



## 6<sup>th</sup> International Conference **APLIMAT 2007**

Faculty of Mechanical Engineering - Slovak University of Technology in Bratislava

*Session: Open Source Software in Research and Education*

### **RIEŠENIE KONTAKTNÝCH PROBLÉMOV POUŽITÍM MEDZEROVÝCH KONEČNÝCH PRVKOV**

**ŽMINDÁK, Milan, MELICHER, Richard, ŠTEFANIČIAK, Maroš, (SK)**

**Abstrakt.** V tomto článku je prezentovaná implementácia medzerového prvku na riešenie 2D kontaktných problémov bez trenia. Je odvodený medzerový prvok a diskutované sú jeho vlastnosti. Pre diskretizáciu kontaktných telies sú použité lineárne štvoruholníkové prvky. Kontakt sa rieši iteračne úpravou Youngovho modulu pružnosti až kým sa dosiahne konvergencia vo všetkých medzerových prvkoch. Výsledky ilustračného príkladu sú porovnané s MKP programom ADINA.

**Kľúčové slová.** Kontakt telies, metóda konečných prvkov, medzerový prvok, kontaktný algoritmus

### **SOLUTION OF CONTACT PROBLEMS USING GAP FINITE ELEMENTS**

**Abstract.** A finite gap element implementation of 2D frictionless contact problems is presented in this paper. The gap element is derived and its properties are discussed. For discretization of contact bodies are used the linear quadrilateral elements. The contact is solved by iteration updating Young's modulus of elasticity as far as achieved converged solution in all gap elements. The results of illustrative example are compared with FEM program ADINA.

**Key words and phrases.** Contact of bodies, finite element method, gap elements, contact algorithm

*Mathematics Subject Classification.* 74-04; 74M15.

# 1 Úvod

Vo všeobecnosti kontaktné problémy v mechanike telies obsahujú trenie medzi dvoma a viacerými tuhými alebo poddajnými telesami. Mechanické zaťaženie sa potom prenáša interakciou medzi týmito telesami. Zložitosť riešenia kontaktu spočíva v nelineárnom charaktere kontaktu, aj v prípade malých deformácií a posunutí a pri elastickom správaní sa materiálu. Dôležitou podmienkou v riešení kontaktu je nepreniknuteľnosť telies. Táto podmienka sa dá do formulácie kontaktu zahrnúť v tvare nerovnic. Na splnenie podmienky nepreniknuteľnosti telies sa používajú dva prístupy: (1) ohraničenia na posunutia sa zahrnú priamo do formulácie; (2) použijú sa tzv. medzerové prvky (gap elements) na spojenie uzlov, ktoré môžu prísť do kontaktu.

V priamom prístupe musíme najprv identifikovať body kontaktu. Po identifikovaní bodov kontaktu, podmienku nepreniknuteľnosti môžeme zaviesť viacerými spôsobmi. Jednou z možností je definovať kontaktné sily použitím metódy Lagrangeových multiplikátorov alebo metódy pokutových funkcií [1]. Iná možnosť je riešiť kontakt ako problém matematického programovania. U metódy pokutových funkcií nesprávne zvolenie pokutového parametra môže spôsobiť prenikanie telies (nízka hodnota pokutového parametra) alebo numerické problémy (vysoký pokutový parameter). Metóda Lagrangeových multiplikátorov je časovo náročná, pretože je potrebné definovať ďalšie premenné (Lagrangeové multiplikátory), ktoré predstavujú kontaktné napätie. Veľmi efektívna metóda je metóda kontaktných podmienok v tvare rovníc [2, 3]. Zahrnutie kontaktných podmienok je v tejto metóde konzistentné s princípom virtuálnych prác pre MKP.

Druhá skupina využíva na splnenie podmienky nepreniknuteľnosti medzerové prvky (MP) s veľkou tuhosťou, ktoré spájajú páry uzlov v oblasti kontaktu [4]. Kontakt sa potom modeluje zadaním nelineárnej charakteristiky tuhosti elementu v závislosti na polohe uzlov. Pokiaľ medzera medzi kontaktnými telesami je nulová je aj tuhosť nulová. V opačnom prípade, t. j. keď dochádza ku kontaktu, medzera je zatvorená a tuhosť má nenulovú hodnotu. Poznamenajme, že nevýhodou použitia medzerových prvkov je, že musíme vedieť, kde nastane kontakt. Napríklad pre gumu, ktorá je z hyperelastického materiálu, je odhad polohy kontaktu zložitý, čo má za následok nesprávny prenos zaťaženia na povrchu kontaktných telies. Výhodou použitia MP je, že nevyžadujú zavádzanie nových neznámych alebo špeciálnych algoritmov riešenia. Preto sa algoritmy riešenia ľahko implementujú do existujúcich programov MKP [5].

Okrem MKP veľmi atraktívne je aj použitie metódy hraničných prvkov (MHP) [6]. Iné metódy využívajú výhody MHP a MKP a sú založené na Bettiho vete o reciprocite a Trefftzových funkciách [7,8]. V [9] sa používajú hybridné Trefftzovske (HT) konečné prvky a stykové (interface elements) Kaliakinove-Liove (KL) prvky. Štúdium metód založených na variačných rovniciach a nerovniciach a ich porovnanie s programom ANSYS je v [10]. Nové riešenia v mechanike kontaktu sú prezentované v [11].

Cieľom tohto príspevku je odvodenie efektívnej procedúry na analýzu 2D kontaktu použitím štvoruholníkových konečných prvkov a nelineárnych MP prvkov. Navrhnutý je

aj algoritmus riešenia a jeho implementácia v jazyku C++. Správnu funkčnosť algoritmu a porovnanie výsledkov s programom ADINA dokumentujú výsledky ilustračného príkladu.

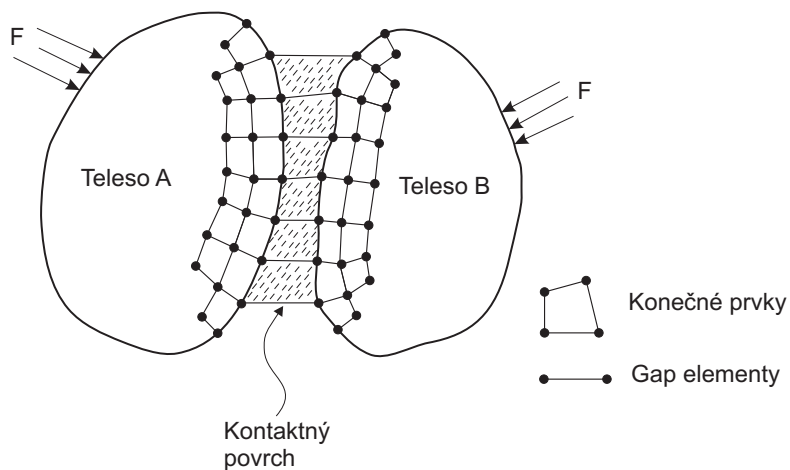
## 2 Vlastnosti medzerových prvkov

Na odvodenie procedúry na analýzu kontaktu telies prostredníctvom MP uvažujeme kontakt na obrázku 1. MP sa bude predlžovať alebo skracovať v závislosti od pohybu telies  $A$  a  $B$ . Jeho dĺžka sa stáva premennou medzerovou vzdialenosťou  $d$ . Je zřejmé, že k predlžovaniu MP dochádza, keď v prvku vzniká ťah, to znamená, že osová deformácia  $\varepsilon$  je kladná. Na druhej strane, MP prvok sa skracuje, keď  $\varepsilon$  je negatívne (tlakové). Všeobecný MP má nasledovné vlastnosti:

- $\varepsilon > -1,0$ ; je to stav pred kontaktom.
- $\varepsilon \leq -1,0$ ; dochádza ku kontaktu. V tomto prípade môže dôjsť k prešmykovaniu ak  $\tau \geq \mu |\sigma|$  alebo ide o šmyk ak  $\tau < \mu |\sigma|$ , kde  $\mu$  je koeficient trenia a  $\tau$  je šmykové napätie.
- $\varepsilon = -1,0$ ; hrúbka MP je nulová a telesá sú v kontakte v tejto oblasti. Keď  $\varepsilon < -1,0$  to znamená, že dochádza k preniknutiu telies a je potrebné modifikovať tuhosť.

Riešenie kontaktu je iteračný proces, ktorý môže byť nestabilný. Na dosiahnutie konvergencie v niekoľkých iteračných krokoch je potrebné definovať adekvátne konvergenčné kritériá, aby nedochádzalo k osciláciám riešenia.

Hore uvedené správanie sa MP sa používa na riešenie jednoduchých kontaktných problémov. Odvodenie prvkov s uvedenými vlastnosťami je problematické, keď medzera medzi telesami sa znižuje alebo sa rovná nule.



Obrázok 1: Kontakt telies použitím medzerových prvkov

Preto namiesto dĺžky MP zavádzame malý parameter  $\delta$  a kontaktný stav sa definuje spolu s veľkosťou medzery. Ďalej sa definuje parameter tzv. indikátor kontaktu  $\mu$

$$\mu = \frac{(d - \delta)}{\delta}. \quad (1)$$

Z rovnice (1) vidíme, že medzera medzi telesami je úplne zatvorená, keď  $\mu$  je rovné nule a telesá sú oddelené, keď  $\mu$  je väčšie ako nula. Ak hodnota indikátora je menšia ako nula, potom dochádza k preniknutiu telies. Je potrebné poznamenať, že ku kontaktu dochádza, keď  $\mu = 0$ , to znamená, že  $d = \delta$ . Nevýhodou MP je keď sa má riešiť kontakt veľmi elastickej konštrukcie, napr. z hyperelastického materiálu s veľmi tuhým telesom. Tu na začiatku kontaktu nepoznáme deformácie elastického telesa a predpokladáme iba páry uzlov, ktoré prídu do kontaktu. Takže pri veľkých deformáciách sú MP umelo donútené k spojeniu uzlov. V tomto prípade každý uzol na povrchu kontaktných telies, bez ohľadu na jeho pozíciu po deformácii môže prenášať silu, ktorá bola pôvodne pridelená MP. To má za následok nesprávny prenos síl.

### 3 Princíp algoritmu riešenia kontaktu

Princíp algoritmu riešenia kontaktu použitím MP je jednoduchý. Predpokladáme, že pred kontaktom je Youngov modul  $E$  zanedbateľne malý tak, že dochádza k pohybu telies. Na druhej strane, keď dôjde k prenikaniu telies, je potrebné uvažovať veľkú hodnotu  $E$ . To má za následok, že moduly jednotlivých prvkov je potrebné spojito meniť až kým element nie je v úplnom kontakte alebo  $E$  má nenulovú hodnotu.

Numerický iteračný algoritmus je odvodený za predpokladu, že osovú napätie v  $i$ -tom prvku  $\sigma^i$  je počas kontaktu konštantné. Potom pre 1. iteráciu platí

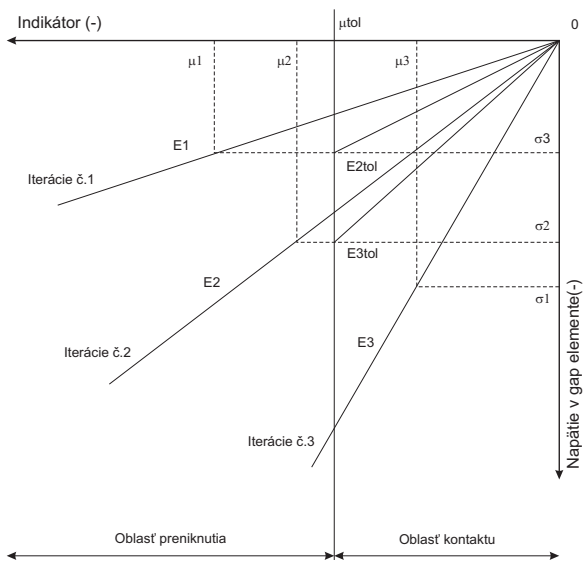
$$\sigma_1^i = E_1^i \mu_1^i = E_2^i \mu_{\text{tol}}, \quad (2)$$

kde  $\mu_{\text{tol}}$  je záporná kontaktná tolerancia s veľmi malou absolútnou hodnotou. Testy ukázali, že uspokojivé výsledky sa získajú pre  $\mu_{\text{tol}} = 0,05$ . Iteračný proces pre prvé tri iterácie je na obrázku 2.

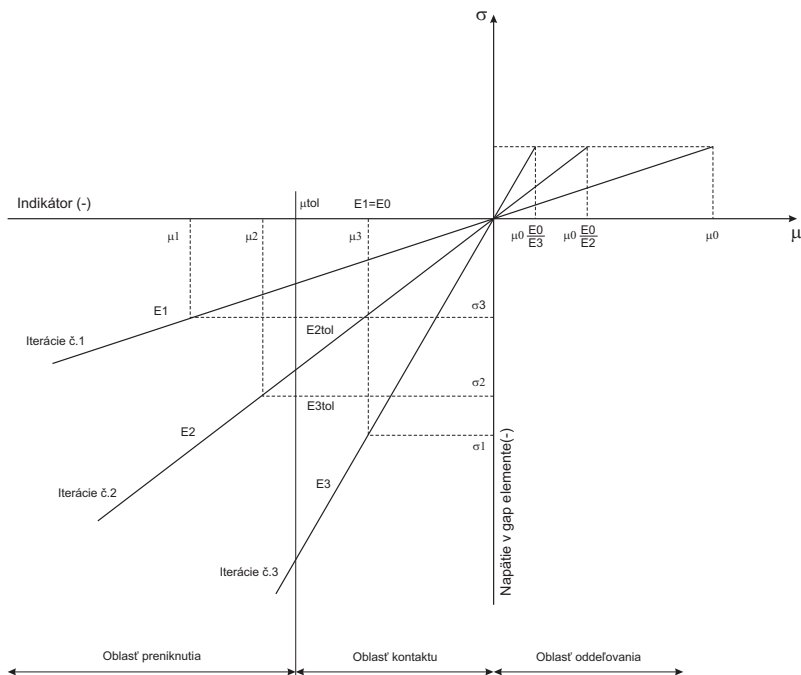
Keďže správanie sa jedného prvku závisí od správania sa ďalších prvkov, tak nasledujúca iterácia vo všeobecnosti nebude spĺňať podmienky kontaktu. Preto kvôli konvergencii sa použije väčšia hodnota  $E$ . Takže  $(k + 1)$ -vá iterácia bude

$$E_{k+1}^i = E_k^i \frac{\mu_k^i}{\mu_{\text{tol}}} (1 + \alpha), \quad (3)$$

kde  $\alpha$  je parameter zrýchlenia na stabilizovanie konvergence riešenia. Testy ukázali, že na dobrú konvergenciu sa odporúča použiť  $\alpha = 0,01$ . Poznamenajme, že stav každého prvku



Obrázok 2: Iteračný proces s modifikáciou Youngovho modulu



Obrázok 3: Iteračný proces pred kontaktom a v kontakte

môžeme ovplyvniť počiatočnými hodnotami indikátora  $\mu_0$  a  $E_0$ . Na obrázku 3 je znázornený iteračný proces pred kontaktom a v kontakte.

Po  $k$ -tej iterácii má indikátor kontaktu tvar

$$\mu_k^{i,acc} = \mu_k^i + \mu_0^i \frac{E_0^i}{E_k^i} \quad (4)$$

a odpovedajúci modul  $E$  prenikajúceho prvku je

$$E_{k+1}^i = \frac{E_0^i \mu_0^i + E_k^i \mu_k^i}{\mu_{tol}} (1 + \alpha). \quad (5)$$

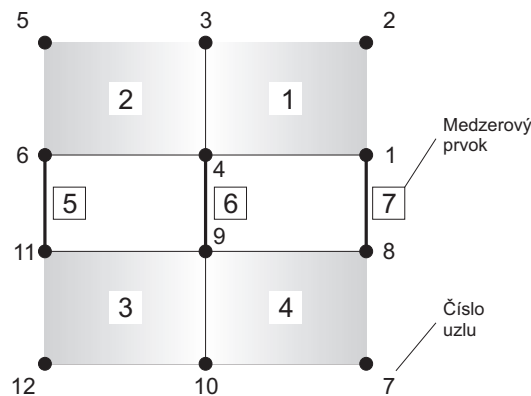
Na druhej strane, modul  $E$  naťahovaných prvkov sa modifikuje zavedením tzv. ťahovej redukčnej konštanty  $\beta$

$$E_{k+i}^i = \beta E_k^i. \quad (6)$$

Tu je potrebné riadiť minimálnu hodnotu  $E$ . Zmenšovanie  $E$  ovplyvňuje konvergenciu a jeho adekvátna redukcia závisí od geometrie a stupňoch zaťaženia v analyzovaných problémoch. Na základe numerických testov bola konvergencia dosiahnutá pre  $\beta = 0,1$  pre jednoduché prípady a  $\beta = 0,25$  pre všeobecnejšie prípady. Algoritmus riešenia je na obrázku 5.

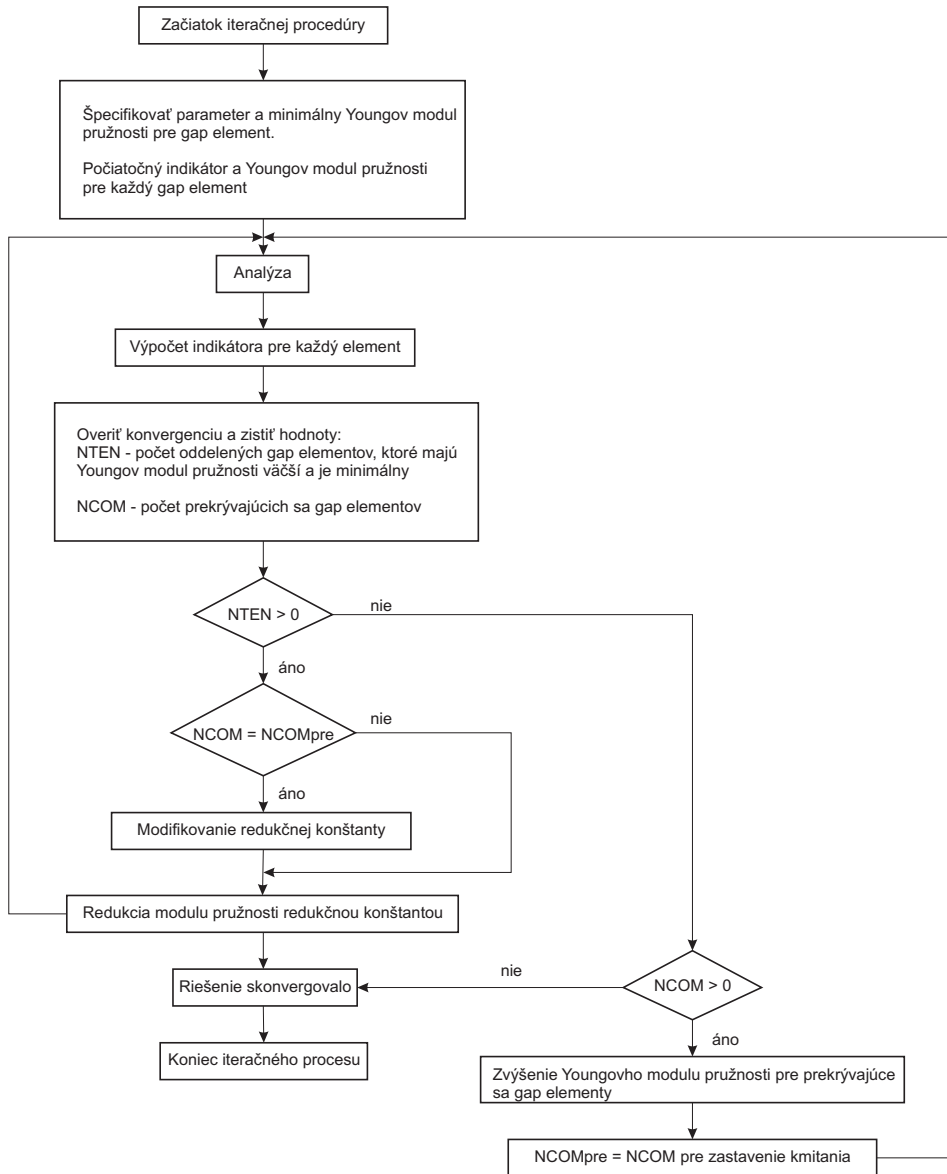
## 4 Popis programu a funkcií

Program používa manuálny preprocessing, t. j. všetky vstupné údaje sa zadávajú ručne pomocou klávesnice. Príklad siete s číslaním prvkov a uzlov je na obrázku 4.



Obrázok 4: Príklad siete prvkov a číslovanie prvkov a uzlov

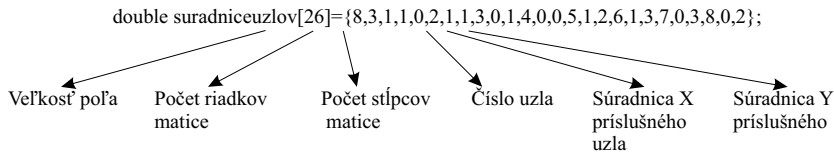
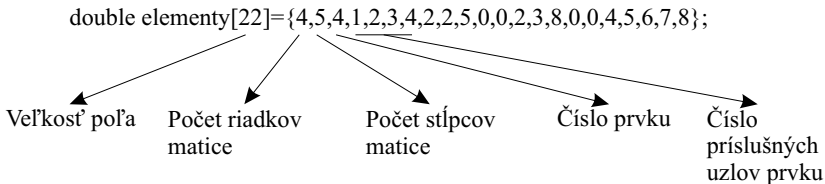
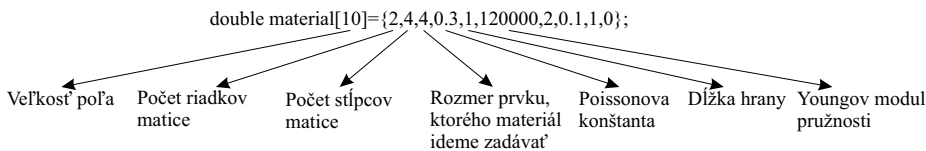
Do súboru `hlavny.cpp` zapíšeme súradnice uzlov do poľa `suradniceuzlov` (obrázok 6) od prvého uzla až po posledný, pričom prvé dve čísla v matici udávajú jej rozmery. Dôležité je uviesť veľkosť poľa (počet čísel v matici). Ďalšie pole, ktoré sa zadáva v súbore `hlavny.cpp`,



Obrázok 5: Algoritmus riešenia

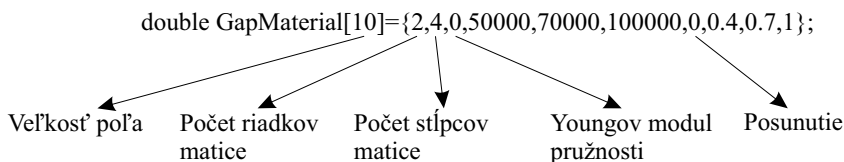
je pole `elementy` (obrázok 7). Do tohto poľa sa zapisujú čísla uzlov príslušného elementu (4 uzly). Keď zapisujeme MP, tak zapíšeme príslušné 2 uzly a dopíšeme dve nuly, aby sme dodržali veľkosť matice.

Nasleduje zadanie materiálu pre použité konečné prvky (obrázok 8). V tomto programe sú použité 2D štvoruzlové prvky [12] a medzerové prvky sú prútové nelineárne prvky [2].

Obrázok 6: Príklad zadania súradníc uzlov `suradniceuzlov`Obrázok 7: Príklad zadania príslušných uzlov prvku do `elementy`

Obrázok 8: Príklad zadania materiálu

Materiál sa zadáva do poľa `material`. Pre 2D prvok sa zadá Poissonova konštanta  $\nu$ , dĺžka hrany a modul  $E$ . Pre medzerový prvok je to veľkosť prierezu, dĺžka prúta a dodá sa nula, aby sa dodržal rozmer matice. Ďalšie potrebné materiálové vlastnosti MP prvkov sa zadávajú do poľa `GapMaterial` (obrázok 9). Ako bolo uvedené, tieto vlastnosti určujú, ako sa bude správať kontakt, čiže je potrebné zadať Youngove moduly  $E$  a k nim príslušné posunutia. Ďalšie pole potrebné na výpočet je pole okrajových podmienok uchytenia (obrázok 10). Do tohto poľa sa zapisujú len tie uzly, v ktorých sú určené nulové posunutia.

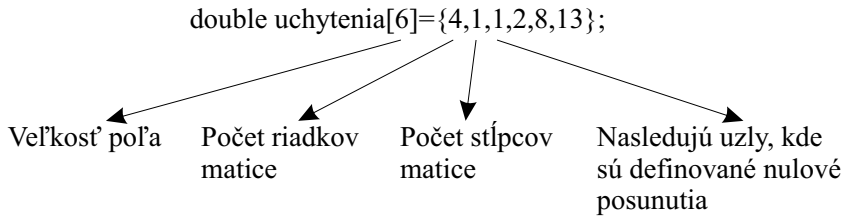


Obrázok 9: Príklad zadania dodatočného materiálu pre medzerové elementy

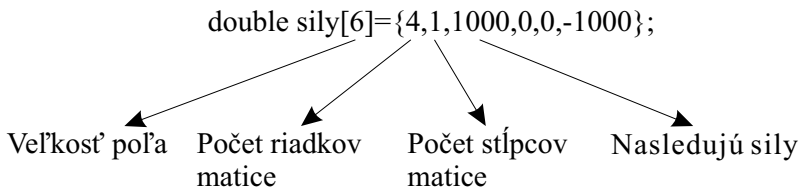
Posledné pole, ktoré potrebujeme na výpočet je pole zaťaženia `sily` (obrázok 11). Ak ide o sily, ktoré pôsobia v smere zápornom, silu zadáme tiež zápornú. Sily zadávame pre všetky uzly, t. j. aj sily v uzloch, ktoré nie sú zaťažené (nulové sily).

Po zadaní potrebných vstupných údajov nasleduje samotný výpočet. Tento vyžaduje kompilovanie súboru `hlavny.cpp` vo voľne šíriteľnom prekladači `Bloodshed Dev-C++`.





Obrázok 10: Príklad zadania uchytení



Obrázok 11: Príklad zadania síl

### Hlavný program používa funkcie:

- matrix.cpp,
- linmaticatuhostistvor.cpp,
- globalnamatica.cpp,
- nonlinmaticatuhostiprutu.cpp,
- linmaticatuhostiprutu.cpp.

Výstupom programu sú posunutia v uzlových bodoch. Algoritmy ďalších funkcií potrebných napr. na výpočet napätí sú napr. v [13, 14].

### Zoznam použitých funkcií:

- *PrintMatrix* – výpis matice.
- *InvMatrix* – inverzia matice.
- *SciMatrix* – sčítanie dvoch matíc.
- *OdcMatrix* – odčítanie dvoch matíc.
- *NasMatrix* – násobenie matíc.
- *TranMatrix* – transpozícia matice.
- *RiesicMatrix* – riešenie systému rovníc  $\mathbf{K} \mathbf{U} = \mathbf{F}$ .
- *LinMaticaTuhostiStvor* – má šesť parametrov. Prvé tri parametre sú Poissonovo číslo, dĺžka hrany a modul  $E$ . Ďalšie dva parametre sú súradnice uzlových bodov  $x$  a  $y$  v globálnom súradnicovom systéme. V poslednom je uložená matica tuhosti prvku.

- *LinMaticaTuhostiPrutu* – má päť parametrov. Prvé tri sú prierez, modul  $E$  a dĺžka. Nasleduje vektor súradníc. Ako posledný je parameter, do ktorého sa ukladá výsledok.
- *GlobalnaMatica* – táto funkcia volá funkcie *LinMaticaTuhostiPrutu* a *LinMaticaTuhostiStvor* na vytvorenie globálnej matice tuhosti. Jej prvý vstup je matica súradníc uzlov, ďalej je to matica prvkov a odpovedajúcich uzlových bodov. Ďalšia matica obsahuje materiálové vlastnosti. Predposledná matica je  $E$  a do poslednej matice sa ukladá globálna matica tuhosti celej konštrukcie.

## 5 Ilustračný príklad

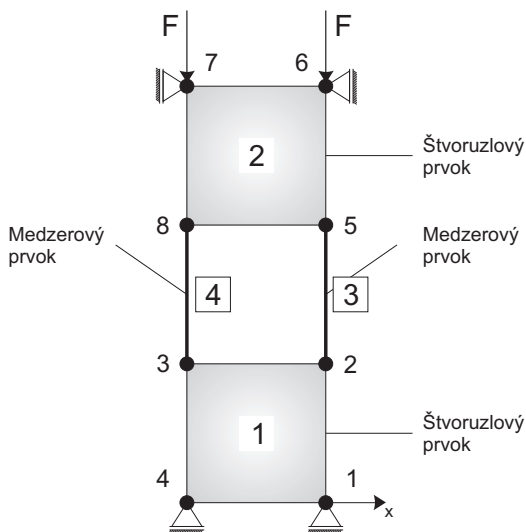
Na obrázku 12 je kontakt dvoch lineárnych štvoruzlových prvkov a dvoch MP. Nelineárna charakteristika prvku je daná v tabuľke 1. Výsledky posunutí v uzlových bodoch sú v tabuľke 2.

Tabuľka 1:

$E$ (Pa)	0,0	50 000	70 000	100 000
$\varepsilon$	0,0	0,4	0,7	1,0

Tabuľka 2:

Číslo uzlu	Posunutie
1	0
2	0
3	$-7,11508E-17$
4	0,166667
5	0,05
6	0,166667
7	0,05
8	0
9	$-0,0334559$
10	1,16667
11	0
12	1,32837
13	0
14	1,32837
15	0,0334559
16	1,16667



Obrázok 12: Príklad

## 6 Záver

V tomto článku bol vyvinutý 2D kontaktný MP. Správanie tohto prvku sa dá ovplyvňovať zadaním vzťahu medzi napätím a deformáciou a iteračný proces je založený na modifikácii modulu pružnosti  $E$ . Výhodou koncepcie použitia MP je, že nemusíme definovať žiadne ďalšie premenné. Nevýhodou použitia koncepcie MP je, že musíme vopred definovať páry bodov, ktoré prídu do kontaktu. Výsledky vyvinutého softvéru sa veľmi dobre zhodujú s programom ADINA na riešenie jednoduchých problémov. Štruktúra softvéru umožňuje dopĺňanie nových typov prvkov kontaktných telies algoritmov a metód. S využitím niektorých funkcií na riešenie zložitejších problémov kontaktu sa uvažuje v systéme IVS MPK [5]. Vyvinutý softvér je open source softvér (OSS) a jeho zdrojové texty sú prístupné na URL adrese [ftp://frcatel.fri.uniza.sk/pub/research/Gap\\_contact.zip](ftp://frcatel.fri.uniza.sk/pub/research/Gap_contact.zip).

### PodĎakovanie

Autori práce ďakujú Agentúre pre podporu vedy a techniky (APVT) za podporu tejto práce (Projekt č. APVT-20-003102).

### Literatúra

- [1] Zhong, Z. H.: *Finite Element Procedures for Contact – Impact Problems*. Oxford University Press, 1993
- [2] Bathe, K. J.: *Finite Element Procedures*. Prentice Hall, Inc. New Jersey, 1996
- [3] Bathe, K. J. – Ellabbasi, N. – Hong, J. W.: *On the reliable solution of contact problems in engineering design*. International Journal of Mechanics and Materials in design, 1, 3–16, 2004
- [4] Bai, X. – Zhao, X.: *Analysis of Large Deformation Elastoplastic Contact through Finite Gap Elements*. Computers and Structures, Vol. 30, No. 4, pp. 975–978, 1988
- [5] Eisner, I. – Lizoň, I. – Melicher, R. – Žmindák, M.: *The application of interactive computational system (ICS FEM) in engineering*. In: Engineering mechanics 2006, Žďár nad Sázavou, Czech Republic
- [6] Vodička, R.: *Boundary Integral Equation Systems for some kinds of Contact Problems*. Strojnícky časopis, 52,4, s. 193–207, 2001
- [7] Letrich, M. – Novák, P. – Žmindák, M.: *Elastický viactelesový kontakt - tuhostná charakteristika valčekového ložiska*. In: Zborník z konferencie CO-MAT-TECH 2004, Trnava, 2004

- [8] Fraštia, Ľ. – Handrik, M.: *Numerický výpočet kontaktu elastických telies pomocou MKP s využitím Trefftzových funkcií*. In: *Nekonvenčné technológie 2006*, Žilina, 2006
- [9] Wang, Y. K. – Dhanasekar, M. – Han, J. J. – Qin, Q. H. – Kang, L. Y.: *Contact modeling and interface finite elements*. In: *10-th International Conference in Numerical Methods in Continuum Mechanics & 4-th workshop on Trefftz Methods (CD ROM)*. Žilina, Slovak Republic, 2005
- [10] Mijar, R. A. – Arora, J. S.: *Study of Variational Inequality and Equality Formulations for Elastic Frictional Contact Problems*. *Archives of Computational Methods in Engineering – State of the art reviews*. Vol. 7, 4, s. 387–449, 2000
- [11] Jaeger, J.: *New Solutions in Contact Mechanics*. WIT Press, 2005
- [12] Sága, M. – Vávro, J. – Kopecký, M.: *Počítačová analýza a optimalizácia mechanických sústav*. ZUSI, Žilina, 2002
- [13] Smith, I. M. – Griffiths, D. V.: *Programming the Finite Element Method, 4-th ed.* John Wiley and Sons Ltd., 2004
- [14] Chessa, J.: *Programming the Finite Element Method with Matlab*. Northwestern University, 2002 <http://www.tam.northwestern.edu/jfc795/Matlab/>

## Kontaktná adresa

**Milan Žmindák, (Doc., Ing., CSc.),**

Žilinská univerzita v Žiline, Strojnícka fakulta,

Katedra aplikovanej mechaniky,

Univerzitná 1, 010 26 Žilina,

tel.: +4215132962, e-mail: Milan.Zmindak@fstroj.utc.sk

**Richard Melicher, (Ing.),**

Žilinská univerzita v Žiline, Strojnícka fakulta,

Katedra aplikovanej mechaniky,

Univerzitná 1, 010 26 Žilina,

tel.: +421 5132954, e-mail: Richard.Melicher@fstroj.utc.sk

**Maroš Štefaničiak, (Ing.),**

Jána Hollého 1058,

034 95 Likavka, email: Maros.Stefaniciak@yahoo.com

**Príloha: Zdrojový text hlavného programu**

```
#include "matrix.cpp"
#include "globalnamatica.cpp"
//PrintMatrix(vstup)
//InvMatrix(vstup,vystup)
//NasMatrix(vstup,vstup,vystup)
//SciMatrix(vstup,vstup,vystup)
//OdcMatrix(vstup,vstup,vystup)
//TranMatrix(vstup,vystup)
int main()
{
double
suradniceuzlov[26]={8,3,1,1,0,2,1,1,3,0,1,4,0,0,5,1,2,6,1,3,7,0,3,8,0,2
};
double elementy[22]={4,5,4,1,2,3,4,2,2,5,0,0,2,3,8,0,0,4,5,6,7,8};
double material[10]={2,4,4,0.3,1,120000,2,0.1,1,0};
double GapMaterial[10]={2,4,0,50000,70000,100000,0,0.4,0.7,1};
double uchytenia[7]={5,1,1,2,8,11,13};
double posunutia[18]={0};
posunutia[0]=16;
posunutia[1]=1;
double sigma[18]={0};
sigma[0]=16;
sigma[1]=1;
double sily[18]={0};
sily[0]=16;
sily[1]=1;
sily[13]=10000;
sily[15]=10000;
double PrirastokSily = 1;
int PocetPrutov=0;
for (int j=0;j<int(elementy[0]);j++)
    if (elementy[j*int(elementy[1])+2]==2)
        PocetPrutov++;
int NodyPrutov[PocetPrutov*2+2];
NodyPrutov[0]=PocetPrutov;
NodyPrutov[1]=2;
int counter=0;
for (int j=0;j<int(elementy[0]);j++)
    if (elementy[j*int(elementy[1])+2]==2)
    {
        for (int u=0;u<2;u++)
            NodyPrutov[counter*2+u+2]=int(elementy[j*int(elementy
[1])+3+u]);counter++;
    }
double ModulPruznosti=((GapMaterial[3]-GapMaterial[2])/(GapMaterial[7]
-GapMaterial[6]));
int NapatovaKonstanta=3;
int SilovaKonstanta=0;
double NapatieVPrutoch[counter];
```

```

for (int i=0;i<counter;i++)
  NapatieVPrutoch[i]=0;
for (int i=int(PrirastokSily);i<=10000;i++)
  {
    if (NapatieVPrutoch[0]>=GapMaterial[NapatovaKonstanta])
      {
        ModulPruznosti=((GapMaterial[NapatovaKonstanta+1]-GapMaterial[
NapatovaKonstanta])/(GapMaterial[NapatovaKonstanta+1+4]-GapMaterial[
NapatovaKonstanta+4]));
        NapatovaKonstanta++;
      }
double GlobMatica[258]={0};
GlobalnaMatica(suradniceuzlov,elementy,material,ModulPruznosti,
GlobMatica);
sily[13]=PrirastokSily;
sily[15]=PrirastokSily;
double pos[int(posunutia[0])+2];
for (int j=0;j<(int(posunutia[0])+2);j++)
  pos[j]=0;
RiesicMatrix(GlobMatica,sily,uchytenia,pos);
for (int j=0;j<(int(posunutia[0]));j++)
  posunutia[j+2]=posunutia[j+2]+pos[j+2];
for (int j=0;j<counter;j++)
  NapatieVPrutoch[j]=0;
for (int j=0;j<posunutia[0];j++)
  {
    for (int u=0;u<(int(NodyPrutov[0])*int(NodyPrutov[1]));u++)
      if (NodyPrutov[u+2]*2 == j+1  $\text{\textbackslash}$ vert  $\text{\textbackslash}$ vert  $\text{\$}$  NodyPrutov[u+2]*2-1
== j+1)
        {
          sigma[j+2]=sigma[j+2]+ModulPruznosti*pos[j+2]/material[8];
        }
  }
if (i==PrirastokSily)
  for (int j=0;j<sily[0];j++)
    if (sily[j+2]!=0)
      {
        if ((j+2) == j/2+j/2+2)
          {
            SilovaKonstanta=-1;
            break;
          }
        else
          {
            SilovaKonstanta=0;
            break;
          }
      }
  }
for (int k=0;k<counter;k++)
  {

```

```
for (int j=0;j<sigma[0];j++)
  for (int u=0;u<NodyPrutov[1];u++)
    if ((NodyPrutov[u+2+k*int(NodyPrutov[0])])*2 == j+1)
      {
        NapatieVPrutoch[k]=sigma[j+2+SilovaKonstanta]-NapatieVPrutoch[k];
      }
  }
}
PrintMatrix(posunutia);
return 0;
}
```

## **6<sup>th</sup> INTERNATIONAL CONFERENCE APLIMAT**

Section Open Source Software in Research and Education

February 6–9, 2007

Bratislava, Slovakia

**Organizers:** Michal Kaukič and Miloš Šrámek

**Reviewers:** Ján Buša, Michal Kaukič, Dušan Mamrilla, Peter Mann, Andrej Petráš, Karel Šotek and Miloš Šrámek

**Editors:** Michal Kaukič, Miloš Šrámek, Ladislav Ševčovič and Ján Buša

ISBN 978-80-969562-7-2

Zborník bol vydaný s podporou SKOSI, n. o.

---

Copyright ©2007 autori príspevkov

Ktokoľvek má dovolenie vyhotoviť alebo distribuovať doslovný opis tohoto dokumentu alebo jeho časti akýmkoľvek médiom za predpokladu, že bude zachované oznámenie o copyrighte a o tom, že distribútor príjemcovi poskytuje povolenie na ďalšie šírenie, a to v rovnakej podobe, akú má toto oznámenie.