

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

**FAKULTA STAVEBNÍ**

**KATEDRA MECHANIKY**



**BAKALÁŘSKÁ PRÁCE**

KVĚTEN 2011

LUKÁŠ KRUPIČKA

## **PROHLÁŠENÍ:**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, pouze za odborného vedení vedoucího práce doc. Ing. Jana Zemana, Ph.D. Dále prohlašuji, že veškeré podklady, ze kterých jsem čerpal, jsou uvedeny v seznamu literatury.

V Praze dne 13. 5. 2011 .....

LUKÁŠ KRUPÍČKA

# **KOMPARATIVNÍ STUDIE DISKRÉTNÍCH MODELŮ PORUŠENÍ**

**(COMPARATIVE STUDY ON DISCRETE MODELS OF FAILURE)**

**ABSTRAKT:**

Práce obsahuje představení dvou rozdílných přístupů k analýze vývoje poškození příhradových konstrukcí. Prvním z nich je metoda nejslabšího článku, předpokládající porušení prutu v okamžiku, kdy energie v něm uložená dosáhne kritické hodnoty. Druhým přístupem je metoda založená na energetickém kritériu, která předpokládá porušení v okamžiku, kdy se disipace rovná rozdílu energií mezi porušenou a neporušenou konstrukcí. V práci jsou dále popsány programy, vypracované v jazyce MATLAB, sloužící k výpočtu těchto konstrukcí za použití efektivního způsobu přepočítávání inverze matice tuhosti. Další část práce obsahuje ukázkou ručního výpočtu konstrukce pomocí zmíněných metod a příklady, řešené pomocí výpočetních programů a porovnání výsledků.

**KLÍČOVÁ SLOVA:**

rovinný příhradový model, metoda nejslabšího článku, metoda založená na energetickém kritériu, porušení, deformace, předepsaný posun, neznámý posun

**ABSTRACT:**

This thesis introduces two different approaches to the failure analysis of lattice models. The first approach is the weakest link method, supposing that the element will fail if the energy cumulated in this element reaches its critical value. The second one is the method using a failure criterion based on the energy release rate. This method supposes that the crack occurs if dissipation is equal to difference between energy of damaged model and energy of entire structure. This thesis also contains a description of programs written in MATLAB to solve these problems, which use an effective way of recalculating inversions of the stiffness matrix. The next part of this work provides an example of hand calculation using these methods and examples solved by computer programs. It also provides comparing of the obtained results.

**KEY WORDS:**

planar lattice model, weakest link method, method using failure criterion based on energy release rate, failure, displacement, prescribed displacement, unknown displacement

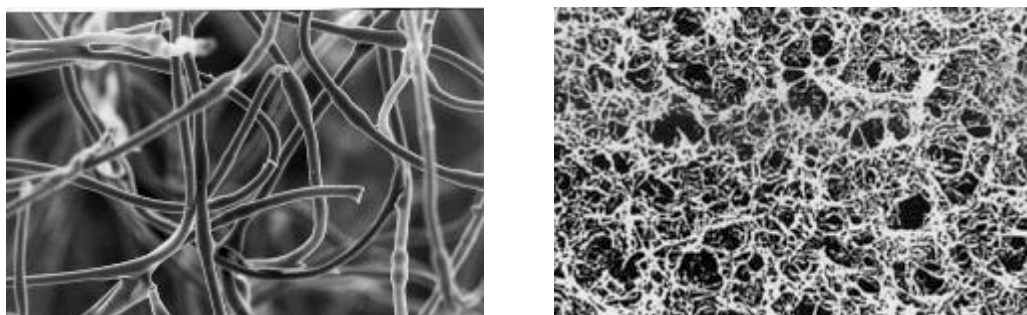
## OBSAH:

|  |           |
|--|-----------|
| Obsah .....  | 05        |
| <b>1 Úvod .....</b>  | <b>07</b> |
| <b>2 Představení porovnávaných metod .....</b>                           | <b>09</b> |
| 2.1 Metoda nejslabšího článku .....                                      | 09        |
| 2.2 Metoda založená na energetickém kritériu .....                       | 09        |
| 2.3 Porovnání metod .....  | 11        |
| <b>3 Popis programu – metoda nejslabšího článku .....</b>                | <b>12</b> |
| 3.1 Úvod .....   | 12        |
| 3.2 Popis zdrojového kódu .....  | 12        |
| 3.2.1 Vstupy .....   | 12        |
| 3.2.2 Tělo programu .....  | 12        |
| 3.2.3 Výstupy .....  | 16        |
| 3.3 Zdrojový kód .....   | 16        |
| 3.3.1 Úvodní část .....  | 16        |
| 3.3.2 Function weakest_link .....  | 16        |
| 3.3.3 Ostatní funkce .....   | 17        |
| 3.4 Shrnutí a výpis podprogramů .....                                    | 19        |
| <b>4 Popis programu – metoda založená na energetickém kritériu .....</b> | <b>20</b> |
| 4.1 Úvod .....   | 20        |
| 4.2 Popis zdrojového kódu .....  | 20        |
| 4.2.1 Vstupy .....   | 20        |
| 4.2.2 Tělo programu .....  | 20        |
| 4.2.3 Výstupy .....  | 21        |
| 4.3 Zdrojový kód .....   | 21        |
| 4.3.1 Úvodní část .....  | 21        |
| 4.3.2 Function Delta_energy_matrix .....                                 | 21        |
| 4.3.3 Ostatní funkce .....   | 23        |
| 4.4 Shrnutí a výpis podprogramů .....                                    | 23        |
| <b>5 Odvození „Woodbury matrix identity“ .....</b>                       | <b>24</b> |
| 5.1 Úvod .....   | 24        |
| 5.2 Odvození .....   | 24        |
| 5.2.1 Odvození pomocí blokové eliminace .....                            | 24        |
| 5.2.2 Odvození pomocí LDU rozkladu .....                                 | 25        |
| 5.3 Přímý důkaz .....  | 27        |
| 5.4.1 Tvar odvozený pomocí blokové eliminace .....                       | 28        |
| 5.4.2 Tvar odvozený pomocí LDU rozkladu .....                            | 28        |
| 5.4 Speciální případ: Sherman – Morrison formula .....                   | 29        |
| 5.4.1 Popis .....  | 29        |
| 5.4.2 Použití .....  | 29        |
| 5.4.3 Ověření .....  | 30        |

|    |  |    |
|----|--|----|
| 6  | <b>Příklad 1 – ruční výpočet</b> .....                       | 31 |
|    | 6.1 Schéma, zatížení .....                                   | 31 |
|    | 6.2 Výpočet – metoda nejslabšího článku .....                | 31 |
|    | 6.3 Výpočet – metoda založená na energetickém kritériu ..... | 33 |
|    | 6.3.1 Předpoklad 1 – porušení prutu č. 1 .....               | 34 |
|    | 6.3.2 Předpoklad 2 – porušení prutu č. 2 .....               | 35 |
|    | 6.4 Porovnání .....  | 36 |
| 7  | <b>Příklad 2 – výpočet pomocí programu</b> .....             | 37 |
|    | 7.1 Schéma, zatížení .....                                   | 37 |
|    | 7.2 Výsledky .....   | 38 |
|    | 7.2.1 Metoda nejslabšího článku .....                        | 38 |
|    | 7.2.2 Metoda založená na energetickém kritériu .....         | 38 |
|    | 7.2.3 Porovnání .....  | 38 |
| 8  | <b>Příklad 3 – výpočet pomocí programu</b> .....             | 40 |
|    | 8.1 Schéma, zatížení .....                                   | 40 |
|    | 8.2 Výsledky .....   | 41 |
|    | 8.2.1 Metoda nejslabšího článku .....                        | 41 |
|    | 8.2.2 Metoda založená na energetickém kritériu .....         | 42 |
|    | 8.2.3 Porovnání .....  | 43 |
| 9  | <b>Příklad 4 – výpočet pomocí programu</b> .....             | 44 |
|    | 9.1 Schéma, zatížení .....                                   | 44 |
|    | 9.2 Výsledky .....   | 45 |
|    | 9.2.1 Metoda nejslabšího článku .....                        | 45 |
|    | 9.2.2 Metoda založená na energetickém kritériu .....         | 47 |
|    | 9.2.3 Porovnání .....  | 49 |
| 10 | <b>Závěr</b> .....   | 50 |
| 11 | <b>Literatura</b> .....                                      | 51 |

# 1 ÚVOD

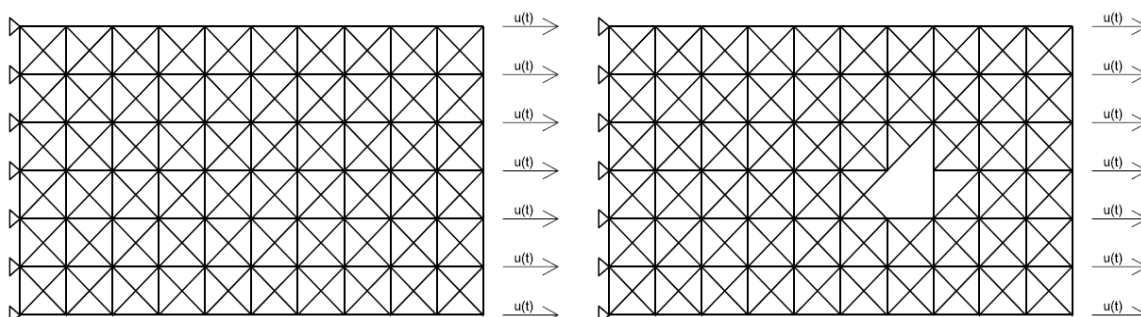
Pokud chceme zkoumat chování některých materiálů, můžeme k jejich modelování použít síť prutových prvků. Tyto sítě se nazývají příhradovými modely. Mnoho přírodních i umělých materiálů obsahuje vláknitou strukturu. U takovýchto materiálů se diskretizace vláken pomocí příhradových prutů přímo nabízí. Pomocí této diskretizace lze numericky zkoumat reakce modelu na postupné zatěžování. Čím hustší síť prutů vytvoříme, tím přesnější výsledky získáme. V určitém čase dojde k porušení prvního vlákna a tím zároveň ke změně tuhosti systému jako celku a k oslabení vnitřní struktury materiálu. Příhradové modely se však používají i k modelování materiálů, které nemají vláknitou vnitřní strukturu, například betonu nebo asfaltu.



Obr. 1.1: Ukázka vláknitých přírodních materiálů



Obr. 1.2: Porušení vláknité struktury přírodního materiálu



Obr. 1.3: Reprezentace vláknité struktury pomocí příhradových modelů

Existuje několik výpočetních metod, které se zabývají předvídáním vývoje porušení těchto modelů. Cílem této práce je porovnání metody nejslabšího článku a metody založené na energetickém kritériu pomocí ručního výpočtu a pomocí skriptů v jazyce MATLAB.



## 2 PŘEDSTAVENÍ POROVNÁVANÝCH METOD

### 2.1 METODA NEJSLABŠÍHO ČLÁNKU

Metoda nejslabšího článku je metoda, která porušení prutu uvažuje v okamžiku, kdy energie v něm uložená přesáhne kritickou hodnotu. Poté nalezneme prut, k jehož porušení dojde nejdříve, tento prut vyjme z konstrukce a počítáme s konstrukcí oslabenou o tento prut. Tento postup opakujeme až do okamžiku, kdy dojde ke kolapsu konstrukce jako celku, případně do okamžiku konce zatěžování. Následující část práce popisuje postup, jak metodou nejslabšího článku zjistíme čas, kdy dojde k porušení prvního prutu:

Energie uložená v prutu se spočte následujícím způsobem:

$$E(t) = \frac{1}{2} r^T K r \quad (2.1)$$

Kde  $r$  je vektor uzlových posunů a  $K$  je matice tuhosti prutu.

V našem případě budeme předpokládat zatížení konstrukce předepsaným posunem styčnicku, které bude lineárně narůstat v čase. Z toho tedy vyplývá, že energie, která bude v prutu uložená, bude narůstat kvadraticky. Obecně však můžeme zatěžovat posunem, jehož hodnota se v čase bude měnit podle obecné časové funkce. K porušení prutu dojde v okamžiku, kdy se hodnota energie v prutu bude rovnat kritické energii.

$$E f(t) = E_{crit} \quad (2.2)$$

Pokud tedy chceme zjistit čas, kdy k porušení prutu dojde, musíme funkci invertovat (Pozn.: Předpokládáme, že funkce je na intervalu kladných čísel prostá.)

$$t_{fail} = [f]^{-1} \frac{E_{crit}}{E} \quad (2.3)$$

V našem případě, kdy uvažujeme lineární nárůst posunů, tzn. kvadratický nárůst energie, je tedy inverzní funkci odmocnina a čas kolapsu prutu spočteme následujícím způsobem:

$$t_{fail} = \sqrt{\frac{E_{crit}}{E}} \quad (2.4)$$

Tento výpočet provedeme pro všechny pruty v konstrukci. Jako první bude porušen prut, jemuž odpovídá nejmenší hodnota času  $t_{fail}$ .

Z předchozího výpočtu jsme zjistili čas kolapsu prvního prutu. Nyní tento prut z konstrukce vyjme, tuto nově získanou konstrukci opět zatížíme, spočteme posuny, energii v prutech a zjistíme čas kolapsu dalšího prutu. Výpočet opakujeme, dokud nedojde ke kolapsu celé konstrukce nebo dokud neskončí zatěžování.

### 2.2 METODA ZALOŽENÁ NA ENERGETICKÉM KRITÉRIU

Tato metoda funguje na podobném principu jako metoda nejslabšího článku. Liší se ve výpočtu času kolapsu jednotlivých prutů. Metoda nejslabšího článku předpokládá kolaps v okamžiku dosažení kritické energie v jednom z prutů, zatímco tato metoda zjišťuje čas

kolapsu pomocí rovnosti disipované energie a rozdílu energií porušeného a neporušeného systému.

Praktický výpočet potom vypadá takto:

1. Počítáme s neporušenou konstrukcí a zjistíme hodnotu energie, která se v ní uloží.

$$Eng = \frac{1}{2} r^T K r \quad (2.5)$$

2. Počítáme s konstrukcí, v níž byl porušen prut 1 a zjistíme hodnotu energie, která se uloží v systému. Tentýž postup opakujeme pro systém s porušeným prutem 2, 3 atd. až projedeme všechny možnosti porušení jednoho prutu. Dostaneme tedy  $n$  hodnot, kde  $n$  je počet prutů.

$$Eng_i = \frac{1}{2} r_i^T K_i r_i \quad (2.6)$$

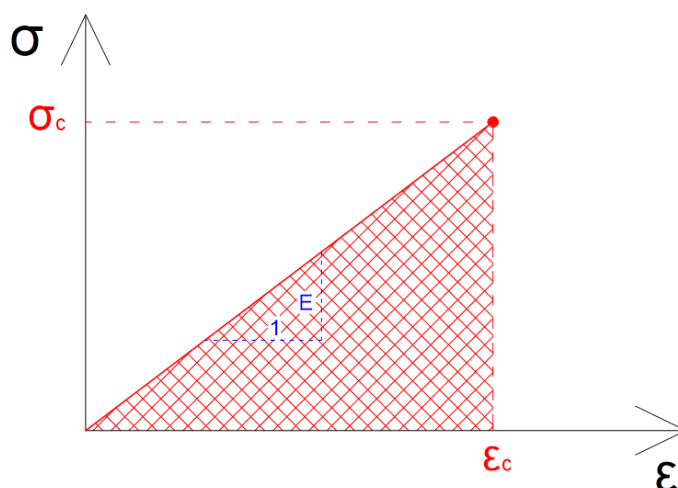
Kde  $Eng_i$  představuje hodnotu energie systému, v němž je porušen prut  $i$ .

3. Spočteme, o kolik se uložená energie v systému snížila oproti energii v neporušeném systému.

$$\Delta Eng_i = Eng_i - Eng \quad (2.7)$$

Kde  $\Delta Eng_i$  představuje změnu energie, při porušení prutu  $i$  oproti energii neporušeného systému. Hodnoty budou záporné a bude jich opět  $n$ .

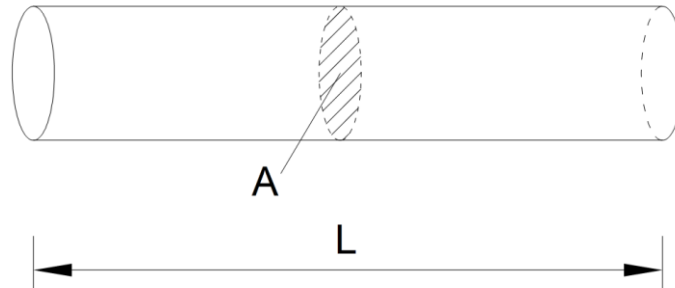
4. Spočteme hodnotu disipace při porušení prutu. Ta se rovná ploše pod pracovním diagramem v okamžiku porušení, vynásobené objemem prutu.



Obr. 2.1.: Pracovní diagram elastického materiálu

Za předpokladu lineárně pružného chování materiálu se plocha pod pracovním diagramem až do porušení rovná:

$$S = \frac{1}{2} \varepsilon_c \sigma_c \quad (2.8)$$



Obr. 2.2: Prut konstantního průřezu

Pro prut konstantní průřezové plochy  $A$  o délce  $L$  se potom disipace spočítá takto:

$$D = \frac{1}{2} \varepsilon_c \sigma_c AL = \frac{1}{2} \frac{\sigma_c}{E} \sigma_c AL = \frac{1}{2} \sigma_c^2 \frac{AL}{E} \quad (2.9)$$

5. Z rovnosti změny energie a disipace můžeme spočítat čas, kdy dojde k porušení prutu.

$$-t^2 \Delta Eng_i = D_i \quad (2.10)$$

$$t_{fail,i} = \sqrt{-\frac{D_i}{\Delta Eng_i}} \quad (2.11)$$

Kde  $t_{fail,i}$  představuje čas kolapsu  $i$ -tého prutu.

Opět získáme  $n$  hodnot  $t_{fail}$  a jako první bude porušen prut, jemuž odpovídá minimální hodnota. Podobně jako u metody nejslabšího článku postup opakujeme, dokud nedojde ke kolapsu celé konstrukce, případně ke konci zatěžování.

## 2.3 POROVNÁNÍ METOD

Obě metody fungují na velmi podobném principu, liší se pouze přístupem k výpočtu času porušení jednotlivých prvků. K porovnání výsledků těchto metod použijeme programy, popsané v další kapitole a ruční výpočet.

## 3 POPIS PROGRAMU – METODA NEJSLABŠÍHO ČLÁNKU

### 3.1 ÚVOD

Zdrojový kód je psán v programovacím jazyku MATLAB, který je nevhodnějším programovacím jazykem, pokud pracujeme na problému, popsaném pomocí maticových rovnic. Deformační metoda k výpočtu staticky neurčitých konstrukcí je tímto případem.

### 3.2 POPIS ZDROJOVÉHO KÓDU

#### 3.2.1 Vstupy

Jako vstupní parametry do programu zadáme následující matice:

|                         |  |
|-------------------------|--|
| <i>xz</i> .....         | matice o rozměru $k \times 2$ , kde $k$ je počet styčníků, obsahující souřadnice jednotlivých styčníků v souřadném systému. Na první pozici $x$ -ovou souřadnici a na druhé pozici $z$ -ovou, přičemž osa $z$ je orientována kladně směrem dolů. |
| <i>elems</i> .....      | matice o rozměru $n \times 2$ , kde $n$ je počet prutů, obsahující topologii soustavy, jinými slovy tato matice popisuje, jaké styčníky jsou propojeny jednotlivými pruty  |
| <i>p</i> .....          | matice předepsaných posunů   |
| <i>sigma_crit</i> ..... | matice o rozměru $n \times 1$ obsahující kritická napětí jednotlivých prutů  |
| <i>A</i> .....          | matice o rozměru $n \times 1$ obsahující průřezové plochy prutů  |
| <i>E</i> .....          | matice o rozměru $n \times 1$ obsahující hodnoty Youngova modulu pružnosti jednotlivých prutů  |
| <i>r</i> .....          | předepsané zatížení posunem styčnicku  |
| <i>limit</i> .....      | čas, kdy chceme zastavit výpočet   |

#### 3.2.2 Tělo programu

Po zadání vstupních parametrů zavoláme funkci *weakest\_link*. Tato funkce tvoří základ celého programu. Vstupními parametry jsou pro ni souřadnice jednotlivých styčníků (*xz*), topologie soustavy (*elems*), vektor předepsaných posunů (*p*), zadané zatížení posunem (*r*), moduly pružnosti prutů (*E*) a pole kritických napětí (*sigma\_crit*) a limitní čas (*limit*). Výstupem této funkce bude matice *Fail\_Matrix*, v jejímž prvním sloupci budou čísla porušených prutů a ve druhém sloupci časy porušení.

Funkce nejdříve zjistí počet styčníků jako velikost vektoru *xz* a tuto hodnotu uloží do proměnné  $k$ . Dále počet prutů jako velikost vektoru *elems* a hodnotu uloží do proměnné  $n$ . Potom je potřeba definovat matici tuhosti celého systému, označíme ji jako  $K$  a definujeme ji jako matici plnou nul o rozměrech  $2k \times 2k$ , neboť budeme pracovat s příhradovými konstrukcemi a jako neznámé tedy uvažujeme pouze svislý a vodorovný posun styčnicku. Dále si ještě připravíme vektor *E\_crit* o velikosti  $n$ , kam budeme později ukládat hodnoty kritických energií, matici *Eng*, kam budeme zapisovat hodnoty energie, která je v jednotlivých prutech uložena, matici *Temp*, kam budeme ukládat časy kolapsů jednotlivých prutů a také výslednou výstupní matici *Fail\_Matrix*.

Výpočet začneme sestavení matice kritických energií *E\_crit*. Kritickou energii prutu počítá funkce *critical\_energy*. Vstupními parametry pro ni jsou souřadnice obou styčnicků, modul pružnosti, průřez a kritické napětí. Z těchto vstupů spočteme kritickou energii a pomocí for cyklu zapíšeme do příslušného řádku připravené matice *E\_crit*.

```

1 function [E_critical]=critical_energy(xa,xb,za,zb,Ee,Ae,sigma_crit)
2     dxab = xb-xa; dzab = zb-za;
3     le=sqrt((dxab)^2 + (dzab)^2);
4     E_critical=1/2*sigma_crit*sigma_crit*Ae*le/Ee;
5 end

```

K sestavení globální matice tuhosti použijeme for cyklus s počtem opakování rovným počtu prutů, tj. hodnotě proměnné  $n$ . Globální matice tuhosti se skládá z příspěvků jednotlivých prutů neboli z příspěvků lokálních matic tuhosti jednotlivých prutů. Tyto lokální matice sestavíme pomocí funkce *matice\_tuhosti*. Jejimi vstupními parametry budou vodorovné a svislé souřadnice obou styčnicků prutu, jeho průřezová plocha a modul pružnosti. Z rozdílů jednotlivých souřadnic spočteme délku prutu a také sinus a cosinus úhlu, který prut svírá s kladnou poloosou osy  $x$ . Lokální matice tuhosti pro tah/tlak se potom určí z následujícího vztahu:

$$K = \frac{EA}{l} \begin{bmatrix} c \\ s \\ -c \\ -s \end{bmatrix} [c \quad s \quad -c \quad -s] \quad (3.1)$$

Kde  $c$  je hodnota funkce cosinus zmíněného úhlu,  $s$  je hodnota funkce sinus,  $E$  je modul pružnosti,  $A$  je průřezová plocha a  $l$  je délka prutu. Lokální matice tuhosti bude mít rozměr  $4 \times 4$ .

```

1 function [Ke, le] = matice_tuhosti(xa,xb,za,zb,Ee,Ae)
2     dxab = xb-xa; dzab = zb-za;
3     le=sqrt((dxab)^2 + (dzab)^2);
4     c = dxab/le; s = dzab/le;
5     Ke = Ee*Ae/le*[c;s;-c;-s]*[c,s,-c,-s];
6 end

```

Ve for cyklu postupně přičítáme příspěvky lokálních matic tuhosti podle kódových čísel do globální matice, která bude mít rozměr  $2k \times 2k$ .

```

23 K(id,id)=K(id,id)+Ki;

```

Nyní můžeme přikročit k výpočtu posunů. Pokud bychom počítali ručně, stačilo by nám sestavit matici tuhosti, odpovídající kódovým číslům neznámých posunů, ze soustavy rovnic spočítat tyto posuny a poté dopočítat reakce. Podobně budeme postupovat i v tomto případě, kdy si globální matici tuhosti můžeme rozdělit na submatice a budeme řešit následující soustavy rovnic:

$$\begin{bmatrix} K(u,u) & K(u,p) \\ K(p,u) & K(p,p) \end{bmatrix} \begin{bmatrix} r(u) \\ r(p) \end{bmatrix} = \begin{bmatrix} f(u) \\ f(p) + R \end{bmatrix} \quad (3.2)$$

Celou globální matici tuhosti známe, předepsané posuny také známe a silové zatížení styčnicků je v našem případě nulové. Z tohoto vztahu tedy získáme dvě rovnice, z nichž lze dopočítat vektor neznámých posunů  $r(u)$  a reakce  $R$ .

$$K(u,u) r(u) + K(u,p) r(p) = f(u) \quad (3.3)$$

$$K(p,u) r(u) + K(p,p) r(p) = f(p) + R \quad (3.4)$$

Spočteme tedy vektor neznámých posunů, případně můžeme spočítat i reakce. Ty ale k dalšímu výpočtu potřebovat nebudeme.

27  $r(u) = -K(u, u) \setminus (K(u, p) * r(p)) ;$

Nyní potřebujeme zjistit, u kterého prutu dojde k porušení. K tomu nám slouží funkce *find\_weakest\_link*. Vstupními parametry této funkce jsou opět souřadnice styčnicků, topologie soustavy, moduly pružnosti, kritická energie a spočítané posuny. energii uloženou v prutu můžeme spočítat z následujícího vztahu:

$$Eng = \frac{1}{2} r^T K r \quad (3.5)$$

Pomocí for cyklu přes počet prutů zjistíme energii ve všech prutech a postupně si tyto hodnoty budeme ukládat do matice *Eng*. Zbývá spočítat už pouze čas kolapsu prutu. Ten spočteme podle vztahu (2.4). Výsledné časy budeme ukládat do matice *Temp*.

19  $Temp = \text{sqrt}( E\_crit ./ Eng ) ;$

V matici *Temp* máme uloženo pole časů kolapsů prutů. V tomto vektoru najdeme minimum a tím najdeme i prut, který zkolabuje jako první. Bude to ten prut, na jehož pozici se tento minimální čas nachází.

Ve funkci *find\_weakest\_link* ještě můžeme spočítat i normálové síly podle následujícího vztahu:

$$N = \frac{EA}{l} \begin{bmatrix} -C \\ -S \\ C \\ S \end{bmatrix} [u_a \quad w_a \quad u_b \quad w_b] \quad (3.6)$$

Kde *u* a *w* jsou vodorovné, resp. svislé posuny krajních styčnicků prutu. Normálové síly můžeme použít v případě, kdy bychom chtěli vyloučit porušování tažených nebo tlačných prutů. To bychom mohli operativně nastavit pomocí podmínky *if*.

Výstupem funkce *find\_weakest\_link* jsou proměnné *emin* a *tmin*, kde *emin* je číslo prvního kolabujícího prutu a *tmin* čas kolapsu.

Vrátíme se zpět do funkce *weakest\_link* a tyto hodnoty zapíšeme do výsledné výstupní matice *Fail\_Matrix*.

Nyní víme, který prut se poruší jako první a v jakém čase, potřebujeme tedy tento prut z konstrukce vyjmout a počítat bez něj. Mohli bychom postupovat stejně jako v předchozím případě, pouze s tím rozdílem, že bychom danému prutu přiřadili nulovou hodnotu modulu pružnosti a výpočet opakovali.

Museli bychom však k vyřešení soustavy rovnic znovu použít Gaussovu eliminaci, která má složitost  $O(n^3)$ . Abychom se tomu vyhnuli, použijeme algoritmus známý pod názvem „Woodbury Matrix Identity“, v našem konkrétním případě můžeme dokonce využít jeho speciální případ tzv. „Sherman-Morrisson Formula“. Tento algoritmus nám umožní spočítat inverzi matice, která se liší od původní matice o součin vektorů hodnoty jedna (tzv. rank one update) aniž bychom museli použít Gaussovu eliminaci. Pokud tedy budeme chtít řešit rovnici  $B_{ii} x = b$ , kde  $B_{ii}$  bude matice, která se liší od matice  $B_i$  pouze součinem vektorů  $uv^T$  a my budeme znát inverzi původní matice  $B_i$ , můžeme algoritmus použít. Platí následující vztah.

Pokud:

$$B_{II} = B_I + uv^T \quad (3.7)$$

Potom:

$$B_{II}^{-1} = (B_I + uv^T)^{-1} = B_I^{-1} - \frac{B_I^{-1}uv^TB_I^{-1}}{1+v^TB_I^{-1}u} \quad (3.8)$$

Řešení rovnice  $B_{II} x = b$  získáme potom jednoduše vynásobením celé rovnice zleva inverzí matice  $B_{II}$ :

$$B_{II}^{-1}B_{II}x = B_{II}^{-1}b$$

$$x = B_{II}^{-1}b$$

Odvození vztahu (3.8) pro výpočet této inverze je věnována zvláštní kapitola 5.

Výpočet vektorů, o které je potřeba změnit původní matici tuhosti po tom, co dojde k porušení prutu, je proveden ve funkci *rank\_one\_update*. Vytvoříme si pomocný vektor délky  $2k$ . Do něj budeme zapisovat příspěvky porušeného prutu podle kódových čísel. Poté vybereme odpovídající příspěvky, které mají vliv na matici  $K(u,u)$  a tím získáme výsledný opravný vektor o délce velikosti  $u$ . Tento vektor označíme proměnnou  $v$ . V našem případě si můžeme výpočet ještě usnadnit, neboť druhý opravný vektor je stejný jako první, pouze je třeba ho transponovat. Nesmíme zapomenout ještě na vliv konstanty  $EA/L$ , kterou je příspěvek prutu do globální matice tuhosti násoben. Abychom zachovali rovnost obou opravných vektorů, rozložíme tento násobek do obou rovnoměrně tím, že ho odmocníme a vynásobíme jím oba vektory. Konečně tedy získáme náš rank-one update. Oprava je se záporným znaménkem, proto je ještě potřeba mírně upravit vztah pro výpočet inverze opravené matice:

$$K(u,u)_{II}^{-1} = (K(u,u)_I - vv^T)^{-1} = K(u,u)_I^{-1} + \frac{K(u,u)_I^{-1}vv^TK(u,u)_I^{-1}}{1-v^TK(u,u)_I^{-1}v} \quad (3.9)$$

K výpočtu neznámých posunů ještě potřebujeme opravit matice  $K(u,p)$  a  $K(p,u)$ , na které měl porušený prut také vliv. K výpočtu reakcí budeme opět potřebovat matici i  $K(p,p)$ , proto je potřeba odečíst příspěvky od porušeného prutu od celé globální matice tuhosti. Nyní můžeme přikročit k výpočtu neznámých posunů. Celou rovnici pro jejich výpočet vynásobíme zleva inverzí opravené matice  $K(u,u)$ :

$$\begin{aligned} K(u,u)r(u) + K(u,p)r(p) &= f(u) \\ K(u,u)^{-1}K(u,u)r(u) + K(u,u)^{-1}K(u,p)r(p) &= K(u,u)^{-1}f(u) \\ I r(u) + K(u,u)^{-1}K(u,p)r(p) &= K(u,u)^{-1}f(u) \end{aligned}$$

Dostáváme tedy výsledný vztah pro výpočet neznámých posunů:

$$r(u) = -K(u,u)^{-1}K(u,p)r(p) + K(u,u)^{-1}f(u) \quad (3.10)$$

V našem případě navíc neuvažujeme silové zatížení, takže výsledný vztah bude v následujícím tvaru:

$$r(u) = -K(u,u)^{-1}K(u,p)r(p) \quad (3.11)$$

Můžeme tedy spočítat neznámé posuny, případně i reakce.

```
50 r(u) = -K_inverzni_opravena*K(u,p)*r(p);
51 R = K(p,u)*r(u)+K(p,p)*r(p);
```

Nyní máme spočtené výsledné neznámé posuny, s jejichž pomocí dokážeme spočítat energii, uloženou v jednotlivých prutech, potažmo i čas, kdy dojde k selhání dalšího prutu. K tomu opět použijeme funkci *find\_weakest\_link*. Tato funkce opět najde nejslabší prut a čas jeho kolapsu, tyto hodnoty zapíšeme na další řádek výstupní matice *Fail\_Matrix* a můžeme znovu opakovat výpočet do té doby, než tento čas dosáhne předem stanoveného limitu, což zajistíme pomocí while cyklu.

Pozn.: V případě, kdy postupným odečítáním příspěvků jednotlivých porušených prutů do matice  $K(u,u)$  dosáhneme toho, že se tato matice stane singulární, dochází ke kolapsu celého systému.

### 3.2.3 Výstupy

Výstupem funkce *weakest\_link* a celého programu je matice *Fail\_Matrix* obsahující v prvním sloupci čísla porušených prutů a ve druhém sloupci časy kolapsů.

## 3.3 Zdrojový kód

### 3.3.1 Úvodní část

```
1 %% EXAMPLE 01
2
3 xz = [0,0;8,0;0,-6;4,-3]; % souradnice stycniku VSTUP
4 elems = [1,3;1,2;1,4;3,4;4,2]; % topologie soustavy
5 p = [1,2,4,5,7]; % predepsane posuny
6 sigma_crit = ones(size(elems,1),1)*1e4; % kriticke napeti
7 A = ones(size(elems,1),1)*0.01; % prurezy jednotlivych prutu
8 E = ones(size(elems,1),1)*10e6; % moduly pruznosti jednotlivych prutu
9 r = zeros(length(xz(:)),1); limit = 10;
10 r(7) = 0.001; % predepsane zatizeni posunem
11
12 [Fail_Matrix] = weakest_link(xz,elems, p, r, E, A, sigma_crit, limit);
13 Fail_Matrix(:,1) % porusene pruty
14 Fail_Matrix(:,2) % casy poruseni
15
```

### 3.3.2 Function weakest\_link

```
1 function [Fail_Matrix]=weakest_link(xz,elems,p,r,E,A,sigma_crit,limit)
2
3 k = size(xz,1); % pocet stycniku
4 n = size(elems,1); % pocet prutu
5 K = zeros(2*k,2*k); % matice tuhosti plna nul, rozmer 2k x 2k
6 u = setdiff(1:2*k, p); % nezname posuny
7 l = zeros(n,1); % delky prutu
8 Fail_Matrix = zeros(n,2); % vysledna vystupni matice
9 E_crit = zeros(n,1); % matice kritickyh energi
10 Eng = zeros(n,1); % matice energie jednotlivych prutu
11 Temp = zeros(n,2); % pomocna matice s casy kolapsu jednotlivych prutu
12
13 %% SESTAVENI MATICE KRITICKYH ENERGI
14 for q = 1:n
15     [E_critical] = critical_energy(xz(elems(q,1),1),
        xz(elems(q,2),1),xz(elems(q,1),2),xz(elems(q,2),2),E(q),A(q),
        sigma_crit(q));
```



```

16     E_crit(q,1)= [E_critical];
17 end
18
19 %% SESTAVENI MATICE TUHOSTI KONSTRUKCE
20 for i = 1:n
21     [Ki,li] = matice_tuhosti(xz (elems (i,1),1),xz (elems (i,2),1),
        xz (elems (i,1),2),xz (elems (i,2),2),E(i),A(i)); % matice tuhosti
        prutu, rozmer 4 x 4
22     id = [2*elems (i,1)-1,2*elems (i,1),2*elems (i,2)-1,2*elems (i,2)];
        % kodova cisla
23     K(id,id) = K(id,id)+Ki; % prispevek lokalni matice tuhosti do
        globalni matice tuhosti podle kodovych cisel
24     l(i) = li;
25 end
26
27 r(u) = -K(u,u)\(K(u,p)*r(p)); % vypocet posunu
28
29 [tmin,emin] = find_weakest_link( xz, elems, E, A, E_crit, r);
30 Fail_Matrix(1,1) = emin;
31 Fail_Matrix(1,2) = tmin;
32
33 pom = 2;
34 while( tmin < limit )
35
36     %% RANK-ONE-UPDATE
37     [c,s] = sinus_cosinus(xz (elems (emin,1),1),xz (elems (emin,2),1),
        xz (elems (emin,1),2),xz (elems (emin,2),2));
38     [v] = rank_one_update(k,c,s,emin,u,elems,E,A,l);
39
40     %% OPRAVA INVERZNI MATICE
41     INV=inv(K(u,u));
42     K_inverzni_opravena = INV+(INV*v*v'*INV)/(1-v'*INV*v);
43
44     %% OPRAVA GLOBALNI MATICE TUHOSTI
45     [Kpr,lp] = matice_tuhosti(xz (elems (emin,1),1),xz (elems (emin,2),1),
        xz (elems (emin,1),2),xz (elems (emin,2),2),E(emin),A(emin));
46     id = [2*elems (emin,1)-1,2*elems (emin,1),
        2*elems (emin,2)-1,2*elems (emin,2)];
47     K(id,id) = K(id,id)-Kpr;
48     E(emin) = 0;
49
50     %% VYPOCET NOVEHO r
51     r(u) = -K_inverzni_opravena*K(u,p)*r(p); % posuny
52     R = K(p,u)*r(u)+K(p,p)*r(p); % reakce
53
54     [tmin,emin] = find_weakest_link( xz, elems, E, A, E_crit, r);
55     Fail_Matrix(pom,1) = emin;
56     Fail_Matrix(pom,2) = tmin;
57     pom = pom+1;
58 end % ukonceni while cyklu
59
60 end % konec funkce

```

### 3.3.3 Ostatní funkce

```

1 function [E_critical] = critical_energy( xa,xb,za,zb,Ee,Ae,sigma_crit )
2
3 dxab = xb-xa; dzab = zb-za;
4 le = sqrt((dxab)^2 + (dzab)^2);
5 E_critical = 1/2*sigma_crit*sigma_crit*Ae*le/Ee;
6
7 end

```

```

1 function [Ke,le] = matice_tuhosti( xa,xb,za,zb,Ee,Ae )
2
3 dxab = xb-xa; dzab = zb-za;
4 le = sqrt((dxab)^2 + (dzab)^2);
5 c = dxab/le; s = dzab/le;
6 Ke = Ee*Ae/le*[c;s;-c;-s]*[c,s,-c,-s];
7
8 end

1 function [tmin,emin] = find_weakest_link(xz,elems,E,A,E_crit,r)
2
3 Normalovky = zeros(size(elems,1),1);
4 pomocna = size(elems,1);
5
6 for m = 1:size(elems,1)
7     Km = matice_tuhosti(xz(elems(m,1),1),xz(elems(m,2),1),
8         xz(elems(m,1),2),xz(elems(m,2),2),E(m),A(m)));
9     Energie = 1/2*[r(2*elems(m,1)-1,1),r(2*elems(m,1),1),
10        r(2*elems(m,2)-1,1), r(2*elems(m,2),1)]*Km*
11        [r(2*elems(m,1)-1,1);r(2*elems(m,1),1);
12         r(2*elems(m,2),1,1);r(2*elems(m,2),1)];
13     [N] = normalova_sila(xz(elems(m,1),1),xz(elems(m,2),1),
14        xz(elems(m,1),2),xz(elems(m,2),2),E(m),A(m),r(2*elems(m,1)-1,1),
15        r(2*elems(m,1),1),r(2*elems(m,2)-1,1),r(2*elems(m,2),1)));
16     Normalovky(m,1) = N;
17     Eng(m,1) = Energie; % matice energii jednotlivych prutu
18     %% PRIPADNE ZABRANENI PORUSENI TLACENYCH PRUTU
19     % if N<0
20     %     Eng(m,1)=0;
21     % end
22 end

18 %% VYBER PRUTU, K JEHOZ PORUSENI DOJDE
19 Temp = sqrt( E_crit ./ Eng );
20 [tmin,emin] = min(Temp(:,1));
21
22 end

1 function [N] = normalova_sila (xa,xb,za,zb,E,A,u_a,w_a,u_b,w_b)
2
3 dxab = xb-xa; dzab = zb-za;
4 l = sqrt((dxab)^2 + (dzab)^2);
5 c = dxab/l; s = dzab/l;
6 N = E*A/l*[-c,-s,c,s]*[u_a;w_a;u_b;w_b];
7
8 end

1 function [c,s] = sinus_cosinus(xa,xb,za,zb)
2
3 dxab = xb-xa; dzab = zb-za;
4 l = sqrt((dxab)^2 + (dzab)^2);
5 c = dxab/l; s = dzab/l;
6
7 end

```

```

1 function [v] = rank_one_update(k,c,s,emin,u,elems,E,A,l)
2
3 v1 = zeros(2*k,1);
4 v1(2*elems(emin,1)-1,1) = c;
5 v1(2*elems(emin,1),1) = s;
6 v1(2*elems(emin,2)-1,1) = -c;
7 v1(2*elems(emin,2),1) = -s;
8 v1 = v1*sqrt(E(emin)*A(emin)/l(emin));
9 PN = length(u);
10 v = zeros(PN,1);
11 for pom = 1:1:PN
12     v(pom,1) = v1(u(pom));
13 end
14
15 end

```

### 3.4 SHRNU TÍ A VÝPIS PODPROGRAMŮ

Program umožňuje analýzu porušení příhradové konstrukce na základě metody nejslabšího článku za použití efektivního způsobu přepočítávání inverze opravené matice tuhosti o příspěvek porušeného prutu, tzv. „Woodbury matrix identity“. Popis tohoto algoritmu a jeho odvození jsou uvedeny v kapitole číslo 5. Program je složen z následujících funkcí:

- *function weakest\_link*
- *function find weakest\_link*
- *function matice\_tuhosti*
- *function sinus\_cosinus*
- *function rank\_one\_update*
- *function critical\_energy*
- *function normalova\_sila*

## 4 POPIS PROGRAMU – METODA ZALOŽENÁ NA ENERGETICKÉM KRITÉRIU

### 4.1 ÚVOD

Tento program je velmi podobný programu pro výpočet metodou nejslabšího článku. Několik funkcí z předchozího programu je použito i zde.

### 4.2 POPIS ZDROJOVÉHO KÓDU

#### 4.2.1 Vstupy

Vstupy zůstávají totožné s programem, popsáním výše.

#### 4.2.2 Tělo programu

Po zadání vstupních parametrů zavoláme funkci *delta\_energy\_matrix*. Tato funkce tvoří základ celého programu podobně jako funkce *weakest\_link* v případě prvního programu. Vstupní parametry zůstávají také stejné: *xz*, *elems*, *p*, *r*, *E*, *sigma\_crit*, *limit*. Výstupem této funkce bude opět matice *Fail\_Matrix* obsahující čísla porušených prutů a ve druhém sloupci časy porušení.

Začátek výpočtu je totožný s prvním programem. Sestavení matice kritických energií je v tomto případě zbytečné. Sestavíme tedy rovnou matici tuhosti neporušené konstrukce, spočítáme neznámé posuny a zjistíme hodnotu energie systému. Připravíme matici *DEM*, kam budeme zapisovat změny energie systému při porušení jednotlivých prutů.

Nyní vstoupíme do while cyklu, který omezíme například časem zastavení zatěžování. Nadefinujeme si matici, kam budeme zapisovat hodnoty disipací jednotlivých prutů (*Matices\_disipaci*) a matici, kam budeme zapisovat časy, kdy dojde k porušení prutů (*Matices\_casu*).

Nyní předpokládejme, že jako první bude porušen prut 1. Konstrukce je tedy o tento prut oslabena a počítáme bez něj. Stejně jako v prvním programu použijeme rank-one update, abychom zjistili inverzi submatice  $K(u,u)$ , ovšem již bez příspěvku prasklého prutu. Opravíme i ostatní submatice a spočítáme, jak se oslabená konstrukce zdeformuje. Případně můžeme opět spočítat i normálové síly a eventuálně zabránit porušení tlačných nebo tažených prutů. Nyní známe vše, co potřebujeme k výpočtu energie oslabeného systému. Spočteme tedy energii systému, který je oproti předchozímu systému oslaben o jeden prut a zjistíme její rozdíl oproti energii neporušené konstrukce. Tuto hodnotu uložíme do matice *DEM*. Dále přikročíme k výpočtu disipace prutu podle vztahu (2.9) a tuto hodnotu si uložíme do matice disipací. Zbývá již spočítat pouze čas, kdy dojde ke kolapsu prutu. Ten spočteme podle vztahu (2.11) tak, že vydělíme hodnotu disipace hodnotou změny energie a podíl odmocníme. Výsledný čas uložíme na odpovídající pozici matice času kolapsů.

```
62 Matices_disipaci(j,1)=D;  
63 Matices_casu(j,1)=sqrt(D/DEM(j,1));
```

Nyní stejný výpočet provedeme za předpokladu, že jako první zkolabuje prut 2, poté výpočet opakujeme pro prut 3 atd. až projedeme všechny pruty. To zařídíme pomocí for cyklu přes

počet prutů. Po tomto for cyklu získáme kompletní matici, obsahující časy, kdy zkolabují jednotlivé pruty. Z této matice stačí vybrat minimum a víme, který prut se poruší jako první. Číslo prutu a čas kolapsu zapíšeme do výsledné výstupní matice `Fail_Matrix`.

```
67 [tmin,emin] = min(Matrice_casu(:,1));
68 Fail_Matrix(pomocna,1)=emin;
69 Fail_Matrix(pomocna,2)=tmin;
```

Předchozím výpočtem jsme zjistili, že praskne prut číslo  $i$ . Jako „původní“ konstrukci tedy nyní předpokládáme konstrukci bez  $i$ -tého prutu. Spočteme její deformaci a energii a stejným postupem hledáme rozdíly energie systému se dvěma prasklými pruty oproti systému s jedním prasklým prutem. Tento postup opakujeme, dokud není splněna podmínka while cyklu.

### 4.2.3 Výstupy

Výstupem funkce `Delta_energy_matrix` a celého programu je stejně jako u prvního programu matice `Fail_Matrix` obsahující v prvním sloupci čísla porušených prutů a ve druhém sloupci časy kolapsů.

## 4.3 ZDROJOVÝ KÓD

### 4.3.1 Úvodní část

```
1 %% EXAMPLE 02
2
3 xz = [0,0;8,0;0,-6;4,-3]; % souradnice stycniku
4 elems = [1,3;1,2;1,4;3,4;4,2]; % topologie soustavy
5 p = [1,2,4,5,7]; % predepsane posuny
6 sigma_crit = ones(size(elems,1),1)*1e4; %kriticke napeti
7 A = ones(size(elems,1),1)*0.01; % Prurezy jednotlivych prutu
8 E = ones(size(elems,1),1)*10e6; % Moduly pruznosti jednotlivych prutu
9 r = zeros(length(xz(:)),1); limit = 10;
10 r(7) = 0.001; % predepsane zatizeni posunem
11
12 [Fail_Matrix] = delta_energy_matrix(xz,elems, p, r, E, A,sigma_crit,
13 limit);
14 Fail_Matrix(:,1)
15 Fail_Matrix(1,2)
```

### 4.3.2 Function `Delta_energy_matrix`

```
1 %% DELTA ENERGY MATRIX
2
3 function [Fail_Matrix] = delta_energy_matrix(xz,elems,
4 p,r,E,A,sigma_crit,limit)
5
6 k = size(xz,1); % pocet stycniku
7 n = size(elems,1); % pocet prutu
8 l = zeros(n,1); % delky prutu
9 Fail_Matrix = zeros(n,2);
10 K = zeros(2*k,2*k); % matice plna nul, rozmer 2k x 2k
11 u = setdiff(1:2*k, p); % nezname posuny
12 tmin = 0;
13 pomocna = 1;
14
15 %% SESTAVENI MATICE TUHOSTI KONSTRUKCE
16
17 for i = 1:n
```

```

18     [Ki,li] = matice_tuhosti(xz(elems(i,1),1),xz(elems(i,2),1),
    xz(elems(i,1),2),xz(elems(i,2),2),E(i),A(i)); % matice tuhosti
    prutu,rozmer 4x4
19     id = [2*elems(i,1)-1,2*elems(i,1),2*elems(i,2)-1,2*elems(i,2)];
    % kodova cisla
20     K(id,id) = K(id,id)+Ki; % prispevek lokalni matice tuhosti do
    globalni matice tuhosti podle kodovych cisel
21     l(i) = li;
22 end
23
24 r(u) = -K(u,u)\(K(u,p)*r(p)); % posuny
25
26 Energy = 1/2*r'*K*r; % energie systemu
27 DEM = zeros(n,1); % matice rozdilu energii
28
29 while( tmin < limit )
30
31     Matice_disipaci = zeros(n,1);
32     Matice_casu = zeros(n,1);
33
34     for j = 1:n
35         K_pom = K;
36         [c,s] = sinus_cosinus(xz(elems(j,1),1),xz(elems(j,2),1),
    xz(elems(j,1),2),xz(elems(j,2),2)));
37
38         %% RANK ONE UPDATE
39         [v] = rank_one_update(k,c,s,j,u,elems,E,A,l);
40         INV= inv(K_pom(u,u));
41         K_inverzni_opravena = INV+(INV*v*v'*INV)/(1-v'*INV*v);
42
43         %% OPRAVA GLOBALNI MATICE TUHOSTI
44         [Kpr,lp] = matice_tuhosti(xz(elems(j,1),1),xz(elems(j,2),1),
    xz(elems(j,1),2),xz(elems(j,2),2),E(j),A(j)); % matice
    tuhosti prutu, rozmer 4 x 4
45         id = [2*elems(j,1)-1,2*elems(j,1),2*elems(j,2)-
    1,2*elems(j,2)];
46         K_pom(id,id) = K_pom(id,id)-Kpr;
47
48         %% VYPOCET NOVEHO R
49         r(u) = -K_inverzni_opravena*K_pom(u,p)*r(p);
50
51         energeticky_rozdil = Energy-1/2*r'*K_pom*r;
52
53         DEM(j,1) = energeticky_rozdil;
54
55         %% VYRAZENI TLACENYCH PRUTU
56         % [N] = normalova_sila(xz(elems(j,1),1),xz(elems(j,2),1),
    xz(elems(j,1),2),xz(elems(j,2),2),E(j),A(j),
    r(2*elems(j,1)-1,1),r(2*elems(j,1),1),
    r(2*elems(j,2)-1,1),r(2*elems(j,2),1)));
57         % if N<0
58         % DEM(j,1)=0;
59         % end
60
61         %% VYPOCET CASU KOLAPSU PRUTU
62         [D] = Disipace(sigma_crit(j),A(j),xz(elems(j,1),1),
    xz(elems(j,2),1),xz(elems(j,1),2),xz(elems(j,2),2),E(j));
63         Matice_disipaci(j,1) = D;
64         Matice_casu(j,1) = sqrt(D/DEM(j,1));
65
66     end % konec for cyklu
67
68     [tmin,emin] = min(Matice_casu(:,1));

```

```

68 Fail_Matrix(pomocna,1) = emin; % zapis do vystupni matice
69 Fail_Matrix(pomocna,2) = tmin; % zapis do vystupni matice
70 pomocna = pomocna+1;
71
72 %% VYPOCET KONSTRUKCE S PRASKLYM PRUTEM EMIN
73 [c,s] = sinus_cosinus(xz(elems(emin,1),1),xz(elems(emin,2),1),
xz(elems(emin,1),2),xz(elems(emin,2),2)));
74 [v] = rank_one_update(k,c,s,emin,u,elems,E,A,1);
75 K_inverzni_opravena = inv(K(u,u))+(inv(K(u,u))*v*v'*inv(K(u,u)))/
(1-v'*inv(K(u,u))*v);
76
77
78 [Kpr,lp] = matice_tuhosti(xz(elems(emin,1),1),xz(elems(emin,2),1),
xz(elems(emin,1),2),xz(elems(emin,2),2),E(emin),A(emin)));
79 id = [2*elems(emin,1)-1,2*elems(emin,1),2*elems(emin,2)-
1,2*elems(emin,2)];
80 K(id,id) = K(id,id)-Kpr;
81 E(emin) = 0;
82
83 r(u) = -K_inverzni_opravena*K(u,p)*r(p);
84 R = K(p,u)*r(u)+K(p,p)*r(p);
85 Energy = 1/2*r'*K*r; % energie kce s prasklym prutem emin
86
87 end % konec while cyklu
88 end

```

### 4.3.3 Ostatní funkce

```

1 function [D] = Disipace(sigma_crit,A,xa,xb,za,zb,E)
2
3 dxab = xb-xa; dzab = zb-za;
4 l = sqrt((dxab)^2 + (dzab)^2);
5 D = 1/2*sigma_crit*sigma_crit*A*l/E;
6 end

```

Všechny ostatní použité funkce jsou totožné s prvním programem a jsou již popsány v kapitole 3.3.3.

## 4.4 SHRNUTÍ A VÝPIS PODPROGRAMŮ

Program umožňuje analýzu porušení příhradové konstrukce pomocí metody, založené na energetickém kritériu. Opět je zde použit efektivní algoritmus pro výpočet inverze opravené matice, který je detailně popsán v následující kapitole. Program je složen z následujících funkcí:

- *function Delta\_energy\_matrix*
- *function Disipace*
- *function matice\_tuhosti*
- *function sinus\_cosinus*
- *function rank\_one\_update*
- *function critical\_energy*
- *function normalova\_sila*

## 5 ODVOZENÍ „WOODBURY MATRIX IDENTITY“

### 5.1 ÚVOD

Woodbury matrix identity je matematický vztah, který nám umožňuje na základě znalosti inverze původní matice zjistit inverzi matice, která se od této původní liší součinem tří matic odpovídajících rozměrů. V literatuře se můžeme setkat s dalšími názvy této formule, např. lemma o inverzní matici, Sherman-Morrison-Woodbury formula nebo pouze Woodbury formula.

Formule má následující tvar:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1} \quad (5.1)$$

Kde  $A$  je původní matice velikosti  $n \times n$ ,  $U$  je matice velikosti  $n \times k$ ,  $C$  matice velikosti  $k \times k$ ,  $V$  matice velikosti  $k \times n$ .

### 5.2 ODVOZENÍ

Následující část práce obsahuje odvození výsledných vztahů na základě informací ze serveru [en.wikipedia.org](http://en.wikipedia.org).

#### 5.2.1 Odvození pomocí blokové eliminace

Budeme řešit následující blok:

$$\begin{bmatrix} A & U \\ V & -C^{-1} \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \quad (5.2)$$

Což je ekvivalentní problému:

$$(A + UCV)X = I \quad (5.3)$$

Roznásobením dostaneme:

$$AX + UY = I \quad (5.4)$$

$$VX - C^{-1}Y = 0 \quad (5.5)$$

Z první rovnice vyjádříme  $X$  a dosadíme do druhé:

$$\begin{aligned} X &= A^{-1}(I - UY) \\ VA^{-1}(I - UY) - C^{-1}Y &= 0 \end{aligned} \quad (5.6)$$

Odstraníme závorku a upravíme:

$$\begin{aligned} VA^{-1} - VA^{-1}UY - C^{-1}Y &= 0 \\ VA^{-1} - (VA^{-1}U + C^{-1})Y &= 0 \\ VA^{-1} &= (VA^{-1}U + C^{-1})Y \end{aligned} \quad (5.7)$$



Celou rovnici vynásobíme zleva maticí  $(VA^{-1}U + C^{-1})^{-1}$  a tím vyjádříme  $Y$ :

$$Y = (VA^{-1}U + C^{-1})^{-1}VA^{-1} \quad (5.8)$$

Nyní dosadíme  $Y$  do rovnice (5.4):

$$AX + U(VA^{-1}U + C^{-1})^{-1}VA^{-1} = I \quad (5.9)$$

Rovnici vynásobíme zleva  $A^{-1}$ :

$$IX + A^{-1}U(VA^{-1}U + C^{-1})^{-1}VA^{-1} = A^{-1}$$

Vyjádříme  $X$ :

$$X = A^{-1} - A^{-1}U(VA^{-1}U + C^{-1})^{-1}VA^{-1}$$

Hledaný vzorec je tedy odvozen:

$$X = (A + UCV)^{-1} = A^{-1} - A^{-1}U(VA^{-1}U + C^{-1})^{-1}VA^{-1} \quad (5.10)$$

### 5.2.2 Odvození pomocí LDU rozkladu

LDU rozklad je rozklad na součin dolní trojúhelníkové, diagonální a horní trojúhelníkové matice. Při odvozování budeme vycházet z následujícího tvaru:

$$\begin{bmatrix} A & U \\ V & C \end{bmatrix} \quad (5.11)$$

Nejdříve najdeme dolní trojúhelníkovou matici:

$$\begin{bmatrix} I & 0 \\ -VA^{-1} & I \end{bmatrix} \begin{bmatrix} A & U \\ V & C \end{bmatrix} = \begin{bmatrix} IA + 0V & IU + 0C \\ -VA^{-1}A + V & -VA^{-1}U + C \end{bmatrix} = \\ \begin{bmatrix} A & U \\ 0 & C - VA^{-1}U \end{bmatrix} \quad (5.12)$$

Podobně získáme i horní trojúhelníkovou matici:

$$\begin{bmatrix} A & U \\ V & C \end{bmatrix} \begin{bmatrix} I & -A^{-1}U \\ 0 & I \end{bmatrix} = \begin{bmatrix} AI + U0 & -AA^{-1}U + UI \\ VI + C0 & -VA^{-1}U + CI \end{bmatrix} = \\ \begin{bmatrix} A & 0 \\ V & C - VA^{-1}U \end{bmatrix} \quad (5.13)$$

Kombinací předchozích dvou postupů získáme matici diagonální:

$$\begin{bmatrix} I & 0 \\ -VA^{-1} & I \end{bmatrix} \begin{bmatrix} A & U \\ V & C \end{bmatrix} \begin{bmatrix} I & -A^{-1}U \\ 0 & I \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & C - VA^{-1}U \end{bmatrix} \quad (5.14)$$

Původní matici vyjádříme z 5.14:

$$\begin{bmatrix} I & 0 \\ -VA^{-1} & I \end{bmatrix}^{-1} \begin{bmatrix} I & 0 \\ -VA^{-1} & I \end{bmatrix} \begin{bmatrix} A & U \\ V & C \end{bmatrix} \begin{bmatrix} I & -A^{-1}U \\ 0 & I \end{bmatrix} \begin{bmatrix} I & -A^{-1}U \\ 0 & I \end{bmatrix}^{-1} = \\ \begin{bmatrix} I & 0 \\ -VA^{-1} & I \end{bmatrix}^{-1} \begin{bmatrix} A & 0 \\ 0 & C - VA^{-1}U \end{bmatrix} \begin{bmatrix} I & -A^{-1}U \\ 0 & I \end{bmatrix}^{-1} \quad (5.15)$$

Získáme tedy výsledný LDU rozklad na součin horní trojúhelníkové, dolní trojúhelníkové a diagonální matice:

$$\begin{bmatrix} A & U \\ V & C \end{bmatrix} = \begin{bmatrix} I & 0 \\ VA^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & C - VA^{-1}U \end{bmatrix} \begin{bmatrix} I & A^{-1}U \\ 0 & I \end{bmatrix} \quad (5.16)$$

Nyní invertujeme obě strany rovnice:

$$\begin{aligned} \begin{bmatrix} A & U \\ V & C \end{bmatrix}^{-1} &= \begin{bmatrix} I & A^{-1}U \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} A & 0 \\ 0 & C - VA^{-1}U \end{bmatrix}^{-1} \begin{bmatrix} I & 0 \\ VA^{-1} & I \end{bmatrix}^{-1} = \\ &= \begin{bmatrix} I & -A^{-1}U \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & (C - VA^{-1}U)^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -VA^{-1} & I \end{bmatrix} = \\ &= \begin{bmatrix} IA^{-1} + 0 & I0 - A^{-1}U(C - VA^{-1}U)^{-1} \\ 0A^{-1} + 0 & 0 + I(C - VA^{-1}U)^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -VA^{-1} & I \end{bmatrix} = \\ &= \begin{bmatrix} A^{-1} & -A^{-1}U(C - VA^{-1}U)^{-1} \\ 0 & (C - VA^{-1}U)^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -VA^{-1} & I \end{bmatrix} = \\ &= \begin{bmatrix} IA^{-1} + A^{-1}U(C - VA^{-1}U)^{-1}VA^{-1} & A^{-1}0 - A^{-1}U(C - VA^{-1}U)^{-1} \\ 0I - (C - VA^{-1}U)^{-1}VA^{-1} & 0 + I(C - VA^{-1}U)^{-1} \end{bmatrix} \end{aligned}$$

Získáme tedy tento vztah:

$$\begin{bmatrix} A & U \\ V & C \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}U(C - VA^{-1}U)^{-1}VA^{-1} & -A^{-1}U(C - VA^{-1}U)^{-1} \\ (C - VA^{-1}U)^{-1}VA^{-1} & (C - VA^{-1}U)^{-1} \end{bmatrix} \quad (5.17)$$

Totéž můžeme provést i jiným způsobem. Nejdříve najdeme horní trojúhelníkovou matici:

$$\begin{aligned} \begin{bmatrix} I & -UC^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A & U \\ V & C \end{bmatrix} &= \\ \begin{bmatrix} IA - UC^{-1}V & IU - UC^{-1}C \\ 0A + IV & 0U + IC \end{bmatrix} &= \begin{bmatrix} A - UC^{-1}V & 0 \\ V & C \end{bmatrix} \end{aligned} \quad (5.18)$$

Podobně získáme i dolní trojúhelníkovou matici:

$$\begin{aligned} \begin{bmatrix} A & U \\ V & C \end{bmatrix} \begin{bmatrix} I & 0 \\ -C^{-1}V & I \end{bmatrix} &= \\ \begin{bmatrix} AI - UC^{-1}V & A0 + UI \\ VI - CC^{-1}V & V0 + CI \end{bmatrix} &= \begin{bmatrix} A - UC^{-1}V & U \\ 0 & C \end{bmatrix} \end{aligned} \quad (5.19)$$

Kombinací předchozích dvou postupů získáme matici diagonální:

$$\begin{bmatrix} I & -UC^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A & U \\ V & C \end{bmatrix} \begin{bmatrix} I & 0 \\ -C^{-1}V & I \end{bmatrix} = \begin{bmatrix} A - UC^{-1}V & 0 \\ 0 & C \end{bmatrix} \quad (5.20)$$

Nyní opět vyjádříme původní matici z 5.20:

$$\begin{aligned} & \begin{bmatrix} I & -UC^{-1} \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} I & -UC^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A & U \\ V & C \end{bmatrix} \begin{bmatrix} I & 0 \\ -C^{-1}V & I \end{bmatrix} \begin{bmatrix} I & 0 \\ -C^{-1}V & I \end{bmatrix}^{-1} = \\ & \begin{bmatrix} I & -UC^{-1} \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} A - UC^{-1}V & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} I & 0 \\ -C^{-1}V & I \end{bmatrix}^{-1} \end{aligned} \quad (5.21)$$

Výsledný rozklad potom bude v následujícím tvaru:

$$\begin{bmatrix} A & U \\ V & C \end{bmatrix} = \begin{bmatrix} I & UC^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A - UC^{-1}V & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} I & 0 \\ -C^{-1}V & I \end{bmatrix} \quad (5.22)$$

Opět invertujeme obě strany rovnice:

$$\begin{aligned} & \begin{bmatrix} A & U \\ V & C \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ -C^{-1}V & I \end{bmatrix}^{-1} \begin{bmatrix} A - UC^{-1}V & 0 \\ 0 & C \end{bmatrix}^{-1} \begin{bmatrix} I & UC^{-1} \\ 0 & I \end{bmatrix}^{-1} = \\ & \begin{bmatrix} I & 0 \\ -C^{-1}V & I \end{bmatrix} \begin{bmatrix} (A - UC^{-1}V)^{-1} & 0 \\ 0 & C^{-1} \end{bmatrix} \begin{bmatrix} I & -UC^{-1} \\ 0 & I \end{bmatrix} = \\ & \begin{bmatrix} I(A - UC^{-1}V)^{-1} + 0 & 0 + 0 \\ -C^{-1}V(A - UC^{-1}V)^{-1} + 0 & 0 + IC^{-1} \end{bmatrix} \begin{bmatrix} I & -UC^{-1} \\ 0 & I \end{bmatrix} = \\ & \begin{bmatrix} (A - UC^{-1}V)^{-1} & 0 \\ -C^{-1}V(A - UC^{-1}V)^{-1} & C^{-1} \end{bmatrix} \begin{bmatrix} I & -UC^{-1} \\ 0 & I \end{bmatrix} \end{aligned}$$

Získáváme tedy tento vztah:

$$\begin{aligned} & \begin{bmatrix} A & U \\ V & C \end{bmatrix}^{-1} = \\ & \begin{bmatrix} (A - UC^{-1}V)^{-1} & -(A - UC^{-1}V)^{-1}UC^{-1} \\ -C^{-1}V(A - UC^{-1}V)^{-1} & C^{-1}V(A - UC^{-1}V)^{-1}UC^{-1} + C^{-1} \end{bmatrix} \end{aligned} \quad (5.23)$$

Nyní porovnáme vztahy 5.17 a 5.23:

$$\begin{aligned} & \begin{bmatrix} A^{-1} + A^{-1}U(C - VA^{-1}U)^{-1}VA^{-1} & -A^{-1}U(C - VA^{-1}U)^{-1} \\ (C - VA^{-1}U)^{-1}VA^{-1} & (C - VA^{-1}U)^{-1} \end{bmatrix} = \\ & \begin{bmatrix} (A - UC^{-1}V)^{-1} & -(A - UC^{-1}V)^{-1}UC^{-1} \\ -C^{-1}V(A - UC^{-1}V)^{-1} & C^{-1}V(A - UC^{-1}V)^{-1}UC^{-1} + C^{-1} \end{bmatrix} \end{aligned}$$

Porovnáním pozice (1,1) získáme výslednou formuli:

$$(A - UC^{-1}V)^{-1} = A^{-1} + A^{-1}U(C - VA^{-1}U)^{-1}VA^{-1} \quad (5.24)$$

### 5.3 PŘÍMÝ DŮKAZ

Woodbury-matrix identity nám umožňuje zjistit inverzní matici. Abychom ověřili správnost, budeme tedy dokazovat, zda součin původní matice a inverzní matice dá jednotkovou matici.

### 5.3.1 Tvar odvozený pomocí blokové eliminace

Vzorec 5.10, odvozený pomocí blokové eliminace má následující tvar:

$$(A + UCV)^{-1} = [A^{-1} - A^{-1}U(VA^{-1}U + C^{-1})^{-1}VA^{-1}]$$

Ověříme, zda součin matic je roven jednotkové matici:

$$\begin{aligned}(A + UCV)(A + UCV)^{-1} &= (A + UCV)[A^{-1} - A^{-1}U(VA^{-1}U + C^{-1})^{-1}VA^{-1}] = \\ AA^{-1} - AA^{-1}U(VA^{-1}U + C^{-1})^{-1}VA^{-1} + UCVA^{-1} - UCVA^{-1}U(VA^{-1}U + \\ C^{-1})^{-1}VA^{-1} &= \\ I + UCVA^{-1} - U(VA^{-1}U + C^{-1})^{-1}VA^{-1} - UCVA^{-1}U(VA^{-1}U + C^{-1})^{-1}VA^{-1} &= \end{aligned}$$

Vytkneme zleva  $(U + UCVA^{-1}U)$ :

$$I + UCVA^{-1} - (U + UCVA^{-1}U)(VA^{-1}U + C^{-1})^{-1}VA^{-1} =$$

Nyní ze součinu vytkneme ještě zleva  $UC$ :

$$\begin{aligned}I + UCVA^{-1} - UC(C^{-1} + VA^{-1}U)(VA^{-1}U + C^{-1})^{-1}VA^{-1} &= \\ I + UCVA^{-1} - UCVA^{-1} &= I. \end{aligned}$$

Důkaz je hotov.

### 5.3.2 Tvar odvozený pomocí LDU rozkladu

Tvar 5.24, odvozený pomocí LDU rozkladu:

$$(A - UC^{-1}V)^{-1} = (A^{-1} + A^{-1}U(C - VA^{-1}U)^{-1}VA^{-1})$$

Důkaz provedeme obdobně jako v předchozím případě:

$$\begin{aligned}(A - UC^{-1}V)(A - UC^{-1}V)^{-1} &= \\ (A - UC^{-1}V)(A^{-1} + A^{-1}U(C - VA^{-1}U)^{-1}VA^{-1}) &= \\ AA^{-1} + U(C - VA^{-1}U)^{-1}VA^{-1} - UC^{-1}VA^{-1} - UC^{-1}VA^{-1}U(C - VA^{-1}U)^{-1}VA^{-1} &= \end{aligned}$$

Vytkneme zprava  $(C - VA^{-1}U)^{-1}VA^{-1}$ :

$$= I - UC^{-1}VA^{-1} + (U - UC^{-1}VA^{-1}U)(C - VA^{-1}U)^{-1}VA^{-1} =$$

Nyní vytkneme ze součinu zleva  $UC^{-1}$ :

$$\begin{aligned}I - UC^{-1}VA^{-1} + UC^{-1}(C - VA^{-1}U)(C - VA^{-1}U)^{-1}VA^{-1} &= \\ I - UC^{-1}VA^{-1} + UC^{-1}VA^{-1} &= I. \end{aligned}$$

Důkaz je hotov.

## 5.4 SPECIÁLNÍ PŘÍPAD: SHERMAN-MORRISON FORMULA

### 5.4.1 Popis

Sherman-Morrison formula je speciálním případem Woodbury-matrix identity. Připomeňme její obecný vzorec:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(VA^{-1}U + C^{-1})^{-1}VA^{-1} \quad (5.25)$$

Sherman-Morrison formula platí pro případ, kdy  $A$  je matice  $n \times n$ ,  $C$  je matice  $1 \times 1$ ,  $u$  je sloupcový vektor  $n \times 1$  a  $v^T$  je řádkový vektor  $1 \times n$ . Potom platí:

$$(A + uv^T)^{-1} = A^{-1} - A^{-1}u(v^T A^{-1}u + 1)^{-1}v^T A^{-1} \quad (5.26)$$

Člen  $v^T A^{-1}u + 1$  je skalár a proto vztah můžeme převést do tohoto tvaru, za předpokladu, že  $A$  je invertovatelná matice a zároveň  $1 + v^T A^{-1}u \neq 0$ :

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1}u} \quad (5.27)$$

### 5.4.2 Použití

Za předpokladu, že známe inverzi matice  $A$  nám tato formule umožňuje spočítat inverzi matice opravené o součin  $uv^T$ . Hlavní výhodou tohoto postupu je jeho výpočetní nenáročnost. V obecném případě, kdy máme matici  $A$  o rozměru  $n \times n$  a vektory  $u, v$ , které mají  $n$  řádků, je výpočetní složitost tohoto procesu  $O(3n^2)$ . V našem konkrétním případě toho využijeme při korekci matice tuhosti, kdy poškozením prutu dojde k opravě původní matice tuhosti právě o příspěvek porušeného prutu. Tento příspěvek lze vyjádřit následujícím součinem:

$$\frac{EA}{l} \begin{bmatrix} c \\ s \\ -c \\ -s \end{bmatrix} [c \quad s \quad -c \quad -s] \quad (5.28)$$

Konstantu před součinem tedy můžeme „rozložit“ do obou vektorů a počítáme s vektory v následujících tvarech:

$$u = \sqrt{\frac{EA}{l}} \begin{bmatrix} c \\ s \\ -c \\ -s \end{bmatrix} \quad (5.29)$$

$$v^T = \sqrt{\frac{EA}{l}} [c \quad s \quad -c \quad -s] \quad (5.30)$$

Porušením prutu samozřejmě musíme jeho příspěvek do původní matice odečíst, proto je ještě potřeba ve výsledném vztahu změnit znaménka. Výsledný tvar, který použijeme při výpočtu, bude potom vypadat takto:

$$(A - uv^T)^{-1} = A^{-1} + \frac{A^{-1}uv^T A^{-1}}{1 - v^T A^{-1}u} \quad (5.31)$$

### 5.4.3 Ověření

Ověříme správnost vzorce.

$$(A + uv^T)(A + uv^T)^{-1} = (A + uv^T) \left( A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1+v^T A^{-1}u} \right) =$$

$$AA^{-1} + uv^T A^{-1} - \frac{AA^{-1}uv^T A^{-1}}{1+v^T A^{-1}u} + \frac{uv^T A^{-1}uv^T A^{-1}}{1+v^T A^{-1}u} =$$

$$I + uv^T A^{-1} - \frac{uv^T A^{-1} - uv^T A^{-1}uv^T A^{-1}}{1-v^T A^{-1}u} =$$

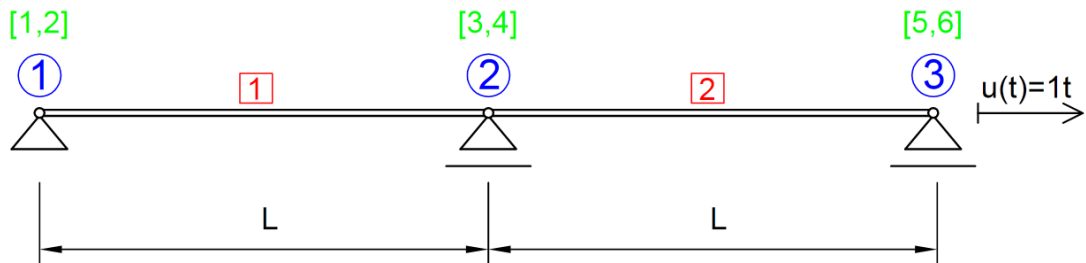
Nyní v čitateli vytkneme zleva  $u$  a zprava  $v^T A^{-1}$ :

$$I - uv^T A^{-1} - \frac{uv^T (1-v^T A^{-1}u)v^T A^{-1}}{1-v^T A^{-1}u} = I - uv^T A^{-1} + uv^T A^{-1} = I.$$

Vztah tedy skutečně platí.

## 6 PŘÍKLAD 1 - RUČNÍ VÝPOČET

### 6.1 SCHÉMA, ZATÍŽENÍ



Obr 6.1: Schéma konstrukce

Konstrukce se skládá ze dvou prutů za sebou. Červeně jsou označena čísla prutů, modře čísla styčníků a zeleně kódová čísla odpovídající deformacím. Uvažujeme pouze namáhání tahem/tlakem, kódová čísla proto odpovídají vodorovnému a svislému posunu. S pootočením nepočítáme. Konstrukci zatěžujeme posunem třetího styčnicku. Tento posun narůstá lineárně v čase.

Oba pruty mají shodnou délku  $L=1m$ , průřezovou plochu  $A=1m^2$ , modul pružnosti  $E=1Pa$ . Hodnota energie, při které dojde k porušení prutu  $E_{crit}=10J$ . Těto kritické hodnotě energie odpovídá kritické napětí  $\sigma_{crit}=4,472Pa$ , neboť:

$$E_{crit} = \frac{1}{2} \sigma_{crit}^2 \frac{AL}{E}$$

$$10 = \frac{1}{2} \sigma_{crit}^2 \cdot 1 \Rightarrow \sigma_{crit} = \sqrt{20} \cong 4,472$$

### 6.2 VÝPOČET - METODA NEJSLABŠÍHO ČLÁNKU

Ruční výpočet začneme sestavením matice tuhosti v tahu/tlaku. K tomu potřebujeme matice tuhosti jednotlivých prutů.

Tato matice má obecný tvar:

$$K = \frac{EA}{L} \begin{bmatrix} c \\ s \\ -c \\ -s \end{bmatrix} [c \quad s \quad -c \quad -s] = \frac{EA}{L} \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -s^2 & -s^2 & cs & s^2 \end{bmatrix}$$

Kde  $c = \frac{\Delta X_{ab}}{l_{ab}}$ ,  $s = \frac{\Delta Z_{ab}}{l_{ab}}$ .  $\Delta Z_{ab}$  resp.  $\Delta X_{ab}$  jsou rozdíly souřadnic styčnicků prutu.

Pro pruty 1 a 2 mají matice tvar:

$$K_1 = \frac{1}{1} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}$$

$$K_2 = \frac{1}{1} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} 3 \\ 4 \\ 5 \\ 6 \end{matrix}$$

Modře jsou vyznačena odpovídající kódová čísla. Podle nich nyní sestavíme globální matici tuhosti.

$$K = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 2 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}$$

Matice předepsaných posunů bude mít tvar:

$$r(p) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1t \\ 0 \end{bmatrix}$$

Neznámým posunem tedy zůstává pouze vodorovný posun styčnicku 2. Tomuto přemístění odpovídá kódové číslo 3. Nyní můžeme přikročit k jeho výpočtu:

$$K(u, u) r(u) + K(u, p) r(p) = f(u)$$

Silové zatížení je nulové. Po dosazení dostaneme:

$$[2] r(u) + [-1 \ 0 \ 0 \ -1 \ 0] \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1t \\ 0 \end{bmatrix} = [0]$$

$$2 r(u) - 1t = 0$$

$$r(u) = \frac{t}{2}$$

Známe tedy vodorovný posun styčnicku 2. Můžeme přejít k výpočtu energie, uložené v prutech.

$$Eng = \frac{1}{2} r^T K r$$

Prut 1:

$$Eng(1) = \frac{1}{2} [0 \ 0 \ 0,5t \ 0] \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0,5t \\ 0 \end{bmatrix} =$$



$$\frac{1}{2}[-0,5t \ 0 \ 0,5t \ 0] \begin{bmatrix} 0 \\ 0 \\ 0,5t \\ 0 \end{bmatrix} = \frac{1}{2}0,5t \ 0,5t = \frac{1}{8}t^2$$

$$Eng(1) = \frac{1}{8}t^2$$

Prut 2:

$$Eng(2) = \frac{1}{2} [0,5t \ 0 \ t \ 0] \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0,5t \\ 0 \\ t \\ 0 \end{bmatrix} =$$

$$\frac{1}{2}[-0,5t \ 0 \ 0,5t \ 0] \begin{bmatrix} 0,5t \\ 0 \\ t \\ 0 \end{bmatrix} = \frac{1}{2}(-0,25t^2 + 0,5t^2) = \frac{1}{8}t^2$$

$$Eng(2) = \frac{1}{8}t^2$$

Zajímá nás čas, kdy dojde k porušení prutu. Tzn., kdy hodnota energie dosáhne v jednom z prutů kritické hodnoty 10.

$$10 = \frac{1}{8}t^2$$

$$t = \sqrt{80} \cong 8,944s$$

Energie v obou prutech narůstá stejným tempem, teoreticky tedy dojde k porušení obou prutů ve stejném čase  $t=8,944s$ .

### 6.3 VÝPOČET – METODA ZALOŽENÁ NA ENERGETICKÉM KRITÉRIU

Nejprve zjistíme, jak by narůstala hodnota energie uložená v systému, za předpokladu, že by konstrukce zůstala neporušená. Z předchozího výpočtu již víme, že vodorovný posun styčnicku 2 bude s časem narůstat podle následujícího vztahu:

$$r(u) = \frac{t}{2}$$

Potom:

$$Eng = \frac{1}{2}r^T Kr$$

$$Eng = \frac{1}{2} [0 \ 0 \ 0,5t \ 0 \ t \ 0] \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 2 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0,5t \\ 0 \\ t \\ 0 \end{bmatrix} =$$

$$\frac{1}{2} \begin{bmatrix} -0,5t & 0 & t-t & 0 & 0,5t & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0,5t \\ 0 \\ t \\ 0 \end{bmatrix} = \frac{1}{2} 0,5t^2$$

$$Eng = \frac{1}{4} t^2$$

### 6.3.1 Předpoklad 1 - porušení prutu č. 1

Nyní předpokládejme, že nejdříve dojde k porušení prutu číslo 1. Globální matice tuhosti potom bude bez příspěvku od tohoto prutu vypadat takto:

$$K(I) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}$$

Přejdeme k výpočtu vodorovného posunu styčnicku 3, za předpokladu, že prut 1 je porušen.

$$K(u, u) r(u) + K(u, p) r(p) = f(u)$$

$$\begin{bmatrix} 1 \end{bmatrix} r(u) + \begin{bmatrix} 0 & 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1t \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix}$$

$$r(u) = t$$

Spočteme energii v systému:

$$Eng(I) = \frac{1}{2} r^T K r = \frac{1}{2} \begin{bmatrix} 0 & 0 & t & 0 & t & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ t \\ 0 \\ t \\ 0 \end{bmatrix} =$$

$$\frac{1}{2} \begin{bmatrix} 0 & 0 & t-t & 0 & -t+t & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ t \\ 0 \\ t \\ 0 \end{bmatrix} = 0$$

$$Eng(I) = 0$$

Hodnota energie v systému tedy poklesla o celou původní hodnotu:

$$\Delta Eng = Eng(I) - Eng = 0 - \frac{1}{4} t^2 = -\frac{1}{4} t^2$$

### 6.3.2 Předpoklad 2 – porušení prutu č. 2

Jestliže předpokládáme, že byl porušen prut číslo 2, potom bude mít globální matice tuhosti následující tvar:

$$K(II) = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}$$

Nyní budeme pokračovat ve výpočtu stejným způsobem. Nejdříve spočítáme posun styčnicku 3 ve vodorovném směru:

$$K(u, u) r(u) + K(u, p) r(p) = f(u)$$

$$[1] r(u) + [-1 \ 0 \ 0 \ 0 \ 0] \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1t \\ 0 \end{bmatrix} = [0]$$

$$r(u) = 0$$

Tento posun bude nulový. Přikročíme k výpočtu energie systému:

$$Eng(II) = \frac{1}{2} r^T K r = \frac{1}{2} [0 \ 0 \ 0 \ 0 \ t \ 0] \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ t \\ 0 \end{bmatrix} =$$

$$\frac{1}{2} [0 \ 0 \ 0 \ 0 \ 0 \ 0] \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ t \\ 0 \end{bmatrix} = 0$$

$$Eng(I) = 0$$

Hodnota energie v systému tedy poklesla, stejně jako v prvním případě, o celou původní hodnotu:

$$\Delta Eng = Eng(II) - Eng = 0 - \frac{1}{4} t^2 = -\frac{1}{4} t^2$$

Nyní spočteme hodnotu disipace:

$$D = \frac{1}{2} \sigma_{crit}^2 \frac{AL}{E} = \frac{1}{2} \sqrt{20}^2 = 10$$

Z předpokladu rovnosti disipace a změny energie můžeme spočítat čas, kdy k porušení dojde. Jelikož jsou oba pruty shodné, dojde k jejich porušení v jednom okamžiku.

$$D = -\Delta Eng$$

$$10 = \frac{1}{4}t^2 \Rightarrow t = \sqrt{40} \cong 6,326s$$

$$t = 6,326s$$

### 6.3.3 POROVNÁNÍ

Ručním výpočtem pomocí metody nejslabšího článku jsme dospěli k závěru, že k porušení konstrukce dojde v čase 8,944s. Zatímco výpočtem metodou jsme zjistili, že by se konstrukce měla porušit v čase 6,326s.

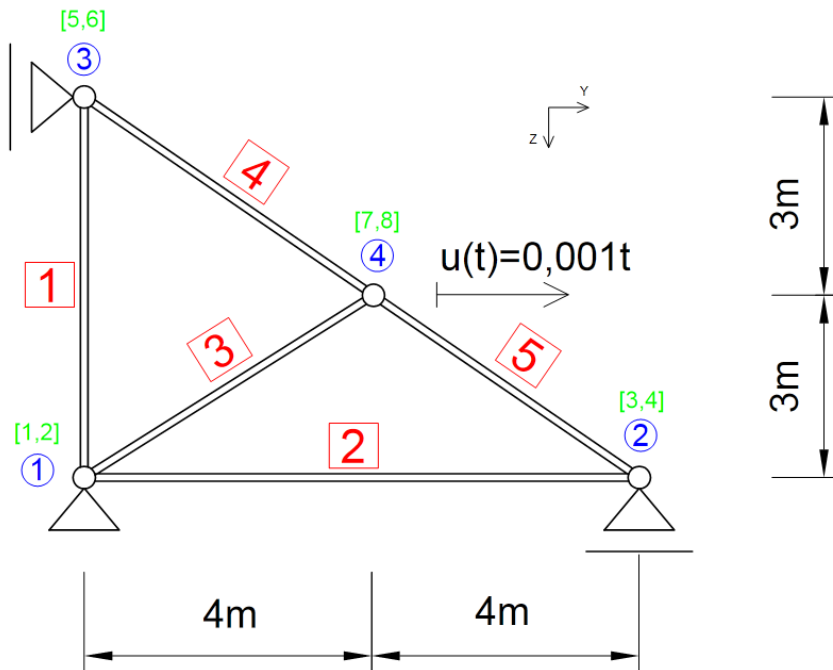
Pozn.: Pokud bychom tento příklad řešili pomocí programů, vstupní parametry by vypadaly následovně:

```
1 %% EXAMPLE 01
2
3 xz = [0,0;1,0;2,0]; % souradnice stycniku VSTUP
4 elems = [1,2;2,3]; % topologie soustavy VSTUP
5 p = [1,2,4,5,6]; % predepsane posuny VSTUP
6 sigma_crit = ones(size(elems,1),1)*4.472; % kriticka napeti VSTUP
7
8 A = ones(size(elems,1),1)*1; % Prurezy jednotlivych prutu VSTUP
9 E = ones(size(elems,1),1); % Moduly pruznosti jednotlivych prutu VSTUP
10 r = zeros(length(xz(:)),1); limit=10;
12 r(5) = 1; % predepsane zatizeni posunem VSTUP
```

Pomocí programu potom získáme výsledky totožné s ručním výpočtem.

## 7 PŘÍKLAD 2 – VÝPOČET POMOCÍ PROGRAMU

### 7.1 SCHÉMA, ZATÍŽENÍ



Obr. 7.1: Schéma konstrukce

Konstrukce se skládá ze čtyř prutů průřezu konstantního  $A=0,01\text{m}^2$ , kritické napětí  $\sigma_{\text{crit}}=10^4\text{Pa}$ , modul pružnosti  $E=10^7\text{Pa}$ . Styčnick 4 je zatížen vodorovným posunem narůstajícím lineárně v čase podle funkce  $0,001t$ . Podepření je zřejmé ze schématu.

Vstupy do obou programů budou tedy v následujících tvarech:

$$\text{Souřadnice styčnicků: } xz = \begin{bmatrix} 0 & 0 \\ 8 & 0 \\ 0 & -6 \\ 4 & -3 \end{bmatrix}$$

$$\text{Topologie soustavy: } elems = \begin{bmatrix} 1 & 3 \\ 2 & 1 \\ 1 & 4 \\ 3 & 4 \\ 4 & 2 \end{bmatrix}$$

Předepsané posuny:  $p = [1,2,4,5,7];$

Zatížení:  $r(7) = 0,001;$

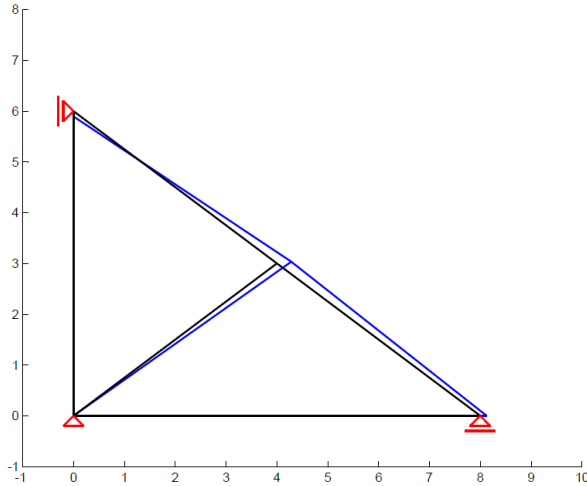
Kritické napětí:  $\sigma_{\text{crit}} = \text{ones}(\text{size}(elems,1),1)*1e4 ;$

Modul pružnosti:  $E = \text{ones}(\text{size}(elems,1),1)*1e7 ;$

## 7.2 VÝSLEDKY

### 7.2.1 Metoda nejslabšího článku

Výsledky výpočtů ukazují, že se poruší prut číslo 3 v tahu a to v čase 5,7458s. Deformace konstrukce v čase těsně před kolapsem bude mít následující tvar:

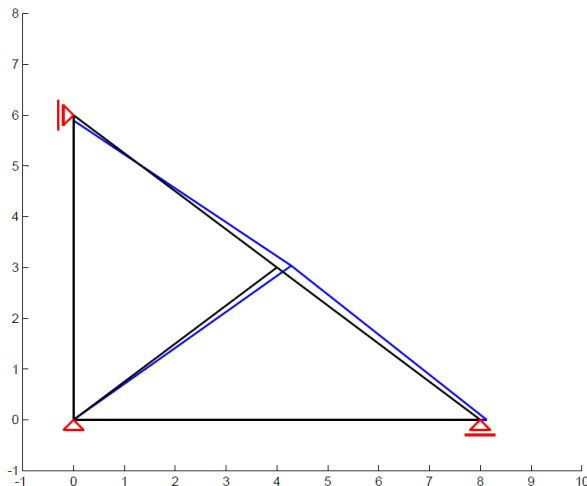


$$r = \begin{bmatrix} 0 \\ 0 \\ 0,0026 \\ 0 \\ 0 \\ 0,0021 \\ 0,0057 \\ -0,0007 \end{bmatrix} \text{ [m]}$$

Obr. 7.2: Deformace těsně před kolapsem

### 7.2.2 Metoda založená na energetickém kritériu

Podle metody založené na principu rovnosti disipace a změny energie systému při porušení se jako první poruší prut číslo 3 v tahu v čase 4,2374s. Deformace těsně před kolapsem bude vypadat takto:

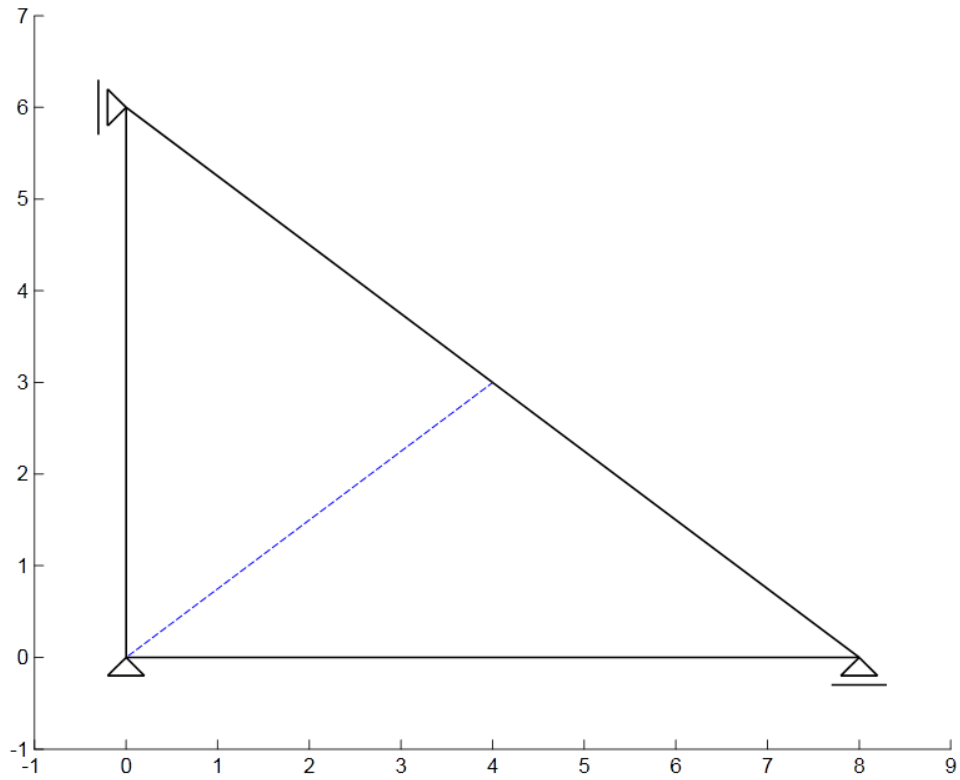


$$r = \begin{bmatrix} 0 \\ 0 \\ 0,0019 \\ 0 \\ 0 \\ 0,0015 \\ 0,0042 \\ -0,0005 \end{bmatrix} \text{ [m]}$$

Obr. 7.3: Deformace těsně před kolapsem

### 7.2.3 Porovnání

Podle obou metod dojde k tahovému porušení prutu číslo 3 a následně ke kolapsu celé konstrukce. Metody se liší v čase, kdy k porušení dojde a samozřejmě taky deformaci systému, při které se prut poruší.



Obr. 7.4: Porušení prutu číslo 3

$$r(u)_{fail}^{WL} = \begin{bmatrix} 0 \\ 0 \\ 0,0026 \\ 0 \\ 0 \\ 0,0021 \\ 0,0057 \\ -0,0007 \end{bmatrix} \text{ [m]}$$

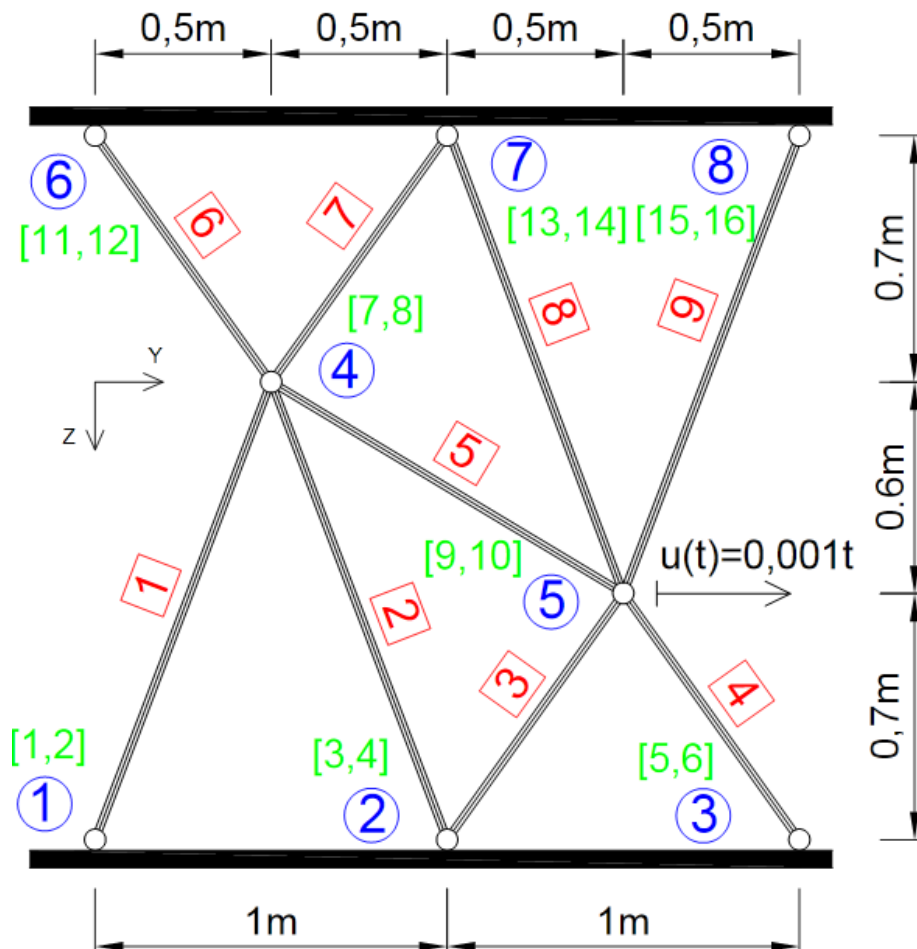
$$t_{fail}^{WL} = 5,7458 \text{ [s]}$$

$$r(u)_{fail}^{ENG} = \begin{bmatrix} 0 \\ 0 \\ 0,0019 \\ 0 \\ 0 \\ 0,0015 \\ 0,0042 \\ -0,0005 \end{bmatrix} \text{ [m]}$$

$$t_{fail}^{ENG} = 4,2374 \text{ [s]}$$

## 8 PŘÍKLAD 3 – VÝPOČET POMOCÍ PROGRAMU

### 8.1 SCHÉMA, ZATÍŽENÍ



Obr. 8.1: Schéma konstrukce<sup>1</sup>

Konstrukce se skládá z devíti prutů konstantního průřezu  $A=1\text{m}^2$ . Moduly pružnosti prutů 1-4 jsou  $10^7\text{Pa}$ , moduly pružnosti prutů 5-9 jsou  $10^6\text{Pa}$ . Kritické napětí prutů je  $1000\text{Pa}$ . Styčník 5 je zatížen vodorovným posunem narůstajícím v čase podle lineární funkce  $0,001t$ , Podepření je zřejmé ze schématu konstrukce.

Ostatní vstupy do obou programů budou vypadat takto:

$$\text{Souřadnice styčnicků: } xz = \begin{bmatrix} 0 & -0.7 \\ 1 & -0.7 \\ 2 & -0.7 \\ 0.5 & 0.6 \\ 1.5 & 0 \\ 0 & 1.3 \\ 1 & 1.3 \\ 2 & 1.3 \end{bmatrix}$$

<sup>1</sup> Geometrie konstrukce převzata z článku: Eliáš, J.; Frantík, P.: *Improved sequentially linear solution procedure*. Engineering Fracture Mechanics, Vol. 77, No. 12 (2010), pp. 2263-2276.



$$\text{Topologie soustavy: } elems = \begin{bmatrix} 1 & 4 \\ 4 & 2 \\ 2 & 5 \\ 5 & 3 \\ 4 & 5 \\ 4 & 6 \\ 4 & 7 \\ 7 & 5 \\ 5 & 8 \end{bmatrix}$$

Předepsané posuny:  $p = [1,2,3,4,5,6,9,11,12,13,14,15,16]$ ;

Zatížení:  $r(9) = 0,001$ ;

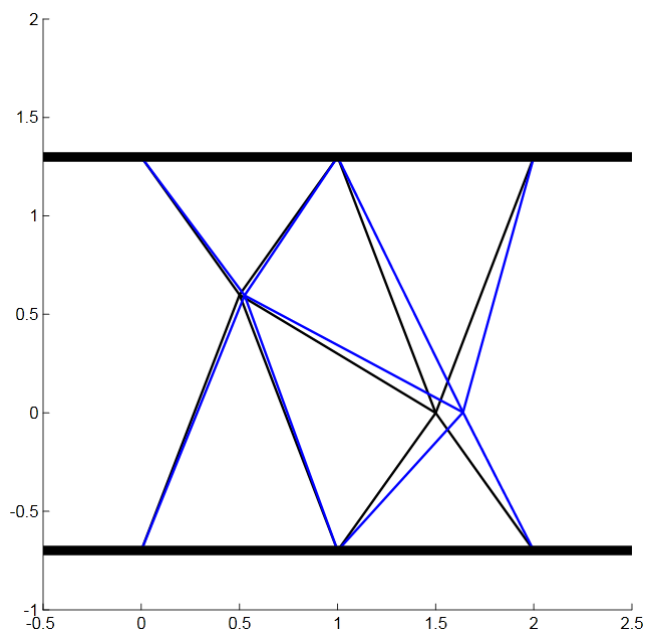
## 8.2 VÝSLEDKY

### 8.2.1 Metoda nejslabšího článku

Výsledkem výpočtu je matice *Fail\_Matrix*, v jejímž prvním sloupci nalezneme čísla prutů podle pořadí, ve kterém dochází jejich porušení a ve druhém sloupci časy porušení od začátku zatěžování. Matice má následující tvar:

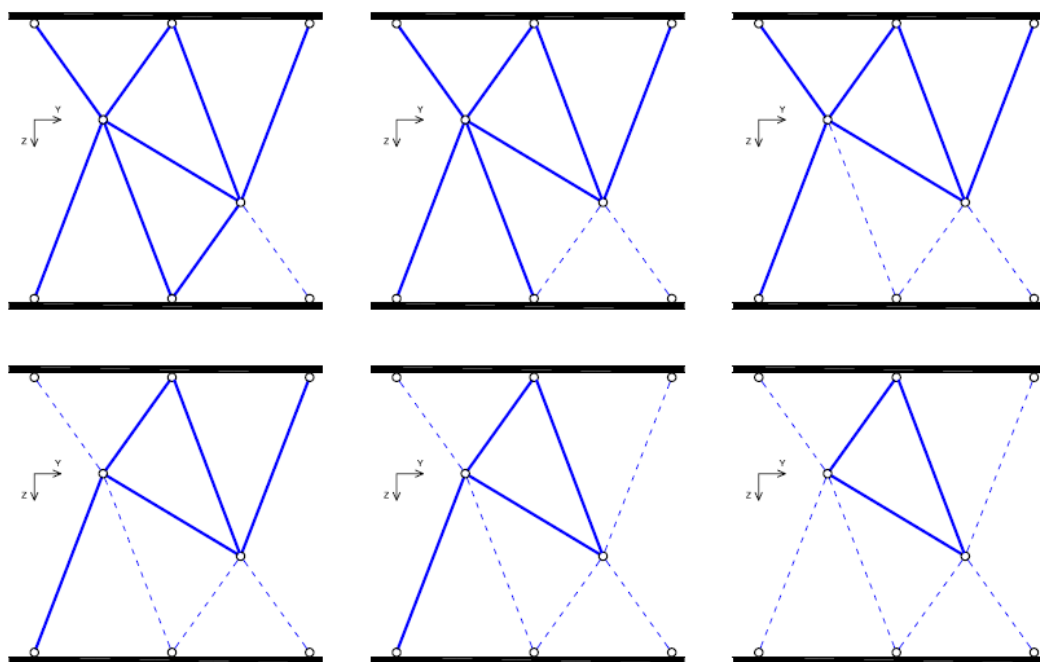
$$Fail\_Matrix^{WL} = \begin{bmatrix} 3 & 0,1444 \\ 4 & 1,3380 \\ 2 & 1,8029 \\ 6 & 2,6810 \\ 9 & 3,6803 \\ 1 & 6,5159 \end{bmatrix}$$

K prvnímu porušení dojde tedy v čase 0,14s. Deformace těsně před prvním porušením bude vypadat následovně:



$$r(u)^{WL} = 10^{-3} \begin{bmatrix} 0,0264 \\ 0,0030 \\ 0,1400 \\ -0,0025 \end{bmatrix} \begin{matrix} 7 \\ 8 \\ 9 \\ 10 \end{matrix}$$

Obr. 8.2: Deformace těsně před porušením prvního prutu



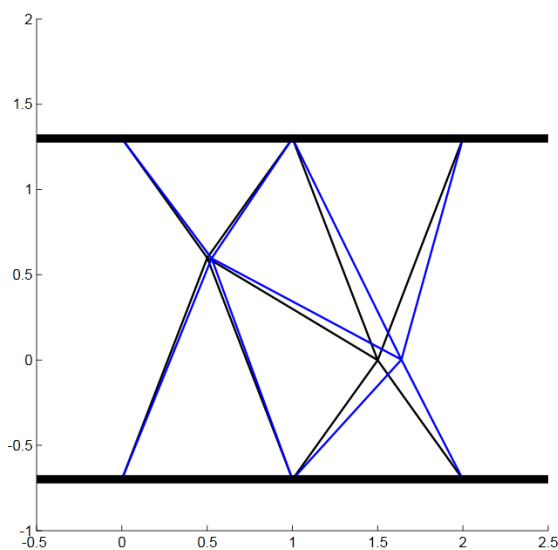
Obr. 8.3: Vývoj porušení konstrukce

### 8.2.2 Metoda založená na energetickém kritériu

Stejně jako v prvním případě je výstupem programu matice *Fail\_Matrix* s čísly porušených prutů a časy jejich porušení.

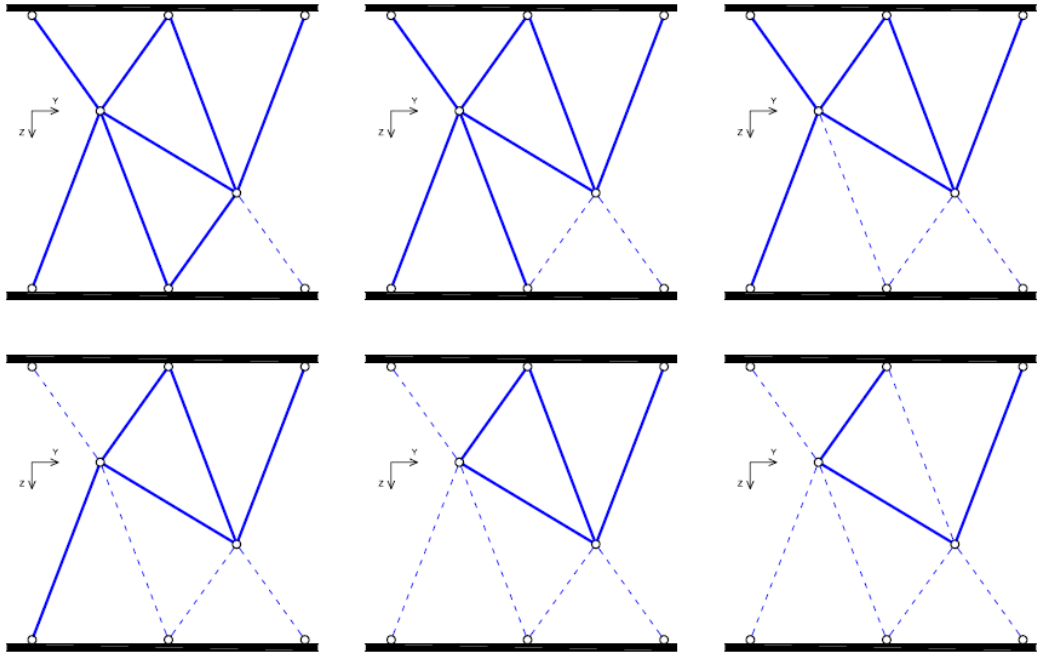
$$Fail\_Matrix^{ENG} = \begin{bmatrix} 3 & 0,1063 \\ 4 & 0,5299 \\ 2 & 0,9898 \\ 6 & 1,8434 \\ 1 & 1,7892 \\ 8 & 2,7436 \end{bmatrix}$$

K prvnímu porušení dojde tedy v čase 0,11s. Deformace těsně před prvním porušením bude vypadat takto:



$$r(u)^{ENG} = 10^{-4} \begin{bmatrix} 0,1885 \\ 0,0212 \\ 1,0000 \\ -0,0179 \end{bmatrix} \begin{matrix} 7 \\ 8 \\ 9 \\ 10 \end{matrix}$$

Obr. 8.4: Deformace těsně před porušením prvního prutu



Obr. 8.5: Vývoj porušení konstrukce

### 8.2.3 Porovnání

V tomto případě se obě metody liší nejen v časech kolapsů jednotlivých prutů, ale i v tom, které pruty zkolabují. Liší se i deformace konstrukce, při které dojde k prvnímu poškození.

$$Fail\_Matrix^{WL} = \begin{bmatrix} 3 & 0,1444 \\ 4 & 1,3380 \\ 2 & 1,8029 \\ 6 & 2,6810 \\ 9 & 3,6803 \\ 1 & 6,5159 \end{bmatrix}$$

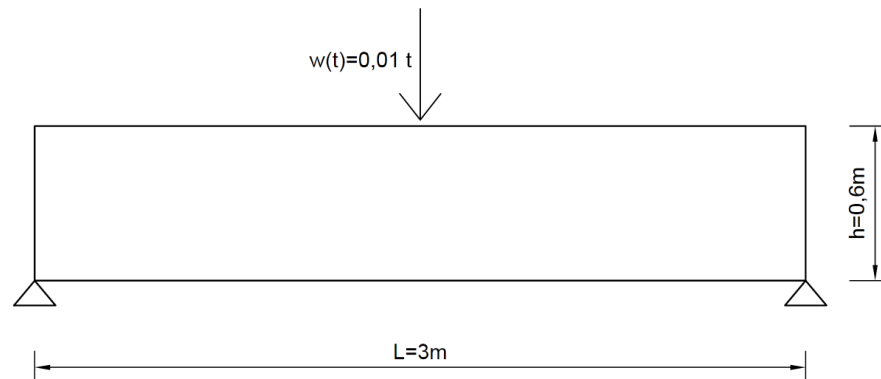
$$Fail\_Matrix^{ENG} = \begin{bmatrix} 3 & 0,1063 \\ 4 & 0,5299 \\ 2 & 0,9898 \\ 6 & 1,8434 \\ 1 & 1,7892 \\ 8 & 2,7436 \end{bmatrix}$$

$$r(u)_{fail}^{WL} = 10^{-3} \begin{bmatrix} 0,0264 \\ 0,0030 \\ 0,1400 \\ -0,0025 \end{bmatrix} \begin{matrix} 7 \\ 8 \\ 9 \\ 10 \end{matrix} [m]$$

$$r(u)_{fail}^{ENG} = 10^{-3} \begin{bmatrix} 0,0189 \\ 0,0021 \\ 0,1000 \\ -0,0018 \end{bmatrix} \begin{matrix} 7 \\ 8 \\ 9 \\ 10 \end{matrix} [m]$$

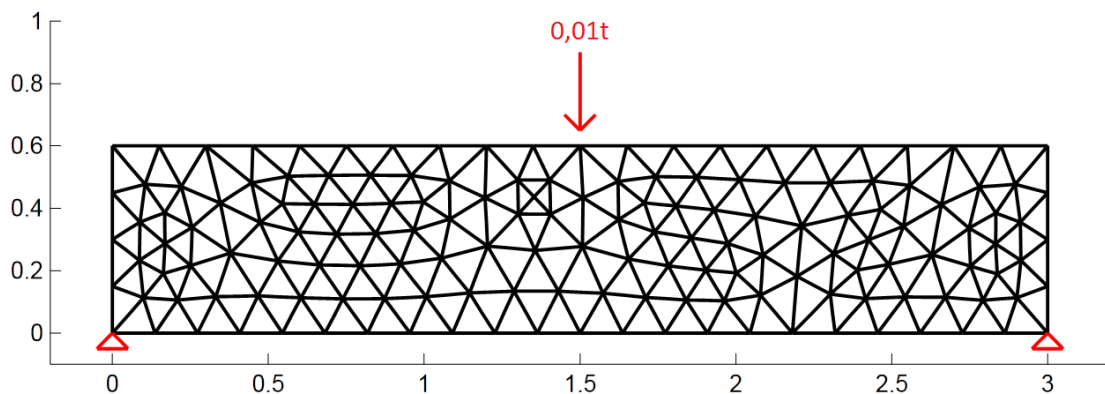
## 9 PŘÍKLAD 4 – VÝPOČET POMOCÍ PROGRAMU

### 9.1 SCHÉMA, ZATÍŽENÍ



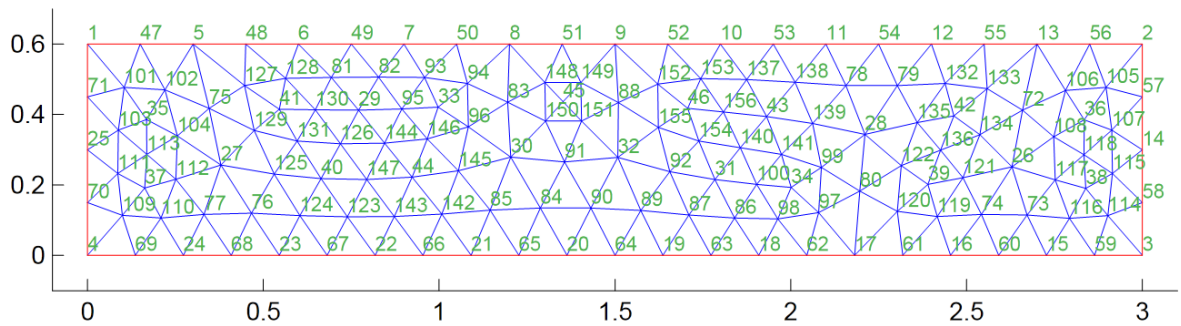
Obr. 9.1: Schéma ohybové zkoušky

V tomto příkladě budeme simulovat zkoušku konstrukce v třibodovém ohybu. Pro jednoduchost budeme vzorek uvažovat jako rovinný model a k reprezentaci jeho struktury použijeme příhradové pruty.



Obr. 9.2: Použitý příhradový model

Použitý příhradový model obsahuje celkem 156 styčníků a 415 prutů. Zatížení prostředního styčnicku u horního povrchu vzorku (styčnick č. 9) posunem bude v čase lineárně narůstat podle funkce  $0,01t$ . Podepřeny budou styčnick č. 2 a styčnick č. 3. Abychom předešli lokálnímu poškození, budeme počítat se zesílením prutů v okolí podpor a v okolí zatížení. Jedná se o všechny pruty vycházející ze styčnicků 4, 3, 9. Zesíleným prutům přiřadíme modul pružnosti  $10^6$  a kritické napětí  $10^5$ . Ostatním prutům potom modul pružnosti  $2,1 \cdot 10^5$  a kritické napětí  $5 \cdot 10^3$ . Průřezová plocha bude u všech prutů stejná a to 0,1. Dalšími vstupními parametry bude opět matice  $xz$  se souřadnicemi styčnicků o rozměru  $156 \times 2$ , matice  $elems$  s topologií soustavy o rozměru  $415 \times 2$ . Zatížení posunem  $r(18)=0,01$ . Předepsané posuny tedy budou  $p=[5,6,7,8,18]$ . Výpočet provedeme pro model se stejnou pevností v tahu i tlaku a pro vzorek s několikanásobně vyšší pevností v tlaku.



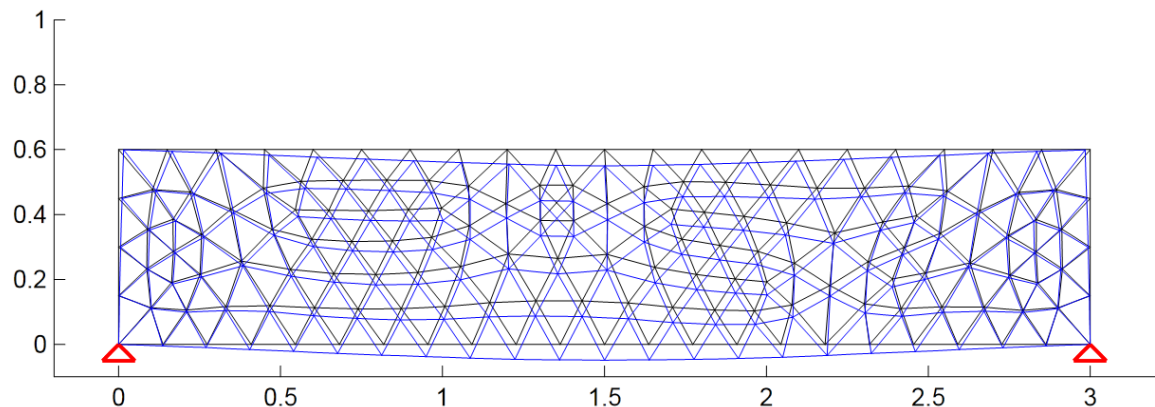
Obr. 9.3: Čísla styčníků v příhradovém modelu

## 9.2 VÝSLEDKY

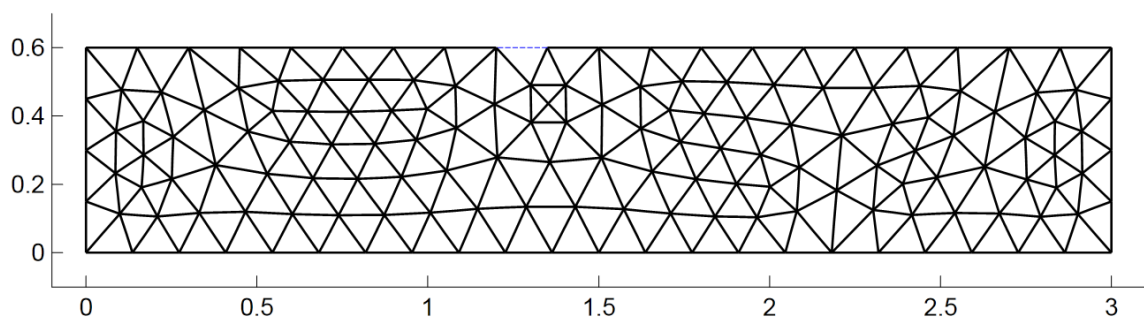
### 9.2.1 Metoda nejslabšího článku

#### *Stejná pevnost v tahu a tlaku*

Příhradový model se začne deformovat. K porušení prvního prutu dojde v čase 4,1922s. Porušen bude prut číslo 196, který se nachází při horním povrchu modelu a spojuje styčnky 8 a 51.



Obr. 9.4: Zdeformovaná konstrukce těsně před porušením

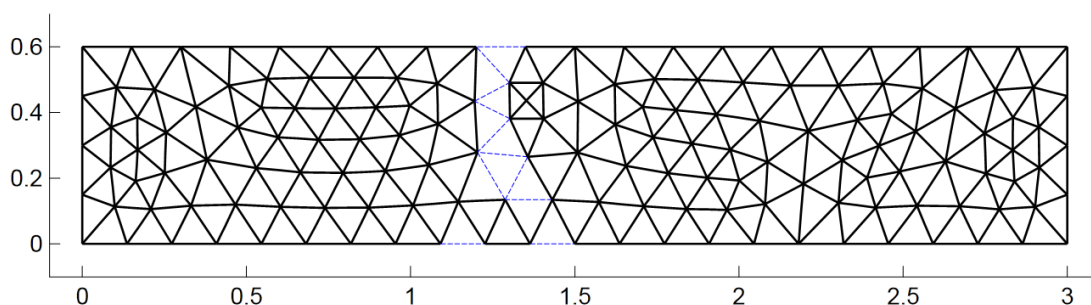


Obr. 9.5: Porušení prvního prutu v čase 4,2s

Stejně jako v předchozích případech je výsledkem výpočtu matice *Fail\_Matrix*, ze které zjistíme, jak se konstrukce bude porušovat dále.

$$Fail\_matrix_I^{WL} = \begin{bmatrix} 196 & 4,1922 \\ 175 & 4,2043 \\ 178 & 4,1862 \\ 104 & 5,7040 \\ 242 & 6,1563 \\ 103 & 9,7756 \\ 105 & 11,6563 \\ 301 & 10,3968 \\ 344 & 13,7741 \\ 255 & 20,7283 \end{bmatrix}$$

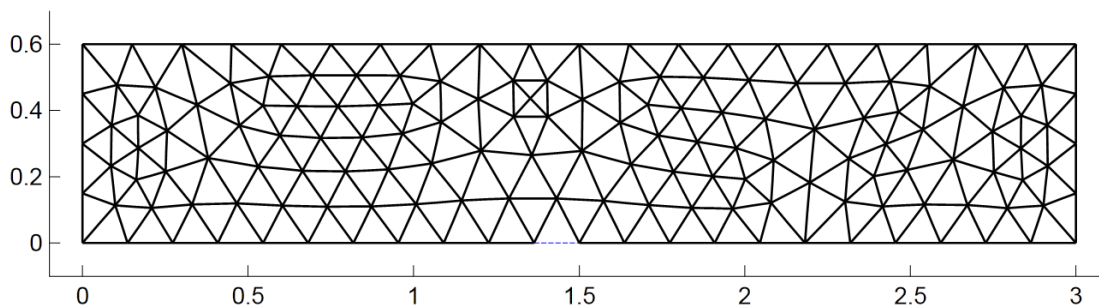
Konstrukce se tedy poruší podle výpočtu metodou nejslabšího článku následujícím způsobem:



Obr 9.6: Porušení konstrukce

### ***Několicí násobně vyšší pevnost v tlaku***

V tomto případě by k prvnímu porušení konstrukce došlo u spodního okraje vzorku. Jako první by byl porušen prut číslo 242 v čase 8.7335s, spojující styčníky 20 a 64.

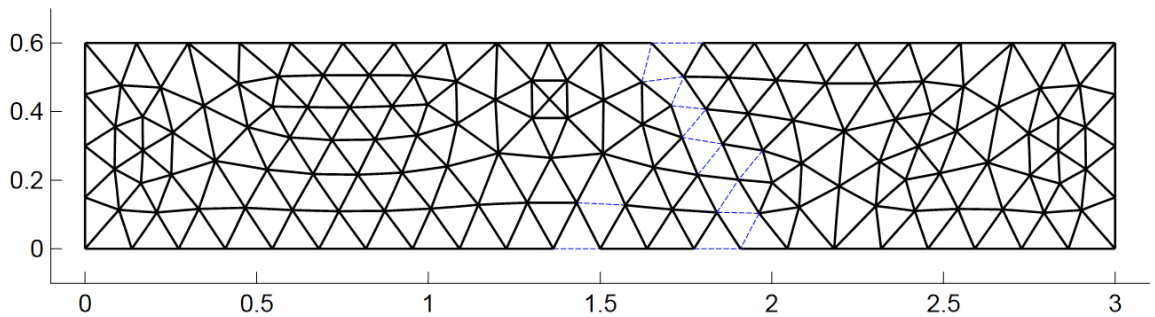


Obr 9.7: První porušení konstrukce s vysokou tlakovou pevností v čase 8,7s

Další vývoj poškození bychom opět mohli vyčíst z výstupní matice *Fail\_matrix*.

$$Fail\_matrix_{II}^{WL} = \begin{bmatrix} 242 & 8,7335 \\ 83 & 9,4835 \\ 197 & 14,3577 \\ 318 & 12,4020 \\ 415 & 12,9695 \\ \vdots & \vdots \\ \vdots & \vdots \end{bmatrix}$$

Celá konstrukce se v tomto případě tedy poruší odlišným způsobem.

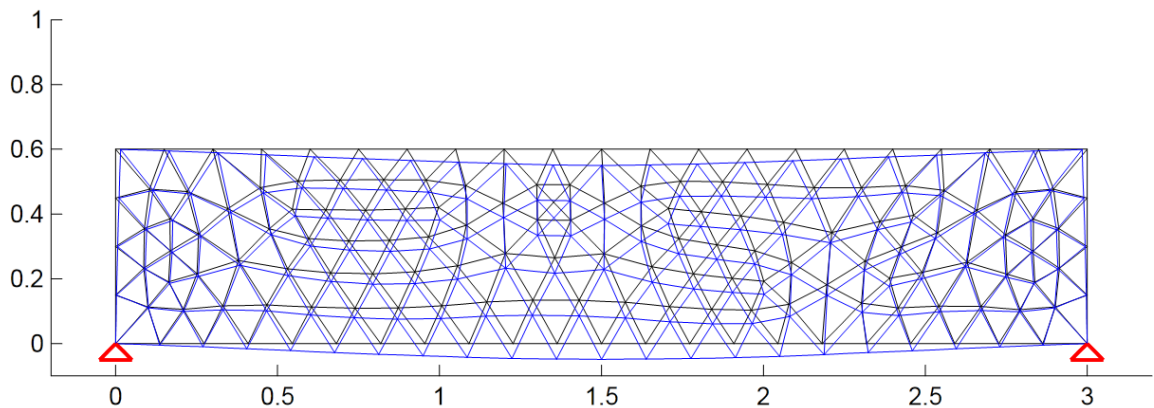


Obr. 9.8: Porušení konstrukce s vysokou pevností v tlaku

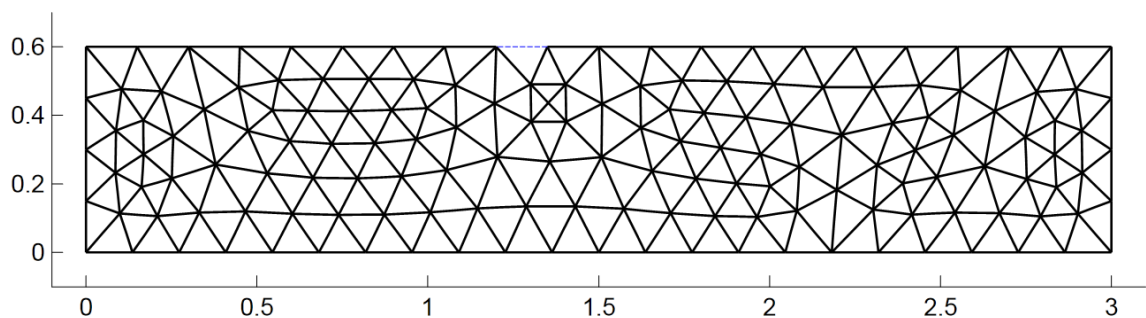
## 9.2.2 Metoda založená na energetickém kritériu

### *Stejná pevnost v tahu a tlaku*

Příhradový model se zdeformuje a k porušení prvního prutu dojde v čase 2,2901s. Porušení bude stejné jako v prvním případě prut číslo 196 při horním povrchu modelu. Deformace těsně před tímto porušením bude v závislosti na čase o něco menší než v případě výpočtu metodou nejslabšího článku.



Obr. 9.9: Zdeformovaná konstrukce těsně před porušením

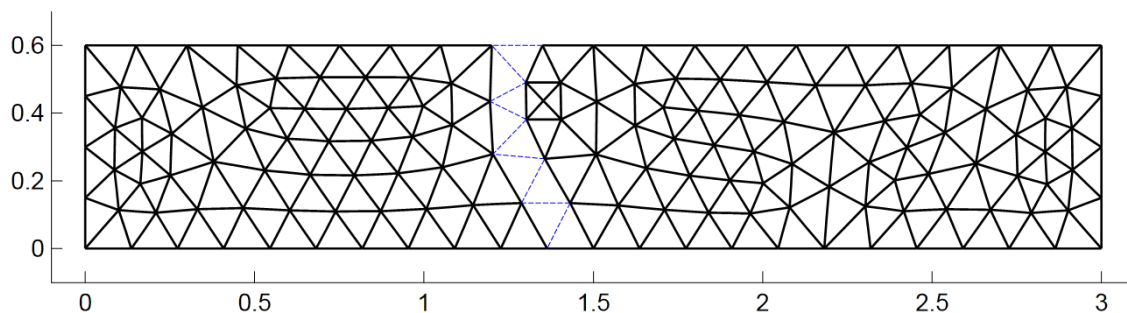


Obr. 9.10: Porušení prvního prutu v čase 2,3s

Výstupní matice výpočetního programu, ze které vyčteme další vývoj porušení modelu, má následující tvar:

$$Fail\_matrix_I^{ENG} = \begin{bmatrix} 196 & 2,2901 \\ 175 & 2,0824 \\ 178 & 2,1762 \\ 104 & 2,7885 \\ 244 & 2,5104 \\ 105 & 4,5417 \\ 176 & 4,7307 \\ 301 & 5,1769 \\ 255 & 5,3324 \end{bmatrix}$$

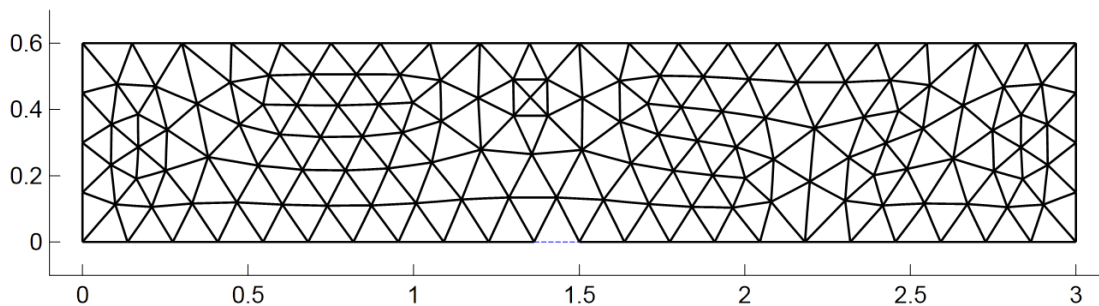
Konstrukce se tedy poruší následujícím způsobem:



Obr. 9.11: Porušení konstrukce

### ***Několicí násobně vyšší pevnost v tlaku***

Pro porovnání výsledků zjistíme touto metodou i průběh deformace vzorku s vysokou pevností v tlaku. Opět se jako první poruší prut 242, tentokrát v čase 3,90s



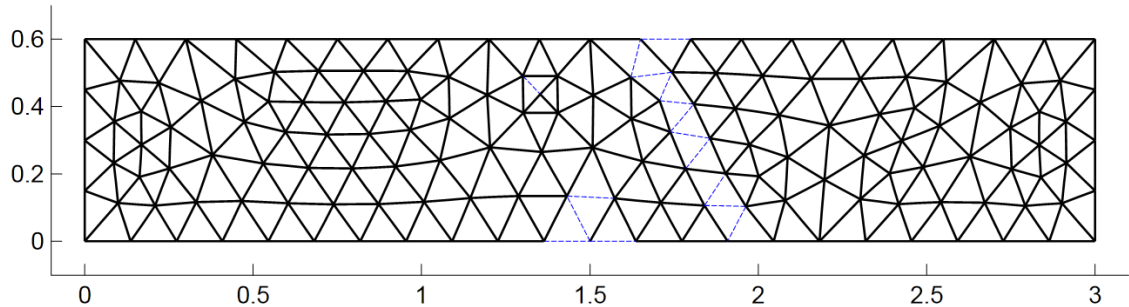
Obr. 9.12: První porušení konstrukce s vysokou tlakovou pevností v čase 3,90s

Další vývoj poškození bychom opět mohli vyčíst z výstupní matice *Fail\_matrix*.

$$Fail\_matrix_{II}^{ENG} = \begin{bmatrix} 242 & 3,8954 \\ 83 & 4,6678 \\ 197 & 5,3486 \\ 318 & 5,2410 \\ 363 & 5,5978 \\ \vdots & \vdots \\ \vdots & \vdots \end{bmatrix}$$



Porušení konstrukce s vysokou pevností v tlaku bude podle výpočtu metodou založenou na energetickém kritériu vypadat takto:



Obr. 9.13: Porušení konstrukce s vysokou pevností v tlaku

### 9.2.3 Porovnání

#### ***Stejná pevnost v tahu a tlaku***

V obou případech se jako první poruší prut číslo 196. Metody se liší v čase, kdy by ke kolapsu mělo dojít, potažmo v předepsané deformaci, kterou je potřeba vyvolat, abychom prut porušili.

$$t_{fail}^{WL} = 4,1922 [s] \qquad t_{fail}^{ENG} = 2,2901 [s]$$

$$r(u)_{fail}^{WL} = 0,005 [m] \qquad r(u)_{fail}^{ENG} = 0,003 [m]$$

#### ***Několikanásobně vyšší pevnost v tlaku***

V případě vzorku s vysokou pevností v tlaku se v obou případech poruší jako první prut číslo 240. Metody se opět liší v čase, kdy k porušení dojde.

$$t_{fail}^{WL} = 8,7335 [s] \qquad t_{fail}^{ENG} = 3,8954 [s]$$

$$r(u)_{fail}^{WL} = 0,009 [m] \qquad r(u)_{fail}^{ENG} = 0,004 [m]$$

## 10 ZÁVĚR

Cílem této práce bylo porovnání vývoje poškození příhradových modelů získaného metodou nejslabšího článku a metodou, která je založena na energetickém kritériu. První metoda předpokládá porušení prutu v okamžiku, kdy energie uložená v prutu dosáhne kritické hodnoty. Druhá metoda vychází z předpokladu rovnosti disipace a rozdílu energie porušené a neporušené konstrukce. Obě metody byly porovnány na čtyřech příkladech pomocí ručního výpočtu a pomocí programu.

Ukazuje se, že metody se mohou lišit nejenom v určení času porušení, ale také v lokalizaci místa, kde k porušení dojde. Zdá se, že energetická metoda předpovídá menší časy porušení, tzn. k porušení je potřeba vyvození menšího zatížení, tato metoda je tedy konzervativnější. Příčiny těchto rozdílů budou předmětem další analýzy v průběhu mého dalšího studia, stejně jako zkoumání dalších přístupů k této problematice.

## 11 LITERATURA

- [1] Jirásek, M.; Konvalinka, P.: *Statika stavebních konstrukcí I*. Praha: ČVUT, 1993, 187 s. ISBN: 80-01-00174-1.
- [2] Konvalinka, P.: *Přednáškové materiály k předmětu 132ANKC – Analýza konstrukcí*. ČVUT v Praze, FSV, katedra mechaniky. Praha 2010.
- [3] Bittnarová, J.: *Přednáškové materiály k předmětu 132SM3 – Stavební mechanika 3*. ČVUT v Praze, FSV, katedra mechaniky. Praha 2010. Dostupný z WWW: <[http://mech.fsv.cvut.cz/download/sm3\\_150dpi.pdf](http://mech.fsv.cvut.cz/download/sm3_150dpi.pdf)>
- [4] Bican, L.: *Lineární algebra a geometrie*. Praha: Academia, 2002, 304 s. ISBN: 978-80-200-1707-9
- [5] C. A. Buizer. *Computational approaches to damage localization using failure criteria based on local and global energy release*. Eindhoven, 2008, 21 s. Bachelor final project. TU/e – Mechanical Engineering Section, Mechanics of Materials. 2008
- [6] Zaplatílek, K.; Doňar, B.: *MATLAB<sup>®</sup> pro začátečníky*. Druhé vydání. Praha: BEN Technická literatura, 2005, 151 s. ISBN 80-7300-175-6.
- [7] Tichá, L.; Cívínová, Z.: *Jak psát vysokoškolské závěrečné práce*. Praha: Ústřední knihovna ČVUT, 2010, 35 s.
- [8] Bernstein, A.: *Rank One Update and the Google Matrix* [online]. Signal Science, LLC. Dostupný z WWW: <<http://www.signalscience.net/files/RankOneAndGooglesPageRank2.pdf>>
- [9] Bindel, D.: Fall 2009. *Sherman-Morrison-Woodbury* [online]. Cornell University, Department of Computer Science. Dostupný z WWW: <<http://www.cs.cornell.edu/~bindel/class/cs6210-f09/lec12.pdf>>
- [10] Brenner, S.; Li-Yeng Sung: *The Sherman-Morrison Formula* [online]. University of South Carolina. Dostupný z WWW: <<http://www.math.sc.edu/~meade/math526-F05/labs/lab07-PartI.pdf>>
- [11] Wikipedia.org. *Woodbury matrix identity* [online]. Dostupný z WWW: <[http://en.wikipedia.org/wiki/Woodbury\\_matrix\\_identity](http://en.wikipedia.org/wiki/Woodbury_matrix_identity)>
- [12] Wikipedia.org. *Sherman-Morrison formula* [online]. Dostupný z WWW: <[http://en.wikipedia.org/wiki/Sherman-Morrison\\_formula](http://en.wikipedia.org/wiki/Sherman-Morrison_formula)>
- [13] Wikipieia.org. *LU Decomposition* [online]. Dostupný z WWW: <[http://en.wikipedia.org/wiki/LU\\_decomposition](http://en.wikipedia.org/wiki/LU_decomposition)>

- [14] Leguillon, D.: *Strength or toughness? A criterion for crack onset at a notch*. European Journal of Mechanics - A/Solids, Vol. 21, No. 1. (2002), pp. 61-72.
- [15] Eliáš, J.; Frantík, P.: *Improved sequentially linear solution procedure*. Engineering Fracture Mechanics, Vol. 77, No. 12 (2010), pp. 2263-2276.

## **PODĚKOVÁNÍ**

Na tomto místě bych rád poděkoval vedoucímu bakalářské práce doc. Ing. Janu Zemanovi, Ph. D. za ochotu a trpělivost při konzultacích, za odbornou pomoc, poskytnutí materiálů a hlavně za skvělý přístup. Dále bych rád poděkoval Ing. Ondřeji Rokošovi za poskytnutí programu na vykreslování prutových konstrukcí a za vysvětlení jeho fungování. Mé poděkování patří samozřejmě i všem ostatním, kteří mě po celou dobu studia podporovali.