

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

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



METODA KONEČNÝCH PRVKŮ I

11. ŘEŠENÍ SOUSTAVY ROVNIC

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	ŘEŠENÍ SOUSTAVY ROVNIC	3
1.1	Globální matice tuhosti systému.....	4
1.2	Způsoby řešení soustavy lineárních rovnic.....	4
1.2.1	Přímé řešiče.....	5
1.2.2	Iterační řešiče.....	6
1.3	Volba řešiče.....	7
1.4	Řešení pomoci rozkladu na podoblasti	8
1.5	Cvičení.....	10
2	LITERATURA	18



1 ŘEŠENÍ SOUSTAVY ROVNIC



OBSAH KAPITOLY

V přednášce bude probrána problematika řešení soustav rovnic generovaných při užití metody konečných prvků. Na začátku budou probrány vlastnosti globální matice tuhosti a tím pádem i celého systému řešených rovnic. Seznámíme se s problematikou přímých a iteračních řešičů. Dále bude naznačena i problematika řešení pomocí rozkladu úlohy na podoblasti.



MOTIVACE:

Soustavy lineárních algebraických rovnic, které vznikají v případě užití metody konečných prvků, jsou charakterizovány vysokým řádem a výrazným pásovým uspořádáním. V úlohách mechaniky kontinua pružného tělesa je matice soustavy řešeného systému symetrická a pozitivně definitní. Pro řešení takovýchto soustav se užívají specializované algoritmy. Symetrie a pásovitosti matice soustavy se využívá k redukci uchovávaných údajů během výpočtu.

Seznámíme se s metodami řešení soustav lineárních rovnic tak, abychom byli schopni správně volit a využít tak efektivitu jednotlivých metod.



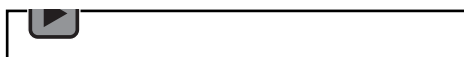
1.1 GLOBÁLNÍ MATICE TUHOSTI SYSTÉMU

Na začátku si uveďme základní vlastnosti globální matice tuhosti. Jedná se o tři následující vlastnosti:

1. Matice je symetrická, tj. $K_{ij} = K_{ji}$.
2. Matice je pásová, tzn. nenulové členy se vyskytují pouze v páse okolo hlavní diagonály.
3. Matice je pozitivně definitní, tj. pro každý vektor $\{x\}$ platí $\{x\}^T [K] \{x\} \geq 0$, přičemž hodnotu nula tento výraz nabyde pouze v případě, že $\{x\} = \{0\}$.



Audio 1.1 Matice tuhosti



První dvě vlastnosti vedou k tomu, že se nemusí v počítači pracovat s celou maticí, ale ukládají se pouze diagonální a naddiagonální prvky ležící v páse nad hlavní diagonálou. Příkladem je například tzv. „skyline“ algoritmus. Práce je (v rámci programu) komplikovanější, na druhou stranu dochází k vysoké úspoře paměti počítače.

1.2 ZPŮSOBY ŘEŠENÍ SOUSTAVY LINEÁRNÍCH ROVNIC

Jak bylo uvedeno v předcházejících kapitolách, přesnost výsledku souvisí s kvalitou vytvořené sítě. Při zahušťování sítě konečných prvků by se mělo řešení blížit k hodnotám exaktního řešení. Na druhou stranu, čím hustší síť, tím delší doba výpočtu. Jak souvisí čas řešení s velikostí elementu?

Odpověď na tuto otázku závisí na mnoho faktorech. Pokud uvažujeme lineární soustavu rovnic, pak procesorový čas pro řešení symetrické, pozitivně definitní soustavy rovnic:

$$[K]\{U\} = \{F\} \quad (11.1)$$

je dán následujícím vztahem:

$$CPU = Cn^\alpha \quad (11.2)$$

kde n je počet neznámých, tj. počet stupňů volnosti celého konečno prvkového modelu. Hodnota C a hodnota α závisí na volbě metody řešení (řešiče) a na tvaru matice tuhosti. Například v případě, že matice soustavy rovnic (globální matice tuhosti) je plná, hodnota exponentu α je rovna 3, pro většinu přímých řešičů. V případě řídké matice (tj. takové, které jsou generovány v metodě konečných prvků) je hodnota α někde mezi 1 a 2, zase v závislosti na zvoleném řešiči a kvalitě matice tuhosti.

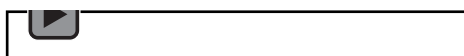
Metod řešení soustav lineárních rovnic můžeme rozdělit v podstatě do dvou základních skupin:

1. přímé metody (finitní)
2. iterační metody

Přímé metody jsou v podstatě přesné metody, udávající konečný algoritmus pro řešení soustavy lineárních rovnic. Jsou-li všechny vstupní parametry dány přesně a provádí-li se aritmetické operace také přesně, je výsledné řešení také přesné.



Audio 1.2 Metody řešení



Nutno poznamenat, že tyto předpoklady nejsou při použití počítače reálné (uložení čísla v počítači, operace), jsou tyto metody v podstatě metody přibližné. Příkladem přímých metod



je například Gaussova metoda. Většina přímých metod, vhodných pro použití na počítačích, je právě obměnou Gaussovy eliminační metody. Další přímé metody jsou například LU rozklad, Ironsova frontální metoda, Croutova metoda, aj.

Pozn.: Frontální metoda se od ostatních metod liší tím, že matice tuhosti se sestavuje průběžně během řešení soustavy rovnic. Matice tuhosti se nikdy nevyskytuje v paměti celá – úspora prostředků. Vznikla v souvislosti s rozvojem metody konečných prvků.

Iterační metody umožňují řešit soustavy rovnic s předepsanou přesností konvergujících nekonečných procesů.



Audio 1.3 Iterační metody



Konstanta C v rovnici (11.2) je výrazně vyšší pro iterativní řešiče oproti přímým. U exponentu α je to naopak, nižší hodnoty jsou pro iterační řešiče, vyšší pro přímé. Hlavní výhodou přímých metod řešení je jejich robustnost. Jsou ideální pro řešení malých a středně velkých problému. Volba mezi typy řešičů není závislá pouze na charakteru řešeného systému rovnic, ale také například na architektuře hardware na kterém je problém řešen. Například při použití víceprocesorových paralelních strojích je efektivnější použití iteračních metod, které jsou téměř lineárně škálovatelné, oproti přímým řešičům které mají omezenou škálovatelnost.

1.2.1 Přímé řešiče

Gaussova eliminace – Nejefektivnější přímá metoda řešení soustavy lineárních algebraických rovnic. Metodu lze obecně aplikovat na libovolnou soustavu rovnic, efektivita v případě metody konečných prvků souvisí s vlastnostmi matice tuhosti, tj. symetrie, pozitivní definitnost a šířka pásu.

Připomeňme si algoritmus řešení. Gaussova eliminační metoda předpokládá, že matice soustavy (11.1) je regulární. Pro jednoduchost budeme předpokládat, že prvek K_{11} a že v každém kroku je příslušný diagonální prvek různý od nuly. Při procesu se vyskytují dva základní kroky. První krok se nazývá **přímý chod** Gaussovy eliminace (triangulizace), při kterém se provádí úprava rozšířené matice soustavy tak, aby pod diagonálou matice soustavy byly samé nule. Druhá část procesu je tzv. **zpětný chod**, při kterém se vypočítají hledané hodnoty vektoru $\{U\}$.

Řešme soustavu:

$$\begin{bmatrix} K_{11} & K_{12} & \dots & K_{1n} \\ K_{21} & K_{22} & \dots & K_{2n} \\ \dots & \dots & \dots & \dots \\ K_{n1} & K_{n2} & \dots & K_{nn} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ \vdots \\ U_n \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{Bmatrix} \quad (11.3)$$

Postupně upravíme do tvaru:

$$\begin{bmatrix} K_{11}^{(0)} & K_{12}^{(0)} & \dots & K_{1n}^{(0)} \\ 0 & K_{22}^{(0)} & \dots & K_{2n}^{(0)} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & K_{nn}^{(n-1)} \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ \vdots \\ U_n \end{Bmatrix} = \begin{Bmatrix} F_1^{(0)} \\ F_2^{(1)} \\ \vdots \\ F_n^{(n-1)} \end{Bmatrix} \quad (11.4)$$

s nenulovými diagonálními koeficienty. Platí:

$$K_{ij}^{(0)} = K_{ij} \quad F_i^{(0)} = F_i \quad i, j = 1, 2, \dots, n \quad (11.5)$$

$$K_{ij}^{(k)} = K_{ij}^{(k-1)} - \frac{K_{ik}^{(k-1)}}{K_{kk}^{(k-1)}} K_{kj}^{(k-1)} \quad F_i^{(k)} = F_i^{(k-1)} - \frac{K_{ik}^{(k-1)}}{K_{kk}^{(k-1)}} F_k^{(k-1)}$$



$$k = 1, 2, \dots, n$$

$$i, j = k + 1, k + 2, \dots, n$$

Při zpětném chodu Gaussovy eliminace se určí hledané neznámé $U_i, i = 1, 2, \dots, n$ dle následujícího vztahu:

$$U_i = \frac{1}{K_{ii}^{(i-1)}} \left(F_i^{(i-1)} - \sum_{j=i+1}^n K_{ij}^{(i-1)} U_j \right), \quad i = n, n-1, \dots, 1 \quad (11.6)$$

$$\sum_{j=i+1}^n K_{ij}^{(i-1)} U_j = 0 \text{ pro } i \geq n$$

Při řešení bylo uvažováno, že každý příslušný diagonální prvek je nenulový. Tento předpoklad nemusí být obecně splněn. Algoritmus zkolabuje v případě, že $K_{ii} = 0$. (pozn.: Diagonální členy jsou označovány jako tzv. pivoty.) Z regularity matice soustavy vyplývá, že v každém kroku můžeme vhodnou výměnou řádků nebo sloupců soustavy dosáhnout toho, že příslušný diagonální prvek je nenulový.

Jestliže je absolutní hodnota některého z dělitelů $K_{kk}^{(k-1)}$ malá (nebo nulová), ve srovnání s absolutní hodnotou prvků $K_{ik}^{(k-1)}, i > k$, dochází při výpočtu k velkým zaokrouhlovacím chybám. Pro zamezení tohoto problému se proto využívá Gaussova eliminace s částečným nebo úplným výběrem hlavního prvku.

Při částečném výběru hlavního prvku se v každém kroku přímého chodu určí prvek $K_{p,k}^{(k-1)}, p \geq k$ s nejvyšší hodnotou (absolutní) a vymění se mezi sebou $p - \text{tý}$ a $k - \text{tý}$ řádek. Další postup je stejný, tak jak byl uveden výše.

Při úplném výběru hlavního prvku se postupuje tak, že se při přímém chodu určí prvek $K_{p,r}^{(k-1)}, p \geq k, r \geq k$ opět s nejvyšší absolutní hodnotou. Proveďte se výměna $p - \text{tého}$ řádku s $k - \text{tým}$ sloupcem a dále pak výměna $r - \text{tého}$ sloupce $k - \text{tým}$. Současně se musí registrovat záměna neznámých.

Výše byl uveden obecný algoritmus Gaussovy eliminační metody. Pro případ diagonální matice soustavy existují varianty tohoto algoritmu, které jsou výkonnější. Nebudeme je však zde rozebírat.

1.2.2 Iterační řešiče

Při užití iteračních metod se vytváří posloupnost postupných aproximací $\{U\}^{(k)}$, která konverguje k přesnému řešení $\{U\}$. Přesné řešení tedy nahrazujeme některou z aproximací $\{U\}^{(k)}$. Nekonečný konvergenční proces nahrazujeme konečným za současného vzniku chyby.

Jestliže posloupnost postupných aproximací konverguje, mluvíme o konvergenčním iteračním procesu. Důležitou otázkou je, kdy je proces konvergentní a je-li vůbec konvergentní. Konvergenci můžeme zapsat následujícím výrazem:

$$\lim_{k \rightarrow \infty} \{U\}^{(k)} = \{U\} \quad (11.7)$$

Otázka konvergence je u všech iteračních úloh velice důležitá.

Odhad času výpočtu je velice nepřesný. Na rozdíl od přímých metod, kde je počet kroků vedoucích k nalezení řešení přesně znám dopředu, u iteračních metod počet kroků dopředu neznáme.

Samotný iterační proces je charakterizován opakovaným řešením rovnice:

$$\begin{aligned} \{U\}^{(k+1)} &= \{U\}^{(k)} + \Delta\{U\}^{(k)} = \{U\}^{(k)} + [Q]^{-1}\{R\}^{(k)} \\ &= \{U\}^{(k)} + [Q]^{-1}(\{F\} - [K]\{U\}^{(k)}) \end{aligned} \quad (11.8)$$

Jako počáteční odhad se zpravidla volí $\{U\}^{(0)} = \{0\}$. Proces iterací je ukončen v případě, kdy reziduum (nevyrovnané síly) je dostatečně malé vůči celkovému vnějšímu zatížení, tj.



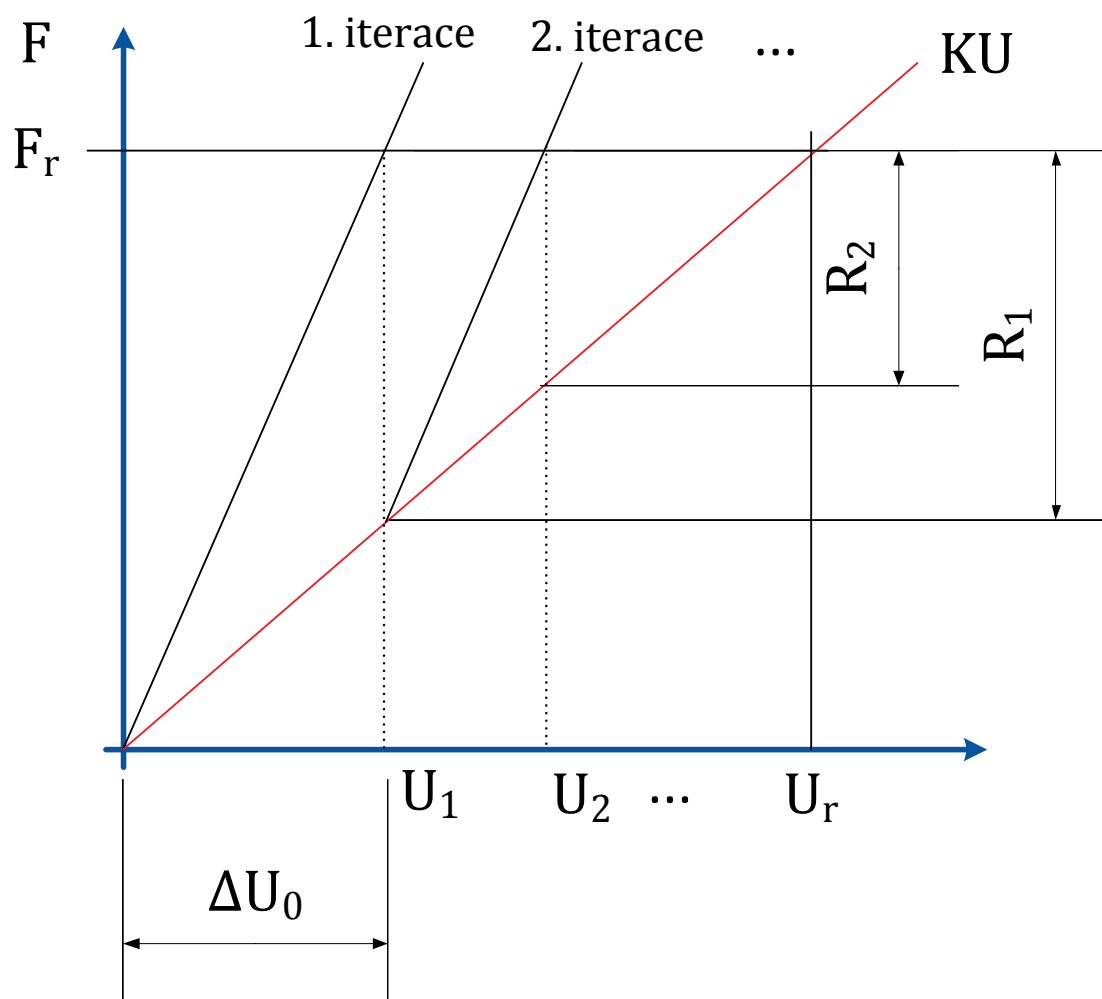
s uspokojivou přesností je dosaženo rovnováhy. Podmínka pro zastavení může nabývat například tvaru:

$$\frac{\{R\}^{(k)T}\{R\}^{(k)}}{\{F\}^{(k)T}\{F\}^{(k)}} \leq \varepsilon_F^2 \quad (11.9)$$

kde ε_F je tolerance, kterou lze ovlivnit rychlost řešení, ale také přesnost řešení.

Jednotlivé iterační metody se liší ve způsobu konstrukce matice $[Q]$ (tzv. *preconditioner*). Většinou se vychází z matice tuhosti soustava. Pro rychlost inverze je nejjednodušší volit $[Q] = \text{diag}([K])$, dostáváme se k tzv. Jacobiho iterační metodě.

Iterační metody mají značný význam, zvláště při řešení velkých soustav rovnic a při nasazení víceprocesorových paralelních počítačů. Existuje velké množství iteračních algoritmů. Výpočetní schémata jsou obvykle jednoduchá. Každý iterační proces má však svá omezení, pro některou soustavu může konvergovat velmi rychle, pro jinou pomalu, nebo dokonce může dojít k divergenci řešení. Graficky znázorněný iterační proces (pro jeden stupeň volnosti) je naznačen na Obr. 1.



Obr. 1 - Schéma iteračního procesu

1.3 VOLBA ŘEŠIČE

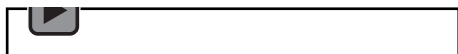
Závisí na velikosti úlohy a použitém hardware počítače. Obecné informace byly uvedeny již na začátku kapitoly. Obecně můžeme říci, že přímé řešení je obvykle spolehlivější a výpočet dospěje s vysokou pravděpodobností k hledanému výsledku s technicky přijatelnou přesností.



Při velkých úlohách však takovýto proces může trvat poměrně dlouho. Naproti tomu iterační řešení může být méně spolehlivé, zejména u hůře podmíněných úloh. Zde může iterační proces konvergovat velmi pomalu, nebo dopočítat výsledek, který je z technického hlediska nepřesný. V jiných případech však může být iterační řešení o mnoho rychlejší než při použití přímých metod a tím pádem může dojít i k řádové úspoře výpočetního času.



Audio 1.4 Volba řešiče

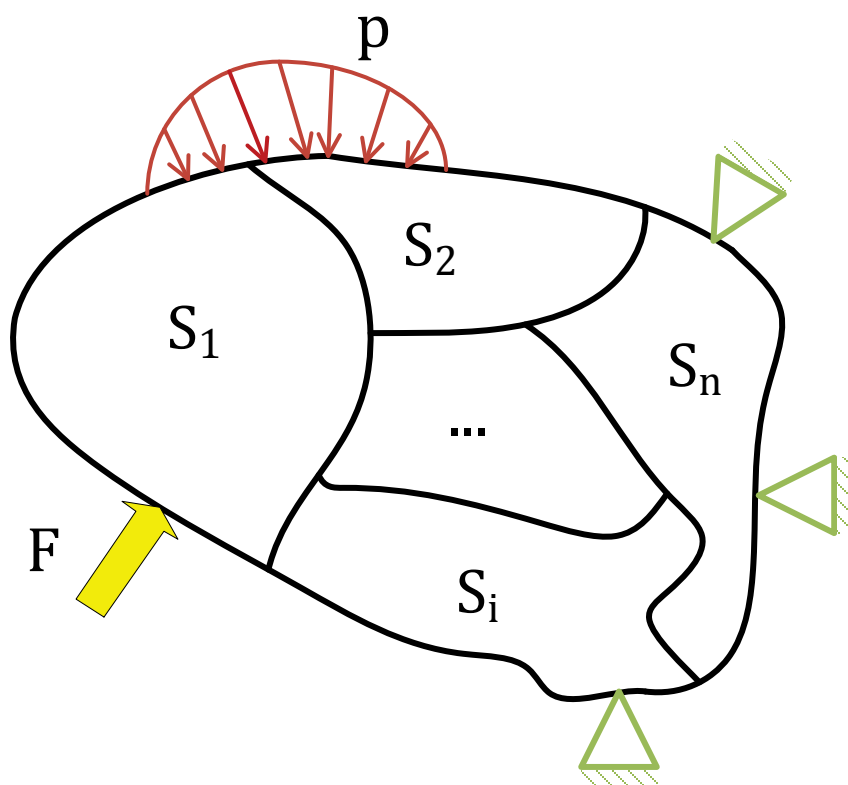
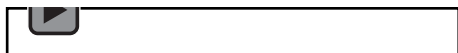


1.4 ŘEŠENÍ POMOCÍ ROZKLADU NA PODOBLASTI

Efektivní způsob řešení velmi rozsáhlých soustav rovnic je rozdělení řešené oblasti na tzv. domény (subdomény, substrukтуры, podoblasti) a provedení předeliminace neznámých na jednotlivých podoblastech před sestavením výsledné soustavy rovnic. Tento přístup nabývá významu v souvislosti s využíváním víceprocesorových paralelních počítačů. Myšlenka domén vede k možnosti paralelního řešení jednotlivých podoblastí na více procesorech současně, což vede k výraznému urychlení celého výpočtu.



Audio 1.5 Podoblasti



Obr. 2- Rozdělení řešení na podoblasti

Uvažme oblast řešení dle Obr. 2, která je rozdělena na n -podoblastí. Vezměme jednu podoblast a napišme pro ni soustavu rovnic ve tvaru:

$$[K_{Si}] \{U_{Si}\} = \{F_{Si}\}. \quad (11.10)$$

Rozdělme deformační parametry, dané maticí $\{U_{Si}\}$, na dvě skupiny a vytvořme tak dvě submatice. První submatice $\{U_H\}$ obsahuje všechny deformační parametry (zobecněná posunutí) ležící na hranici oblasti S_i . Druhá submatice $\{U_V\}$ obsahuje všechny deformační



parametry ležící uvnitř oblasti S_i . Předchozí rovnici XXX D1 přepíšme do následujícího tvaru:

$$\begin{bmatrix} [K_{HH}] & [K_{HV}] \\ [K_{VH}] & [K_{VV}] \end{bmatrix} \begin{Bmatrix} \{U_H\} \\ \{U_V\} \end{Bmatrix} = \begin{Bmatrix} \{F_H\} \\ \{F_V\} \end{Bmatrix} \quad (11.11)$$

Submatice $[K_{HH}]$ obsahuje tuhostní parametry uvnitř skupiny hraničních uzlů, submatice $[K_{VV}]$ sdružuje tuhostní parametry pro skupinu vnitřních uzlů. Obdobně silové submatice, $\{F_H\}$ zahrnuje zatížení působící na hranici podoblasti, $\{F_V\}$ zahrnuje zatížení na uzly uvnitř podoblasti. Rovnici (11.11) můžeme rozepsat na dvě samostatné rovnice:

$$[K_{HH}]\{U_H\} + [K_{HV}]\{U_V\} = \{F_H\} \quad (11.12)$$

$$[K_{VH}]\{U_H\} + [K_{VV}]\{U_V\} = \{F_V\}$$

Z druhé rovnice (11.12) vyjádříme posuvy $\{U_V\}$ a dosadíme do první rovnice (11.12), získáme:

$$([K_{HH}] - [K_{HV}][K_{VV}]^{-1}[K_{VH}])\{U_H\} = \{F_H\} - [K_{HV}][K_{VV}]^{-1}\{F_V\} \quad (11.13)$$

$$[\widehat{K}_{Si}]\{U_H\} = \{\widehat{F}_{Si}\}.$$

Jak lze vidět z předchozí rovnice, všechny vnitřní deformační parametry byly odstraněny a výsledná matice $[\widehat{K}_{Si}]$ má podstatně menší dimenzi, než původní matice $[K_{Si}]$. Obdobně se vše proveden pro zbývající podoblasti dané úlohy. Výsledná matice tuhosti celé soustavy se sestaví z předeliminovaných matic („se stříškou“). Výsledná soustava rovnic pak tedy obsahuje pouze neznáme parametry na hraničních podoblastích a je významně menší než původní soustava, která by vznikla klasickým postupem sestavení. Řešení jedné úlohy se tak rozpadá na sekvenci úloh o podstatně menších rozměrech. Toho lze využít při řešení velkých úloh na hardware s malou kapacitou nebo již výše zmiňovaných paralelních strojích.

Pokud se s určitou podoblastí pracuje často a opakovaně, je výhodné takovouto podoblast zařadit do knihovny existujících prvků a vytvořit tzv. **makroprvek**.



1.5 CVIČENÍ

Nejsložitějším typem 2D prvků jsou skořepinové prvky. Tyto prvky přenášejí v každém uzlu až 6 stupňů volnosti (3 posuvy, 3 natočení). Používají se při analýze konstrukcí, které obsahují velké množství plechů – těles, u nichž je jeden rozměr menší než ostatní.

Používají se i u úloh, kde bychom mohli snadno použít nosníkové prvky (obsahují I, U profily apod.). Výhodou skořepin je, že dokáží lépe popsat spojovací místa nosníků. V místa spojů nosníků často obsahují zaoblení, otvory (šrouby apod.), svary atd. Tyto spoje „nelze modelovat“ nosníkovými prvky, ale pomocí skořepinových prvků je možné je modelovat s dostatečnou přesností. Úloha „skořepinová“ je o řád větší než úloha „nosníková“.

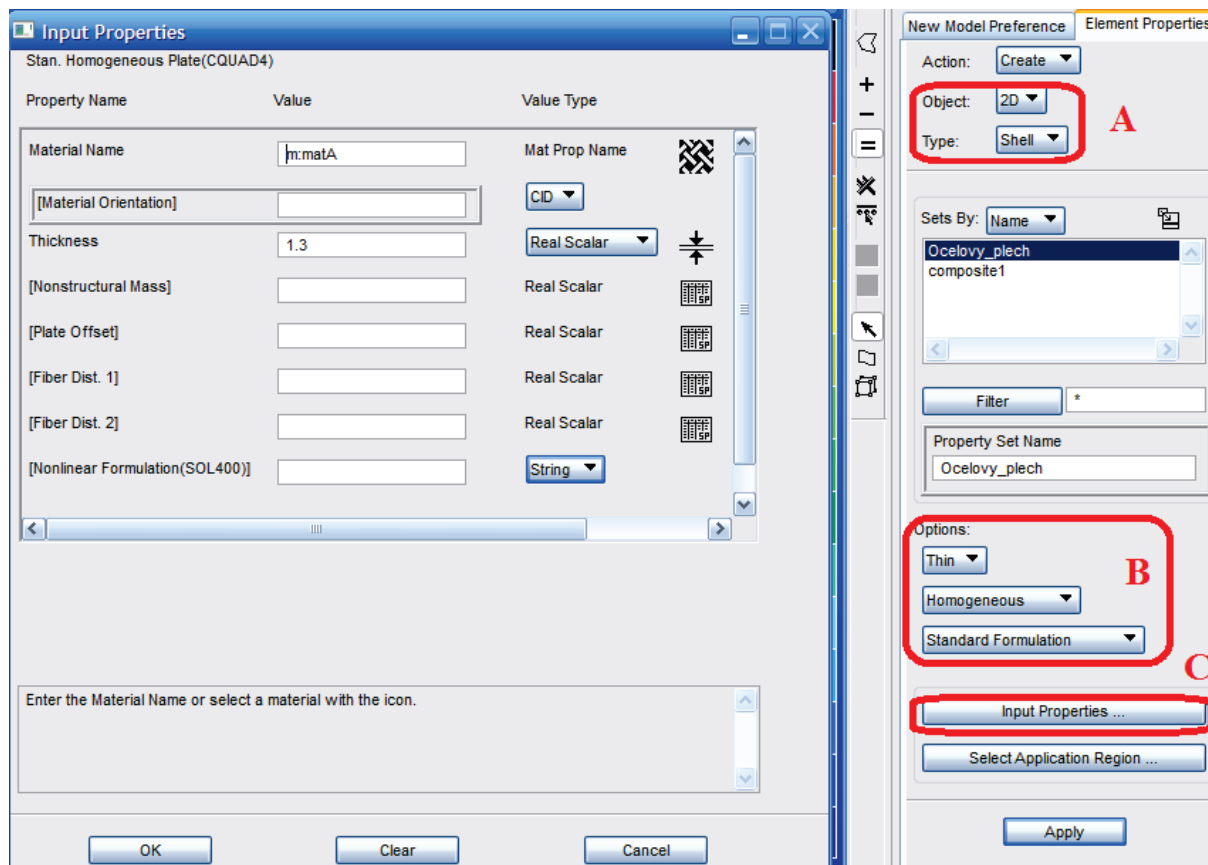
Pomocí skořepin nelze (není vhodné) modelovat libovolný tvar (kruh, obdélník) a často to ani není vhodné z hlediska požadavků na přesnost, rychlost či náročnost při vytváření modelu. Používají se tedy kombinace nosníkových (1D) a skořepinových (2D) (případně i objemových (3D)) prvků.

Existuje také tzv. solid-shell prvek – který spojuje výhody skořepinových a objemových prvků. V současné době klade tento prvek vysoké požadavky na model a v praktických (obvykle složitějších) případech je vhodnější využít skořepinový nebo objemový prvek (případně kombinaci viz kap.8 kontaktní úlohy). V této práci se tímto typem prvku nebudeme zabývat.

Skořepina je v nejjednodušším tvaru definována materiálem a tloušťkou. Vzhledem k tomu, že skořepina je velmi často používaný typ prvku, projdeme si nastavení „Element properties“ podrobněji:

- V nabídce Element properties najdeme skořepiny pod položkou 2D/Shell (viz obr. 1 A).
- Options – základní rozlišení, podobně jako u nosníků můžeme rozlišit (viz obr. 1 B).
 - Thin – tenký plech
 - Thick – tlustý plech
 - Field Point Mesh – používá se u akustické analýzy - nebudeme využívat.
- Options – typ materiálu (viz obr. 1 B).
 - Homogeneous – nejčastěji používaný homogenní materiál
 - Laminate – definice kompozitního materiálu
 - Equivalent Section – definice materiálu dle funkce vrstvy (Membrane, Bending, Shear, Coupling) – nebudeme využívat
- Options – použitá teorie (viz obr. 1 B).
 - Standart Formulation – budeme používat
 - Revised Formulation
 - Atd.
- Input Properties
 - Material Name – základní materiál (musíme zadat)
 - Material Orientation – orientaci materiálu u anisotropních materiálů
 - Thickness – tloušťka plechu (musíme zadat).
 - Nonstructural Mass – zadání hmotnosti ne na základě hustoty materiálu.
 - Plate Offset – posunutí střednice plochy
 - Fiber Dist. 1, Fiber Dist. 2 – vzdálenosti krajních vláken
 - Nonlinear Formulation – nastavení v případě využití řešiče SOL 400.

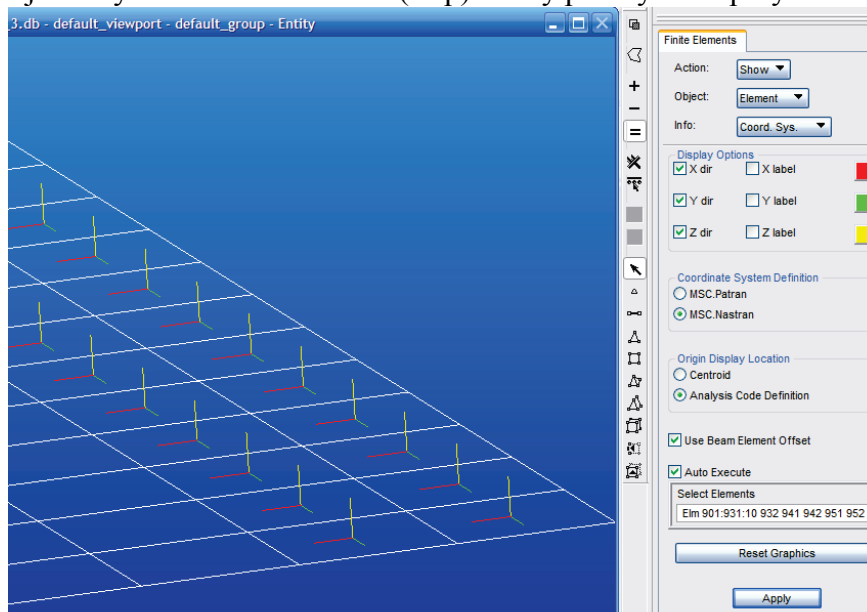




Obr. 1 Vytvoření Element properties skořepinového prvku

Na vetknutém nosníku si ukážeme specifika skořepinových prvků:

U skořepin rozlišujeme horní (Top) a spodní (Bottom) stranu plochy. Toto rozlišení je důležité při zadávání okrajových podmínek. Nejčastěji se s tímto můžeme setkat při zadávání tlaku (Pressure) nebo u kontaktů. Na rozlišení stran ploch (Top a Bottom) musíme pamatovat i při zobrazování výsledků. K rozlišení horní strany plochy (Top) a spodní strany plochy (Bottom) můžeme využít souřadný systém v elementu, viz obr. 2. Na obrázku je horní strana plochy skořepiny (Top) směrem nahoru, což je patrné ze směru osy z. Osa z jde tedy směrem ven z horní (Top) strany plochy skořepiny.

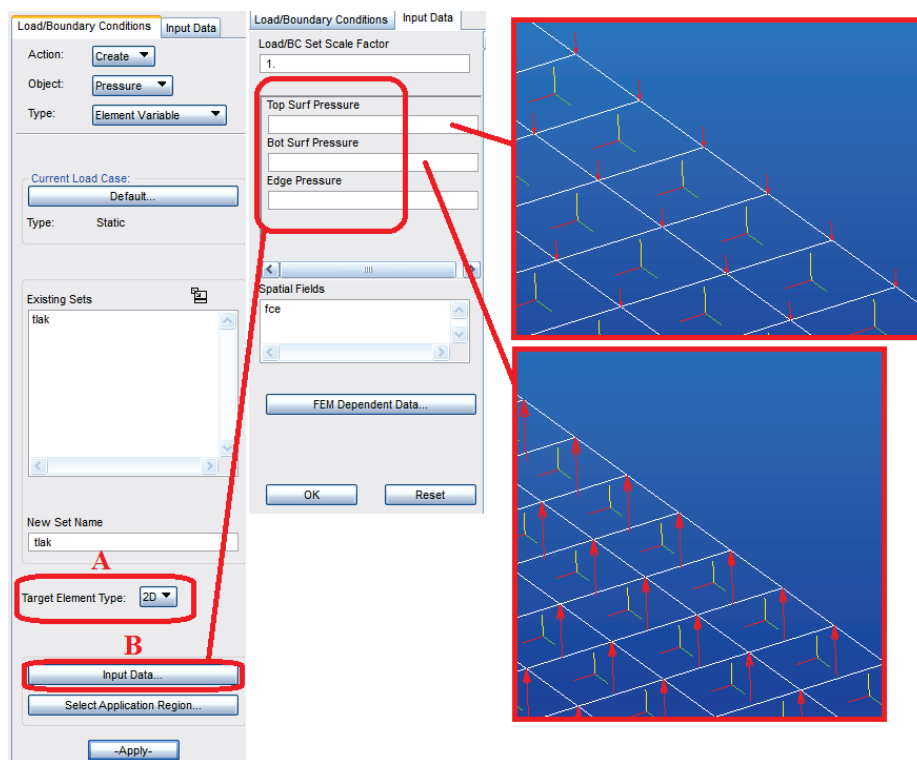


Obr. 2 Souřadný systém v elementu – určení horní strany skořepiny.

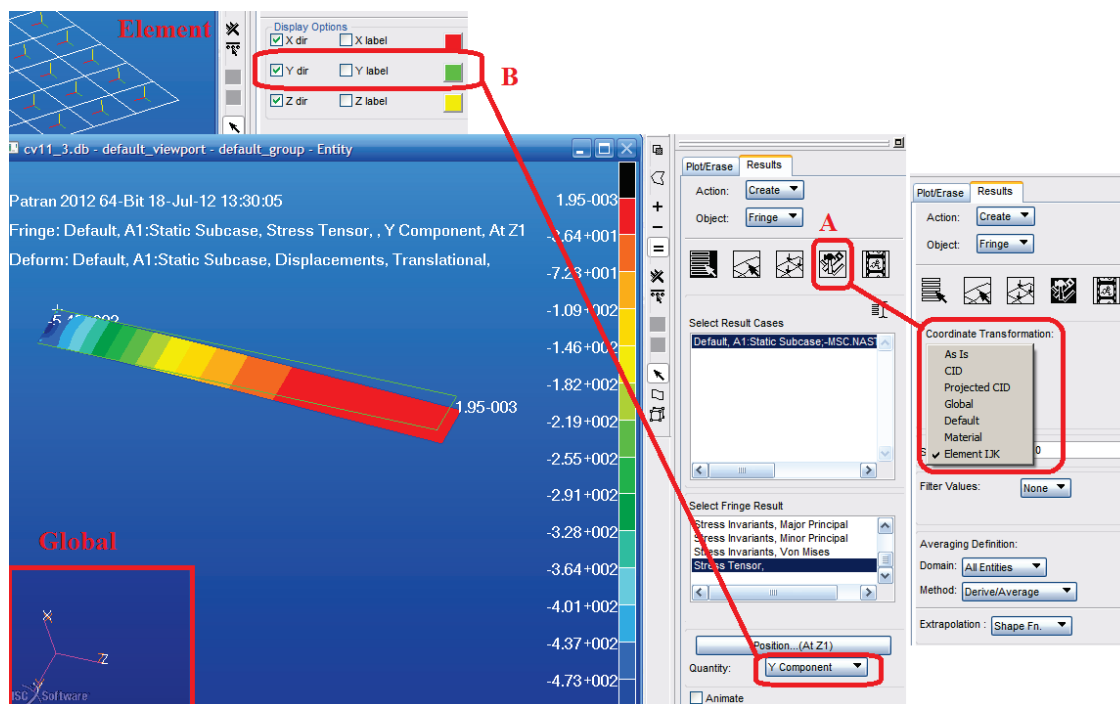


Při zadávání tlaku u skořepinových prvků (viz obr. 2 A – 2D) máme na výběr ze tří možností:

- Top Surf Pressure – tlak působí kolmo na horní stranu plochy (Top)
- Bot Surf Pressure – tlak působí kolmo na spodní stranu plochy (Bottom)
- Edge Pressure – tlak působí na hranu, můžeme využít např. při axisymetrii, rovinné napjatosti a rovinné deformaci.



Obr. 3 Tlak působící na horní nebo spodní plochu skořepiny.

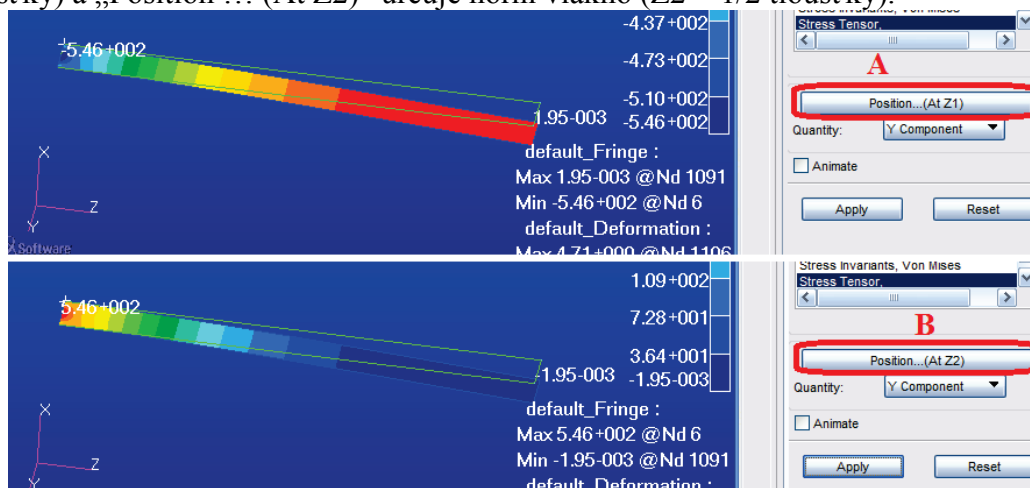


Obr. 4 Zobrazení výsledků – souřadný systém.



Podobně jako při zadávání tlaku zobrazujeme výsledky. Například napětí (Stress Tensor) ve směru osy se bude lišit dle zvoleného souřadného systému (osa y v globálním souřadném systému se může lišit od osy y v souřadném systému elementu). Korektně je možné zobrazit výsledky přes příkaz Fringe, kde můžeme nastavit „správný“ souřadný systém viz obr.4 A, B. Souřadný systém „Element IJK“ označuje souřadný systém elementu (viz obr.2). „Global“ označuje globální souřadný systém, který se obvykle zobrazuje vlevo dole.

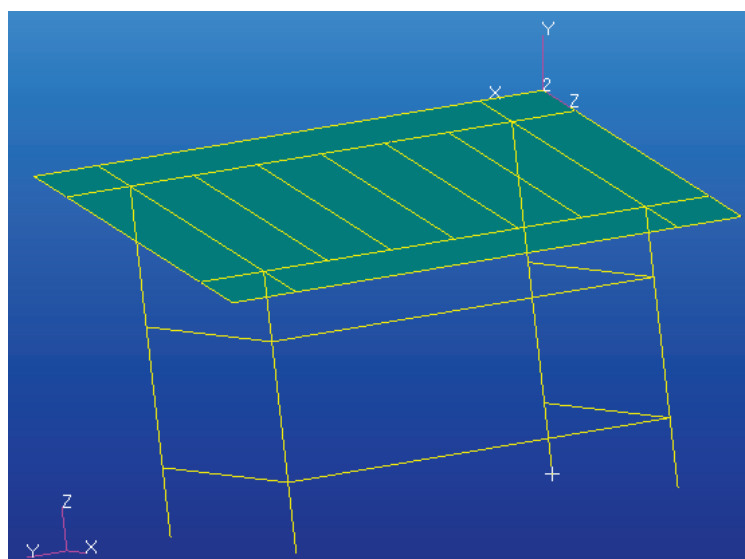
U skořepiny musíme také rozlišovat spodní (Bottom) a horní (Top) stranu plochy (vlákno). V našem příkladu s jednoduchým ohybem při zatížení tlakem na horní (Top) stranu plochy bude v horních vláknech tahové (kladné) a spodních vláknech tlakové (záporné). Z obr. 5 je zřejmé, že „Position ... (At Z1)“ určuje spodní vlákno ($Z1 = -1/2$ tloušťky) a „Position ... (At Z2)“ určuje horní vlákno ($Z2 = 1/2$ tloušťky).



Obr. 5 Zobrazení výsledků – spodní a horní vlákno.

Velmi častá a výhodná je kombinace skořepina – nosníkový prvek (kombinace skořepina – objemový prvek viz kap. 8). Postup při řešení jednoduché úlohy, ve které jsou kombinovány skořepinové a nosníkové prvky:

- Vytvoření modelu – pro nosníky použijeme křivky, pro skořepiny plochy.

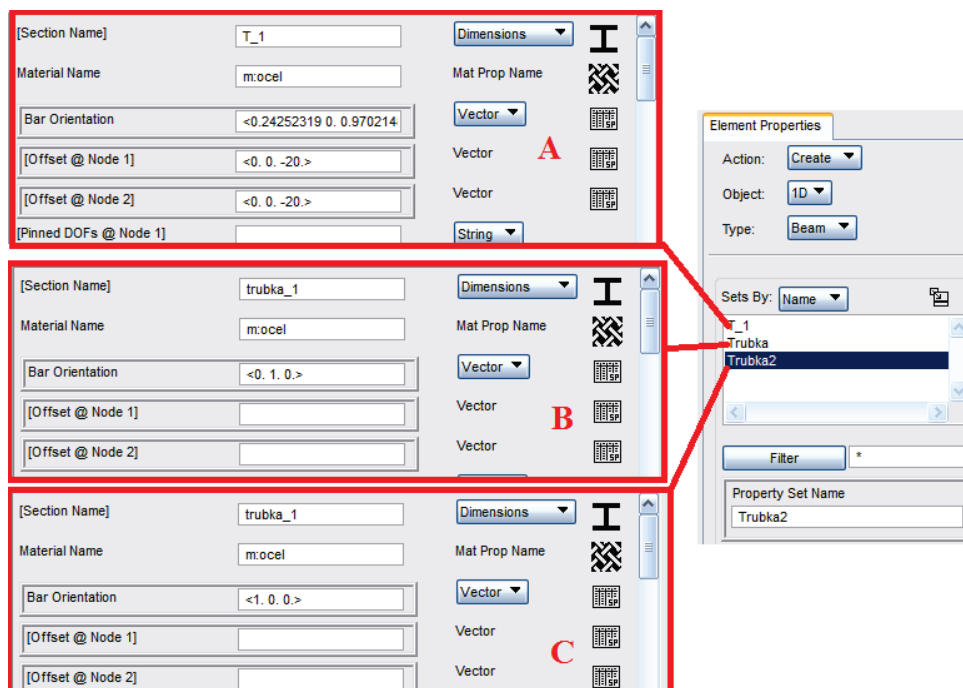


Obr.6 Geometrický model zastávky.

- Vytvoření materiálu – ocel
- Vytvoření „Element Properties“ viz obr. 7.

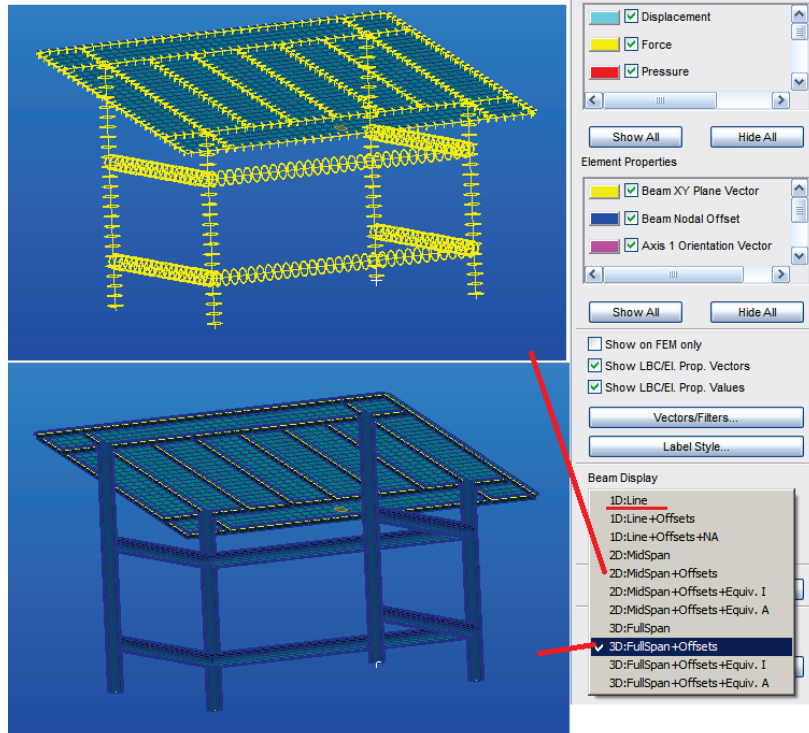


- Trubka – viz obr. 7 B, C. Vzhledem k orientaci trubek jsme museli vytvořit dva průřezy, lišící se orientací průřezu.
- T profil – viz obr. 7 A. Profily jsou natočené tak aby přímo doléhaly na skořepinu (Bar Orientation – např. pomocí vytvořeného souřadného systému). Dále musíme nosník (nebo skořepinu) posunout (Offset @Node 1, Offset @Node 2) tak, aby na sebe navazovaly části nosníku a skořepiny.
- Plech (skořepina) – viz obr. 1.
- Atd.



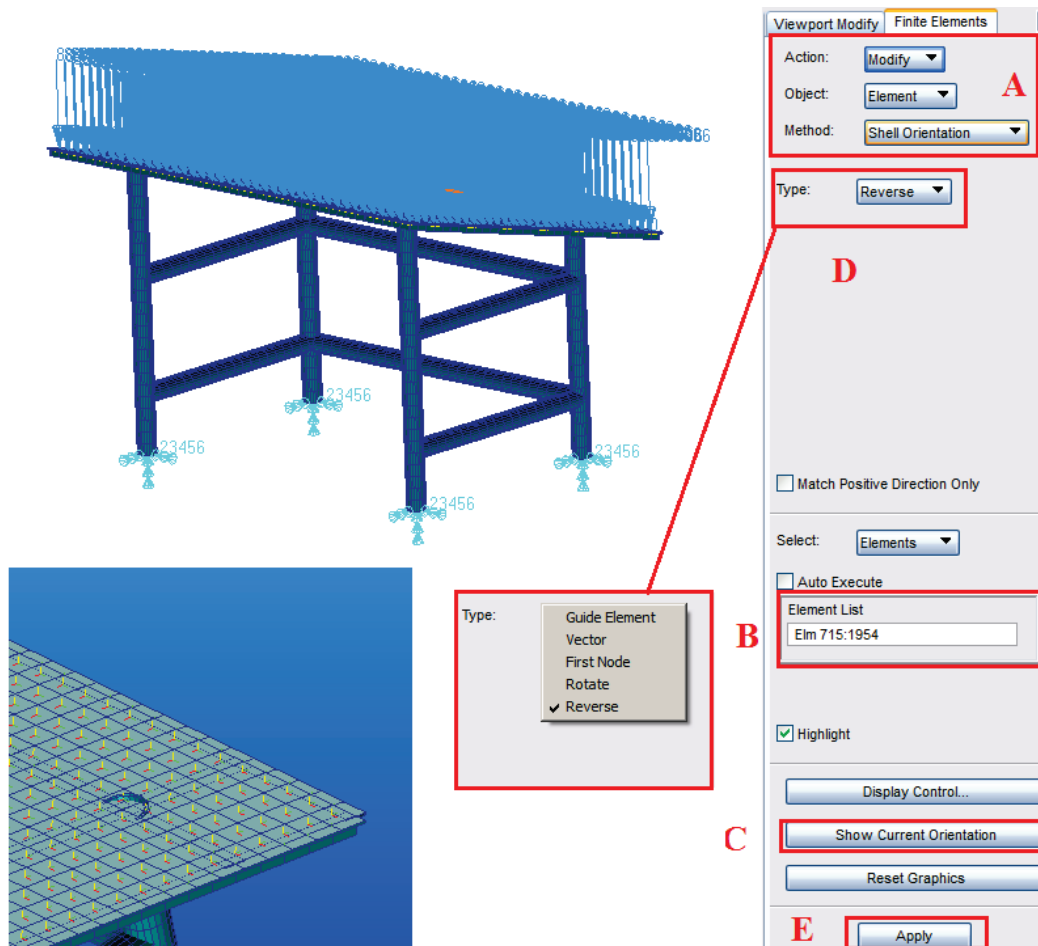
Obr. 7 Vytvoření „Element Properties“ pro jednotlivé profily.

- Vytvoření sítě konečných prvků (nosníkové i skořepinové prvky) viz obr. 8. Zobrazení nosníkových prvků bylo popsáno v kap. 9. (Display / Load/BC/Elem Props...). Nesmíme zapomenout na příkaz Finite Elements/Equivalence, který „sečte“ uzly a tím spojí těleso do jednoho. Příkaz můžeme použít, neboť použité nosníkové a skořepinové prvky mají v uzlech stejný počet stupňů volnosti.

1D:Line nezobrazí profil nosníku

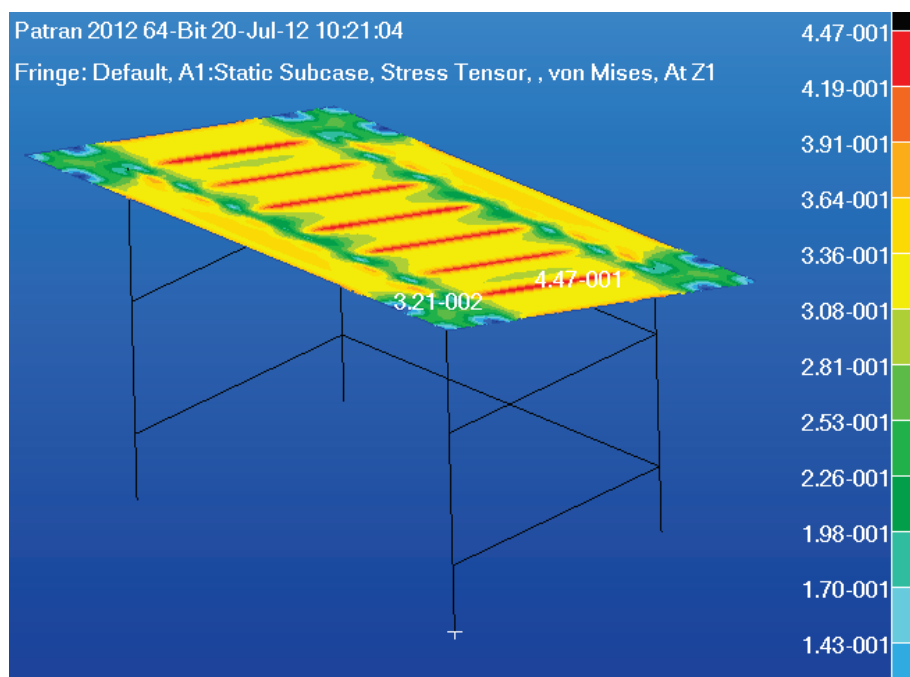
Obr. 8 Síť konečných prvků – různé zobrazení nosíkových prvků.

- Zadáme okrajové podmínky. Při zadávání okrajových podmínek zkontrolujeme orientaci souřadného systému elementu u skořepin. Orientace souřadného systému závisí na orientaci ploch. Orientace souřadného systému elementu má vliv na některé typy okrajových podmínek (Pressure, Total Load atd.) a samozřejmě také na výsledky !!!! V případě chybné orientace souřadného systému, použijeme příkaz Modify/Element/ Shell Orientation viz obr.9 A. Po výběru elementů (obr. 9 B) zde můžeme zobrazit stávající stav (obr. 9 C) a nevyhovující elementy „opravit“. Např. způsob opravy (Reverse) - obr. 9 D, výběr nevyhovujících elementů (Element List) B, aplikace (Apply) E).



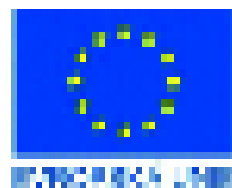
Obr.9 Okrajové podmínky, kontrola – obrácení normály plochy.

- Řešení úlohy (můžeme použít SOL101).
- Zobrazení výsledků (viz. obr.10). Popis byl uveden v prvním příkladu, u nosníkových prvků byl základní popis uveden v kap. 9.



Obr. 10 Průběh redukovaného napětí na „spodní“ straně skořepiny.





INVESTICE DO ROZVOJE VZDELÁVÁNÍ

Metoda konečných prvků 1

Výpocetová v rámci projektu
CZ.1.07/2.2.00/15.0463



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/>

