

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

ZPRACOVÁNÍ PŘEHLEDU MKP APLIKACÍ NA BÁZI OPEN SOURCE

SUMMARY OF FEM APPLICATION ON THE BASIS OPEN SOURCE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ TOMÁNEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. TOMÁŠ NÁVRAT, Ph.D.

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav mechaniky těles, mechatroniky a biomechaniky

Akademický rok: 2009/10

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Tománek Jiří

který/která studuje v **bakalářském studijním programu**

obor: **Strojní inženýrství (2301R016)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Zpracování přehledu MKP aplikací na bázi Open Source

v anglickém jazyce:

Summary of FEM application on the basis Open Source

Stručná charakteristika problematiky úkolu:

Práce je zaměřena na zpracování přehledu současně dostupných MKP programů na bázi otevřeného kódu. Student vybere jeden nebo dva programy, ve kterých se naučí pracovat a zpracuje v nich jednoduchou rovinnou úlohu. Získané výsledky vzájemně srovná příp. je porovná s výsledky z dostupného komerčního softwaru.

Cíle bakalářské práce:

- Studium a seznámení se s teoretickými základy metody konečných prvků.
- Zpracování přehledu programů (zdroj: internet)
- Výběr programu a zpracování jednoduché rovinné úlohy
- Srovnávací analýza

Seznam odborné literatury:

Kolář,V., Kratochvíl,J., Leitner,F., Ženíšek,A.: Výpočet plošných a prostorových konstrukcí metodou konečných prvků, SNTL Praha, 1979

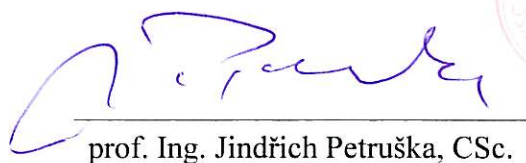
Petruška, J.: Počítačové metody mechaniky II, Metoda konečných prvků, FSI VUT Brno 2003, výukový text

Vedoucí bakalářské práce: Ing. Tomáš Návrat, Ph.D.

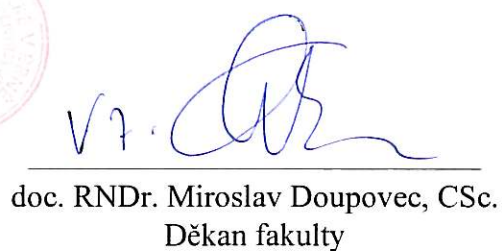
Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2009/10.

V Brně, dne 19.11.2009

L.S.



prof. Ing. Jindřich Petruška, CSc.
Ředitel ústavu



doc. RNDr. Miroslav Doupovec, CSc.
Děkan fakulty

ABSTRAKT

Práce je zaměřena na zpracování přehledu současně dostupných MKP programů na bázi otevřeného kódu, které jsou volně k dispozici na internetu. Jsou zde postupně prezentovány tři programy (Salome, Roshaz, Ansys), ve kterých byla pro ukázkou zpracována jednoduchá rovinná úloha. Na závěr bylo provedeno srovnání získaných výsledků a postupů při zadávání rovinných úloh ve třech výše zmíněných programech.

ABSTRACT

This work focuses on the summary of currently available FEM applications on the Open Source basis which are freely available on the Internet. There are three programs presented (Salome, Roshaz and Ansys) and in each of them there was an easy simple plane performed and demonstrated. In the final part, a comparison of the results and the procedures for the award plane task in the three above mentioned programs was carried out.

KLÍČOVÁ SLOVA

Metoda konečných prvků, Open Source, rovinná úloha, Salome, Roshaz, Ansys

KEYWORDS

Finite element Method, Open Source, plane task, Salome, Roshaz, Ansys

TOMÁNEK, J. *Zpracování přehledu MKP aplikací na bázi Open Source*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2010. 57 s. Vedoucí bakalářské práce Ing. Tomáš Návrát, Ph.D

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma *Zpracování přehledu MKP aplikací na bázi Open Source* jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

V Brně dne

.....

(Jiří Tománek)

PODĚKOVÁNÍ

Tímto bych rád poděkoval svému vedoucímu bakalářské práce, panu Ing. Tomáši Návratovi, Ph.D. za cenné rady a připomínky, které mi ochotně dával a zejména za jeho trpělivost a čas, který mi věnoval při přípravě této bakalářské práce. Dále bych chtěl poděkovat panu Ing. Petru Vosynkovi, který spolu s panem Ing. Tomášem Návratem, Ph.D. otevřeli kurz *Metoda konečných prvků a výpočetní systém Ansys*, který mi pomohl získat určité základy v MKP.

Dále bych chtěl poděkovat svým přátelům, rodině a přítelkyni, kteří mne po celou dobu studia podporovali.

ÚVOD.....	3
1. ÚVOD DO TEORIE MKP	4
1.1. NĚKOLIK ÚVODNÍCH SLOV	4
1.2. HISTORIE MKP	5
1.3. TEORETICKÝ ZÁKLAD MKP	5
1.3.1. Rovnice rovnováhy.....	6
1.3.2. Rovnice geometrické	6
1.3.3. Konstitutivní vztahy.....	7
2. MKP PROGRAMY NA BÁZI OPEN SOURCE	8
2.1. PODSTATA OPEN SOURCE	8
2.2. PŘEHLED PROGRAMŮ NA BÁZI OTEVŘENÉHO KÓDU	8
2.3. ROZBOR NĚKTERÝCH VOLNĚ ŠÍŘITELNÝCH PROGRAMŮ	10
2.3.1. Impact	11
2.3.2. CalculiX.....	11
2.3.3. Elmer.....	12
3. ROVINNÁ ÚLOHA.....	13
3.1. ZADÁNÍ	13
3.2. ROZBOR.....	13
4. ŘEŠENÍ ROVINNÉ ÚLOHY V PROGRAMU SALOME	15
4.1. ZÁKLADNÍ POPIS	15
4.2. POSTUP PŘI ŘEŠENÍ ROVINNÉ ÚLOHY	16
5. PROGRAM ROSHAZ.....	25
5.1. ZÁKLADNÍ POPIS	25
5.2. POSTUP PŘI ŘEŠENÍ ROVINNÉ ÚLOHY	26
6. ŘEŠENÍ ROVINNÉ ÚLOHY V PROGRAMU ANSYS	35
6.1. ZÁKLADNÍ POPIS	35
6.2. POSTUP PŘI ŘEŠENÍ ROVINNÉ ÚLOHY	35
7. SROVNÁVACÍ ANALÝZA	48
7.1. SROVNÁNÍ Z HLEDISKA ZPRACOVÁNÍ A SROZUMITELNOSTI NÁPOVĚDY	48
7.2. SROVNÁNÍ POSTUPU PŘI VYTVOŘENÍ MODELU GEOMETRIE	49
7.3. SROVNÁNÍ POSTUPU PŘI VYTVOŘENÍ SÍTĚ KONEČNÝCH PRVKŮ	50
7.4. SROVNÁNÍ POSTUPU PŘI ZADÁVÁNÍ OKRAJOVÝCH PODMÍNEK A ŘEŠIČE	51
7.5. SROVNÁNÍ POSTUPU PŘI VYKRESLENÍ VYPOČTENÝCH VÝSLEDKŮ.....	52
7.6. SROVNÁNÍ VYPOČTENÝCH VÝSLEDKŮ	52

7.7. PŘEHLED PROGRAMŮ A MOŽNOSTI JEJICH POUŽITÍ	53
ZÁVĚR.....	54
SEZNAM POUŽITÝCH VELIČIN.....	55
SEZNAM LITERATURY.....	56
PŘÍLOHY	57

ÚVOD

Tato práce se zabývá v současné době velmi populární a rozšířenou numerickou metodou a tou je metoda konečných prvků (MKP). V první kapitole této práce je popsáno, proč je vhodné se touto metodou zabývat a proč nám šetří náš čas, co tato metoda dokáže řešit a v jakých inženýrských oborech ji můžeme výhodně použít. Dále popisuje historický vývoj a počáteční myšlenky dvou nezávisle pracujících zakladatelů. Poté je zde uveden základní princip a teoretický základ MKP teorie. Dále je také stručně popsán přehled základních neznámých rovnic, které potřebujeme vyřešit. V druhé kapitole je vysvětlen pojem slova Open Source v oblasti metody konečných prvků a jsou zde uvedeny výhody a nevýhody této licence. V další části práce je vytvořen stručný přehled MKP programů (*tab.1*) na bázi Open Source, který obsahuje název a typ programu, dále na kterém operačním systému můžeme tento program použít, typ analýzy, který dokáže řešit a nakonec [www odkaz](#), ze kterého můžeme program bezplatně stáhnout. Z několika desítek těchto dostupných volně šiřitelných programů jsou některé stručně popsány a poté jsem vybral jeden, který je popsán detailněji a následně jsou rozebrány jednotlivé kroky při postupu jednoduché rovinné úlohy řešené pomocí již zmíněné metody konečných prvků. V práci se provádí řešení jednoduché rovinné úlohy z toho důvodu, abychom si pak byli schopni ověřit správnost vypočtených výsledků pomocí snadného analytického řešení a zjistili tak, jestli nám program vypisuje přibližně stejné výsledky. V další části práce je tato úloha řešena také pomocí tzv. „demo“ verze komerčního programu a na závěr pro porovnání v komerčním programu - Ansysu. V samotném závěru této práce je provedena srovnávací analýza z několika hledisek a následně je vytvořena *tab. 2*, kde je uveden přehled použitých programů v této práci a také jsou uvedeny typy úloh, které jednotlivé programy dokáží řešit. Toto téma bakalářské práce jsem si vybral z toho důvodu, protože mne tato problematika zajímá a téma řešeného problému mi přijde v dnešní době jako velice aktuální.

1. ÚVOD DO TEORIE MKP

1.1. Několik úvodních slov

Ve strojírenství je velmi důležité, aby konstruktér při návrhu nové součásti, stroje apod. dokázal odhadnout chování v různých situacích. Konstruktér musí učinit několik výpočtů, aby mohl dostatečně dimenzovat součásti nového stroje. Nejenom, že jsou kladeny nároky na minimální finanční náklady, nízkou hmotnost, vysokou tuhost a životnost konstrukce, ale také navržená součást musí splňovat jak správnou funkci, tak především různé bezpečnosti. Většinu inženýrských problémů ovšem nelze řešit analyticky, protože se jedná o složité matematické operace, které člověk není schopen v reálném čase vypočítat. V současné době je ovšem velmi účinné využívat systém CAE (Computer Aided Engineering), což znamená počítačem podporované konstruování a různé typy výpočtů a analýz. Pod CAE můžeme zahrnout jednu z nejmodernějších numerických metod s názvem metoda konečných prvků. Díky své obecné matematické formulaci umožňuje MKP řešit např. tyto problémy: v mechanice pevných těles dokáže řešit jak statické strukturální úlohy (provedení deformačně napěťové analýzy), tak i úlohy dynamické (kmitání, stanovení vlastních frekvencí) a také lze tuto metodu použít v oblasti lomové mechaniky. Pomocí MKP můžeme také řešit přenos tepla, proudění kapalin a plynů, elektromagnetické pole apod.

Podle druhu licence bychom mohli MKP programy rozdělit do dvou kategorií. V první by se jednalo o software komerčního typu, který je navržen týmem odborníků a je ověřen různými certifikáty apod. Komerční programy jsou také jednodušší z hlediska ovládání pro běžného uživatele, protože obsahují všechny potřebné nástroje k provedení analýzy (preprocessing, generování sítě konečných prvků, řešiče, postprocessing). Další výhodou je zde technická podpora, která se provádí ve formě různých školení apod. Naopak nevýhodou je příliš vysoká cena (řádově stovky tisíc korun), a proto je tento software dostupný spíše pro „velké“ firmy, které si mohou dovolit zaplatit tuto licenci. Mezi nejúspěšnější komerční programy v oblasti MKP patří např. Ansys, Abaqus, Nei Software, Adina, Cosmos. V druhé kategorii rozdělení by šlo o tzv. Open Source software, který je podrobně rozebrán v druhé kapitole této práce. Obsah této kapitoly je čerpán z následujících zdrojů: [5], [9].

1.2. Historie MKP

Kořeny metody konečných prvků spadají do 40. let 20. století. Myšlenky této metody můžeme poprvé spatřit v práci ruského inženýra Alexandra Hrennikoffova (1941) a německého matematika Richarda Couranta (1942). I když postupy těchto zakladatelů byly rozdílné, měly jednu společnou charakteristiku: rozdělení spojitého tělesa na diskrétní části. Hrennikoffov rozděluje části pomocí mřížky, zato Courant použil při řešení eliptické parciální diferenciální rovnice (ty byly odvozeny již dříve Rayleighem, Ritzem a Galerkinem) druhého stupně a oblast rozdělil do konečného počtu trojúhelníkových částí.

Vývoj metody je často spojován s rokem 1956 (kde byla poprvé zmíněna v publikaci [1]), jejíž pomocí se začaly řešit úlohy statické strukturální mechaniky pomocí číslicového počítače, který dokázal vyřešit větší počet soustav algebraických rovnic. Název této metody vznikl z anglické verze *The Finite Element Method* (FEM) v roce 1960, kde byl poprvé zmíněn v článku. Jak už samotný název napovídá, základním kamenem metody je prvek konečných rozměrů, na rozdíl od klasické analytické pružnosti, kde prvek považujeme za nekonečně malý element. Později roku 1967 Prof. Zienkiewicz poprvé napsal první knihu o MKP [2]. Obsah této kapitoly je čerpán z následujících zdrojů: [3], [4], [6].

1.3. Teoretický základ MKP

Základním principem MKP je rozdělení (diskretizace) tělesa na mnoho malých částí (konečných rozměrů). Složitě tvary se převedou na jednoduché - obdélníky, čtyřúhelníky, trojúhelníky atd. Podstatnou výhodou takového rozdělení je to, že jednoduché tvary se snadno matematicky popisují. Kromě rozdělení na jednoduché prvky je také charakteristický počet a poloha uzlů. Uzly spolu s elementy nám vytvářejí tzv.sít'. Hustota sítě nám podstatně ovlivňuje dosaženou přesnost výsledků.

Teoretickým základem MKP je tzv. Lagrangeův variační princip, nebo jinak řečeno: deformační varianta. V pružnosti-pevnosti při řešení obecného prostorového tělesa hledáme řešení pro 15 neznámých proměnných funkcí. Získáme tedy 15 rovnic, které musí být doplněny okrajovými podmínkami (geometrické nebo silové), přičemž se nikdy těchto 15 rovnic neřeší současně. Pomocí vzájemného dosazování obecných rovnic pružnosti se postupně vyloučí nějaké neznámé funkce. Poté dostaneme pouze vztahy, které v deformační variantě obsahují pouze neznámé posuvy, které počítáme ve všech uzlech.

Z těchto posuvů se poté dopočítají přetvoření a následně napětí. Tato metoda je nejpoužívanější jak při numerickém řešení, tak i při analytickém.

Jedná se o těchto 15 neznámých funkcí:

- tři posuvy u, v, w
- šest přetvoření $\varepsilon_x, \varepsilon_y, \varepsilon_z, \gamma_{xy}, \gamma_{yz}, \gamma_{zx}$
- šest napětí $\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{zx}$

1.3.1. Rovnice rovnováhy

Jedná se o tři parciální diferenciální rovnice. Tyto rovnice představují podmínky vnitřní rovnováhy elementárního prvku, na který kromě působení složek napětí působí také vnější objemová síla (např. odstředivá, gravitační, ...).

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

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

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

1.3.2. Rovnice geometrické

Jedná se o šest parciálních diferenciálních rovnic, které vyjadřují vztahy mezi posuvy a přetvořeními (podmínkou je malé přetvoření, řádově 10^{-2} a menší).

$$\varepsilon_x = \frac{\partial u}{\partial x} \quad (1.1.4)$$

$$\varepsilon_y = \frac{\partial v}{\partial y} \quad (1.1.5)$$

$$\varepsilon_z = \frac{\partial w}{\partial z} \quad (1.1.6)$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \quad (1.1.7)$$

$$\gamma_{yz} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \quad (1.1.8)$$

$$\gamma_{xz} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \quad (1.1.9)$$

1.3.3. Konstitutivní vztahy

Jedná se o zbývajících šest lineárních algebraických rovnic. Tyto vztahy ovšem platí pro homogenní, izotropní, lineárně pružný materiál.

$$\varepsilon_x = \frac{1}{E} [\sigma_x - \mu(\sigma_y + \sigma_z)] \quad \gamma_{xy} = \frac{1}{G} \tau_{xy} \quad (1.1.10)$$

$$\varepsilon_y = \frac{1}{E} [\sigma_y - \mu(\sigma_x + \sigma_z)] \quad \gamma_{yz} = \frac{1}{G} \tau_{yz} \quad (1.1.11)$$

$$\varepsilon_z = \frac{1}{E} [\sigma_z - \mu(\sigma_x + \sigma_y)] \quad \gamma_{xz} = \frac{1}{G} \tau_{xz} \quad (1.1.12)$$

$$G = \frac{E}{2(1 + \mu)} \quad (1.1.13)$$

U statické strukturální úlohy se numericky řeší soustava algebraických rovnic (lineárních nebo nelineárních) pomocí konečného počtu neznámých hodnot a tím se nahrazuje řešení parciálních diferenciálních rovnic. Pomocí diskretizace se pak v jednotlivých uzlech prvků řeší neznámé posuvy pomocí následující rovnice:

$$K \cdot U = F \quad (1.1.14)$$

Obsah této kapitoly je čerpán z následujících zdrojů: [3], [6], [11].

2. MKP PROGRAMY NA BÁZI OPEN SOURCE

2.1. Podstata Open Source

Podle druhu licence máme k dispozici kromě komerčních softwarů také tzv. Open Source software. V této práci však budu používat český název a to: program na bázi otevřeného kódu. Je tedy ještě nutné si vysvětlit významy těchto výše zmíněných anglických slov. Význam slova Open je „otevřený, přístupný“, z čehož vyplývá, že tento program je volně k dispozici. Druhé slovo Source znamená text programu, který je napsán v nějakém programovacím jazyce (C++, Python, Java, ...). Tomuto textu ovšem rozumí pouze člověk. Aby tomu rozuměl i počítač, je nutné tento text zkompileovat, což pro člověka znamená naprosto nepochopitelný text složený z různých znaků a čísel.

To, že programy na bázi otevřeného kódu jsou volně k dispozici je obrovská výhoda, jak pro malé firmy, tak i pro studenty, protože se zde neplatí žádná licence. Další výhodou tohoto programu je, že každý uživatel si může tento kód otevřít, různě upravit a přizpůsobit svým potřebám, popřípadě ho může za jistých podmínek dál volně šířit. Ovšem ne každý uživatel je schopný programátor, a tak mu víceméně nezbyvá nic jiného, než buď si koupit komerční software nebo se naučit programovat a nebo také může doufat, že program, který má k dispozici už je v takové fázi, ve které lze provést požadovaná analýza apod. Programy na bázi otevřeného kódu jsou vytvářeny menšími skupinami nebo jednotlivci z výzkumných ústavů, univerzit apod. Ovšem nevýhodou je ta skutečnost, že některé programy neobsahují celou sadu nástrojů v jednom programu. K provedení analýzy je tedy potřebné jednotlivé aplikace doplnit v jiných dostupných volně šiřitelných programech. Obsah této kapitoly byl čerpán z těchto zdrojů: [5], [7], [8].

2.2. Přehled programů na bázi otevřeného kódu

V dnešní době si na internetu můžeme vybrat z velkého množství MKP programů, které jsou na bázi otevřeného kódu. Většinou se nejedná o programy, které obsahují vše potřebné pro výpočet, ale jen určité části programu, jako např. vytváření geometrie, síť konečných prvků, nebo různé matematické řešiče, knihovny apod. V následující tabulce je uveden stručný seznam některých programů, které jsem našel na internetu.

název programu	typ	typ OS	typ analýzy	www odkaz
Aladdin	t	Lin	S, D	http://www.isr.umd.edu/~austin/aladdin.html
CalculiX	s	Win, Lin	S, E, T	http://www.calculix.de/ http://www.dhondt.de/
Curly3D	l	Win, Lin	J	http://curly3d.sourceforge.net/
Deal.II	l	Win, Lin	S, T	http://www.dealii.org/
Elmer	s	Win, Lin, Mac	S, E, T, J	http://www.csc.fi/english/pages/elmer
Felib	l	Win, Lin	S	http://www.softeng.cse.clrc.ac.uk/felib3/
FEMM	s	Win, Lin	E	http://www.femm.info/wiki/HomePage
Felt	s	Win, Lin	S, T	http://felt.sourceforge.net/
Fenics	l	Lin, Mac	S, T, CFD	http://www.fenics.org/wiki/FEniCS_Project
FreeFem++	PD E	Win, Lin, Mac	S, E, T	http://www.freefem.org/ff++/index.htm
Getfem ++	l	Win, Lin	S, E, T	http://home.gna.org/getfem/getfem_intro
Gmsh	s	Win, Lin, Mac	S, E, T	http://www.geuz.org/gmsh/
Impact	s	Win, Lin	D	http://impact.sourceforge.net/
libMesh	l	Lin	S, T, J	http://libmesh.sourceforge.net/index.php
LISA (demo)	s	Win	S, E, T, CFD	http://www.lisa-fet.com/index.htm
ModuleF	l	Win, Lin	S, E, T, J	http://www-rocq.inria.fr/modulef/english.html
Ofeli	l	Win, Lin	S, E, T, CFD	http://www.ofeli.net/
OOFEM	l	Lin	S, CFD	http://www.oofem.org/cz/oofem.html

OpenFOAM	l	Lin	S, T, J	http://www.openfoam.com/
Open CASCADE	s	Win, Lin	S, J	http://www.opencascade.org/
NGSolve	l	Win, Lin	S, E, T	http://www.hpfem.jku.at/ngsolve/index.html
ROSHAZ (demo)	s	Win	S	http://www.roshaz.com/home.html
Salome	s	Lin	S, T, CFD	http://www.salome-platform.org/
SLFFEA	s	Lin	E, CFD	http://slffea.sourceforge.net/
Z88	s	Win, Lin, Mac	S	http://www.z88.org/

Tab.1 Přehled programů na bázi otevřeného kódu

Legenda:

Typ programu: s – řešič (solver) , l – knihovna (library), t – toolbox,

PDE – parciální diferenciální rovnice (Partial differential equation)

Typ operačního systému: Win – Windows, Lin – Linux, Mac – Macintosh

Typ analýzy: S – statická strukturální, D - dynamická, E – elektromagnetismus,

T – přenos tepla, CFD – mechanika tekutin

2.3. Rozbor některých volně šiřitelných programů

V této části kapitoly jsou rozebrány některé programy, ve kterých jsem chtěl původně zpracovat jednoduchou rovinnou úlohu, protože všechny tyto programy jsou spustitelné na operačním systému Windows a zdály se zpočátku jednoduché. Nakonec jsem ani jeden program nevyužil, protože zaprvé pomocí programu Impact můžeme řešit pouze úlohy dynamických jevů. Za druhé program CalculiX mi přišel jako příliš složitý program, ve kterém se podle mého názoru nedá vůbec orientovat a tvorba jednotlivých kroků při řešení úlohy je spíše na úrovni programování, což pro mne jako „neprogramátora“ nebylo přijatelné. Důkazem toho je ta skutečnost, že v nápovědě je uveden jednoduchý příklad (řešení vetknutého nosníku na konci zatíženého silou) s počtem 22 stran. V posledním případě

program Elmer po importování geometrie automaticky vykresloval velmi hrubou síť pomocí trojúhelníkových elementů a tuto síť se mi nijak nepodařilo editovat, jestli je to ovšem možné. Proto jsem se nakonec rozhodl pro vyřešení rovinné úlohy v jiném volně šiřitelném programu.

2.3.1. Impact

Tento program byl navržen tak, aby byl jednoduchou alternativou k vytvoření moderních MKP kódů. První verze byla vytvořena v roce 1989. Cílem vývoje tohoto programu bylo umožnit programátorům snadný způsob, jak přidat funkce programu bez nutnosti zadávat dalších částí kódů. Kódy jsou používány k simulaci dynamických jevů a jsou schopny zahrnovat velké deformace, např. srážky jakéhokoliv druhu, tvářecí operace apod. Impact je napsaný v programovacím jazyce Java. Tato volba jazyka se může zdát zvláštní, protože Java je pomalá, ale na druhou stranu je výhodná v tom, že je jednoduchá a nezávislá na operačním systému. Celý kód provádí výpočet zrychlení tělesa za použití malých přírůstků v čase k převedení tohoto zrychlení do malého posunutí tělesa. Toto posunutí je poté použito pro výpočet reakční síly od soustavy. Následně je síla použita pro výpočet zrychlení. Tento proces se pořád opakuje.

Impact se skládá z následujících částí:

- Preprocesor
- Procesor
- Post procesor
- Graf

Preprocessor je výchozím bodem pro uživatele. Používá se pro vytváření geometrie, sítě konečných prvků, stanovení geometrických a silových okrajových podmínek a také pro nastavení řešení „time step“. Preprocessor pracuje v plném 3D zobrazení. V processoru se provádí výpočet. Výsledky jsou automaticky zapsány do *.res souboru, který se načte do postprocessoru. Ten se používá pro zobrazení výsledků.

2.3.2. CalculiX

CalculiX byl vyvinut v Německu v roce 1998. Tímto programem lze vytvářet modely, počítat a následně zobrazovat výsledky. Obsahuje nástroje pro vytvoření analýzy preprocessor (CGX) a postprocessor (CCX). Ovládání obou modulů se provádí přes příkazový řádek, v kterém se dá pouze psát, jestliže je aktivní hlavní nabídkové okno. Řešitel využívá

Abaqusový vstupní formát, proto je možné použít komerční preprocessory. CalculiX je tedy schopen přečíst formáty i jiných komerčních programů jako např. Nastran, Abaqus, Ansys, Duns – CFD, Isaac- CFD a OpenFoam a to ve fázi preprocessoru a nebo postprocessoru.

CalculiX dokáže řešit úlohy statické strukturální (lineární, nelineární), frekvenční, dynamické (lineární, nelineární), vzpěru, přenosu tepla, Laplaceovy a Helmholtzovy problémy této analogie: elektrostatika, mazání, difúzní přenos hmoty, tok podzemní vody, akustika apod.

2.3.3. Elmer

Tento program byl vytvořen v roce 1995 ve spolupráci s finskými univerzitami, výzkumných ústavů a průmyslu. Elmer byl původně navržen pro úlohy dynamiky (stlačitelných a nestlačitelných) tekutin. Dnes je však rozšířen o úlohy: statické strukturální a kvantové mechaniky, elektromagnetismu, přenosu tepla a akustiky. Model je popsán parciálními diferenciálními rovnicemi. Tento program je napsán ve Fortran90, C a C++.

Elmer je rozdělen do několika samostatných spustitelných souborů, které také mohou být použity samostatně a skládá se z těchto částí:

- ElmerGUI
 - o je grafické uživatelské rozhraní pro definování a generování vstupních dat pro ElmerSolver (geometrie, okrajové podmínky, fyzikální vlastnosti)
 - o je schopen řešit 2D a 3D úlohy
 - o generuje síť a dokáže import sítě vytvořenou jiným programem
- ElmerSolver
 - o zpracovává vstupní data a vygenerovanou síť
- ElmerPost
 - o je nástroj na vizualizaci pro zpracování číselných výsledků vypočtených v ElmerSolver nebo také v jiných MKP programech

3. ROVINNÁ ÚLOHA

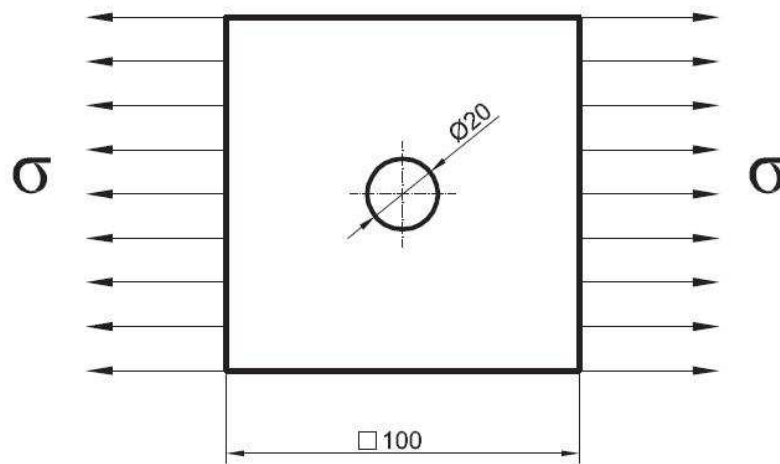
Rovinná úloha je zde zvolena z toho důvodu, abychom si pak byli schopni ověřit správnost vypočtených výsledků pomocí snadného analytického řešení a zjistili tak, jestli nám programy vypisují přibližně stejné výsledky.

3.1. Zadání

U součásti dle *obr. 1* proveďte deformačně-napěťovou analýzu. Součást je vyrobena z ocelového plechu.

Zadané parametry:

- modul pružnosti v tahu $E = 2,1 \cdot 10^5 \text{ MPa}$
- Poissonova konstanta $\mu = 0,3$
- tloušťka plechu $t = 1 \text{ mm}$
- tahové napětí $\sigma = 100 \text{ MPa}$



Obr. 1 Zadaná součást

3.2. Rozbor

Tato úloha je symetrická podle vodorovné a svislé osy, proto při samotném řešení použijeme pouze $\frac{1}{4}$ součásti (*obr. 2*). Což je výhodné jak z hlediska řešení menší soustavy rovnic, tak i kvůli vazbám. Těleso musí být nepohyblivé, proto použijeme okrajové podmínky, které jsou na *obr. 2.*, kde červená barva zobrazuje geometrickou okrajovou podmínku, která zajišťuje nulový posuv v ose y . Zelená barva zobrazuje geometrickou

podmínku, která zajišťuje nulový posuv v ose x . Poslední okrajová podmínka a to silová je vyznačena žlutou barvou. V této úloze se zde využije teorie rovinné napjatosti, kde v našem případě jedno hlavní napětí $\sigma_z = 0 \text{ MPa}$, protože tloušťka je zanedbatelná vůči ostatním rozměrům.

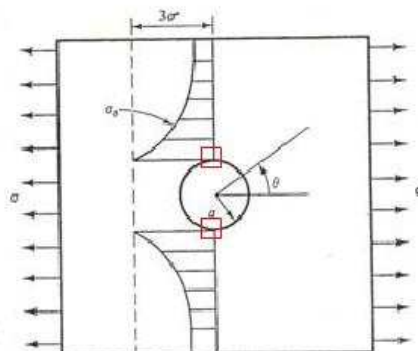


Obr. 2 Čtvrtina součásti s barevným vyznačením okrajových podmínek

V této úloze představuje díra koncentrátor napětí, takže v této oblasti očekáváme nárůst napětí. Pokud je díra zanedbatelná vůči ostatním rozměrům, pak tvarový součinitel α se rovná 3 (obr. 3). Analytickým přístupem by se hodnota napětí řešila následovně:

$$\sigma = \alpha \cdot \sigma_{nom} = 3 \cdot 100 \text{ MPa} = 300 \text{ MPa}$$

Hodnotu tvarového součinitele můžeme snadno nalézt na internetu:



$$(K_g)_{max} = \frac{\sigma_{max}}{\sigma_{nom}} = 3$$

Obr. 3 Hodnota tvarového součinitele [12]

4. ŘEŠENÍ ROVINNÉ ÚLOHY V PROGRAMU SALOME

4.1. Základní popis

Salome vznikl ve Francii v roce 2000 a je mateřskou společností OpenCASCADE, který se specializuje na vývoj a integraci numerické simulace kódů (pro preprocessor a postprocessor). Tento program dokáže řešit úlohy statické strukturální mechaniky, mechaniky tekutin a termomechaniky. Tvůrci programu Salome jako jedny z mála volně šiřitelných programů nabízejí školení (pro začátečníky i pokročilé), které už ovšem není bezplatné, také on-line poradenství již není zdarma.

Tento program obsahuje několik modulů:



Obr. 4 Moduly

- Geometry - V tomto modulu se vytváří geometrie součástí. Pro rovinnou úlohu lze použít tzv. „sketch“, kde lze kreslit geometrie pouze pomocí přímek, křivek nebo oblouků. Pro prostorové modely se vyplatí použít import z jiného softwaru, protože zde nejsou tak pohodlné modelovací nástroje jako např. v Inventoru. Proto je výhodné model vytvořený v Inventoru exportovat s použitím přípony *.igs a následně importovat do Salome, který kromě formátu *.igs také podporuje *.iges, *.step, *.stp a *.sat.
- Mesh - Použitím tohoto modulu lze vytvářet a editovat síť. Je zde také možnost importu sítě z jiného softwaru pomocí formátů *.dat, *.unv, *.med. Pokud bychom chtěli spustit řešení v jiném softwaru, je zde možnost exportu do souboru s příponami: *.dat, *.med, *.unv, *.stl.
- Post-Pro - Graficky zobrazuje vypočítané výsledky (=postprocessor).

- Supervisor - Tento nástroj dovoluje popsat a řídit rozdílně spojené schopnosti mezi distribuované komponenty a zvláště řešitelské moduly. Komponenta může být buďto existující SALOME modul provozovaný na vlastním počítači a na svém vlastním operačním systému (tj. distribuovaná komponenta) nebo Python komponenta provozovaná lokálně a navržená uživatelem.
- Med - Účelem tohoto modulu je poskytnout standard pro ukládání a obnovu počítačových údajů vztahujících se sítí a polí a usnadnit výměnu mezi kódy a řešiči.
- Aster - Řešič, který zpracovává vstupní data a vygenerovanou síť.

Program Salome, jako jeden z mála bezplatných programů, obsahuje celou sadu nástrojů pro vytvoření analýzy. Není zde nutné programovat a přidávat další aplikace (např. řešič Aster je již dodatečně vložen), což je pro běžného uživatele velice výhodné.

Obsah této kapitoly je čerpán ze zdroje [16].

4.2. Postup při řešení rovinné úlohy

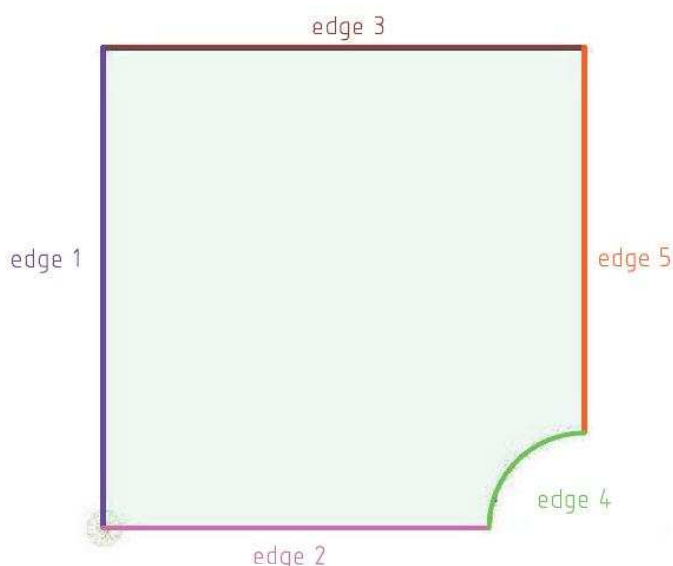
V prvním kroku zapneme modul *Geometry*, kde nejprve vybereme *New Entity* a pak se z nabídky vybere *Sketch*. Následně se pomocí velmi jednoduchých příkazů nakreslí geometrie.



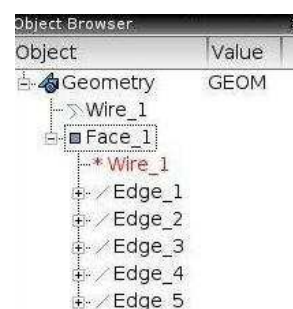
Obr. 5 Nabídka

Poté se vytvoří tzv. face, aby byla vytvořena plocha (obr. 6a). Tato plocha se následně rozdělí na jednotlivé „edge“ (hrany), jak je naznačeno na obr. 6b. Na tyto hrany pak zadáme silové a geometrické okrajové podmínky.

Pozn. Na obr. 6a je vytvořená plocha ještě dodatečně doplněna barevným označením jednotlivých „edge“.



Obr. 6a Vytvořená plocha



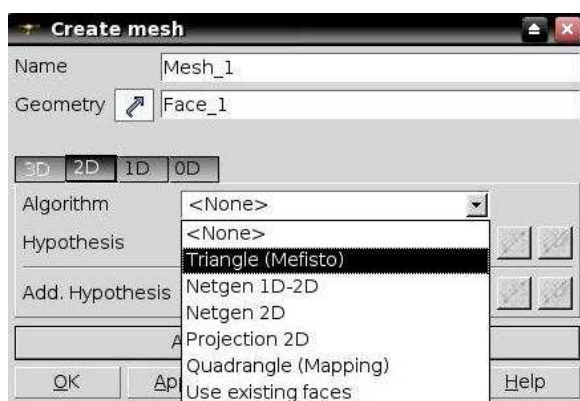
Obr. 6b Rozdělené hrany znázorněné ve stromku

V dalším kroku se z modulu *Geometry* přejde na modul *Mesh*, kde nejprve vybereme záložku *Mesh*. Pak se pomocí příkazu *Create mesh* (obr. 7) zadávají všechny potřebné parametry k vytvoření sítě konečných prvků.

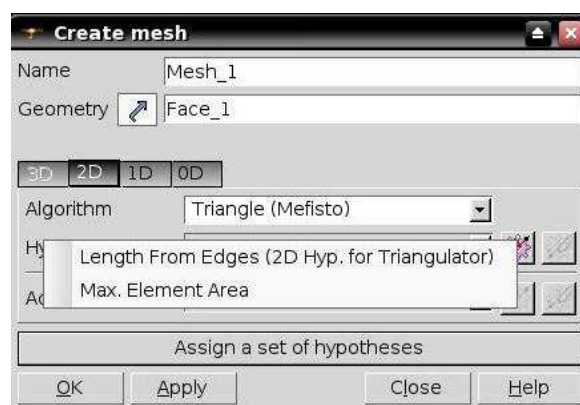


Obr. 7 Nabídka v záložce Mesh

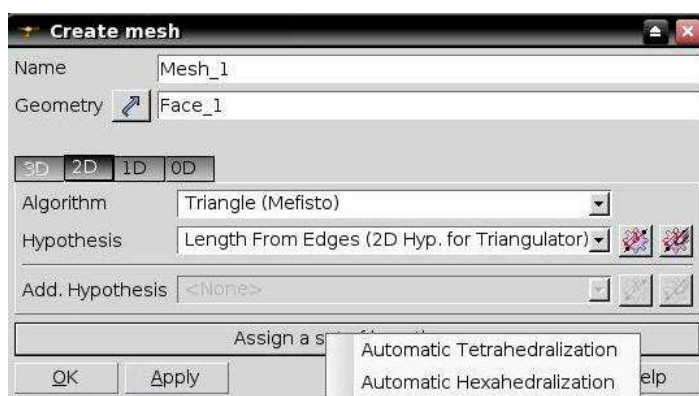
Nejprve se vybere tvar prvku, z kterého bude síť vytvořena. V našem případě se budeme zabývat volbou trojúhelníkového nebo čtyřúhelníkového elementu. Aby bylo možné vytvořit mapovanou síť (pomocí čtyřúhelníkového elementu) je třeba spojit protilehlé přímky (v jednu křivku) naproti oblouku, což se mi bohužel nepodařilo, a proto je použita síť vytvořená pomocí trojúhelníkových elementů – *Triangle (Mefisto)* (obr. 8a). V dalším kroku se zvolí *Length From Edges* (obr. 8b), pomocí něhož nastavíme délku hrany elementu sítě. Následně vybereme *Automatic Tetrahedralization* (obr. 8c) (*Hexahedralization* nemůžeme použít, protože nevytváříme mapovanou síť), pak se nám otevře nové okno, kde zvolíme hustotu sítě (obr. 8d).



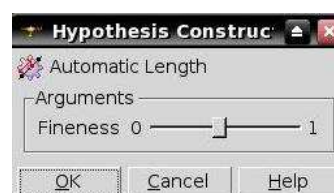
Obr. 8a Výběr tvaru prvku



Obr. 8b Výběr délky hrany elementu sítě

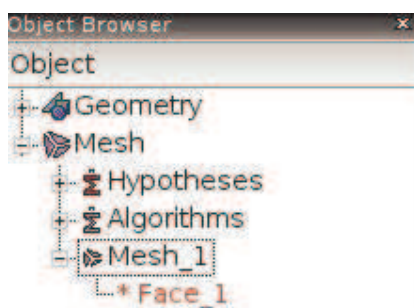


Obr. 8c Spuštění příkazu hustoty sítě



Obr. 8d Zvolení hustoty sítě

Ve stromku rozklikneme položku *Mesh* a poté klikneme pravým tlačítkem myši na *Mesh_1* (obr. 9a). Nyní se nám zobrazí nabídka (obr. 9b), kde se po provedení příkazu *Compute* vygenerovala síť (obr. 9d) a program následně vypsál tabulku (obr. 9c), která obsahuje základní informace o síti, jako např. počet uzlů, elementů apod. Na obr. 9b ještě můžeme, mimo jiné, vidět *Edit Mesh/Sub-mesh*, kde bychom snadno mohli změnit vytvořenou síť.



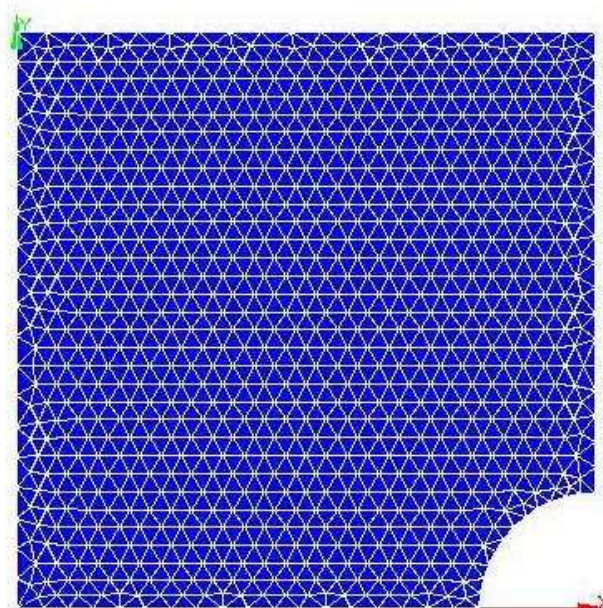
Obr. 9a Mesh_1



Obr. 9b Nabídka

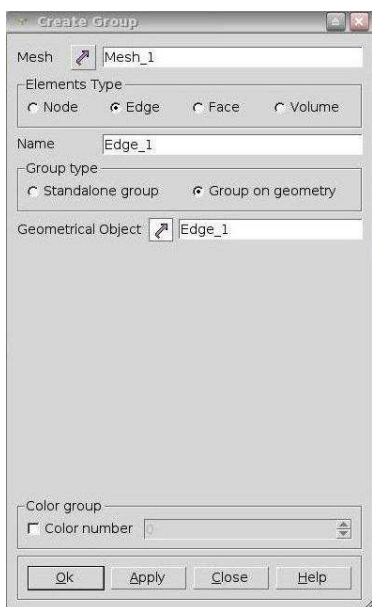
Mesh computation succeed			
Compute mesh			
Name Mesh_1			
Mesh Infos			
	Total	Linear	Quadratic
Nodes :	935		
Edges :	83	83	0
Faces :	1785	1785	0
Triangles :	1785	1785	0
Quadrangles :	0	0	0
Polygons :	0		
Volumes :	0	0	0
Tetrahedrons :	0	0	0
Hexahedrons :	0	0	0
Pyramids :	0	0	0
Prisms :	0	0	0
Polyhedrons :	0		

Obr. 9c Vypsaná tabulka

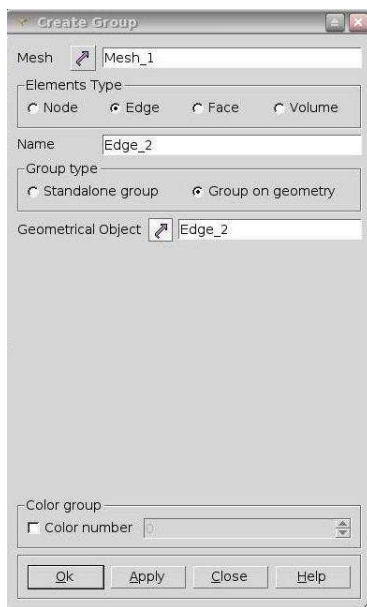


Obr. 9d Vygenerovaná síť

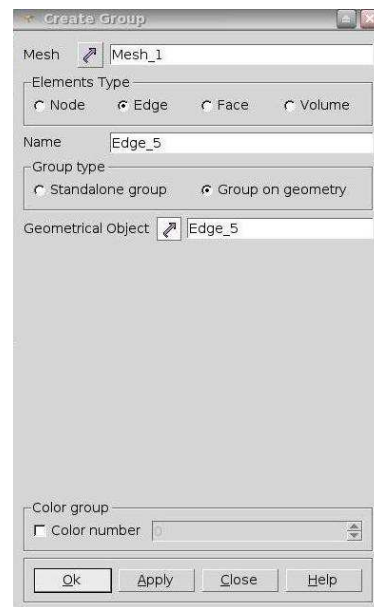
Na *obr. 9b* je také vidět příkaz *Create Group*, který použijeme k vytvoření geometrických a silových okrajových podmínek pro řešič Aster. *Group* musíme vytvořit pro každou hranu zvlášť, protože geometrické a silové okrajové podmínky jsou aplikované na tři různé hrany.



Obr. 10a Vytvoření první *group*

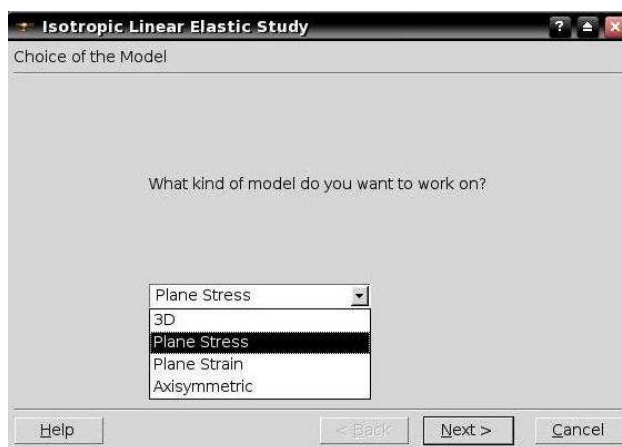


Obr. 10b Vytvoření druhé *group*



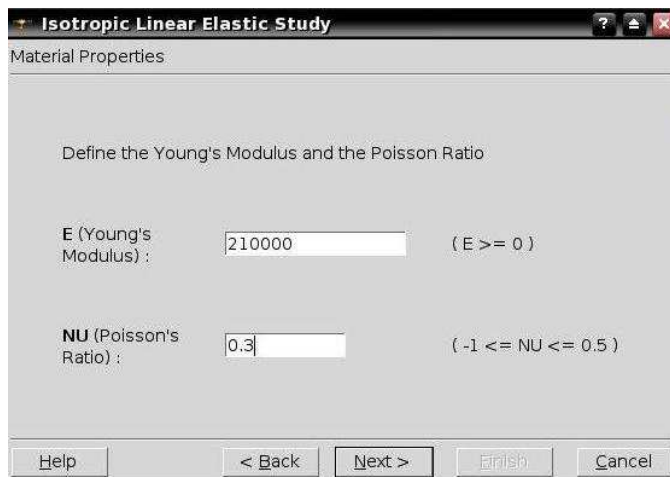
Obr. 10c Vytvoření třetí *group*

Dalším krokem je spuštění *Aster* modulu, kde se nejprve zvolí typ úlohy (lineární elasticita) a pak se vybere druh analýzy. V nabídce máme možnost zvolit tyto analýzy: 3D, rovinná napjatost, rovinné přetvoření, axisymetrie, jak je vidět na *obr. 11*. V našem případě jde o rovinnou napjatost (vyplývá z rozboru z 3.2.), proto vybereme *Plane Stress*.



Obr. 11 Výběr analýzy

Dále máme možnost vybrat si v jakém druhu objektu bude Aster pracovat, máme na výběr buď geometrii a nebo síť, my volíme síť. Poté musíme definovat modul pružnosti v tahu a Poissonovo číslo, které je patrné ze zadání z 3.1.



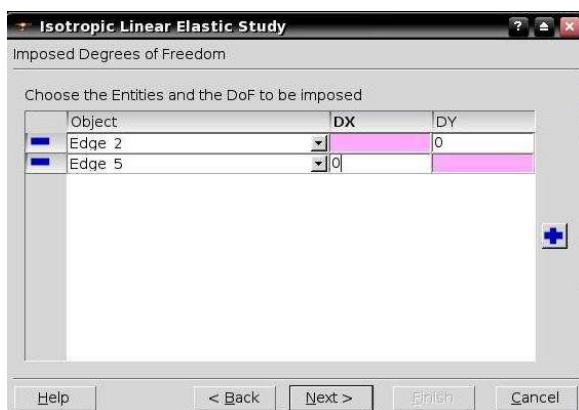
Obr.12 Modul pružnosti v tahu a Poissonovo číslo

Následuje zadání geometrických okrajových podmínek, kde jsou pomocí *obr. 6a* aplikovány:

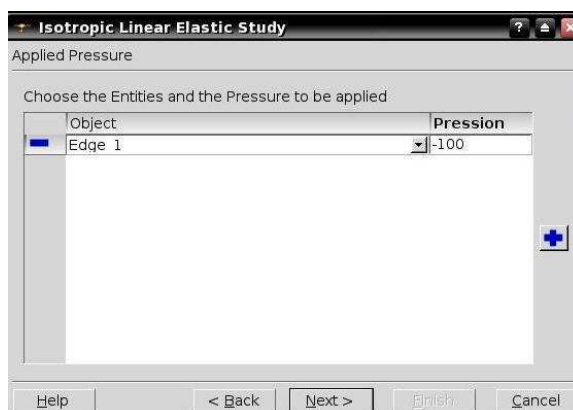
- nulový posuv v ose y, který se zadá na „edge_2“ (*obr. 13a*)
- nulový posuv v ose x, který se zadá na „edge_5“ (*obr. 13a*)

A také musíme podle *obr. 6a* aplikovat silovou okrajovou podmínku:

- ta se zadá na „edge_1“ (*obr. 13b*)



Obr. 13a Zadávání geometrických okrajových podmínek



Obr. 13b Zadávání silové okrajové podmínky

V posledním kroku se pomocí příkazu *Finish* spustí řešič. Na *obr. 14* je vidět okno, které se ukáže uživateli během řešení úlohy, kde se po vypočtení úlohy toto okno automaticky zavře.

```

copying .../as/analyza.comm [ OK ]
copying .../as/analyza.mail.med [ OK ]
<INFO> Parameters : memory 128 MB - time limit 120 s

-----
Content of /tmp/interactif,31683 before execution

total 6916
drwxrwxrwx 5 caelinux caelinux 220 Mar 6 13:36 .
drwxrwxrwt 30 root root 780 Mar 6 13:36 ..
-rw-rw-rw- 1 caelinux caelinux 931 Mar 6 13:36 31683.export
-rw-rw-rw- 1 caelinux caelinux 48 Mar 6 13:36 31683.para
lrwxrwxrwx 1 caelinux caelinux 53 Mar 6 13:36 asteru -> /union/opt/SALOME-MECA-2008
,1-GPL/aster/STA9,2/asteru
-rw-rw-rw- 1 caelinux caelinux 6963204 Mar 6 13:36 elem,1
-rw-rw-rw- 1 caelinux caelinux 1292 Mar 6 13:36 fort,1,1
-rw-rw-rw- 1 caelinux caelinux 78699 Mar 6 13:36 fort,20
drwxrwxrwx 17 caelinux caelinux 340 Mar 6 13:36 Python
drwxrwxrwx 2 caelinux caelinux 40 Mar 6 13:36 REPE_OUT
drwxrwxrwx 2 caelinux caelinux 40 Mar 6 13:36 RESU_ENSIGHT

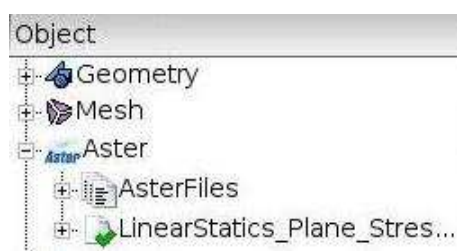
-----
Code_Aster run

<INFO> Command line 1 :
<INFO> ./asteru Python/Execution/E_SUPERV.py -eficas_path ./Python -commandes fort,1 -rep
none -num_job 31683 -mode interactif -ORBInitRef NameService=corbaname::localhost:2810 -r
ep_outils /union/opt/SALOME-MECA-2008,1-GPL/aster/outils -rep_mat /union/opt/SALOME-MECA-2
008,1-GPL/aster/STA9,2/materiau -rep_dex /union/opt/SALOME-MECA-2008,1-GPL/aster/STA9,2/da
ta -suivi_batch -memjeveux 32 -tpmax 120 | tee fort,6

```

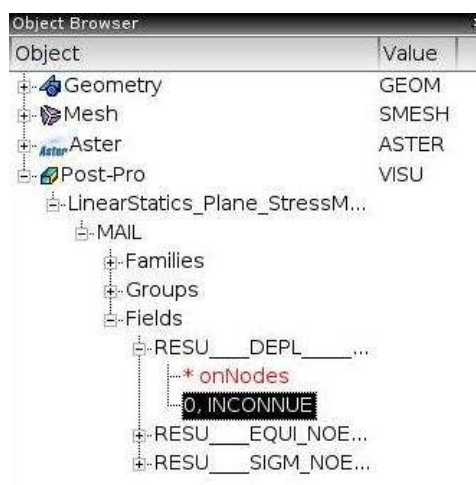
Obr. 14 Okno zobrazující průběh výpočtu

V levé části okna je ve stromku po rozkliknutí modulu Aster vidět zelená fajfka (*obr. 15*), která nám ukazuje, že výpočet proběhl správně a bez problémů. Pokud by fajfka byla oranžová, výpočet neproběhl správně a je třeba změnit např. okrajové podmínky apod.

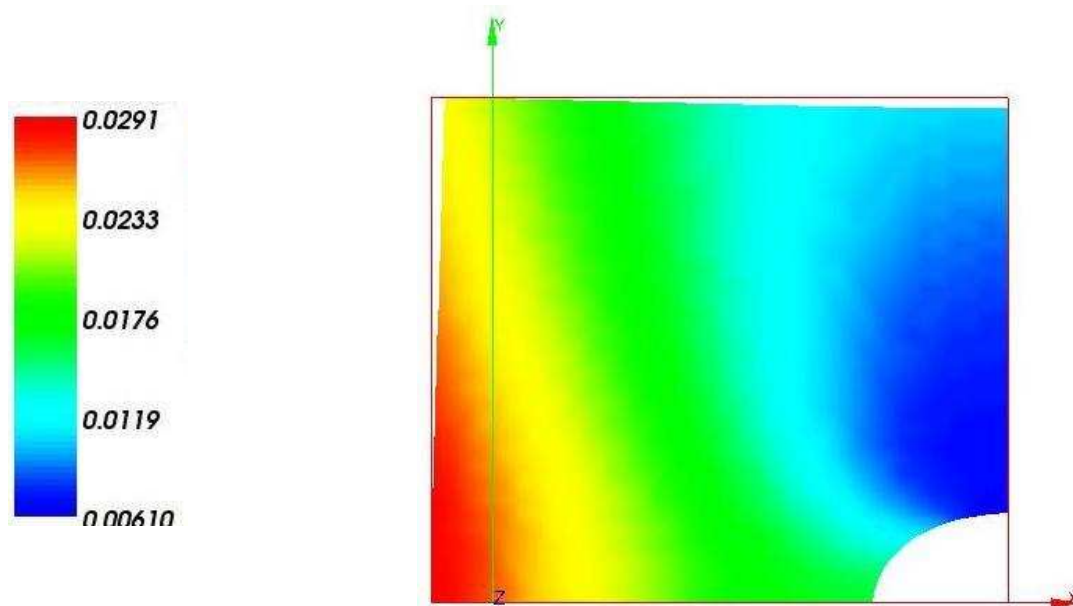


Obr. 15 Potvrzení správného výpočtu ve stromku

Následuje spuštění posledního modulu a to postprocessoru, který nám vykreslí vypočtené výsledky. Pro nás je nejdůležitější vykreslení celkového posuvu, abychom si ověřili správnost zadaných okrajových podmínek. To uděláme tak, že ve stromku postupně rozklikneme položky *Post-pro – Linear Statics Plane Stress – Mail – Fields – Resu depl* a na závěr klikneme pravým tlačítkem myši na položku *0, INCONNUE* (obr. 16) a následně vybereme *Deformed shape*. Na obr. 17 je použito 200 násobné zvětšení, aby celkové posuvy byly jasně viditelné.

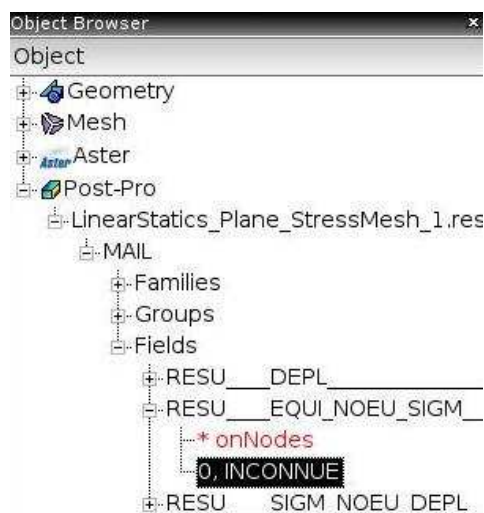


Obr. 16 Položka 0,INCONNUE ve stromku pro vykreslení celkového posuvu

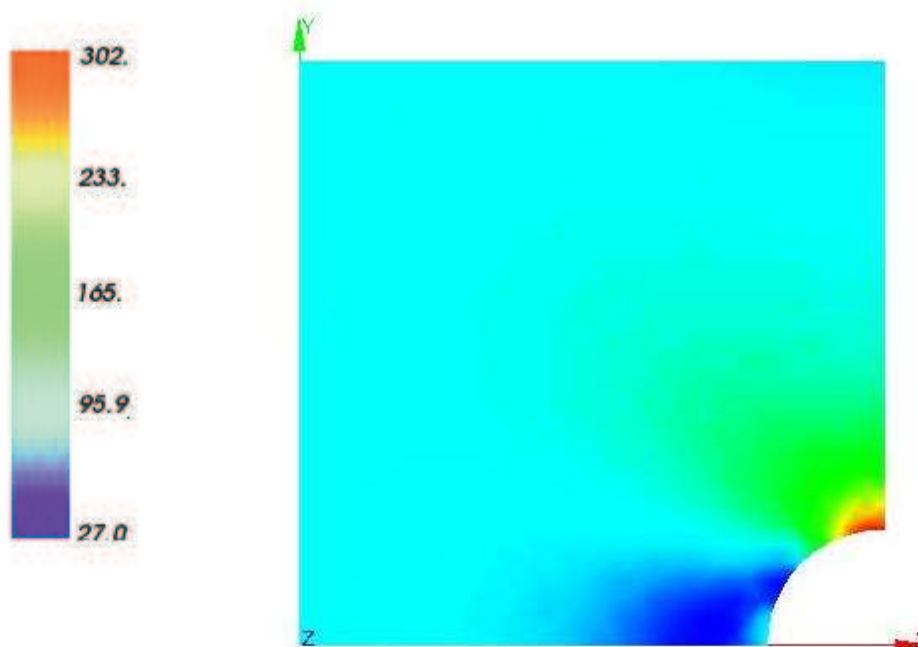


Obr. 17 Vykreslení celkového posuvu v mm

Poté nás zajímá rozložení redukovaného napětí. Postup bude podobný jako u vykreslení celkového posuvu, ale s rozdílem, že ve stromku místo *Resu depl* zvolíme *Resu Equi Noeu Sigm* a na závěr opět klikneme pravým tlačítkem myši na položku *0, INCONNUE* (obr. 18) a následně vybereme *Scalar map*. Otevře se nám nové okno a zde zadáme *VMIS*, což znamená, že nám vykreslí výsledné rozložení redukovaného napětí podle podmínky HMH (Von Misses).



Obr. 18 Položka *0,INCONNUE* ve stromku pro vykreslení rozložení redukovaného napětí



Obr. 19 Vykreslení rozložení redukovaného napětí v MPa

5. PROGRAM ROSHAZ

5.1. Základní popis

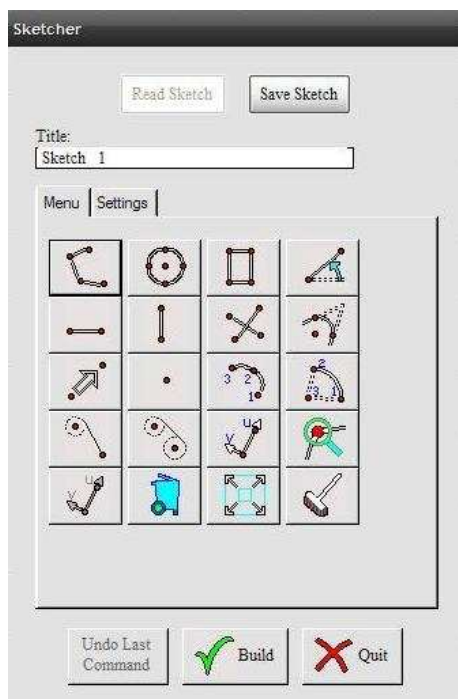
Roshaz je komerční software, který ale také poskytuje bezplatnou demo verzi a díky této verzi je vyřešena rovinná úloha z 3. kapitoly. Roshaz poskytuje dva druhy demo verze. Jednou z nich je plné užití bez žádného omezení v počtu uzlů, ale lze ji užít pouze na jeden měsíc a druhá verze poskytuje řešení úlohy pouze s maximálním počtem 5000 uzlů. V této práci je použita verze s maximálním počtem 5000 uzlů. Tento program byl vytvořen v Anglii v roce 2002 a dokáže řešit úlohy lineární statické strukturální analýzy. Pokud bychom dodatečně nainstalovali bezplatný řešič CalculiX, pak Roshaz s tímto řešičem dokáže řešit také úlohy nelineární statiky, plasticity, frekvencí a termomechaniky. I když se jedná o demo verzi, on-line technická podpora je rychlá a bezplatná. Je zde také možnost výuky formou odborného školení apod.

Výhodou tohoto programu je široký výběr možnosti importu:

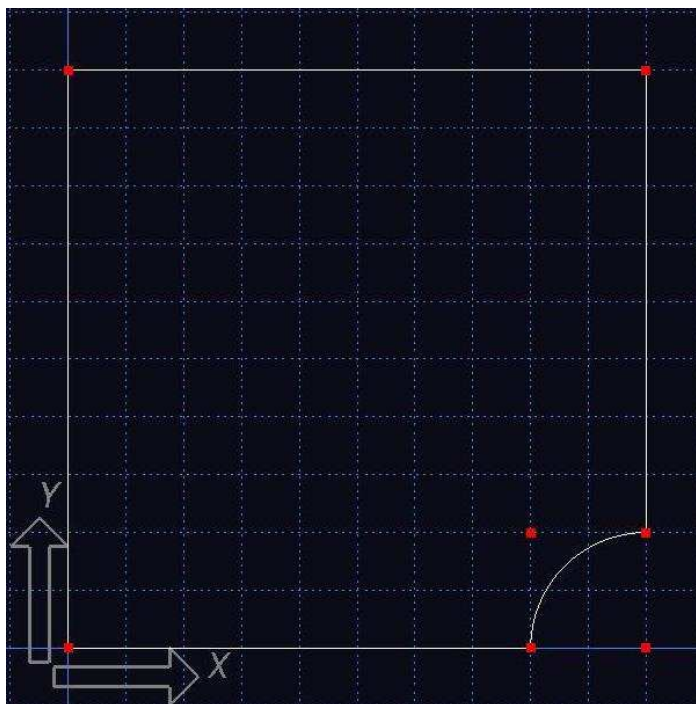
- geometrie - geometrii můžeme vytvořit v AutoCadu a uložit s příponou *.dxf
nebo v Inventoru a použít příponu *.iges nebo lze také využít: *.step,
*.stl, *.vrmf
- sítě – z následujících programů: Nastran, Lusas, Abaqus, Calculix , Cadfix, ...
- výsledků s těmito příponami: *.fil, *.prv, *.out, *.frd, *.nrs, *.fef

5.2. Postup při řešení rovinné úlohy

Po spuštění programu se v prvním kroku v záložce *Geometry* vybere z nabídky *Sketcher* a následně se pomocí jednoduchých příkazů nakreslí geometrie (obr. 20b).



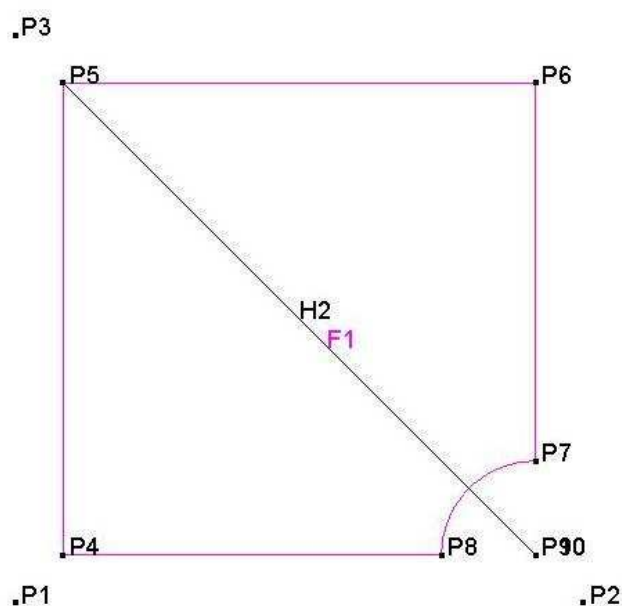
Obr. 20a Sketcher



Obr.20b Geometrie

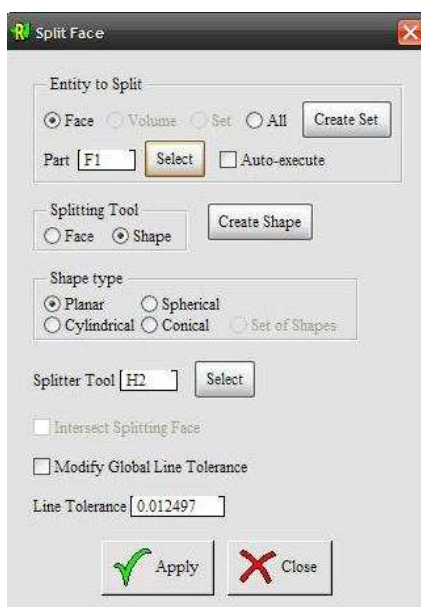
Po vytvoření geometrie (označeno jako Face1 = F1) se zvolí zviditelnění bodů a poté v záložce *Geometry* vybereme položku *Shape* a následně *Plane*, která nám umožní vytvoření roviny (označeno jako H2), která je zadána pomocí dvou zviditelněných bodů (P5 a P9) a poslední bod (P10) musíme vytvořit, ten je dán souřadnicemi [50, 0, 50]. Tuto rovinu použijeme k rozdělení na dvě části u vytvořené geometrie a tím získáme dvě symetrické plochy. Jednotlivá plocha obsahuje čtyři hrany a tím pádem jsme schopni vytvořit mapovanou síť pomocí čtyřúhelníkového elementu. Geometrie s rovinou a viditelnými body je naznačena na obr. 21

Pozn. Na obr. 21 bod P9 není dostatečně vidět z toho důvodu, protože jeho souřadnice [50, 0, 0] se v rovině x-y shodují se souřadnicemi bodu P10 [50, 0, 50].

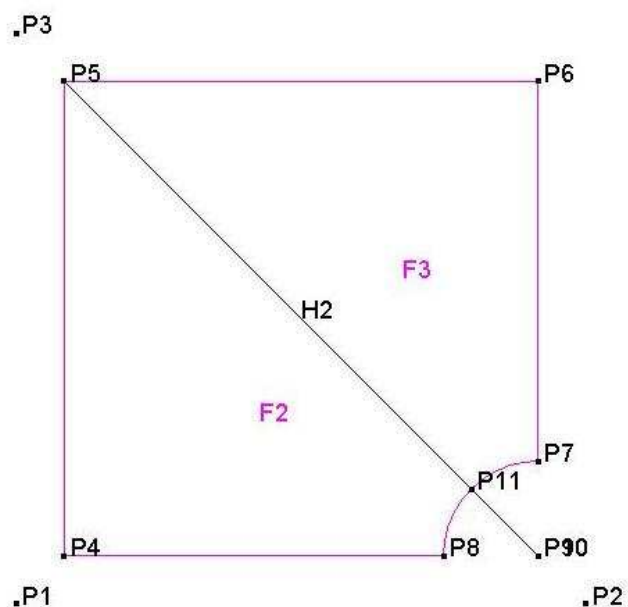


Obr. 21 Geometrie s rovinou a viditelnými body

V dalším kroku se provede rozdělení plochy, pomocí jednoduchého příkazu *Split Face* (obr. 22a), který najdeme v záložce *Tools*. Rozdělení se provede pomocí již vytvořené roviny (H2). Následně nám vzniknou dvě symetrické plochy – F2 a F3 (obr. 22b).



Obr. 22a Příkaz pro rozdělení plochy

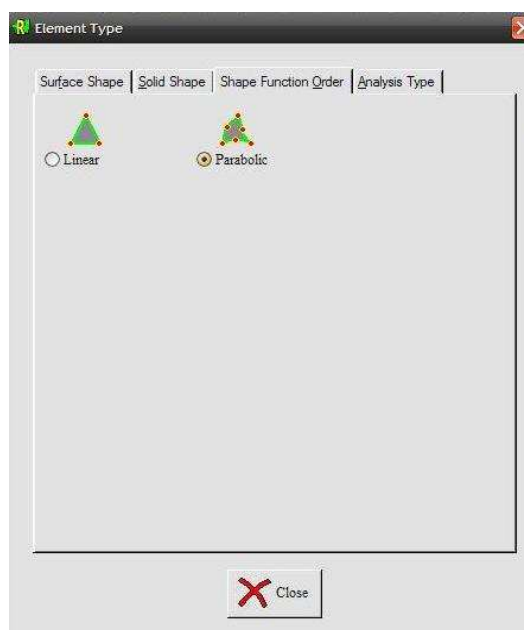


Obr. 22b Rozdělená plocha

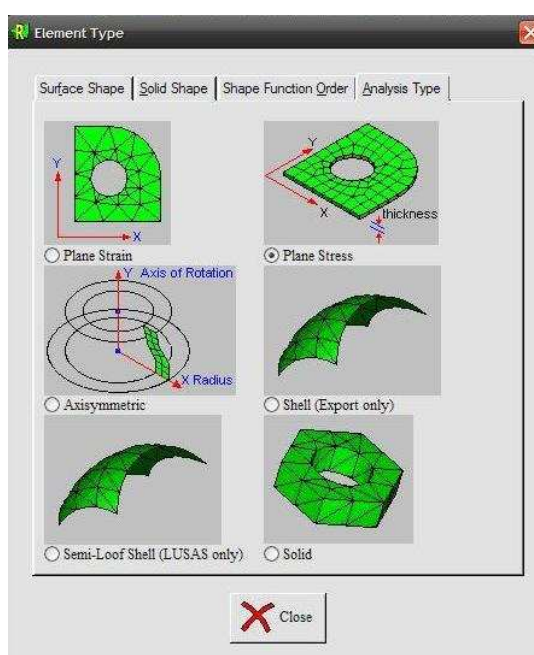
Nyní budeme definovat jednotlivé zadání sítě konečných prvků. Ze záložky *Mesh* vybereme *Shell/Solid Element Type*. Otevře se nám okno (obr. 23a) s výběrem tvaru buď trojúhelníkového nebo čtyřúhelníkového prvku sítě, v našem případě vybíráme čtyřúhelník. Záložku *Solid Shape* vynecháme, protože v této záložce by se vybíral tvar elementu pro prostorovou úlohu. Následuje záložka *Shape Function Order* (obr. 23b), kde vybereme možnost *Parabolic*. V posledním kroku této nabídky máme na výběr několik typů analýz (obr. 23c), my však vybereme *Plane Stress*, protože se jedná o rovinnou úlohu.



Obr. 23a Tvar elementu sítě



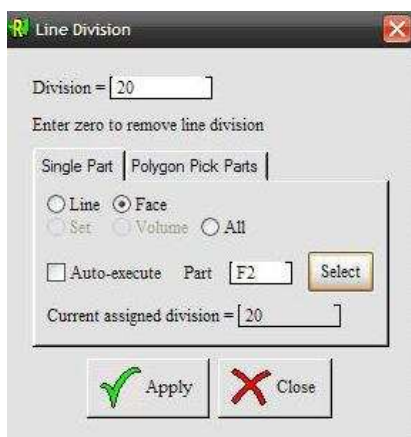
Obr. 23b Tvar funkce



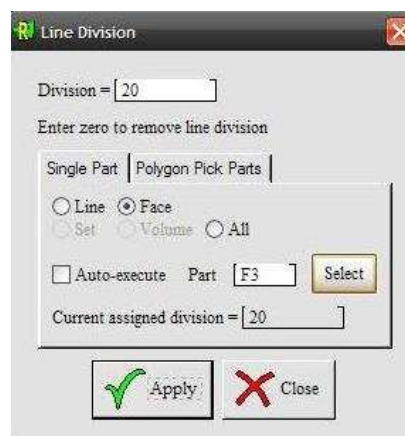
Obr. 23c Typy analýz

V záložce *Mesh* vybereme *Line Division*, pomocí něhož zvolíme na jednotlivé plochy F2 (obr. 24a) a F3 (obr. 24b) rozdělení přímky L8 na 20 stejných úseků.

Pozn. Přímka L8 je spojnice bodů P5 a P11, jak je vidět na obr. 22b.

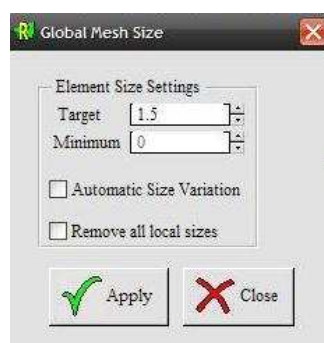


Obr. 24a Rozdělení přímky na ploše F2

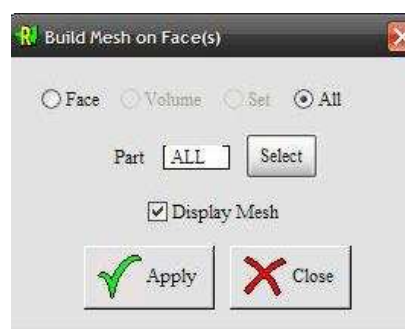


Obr. 24b Rozdělení přímky na ploše F3

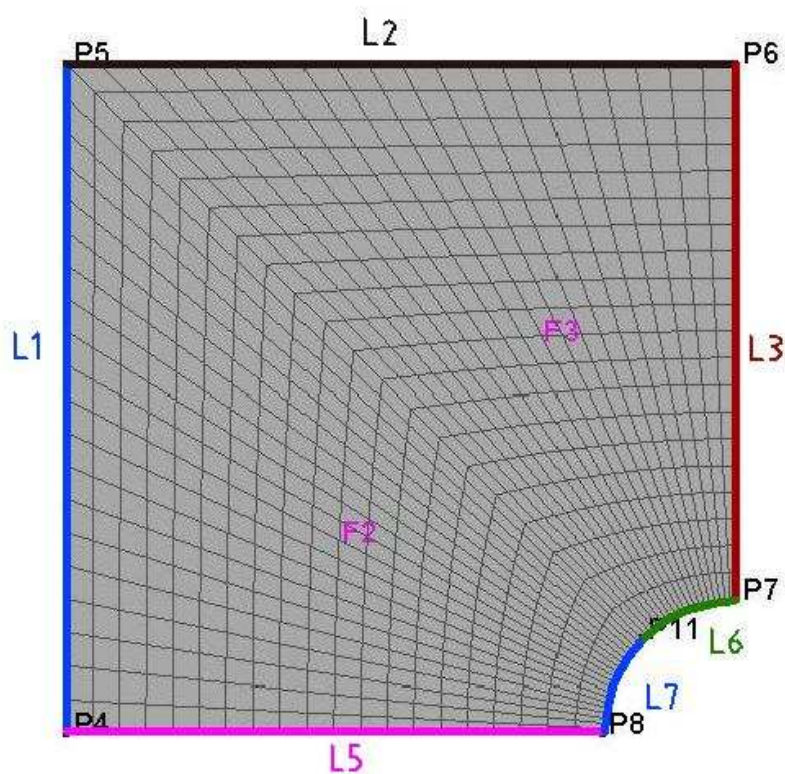
Opět v záložce *Mesh* zvolíme *Size* a pak *Global*, což znamená nastavení velikosti elementu celé sítě (obr. 25a). Na výběr je také *Local*, kde bychom mohli zadat lokální velikost sítě. Nyní máme zadáno vše potřebné k vytvoření sítě, proto na závěr v záložce *Mesh* zvolíme *Build* a pak *Surface*, v této nabídce vybereme *All* (obr. 25b), což znamená, že nám vytvoří síť ve všech plochách (F2 a F3). Vytvořená síť je vidět na obr. 26, kde jsou dodatečně barevně doplněny názvy jednotlivých „lines“.



Obr. 25a Nastavení velikosti sítě

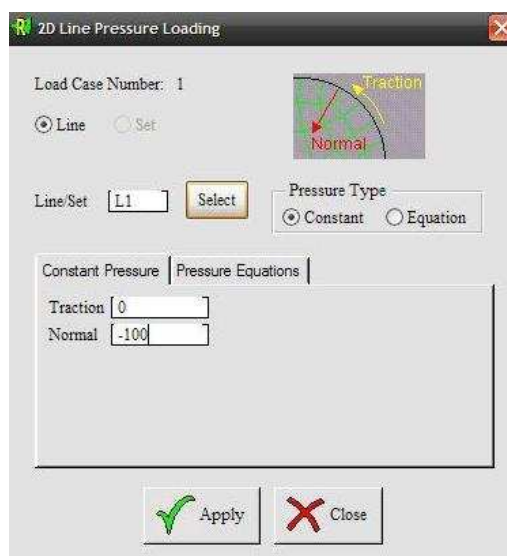


Obr. 25b Vybrání ploch



Obr. 26 Vygenerovaná mapovaná síť

Nyní přejdeme do záložky *Attributes*, kde vybereme položku *Loading* a následně *Case Title*, kde zadáme název silové okrajové podmínky. Bez tohoto názvu nemůžeme zadat jednotlivé zatížení, proto je nutné ho nějakým způsobem pojmenovat. V našem případě jsem zadal název „napeti 100MPa“. V záložce *Attributes* zvolíme opět *Loading* a *Line pressure 2D*. V kolonce *Normal* vyplníme hodnotu -100, což znamená tahové napětí o velikosti 100MPa. Minus je uvedeno z toho důvodu, že je to tahové napětí a směr napětí míří z tělesa ven. Na závěr vybereme přímku, na kterou bude působit tahové napětí, což v našem případě je přímka L1.



Obr. 27 Zadání silové okrajové podmínky

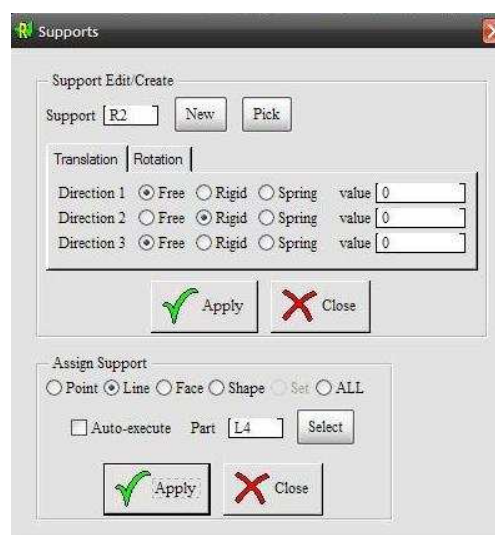
V dalším kroku budeme definovat geometrické okrajové podmínky. V záložce *Attributes* vybereme položku *Supports*. První okrajová podmínka bude představovat nulový posuv v ose x, proto zaškrtneme *Rigid* v *Direction 1*. *Direction 2* a *Direction 3* nejsou nijak omezeny, proto necháme *Free*. Po stisknutí *Apply* (nahore) můžeme vybrat místo, kde bude podmínka zadána. My zvolíme *Line* a následně vybereme L3, jak je zřejmé z *obr. 28a*. Poté znovu potvrdíme *Apply*. Druhá okrajová podmínka bude představovat nulový posuv v ose y, proto zaškrtneme *Rigid* v *Direction 2*. *Direction 1* a *Direction 3* nejsou nijak omezeny, proto znovu necháme *Free*. Opět nahore stiskneme *Apply* a tentokrát vybereme L4 (*obr. 28b*). A znovu potvrdíme *Apply*.

Pozn. Na L3 se později bude ukazovat „RFF“, což znamená „Rigid“(pevný) v ose x a „Free“(volný) v osách y a z.

Obdobně to bude vypadat na L4, ale s tím rozdílem, že bude napsáno „FRF“

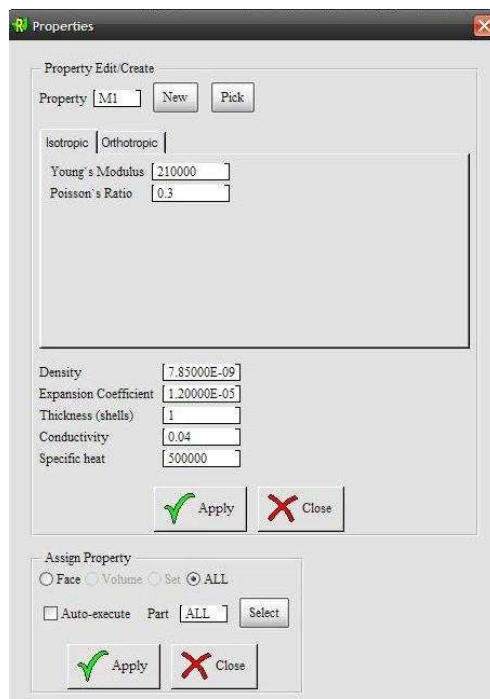


Obr. 28a První geometrická okrajová podmínka



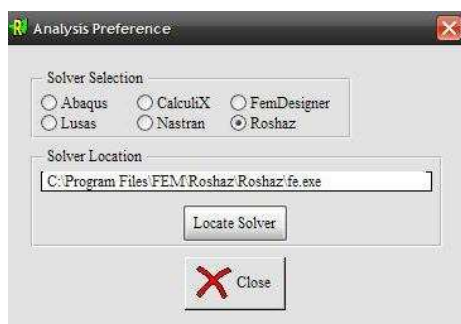
Obr. 28b Druhá geometrická okrajová podmínka

Před spuštěním řešiče musíme ještě definovat materiálové vlastnosti, které bude mít daná součást. V záložce *Attributes* vybereme položku *Properties*. Dle zadání z 3.1. napíšeme jednotlivé hodnoty a vybereme *Apply* a pak ještě zvolíme *All*, abychom materiálové vlastnosti zadali na F2 i F3, poté znovu *Apply*.

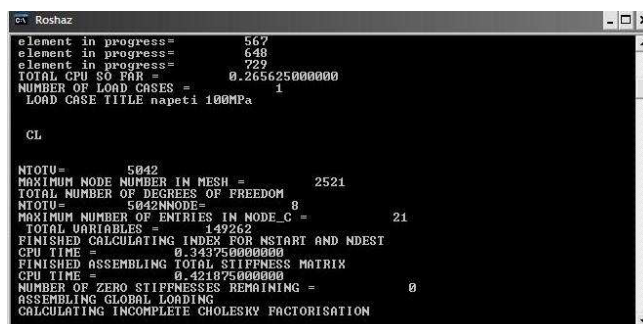


Obr. 29 Definování modulu pružnosti v tahu a Poissonovo čísla

Nyní můžeme spustit řešič, který nalezneme v záložce *Analysis*, následně *Default Solver*. Objeví se nám okno, které je na obr. 30a. Zde si můžeme vybrat pomocí kterého řešiče budeme úlohu počítat. Po výběru řešiče *Roshaz* opět vybereme záložku *Analysis* a nakonec *Submit*, tím se spustí řešič a začne počítat (obr. 30b).

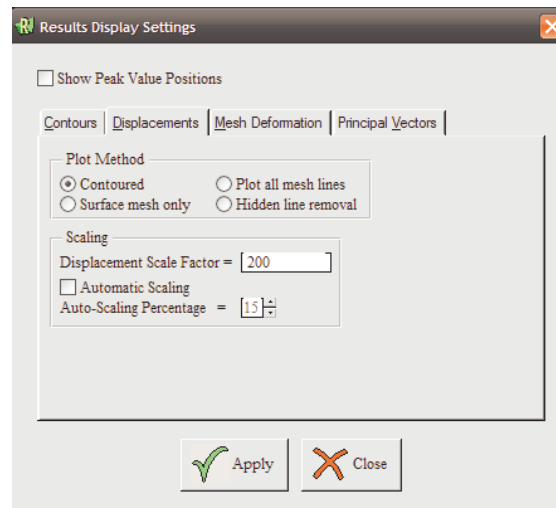


Obr. 30a Výběr řešiče

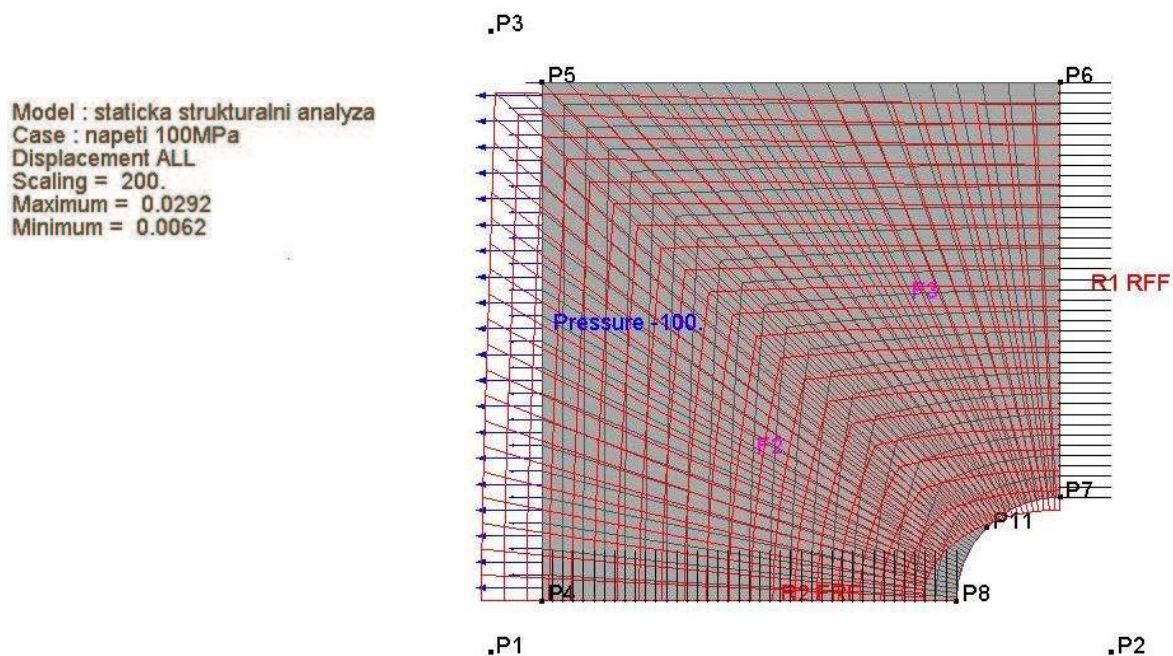


Obr. 30b Okno zobrazující průběh výpočtu

Opět nejprve vykreslíme celkový posuv, abychom si ověřili správnost zadaných okrajových podmínek. V záložce *Results* zvolíme *Display settings*, poté *Displacements* (obr. 31). Aby celkový posuv byl jasně viditelný, vyplníme *Displacement Scale Factor* 200, což znamená, že měřítko pro vykreslení celkových posuvů je 200x zvětšené.

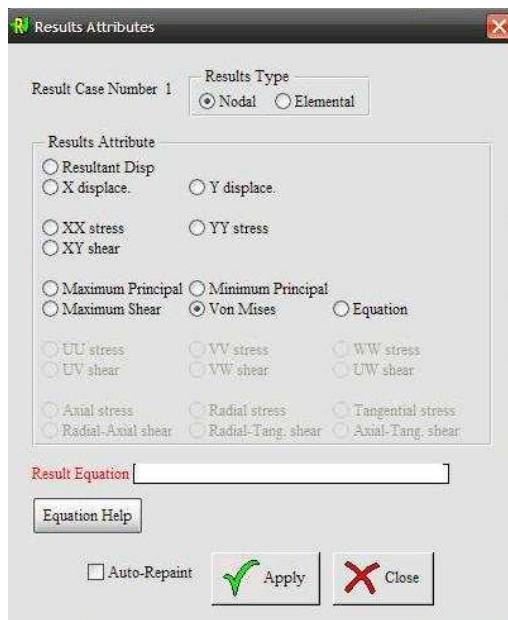


Obr. 31 Nastavení velikosti měřítka celkového posuvu

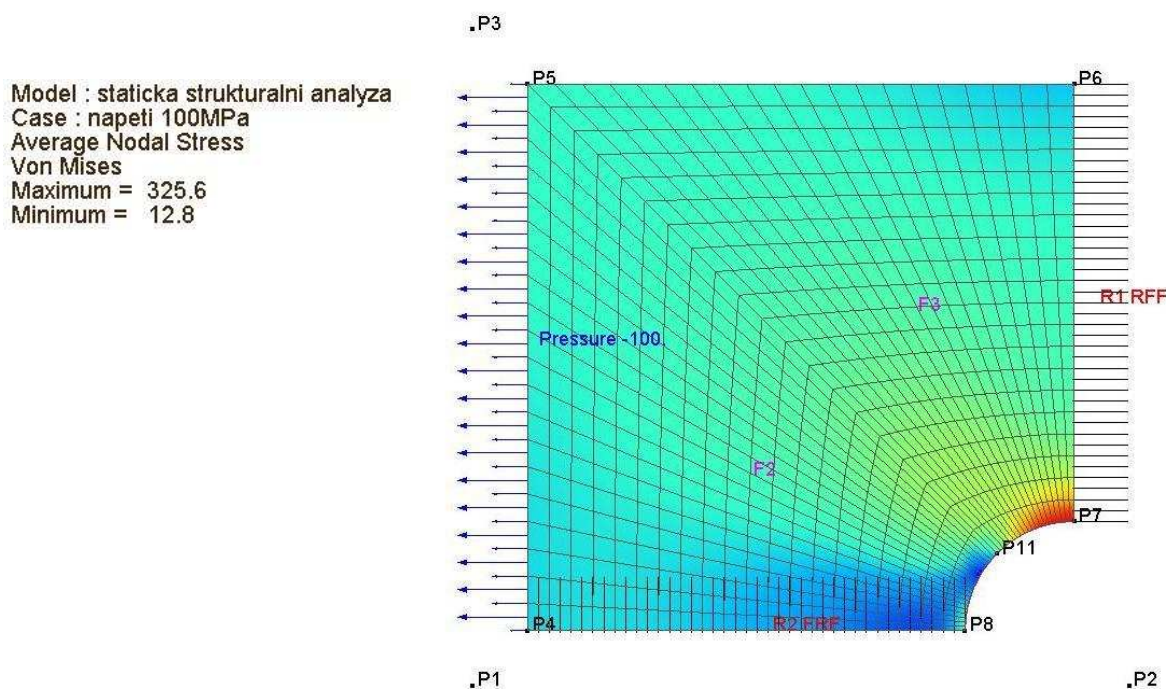


Obr. 32 Vykreslení celkového posuvu v mm

Poté nás zajímá rozložení redukovaného napětí, které zvolíme ze záložky *Results*, pak *Select Attributes* a nakonec vybereme *Von Mises* (obr. 33), což představuje redukované napětí podle podmínky HMM.



Obr. 33 Výběr redukovaného napětí Von Mises



Obr. 34 Vykreslení rozložení redukovaného napětí v MPa

6. ŘEŠENÍ ROVINNÉ ÚLOHY V PROGRAMU ANSYS

6.1. Základní popis

Jako poslední program pro vyřešení jednoduché rovinné úlohy je použit komerční program Ansys. První verze tohoto programu vznikla v roce 1970 v USA. V dnešní době patří mezi nejuniverzálnější a nejvyspělejší MKP výpočetní programy. Z toho ovšem vyplývá nevýhoda a tou je velmi drahá licence tohoto programu. Řádově se pohybuje od stovek tisíc až k miliónu Kč. Jednotlivé ceny se liší podle druhu licence, počtu jader procesorů, druhu řešení fyzikální úlohy apod. Ansys poskytuje dva druhy licence produktu: komerční a akademickou. Komerční licence produktu je určena k vytváření analýz, které slouží k dosažení zisku firmy. Naopak akademická licence produktu (levnější než komerční) slouží k výuce na středních a vysokých školách a k výzkumu. V této práci je použita akademická licence Ansysu (verze 10), přestože existují novější verze. Pracovní prostředí je podobné programu Salome a budou zde lépe vidět rozdílné postupy při provádění deformačně napěťové analýzy v těchto programech.

Ansys dokáže řešit úlohy statické strukturální (lineární, nelineární), dynamické, elektromagnetismus, dynamiku kapalin, akustiku, lomovou mechaniku apod. Dále dokáže import geometrie z několika CAD formátů, jako např. iges, Catia, Pro/engineer, Para, Heal, SAT atd.

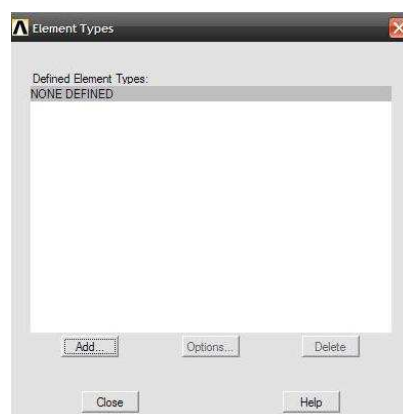
6.2. Postup při řešení rovinné úlohy

V prvním kroku spustíme program Ansys. Následně se nám otevře okno klasického prostředí a nyní můžeme začít vytvářet statickou strukturální analýzu.

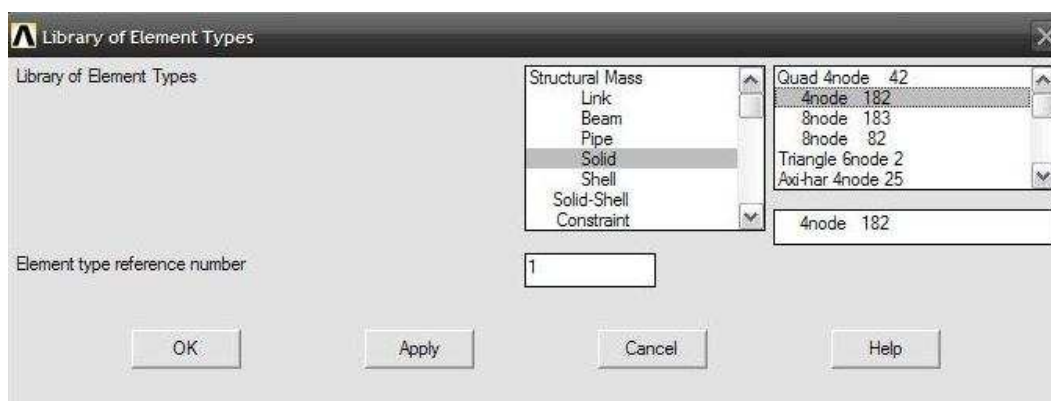
V levé části máme hlavní menu (*obr. 35a*), kde nejprve rozklikneme položku *Preprocessor*. V prvním kroku musíme definovat typ prvku pro přípravu vstupních dat, proto zvolíme *Element Type*. Poté stiskneme *Add/Edit/Delete*. Otevře se nám seznam (*obr. 35b*) načtených typů prvků, protože zde nemáme nadefinovány žádné prvky, stiskneme proto *Add*. V následujícím okně vybereme *Structural Mass*, pak *Solid* a nakonec *4node 182* (*obr. 35c*), protože chceme mít prvky čtyřúhelníkového tvaru.



Obr. 35a Hlavní menu

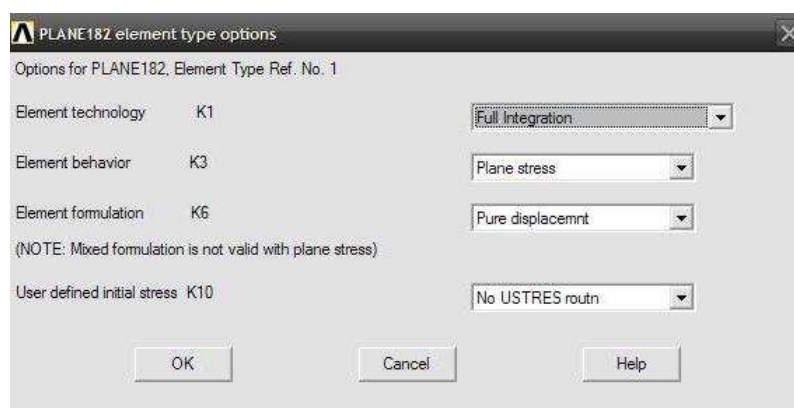


Obr. 35b Seznam načtených prvků



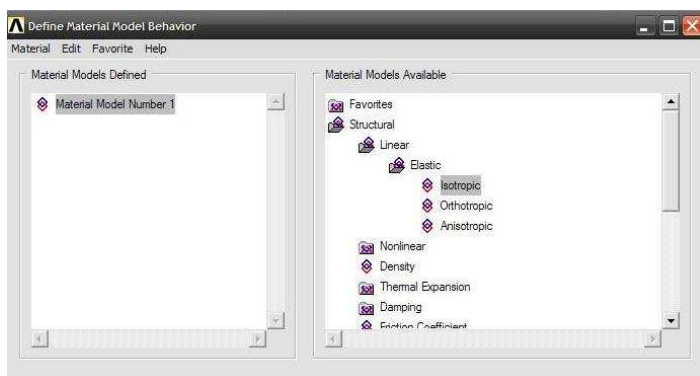
Obr. 35c Výběr elementu

Poté se vrátíme do položky *Element type*, kde vybereme *Options* a zde můžeme nastavit, zda chceme řešit rovinnou napjatost, rovinnou deformaci apod.

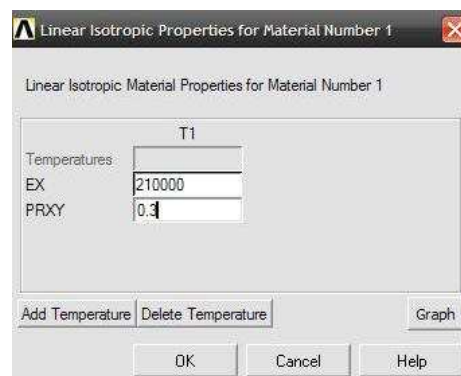


Obr. 36 Výběr rovinné napjatosti

Nyní vybereme model materiálu, proto v *Preprocessoru* zvolíme *Material Props*, následně *Material Models*. Potom vybereme *Structural – Linear – Isotropic*, jak je vidět na *obr. 37a*. Po rozkliknutí poslední položky *Isotropic* se nám objeví okno (*obr. 37b*), kde nejprve definujeme modul pružnosti v tahu a pak Poissonovu konstantu, jak plyne ze zadání z 3.1.



Obr. 37a Výběr materiálového modelu

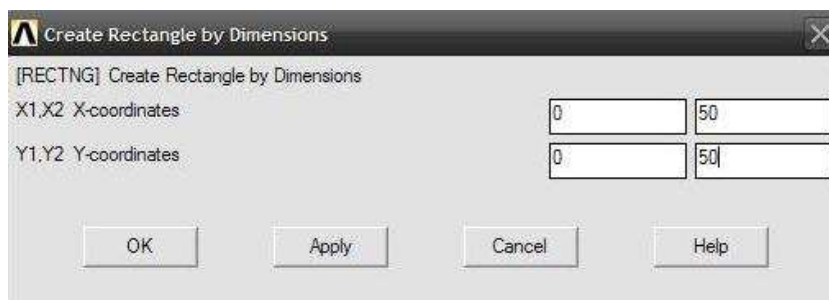


Obr. 37b Definování modulu pružnosti v tahu a Poissonovo čísla

Nyní můžeme přejít k vytváření vlastní geometrie modelu, proto v záložce *Preprocessor* vybereme *Modeling – Create – Areas – Rectangle – By Dimensions* (*obr. 38a*), protože chceme vytvořit čtvercovou plochu, jak plyne ze zadání z 3.1. Na *obr.38b* jsou vidět souřadnice bodů v osách *x* a *y*.

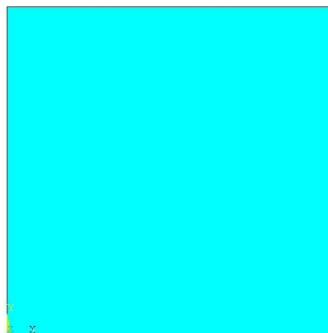


Obr. 38a Příkaz pro vytvoření obdélníkové plochy



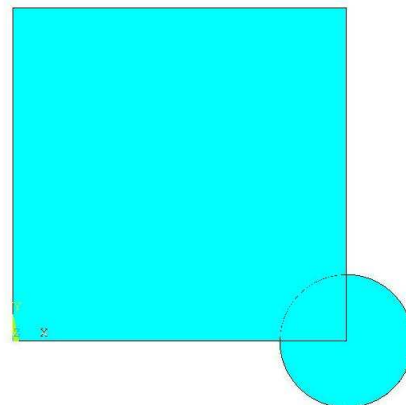
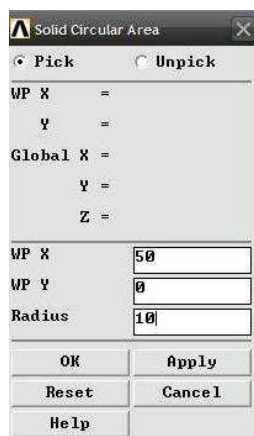
Obr. 38b Souřadnice bodů

Po stisknutí tlačítka *OK* se nám vytvoří čtvercová plocha z výše zadaných souřadnic.



Obr. 39 Vytvořená čtvercová plocha

V dalším kroku vytvoříme kruhovou plochu v pravém dolním rohu. Postup je stejný jako při vytváření čtvercové plochy s tím rozdílem, že místo *Rectangle* zvolíme *Circle* a pak *Solid Circle* (obr. 40a). Ukáže se nám nové okno, kde zadáme souřadnice a poloměr kruhové plochy (obr. 40b). Na obr. 40c je vidět čtvercová a kruhová plocha.

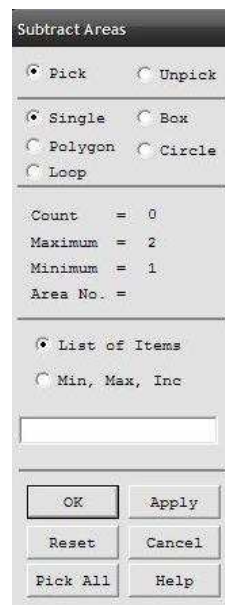


Obr. 40a Příkaz pro vytvoření kruhové plochy

Obr. 40b Souřadnice kruhové plochy

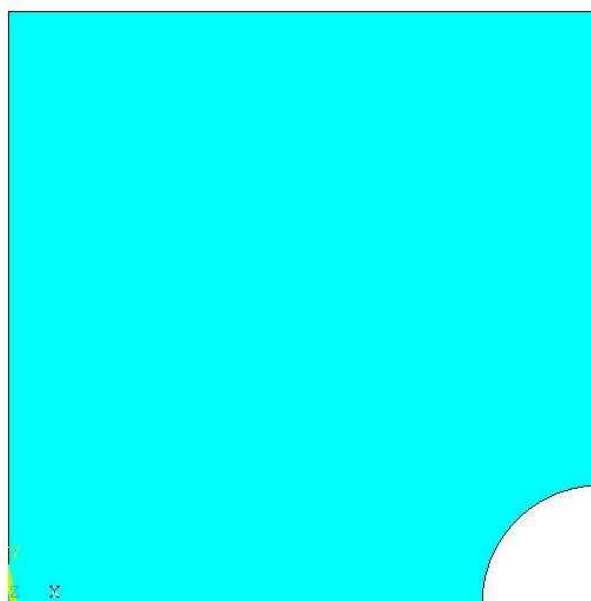
Obr. 40c Vytvořená čtvercová a kruhová plocha

Nyní potřebujeme kruhovou plochu odebrat od čtvercové plochy, abychom v pravém dolním rohu měli $\frac{1}{4}$ díry. To provedeme pomocí „Booleanských“ operací a to následovně: v záložce *Modeling* vybereme *Operate – Booleans – Subtract – Areas* (obr. 41a). Otevře se nám nové okno (obr. 41b), kde nejprve vybereme plochu (čtvercová), ze které budeme materiál odebírat, stiskneme *OK* a pak vybereme odebíranou plochu (kruhová) a pak opět stiskneme *OK*. Tímto nám vznikla geometrie, která odpovídá obr. 41c.



Obr. 41a Příkaz k vytvoření Booleanské operace

Obr. 41b Odebírání kruhové plochy



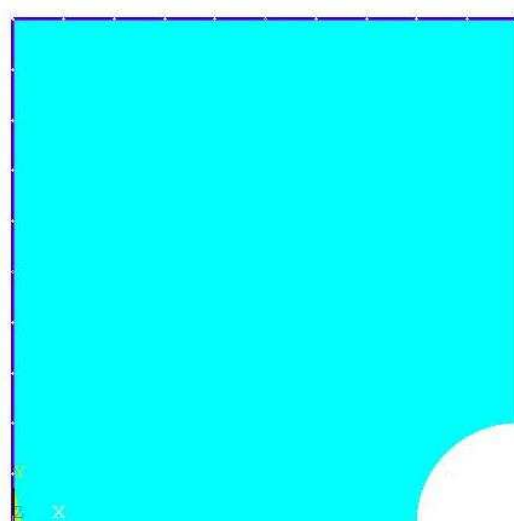
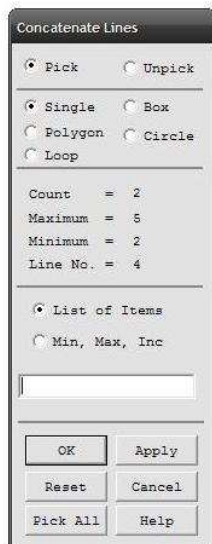
Obr. 41c Výsledná plocha

V dalším kroku si připravíme model tak, abychom mohli vytvořit mapovanou síť, protože máme nadefinovaný čtyřúhelníkový prvek, je třeba spojit přímky na protilehlé straně oblouku, to uděláme následovně: v *Preprocessoru* vybereme *Meshing – Concatenate – Lines* (obr. 42a). Pak se nám objeví další okno, kde vybereme hrany, které chceme spojit (obr. 42b). Nakonec stiskneme *OK*.

Pozn. Na obr. 42b jsou vybrané hrany zbarveny modrou barvou



Obr. 42a Příkaz k spojení hran

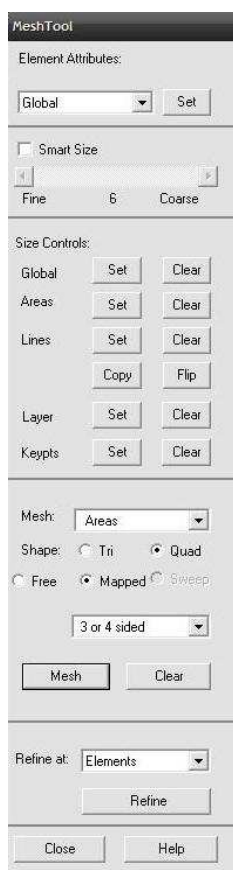


Obr. 42b Výběr hran

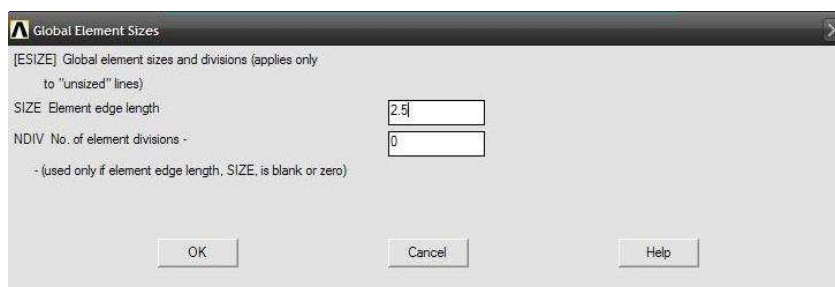
Nyní můžeme přejít k samotnému vytváření sítě konečných prvků. V záložce *Preprocessor* vybereme opět *Meshing* a pak *MeshTool* (obr. 43a). Tím se nám otevře okno, kde nastavíme jednotlivé parametry sítě. Nejprve si zvolíme délku hrany elementu a to uděláme tak, že v otevřeném okně (obr. 43b) u položky *Global* zvolíme *Set*, kde se nám otevře další okno a zde do horního řádku napíšeme 2.5 (obr. 43c), což znamená, že délka hrany jednotlivých prvků bude 2,5mm.



Obr. 43a Příkaz ke spuštění
vytváření sítě

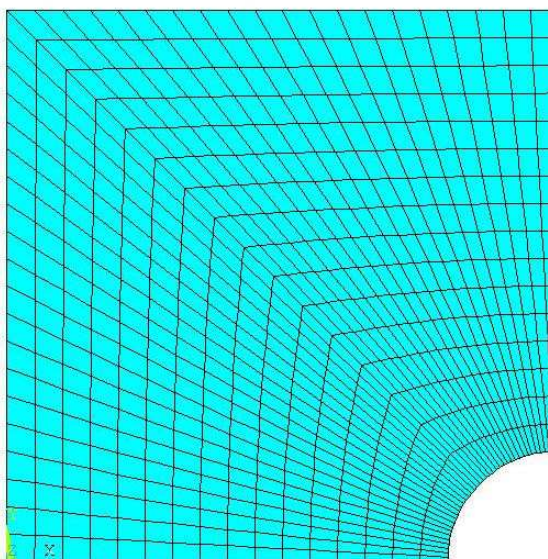


Obr. 43b Okno s parametry pro vytváření sítě



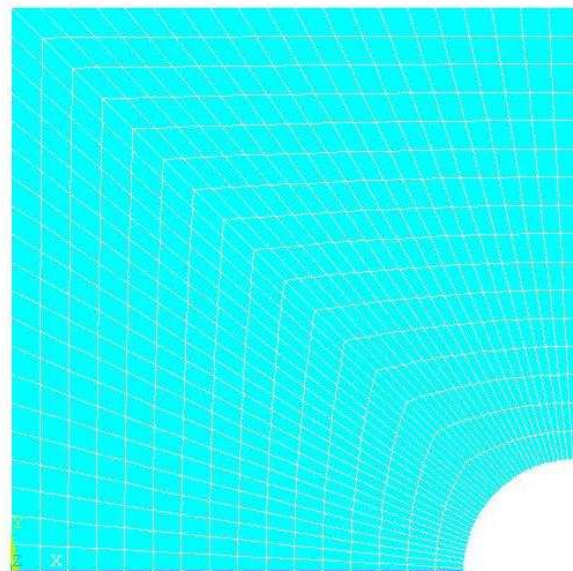
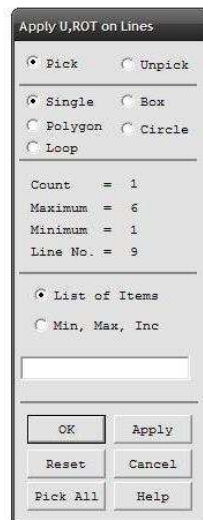
Obr. 43c Nastavení délky hrany elementu sítě

Pak v dolní části okna zaškrtneme *Mapped*, abychom měli mapovanou síť konečných prvků. Poté klikneme na tlačítko *Mesh*, které nám otevře další okno. My pouze klikneme na naší vytvořenou plochu a pak v tomto okně klikneme na tlačítko *OK*. Tím se nám vytvoří mapovaná síť, jak je patrné z *obr. 44*.



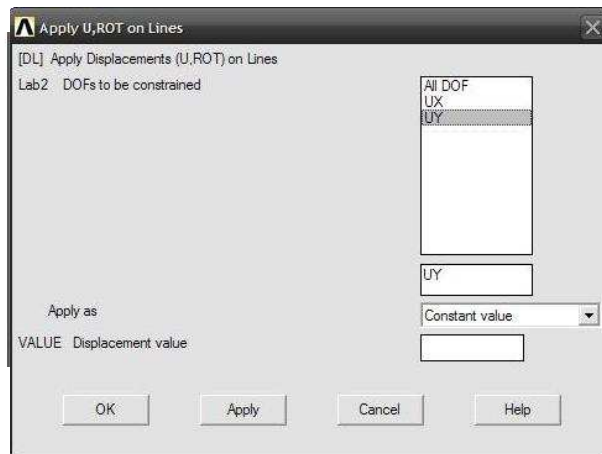
Obr. 44 Vygenerovaná mapovaná síť

Před samotným řešením se ještě musí zadat geometrické a silové okrajové podmínky. Podle obr. 2 geometrické podmínky zadáme následovně: v záložce *Preprocessor* zvolíme *Loads – Define Loads – Apply – Structural – Displacement – On Lines* (obr. 45a). Nyní se nám zobrazí okno a podle obr. 2 určíme první geometrickou podmínku. Vybereme hranu, která leží na ose *x*, kde je tato hrana rozdělena do několika uzlů. V těchto uzlech budeme předepisovat velikost posuvu. (obr. 45b). Po výběru hrany klikneme v okně na tlačítko *OK*. Poté se nám zobrazí nové okno (obr. 45c), zde vybereme *UY*, což znamená posuv v ose *y*. V řádku *Displacement Value* nic nevyplňujeme, protože prázdnému místu Ansys automaticky přiřazuje hodnotu nula.



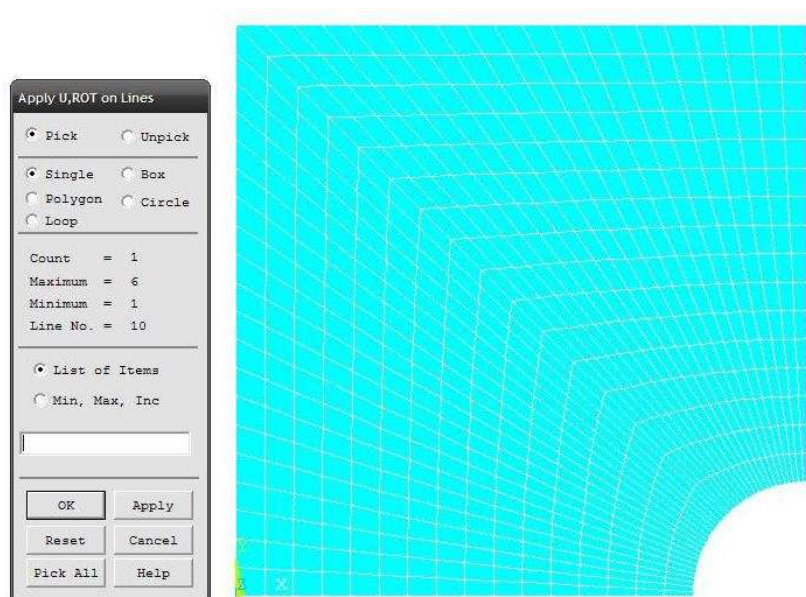
Obr. 45a Příkaz pro vytvoření geometrických okrajových podmínek

Obr. 45b První geometrická okrajová podmínka

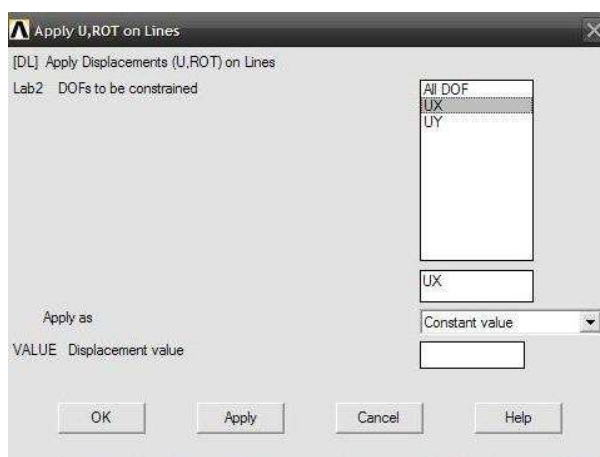


Obr. 45c Výběr nulového posuvu v *y*-vé ose

Postup při vytváření druhé geometrické podmínky je totožný jako u vytváření první geometrické okrajové podmínky, ale jen s rozdílem, že tentokrát vybereme hranu, která leží na ose y , jak je patrné z *obr. 46a*. Poté se nám znovu zobrazí nové okno (*obr. 46b*), zde vybereme UX , což znamená posuv v ose x . V řádku *Displacement Value* opět nic nevyplňujeme, protože posuv bude mít opět nulovou hodnotu.

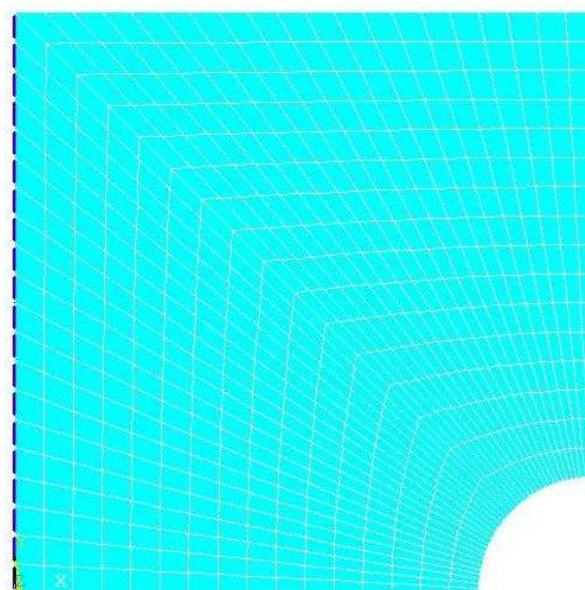
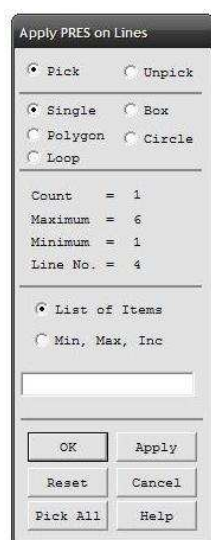


Obr. 46a Druhá geometrická okrajová podmínka



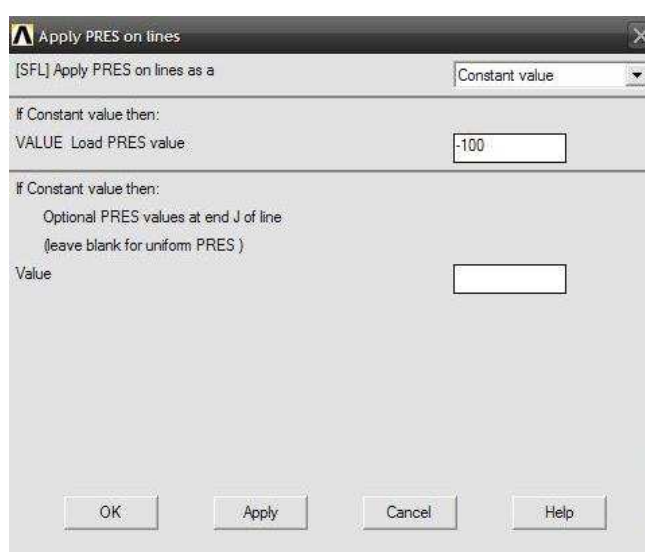
Obr. 46b Výběr nulového posuvu v x -vé ose

Následuje zadání silové okrajové podmínky, která se zadá následovně: v položce *Preprocessor* vybereme *Loads – Define Loads – Apply – Structural – Pressure – On Lines* (obr. 47a). Označíme hranu, která leží na ose *y* (obr. 47b) a klikneme na tlačítko *OK*. Otevře se nám nové okno (obr. 47c), kde do horního rámečku napíšeme hodnotu *-100*, což znamená tahové napětí o hodnotě *100MPa*, opět znaménko mínus je z toho důvodu, protože je to tahové napětí a směr napětí míří z tělesa ven.



Obr. 47a Příkaz pro vytvoření silové okrajové podmínky

Obr. 47b Silová okrajová podmínka



Obr. 47c Vyplnění číselné hodnoty tahového napětí

Nyní je vše potřebné pro výpočet nadefinováno, takže v dalším kroku vybereme v záložce *Solution – Solve – Current LS* a po stisknutí tlačítka *OK* (obr. 48b) se spustí řešič. Pokud výpočet proběhl správně, objeví se po skončení výpočtu nové okno (obr. 48c).



Obr. 48a Příkaz pro spuštění řešiče



Obr. 48b Spuštění řešiče

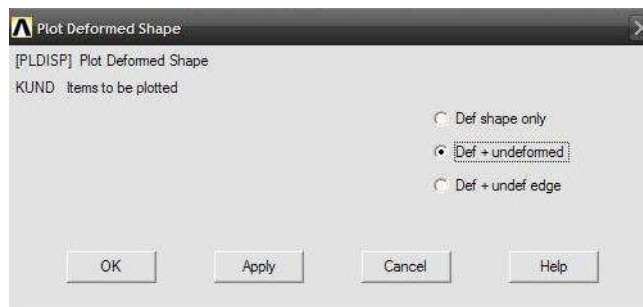


Obr. 48c Okno vyřešené úlohy

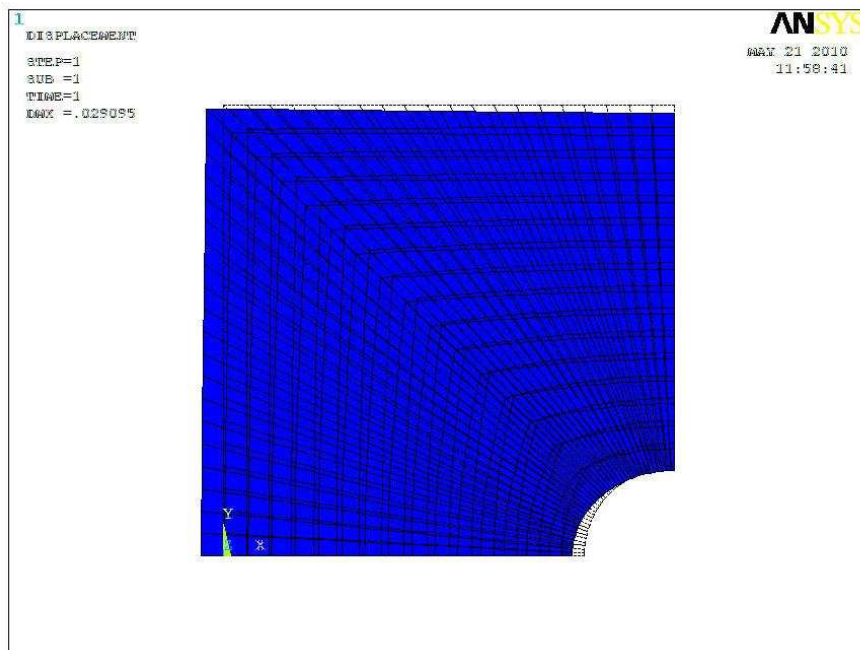
Následuje vykreslení vypočítaných výsledků. A jako v předchozích programech nás nejprve zajímá vykreslení celkových posuvů. V záložce *General Postproc* zvolíme položku *Plot Results – Deformed Shape* (obr. 49a). Po zobrazení nového okna klikneme na *Def + undeformed* (obr. 49b), což nám vykreslí deformovaný a nedeformovaný tvar (obr. 49c).



Obr. 49a Příkaz pro vykreslení celkového posuvu

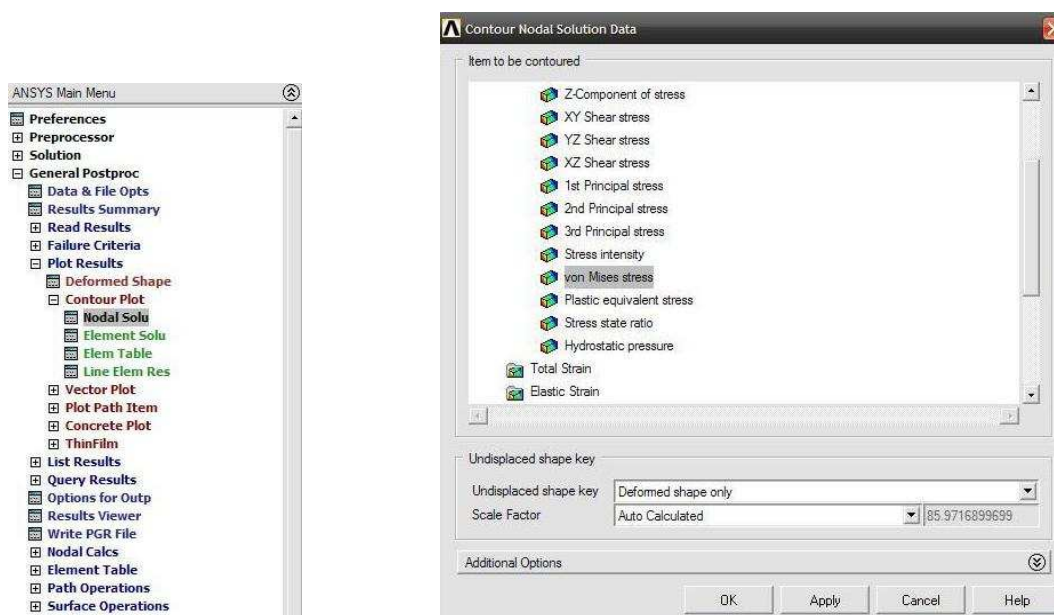


Obr. 49b Výběr pro vykreslení celkového posuvu



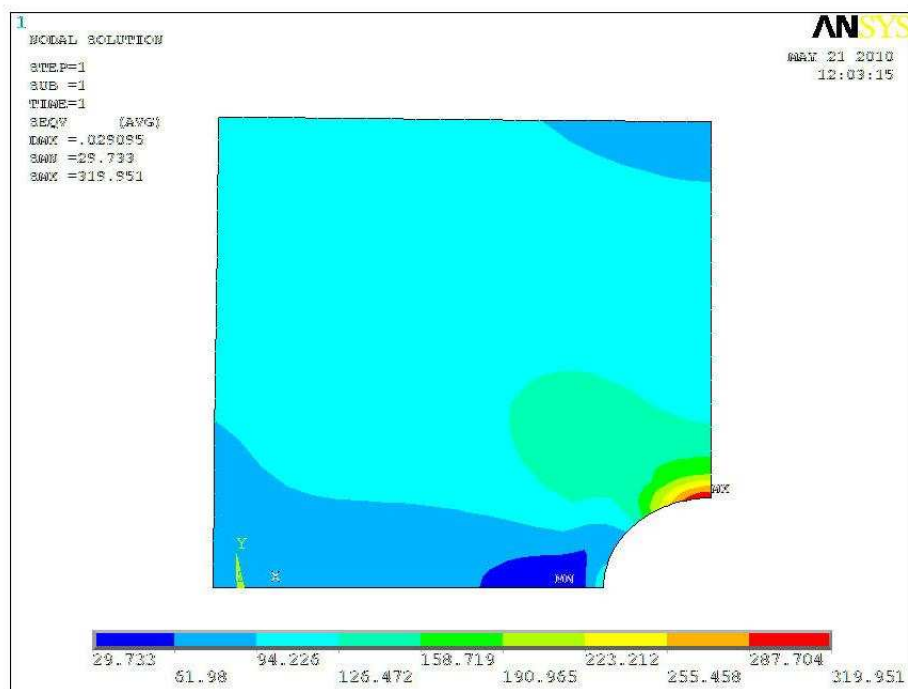
Obr. 49c Vykreslení celkového posuvu v mm

Poté nás opět zajímá rozložení redukovaného napětí. Postup je stejný jako při vykreslení deformovaného tvaru, ale s tím rozdílem, že v položce *Plot results* nyní zvolíme *Nodal Solu* (obr. 50a). Tím se nám otevře nové okno, kde máme na výběr, zda chceme vykreslit různé druhy napětí nebo přetvoření. My si v položce *Nodal Solution* vybereme *Stress* a následně *von Mises stress*, což je opět jako u předchozích příkladů redukované napětí podle podmínky HMH (obr. 50b). Rozložení redukovaného napětí je vidět na obr. 50c.



Obr. 50a Příkaz pro vykreslení napětí

Obr. 50b Výběr redukovaného napětí von Mises



Obr. 50c Vykreslení rozložení redukovaného napětí v MPa

7. SROVNÁVACÍ ANALÝZA

V této práci jsou použity dohromady tři různé programy (Salome, Roshaz, Ansys) pro řešení jednoduché rovinné úlohy. Při vytváření této úlohy jsou jednotlivé postupy rozdílné a v následujících podkapitolách je uvedeno jejich srovnání.

7.1. Srovnání z hlediska zpracování a srozumitelnosti nápovědy

Při zpracování rovinné úlohy bylo nejprve potřeba se seznámit s jednotlivými programy. K dispozici je v každém programu nápověda, která by nám měla ulehčit práci s programem a vyřešit některé problémy, se kterými bychom se mohli setkat.

Salome

Jelikož je tento program stáhnut z internetu, tak při pozornějším prozkoumání internetových stránek můžeme nalézt několik příkladů, ve kterých je stručně popsán postup v některých modulech. Z této internetové nápovědy jsem použil pouze informace týkající se vytváření geometrie a sítě, protože tyto moduly jsou podle mého názoru nejlépe popsány.

V ostatních příkladech je spíše ukázka toho, co v programu můžeme vytvářet a co nám program dokáže vykreslit. Nápověda týkající se řešiče a vykreslení výsledků už není podle mého názoru tak jasná.

Jestliže potřebujeme získat podrobnější informace o programu, pak se nám nabízí možnost použití nápovědy po spuštění programu a to tak, že v horní liště vybereme položku *Help*. Nápověda je rozdělena do jednotlivých částí podle modulů. Obsah této nápovědy je už lépe zpracovaný a není tak stručný jako na internetu, protože jednotlivé informace už nejsou psány jen v bodech.

Pokud by nám nepomohla ani tato nápověda, pak bychom mohli využít poradenství prostřednictvím internetu. Jelikož už je to konkrétní problém a musí s námi komunikovat nějaká osoba, je nutné za tuto službu zaplatit.

Poslední možností je využití výukových kurzů, kterých bychom se mohli zúčastnit. Nevýhodou je ovšem ta skutečnost, že tyto kurzy probíhají pouze ve Francii.

Roshaz

V tomto programu se nám nabízejí dvě možnosti nápovědy. Jednou z možností je, že po spuštění programu vybereme v základním menu položku *Help*. Tato nápověda však

obsahuje pouze informace o příkazech, které můžeme v programu nalézt. Není zde uveden žádný konkrétní příklad, ve kterém bychom mohli pochopit práci v tomto programu. Jelikož je program vybaven grafickým zobrazením jednotlivých příkazů a vše je zcela jasné a srozumitelné, tak tuto nápovědu ani nevyužijeme.

Pokud bychom však měli nějaký konkrétní problém, který bychom nemohli vyřešit, tak se nám nabízí možnost zkontaktovat se (např. pomocí elektronické pošty) s vývojáři tohoto programu. I když se jedná o konkrétní případ, je tato odborná pomoc bezplatná a velice rychlá a vývojáři jsou velice ochotní. V této práci je tak vyřešen problém s konstrukcí mapované sítě.

Poslední možností je využití výukových kurzů probíhajících v Anglii, kde je tento program vyvinut.

Ansys

Vzhledem ke skutečnosti, že je Ansys komerčním programem, je jeho nápověda obsáhlá a kvalitně provedená. Jsou zde uvedeny příklady a postup při vytváření jednoduchých úloh. Vše je graficky doplněno obrázky, což činí nápovědu velice přehlednou.

7.2. Srovnání postupu při vytváření modelu geometrie

Salome

Postup při vytváření geometrie je podle mého názoru celkem složitý, neboť geometrii můžeme vytvořit pouze pomocí přímek, křivek a nebo oblouků. Proto v případě složitější rovinné úlohy nebo u prostorového modelu je vhodnější geometrii importovat z jiného programu, což není problém, pokud vlastníme nějaký komerční modelovací program, jako např. AutoCad nebo Inventor (naopak s modelovacím programem na bázi otevřeného kódu jsem se nesešel).

Roshaz

V tomto programu se geometrie modeluje podstatně lépe. Nabídka, která se objeví po provedení příkazu *Sketcher* obsahuje více možností pro kreslení geometrie. Pomocí grafického zobrazení jednotlivých příkazů (*obr. 20a*) je jasně pochopitelné, co daný příkaz dokáže. Další možností je také využití importu, kde bychom snadno vložili prostorová tělesa.

Ansys

V programu Ansys je vytvoření geometrie také celkem snadné, pod položkou *Modeling* máme výběr z několika příkazů. Oproti programu Roshaz zde není doplněno grafické zobrazení jednotlivých příkazů, což je ovšem vyřešeno v novějších verzích.

7.3. Srovnání postupu při vytváření sítě konečných prvků

Salome

V další části se vytváří síť konečných prvků, kde postup při vytváření sítě pomocí trojúhelníkových elementů je celkem jednoduchý a přehledný. Naopak kdybychom chtěli použít čtyřúhelníkový element pro vytvoření mapované sítě, tak bychom museli spojit naproti oblouku protilehlé hrany v jednu hranu a na to jsem nenašel žádný příkaz. Ani v modulu *Geometry* nešlo nějakým způsobem jednotlivé dvě hrany spojit, což mne trochu zklamalo, protože v jako jediném programu nebyla vytvořena mapovaná síť.

Roshaz

Postup při přípravě pro vytvoření mapované sítě v tomto programu je také vcelku jednoduchý. Spočívá v tom, že plocha se rozdělí na dvě symetrické plochy, kde každá plocha má čtyři vrcholy a tím pádem můžeme použít čtyřúhelníkový element pro vytvoření mapované sítě. Dále samotné vytváření sítě konečných prvků je jasné a přehledné. Opět nám pomáhá grafické zobrazení jednotlivých příkazů, z kterého snadno a rychle vybereme příslušný příkaz.

Ansys

V tomto programu se také síť vytváří velice rychle a jednoduše. Pomocí příkazu *Concatenate* pouze vybereme hrany, které chceme spojit a ty se hned snadno spojí. V našem případě to byly hrany, které ležely naproti oblouku. Při zadávání velikosti sítě máme oproti výše zmíněným programům širší výběr možností. Buď si můžeme vybrat velikost hrany každého elementu a nebo si můžeme vybrat, na kolik částí bude hrana rozdělena. Stejně jako u programu Roshaz můžeme velice snadno vytvořit lokální zjemněnou síť, což se mi v programu Salome opět nepodařilo.

7.4. Srovnání postupu při zadávání okrajových podmínek a řešiče

Salome

Další nejasností v programu Salome je zadávání geometrických a silových okrajových podmínek, protože pod příkazem *Create Group* si asi ne každý člověk dokáže představit definování těchto podmínek. Po spuštění příkazu *Create Group* už snadno můžeme zadat tyto podmínky buď na uzel, hranu, plochu a nebo na objem. Nulové posuvy v příslušných osách a také definování velikosti tahového napětí se provádí až v samotném řešiči Aster. Je tedy vidět, že až po spuštění modulu *Aster* a nadefinování ostatních parametrů máme úplně určeny okrajové podmínky a tím pádem definování těchto podmínek není zadáno hned v jednom příkazu, ale je to takhle rozkouskované. Další záhada se týká v oblasti výpočtu. Pokud výpočet proběhne správně, ukáže se nám zelená fajfka (*obr. 15*), ale pokud by výpočet neproběhl správně, pak je fajfka oranžová. Vytvoří se nám soubor, který obsahuje průběh výpočtu a případné chyby, nevýhodou je ovšem to, že tento text je napsán pouze ve francouzském jazyce. Což je na druhou stranu celkem pochopitelné, protože tento program byl vyvinut ve Francii a jelikož je to Open Source, pak nemůžeme očekávat překlad do ostatních jazyků, jako např. do anglického jazyka.

Roshaz

Při zadávání geometrických a silových okrajových podmínek v programu Roshaz také není problém. Jednotlivé hodnoty posuvů a tahového napětí napíšeme hned do příslušného řádku a tím pádem je pak nemusíme definovat zvlášť jako u programu Salome. Jedinou zvláštností u zadávání silových okrajových podmínek je to, že musíme vyplnit název. Domnívám se, že je to z toho důvodu, aby při samotném vykreslení výsledků bylo v levé části tabulky jasně vidět, jaké tam působí zatížení. U programu Roshaz se vyplatí doinstalovat volně šiřitelný program CalculiX, který nám rozšíří možnosti při řešení z úloh staticky strukturálních také o úlohy nelineární statiky, frekvence a termomechaniky.

Ansys

V posledním programu je zadávání těchto podmínek trochu „schované“. Např. při definování geometrických okrajových podmínek musíme v položce *Preprocessor* rozkliknout *Loads – Define Loads – Apply* a až nyní můžeme vybrat *Displacement*, kde po výběru hrany již snadno definujeme nulové posuvy v příslušných osách. Stejný postup je při zadávání silové okrajové podmínky, místo *Displacement* v našem případě vybereme položku *Pressure*.

Ovšem v novějších verzích už je tento malý nedostatek odstraněn a definování těchto podmínek je již snadné, rychlé a přehledné. Dále pokud se vyskytne problém při výpočtu, ukáže se chybový řádek, ve kterém je napsáno v čem spočívá problém, resp. co musíme změnit nebo nadefinovat.

7.5. Srovnání postupu při vykreslení vypočtených výsledků

Salome

Samotné vykreslení v *postprocessoru* také není na první pohled jasné, protože když bychom použili ikonu nabízející vykreslení těchto výsledků, tak se nám ukáže, že nejsou vybraná žádná data. Proto tedy musíme v stromku v části *preprocessor* rozkliknout do posledního okna 0, *INCONNUE* a pak teprve pravým tlačítkem myši se nám zvolí nabídka jaké výsledky chceme vykreslit. Máme na výběr pro vykreslení různých napětí, jako např. podle podmínky HMH nebo $\max \tau$ atd.

Roshaz

Vykreslení vypočtených výsledků je opět jasné a naprosto přehledné. Nabízí vykreslení celkových posuvů, minimálního nebo maximálního napětí, dále smykové a redukované napětí.

Ansys

Oproti zbývajícím programům Ansys nabízí k vykreslení vypočtených výsledků nejširší možnosti. Dokáže vykreslit všechna hlavní napětí a přetvoření, smyková napětí, redukovaná napětí atd. Dále můžeme vytvořit určitou cestu a pomocí této cesty můžeme provést vykreslení rozložení např. napětí, tímto způsobem můžeme získat vykreslení např. napětí v různých řezech. Další možností je vykreslení různých grafů apod.

7.6. Srovnání vypočtených výsledků

Jelikož tato numerická metoda využívá při výpočtu deformační variantu, vypočetly se tedy nejprve hodnoty celkových posuvů a z těchto hodnot jsou následně dopočteny hodnoty složek napětí. Jelikož jsme použili jednotky z SI soustavy, což v našem případě znamená, že vytvořená geometrie a hodnoty geometrických okrajových podmínek jsou v *mm* a v případě silové geometrické podmínky v *MPa*. Pak nám vypočtené výsledky vyjdou v *mm* a v *MPa*.

Jednotlivé hodnoty vypočtených výsledků jsou seřazeny za sebou v tomto pořadí: (Salome, Roshaz, Ansys):

- hodnota celkového posuvu: 0,0291mm, 0,0292mm, 0,029095mm
- maximální hodnota redukovaného napětí (HMH): 302MPa, 325,6MPa, 319,951MPa

Je tedy vidět, že v případě vypočtené hodnoty celkového posuvu se výsledky ve všech programech relativně shodují. Naopak u hodnoty redukovaného napětí je rozdíl přibližně 20MPa. To může být způsobeno rozdílným tvarem prvku sítě a nebo také rozdílnou hustotou vygenerované sítě. U této úlohy je však velice jednoduché provést si kontrolu pomocí analytického řešení, kde hodnota redukovaného napětí vyšla 300 MPa.

7.7. Přehled programů a možnosti jejich použití

V *tab. 2* je uveden přehled použitých programů v této práci a také jsou uvedeny typy úloh, které jednotlivé programy dokáží řešit.

program	typ analýzy					
	S	D	M	E	T	CFD
Salome	•	–	•	–	•	•
Roshaz	•	–	–	–	–	–
Roshaz + CalculiX	•	–	•	–	•	–
Ansys	•	•	•	•	•	•

Tab. 2 Přehled programů a možnosti jejich použití

Legenda:

S – statická strukturální

E – elektromagnetismus

D – dynamická

T – přenos tepla

M – modální

CFD – mechanika tekutin

ZÁVĚR

V této práci je stručně popsán historický vývoj a matematický základ metody konečných prvků. Dále je zde vysvětlen význam slova Open Source v oblasti MKP. V další části je uveden stručný přehled MKP programů (*tab.1*) na bázi otevřeného kódu, který obsahuje název a typ programu, dále na kterém operačním systému můžeme tento program použít, typ analýzy, který dokáže řešit a nakonec www odkaz, ze kterého můžeme program bezplatně stáhnout. Z tohoto přehledu jsou vybrány některé programy, ve kterých měla být původně vytvořena statická strukturální úloha. Nakonec byl vybrán jiný volně šiřitelný program a to Salome, který je popsán detailněji a jsou zde také rozebrány jednotlivé kroky při postupu jednoduché rovinné úlohy. Totéž bylo provedeno u dvou komerčních programů Roshaz a Ansys. Jsou zde tedy použity celkem tři programy pro řešení jednoduché rovinné úlohy.

Ze srovnávací analýzy z kapitoly 7 je patrné, že volně šiřitelné programy nejsou psány lidmi, kteří se tomu intenzivně dennodenně věnují, ale tyto programy vytvářejí víceméně ve svém volném čase. Což bych řekl, že je od nich velice laskavé, neboť z toho nemají žádný zisk. Dále oceňuji, že grafické prostředí v programu se vcelku podobá komerčnímu programu Ansys (verzi 10). Ale samotné pochopení programu mi přijde jako složité, a proto si myslím, že se vyplatí použít např. dokumentaci k tomuto programu, která je k dispozici na internetu [10] a nebo také můžeme navštívit různá školení. V případě, že bychom měli nějaký konkrétnější technický dotaz, mohli bychom využít online technické podpory, ale stejně jako u školení už za to budeme muset něco zaplatit. Tato částka určitě bude zanedbatelná vůči částce, kterou bychom zaplatili za licenci komerčního programu. Salome by mohli zejména využít studenti, kteří nemají finance na zaplacení licence komerčního softwaru. Menší nevýhodou je ovšem to, že Salome je spustitelný pouze na operačním systému Linux. To ale nemusí být takový problém, protože v dnešní době lze do Windows nainstalovat virtuální operační systém (výše zmíněný Linux), což je také Open Source. V případě použití programu Roshaz se už jedná o komerční software, kde jeho cena v plné verzi vyjde v přepočtu na 12 000 Kč, což mi přijde jako velice přijatelné a tuto částku by si mohly určitě dovolit i např. menší firmy, které nepotřebují každý den provádět tyto analýzy. Na závěr program Ansys, který přijde firmu z hlediska pořízení licence velmi draho. Na druhou stranu se ovšem investice do tohoto programu vrátí, pokud se daná firma zabývá těmito analýzami. Ansys nabízí řešení široké škály fyzikálních úloh.

SEZNAM POUŽITÝCH VELIČIN

Uvedené veličiny jsou uspořádány podle výskytu v textu.

Označení	Měřicí jednotka	Význam
u, v, w	[mm]	posuv v osách x, y, z
$\varepsilon_{x,y,z}$	[-]	délková přetvoření v osách x, y, z
$\gamma_{xy,yz,xz}$	[-]	úhlová přetvoření v rovinách xy, yz, xz
$\sigma_{x,y,z}$	[MPa]	normálová napětí v osách x, y, z
$\tau_{xy,yz,xz}$	[MPa]	smyková napětí v rovinách xy, yz, xz
$\rho_{x,y,z}$	[Nm ⁻³]	objemová síla v osách x, y, z
E	[MPa]	modul pružnosti v tahu
μ	[-]	Poissonovo číslo
G	[MPa]	modul pružnosti ve smyku
K	[Nmm]	matice tuhosti
U	[mm]	matice neznámých parametrů
F	[N]	matice zatížení

SEZNAM LITERATURYKnihy a výukový text:

- [1] Turner, M.,J., Clough,R.,W., Matrin, H.,C., Topp,L., J.: Stress and deflection analysis of complex structures, J.Aero.Sci. 23, (1956), 805-823
- [2] Zienkiewicz,O.,C., Cheung,Y.K.: The finite element method in structural and continuum mechanics, New York-London McGraw-Hill,1967
- [3] Petruška, J.: Počítačové metody mechaniky II, Metoda konečných prvků, FSI VUT Brno 2003, výukový text

Internetové odkazy:

- [4] http://www.sv.vt.edu/classes/MSE2094_NoteBook/97ClassProj/num/widas/history.html
- [5] <http://people.tuke.sk/ladislav.sevcovic/aplimat08/prispevky/sebera-08.pdf>
- [6] <http://wood.mendelu.cz/cz/sections/FEM/?q=book/export/html/45>
- [7] http://cs.wikipedia.org/wiki/Open_source_software
- [8] <http://www.pooh.cz/aktovka-x/a.asp?a=2006553>
- [9] <http://www.spstrutnov.cz/o-skole/projekty/technicke-vypocty/skripta-fem-mkp.pdf/view>
- [10] <http://www.salome-platform.org/user-section/online-documentation/>
- [11] <http://www.umt.fme.vutbr.cz/~tnavrat/mkp/?q=cs/uvod>
- [12] <http://www.mece.ualberta.ca/courses/mec403/2009-W/Stress%20Concentration-09-handout.pdf>

PŘÍLOHY

CD-ROM

- obsahuje elektronickou verzi bakalářské práce ve formátech *.doc a *.pdf
- obsahuje vytvořené programy z programů Salome, Roshaz, Ansys
- obsahuje všechny obrázky použité v této bakalářské práci