

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

**VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA STROJNÍ**



METODA KONEČNÝCH PRVKŮ I

2. MATEMATICKÁ TEORIE PRUŽNOSTI

Ing. Martin Fusek, Ph.D.

Ing. Jaroslav Rojíček, Ph.D.

Ostrava 2013

© Ing. Martin Fusek, Ph.D., Ing. Jaroslav Rojíček, Ph.D.

© Vysoká škola báňská – Technická univerzita Ostrava

ISBN 978-80-248-3023-0



Tento studijní materiál vznikl za finanční podpory Evropského sociálního fondu (ESF) a rozpočtu České republiky v rámci řešení projektu: CZ.1.07/2.2.00/15.0463, MODERNIZACE VÝUKOVÝCH MATERIÁLŮ A DIDAKTICKÝCH METOD

OBSAH

1	MATEMATICKÁ TEORIE PRUŽNOSTI	3
1.1	Úvod	4
1.2	Předpoklady matematické teorie pružnosti.....	4
1.3	Napjatost.....	4
1.4	Deformace	5
1.5	Základní rovnice matematické teorie pružnosti	5
1.5.1	Rovnice rovnováhy elementární krychle ve složkách napětí.....	6
1.5.2	Geometrické rovnice a rovnice kompatibility.....	7
1.5.3	Fyzikální rovnice	9
1.5.4	Okrajové podmínky.....	10
1.6	Základní způsoby řešení soustav rovnic matematické teorie pružnosti	11
1.6.1	Deformační varianta řešení	12
1.6.2	Silová varianta	12
1.6.3	Řešitelnost	12
1.7	Zjednodušení úloh matematické teorie pružnosti.....	12
1.7.1	Zjednodušení na dvojrozměrný problém	13
1.7.2	Zjednodušení na jednodimenzionální úlohu (jednoosá napjatost)	15
1.7.3	Nosníky a skořepiny	15
1.8	Cvičení.....	16
2	LITERATURA	42



1 MATEMATICKÁ TEORIE PRUŽNOSTI



OBSAH KAPITOLY

V této přednášce se seznámíte s:

1. Matematickou teorii pružnosti
2. Možnostmi idealizace problému
3. Různým způsobem zápisu rovnic



MOTIVACE:

Připomeňme, že během vývoje teorie pružnosti lze rozeznat dvě základní vývojové větve. Jedna by se dala nazvat jako „nauka o pružnosti a pevnosti“, a dala by se charakterizovat tak, že se jedná o větev vzniklou z potřeb průmyslu, sloužící přímo k navrhování popřípadě posouzení jednotlivých konstrukcí. Stručně řečeno jedná se o technické, chceme-li inženýrské, řešení problémů. Druhá větev by se dala nazvat jako „matematická teorie pružnosti“, která usiluje především o matematickou dedukci a správnost výsledných vzorců. Už z nabytých znalostí je jasné, že obě uvedené vývojové větve nejsou, a ani nemohou být, v protikladu, protože mají společný základ. Oba dva přístupy se navzájem doplňují a současná teorie pružnosti je vlastně syntézou obou dvou přístupů. V této přednášce stručně shrneme poznatky týkající se poznatků matematické teorie pružnosti, které budou v dalších částech textu využívány.



1.1 ÚVOD

Předmětem matematické teorie pružnosti je vyšetřování stavů napjatosti a deformace obecného pružného tělesa.

Úlohy technické pružnosti a pevnosti (tak, jak byla prezentována v předmětu pružnost a pevnost) obvykle pracují s jednoduchými tělesy, jako jsou například pruty, desky, skořepiny aj. Tyto náhradní výpočetní tělesa vznikly zjednodušením obecně zatíženého prostorového tělesa na základě nějaké teorie. Díky tomuto vystačíme s relativně jednoduchým matematickým aparátem. Na druhou stranu však díky zjednodušením a předpokladům přicházíme o část informace o napětově-deformačním chování tělesa.

Naproti tomu matematická teorie pružnosti předpokládá obecná řešení napjatosti a deformace těles v libovolném tělese. Zavedeme-li však stejná zjednodušení jako v případě technického přístupu k problematice pevnosti a pružnosti, dostaneme obvykle stejné výsledky, jako nám poskytlo technické řešení problémů.

Soustava základních rovnic matematické teorie pružnosti je však obtížně řešitelná analytickými postupy. Můžeme říci, že v uzavřeném tvaru je řešitelná jen malá skupina problémů. V případě, kdy není známo exaktní matematické řešení je možno využít řešení pomocí matematické analogie nebo využití přibližných numerických metod.

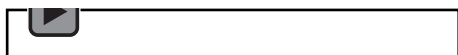
1.2 PŘEDPOKLADY MATEMATICKÉ TEORIE PRUŽNOSTI

Předmětem matematické teorie pružnosti je vyšetřování stavů napjatosti a deformace obecného pružného tělesa. Soustava rovnic matematické teorie pružnosti je odvozena za předpokladu:

- izotropního a homogenního materiálu s dokonale pružnými vlastnostmi,
- předpokládají pouze velmi malé deformace,
- rovnice rovnováhy jsou sestavovány na nedeformovaném tělese.



Audio 1.1 Teorie pružnosti

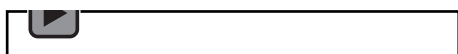


1.3 NAPJATOST

Stav napjatosti v obecném bodě tělesa popisuje symetrický **tenzor napjatosti**, jehož složky můžeme zapsat do následující matice:



Audio 1.2 Napjatost



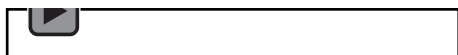
$$[\sigma] = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} \equiv \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \equiv \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \quad (2.1)$$

$$\equiv \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}.$$

Tenzor napjatosti nese informaci o intenzitě vnitřních sil v části (v daném bodě tělesa).



Audio 1.3 Tenzor napjatosti



Jsou-li známy složky tenzoru napětí v obecném bodě namáhaného tělesa, můžeme určit napjatost na obecně skloněné rovině. Můžeme definovat **vektor napětí**:

$$\{v\} = \frac{d\{F\}}{dS} = [\sigma]\{n\}, \quad (2.2)$$

kde $\{v\}$ je vektor napětí, elementární vnitřní síla $\{F\}$, elementární ploška dS a $\{n\}$ normálový vektor elementární plošky.

Často se používá inženýrský zápis ve formě vektoru¹:

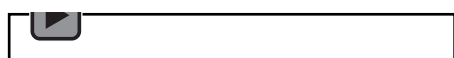
$$\{\sigma\} = \{\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{xz}, \tau_{yz}\}^T \quad (2.3)$$

1.4 DEFORMACE

Změna tvaru tělesa je popsána vektorovým polem posuvů. Maticově můžeme posuvy zapsat následující rovnicí.



Audio 1.4 Deformace



$$\{u\} = \begin{Bmatrix} u(x, y, z) \\ v(x, y, z) \\ w(x, y, z) \end{Bmatrix}. \quad (2.4)$$

Deformační stav tělesa je charakterizován infinitesimálním tenzorem deformace:

$$[\varepsilon] = \begin{bmatrix} \varepsilon_x & \frac{1}{2} \gamma_{xy} & \frac{1}{2} \gamma_{xz} \\ \frac{1}{2} \gamma_{yx} & \varepsilon_y & \frac{1}{2} \gamma_{yz} \\ \frac{1}{2} \gamma_{zx} & \frac{1}{2} \gamma_{zy} & \varepsilon_z \end{bmatrix} \equiv \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{bmatrix} \equiv \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix} \quad (2.5)$$

$$\equiv \begin{bmatrix} \varepsilon_1 & 0 & 0 \\ 0 & \varepsilon_2 & 0 \\ 0 & 0 & \varepsilon_3 \end{bmatrix}.$$

Tenzor deformace obsahuje informaci o poměrném prodloužení a úhlové změně v okolí zkoumané částice.



Audio 1.5 Tenzor deformace



Často se, také jako u napětí, používá vektorový (inženýrský) zápis ve formě:

$$\{\varepsilon\} = \{\varepsilon_x, \varepsilon_y, \varepsilon_z, \gamma_{xy}, \gamma_{xz}, \gamma_{yz}\}^T \quad (2.6)$$

Nezapomeňme, že platí:

$$\gamma_{xy} = 2\varepsilon_{xy} = 2\varepsilon_{12}, \gamma_{xz} = 2\varepsilon_{xz} = 2\varepsilon_{13}, \gamma_{yz} = 2\varepsilon_{yz} = 2\varepsilon_{23} \quad (2.7)$$

1.5 ZÁKLADNÍ ROVNICE MATEMATICKÉ TEORIE PRUŽNOSTI

Úlohou teorie pružnosti je určit v tělese vyplňujícím objem V a ohraničené povrchem S tři pole:

- vektorové pole posunutí,

¹ Správně by mělo být spíše uvedeno „pseudovektorový“ zápis, protože se nejedná o vektor ve smyslu matematickém či fyzikálním, ale spíše jako forma zápis jednotlivých složek do schématu vektoru.



- tenzorové pole deformace,
- tenzorové pole napětí.

K určení patnácti neznámých funkcí máme soustavu základních rovnic matematické teorie pružnosti (diferenciální formulace úlohy). Můžeme jí rozdělit na tři hlavní skupiny:

1. rovnice rovnováhy,
2. geometrické rovnice,
3. fyzikální rovnice.



Audio 1.6 Modelování

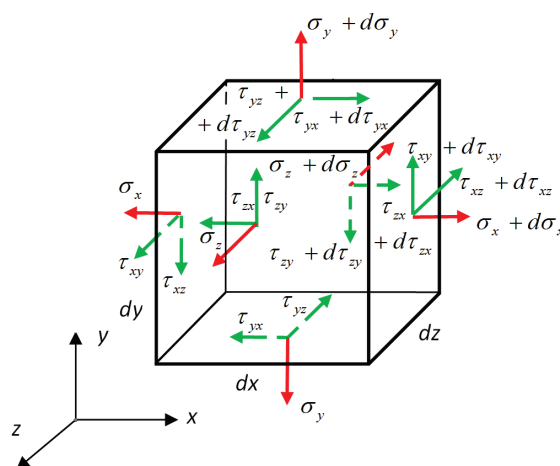
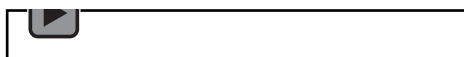


1.5.1 Rovnice rovnováhy elementární krychle ve složkách napětí

Rovnice rovnováhy elementární krychle ve složkách napětí se také označují jako tzv. Navierovy rovnice. Rovnice vyjadřují podmínky pro dosažení statické rovnováhy v libovolné elementární částici tělesa – v bodě tělesa. Mohou rovněž vyjadřovat podmínku rovnováhy podle d'Alembertova principu v případě řešení dynamického děje. Poněvadž chceme určit napjatost, nestačí napsat obvyklé podmínky rovnováhy pro vnější síly a reakce, ale je třeba uvážit rovnováhu každého libovolného bodu tělesa.



Audio 1.7 Rovnice rovnováhy



Obr. 1 - Elementární krychle

Při řešení tělesa v pravouhlých souřadnicích rozeznáváme dva tvary elementárního tělesa (bodu). Pro bod tělesa, který leží uvnitř, lze uvažovat jeho tvar jako elementární krychli, viz Obr. 1. Na povrchu tělesa nemůže (obecně) vzniknout hranolek, bod zde si můžeme představit ve tvaru elementárního čtyřstěnu. Povrchový element může být uchycen (vazba) nebo zde může působit vnější zatížení – o okrajových podmínkách viz později.

Kromě povrchového zatížení může na těleso působit objemová síla (hmotová síla). Její složky ve směru os x, y, z budeme značit X, Y, Z a jejich rozměr je jednotka síly na jednotku objemu (např. $[N/m^3]$).

Podmínku rovnováhy sil ve směru osy x můžeme vyjádřit následující rovnicí:

$$\left(\sigma_x + \frac{\partial \sigma_x}{\partial x} dx \right) dydz - \sigma_x dydz +$$



$$\begin{aligned}
& + \left(\tau_{yx} + \frac{\partial \tau_{yx}}{\partial y} dy \right) dx dz - \tau_{yx} dx dz + \\
& + \left(\tau_{zx} + \frac{\partial \tau_{zx}}{\partial z} dz \right) dx dy - \tau_{zx} dx dy + \\
& + X dx dy dz = 0.
\end{aligned} \tag{2.8}$$

Po úpravě, zkrácením elementárním objemem $dx dy dz$ získáme rovnici rovnováhy ve směru osy x . Obdobně můžeme sestavit rovnice rovnováhy pro následující dva směry, tedy osu y a z . Výsledný systém rovnic má pak následující tvar:

$$\begin{aligned}
\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + X &= 0 \\
\frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + \frac{\partial \tau_{xy}}{\partial x} + Y &= 0 \\
\frac{\partial \sigma_z}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + Z &= 0
\end{aligned} \tag{2.9}$$

Maticově lze tyto rovnice zapsat ve formě:

$$\begin{aligned}
\{\partial\}[\sigma] + \{X\} &= 0 \\
\{\partial\} &= \left\{ \frac{\partial}{\partial x} \quad \frac{\partial}{\partial y} \quad \frac{\partial}{\partial z} \right\} \\
[\sigma] &= \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} \\
\{X\} &= \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}
\end{aligned} \tag{2.10}$$

kde: $[\sigma]$ je matice tenzoru napětí, $\{\partial\}$ je vektor diferenciálních operátorů a $\{X\}$ je vektor objemových sil.

V indexové formě zápisu, s využitím Einsteinova sčítacího pravidla, můžeme rovnice rovnováhy zapsat v následujícím tvaru:

$$\sigma_{ij,i} + f_j = 0. \tag{2.11}$$

Dále lze napsat statické podmínky rovnováhy momentové. Tyto rovnice však pro nás nepřinesou nic nového, protože vyjadřují jen zákon o sdruženosti smykových napětí.

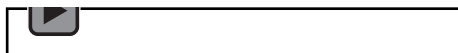
1.5.2 Geometrické rovnice a rovnice kompatibility

Chceme-li popsat deformaci celého tělesa, musíme pro každý bod tělesa $P(x, y, z)$ popsat polohu též po deformaci $P(x + u, y + v, z + w)$. Posuvy u, v, w jsou spojitými funkcemi souřadnic. Požadavek na spojitost funkcí vychází z předpokladu, že spojitost tělesa nebude během deformace porušena. V případě lineární teorie pružnosti se geometrické rovnice Cauchyho rovnice.





Audio 1.8 Geometrická rovnice



Těchto šest rovnic tvoří další část systému rovnic obecné matematické teorie pružnosti. Připomeňme, že platí pouze pro malé deformace (velice orientačně cca do 1%). Pro zopakování mají následující tvar:

$$\begin{aligned}
 \varepsilon_x &= \frac{\partial u}{\partial x} \\
 \varepsilon_y &= \frac{\partial v}{\partial y} \\
 \varepsilon_z &= \frac{\partial w}{\partial z} \\
 \gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\
 \gamma_{xz} &= \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \\
 \gamma_{yz} &= \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}
 \end{aligned} \tag{2.12}$$

Maticově lze tyto rovnice zapsat v následující formě:

$$\begin{aligned}
 \{\varepsilon\} &= [\partial]\{u\} \\
 \{\varepsilon\} &= \{\varepsilon_x, \varepsilon_y, \varepsilon_z, \gamma_{xy}, \gamma_{xz}, \gamma_{yz}\}^T \\
 [\partial] &= \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \end{bmatrix} \\
 \{u\} &= \{u, v, w\}^T.
 \end{aligned} \tag{2.13}$$

kde: $[\partial]$ je matice operátorů (operátorová matice), $\{\varepsilon\}$ vektorový zápis deformace, $\{u\}$ vektor posuvů.

V indexové formě zápisu, s využitím Einsteinova sčítacího pravidla, můžeme rovnice rovnováhy zapsat v následujícím tvaru:

$$\varepsilon_{ij} = \frac{(u_{i,j} + u_{j,i})}{2}. \tag{2.14}$$

Šest složek deformace (přetvoření) jsou vyjádřeny pomocí tří složek posuvů (u, v, w). Složky přetvoření tedy nejsou zcela nezávislými funkcemi souřadnic. Mají-li popisovat deformace spojitého tělesa, musí pro ně platit deformační podmínky, které získáme tak, že vyloučíme



posuvy z Cauchyho rovnic. Získáme tzv. rovnice kompatibility. **Rovnice kompatibility** deformací mají následující tvar:

$$\begin{aligned}\frac{\partial^2 \gamma_{xy}}{\partial x \partial y} &= \frac{\partial^2 \varepsilon_x}{\partial y^2} + \frac{\partial^2 \varepsilon_y}{\partial x^2} \\ \frac{\partial^2 \gamma_{yz}}{\partial y \partial z} &= \frac{\partial^2 \varepsilon_y}{\partial z^2} + \frac{\partial^2 \varepsilon_z}{\partial y^2} \\ \frac{\partial^2 \gamma_{zx}}{\partial z \partial x} &= \frac{\partial^2 \varepsilon_z}{\partial x^2} + \frac{\partial^2 \varepsilon_x}{\partial z^2}\end{aligned}\quad (2.15)$$

$$\begin{aligned}2 \frac{\partial^2 \varepsilon_x}{\partial y \partial z} &= \frac{\partial}{\partial x} \left[-\frac{\partial \gamma_{yz}}{\partial x} + \frac{\partial \gamma_{xz}}{\partial y} + \frac{\partial \gamma_{xy}}{\partial z} \right] \\ 2 \frac{\partial^2 \varepsilon_y}{\partial z \partial x} &= \frac{\partial}{\partial y} \left[-\frac{\partial \gamma_{zx}}{\partial y} + \frac{\partial \gamma_{yx}}{\partial z} + \frac{\partial \gamma_{yz}}{\partial x} \right] \\ 2 \frac{\partial^2 \varepsilon_z}{\partial x \partial y} &= \frac{\partial}{\partial z} \left[-\frac{\partial \gamma_{xy}}{\partial z} + \frac{\partial \gamma_{zy}}{\partial x} + \frac{\partial \gamma_{zx}}{\partial y} \right]\end{aligned}$$

V indexové formě zápisu, s využitím Einsteinova sčítacího pravidla, můžeme rovnice rovnováhy zapsat v následujícím tvaru:

$$\varepsilon_{ij,mn} + \varepsilon_{mn,ij} = \varepsilon_{in,jm} + \varepsilon_{jm,in}. \quad (2.16)$$

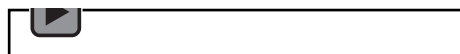
Rovnice vyjadřují geometrickou spojitost deformovatelného tělesa a musí být splněny nezávisle na tom, zda je těleso pružné či nikoliv. Nezapomeňme však, že platí jen pro malé deformace. Lze dokázat, že těchto šest rovnic dostatečně vyjadřuje podmínku zachování spojitosti tělesa a není možno nalézt další nezávislé rovnice kompatibility jinou úpravou Cauchyho rovnic.

1.5.3 Fyzikální rovnice

Složky napětí, vstupující do rovnic rovnováhy (Naviérových r. r.) a složky přetvoření v Cauchyho rovnicích (popř. v rovnicích kompatibility) jsou navzájem vázány Hookovým zákonem.



Audio 1.9 Fyzikální rovnice



Připomeňme však, že tento vztah platí pouze, je-li těleso pružné, izotropní a homogenní

$$\begin{aligned}\varepsilon_x &= \frac{1}{E} [\sigma_x - \mu \cdot (\sigma_y + \sigma_z)] \\ \varepsilon_y &= \frac{1}{E} [\sigma_y - \mu \cdot (\sigma_x + \sigma_z)] \\ \varepsilon_z &= \frac{1}{E} [\sigma_z - \mu \cdot (\sigma_x + \sigma_y)] \\ \tau_{xy} &= G \cdot \gamma_{xy} \\ \tau_{xz} &= G \cdot \gamma_{xz} \\ \tau_{yz} &= G \cdot \gamma_{yz}\end{aligned}\quad (2.17)$$

kde E je modul pružnosti v tahu, G modul pružnosti ve smyku a μ Poissonovo číslo. Maticově můžeme předchozí rovnici přepsat do tvaru:

$$\{\sigma\} = [E]\{\varepsilon\}$$



$$\{\sigma\} = \{\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{xz}, \tau_{yz}\}^T$$

$$[E] = \frac{E}{(1+\mu)(1-2\mu)} \begin{bmatrix} 1-\mu & \mu & \mu & 0 & 0 & 0 \\ \mu & 1-\mu & \mu & 0 & 0 & 0 \\ \mu & \mu & 1-\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}(1-2\mu) & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}(1-2\mu) & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}(1-2\mu) \end{bmatrix} \quad (2.18)$$

$$\{\varepsilon\} = \{\varepsilon_x, \varepsilon_y, \varepsilon_z, \gamma_{xy}, \gamma_{xz}, \gamma_{yz}\}^T$$

kde $[E]$ je matice elastických konstant, $\{\sigma\}$ je vektor složek tenzoru napětí a $\{\varepsilon\}$ je vektor složek tenzoru deformace.

Pomocí tenzorového zápisu (indexového) lze předchozí vztahy zapsat následovně:

$$\sigma_{ij} = E_{ijkl} \varepsilon_{kl}. \quad (2.19)$$

kde E_{ijkl} je tenzor elastických konstant.

$$E_{ijkl} = \lambda \delta_{ij} \delta_{kl} + G(\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}), \quad (2.20)$$

kde G je modul pružnosti ve smyku a λ je Lámého konstanta a δ_{ij} je tzv. Kroneckerovo delta:

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

$$\lambda = \frac{\mu E}{(1+\mu)(1-2\mu)}, \quad (2.21)$$

$$\delta_{ij} = 1 \text{ jestliže } i = j, \delta_{ij} = 0 \text{ jestliže } i \neq j.$$

Obecněji můžeme přepsat předchozí rovnice, s uvažováním počáteční napjatosti $\{\sigma^0\}$ popřípadě σ_{ij}^0 a počáteční deformace $\{\varepsilon^0\}$ popřípadě ε_{kl}^0 , do následujícího tvaru:

$$\{\sigma\} = \{\sigma^0\} + [E](\{\varepsilon\} - \{\varepsilon^0\}), \quad (2.22)$$

popřípadě:

$$\sigma_{ij} = \sigma_{ij}^0 + E_{ijkl}(\varepsilon_{kl} - \varepsilon_{kl}^0). \quad (2.23)$$

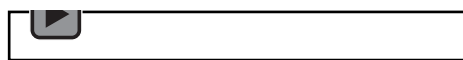
1.5.4 Okrajové podmínky

Rovnice (1), (2) a (4) jsou soustavou 15 rovnic (9 parciálních diferenciálních rovnic a 6 algebraických rovnic), kterým musí vyhovovat napjatost a deformace pružného tělesa. Soustava poskytuje obecně nekonečně mnoho řešení. Je nutno najít to, které vyhovuje okrajovým podmínkám. Rozoznáváme dva druhy okrajových podmínek.

Na hranici tělesa mohou být předepsány hodnoty složek posuvů u , v , w . Takovéto okrajové podmínky předepisují geometrickou vazbu tělesa s jeho okolím. Proto se tyto okrajové podmínky nazývají **geometrické okrajové podmínky**.



Audio 1.10 Okrajové podmínky



Někdy se též nazývají **kinematické okrajové podmínky**. Předepisují hodnotu složek vektoru posunutí na části hranice tělesa S_u .



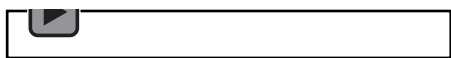
$$\{u\}|_{S_u} = \{\bar{u}\}. \quad (2.24)$$

kde $\{\bar{u}\}$ je vektor předepsaných posunutí.

Druhou skupinou okrajových podmínek jsou **statické** nebo taky **silové okrajové podmínky**. Okrajové podmínky vyjadřují statickou vazbu tělesa s jeho vnějším okolím.



Audio 1.11 Okrajové podmínky



V tomto případě jsou předepsány na hranici S_t hodnoty složek vektoru vnějšího zatížení.

$$\{\bar{t}\}|_{S_t} = [n]\{\sigma\}. \quad (2.25)$$

$$\{\bar{t}\}|_{S_t} = \{t_x, t_y, t_z\}^T$$

$$[n] = \begin{bmatrix} n_x & 0 & 0 & n_y & n_z & 0 \\ 0 & n_y & 0 & n_x & 0 & n_z \\ 0 & 0 & n_z & 0 & n_x & n_y \end{bmatrix}$$

$$\{\sigma\} = \{\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{xz}, \tau_{yz}\}^T$$

kde $\{\bar{t}\}$ je vektor plošného povrchového zatížení, $[n]$ je matice směrových kosinů plochy (její normály) na které působí povrchové zatížení ($n_x = \cos\alpha, n_y = \cos\beta, n_z = \cos\gamma$) a $\{\sigma\}$ je vektor uspořádaných složek tenzoru napětí.

V tenzorové notaci:

$$t_j = \sigma_{ij}n_i. \quad (2.26)$$

1.6 ZÁKLADNÍ ZPŮSOBY ŘEŠENÍ SOUSTAV ROVNIC MATEMATICKÉ TEORIE PRUŽNOSTI

Úlohou pružnosti je určit vektorové pole posuvů a dvě tenzorová pole přetvoření (deformací) a napětí. Řešení problémů pružného tělesa lze rozdělit na tři základní úlohy:

1. **Přímá úloha.** Pružné těleso je popsáno svým tvarem (geometrií), jsou zadány fyzikální vlastnosti materiálu a je zatíženo objemovými a vnějšími povrchovými silami. Úkolem je stanovit tenzorové pole napětí a deformace a vektorové pole posunutí v celém tělese (uvnitř i na povrchu).
2. **Inverzní úloha.** Pružné těleso je popsáno geometrií a fyzikálními vlastnostmi materiálu. Předepsáno je pole napětí a posuvů. Úlohou je stanovit vnější zatížení, která daná pole vyvolaly, popřípadě zjistit okrajové podmínky, kterým dané funkce pole napětí vyhovují.
3. **Poloinverzní úloha.** Pro výše zadané těleso jsou částečně zadány silové veličiny a částečně zadány posuvy. Uvnitř tělesa jsou známy pouze některé složky tenzoru napětí.

Nejčastěji se v pružnosti řeší první jmenovaná, tj. přímá úloha. Soustavu rovnic matematické teorie pružnosti tvoří celkem 15 rovnic s příslušnými okrajovými podmínkami. Většinou se neřeší přímo tyto rovnice, ale upraví se do tvaru, kdy jsou některé veličiny vhodnou matematickou úpravou odstraněny. Soustavu rovnic můžeme řešit obecně třemi způsoby – deformační, silovou a smíšenou metodou.




Audio 1.12 Soustava rovnic


Deformační metodou se úloha řeší ve složkách posuvů u, v, w . Silovou metodou se úloha řeší ve složkách tenzoru napětí $\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{xz}, \tau_{yz}$. U smíšené metody řešení se využívají složky napětí i ve složky posuvů.

Standardně se však rovnice matematické teorie pružnosti upravují buď na deformační, nebo silovou variantu. Obě dvě varianty jsou si z hlediska matematiky rovnocenné. V praktických aplikacích se častěji využívá deformační varianta, neboť má o tři rovnice méně a lépe se pracuje se smíšenými okrajovými podmínkami.

1.6.1 Deformační varianta řešení

Při deformační variantě řešení se berou jako hlavní neznámá posuvy. Z Cauchyho rovnic (2.12) se dosadí do rovnic fyzikálních (konstitutivních) (2.17) a odtud pak do Navierových rovnic rovnováhy (2.9). Rovnice rovnováhy jsou tedy vyjádřeny v posuvech. Tyto rovnice musí splňovat všechny předepsané okrajové podmínky. V případě izotropního Hookeovského materiálu získáme tzv. **Lamé-Navierovy rovnice**. Ty představují soustavu tří lineárních parciálních diferenciálních rovnic druhého řádu pro tři neznámé složky posuvu. Jejich tvar v indexové notaci je následující:

$$u_{i,jj} + \frac{1}{1-2\mu} u_{k,ki} + \frac{f_i}{G} = 0. \quad (2.27)$$

1.6.2 Silová varianta

Při silové variantě se vyjadřují rovnice rovnováhy v napětích. Do rovnic kompatibility deformace (2.15) se dosadí deformace z konstitutivní rovnice (2.12). Takto získáme rovnice kompatibility vyjádřeny v napětích. Dosazením rovnic rovnováhy (2.9), po příslušných matematických úpravách, do takto upravených rovnic kompatibility lze pro materiál řídicí se Hookeovým zákonem (2.17) získat po úpravách systém celkem šesti lineárních parciálních diferenciálních rovnic druhého řádu pro šest složek tenzorového pole napjatosti. Takto získané rovnice se nazývají **Beltram-Mitchelovy rovnice**.

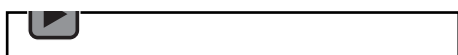
$$\sigma_{ij,kk} + \frac{1}{1+\mu} \sigma_{kk,ij} = -(f_{i,j} + f_{j,i}) - \frac{\mu}{1-\mu} \delta_{ij} f_{k,k}. \quad (2.28)$$

1.6.3 Řešitelnost

Soustava rovnic pro určení napjatosti a deformace pružného tělesa je natolik složitá, že řešení lze získat jen pro jednoduché tvary těles a pro určité typy okrajových podmínek. Lze dokázat, že pokud nalezneme řešení, které vyhovuje všem rovnicím matematické teorie pružnosti, je řešení jednoznačné.

1.7 ZJEDNODUŠENÍ ÚLOH MATEMATICKÉ TEORIE PRUŽNOSTI

Řešení obecných prostorových úloh je velice obtížné. V mnoha případech můžeme úlohu zjednodušit na dvojrozměrnou (rovinnou), popřípadě na jednorozměrnou. Tímto dosáhneme značného zjednodušení popisu a řešení úlohy.


Audio 1.13 Řešení obecných prostorových úloh


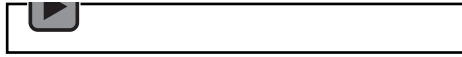
1.7.1 Zjednodušení na dvojrozměrný problém

Rovnice matematické teorie pružnosti se pro rovinný stav namáhání značně zjednoduší, protože všechna napětí, deformace a posuvy jsou pouze funkcemi dvou proměnných x a y . Rozoznáváme dva základní případy – rovinná deformace, rovinná napjatost.

Dále pak zde můžeme zařadit osově souměrnou úlohu (vzhledem k ose musí být souměrná: geometrie, okrajové podmínky a materiál).



Audio 1.14 Zjednodušení

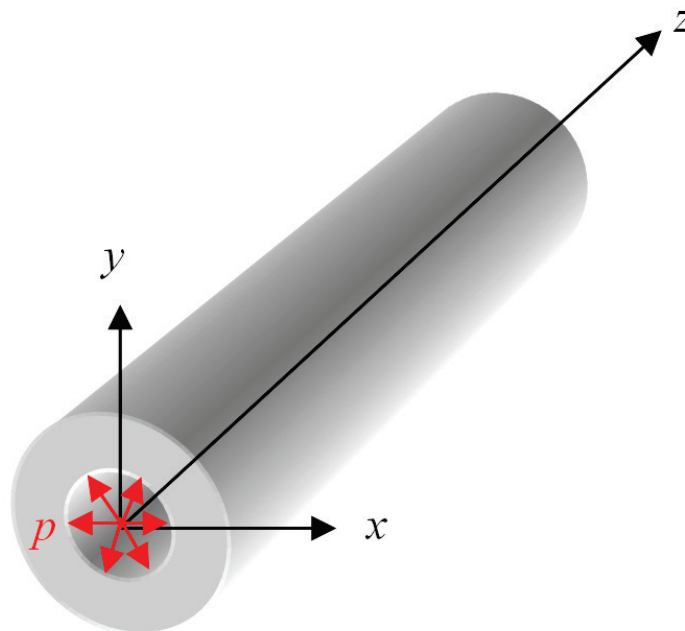


1.1.1.1 Rovinná deformace

Případ **rovinné deformace** je dán podmínkou, aby složka posuvu v jednom směru byla nulová. Bude-li nulový posuv ve směru osy z , musí platit $w = 0$. V tomto případě budou pole posuvů, deformací a napětí následující:

$$\begin{aligned} \{u\} &= \{u, v, 0\}^T \\ \{\varepsilon\} &= \{\varepsilon_x, \varepsilon_y, 0, \gamma_{xy}, 0, 0\}^T \\ \{\sigma\} &= \{\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, 0, 0\}^T \end{aligned} \quad (2.29)$$

Takový stav deformace je přibližně dosažen u těles, jejichž rozměr ve směru osy z je převládající, tj. výrazně větší než v dalších dvou směrech a vyšetřovaný průřez je dostatečně vzdálen od konců tělesa. Další podmínkou je, aby zatížení působilo ve směru kolmém na osu z a současně nebylo funkcí této souřadnice. Jako příklad můžeme uvést dlouhou tlustostěnnou trubku zatíženou vnitřním přetlakem – viz Obr. 2.



Obr. 2 - Příklad rovinné deformace

Pozn.: Můžeme se setkat i s pojmem zobecněná rovinná deformace. V tomto případě se předpokládá, že posuv w je nenulový, ale konstantní. Dále se tímto případem nebudeme zabývat.

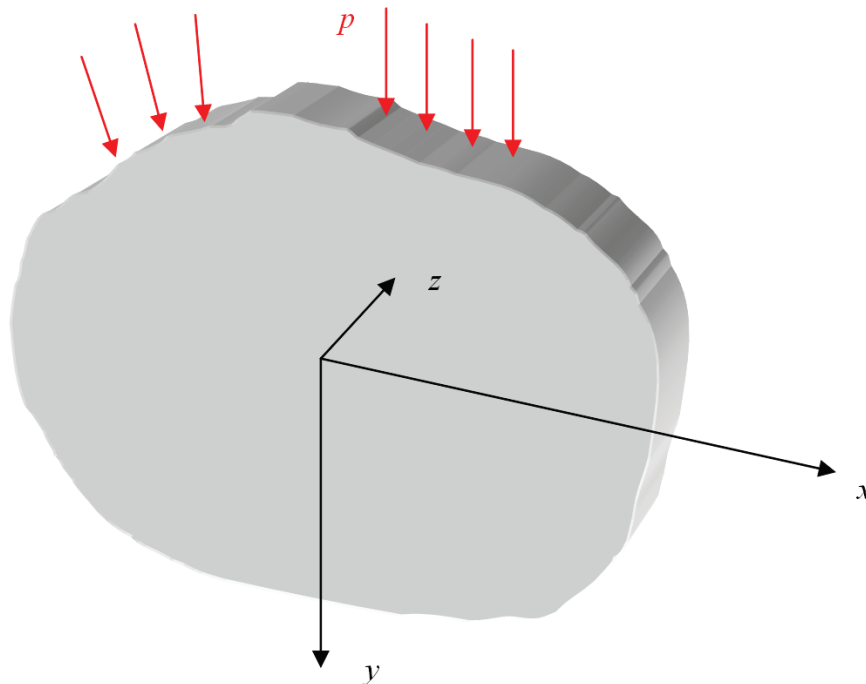


1.1.1.2 Rovinná napjatost

Pro případ **rovinné napjatosti** je hlavní podmínkou, aby všechny složky napětí, působící na element tělesa, byly rovnoběžné s jednou rovinou. Složky tenzoru napětí ve směru osy z jsou rovny nule a zksoy jsou nenulové pouze v rovině xy . V tomto případě budou pole posuvů, deformací a napětí následující:

$$\begin{aligned}\{u\} &= \{u, v, w\}^T \\ \{\varepsilon\} &= \{\varepsilon_x, \varepsilon_y, \varepsilon_z, \gamma_{xy}, 0, 0\}^T \\ \{\sigma\} &= \{\sigma_x, \sigma_y, 0, \tau_{xy}, 0, 0\}^T\end{aligned}\quad (2.30)$$

Popsaný stav napjatosti se vyskytuje např. v tenké desce (viz Obr. 3) a na nezátíženém povrchu tělesa.



Obr. 3 - Příklad rovinné napjatosti

1.1.1.3 Osově souměrnou úlohu

Pokud úloha splňuje podmínky osové symetrie, tj. vzhledem k ose musí být souměrná jednak geometrie, dále pak okrajové podmínky a v neposlední řadě materiálové vlastnosti, je s výhodou použít právě tento typ zjednodušení. Problém je pak výhodné vyjádřit ne v pravoúhlém kartézském souřadném systému jako doposud, ale zapsat všechny potřebné vztahy v závislosti na cylindrickém (válcovém) souřadném systému se souřadnicemi r, φ, z . Pokud jsou splněny podmínky osové symetrie, a osa symetrie je osa z , pak všechny veličiny jsou nezávislé na souřadnici φ . Úlohu můžeme řešit pouze v meridiánové rovině dané rovinou $r - z$.

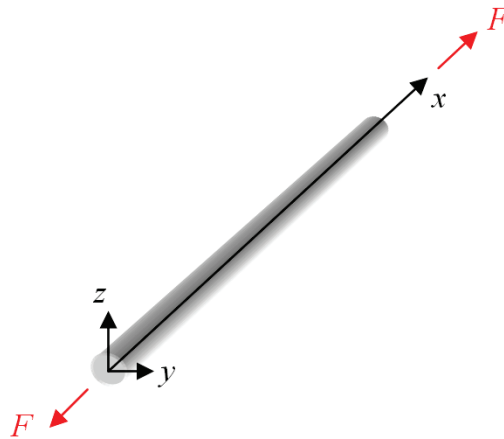
$$\begin{aligned}\{u\} &= \{u, 0, w\}^T = \{u_r, u_\varphi = 0, u_z\}^T \\ \{\varepsilon\} &= \{\varepsilon_{rr}, \varepsilon_{\varphi\varphi}, \varepsilon_{zz}, 0, \varepsilon_{rz}, 0\}^T \\ \{\sigma\} &= \{\sigma_{rr}, \sigma_{\varphi\varphi}, \sigma_{zz}, 0, \sigma_{rz}, 0\}^T\end{aligned}\quad (2.31)$$



1.7.2 Zjednodušení na jednodimenzionální úlohu (jednoosá napjatost)

Jednorozměrné těleso (tyčové těleso) je takové, jehož jeden rozměr (např. ve směru osy x) je o mnoho větší než ostatní dva. Současně silové zatížení je aplikováno pouze ve směru převládajícího rozměru. Tento případ odpovídá namáhání prutu tahem a tlakem – viz Obr. 4. V tomto případě budou pole posuvů, deformací a napětí následující:

$$\begin{aligned} \{u\} &= \{u, 0, 0\}^T \\ \{\varepsilon\} &= \{\varepsilon_x, \varepsilon_y, \varepsilon_z, 0, 0, 0\}^T \\ \{\sigma\} &= \{\sigma_x, 0, 0, 0, 0, 0\}^T \end{aligned} \quad (2.32)$$



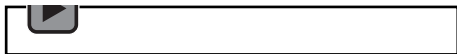
Obr. 4 - Jednorozměrná napjatost

1.7.3 Nosníky a skořepiny

Další typ zjednodušení úloh teorie pružnosti je přijetí další dodatečných zjednodušení. Tyto předpoklady vedou na definice těles, jako jsou nosníky a skořepiny.



Audio 1.15 Zjednodušení



Nosníkové těleso je podobné tyčovému jednorozměrnému tělesu. Rozdíl je zde ve způsobu zatížení. Zatížení zde leží ve směru kolmém k ose tělesa. Takže bude-li osa tělesa (převládající rozměr tělesa) ve směru osy x , pak průhyb bude ve směru osy y (popř. z). Detailněji byla problematika probána v základním kurzu pružnosti a pevnosti a bude stručně zopakována později ve skriptech.

Obdobně můžeme získat rovnice pro výpočet napěťově-deformační odezvy pro desková a skořepinová tělesa. Takováto tělesa mají převládající dva rozměry (např. ve směrech x a y) a zatížena jsou ve směru třetím. Detailněji bude problematika probírána v předmětu Pružnost a pevnost II.

1.8 CVIČENÍ

Soubory při práci s programem Patran, Nastran, atd

Program Patran je určen k přípravě modelu a k zpracování výsledků. Samotný výpočet provádí tzv. řešič, který volíme v nabídce **Analysis Preference** (např. Ansys, MSC Nastran, MSC Marc). V jednotlivých řešičích jsou dostupné různé typy analýz (Analysis Type) – Structural, Thermal, Coupled atd.

Patran vytváří několik základních typů souborů:

- **Nazev_souboru.db** Databáze reprezentující geometrický a konečnoprvkový model – binární soubor, je poměrně velký.
- **Nazev_souboru.db.jou** Journal file – obsahuje jednotlivé příkazy prováděné programem při vytváření databáze. Je možné využít k opakovanému spuštění, opravám, přípravě maker apod. Jedná se o textový soubor.

```

$# Creating journal file
$# F:\Pracovni\Jarek\Granty\MKPI\Vymazat\cviceni_2.db.jou at 12-Apr-12 12:33:26
util_file_rebuild.start("C:\MSC\Software\Patran_x64\20121\template.db", @
"F:\Pracovni\Jarek\Granty\MKPI\Vymazat\cviceni_2.db")
$# Synchronizing the User Data Manager
set_current_dir( "F:\Pracovni\Jarek\Granty\MKPI\Vymazat" )
$# The current working directory has been changed to the database location
$# shown.
$#
$#
util_pref_analysis.set_analysis_preference( "MSC.Marc", "Structural", ".dat", @".t16", "No Mapping" )
$# Changing results display tool settings - DeformedScale: Model=0.1 to
$# DeformedScale:True=1..
$# Loading... msc_marc_defn_dflt_tbs( MSC.Marc, Structural )
$# session file patran.ses.104 stopped recording at 12-Apr-12 12:42:08
$# session file F:\Pracovni\Jarek\Granty\MKPI\Vymazat\patran.ses.01 started
$# recording at 12-Apr-12 12:42:08
$# Build: 19.0.132332 wed Jan 25 01:11:45 PST 2012
$# Recorded by: Patran 2012 64-Bit
STRING ge_create_solid_pri_created_ids[VIRTUAL]
ge_const_solid_block("1", "1.0", "1.0", "1.0", 0, "", "coord 0", "[0 0 0]", @
ge_create_solid_pri_created_ids )

```

- **Nazev_souboru.ses.01** Session file – obsahuje jednotlivé příkazy prováděné programem v jednotlivé seanci (blok od zapnutí do vypnutí programu Patran). Je menší než Journal file, lépe se zde orientuje. Poslední číslo v příponě názvu (01, 02, atd.) označuje, kolikrát jsme již s programem pracovali. Každý blok vytvoří nový session file. Session file je textový soubor, mající obdobnou strukturu jako Journal File.

- **Nazev_souboru.out** Neutral file – obsahuje informace o modelu, je možné použít jako zálohu. Jedná se o textový soubor.

```

25      0      0      1      0      0      0      0      0
P3/PATRAN Neutral File from: F:\Pracovni\Jarek\Granty\MKPI\Vymazat\cviceni_2.db
26      0      0      1      0      0      1      0      0
12-Apr-12 13:17:07      3.0
3      1      1      20      0      0      0      0      0
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.10000000E+01 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.21000000E+06 0.21000000E+06 0.21000000E+06 0.30000012E+00
0.30000012E+00 0.30000012E+00 0.80769226E+05 0.80769226E+05 0.80769226E+05
0.00000000E+00 0.28269228E+06 0.12115385E+06 0.12115385E+06 0.28269228E+06
0.12115385E+06 0.28269228E+06 0.80769226E+05 0.00000000E+00 0.00000000E+00
0.80769226E+05 0.00000000E+00 0.80769226E+05 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00 0.00000000E+00
31      1      0      1      0      0      0      0      0
1.00000000E+0 0.00000000E+0 1.00000000E+0 0 0 0

```

- **Nazev_souboru.db.bkup** Databáze – Zálohová databáze. (Pro „zálohu“ je možné použít také Journal file, session file, nebo příkaz **Save a Copy**).

Journal file, PCL – základní struktura



Journal File - obsahuje jednotlivé příkazy prováděné programem a komentáře popisující jednotlivé kroky. Struktura a příkazy skriptovacího jazyka tzv. **PATRAN Command Language (PCL)** je možné najít v "Helpu". Struktura Journal File je poměrně přehledná a můžeme ji rozdělit do několika částí:

- **Popis, text, komentář** – existuje několik možností zapsání popisu - komentáře.

```
$# Komentář
```

```
$ Komentář
```

```
# Komentář
```

```
/* Komentář */
```

```
/*
```

```
* Komentář v bloku
```

```
*/
```

- **Proměnné** (musí začínat znakem) – V prvním kroku proměnné deklarujeme, ve druhém přiřadíme hodnotu. Proměnné jsou nejčastěji:
 - **Celá čísla** – **INTEGER**
 - **Reálná čísla** - **REAL**
 - **Řetězce znaků** - **STRING jméno[počet znaků]**
 - **Logické**, 1-pravda-TRUE, 0-nepravda-FALSE - **LOGICAL**,
 - **Pole proměnných**
 - **Atd.**

```
$# Deklarace proměnné
```

```
INTEGER a
```

```
$# Přiřazení hodnoty
```

```
a = 6
```

```
REAL x, y, Radius,
```

```
x = -5.06; y = 10.E-2;
```

```
Radius = 8.3
```

```
$# Přiřazení hodnoty přímo při deklaraci
```

```
REAL z = -8.3
```

```
$# Deklarace řetězce znaků
```

```
STRING name[20]
```

```
name = "Síla"
```

```
$# Deklarace logické proměnné,
```

```
LOGICAL Pravda
```

```
Pravda = TRUE
```

```
Pravda = FALSE
```

```
$# Pole reálných hodnot
```

```
REAL TABLE(2,3) = [ 10, 20, 30, 11, 21, 31 ]
```

- **Operátory a matematické výrazy**

```
$# součet, rozdíl, násobení, dělení, závorky
```

```
Real c = 1*(2/x + 2-Radius)
```

```
$# mocniny
```

```
Real d = c**3
```

```
$# porovnání u podmíněných příkazů
```

```
c > d
```



```

c <= Radius
$# zvýšení, snížení, přiřazení
INTEGER a = 6
INTEGER b, c, d
b += a
c -= a
d = a
$# a = 6, b = 7, c = 5, d = 6

```

- **Příkazy cyklu For, While, Repeat**

```

INTEGER i,j
Integer   pole(10,10)
FOR (j = 1 TO 10)
    FOR (i = 1 TO 10)
        pole (i, j) = i*j
    END FOR
END FOR

```

- **Příkazy podmíněné IF (Logický výraz) THEN příkaz, apod.**

```

LOGICAL Pravda = TRUE
Real pi
IF (Pravda) THEN
    pi=3.14
END IF

```

- **Subrutiny a funkce** - Příkazy jsou často na několik řádků, kde znak @ označuje pokračování na dalším řádku. Textové řetězce se dávají do uvozovek.

```

uil_file_rebuild.start("C:\MSC.Software\Patran_x64\20121\template.db", @
"F:\Pracovni\Jarek\Granty\MKP1\Vymazat\Cviceni_2.db")

```

```

$# Funkci můžeme také vytvořit
FUNCTION Jméno( arglist )
    Deklarace proměnných

    příkazy

    RETURN value
END FUNCTION

```

```

$# Vytvořenou funkci můžeme v programu zavolat
!!INPUT Jméno

```

- **Výstupy do souboru, textové výstupy do okna historie apod.**

```

$# Výpis do okna historie !$
!$` SQRT(250.) * 12.4`
$ 196.0612
$# Výpis do okna historie pomocí funkce write
WRITE (SQRT(250.) *12.4)

```

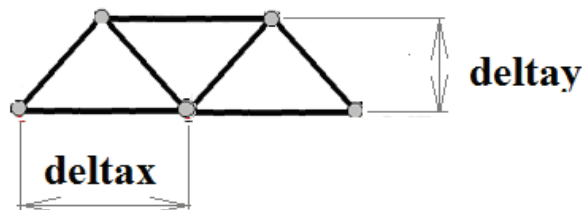
- **Dále následují dva jednoduché příklady**



```

$# Příklad1 vykreslení oblouku
$# Deklarace
integer i
real R = 3.1
real x,y
real fi = 5
real z = 0
string poloha1 [VIRTUAL]
string poloha2 [VIRTUAL]
$# Vytvoreni bodu - Point
STRING asm_create_grid_xyz_created_ids[VIRTUAL]
asm_const_grid_xyz( "1", "[0 0 0]", "Coord 0", asm_create_grid_xyz_created_ids )
FOR (i = 1 TO 10)
    x = R * mth_cosd (fi*i)
    y = R * mth_sind (fi*i)
    poloha1 = "[ " // str_from_real ( x ) // " "
    poloha1 = poloha1 // str_from_real ( y ) // " "
    poloha1 = poloha1 // str_from_real ( z ) // " ]"
    asm_const_grid_xyz( "#", poloha1 , "Coord 0", asm_create_grid_xyz_created_ids )
END FOR
$# Vytvoreni cary - curve
STRING asm_line_2point_created_ids[VIRTUAL]
FOR (i = 2 TO 10)
    poloha1 = "Point" // str_from_integer ( i )
    poloha2 = "Point" // str_from_integer ( i+1 )
    asm_const_line_2point( "#", poloha1, poloha2, 0, "", 50., 1,
asm_line_2point_created_ids )
END FOR
$# Konec prikladu1

```



```

$# Příklad2 vykreslení jednoduché prutové soustavy
$# Vykreslení prutové konstrukce
$# Deklarace
integer i,nx
real x,y,deltax,deltay
nx=5
deltax=1000
deltay=1000
string poloha1 [VIRTUAL]
string poloha2 [VIRTUAL]
$# Vytvoreni bodu - Point
STRING asm_create_grid_xyz_created_ids[VIRTUAL]
FOR (i = 1 TO nx)
    x = (i-1)*deltax
    poloha1 = "[ " // str_from_real ( x ) // " 0 0]"

```



```

asm_const_grid_xyz( "#", poloha1, "Coord 0", asm_create_grid_xyz_created_ids )
END FOR
FOR (i = 1 TO (nx-1))
  x = (i-1)*deltax+deltax/2
  poloha1 = "[ " // str_from_real ( x )
  poloha1 = poloha1 // str_from_real ( deltay ) // " 0]"
  asm_const_grid_xyz( "#", poloha1, "Coord 0", asm_create_grid_xyz_created_ids )
END FOR

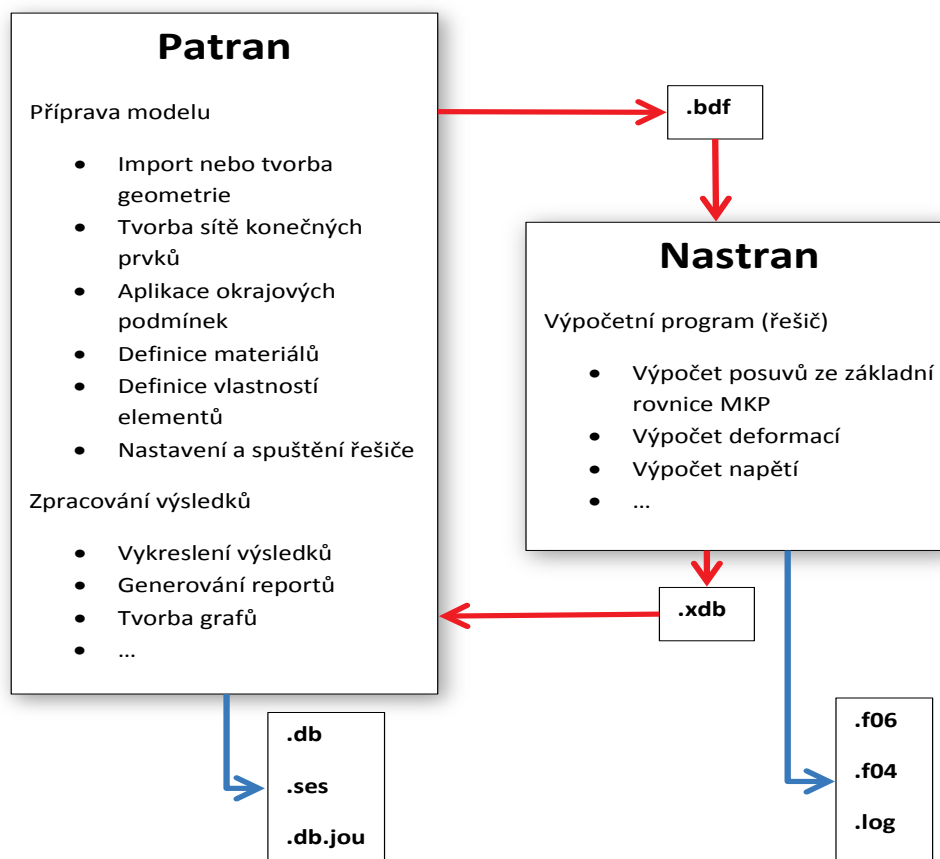
$# Vytvoreni cary - curve
STRING asm_line_2point_created_ids[VIRTUAL]
FOR (i = 1 TO (nx-1))
  poloha1 = "Point" // str_from_integer ( i )
  poloha2 = "Point" // str_from_integer ( i+1 )
  asm_const_line_2point( "#", poloha1, poloha2, 0, "", 50., 1,
asm_line_2point_created_ids )
END FOR
FOR (i = 1 TO (nx-1))
  poloha1 = "Point" // str_from_integer ( i+nx-1 )
  poloha2 = "Point" // str_from_integer ( i+nx )
  asm_const_line_2point( "#", poloha1, poloha2, 0, "", 50., 1,
asm_line_2point_created_ids )
END FOR
FOR (i = 1 TO (nx-1))
  poloha1 = "Point" // str_from_integer ( i )
  poloha2 = "Point" // str_from_integer ( i+nx )
  asm_const_line_2point( "#", poloha1, poloha2, 0, "", 50., 1,
asm_line_2point_created_ids )
END FOR
FOR (i = 1 TO (nx-1))
  poloha1 = "Point" // str_from_integer ( i+1 )
  poloha2 = "Point" // str_from_integer ( i+nx )
  asm_const_line_2point( "#", poloha1, poloha2, 0, "", 50., 1,
asm_line_2point_created_ids )
END FOR
$# konec prikladu2

```

Ke komunikaci mezi Patranem a vybraným řešičem (MSC Marc, Nastran, Ansys atd.) pak slouží soubory typu .bdf, .dat, xdb atd. Řešiče pak při řešení používají soubory s příponami .f06, .f04, .log atd. U vybraných řešičů si tyto soubory projdeme.

- **Nastran** – Program Nastran je konečnoprvkový řešič. Jméno programu je odvozeno z **NA**sá **STR**uctural **AN**alysis System. Program je primárně napsán v jazyce Fortran a v současnosti obsahuje více než milión řádků kódu. Je dostupný pod mnoha operačními systémy a může být spuštěn na osobním počítači i na velkých superpočítačích. Program jako takový se skládá z množství modulů, kde každý modul odpovídá za určitou specifickou operaci (aplikace okrajových podmínek, generování matic, vlastní řešení, atd.). Jednotlivé moduly jsou kontrolovány pomocí vnitřního jazyka DMAP (Direct Matrix Abstraction Program).



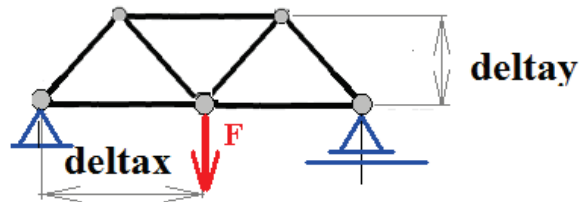


Obrázek 1- Vztah mezi programy Nastran a Patran

Program obsahuje velké množství analýz. Každý typ analýzy je označen jako sekvence řešení (solution sequence) a má svoje kódové číslo. Tak například sekvence 101 je lineární statická analýza, 129 je nelineární transienční analýza, 700 explicitní nelineární analýza, atd. Program Nastran neposkytuje žádné grafické rozhraní, neposkytuje žádné funkce pro grafickou tvorbu modelu. Všechny vstupy a výstupy programu jsou zprostředkovány pomocí textových souborů. Základní typy souborů jsou:

- **Jméno_souboru.bdf** – vstupní soubor. Obsahuje definici modelu, je to textový soubor. Umožňuje přidávat vstupy nepodporované v Patranu. Nastranovský soubor můžeme rozdělit do pěti sekcí.
 1. Nastavení Nastranu (Nastran Statement) je určena na modifikaci defaultního nastavení nastranu. Není povinná.
 2. Správa souborů (File Management Section) je určena k nastavení umístění souborů, restartu apod. Není povinná
 3. Výkonná část (Executive Control Section) je určena k nastavení typu řešení, času řešení, modifikací apod. Je povinná a začíná identifikační značkou **ID A,B**

4. Část definice (Case Control Section) je určena k nastavení výstupů, určuje následující sekci. je povinná a začíná klíčovým slovem **CEND**.
5. Datová část (Bulk Data Section) – Definice modelu, zatížení a okrajových podmínek. Je povinná a začíná klíčovým slovem **BEGIN BULK** a končí klíčovým slovem **ENDDATA**, které ukončuje vstupní soubor.



\$ Jednoduchý model prutové konstrukce v Nastranovském souboru může vypadat např. takto...

\$ 1. Část Nastavení a 2. část - správa souborů je nepovinná.

\$ Linear Static Analysis, Database – zvolený řešič

ID Testovací_uloha, priklad

SOL 101

TIME 1

\$ Část definice

CEND

\$ Název

TITLE = Testovací_uloha

ECHO = NONE

SUBCASE 1

\$ Subcase name : Default

SUBTITLE=Default

SPC = 4

LOAD = 5

DISPLACEMENT(SORT1,REAL)=ALL

SPCFORCES(SORT1,REAL)=ALL

STRESS(SORT1,REAL,VONMISES,BILIN)=ALL

\$ Datová část

BEGIN BULK

PARAM POST 0

PARAM PRTMAXIM YES

\$ definice geometrie

\$	bod	x	y	z
Grid	1	0.	0.	0.
Grid	2	1000.	0.	0.
Grid	3	2000.	0.	0.
Grid	4	500.	1000.	0.
Grid	5	1500.	1000.	0.

\$ definice elementů

\$	číslo	mat.	bod1	bod2
CROD	1	1	1	2
CROD	2	1	2	3
CROD	3	1	4	5



```

CROD 4      1      1      4
CROD 5      1      2      4
CROD 6      1      2      5
CROD 7      1      3      5
$ Definice materialu
MAT1 1 210000. 80769.2 .3
$ Definice zatěžovacího stavu : Default
SPCADD 4 1 2 3
LOAD 5 1. 1. 1
$ Vlastnosti elementu R = 10 mm
$      typ      mat.      S
PROD 1 1 314
$ Definice vazeb
$      ozn.      dof      bod1      bod2
SPC1 1 123 1
SPC1 2 2 3
SPC1 3 3 1 2 3 4 5
$ Definice sil
$      ozn.      bod      sila      x      y      z
FORCE 1 2 0 1000. 0. -1. 0.
$ Ukončení datové sekce
ENDDATA 0103f4b3

```

- **Jméno_souboru.f06** – je základní Natranovský výstupní soubor. Obsahuje výsledky řešení, hlášení informační a hlášení chybová apod. Je to textový soubor. Např. výsledky mohou být zobrazeny v tomto tvaru:

```

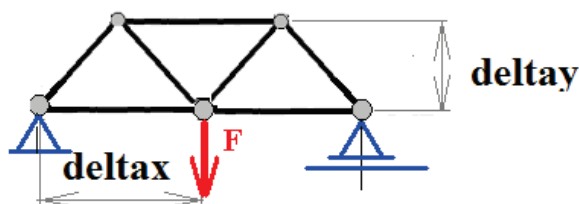
              D I S P L A C E M E N T   V E C T O R
POINT ID.  TYPE      T1      T2      T3      R1      R2      R3
   1      G      0.0      0.0      0.0      0.0      0.0      0.0
   2      G      0.0      0.0      0.0      0.0      0.0      0.0
   3      G      1.516530E-02  -5.805925E-02  0.0      0.0      0.0      0.0
1  TESTOVACI_ULOHA                                APRIL 13, 2012 MD NASTRAN 4/27/11 PAGE 10
0
E L E M E N T   A X I A L   S T R E S S E S   I N   R O D   E L E M E N T S   ( C R O D )
ID.  AXIAL  SAFETY  TORSIONAL  SAFETY  ELEMENT  AXIAL  SAFETY  TORSIONAL  SAFETY
   1  STRESS MARGIN  STRESS  MARGIN  ID.  STRESS MARGIN  STRESS MARGIN
1  TESTOVACI_ULOHA  -4.503865E+01  0.0      2  3.184713E+01  0.0
                                APRIL 13, 2012 MD NASTRAN 4/27/11 PAGE 11

```

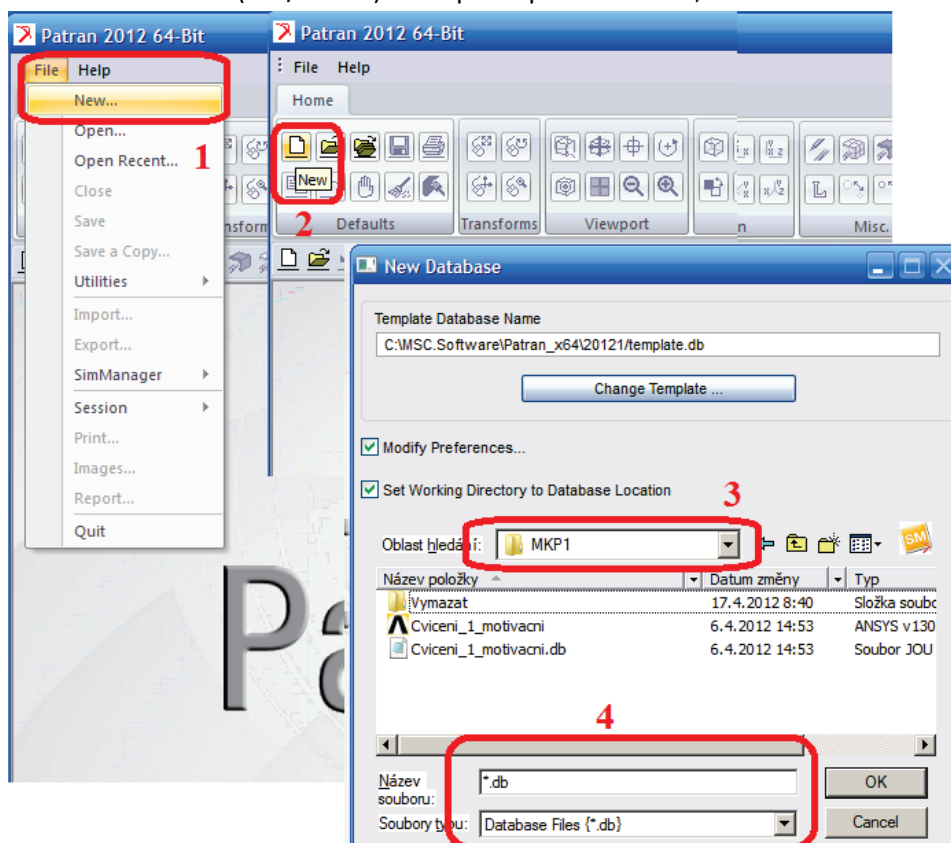
- **Jméno_souboru.f04** – obsahuje historii řešení. Je to textový soubor.
- **Jméno_souboru.log** – Základní informace oběhu programu. Je to textový soubor.

V poslední části projdeme příkazy pro vytvoření prutové soustavy. Budeme postupovat po jednotlivých krocích.





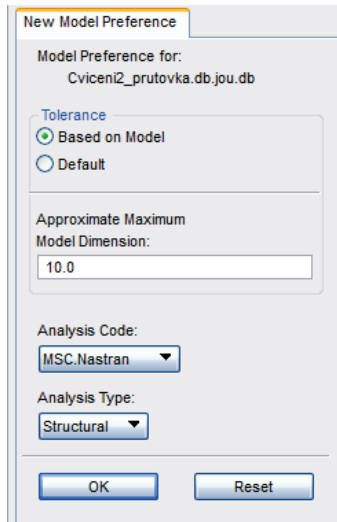
- Po spuštění programu vytvoříme novou databázi (obr. 1). Novou databázi vytvoříme přes hlavní menu viz obr.1 bod1 (File/New...) nebo přes aplikační menu, záložka Defaults viz obr.1 bod 2.



Obr. 1 Vytvoření nové databáze.

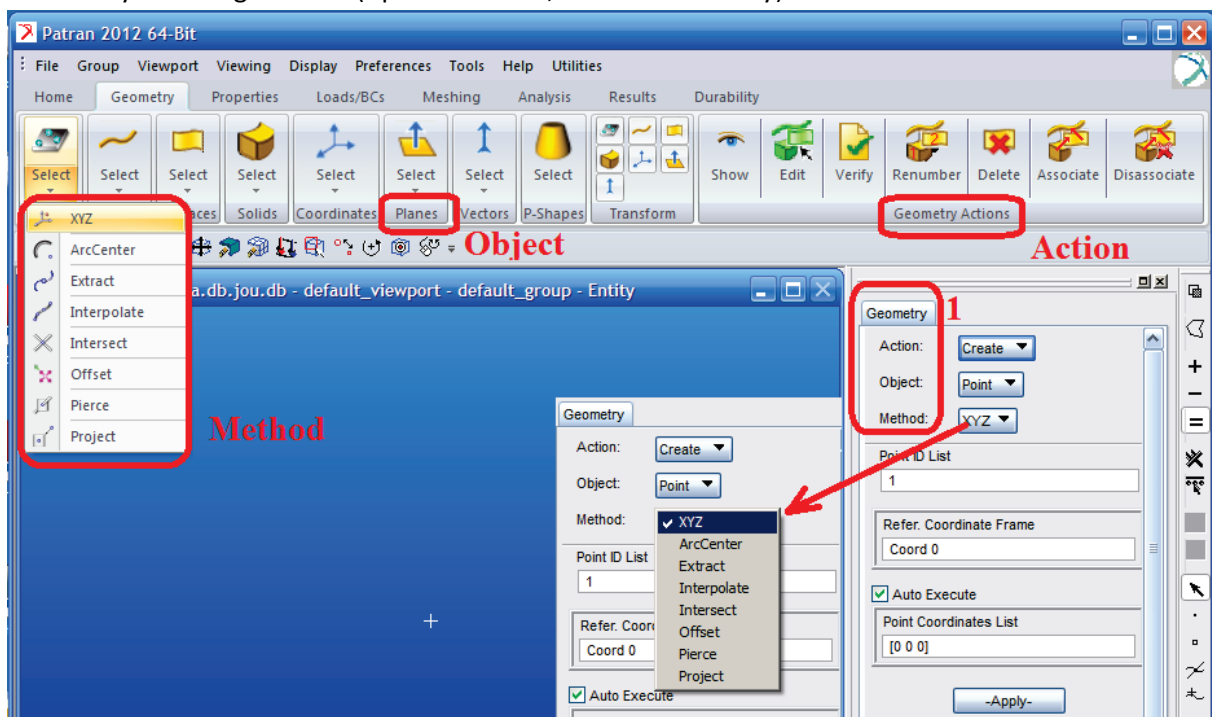
V dialogovém okně, které se objeví, můžeme nastavit pracovní adresář obr.1 bod3 a nový název databáze obr.1 bod 4. V okně je dále možné změnit šablonu (template.db), kde je možné nastavit různé vlastnosti, se kterými začínáme práci v patranu – např. řešič nebo pozadí. Přepínač Modifikace preferencí... po spuštění spustí nabídku nastavení preferencí viz. obr 2. Přepínač Set Working Directory to Database Location zajistí, že všechny vytvořené soubory (.db, .jou, .ses, .bdf, atd.) se budou ukládat do adresáře, kde vytvoříme databázi.





Obr.2 Nastavení preferencí

2. Vytvoříme geometrii (Aplikační menu, záložka Geometry).



Obr. 3 Aplikační menu, Záložka Geometry

Máme dvě základní možnosti jak vybrat jednotlivé příkazy. Přes aplikační menu, přes aplikační formulář (po vybrání záložky Geometry), nebo nejčastěji jejich kombinací. Obě možnosti obsahují stejnou nabídku – Action, Object a Method. Dále zadáme požadované hodnoty do aplikačního formuláře.

V prvním kroku vytvoříme body (Point), které v dalším kroku spojíme pomocí čar (Curve).



Obr. 4 Aplikační formulář pro vytvoření bodu (Point)

Popis Aplikačního formuláře:

- **Point ID List**, volíme pořadové číslo bodu. Každý bod má rozdílné pořadové číslo, defaultně se vždy zvětší o 1. Místo čísla zde můžeme uvést znak #, který zvolí pořadové číslo nejbližší vyšší než nejvyšší již vytvořený bod.
- **Refer. Coordinate Frame** volíme souřadný systém.
- Přepínač **Auto Execute** automaticky po vyplnění formuláře provede příkaz. V řadě případů je výhodnější vypnout, aby se příkaz neprováděl automaticky několikrát po sobě.
- **Point Coordinate List** – zadáme souřadnice bodu, který chceme vytvořit. Bod je definován v hranatých závorkách, např. [0 0 0], [0, 10, 0], [0 , 20.5, 0].
- **Apply** – provedení příkazu.

V příkazovém řádku se po provedení vypíše provedený příkaz v **PCL jazyku**, viz obr.5 a jeho popis. V případě chybného zadání se zde také objeví červeně chyba – Error s jednoduchým popisem.

```

## Synchronizing the User Data Manager
set_current_dir("F:\Pracovni\Jarek\Granty\MKP1\Vymazat")
## The current working directory has been changed to the database location shown.
ga_viewport_size_set("default_viewport", 8884514, 7204785, 1)
STRING asm_create_grid_xyz_created_ids[VIRTUAL]
asm_const_grid_xyz("#", "[0 0 0]", "Coord 0", asm_create_grid_xyz_created_ids )
## 1 Point created: Point 1

asm_const_grid_xyz("2", "[0, 10, 0]", "Coord 0", asm_create_grid_xyz_created_ids )
## 1 Point created: Point 2
asm_const_grid_xyz("0", "[ , 20.5, ]", "Coord 0", asm_create_grid_xyz_created_ids )
## Error reported from application APP
## Unable to process list: '[ , 20.5, ]'
asm_const_grid_xyz("3", "[0, 20.5, 0]", "Coord 0", asm_create_grid_xyz_created_ids )
## 1 Point created: Point 3

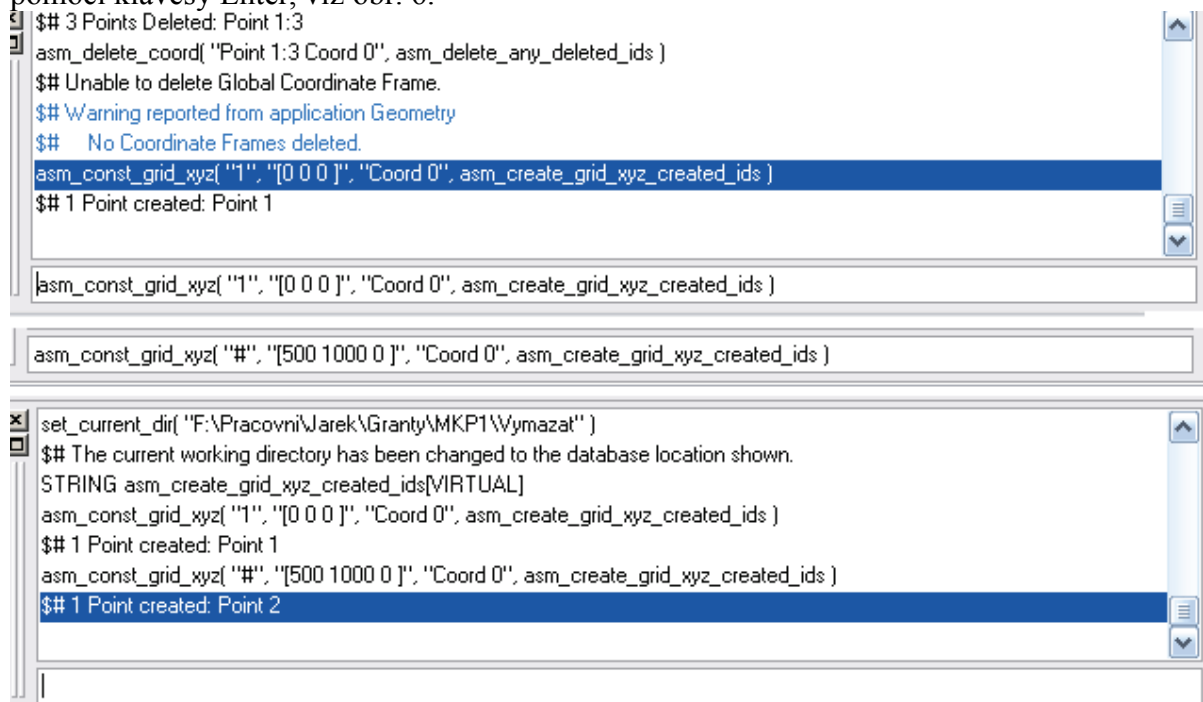
```

Obr. 5 Vytvoření bodu v příkazovém řádku.

Často se objeví také modře varování – Warning. Pomocí příkazového řádku můžeme příkaz zopakovat. Kliknutím na okno historie se příkaz zkopíruje do příkazového řádku. Po korekci

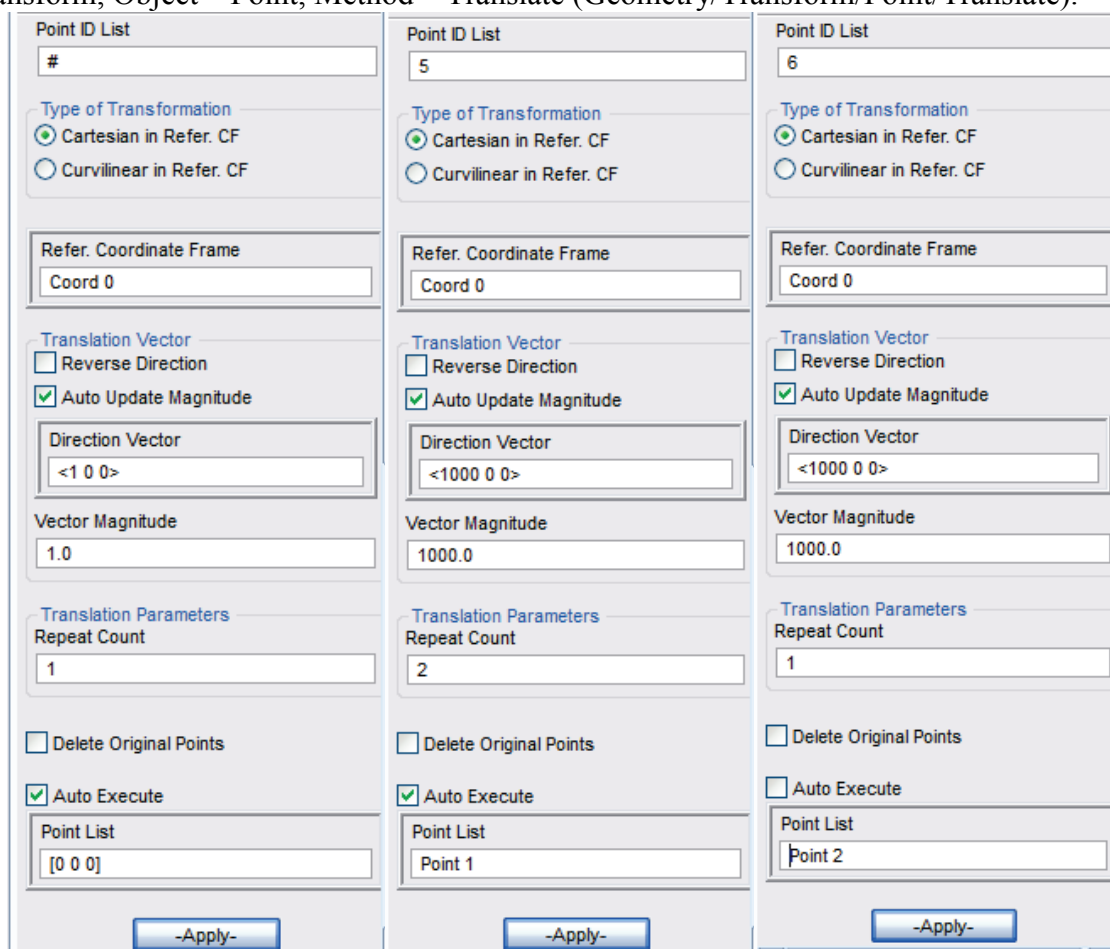


příkazu – změna označení bodu 1 na 2 nebo # a zadání nových souřadnic spustíme příkaz pomocí klávesy Enter, viz obr. 6.



Obr. 6 Práce s oknem historie a příkazovým řádkem.

Vytvořené body zkopírujeme pomocí příkazu záložka – Geometry, Action – Transform, Object – Point, Method – Translate (Geometry/Transform/Point/Translate).



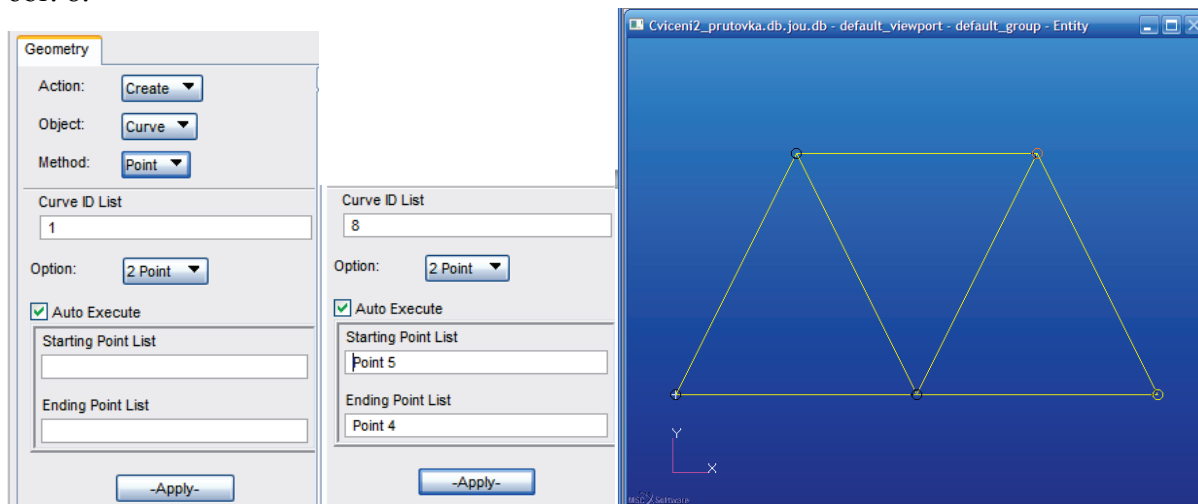
Obr. 7 Aplikační formulář pro kopírování (Transform/Point/Translate).



Popis Aplikačního formuláře:

- **Point ID List**, volíme pořadové číslo bodu (číslo nebo #).
- **Type of Transformation** – volíme souřadný systém. Volba Cartesian in Refer. CF znamená, že Translation Vector bude aplikován v Referenčním souřadném systému ve směru os x, y, z. Volba Curvilinear in Refer. CF znamená, že Translation Vector bude aplikován ve válcovém (cylindrický) souřadném systému R, φ , z, nebo kulovém (sférickém) souřadném systému.
- **Refer. Coordinate Frame** volíme souřadný systém.
- **Reverse Direction** – Přepínač na obrácení směru vektoru.
- **Auto Update Magnitude** – Přepínač na automatický výpočet posuvné vzdálenosti (Vector Magnitude) z vektoru směru (Direction Vector).
- **Direction Vector** – směr ve kterém se generují nové body. Zadává se ve tvaru $\langle 1\ 0\ 0 \rangle$, $\langle 1, 0, 0 \rangle$ apod.
- **Vector Magnitude** – Posuvná vzdálenost (dopočítává se z vektoru směru při zapnutí přepínače)
- **Repeat Coun** – počet kopií, které se mají vytvořit.
- **Delete Original Points** – přepínač při zapnutí smaže původní bod (Point).
- Přepínač **Auto Execute** automaticky po vyplnění formuláře provede příkaz.
- **Point list** – bod nebo body, které mají být kopírovány. Zadává se ve tvaru, [0 0 0], Point 1; Point 1,3; Point 1:3 apod.

Jednotlivé body (Point) spojíme pomocí čar (Curve). Příkaz Geometry/Create/Curve/Point viz obr. 8.



Obr. 8 Aplikační formulář pro kopírování (Transform/Point/Translate), model vytvořený z čar.

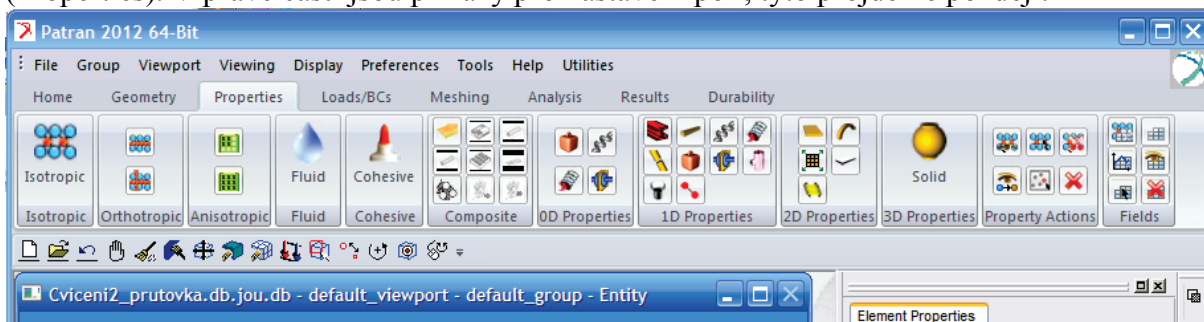
Popis Aplikačního formuláře:

- **Curve ID List**, volíme pořadové číslo čáry (číslo nebo #).
- **Option** – vybíráme počet bodů, které chceme proložit čarou (2 Point, 3 Point, 4 Point).
- Přepínač **Auto Execute** automaticky po vyplnění formuláře provede příkaz.
- **Starting Point List** – První bod čáry. Zdává se ve tvaru Point 1.
- **Ending Point List** – Poslední bod čáry Zdává se ve tvaru Point 2.

Body mezi kterými chceme vytvořit přímku (křivku) můžeme vybrat také pomocí myši.

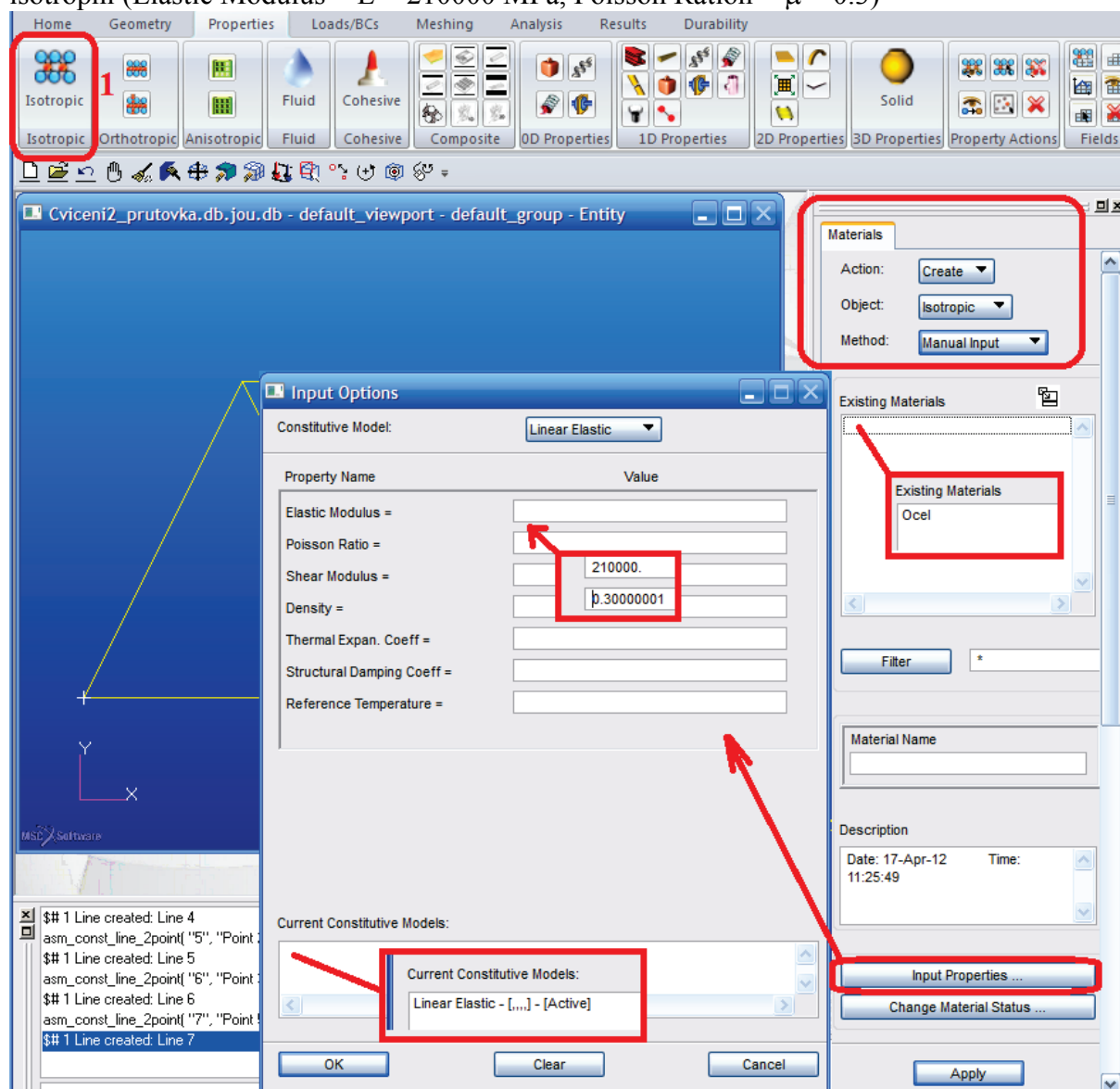


Dalším krokem bude vytvoření vlastností – Properties. Aplikací menu – záložka Properties. Po vybrání záložky Properties se nabídka v aplikacím menu změní, viz obr. 9. V levé části aplikacního menu jsou nabídky materiálů, uprostřed nastavení vlastností (Properties). V pravé části jsou příkazy pro nastavení polí, tyto projdeme později.



Obr. 9 Nabídka Aplikacního menu – Properties.

Naše prutová konstrukce bude vyrobena z oceli, u níž předpokládáme chování homogenní a isotropní (Elastic Modulus = $E = 210000$ MPa, Poisson Ratio = $\mu = 0.3$)

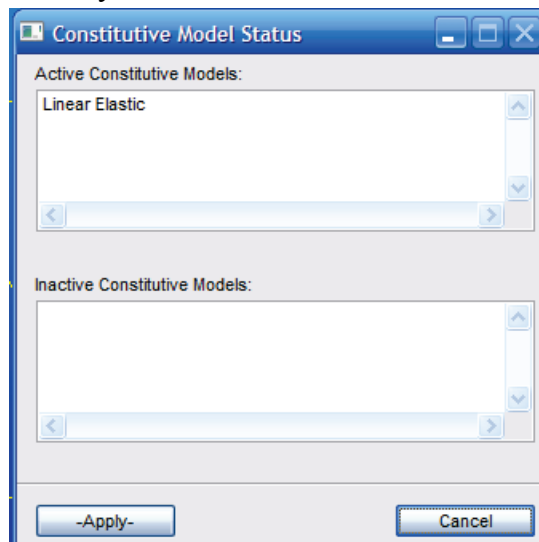


Obr. 10 Volba materiálu

Popis Aplikacního formuláře:



- **Existing Materials**, vidíme zde všechny již vytvořené materiály.
- **Filter** – filtr na redukce a zjednodušení výběru materiálu
- **Material Name** – jméno materiálu, který chceme vytvořit. Musíme zadat – např. Ocel.
- **Description** – Jednoduchý popis vybraného materiálu. Automaticky datum a čas, zbytek můžeme dopsat.
- **Input Properties** – Tabulka pro vkládání hodnot materiálových parametrů
 - **Constitutive model** – nabídka materiálových modelů (Linear elastic, Nonlinear elastic, Hyperelastic, Elastoplastic, Failure, Creep, Viskoelastic atd.).
 - **Property Name** – obsahuje název jednotlivých parametrů a okno pro zadání hodnoty (Elastic Modulus – modul pružnosti v tahu, Poisson Ratio – Poissonovo číslo, Shear Modulus – modul pružnosti ve smyku, Density – hustota atd.)
 - **Current Constitutive Models** – obsahuje vybraný model a jeho status.
- **Change Material Status ...** - zde můžeme aktivovat, deaktivovat jednotlivé konstituční modely.



Obr. 11 Aktivace materiálových konstitučních modelů.

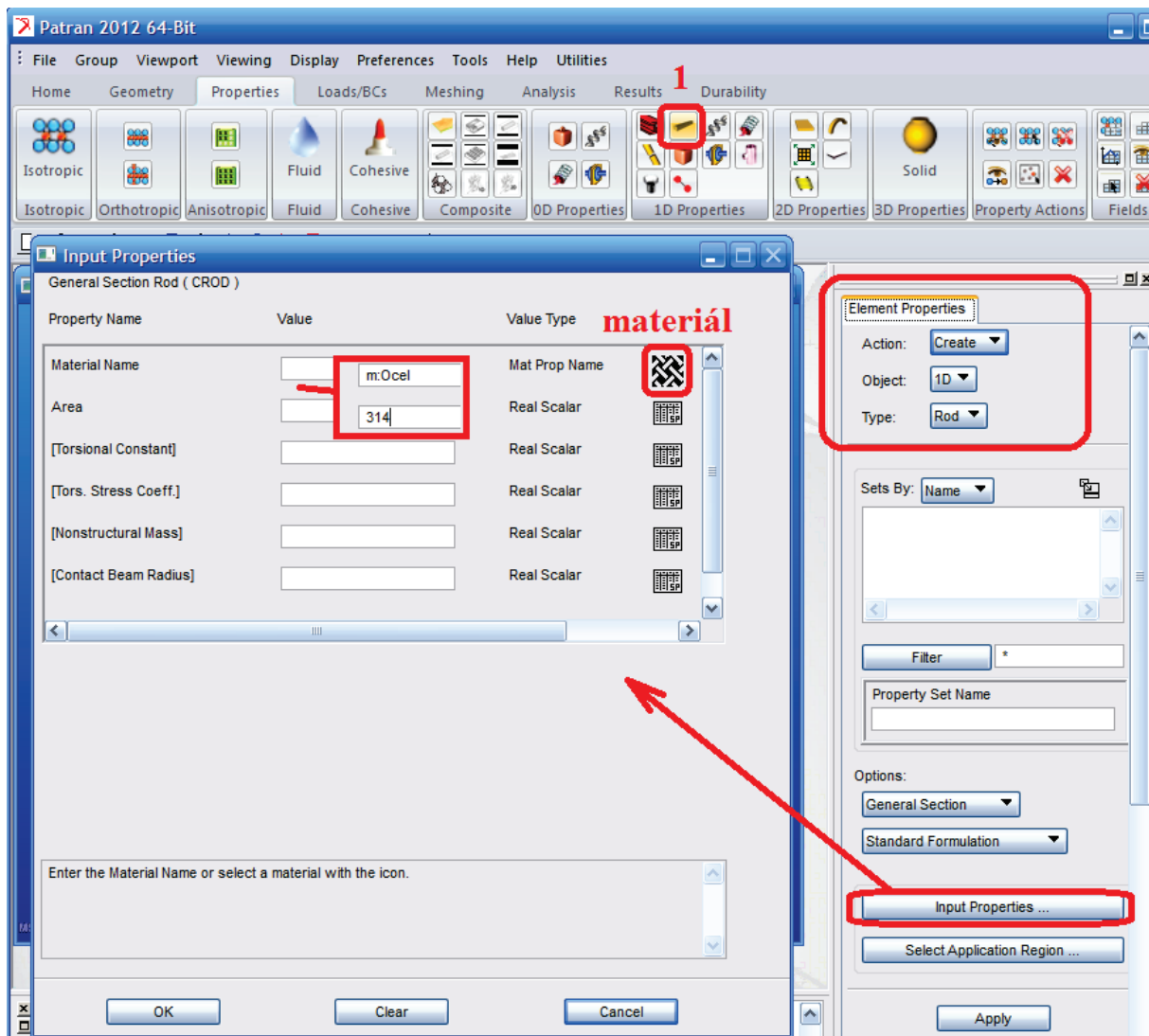
Dále vytvoříme „Element Properties“. Aplikační menu – záložka Properties. Pro naši úlohu se nejlépe hodí 1D prvky – vybereme 1D Properties – typ prvku Rod. Tento prvek je možné použít při zatěžování tahem-tlakem a krutem.

Popis Aplikačního formuláře:

- **Sets By /Name** - vidíme zde všechny již vytvořené materiály. Name reprezentuje typ řazení – dle jména.
- **Filter** – filtr na redukce a zjednodušení výběru materiálu
- **Property Set Name** – Spojuje tvar průřezu s materiálem. Musíme zadat – např. Ocel_tyc.
- **Options – General Section** – další volba. V tomto případě výběr klasické tyče a trubky.
- **Options – Standart Formulation** - Ovlivňuje počet a typ parametrů, které dále nastavujeme.

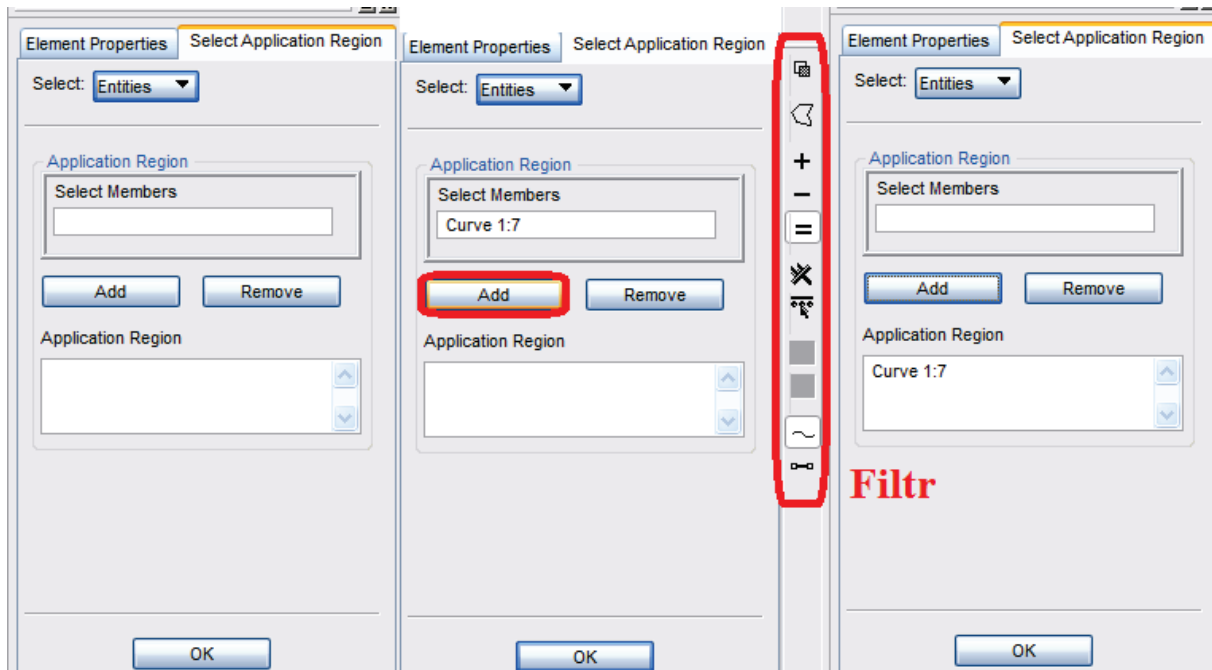


- **Input Properties** – Tabulka pro vkládání hodnot
 - **General Section Rod (CROD)** – typ – název elementu
 - **Property name** – obsahuje tabulku pro zadání jednotlivých parametrů
 - **Material Name** – materiál vybíráme z nabídky již vytvořených materiálů
 - **Area** – plocha průřezu, dle zadaného průřezu (např R=10mm tedy plocha průřezu $S = 314 \text{ mm}^2$).
 - **Torsional Constant** – torzní tuhost, nebudeme vyplňovat.
 - **Atd.** v našem případě nebudeme vyplňovat.
- **Select Application Region** – Výběr oblastí, kde budou tyto vlastnosti platit. Je možné vybrat geometrii, nebo elementy. V našem případě jsme ještě síť nevytvořili, vybereme tedy geometrii (curve) viz obr.12.



Obr. 11 Zadání Element Properties.





Obr. 12 Selekcce – výběr entit

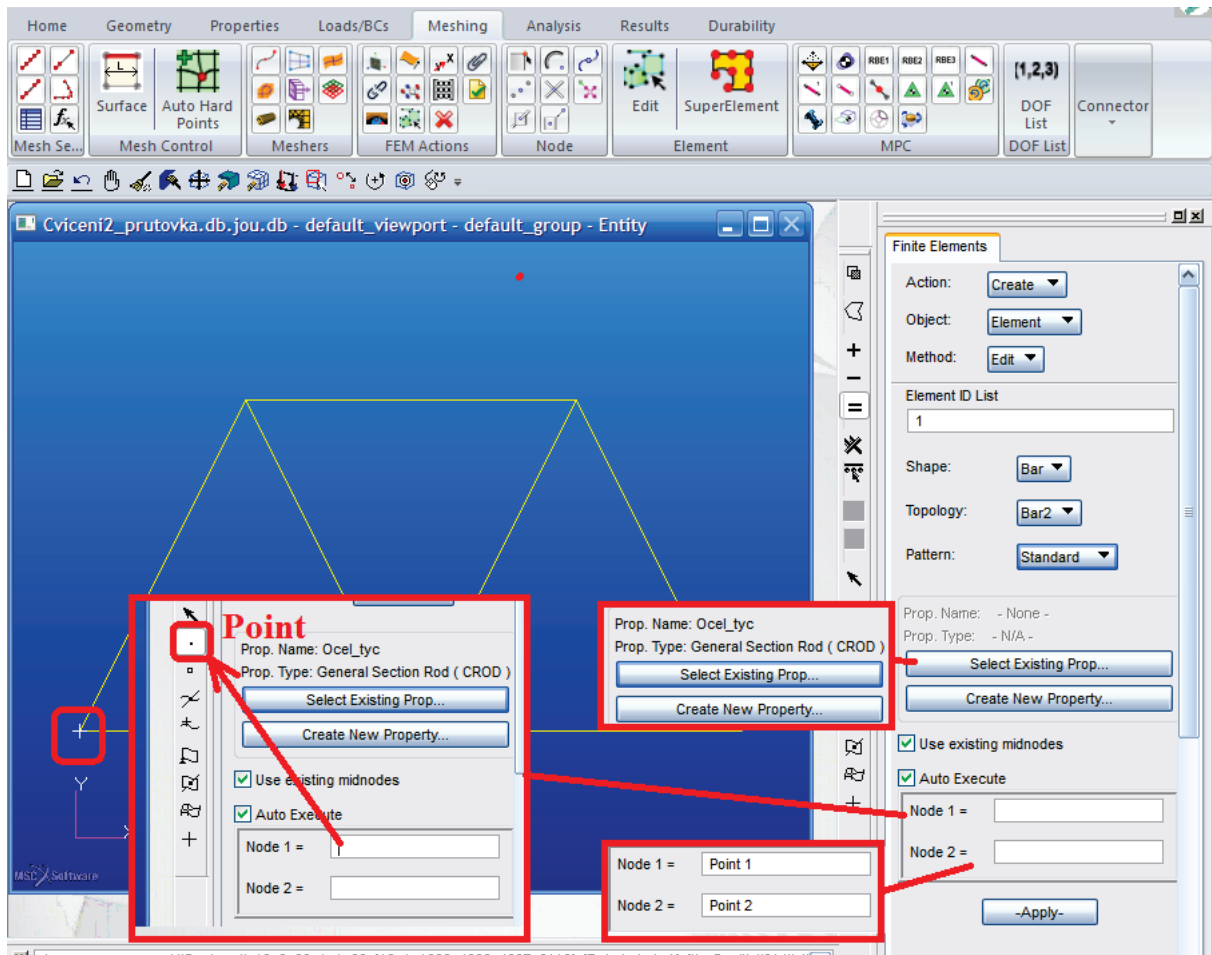
Popis Aplikačního formuláře:

- **Select** – způsob výběru – Entites (body, čáry, plochy, objemy) nebo Regions (Groups). Podrobněji projdeme později.
- **Select Memebers**– vybrané entity (přidáme do Application Region pomocí Add). K výběru jednotlivých entit slouží také Selekční Filtr.
- **Application Region** – entity na které bude aplikován nadřazený příkaz (Element properties).

Dále vytvoříme síť konečných prvků - Meshing. Aplikační menu – záložka Meshing. Ačkoliv jsme si vytvořili model z přímek (curve) vytvoříme přímo elementy – Meshing/Create/Element/Edit viz obr. 13.

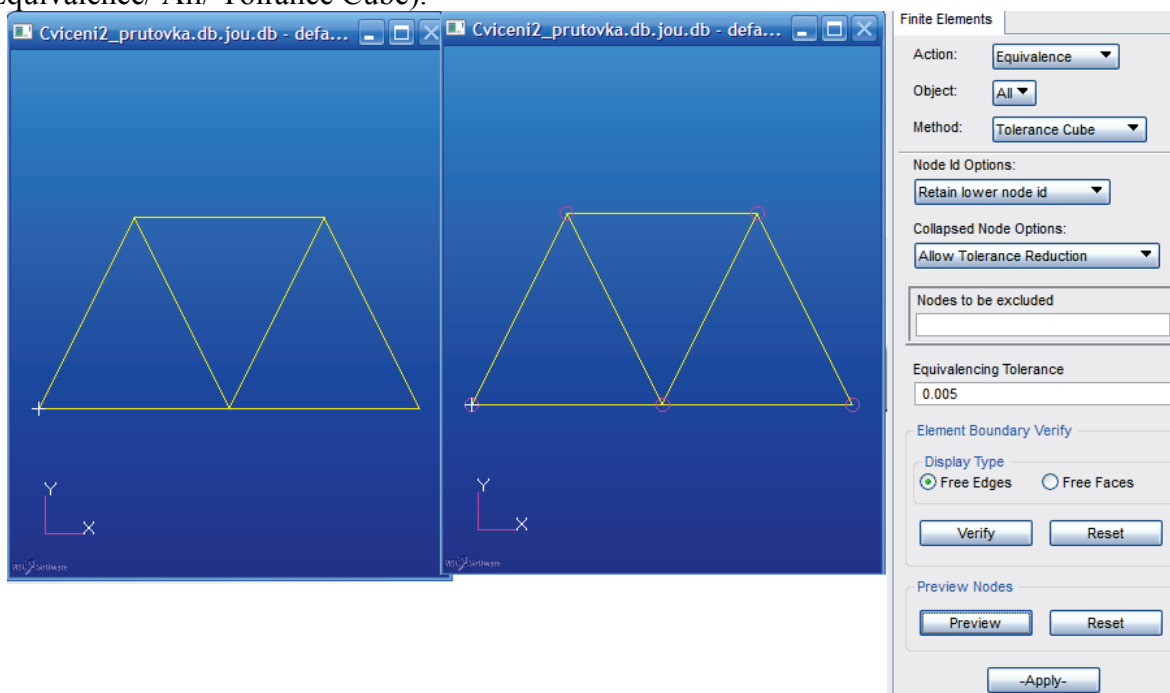
Popis Aplikačního formuláře:

- **Element ID List**, volíme pořadové číslo elementu (číslo nebo #).
- **Shape** – týká se vytvářeného elementu
- **Topology** – týká se vytvářeného elementu
- **Pattern** – týká se vytvářeného elementu
- **Select Existing Prop...** - výběr vlastností, vybereme Ocel_tyc.
- **Create New Property ..** – Vytvoření nových vlastností.
- **Use existing midnodes** – řídí vytváření meziuzlů.
- Přepínač **Auto Execute** automaticky po vyplnění formuláře provede příkaz.
- **Node 1** - první uzel (bod) elementu. Při vytváření elementů musíme přenastavit selekční filtr. V modelu máme zatím pouze body (Point) ne uzly (Node). Přenastavíme tedy filtr a elementy vytvoříme z bodů (Point), viz obr. 13.
- **Node 2** - druhý uzel (bod) elementu



Obr. 13 Vytvoření elementů

Vytvoříme 7 elementů s vlastnostmi Ocel_tyc – jednotlivé pruty prutové soustavy. Při vytváření mohly v jednotlivých bodech vzniknout 2 uzly, tyto musíme spojit (Finite Element/ Equivalence/ All/ Tolerance Cube).



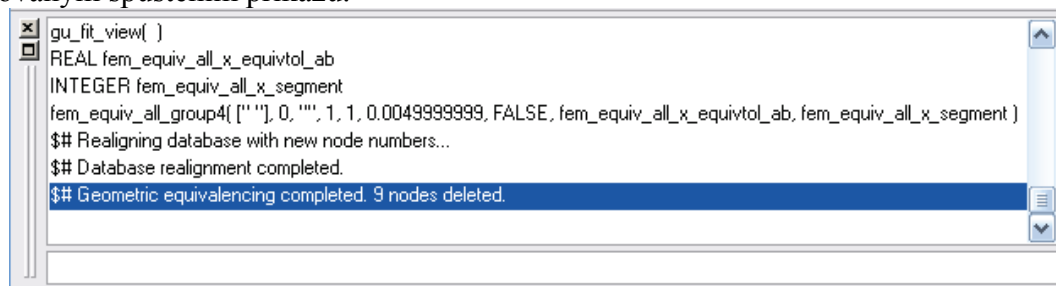
Obr. 14 Korekce sítě, Preview Nodes zobrazí uzly určené k sloučení.



Popis Aplikačního formuláře:

- **Node id Options** – nastavení, který uzel ze spojovaných se zachová, který zruší.
- **Collapsed Node Options** – Nastavení kontroly a chování uzlů a elementů.
 - **Allow Tolerance Reduction** – redukuje toleranci, aby byla větší než je hrana elementu.
 - **Allow Collapsed Elements** – dovolí redukci i přes chyby (collapsed) elementů
 - **Stop on Collapsed Elements** – zastaví se při chybě (collapsed) elementů.
- **Nodes to be excluded** – uzly, které budou vyjmuty z kontroly.
- **Equivalencing Tolerance** – vzdálenost. Jsou-li uzly blíže, program se je pokusí spojit.
- **Element boudary verify – Free Edges** – přepínač na zobrazení hrany, na nichž jsou nespojitě elementy.
- **Element boudary verify – Free Faces** – přepínač na zobrazení plochy, na nichž jsou nespojitě elementy.
- **Element boudary verify – Verify** – tlačítko provede verifikaci a zobrazí výsledky
- **Element boudary verify – Reset** – tlačítko provede reset (vynulování) verifikace
- **Preview Nodes – Verify** – tlačítko provede verifikaci a zobrazí výsledky – uzly, které odpovídají definici a budou se spojovat
- **Preview Nodes – Reset** – tlačítko provede reset (vynulování) verifikace
- **Apply** - Provede sloučení uzlů

Výsledek sloučení je vhodné zkontrolovat např. pomocí okna historie, viz obr. 15, nebo opakovaným spuštěním příkazu.



Obr. 15 Informace o sloučení v okně historie.

Nyní přidáme okrajové podmínky (aplikační menu, záložka Loads/BCs). Přidáme posuvy do jednotlivých uzlů dle zadání (Loads/BCs / Create/Displacement/Nodal).

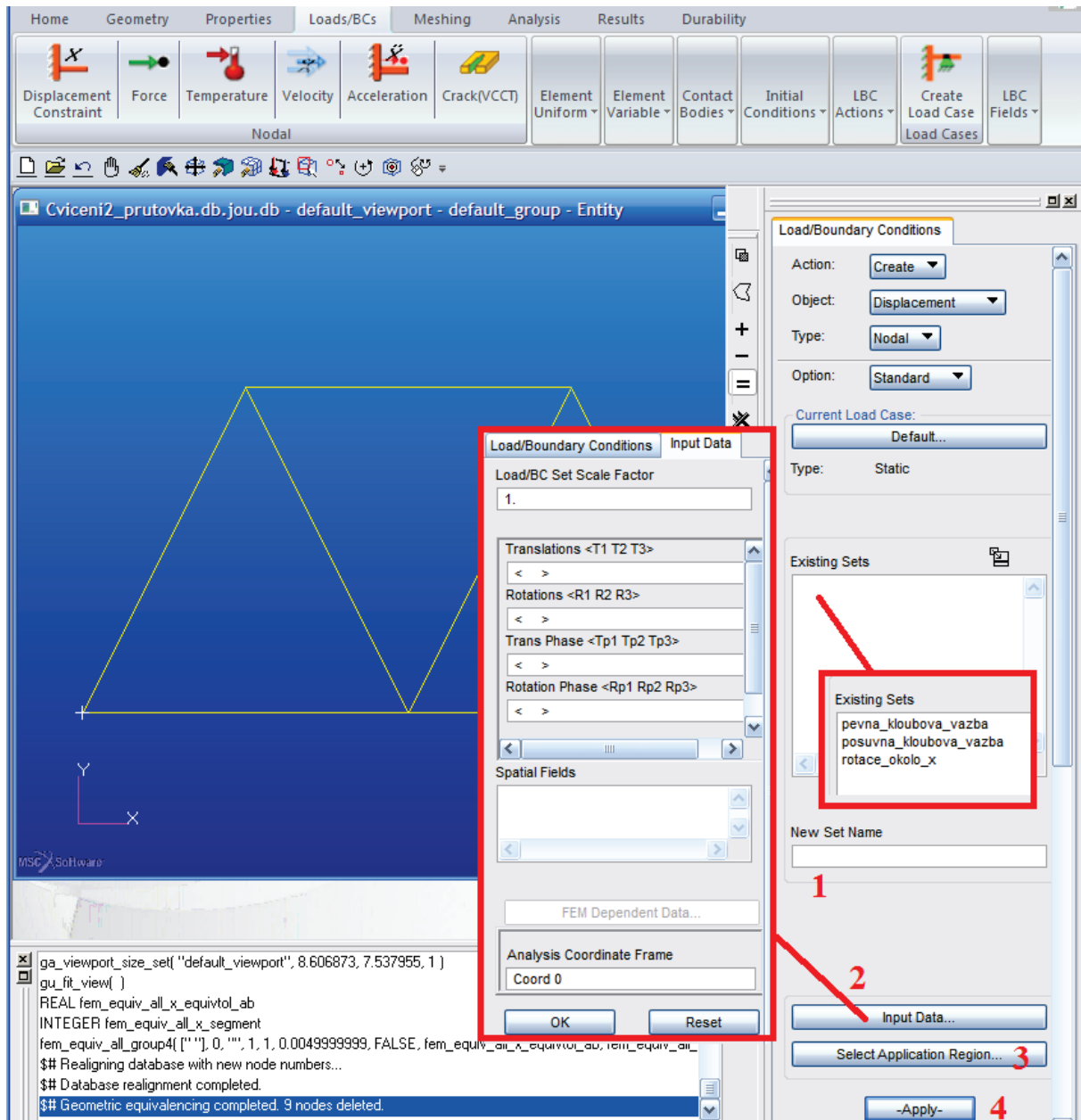
Popis Aplikačního formuláře:

- **Option** – nastavení typu vazby (pro náš případ neměníme).
- **Current Load Case** – Výběr aktuálního zatěžovacího stavu (pro náš případ neměníme).
- **Existing Sets** – tabulka obsahující všechny vytvořené vazby.



- **New Set Name** – jméno vytvářené vazby např. pevna_kloubova, posuvna_kloubova apod. Musíme zadat.
- **Input Data** – formulář na zadání posuvů.
 - **Load/BC Set Scale Factor** – velikostní faktor
 - **Translations < T1 T2 T3 >** - posuvy v jednotlivých směrech os např. < 0 0 0 >, < , , 0 >, < 5 , , 0 >.
 - **Rotations < R1 R2 R3 >** - rotace okolo jednotlivých os v případě, že to má pro daný typ prvku smysl. Definice je stejná jako u posuvů.
 - **Trans Phase < Tp1 Tp2 Tp3 >** - fázový posuv posunutí při nesoufázovém zatěžování. Zadáváme úhly.
 - **Rotation Phase < Rp1 Rp2 Rp3 >** - fázový posuv rotace při nesoufázovém zatěžování. Zadáváme úhly.
 - **Spatial Fields** – okrajové podmínky zadané pomocí tabulky.
 - **FEM Depentd Data** – okrajové podmínky zadávané pomocí tabulky.
 - **Analysis Coordinate Frane** – definuje souřadný systém.
- **Selekt Application Region** – vybereme místo, kde bude okrajová podmínka aplikovaná. Používáme selekční filtr.
- **Apply** – vytvoří nový BC Set





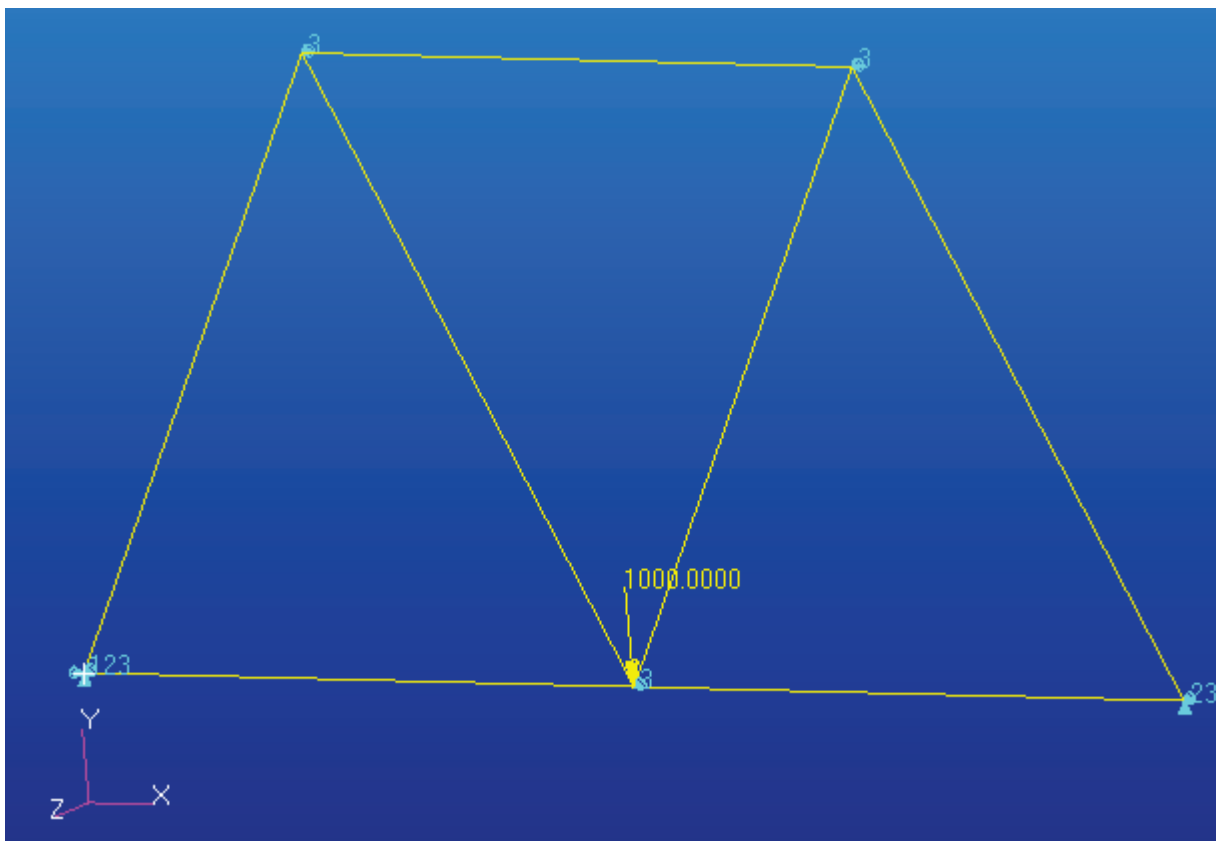
Obr. 16 Zadání posunutí do uzlů.

Přidáme zatížení (síly nebo momenty) do jednotlivých uzlů dle zadání (Create/Force /Nodal).
Popis Aplikačního formuláře:

- **Current Load Case** – Výběr aktuálního zatěžovacího stavu (pro náš případ neměníme).
- **Existing Sets** – tabulka obsahující všechny vytvořené vazby.
- **New Set Name** – jméno vytvářené síly např. síla apod. Musíme zadat.
- **Input Data** – formulář na zadání sil (obdobné jako u posuvů).
 - **Load/BC Set Scale Factor** – velikostní faktor
 - **Force < F1 F2 F3 >** - síly v jednotlivých směrech os např. < 1000 0 0 >, < , ,100 >, < 500 , ,0 >.
 - **Moment < M1 M2 M3 >** - Momenty



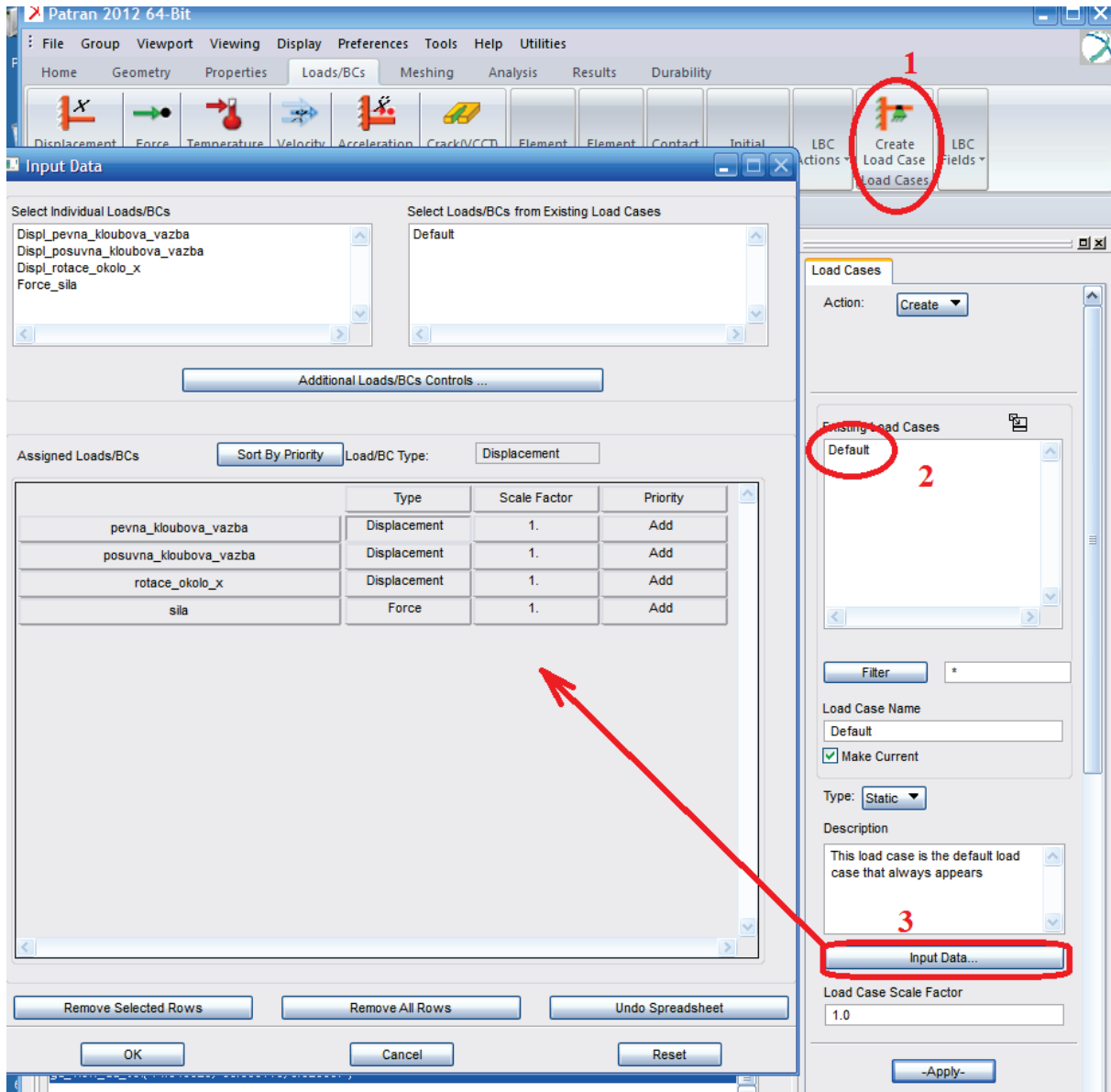
- **Force Phase < Fp1 Fp2 Fp3 >** - fázový posuv sil při nesoufázovém zatěžování. Zadáváme úhly.
- **Moment Phase < Mp1 Mp2 Mp3 >** - fázový posuv momentů při nesoufázovém zatěžování. Zadáváme úhly.
- **Spatial Fields** – okrajové podmínky zadané pomocí tabulky.
- **FEM Depentd Data** – okrajové podmínky zadávané pomocí tabulky.
- **Analysis Coordinate Frane** – definuje souřadný systém.
- **Selekt Application Region** – vybereme místo, kde bude okrajová podmínka aplikovaná. Používáme selekční filtr.
- **Apply** – vytvoří nový BC Set



Obr. 17 Okrajové podmínky v grafickém okně.

Nyní zkontrolujeme, zda se všechny okrajové podmínky uložily do správné množiny – Load Cases. V tomto příkladu jsme využili defaultně vytvořené množiny – jménem Default viz. Obr. 18.





Obr. 18 Okrajové podmínky v Load Cases.

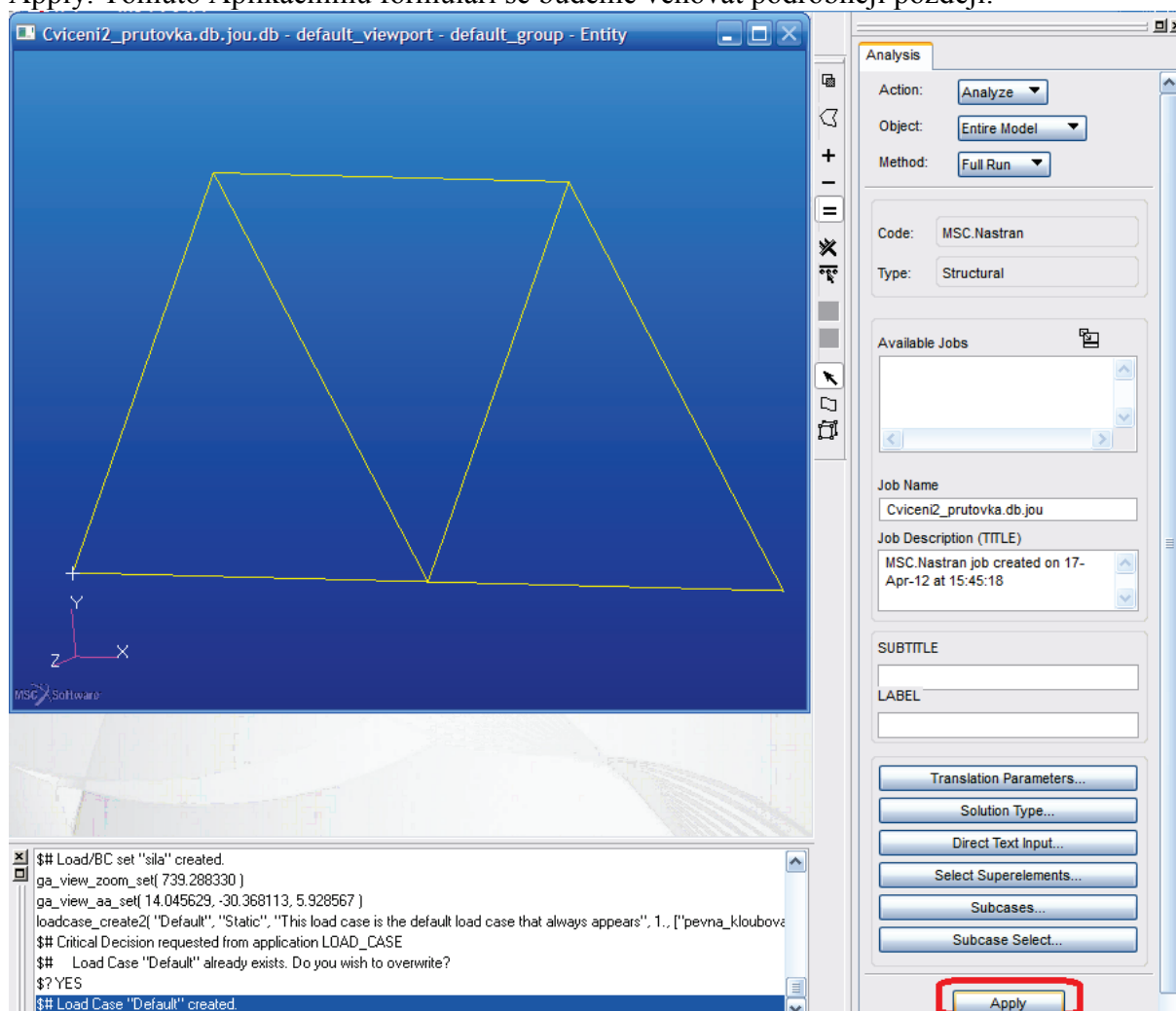
Popis Aplikačního formuláře:

- **Existing Load Cases** – tabulka obsahující všechny vytvořené množiny.
- **Filter** – filtr na redukce a zjednodušení výběru
- **Load Case Name** – jméno vytvářené množiny vazeb např. Default. Musíme zadat nebo vybrat.
- **Make Current** – vybraná množina (Load Case) je aktivní (pracuje se s ní).
- **Type** – typ úlohy.
- **Description** – popis množiny vazeb (Load Case).
- **Input Data** – formulář pro výběr jednotlivých okrajových podmínek a zatížení.
 - **Select Individual Loads/BCs** – všechna zatížení a BC.
 - **Select Loads/BCs from Existing Load Cases** – všechny množiny vazeb (Load Case)
 - **Additional Loads/BCs Controls ...** - možnost nastavení poměrů mezi jednotlivými okrajovými podmínkami/ zatíženími.



- **Assigned Loads/BCs** – okrajové podmínky a zatížení patřící do vybrané množiny vazeb (Load Case).
- **Remove Selected Rows** – vymazání vybraného řádku (okrajové podmínky) z vybrané množiny vazeb (Load Case)
- **Remove all Rows** - vymazání všech řádků z vybrané množiny vazeb (Load Case)
- **Undo Spreadsheet** – krok zpět
- **OK** – potvrzení změn.
- **Load Case Scale Factor** – velikostní faktor
- **Apply** – vytvoří nové Load Case, nebo přepíše původní.

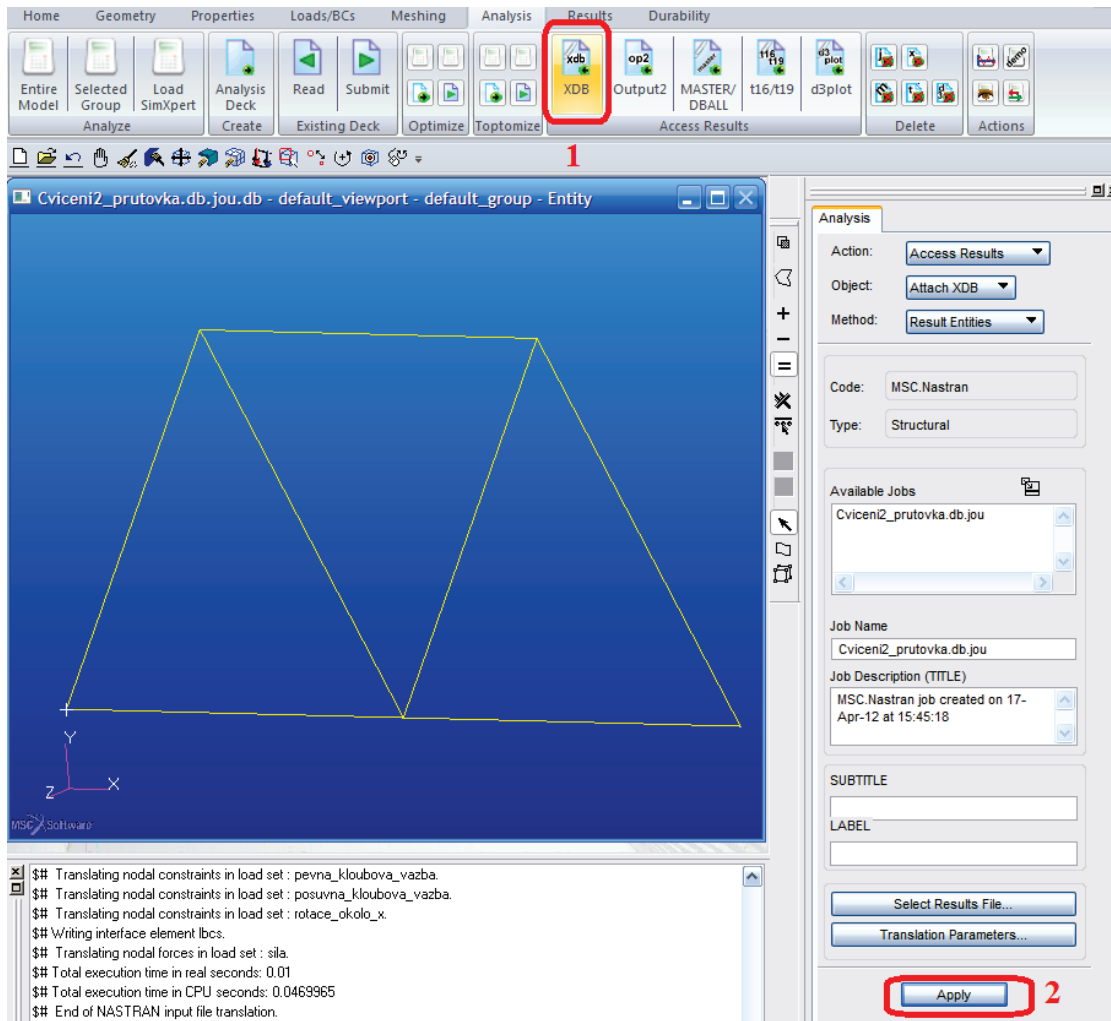
Tímto jsme vše připravili k výpočtu. Přejdeme v aplikačním menu do záložky Analysis. V tomto jednoduchém příkladu je zde již vše přednastaveno a můžeme rovnou spustit řešení Apply. Tomuto Aplikačnímu formuláři se budeme věnovat podrobněji později.



Obr. 19 Řešení úlohy.

Patran následně převede úlohu do souboru .bdf a spustí řešení ve vybraném řešiči – MSC Nastran. Po ukončení řešení uloží výsledky do souboru .xdb. O ukončení řešení nás informuje výpis v okně historie, zvukový signál atd. Informace o řešení, jeho průběhu, chybách je možno najít v souborech s příponou .f06, .f04, .log. Výsledky řešení pak načteme do patranu (.xdb), viz obr.20.

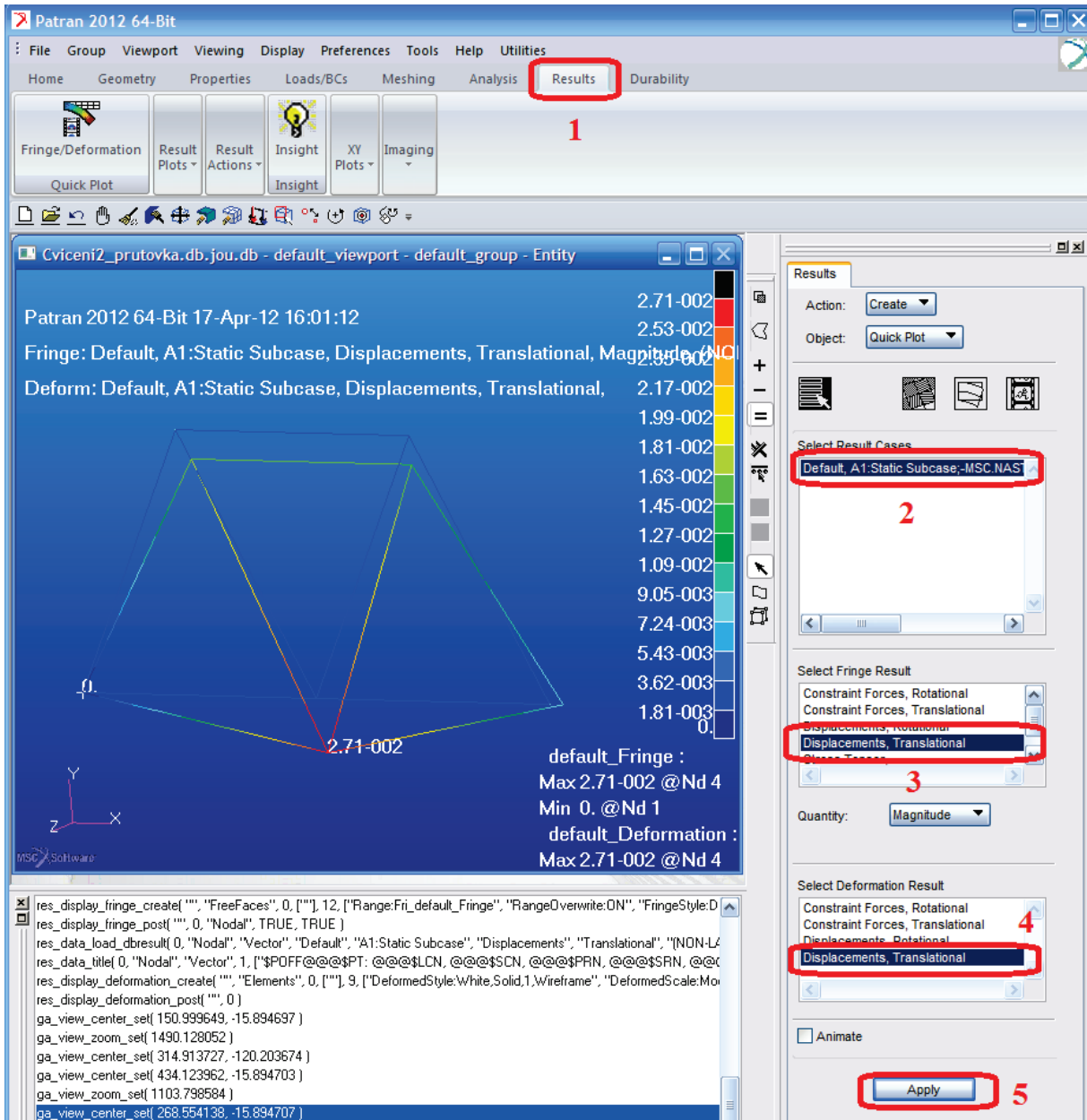




Obr. 20 Načtení výsledků do Patranu.

K prohlížení výsledků používáme v aplikačním menu do záložku Results. Vybereme krok řešení a požadované výsledky a pomocí Apply zobrazíme viz. obr.21 Tomuto Aplikačnímu formuláři se budeme věnovat podrobněji později.





Obr. 21 Výsledky řešení.



2 LITERATURA

LENERT, J. Úvod do metody konečných prvků. skripta VŠB-TU Ostrava, 1999.

KOLÁŘ, V., NĚMEC, I., KANICKÝ, V. FEM Principy a praxe metody konečných prvků. Computer Press, 1997.

VALCHÁŘOVÁ, J. Soudobé numerické metody v mechanice kontinua. SNTL. Praha. 1986.

BEER, G., WATSON, J.O. Introduction to Finite and Boundary Element Methods for Engineers. New York. 1992.

BITTNAR, Z., ŠEJNOHA, J. Numerické metody mechaniky 1. Vydavatelství ČVUT. Praha. 1992. ISBN 80-01-00855-X.

BITTNAR, Z., ŠEJNOHA, J. Numerické metody mechaniky 2. Vydavatelství ČVUT. Praha. 1992. ISBN 80-01-00901-7.

ZIENKIEWICZ, O., C., TAYLOR, R., L. The Finite Element Method, Fifth edition. Butterworth-Heinemann. Oxford. 2000. ISBN 0-7506-5049-4.

NEČAS, J., HLAVÁČEK, I. Úvod do matematické teorii pružných a pružně plastických těles. SNTL. Praha. 1983.

Internet:

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

<http://www.colorado.edu/engineering/cas/courses.d/IFEM.d/>

