\Gamma_c)$, $g \geq 0$. Můžeme tedy definovat zobrazení Φ , které přiřadí každému g součin $g = FT_n(\mathbf{u}(g))$, přičemž $T_n(u(g)) \geq 0$ je normálové kontaktní napětí odpovídající $u(g)$. Klasický Coulombův zákon tření odpovídá případu, kdy je $g = F(T_n(u(g)))$, tj. pevnému bodu zobrazení Φ . Pro jeho určení můžeme použít *metodu postupných aproximací*. Začneme se zadaným $g^{(0)}$ a generujeme $g^{(l)}$ pomocí iteračního předpisu

$$g^{(l+1)} = \Phi(g^{(l)}), \quad l = 1, 2, \dots \quad (1.7)$$

Tento iterační proces konverguje za předpokladu, že Φ je *kontraktivní*, což je garantováno pro dostatečně malé F ([15]). Další podrobnosti jsou v [16, 18].

2 Dekompozice oblastí a diskretizace kontaktní úlohy

V této části si popíšeme rozdělení oblastí Ω^p na podoblasti a diskretizaci úlohy (1.4).

Oblasti Ω^p rozdělíme na čtyřstěny $\mathcal{T}$ s maximálním průměrem h , přičemž předpokládáme, že dělení je regulární a konzistentní s ohledem na dekompozici hranice $\partial\Omega^p$ na Γ_u^p , Γ_t^p a Γ_c^p . V dalším se omezíme na případ, kdy průsečík mezi hranicemi dvou odlišných oblastí $\partial\Omega^p \cap \partial\Omega^q$, $p \neq q$, je buď prázdný, vrchol, celá hrana nebo celá stěna.

Nechť oblasti Ω^p jsou rozloženy na nepřekrývající se podoblasti $\Omega^{p,i}$, $i = 1, \dots, n_p$ a každá z nich je sjednocením konečných prvků z $\mathcal{T}$. V $\Omega^{p,i}$ zavedeme prostor $V_h^{p,i}$ následovně:

$$V_h^{p,i} = \left\{ \mathbf{v}^{p,i} \in (C(\Omega^{p,i}))^3 : \mathbf{v}^{p,i}|_{\mathcal{T}} \in (P_1(\mathcal{T}))^3 \text{ pro všechny } \mathcal{T} \subset \Omega^{p,i}, \right. \\ \left. \mathbf{v}^{p,i}|_{\partial\Omega^{p,i} \cap \Gamma_u^p} = 0 \right\}, \quad (2.1)$$

kde $P_m(\mathcal{T})$ popisuje množinu všech mnohočlenů na $\mathcal{T}$ stupně $\leq m$. Nakonec definujeme prostor $V_h = \prod_{p=1}^s \prod_{i=1}^{n_p} V_h^{p,i}$.

Nahradíme-li V za V_h a použijeme-li *lepící podmínku* $\mathbf{v}^{p,i}(\mathbf{x}) = \mathbf{v}^{p,j}(\mathbf{x})$ pro libovolné $\mathbf{x}$ na rozhraní $\partial\Omega^{p,i} \cap \partial\Omega^{p,j}$, $i \neq j$, můžeme přepsat aproximaci kontaktního problému s daným třením (1.4) na algebraickou formulaci

$$\min_{\mathbf{u}} \frac{1}{2} \mathbf{u}^T \mathbb{K} \mathbf{u} - \mathbf{u}^T \mathbf{f} + \sum_{k=1}^m g_k \|(\mathbb{T}_1 \mathbf{u})_k, (\mathbb{T}_2 \mathbf{u})_k\| \\ \text{za podmínky } \mathbb{N} \mathbf{u} \leq \mathbf{d}, \quad \mathbb{B}_E \mathbf{u} = 0. \quad (2.2)$$

Zde $\mathbb{K}$ popisuje pozitivně semidefinitní blokově diagonální matici tuhosti, $\mathbf{f}$ je vektor zatížení, $\mathbb{N}$ a $\mathbf{d}$ jsou matice a vektor z podmínky nepronikání a $\mathbb{B}_E$ je matice z lepících a Dirichletových podmínek. Člen se sumou v rovnici (2.2) vzniká v důsledku užití numerické kvadraturní formule v (1.6). Matice $\mathbb{T}_1$, $\mathbb{T}_2$ popisují projekce posunutí v uzlech ležících na hranici Γ_c k tečné rovině a g_k jsou hodnoty meze skluzu $\mathbf{g}$.

Zdůrazněme, že úloha (2.2) je nediferencovatelná z důvodu $\mathbb{R}^2$ -normy v sumačním členu. Z tohoto důvodu zavedeme dva typy Lagrangeových multiplikátorů $\boldsymbol{\lambda}_t = (\boldsymbol{\lambda}_{t_1}^T, \boldsymbol{\lambda}_{t_2}^T)^T$ a $\boldsymbol{\lambda}_c = (\boldsymbol{\lambda}_I^T, \boldsymbol{\lambda}_E^T)^T$. Zatímco první Lagrangeův multiplikátor odstraňuje nediferencovatelnost, druhý odpovídá omezením ve vztahu (2.2). Označme

$$\mathbb{B} = \begin{bmatrix} \mathbb{T}_1 \\ \mathbb{T}_2 \\ \mathbb{N} \\ \mathbb{B}_E \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{d} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_{t_1} \\ \boldsymbol{\lambda}_{t_2} \\ \boldsymbol{\lambda}_I \\ \boldsymbol{\lambda}_E \end{bmatrix} \quad (2.3)$$

a zavedme množinu Lagrangeových multiplikátorů

$$\Lambda(\mathbf{g}) = \{ \boldsymbol{\lambda} : \|(\lambda_{t_1,k}, \lambda_{t_2,k})\| \leq g_k, \quad \lambda_{I,k} \geq 0 \quad \forall k \}. \quad (2.4)$$

Úloha (2.2) je ekvivalentní sedlo-bodové úloze:

$$\text{Najdi } (\mathbf{u}, \boldsymbol{\lambda}) \text{ takové, že } \mathcal{L}(\mathbf{u}, \boldsymbol{\lambda}) = \sup_{\boldsymbol{\mu} \in \Lambda(\mathbf{g})} \min_{\mathbf{v}} \mathcal{L}(\mathbf{v}, \boldsymbol{\mu}), \quad (2.5)$$

kde $\mathcal{L}$ je Lagrangeův funkcionál k úloze (2.2) definovaný vztahem

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\lambda}) = \frac{1}{2} \mathbf{u}^T \mathbb{K} \mathbf{u} - \mathbf{u}^T \mathbf{f} + \boldsymbol{\lambda}^T (\mathbb{B} \mathbf{u} - \mathbf{c}). \quad (2.6)$$

Z minima v rovnici (2.5) obdržíme algebraickou rovnici

$$\mathbb{K} \mathbf{u} - \mathbf{f} + \mathbb{B}^T \boldsymbol{\lambda} = \mathbf{0}. \quad (2.7)$$

Tato rovnost platí tehdy a jen tehdy, jestliže $\mathbf{f} - \mathbb{B}^T \boldsymbol{\lambda} \in \text{Im} \mathbb{K}$, což je ekvivalentní podmínce kolmosti $\mathbf{f} - \mathbb{B}^T \boldsymbol{\lambda} \perp \text{Ker} \mathbb{K}$. Užitím matice $\mathbb{R}$ plné hodnosti, jejíž sloupce tvoří bázi jádra matice $\mathbb{K}$, můžeme psát

$$\mathbb{R}^T (\mathbf{f} - \mathbb{B}^T \boldsymbol{\lambda}) = \mathbf{0}. \quad (2.8)$$

Pokud je podmínka (2.8) splněna, pak vektor $\mathbf{u}$ splňující rovnici (2.7) vypadá následovně

$$\mathbf{u} = \mathbb{K}^\dagger (\mathbf{f} - \mathbb{B}^T \boldsymbol{\lambda}) + \mathbb{R} \boldsymbol{\alpha}, \quad (2.9)$$

kde $\mathbb{K}^\dagger$ popisuje zobecněnou inverzi k matici $\mathbb{K}$. Poslední výraz může být použit k eliminaci první neznámé $\mathbf{u}$ v úloze (2.5). Po jednoduché algebraické úpravě a změně znaménka obdržíme minimalizační problém pouze v druhé proměnné $\boldsymbol{\lambda}$:

$$\begin{aligned} & \min \frac{1}{2} \boldsymbol{\lambda}^T \mathbb{F} \boldsymbol{\lambda} - \boldsymbol{\lambda}^T \tilde{\mathbf{h}} \\ & \text{za podmínky } \boldsymbol{\lambda} \in \Lambda(\mathbf{g}), \quad \mathbb{G} \boldsymbol{\lambda} = \mathbf{e}, \end{aligned} \quad (2.10)$$

kde $\mathbb{F} = \mathbb{B} \mathbb{K}^\dagger \mathbb{B}^T$ je pozitivně definitní matice, $\tilde{\mathbf{h}} = \mathbb{B} \mathbb{K}^\dagger \mathbf{f} - \mathbf{c}$, $\mathbb{G} = \mathbb{R}^T \mathbb{B}^T$ má plnou řádkovou hodnotu a $\mathbf{e} = \mathbb{R}^T \mathbf{f}$.

Problém (2.10) můžeme upravit použitím ortogonálních projektorů. Definujme projektory na $\text{Im} \mathbb{G}^T$ a $\text{Ker} \mathbb{G}$ následovně:

$$\mathbb{Q} = \mathbb{G}^T (\mathbb{G} \mathbb{G}^T)^{-1} \mathbb{G} \quad \text{a} \quad \mathbb{P} = \mathbb{I} - \mathbb{Q}. \quad (2.11)$$

Jelikož $\mathbb{R}^m = \text{Im} \mathbb{G}^T \oplus \text{Ker} \mathbb{G}$, řešení $\boldsymbol{\lambda}$ úlohy (2.10) může být jednoznačně rozloženo na

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_Q + \boldsymbol{\lambda}_P, \quad (2.12)$$

kde $\boldsymbol{\lambda}_Q = \mathbb{Q} \boldsymbol{\lambda}$ a $\boldsymbol{\lambda}_P = \mathbb{P} \boldsymbol{\lambda}$. Protože $\mathbb{G} \boldsymbol{\lambda}_P = 0$, dostáváme z rovnostních vazeb v (2.10), že $\mathbb{G} \boldsymbol{\lambda} = \mathbb{G} \boldsymbol{\lambda}_Q = \mathbf{e}$ a proto $\boldsymbol{\lambda}_Q$ můžeme vypočítat podle vzorce:

$$\boldsymbol{\lambda}_Q = \mathbb{G}^T (\mathbb{G} \mathbb{G}^T)^{-1} \mathbf{e}. \quad (2.13)$$

Použitím rozkladu (2.12) ve funkcionálu v úloze (2.10) obdržíme

$$\begin{aligned} \frac{1}{2} \boldsymbol{\lambda}^T \mathbb{F} \boldsymbol{\lambda} - \boldsymbol{\lambda}^T \tilde{\mathbf{h}} &= \frac{1}{2} \boldsymbol{\lambda}_Q^T \mathbb{F} \boldsymbol{\lambda}_Q - \boldsymbol{\lambda}_Q^T \tilde{\mathbf{h}} + \boldsymbol{\lambda}_P^T \mathbb{F} \boldsymbol{\lambda}_Q + \frac{1}{2} \boldsymbol{\lambda}_P^T \mathbb{F} \boldsymbol{\lambda}_P - \boldsymbol{\lambda}_P^T \tilde{\mathbf{h}} \\ &= \text{const.} + \frac{1}{2} \boldsymbol{\lambda}_P^T \mathbb{F} \boldsymbol{\lambda}_P - \boldsymbol{\lambda}_P^T (\tilde{\mathbf{h}} - \mathbb{F} \boldsymbol{\lambda}_Q). \end{aligned} \quad (2.14)$$

Proto je (2.10) ekvivalentní s úlohou

$$\begin{aligned} & \min \frac{1}{2} \boldsymbol{\lambda}_P^T \mathbb{F} \boldsymbol{\lambda}_P - \boldsymbol{\lambda}_P^T \tilde{\mathbf{h}} \\ & \text{za podmínky } \boldsymbol{\lambda}_P + \boldsymbol{\lambda}_Q \in \Lambda(\mathbf{g}), \quad \mathbb{G} \boldsymbol{\lambda}_P = 0, \end{aligned} \quad (2.15)$$

kde $\mathbf{h} = \tilde{\mathbf{h}} - \mathbb{F}\boldsymbol{\lambda}_Q$. Protože řešení $\boldsymbol{\lambda}_P$ z (2.15) patří do $\text{Ker}\mathbb{G}$, platí $\boldsymbol{\lambda}_P = \mathbb{P}\boldsymbol{\lambda}_P$ a proto $\boldsymbol{\lambda}_P$ řeší také úlohu

$$\begin{aligned} & \min \frac{1}{2} \boldsymbol{\lambda}_P^T \mathbb{P} \mathbb{F} \mathbb{P} \boldsymbol{\lambda}_P - \boldsymbol{\lambda}_P^T \mathbb{P} \mathbf{h} \\ & \text{za podmínky} \quad \boldsymbol{\lambda}_P + \boldsymbol{\lambda}_Q \in \Lambda(\mathbf{g}), \quad \mathbb{G} \boldsymbol{\lambda}_P = \mathbf{0}. \end{aligned} \quad (2.16)$$

Zdůrazněme, že řešení (2.16) je jednoznačné, protože Hessián $\mathbb{P} \mathbb{F} \mathbb{P}$ je pozitivně definitní na $\text{Ker}\mathbb{G}$, což plyne ze vztahu

$$\boldsymbol{\delta}^T \mathbb{P} \mathbb{F} \mathbb{P} \boldsymbol{\delta} = \boldsymbol{\delta}^T \mathbb{F} \boldsymbol{\delta} > 0, \quad (2.17)$$

který platí pro každé $\boldsymbol{\delta} \in \text{Ker}\mathbb{G} \setminus \{\mathbf{0}\}$. Úlohy (2.15) a (2.16) mají proto stejné řešení.

Úlohu (2.16) řešíme algoritmem SMALBE, který představíme v následující kapitole. Po vyřešení úlohy (2.16) se vypočítají tuhé pohyby:

$$\boldsymbol{\alpha} := \left(\mathbb{R}^T \tilde{\mathbb{B}}^T \tilde{\mathbb{B}} \mathbb{R} \right)^{-1} \mathbb{R}^T \tilde{\mathbb{B}}^T \left(\tilde{\mathbf{c}} - \tilde{\mathbb{B}} \mathbf{K}^\dagger (\mathbf{f} - \mathbb{B}^T \boldsymbol{\lambda}) \right), \quad (2.18)$$

kde $\tilde{\mathbb{B}}$ je redukovaná matice $\mathbb{B}$, ze které vezmeme pouze řádky fixující tuhé pohyby. Potom se dopočítají posunutí podle rovnice (2.9).

Výše uvedené vztahy jsme odvodili pro případ ve 3D. Situace ve 2D je velmi podobná. Je třeba provést pouze tyto změny: oblasti Ω^p rozdělíme na trojúhelníky, průsečík dvou odlišných oblastí $\partial\Omega^p \cap \partial\Omega^q$, $p \neq q$, je buď prázdný, vrchol nebo celá hrana. V $\Omega^{p,i}$ zavedeme prostor konečných prvků $V_h^{p,i}$, který definujeme následovně:

$$\begin{aligned} V_h^{p,i} = \left\{ \mathbf{v}^{p,i} \in (C(\Omega^{p,i}))^2 : \mathbf{v}^{p,i}|_{\mathcal{T}} \in (P_1(\mathcal{T}))^2 \text{ pro všechny } \mathcal{T} \subset \Omega^{p,i}, \right. \\ \left. \mathbf{v}^{p,i}|_{\partial\Omega^{p,i} \cap \Gamma_u^p} = 0 \right\}. \end{aligned} \quad (2.19)$$

Sumační člen v úloze (2.2) se redukuje na $\mathbf{g}^T |\mathbf{T} \mathbf{u}|$ a Lagrangeův multiplikátor $\boldsymbol{\lambda}_t = (\boldsymbol{\lambda}_{t_1}^T, \boldsymbol{\lambda}_{t_2}^T)^T$ na prosté $\boldsymbol{\lambda}_t$. Matice $\mathbb{B}$ a vektory $\mathbf{c}$ a $\boldsymbol{\lambda}$ vypadají takto:

$$\mathbb{B} = \begin{bmatrix} \mathbb{T} \\ \mathbb{N} \\ \mathbb{B}_E \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \mathbf{0} \\ \mathbf{d} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{\lambda} = \begin{bmatrix} \boldsymbol{\lambda}_t \\ \boldsymbol{\lambda}_I \\ \boldsymbol{\lambda}_E \end{bmatrix}. \quad (2.20)$$

3 Řešiče kontaktních úloh

V této kapitole podrobně popíšeme algoritmy SMALBE a MPRGP pro řešení kontaktní úlohy (2.16).

3.1 Minimalizace s rovnostními a nerovnostními vazbami

V této části se zaměříme na popis efektivního algoritmu SMALBE (Semimonotonic Augmented Lagrangian algorithm for Bound and Equality constraints) pro řešení úloh kvadratického programování s rovnostními a nerovnostními vazbami. Předpokládejme úlohu

$$\min_{\mathbf{x} \in \Omega_{BE}} f(\mathbf{x}), \quad (3.1)$$

kde $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \mathbb{A} \mathbf{x} - \mathbf{x}^T \mathbf{b}$, $\mathbf{b}$ a $\mathbf{l}$ jsou n -rozměrné sloupcové vektory, $\mathbb{A}$ je $n \times n$ symetrická pozitivně definitní matice, $\mathbb{B} \in \mathbb{R}^m \times n$ a množina

$$\Omega_{BE} = \{\mathbf{x} \in \mathbb{R}^n : \mathbb{B}\mathbf{x} = \mathbf{o} \text{ a } \mathbf{x} \geq \mathbf{l}\}. \quad (3.2)$$

Připouštíme lineárně závislé řádky v matici $\mathbb{B}$ a hodnoty $l_i = -\infty$. Dále předpokládáme, že $\mathbb{B} \neq \mathbf{0}$ a nemá plnou sloupcovou hodnotu, tj. $\text{Ker} \mathbb{B} \neq \{\mathbf{o}\}$. Zdůrazněme, že i obecnější problémy kvadratického programování mohou být redukovány na úlohu (3.1) pomocí duality nebo vhodné substituce proměnných nebo modifikace funkcionálu f .

Hlavní myšlenkou algoritmu SMALBE je pracovat odděleně s rovnostními a nerovnostními vazbami, tj. pomocná úloha s nerovnostními omezeními bude řešena ve vnitřní smyčce a ve vnější se budou aktualizovat Lagrangeovy multiplikátory pro splnění rovnostní vazby (viz Conn, Gould and Toint [10, 11]). Mezi techniky užívané v algoritmu patří kontrola přesnosti pomocných úloh a pravidlo pro aktualizaci penalizačního parametru.

3.1.1 KKT podmínky a projektovaný gradient

Řešení problému (3.1) je jednoznačné, protože množina Ω_{BE} je neprázdná, uzavřená a konvexní a o funkci f předpokládáme, že je ryze konvexní. Speciální tvar přípustné množiny Ω_{BE} umožňuje použít specifický tvar KKT podmínek, který zcela popisuje jednoznačnost řešení (3.1). Obohacením Lagrangianu o rovnostní vazby

$$\mathbf{L}(\mathbf{x}, \boldsymbol{\lambda}, \varrho) = \frac{1}{2}\mathbf{x}^T \mathbb{A} \mathbf{x} - \mathbf{x}^T \mathbf{b} + \mathbf{x}^T \mathbb{B}^T \boldsymbol{\lambda} + \frac{\varrho}{2} \|\mathbb{B}\mathbf{x}\|^2, \quad (3.3)$$

můžeme jednoduše vyjádřit KKT podmínky pro úlohu (3.1) pomocí jeho gradientu

$$\mathbf{g} = \mathbf{g}(\mathbf{x}, \boldsymbol{\lambda}, \varrho) = \nabla_{\mathbf{x}} \mathbf{L}(\mathbf{x}, \boldsymbol{\lambda}, \varrho) = (\mathbb{A} + \varrho \mathbb{B}^T \mathbb{B})\mathbf{x} - \mathbf{b} + \mathbb{B}^T \boldsymbol{\lambda}. \quad (3.4)$$

Po úpravě dostaneme, že přípustný vektor $\mathbf{x} \in \Omega_{BE}$ je řešením (3.1) tehdy a jen tehdy, když

$$\mathbf{g} \geq \mathbf{o} \text{ a } \mathbf{g}^T(\mathbf{x} - \mathbf{l}) = 0, \quad (3.5)$$

nebo obdobně

$$\mathbf{g}^P = \mathbf{o}, \quad (3.6)$$

kde $\mathbf{g}^P$ je projektovaný gradient (definice viz popis algoritmu MPRGP).

3.1.2 Algoritmus SMALBE

Následující algoritmus je modifikací algoritmu SMALE z [1], kapitola 4.6.1. Základním rozdílem je, že algoritmus SMALBE řeší úlohu s nerovnostními vazbami ve vnitřní smyčce s kontrolou přesnosti realizovanou pomocí Euklidovské normy projektovaného gradientu. Ve srovnání s originálním algoritmem, navrženým Connem, Glouldem and Tointem [10, 11], se algoritmus SMALBE liší v adaptivní kontrole přesnosti představené autory Dostálem, Friedlanderem a Santosem [12] a v řízení regularizačního parametru ϱ , které bylo publikováno prof. Dostálem v [13]. Kompletní algoritmus SMALBE vypadá následovně.

Mějme symetrickou pozitivně definitní matici $\mathbb{A} \in \mathbb{R}^{n \times n}$, $\mathbb{B} \in \mathbb{R}^{m \times n}$ a n -rozměrné vektory $\mathbf{b}, \mathbf{l}$.

Krok 0 {Inicializace.}

Vyber $\eta > 0$, $\beta > 1$, $M > 0$, $\rho_0 > 0$, $\boldsymbol{\lambda}^0 \in \mathbb{R}^m$

opakuj pro $k = 0, 1, 2, \dots$

Krok 1 {Vnitřní iterace s adaptivní kontrolou přesnosti.}

Najdi $\mathbf{x}^k \geq \mathbf{l}$ takové, že

$$\|\mathbf{g}^P(\mathbf{x}^k, \boldsymbol{\lambda}^k, \varrho_k)\| \leq \min\{M\|\mathbb{B}\mathbf{x}^k\|, \eta\} \quad (3.7)$$

Krok 2 {Update Lagrangeových multiplikátorů.}

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \varrho_k \mathbb{B}\mathbf{x}^k \quad (3.8)$$

Krok 3 {Přepočítání ϱ za podmínky, že zvýšení Lagrangiánu není dostatečné.}

jestliže $k > 0$ a $\mathbf{L}(\mathbf{x}^k, \boldsymbol{\lambda}^k, \varrho_k) < \mathbf{L}(\mathbf{x}^{k-1}, \boldsymbol{\lambda}^{k-1}, \varrho_{k-1}) + \frac{\varrho_k}{2}\|\mathbb{B}\mathbf{x}^k\|^2$

$\varrho_{k+1} = \beta\varrho_k$

jinak

$\varrho_{k+1} = \varrho_k$

end jestliže

end opakuj pro

V *Kroku 1* můžeme použít libovolný algoritmus pro minimalizaci ryze konvexní kvadratické funkce vzhledem k nerovnostním omezením do té doby, dokud je zaručena konvergence projektovaného gradientu k 0. Takovým algoritmem je např. algoritmus MPRGP popsáný v následující kapitole.

Lemma 3.1. *Nechť $M > 0$, $\boldsymbol{\lambda} \in \mathbb{R}^m$, $\eta > 0$ a $\varrho \geq 0$. Označme $\{\mathbf{y}^k\}$ libovolnou sekvenci takovou, že $\mathbf{y}^k \geq \mathbf{l}$ a $\mathbf{g}^P(\mathbf{y}^k, \boldsymbol{\lambda}, \varrho)$ konvergují k nulovému vektoru. Potom $\{\mathbf{y}^k\}$ konverguje k jednoznačnému řešení $\hat{\mathbf{x}}$ problému (3.1), nebo existuje index k takový, že*

$$\|\mathbf{g}^P(\mathbf{y}^k, \boldsymbol{\lambda}, \varrho)\| \leq \min\{M\|\mathbb{B}\mathbf{y}^k\|, \eta\}. \quad (3.9)$$

Poznámka V praktické realizaci se používá modifikace algoritmu SMALBE, která pracuje i s omezeními shora, tj. $\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$.

3.2 Minimalizace kvadratické funkce s jednoduchými lineárními omezeními

V této části si ukážeme algoritmus MPRGP na minimalizaci kvadratické funkce na konvexní množině definované jednoduchými lineárními omezeními. Tato omezení plynou z definice množiny $\Lambda(\mathbf{g})$.

Pokud odbočíme od dosavadního značení a zavedeme-li nové, budeme řešit úlohu:

$$\min_{\mathbf{x} \in \Upsilon} f(\mathbf{x}) = \min_{\mathbf{x} \in \Upsilon} \frac{1}{2} \mathbf{x}^T \mathbb{A} \mathbf{x} - \mathbf{x}^T \mathbf{b}, \quad \Upsilon = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}, \quad (3.10)$$

kde symbolem $\leq$ rozumíme nerovnosti mezi odpovídajícími složkami vektorů, $\mathbb{A} \in \mathbb{R}^{n \times n}$ je symetrická, pozitivně definitní matice a $\mathbf{l}, \mathbf{u} \in (\mathbb{R} \cup \{-\infty, \infty\})^n$. V tomto případě existuje právě jedno řešení úlohy (3.10).

Algoritmus MPRGP kombinuje metodu sdružených gradientů s vhodnou strategií, která se stará o splnění jednoduchých omezení. Nyní krátce popíšeme základní myšlenky a vlastnosti.

Nejprve si zavedeme několik označení. Nechť $\mathcal{N} = \{1, 2, \dots, n\}$ je množina všech indexů. Projekci vektoru $\mathbf{x}$ na množinu Υ definujeme takto:

$$P_{\Upsilon}(\mathbf{x}) = \mathbf{y} : \mathbf{y}_i = \max\{\mathbf{l}_i, \min\{\mathbf{u}_i, \mathbf{x}_i\}\}. \quad (3.11)$$

Gradient $\mathbf{g} := \mathbf{g}(\mathbf{x})$ funkce f v bodě $\mathbf{x} \in \mathbb{R}^n$ je roven

$$\mathbf{g} = \mathbb{A} \mathbf{x} - \mathbf{b}. \quad (3.12)$$

Řešení $\bar{\mathbf{x}}$ úlohy (3.10) musí splňovat tyto Karush-Kuhn-Tuckerovy podmínky:

$$\left. \begin{aligned} \bar{\mathbf{x}}_i = \mathbf{l}_i &\Rightarrow \bar{\mathbf{g}}_i \geq 0, \\ \bar{\mathbf{x}}_i = \mathbf{u}_i &\Rightarrow \bar{\mathbf{g}}_i \leq 0, \\ \mathbf{l}_i < \bar{\mathbf{x}}_i < \mathbf{u}_i &\Rightarrow \bar{\mathbf{g}}_i = 0, \quad i \in \mathcal{N}, \quad \bar{\mathbf{g}}_i = \mathbf{g}_i(\bar{\mathbf{x}}). \end{aligned} \right\} \quad (3.13)$$

Množinu všech indexů $\mathcal{A}(\mathbf{x})$, které splňují $\mathbf{x}_i = \mathbf{l}_i$ nebo $\mathbf{x}_i = \mathbf{u}_i$ nazveme *aktivní množinou*:

$$\mathcal{A}(\mathbf{x}) = \{i \in \mathcal{N} : (\mathbf{x}_i = \mathbf{l}_i) \vee (\mathbf{x}_i = \mathbf{u}_i)\} \quad (3.14)$$

a její doplněk $\mathcal{F}(\mathbf{x})$ do $\mathcal{N}$ nazveme *volnou množinou*. Karush-Kuhn-Tuckerovy podmínky lze napsat pomocí částí gradientu $\mathbf{g}$. Proto definujeme *volný gradient* $\boldsymbol{\varphi}$ a *uříznutý gradient* $\boldsymbol{\beta}$:

$$\begin{aligned} \boldsymbol{\varphi}_i(\mathbf{x}) &= \mathbf{g}_i(\mathbf{x}) \text{ pro } i \in \mathcal{F}(\mathbf{x}), & \boldsymbol{\varphi}_i(\mathbf{x}) &= 0 \text{ pro } i \in \mathcal{A}(\mathbf{x}), \\ \boldsymbol{\beta}_i(\mathbf{x}) &= 0 \text{ pro } i \in \mathcal{F}(\mathbf{x}), & \boldsymbol{\beta}_i(\mathbf{x}) &= \mathbf{g}_i^{\pm}(\mathbf{x}) \text{ pro } i \in \mathcal{A}(\mathbf{x}), \end{aligned} \quad (3.15)$$

kde $\mathbf{g}_i^{\pm}(\mathbf{x}) = \min\{\mathbf{g}_i, 0\}$ pro $\mathbf{x}_i = \mathbf{l}_i$, respektive $\mathbf{g}_i^{\pm}(\mathbf{x}) = \max\{\mathbf{g}_i, 0\}$ pro $\mathbf{x}_i = \mathbf{u}_i$. Je vidět, že Karush-Kuhn-Tuckerovy podmínky jsou splněny právě tehdy, když *projektovaný gradient* $\boldsymbol{\nu}(\mathbf{x}) := \boldsymbol{\varphi}(\mathbf{x}) + \boldsymbol{\beta}(\mathbf{x})$ je roven nule.

Než přistoupíme k popisu MPRGP, nastíníme jednoduché myšlenky Polyakova algoritmu, který byl jeden z důvodů vzniku MPRGP. Algoritmus, který navrhl Polyak, využívá

efektivní metody sdružených gradientů k minimizaci f , pokud se nemění aktivní množina. Tato minimizace vede buď k nalezení minima na konvexní množině definované aktivními vazbami, nebo vygeneruje bod mimo množinu Υ . V prvním případě algoritmus odejme z aktivní množiny indexy, ve kterých jsou narušeny Karush-Kuhn-Tuckerovy podmínky a redukuje tak aktivní množinu. V druhém případě je krok sdružených gradientů zkrácen tak, aby vedl na hranici množiny Υ , což vede k rozšíření aktivní množiny obvykle o jeden index. Při každé redukci aktivní množiny je funkce f minimizovaná pro stav, ve kterém $\mathbf{x}_i = \mathbf{l}_i$, nebo $\mathbf{x}_i = \mathbf{u}_i$ pro všechny indexy aktivní množiny. Jelikož Polyakovův algoritmus v každém kroku snižuje hodnotu funkce f , tento stav nikdy v dalších krocích nenastane. Protože je počet kombinací omezení shora a zdola ve všech indexech aktivní množiny konečný a počet všech různých aktivních množin je rovněž konečný, plyne z finitnosti metody sdružených gradientů, že počet kroků Polyakova algoritmu je také konečný.

Algoritmus MPRGP řeší úlohu (3.10), na rozdíl od Polyakova algoritmu, připouští zopakování některých stavů. Je však rovněž finitní a je rychlejší. Zahrnuje tři typy kroků, z nichž každý nakládá jiným způsobem s aktivní množinou $\mathcal{A}(\mathbf{x})$, a obsahuje rozhodovací kritérium, jenž určuje, který z těchto kroků vybrat.

Krok sdružených gradientů: nová hodnota $\mathbf{x}$ je definována takto

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_{cg} \mathbf{p}^k, \quad \alpha_{cg} = \frac{\mathbf{g}(\mathbf{x}^k)^T \mathbf{p}^k}{(\mathbf{p}^k)^T \mathbb{A} \mathbf{p}^k}, \quad (3.16)$$

kde $\mathbf{p}^k$ je sdružený gradient. Pokud se jedná o počáteční krok začínající v bodě $\mathbf{x}^s$, je $\mathbf{p}^s = \boldsymbol{\varphi}(\mathbf{x}^s)$. V následujících krocích je sdružený gradient počítán rekurentně:

$$\mathbf{p}^{k+1} = \boldsymbol{\varphi}(\mathbf{x}^k) - \gamma \mathbf{p}^k, \quad \gamma = \frac{\boldsymbol{\varphi}(\mathbf{x}^k)^T \mathbb{A} \mathbf{p}^k}{(\mathbf{p}^k)^T \mathbb{A} \mathbf{p}^k}. \quad (3.17)$$

Základní vlastností sdružených gradientů $\mathbf{p}^s, \dots, \mathbf{p}^k$ je jejich $\mathbb{A}$ -ortogonalita, tj. $(\mathbf{p}^i)^T \mathbb{A} \mathbf{p}^j = 0$, $i, j \in \{s, \dots, k\}$, $i \neq j$. Navíc je známo, že

$$f(\mathbf{x}^{k+1}) = \min_{\mathbf{y} \in \text{Span}\{\mathbf{p}^s, \dots, \mathbf{p}^k\}} f(\mathbf{x}^s + \mathbf{y}). \quad (3.18)$$

V případě, že se nemění aktivní množina, algoritmus využívá metodu sdružených gradientů k minimizaci f .

Expanze aktivní množiny: při postupných krocích sdružených gradientů je možné, že následující krok $\mathbf{x}^{k+1} \notin \Upsilon$. Proveďte se krok zvaný expanze a následně se restartuje metoda sdružených gradientů. Expanze se skládá ze dvou částí. Nejprve se provede krok ve směru sdružených gradientů zkrácený na přípustnou množinu, která se tímto krokem zpravidla o jeden index rozšíří:

$$\mathbf{x}^{k+\frac{1}{2}} = \mathbf{x}^k - \alpha_{\Upsilon} \mathbf{p}^k, \quad \alpha_{\Upsilon} = \max\{\alpha : \mathbf{x}^k - \alpha \mathbf{p}^k \in \Upsilon\}. \quad (3.19)$$

Následně se provede projektovaný krátký krok ve směru volného gradientu:

$$\mathbf{x}^{k+1} = P_{\Upsilon} \left(\mathbf{x}^{k+\frac{1}{2}} - \bar{\alpha} \boldsymbol{\varphi} \left(\mathbf{x}^{k+\frac{1}{2}} \right) \right), \quad \bar{\alpha} \in (0, \|\mathbb{A}^{-1}\|). \quad (3.20)$$

Tento krok, stejně jako předchozí, snižuje hodnotu minimizované funkce, což zaručuje jeho délka viz [19]. Na rozdíl od Polyakova algoritmu to vede k většímu rozšíření aktivní množiny.

Redukce aktivní množiny-proportioning: předchozí dva typy kroků využívaly ke snížení hodnoty funkce f směr $\varphi(\mathbf{x}^k)$ volné části projektovaného gradientu $\nu(\mathbf{x}^k)$. Krok zvaný proportioning minimizuje funkci f ve směru uříznutého gradientu $\beta(\mathbf{x}^k)$:

$$\begin{aligned} \mathbf{x}^{k+1} &= \mathbf{x}^k - \alpha_{prop} \beta(\mathbf{x}^k), & \alpha_{prop} &= \min\{\alpha_{\Upsilon}, \alpha_{cg}\}, \\ \alpha_{\Upsilon} &= \max\{\alpha : \mathbf{x}^k - \alpha \beta(\mathbf{x}^k) \in \Upsilon\}, & \alpha_{cg} &= \frac{\mathbf{g}(\mathbf{x}^k)^T \beta(\mathbf{x}^k)}{\beta(\mathbf{x}^k)^T \mathbb{A} \beta(\mathbf{x}^k)}. \end{aligned} \quad (3.21)$$

Jelikož se mohou změnit pouze ty složky vektoru $\mathbf{x}$, které přísluší aktivní množině, vede proportioning k redukci aktivní množiny.

O tom, kterou část $\varphi(\mathbf{x}^k)$ nebo $\beta(\mathbf{x}^k)$ projektovaného gradientu $\nu(\mathbf{x}^k)$ použijeme k minimizaci v následujícím kroku, rozhoduje kritérium:

$$\|\beta(\mathbf{x}^k)\|^2 \leq \Gamma^2 \tilde{\varphi}(\mathbf{x}^k)^T \varphi(\mathbf{x}^k), \quad (3.22)$$

kde

$$\tilde{\varphi}_i(\mathbf{x}) = \max \left\{ \frac{\mathbf{x}_i - \mathbf{u}_i}{\bar{\alpha}}, \min \left\{ \frac{\mathbf{x}_i - \mathbf{l}_i}{\bar{\alpha}}, \varphi_i(\mathbf{x}) \right\} \right\}. \quad (3.23)$$

Pravá strana (3.22) závisí jak na volném gradientu, tak na jeho části, která může být použita v druhé části kroku expanze aktivní množiny. Konstanta $\Gamma \geq 0$ udává, jak přesně minimizovat f , pokud se nezmenšuje aktivní množina. Při volbě $\Gamma = 0$ se algoritmus chová stejně jako Polyakovův algoritmus, ideální volba je $\Gamma = 1$. Pokud je kritérium (3.22) splněno, nazýváme $\mathbf{x}^k$ proporcionální a použijeme krok sdružených gradientů nebo krok expanze aktivní množiny, pokud by krok sdružených gradientů vedl k narušení podmínky $\mathbf{x}^{k+1} \in \Upsilon$. V případě, že kritérium splněno není, použijeme krok redukce aktivní množiny. Takto vypadá stručný zápis algoritmu, kde konstanta ϵ říká, jakou požadujeme přesnost:

Krok 0 {Inicializace.}

Nechť $k = 0$, $\mathbf{g} = \mathbb{A}\mathbf{x}^0 - \mathbf{b}$, $\mathbf{p} = \varphi(\mathbf{x}^0)$

dokud $\|\nu(\mathbf{x}^k)\| > \epsilon$

jestliže $\|\beta(\mathbf{x}^k)\|^2 \leq \Gamma^2 \tilde{\varphi}(\mathbf{x}^k)^T \varphi(\mathbf{x}^k)$

Krok 1 { $\mathbf{x}^k$ je proporcionální. Zkus sdružené gradienty.}

$\alpha_{cg} = \mathbf{g}^T \mathbf{p} / \mathbf{p}^T \mathbb{A} \mathbf{p}$, $\mathbf{y} = \mathbf{x}^k - \alpha_{cg} \mathbf{p}$

$\alpha_{\Upsilon} = \max\{\alpha : \mathbf{x}^k - \alpha \mathbf{p} \in \Upsilon\}$

jestliže $\alpha_{cg} \leq \alpha_{\Upsilon}$

Krok 2 {Krok sdružených gradientů.}

$\mathbf{x}^{k+1} = \mathbf{y}$, $\mathbf{g} = \mathbf{g} - \alpha_{cg} \mathbb{A} \mathbf{p}$

$\gamma = \varphi(\mathbf{y})^T \mathbb{A} \mathbf{p} / \mathbf{p}^T \mathbb{A} \mathbf{p}$, $\mathbf{p} = \varphi(\mathbf{y}) - \gamma \mathbf{p}$

jinak

Krok 3 {Expanze aktivní množiny.}

$\mathbf{nx}_1 = \mathbf{x}^k - \alpha_{\Upsilon} \mathbf{p}$, $\mathbf{nx}_2 = P_{\Upsilon}(\mathbf{x}^k - \alpha_{cg} \mathbf{p})$,

jestliže $f(\mathbf{nx}_1) \leq f(\mathbf{nx}_2)$

$\mathbf{x}^{k+\frac{1}{2}} = \mathbf{n}\mathbf{x}_1, \mathbf{g} = \mathbf{g} - \alpha_{\mathbf{r}}\mathbb{A}\mathbf{p}$

jinak

$\mathbf{x}^{k+\frac{1}{2}} = \mathbf{n}\mathbf{x}_2, \mathbf{g} = \mathbb{A}\mathbf{x}^{k+\frac{1}{2}} - \mathbf{b}$

konec jestliže

$\mathbf{x}^{k+1} = P_{\mathbf{r}}\left(\mathbf{x}^{k+\frac{1}{2}} - \bar{\alpha}\varphi\left(\mathbf{x}^{k+\frac{1}{2}}\right)\right)$

$\mathbf{g} = \mathbb{A}\mathbf{x}^{k+1} - \mathbf{b}, \mathbf{p} = \varphi(\mathbf{x}^{k+1})$

konec jestliže

jinak

Krok 4 $\{\textit{Proportioning - redukce aktivní množiny.}\}$

$\mathbf{d} = \boldsymbol{\beta}(\mathbf{x}^k), \alpha_{cg} = \mathbf{g}^T\mathbf{d}/\mathbf{d}^T\mathbb{A}\mathbf{d}$

$\alpha_{\mathbf{r}} = \max\{\alpha : \mathbf{x}^k - \alpha\mathbf{d} \in \mathbf{r}\}, \alpha_{prop} = \min\{\alpha_{\mathbf{r}}, \alpha_{cg}\}$

$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_{prop}\mathbf{d}, \mathbf{g} = \mathbb{A}\mathbf{x}^{k+1} - \mathbf{b}, \mathbf{p} = \varphi(\mathbf{x}^{k+1})$

konec jestliže

$k = k + 1$

konec dokud

Krok 5 $\{\textit{Konec.}\}$

$\mathbf{x} = \mathbf{x}^k.$

Tento algoritmus lze zobecnit i pro případ separovatelných konvexních omezení [20], což umožňuje řešit 3D-kontaktní úlohy se třením

4 Rozhraní COMSOL MATLAB

Rozhraní mezi COMSOLEM a OOSOLEM je složeno ze dvou rozhraní COMSOL MATLAB a MATLAB OOSOL. Druhé v pořadí již bylo implementované dříve Doc. Vondrákem a Doc. Kozubkem z Katedry aplikované matematiky, VŠB-TU Ostrava. Z tohoto důvodu se v následujícím zabývám pouze implementací rozhraní mezi systémy MATLAB a COMSOL zaměřeném na řešení kontaktních úloh.

4.1 Co je to COMSOL

Systém COMSOL Multiphysic vytvořila švédská společnost COMSOL®. Tento systém umožňuje modelování a simulaci fyzikálních procesů popsaných parciálními diferenciálními rovnicemi (PDR) pomocí metody konečných prvků. Verze 3.3 má rozšířenou knihovnu materiálů s více jak 2500 položkami roztříděnou do skupin podle typů a fyzikálních vlastností. V knihovně můžeme materiály vyhledávat podle názvu nebo podle označení v normě DIN a UNS. V této verzi je možné importovat datové soubory ve standardu CHEMKIN®, který umožňuje číst formáty dat používané v chemickém průmyslu v oblasti spalování, chemických reakcí v reaktorech a v oblasti chemických procesů v atmosféře. COMSOL Multiphysics dále nabízí zlepšený CAD Import Module s novou možností opravit chyby v importované geometrii např. mezery, přesahy nebo odstranit malá zaoblení, plošky a krátké hrany. Uživatelé mohou interaktivně připravit model pro vhodné generování FEM sítě a následnou FEM analýzu. V knihovně vzorových modelů - Model Library - je umístěna řada vzorově řešených modelových úloh.

CAD Import Module je dále obohacen o zlepšený import sestav geometrických modelů z mnoha významných CAD systémů jako je SolidWorks®, Solid Edge®, Autodesk Inventor® a mnoho dalších. Modul COMSOL Multiphysics 3.3 je obohacen o mnoho nástrojů, které zajistí matematickou a fyzikální definici těles a ploch tak, aby importovaná geometrie splňovala požadavky pro následnou FEM analýzu. Uživatel má pod kontrolou ořezání hran jednotlivých ploch, popřípadě vyplnění mezer ve spojích čar a hran.

COMSOL Multiphysics je dynamicky se rozvíjející program pro modelování a simulaci fyzikálních procesů, které jsou popsány pomocí parciálních diferenciálních rovnic. Tato metoda je známá také pod názvem EBM - Equation Based Modeling, kde základem řešení je knihovna parciálních diferenciálních rovnic popisujících vybrané úlohy. Uživatel není omezen aplikačními režimy z této knihovny, ale díky otevřenosti systému může volně definovat své vlastní úlohy. Většina úloh v COMSOL Multiphysics je řešena metodou konečných prvků. Program je určen k modelování a k simulaci úloh z oblasti strojírenství, elektrotechniky, chemie a z různých oblastí fyziky, jako je například přenos tepla, dynamika tekutin, akustika, vibrace, elektromagnetismus, šíření vln, strukturální mechanika, difúze, chemické reakce, geofyzika, biologie, palivové články, polovodiče, vedení tepla, kvantová mechanika a mnoho dalších. COMSOL Multiphysics usnadňuje pochopení řady fyzikálních a technických procesů díky názorné grafice, snadné změně vstupních parametrů a možnosti simulovat dané úlohy. Řešení je možné vidět v krátké době z několika úhlů pohledu a efektivně tak najít optimální variantu. K největším přednostem patří srozumí-

telnost, otevřenost, schopnost řešit více fyzikálních jevů současně (multifyzikální úlohy) a velké množství předzpracovaných typových příkladů. Ovšem jeho největší výhodou je napojení na MATLAB na bázi Klient Server.

4.2 Co je to MATLAB

MATLAB (MATrix LABoratory = maticová laboratoř) je výkonné interaktivní prostředí pro vědecké výpočty. Spojuje technické výpočty s vizualizací dat a programovací jazyk v jednom prostředí. Společně s velkým množstvím dostupných modulů tak vytváří ideální prostředek pro inženýry, vědce, matematiky a učitele při řešení problémů z mnoha oblastí. Aplikaci vyvíjí firma MathWorks a dodává ji s celou řadou rozšíření a toolboxů.

Matlab je používán po celém světě. Existuje proto široká komunita, která s ním pracuje, včetně množství dokumentace a ukázkových příkladů. Firma MathWorks na svých stránkách provozuje Matlab Central, což můžeme chápat jako otevřené rozhraní pro výměnu informací mezi uživateli Matlabu a Simulinku.

V prostředí Matlabu je řešení technických problémů mnohem snazší, než v jiných programovacích jazycích popřípadě vývojových prostředích, protože obsahuje velké množství již předdefinovaných funkcí, které usnadňují vývoj. K těmto funkcím patří například přímá podpora komplexních čísel, sofistikované řešiče soustav lineárních algebraických rovnic, práce s velkými řídkými maticemi nebo řešení soustav obyčejných diferenciálních rovnic.

4.3 Vytváření modelů v COMSOLu

Program COMSOL používáme pro vytvoření konkrétních modelů, které se poté dále zpracovávají implementovanými programy v systému MATLAB. Celý další popis COMSOLu je zaměřen na řešení stacionárních kontaktních problémů.

Nejdříve si zvolíme typ úlohy, který chceme řešit (např. rovinná deformace nebo napjatost, osově symetrická úloha nebo plná 3D úloha). Potom nakreslíme geometrii úlohy pomocí vestavěného kreslicího modulu nebo ji nainportujeme z jiného systému pomocí Import Cad Module.

Dále předepíšeme okrajové podmínky. Každá hranice objektu je defaultně nastavena jako Neumannova hranice (nulové působící síly). Toto nastavení si samozřejmě upravíme podle našich přání a požadavků. Můžeme si vybrat z těchto možností:

- *Free* - nulové působící síly
- *Fixed* - nulové posunutí ve všech směrech
- *Roller* - nulové posunutí v normálovém směru
- *Prescribed displacement* - předepsané posunutí
- *Symmetry plane* - rovina symetrie
- *y-z, x-z, x-y symmetry plane* - rovina symetrie je rovina y-z, x-z, resp. x-y

- *Antisymmetry plane* - antisymetrie (nulová posunutí v tečném směru)
- *y-z, x-z, x-y antisymmetry plane* - rovina antisymetrie je rovina y-z, x-z, resp. x-y

U hranic můžeme dále nastavit působení povrchových sil v libovolném směru.

V nastavení pro jednotlivé části tělesa (**Subdomain Settings**) nastavíme hodnoty pro Youngův modul, Poissonovu konstantu, působení objemových sil a mnoho dalších nastavení podle naší potřeby.

Zde můžeme také nastavit kontaktní páry (hranice v kontaktu). U těchto kontaktních hranic zadáváme, která je *master* a která *slave*. Dále si zvolíme, zda chceme kontakt se třením nebo bez. Coulombův model tření je realizován pomocí kontaktních prvků a penalty podobně jako v ANSYSu a tudíž s obrovskými problémy při řešení kontaktních úloh s více volnými tělesy. Naše katedra vyvinula několik algoritmů, které tyto úlohy dokáží hravě zvládnout.

Po provedení předchozích nastavení jsme schopni vygenerovat síť (mesh). Síť vytvoříme pomocí příkazu **Initialize Mesh**, který nám vygeneruje trojúhelníkovou síť ve 2D a ve 3D síť ze čtyřstěnů. Takhle vytvořenou síť, ať se jedna o trojúhelníky nebo čtyřstěny, můžeme dále zjemňovat, buď jako celek pomocí příkazu **Refine Mesh**, nebo pouze vybranou část pomocí příkazu **Refine Selection**. Kromě trojúhelníků a čtyřstěnů je možné použít obdélníky, bricky, prizmy a další prvky.

Po provedení všech předchozích kroků máme hotový model kontaktní úlohy. Tento model uložíme pod názvem, který nám vyhovuje, jako MATLABovský „m“ soubor.

Při jeho spuštění se nám vygeneruje struktura *fem*, ve které jsou uloženy veškeré informace o modelu, které potřebujeme pro další práci.

4.4 Struktura FEM

Struktura FEM obsahuje kompletní definici řešené kontaktní úlohy. Její položky jsou *version*, *appl*, *sdim*, *frame*, *shape*, *sshape*, *border*, *units*, *equ*, *bnd*, *pnt*, *var*, *elemmiph*, *eleminitmiph*, *draw*, *geom* a *mesh*. Dále se zaměříme pouze na popis těch položek, které jsou pro nás důležité.

4.4.1 Položka version

Tato položka má v sobě uloženou verzi COMSOLu, kterou byl model vytvořen, datum vytvoření atd.

4.4.2 Položka appl

Položka *appl* obsahuje veškerou fyziku modelu. To znamená, že jsou zde uloženy hodnoty Youngova modulu pružnosti, Poissonova konstanta, materiálové vlastnosti jednotlivých objektů, působící síly, okrajové podmínky atd.

4.4.3 Položka **sdim**

Zde jsou uvedeny názvy prostorových proměnných (souřadnic) a jejich počet.

4.4.4 Položka **shape** a **sshape**

Slouží k definici tvarových funkcí (volba typu elementu).

4.4.5 Položka **border**

Zpřístupňuje možnost zadávání okrajových podmínek na vnitřní hranice mezi podoblastmi.

4.4.6 Položka **units**

COMSOL podporuje následující jednotkové systémy:

- systém SI, mezinárodní systém jednotek. Tento systém je defaultně nastaven
- systém CGSA, CGS systém používá centimetr, gram a sekundu jako základní jednotky délky, hmotnosti a času. Zbylé základní jednotky jsou ze systému SI.
- Elektromagnetické jednotky (EMU). Tento systém je založen na Ampérově zákoně.
- Elektrostatické jednotky (ESU). Je založen na Coulombově zákoně pro síly mezi dvěma navzájem se ovlivňujícími body.
- systém MPa, používá se v mechanice v případě, že Newton a Megapascal jsou přirozeně odvozené jednotky pro sílu a tlak.

COMSOL pamatuje i na jiné systémy jednotek, hlavně pro anglosaské země.

4.4.7 Položka **bnd**

Slouží k definování kontaktních párů. Rozlišují se dva typy: *Contact* nebo *Identity*, kde *Contact* označuje kontaktní hranice a *Identity* udává, které hranice jsou totožné, to jest mají být slepeny lepicí podmínkou, položka **name** udává jméno tohoto páru a položky **src** a **dst** udávají indexy master a slave hranice.

4.4.8 Položka **var**

Pro mnoho aplikačních módů generuje systém dodatečné proměnné. Jejich jména a definice jsou zadány defaultně.

4.4.9 Položka **draw**

Popisuje geometrii objektu pro vykreslování. Generuje se pokaždé při opuštění kreslicího módu, byla-li změněna geometrie. Obsahuje tedy veškeré geometrické informace, jako jsou například názvy jednotlivých objektů, jejich rozměry, typ atd.

4.4.10 Položka geom

Analyzovaná geometrie se ukládá do položky `geom`. Analýza se provádí pomocí funkcí `geomcsg` a `geomgroup`, kde funkce `geomcsg` rozdělí jednu oblast na více podoblastí a funkce `geomgroup` rozdělí oblast na více samostatných (fyzicky oddělených) oblastí, které je pak nutno slepit lepicí podmínkou.

4.4.11 Položka mesh

Položka `mesh` je opět struktura, která obsahuje kompletní informace o konečně prvkové síti pro modelovanou geometrii. Obsahuje položky `P`, `E` a `T`, jejichž význam je následující:

- V položce `P` jsou uloženy informace o souřadnicích jednotlivých uzlů. Ve 2D je $P = [X, Y]^T$, kde $X(n)$, $Y(n)$ jsou souřadnice n -tého uzlu sítě. Ve 3D je pak $P = [X, Y, Z]^T$, kde $X(n)$, $Y(n)$, $Z(n)$ jsou souřadnice n -tého uzlu sítě.
- V položce `T` jsou uloženy informace o jednotlivých objemových elementech (trojúhelníky, obdélníky, čtyřstěny, šestistěny, atd.). Ve 2D je $T = [I, J, K, S]^T$, kde $I(n)$, $J(n)$, $K(n)$ jsou indexy vrcholu n -tého trojúhelníkového elementu a $S(n)$ je index podoblasti, do které tento element patří. Ve 3D je pak $T = [I, J, K, L, S]^T$, kde $I(n)$, $J(n)$, $K(n)$, $L(n)$ jsou indexy vrcholu n -tého čtyřstěnného elementu (jehlanu) a $S(n)$ je index podoblasti, do které tento element patří. Podobně se to ukládá i pro obdélníky, kvádry a elementy vyššího stupně
- V položce `E` máme uloženy informace o hraničních elementech (ploškách). Každý sloupec popisuje jeden hraniční element. Ve 2D to vypadá následovně: $E = [I, J, \alpha_1, \alpha_2, L, M, N]^T$, kde $I(n)$, $J(n)$ jsou indexy vrcholů n -tého hraničního elementu (úsečky), $\alpha_1(n)$, $\alpha_2(n)$ souvisejí s parametrizací části hranice $L(n)$ popsané racionální Bezierovou křivkou a $M(n)$, $N(n)$ jsou indexy levé a pravé sousední podoblasti pro vnitřní části hranice, pro vnější části hranice je jeden z indexů roven 0. Ve 3D to vypadá takto: $E = [I, J, K, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, L, M, N]^T$, kde $I(n)$, $J(n)$, $K(n)$ jsou indexy vrcholů n -tého hraničního elementu (trojúhelníku), $\alpha_i(n)$ souvisejí s parametrickým vyjádřením části hranice $L(n)$ popsané racionální Bezierovou plochou a $M(n)$, $N(n)$ jsou indexy levé a pravé sousední podoblasti pro vnitřní části hranice vzhledem k orientaci hran hraničního elementu, pro vnější části hranice je jeden z indexů roven 0, pro $M(n) < N(n)$ jsou indexy uzlů $I(n)$, $J(n)$, $K(n)$ uspořádány tak, že podle pravidla pravé ruky palec ukazuje z podoblasti s indexem $M(n)$ do podoblasti s indexem $N(n)$, pro $M(n) > N(n)$ směřuje z $N(n)$ do $M(n)$

Pomocí této struktury jsme schopni sestavit objekty z podmínky nepronikání a ze zákona tření.

4.4.12 Funkce Meshextend

Další důležitou funkcí v COMSOLu je funkce *meshextend*, která např. vytváří mapování uzlů na stupně volnosti.

4.4.13 Funkce Xmeshinfo

Funkce *xmeshinfo* poskytuje informace o počtu elementů, uzlů a stupňů volnosti. Pomocí ní získáme také mapování uzlů na stupně volnosti a naopak.

4.4.14 Funkce Assemble

Funkce *assemble* sestaví matici tuhosti $\mathbb{K}$, vektor zatížení $\mathbf{f}$ a matici $\mathbb{B}_e$ z rovnostních omezení (Dirichletovy a lepící podmínky) pro námi zvolenou úlohu. Zbývající objekty z podmínek nepronikání a zákona tření se sestávají ve vytvořeném rozhraní.

4.4.15 Funkce Geominfo

Tato funkce nám vrací informace o geometrii. Můžeme se pomocí ní doptat na dimenzi úlohy, na informace o dekompozici oblasti apod.

4.5 Funkce z rozhraní COMSOL MATLAB

Hlavním cílem bylo vytvořit rozhraní, které umožní aplikovat na modely vytvořené v COMSOLu algoritmy vyvíjené katedrou aplikované matematiky. Toto rozhraní sestává z následujících funkcí.

4.5.1 Funkce bc_contact_pair2d

Vstupním argumentem je nám již známá struktura *fem* a výstupní argumenty jsou objekty: $\mathbb{N}$, $\mathbb{T}$, $\mathbf{d}$, $\mathbf{g}$.

V této funkci procházíme jednotlivé kontaktní páry a pro každý pár určíme uzly na master hranici a odpovídající povrchové elementy na slave hranici, což jsou elementy, do kterých směřují normály sestrojené v master uzlech.

Mezery mezi tělesy se určí jako vzdálenosti master uzlů a odpovídajících kontaktních plošek.

Matice $\mathbb{N}$ a $\mathbb{T}$ se sestaví jako projekce posunutí do normálového, resp. tečného směru.

4.5.2 Funkce bc_contact_pair3d

Funkce *bc_contact_pair3d* je implementovaná podobně jako funkce *bc_contact_pair2d*, ale pro 3D.

4.5.3 Funkce cpdata

Funkce `cpdata` má za úkol vytvořit všechny potřebné matice a vektory, které potřebujeme pro práci s algoritmy na řešení kontaktních úloh.

Vstupním parametrem je opět struktura *fem* a výstupním parametrem je struktura *data*, která obsahuje všechny objekty (matice a vektory) z algebraické formulace (2.16).

Popis samotné funkce je velice jednoduchý. Zjistíme si prostorovou dimenzi úlohy. Zavoláme funkci *assemble* pro výpočet matic a vektorů $\mathbb{K}$, $\mathbf{f}$ a $\mathbb{B}_e$. Poté podle dimenze úlohy voláme funkci buď pro 2D *bc_contact_pair2d*, nebo pro 3D *bc_contact_pair3d* a sestavíme matice a vektory $\mathbb{N}$, $\mathbf{d}$, $\mathbb{T}$, $\mathbf{g}$. Nakonec provedeme výpočet zbývajících objektů pomocí funkce *gen_lu*. Tímto úkol této metody končí.

4.5.4 Funkce gen_lu

Funkce *gen_lu* sestavuje objekty pro efektivní násobení maticí $\mathbb{K}^\dagger$ pomocí *lu* rozkladu. Nejdříve se provede přeuspořádání matice $\mathbb{K}$ pomocí přeuspořádávajícího algoritmu *symamd* (symmetric approximate minimum degree permutation). Na přeuspořádanou matici $\mathbb{K}$ zavoláme *lu* rozklad bez pivotizace. Na diagonále matice $\mathbb{U}$ budou nuly na místech odpovídajícím lineárně závislým řádkům, počet těchto lineárně závislých řádků je dimenze jádra, jehož bázi lze jednoduše dopočítat.

5 Příklady

V této kapitole budeme ilustrovat funkčnost vytvořeného rozhraní společně s algoritmy pro řešení kontaktních úloh. Vše ukážeme na modelových úlohách Hertzova typu ve 2D a 3D.

5.1 Hertzova úloha ve 2D

Hertzovu úlohu definujeme následovně: předpokládejme nekonečně dlouhý ocelový válec položený na hliníkové podložce (tato podložka je výseč válce s nekonečným poloměrem), obě tělesa jsou elastické. Válec je podroben bodové zátěži podél celého vrchu (viz obr. 2). Cílem je nalezení rozložení kontaktního napětí a délky kontaktu mezi podložkou a válcem. Analytické řešení kontaktního napětí jako funkce x -ové souřadnice vypadá následovně:

$$P = \sqrt{\frac{F_n E}{2\pi R} \times \left(1 - \left(\frac{x}{a}\right)^2\right)} \quad (5.1)$$

a délka kontaktu a je dána vztahem:

$$a = \sqrt{\frac{8F_n R}{\pi E}}, \quad (5.2)$$

kde F_n je působící bodová síla, E je svázaný modul pružnosti a R je svázaný poloměr.

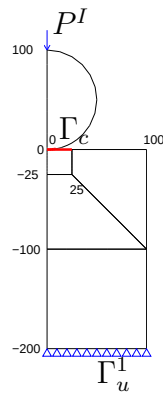
Tyto veličiny jsou definovány následovně:

$$E = \frac{2E_1 E_2}{E_2(1 - v_1^2) + E_1(1 - v_2^2)}, \quad (5.3)$$

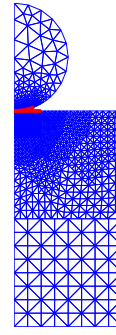
$$R = \lim_{R_2 \rightarrow \infty} \frac{R_1 R_2}{R_1 + R_2} = R_1. \quad (5.4)$$

Tato úloha vede ve 2D na úlohu rovinné deformace s kontaktem (viz obr. 2). Na obrázku je zobrazena pouze polovina geometrie z důvodu symetrie. Hranice Γ_u^1 znázorňuje hranici s Dirichletovou podmínkou, hranice Γ_c tvoří kontaktní pár a P^I je síla působící v bodě $(0, 100)$ o hodnotě 35kN. Zobrazené souřadnice jsou uvedeny v milimetrech. Na obrázku 3 je znázorněna vygenerovaná trojúhelníková síť. V místě předpokládaného kontaktu je síť hustší pro dosažení přesnějšího výpočtu. Obsahuje 2421 trojúhelníkových elementů a 1355 uzlů. Z toho 279 uzlů leží na povrchu a 48 leží v kontaktní zóně. Z toho 16 patří k master hranici a 32 ke slave hranici.

Při výpočtu kontaktního problému používáme algoritmus SMALBE. Jeho parametry byly zvoleny následovně: $\eta = 0.01$, $\beta = 10$, $M = 1$, $\rho_0 = 0.1$, $\lambda^0 = \mathbf{0}$, $\varepsilon = 10^{-4}$, kde ε je přesnost v ukončovací podmínce algoritmu SMALBE. Ostatní parametry viz kapitola 3.1.2.



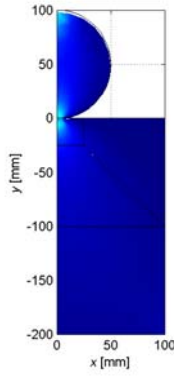
Obrázek 2: Geometrie úlohy v mm.



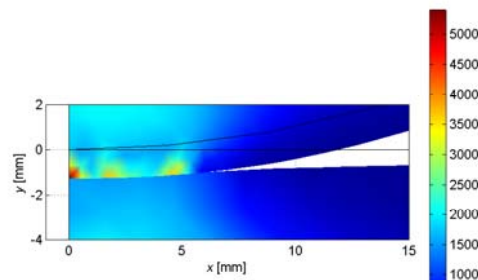
Obrázek 3: Užitá trojúhelníková síť

5.1.1 2D Hertzova úloha bez tření

Dosažené výsledky jsou zobrazeny na obrázcích 4 až 9. Obrázek 4 nám zobrazuje rozložení von Misesova napětí v MPa a deformace těles v mm. Obrázek 5 je přiblížení obrázku 4 v okolí kontaktní zóny. Na obrázku 6 vidíme rozložení kontaktního napětí a obrázek 7 je opět přiblížení obrázku 6 v okolí kontaktní zóny.



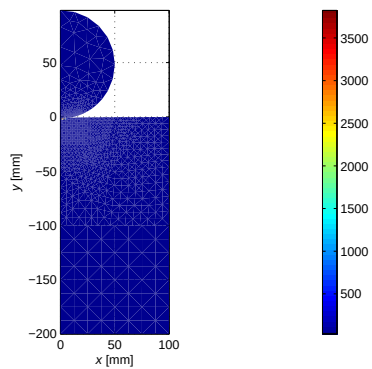
Obrázek 4: Rozložení von Misesova napětí v MPa a deformace těles v mm.



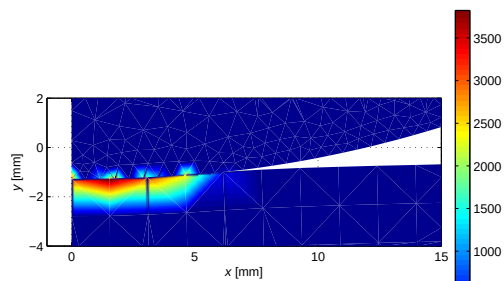
Obrázek 5: Přiblížení obrázku 4 v okolí kontaktní zóny.

Na obrázku 8 je znázorněno vypočtené a analytické kontaktní napětí v MPa v místě kontaktu a na obrázku 9 je zobrazen skok v posunutích v normálovém směru $[u_n]$ v mm a počáteční mezera mezi kontaktními hranicemi.

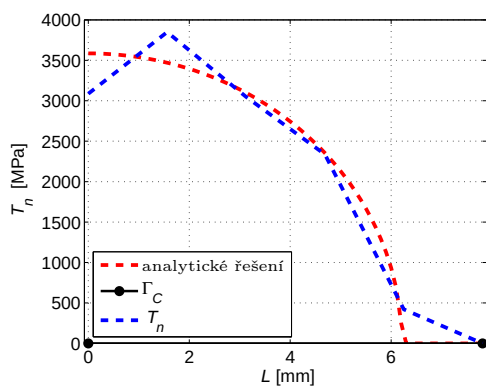
Historie algoritmu SMALBE je uvedena v tabulce 1. Kde SMALBE it. je číslo vnější iterace, $L(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\rho})$ hodnota Lagrangiánu, $\|\boldsymbol{\mu}(\mathbf{x})\|$ norma projektovaného gradientu, $\|\mathbb{B}\mathbf{x}\|$ chyba splnění rovnostních vazeb $\mathbb{B}\mathbf{x} = \mathbf{0}$ v algoritmu SMALBE, Exp je počet expanzních kroků, $Prop$ počet proporcionalizací, Cgm počet kroků sdružených gradientů, No_A počet



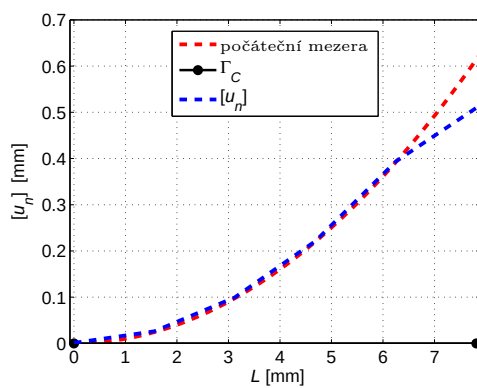
Obrázek 6: Rozložení kontaktního napětí v MPa.



Obrázek 7: Přiblížení obrázku 6 v okolí kontaktní zóny.



Obrázek 8: Normálové napětí v MPa na hranici Γ_c .



Obrázek 9: Skok v posunutích v normálovém směru v mm na Γ_c .

násobení Hessiánem, ρ penalta a M přesnost ukončení algoritmu MPRGP. Celý výpočet problému trval 1.8125 sekund na PC s procesorem Intel Pentium 1.8GHz a pamětí 1GB.

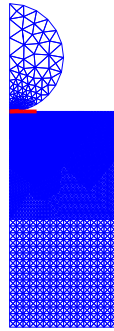
SMALBE it.	$L(\mathbf{x}, \boldsymbol{\mu}, \rho)$	$\ \boldsymbol{\mu}(\mathbf{x})\ $	$\ \mathbb{B}\mathbf{x}\ $	Exp	Prop	Cgm	No_A	ρ	M
0	0	2.35	0	0	0	0	1	0.001	0.01
1	$-1.465506 \cdot 10^4$	$9.20 \cdot 10^{-1}$	$1.46 \cdot 10^{-1}$	6	0	11	24	0.001	0.01
2	$-1.562683 \cdot 10^4$	$7.20 \cdot 10^{-2}$	$1.14 \cdot 10^{-2}$	0	0	7	8	0.001	0.01
3	$-1.563275 \cdot 10^4$	$1.25 \cdot 10^{-2}$	$1.10 \cdot 10^{-3}$	0	0	1	2	0.001	0.01
4	$-1.563304 \cdot 10^4$	$1.23 \cdot 10^{-2}$	$8.36 \cdot 10^{-4}$	0	0	2	3	0.001	0.001
5	$-1.563312 \cdot 10^4$	$9.93 \cdot 10^{-3}$	$1.03 \cdot 10^{-3}$	0	0	1	2	0.001	0.0001
6	$-1.563375 \cdot 10^4$	$2.25 \cdot 10^{-3}$	$3.06 \cdot 10^{-4}$	0	0	16	17	0.001	0.0000
7	$-1.563376 \cdot 10^4$	$2.14 \cdot 10^{-4}$	$1.99 \cdot 10^{-5}$	0	0	2	3	0.001	0.0000

Tabulka 1: Historie algoritmu SMALBE.

5.1.2 2D Hertzova úloha bez tření, jemnější síť

Na obrázku 10 je znázorněno pokrytí těles jemnější trojúhelníkovou sítí ve srovnání s předchozím příkladem se zjemněním sítě v místě předpokládaného kontaktu.

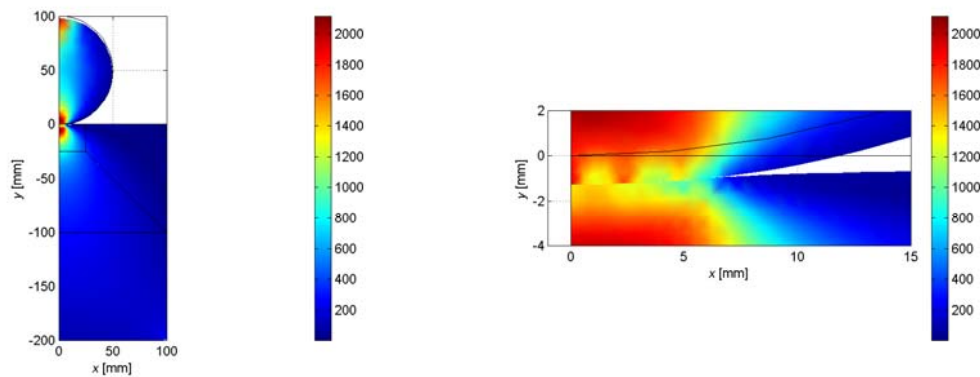
Dosažené výsledky jsou zobrazeny na obrázcích 11 až 16. Obrázek 11 nám zobrazuje rozložení von Misesova napětí v MPa a deformace těles v mm. Obrázek 12 je přiblížení obrázku 11 v okolí kontaktní zóny. Na obrázku 13 vidíme rozložení kontaktního napětí a obrázek 14 je opět přiblížení obrázku 13 v okolí kontaktní zóny.



Obrázek 10: Použitá trojúhelníková síť.

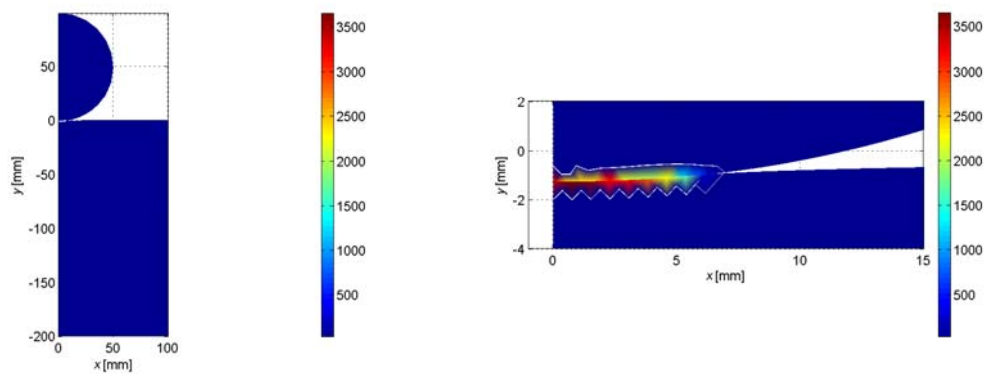
Vygenerovaná síť obsahuje 16242 trojúhelníkových elementů a 8426 uzlů. Z toho 600 uzlů leží na hranici a 48 leží v kontaktní zóně. Z toho 16 patří k master hranici a 32 ke slave hranici.

Na obrázku 15 je znázorněno vypočtené a analytické kontaktní napětí v MPa v místě kontaktu a na obrázku 16 je zobrazen skok v posunutích v normálovém směru $[u_n]$ v mm a počáteční mezera mezi kontaktními hranicemi.



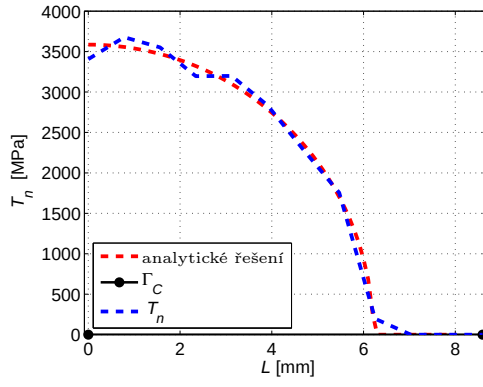
Obrázek 11: Rozložení von Misesova napětí v MPa a deformace těles v mm.

Obrázek 12: Přiblížení obrázku 11 v okolí kontaktní zóny.

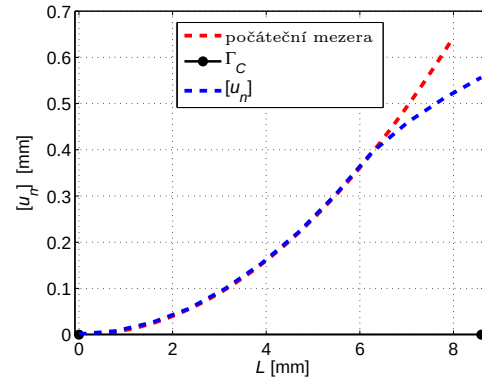


Obrázek 13: Rozložení kontaktního napětí v MPa.

Obrázek 14: Přiblížení obrázku 13 v okolí kontaktní zóny.



Obrázek 15: Normálové napětí v MPa na hranici Γ_c .



Obrázek 16: Skok v posunutích v normálovém směru v mm na Γ_c .

Z obrázku 15 vidíme, že pro jemnější síť dostáváme přesnější rozložení kontaktních napětí a délku kontaktní hranice.

Historie algoritmu SMALBE je uvedena v tabulce 1. Celý výpočet problému trval 7.8438 sekund na stejném počítači jako v předchozím příkladu.

SMALBE it.	$L(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\rho})$	$\ \boldsymbol{\mu}(\mathbf{x})\ $	$\ \mathbb{B}\mathbf{x}\ $	Exp	Prop	Cgm	No_A	$\boldsymbol{\rho}$	M
0	0	1.74	0	0	0	0	1	0.001	0.01
1	$-4.621140 \cdot 10^3$	$6.57 \cdot 10^{-2}$	$7.51 \cdot 10^{-3}$	3	0	18	25	0.001	0.01
2	$-4.624419 \cdot 10^3$	$9.66 \cdot 10^{-3}$	$3.27 \cdot 10^{-4}$	0	0	4	5	0.001	0.01
3	$-4.624408 \cdot 10^3$	$1.21 \cdot 10^{-2}$	$3.27 \cdot 10^{-4}$	0	0	0	1	0.001	0.01
4	$-4.624436 \cdot 10^3$	$8.71 \cdot 10^{-3}$	$6.97 \cdot 10^{-4}$	0	0	1	2	0.001	0.001
5	$-4.624804 \cdot 10^3$	$2.02 \cdot 10^{-3}$	$1.24 \cdot 10^{-4}$	0	0	11	12	0.001	0.0001
6	$-4.624822 \cdot 10^3$	$1.43 \cdot 10^{-4}$	$8.24 \cdot 10^{-5}$	0	0	16	17	0.001	0.0001

Tabulka 2: Historie algoritmu SMALBE.

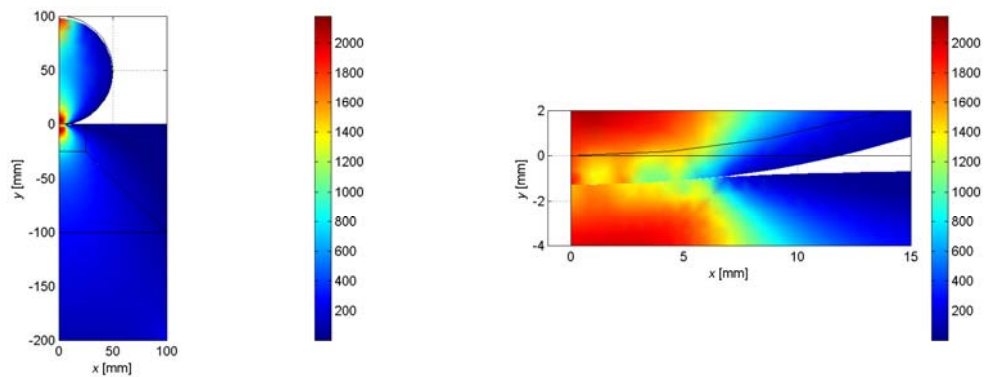
Jak je vidět na obrázcích 11 až 16, pro model s jemnější trojúhelníkovou sítí jsme dostali lepší výsledky, které jsou blíže analytickému řešení.

5.1.3 2D Hertzova úloha s Coulombovým třením

Nyní si ukážeme řešení modelu s jemnější sítí z předchozí kapitoly s uvažováním Coulombova tření. Hodnotu koeficientu tření nastavíme na 0.1.

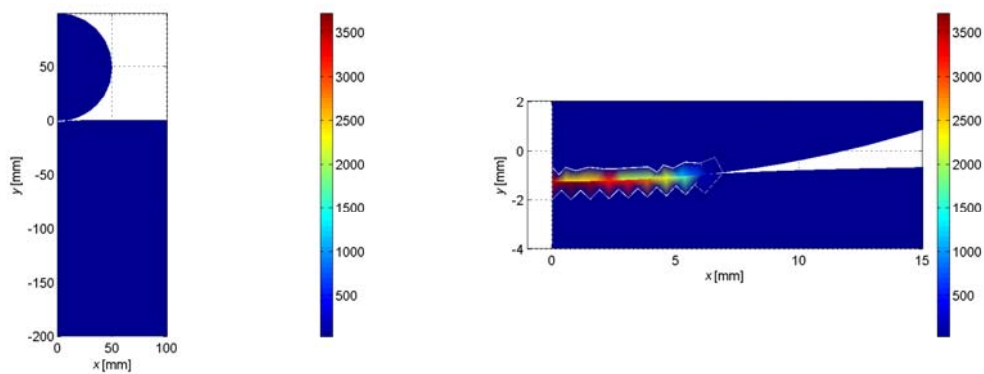
Dosažené výsledky jsou ilustrovány na obrázcích 17 až 25. V tomto případě navíc zobrazujeme rozložení tečného kontaktního napětí na obrázku 21 a jeho přiblížení v okolí zóny kontaktu na obrázku 22.

Na obrázku 23 vidíme mez skluzu $g = FT_n$ a tečné napětí T_t v MPa. Konečně, na obrázku 24 a 25 jsou zobrazeny skoky v posunutích v tečném a normálovém směru.



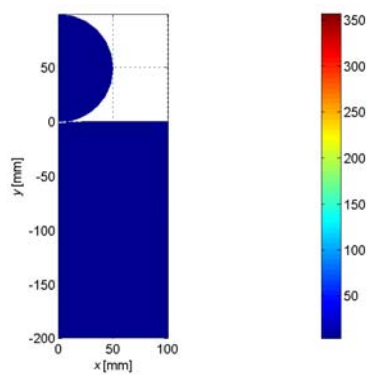
Obrázek 17: Rozložení von Misesova napětí v MPa a deformace těles v mm.

Obrázek 18: Přiblížení obrázku 17 v okolí kontaktní zóny.

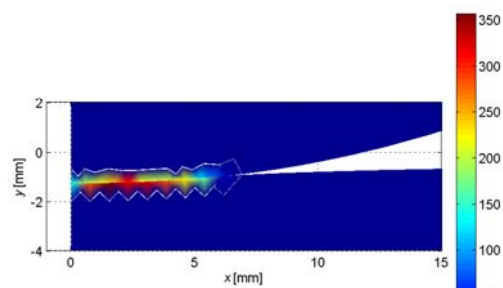


Obrázek 19: Rozložení normálového kontaktního napětí v MPa.

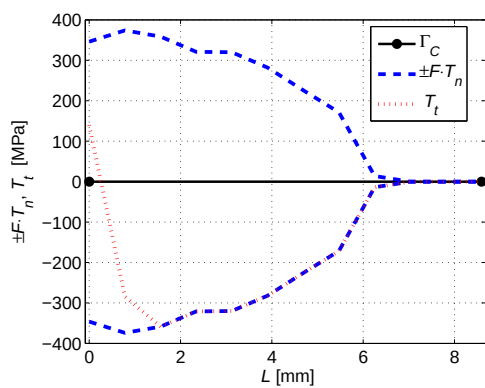
Obrázek 20: Přiblížení obrázku 19 v okolí kontaktní zóny.



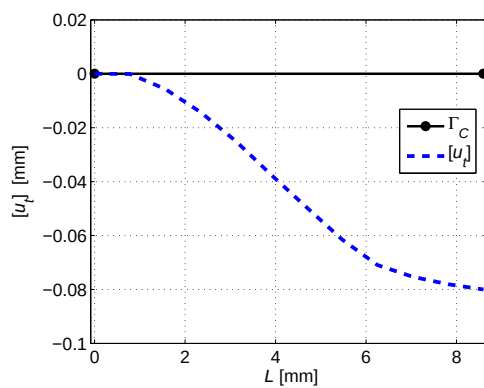
Obrázek 21: Rozložení tečného kontaktního napětí v MPa.



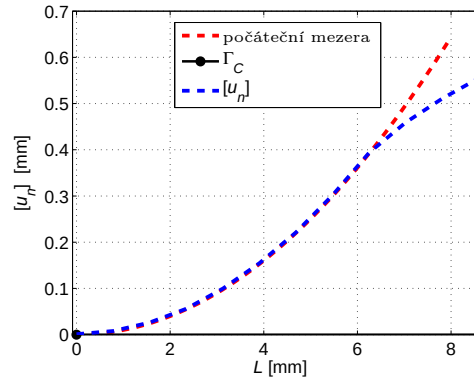
Obrázek 22: Přiblížení obrázku 21 v okolí kontaktní zóny.



Obrázek 23: Mez skluzu a tečné napětí v MPa na hranici Γ_c .



Obrázek 24: Skok v posunutích v tečném směru na Γ_c .



Obrázek 25: Skok v posunutích v normálním směru na Γ_c .

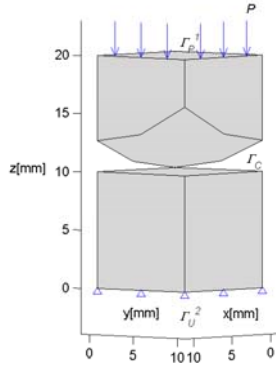
5.2 Hertzova úloha ve 3D

Jako úlohu Hertzova typu ve 3D uvažujme kvádr s kulovou spodní stěnou položený na jiný kvádr. Z důvodů symetrie vezmeme jen jednu čtvrtinu celku (viz obr. 26). Tlak na hranici Γ_p^1 je -500 MPa a hranice Γ_u^2 je pevně zafixována. Po stranách jsou definovány podmínky symetrie.

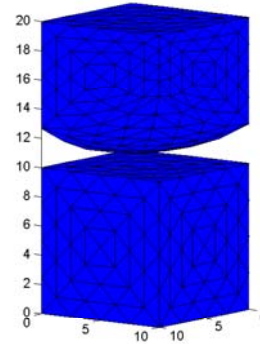
Při výpočtu kontaktního problému používáme opět algoritmus SMALBE. Nastavení parametrů algoritmu zůstává stejné.

5.2.1 3D Hertzova úloha bez tření

Na obrázku 26 vidíme geometrii modelu a obrázek 27 nám zobrazuje vygenerovanou síť ze čtyřtěstů.



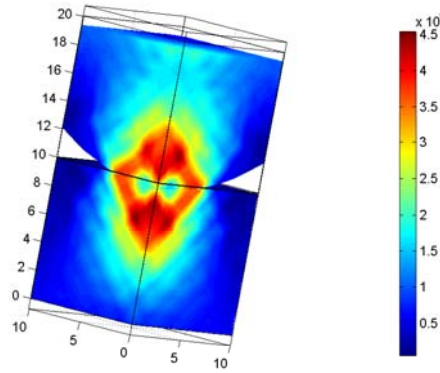
Obrázek 26: Geometrie úlohy.



Obrázek 27: Užitá síť složená ze čtyřtěstů.

Dosažené výsledky jsou zobrazeny na obrázcích 28 až 32. Obrázek 28 nám ukazuje rozložení von Misesova napětí.

Úloha obsahuje 3072 čtyřstěňů a 730 uzlů. Celou hranici tvoří 768 trojúhelníkových plošek, z toho 8 plošek patří ke kontaktní hranici master a 8 plošek patří ke kontaktní hranici slave.



Obrázek 28: Rozložení von Misesova napětí v MPa a deformace těles v mm.

Historie algoritmu SMALBE je uvedena v tabulce 3. Celý výpočet problému trval 2.8438 sekund.

SMALBE it.	$L(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\rho})$	$\ \boldsymbol{\mu}(\mathbf{x})\ $	$\ \mathbb{B}\mathbf{x}\ $	Exp	Prop	Cgm	No_A	$\boldsymbol{\rho}$	M
0	$4.462736 \cdot 10^{-14}$	4.86	$2.57 \cdot 10^{-14}$	0	0	0	1	0.001	0.01
1	$-5.994776 \cdot 10^4$	2.77	$6.77 \cdot 10^{-1}$	11	0	15	38	0.001	0.01
2	$-8.068615 \cdot 10^4$	$2.03 \cdot 10^{-1}$	$5.65 \cdot 10^{-2}$	0	0	10	11	0.001	0.01
3	$-8.082933 \cdot 10^4$	$2.24 \cdot 10^{-2}$	$5.17 \cdot 10^{-3}$	0	0	4	5	0.001	0.01
4	$-8.083099 \cdot 10^4$	$7.69 \cdot 10^{-3}$	$4.45 \cdot 10^{-4}$	0	0	2	3	0.001	0.01
5	$-8.083097 \cdot 10^4$	$1.06 \cdot 10^{-2}$	$4.45 \cdot 10^{-4}$	0	0	0	1	0.001	0.01
6	$-8.083098 \cdot 10^4$	$1.45 \cdot 10^{-2}$	$9.17 \cdot 10^{-4}$	0	0	1	2	0.001	0.001
7	$-8.083105 \cdot 10^4$	$1.16 \cdot 10^{-2}$	$7.71 \cdot 10^{-4}$	0	0	1	2	0.001	0.0001
8	$-8.083457 \cdot 10^4$	$4.26 \cdot 10^{-4}$	$4.75 \cdot 10^{-4}$	0	0	14	15	0.001	0.0001

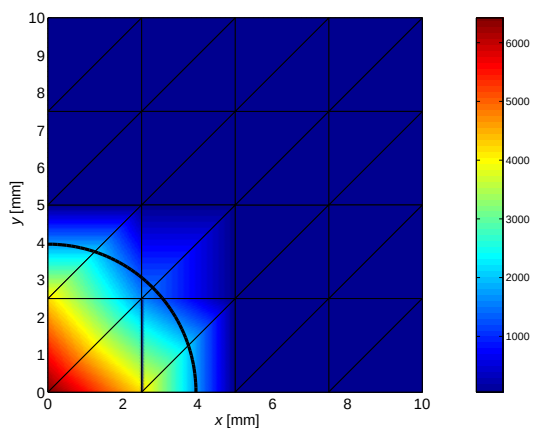
Tabulka 3: Historie algoritmu SMALBE.

Rozložení normálového napětí v MPa a posunutí v mm v jednotlivých směrech na horní ploše kváдру, kde dochází ke kontaktu, je vidět na obrázcích 29 až 32. Tlustá černá čára je hranice kontaktní zóny.

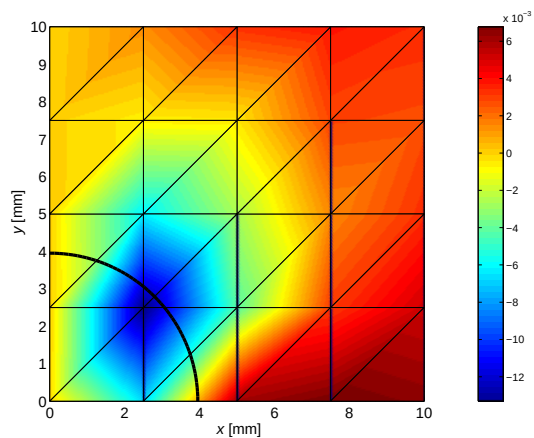
5.2.2 3D Hertzova úloha bez tření s rozložením těles na 16 podoblastí

Z důvodu efektivního řešení úlohy na jemnější síti rozložíme tělesa na 16 podoblastí, jak je patrné z obrázku 33. Síť ze čtyřstěňů je vidět na obrázku 34.

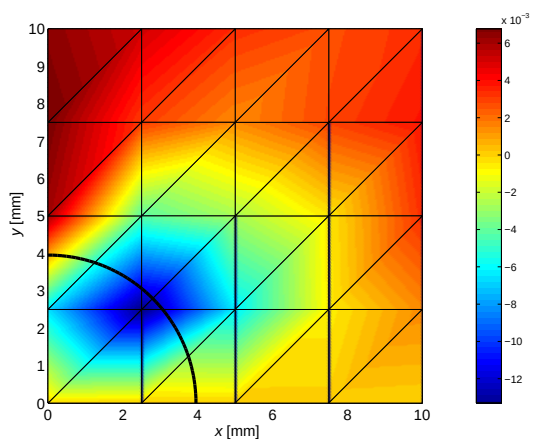
Dosažené výsledky jsou zobrazeny na obrázcích 35 až 39. Obrázek 35 nám zobrazuje rozložení von Misesova napětí v MPa a deformace v mm.



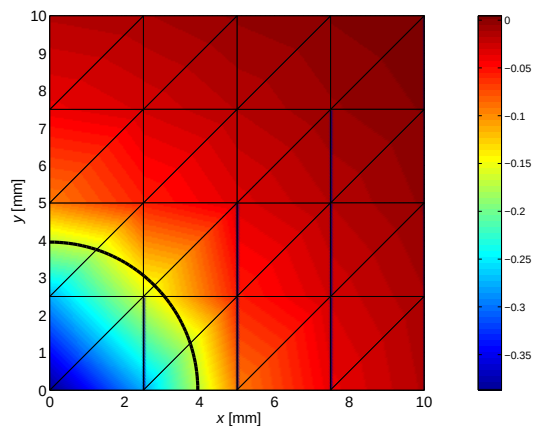
Obrázek 29: Rozložení normálového napětí v MPa.



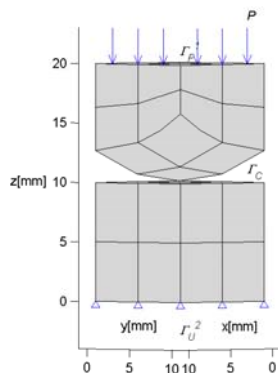
Obrázek 30: Posunutí ve směru x v mm.



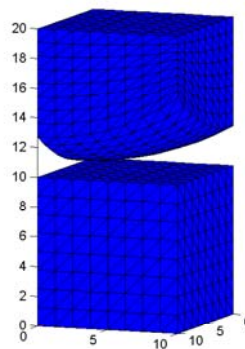
Obrázek 31: Posunutí ve směru y v mm.



Obrázek 32: Posunutí ve směru z v mm.

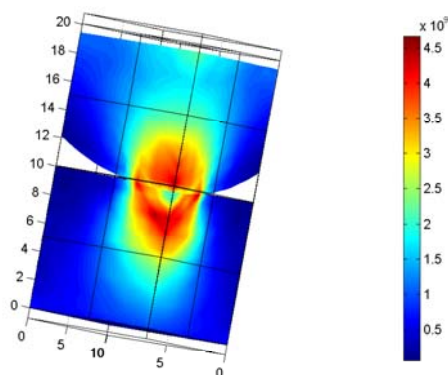


Obrázek 33: Geometrie úlohy.



Obrázek 34: Užitá síť složená ze čtyřstěnů.

Úloha obsahuje 6144 čtyřstěnů a 2000 uzlů. Povrch tvoří 3072 trojúhelníkových plošek, z toho 32 plošek patří ke kontaktní hranici master a 32 plošek patří ke kontaktní hranici slave.



Obrázek 35: Rozložení von Misesova napětí v MPa a deformace těles v mm.

Historie algoritmu SMALBE je uvedena v tabulce 4. Celý výpočet problému trval 25.2813 sekund.

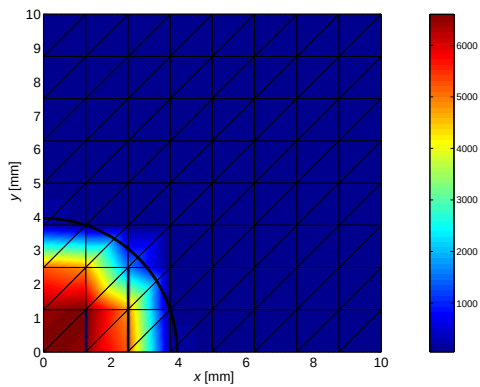
Rozložení normálového napětí v MPa a posunutí v mm v jednotlivých směrech na horní ploše kvádrů, kde dochází ke kontaktu je vidět na obrázcích 36 až 39. Tlustá černá čára je hranice kontaktní zóny.

5.2.3 3D Hertzova úloha s daným třením

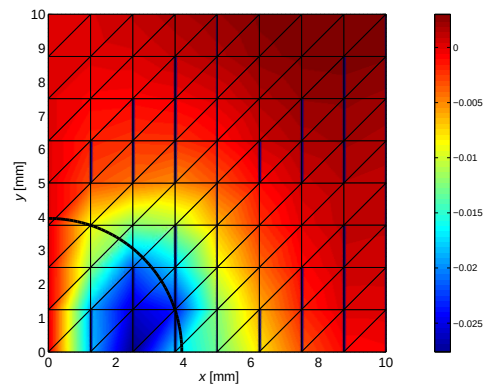
Nyní si přeřešíme úlohu z kapitoly 5.2.1 s uvažováním tření a s jemnější sítí. Hodnotu meze skluzu nastavíme na 200 MPa.

SMALBE it.	$L(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\rho})$	$\ \boldsymbol{\mu}(\mathbf{x})\ $	$\ \mathbb{B}\mathbf{x}\ $	Exp	Prop	Cgm	No_A	$\boldsymbol{\rho}$	M
0	9.741196	4.99	$3.82 \cdot 10^{-3}$	0	0	0	1	0.001	0.01
1	$-4.053604 \cdot 10^4$	3.45	$9.84 \cdot 10^{-1}$	26	0	23	76	0.001	0.01
2	$-7.830761 \cdot 10^4$	$4.20 \cdot 10^{-1}$	$1.70 \cdot 10^{-1}$	4	0	29	38	0.001	0.01
3	$-7.929830 \cdot 10^4$	$9.01 \cdot 10^{-2}$	$3.74 \cdot 10^{-2}$	0	0	16	17	0.001	0.01
4	$-7.934758 \cdot 10^4$	$2.04 \cdot 10^{-2}$	$7.89 \cdot 10^{-3}$	1	0	8	11	0.001	0.01
5	$-7.934906 \cdot 10^4$	$1.01 \cdot 10^{-2}$	$2.74 \cdot 10^{-3}$	0	0	3	4	0.001	0.01
6	$-7.934955 \cdot 10^4$	$1.79 \cdot 10^{-2}$	$1.19 \cdot 10^{-3}$	0	0	3	4	0.001	0.01
7	$-7.934964 \cdot 10^4$	$1.68 \cdot 10^{-2}$	$1.18 \cdot 10^{-3}$	0	0	1	2	0.001	0.001
8	$-7.934974 \cdot 10^4$	$1.13 \cdot 10^{-2}$	$7.94 \cdot 10^{-4}$	0	0	1	2	0.001	0.0001
9	$-7.935167 \cdot 10^4$	$4.30 \cdot 10^{-3}$	$7.26 \cdot 10^{-4}$	0	0	24	25	0.001	0.0000
10	$-7.935170 \cdot 10^4$	$4.50 \cdot 10^{-4}$	$7.36 \cdot 10^{-5}$	0	0	10	11	0.001	0.0000

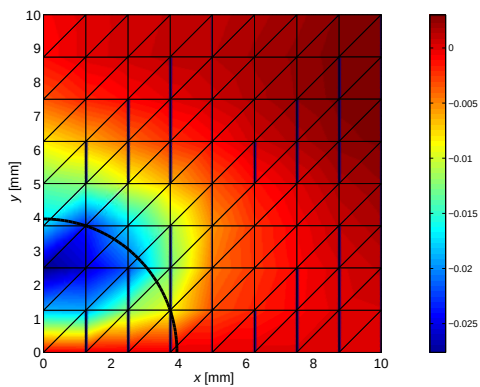
Tabulka 4: Historie algoritmu SMALBE.



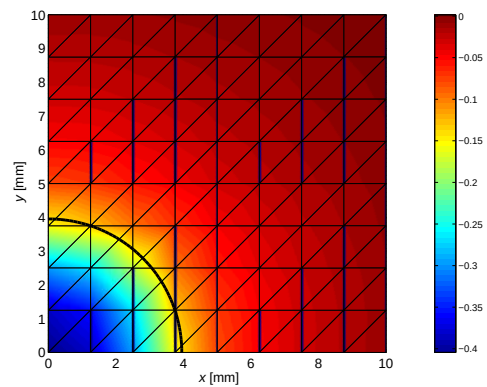
Obrázek 36: Rozložení normálového napětí v MPa.



Obrázek 37: Posunutí ve směru x v mm.

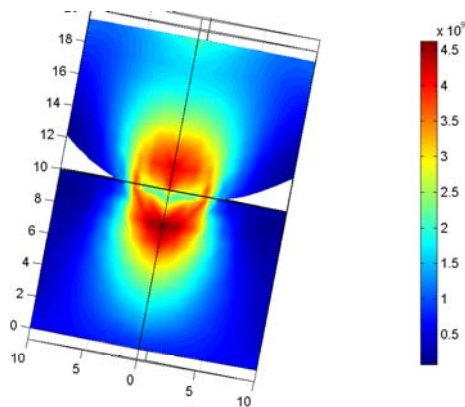


Obrázek 38: Posunutí ve směru y v mm.



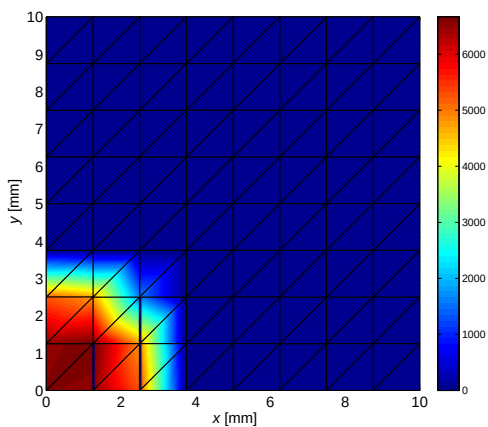
Obrázek 39: Posunutí ve směru z v mm.

Dosažené výsledky jsou zobrazeny na obrázcích 40 až 46. Obrázek 40 nám ukazuje rozložení von Misesova napětí a deformace v mm.

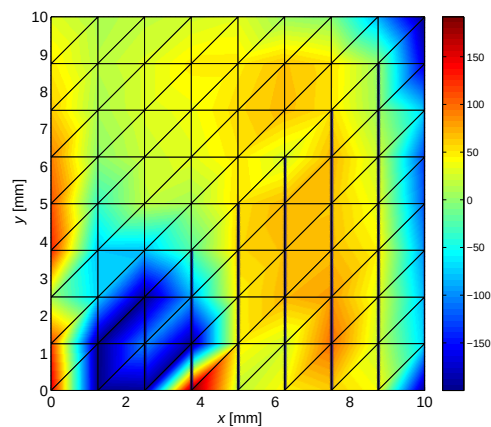


Obrázek 40: Rozložení von Misesova napětí v MPa a deformace těles v mm.

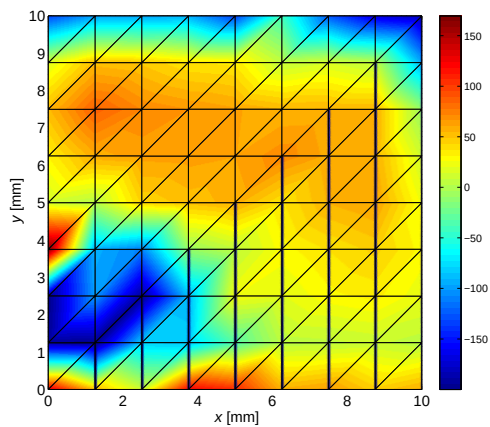
Rozložení normálového a složek tečného napětí v MPa a posunutí v mm v jednotlivých směrech na horní ploše kvádru, kde dochází ke kontaktu, je vidět na obrázcích 41 až 46. Celý výpočet problému trval 15.8561 sekund.



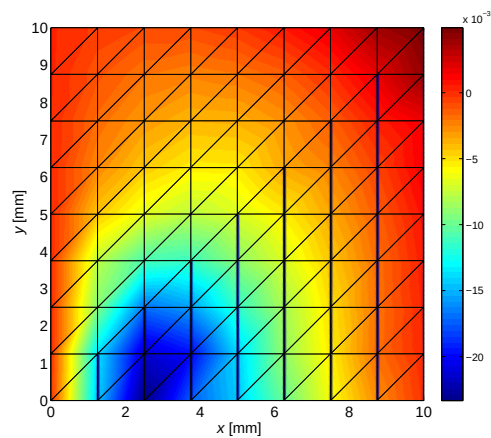
Obrázek 41: Rozložení normálového napětí v MPa.



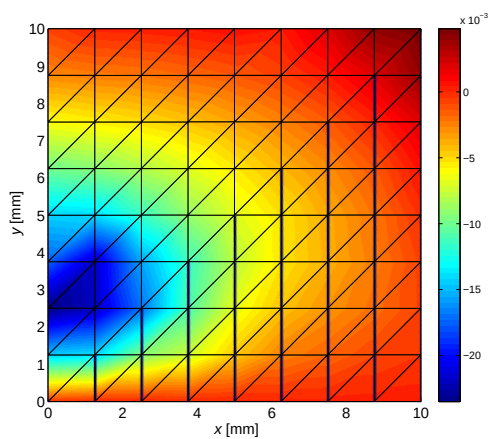
Obrázek 42: Rozložení tečného napětí $T_{t,x}$ v MPa.



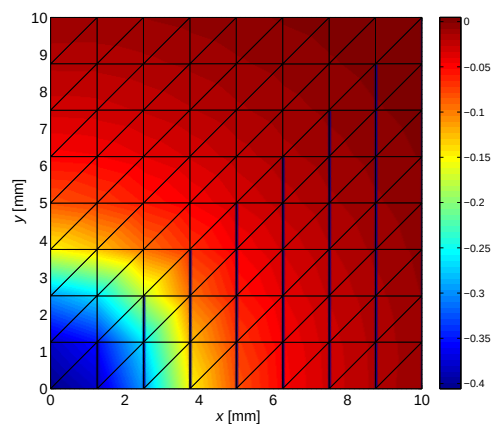
Obrázek 43: Rozložení tečného napětí $T_{t,y}$ v MPa.



Obrázek 44: Posunutí ve směru x v mm.



Obrázek 45: Posunutí ve směru y v mm.



Obrázek 46: Posunutí ve směru z v mm.

Závěr

Bylo implementováno rozhraní, které umožňuje napojení kontaktních úloh modelovaných v systému COMSOL na řešiče implementované v MATLABu a OOSOLu. Jeho součástí jsou funkce pro sestavení matic a vektorů z podmínek nepronikání a zákona tření. Sestavení matice tuhosti, vektoru zatížení a objektů z rovnostních vazeb (Dirichletovy a lepící podmínky) je realizováno pomocí COMSOLovské funkce assemble. Dále byly implementovány efektivní násobící procedury pro násobení Hessiánem z úlohy (2.16) a pseudoinverzí $\mathbb{K}^\dagger$.

Na takto vytvořené matice a vektory je aplikován numericky i paralelně škálovatelný algoritmus SMALBE vyvinutý prof. Dostálem z Katedry aplikované matematiky, VŠB-TU Ostrava.

Přínos této práce je ilustrován na řadě modelových úloh Herzova typu ve 2D a 3D.

V této práci bych rád pokračoval v doktorském studiu, kde bych se rád pokusil o paralelní řešení a další modifikace, které by vyplynuly z požadavků naší katedry.

Reference

- [1] Z. Dostál: *Optimal Quadratic Programing Algorithms with Applications to Variational Inequalities, 1st Edition*, Monograph (2007)
- [2] J. Haslinger, R. Kučera, Z. Dostál: *An algorithm for the numerical realization of 3D contact problems with Coulomb friction*, Journal of Computational and Applied Mathematics 164-165 (2004) 387-408
- [3] J. Haslinger, T. Kozubek, R. Kučera, G. Peichl: *Projected Schur complement method for solving non-symmetric systems arising from a smooth fictitious domain approach*, Numer. Linear Algebra Appl. (2007) 713-739
- [4] R. Kučera, J. Haslinger, Z. Dostál: *Quadratic programming with separable convex constraints and solving of 3D contact problems with friction*, 2006.
- [5] P. Karban: *Výpočty a simulace v programech Matlba a Simulink*, Computer Press, a.s., Brno, 2006, ISBN 80-251-1301-9
- [6] O. Vlach: *Modelování kompozitů pomocí řešičů založených na dualitě*, diplomová práce, Ostrava, 2001
- [7] O. Vlach: *Aproximace a numerická realizace kontaktních problémů se třením a koeficientem tření závislých na řešení*, disertační práce, Ostrava, 2005
- [8] Dokumentace pro Comsol 3.3
- [9] Internetové stránky http://www.humusoft.cz/pub/femlab/tk_07/comsol33a.rtf.
- [10] A.R. Conn, N.I.M. Gould, Ph.L. Toint: *A globally convergent augmented Lagrangian algorithm for optimization general constraints and simple bounds*, SIAM Journal and Numerical Analysis, 28, 545-572, (1991)
- [11] A.R. Conn, N.I.M. Gould, Ph.L. Toint: *LANCELOT: A FORTRAN Package for Large Scale Nonlinear Optimization, (Release A), no. 17 in Springer Series Computational Mathematics*, Springer-Verlag, New York (1992)
- [12] Z. Dostál, A. Friedlander, S.A. Santos: *Augmented Lagrangians with adaptive precision control for quadratic programing with simple bounds and equality constraints*, SIAM J. Optimiz., 13, 1120-1140 (2003)
- [13] Z. Dostál: *Inexact semi-monotonic Augmented Lagrangians with optimal feasibility convergence for quadratic programming with simple bounds and equality constraints*, SIAM J. Numer. Analysis, 45, 2, 500-513 (2007)
- [14] I. Hlaváček, J. Haslinger, J. Nečas, J. Lovíšek: *Solution of Variational Inequalities in Mechanics*. Springer, Berlin, 1988.

- [15] J. Haslinger: *Approximation of the Signorini problem with friction, obeying Coulomb law*. Math. Methods Appl. Sci, 5, 1983, pp. 422-437.
- [16] I. Hlaváček, J. Haslinger, J. Nečas, J. Lovíšek: *Numerical Solution of Variational Inequalities*, Springer Series in Applied Mathematical Sciences, Vol. 66, Springer, New York, 1988.
- [17] N. Kikuchi, J.T. Oden: *Contact Problems in Elasticity*, SIAM, Philadelphia, PA, 1988.
- [18] G. Duvaut, J.L. Lions: *Inequalities in mechanics and physics*, Grundlehren der mathematischen Wissenschaften, Vol 219, Springer, Berlin, 1976.
- [19] Z. Dostál, J. Schöberl: *Minimizing quadratic functions subject to bound constraints with the rate of convergence and finite termination*, Comput. Optimiz. and Appl., 30,1, 2003, 23-44.
- [20] R. Kučera: *Convergence rate of an optimization algorithm for minimizing quadratic functions with separable convex constraints*, přijato k publikaci v SIAM J. Optimization, 2008.