

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A
BIOMECHANIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND
BIOMECHANICS

APLIKACE ALGORITMU METODY KONEČNÝCH PRVKŮ NA ROVINNOU ÚLOHU

APPLICATION OF FEM TO THE PLANE TASK

BAKALÁŘSKÁ PRÁCE
BACHELOR THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ PAVLŮ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. TOMÁŠ NÁVRAT, Ph.D.

BRNO 2008

ANOTACE

Cílem práce je naprogramovat algoritmus metody konečných prvků pro jednoduchou rovinnou úlohu. Programový systém ANSYS bude využit pro přípravu modelu, pro zobrazení vypočtených výsledků a pro verifikaci algoritmu

ANNOTATION

Purpose of this work is to programmed FEM algorithm for evaluation of simple plane task. Software solution ANSYS will be used for preparation of model and verification of our algorithm.

Obsah:

Zadání závěrečné práce	3
Anotace	5
1 Úvod	8
1.1 Teorie MKP	8
1.1.1 Historie MKP	8
1.1.2 Matematický základ MKP	8
1.2 Přehled používaných MKP programu	15
2 Příprava vstupních dat pomocí programu ANSYS	18
2.1 Vytvoření modelu	18
2.2 Výpočet deformace, export dat	22
3 Program PLANE42	27
3.1 Vývojový diagram programu	27
3.2 Rozbor vývojového diagramu	27
4 Popis uživatelského rozhraní PLANE42	29
4.1 Načtení hodnot	30
4.2 Styly zobrazení	31
4.3 Nastavení modelu a výpočet deformace	33
5 Srovnání výsledků s programem ANSYS	35
5.1 Srovnání modelu pro rovinnou napjatost	35
5.2 Srovnání modelu pro rovinnou deformaci	37
6 Závěr	39
Seznam použité literatury	40

1 Úvod

1.1 Teorie MKP

1.1.1 Historie MKP

Za počátek vývoje metody konečných prvků lze považovat práce německého matematika Richarda Couranta a ruského inženýra Alexandra Hrennikoffa [4].

Svoje práce napsali nezávisle na sobě a též jejich postupy se do značné míry lišily. Společným prvkem obou prací byla myšlenka rozdělit spojité těleso na diskrétní části, avšak v samotném rozdělení se již rozcházel. Zatímco Alexander Hrennikoff rozděloval těleso na mřížku, Richard Courant využil k rozdělení diskrétní trojúhelníkovou síť, přičemž toto rozdělení využil k řešení eliptických parciálních diferenciálních rovnic, které vznikly u problému krutu válce.

Metoda získala přesný matematický základ v práci amerických matematiků George Fixe a Gilberta Stranga [5] a to pomohlo její generalizaci a použití v jiných oblastech jako je dynamika tekutin nebo elektromagnetismus.

1.1.2 Matematický základ MKP

Základní myšlenka MKP vychází z rozdělení spojitěho problému na spoustu menších diskrétních částí, které nazýváme elementy. S trochou zjednodušení se jedná o diskretizaci spojitě N rozměrné úlohy.

Při nasazení algoritmu MKP na úlohu pružnosti-pevnosti musíme nejdříve určit matici D, která nám udává závislost napětí na přetvoření na nekonečně malém prvku a k tomu je potřeba několik základních veličin.

Veličiny a rovnice z pružnosti-pevnosti

15 neznámých proměnných

tři posuvy u, v, w

šest přetvoření $\varepsilon_x, \varepsilon_y, \varepsilon_z, \gamma_{xy}, \gamma_{yz}, \gamma_{zx}$

šest napětí $\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{zx}$

15 rovnic (podmínek)

rovnice rovnováhy

představují rovnice vnitřní rovnováhy prvku a vnější objemovou sílu například gravitační, odstředivá a magnetické pole atd....

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} + o_x = 0 \quad (1.1.1)$$

$$\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} + o_y = 0 \quad (1.1.2)$$

$$\frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_z}{\partial z} + o_z = 0 \quad (1.1.3)$$

rovnice geometrické

jsou rovnice udávající závislost posuvů a přetvoření

$$\varepsilon_x = \frac{\partial u}{\partial x} \quad \varepsilon_y = \frac{\partial v}{\partial y} \quad \varepsilon_z = \frac{\partial w}{\partial z} \quad (1.1.4)$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \quad \gamma_{yz} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \quad \gamma_{zx} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \quad (1.1.5)$$

konstitutivní vztahy

jsou vztahy mezi deformací a napjatostí pro lineárně pružný, izotropní Hookovský materiál.

$$\begin{aligned} \varepsilon_x &= \frac{1}{E} [\sigma_x - \mu(\sigma_y + \sigma_z)] & \gamma_{xy} &= \frac{1}{G} \tau_{xy} \\ \varepsilon_y &= \frac{1}{E} [\sigma_y - \mu(\sigma_x + \sigma_z)] & \gamma_{yz} &= \frac{1}{G} \tau_{yz} \\ \varepsilon_z &= \frac{1}{E} [\sigma_z - \mu(\sigma_x + \sigma_y)] & \gamma_{zx} &= \frac{1}{G} \tau_{zx} \end{aligned} \quad (1.1.6)$$

Modul pružnosti ve smyku G je materiálová veličina, kterou lze vypočítat pomocí vztahu

$$G = \frac{E}{2(1+\mu)} \quad (1.1.7)$$

a vlastní matici D určíme podle $\sigma = D \cdot \varepsilon$ (1.1.8)

$$D = \begin{bmatrix} A & B & B \\ B & A & B \\ B & B & A \end{bmatrix} \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix} \quad \sigma = \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} \quad (1.1.9)$$

$$A = 2G + \lambda = \frac{E}{1+\mu} \cdot \frac{1-\mu}{1-2\mu} \quad (1.1.10)$$

$$B = \lambda = \frac{E\mu}{(1+\mu) \cdot (1+\mu)} \quad (1.1.11)$$

$$D = \frac{E}{1-\mu^2} \begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1-\mu}{2} \end{bmatrix} \quad \varepsilon = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \end{bmatrix} \quad \sigma = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} \quad (1.1.12)$$

pro rovinnou napjatost

$$D = \frac{E \cdot (1 - \mu)}{(1 + \mu)(1 - 2\mu)} \begin{bmatrix} 1 & \frac{\mu}{1 - \mu} & 0 \\ \frac{\mu}{1 - \mu} & 1 & 0 \\ 0 & 0 & \frac{1 - 2\mu}{2 \cdot (1 - \mu)} \end{bmatrix} \quad \varepsilon = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} \quad \sigma = \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} \quad (1.1.13)$$

pro rovinnou deformaci

nyiní musíme urřit vlastní tvary funkcí uvnitř prvku

určení bázové funkce pro bod i

Tvarová (bázová) funkce musí být nad prvkem lineární. V bodě i bude 1 a v bodech j a k musí být 0. Z toho vyplývá, že vezmeme natočenou rovinu v prostoru, která je určena třemi body. V bodě i bude $z=1$ a v bodech j a k bude $z=0$. Osy x a y budou souřadnice

$$\text{bodů } x \text{ a } y. \text{ Funkci roviny urříme jednoduše } z_x = \frac{2 \cdot S}{ly} \quad \frac{1}{z_x} = \frac{ly}{2 \cdot S} \quad (1.1.14)$$

$$f_x = 1 - \frac{ly}{2 \cdot S} \cdot x \quad (1.1.15)$$

$$\text{stejným způsobem urřím } z_y = \frac{2 \cdot S}{lx} \quad \frac{1}{z_y} = \frac{lx}{2 \cdot S} \quad f_y = 1 - \frac{lx}{2 \cdot S} \cdot y \quad (1.1.16)$$

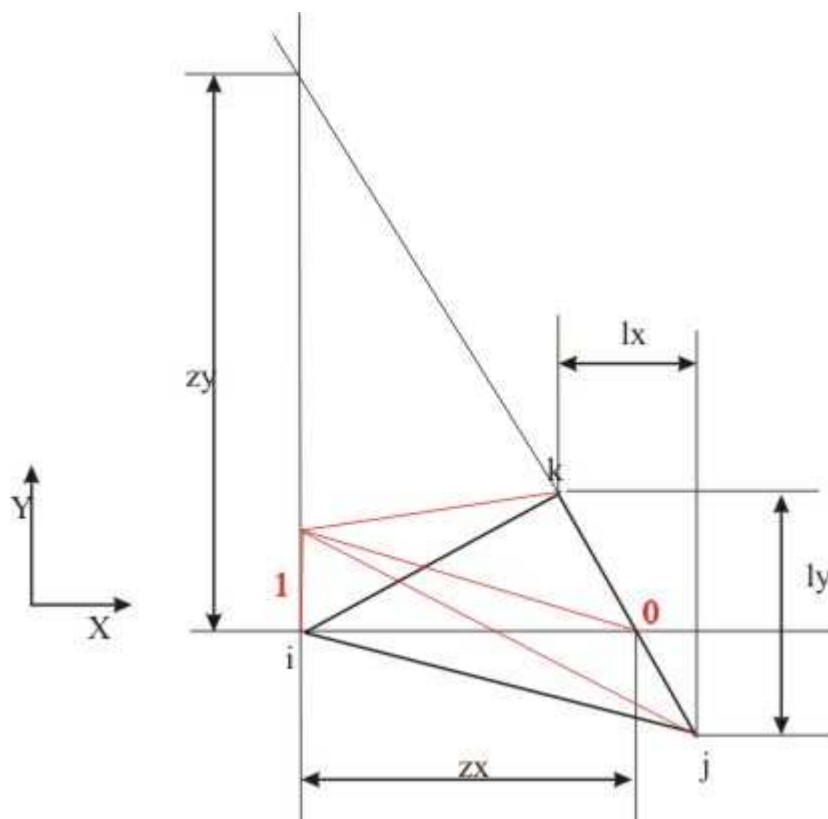
plochu trojúhelníka urříme snadno pomocí vektorů \underline{ij} a \underline{ik}

$$2 \cdot S = (iy - jy) \cdot (ix - xy) - (iy - ky) \cdot (ix - kx) \quad (1.1.17)$$

$$ly = k \cdot y - j \cdot y \quad lx = j \cdot x - k \cdot x \quad (1.1.18)$$

a poté složím dohromady

$$N1 = 1 - \frac{lx}{2 \cdot S} \cdot y - \frac{ly}{2 \cdot S} \cdot x \quad (1.1.19)$$



Obr. 1 určení bázové funkce

$$\mathbf{N}_u = [N_1 \quad N_2 \quad N_3] \mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{N} \cdot \boldsymbol{\delta} \quad \boldsymbol{\delta} = \begin{bmatrix} u1 \\ v1 \\ u2 \\ v2 \\ u3 \\ v3 \end{bmatrix} \quad (1.1.20)$$

$$\mathbf{N} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix} \quad (1.1.21)$$

$$\mathbf{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \quad (1.1.22)$$

maticy \mathbf{B} udává tvar funkce přetvoření a to uděláme tak, že postupně zderivujeme bázové funkce. A jelikož jsou naše bázové funkce lineární, bude Matice \mathbf{B} obsahovat konstantní hodnoty.

$$\mathbf{B} = \mathbf{L} \cdot \mathbf{N} \quad (1.1.23)$$

Vnitřní energie napjatosti trojúhelníku je dána

$$W_k = \frac{1}{2} \iint_s \sigma \varepsilon t_k dx dy \quad (1.1.24)$$

dosadíme napětí a přetvoření pomocí matic

$$\sigma = D \cdot B \cdot \delta = \delta^T \cdot B^T \cdot D \quad \varepsilon = B \cdot \delta \quad (1.1.25)$$

$$W_k = \frac{1}{2} \delta^T \left(\iint_s B^T D B t_k dx dy \right) \delta \quad W_k = \frac{1}{2} \delta^T K \delta \quad \Rightarrow K = \iint_s B^T D B t_k dx dy \quad (1.1.26)$$

a vyjádříme si matici tuhosti K (jelikož jsou B, D, t_k konstanty, integrujeme pouze plochu trojúhelníku)

$$K_k = t_k S_k B^T D_k B \quad (1.1.27)$$

S_k je plocha trojúhelníku

t_k je tloušťka trojúhelníku

Vytvoření tabulky pro globální kódová čísla neznámých

Nejprve potřebujeme přiřadit u každého uzlu (nodu) čísla pro jeho neznámé. To uděláme tak že začneme od prvního uzlu a přiřadíme číslo neznámé deformaci u a poté deformaci v . Jestliže je na deformaci uzlu přiřazena vazba, pak vložíme nulu, v jiném případě číslo vždy o jedno větší než předchozí.

Uzel	1	2	3	4	5	6	7	8
Index u	0	1	3	5	7	8	10	12
Index v	0	2	4	6	0	9	11	13

Vytvoření tabulky pro globální kódová čísla prvků

Z tabulky globálních kódových čísel neznámých vybereme čísla neznámých pro každý uzel prvku a vložíme je do řádku tabulky. Toto provedeme pro všechny prvky našeho modelu.

Prvek	Lokální indexy					
	1(ui)	2(vi)	3(uj)	4(vj)	5(uk)	6(vk)
1	1	2	0	0	7	0
2	0	0	3	4	8	9
3	3	4	10	11	12	13

Vytvoření lokální matice K

Vybíráme postupně všechny prvky (elementy) a pomocí souřadnic vrcholů jednotlivých trojúhelníků sestavíme matici tuhosti, zvlášť pro každý element a vždy po jejím sestavení ji zapíšeme do globální matice tuhosti.

Vytvoření globální matice K

Do globální matice tuhosti vkládáme lokální matice, tak že postupně vybíráme prvky lokální matice 6×6 a souřadnice vloženého místa vybíráme z tabulky globálních kódových čísel prvků. Její velikost je $n \times n$ kde n je počet neznámých

Například matice pro prvek 1 v naší tabulce pozice 1:5 odpovídá pozici v globální matici souřadnice 1:7

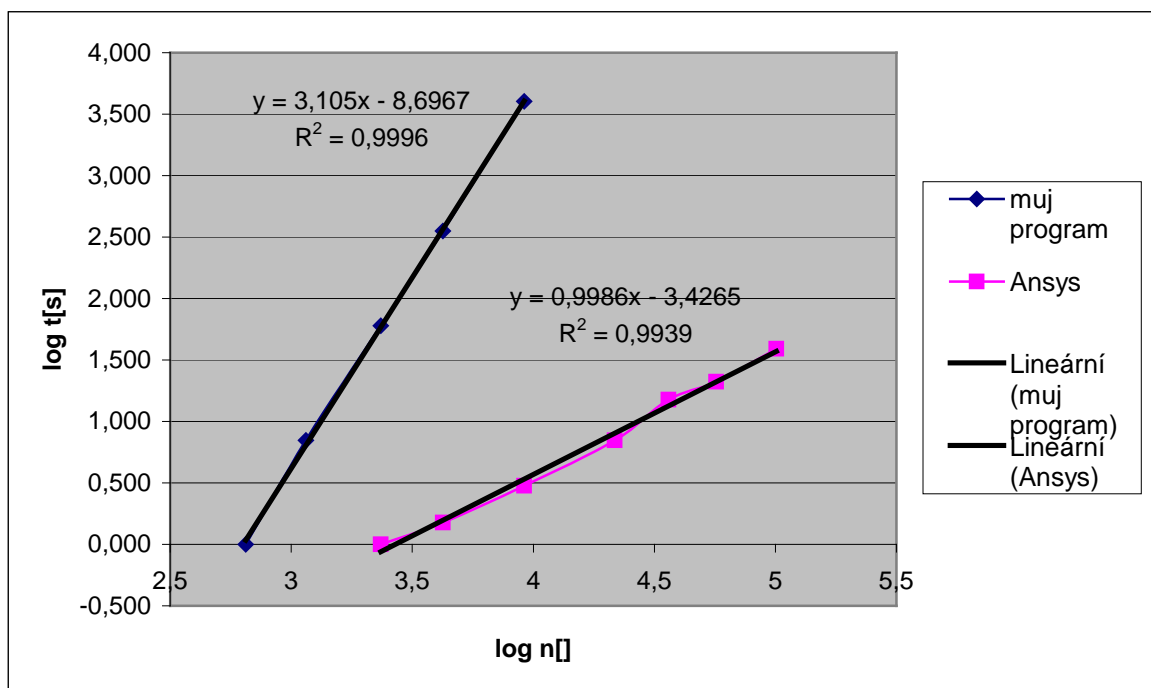
Samozřejmě že pro větší soustavy je čtvercová matice nevýhodná, protože její velikost roste kvadraticky a roste i počet prvků, které musíme projít k eliminaci jedné neznámé. Neboli když máme matici pro 1000 prvků, musíme sestavit matici 1000×1000 a pro její vyřešení postupujeme postupnou eliminací. Musíme pro eliminaci první proměnné přechíst celý řádek (vynásobit poměrem odečítané proměnné) a odečíst ho od nižšího. S trochou zjednodušení musíme operaci odečtení provést 1000krát pro první proměnnou na jenom řádku a na všech 1000ci řádcích to znamená 1000×1000 operací pro první 999×999 pro druhou proměnnou atd.

A to celé musíme provést pro všech 1000 proměnných, to znamená zhruba nárůst s třetí mocninou.

Proto musíme nějak matici zmenšit. Nejjednodušší je skládat pouze trojúhelníkovou matici, protože matice tuhosti je symetrická, ale tímto ušetříme pouze polovinu což je nedostatečné a proto se skládá matice pásová. Šířka pásu je závislá na počtu neznámých prvků a na tom, kolik prvků obsahuje společný uzel (nod). Důležité je, že šířka pásové matice je velmi malá a skoro nezávislá na zvyšování počtu prvků. Naproti tomu u pásové matice 1000 např. tloušťka 12 obsahuje pouze 12×1000 prvků. K eliminaci jednoho stačí projít 12 řádků po 12 operacích poměrového odečtení na každý. To znamená, že pokud se nám nezvedne tloušťka pásu, tak nám s počtem neznámých narůstá doba řešení lineárně, místo s třetí mocninou. To lze ukázat na porovnání mé úlohy, kde používám čtvercovou matici a ANSYSU, který používá pásovou matici.

můj čas	1	7	60	355	4020				
Ansys čas	1	1	1	1,5	3	7	15	21	39
nody	338	592	1199	2153	4641	10928	18200	28600	50600
Neznámé	648	1148	2346	4236	9180	21698	36197	56945	100861

log můj čas	0,000	0,845	1,778	2,550	3,604				
log Ansys čas	0,000	0,000	0,000	0,176	0,477	0,845	1,176	1,322	1,591
log neznámé	2,812	3,060	3,370	3,627	3,963	4,336	4,559	4,755	5,004



Obr. 2 logaritmická závislost výpočetní doby na počtu neznámých

Tento graf je zobrazen v logaritmických závislostech a udává závislost potřebného času pro řešení, na počtu neznámých v úloze. Tyto časy jsou na 2GHz počítači, ale na jeho rychlosti tu nezáleží, protože tu jde o porovnání stupně závislosti

Na tomto příkladu je vidět, že opravdu čas pro řešení pásové matice roste lineárně a čas pro řešení čtvercové matice jde s třetí mocninou. Z toho vyplývá, že pásová matice je pro větší soustavy nutností. Já jsem zvolil matici čtvercovou pouze pro její snadnou implementaci, protože pro sestavení pásové matice se musí neznámé v soustavě setřídit podle speciálních algoritmů. Například už tím, že pořadí bodů v trojúhelníku bereme podle otáčení hodinových ručiček, se nám většina hodnot (ne všechny) přesune směrem k diagonále matice.

Samozřejmě existuje spousta jiných řešičů. Například iterativní metoda. Ta pracuje na principu, že nějak nastavíme naše neznámé (výsledné hodnoty deformací) a spočítáme přes matici K jejich silový výsledek, který porovnáme se zadanými silami a vypočítáváme gradient, pomocí kterého upravujeme naši odhadovanou deformaci. Toto provádíme, dokud nesplníme naše požadavky na přesnost nebo výpočetní dobu. Tato metoda je sice daleko rychlejší než přímé metody, ale zase je její přesnost závislá na době, kterou necháme soustavu řešit a nemůžeme předem přesně říci, jak dlouho budeme pro požadovanou přesnost počítat. A v některých případech se nemusíme ani dobrat k výsledku - to souvisí s problémy iterativních metod, jako je uvíznutí v lokálním minimu atd. Samozřejmě i toto se dá nějakými způsoby řešit, ale vnáší to do řešení spoustu problémů a neurčitelnost. Přes své nevýhody jsou ale pro svou vysokou rychlost s úspěchem využívány.

1.2 Přehled používaných MKP programu

V dnešní době existuje velké množství softwaru, které aplikuje MKP na nejrůznější problémy v technické praxi. Zde je přehled některých z nich.

ABAQUS

ABAQUS/Standard

Umožňuje řešit

- Lineární problémy
- Nelineární problémy

Materiálové modely

- Elastické
- Plastické
- Creepové

Elementy

- všechny základní topologie
- nekonečné (infinite) prvky
- uživatelsky definované elementy
- speciální elementy (kapalinové, pneumatické, kontaktní...)

ABAQUS/Explicit – rychlé dynamické děje a velká přetvoření včetně destruktivních poškození

ABAQUS/Aqua – modul pro výpočty s tekutinami

ABAQUS/Pre - preprocessor

ABAQUS/Post – postprocessor

<http://www.simulia.com/>

LS-DYNA

Umožňuje řešit

- Nelineární dynamika
- Lineární analýzy
- Teplotní analýzy
- Fluidní analýzy
- Kontaktní úlohy

Materiálové modely

- Kovy
- Plasty
- Keramika
- Kapaliny

Knihovna elementů

- všechny základní topologie
- beamy s nulovou délkou
- Nodal masses

<http://www2.lstc.com/>

ANSYS

Umožňuje řešit	-Lineární/nelineární úlohy -Statické/Dynamické úlohy -Fluidní dynamiku -Teplotní, elektrostatické, magnetické úlohy
Materiálové modely	-isotropní/anisotropní -creepové/viskoelasticitní -kompozitní -hyperelastické
Knihovna elementů	-všechny základní topologie -kontrolované (přepínané) -uživatelsky definované elementy

<http://www.Ansys.com>

Nastran

Umožňuje řešit	-Lineární/nelineární úlohy -Statické/Dynamické úlohy -Fluidní dynamiku -Teplotní úlohy
Materiálové modely	-isotropní/anisotropní -creepové -kompozitní -hyperelastické
Knihovna elementů	-všechny základní topologie -lana -automatické kontaktní prvky -interpolační

<http://www.nastran.com/engineeringsoftware/femap>

Pro/Mechanica

Modul Pro/Mechanica je součástí systému Pro/ENGINEER[®] a řeší strukturální chování v oblasti statiky, vlastních frekvencí, dynamiky, vzpěrné stability, kontaktu, velkých deformací, teplotních zátěžových stavů, dále životnosti

free FEM řešiče

FELyX - v Finite Element library experiment je objektově orientovaný kód napsaný v C++ obsahuje některé elementy pro mechaniku pevných látek, redukci šířky pásu matice, skyline řešič a několik iteračních řešičů.

Getfem++ tento projekt poskytuje knihovnu v C++ pro konečně prvkové metody umožňuje řešit lineární i nelineární problémy libovolné dimenze, také neobsahuje implementovaný pre a postprocesor, ale s rozhraním pro Matlab a Python a umí importovat formát z několika free meshovacích programů.

<http://home.gna.org/getfem/>

OOFEM – objektově orientovaný konečně prvkový řešič pro řešení úloh z mechaniky pevných látek a mechaniky tekutin.

<http://www.oofem.org/>

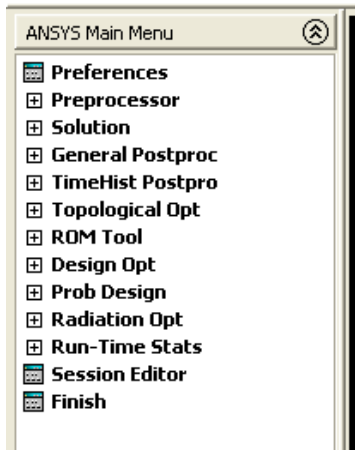
Impact - OpenSource MKP řešič pro velké rychlosti a velké deformace jako nárazy aut a tváření lisováním

<http://impact.sourceforge.net/>

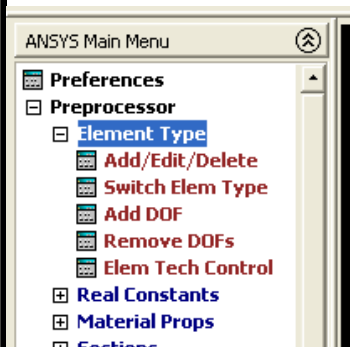
2 PŘÍPRAVA VSTUPNÍCH DAT POMOCÍ PROGRAMU ANSYS

2.1 Vytvoření modelu

Pro přípravu vstupních dat musíme nejdříve v programu ANSYS vybrat vhodný prvek. To uděláme tak, že otevřeme nastavení *Preprocessor* a v něm *Element Type* poté stiskneme *Add/Edit/Delete*.

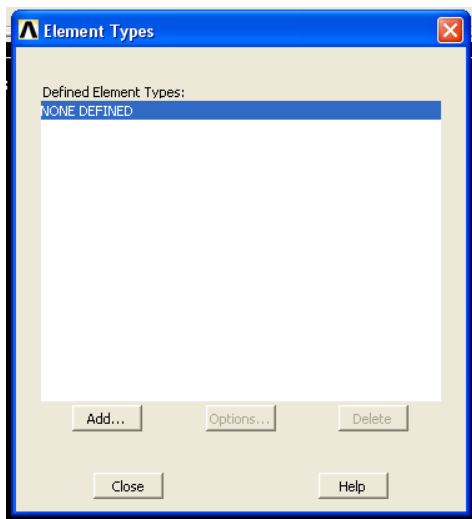


Obr. 3



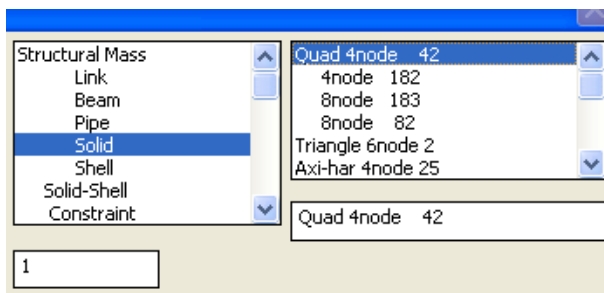
Obr. 4

Otevře se nám zobrazení načtených typů prvků, tam klikneme na *Add*.



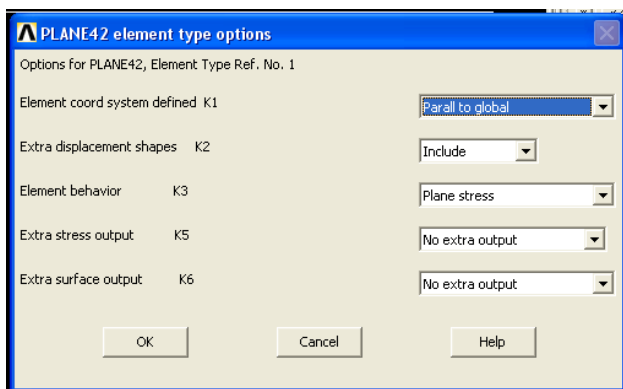
Obr. 5

V následujícím okně vybereme *Structural Mass – solid - Quad 4node 42*.



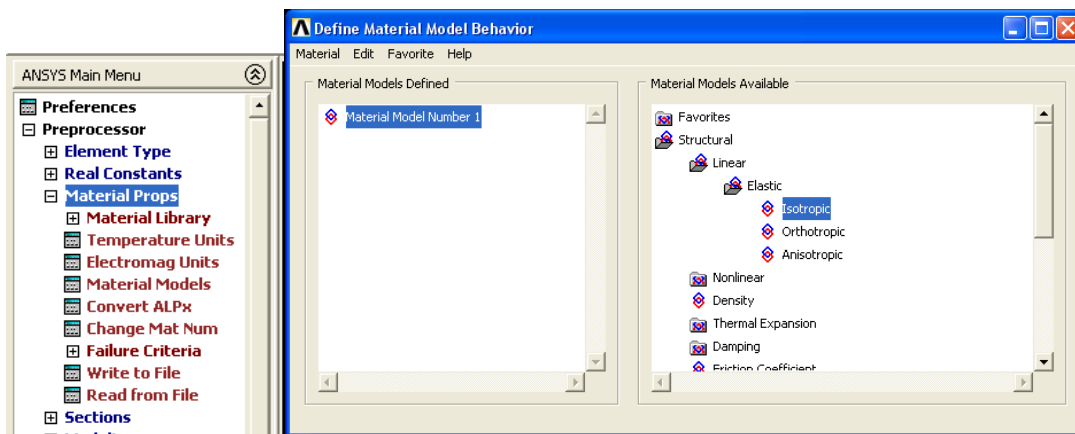
Obr. 6

Poté se vrátíme opět do zobrazení načtených typů prvků a pod tlačítkem *options* můžeme nastavit, jestli chceme řešit rovinnou napjatost, nebo rovinnou deformaci.



Obr. 7

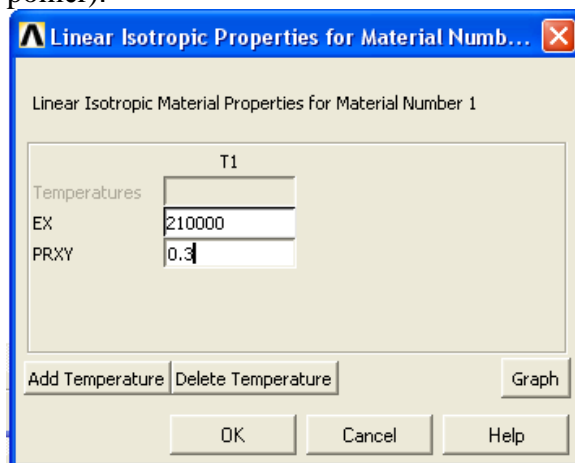
Jestliže máme vybrán prvek tak si musíme nastavit materiál. Ten nastavujeme v *Preprocessor – Material Props* kde klikneme na *Material Models*. Potom se nám otevře okno pro výběr materiálu, tam vybereme *Structural – Linear – Isotropic*.



Obr. 8

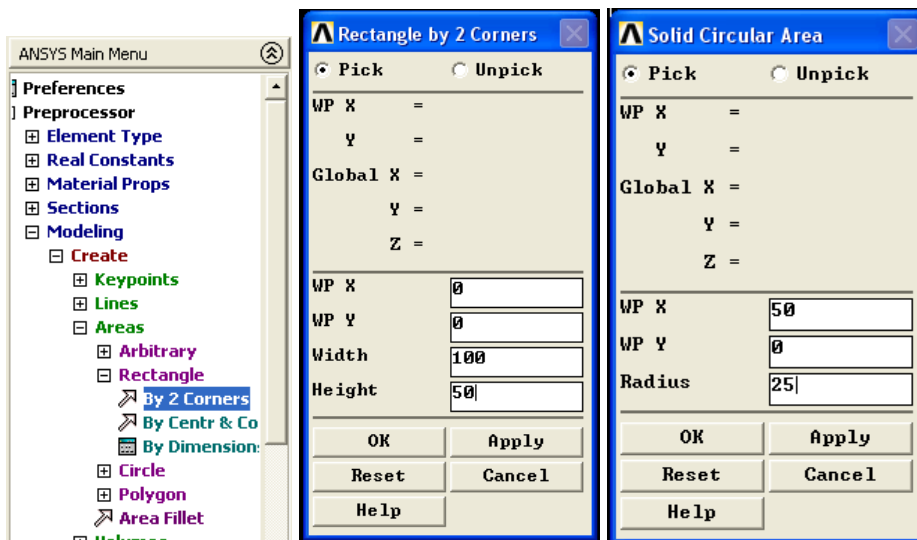
Obr. 9

Potom nastavíme parametry tohoto materiálu, tedy EX (tuhost), PRXY (Poissonův poměr).



Obr. 10

Jestliže máme nastavování za sebou, přejdeme k vlastní tvorbě modelu. Tvorba modelu se provádí opět v preprocesoru v položce *modeling - create*, tam vybereme například *rectangle a by 2 Corners* pro vytvoření obdélníku a *Circle* pro vytvoření kruhu.

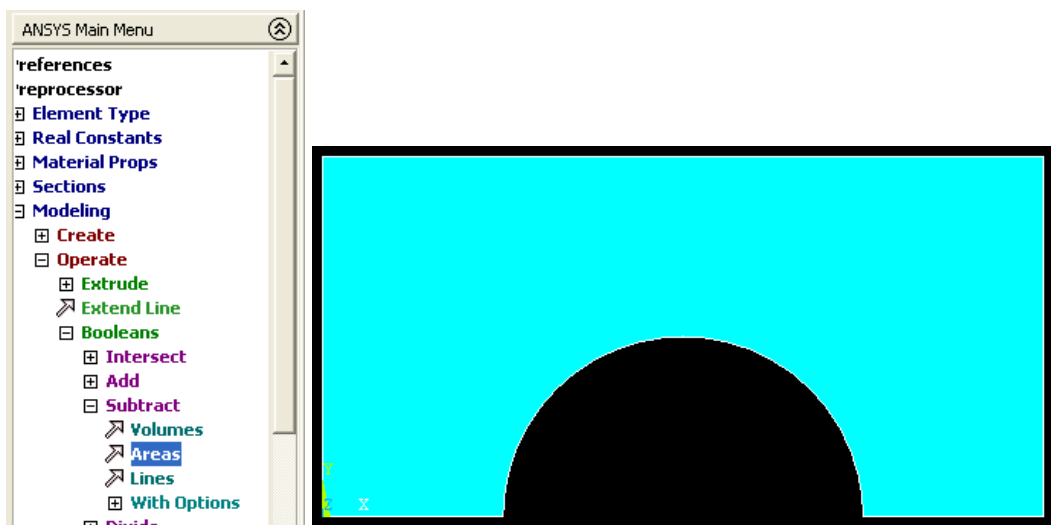


Obr. 11

Obr. 12

Obr. 13

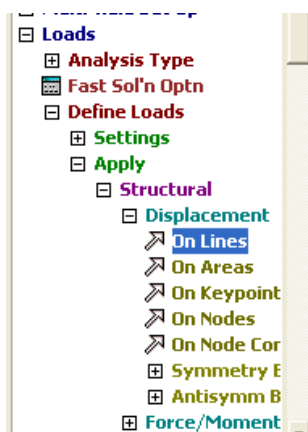
Pro spojení těchto částí použijeme v položce *Modeling* volbu *Operate* a *Subtract* pro odečtení jednotlivých částí.



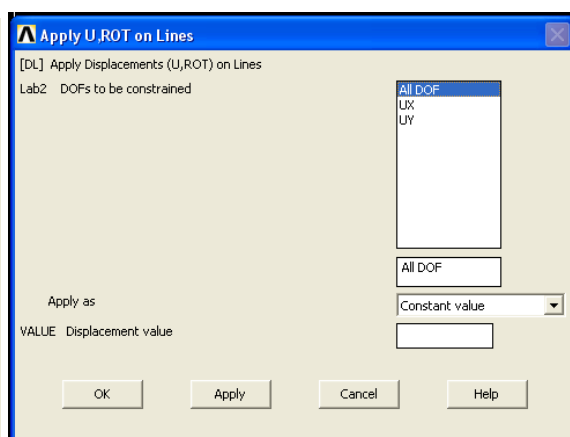
Obr. 14

Obr. 15 jednoduchá úloha

Pokud máme vytvořený model, musíme mu nastavit jeho vazbové podmínky. To provádíme v *Preprocesoru* položkou *Loads - Define loads - Apply - Structural - Displacement - On lines*, kde definujeme omezující vazbu na čáru. Po rozkliknutí je vidět, že máme možnost pouze UX UY, nebo oboje, což je jediné omezení možné pro náš prvek, protože naše prvky nepřenáší moment pomocí neznámých.

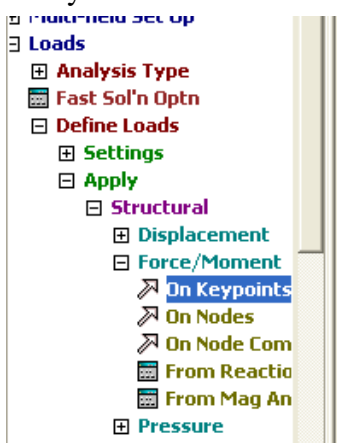


Obr. 16

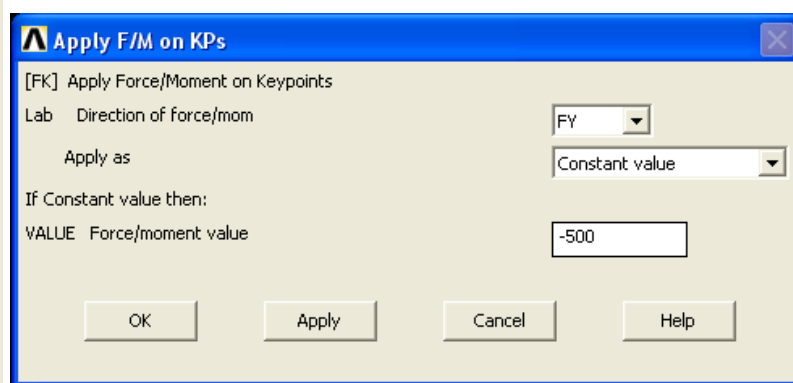


Obr. 17 nastavení vazbových podmínek

Po vazbách v položce *Loads* ještě nastavíme síly pod *Force/Moment* a položka *On keypoints* rozhodně musíme použít *Force* a ne *Pressure*, protože tam se jinak exportují hodnoty a můj algoritmus používá síly v nodech. Zase se nám zde objeví *FX* a *FY* a žádný moment.

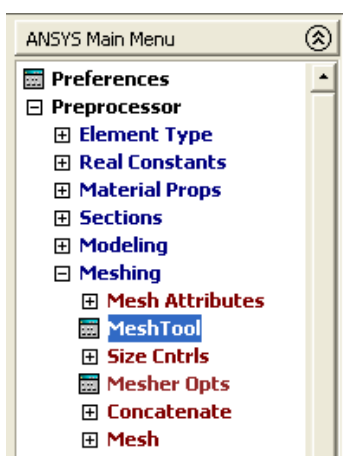


Obr. 18

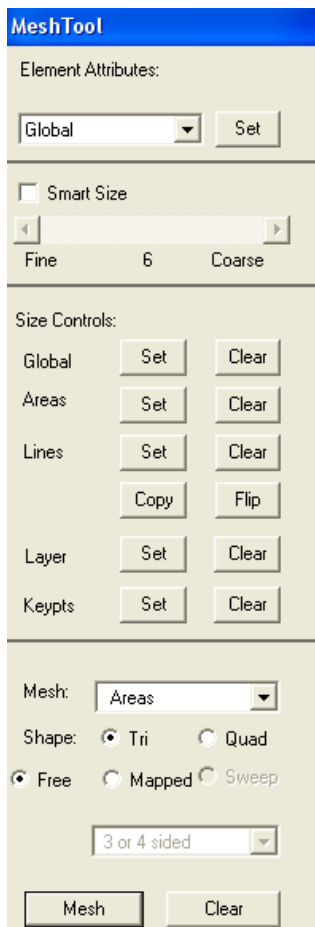


Obr. 19 nastavení sil

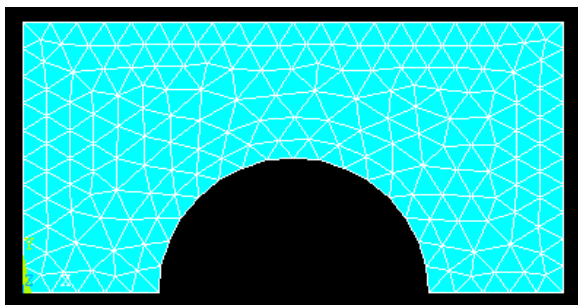
Po nastavení nám u modelu zbývá pouze rozdělit naši vytvořenou oblast na elementy a to uděláme v *Preprocesoru* v položce *Meshing* odkaz *MeshTool* tím se nám otevře dialog pro nastavování parametrů *Meshování*, tam musíme nastavit kousek nad tlačítkem *Mesh* u volby *Shape* položku *Tri*. Tím nastavíme trojúhelníkový typ prvku místo lepšího čtyřúhelníkového. Dále nastavíme u položky *Global* tlačítkem *Set* standardní velikost trojúhelníkového prvku, čímž volíme složitost soustavy. Na závěr stlačíme tlačítko *Mesh* a vygenerujeme síť pro naši úlohu.



Obr. 20



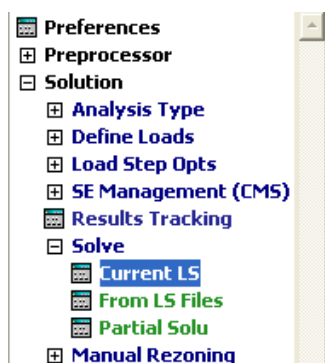
Obr. 21



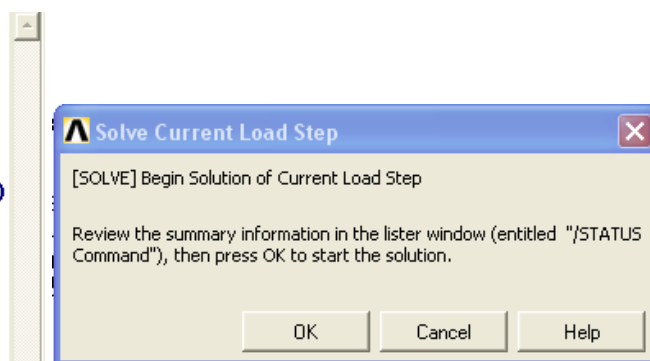
Obr. 22 úloha rozdělená na elementy (funkcí Mesh)

2.2 Výpočet deformace, export dat

Nakonec když je úloha kompletně sestavena ji necháme vyřešit pomocí položky v *Solution* pod *Solve* příkazem *Curent LS*, ten nám ještě zobrazí nabídkové okno, které se nás zeptá, jestli chceme soustavu opravdu vyřešit.

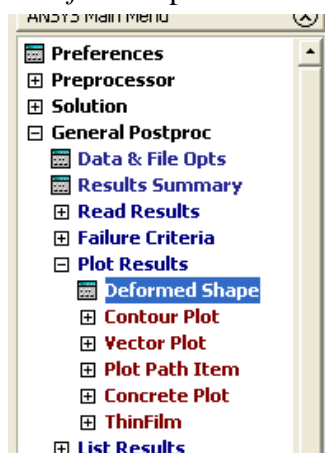


Obr. 23

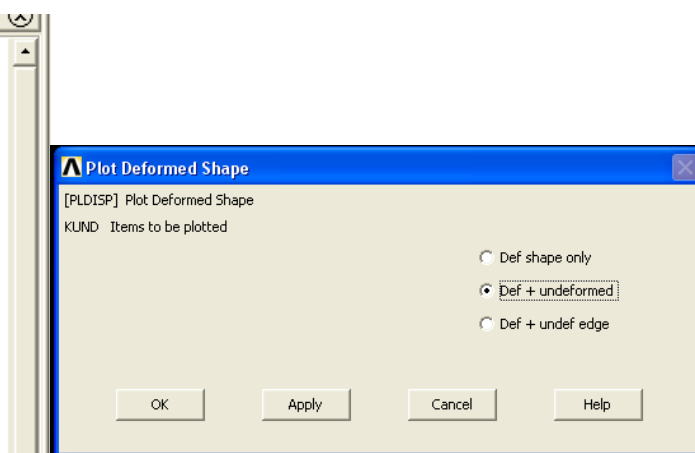


Obr. 24

Jakmile je zadaná úloha vyřešena, musíme si nechat zobrazit alespoň jedno řešení. V případě že tak neučiníme, ANSYS nám nedovolí vyexportovat hodnoty deformací. Necháme si tedy zobrazit třeba základní deformovaný tvar pomocí *General Postproc* volbou *Plot Result* položkou *Deformed Shape*, tam si třeba zobrazíme *Def + undeformed* pro zobrazení deformovaného a nedeformovaného tvaru.

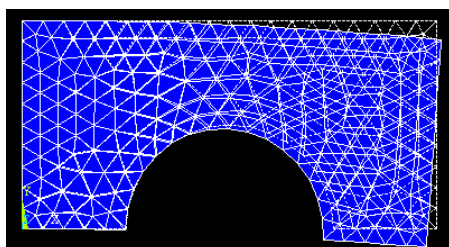


Obr. 25



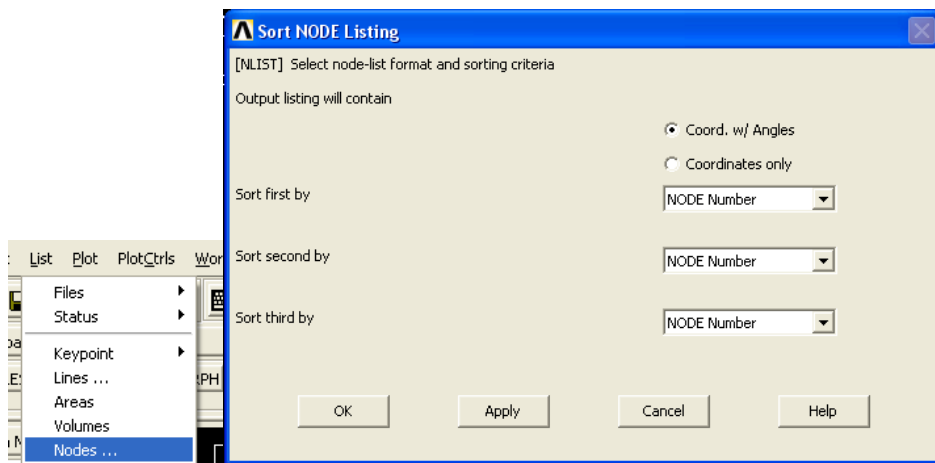
Obr. 26

Výsledné zobrazení není jen nutností pro export porovnávacích dat, ale slouží nám pro kontrolu, jestli byla soustava správně vyřešena.



Obr. 27 zobrazení deformace úlohy

A nyní, když máme vše připraveno, můžeme se pustit do vlastního exportování dat. To provádíme v horní liště programu, v položce *List*. Tam klikneme na *Nodes* a otevře se nám dialog pro nastavení možnosti zobrazení. Nic tam neměníme a otevře se nám okno, které vypisuje veškeré nody v úloze.



Obr. 28

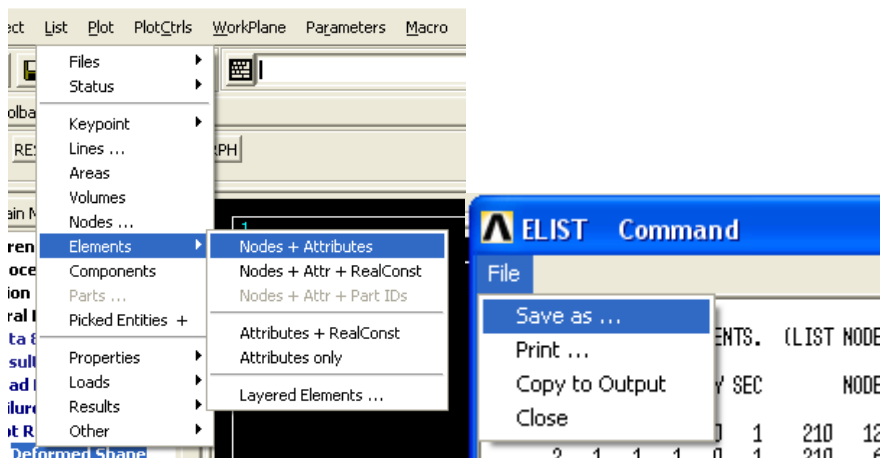
Obr. 29

Tam už nám stačí pouze kliknout na *File* a *Save* a hodnoty uložit.

NODE	X	Y	Z	THXY	THYZ	THZX
1	0.0000	0.0000	0.0000	0.00	0.00	0.00
2	25.000	0.0000	0.0000	0.00	0.00	0.00
3	5.0000	0.0000	0.0000	0.00	0.00	0.00
4	10.000	0.0000	0.0000	0.00	0.00	0.00
5	15.000	0.0000	0.0000	0.00	0.00	0.00
6	20.000	0.0000	0.0000	0.00	0.00	0.00
7	50.000	25.000	0.0000	0.00	0.00	0.00
8	45.123	24.520	0.0000	0.00	0.00	0.00
9	40.433	23.097	0.0000	0.00	0.00	0.00
10	36.111	20.787	0.0000	0.00	0.00	0.00
11	32.322	17.678	0.0000	0.00	0.00	0.00

Obr. 30 zobrazení souřadnic nodů

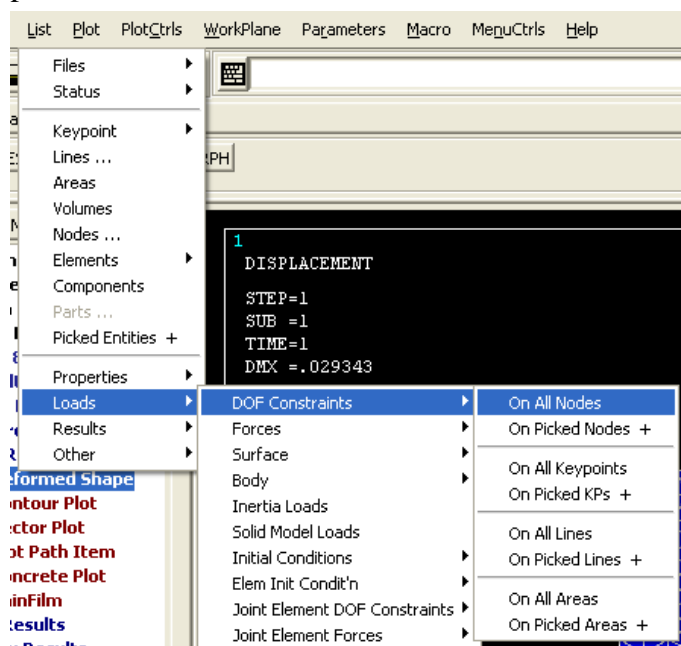
Jako další ukládáme elementy. Opět ve volbě *List*, pod položkou *Elements*, tam pak klikneme na volbu *Nodes + Attributes*, pak se nám otevře okno s vypsányými prvky a opět dáme, *File* a *Save as* pro uložení. Pro jistotu ještě podotknu, že musíme postup pro ukládání dodržet, protože v jiném formátu to můj program neumí načíst.



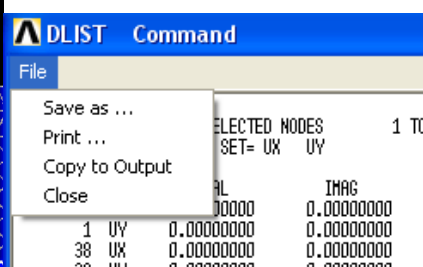
Obr. 31

Obr. 32

Dále opět v položce *List*, pod položkou *Loads*, odklepeme *DOF Constraints* a volbu *On All Nodes*. Tím se nám otevře okno s vazbovými podmínkami a ty opět uložíme pomocí *File Save as*.

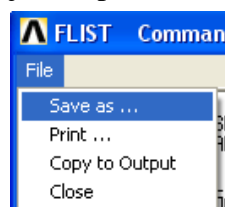


Obr. 33



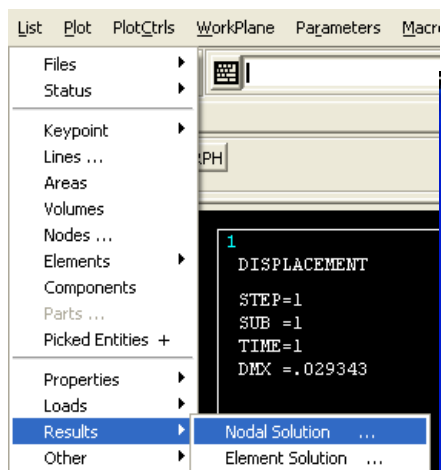
Obr. 34

V položce *Loads* ještě klikneme na *Forces* a uložíme síly pomocí dialogového okna jako u předchozích úloh.

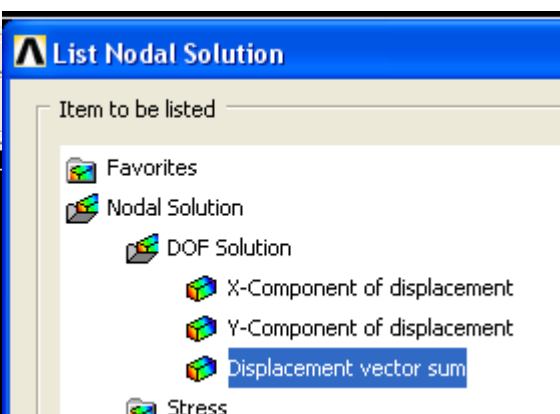


Obr. 35

A nyní export dat, který se dělá pouze jako kontrola správnosti řešení vlastního programu. Tj. už vypočtených hodnot deformace. Opět otevřeme *List* a položku *Result* a *Nodal Solution*. Tam se nám otevře výběr úloh pro řešení, jako jsou různá napětí, deformace a jiné. My si vybereme *DOF Solution* a *Displacement vector sum*.

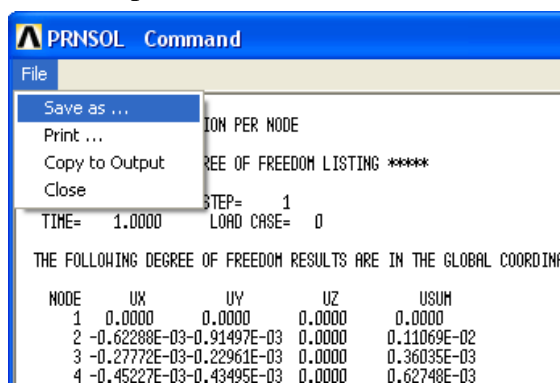


Obr. 36



Obr. 37

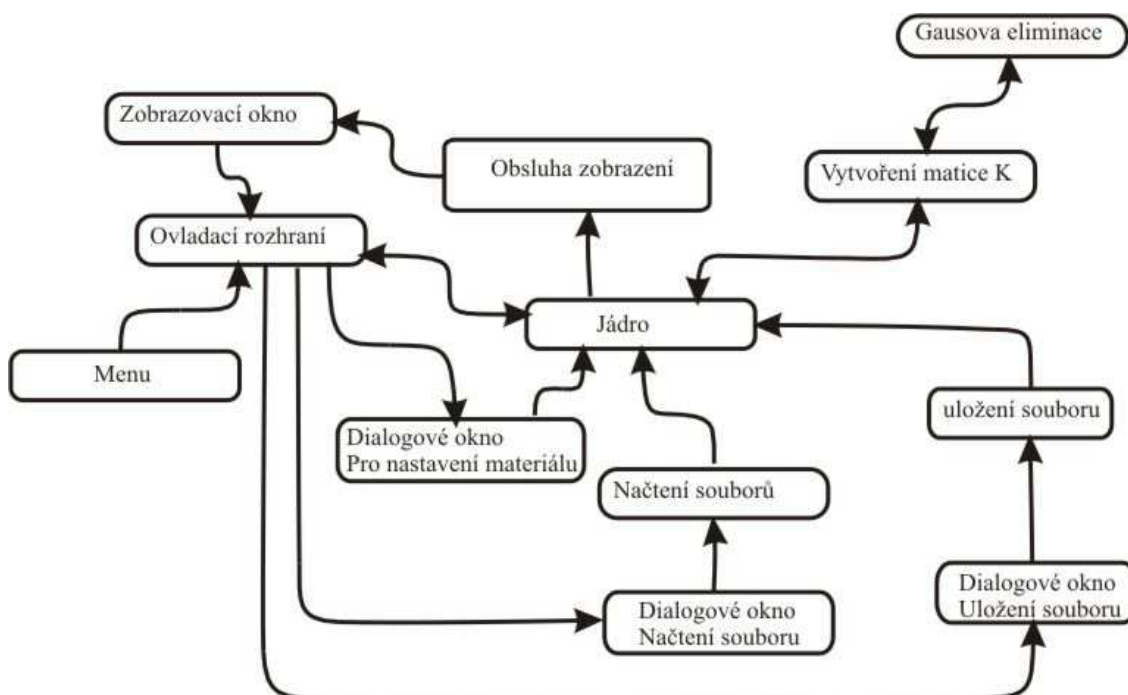
Nyní se nám zobrazí hodnoty deformací v ose x, v ose y a velikost vektoru deformace, tyto hodnoty si můžeme pro porovnání opět uložit a později načíst třeba v Excelu a zobrazit v grafu a porovnat s daty z mého programu. Toto porovnání je zobrazeno v další kapitole



Obr. 38

3 PROGRAM PLANE42

3.1 Vývojový diagram programu



Obr. 38 blokové struktura programu

3.2 Rozbor vývojového diagramu

Ve své podstatě se tento program většinou skládá ze samostatných bločků, které spolu příliš nespolutpracují, kromě sdílení paměti.

Takže ze začátku program vyčkává, než z ovládacího rozhraní dostane nějaký příkaz. Pomocí ovládacího rozhraní vyšleme příkaz pro otevření dialogového okna pro nahrávání souborů. Jakmile se nám otevře naše dialogové okno, tak můžeme nastavit cestu k souboru, který chceme načíst. Jakmile cestu nastavíme, klepneme na tlačítko načtení a cesta k souboru se předá proceduře, která otevře soubor a pokusí se z něj načíst požadovaná data. Načtená data se pak předají do jádra a tam zůstanou.

Jakmile takhle načteme všechna data, tak musíme pomocí ovládacího rozhraní dát příkaz k otevření dialogového okna pro nastavení materiálu. Toto okno po našem nastavení opět předá své parametry do jádra a tam se uloží.

Poté opět pomocí ovládacího rozhraní dáme příkaz pro výpočet vlastní soustavy. To se provede tak, že jádro předá ukazatel na nody, elementy, vazby a síly, které jsou uloženy v paměti jako čtyři různá pole. Algoritmus poté určí počet neznámých a alokuje si globální matici K, která zabírá spoustu paměti. Jakmile ji alokuje tak začne sestavovat jednotlivé lokální matice, které pak vkládá do matice globální. Samozřejmě se kontroluje, zdali nedojde k nějakému špatnému načtení, které by způsobilo pád programu. Dojde li ke špatnému načtení, bude pouze zobrazena varovná hláška a

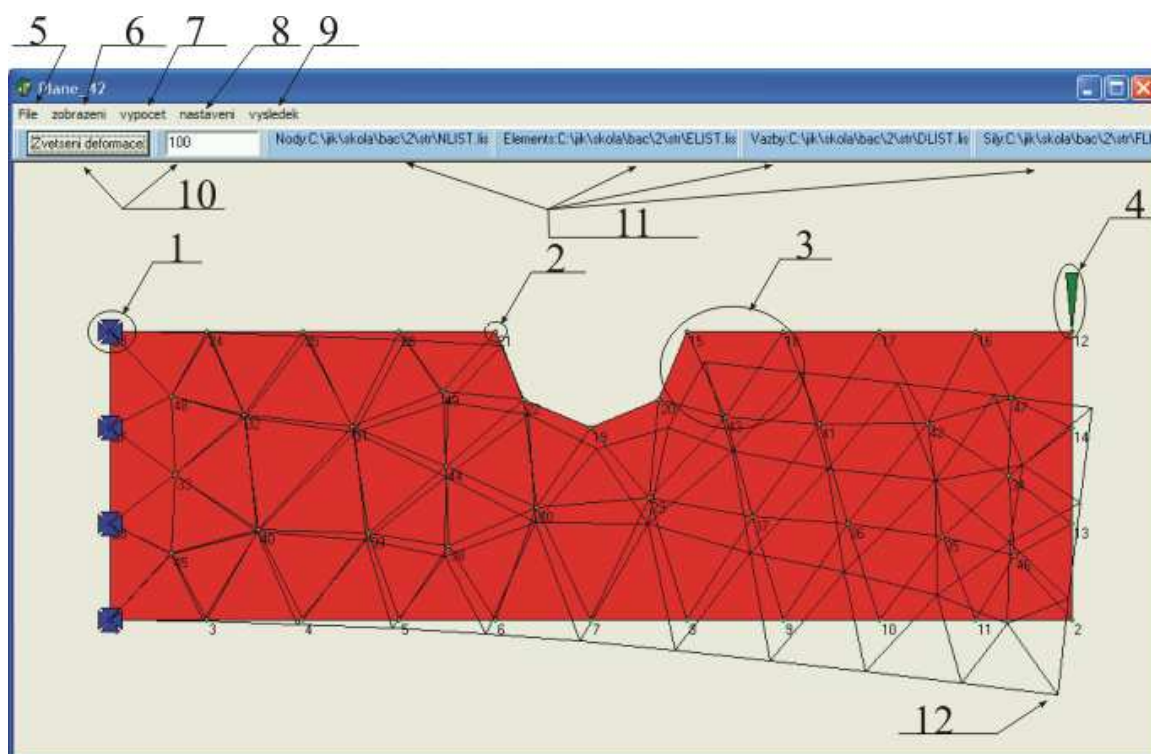
výpočet je ukončen. Chybová hláška je zobrazena i při matici s nulovým determinantem.

Jakmile je pomocí Gaussovy eliminace vyřešena naše soustava rovnic, je globální matice tuhosti uvolněna a hodnoty deformace jsou vsazeny do pole nodů. Ke každému nodu bude přiřazena jeho deformace. Zároveň jsou číselné hodnoty převedeny na text a vloženy do paměti listboxu.

Opět pomocí ovládacího rozhraní můžeme otevřít dialog pro zobrazení číselných hodnot deformací. V tomto dialogu můžeme tyto hodnoty uložit do textového souboru, ve kterém jsou hodnoty odděleny tabulátorem, aby je bylo možno snadno načíst v programu Excel. Ten umí textové hodnoty načíst jako čísla a rozdělit je do sloupců pomocí tabulátorů jako oddělovačů.

Posledním nepopsaným blokem je zobrazení. Zobrazování je spouštěno nejen žádostí od jádra, ale i voláním operačního systému o překreslení okna a samozřejmě při načtení hodnot, které se zobrazují. Zobrazovací okno zobrazuje pomocí základního vykreslování všechny potřebné části. Zobrazení nodů je vyobrazeno jako malé zelené puntíky, které mají vpravo dole své pořadové číslo. Elementy jsou vyobrazeny jako červené trojúhelníky ohraničené černou linkou. Síly jsou zelené trojúhelníky představující šipku ukazující směr působení. Vazby jsou zobrazeny modře vždy dvěma trojúhelníky ve směru omezení. Deformace je pouze zobrazena jako nevyplněné trojúhelníky, které jsou tvořeny třemi černými čarami. Různé formy zobrazení lze nastavit opět v ovládacím rozhraní a zvětšení a posunutí modelu lze přizpůsobit pro zobrazované okno.

4 POPIS UŽIVATELSKÉHO ROZHRAŇÍ PLANE42



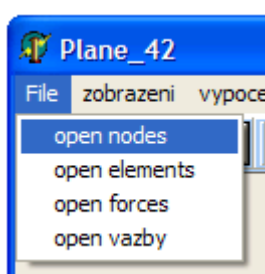
Obr. 39 okno programu s označením jednotlivých částí

Legenda:

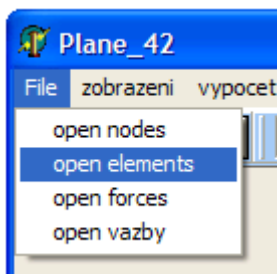
- 1 Vazby jsou zobrazeny modrými šipečkami ve směru omezení.
- 2 Nody jsou zobrazeny zelenými body a pořadovým číslem umístěným vlevo dole.
- 3 Prvky (elementy) jsou zobrazeny červenými trojúhelníky.
- 4 Síly jsou zobrazeny zelenými šipkami v osách x, y ve směru působení.
- 5 Položka File, v menu rozvine roletku s načítáním příslušných souborů.
- 6 Položka zobrazení v menu rozvine roletku na nastavení zobrazení.
- 7 Položka výpočet v menu po stisknutí spustí výpočet deformace.
- 8 Položka nastavení v menu otevře dialog pro nastavení materiálu.
- 9 Položka výsledek v menu otevře okno s hodnotami posunů v jednotlivých nodech.
- 10 Zvětšení deformace nastavuje zvětšení zobrazení posunu deformace.
- 11 Na liště pod menu jsou vypsány načtené soubory.
- 12 Deformace je zobrazeny tenkými černými linkami.

4.1 Načtení hodnot

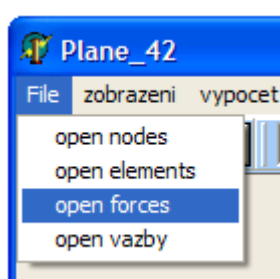
Data potřebná k výpočtu modelu deformace sestávají z nodů, elementů sil a vazebních podmínek. Všechny tyto kroky provedeme v submenu *File* horní lišty. Viz. příložené obrázky 41 – 44. Po vybrání příslušné položky se otevře klasický dialog *Open File* viz obrázek 45.



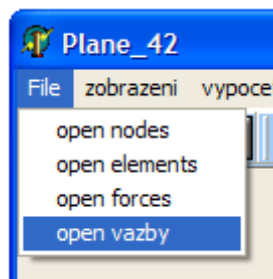
Obr. 41



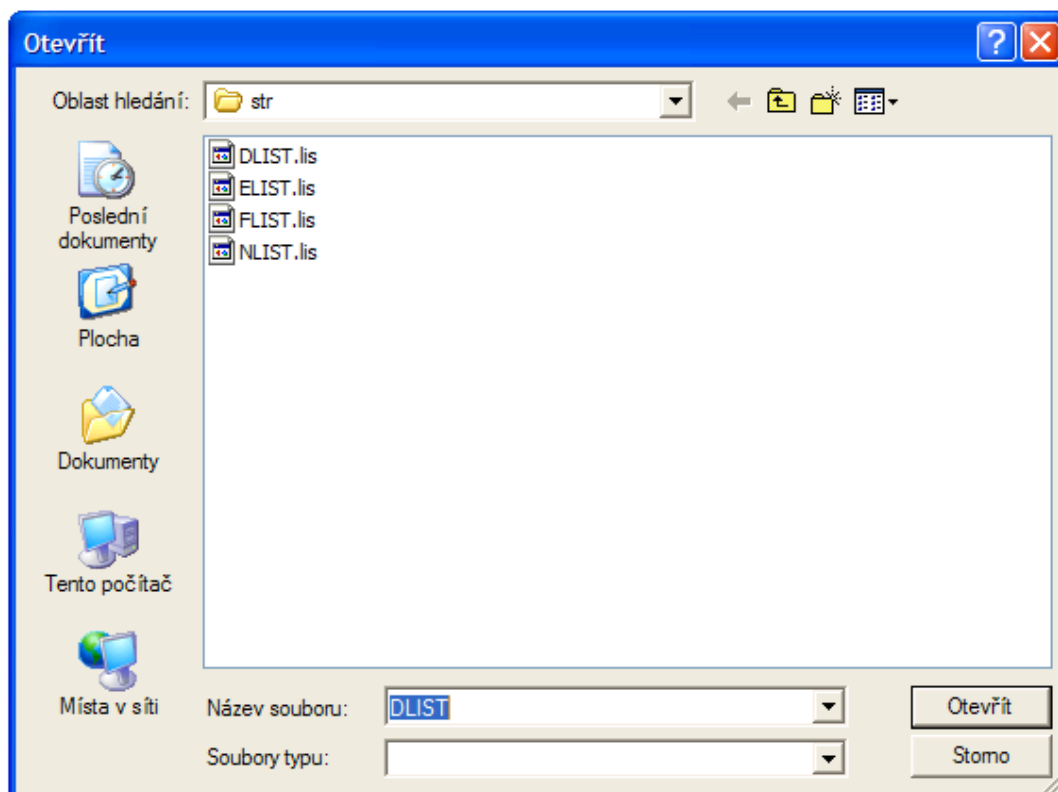
Obr. 42



Obr. 43



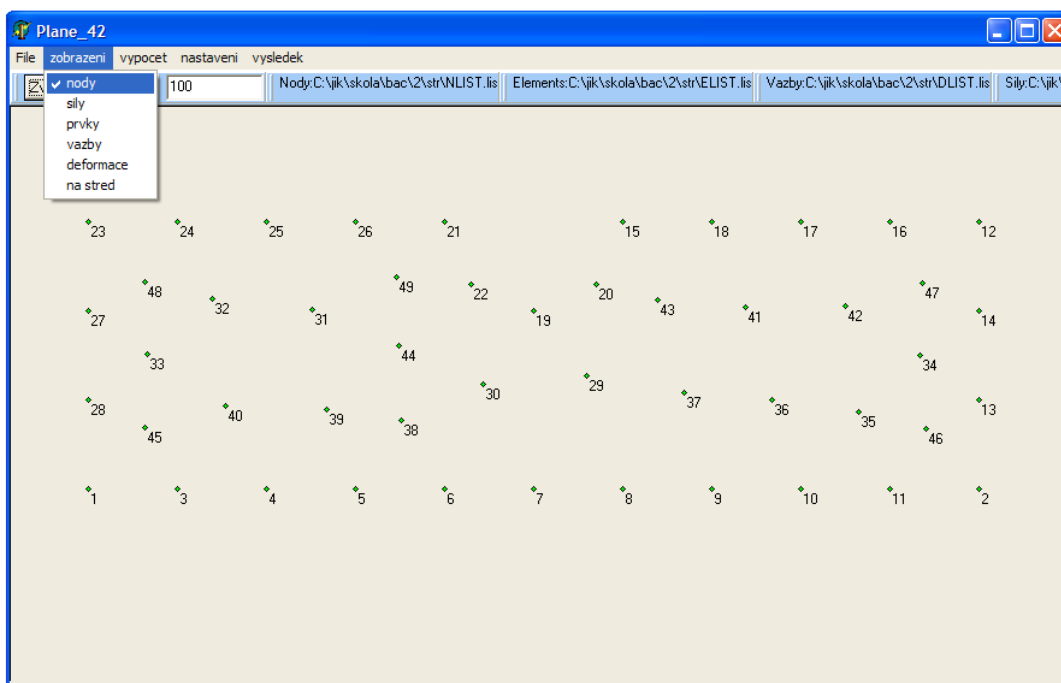
Obr. 44



Obr. 45 dialogové okno pro načtení souborů

4.2 Styly zobrazení

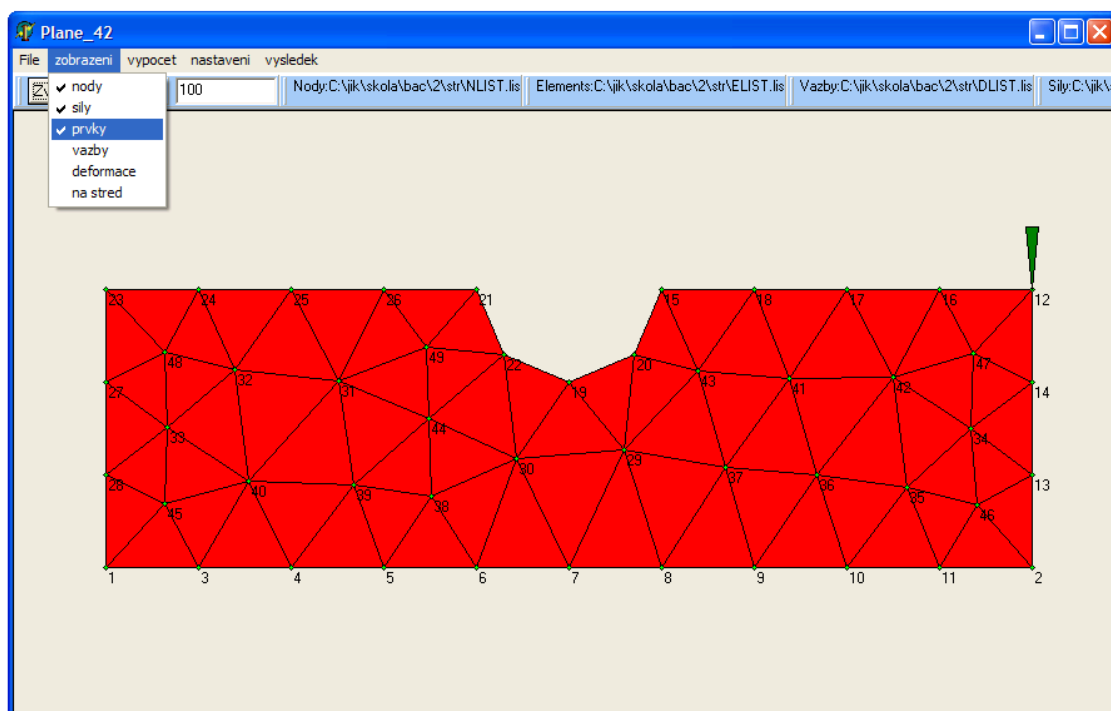
Submenu *zobrazení* umožňuje nastavit, které komponenty modelu se nám zobrazují. Nastavení probíhá pomocí checkboxů, takže můžeme zobrazovat libovolnou kombinaci komponent. Poslední položka *na střed* centruje model na střed pracovní polohy a upraví jeho velikost na nejvhodnější vzhledem k parametrům pracovní plochy.



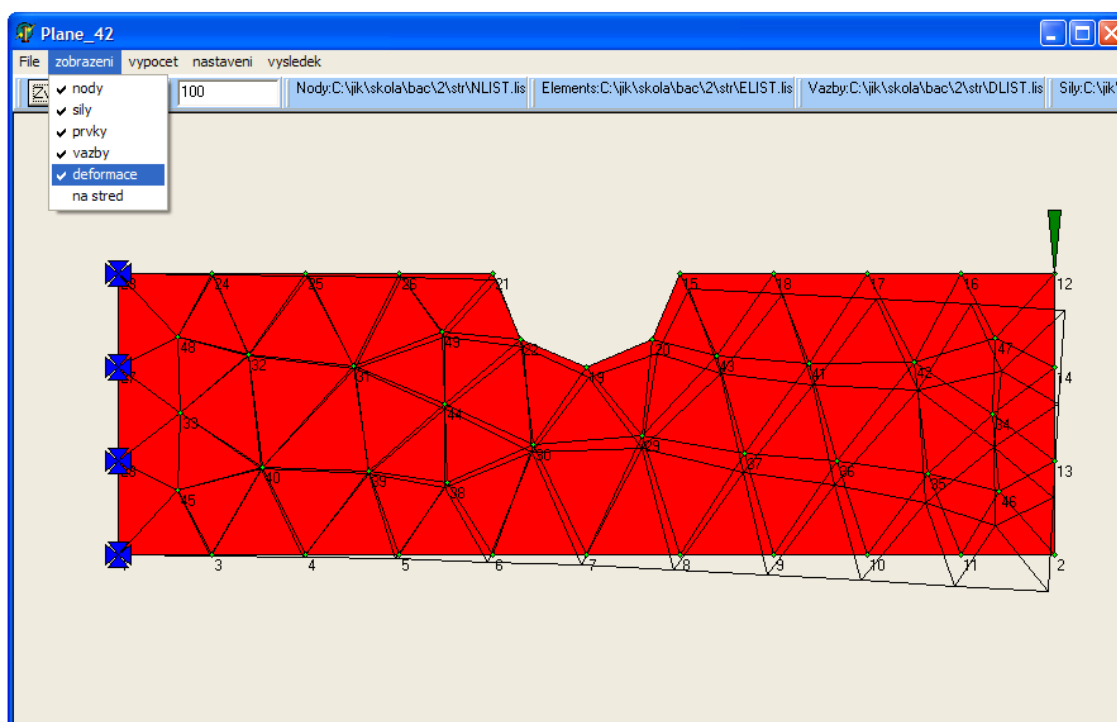
Obr. 46 (zobrazení nodů)



Obr. 47 (zobrazení nodů a sil)



Obr. 48 (zobrazení nodů, sil a prvků)



Obr. 49 (zobrazení nodů, sil, prvků a vazeb)

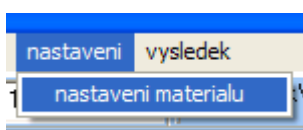
4.3 Nastavení modelu a výpočet deformace

Vlastnosti materiálu nastavíme pomocí položky, *nastavení materiálu* v submenu *nastavení* (obr. 50). Jsou to vlastnosti lineárního izotropního hookovského materiálu. (obr. 51) také zde nastavujeme, jestliže se jedná o výpočet rovinné napjatosti, nebo rovinné deformace.

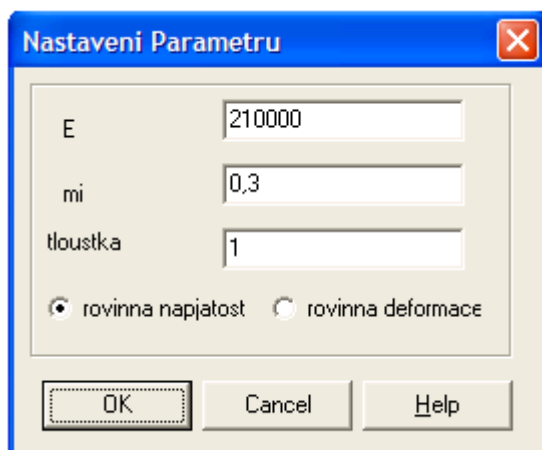
E – Youngův modul pružnosti

μ – Poissonův poměr

tloušťka – tloušťka materiálu

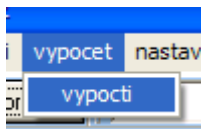


Obr. 50 volba nastavení materiálu



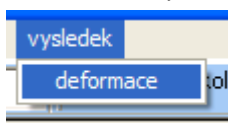
Obr. 51 dialog pro nastavení materiálu

Vypočtení vlastní deformace, provedeme pomocí položky, *vypočti* v submenu *vypočet*. Tím spustíme výpočet, který může podle rozsahu řešené úlohy trvat až několik minut.



Obr. 52 spuštění výpočtu deformace

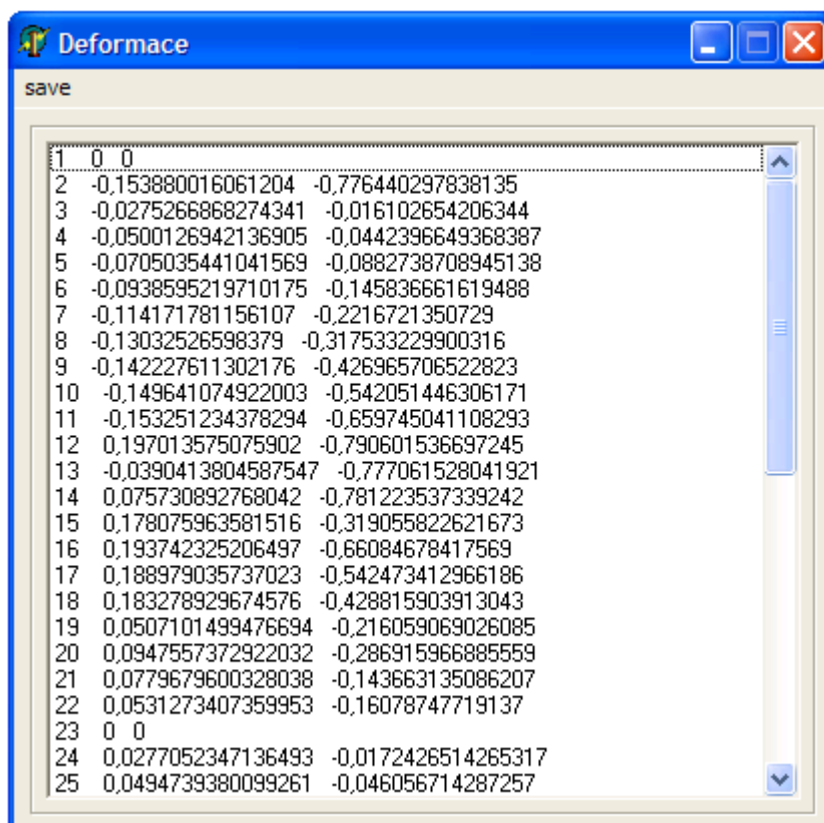
Číselné zobrazení vlastních vypočtených hodnot provedeme pomocí položky *deformace* v submenu *výsledek*.



Obr. 53 zobrazení okna s hodnotami deformace

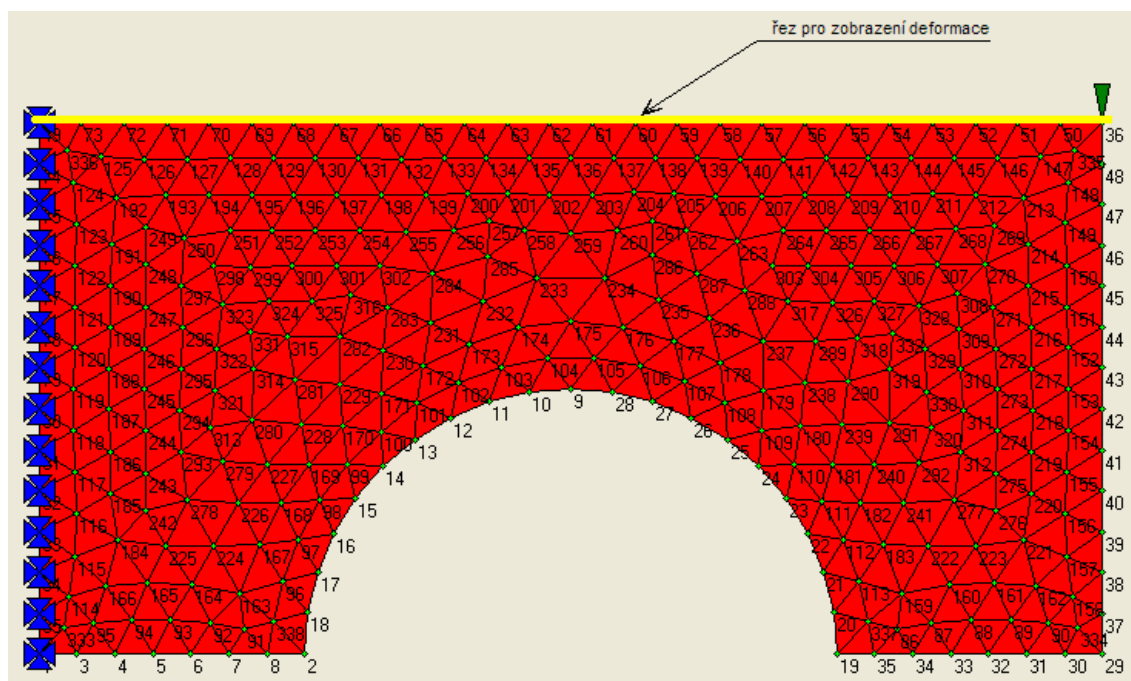
V následně zobrazeném dialogu jsou zobrazeny informace o deformaci. První sloupec zobrazuje pořadové číslo nodu, druhý sloupec udává deformaci v x-ové ose, poslední sloupec deformaci v ose y (Obr 54).

Pomocí tlačítka *save* otevřeme dialog pro uložení deformace do souboru v textovém formátu.



Obr. 54 okno zobrazující hodnoty deformací v nodech

5. SROVNÁNÍ VÝSLEDKŮ S PROGRAMEM ANSYS



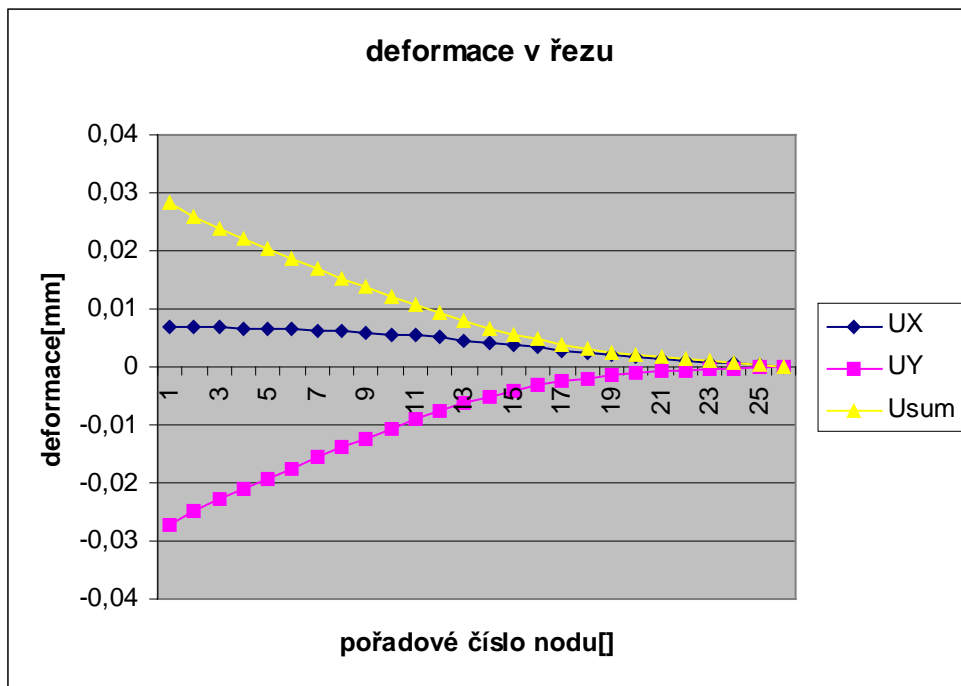
Obr. 55 vyznačení řezu, ve kterém provádím porovnání hodnot

Ve všech dále uvedených srovnávacích grafech zobrazuji deformaci v nodech, které leží ve výše uvedeném žlutém řezu.

5.1 Srovnání modelu pro rovinnou napjatost

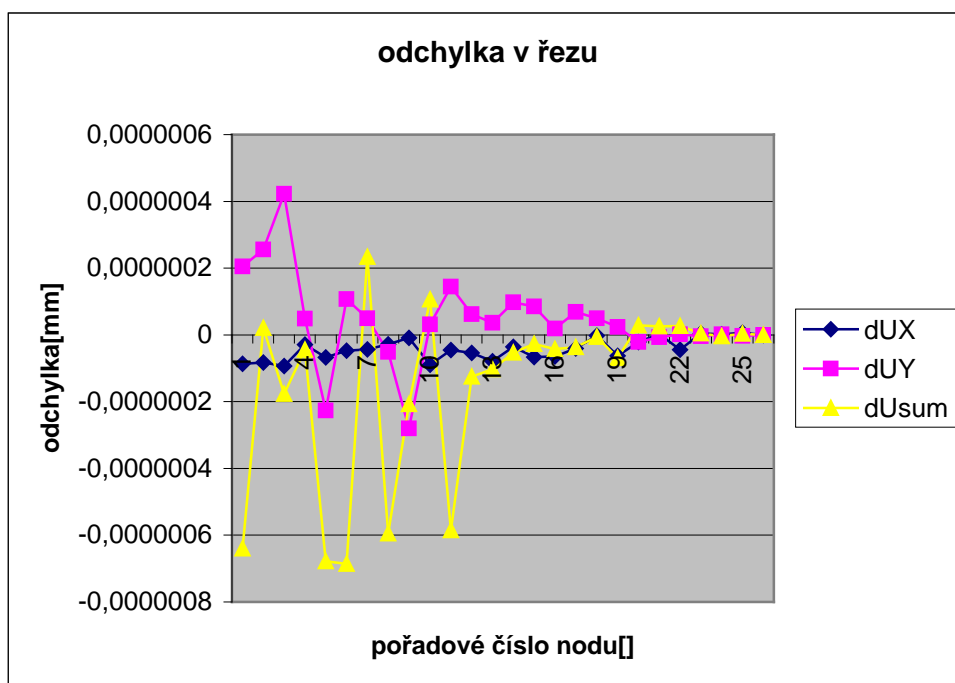
V tomto grafu zobrazuji deformaci v mm pro nody v řezu zprava doleva jako na obrázku. Uvádím zde pouze jeden graf pro závislost deformace, protože se vypočtené hodnoty z ANSYSU s těmi mými shodují, takže by se překrývaly.

Navíc je z tohoto grafu celkem vidět, že deformace je spojitá a proto můžeme tato data použít pro srovnávání.



Obr. 56 graf závislosti deformace vyznačených nodů

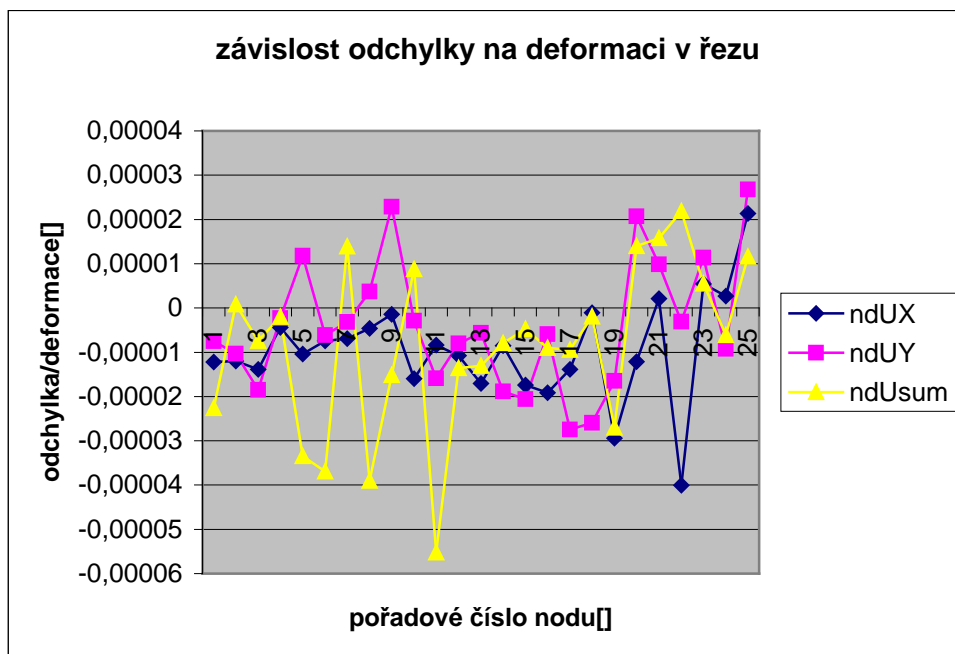
Proto v dalším grafu už zobrazují rozdíl hodnot vypočtené deformace mým programem a programem ANSYS, tento rozdíl udává základní představu o typu odchylky jednotlivých hodnot a její závislost.



Obr. 57 graf závislosti odchylky deformace vyznačených nodů

Z tohoto grafu je jasně vidět, že je velikost maximálních odchylek hodnot mezi mým programem a ANSYSEM závislá na velikosti deformace.

Proto další graf vypočteme jako odchylku mého programu od ANSYSU a dělím jí absolutní hodnotou deformace. Tj. závislost rozdílu vypočtené deformace na velikosti deformace.

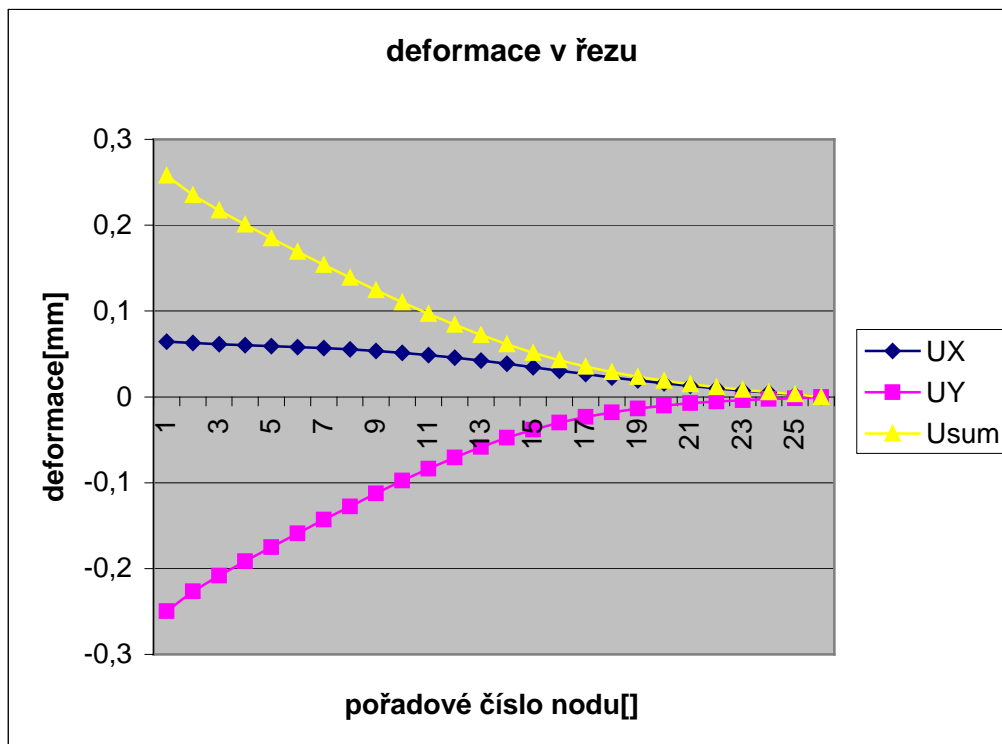


Obr. 58 graf závislosti odchylky v poměru k deformaci vyznačených nodů

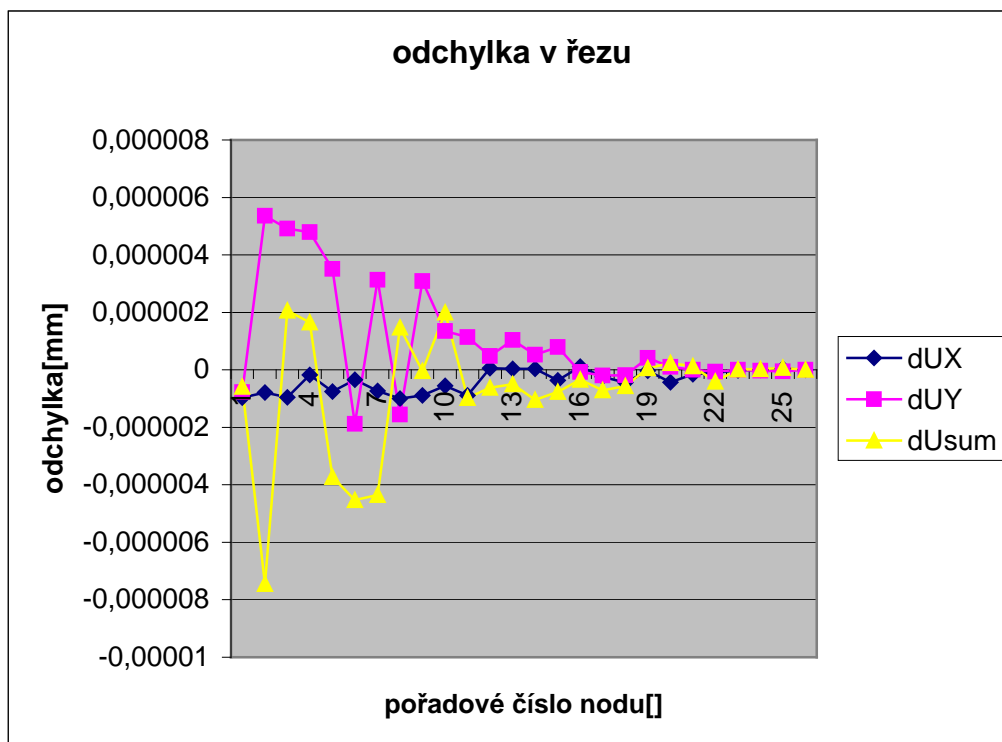
Z tohoto grafu už je jasně vidět, že závislost odchylky řešení je pouze na poslední číslici vyexportovaných dat z ANSYSU a to je jen jedna dvěstěpadesátina procenta, což se dá považovat za úspěch. Vezmeme-li v úvahu rozdíl mezi 3úhelníkovými prvky a vyššími typy prvků,

5.2 Srovnání modelu pro rovinnou deformaci

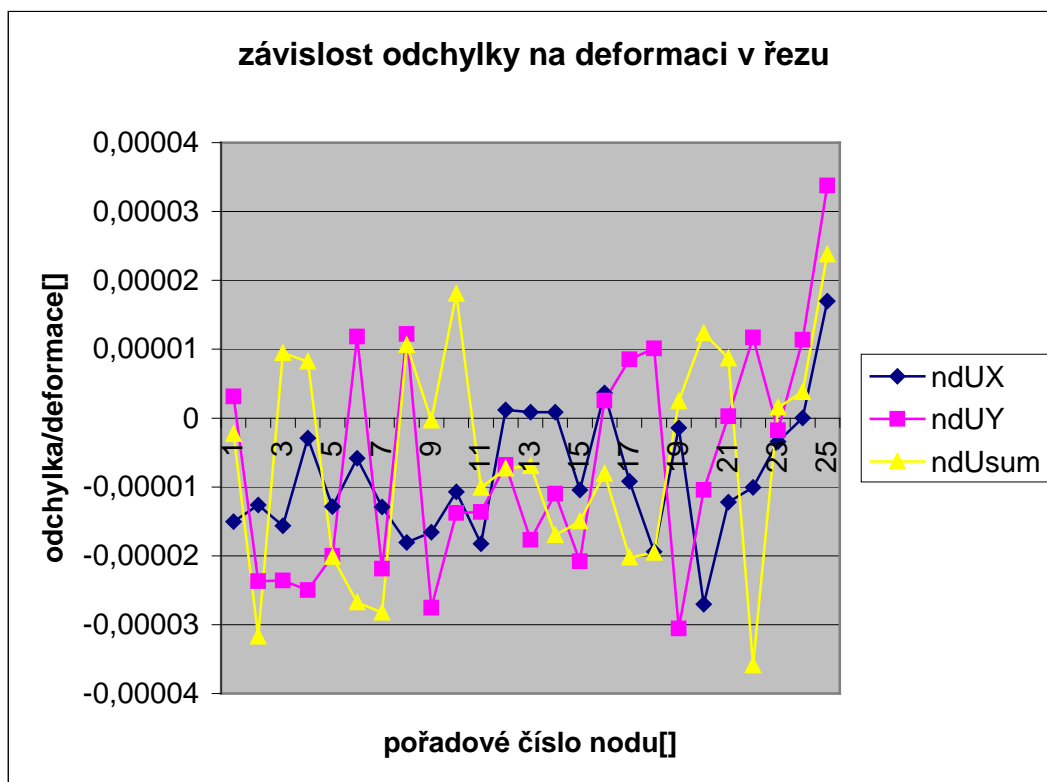
Nyní provedu stejné srovnání pro řešení rovinné deformace namísto rovinné napjatosti. Pro stejnou úlohu, protože je to stejný postup zobrazím pouze grafy bez popisu



Obr. 59 graf závislosti deformace vyznačených nodů



Obr. 60 graf závislosti odchylky deformace vyznačených nodů



Obr. 61 graf závislosti odchylky v poměru k deformaci vyznačených nodů

6. Závěr

Na závěr lze říci, že programy MKP mají velmi široké použití. Nejen díky přesným výsledkům dosahovaných pomocí této metody, ale i díky možnému modulovému přístupu při návrhu MKP programů. Jednotlivé specializované moduly pracují pomocí společného základu. Proto lze malou změnou řešit naprosto odlišné úlohy.

Touto prací, aneb naprogramováním základního a přitom přesného a velmi dobře pracujícího algoritmu MKP bylo dokázáno, že není až takový problém vytvořit nějaký MKP algoritmus nebo jednodušeji a efektivněji pouze přidat MKP prvek do nějakého komerčního či volně šiřitelného MKP programu umožňující řešení standardních i velmi nestandardních úloh (např. předpověď počasí).

Můžeme tedy celkem jednoduše řešit i úlohy n-rozměrných polí, pro které žádný software neexistuje nebo existuje, ale je neoptimalizovaný nebo velmi drahý a analytické řešení je neúměrně složité.

Seznam použité literatury

- [1] Libor Čermák. *Algoritmy metody konečných prvků*. Brno, září 2005
- [2] J.Petruška. *Počítačové metody mechaniky II*. Brno, Listopad 2003
- [3] Ondráček, E.; Vrbka, J.; Janíček, P.: *Mechanika těles – pružnost a pevnost II*, skriptum FS VUT Brno, 2002
- [4] Alexander Hrennikoff. *Solution of problems of elasticity by the frame-work method*, ASME J. Appl. Mech. 8 (1941), A619–A715
- [5] Strange, G.; Fix, G. J. *An Analysis of the Finite Element Method* 1973