



národní
úložiště
šedé
literatury

Modifikace FEC - kód pro dynamickou kontaktní úlohu v lineární pružnosti

Lanzendörfer, Martin

2006

Dostupný z <http://www.nusl.cz/ntk/nusl-36082>

Dílo je chráněno podle autorského zákona č. 121/2000 Sb.

Tento dokument byl stažen z Národního úložiště šedé literatury (NUŠL).

Datum stažení: 03.10.2024

Další dokumenty můžete najít prostřednictvím vyhledávacího rozhraní [nusl.cz](http://www.nusl.cz) .



Institute of Computer Science
Academy of Sciences of the Czech Republic

Modifikace FEC – kód pro dynamickou kontaktní úlohu v lineární pružnosti

M. Lanzendörfer

Technical report No. 980

November 2006



Institute of Computer Science
Academy of Sciences of the Czech Republic

Modifikace FEC – kód pro dynamickou kontaktní úlohu v lineární pružnosti

M. Lanzendörfer

Technical report No. 980

November 2006

Abstrakt:

Kód FEC byl sestaven k řešení (stacionárních) kontaktních problémů v lineární pružnosti. Vypracovaná modifikace tohoto kódu je připravena k řešení dynamických kontaktních problémů v lineární pružnosti. Časová diskretizace je provedena jednoduchou semi-implicitní metodou, přibližné řešení je hledáno metodou konečných prvků.

Předložená studie je výzkumnou zprávou projektu MPO ČR č. FT-TA/087 za rok 2006.

Keywords:

metoda konečných prvků, dynamická kontaktní úloha, matematické programování

1 Úvod

1.1 Úvod, motivace

Matematický model jednostranného kontaktu pružných těles má rozsáhlé důležité aplikace od průmyslového vývoje až po například výzkum biomechaniky lidského skeletu. Program s pracovním názvem FEC (resp. FEC 3D), sestavený za účelem řešení stacionárních kontaktních úloh v lineární pružnosti metodou konečných prvků ve třech prostorových dimenzích, byl již na ÚI AV ČR používán ke studiu některých aplikací v biomechanice a v geomechanice. Software FEC je popsán v technických zprávách [5], [8] a [10]. Výchozí kód používá formulaci kontaktu pomocí podmínek Signoriniho typu společně s Trescovým modelem tření.

Zde prezentovaná modifikace je připravena pro numerické simulace dynamického modelu pomocí semi-implicitní časové diskretizace s pevně daným časovým krokem. Ve formulaci jsou oproti původní stacionární verzi zahrnuty setrvačný člen a tlumící člen, dále mohou oproti původní verzi hodnoty okrajových podmínek (Dirichletovy a Neumannovy) obecně záviset na čase. V každém časovém kroku je řešena úloha, jejíž struktura je totožná se strukturou stacionární úlohy řešené původní verzí kódu FEC.

Předložená verze nezahrnuje tření, řeší tedy jen dynamickou Signoriniho kontaktní úlohu. Implementace však zohledňuje následné přímočaré rozšíření jak na úlohy se třením, tak na úlohy v lineární vazko-pružnosti. Motivací pro vývoj kódu jsou zejména aplikace v biomechanice lidských kloubů a jejich náhrad, viz například [12], [16]. Vývoj byl součástí projektu MPO ČR pod označením FT-TA/087.

1.2 Fyzikální formulace

Mějme oblast Ω představující \bar{s} jednotlivých pružných těles $\Omega = \cup_{s=1}^{\bar{s}} \Omega^s \subset \mathbb{R}^3$ ve třírozměrném prostoru. Hranici oblasti rozdělme disjunktčně podle uvažovaných okrajových podmínek na $\partial\Omega = \Gamma_{\mathbf{F}} \cup \Gamma_{\mathbf{u}} \cup \Gamma_c$, přičemž $\Gamma_c = \cup_{r,s} \Gamma_c^{rs}$, kde $\Gamma_c^{rs} = \Gamma_c^r \cap \Gamma_c^s$ představuje kontaktní rozhraní mezi tělesy Ω^r a Ω^s , $\Gamma_c^s \subset \partial\Omega^s$. Uvažujme časový interval $I = (0, T)$, v němž budeme vývoj mechanického systému studovat. Hledáme posunutí $\mathbf{u} : I \times \Omega \rightarrow \mathbb{R}^3$, splňující následující:

V každém tělese Ω^s je splněna pohybová rovnice

$$\rho \frac{\partial^2}{\partial t^2} \mathbf{u} + \alpha \frac{\partial}{\partial t} \mathbf{u} - \operatorname{div} \boldsymbol{\sigma}(\mathbf{u}) = \mathbf{g} \quad \text{v } I \times \Omega, \quad (1.1)$$

kde $\rho > 0$ je materiálová hustota (konstantní na tělese Ω^s) a $\alpha > 0$ je (malý) tlumící parametr. Pro tensor napětí $\boldsymbol{\sigma}$ uvažujeme vztah daný zobecněným Hookovým zákonem, $\boldsymbol{\sigma}$ je tedy lineárně závislé na linearizovaném tensoru deformace (tensoru malých deformací) $\mathbf{e}(\mathbf{u})$:

$$\boldsymbol{\sigma}(\mathbf{u}) = \mathbf{C} \mathbf{e}(\mathbf{u}) \quad (1.2)$$

$$\mathbf{e}(\mathbf{u}) = \frac{1}{2} (\nabla \mathbf{u} + \nabla^T \mathbf{u}) \quad (1.3)$$

a platí podmínky symetrie a podmínky pozitivní definitnosti, viz například [5], [9]. Program FEC je navíc připraven a používán pro případ izotropního materiálu; pro každou oblast Ω^i jsou zadávány dvě materiálové konstanty: Youngův modul pružnosti E a Poissonova konstanta σ .

Na hranici $\Gamma_{\mathbf{u}}$ uvažujeme tělesa pevně uchycena, posunutí je tedy předepsáno pomocí Dirichletovy okrajové podmínky

$$\mathbf{u} = \mathbf{u}_D \quad \text{na } I \times \Gamma_{\mathbf{u}}, \quad (1.4)$$

zatímco na hranici $\Gamma_{\mathbf{F}}$ je předepsáno zatížení těles pomocí Neumannovy okrajové podmínky

$$\boldsymbol{\sigma} \mathbf{n} = \mathbf{f} \quad \text{na } I \times \Gamma_{\mathbf{F}}, \quad (1.5)$$

kde \mathbf{n} značí normálový vektor k hranici $\Gamma_{\mathbf{F}}$. Jak \mathbf{u}_D tak \mathbf{f} se mohou měnit v závislosti na čase.

Na kontaktním rozhraní předepisujeme Signoriniho podmínku nepronikání

$$\begin{aligned} [\mathbf{u} \cdot \mathbf{n}]^{rs} &\leq 0, & \boldsymbol{\sigma}_{\mathbf{n}} &\leq 0 \\ \boldsymbol{\sigma}_{\mathbf{n}} [\mathbf{u} \cdot \mathbf{n}]^{rs} &= 0 \end{aligned} \quad \text{na } \Gamma_c, \quad (1.6)$$

kde $[*]^{rs} := *^r - *^s$ značí skok hodnoty $*$ na rozhraní těles Ω^r a Ω^s , kde \mathbf{n} značí jednotkový normálový vektor ke kontaktní hranici Γ_c^{rs} a kde $\boldsymbol{\sigma}_n = (\boldsymbol{\sigma} \mathbf{n}) \cdot \mathbf{n}$ je normálová část tensoru napětí $\boldsymbol{\sigma}$.

1.3 Časová diskretizace

Zvoleným schématem diskretizace v časové proměnné bylo semi-implicitní schéma, takové, aby výsledná úloha v každém časovém kroku měla strukturu shodnou se stacionární úlohou řešenou v původním kódu FEC. Pro pevně zvolenou velikost časového kroku $\tau > 0$ uvažujeme jednotlivé časové hladiny

$$t_k = k\tau, \quad k = 1, 2, \dots$$

Rovnici (1.1) pak diskretizujeme v čase jako (horní index k značí vždy hodnotu v čase t_k):

$$\rho \left(\frac{\mathbf{u}^k - 2\mathbf{u}^{k-1} + \mathbf{u}^{k-2}}{\tau^2} \right) + \alpha \left(\frac{\mathbf{u}^k - \mathbf{u}^{k-1}}{\tau} \right) - \operatorname{div} \boldsymbol{\sigma}(\mathbf{u}^k) = \mathbf{g} \quad \text{v } \Omega, \quad (1.7)$$

což po přeuspořádání členů dává (pravou stranu označíme jako \mathbf{F}^k):

$$\left(\frac{\rho}{\tau^2} + \frac{\alpha}{\tau} \right) \mathbf{u}^k - \operatorname{div} \boldsymbol{\sigma}(\mathbf{u}^k) = \mathbf{g} + \left(2\frac{\rho}{\tau^2} + \frac{\alpha}{\tau} \right) \mathbf{u}^{k-1} - \frac{\rho}{\tau^2} \mathbf{u}^{k-2} \equiv \mathbf{F}^k. \quad (1.8)$$

Výsledná úloha řešená v jednom časovém kroku je tedy stejného typu jako stacionární formulace $\operatorname{div} \boldsymbol{\sigma}(\mathbf{u}) = \mathbf{g}$ a bude také řešena stejnými prostředky. Oproti stacionární formulaci je potřeba jen zahrnout první člen na levé straně (1.8) a uvažovat pro každé k obecně různou pravou stranu \mathbf{F}^k .

Okrajové podmínky (1.4), (1.5) a (1.6) jednoduše vyžadujeme na každé časové hladině

$$\mathbf{u}^k = \mathbf{u}_D^k \quad \text{na } \Gamma_u, \quad (1.9)$$

$$\boldsymbol{\sigma}^k \mathbf{n} = \mathbf{f}^k \quad \text{na } \Gamma_F, \quad (1.10)$$

$$\begin{aligned} [\mathbf{u}^k \cdot \mathbf{n}]^{rs} \leq 0, \quad \boldsymbol{\sigma}_n^k \leq 0, \\ \boldsymbol{\sigma}_n^k [\mathbf{u}^k \cdot \mathbf{n}]^{rs} = 0 \end{aligned} \quad \text{na } \Gamma_c, \quad (1.11)$$

hodnoty \mathbf{f}^k a \mathbf{u}_D^k se však opět mohou pro různá k lišit.

1.4 Variační formulace v daném časovém kroku

Zopakujme zde variační formulaci úlohy, na níž je postaveno řešení metodou konečných prvků a jež se v principu neliší od formulace popsané v [5], [6], [7].

Označme $\mathbf{W} = \mathbf{W}^{1,2}(\Omega^1)^3 \times \dots \times \mathbf{W}^{1,2}(\Omega^{\bar{s}})^3$, kde $\mathbf{W}^{1,2}(\Omega^s)$ je standardní Sobolevův prostor. Pak množina přípustných posunutí v čase t_k (přípustných z hlediska Dirichletovy okrajové podmínky a Signoriniho kontaktní podmínky) na dané časové hladině t_k je

$$\mathbf{K}^k = \left\{ \mathbf{v} \mid \mathbf{v} \in \mathbf{W}, \quad \begin{array}{ll} \mathbf{v} = \mathbf{u}_D^k & \text{na } \Gamma_D \\ [\mathbf{v} \cdot \mathbf{n}]^{rs} \leq 0 & \text{na } \cup_{r,s} \Gamma_c^{rs} \end{array} \right\}. \quad (1.12)$$

Funkcionál potenciální energie v čase t_k definujeme jako

$$\mathcal{L}^k(\mathbf{v}) = \frac{1}{2} A(\mathbf{v}, \mathbf{v}) - F^k(\mathbf{v}), \quad (1.13)$$

kde

$$A(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \boldsymbol{\sigma}(\mathbf{e}(\mathbf{u})) : \mathbf{e}(\mathbf{v}) \, d\mathbf{x} + \int_{\Omega} \left(\frac{\rho}{\tau^2} + \frac{\alpha}{\tau} \right) \mathbf{u} \cdot \mathbf{v} \, d\mathbf{x} \quad (1.14)$$

$$F^k(\mathbf{v}) = \int_{\Omega} \mathbf{F}^k \cdot \mathbf{v} \, d\mathbf{x} + \int_{\Gamma_F} \mathbf{f}^k \cdot \mathbf{v} \, d\mathbf{s}. \quad (1.15)$$

Poznamenejme, že v každém tělese Ω^s , $s = 1, \dots, \bar{s}$ uvažujeme obecně různé konstanty $\rho = \rho^s$ a $\alpha = \alpha^s$; hodnota $\left(\frac{\rho}{\tau^2} + \frac{\alpha}{\tau} \right)$ tedy obecně není konstantní v oblasti Ω . Obdobně pro $\mathbf{x} \in \Omega^s$ uvažujeme

$\boldsymbol{\sigma}(\mathbf{e}(\mathbf{u})) : \mathbf{e}(\mathbf{v}) = \sum_{ijkl} C_{ijkl}^s e_{ij}(\mathbf{u}^s) e_{kl}(\mathbf{v}^s)$, kde elastické konstanty C_{ijkl}^s (respektive konstanty E, σ , které je v našem modelu definují) jsou opět na každém tělese obecně různé. Člen \mathbf{F}^k v rovnici (1.15) jsme definovali v (1.8), zahrnuje předepsané objemové síly a hodnoty řešení v minulých dvou časových krocích. Člen \mathbf{f}^k je dán zatížením na části hranice Γ_F , tedy hodnotou Neumannovy okrajové podmínky (1.10) v čase t_k .

Řešením \mathbf{u}^k naší úlohy na časové hladině t_k je minimum funkcionálu $\mathcal{L}^k(\mathbf{v})$ v množině přípustných posunutí \mathbf{K}^k

$$\mathbf{u}^k \in \mathbf{K}^k : \quad \mathcal{L}^k(\mathbf{u}^k) \leq \mathcal{L}^k(\mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{K}^k. \quad (1.16)$$

Za řešení dynamické úlohy diskretizované v čase pak považujeme posloupnost $\{\mathbf{u}^k\}_{k=1}^{\bar{k}}$ (kde $t_{\bar{k}} \geq T$), případně vhodně zvolenou interpolaci, která z této posloupnosti vychází.

1.5 Aproximace konečnými elementy

Mějme připravenou regulární triangulaci \mathcal{T}_h (h značí velikost nejdelší hrany triangulace) jednotlivých těles Ω^s , s čtyřstěnnými elementy (simplexy) T_{hn} , $n = 1, \dots, N_{el}$ a s vrcholy a_m , $m = 1, \dots, N_v$. Předpokládáme, že tělesa uvažovaná ve vzájemném kontaktu mají společnou hranici Γ_c^s a že vrcholy sousedících elementů z Γ_c^r a Γ_c^s vždy tvoří pár (v němž oba vrcholy mají stejné souřadnice a jemžž přísluší normálový vektor \mathbf{n}).

Množiny \mathbf{W} a \mathbf{K}^k aproximujeme konečně-dimenzionálními množinami

$$\mathbf{W}_h = \{ \mathbf{v}_h \in \mathcal{C}(\Omega^1)^3 \times \dots \times \mathcal{C}(\Omega^s)^3 \mid \mathbf{v}_h|_T \in P_1(T)^3, \forall T \in \mathcal{T}_h \} \quad (1.17)$$

$$\mathbf{K}_h^k = \left\{ \mathbf{v}_h \in \mathbf{W}_h \mid \begin{array}{l} \mathbf{v}_h(a_m) = \mathbf{u}_D^k(a_m), \forall a_m \in \Gamma_D \\ [\mathbf{v}_h(a_m^{rs}) \cdot \mathbf{n}]^{rs} \leq 0, \forall a_m^{rs} \in \Gamma_c^{rs} \end{array} \right\}. \quad (1.18)$$

Za řešení \mathbf{u}_h^k aproximované úlohy v čase t_k pak považujeme aproximaci $\mathbf{v} \in \mathbf{K}_h^k$, pro kterou nastane minimum funkcionálu $\mathcal{L}_h^k(\mathbf{v})$ v množině \mathbf{K}_h^k .

Vektor posunutí $\mathbf{v}_h \in \mathbf{K}_h^k$ je určen (volbou vhodné báze prostoru \mathbf{W}_h) vektorem $\mathbf{y} \in \mathbb{R}^{N_{dof}}$, obsahujícím složky vektoru posunutí ve vrcholech a_m . Číslo N_{dof} je počet stupňů volnosti, v našem případě rovné trojnásobku počtu vrcholů, které neleží na Γ_D , tedy těch vrcholů, na kterých není posunutí dáno explicitně Dirichletovou okrajovou podmínkou. Vektor \mathbf{y} je tedy algebraickou reprezentací vektoru posunutí \mathbf{v}_h . Obdobně na jednotlivém elementu triangulace budeme značit $\Delta \in \mathbb{R}^{12}$ vektor obsahující 3 složky posunutí ve 4 vrcholech čtyřstěnu. Prvky Δ jsou výběrem z prvků vektoru \mathbf{y} .

Výraz $\mathcal{L}^k(\mathbf{v}_h)$ lze vyjádřit jako součet příspěvků za všechny elementy, tj. za objemy jednotlivých čtyřstěnu a za ty jejich stěny, které leží na hranici Γ_F nebo na kontaktní hranici Γ_c . Jelikož funkce z prostoru \mathbf{W}_h jsou lineární na každém elementu T_{hn} , jejich hodnoty i derivace uvnitř T_{hn} jsou jednoznačně určeny (a závisí na nich lineárně) hodnotami ve 4 vrcholech čtyřstěnu tvořícího T_{hn} (a obdobně, hodnoty na stěně jsou dány hodnotami v jejích 3 vrcholech). Na elementu T_{hn} lze proto vyjádřit bilineární člen $A(\mathbf{v}_h, \mathbf{v}_h)$ jako

$$\frac{1}{2} A(\mathbf{v}_h, \mathbf{v}_h) \Big|_{T_{hn}} = \frac{1}{2} \Delta^T \mathbf{C}_n \Delta, \quad (1.19)$$

kde definujeme $\mathbf{C}_n \in \mathbb{R}^{12 \times 12}$ na základě (1.14). Obdobně lineární člen $F^k(\mathbf{v}_h)$ napíšeme jako

$$F^k(\mathbf{v}_h) \Big|_{T_{hn}} = \Delta^T \mathbf{b}_n^k, \quad (1.20)$$

příčemž vektor $\mathbf{b}_n^k \in \mathbb{R}^{12}$ je definován tak, že platí (1.15).

Funkcionál $\mathcal{L}_h^k(\mathbf{v}_h)$ je tedy algebraicky reprezentován funkcí

$$\mathcal{L}_h^k(\mathbf{y}) = \frac{1}{2} \mathbf{y}^T \mathbf{C} \mathbf{y} + \mathbf{y}^T \mathbf{b}^k, \quad (1.21)$$

kde matice $\mathbf{C} \in \mathbb{R}^{N_{dof} \times N_{dof}}$ a vektor $\mathbf{b}^k \in \mathbb{R}^{N_{dof}}$ vzniknou sečtením příspěvků \mathbf{C}_n a \mathbf{b}_n^k za všechny elementy T_{hn} (a eliminací neznámých v těch vrcholech, kde je posunutí dáno hodnotou Dirichletovy okrajové podmínky). Matice \mathbf{C} je symetrická, pozitivně definitní, řídká a blokově diagonální (její bloky odpovídají jednotlivým tělesům).

Signoriniho podmínku nepronikání na kontaktní hranici, aproximovanou nerovností (1.18)₂ vyžadovanou jen ve vrcholech elementů

$$[\mathbf{v}_h(a_m^{rs}) \cdot \mathbf{n}]^{rs} \leq 0, \forall a_m^{rs} \in \Gamma_c^{rs}, \quad (1.22)$$

lze napsat v každé kontaktní dvojici vrcholů jako

$$\mathbf{A}_m^T \tilde{\mathbf{\Delta}} \leq 0, \quad (1.23)$$

kde $\tilde{\mathbf{\Delta}} \in \mathbb{R}^6$ obsahuje vektory posunutí ve dvou vrcholech tvořících kontaktní dvojici a_m^{rs} a kde $\mathbf{A}_m \in \mathbb{R}^6$ je vektor odpovídající podmínce (1.22) (v podstatě lze říci $\mathbf{A}_m^T = [\mathbf{n}, -\mathbf{n}]$). Aproximovaná kontaktní podmínka pro celou úlohu lze tedy vyjádřit jako nerovnost

$$\mathbf{A}\mathbf{y} \leq \mathbf{0}, \quad (1.24)$$

kde $\mathbf{A} \in \mathbb{R}^{N_{cont} \times N_{dof}}$ a $\mathbf{0} \in \mathbb{R}^{N_{cont}}$, N_{cont} je počet kontaktních párů, většinou platí $N_{cont} \ll N_{dof}$.

Celá úloha je diskretizací metodou konečných prvků převedena na úlohu kvadratického programování

$$\begin{aligned} \frac{1}{2}\mathbf{y}^T \mathbf{C}\mathbf{y} + \mathbf{y}^T \mathbf{b}^k &\longrightarrow \min \\ \mathbf{A}\mathbf{y} &\leq \mathbf{0}. \end{aligned} \quad (1.25)$$

Tato úloha je řešena metodou sružených projektovaných gradientů, popsanou například v [5], [6], a [9].

2 Implementace

Pro detailnější popis struktury implementace zvolené v programu FEC odkazujeme zejména na [5] a [10], zde se omezíme na stručný popis odlišností dynamické verze od verze původní. Souhrnně řečeno se jedná o kód napsaný v jazyce Fortran 77, bez použití žádných dalších specializovaných knihoven. Data jsou uložena ve dvou globálních (staticky alokovaných) polích, jednom integerovém a jednom realovém. Pro reprezentaci řídkých matic je použit formát CSR – *Compressed Sparse by Rows*, případně varianta CSC – *Compressed Sparse by Columns* s uložením po sloupcích, u symetrických matic jsou reprezentovány jen prvky na a nad diagonálou, viz [5].

V každém vrcholu síť může být proměnný počet stupňů volnosti, obecně v rozmezí od 0 do 3, v naší verzi (jelikož neuvažujeme bilaterální kontaktní podmínky, v původní verzi FEC zahrnuté) buďto 3 nebo v případě Dirichletovy okrajové podmínky 0 stupňů volnosti. K uložení této informace a zejména globálních indexů příslušných stupňů volnosti slouží integerové pole $KOD(J,I)$, $1 \leq J \leq 3$, $1 \leq I \leq \text{NoVertices}$.

Okrajové podmínky, v původním FEC charakterizované dvěma realovými poli H1, H2, jsou v dynamické verzi zjišťovány voláním podprogramu

```
subroutine GetBCValue(HH,iNode,jAxis,bcType,VX,VY,VZ,dTime),
```

který vrací hodnotu HH složky ve směru jAxis okrajové podmínky typu bcType (buďto Dirichletova nebo Neumannova o.p.), ve vrcholu číslo iNode se souřadnicemi VX,VY,VZ; přičemž se ptáme na hodnotu zadanou pro čas dTime. Podprogram GetBCValue je ve zdrojovém souboru bc.f a uživatel v něm okrajové podmínky zadává (např. funkcí) před samotným překladem fortranovského zdrojového kódu. Důvodem pro tuto změnu byla potřeba zadávat různé hodnoty okrajových podmínek v závislosti na čase t.

Program lze rozdělit na dvě hlavní části. V první jsou načtena vstupní data (parametry programu, geometrie úlohy a její materiálové parametry), viz sekce 3. Na základě nich jsou připraveny datové struktury popisující síť konečných prvků, okrajové podmínky, vektor řešení, strukturu stupňů volnosti (vektor řešení). Tyto základní stavební prvky programu FEC zde popisovat nebudeme, stručný popis datových struktur a procesu jejich sestavení lze nalézt v [10].

V první části jsou také sestaveny matice \mathbf{C} a \mathbf{A} a připraveny ty datové struktury potřebné pro sestavení vektoru \mathbf{b}^k , které nezávisí na časově závislých datech (tedy na hodnotách okrajových podmínek ani na hodnotách řešení z předchozích časových hladin). Druhá část pak obsahuje cyklus přes jednotlivé časové hladiny, ve kterém je sestaven vektor \mathbf{b}^k příslušující časovému kroku k a je realizován výpočet řešení.

Před časovým cyklem

Vstupní data a parametry programu jsou popsány v sekci 3. Příprava datových struktur popisujících stupně volnosti a tvořících tak základ pro sestavení algebraických úloh (1.25), probíhá zcela stejně jako v původním kódu FEC, a to v podprogramech

```
subroutine ALLP(...)  
subroutine NUMBERIN(...).
```

Jedním z hlavních výstupů je integerové pole KOD, tvořící spojení mezi vrcholy sítě a mezi stupni volnosti (indexy vektoru řešení).

Samotná příprava k sestavení algebraických úloh obsahuje:

Sestavení matice C na základě výrazu (1.14).

Matice C se tvoří, stejně jako ve stacionární verzi FEC, uvnitř podprogramu

```
subroutine ELSTIFF(...),
```

kteřý je popsán v [10], v sekci 3.2. Algoritmus postupuje element po elementu a pro každý z nich provede čtyři kroky:

- Vytvoření lokální matice (malé, husté) \mathbf{C}_n , viz (1.19).
V této, dynamické verzi je oproti verzi původní přidána matice reprezentující skalární součin $\int_{\mathcal{T}_n} \mathbf{u}_h \cdot \mathbf{v}_h \, d\mathbf{x}$, přenásobená konstantou $(\frac{\rho}{\tau^2} + \frac{\alpha}{\tau})$ (značenou `tRhoAlpha`).
- Eliminace řádků a sloupců odpovídajících vrcholům s Dirichletovou okrajovou podmínkou, voláním procedury

```
subroutine ELIMEL(...).
```

V původní verzi FEC se při tomto procesu tvořily příspěvky do vektoru pravé strany SP na základě hodnot okrajové podmínky. V dynamické verzi je nutno hodnoty okrajové podmínky použít až uvnitř časového cyklu, proto se namísto toho ukládají potřebná data do souboru 'MVH.DAT' a z nich se později tvoří matice MVH, jež, aplikována v každém časovém kroku na aktuální vektor hodnot okrajové podmínky VH, dává teprve správné příspěvky k vektoru pravé strany SPdyn.

- Jednotlivé členy lokální matice jsou vloženy do lokální řádkové matice CLoc, tentokrát s globálními indexy řádků a sloupců.
- Lokální matice CLoc je po blocích vložena do globální matice C. Matice C je reprezentována v řádkovém formátu CSC.

Příprava matice MVH potřebné pro sestavování (assembling) hodnot Dirichletových okrajových podmínek v každém časovém kroku. Matice MVH je vytvořena podprogramem

```
subroutine AssVH(MVH,...),
```

ze záznamů v souboru 'MVH.DAT', vytvořených v průběhu sestavování matice C. Matice MVH je reprezentována ve formátu CSC.

Příprava matic MPrevX, MPPrevX a MPrevVH, MPPrevVH pro sestavování (assembling) příspěvků z předcházejících časových hladin do pravé strany rovnice (1.8). Matice MPrevX a MPPrevX odpovídají (po přenásobení vektory $\mathbf{X}(i_XPrev)$ a $\mathbf{X}(i_XPPrev)$ v každém časovém kroku) členům $(2\frac{\rho}{\tau^2} + \frac{\alpha}{\tau}) \mathbf{u}_h^{k-1}$ a $-\frac{\rho}{\tau^2} \mathbf{u}_h^{k-2}$. Matice MPrevVH a MPPrevVH k tomu analogickým způsobem doplňují informaci o hodnotách Dirichletových okrajových podmínek \mathbf{u}_D^{k-1} , \mathbf{u}_D^{k-2} , která v algebraické reprezentaci vektorů řešení $\mathbf{X}(i_XPrev)$, $\mathbf{X}(i_XPPrev)$ není obsažena.

Příspěvky jednotlivých elementů jsou nejdříve generovány do souboru 'REAR.DAT' podprogramem


```
subroutine RStiff(...),
```

a samotné matice jsou pak postupně vytvořeny podprogramy

```
subroutine AssRearX( MPrevX, t2RhoAlpha, ...)  
subroutine AssRearX( MPPrevX, tRho, ...)  
subroutine AssRearVH( MPrevVH, t2RhoAlpha, ...)  
subroutine AssRearVH( MPPrevVH, tRho, ...).
```

Sestavení matice A reprezentující Signoriniho kontaktní podmínku (1.22), probíhá stejným způsobem jako v původní verzi FEC.

Příspěvky za jednotlivé kontaktní dvojice vrcholů se nejdříve v podprogramu

```
subroutine CONCNS(...)
```

vygenerují do souboru 'CNS.DAT', z něhož se v podprogramu

```
subroutine ASSA(...)
```

vytvoří matice A, reprezentovaná v řídkém formátu CSR. Tento proces provádíme jen jednou před časovým cyklem, jelikož neuvažujeme přítomnost Dirichletovy okrajové podmínky ve vrcholu ležícím zároveň na kontaktní hranici.

Časový cyklus

Posloupnost řešení v jednotlivých časových hladinách t_k je realizována časovým cyklem opakovaným až do dosažení požadované časové hladiny. Krok cyklu obsahuje:

Zvýšení časové hladiny přičtením 1 k číslu časové hladiny k (`iTime`) a přičtením časového kroku τ (`dTimeStep`) k aktuálnímu času t (`dTime`).

Inicializace vektoru řešení u_h^k (značeného X). Uchovávají se 2 vektory řešení z předchozích časových hladin, jejichž obsah by však bylo zbytečné v každém časovém kroku kopírovat. Proto je zde pole tří ukazatelů na realová pole, z nichž jeden ukazuje na aktuální vektor řešení X(`i_X0`) a dva na vektory řešení z předchozích dvou časových hladin X(`i_XPrev`) a X(`i_XPPrev`). V tomto kroku se ukazatele rotují a nový vektor řešení se vynuluje:

```
i_XPPPom = i_XPPrev  
i_XPPrev = i_XPrev  
i_XPrev = i_X0  
i_X0 = i_XPPPom  
CLEAN_VECTOR( X(i_X0), ...)
```

Sestavení částí závislých na časové hladině do vektoru b^k .

Některé z příspěvků, které dávají v součtu vektor pravé strany b^k (značený SPdyn), musejí být přepočítány v každé časové hladině. Jedná se o následující části (a o následující postup jejich sestavování):

- Začni vždy s částí vektoru pravé strany SPdyn, nezávislou na čase (a tedy sestavenou před započítáním časového cyklu do vektoru SP).

```
COPY(SPdyn, SP, ...)  
// copy vector SP->SPdyn
```

- Přidej členy závislé na Dirichletově o.p. do SPdyn.

Hodnoty Dirichletovy okrajové podmínky jsou reprezentovány vektorem VH(`i_VH`) (vektory z předcházejících dvou hladin jsou ukládány obdobným způsobem jaký jsme popisovali u vektorů řešení). Aktuální vektor VH je získán voláním podprogramu

```
subroutine UpdateBV(VH,...,VX,VY,VZ,dTime).
```

Příspěvek do vektoru pravé strany vznikne aplikací matice MVH na aktuální vektor VH:

```
call mAddMatrijxVector(SPdyn,MVH,VH,...).
// add matrix MVH times vector VH into vector SPdyn
```

Matice MVH byla vytvořena v průběhu tvorby matice C před započítáním časového cyklu.

- Přidej členy závislé na Neumannově o.p. do SPdyn. Viz (1.10) a (1.15).

Tento proces probíhá stejně jako v původní stacionární verzi programu FEC, jen je opakován na každé časové hladině. Přidání těchto členů probíhá v podprogramech

```
subroutine BNStiff(...)
// Assemble Neumann b.c. values into file 'BNS.DAT'
```

```
subroutine ASSH(...,SPdyn)
// Assemble data from 'BNS.DAT' into vector SPdyn.
```

- Přidej členy závislé na řešeních v předcházejících časových hladinách \mathbf{u}_h^{k-1} , \mathbf{u}_h^{k-2} do SPdyn. Viz (1.8) a (1.15).

Příspěvek do vektoru pravé strany vznikne aplikací předem (před časovým cyklem) připravených matic MPrevX, MPPrevX a MPrevVH, MPPrevVH na vektory řešení z předcházejících časových hladin \mathbf{u}_h^{k-1} , \mathbf{u}_h^{k-2} a na vektory hodnot Dirichletových okrajových podmínek z předcházejících časových hladin \mathbf{u}_D^{k-1} , \mathbf{u}_D^{k-2} :

```
call msAddMatrijxVector(SPdyn,MPrevX,X(i_XPrev),...)
call msAddMatrijxVector(SPdyn,MPPrevX,X(i_XPPrev),...)
call msAddMatrijxVector(SPdyn,MPrevVH,VH(i_VHPrev),...)
call msAddMatrijxVector(SPdyn,MPPrevVH,VH(i_VHPPrev),...)
```

Řešení vzniklé algebraické úlohy. Takto sestavená úloha kvadratického programování (1.25)

$$\frac{1}{2}\mathbf{y}^T \mathbf{C} \mathbf{y} + \mathbf{y}^T \mathbf{b}^k \longrightarrow \min$$

$$\mathbf{A} \mathbf{y} \leq \mathbf{0}$$

je řešena metodou sdružených projektovaných gradientů, popsanou například v [5], [6], [9] včetně popisu její implementace ve FEC. Metoda je implementována v podprogramu

```
subroutine CG1(C,A,SPdyn,X(i_X0),...)
```

Export spočteného vektoru posunutí. Vektor řešení v čase dTime je zapsán do souboru out.gmv.XXXX, kde za XXXX je dosazena hodnota 1000+iTime. Export je proveden v podprogramu

```
subroutine OUT2GMV3(...,X(i_X0),1000+iTime,dTime),
```

pro strukturu ukládaných dat viz následující sekce.

3 Vstupní a výstupní data, použití kódu, parametry

Program FEC po překladu ze zdrojových kódů tvoří jediný spustitelný soubor fec06.x, který se spouští bez dalších parametrů. Veškerá data jsou programem načtena ze souborů popsaných níže. Program si vytváří v pracovním adresáři soubory *.DAT, jejichž účel je však dočasný a nejsou výstupem výpočtu. Výsledkem běhu je (v případě dynamické úlohy) řada výstupních souborů *.gmv.* v adresáři gmv/, viz dále.

Geometrická data pro současnou verzi programu FEC jsou rozdělena do tří souborů: USR_FIRST.TXT, USR_MATERIALS.TXT a USR_MESH.TXT.

Soubor USR_FIRST.TXT se načítá jako první a obsahuje informace o rozměrech zadávaných datových polí. Nejdůležitější položky jsou (zápis v souboru odpovídá názvům proměnných ve zdrojovém kódu):

NoTetras – počet elementů sítě (čtyřstěnů)

NoVertices – počet vrcholů

NoBoundaries – počet stěn elementů na hranici oblasti

NoSubdomains – počet těles, resp. oblastí s různými materiálovými vlastnostmi

NoContactBoundaries – počet (dvojic) stěn elementů na kontaktní hranici

NoContactPairs – počet dvojic vrcholů na kontaktní hranici (tj. dvojic, na kterých je realizována kontaktní podmínka nepronikání)

Materiálové vlastnosti jsou zadány v souboru `USR_MATERIALS.TXT`, zápis položek odpovídá datové struktuře uvnitř programu, index `I` odpovídá číslu podoblasti (tělesa):

(Weight(J,I), J=1,3)

vektor síly působící na jednotku hmotnosti

EStiff(I), SigStiff(I)

Youngův modul pružnosti E a Poissonova konstanta σ

RhoDensity(I), AlphaDamping(I)

materiálová hustota ρ a tlumící koeficient α

Hodnoty polí popisujících geometrii sítě, jsou zadány v souboru `USR_MESH.TXT`. Zápis položek opět odpovídá zápisu uvnitř zdrojového kódu:

VX(I), VY(I), VZ(I)

souřadnice vrcholu s indexem `I`, `I=1..NoVertices`

(ITNODE(J,I), J=1,5)

indexy vrcholů (`J=1..4`) čtyřstěnu `I`, `I=1..NoTetras`;

číslo podoblasti (`J=5`) ke které čtyřstěn náleží

(IBNDRY(J,I), J=1,5)

indexy vrcholů (`J=1..3`) stěny `I`, `I=1..NoBoundaries`;

typ okrajové podmínky (`J=4`) zadané na této stěně:

1 homogenní (nulová) Neumannova o.p.

2 nehomogenní Neumannova o.p., tj. předepsané zatížení

-1 Dirichletova o.p., tj. předepsané posunutí

-2 unilaterální kontakt

-3 (bilaterální kontakt, v této verzi neimplementováno)

0 vnitřní stěna, tj. stěna mezi oblastmi, které jsou součástí stejného tělesa (ale liší se např. materiálovými vlastnostmi)

(ICCB(J,I), J=1,3)

indexy stěn (`J=1,2`) přiléhajících k sobě na kontaktní hranici, `I=1..NoContactBoundaries`

(ICP(J,I), J=1,2)

indexy vrcholů tvořících kontaktní dvojici, `I=1..NoContactPairs`

CPX(I), CPY(I), CPZ(I)

souřadnice normálového vektoru příslušejícího ke kontaktní dvojici, `I=1..NoContactPairs`

Časová diskretizace je řízena hodnotami zadanými v souboru `TIME.IN`, v následujícím pořadí:

bTmDiscretisation – určuje chování programu:

- 0 výpočet stacionárního řešení, tj. žádná časová diskretizace, hodnoty okrajových podmínek se uvažují v čase $dTime=dTmStart$
- 2 výpočet dynamické úlohy
- 1 žádný výpočet (načte datové struktury a exportuje síť ve výstupním formátu, bez hodnot řešení)

$dTmStart$ – začátek časového intervalu

$dTmEnd$ – konec časového intervalu

$dTimeStep$ – velikost časového kroku

Řešení u_h^k spočtené na každé časové hladině t_k je zapisováno do výstupního souboru `gmw/out.gmw.XXXX`. Výstupní soubor je určen pro vizualizaci v programu `Gmw` (*General Mesh Viewer*), který lze nalézt na internetové adrese Los Alamos National Laboratory: <http://laws.lanl.gov/XCM/gmw/>, viz [17]. Soubor obsahuje popis sítě konečných prvků a vektor posunutí v každém vrcholu sítě. Geometrie oblasti je deformována podle aktuálních hodnot řešení, tj. souřadnice vrcholů sítě jsou posunuty o vektor posunutí spočtený v daném časovém kroce, přenásobený konstantou `DisplacementMagnification` určenou v podprogramu `OUT2GMV3()`.

4 Příklad použití

Motivací pro vývoj kódu byly zejména aplikace v biomechanice kloubů a jejich náhrad, program byl tedy během vývoje testován také na jednoduchých akademických úlohách, které se podobají skutečným aplikacím. Jako příklad zde uvádíme dva takové modely. Obě uváděné geometrie byly připraveny pomocí programu `NETGEN` (viz <http://www.hpfem.jku.at/netgen>), kterým byla vygenerována síť konečných prvků. Takto vytvořená síť byla převedena do vstupního formátu programu `FEC` pomocí software `Matlab`.

Na obrázku 4.1 je vyobrazena geometrie jednoduchého modelu, složeného z kulové hlavičky uložené v mělké jamce. Mezi vsazenou hlavičkou a jamkou je zadána kontaktní podmínka, vnější povrch jamky je pevně uchycen (je na něm předepsáno nulové posunutí) zatímco horní ploška hlavičky je manipulována v čase předepsaným posunutím. Konkrétně pro zde uvedený pokus, je horní ploška hlavičky posunuta v rámci roviny xy ; je posouvána plynule v časovém intervalu od $t = 0$ do $t = 0.005$ kde se její pohyb zastaví. Na ostatních částech povrchu je předepsáno nulové zatížení. Na stránce 13 jsou vyobrazeny dva pohledy na řešení v časech $t = 0.0$, 0.002 , 0.004 a $t = 0.006$. V levé části je celkový pohled, zobrazené hodnoty odpovídají posunutí ve směru osy y . Vpravo jsou zobrazeny vektory posunutí, řez je veden v rovině $\{x = 0\}$.

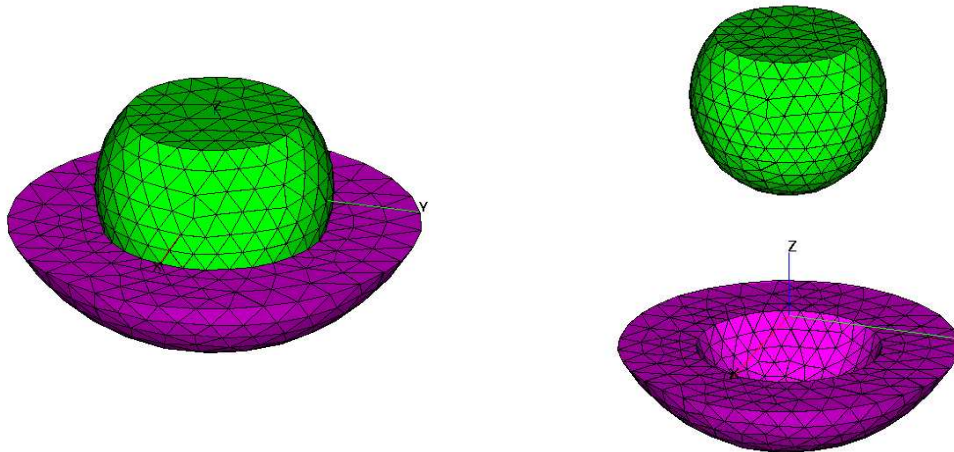
Uvedený model svou strukturou vzdáleně odpovídá studovaným modelům kloubního spojení v kyčelním kloubu, či kontaktu mezi hlavičkou a jamkou v umělém kyčelním kloubu.

Spuštění a běh programu vypadá zhruba následujícím způsobem: program se spustí v adresáři obsahujícím všechny zdrojové soubory, před započítím časového cyklu proběhne načtení dat a sestavení datových struktur nezáviselých na časové hladině.

```

NoTetras= 2655
NoVertices= 825
NoBoundaries= 1344
NoSubdomains= 2
NoBCPrimalNodes= 0
NoPrimalBoundaries= 1
NSVR= 0
NoBCCrossNodes= 0
NoContactBoundaries= 222
NoContactPairs= 127
NoBiContactBoundaries= 0
NoBiContactPairs= 0

```



Obrázek 4.1:

```

NoPointSurfForces= 0
[export GMV problem setting to gmv/out.gmv.0000    ...done]
Time: -0.01 -- 0.03 by 0.001
[export GMV boundary conditions to gmv/out.gmv.5000 ...done]
ELSTIFF: element no 1000
ELSTIFF: element no 2000
RStiff: element no. 1
RStiff: element no. 1001
RStiff: element no. 2001
AssRearX: MR entry no 1
AssRearX: MR entry no 50001
AssRearX: MR entry no 1
AssRearX: MR entry no 50001

```

NV	NoTetras	NEQ	NCP	LIC	LJA
825	2655	1842	127	29374	762

Údaje na posledním řádku značí, v uvedeném pořadí: počet vrcholů sítě, počet elementů, výsledný počet rovnic (tj. počet stupňů volnosti, dimenze vektoru řešení), počet kontaktních dvojic. LIC, respektive LJA, značí počet nenulových prvků matice C, respektive matice A. Následuje časový cyklus, který v každém časovém kroku vypisuje průběh řešení minimalizační úlohy:

```

[***Time iteration*** no. 12 time= 0.002 ..
ITERATION No.    1
  errcgc= 0.78739E+04
  ill1= 127
  ill2= 127
  cgc it    51
  Dtime= 0.52992E-01
ITERATION No.    2
  errcgc= 0.10361E-05
  ill1= 127
  ill2= 126
  cgc it    25
  Dtime= 0.26996E-01
.
.

```

```

.
.
ITERATION No. 195
  errcgc= 0.46627E-03
  il1=    99
  il2=    98
  cgc it      2
  Dtime= 0.19989E-02
ITERATION No. 196
  errcgc= 0.64521E-03
  il1=    99
  il2=    98
  cgc it      2
  Dtime= 0.19999E-02
ITERATION No. 197
  errcgc= 0.59859E-03
  il1=    99
[***Time iteration*** no. 12 time= 0.002 ...done]
[export GMV solution to gmv/out.gmv.1012 ...done]

```

V tomto případě výsledný vektor řešení obsahoval 99 aktivních kontaktních podmínek (tedy těch dvojic vrcholů, na kterých platila rovnost normálových složek posunutí), výsledný vektor řešení byl zapsán do souboru `gmv/out.gmv.1012`.

Jako druhý příklad uvádíme geometrii, předjímající plánované studie kolenního kloubu. Jak je vidět na obrázku 4.2, model se opět skládá ze dvou těles ve vzájemném kontaktu. Spodní těleso je upevněno předepsáním nulového posunutí na ploše tvořící jeho dolní konec, zatímco horní těleso je tentokrát předepsáním posunutí na jeho vrchní ploše prudce stlačeno ve směrem dolů. Na ostatních částech povrchu obou těles je předepsáno nulové zatížení.

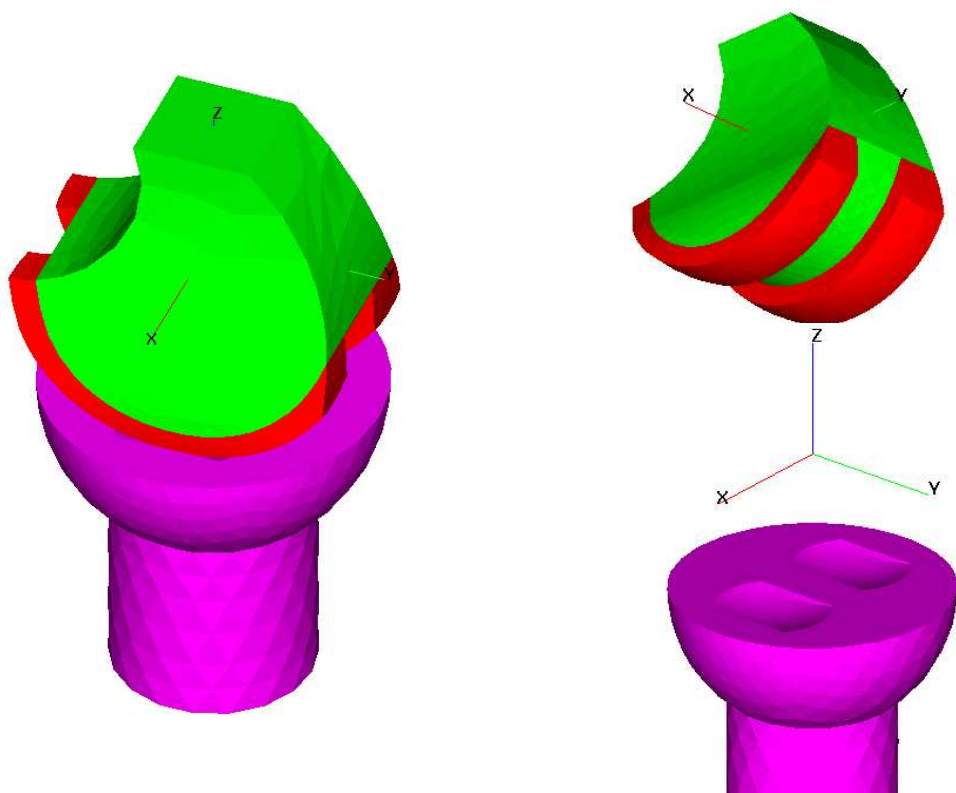
Na straně 14 je naznačena postupná distribuce deformace ve třech různých časových okamžicích. První řada znázorňuje hodnoty velikosti vektoru posunutí, další tři řady pak ukazují hodnoty posunutí ve směru osy x (tj. ve směru osy protínající oba kondyly). V dolních dvou řadách jsou od sebe tělesa oddělena jen pro názornost.

5 Shrnutí

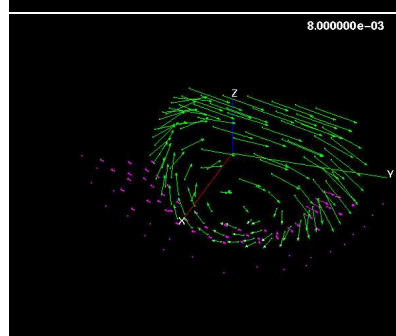
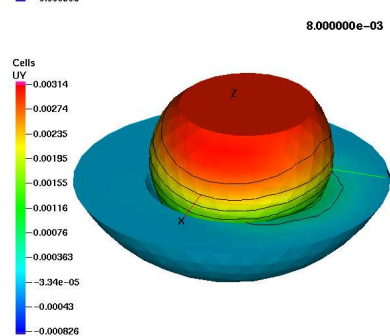
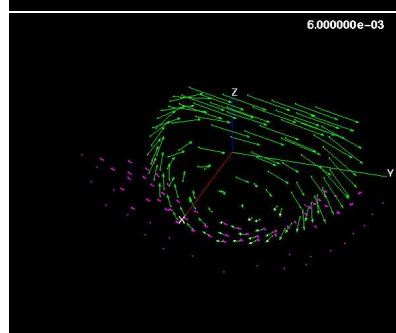
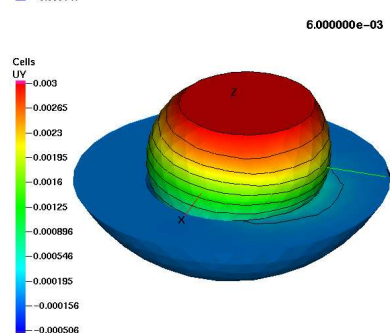
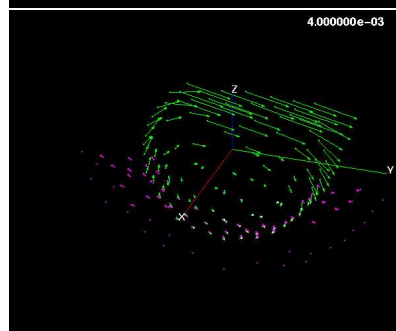
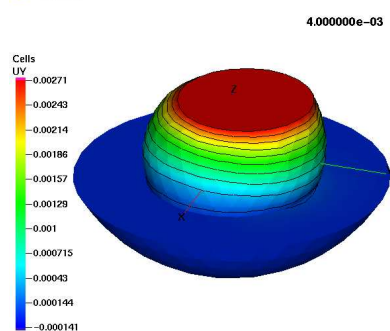
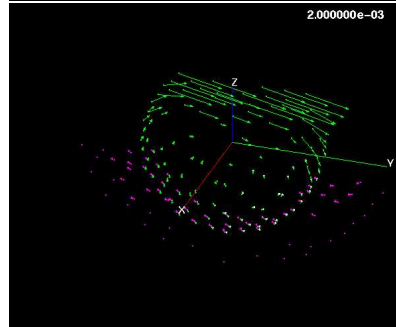
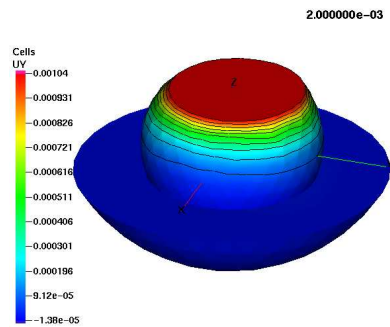
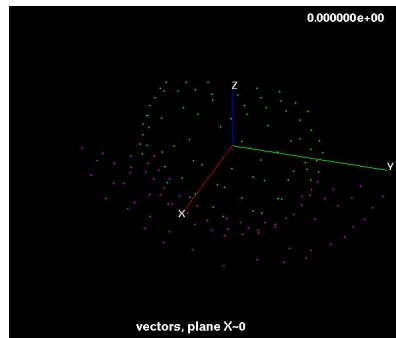
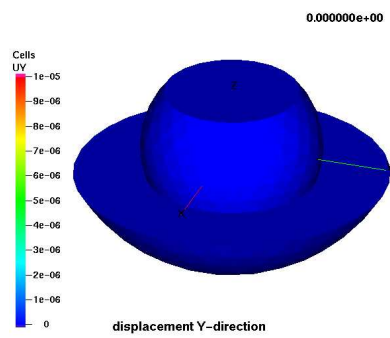
Předložená práce popisuje přístup, použitý k modifikaci kódu FEC za účelem numerického řešení dynamické kontaktní úlohy. Popisuje základní formulaci úlohy, zvolený způsob její diskretizace v čase a její aproximaci konečnými prvky založenou na variační formulaci. Jsou uvedeny hlavní změny vůči původní implementaci stacionární verze programu FEC, je popsán způsob použití programu a způsob zadání dat. V poslední části zprávy jsou uvedeny dva modelové příklady, na kterých byl program testován a které zároveň naznačují, jak může být vyvinutý software použit při studiu napjatostních poměrů v kloubních systémech.

Stávající verze je předvojem dále vyvíjené implementace, zahrnující model tření a vazko-pružnou reologii těles v kontaktu. Primární aplikací popisovaného programu a motivací k jeho vývoji je studium a stanovení podmínek stability umělých náhrad v lidském organismu a numerická analýza umělých náhrad kloubů ve vazbě na CT a MRI snímkování. Tyto studie, stejně jako zmíněná plánovaná rozšíření implementace, budou předmětem řešení v příštím roce.

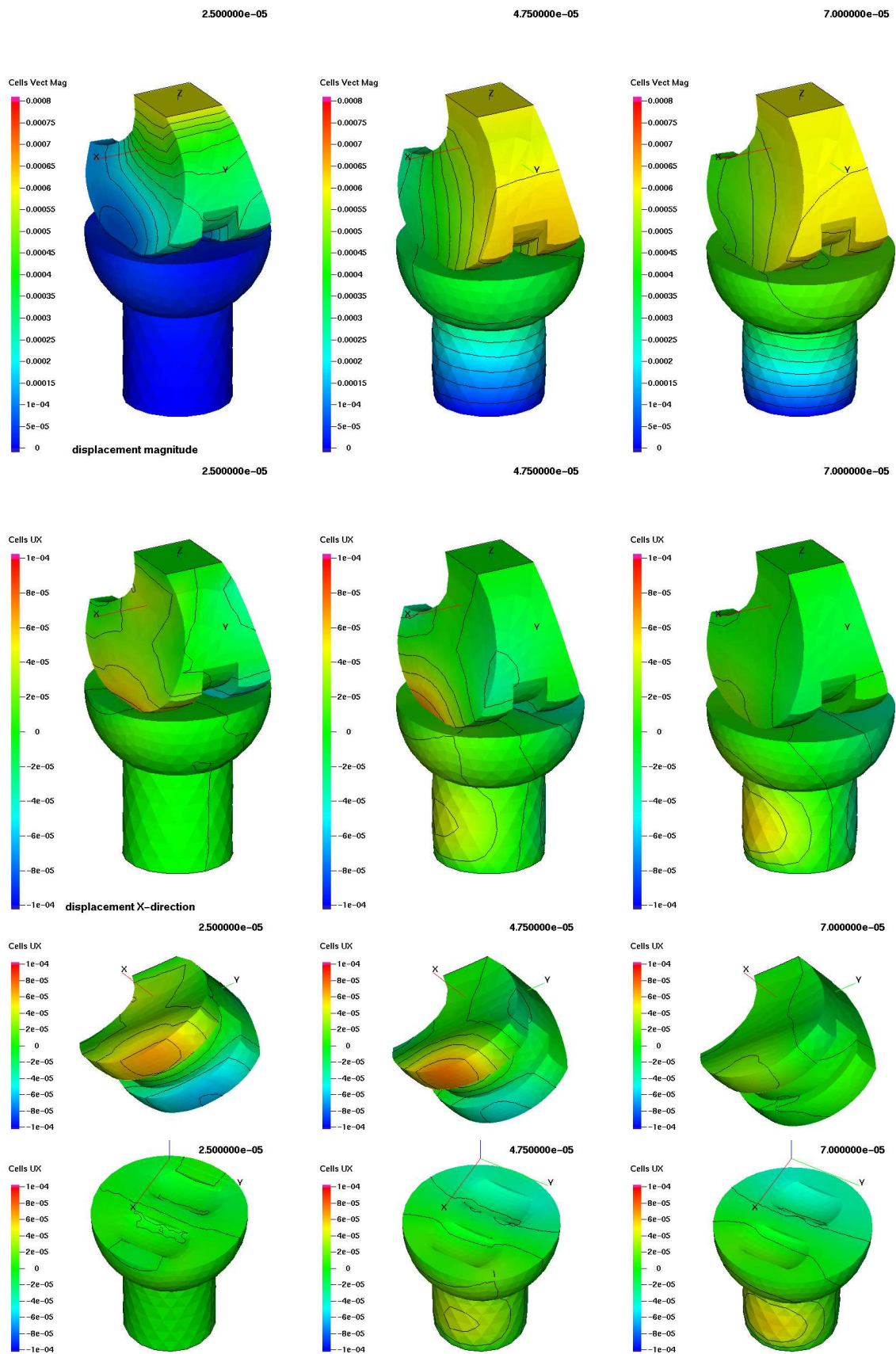
Předložená studie je výzkumnou zprávou projektu MPO ČR č. FT-TA/087 za rok 2006.



Obrázek 4.2:



Obrázek 4.3:



Obrázek 4.4:

Literatura

- [1] C. Eck and J. Jarušek, “Existence results for the static contact problem with coulomb friction,” *Math. Models Methods Appl. Sci.*, vol. 8, pp. 445–468, 1998.
- [2] R. M. Ferencz and T. J. R. Hughes, “Iterative finite element solutions in nonlinear solid mechanics,” *Handbook of Numerical Analysis*, eds. P.G. Ciarlet and J.L. Lions, vol. V, Numerical Methods for Solids, pp. 3–178, 1998.
- [3] J. Haslinger, I. Hlaváček, and J. Nečas, “Numerical methods for unilateral problems in solid mechanics,” *Handbook of Numerical Analysis*, eds. P.G. Ciarlet and J.L. Lions, vol. IV, pp. 313–485, 1996.
- [4] I. Hlaváček and J. Nedoma, “On a solution of a generalized semi-coercive contact problem in thermoelasticity,” *Math. and Computers in Simulation*, vol. 60, pp. 1–17, 2002.
- [5] Z. Kestřánek, “Numerical analysis of the contact problem, comparison of methods for finding the approximate solution,” Tech. Rep. No. 648, Institute of Computer Science, AS CR, September 1995.
- [6] Z. Kestřánek and J. Nedoma, “The conjugate projected gradient method – numerical tests and results,” Tech. Rep. No. 677, Institute of Computer Science, AS CR, August 1996.
- [7] J. Nedoma and Z. Kestřánek, “Preconditioners for contact problems in elasticity,” Tech. Rep. No. 739, Institute of Computer Science, AS CR, February 1998.
- [8] Z. Kestřánek and J. Nedoma, “FEC - a code for contact problems in thermoelasticity,” Tech. Rep. No. 740, Institute of Computer Science, AS CR, February 1998.
- [9] Z. Kestřánek, *Numerical analysis of the 3D contact problem of Signorini type with friction in thermoelasticity. H-version of the finite element approximation*. PhD thesis, Technical University Prague, 1999.
- [10] Z. Kestřánek, “Description of the finite element program for contact problem,” Tech. Rep. No. 831, Institute of Computer Science, AS CR, February 2001.
- [11] J. Nečas and I. Hlaváček, *Mathematical Theory of Elastic and Elasto-Plastic Bodies: An Introduction*. Elsevier, 1981.
- [12] J. Nedoma, “Matematické modely umělých náhrad kloubu ve vazbě na navigované operační techniky s využitím CT a MRI I. Dynamické zatěžování TEP a TKR, matematické 2D a 3D modely,” Tech. Rep. No. 950, Institute of Computer Science, AS CR, 2005.
- [13] J. Nedoma, “On the dynamic contact problem with Coulomb friction in nonlinear elasticity,” *připravuje se do tisku*, 2006.
- [14] J. Nedoma, “Solvability of dynamic multibody contact problems with Coulomb friction and damping in visco-elasticity with short memory,” *Připravuje se do tisku*, 2006.
- [15] J. Nedoma, J. Stehlík, M. Bartoš, F. Denk, V. Džupa, J. Fousek, I. Hlaváček, Z. Klézl, and I. Květ, *Biomechanika lidského skeletu a umělých náhrad jeho částí*. Universita Karlova v Praze – Nakladatelství Karolinum, 2006.

- [16] J. Nedoma, I. Hlaváček, J. Daněk, and M. Lanzendörfer, “Matematické modely umělých náhrad kloubu ve vazbě na navigované operační techniky s využitím CT a MRI II. Algoritmy umožňující numerickou analýzu umělých náhrad kloubu ve vazbě na CT, MRI a navigovanou operační techniku,” Tech. Rep. No. 951, Institute of Computer Science, AS CR, November 2005.
- [17] P. W. Mendius (Edited by), *GMV User Manual*. Los Alamos National Laboratory, <http://www-xdiv.lanl.gov/XCM/gmv/doc.color.pdf>, 2006.
- [18] M. H. Sadd, *Elasticity. Theory, applications and numerics*. Elsevier, 2005.