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 METODY KONEČNÝCH PRVKŮ NA ŘEŠENÍ ROVINNÝCH ÚLOH

APPLICATION OF FINITE ELEMENT METHOD TO SOLVE PLANE PROBLEMS

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE — DAVID LANGER
AUTHOR

VEDOUCÍ PRÁCE — ING. TOMÁŠ NÁVRAT, PH.D.
SUPERVISOR

BRNO 2015

Abstrakt

Práce je zaměřena na problematiku aplikace metody konečných prvků na rovinné úlohy mechanického namáhání strojních součástí.

První část práce má za cíl umožnit čtenáři udělat si přehled o používaných programech pracujících s metodou konečných prvků a jejich možností, a to jak volně dostupných tak komerčních.

Druhá část se věnuje možnosti použití volně dostupných prostředků pro provádění výpočtů MKP.

Poslední část práce tvoří dokumentace programu výpočtu který byl k práci vytvořen. V dokumentaci je detailněji přiblížen postup zvoleného typu výpočtu.

Klíčová slova

Metoda konečných prvků, Výpočetní systémy, Volně dostupné prostředky pro inženýrské výpočty

Abstract

The subject of this thesis is application of finite element method to solve planar problems of mechanical stress of machine components.

The goal of the first part of the thesis is to provide the reader with an overview of commonly used FEM software and it's capabilities, both freeware and commercial solutions.

Second part of the thesis deals with the usage of freeware for FEM calculations.

The last part of the thesis consists of documentation to the FEM software created as a subject of this thesis. The documentation provides a detailed view of the selected type of calculation.

Keywords

Finite element method, Engineering software, Freeware usable for engineering calculations

Bibliografická citace

LANGER, D. *Aplikace metody konečných prvků na řešení rovinných úloh.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2015. 22 s. Vedoucí bakalářské práce Ing. Tomáš Návrat, Ph.D..

Prohlášení o původnosti práce

Prohlašuji, že jsem bakalářskou práci Aplikace metody konečných prvků na řešení rovinných úloh vypracoval samostatně za pomoci vedoucího Ing. Tomáše Návrata Ph. D. a uvedl v seznamu literatury všechny použité zdroje.

V Brně, dne Podpis

Obsah

Úvod

Metoda konečných prvků je numerická metoda sloužící k simulování velkého rozsahu výpočetních úloh, od mechanického namáhání přes vedení tepla až po proudění tekutin. Pro účely této práce se budeme zabývat statickými problémy deformace plošných těles.

Její základ leží ve variačních principech, tedy odvětví zabývající se maximalizací a minimalizací funkcionálů. Konkrétně pro mechanické výpočty se jedná o Lagrangeův variační princip. Ten říká, že ze všech kinematicky přípustných stavů pružného tělesa nastává takový, jaký uděluje celkové potenciální energii Π stacionární hodnotu.

Kinematicky přípustným stavem se rozumí stav, kde posuvy v celé řešené oblasti jsou spojité, mají po částech spojitou derivaci, splňují geometrické okrajové podmínky a přípustná přetvoření jsou s nimi spojeny geometrickými rovnicemi.

V minulosti se jako jedna z metod výpočtu používala tzv. Ritzova metoda, která má s MKP úzce související algoritmus.

Clekovou potenciální energii je pro numerické řešení nutno vztáhnout na konečný počet parametrů. K této aproximaci se používá součet předem známých tzv. bázových funkcí. Neznámými koeficienty těchto funkcí v deformační variantě výpočtu jsou složky posuvů.

Ritzova metoda definuje bázové funkce na celé řešené oblasti. Pro složitější geometrické obrazce je konstrukce bázových funkcí velmi složitá či zcela nemožná.

Metoda konečných prvků, jak název napovídá, aproximuje řešenou oblast sítí geometricky jednoduchých podoblastí, tzv. prvků. Každý prvek má efinované vlastní bázové funkce, které mají tvar polynomu. Tento postup výrazně zjednodušuje řešení. Je však nutno dbát na to, aby zvolená síť dostatečně vykrývala aproximovanou oblast a její hranici.

Po minimalizaci takto sestaveného funkcionálu Π za pomoci systému rovnic obecné pružnosti a pevnosti dostaneme soustavu lineárních rovnic, jejíž jedinými neznámými jsou právě hledané složky posuvů. Po jejich získání je možné pomocí geometrických a konstitutivních vztahů dopočítat potřebné hodnoty přetvoření a napětí.

Přehled používaných MKP programů

ANSYS

V současnosti jeden z nejpoužívanějších komerčních výpočetních programů na trhu. ANSYS pokrývá výpočty v řadě odvětví, včetně například mechanických výpočtů a výpočtů v elektronice a elektrotechnice. Obsahuje řadu nadstaveb které zesnadňují práci, jako například grafické rozhraní Workbench, podporu cloudových výpočtů či data management.

Abaqus

Jeden z nejstarších programů na výpočty MKP, programovaný ve volně dostupném jazyce Python. Jeho výhodou je, že ve svém balíku aplikací nabízí také CAE software. Program původně vznikl pro řešení nelineárních problémů, tedy například pro výpočet materiálů s pružně plastickým modelem.

ADINA

ADINA je další z široké škály výpočetních programů. V průmyslu a akademickém prostředí se využívá na výpočty mechanického zatížení, tepelných výpočtů a úloh elektromagnetismu.

Autodesk Inventor

Inventor je program primárně určen pro vytváření technické dokumentace, avšak má v sobě vestavěnou podporu MKP výpočtů. Jeho výhodou oproti ostatním systémům z vzdělávacího hlediska je nabízení studentských licencí.

GetFEM++

Jeden z volně dostupných programů pro výpočty MKP. Jeho předností kromě ceny je také možnost propojit jej s jinými programy, a to Pythonem, MATLABem and SciLabem jako post-processory či využívání výpočtového modelu o neurčených dimenzích, jež mu umožňuje výpočet interakcí např. mezi 1D a 3D problémem.

Agros2D

Jednodušší volně dostupný výpočtář pro 2D problémy z mnoha oblastí. Podporuje skriptování v jazyce Python.

Volně využitelné prostředky jsou dnes již nedílnou součástí práce s počítačem. Jejich dostupnost ocení nejen studující a vědci, ale i menší programátorské firmy či pouze lidé se zájmem o věc. Dnešní freeware a open source produkty jsou zároveň svými možnostmi srovnatelné s koverčně dostupnými řešeními a jsou základem i mnoha obchodně úspěšných produktů.

Fortran

Překladače Fortranu se používají na nejrychlejších počítačích světa. Jedná se o jeden z nejstarších jazyků pro numerické výpočty a používal se například na výpočty předpovědi počasí, letových drah kosmických raket apod.

Jeho nevýhodou je nepřehledná škála různých verzí, které sice jsou většinou zpětně kompatibilní, ale není to pravidlem. Výhody novějších verzí jsou například podpora objektově orientovaného programoání či souběžného programování.

Object pascal

Object pascal je, jak již název napovídá, objektově orientovanou nástavbou jazyka pascal. Jedním z dostupných překladačů je například program Delphi. Pascal je již velmi starý programovací jazyk což se projevuje na komfortu práce s ním. Je nutno například deklarovat proměnné, což je věc která u novějších jazyků odpadá a usnadňuje tak práci a lépe využívá operační paměť.

PHP

PHP je určením jazyk na vývoj webových aplikací. Jeho pomocí je možné snadno naprogramovat cloudové aplikace, kde více uživatelů může pro svou práci využít výpočetní potenciál serveru. Nevýhodou je nutnost instalace dalších knihoven jelikož jazyk není primárně určen k výpočtům a nemožnost práce na lokálním zařízení bez instalace doplňkových programů.

Python

Python je relativně nový programovací jazyk který vznikl teprve v roce 1991. Jeho předností je podpora většiny běžných platforem a možnost integrace s jazyky C, Java a platformou .NET, což mu umožní využívat knihovny určené pro tyto jazyky. Velkou předností je také uživatelský komfort a jeho jednoduchost na naučení. Python je v základní verzi velmi rychlý a je možno doplnit jej knihovnami které jeho výkon ještě zlepší.

Octave

Jedná se o volně dostupný klon komerčního MATLABu. Program je zaměřen na práci s maticemi, což je vhodné pro určité typy vědeckých výpočtů jako je právě metoda konečných prvků.

V základní verzi neobsahuje žádné grafické rozhraní, je však možné jej doplnit použitím nástaveb.

Určení programu a volba prostředí

Program je určen pro výpočty 2D problémů mechanického namáhání metodou konečných prvků. Vzhledem k povaze výpočtu byl jako vývojové prostředí vybrán MATLAB z důvodů jaho optimalizace pro daný typ výpočtů a uznadnění práce se získanými daty.

Součásti programu

Většina MKP programů je rozdělena na tři moduly. Prvním je preprocessor, který slouží k vytvoření geometrie, zadání okrajových podmínek a zatížení. Tato data jsou poté zpracována řešičem, jehož výstupem jsou číselné hodnoty určitého parametru, v tomto případě hodnoty posuvů ve dvou osách. Tato data je možné načíst postprocessorem, který je uspořádá a vykreslí. Vytvářený program se drží tohoto složení.

Teoretický základ

Trojúhelníkový prvek [1]

Jako tělesové (též masivní, anglicky „solid“ prvky) budeme označovat prvky umožňující diskretizaci spojitého prostředí, ať již v úlohách rovinných nebo prostorových. Nejjednodušším reprezentantem této kategorie ve 2D prostoru je trojúhelníkový prvek s lineárními bázovými funkcemi

Rovinný trojúhelníkový prvek má tři uzly ve vrcholech se šesti deformačními parametry.

Obě nezávislé složky posuvů v rovině, *u* a *v*, jsou nad prvkem aproximovány stejným typem polynomu. Základní tvar aproximační funkce:

$$u(x,y) = \begin{pmatrix} 1 & x & y \end{pmatrix} . \begin{pmatrix} a1 \\ a2 \\ a3 \end{pmatrix} \quad v(x,y) = \begin{pmatrix} 1 & x & y \end{pmatrix} . \begin{pmatrix} a4 \\ a5 \\ a6 \end{pmatrix} \tag{1.1}$$

Jestliže zapíšeme složky posuvů ve vrcholech prvku do matice deformačních parametrů

$$\boldsymbol{\delta} = [u_1, u_2, u_3, v_1, v_2, v_3]^T$$

můžeme je vyjádřit pomocí známých souřadnic vrcholů prvků, které jsou navzájem různé a neleží na jedné přímce jako

$$\boldsymbol{\delta} = \mathbf{S} \cdot \mathbf{a} = \begin{pmatrix} 1 & x1 & y1 & 0 & 0 & 0 \\ 1 & x2 & y2 & 0 & 0 & 0 \\ 1 & x3 & y3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x1 & y1 \\ 0 & 0 & 0 & 1 & x2 & y2 \\ 0 & 0 & 0 & 1 & x3 & y3 \end{pmatrix} \begin{pmatrix} a1 \\ a2 \\ a3 \\ a4 \\ a5 \\ a6 \end{pmatrix}$$

Vyjádřením **a** a dosazením do 1.1 dosáhneme obvyklého vyjádření aproximovaných funkcí *u* a *v* v závislosti na deformačních parametrech ve vrcholech prvku:

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{N}.\boldsymbol{\delta}$$

kde **N** je matice bázových funkcí.

K sestavení matice tuhosti musíme nejprve vyjádřit napětí a přetvoření prostřednictvím nezávislé funkce posuvů. S využitím geometrických rovnic a konstitutivních vztahů, aplikovaných na rovinnou úlohu, získáme složky přetvoření

$$\boldsymbol{\varepsilon} = \mathbf{L} \cdot \mathbf{N} \cdot \boldsymbol{\delta} = \mathbf{B} \cdot \boldsymbol{\delta}$$

Kde **L** je matice lineárních operátorů. **B** je matice tvarových funkcí přetvoření získaná z bázových funkcí N parciálními derivacemi, v tomto případě má tvar

$$B = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Složky napětí v rovině získáme za předpokladu platnosti hookeova zákona

$$\boldsymbol{\sigma} = \mathbf{D} \cdot \boldsymbol{\varepsilon} = \mathbf{D} \cdot \mathbf{B} \cdot \boldsymbol{\delta}$$

Matice materiálových konstant může nabývat různých tvarů podle toho, zda řešíme úlohu rovinné napjatosti, rovinné deformace nebo rotačně symetricklou. Pro potřeby práce bude uvažována varianta rovinné napjatosti.

$$\mathbf{D} = K \begin{pmatrix} 1 & C & 0 \\ C & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1 - C) \end{pmatrix}$$

$$C = \mu$$

$$K = \mu \cdot (1 - \mu^2)^{-1}$$

Dosazením $\boldsymbol{\sigma}$ a $\boldsymbol{\varepsilon}$ do výrazu pro energii napjatosti vyjádříme matici tuhosti trojúhelníkového prvku:

$$\mathbf{K_k} = \mathbf{S}^{-1T}\mathbf{B}^{T}\mathbf{DBS}^{-1} \cdot 0{,}5 \cdot h_k \cdot S$$

Kde h_k je tloušťka prvku a S je jeho plocha.

Globální matice tuhosti [2]

Po sestavení matic tuhostí prvků přistoupíme k sestavení globální matice tuhosti. K jejímu správnému sestavení slouží tzv. indexová a kódová čísla. Postup pro jejich získání je následující:

1. Očíslují se uzly sítě.
2. Každému prvku se přiřadí indexové číslo tvořené čísly uzlů které jsou vrcholy daného prvku.
3. Očísluje se každý posuv. Zdrojové materiály uvádějí, že známé posuvy se číslují nulou a dále se poté ve výpočtu nezahrnují, pro toto programové řešení však bylo výhodnější posuvy očíslovat tak jak jdou v řadě a až poté je z výpočtu vypustit.
4. Každému prvku se přiřadí kódové číslo. To je tvořeno šesti číslicemi v závislosti na jeho indexovém čísle, a to tak, že první jsou čísla posuvů daných uzlú v ose X, následované čísly posuvů v ose Y.

Nyní se již může sestavit matici tuhosti ešené oblasti. Matice tuhostí prvků jsou řádu 6, řád globální matice bude roven počtu posuvů. Sloupce a řádky každé matice tuhosti prvku očíslujeme podle kódového čísla prvku, sloupce a řádky globální matice očíslujeme vzestupně.

Do globální matice vsazujeme prvky z matic tuhostí prvků na místa odpovídající souřadnicím danými kódovými čísly v místě prvku. Prvky na stejné pozici přičítáme.

Řešení soustavy

Nyní zbývá již jen vyřešit soustavu rovnic

$$\mathbf{K} \cdot \mathbf{u} = \mathbf{f}$$

kde **f** je vektor zatěžujících sil. Matice tuhosti je však zatím singulární. Aby bylo možno soustavu řešit, je potřeba předepsat dostatečný počet okrajových podmínek. V dané soustavě se projeví tak, že jsou každý řádek a sloupec obsahující některý ze známých posuvů z matice tuhosti a každý řdek z vektoru zatížení vypuštěny. Po této úpravě nabývá soustava jediného řešení. Získané posuvy je dále možno zpracovávat postprocessorem.

Preprocessor

Soubor: `mkp_preprocessor.m`

Tento krok je možno přeskočit pokud již máte uložená data buď z předdefinovaných nebo z předchozího užití preprocessoru. Pto načtení předchozích dat použijte funkci `load()`.

Cílem preprocessoru je nastavit prostředí pro daný výpočet, načíst od uživatele potřebná data o materiálu, geometrii, uložení a zatížení simulovaného tělesa.

Vzhledem k tomu, že se vstupy a výstupy zadávají a zobrazují v základních jednotkách SI, budou čísla se kterými program pracuje malá. Proto je zvolen patřičný formát čísel. Program u všech uživatelských vztupů poskytuje informace o jednotkách a formátu zadávaných dat.

Programu je dále nutno zadat potřebné materiálové konstanty, konkrétně modul pružnosti v tahu E a poissonův poměr μ. Z nich se dopočítá modul pružnosti ve smyku G a matice materiálových konstant **D**.

Nyní se přistoupí k vytvoření geometrie. Prvním údajem je tloušťka součásti h. Tvar se zadává pomocí uzlů budoucích prvků, což dává uživateli kontrolu z hlediska pozdějšího umístění vazeb, zatížení a jemnosti sítě.

Program poté uzly přiřadí prvkům a sestaví indexová čísla prvků pomocí vestavěné funkce `delaunay()`. Dále se prvkům přiřadí kódová čísla, která budou použita řešičem pro sestavení matice tuhosti.

Skript dále spočítá plochy všech prvků. Jelikož známe souřadnice vrcholů včech trojúhelníků, nejjednodužší je spočítat plochy z délek stran pomocí Heronova vzorce:

$$s = \frac{1}{2}(a + b + c)$$

$$S = \sqrt{s(s-a)(s-b)(s-c)}$$

Heronův vzorec [3]

Nyní musí uživatel zadat okrajové podmínky a zatěžující síly. Je nutné dbát na to, aby bylo uložení nepohyblivé. Uživateli je zobrazen klíč s geometrií a kódy posuvů a sil působících v daných osách.

Po správném zadání všech parametrů preprocessor ohlásí že je hotov. Nyní je možné zadání uložit zavoláním funkce `save()`.

Solver

Soubor: `mkp_solver.m`

Na začátku se řešič zeptá, jakým způsobem má danou soustavu řešit. Ve většině případů postačí řešení normální (možnost false). Pokud je výsledná matice tuhosti špatně podmíněná, matlab to uživateli ohlásí. Poté je možné spusit program znovu a použít řešení s opravou, který upraví soustavu tak, aby byla její podmíněnost zlepšena.

Dále je zavedena matice tvarových funkcí přetvoření **B**.

Poté program přistoupí k sestavení matic tuhosti jednotlivých prvků $\mathbf{K}_k$ a globální matice tuhosti **K**.

Následuje aplikace okrajových podmínek na matici tuhosti a vektor zatížení.

Poté již jen proběhne řešení dle volby. Pokud se volí normální řešič, používá program funkci `linsolve()`, pokud je zvolen řešič pro špatně podmíněné matice, pak funkci `pcg()`.

V závěru jsou pak již známé posuvy doplněny posuvy vypočtenými a řešič ohlásí konec.

Postprocessor

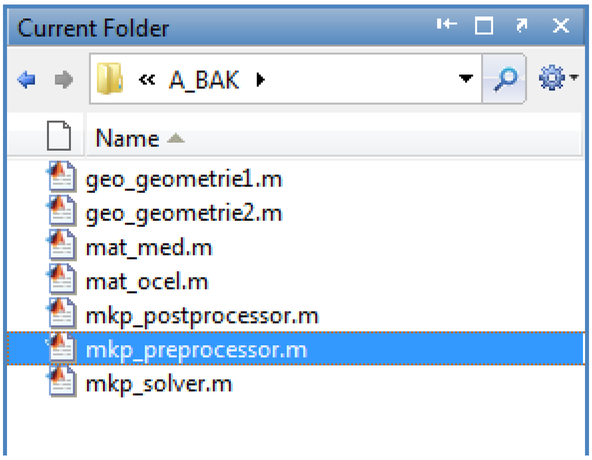
Soubor: `mkp_postprocessor.m`

Získaná data je možno graficky zobrazit pomocí postprocessoru. Jediným vstupním parametrem je měřítko posuvů. Obecně v pružnosti a pevnosti platí, že modelové výpočty jsou přesné pouze na velmi malých deformacích, které je potřeba zětšit aby byly vůbec viditelné na obrazovce.

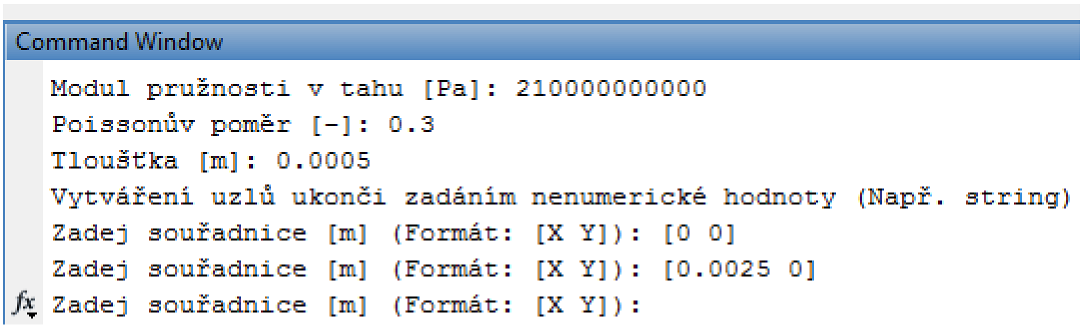
Výstupem postprocessoru je dvojí graf, který zobrazuje srovnání nedeformovaného a deformovaného stavu.

Řešení úlohy vytvořeným programem

Jako první je nutno spustit preprocessor.



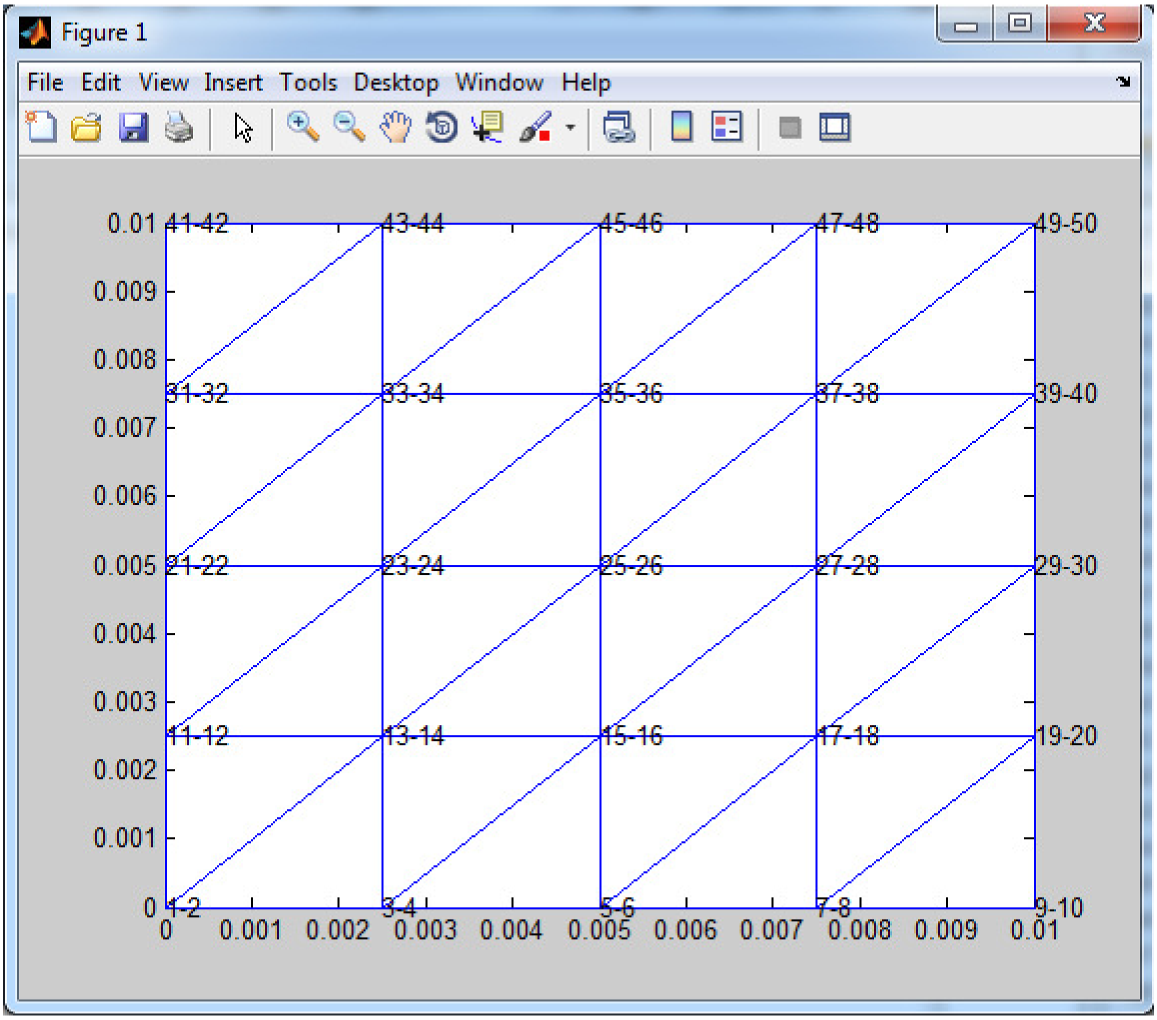
V preprocessoru se vyplní základní data o materiálu a poté můžete začít zadávat uzly geometrie.



Zadávání lze kdykoliv ukončit vložením nenumerické hodnoty.

```
Zadej souřadnice [m] (Formát: [X Y]): [0.01 0.01]
Zadej souřadnice [m] (Formát: [X Y]): false
```

Po dokončení se zobrazí náhled nedeformované sítě společně s klíčem pro zadávání okrajových podmínek.



Pro naši úlohu zvolíme rotační vazbu v levém spodním rohu a obecnou podporu zamezující pohyb v záporném směru osy Y v pravém spodním rohu. Z klíče je patrné které kódy je třeba zadat.

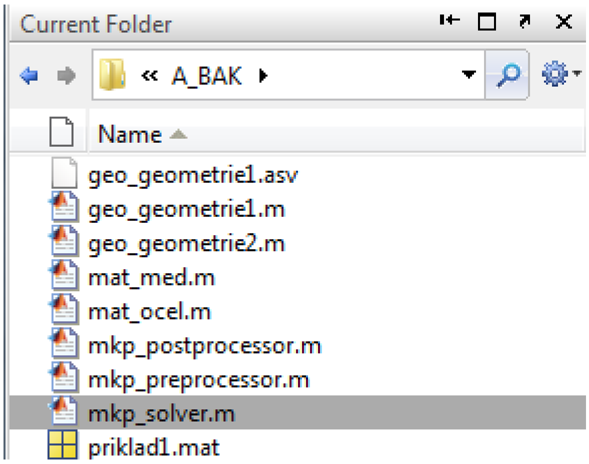
```
Zobrazen klíč pro zadávání okrajových podmínek. První číslo použij pro osu X, druhé pro Y.
Zadávání známých posuvů ukonči zadáním nenumerické hodnoty.
Zadej známé posuvy [m] ([CISLO_POSUVU HODNOTA]: [1 0]
Zadej známé posuvy [m] ([CISLO_POSUVU HODNOTA]: [2 0]
Zadej známé posuvy [m] ([CISLO_POSUVU HODNOTA]: [10 0]
Zadej známé posuvy [m] ([CISLO_POSUVU HODNOTA]: false
```

To samé opakujeme pro zatížení. Jako zatížení bude působit trojice sil v kladném směru osy X, každá o velikosti 1N.

```
Zadávání zatížení ukonči zadáním nenumerické hodnoty.
Zadej zatížení [N] ([CISLO_SILY HODNOTA]): [9 1]
Zadej zatížení [N] ([CISLO_SILY HODNOTA]): [29 1]
Zadej zatížení [N] ([CISLO_SILY HODNOTA]): [49 1]
Zadej zatížení [N] ([CISLO_SILY HODNOTA]): false
```

Nyní je možné data uložit pro pozdější využití.

```
Preprocessor - hotovo.
>> save('priklad1');
>> |
```



Nyjí je již možné nechat danou simulaci spočítat spuštěním řešiče. Ten má jediný vstup, totiž dotaz na metodu řešení. Na první pokus je vždy dobré použít standardní metodu. Pokud MATLAB zahlásí že je matice K špatně podmíněná, načtěte zadání znovu a solver spusťte ve speciálním režimu.

Command Window

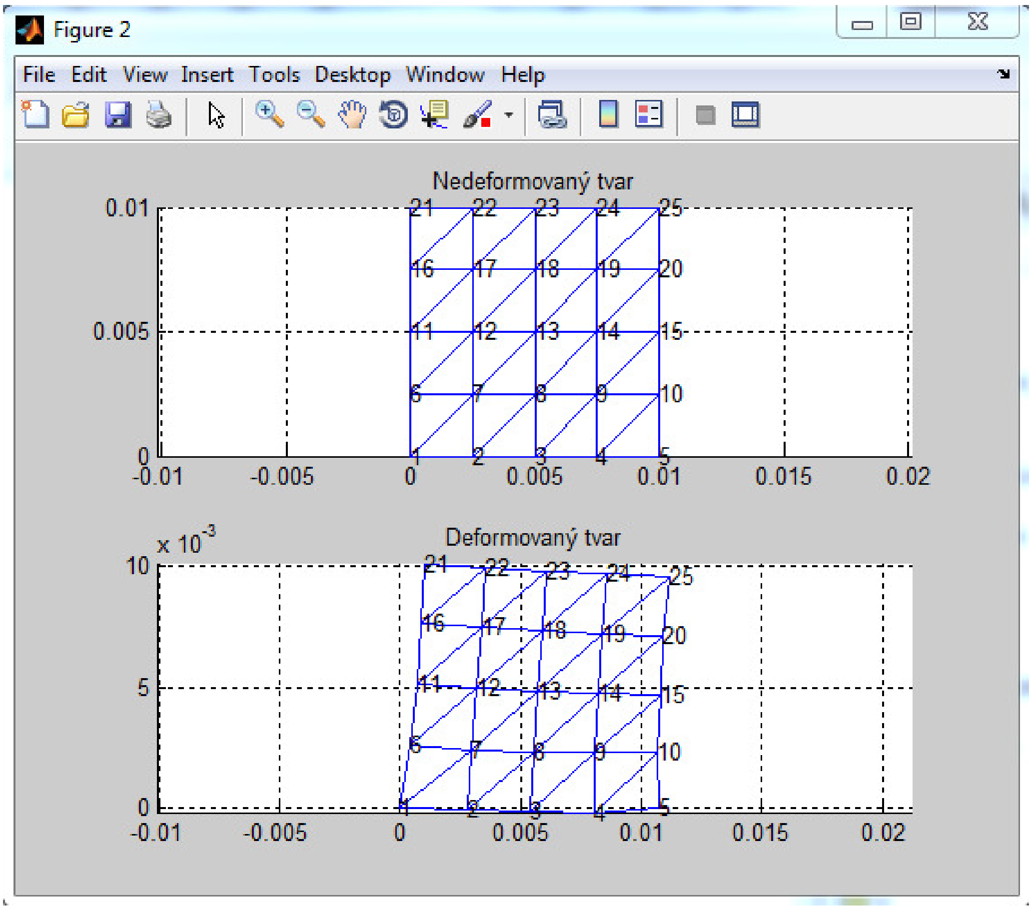
```
Použít solver pro špatně podmíněné matice? (true/false): false
Solver - hotovo.
```

Nyní již jsou spočítané posuvy v jednotlivých uzlech. Tato data je možné uložit a zobrazit postprocessorem. Jediným vstupem je měřítko posuvů. Předpokladem pro funkčnost rovnic obecné pružnosti a pevnosti je totiž že deformace jsou malé. Pro pohodlné zobrazení je třeba je zvětšit,

Command Window

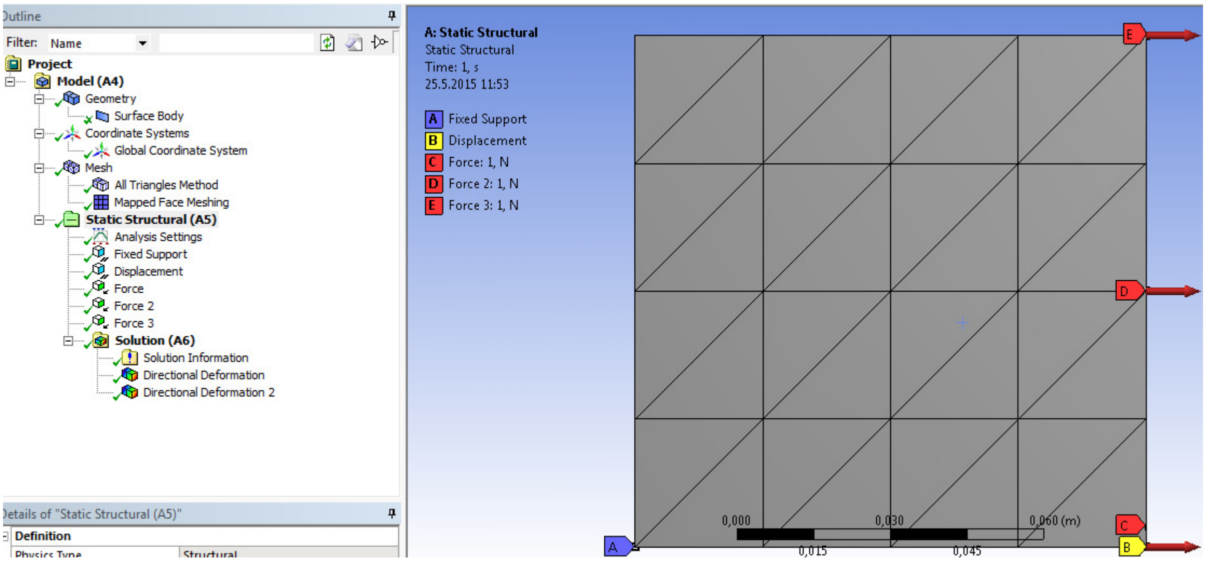
```
fx Měřítko deformaci: 1000
```

```
Command Window

Posuvy
  1.0e-005 *

                  0                   0
   0.02633148790520 0  -0.00551201155186 6
```

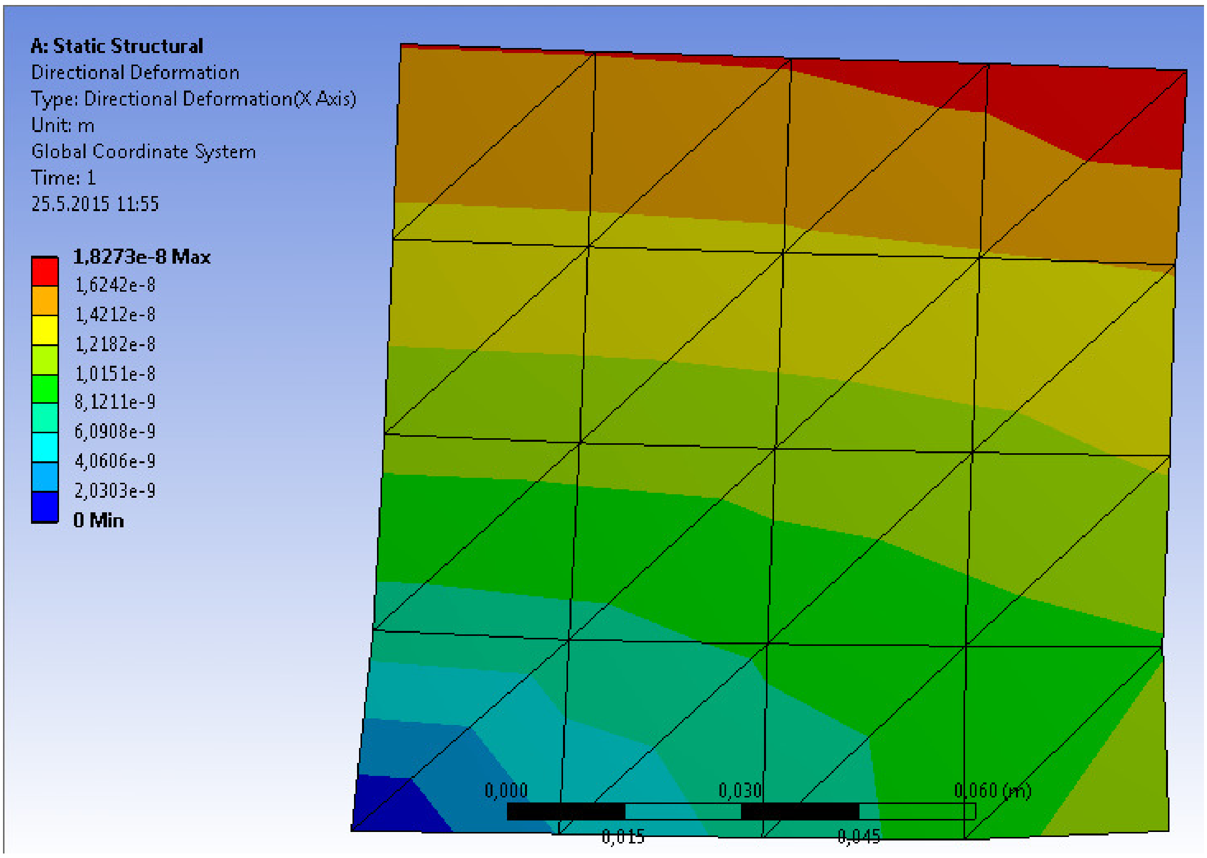


Řeení systémem ANSYS

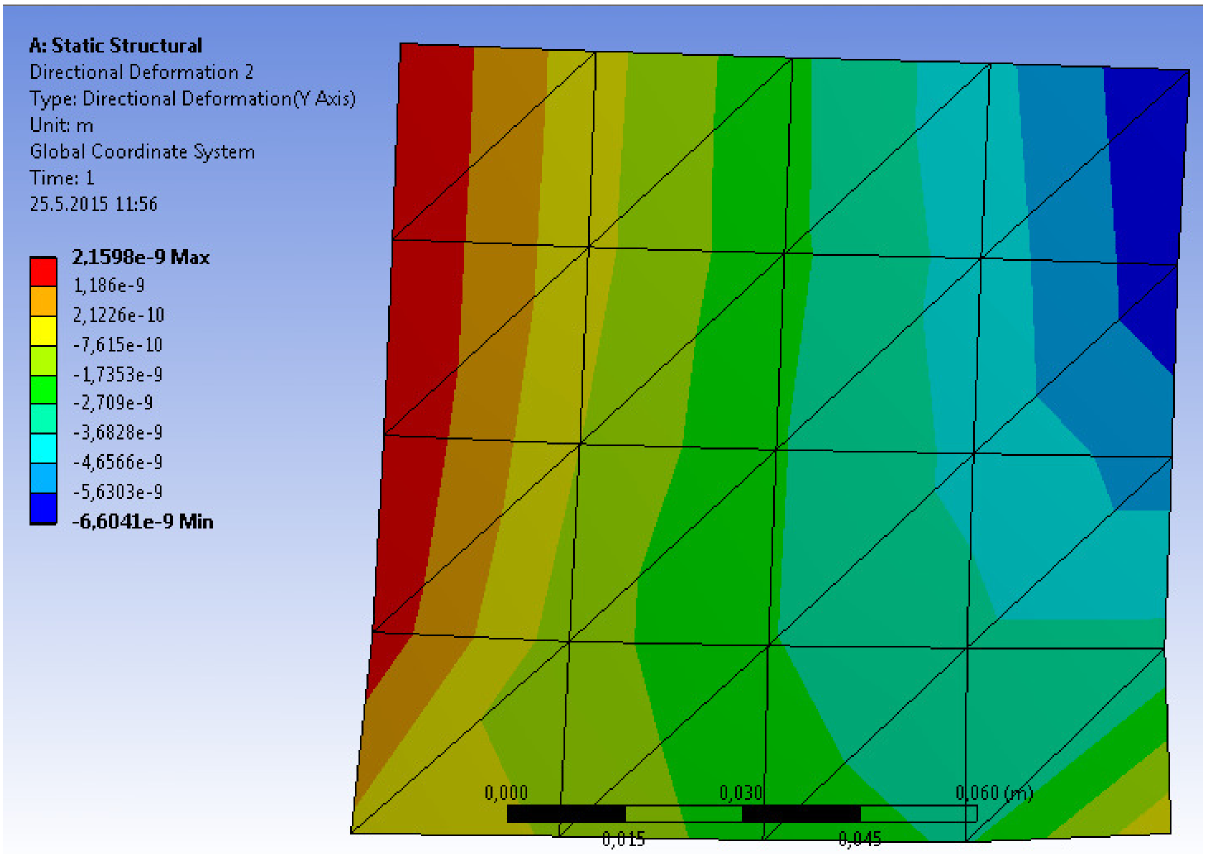
Pro srovnání je uveden stejný výpočet provedený v systémy ANSYS Workbench.



Geometrie a okrajové podmínky



Posuvy v ose X



Posuvy v ose Y

V současné době je možné rychle provádět výpočty metodou konečných prvků bez jakýchkolim nákladů na potřebný software. Existují k dispozici nástroje kvalitativně srovnatelné s komerčním software, které běří na volně dostupných platformách jako například Linux. Nevýhodou většiny těchto nástrojů z hlediska firem je že jsou licencované pouze pro nekomerční užití.

Komerční systémy na druhou stranu poskytují daleko větší komfort při používání, mají více funkcí zrychlujících práci na velkých projektech a v neposlední řadě rychlou zákaznickou podporu.

Ve své nejzákladnější variantě není metoda konečných prvků složitá na pochopení. Nevýhodou zvolené formy, konkrétně trojúhelníkového prvku, je jeho nepřesnost. Postprocessory s funkcí vyhlazování dat jako například ANSYS workbench mají tendenci skrývat špičkové hodnoty.

V neposlední řadě také ANSYS v základu trojúhelníkový prvek nepoužívá, ale nahrazuje ho čtyřuzlovým prvkem se dvěma uzly spojenými v jeden. Výsledný postup výpočtu se pak značně odlišuje a spolu s ním i výsledky.

Seznam použitých zdrojů

[1] PETRUŠKA, Jindřich. *MKP v inženýrských výpočtech* [online]. Brno, 2011 [cit. 2015-05-25]. Dostupné z: http://www.umt.fme.vutbr.cz/img/fckeditor/file/opory/RIV/MKP2011.pdf

[2] VLADIMÍR, Kolář. *Výpočet plošných a prostorových konstrukcí metodou konečných prvků*. 2. přepracované vydání. Praha: SNTL, 1979.

[3] Matematika.cz. *Matematika.cz* [online]. Nová Média [cit. 2015-05-25]. Dostupné z: http://www.matematika.cz/obsah-trojuhelniku