

# Modely tvárného porušení pro simulaci hlubokého tažení MKP

Bc. Vilém Klepač

2018



# Prohlášení

Prohlašuji, že jsem diplomovou práci na téma *Modely tvárného porušení pro simulaci hlubokého tažení MKP* tvořil samostatně pod vedením Doc. Ing. Miroslava Španiela, CSc. a s využitím svých znalostí a literárních zdrojů, jež jsou v této práci uvedeny.

v Praze dne 17.8.2018

Vilém Klepač



## Poděkování

Děkuji vedoucímu práce panu Doc., Ing. Miroslavu Španielovi, CSc. za předání cenných zkušeností a pomoc, která dostala tuto práci do své finální podoby. Za velmi přínosné a komplexní zadání děkuji firmě Comtes FHT, jmenovitě panu Ing. Antonínu Prantlovi, CSc. Dále děkuji kolegům z Ústavu termomechaniky Akademie věd za technickou podporu. V neposlední řadě děkuji své rodině za celoživotní podporu nejen ve studiu. Speciální poděkování patří Štěpánce Šedivákové za udržování našeho útulného a bezpečného zázemí v těžkých časech závěru studia.



## **Abstrakt**

Tato diplomová práce se zabývá laděním modelu porušení materiálu pro simulace hlubokého tažení pomocí MKP. Operaci hlubokého tažení představuje experiment punch testu podle Nakazimy. Referenční sledovanou veličinou je hodnota síly na kulovém razníku během posuvu. Na základě porovnání závislostí získaných simulací a experimentálním měřením jsou naladěny parametry modelu porušení. Obdržené výsledky jsou rovněž porovnány s analytickým řešením.

## **Abstract**

Target of this thesis is to find parameters of ductile damage criterion for FEM deep drawing simulation. Deep drawing operation is represented by Nakazima punch test. The reference value is force on punch during its displacement. Based on comparing of curves obtained from simulation and experiment, parameters of ductile damage model are found. Obtained results are compared with analytical solution.

## **Klíčová slova**

Nakazima punch test, MKP, tvárné porušení, analytické řešení, optimalizace

## **Keywords**

Nakazima punch test, FEM, ductile damage, analytical solution, optimization

# Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>Analytické řešení</b>	<b>6</b>
2.1	Rekonstrukce experimentu . . . . .	6
2.1.1	Vyšetření odezvy síla - posuv . . . . .	7
2.1.2	Iniciace trhliny . . . . .	9
2.2	Závěr a komentář výsledků získaných z analytického řešení . . . . .	10
<b>3</b>	<b>Simulace metodou konečných prvků</b>	<b>11</b>
3.1	Úvod do prostředí ABAQUS . . . . .	11
3.2	Popis fyzikální předlohy úlohy . . . . .	12
3.3	2D úloha . . . . .	12
3.4	Rotačně symetrická úloha . . . . .	13
3.5	Cyklicky symetrická úloha . . . . .	15
3.5.1	Cyklicky symetrický model s dokonale tuhým razníkem . . . . .	15
3.5.2	Cyklicky symetrický model s poddajným razníkem . . . . .	15
3.5.3	Porovnání úloh Rigid a Deformable . . . . .	18
3.6	Plná 3D úloha . . . . .	19
3.7	Porovnání a komentář dosažených výsledků analýzy MKP . . . . .	22
<b>4</b>	<b>Teoretické podklady pro identifikaci modelu porušení</b>	<b>25</b>
4.1	Základy teorie rovinných úloh . . . . .	25
4.1.1	Hookeův zákon . . . . .	25
4.1.2	Úloha rovinné napjatosti . . . . .	26
4.1.3	Úloha rovinné deformace . . . . .	27
4.2	Základy matematiky pro mechaniku . . . . .	28
4.2.1	Tenzory . . . . .	28
4.2.2	Deviatorická složka napětí . . . . .	29
4.2.3	Invarianty . . . . .	29
4.3	Základy optimalizace, přehled některých optimalizačních úloh . . . . .	30
4.3.1	Rosenbrockova metoda . . . . .	30
4.4	Modely tvárného porušení . . . . .	30
4.4.1	Triaxialita a Lodeho úhel . . . . .	31
4.5	Filozofie mechaniky poškození . . . . .	33
4.6	Kritérium porušení Rice-Tracey . . . . .	33
4.7	Model porušení Bai-Wierzbicki . . . . .	34
4.7.1	Redukce počtu parametrů modelu Bai-Wierzbicki . . . . .	35



<b>5</b>	<b>Kalibrace modelu Bai-Wierzbicki pro materiál DC01 pomocí Nakazima experimentu a tahové zkoušky</b>	<b>36</b>
5.1	Třídění dat získaných z výpočtového softwaru . . . . .	37
5.2	Sestavení cílové funkce . . . . .	41
5.3	Minimalizace cílové funkce . . . . .	43
5.4	Shrnutí procesu kalibrace materiálového modelu . . . . .	45
<b>6</b>	<b>Závěr a komentář výsledků</b>	<b>47</b>
6.1	Porovnání dosažených výsledků analytického výpočtu, experimentu a simulace MKP . . . . .	47
6.2	Závěrečné shrnutí . . . . .	53
<b>A</b>	<b>Skript k načtení výsledků z programu ABAQU - BW.py</b>	<b>54</b>
<b>B</b>	<b>Zpracování načtených dat a vytvoření cílové funkce</b>	<b>57</b>
<b>C</b>	<b>Funkce hledání minima</b>	<b>60</b>

# Použité symboly a jejich význam

Název	Jednotka	Označení
$N_r$	$[N]$	Vnitřní membránová síla v radiálním směru
$N_{\Theta}$	$[N]$	Vnitřní membránová síla v tečném směru
$\sigma$	$[N \text{ mm}^{-2}]$	Obecné napětí v materiálu
$\sigma_r$	$[N \text{ mm}^{-2}]$	Membránové napětí v radiálním směru
$\sigma_{\Theta}$	$[N \text{ mm}^{-2}]$	Membránové napětí v tečném směru
$\epsilon$	$[-]$	Obecná deformace materiálu
$\epsilon_{\Theta}$	$[-]$	Tečná deformace
$\epsilon_r$	$[-]$	Radiální deformace
$h$	$[mm]$	Tloušťka materiálu
$R_b$	$[mm]$	Poloměr razníku/Poloměr křivosti v radiálním směru
$R_{\Theta}$	$[mm]$	Poloměr křivosti v tečném směru
$p$	$[N \text{ mm}^{-2}]$	Kontaktní tlak
$P$	$[N]$	Tvářecí síla
$r_c$	$[mm]$	Poloměr průmětu do horizontální roviny
$\psi_c$	$[rad]$	Úhel mezi osou razníku a radius vektorem bodu hranice kontaktu
$\psi_{cf}$	$[rad]$	Úhel mezi osou razníku a radius vektorem v okamžiku vzniku trhliny
$C_0$	$[N \text{ mm}^{-2}]$	Silový modul
$n$	$[-]$	Exponent zpevňování materiálu
$\eta$	$[-]$	Triaxialita napětí
$\sigma_m$	$[N \text{ mm}^{-2}]$	Napětí HMH
$\bar{\sigma}$	$[N \text{ mm}^{-2}]$	Ekvivalentní napětí
$D$	$[-]$	Parametr poškození
$D_c$	$[-]$	Parametr poškození v okamžiku vzniku trhliny
$\sigma_{ij}$	$[N \text{ mm}^{-2}]$	Tenzor napětí
$C_{ijkl}$	$[N \text{ mm}^{-2}]$	Tečný modul
$\epsilon_{kl}$	$[mm]$	Tenzor deformace
$\delta_{ij}$	$[-]$	Kroneckerovo delta
$\lambda, \mu$	$[-]$	Lamého konstanty
$E$	$[N \text{ mm}^{-2}]$	Youngův modul pružnosti
$\nu$	$[-]$	Poissonovo číslo
$\tau_{ij}$	$[N \text{ mm}^{-2}]$	Smykové napětí
$A$	$[-]$	Transformační matice

<b>Název</b>	<b>Jednotka</b>	<b>Označení</b>
$S_{ij}$	$[N \text{ mm}^{-2}]$	Deviatorická složka napětí
$I_{1-3}$	$[-]$	Hlavní invarianty napětí
$J_{1-3}$	$[-]$	Invarianty deviátoru napětí
$\Theta$	$[rad]$	Lodeho úhel
$\bar{\Theta}$	$[-]$	Normalizovaný Lodeho parametr
$C_n$	$[-]$	Parametr modelu porušení
$F_{1-3}$	$[-]$	Funkce triaxiality napětí
$F$	$[mm]$	Cílový funkcionál
$N$	$[-]$	Počet porovnávaných vzorků
$y_i$	$[mm]$	Velikost dosaženého posuvu během experimentu
$y_i^*$	$[mm]$	Spočítaná maximální velikost posuvu

# Kapitola 1

## Úvod

Moderní strojírenství v současné době jde nezadržitelným tempem vpřed. Jedním z hybatelů technického pokroku v obecném slova smyslu je zejména vývoj výpočetní techniky. To umožňuje výpočtářům výrazně zkrátit dobu pevnostních výpočtů nebo zvětšit počítanou síť. Softwarovým vývojářům zase umožňuje rozšířit možnosti výpočetních softwarů. Ty jsou dodávány ve stále se zdokonalujících uživatelských prostředích umožňujících řešení dalekosáhlejších a komplikovanějších problematik, než tomu bylo např. před třiceti lety. Ruku v ruce s vývojem jde ale i fakt, že se z výpočetních softwarů stávají pro uživatele "černé skříně", do kterých na jedné straně uživatel vloží vstupní data, na druhé si vyzvedne výsledek. Je proto důležité zkoušet výpočty ověřovat a nevěřit bezmezně spočítaným výsledkům.

Současně jsou neustále zvyšovány nároky na výrobky. Parametrem kvality výrobku je nejen jeho základní funkčnost, ale i náklady na jeho výrobu, optimální využití jeho únosnosti, atd. Je zřejmé, že pro optimální návrh např. výrobního procesu či provozu součásti, je třeba znát co nejlépe vlastnosti použitého materiálu. Je nutné vhodně stanovit důležité parametry, které musí konstruktér o materiálu znát, a ty následně správně identifikovat. Je přitom potřeba znát i pozadí fyzikální podstaty problému. Lze si pak totiž lépe udělat představu o věrohodnosti (či nevěrohodnosti) získaných hodnot. Tato práce má skloubit výše uvedené fenomény. Její praktickou částí je identifikace materiálových parametrů z experimentálně získaných dat.

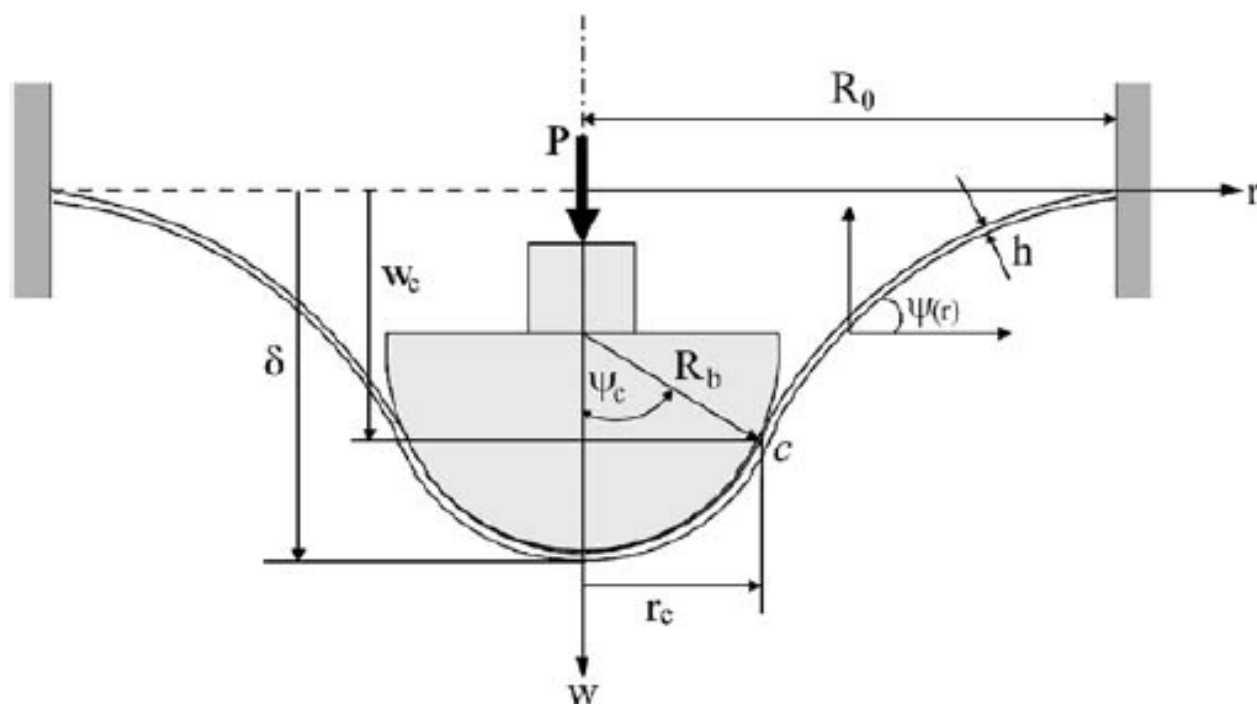
Zadáním této práce bylo najít parametry modelu porušení zadaného materiálu, konkrétně jde o materiál *DC01* dodaný firmou Comtes FHT. Značná část objemu práce spočívá v hledání optimální varianty MKP simulace tváření plechu kulovým razníkem. Na základě porovnání výpočtu s experimentálně získanými daty poté určuje koeficienty modelu porušení. Práce je rozdělena do následujících statí: v kapitole 2 bude čtenář uveden do problematiky hlubokého tažení na rekonstrukci analytického přístupu uvedeným v článku [1]. Kapitola 3 následně vysvětlí přístupy k simulaci MKP. Budou popsána úskalí, na která autor během řešení narazil. Závěrem kapitola konstatuje přístup, který se osvědčil jako nejvhodnější. Po nalezení vhodného způsobu simulace Nakazima punch testu přijde na řadu hlavní náplň zpracovávaného tématu, a to hledání parametrů modelu porušení. Teorie k této problematice je uvedena v kapitole 4 popisující základní znalosti o tenzorech, které byly potřebné ke správnému sestavení použitého materiálového modelu. Další teoretickou oblastí bude problematika optimalizace cílové funkce. Optimalizací byly stanoveny parametry použitého modelu porušení. Model porušení byl do simulace zaveden jako materiálový parametr, se kterým byla následně simulace opakována. Dosažené výsledky a jejich porovnání s experimentálními daty nalezne čtenář v kapitole 6.1. v kapitole 6.2 bude práce celkově shrnuta po stránce přístupu k problému a bude zhodnocen její přínos jak pro autora tak pro zadavatele práce - firmu Comtes FHT.

# Kapitola 2

## Analytické řešení

### 2.1 Rekonstrukce experimentu

Hlavním předmětem této práce je numerická simulace punch testu, zejména pak proces tvár-  
ného porušení během procesu. Je však vhodné zabývat se i řešením analytickým. To totiž  
může poskytnout přijatelnou aproximaci zadaného problému, rychlejší výpočet očekávaných  
závislostí, v neposlední řadě samozřejmě hlubší vhléd do podstaty zkoumaných dějů. Jak je  
ale známo, analytická řešení vyžadují celou řadu zjednodušení. Zjištěné závěry se tak mohou  
lišit nejen od numerické simulace, ale zejména od reality. Ověření aplikovatelnosti numeric-  
kého modelu uvedeného v literatuře je součástí této práce. Následující kapitola přebírá vztahy  
odvozené v článku [1]. Pomocí nich bude určena závislost silové odezvy a porušení materiálu  
během tváření. V kapitole 6.1 budou zjištěné hodnoty porovnány s výsledky MKP simulace  
a Nakazima experimentu.



Obrázek 2.1: Geometrie konfigurace punch testu

### 2.1.1 Vyšetření odezvy síla - posuv

Pro řešení úlohy je zavedeno několik předpokladů. První z nich je ten, že v oblasti velkých plastických deformací lze zanedbat ohybovou tuhost plechu. Druhým předpokladem je nulová tečná deformace -  $\epsilon_\theta = 0$ . Dále jsou pro pozdější účely zavedeny vnitřní síly:

$$N_r = \sigma_r h \quad (2.1)$$

$$N_\theta = \sigma_\theta h \quad (2.2)$$

Na obrázku 2.1 je úloha zobrazena již v zatíženém stavu. Na tvářeném plechu lze najít dva regiony: region v *kontaktu* a *mimo kontakt*. Je nasnadě, že silová rovnováha bude v každém regionu jiná. Pro region v kontaktu bude za předpokladu nulové ohybové tuhosti platit známá Laplaceova rovnice pro tenkostěnnou membránu

$$\frac{N_r}{R_b} + \frac{N_\theta}{R_\theta} = p \quad (2.3)$$

Symbol  $p$  značí hodnotu kontaktního tlaku mezi razníkem a deformovanou deskou (plechem).

$$P = 2\pi N_r r_c \quad (2.4)$$

za  $R_b$  - poloměr razníku - lze dosadit

$$r_c = R_b \sin(\psi_c) \quad (2.5)$$

$r_c$  je poloměr průmětu hranice regionu kontaktu do horizontální roviny. Vznikne tak vztah

$$P = 2\pi r_c N_r \sin(\psi_c) \quad (2.6)$$

Napěťová odezva materiálu na zatížení je určena známou mocninnou závislostí

$$\sigma = C_0 \epsilon^n \quad (2.7)$$

- $C_0$  je silový modul
- $n$  - exponent zpevnování materiálu

Tyto koeficienty byly stanoveny proložení exponenciální regresní křivky naměřenými daty.

Po dosazení do vztahu 2.7 za deformaci  $\epsilon$  a vynásobení výrazu aktuální tloušťkou plechu je

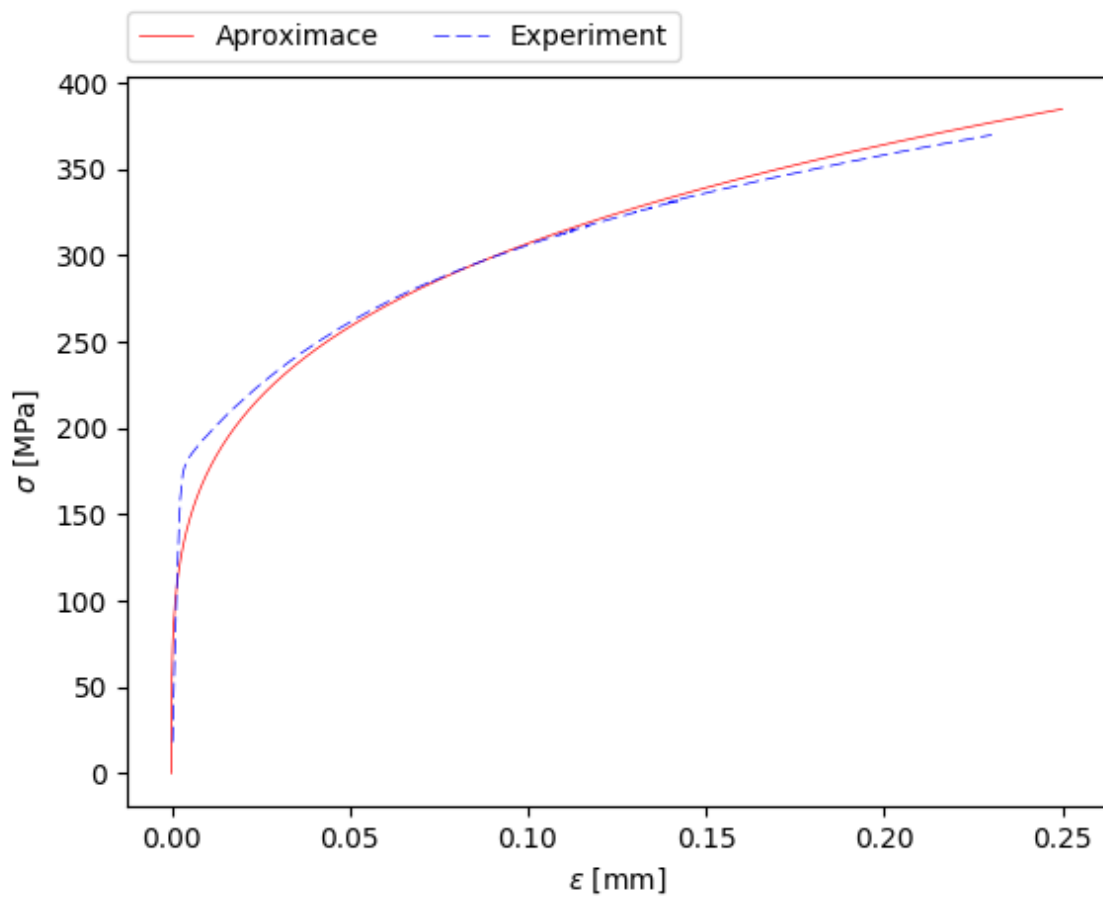
$$N_r = C_0 \left( \frac{1}{2} \tan^2(\psi)^n \right) h_0 \cos(\psi) \quad (2.8)$$

a výraz pro sílu na razníku v závislosti na úhlu  $\psi_c$ , což je úhel mezi osou razníku a radius vektorem bodu hranice regionu kontaktu, viz obr.2.1, má podobu

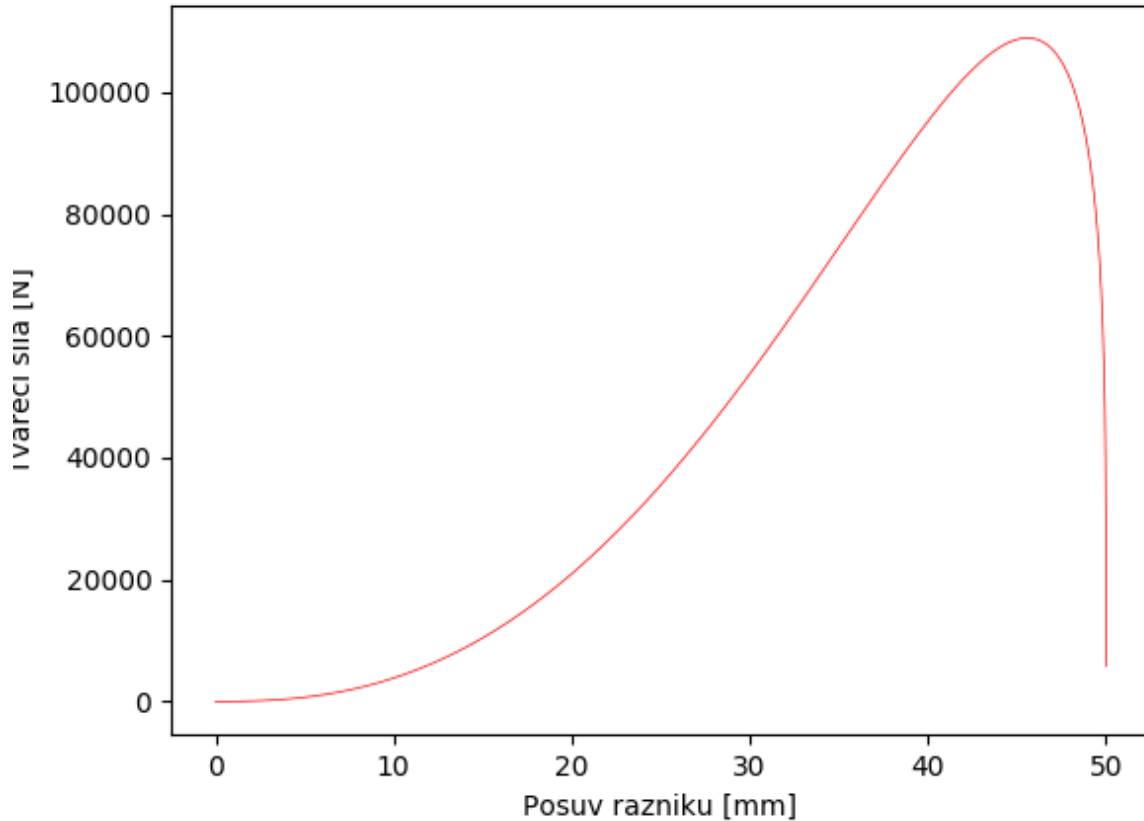
$$P(\psi_c) = 2\pi r_c C_0 h_0 R_b \left( \frac{1}{2} \tan^2(\psi_c) \right)^n \cos \psi_c \sin^2(\psi_c) \quad (2.9)$$

Průběh síly v závislosti na posuvu bez uvažování změn ve struktuře materiálu (bez uvažování poškození materiálu) je na obr. 2.3.

Jak bude ukázáno v kapitole 6.1, hodnoty síly odpovídají řádově hodnotám naměřeným v experimentu.



Obrázek 2.2: Zobrazení křivky regresní závislosti a její porovnání s křivkou získaných dat



Obrázek 2.3: Průběh síly v pístu testovacího zařízení v závislosti na posuvu kulového razníku získaný analytickou aproximací

### 2.1.2 Inicializace trhliny

Pro popis vzniku trhliny v materiálu jsou zavedeny modely, podle kterých je trhlina iniciována. Modely porušení jsou podrobněji popsány v kapitole 4.4. Na tomto místě je pouze zavedena veličina *triaxialita napětí* -  $\eta$

$$\eta = \frac{\sigma_m}{\bar{\sigma}} \quad (2.10)$$

jako poměr prvního invariantu tenzoru napjatosti a ekvivalentního napětí. Dále je zaveden parametr porušení  $D_c$  (z anglického slova *damage*). Jde o číselné vyjádření míry kumulovaného poškození během zatěžování. Není proto překvapující, když je vyjádřen jako integrální hodnota

$$D_c = \int_0^{\epsilon^f} f\left(\frac{\sigma_m}{\bar{\sigma}}\right) d\epsilon \quad (2.11)$$

Pro hodnoty triaxiality v intervalu  $(-\frac{1}{3} < \eta < 1.0)$  lze aproximovat tuto závislost jako lineární, tedy

$$D_c = \int_0^{\epsilon^f} \eta d\bar{\epsilon} \quad (2.12)$$



Následný úhel  $\psi_{c_f}$  je určen z analytického řešení v článku [1] vztahem

$$\psi_{c_f} = \left( \frac{D_c \sqrt{3}}{\eta} \right) \quad (2.13)$$

Velikost hodnoty indentace razníku v okamžiku porušení má pak hodnotu

$$\delta_f = R_b \sin(\psi_{c_f}) \quad (2.14)$$

## 2.2 Závěr a komentář výsledků získaných z analytického řešení

v této kapitole byla aplikována odvození z článku [1], který slouží jako její předloha. Jejím smyslem bylo extrahovat vztahy, které vedou ke zjištění maximální indentace razníku do okamžiku vzniku trhliny a následného kolapsu geometrie vzorku. Výsledky zjištěné tímto postupem jsou shrnuty v kapitole 6.1

# Kapitola 3

## Simulace metodou konečných prvků

Metoda konečných prvků (MKP) představuje silný nástroj co do možnosti predikce chování těles při různých zatíženích, v našem případě jde o proces tváření. Je ale přitom potřeba výpočtovému systému zadat parametry řešené úlohy tak, aby výsledky korespondovaly s fyzikální podstatou daného problému. Současně je přitom třeba vzít do úvahy délky výpočtových časů, náročnost preprocessingu atd. Počítačová technika a použité matematické nástroje umožňují přijmout řadu aproximací, které mohou úlohu do jisté míry zjednodušit a přitom neznehodnotit její výsledky. Je ale potřeba, aby uživatel věděl, jak se tyto nástroje chovají a za jakých předpokladů je lze použít. Tato kapitola by se měla věnovat právě těm zjednodušením, které byly použity pro řešení zadaného problému.

### 3.1 Úvod do prostředí ABAQUS

Abaqus je MKP software vyvíjený firmou Dassault Systemes. Vyniká mezi ostatními MKP softwary zejména svojí robustností. Pro uživatele je také velmi důležitá kvalitní dokumentace [2]. Silným nástrojem je dále možnost spouštění uživatelem napsaných scriptů a rutin. Úlohu tak lze přizpůsobit na míru řešeného problému. To vše dělá ABAQUS velmi komplexním a univerzálním, zároveň ale bohužel poměrně drahým.

Tvorba modelu probíhá v programu ABAQUS v několika modulech:

- **Part** - Jsou modelovány díly - party - analyzované v úloze. ABAQUS disponuje modelářem, existuje ale i možnost součást importovat například ve formátu *.stp*.
- **Property** - Slouží k zadání materiálu jednotlivých zkoumaných dílů. Těmto materiálům jsou přiřazeny jejich fyzikální vlastnosti - Youngův modul, Poissonovo číslo, etc.
- **Assembly** - Sestavení jednotlivých dílů dohromady.
- **Step** - Zadávání kroků úlohy. Programu je tímto způsobem zadáno např.: „Vezmi vytvořený nosník a aplikuj axiální sílu. Následně vynuť na jeho konci posuv 0,1 mm.“ V modulu Step je předepsána i velikost a způsob inkrementace nebo typ řešiče.
- **Interaction** - Předepisování kontaktů (jsou-li), zavádění cyklické symetrie, svázání tělesa s referenčním bodem apod.
- **Load** - Předepisování okrajových podmínek a zatížení.

- **Mesh** - Vytváření sítě (meshe). Jak bude vidět později, síť významně ovlivňuje výsledky řešení.
- **Optimization** - Optimalizace úlohy
- **Job** - Generování vstupního souboru (input file) a jeho následné předložení řešiči k výpočtu.
- **Visualization** - Zobrazení výsledků získaných výpočtem.

Pořadí těchto modulů dává uživateli návod pro vytvoření zadání úlohy a možnost postupovat „krok za krokem“.

## 3.2 Popis fyzikální předlohy úlohy

Pro snazší čtenářovu orientaci v následujících kapitolách bude nyní popsán průběh úlohy. Během jejího řešení bylo zavedeno pojmenování pro označení jednotlivých dílů, které je dodržováno během celé práce. Je převzato z terminologie manuálu ABAQUs [2] a je následovné:

- Punch - kulový razník zatěžující tvářený plech
- Holder - přidržovač plechu
- Die - zápustka
- Blank - tvářený plech

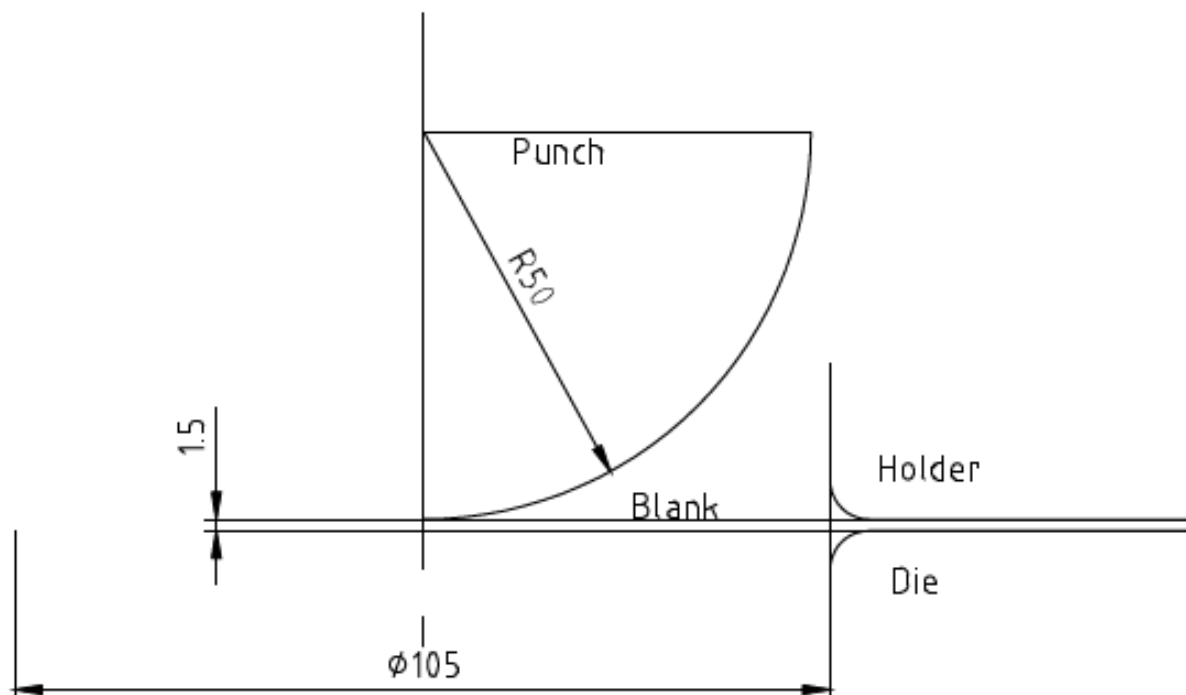
Tělesa Punch, Holder a Die jsou modelována jako objekty dokonale tuhé. Blank je modelováno jako *poddajné (deformable)* těleso. Úloha je vymodelována podle schématu na obr. 3.1.

Úlohu lze rozdělit do dvou kroků. V prvním je vyvozena síla na Holder tak, že plech je chycen mezi přidržovačem a zápustkou (Holder a Die). V simulaci MKP je toto reprezentováno krokem *Holder force*. Je tak zamezeno prokluzování plechu během procesu tváření. Ve druhém kroku, v simulaci označeném jako *Move punch*, je přistoupeno k samotnému tváření. Plech je natvrdo zatížen posuvem kulového razníku. Úloha končí v okamžiku porušení plechu. Do úlohy významně promlouvá koeficient tření zejména mezi razníkem a plechem. Referenční veličinou, na jejímž základě budou porovnávány výsledky jednotlivých variant analýzy, je hodnota síly na razníku.

## 3.3 2D úloha

2D (rovinná úloha) je typ úlohy, kdy jeden rozměr součásti má řádově nižší hodnotu než zbylé dva rozměry. Typicky jde například o tenkostěnné nádoby, desky atp. To umožňuje problém zjednodušit a šetřit tak výpočetní kapacitu ať už v řešičích MKP, či při analytických výpočtech. Rozlišují se dva základní typy 2D úloh - rovinná napjatost a rovinná deformace. Tyto úlohy jsou popsány v kapitole 4. Elementy pro úlohy rovinné napjatosti, resp. rovinné deformace jsou obsaženy v řešičích MKP.

Při vyšetřování popisu punch testu daného zadáním posloužila 2D úloha jako jakýsi velmi zjednodušený přístup k problematice. Je třeba si uvědomit, že toto řešení není symetrické podél osy, ale podle roviny, takže ani nereprezentuje zadanou konfiguraci. Cílem bylo především



Obrázek 3.1: Schema úlohy Nakazima punch testu

autorovo vpravení se do prostředí ABAQUs a zažití předepisování správných parametrů úlohy. Porovnání získaných výsledků s výsledky dalších typů úloh poslouží k určení toho, jak moc "hrubě" je možno k zadanému problému přistoupit a s jakou tolerancí pak musíme považovat dané výsledky za validní.

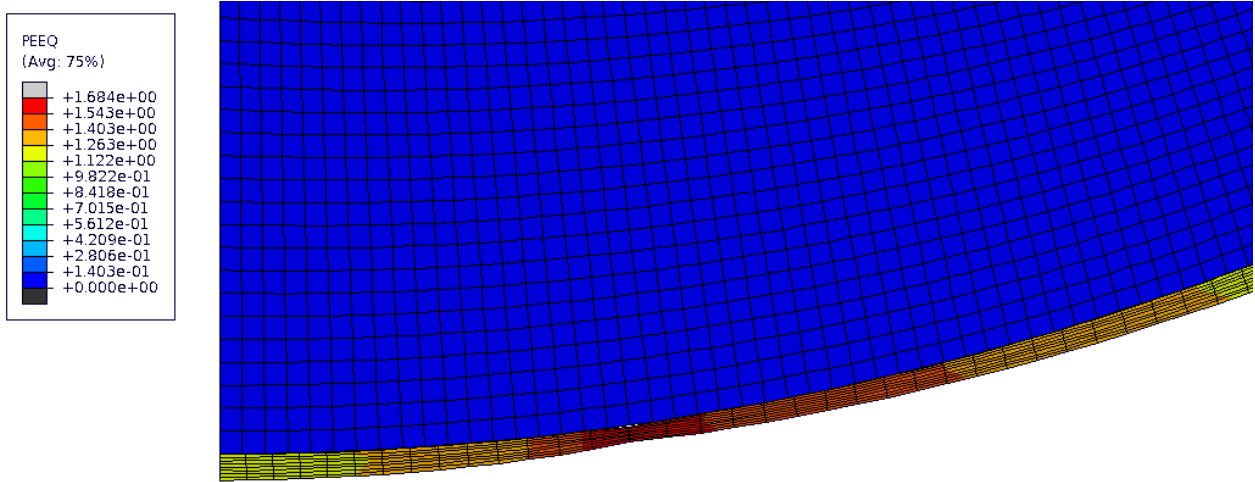
### 3.4 Rotačně symetrická úloha

Tento typ již reprezentuje geometrii punch testu resp. jejího zadání. Obtížně však bude reprezentovat okolí kritického časového inkrementu, za který považujeme porušení plechu. Další otázka vyvstává nad problémem dvojí křivosti plechu v deformovaném stavu v kontaktu plechu se zápustkou.

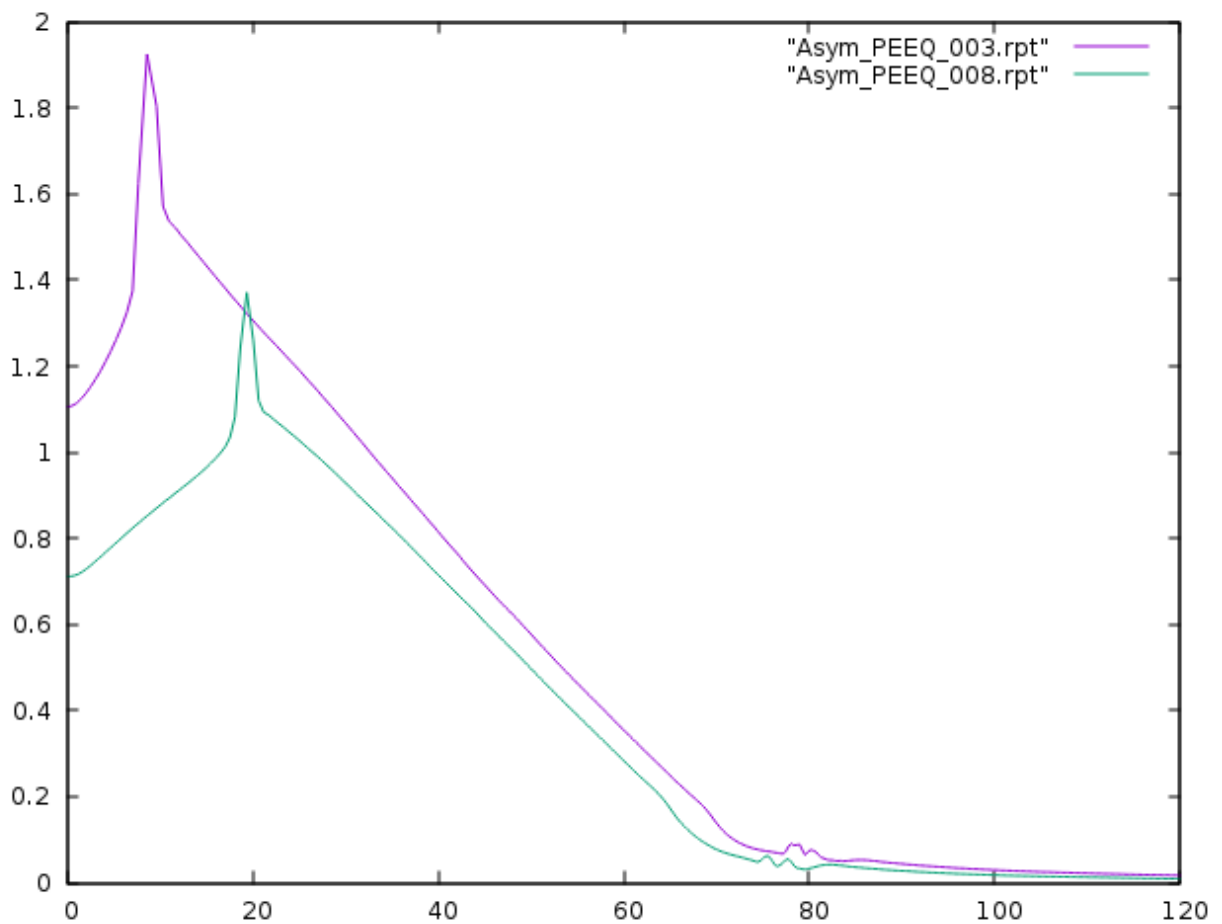
obr. 3.2 dobře zobrazuje místo největší plastické deformace, které odpovídá fyzikálnímu předpokladu vycházejícího z analytického řešení<sup>1</sup>. Pro získání podrobnějších výsledků ale není tato metoda vhodná. Hodnoty síly v závislosti na posunutí neudávají věrohodné výsledky.

Vzhledem k nízké výpočtové náročnosti byla tato úloha provedena pro dvě hodnoty koeficientu tření. Graf zobrazuje, jak právě tření ovlivňuje odezvu při simulaci punch testu, kdy vyšší hodnoty tření posouvají oblast největší plastizace dál od středu geometrie.

<sup>1</sup>Poloha kritického místa, kde nakonec dojde k porušení materiálu, je důležitá při posuzování kvality provedení experimentu punch testu.



Obrázek 3.2: Průběh plastické deformace v ose symetrické úlože



Obrázek 3.3: Porovnání průběhu plastické deformace v radiálním směru od středu plechu

## 3.5 Cyklicky symetrická úloha

Cyklickou symetrii lze použít v případech, kdy existuje osa symetrie, okolo které jsou rozmístěny tvarově shodné prvky, které bychom do úlohy zadávali pouze jejich opakovaným vkládáním. Je nasnadě si uvědomit, že takový postup prodlužuje čas preprocessingu, postprocessingu i výpočtový čas. Je tedy vzhledem k požadavkům na MKP výhodnější tento díl vložit jednou a následně zadat programu, kolikrát bude prvek okolo osy rozložen. Typickým zástupcem cyklicky symetrických těles je například turbína. Ta je sestavena z tělesa, do kterého jsou přes zámky zasazeny lopatky. Z podstaty věci budou všechny lopatky zatíženy stejně, tj. zjistili bychom v nich stejná pole napětí i deformací. Není tedy potřeba řešit každou zvlášť, stačí řešení jedné lopatky. Je však nutné uvažovat jejich hmotnostní a setrvačné účinky. Proto je zaváděna cyklická symetrie.

Úloha punch testu neobsahuje cyklicky se opakující prvky. Je rotačně symetrická. Lze tudíž řešit pouze část 3D tělesa a vyšetřit namáhání v této vyjmuté části. Při řešení našeho zadání byla vyjmuta část zobrazená na obr. 3.4. V úloze cyklické symetrie byl model rozvětven tak, že byl uvažován razník deformovatelný a analyticky tuhý.

Nyní je třeba čtenáři vysvětlit důvody, pro které je vhodné sledovat řešení s tuhým i poddajným přípravkem. První důvod je ten, že je dobré MKP výpočty neustále ověřovat a verifikovat. Během řešení úlohy vyšlo najevo, že pokud jediné, co ve výpočtu změním, je nahrazení deformovatelného razníku za tuhý, neobdržíme stejné výsledky a i stabilita výpočtu se v obou případech liší. Postup do fáze výpočtu, který je možno považovat za platný, proběhl v několika krocích. Ty budou nyní podrobněji rozebrány.

### 3.5.1 Cyklicky symetrický model s dokonale tuhým razníkem

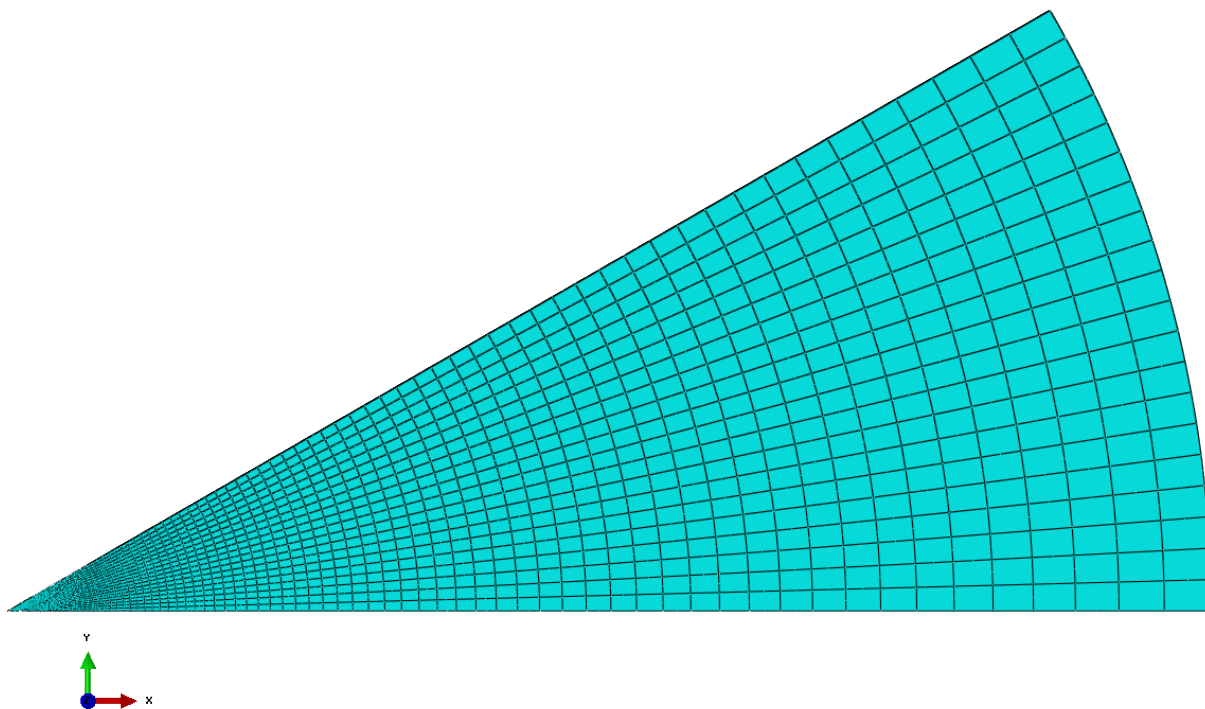
Hlavními modifikovanými parametry úlohy jsou tvar sítě resp. elementy na tělese Blank a třecí koeficient. Jako první konvergující se ukázala varianta s elementy C3D8R, což jsou 3D lineární elementy s osmi integračními body. Síť byla vytvořena metodou sweep. Výsledný tvar sítě je na obr. 3.4

Tato úloha dokonvergovala v poměrně příznivém čase přibližně čtyř hodin na jedno CPU. Obr.3.5 ukazuje, že poloha regionu největší míry plastizace se shoduje s tím, co bylo zjištěno v úloze rotačně symetrické a co odpovídá analytickému výpočtu.

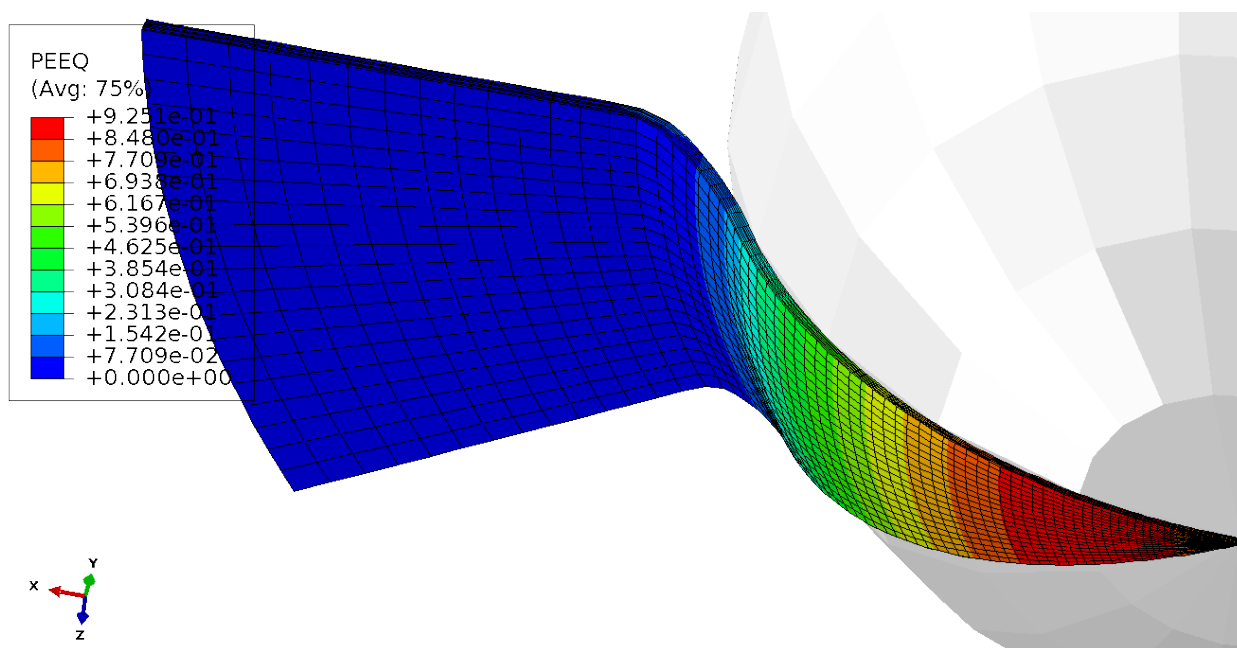
Při pohledu na obr. 3.6 je však zřejmé, že hustota sítě v kontaktu se zápusťkou není dostatečná pro získání korektních výsledků. Pro zlepšení přesnosti výpočtu zejména v okolí kontaktu těles *Blank* a *Die*, kde dochází k dvojímu zakřivení plechu, byla modifikována síť (viz obr. 3.7). Ve snaze kompenzovat nárůst počtu elementů, který tak vznikl, byla snížena hustota sítě v kontaktu *Holder* a *Die*. Dále byl změněn typ elementů na kvadratický - C3D20R. Tvar sítě je vidět na obr. 3.7

### 3.5.2 Cyklicky symetrický model s poddajným razníkem

Zadání úlohy se od předchozího liší v tom, že razník je modelován jako *Deformable* těleso. Musí být tudíž vysíťováno. Vazbou *Rigid body* lze ale předložit tyto uzly do výpočtu jako pevně svázané, a prvky tohoto tělesa tak v samotném výpočtu resp. matici tuhosti nefigurují. Výrazně je tak výpočtu odlehčeno. Stále je ale třeba mít na paměti fakt pozměněné geometrie vysíťovaného povrchu koule. Ta totiž přestává mít ideálně kulový tvar a je sestavena z plošek. Tuto vadu lze do jisté míry podchytit při zadávání vlastností kontaktu, kdy byla zahrnuta va-



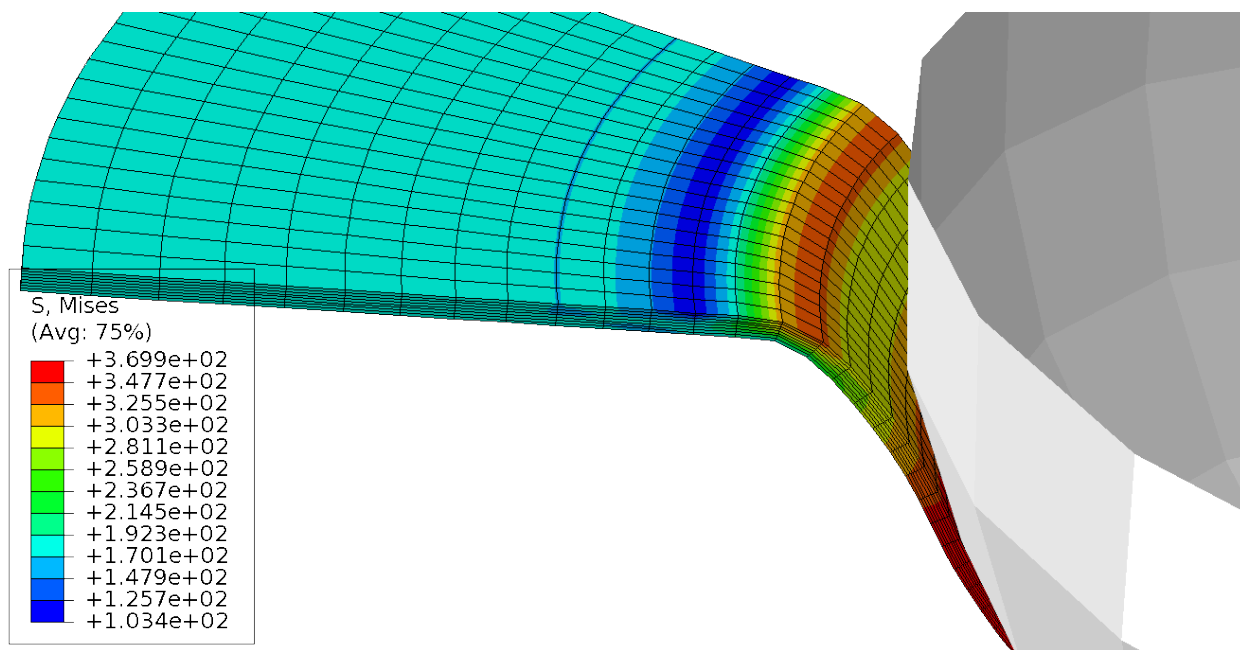
Obrázek 3.4: Prvotní tvar sítě v úloze cyklické symetrie



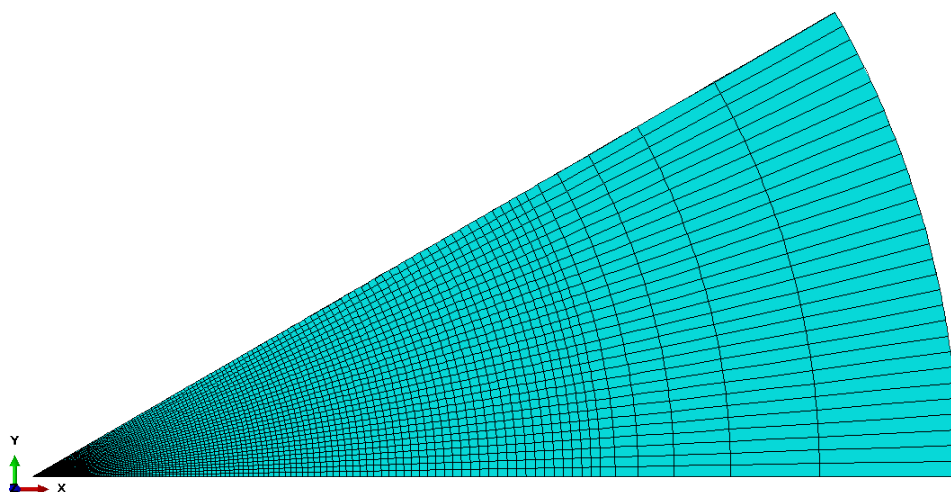
Obrázek 3.5: Rozložení ekvivalentní plastické deformace v úloze s tuhým razníkem

rianta automatického vyhlazení povrchů v kontaktech, což se ukázalo pro tuto úlohu jako vyhovující.

Při analýzách, kdy bylo použito lineárních elementů a nebylo zahrnuto automatické vyhla-



Obrázek 3.6: Rozložení napětí v místě kontaktu plechu a přípravku

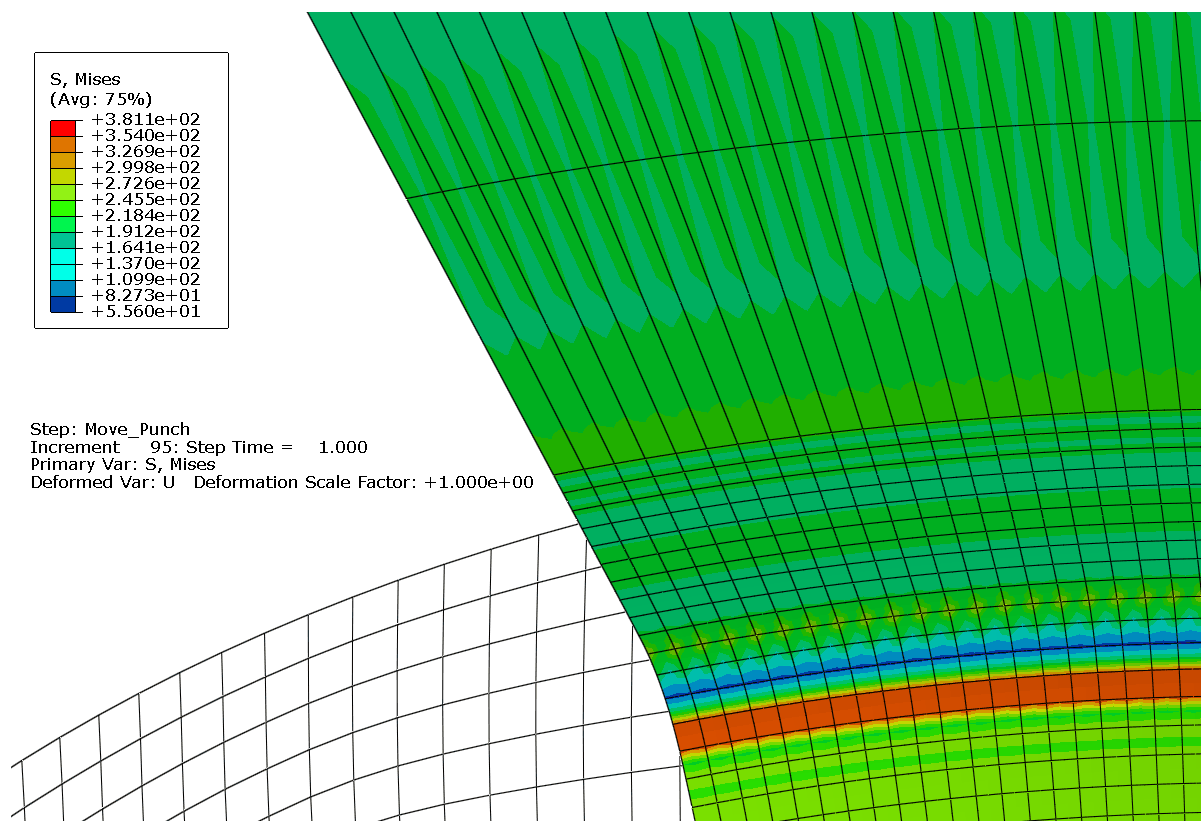


Obrázek 3.7: Tvar první optimalizace sítě

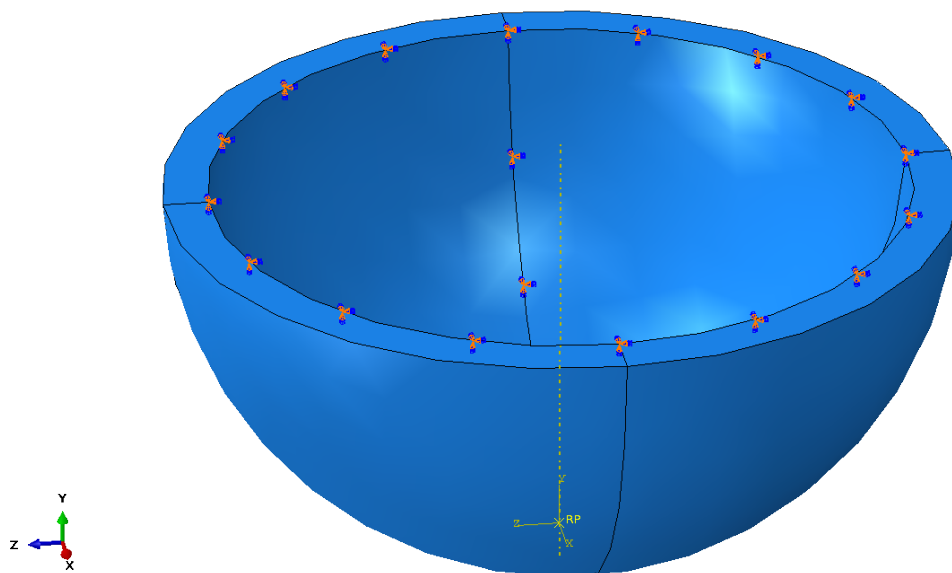
zování síťovaných zakřivených ploch v kontaktech, docházelo zejména v místech dvojí křivosti ke koncentraci napětí v uzových bodech. Zlepšení sítě a změna typu elementů na kvadratické tyto negativní jevy odstranily. Daní za to je výrazně delší výpočtový čas. Ten oproti geometricky shodné analýze v provedení *Rigid* narostl asi 24 krát. Při použití jednoho CPU by výpočet trval zhruba 8 dnů.

Pro širší rozhled byla provedena i úloha s čistě poddajným razníkem. Vymodelován byl přitom pouze vnější povrch razníku v tloušťce pěti elementů - viz obr. 3.9. Tím byl vytvořen kompromis mezi geometrií a výpočtovou náročností úlohy.





Obrázek 3.8: Zobrazení pole napětí HMH po optimalizaci sítě v úloze s deformovatelným razníkem



Obrázek 3.9: Poddajný razník

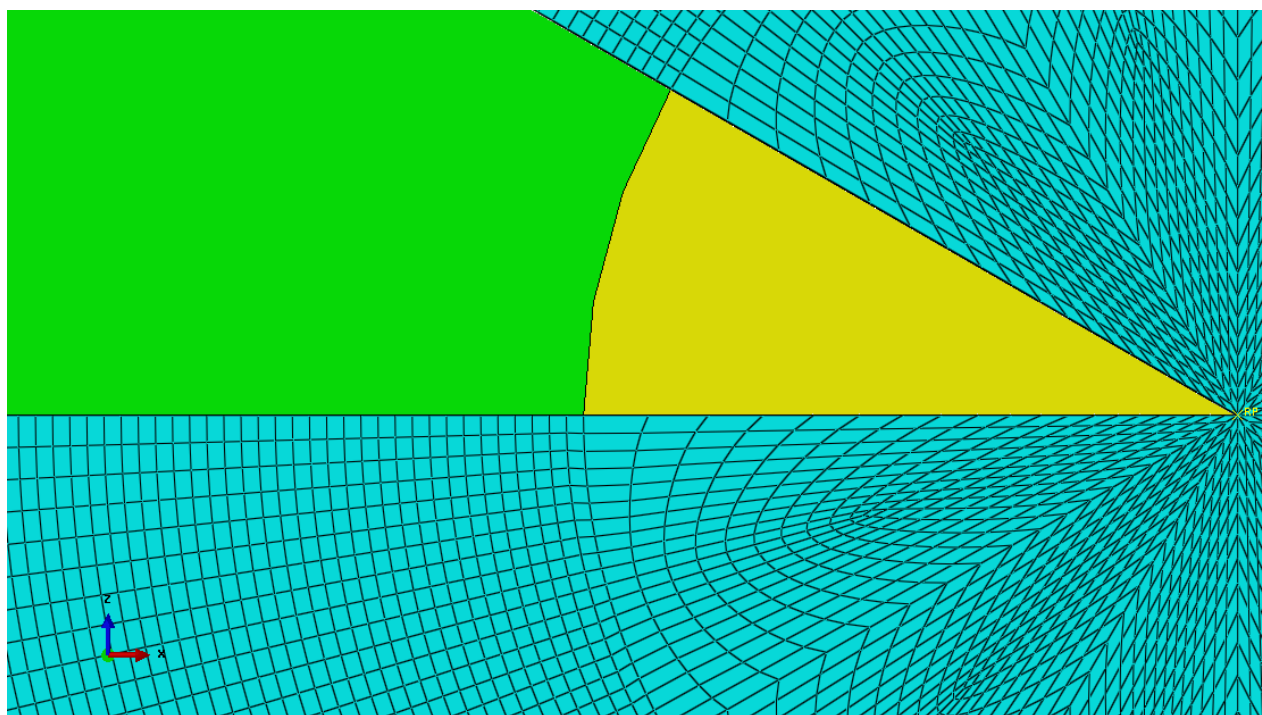
### 3.5.3 Porovnání úloh Rigid a Deformable

Hlavním zjištěním, které vyplynulo z cyklicky symetrické úlohy, je určitá nevyzpytatelnost analyticky tuhých těles v kontaktních úlohách. V tomto typu úlohy opakovaně docházelo k pro-

blémům s konvergencí na startu druhého stepu, tedy začátku tvářecího procesu, kdy kulový razník "zajíždí" do plechu a tak s ním přichází do kontaktu. Jako cesta nejmenšího odporu se ukázala právě varianta *Deformable*, ať už při použití vazby *Rigid body*, která v podstatě vyrábí z deformovatelného tělesa tuhé, či při analýze s deformovatelným "vybraným" razníkem. Vazbu *Rigid body* by mělo teoreticky suplovat předepsání společného posuvu všech uzlů razníku. Je proto paradoxem, že výpočet havaroval, a to opět na začátku Stepu 2.

### 3.6 Plná 3D úloha

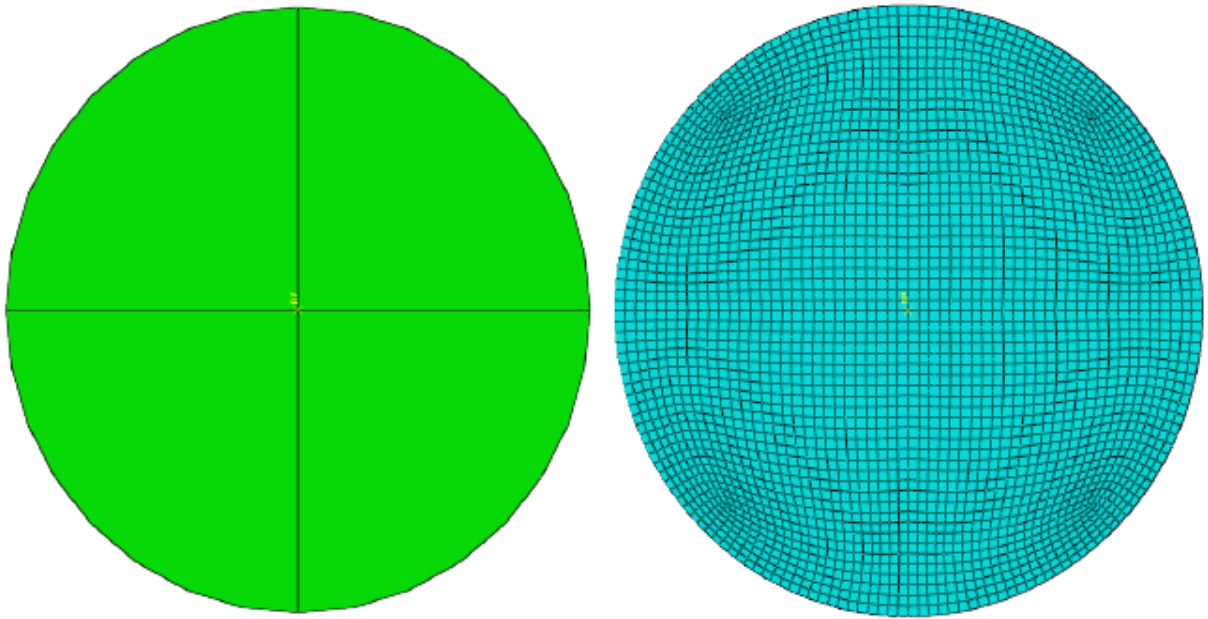
V plné 3D úloze se, stejně jako v přechozích analýzách, jeví jako nejdůležitější tvar sítě. Vzhledem k povaze úlohy je důležité zachovat síť směrově nezávislou. Jak ale ukázaly pokusy během procesu síťování, absolutní izotropie sítě nelze dosáhnout. I přes to, že je geometrie plechu rozdělena pomocí nástroje *partition* na segmenty shodné s tím, který byl použit v úloze cyklicky symetrické, nevytvoří síťovač Abaqusu shodnou síť pro každý segment, ale takovou, jaká je na obr. 3.10. Elementy na středu plechu jsou velmi deformovány a síť postrádá pravoúhlost šestistěnných elementů. Byla proto změněna strategie síťování. Geometrie plechu byla rozdělena na čtyři segmenty po 90° kruhové výseče. Tato geometrie umožňuje vytvoření strukturované sítě, která většinou vykazuje pravidelnější tvar elementů.



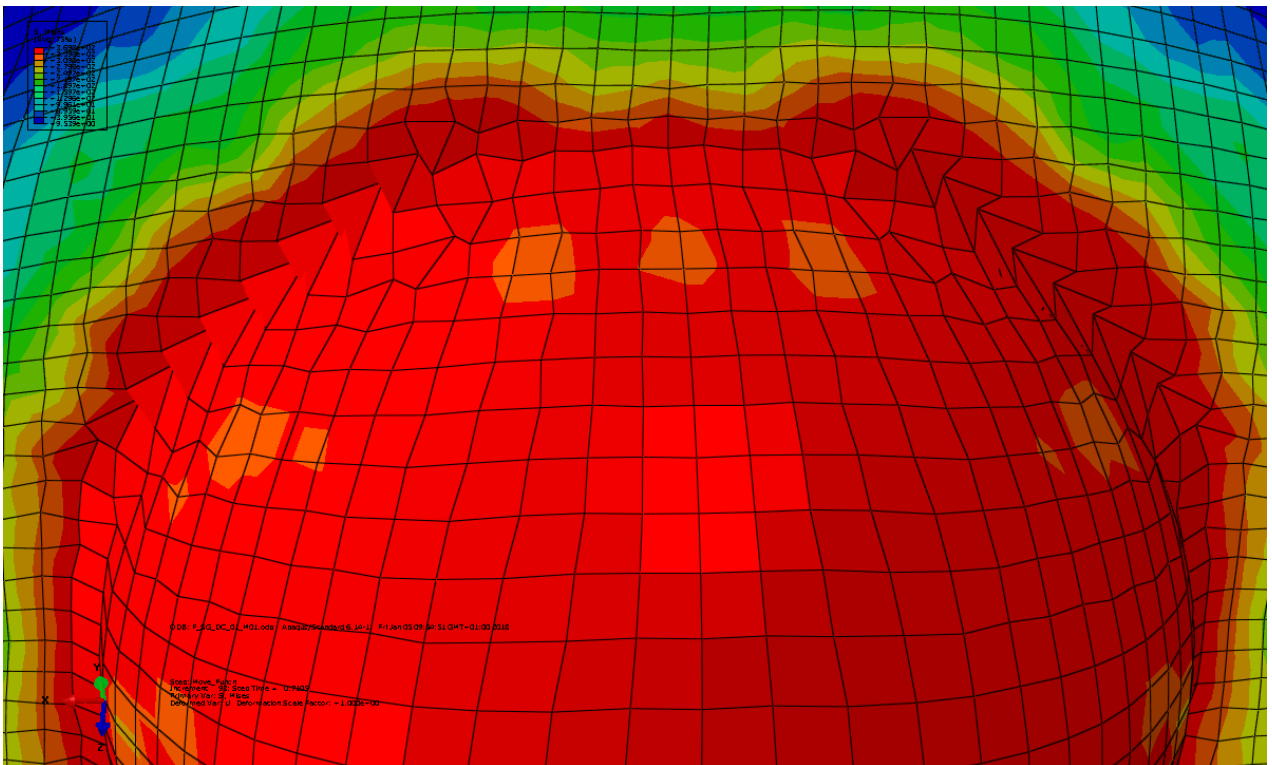
Obrázek 3.10: Nevhodný tvar sítě na středu geometrie analyzovaného plechu

Jak ukazuje obr 3.12, celkový tvar sítě ale pravidelný není. Výsledky pravděpodobně budou směrově závislé. Míru směrové závislosti ovlivní hustota sítě, kterou však nelze zvyšovat neomezeně.

Výsledky řešení s takovou sítí se neukázaly jako vyhovující. Kromě směrové závislosti vykazují i velkou chybu v okolí kontaktu plechu a zápustky. Za pozornost také stojí pozice elementů vůči geometrii přípravku, vlivem které vzniká nekonzistentnost polí napětí a deformací. Stejně jako v předchozích typech úloh je třeba síť v místě kontaktu se zápustkou zjemnit a zajistit



Obrázek 3.11: Rozdělení geometrie a výsledný tvar sítě

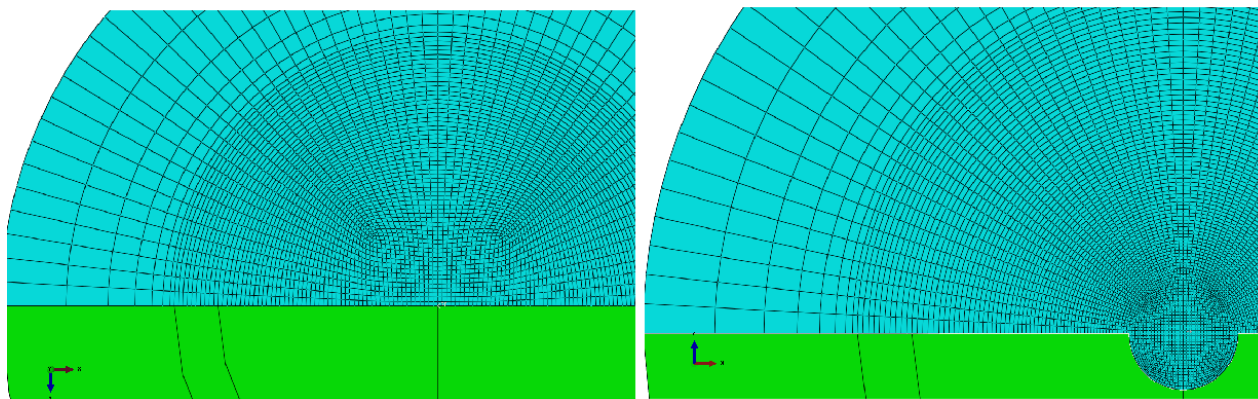


Obrázek 3.12: Deformovaná 3D geometrie plechu

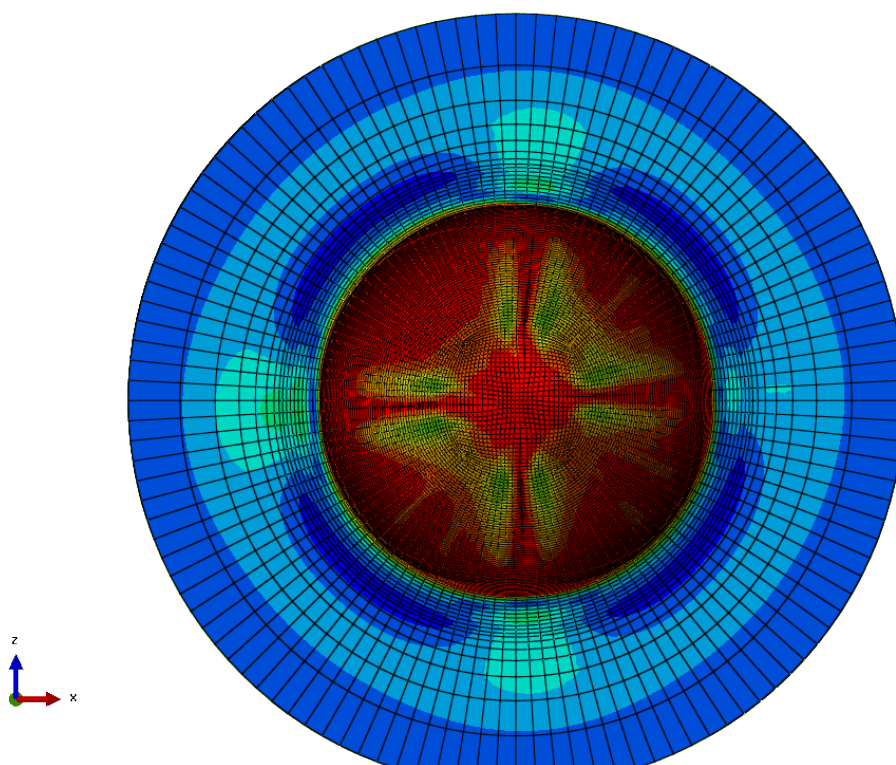
její co možná největší respektování geometrie úlohy. Ideální požadavek by tedy byl vytvořit síť protnutím kružnic a přímek ve směru jejich normál. Kružnice v místě kontaktu s přípravkem by byly kladeny s větší hustotou. Problém ale vznikne v středu plechu, kde by se musely všechny

normály setkat v jednom bodě. Vzhledem k filozofii MKP to není žádoucí, síťovač Abaqusu navíc takovou síť ani neumožní vytvořit.

Byly tedy zvoleny dva přístupy k vysíťování středové oblasti plechu. První přístup spočíval v ponechání plné, nedělené kruhové geometrie středu a vytvoření sítě metodou *sweep*. Druhým přístupem bylo rozdělení středu na čtyři výseče po  $90^\circ$  a použití strukturované sítě. Metody síťování a výsledné tvary sítě jsou na obr 3.13.

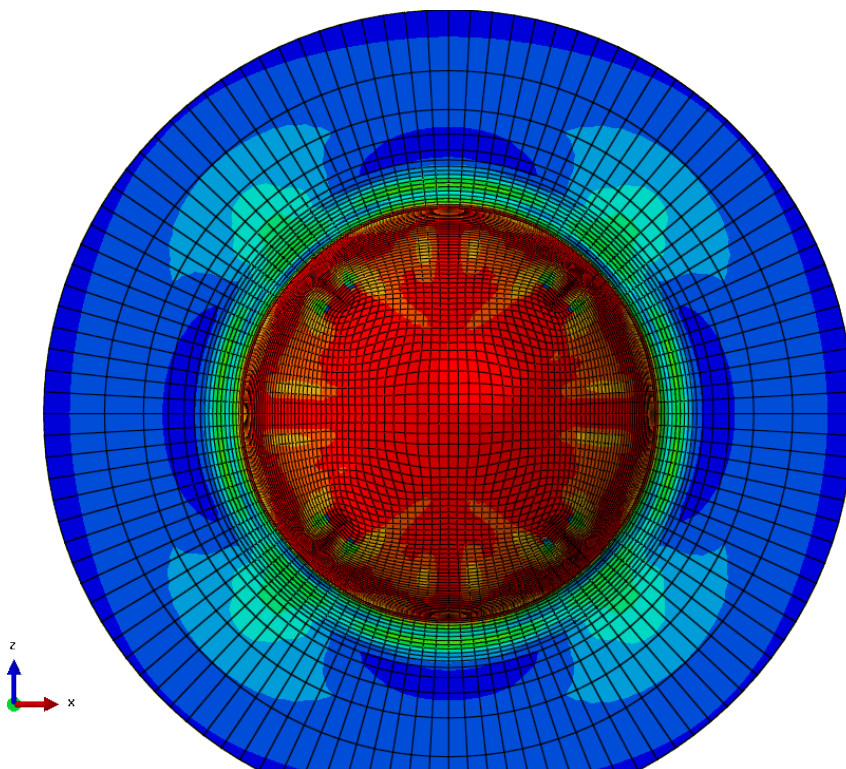


Obrázek 3.13: Porovnání sítí ve středové části. Vpravo je ve středu plechu síť tvořena metodou *sweep*



Obrázek 3.14: Zobrazení pole napětí *HMH* na geometrii s použitou metodou *sweep*

Stejně jako v úlohách cyklicky symetrických bylo přistoupeno k náhradě lineárních elementů za kvadratické. Počet uzlů se tak dostal na číslo 246 000, což je zhruba čtyřnásobek oproti úloze s lineárními elementy. I tak nejde z hlediska výpočtových úloh o nijak významný rozsah, nárůst výpočtového času je však extrémní. Spočítat takovou úlohu trvalo



Obrázek 3.15: Zobrazení pole napětí HMM, síť tvořená pouze jako *structured*

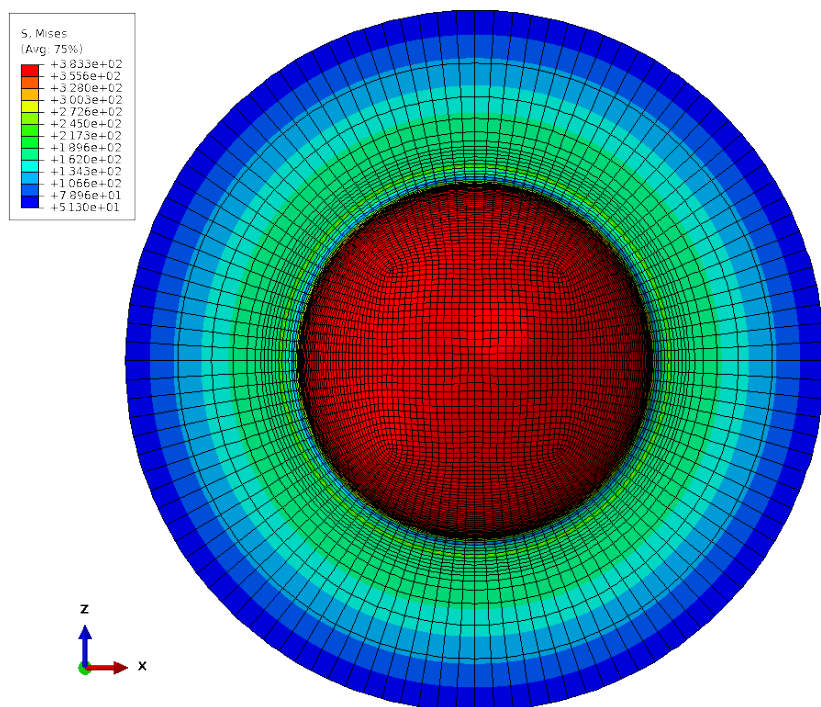
21 dní. Příčinou je pravděpodobně kontakt razníku a plechu. Kontaktní úlohy jsou stále velmi živou problematikou v oblasti MKP výpočtů. Z obrázků je vidět, že síť s kvadratickými elementy poskytla rovnoměrnější rozložení pole napjatosti, nicméně vzhledem k extrémním výpočtovým nárokům nelze považovat tuto úlohu za optimální.

### 3.7 Porovnání a komentář dosažených výsledků analýzy MKP

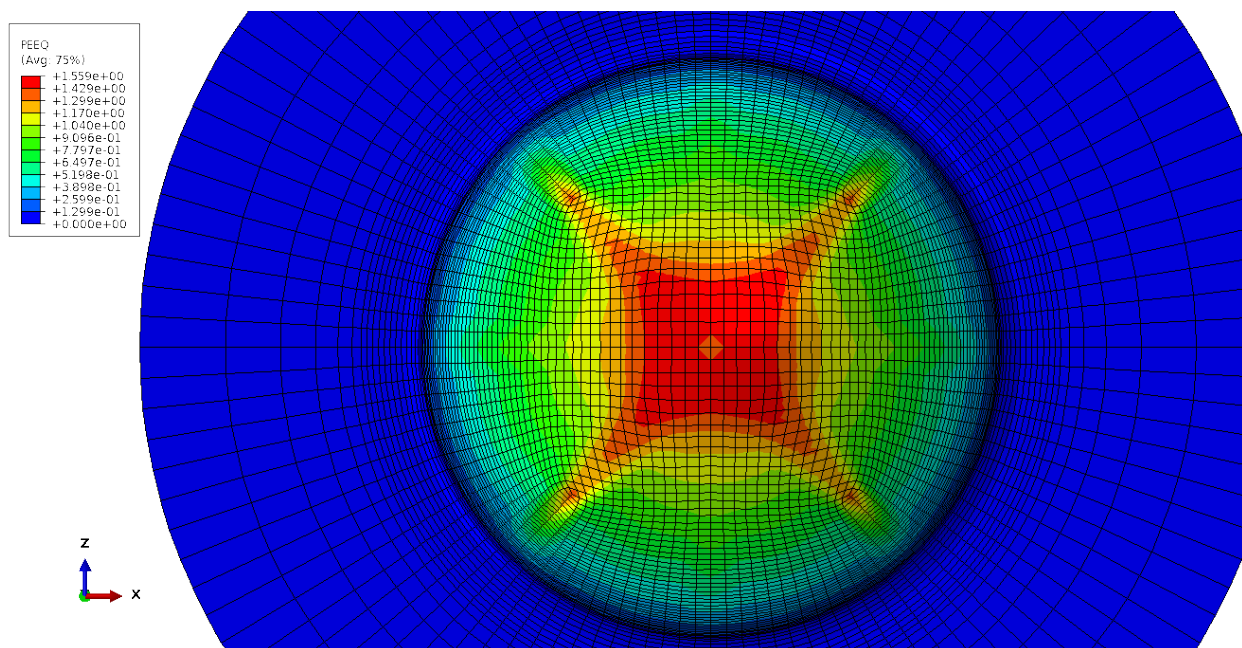
Pro všechny typy úloh byly zvoleny shodné velikosti elementů v závislosti na jejich vzdálenosti od středu plechu, a to ve dvou variantách - *Prostá* a *Optimalizovaná*. Porovnání výsledků získaných při takto nastavené úloze dává poměrně dobrou představu o tom, jak silně tvar sítě ovlivňuje výpočet a jeho stabilitu.

Je pozitivem, že všechny úlohy shodně určily region největší plastizace. Ten je ve vzdálenosti ne větší než 10% průměru razníku. Tak to předepisuje i norma Nakazima experimentu.

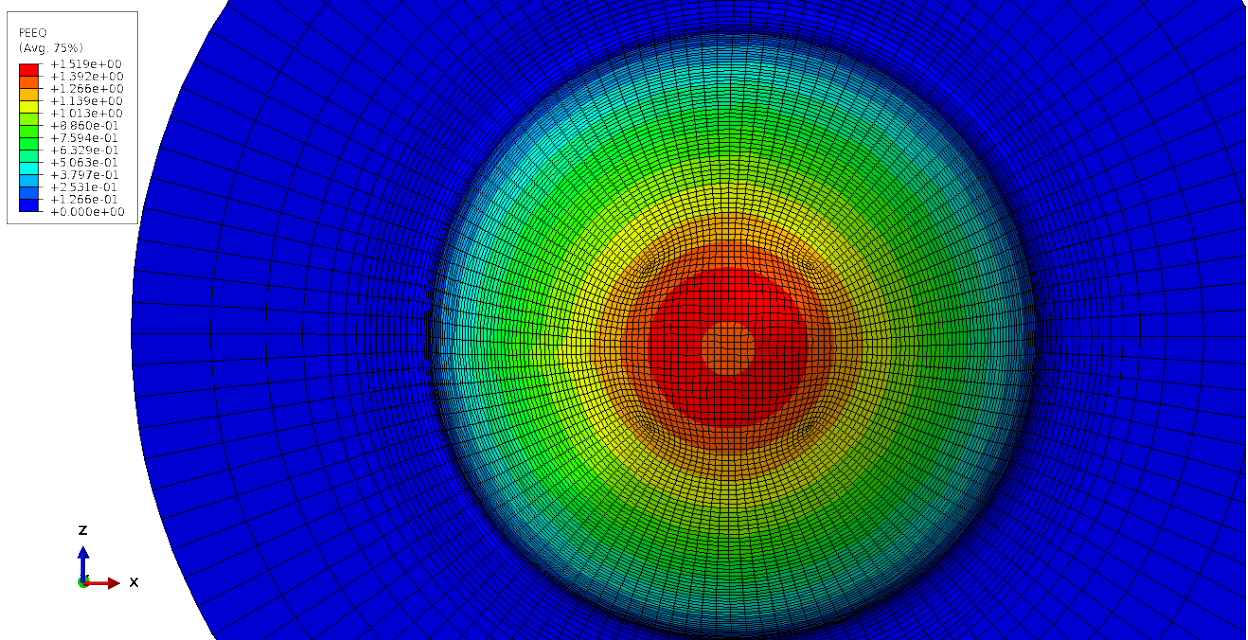
Jak již bylo zmíněno, je sledovaným parametrem zejména síla na razníku, resp. závislost na posuvu razníku během procesu tváření. Úloha cyklicky symetrická i úloha s použitou plnou 3D geometrií poskytly křivku zátěžné síly, která tvarem odpovídá experimentálním datům. Cyklicky symetrická úloha se vyznačuje menším počtem elementů. Není tudíž překvapující, že i výpočtový čas je významně kratší. Pro další simulace např. při zkoumání vlastností jiných materiálů se tudíž jeví jako nejvhodnější využívat právě cyklicky symetrický model, který zachovává symetrii úlohy, podá dostatečnou informaci o rozložení napětí ve vzorku a jeho výpočtový čas je v řádu hodin. To ovšem neplatí v případě zkoumání mechanismu porušení, kdy mizí symetrie modelu. Cyklicky symetrické modely jsou tudíž v tomto případě problematické.



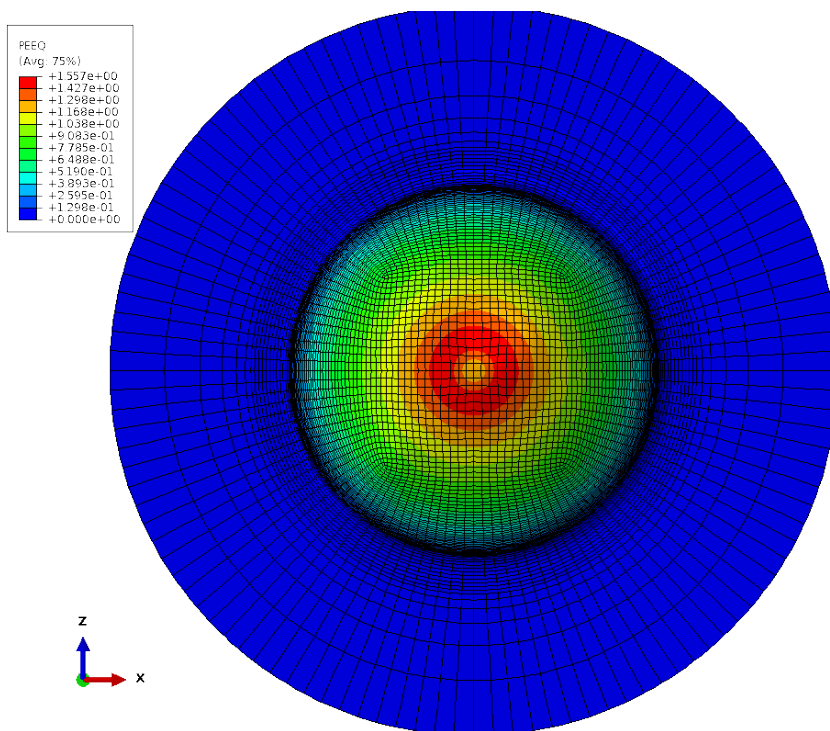
Obrázek 3.16: Zobrazení pole napětí HMM, síť tvořená pouze jako *structured*, použity kvadratické elementy



Obrázek 3.17: Zobrazení pole Ekvivalentní plastické deformace, síť tvořená pouze jako *structured*, použity lineární elementy



Obrázek 3.18: Zobrazení pole Ekvivalentní plastické deformace, střed síťován metodou sweep, použity lineární elementy



Obrázek 3.19: Zobrazení pole ekvivalentní plastické deformace, střed síťován metodou sweep, použity kvadratické elementy

# Kapitola 4

## Teoretické podklady pro identifikaci modelu porušení

V následujících kapitolách bude vysvětlena teorie, jíž je použito ke správnému popisu stavu napjatosti v materiálu. Tyto základní poznatky budou následně aplikovány pro naladění parametrů modelu poškození materiálu *DC01*, který byl během práce používán. Výstupem kapitoly je skript, který umožní provádět tuto proceduru téměř automaticky.

### 4.1 Základy teorie rovinných úloh

#### 4.1.1 Hookeův zákon

Problematika rovinných úloh probíraná v následujících dvou kapitolách má základ v úpravách Hookeova zákona. Bude zde tedy uvedena jeho zobecněná verze. Předpokladem je, že vlastnosti zkoumaného materiálu jsou lineární, tedy že platí lineární odezva deformace těles na zatížení a naopak. Tuto závislost můžeme předepsat jako

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl} \quad (4.1)$$

Tenzor  $C_{ijkl}$  je tzv. tečný modul umožňující vyjádření přírůstku napětí v závislosti na přírůstku malých deformací. Tento modul bývá také nazýván Hessovou maticí. Jde o tenzor čtvrtého řádu, v indexové notaci má proto čtyři indexy. Rozsah indexu je 1 až 3. Hessova matice má 81 prvků. Vzhledem k tomu, že je tenzor napětí symetrický, je symetrická i Hessova matice. Lze tak volně zaměňovat indexy  $i$   $j$  a  $k$   $l$ . Toto je nazýváno *malými symetriemi*, které redukuje počet koeficientů na 36. Důležitou vlastností je i *velká symetrie* plynoucí ze závislosti pro deformační energii. Díky ní lze zaměňovat i dvojice indexu  $ij$  a  $kl$ . Velká symetrie redukuje počet koeficientů na 21.



Pro izotropní tenzor  $C_{ijkl}$  lze psát

$$C_{ijkl} = \alpha\delta_{ij}\delta_{kl} + \beta\delta_{ij}\delta_{kl} + \gamma\delta_{ij}\delta_{kl} \quad (4.2)$$

čímž po dosazení vznikne

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl} = (\alpha\delta_{ij}\delta_{kl} + \beta\delta_{ij}\delta_{kl} + \gamma\delta_{ij}\delta_{kl})\epsilon_{kl} = \alpha\delta_{ij}\epsilon_{kk} + \beta\delta_{ik}\epsilon_{kj} + \gamma\delta_{il}\epsilon_{jl} \quad (4.3)$$

je známo, že tenzor deformace  $\epsilon$  je symetrický, tj.

$$\epsilon_{ij} = \epsilon_{ji} \quad (4.4)$$

z čehož vzejde

$$\sigma_{ij} = \alpha\delta_{ij}\epsilon_{kk} + (\beta + \gamma)\epsilon_{ij} \quad (4.5)$$

a to lze upravit do tvaru

$$\sigma_{ij} = \lambda\delta_{ij}\epsilon_{kk} + (2\mu)\epsilon_{ij} \quad (4.6)$$

Tímto postupem byl získán zobecněný Hookeův zákon pro lineární, izotropní a elastické těleso stanovený Cauchym v roce 1822.  $\lambda, \mu$  jsou tzv. Lamého konstanty. v maticové formě je tvar zobecněného Hookeova zákona

$$[\sigma] = \lambda(\text{tr}[\epsilon])[I] + 2\mu[\epsilon] \quad (4.7)$$

Pro následující kapitoly bude rovněž potřeba znát i inverzní vztah Hookeova zákona, tedy že  $\epsilon_{ij} = \epsilon_{ij}(\sigma_{ij})$ . Hookeův zákon bude zúžen v indexech  $i$  a  $j$ .

$$\sigma_{kk} = \lambda\delta_{kk}\epsilon_{kk} + 2\mu\epsilon_{kk} = (3\lambda + 2\mu)\epsilon_{kk} \quad (4.8)$$

$$\sigma_{ij} = \frac{\lambda\delta_{ij}\sigma_{kk}}{(3\lambda + 2\mu)} + 2\mu\epsilon_{ij} \quad (4.9)$$

Vyjádřením  $\epsilon_{ij}$  vznikne tvar Hookeova zákona:

$$\epsilon_{ij} = \frac{1}{2\mu} \left[ \sigma_{ij} - \frac{\lambda\delta_{ij}\sigma_{kk}}{(3\lambda + 2\mu)} \right] \quad (4.10)$$

Pro materiály, které jsou obecně označovány jako *hookeovské*, existuje pět materiálových konstant -  $\lambda, \mu, E, \nu, k$ . Z nich jsou vždy pouze dvě nezávislé. Velice často bývá využívána verze Hookeova zákona vyjádřená pomocí  $E$  a  $\nu$ . Tento tvar bude použit i v následujících statích.

## 4.1.2 Úloha rovinné napjatosti

Typickou úlohou rovinné napjatosti jsou např. tenké desky nebo úlohy napjatosti v tenkostěnné válcové nádobě. Předpokladem této úlohy je nulová složka napětí ve třetí ose, tedy

$$\sigma_{zz} = \tau_{xz} = \tau_{yz} = 0 \quad (4.11)$$

Vztah pro tenzor napětí tak přechází do tvaru

$$\begin{bmatrix} \sigma_{xx} & \tau_{xy} \\ \tau_{yx} & \sigma_{yy} \end{bmatrix} \quad (4.12)$$

$$\sigma_{ij} = \frac{E}{(1 + \nu)} \left[ \epsilon_{ij} + \frac{\nu}{1 + 2\nu} (\delta_{ij} \epsilon_{kk}) \right] \quad (4.13)$$

$$\epsilon_{ij} = \frac{1}{E} \left[ (1 + \nu) \sigma_{ij} - \nu (\delta_{ij} \sigma_{kk}) \right] \quad (4.14)$$

Složka  $\sigma_z = 0$ . Rovnice pro  $\epsilon_z$  tak přejde do tvaru

$$\epsilon_z = \frac{1}{E} [-\nu(\sigma_x + \sigma_y)] \quad (4.15)$$

deformace v  $x$  a  $y$  mají tvar

$$\epsilon_x = \frac{1}{E} [\sigma_x - \nu(\sigma_x + \sigma_y)] \quad (4.16)$$

$$\epsilon_y = \frac{1}{E} [\sigma_y - \nu(\sigma_x + \sigma_y)] \quad (4.17)$$

Kompletní tenzor deformace pro stav rovinné napjatosti

$$\begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} & 0 \\ \epsilon_{yx} & \epsilon_{yy} & 0 \\ 0 & 0 & \epsilon_{zz} \end{bmatrix} \quad (4.18)$$

### 4.1.3 Úloha rovinné deformace

Jak již vyplývá z názvu této stati, jedná se o takový stav zatěžování, kdy je velikost deformace v jednom ze směrů zanedbatelná a je tudíž funkcí pouze dvou souřadnic - v tomto případě souřadnic  $x$  a  $y$ . Tenzor deformace přejde do tvaru:

$$\begin{bmatrix} \epsilon_{xx} & \epsilon_{xy} \\ \epsilon_{yx} & \epsilon_{yy} \end{bmatrix} \quad (4.19)$$

Tento tenzor lze opět dosadit do Hookeova zákona. Složky deformace ve směru  $z$  jsou rovny nule, vztah pro  $\sigma_z$  je roven:

$$\sigma_z = \frac{E}{(1 + \nu)(1 - 2\nu)} [\epsilon_x + \epsilon_y] \quad (4.20)$$

Dále

$$\sigma_x = \frac{E}{(1 + \nu)} \left[ \epsilon_x + \frac{\nu}{1 - 2\nu} (\epsilon_x + \epsilon_y) \right] \quad (4.21)$$

$$\sigma_y = \frac{E}{(1 + \nu)} \left[ \epsilon_y + \frac{\nu}{1 - 2\nu} (\epsilon_x + \epsilon_y) \right] \quad (4.22)$$

$$\tau_y = \frac{E}{(1 + \nu)} \epsilon_{xy} \quad (4.23)$$

## 4.2 Základy matematiky pro mechaniku

### 4.2.1 Tenzory

Samostatnou kapitolou matematiky pro mechaniku je tenzorový počet. Vzhledem k problematice, která bude probírána v následujících kapitolách, je vhodné se o tensorech alespoň ve zkratce zmínit.

Během bádání v oboru mechaniky bylo vypořádáno, že velikost složek napětí je závislá na směru, ve kterém se na napjatost díváme. Současně je také možné napjatost promítnout do směru jiného. Při použití náležitých transformačních pravidel se nedopustíme chyby. Napjatost v libovolném bodě kontinua má obecně devět složek. Vlivem *zákona o sdružených smykových napětích* je ale nezávislých složek obecně šest. Připomeňme tedy tvar obecného tenzoru napětí

$$\begin{bmatrix} \sigma_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \sigma_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \sigma_{33} \end{bmatrix} \quad (4.24)$$

Ten je symetrický, hodnoty mimodiagonálních složek jsou tedy:  $\tau_{12} = \tau_{21}$ ,  $\tau_{13} = \tau_{31}$  a  $\tau_{23} = \tau_{32}$ . Je ovšem důležité vědět, že významem i fyzikální podstatou jsou všechny složky naprosto rozdílné, ale číselné hodnoty složek s prohozenými indexy jsou stejné. Lze tudíž použít uvedený zápis.

Pokud chceme na tenzor napětí pohlédnout z jiného směru, je třeba ho transformovat. k tomu slouží tzv. transformační matice, většinou značená jako  $\mathbf{A}$ . V podstatě jde o *matici směrových kosinů* úhlů, o které je v prostoru natočen nový souřadný systém vůči tomu původnímu.

$$[A] : x \rightarrow x' \quad (4.25)$$

Základní transformační pravidlo má pak tvar

$$t' = [A]t, \quad t'_i = A_{ij}v_j \quad (4.26)$$

$$t = [A]^{-1}t', \quad t_i = A_{ji}v'_j \quad (4.27)$$

Vztahy 4.26 a 4.27 jsou transformční pravidla definující tenzor 1. řádu. Rozšířenější označení pro takové veličiny je *vektor*. Tenzor napjatosti je soustava vektorů (*tenzorů prvního řádu*) napětí a jedná se tak o tenzor druhého řádu. V praxi je zavedeno toto označení:

- Tenzor 0. řádu - skalár
- Tenzor 1. řádu - vektor
- Tenzor 2. řádu - tenzor

Pro tenzory vyšších řádů je užíván explicitně výraz *tenzor n-tého řádu*.<sup>1</sup>

---

<sup>1</sup>v inženýrské praxi nebývají tenzory vyšších řádů hojně využívány. Kromě prvního řádu pak má navíc smysl používat pouze tenzory sudých řádů.

## 4.2.2 Deviatorická složka napětí

Slovo *deviace* znamená v překladu odchylku od cesty, resp. nějakého standartu. V mechanice poddajných těles má toto slovo význam odchylky od hydrostatické napjatosti. Mějme kvádr umístěný do kartézského souřadného systému. Pokud je tento kvádr zatížen pouze tlakem ve směru  $y$ , bude zřejmě jediná složka tenzoru napjatosti nenulová -  $\sigma_{22} = -p$ . Tento tenzor zapíšeme jako součet dvou částí:

$$\sigma = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -p & 0 \\ 0 & 0 & 0 \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{p}{3} & 0 & 0 \\ 0 & -\frac{2p}{3} & 0 \\ 0 & 0 & \frac{p}{3} \end{bmatrix}}_{\text{deviatorická složka}} + \underbrace{\begin{bmatrix} -\frac{p}{3} & 0 & 0 \\ 0 & -\frac{p}{3} & 0 \\ 0 & 0 & -\frac{p}{3} \end{bmatrix}}_{\text{sférická složka}} \quad (4.28)$$

Symbolicky lze zapsat tuto formulaci do tvaru

$$\sigma_{ij} = S_{ij} - \frac{1}{3}\delta_{ij}tr(\sigma) \quad (4.29)$$

Tenzor napětí je rozdělen na deviatorickou složku a sférickou složku napětí:  $S_{ij}$  a  $\frac{1}{3}\delta_{ij}tr(\sigma)$ .

## 4.2.3 Invarianty

Invariant je tenzorový výraz, na který nemá vliv transformace mezi souřadnými systémy. Invarianty tenzoru napětí jsou označovány  $I$ . Z výše uvedeného musí platit:

$$I = f(\sigma_{ij}) = f(\sigma'_{ij}) \quad (4.30)$$

Tenzory mají tři hlavní invarianty. Lineární

$$I_1 = tr(\sigma) \quad (4.31)$$

kvadratický

$$I_2 = det \begin{vmatrix} \sigma_{11} & \tau_{12} \\ \tau_{21} & \sigma_{22} \end{vmatrix} + det \begin{vmatrix} \sigma_{22} & \tau_{23} \\ \tau_{32} & \sigma_{33} \end{vmatrix} + det \begin{vmatrix} \sigma_{11} & \tau_{13} \\ \tau_{31} & \sigma_{33} \end{vmatrix} \quad (4.32)$$

a kubický

$$I_3 = det |\sigma| \quad (4.33)$$

Invarianty deviatoru napětí mají tvar:

$$\begin{aligned} J_1 &= tr(S) = 0 \\ J_2 &= \frac{1}{2} S_{ij}S_{ji} \\ J_3 &= \frac{1}{3} S_{ij}S_{jk}S_{kl} \end{aligned} \quad (4.34)$$

První invariant deviatoru napětí je vždy nulový.

## 4.3 Základy optimalizace, přehled některých optimalizačních úloh

Úloha optimalizace je nejen v moderním strojírenství důležitým oborem, díky kterému lze vhodně volit parametry syntézy technického problému. Velmi typickým odvětvím je problematika návrhu mechanismů a robotů, od kterých je vyžadována např. optimální ovladatelnost ve vymezeném pracovním prostoru, minimalizace zástavbového prostoru atd. Dalších odvětví s podobnými požadavky lze ale najít celou řadu, zdaleka se nemusí jednat jen o mechanismy. Koneckonců i ve zde řešené úloze ladění materiálového modelu porušení bude optimalizace využito.

Existuje množství optimalizačních metod. Jsou děleny do dvou hlavních skupin: lokální a globální. Jak již toto označení napovídá, lokální metody nejsou, na rozdíl od globálních, schopny rozpoznat, zda se jedná o extrém funkce lokální nebo globální. Mezi takové metody jsou řazeny např. metoda *simplexová* nebo *Rosenbrockova metoda*.

Globální metody využívají k optimalizaci prvek pravděpodobnosti. Jejich algoritmy jsou často založeny na jevech pozorovaných v přírodě. Mezi nejznámější patří tzv. *Metoda simulovaného žihání*, *genetické algoritmy*, *metoda ptačího hejna*.

### 4.3.1 Rosenbrockova metoda

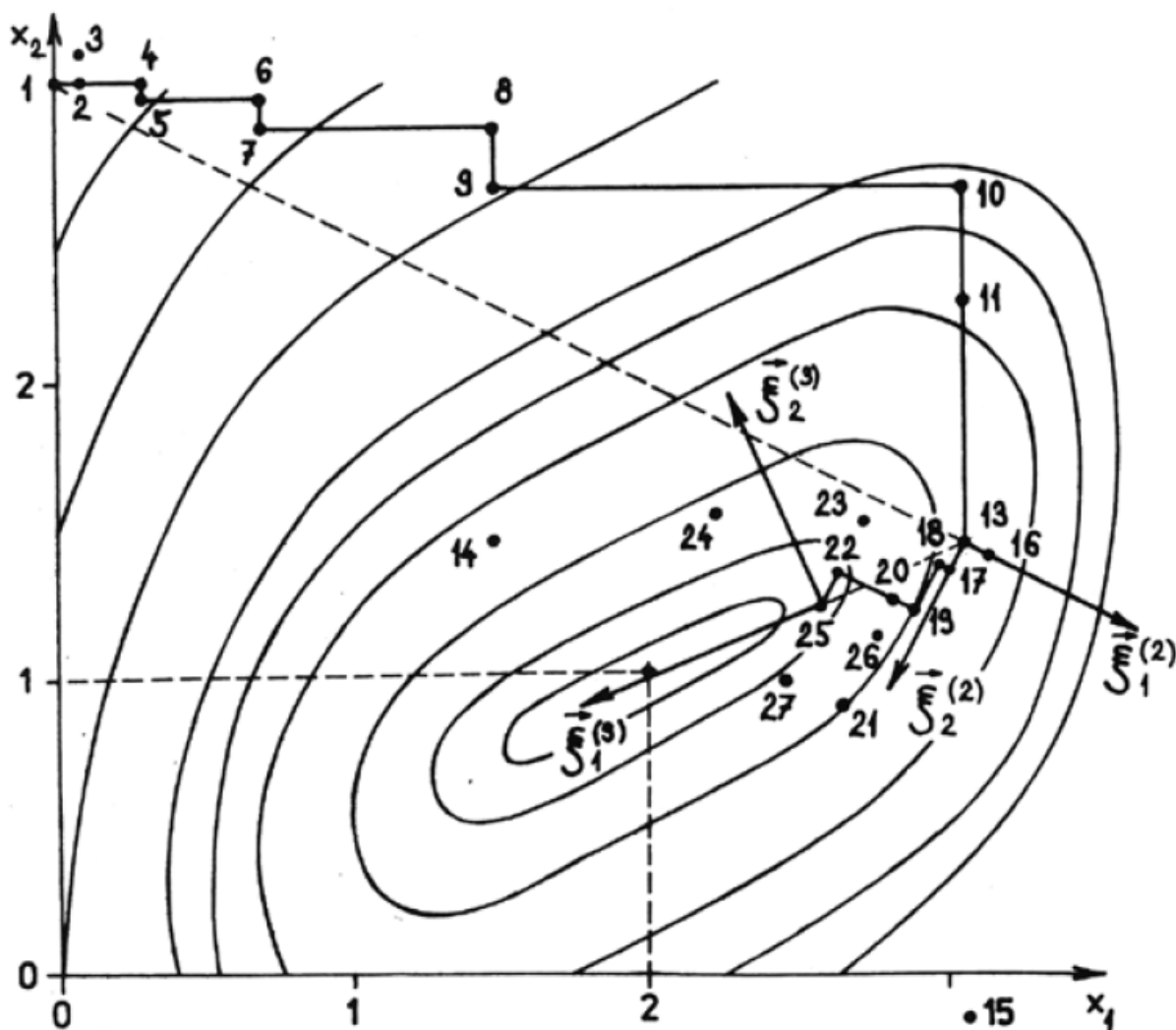
Rosenbrockova metoda je lokální negradientní metoda, jejíž algoritmus sestává z postupných jednosměrných iterací ve směru poklesu minimalizované funkce. Kroky algoritmu jsou následující:

1. Je určen počáteční bod optimalizace - počáteční odhad. Dále je zavedeno  $n$  lineárně nezávislých směrů  $\vec{\xi}_1^k$ , kde  $k$  je číslo zkoumaného bodu funkce:  $\vec{\xi}_1^{(1)} = [1, 0, \dots, 0]^T$ , až po  $\vec{\xi}_1^{(2)} = [0, 0, \dots, 1]^T$ .
2. Je zjištěna funkční hodnota v bodě počátečního odhadu, který je prozatím nejpříznivější. Ve směru  $\vec{\xi}_1^1$  je učiněn krok o hodnotu kroku  $\alpha \cdot s$ .
3. Pokud je funkční hodnota po učinění kroku příznivější, je tento bod uložen jako nový nejpříznivější. Pokud ne, je učiněn krok o hodnotě  $-\beta \cdot s$ . To samé je poté prováděno ve zbylých dimenzích.
4. Následně je celý postup opakován. Lokální souřadnicový systém je ale natočen tak, aby další krok směřoval ve směru efektivního postupu minulé iterace.

Graficky je Rosenbrockova metoda znázorněna na obr. 4.1

## 4.4 Modely tvárného porušení

Procesy lomu lze rozdělit do dvou skupin, a to lom na křehký a tvárný, přičemž pro účely této práce je klíčový druhý jmenovaný. k tvárnému lomu součásti dochází po vyčerpání plastické zásoby materiálu. Je přitom závislý na lokální napjatosti, teplotě, rychlosti deformace atd. Je tedy zřejmé, že při přesném navrhování tvářecího procesu nestačí základní pevnostní hypotézy a materiál je tak nutno kalibrovat tak, aby predikce porušení materiálu byla co nejpřesnější.



Obrázek 4.1: Ukázka Rosenbrockovy metody pro případ optimalizace dvouparametrické funkce [3]

Kalibrace je prováděna například porovnáním dat experimentu a výpočtu MKP. Na základě sledování hodnot napěťové triaxiality a Lodeho parametru lze zjistit parametry funkce poškození.

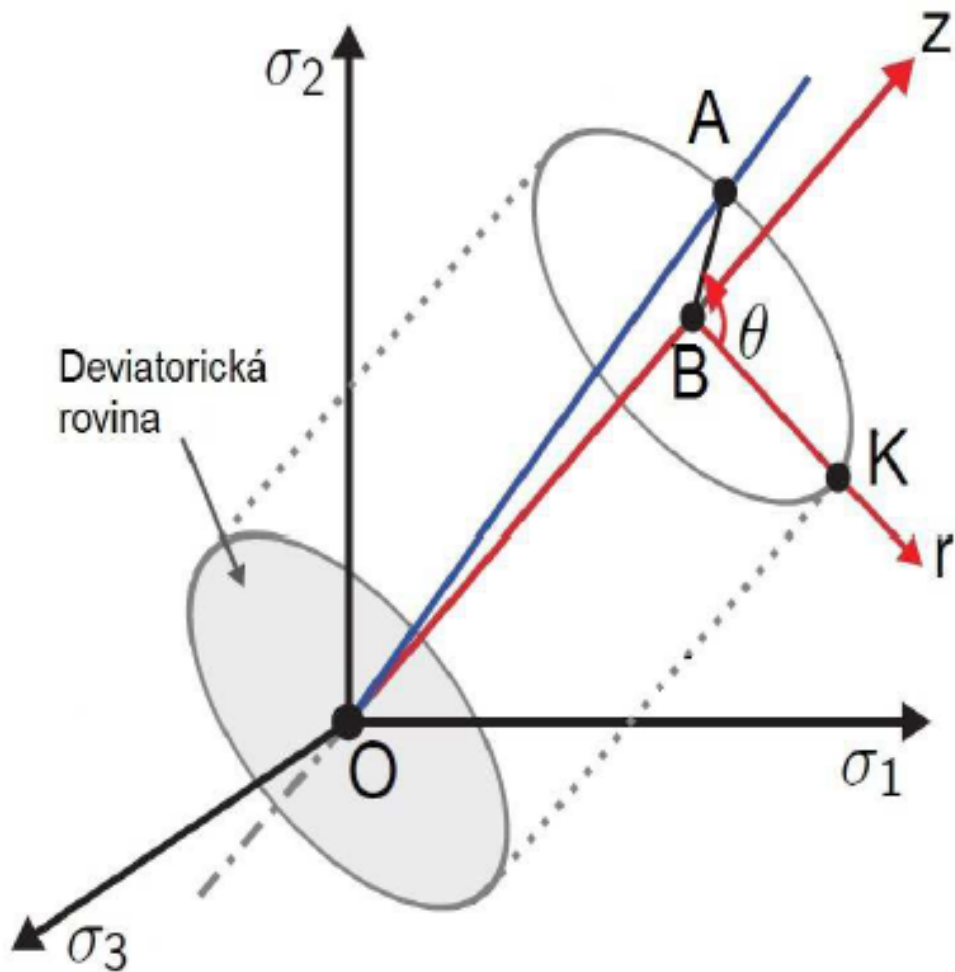
#### 4.4.1 Triaxialita a Lodeho úhel

Jak bylo naznačeno v předchozím odstavci, v problematice ladění modelů tvárného porušení hrají klíčovou roli dvě veličiny: *Triaxialita* a *Lodeho úhel*

*Triaxialita* je podíl hydrostatického napětí  $\sigma_H$  a Misesova napětí  $\sigma_{eq}$ . Vztah pro její výpočet tak vypadá

$$\eta = \frac{\sigma_H}{\sigma_{eq}} = \frac{\frac{1}{3}(\sigma_1 + \sigma_2 + \sigma_3)}{\frac{1}{\sqrt{2}}\sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}} \quad (4.35)$$

Tato veličina zohledňuje vliv hydrostatické napjatosti na poškození materiálu.



Obrázek 4.2: Haigh-Westgaardův prostor hlavních napětí [4]

Dále mějme *Haigh-Westgaardův* prostor hlavních napětí. Osa prvního oktantu splývá s vektorem hydrostatické napjatosti. Uvažujme rovinu, která je kolmá na vektor hydrostatické napjatosti a zároveň prochází počátkem. Tato rovina nese název *Deviatorická rovina*. (O deviatoru napětí bylo napsáno v kapitole 4.2.2.)

Z rovnice 4.29 lze určit hodnoty složek deviatoru napětí. Je zvykem popisovat stav napjatosti materiálu v prostoru hlavních napětí pomocí cylindrického systému  $(r, \Theta, z)$ , kde souřadnice  $z$  splývá s vektorem hydrostatické napjatosti. Z kapitoly 4.2.2 je již známo, že celková napjatost je součtem *hydrostatické* a *deviatorické* složky. To je reprezentováno v prostoru hlavních napětí výslednicí  $|OA|$ . Velikost deviatorické složky přitom odpovídá velikosti ekvivalentního von Misesova napětí,

$$\sigma_{eq} = \frac{1}{\sqrt{2}} \sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2} \quad (4.36)$$

přes něž se lze z výše uvedených vztahů dostat až do tvaru

$$\sigma_{eq} = \sqrt{\frac{3}{2}S_{ij}S_{ij}} = \sqrt{3J_2}. \quad (4.37)$$

Vztah pro Lodeho úhel má tvar

$$\Theta = \frac{1}{3} \arccos \left( \frac{3\sqrt{3}J_3}{2\sqrt{J_2^3}} \right) \quad (4.38)$$

Později bude v této práci využit model popisu lomové plochy *Bai Wierzbicki*, který pracuje s normalizovanou hodnotou *Lodeho úhlu*

$$\bar{\Theta} = 1 - \frac{6\Theta}{\pi}, \quad \bar{\Theta} \in \langle -1, 1 \rangle \quad (4.39)$$

Tato kapitola uvedla základní matematické nástroje využívané v mechanice poddajných těles. Ty byly následně aplikovány v této kapitole pro určení *Triaxiality* a *Lodeho úhlu* - dvou stěžejních veličin sloužících k určení lomové plochy. Poznatky doposud získané budou nyní aplikovány na sestavení lomové plochy zkoumaného materiálu.

## 4.5 Filozofie mechaniky poškození

Před přikročením k popisu konkrétních modelů porušení bude ještě jednou stručně shrnuta filozofie, na které jsou tyto modely založené. Byly zavedeny veličiny *triaxiality* a *Lodeho úhlu*. Další důležitou veličinou je ale i hodnota  $\epsilon_f$  - akumulovaná plastická deformace při porušení za předpokladu, že se během plastizace nemění parametry lomové funkce (*Lodeho úhel* a *triaxialita*). Vztah 4.40, který bude použit v kapitole 4.6, představuje lineární kumulační zákon zohledňující nekonstantní průběh hodnoty triaxiality. Pochopitelně platí i pro případ konstantního průběhu triaxiality, kdy porušení nastane při dosažení hodnoty  $\epsilon_f$ . Obecně tedy lze konstatovat, že lomové funkce zohledňují nejen vliv velikosti napětí na porušení součásti, ale i způsob zatěžování.

## 4.6 Kritérium porušení Rice-Tracey

Na začátku historie řešení problematiky modelů tvárného porušení modely pracovaly s představou růstu dutin ve struktuře ideálně plastického materiálu. s modelem pro 2D kontinuum pracoval McClintock. [5]. Na jeho práci dále navazovali při práci Rice a Tracey, kteří tento přístup rozšířili pro kontinuum ve 3D. Oba přístupy se shodovaly v konstatování, že na růst trhlin ve struktuře materiálu má vliv triaxialita napětí. (O triaxialitě napětí bylo napsáno v kapitole 4.4.1). Lomová funkce má tvar  $\bar{\epsilon}_f(\eta)$

$$\bar{\epsilon}_f(\eta) = C e^{-\frac{3}{2}\eta}, \quad D = \int \frac{1}{\bar{\epsilon}_f(\eta)} d\bar{\epsilon}_{pl} = \frac{1}{C} \int e^{\frac{3}{2}\eta} d\bar{\epsilon}_{pl} \quad (4.40)$$

Tato funkce má jediný materiálový parametr  $C$ . Ten lze zjistit z jediného experimentu. Přesto, že se jedná o poměrně jednoduchý předpis, je kritérium Rice-Traceyho dobře využitelné pro celou řadu technických aplikací. Navíc jsou z tohoto předpisu odvozeny další komplikovanější modely porušení.



## 4.7 Model porušení Bai-Wierzbicki

Jedním z modelů stávicích na základě modelu Rice-Tracey a jeho matematického předpisu 4.40 je právě model Bai-Wierzbicki. Ten byl použit i pro aplikaci v MKP výpočtech v této práci. Zobecněné kritérium Rice-Tracey bylo použito ve významných bodech normalizovaného Lodeho úhlu:

případ jednoosého tahu

$$\bar{\epsilon}_f(\eta, 1) = C_1 e^{-C_2 \eta} \quad (4.41)$$

případ rovinné deformace

$$\bar{\epsilon}_f(\eta, 0) = C_3 e^{-C_4 \eta} \quad (4.42)$$

případ jednoosého tlaku

$$\bar{\epsilon}_f(\eta, -1) = C_5 e^{-C_6 \eta} \quad (4.43)$$

Deformace jakožto funkce triaxiality a Lodeho úhlu  $\bar{\epsilon}_f(\eta, \bar{\Theta})$  je poté předepsána jako parabolická funkce.

$$\bar{\epsilon}_f(\eta, \bar{\Theta}) = F_1(\eta) \bar{\Theta}^2 + F_2(\eta) \bar{\Theta} + F_3(\eta) \quad (4.44)$$

$F_{1-3}$  jsou funkce triaxiality napětí a jejich tvar je odvozen dosazením podmínek 4.41 až 4.43.

$$\begin{aligned} F_1(\eta) &= \frac{1}{2} (C_1 e^{-C_2 \eta} + C_5 e^{-C_6 \eta}) - C_3 e^{-C_4 \eta} \\ F_2(\eta) &= \frac{1}{2} (C_1 e^{-C_2 \eta} + C_5 e^{-C_6 \eta}) \\ F_3(\eta) &= -C_3 e^{-C_4 \eta} \end{aligned}$$

Tím je vyjádřena funkce  $\bar{\epsilon}_f(\eta, \bar{\Theta})$ . Nyní bude zavedena bezrozměrná veličina  $D$  - parametr poškození<sup>2</sup>. Jeho hodnotu bude v tomto případě udávat poměr přírůstku plastické deformace a

$$D = \int_0^{\epsilon_f^*} \frac{d\bar{\epsilon}_{pl}}{\bar{\epsilon}_f(\eta, \bar{\Theta})} \quad (4.45)$$

Plocha poškození je sestavena z bodů, které jsou charakterizovány následovně:

1. Jde o stav zatížení, kdy během experimentu dosáhla síla svého maxima
2. Okamžik, kdy došlo k maximální kumulaci poškození. Je přitom nutné správně určit místo na geometrii, kde začne docházet k porušení materiálu. V tomto místě je lomová funkce kalibrována

Pokud není známo, v kterém místě má k porušení dojít, je třeba určit více podezřelých bodů. Oblast iniciace poškození pak lze určit z experimentu. Určení místa iniciace trhliny vyžaduje individuální přístup pro každý vzorek nebo součást.

---

<sup>2</sup>Tento parametr byl již použit v kapitole 2

### 4.7.1 Redukce počtu parametrů modelu Bai-Wierzbicki

Zopakujme, že zjištění každého parametru  $C_{1-6}$  vyžaduje jeden vzorek, dohromady tedy šest zkušebních vzorků. Přitom platí požadavek na rozdílný průběh stavu napjatosti. To činí formulaci tohoto modelu pro konkrétní materiál poměrně nákladnou. Počet laděných parametrů  $C$  lze ale redukovat. Model Bai-Wierzbicki je obecně nesymetrický. Řada materiálů ovšem poskytuje symetrickou lomovou plochu. Symetrii matematické formulace modelu Bai-Wierzbicki lze zavést předpisem  $C_5 = C_1$ ,  $C_6 = C_2$ . Tak dojde ke snížení počtu parametrů na čtyři. Dalšího zjednodušení lze dosáhnout založením kritéria na modelu Rice-Tracey (viz 4.40). To je splněno předpisem  $C_2 = C_4 = \frac{3}{2}$ . Došlo tak k redukci kritéria ze šesti na dva parametry. To znamená potřebu pouze dvou zkušebních vzorků a zjednodušení procesu optimalizace parametrů. (Teorie procesu optimalizace je v kapitole 4.3, samotný proces optimalizace parametrů  $C$  je popsán v kapitole 5)

# Kapitola 5

## Kalibrace modelu Bai-Wierzbicki pro materiál DC01 pomocí Nakazima experimentu a tahové zkoušky

Modely resp. plochy porušení mají řadu parametrů, pro jejichž zjištění je potřeba minimálně stejná řada experimentů na různých vzorcích (viz kapitola 4.4). V této práci je použit popis lomové plochy *Bai-Wierzbicki*. Následující odstavce považujeme za jeden z možných návodů na zjištění hledaných parametrů funkce poškození. Experimentální data byla dodána firmou Comtes FHT.

V kapitole 3 byl popsán přístup k úloze pomocí řešiče MKP. Byly zde nastíněny různé možnosti přístupu k úloze. Vzhledem k porušení symetrie vzorku během degradace materiálu je nutné použití úlohy s plnou geometrií modelu. Na tomto vzorku byla optimalizována síť a spuštěná MKP analýza. Stejně tak byla analýza provedena pro tahovou zkoušku plochého vzorku. Z těchto analýz byly pomocí *CLI-Command Line Interface*, poměrně silného nástroje výpočtového softwaru, extrahovány následující veličiny:

- Reakční síly
- Ekvivalentní plastické deformace
- Misesova ekvivalentního napětí  $\sigma_{eq}$
- Hlavních napětí a složek napětí  $\bar{\sigma}$

Hodnoty veličin jsou vynášeny v závislosti na inkrementálním (výpočtovém) čase. Z nich se již lze přes úpravy a dosazení do dříve uvedených vzorců dostat k veličinám potřebným pro zjištění aktuální míry poškození. Z výsledků simulace byly v uzlech, ve kterých lze z pozorování experimentu očekávat porušení materiálu, vypsány hodnoty napěťové triaxiality, efektivní plastická deformace a síla na razníku.

Celý proces je rozdělen pro přehlednost do dvou souborů. První skript, *BW.py*, má za úkol třídit výsledky uložené do tabulkového souboru *Results.xlsx*. Jeho příklad je na obr. 5.1. Druhým souborem je *Pole.py*. Ten již pracuje s vytríděnými daty. Z nich počítá hodnoty triaxiality a Lodeho parametru a dosazuje je do předpisu plochy porušení tak, jak ho navrhli Bai-Wierzbicki. Ze zjištěného inkrementálního času, ve kterém má dojít k porušení, je určena jed-

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	PEEQ	Max Princ	Mid Princ	Min Princ	S11	S22	S33	S12	S13	S23	Mises	Time	RF
2	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.
3	0.	-0.0002753	-0.03047	-0.03553	-0.03507	-0.00029	-0.03091	0.0006688	-0.00143	0.0002092	0.03302	0.025	0.01472
4	0.	-0.0007817	-0.1135	-0.1282	-0.1269	-0.0008122	-0.1148	0.001821	-0.004153	0.0005697	0.1208	0.05	0.0409
5	0.	-0.001755	-0.3006	-0.334	-0.331	-0.001812	-0.3035	0.004018	-0.009447	0.001257	0.3169	0.075	0.09099
6	0.	-0.002533	-0.6932	-0.7437	-0.7393	-0.002583	-0.6977	0.005512	-0.01428	0.001724	0.7173	0.1125	0.1284
7	0.	-0.002086	-1.192	-1.239	-1.235	-0.002105	-1.196	0.004342	-0.01335	0.001359	1.214	0.1688	0.1064
8	0.	-0.0009076	-1.693	-1.729	-1.726	-0.0009133	-1.697	0.002443	-0.009962	0.0007652	1.71	0.2531	0.06534
9	0.	0.0005016	-2.268	-2.287	-2.285	0.0004969	-2.27	0.00453	-0.005295	0.0008112	2.278	0.3797	0.04321
10	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	0.00419	NoValue	0.001905	NoValue	0.3797	NoValue
11	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	-0.0001227	NoValue	-0.0006001	NoValue	0.3797	NoValue
12	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	-0.0004434	NoValue	0.0004328	NoValue	0.3797	NoValue
13	0.	0.001503	-2.97	-3.004	-2.972	0.001495	-3.002	0.006342	0.009016	0.0001105	2.99	0.5695	0.04457
81	0.001197	-2.679	-37.14	-205.1	-51.88	-2.832	-190.2	0.7906	47.53	0.2455	187.6	1.029	64.85
82	0.001765	-2.987	-24.18	-201.	-39.59	-3.278	-185.3	1.571	50.08	0.4878	188.3	1.033	78.31
83	0.001951	-2.963	-19.72	-199.4	-35.26	-3.393	-183.4	1.925	50.91	0.5978	188.6	1.035	83.92
84	0.002215	-2.446	-12.77	-196.3	-28.17	-3.424	-180.	2.605	52.16	0.809	189.	1.037	105.5
85	0.002301	-2.007	-10.95	-195.4	-26.12	-3.335	-178.9	2.832	52.5	0.8794	189.1	1.038	96.44
86	0.002406	-1.091	-8.942	-194.1	-23.47	-3.208	-177.5	3.164	52.92	0.9826	189.2	1.039	112.4
87	0.002577	2.583	-6.156	-191.1	-17.43	-3.098	-174.1	3.847	53.85	1.195	189.5	1.041	114.9
88	0.002722	7.974	-4.793	-187.8	-11.5	-2.572	-170.5	4.678	54.64	1.453	189.7	1.043	125.2
89	0.002737	17.47	-0.9391	-183.1	-2.427	1.138	-165.3	6.03	55.96	-6.157	192.1	1.047	149.8
90	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	10.85	NoValue	1.047	NoValue
91	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	-7.274	NoValue	1.047	NoValue
92	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	NoValue	10.07	NoValue	1.047	NoValue
93	0.002737	20.54	0.4041	-177.3	0.8325	2.479	-159.7	6.412	55.16	-6.445	188.7	1.049	159.6

Obrázek 5.1: Příklad sady výsledků získané z programu Abaqus

notná veličina komparace experimentu a simulace, to jest deformace, kterou jsou vzorky zatěžovány. Dále je předepsána cílová funkce ve tvaru:

$$F = \frac{1}{N} \sum_{i=0}^N (y_i - y_i^*)^2 \quad (5.1)$$

Její parametry jsou přitom parametry  $C_i$ . Nalezením minima cílové funkce, tedy minimalizací rozdílu mezi simulací a skutečností, jsou zjištěny parametry C. Funkce Bai - Wierzbicki je následně implementována jako materiálová vlastnost do modelu v MKP. Optimalizátor funkce je rovněž součástí skriptu *pole.py*.

Kauzalita procesu je taková, že *pole.py* cyklicky spouští *BW.py*, přičemž mu předá proměnnou  $l$ . Ta určuje, z kolikátého listu se mají importovat hodnoty. Další postup algoritmu by měl objasnit *step-by-step* popis obou skriptů.

Sledujme nyní obrázek 5.1. Již bylo zmíněno, že řídicí veličinou tvárného porušení je *Ekvivalentní plastická deformace*. Její hodnoty v průběhu zatěžování jsou zaznamenány ve sloupci A resp. *PEEQ*. Postupem zleva doprava jsou uloženy hodnoty hlavních napětí,  $\sigma_{xx}$ ,  $\sigma_{yy}$ ,  $\sigma_{zz}$  a smykových napětí, ekvivalentního Misesova napětí, inkrementálního času a reakční síly<sup>1</sup>.

## 5.1 Třídění dat získaných z výpočtového softwaru

Jako na začátku každého skriptu je vidět import důležitých knihoven potřebných pro práci s různými typy objektů (viz obr. 5.2). Za povšimnutí stojí zejména knihovna *xlrd*, která umožňuje pracovat s tabulkovým souborem. Ačkoliv Python dokáže velice dobře importovat data i z textových souborů, je pro některé potřebné mezikroky výhodnější použít právě tabulkový formát, jako tomu bylo v tomto případě.

<sup>1</sup>Myšlena je reakční síla na razníku v případě Nakazima testů. Pro kalibraci modelu její hodnota není zajímavá, důležitá je ale pro porovnání experimentu a simulace. Je proto zaznamenávána pro pozdější potřebu

```

1
2 import xlrd
3 import matplotlib.pyplot as plt
4 import scipy as sp
5 from scipy.optimize import fmin
6 from matplotlib import cm
7 import mpl_toolkits.mplot3d.axes3d as axes3d
8 from scipy import interpolate
9 import matplotlib.font_manager
10 import numpy as np
11 import scipy.optimize as optimize
12
13
14
15 SESIT=xlrd.open_workbook('Results.xlsx')
16 LIST=SEKIT.sheet_by_index(1)
17 print "SEKIT", l
18
19
20 PEEQes=[] #Pole hodnot plasticke deformace
21 MAXes=[] #...Sigma1 hlavnich napeti
22 MIDes=[] #...Sigma2
23 MINes=[] #...Sigma3
24 MISESes=[] #...Misesova napeti
25 Sigma11s=[] #... A tak dale
26 Sigma22s=[]
27 Sigma33s=[]
28 Sigma12s=[]
29 Sigma13s=[]
30 Sigma23s=[]
31 TIMEs=[]

```

Obrázek 5.2: Import knihoven potřebných pro práci s *xlsx* soubory a vytvoření polí, do kterých budou načítána data

Proměnná *SEKIT* určuje, z jakého souboru se mají hodnoty zpracovávat. Prostřednictvím tečkové notace by samozřejmě bylo možno pracovat s daty z libovolného množství souborů. Náš problém si vystačil s jedním souborem - *Results.xlsx*. Proměnná *LIST* určuje s jakým konkrétním listem je momentálně pracováno. Na řádcích 20 až 31 jsou vytvořena zatím prázdná pole, do kterých budou následně uloženy příslušné hodnoty. Koncovka *-es* zde funguje stejně jako v angličtině, tedy značí množné číslo. To slouží pro orientaci v proměnných, kdy se jedná o pole a kdy o skalární proměnnou.

Na řádku 34 (obr. 5.3) je spuštěn cyklus přes všechny neprázdné řádky výsledkové tabulky v konkrétním listu. Postupně jsou vytvořeny proměnné, jimž je přiřazena hodnota na aktuál-

```

34 for i in range(0,LIST.nrows-1):
35
36     PEEQ=LIST.cell(i+1,0).value
37     MAX=LIST.cell(i+1,1).value
38     MID=LIST.cell(i+1,2).value
39     MIN=LIST.cell(i+1,3).value
40     Sigma11=LIST.cell(i+1,4).value
41     Sigma22=LIST.cell(i+1,5).value
42     Sigma33=LIST.cell(i+1,6).value
43     Sigma12=LIST.cell(i+1,7).value
44     Sigma13=LIST.cell(i+1,8).value
45     Sigma23=LIST.cell(i+1,9).value
46     MISES=LIST.cell(i+1,10).value
47     TIME=LIST.cell(i+1,11).value
48
49
50
51     if PEEQ=='NoValue' or PEEQ=='0.': #Podminka pro obejiti hlasky NoValue z rpt file
52         continue
53     else:
54         PEEQes.append(float(PEEQ)) #Plneni poli jednotlivych velicin hodnotami
55         MAXes.append(float(MAX))
56         MIDes.append(float(MID))
57         MINes.append(float(MIN))
58         Sigma11s.append(float(Sigma11))
59         Sigma22s.append(float(Sigma22))
60         Sigma33s.append(float(Sigma33))
61         Sigma12s.append(float(Sigma12))
62         Sigma13s.append(float(Sigma13))
63         Sigma23s.append(float(Sigma23))
64         MISESes.append(float(MISES))
65         TIMES.append(float(TIME))

```

Obrázek 5.3: Načítání hodnot do polí proměnných

ním řádku (v aktuálním čase) a příslušném sloupci. Tyto proměnné jsou skalární. Na řádku 51 je vytvořen mechanismus pro ošetření případu, kdy byla ve výsledkovém souboru vypsána hodnota "NoValue", případně kdy je plastická deformace nulová. Pokud se v aktuální buňce plastické deformace objeví hodnota "0", nebo "NoValue", je tento řádek přeskočen. Pokud je hodnota jiná, je připsána na konec příslušného pole proměnných.

```

77 for j in range(0, len(PEEQes)):
78     MAX=MAXes[j]
79     MID=MIDes[j]
80     MIN=MINes[j]
81     MISES=MISESes[j]
82     TRIAX=0.33*(MAX+MID+MIN)/(MISES) #TRIAxIALITA
83     TRIAXes.append(TRIAX) #NACTENI AKTUALNI TRIAXIALITY DO POLE TRIAXIALIT
84
85
86     Sigma11=Sigma11s[j]
87     Sigma22=Sigma22s[j]
88     Sigma33=Sigma33s[j]
89     Sigma12=Sigma12s[j]
90     Sigma13=Sigma13s[j]
91     Sigma23=Sigma23s[j]
92

```

Obrázek 5.4: Výpočet hodnoty triaxiality a zadefinování složek tenzoru napjatosti

Podle vztahu 4.35 je vypočtena hodnota triaxiality v aktuálním inkrementálním čase. Stejně jako v předchozím případě je její hodnota postupně načítána do pole vytvořeného na začátku skriptu. Pro aktuální inkrement jsou také zadefinovány složky tenzoru napjatosti.

Na obr.5.5 je vidět zadefinování tenzoru  $\sigma$ . Jak uvádí vztah 4.39, pro výpočet Lodeho úhlu (a následně Lodeho parametru) je třeba znát druhý a třetí invariant deviátoru napětí. (O deviatorické složce napětí bylo pojednáno v sekci 4.2.2). Deviatorická složka tenzoru napjatosti

```

105 SIGMA=np.array([[Sigma11, Sigma12, Sigma13],
106                [Sigma12, Sigma22, Sigma23],
107                [Sigma13, Sigma23, Sigma33]])
108
109
110
111
112 STOPA=np.trace(SIGMA)
113 I=np.eye(3)
114 DEVIATOR=SIGMA-0.33*STOPA*I
115
116
117 J2=0.167*((Sigma11-Sigma22)**2+(Sigma22-Sigma33)**2+(Sigma33-Sigma11)**2)
118 J3=DEVIATOR[0,0]*DEVIATOR[1,1]*DEVIATOR[2,2]
119
120 citatel=(3*3**(0.5)*J3)
121
122 jmenovatel=(2*(J2**3)**(0.5))
123
124 pomer=citatel/jmenovatel
125
126 if (pomer <= (-1)) and (pomer>-1.1):
127     LODE=np.pi
128
129 elif (pomer >= (1)) and (pomer<1.1):
130     LODE=0
131
132 elif ((-1)<pomer<(1)):
133
134     LODE=0.33*np.arccos((3*3**(0.5)*J3)/(2*(J2**3)**(0.5)))
135
136 else:
137     print "CHYBA! ARGUMENT LODEHO UHLU JE", pomer
138     print "OCEKAVANA HODNOTA V INTERVALU -1, 1"
139     print "CHYBA JE V LISTU", m+1 , "RADEK", j+1
140     continue
141
142 LODE_norm=1-(6*LODE)/(np.pi)
143
144 #print "LODE_norm", LODE_norm
145
146 LODE_normes.append(LODE_norm)
147
148
149
150

```

Obrázek 5.5: Definice matice  $\sigma$  a výpočet Lodeho úhlu

je podle vztahu 4.28 na řádce 114. Druhý invariant deviátoru napětí odpovídající hodnotě napětí HMH a třetí invariant jako součin prvků na diagonále jsou zavedeny jako skalární proměnné J2 a J3. Na řádce 135 začíná mechanismus, který koriguje případnou numerickou odchylku při zpracování výsledků v okolí krajních hodnot definičního oboru funkce  $\arccos()$ , případně dává uživateli vědět, že hodnota Lodeho úhlu nedává z fyzikálního hlediska smysl. Vypsána je daná nevěrohodná varianta a místo, kde k chybě došlo.

Následně je vypočtena hodnota *normalizovaného Lodeho úhlu*, která je připsána jako prozatím poslední hodnota pole zadaného v úvodu skriptu. Cyklus se opakuje pro další řádek v sešitě výsledků. Toto je konec prvního skriptu.

```

5 PEEQp= []
6 TRIAXp= []
7 LODE_normp= []
8 TIMEp= []
9
10 PERIODS= []
11 EXPERIMENTS= []
12
13 for l in range(0,2):
14     execfile ('BW.py')
15
16     PEEQp.append(PEEQes)
17     TRIAXp.append(TRIAXes)
18     LODE_normp.append(LODE_normes)
19     TIMEp.append(TIMEs)
20
21     #Vytvoreni pole celkove casove peridy analyzy
22     RESULTSexp=SESIT.sheet_by_index(-1)
23     Period=RESULTSexp.cell(l+1, 3).value
24     PERIODS.append(float(Period))
25
26     #Vytvoreni pole vyslenych posuvu analyzy
27     Experiment=RESULTSexp.cell(l+1,1).value
28     EXPERIMENTS.append(float(Experiment))
29

```

Obrázek 5.6: Začátek skriptu *pole.py*, zdefinování 2D polí pro načtení hodnot z více experimentů

## 5.2 Sestavení cílové funkce

Na začátku skriptu *Pole.py* (obr.5.6) jsou opět vytvořena prázdná pole. K nim jsou cyklicky připojována pole výsledků získaná ve skriptu *BW.py*. Tato pole budou tudíž dvourozměrná a pro rozlišení jsou označeny koncovkou "p". (Například tedy *LODE\_normp* pro sadu dat *Lodeho* normalizovaného úhlu). Proměnné *PERIODS* a *EXPERIMENTS* budou vysvětleny níže.

Tento skript v cyklu spouští soubor *BW.py*. Získané sady dat pro každou simulaci jsou postupně připojovány k polím s koncovkou "p". Proměnná *EXPERIMENTS* obsahuje hodnoty posuvu, při kterém došlo k porušení vzorku. Zastavme se u proměnné *PERIODS*. Některé simulace lze provést v jednom kroku (stepu, viz kapitola 3). Inkrementální čas je tak načítán v intervalu  $\langle 0, 1 \rangle$ . Jak je ale popsáno v kapitole 3.2, simulace plné geometrie byla provedena ve dvou krocích. Z toho důvodu je samotný proces tváření prováděn v intervalu výpočtového času  $\langle 1, 2 \rangle$ . *PERIODS* je pole hodnot *Total Time Period*. Díky tomuto zohlednění se lze dopočítat korektní hodnoty posuvu v následujícím kroku.

Je zdefinována funkce s názvem *F*, která má dva proměnné parametry - *C1* a *C3*<sup>2</sup>. Ostatní parametry jsou zavedeny jako lokální proměnné a nabývají konstantních hodnot, případně

<sup>2</sup>Kapitola 4.7 říká, že model Bai-Wierzbicki má celkem 6 parametrů, tuto závislost lze ale zredukovat na dva, což bylo provedeno v tomto případě z důvodu snadnější optimalizace cílové funkce, která ještě bude popsána.



```

35 def F(C1, C3):
36     POSUVY=[]
37
38     C2=float(3/2)
39     C4=C2
40     C5=C1
41     C6=C2
42
43
44
45     for m in range(0,2):
46
47         D=0 #Definice promenne parametru D
48         k=0
49         while D<1 and k<(len(PEEQes)-1):
50             deps_pl=(PEEQp[m][k+1]-PEEQp[m][k])
51             triax=(TRIAXp[m][k]+TRIAXp[m][k+1])*0.5
52             lode=(LODE_normp[1][k]+LODE_normp[1][k+1])*0.5
53
54             F1=0.5*(C1*e**(-C2*triax)+C5*e**(-C6*triax))-C3*e**(-C4*triax)
55             F2=0.5*(C1*e**(-C2*triax)-C5*e**(-C6*triax))
56             F3=C3*e**(-C6*triax)
57
58             eps=F1*lode**2+F2*lode+F3
59
60             D=D+(deps_pl)/(eps)
61             k=k+1
62             time=TIMEp[m][k]
63
64             if PERIODS[m]==1.0:
65
66                 posuv=70*time
67
68             elif PERIODS[m]==2.0:
69
70                 posuv=70*(time-1)
71             POSUVY.append(posuv)
72         SUMA=0
73         for n in range(0,2):
74             SUMA=SUMA+(POSUVY[n]-EXPERIMENTS[n])**2
75
76     return (1/m)*SUMA

```

Obrázek 5.7: Definice funkce Bai-Wierzbicki, výpočet momentu porušení

jsou rovny C1 nebo C2. Dále je zaveden parametr poškození D. Je to integrální hodnota, tedy postupně narůstá podle rovnice 4.45. Tato rovnice je ve vytvořeném programu počítána následovně:  $d\bar{\epsilon}_{pl}$  je počítána jako rozdíl mez hodnotami plastické deformace následujícího a současného stavu. Pro veličiny  $\eta$  a  $\epsilon$  jsou dosazeny interpolované hodnoty mezi následujícím a současným stavem. Podle předpisu funkce Bai-Wierzbicki uvedeného v kapitole 4.7 jsou spočítány koeficienty F1, F2, F3 do vztahu pro  $\bar{\epsilon}_f(\eta, \Theta)$  v programu na řádce 58 označené jako eps. Podíl  $d\bar{\epsilon}_{pl}$  a  $\bar{\epsilon}_f$  znamená aktuální přírůstek poškození. Sečtením všech přírůstků obdržíme celkovou míru poškození. Jakmile její hodnota dosáhne čísla 1, znamená to technicky iniciaci trhliny. Podmínka pro zastavení *while* smyčky z řádku 49 (obr. 5.7) je splněna a smyčka je tudíž přerušena. Program zjistí aktuální hodnotu výpočtového času.

Deformace, kterou je vzorek zatěžován, je přímo úměrná inkrementálnímu času. Pro všechny vzorky byla zvolena hodnota 70 mm, tj. 100 % posuvu je 70 mm<sup>3</sup>. Proměnná posuv je tak

<sup>3</sup>MKP softwary ve skutečnosti nepoužívají jednotky, jsou bezrozměrné. Ve vztahu k úloze lze ale říci, že jde

dána vztahem na řádce 65:  $70 * time$ . Proměnná *time* je hodnota výpočtového času v okamžiku, kdy by mělo teoreticky dojít ke vzniku trhliny. V případě, že je simulace prováděna ve dvou krocích, použije program formu na řádce 70:  $70 * (1 - time)$ . Aktuální hodnota kritického posuvu je přiřazena na konec pole kritických posuvů příkazem `POSUVY.append(posuv)` (řádek 71).

Proměnná *SUMa* představuje sumaci ze vzorce 5.1. Postupně jsou do této proměnné načítány rozdíly mezi dosaženými posuvy v simulaci a experimentu, resp. jejich kvadrát. Celá sumace je vynásobena převrácenou hodnotou počtu variant vzorků. Tvar cílové funkce v podobě kódu je na řádce 76 a je výstupem funkce *F*.

## 5.3 Minimalizace cílové funkce

V předchozích dvou kapitolách se skript dopracoval do fáze, kdy po vytřídění dat a jejich načtení do přehledných polí byla získána cílová funkce. Ta má v tomto případě dva parametry - *C1* a *C3*. Naším cílem je najít takovou hodnotou těchto parametrů, kdy bude cílová funkce nabývat nejmenší hodnoty. Jinými slovy, bude tak dosaženo nejmenšího možného rozdílu mezi dosaženou deformací v experimentu a v simulaci. Jednou z variant je zkoušet dosazovat do funkce různé hodnoty a sledovat odezvu. O mnoho efektivnější je sledovat chování funkce v okolí počátečního odhadu a hledat postupně na funkčních hodnotách cestu vedoucí k jejímu minimu (v případě hledání minima). V této práci byla použita metoda, která se nejvíce podobá Rosenbrockově metodě uvedené v odstavci 4.3.1. Optimalizační proces bude nyní popsán.

Název funkce *Fminx* naznačuje, že jde o iteraci ve směru *x*. Nejprve je dosazena souřadnice počátečního odhadu, (souřadnice v *y* je reprezentovaná proměnnou *k* a zůstane v tomto případě neměnná). Tato funkční hodnota je uložena do proměnné *hodnota*. Ve směru *x* je hodnota souřadnice zvětšena o velikost kroku definovanou na řádce 102 a opět je zavolána funkce s touto novou souřadnicí. Její hodnota je uložena do proměnné *nova*. Pokud je hodnota *nova* menší, než *hodnota*, bylo postupováno správným směrem a novým výchozím bodem je právě nová souřadnice. Pokud je hodnota *nova* větší než *hodnota*, je použita proměnná *směr*. Ta nabývá hodnot 1 nebo -1 a tím mění směr postupu na novou souřadnici. Současně dojde k mírnému zjemnění velikosti kroku na 98 % jeho původní velikosti. (Obecně lze samozřejmě zjemnění kroku určit libovolně). Zaznamenáno je také, že došlo ke změně směru. Maximální počet takových změn je stanoven na 1000. Tento parametr lze ale libovolně měnit. Novým výchozím bodem je hodnota, která je směrem "doleva" od původního výchozího bodu. Pokud dojde k velkému počtu změn, znamená to, že funkce osciluje okolo oblasti minima. Proces je opakován tak dlouho, dokud není dosaženo maximálního počtu změn a poslední hodnota je označena za minimum v tomto směru. Jak lze vidět z obr.5.9, tento proces je následně opakován pro směr *y*.

Byly provedeny iterace ve směru *x* a *y*. Pro šetření minima některých základních analytických funkcí je toto dostačující. Cílová funkce se ale vyznačuje tím, že toho mnoho nevíme o jejím tvaru. Je proto nutné zjistit, jestli není možné tento proces zopakovat. K tomu slouží část kódu na obr. 5.10. Dosazovány jsou souřadnice předchozích minim. Pokud je rozdíl mezi minimy menší než zvolená hodnota (zde  $10^{-5}$ ), je oznámeno nalezení minima a cyklus hledání je ukončen, vypsány jsou souřadnice minima.

---

o milimetry.

```

100  f Fminx(i, k):
101      #k=2
102      krok=0.5
103      smer=1
104      j=0
105      minula=0
106      pocetzmen=0
107      zjemneni=0.98
108
109      while pocetzmen<1000:
110
111          hodnota=F(i, k)
112          i=i+krok*smer
113          nova=F(i, k)
114          #print "i", i
115
116          if nova<hodnota:
117              nova=hodnota
118              i=i+krok*smer
119          else:
120              smer=smer*(-1)
121
122              krok=krok*zjemneni
123              i=i+2*krok*smer
124              pocetzmen=pocetzmen+1
125
126              nova=F(i, k)
127
128          minula=nova
129      return i

```

Obrázek 5.8: Jednosměrná iterace v hledání minima - směr  $x$

```

132 def Fminy(i, k):
133     #k=2
134     krok=0.5
135     smer=1
136     j=0
137     minula=0
138     pocetzmen=0
139     zjemneni=0.98
140
141     while pocetzmen<1000:
142
143         hodnota=F(i, k)
144         k=k+krok*smer
145         nova=F(i, k)
146         #print "k", k
147
148         if nova<hodnota:
149             nova=hodnota
150             k=k+krok*smer
151         else:
152             smer=smer*(-1)
153
154             krok=krok*zjemneni
155             k=k+2*krok*smer
156             pocetzmen=pocetzmen+1
157
158             nova=F(i, k)
159
160             minula=nova
161     return k

```

Obrázek 5.9: Jednosměrná iterace v hledání minima - směr  $y$

## 5.4 Shrnutí procesu kalibrace materiálového modelu

V této kapitole byl popsán proces, kterým bylo dosaženo nalezení parametrů modelu porušení Bai-Wierzbicki. Využity k tomu byly poznatky posané v kapitolách předcházejících, tj.

```

164 def Fmin3D(j,l):
165     r=0
166     #Hledani minima v prvni smeru
167     minx=Fminx(j,l)
168     miny=Fminy(minx,l)
169     #Hledani noveho minima ve smeru x
170
171     while r<1:
172
173         minxn=Fminx(minx, miny)
174         minyn=Fminy(minxn, miny)
175
176         if (minxn-minx>1e-5) and (minyn-miny>1e-5):
177             miny=minxn
178             miny=minyn
179         else:
180             print "nalezeno minimum"
181             r=2
182
183     print "minx", minx, "miny", miny
184
185 Fmin3D(5,4.2)

```

Obrázek 5.10: Jednosměrná iterace v hledání minima - směr  $y$

základy matematiky pro mechaniku (4.2), základní znalosti o optimalizačních úlohách (4.3) a samozřejmě formulace modelu porušení Bai-Wierzbicki (4.7). Využito bylo skriptovacího prostředí Python. Plná verze skriptů je uvedena v příloze. Parametry získané z procesu optimalizace funkce byly aplikovány do MKP simulace. Porovnání dosažených výsledků je uvedeno v kapitole 6.1.

# Kapitola 6

## Závěr a komentář výsledků

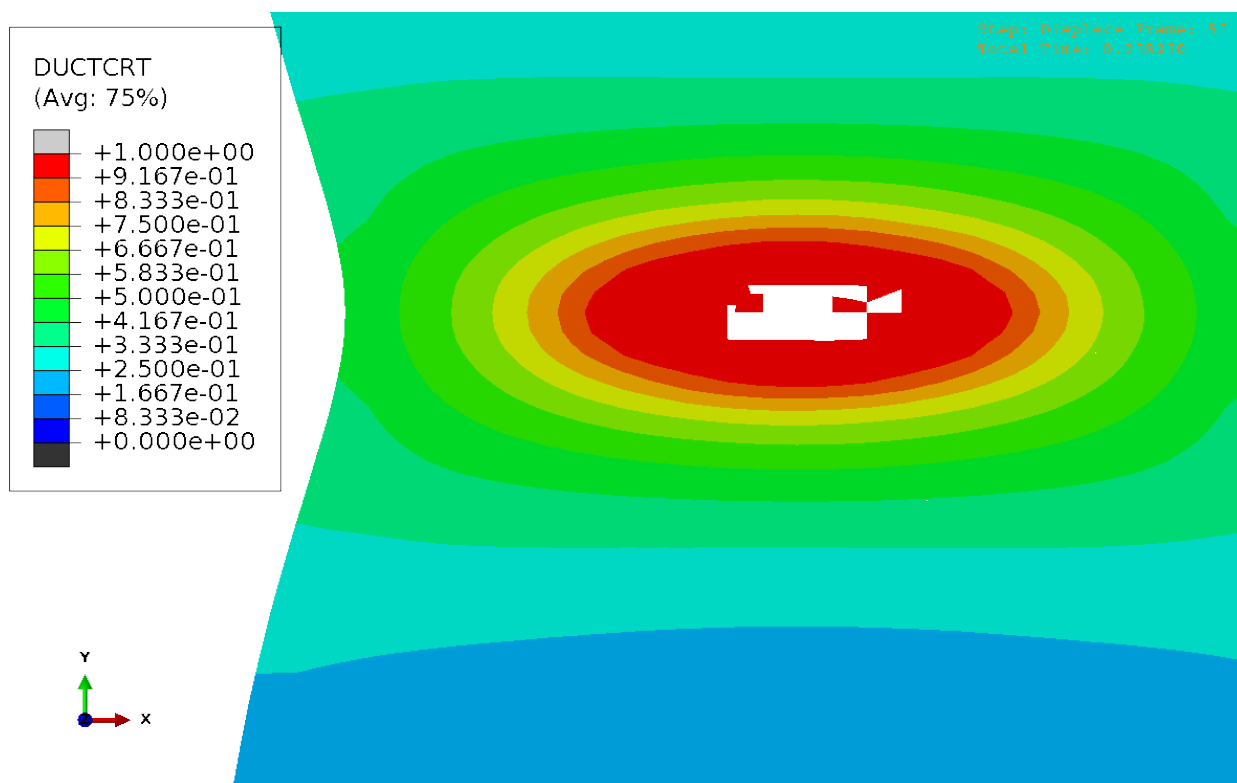
### 6.1 Porovnání dosažených výsledků analytického výpočtu, experimentu a simulace MKP

v dosavadním postupu práce byly odladěny parametry materiálového modelu porušení Bai-Wierzbicki. Ten byl implementován do uživatelské subrutiny spouštěné společně se vstupním souborem analýzy. Model porušení v analýze MKP funguje v tomto případě tak, že pokud parametr porušení  $D$  dosáhne v některém elementu hodnoty 1, je zahájen proces degradace tohoto elementu postupným snižováním *Youngova modulu*. Tím je simulováno šíření trhliny tělesem. Parametry porušení byly odladěny na dvou zkouškách - tahové zkoušce a Nakazima punch testu (viz kapitola 5). Je tudíž na místě ověřit, jak tento model odpovídá danému experimentu. Závislost napětí na deformaci během tahové zkoušky je zobrazena na obr. 6.5. Pro snazší orientaci mezi jednotlivými typy úloh *punch testu* je pro čtenáře vyvořena tabulka 6.1.

Na grafu 6.6 a 6.7 jsou srovnány křivky silové odezvy v pístu razníku při Nakazima punch testu. Konkrétně obr. graf 6.6 porovnává vývoj síly s posuvem mezi dvěma různě vysítovanými modely. (O přístupech k síťování bylo napsáno v kapitole 3). Jako "tužší" se ukázal model, jehož střed geometrie byl síťován metodou *sweep*. Síť na středu plechu tvořená metodou *structured* naopak vykazala větší poddajnost. Přitom jak experimentu, tak analytickému řešení spíše odpovídá síť tvořená jako *sweep*. Po připojení uživatelské subrutiny ale nebyla tato simulace schopna dokonvergovat k žádnému výsledku. Pro simulace se zavedeným modelem Bai-Wierzbicki proto byla použita síť *structured*. Na grafu 6.7 si lze dále všimnout, že obě

Úloha	Popis úlohy	Číslo obrázku	Číslo stránky
a	Analytické řešení	2.1	6
B	Střed plechu síťován metodou <i>sweep</i>	3.3 vpravo	14
C	Střed plechu síťován metodou <i>structured</i>	3.3 vlevo	14
D	Střed plechu síťován metodou <i>structured</i> , implementován model porušení BW	6.3 a 6.4	49
E	Experiment	6.2	49

Tabulka 6.1: Přehled provedených úloh a jejich označení

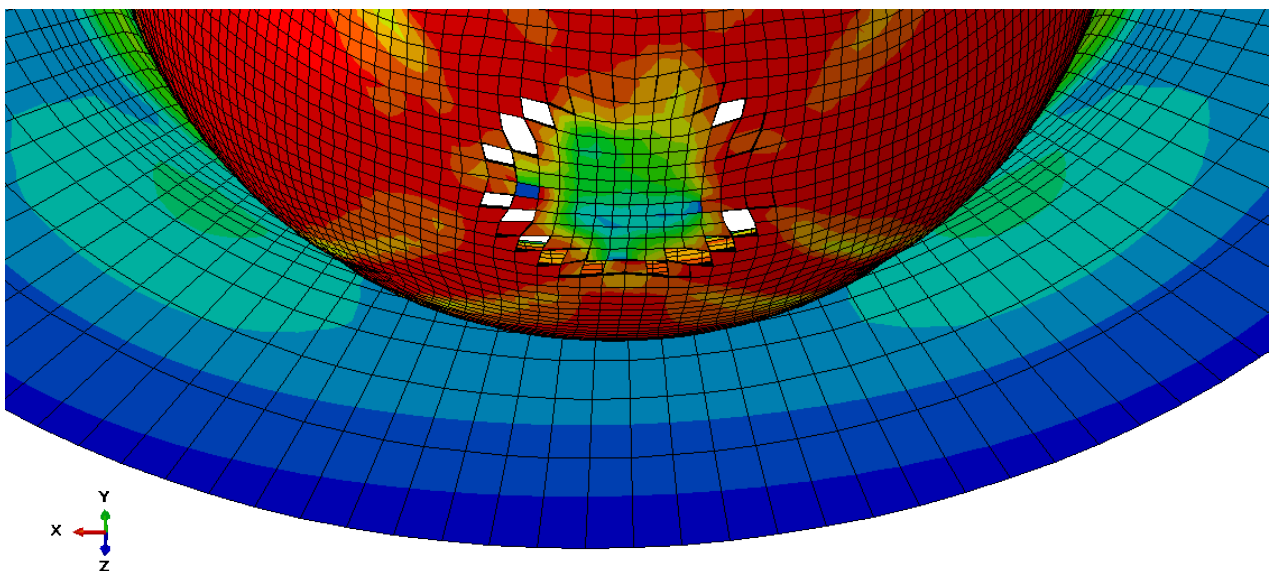


Obrázek 6.1: Simulace tahové zkoušky s modelem tvárného porušení

simulace, ať už s modelem porušení nebo bez něj, dávají možnost shodně určit okamžik iniciace trhliny. v případě "jednoduché" simulace bez subrutiny je to ten okamžik, kdy dochází k prudkému nárůstu tvářecí síly. Ten je dán kolapsem sítě a jejím zamčením. Odpovídá to stavu, kdy v materiálu došlo k vyčerpání zásoby plasticity. Simulace se spuštěnou uživatelskou rutinou na to reaguje degradací a následným odebráním elementu. Lze vidět, že simulace MKP se blíží spíše průběhu síly získanému z analytického řešení. Neshoda mezi křivkou experimentu a zbylými křivkami by mohla znamenat chybu během provádění experimentu. Na vině může být například velikost tření mezi razníkem a vzorkem. Dodržení podmínek tření se ukázalo jako klíčové pro opakovatelnost experimentu. Za pozornost stojí pokles síly v případě MKP simulace bez implementace modelu porušení na obr. 6.6, která sleduje trend daný analytickým řešením. Ten je daný goniometrickými funkcemi vystupujícími ve vztahu pro sílu na razníku uvedeném v kapitole 2. Zde si autor dovolí ještě jednu poznámku k analytickému řešení. Funkce síly má předpis jako  $P = P(\psi, h_0)$ . Parametr  $h_0$  je zde uvažován až do vzniku trhliny jako konstantní. Uvedená křivka síly na razníku z analytického řešení je zobrazena v intervalu  $\langle 0, \pi \rangle$ . Pokud by byl tento interval rozšířen například na  $\langle 0, 2\pi \rangle$ , síla by přešla do záporných hodnot. To technicky nedává smysl, ale je to zajímavá ukázka toho, proč síla při hodnotě posuvu blízké se velikosti poloměru razníku výrazně klesá. Tento předpoklad potvrzuje i graf 6.6, resp. křivka závislosti platná pro simulaci MKP bez implementovaného modelu porušení. Dalším zajímavým jevem je nárůst tvářecí síly po připojení subrutiny. Jev je pozorovatelný jak na případě tahové zkoušky (obr. 6.5), tak na případě simulace punch testu. Tento nárůst (případně pokles) ovlivňuje parametr *energie*, zohledňující vliv velikosti elementů. Parametr energie by bylo možno ladit metodou pokus-omyl a nastavit ho na takovou hodnotu, aby křivky navzájem odpovídaly přesněji. Vzhledem k časové náročnosti simulací byl pro účely práce zachován stav v práci uvedený.

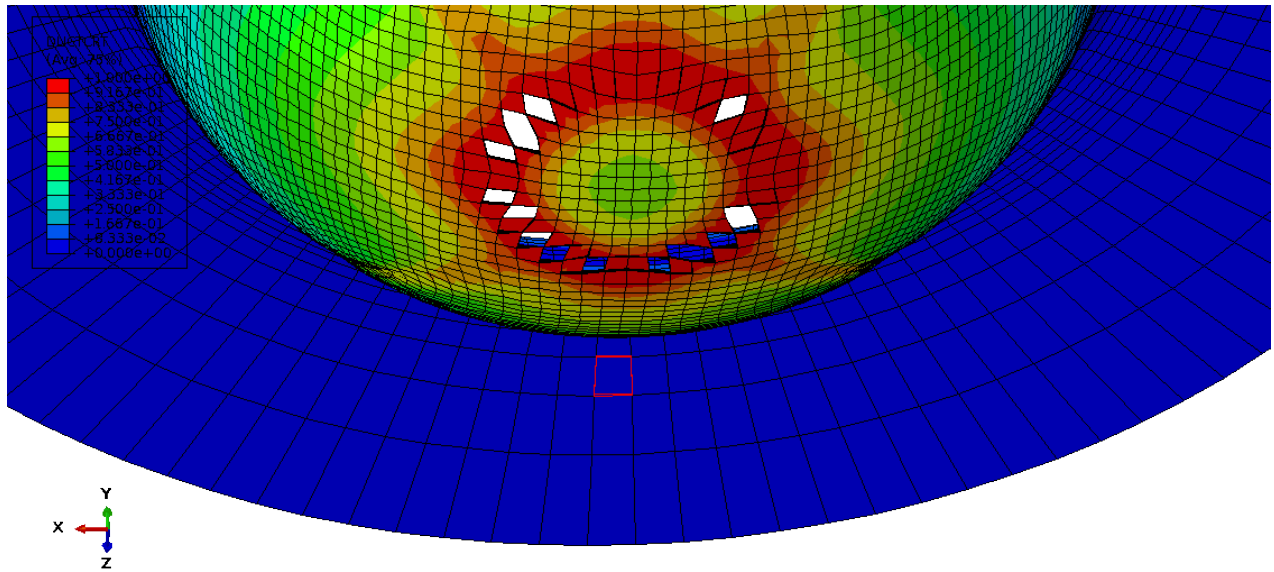


Obrázek 6.2: Zkušební vzorek po Nakazima punch testu

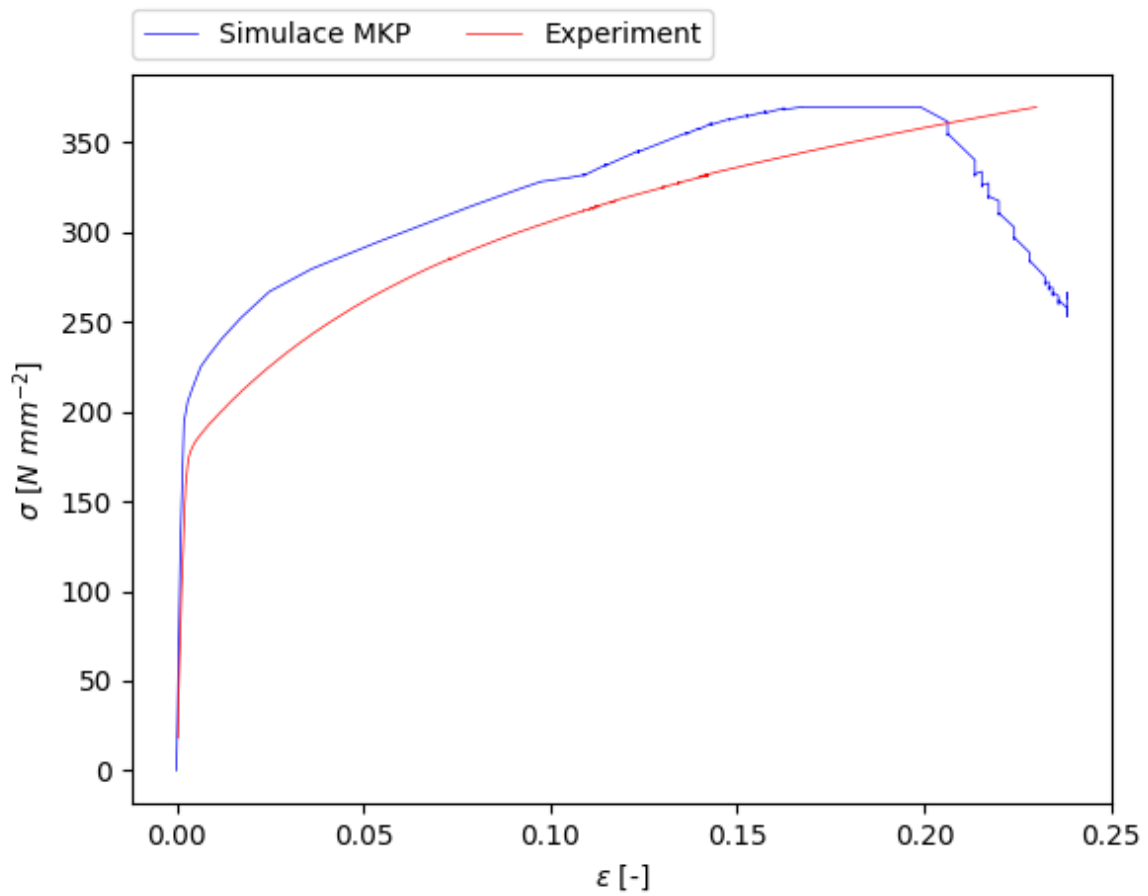


Obrázek 6.3: Simulace punch testu implementovaným modelem poškození, zobrazení pole Misesova napětí.

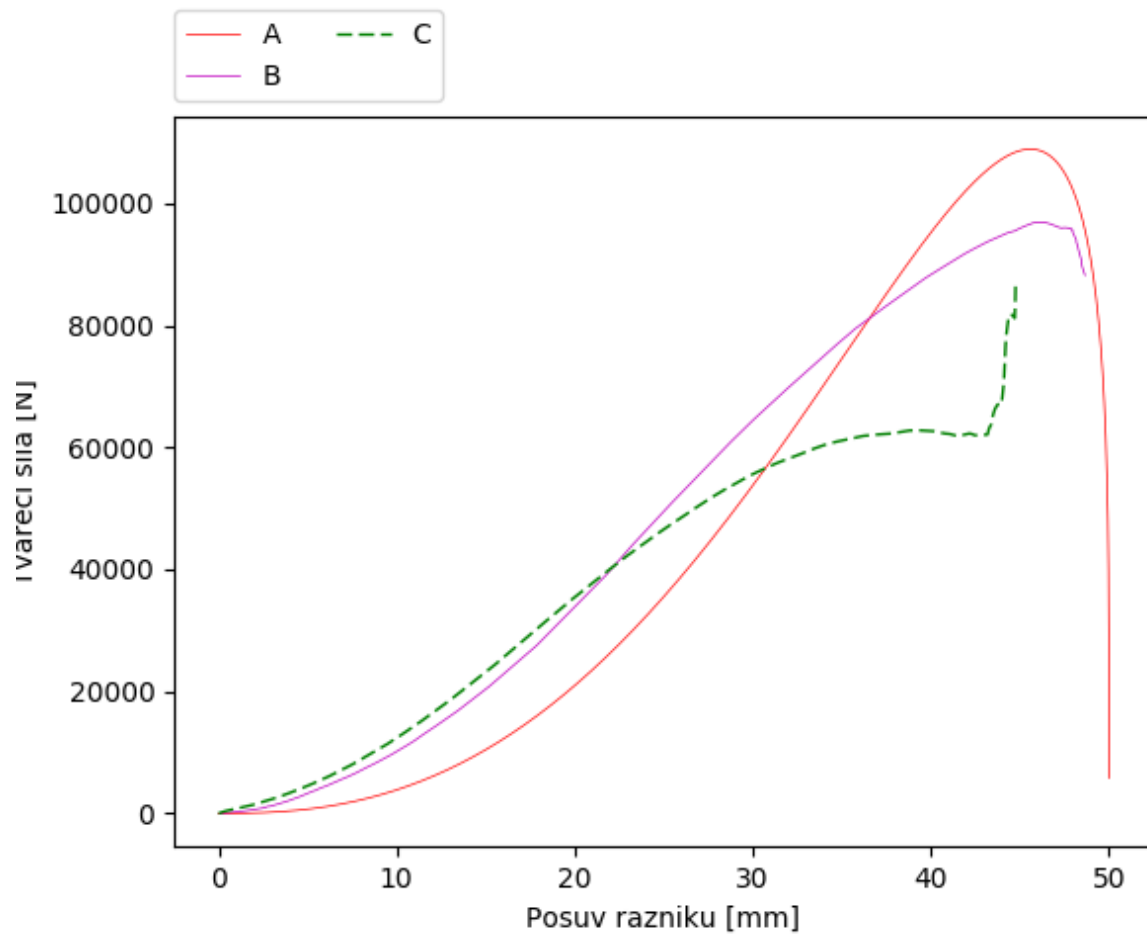




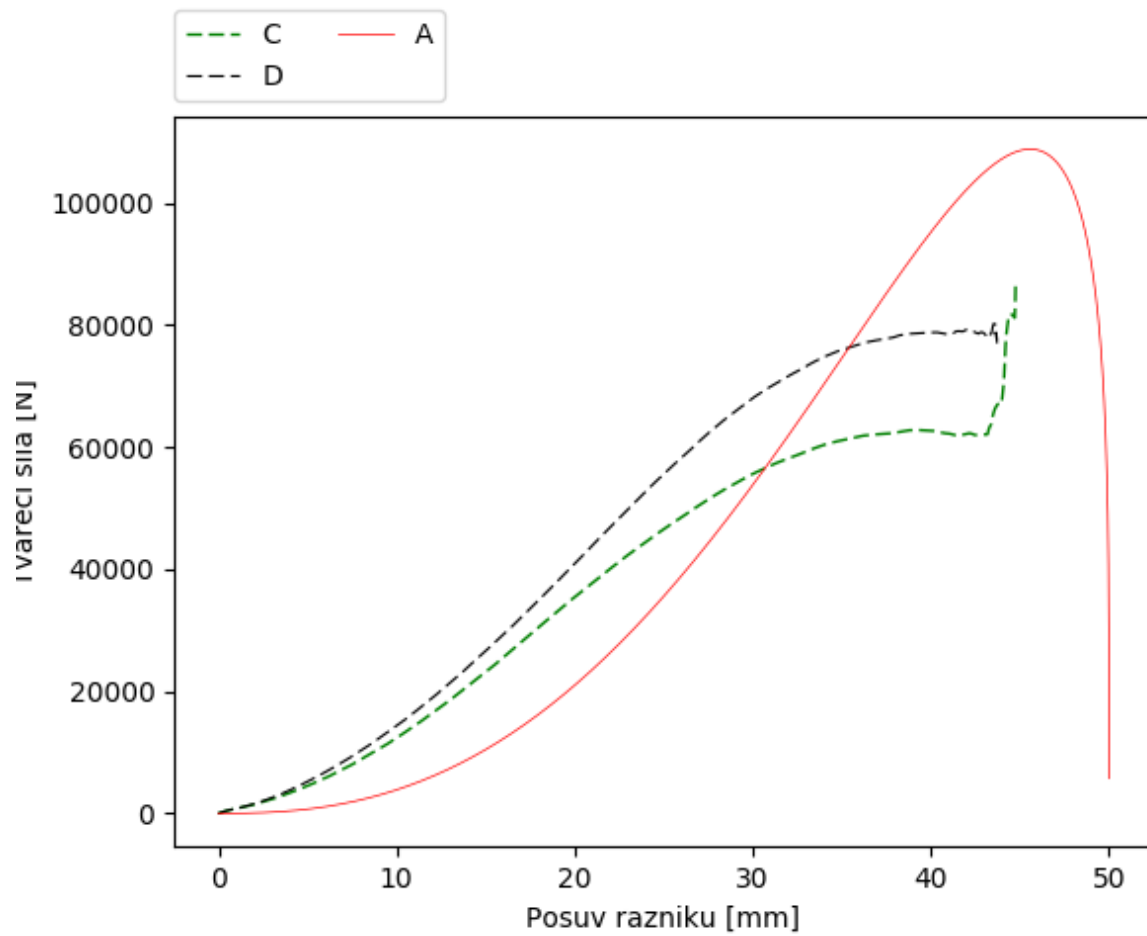
Obrázek 6.4: Simulace punch testu implementovaným modelem poškození, pole parametru poškození.



Obrázek 6.5: Porovnání křivek silové odezvy



Obrázek 6.6: Porovnání křivek silové odezvy pro dvě různé sítě MKP



Obrázek 6.7: Porovnání křivek silové odezvy v závislosti na posunutí kulového razníku

## 6.2 Závěrečné shrnutí

V předložené práci byl nejprve v kapitole 2 proveden analytický přístup k punch testu řešený v článku [1]. Tento problém byl následně převeden do prostředí MKP. V softwaru ABAQUS byla řešena úloha punch testu podle Nakazimy. V kapitole 3 byly vysvětleny různé možnosti přístupu k úloze. Vysvětlena byla důležitost tvaru sítě. Ten totiž významně ovlivňuje kvalitu získaných výsledků. Na základě MKP analýz a experimentálních dat dodaných firmou Comtess FHT byly v kapitole 4 naladěny parametry modelu porušení Bai-Wierzbicki. Postup identifikace materiálu se ukázal jako komplexní problém zahrnující celou škálu prostředků o současném poznání o mechanice a jejich matematických popisech. Je to např. tenzorový počet nebo problematika optimalizace cílové funkce. Samozřejmě důležitá je i obecná znalost o modelech tvárného porušení. Jejich velmi přehledné a srozumitelné vysvětlení lze najít v práci [4].

Na výše zmíněných teoretických základech byla sestaven sada skriptů, jejímž výstupem jsou optimalizované parametry použité plochy porušení. Ruku v ruce se strojírenskou stránkou problému šla potřeba neméně významná. Tou je zpracování výsledků získaných ať už experimentem, nebo jejich extrahováním z výpočtového softwaru. Vzhledem k množství opakování simulací je obecně třeba eliminovat riziko vzniku chyby vlivem lidského faktoru. Opět za pomoci skriptovacího prostředí byla vytvořena sada programů, které získané výsledky zpracovávají a ukládají do přehledné formy. Schopnost algoritmizovat práci se získanými daty je velmi významným přínosem do autorova dovednostního portfolia. Skripty použité k této práci jsou ve své plné podobě v příloze této práce a v elektronické podobě na přiloženém CD.

V závěru práce je v kapitole 6.1 čtenáři graficky zobrazeno, jak závislosti po implementaci modelu porušení, kdy je počítačový model schopen po kumulaci poškození dané elementy vyjmout, korespondují s experimentálně získanými daty.

Ačkoliv míru shody experimentu a simulace hodnotí autor jako uspokojivou, nelze se v žádném případě domnívat, že v této práci bylo vystiženo přesné chování materiálu během jeho tváření. Porušování materiálu a iniciace trhliny, ať už jde o porušování křehké nebo tvárné, je složitý proces odehrávající se na úrovni krystalů a zrn materiálu uvnitř součásti. Na takové úrovni ale inženýři součásti nenavrhují, důležité jsou jejich vlastnosti navenek. Lze se proto uchýlit k fenomenologickým přístupům, kdy jsou na základě zkoušení materiálu určeny jeho vlastnosti v dalším postupu (provozu, výrobě atd). Takový přístup byl použit i v této práci.

Roli zde samozřejmě hraje i statistika. Čím větší portfolio vzorků by bylo zvoleno, tím lépe by fenomenologický model postihoval rozdílné druhy namáhání. Netřeba dodávat, že tím více by jeho kalibrace stála. Vzniká tak malá, ale důležitá optimalizační úloha pro inženýra, který musí zvolit hlediska a rozhodnout, jak přesně má být materiál kalibrován, případně podle jakého fenomenologického modelu.

Cílem této práce bylo naladění materiálového modelu ze zadaných experimentů. Byly provedeny experimenty tahová zkouška a Nakazima punch test. Použitý laděný materiál nese označení DC01. Výsledkem práce jsou simulace s implementovaným a naladěným materiálovým modelem. Lze tak konstatovat, že zadání bylo splněno. Ačkoliv tento proces zkonsumoval značné množství času, vyprodukoval nástroje, které jsou v budoucnu schopny takový postup pro zkoumání dalších materiálů značně urychlit, a to hlavně s minimální podporou obsluhy. Největší časovou zátěží tak nadále bude výpočtový čas simulace a čas se vzorky na zkušebně, tedy ryze inženýrské činnosti. Tato práce tak kromě splnění svého zadání poskytla i možnost zjednodušení procesu identifikace materiálu jejímu uživateli.

# Příloha A

## Skript k načtení výsledků z programu ABAQU - BW.py

```
import xlrd #knihovna pro práci s excelovským formátem
import numpy as np

SESIT=xlrd.open_workbook('Results.xlsx') #Zadefinovani excelovskeho souboru, ze
ktereho budou data importovana
LIST=SESIT.sheet_by_index(1) #Urceni sesitu v excelovskem souboru, ze ktereho budou
data extrahovana. Index "1" je importovan ze souboru 2Dpole.py, který spouští tento
skript
#Definovani poli promennych, do kterych jsou ukladany hodnoty v bunkach PEEQes=[]
#Pole hodnot plasticke deformace
MAXes=[] #...Sigma1 hlavnich napeti
MIDes=[] #...Sigma2
MINes=[] #...Sigma3
MISESes=[] #...Misesova napeti
Sigma11s=[] #... a tak dale
Sigma22s=[]
Sigma33s=[]
Sigma12s=[]
Sigma13s=[]
Sigma23s=[]
TIMES=[]
#Ukladani hodnot z i-teho radku
for i in range(0,LIST.nrows-1):
    PEEQ=LIST.cell(i+1,0).value
    MAX=LIST.cell(i+1,1).value
    MID=LIST.cell(i+1,2).value
    MIN=LIST.cell(i+1,3).value
    Sigma11=LIST.cell(i+1,4).value
    Sigma22=LIST.cell(i+1,5).value
    Sigma33=LIST.cell(i+1,6).value
    Sigma12=LIST.cell(i+1,7).value
    Sigma13=LIST.cell(i+1,8).value
```

```

Sigma23=LIST.cell(i+1,9).value
MISES=LIST.cell(i+1,10).value
TIME=LIST.cell(i+1,11).value
if PEEQ=='NoValue' or PEEQ=='0.': #Podminka pro obejiti hlasky NoValue z rpt
file (rpt je vysledkovy soubor z programu ABAQUS)
    continue
else:
PEEQes.append(float(PEEQ)) #Plneni poli jednotlivych velicin hodnotami
MAXes.append(float(MAX))
MIDes.append(float(MID))
MINes.append(float(MIN))
Sigma11s.append(float(Sigma11))
Sigma22s.append(float(Sigma22))
Sigma33s.append(float(Sigma33))
Sigma12s.append(float(Sigma12))
Sigma13s.append(float(Sigma13))
Sigma23s.append(float(Sigma23))
MISESes.append(float(MISES))
TIMES.append(float(TIME))
#Vytvoreni pole hodnot, se kterymi se nasledne pocita - traxialita, normovany LODEho
uhel
TRIAXes=[] #Definice obou poli,
LODE_normes=[] #do kterych se budou hodnoty nacistat
for j in range(0, len(PEEQes)):
    MAX=MAXes[j]
    MID=MIDes[j]
    MIN=MINes[j]
    MISES=MISESes[j]
    TRIAX=0.33*(MAX+MID+MIN)/(MISES) #TRIAXIALITA
    TRIAXes.append(TRIAX) #NACTENi AKTUALNi TRIAXIALITY DO POLE TRIAXIALIT
#Vypocet LODEHO uhlu

Sigma11=Sigma11s[j]
Sigma22=Sigma22s[j]
Sigma33=Sigma33s[j]
Sigma12=Sigma12s[j]
Sigma13=Sigma13s[j]
Sigma23=Sigma23s[j]

if (Sigma11==0) and (Sigma22==0) and (Sigma33==0):

    Sigma11=MAX
    Sigma22=MID
    Sigma33=MIN

else:
    jentakabysenereklo=1 #Toto nedela nic, ale prikaz "continue"nefungoval. Nevim

```

```

proc
#Matice tenzoru napeti

    SIGMA=np.array([[Sigma11, Sigma12, Sigma13], [Sigma12, Sigma22, Sigma23], [Sigma13,
Sigma23, Sigma33]])
    #Vypocet deviatoru napeti
    STOPA=np.trace(SIGMA)
    I=np.eye(3)
    DEVIATOR=SIGMA-0.33*STOPA*I
#Vypocet druheho a tretiho invariantu deviatoru napeti
    J2=0.167*((Sigma11-Sigma22)**2+(Sigma22-Sigma33)**2+(Sigma33-Sigma11)**2)
    J3=DEVIATOR[0,0]*DEVIATOR[1,1]*DEVIATOR[2,2]

    citatel=(3*3**(0.5)*J3)

    jmenovatel=(2*(J2**3)**(0.5))

    pomer=citatel/jmenovatel
#Vypocet Lodeho uhlu a osetreni vyskytu nejake numericke nestability

    if (pomer <= (-1)) and (pomer>-1.1):

        LODE=np.pi

    elif (pomer >= (1)) and (pomer<1.1):

        LODE=0

    elif ((-1)<pomer<(1)):

        LODE=0.33*np.arccos((3*3**(0.5)*J3)/(2*(J2**3)**(0.5)))
    else:

        print "CHYBA! ARGUMENT LODEHO UHLU JE", pomer

        print "OCEKAVANA HODNOTA v INTERVALU -1, 1"

        print "CHYBA JE v LISTU", m+1 , "RADEK", j+1
        continue

#Normalizovany Lodeho parametr

    LODE_norm=1-(6*LODE)/(np.pi)

    LODE_normes.append(LODE_norm)

```

# Příloha B

## Zpracování načtených dat a vytvoření cílové funkce

```
import scipy.optimize as optimize
import numpy

    #Zadefinovani pole poli ziskanych opakovanym spoustenim skriptu BW_Beta (tj.
    2D pole)
    PEEQp=[] #Pole promennych PEEQ
    TRIAXp=[] #Pole triaxialit
    LODE_normp=[] #a tak dale...
    TIMEp=[]
    PERIODS=[]#Pomucka na osetreni promennych celkovych delek inkrementalnich casu
    EXPERIMENTS=[]#Pole dat z experimentu

#Cyklicke spousteni skriptu BW_Beta, index oznacuje cislo listu v souboru Results.xlsx
for l in xrange(0,2):
    execfile ('BW_Beta.py')#Spusti skript BW_Beta s indexem "l"

#Pole hodnot ulozene skriptem BW_Beta je prirazeno na konec 2D poli
    PEEQp.append(PEEQes)
    TRIAXp.append(TRIAXes)
    LODE_normp.append(LODE_normes)
    TIMEp.append(TIMEs)

#Vytvoreni pole celkove casove periody analyzy
    RESULTSexp=SESIT.sheet_by_index(-1)
    Period=RESULTSexp.cell(1+1, 3).value
    PERIODS.append(float(Period))

#Vytvoreni pole vyslenych posuvu analyzy
    Experiment=RESULTSexp.cell(1+1,1).value
    EXPERIMENTS.append(float(Experiment))

e=np.e #Eulerovo cislo
```



```

#Vytvoreni cilove fce. Jeji parametry jsou dva - C1 a C3
def F(C1, C3):

#Pole hodnot posuvu pri poruseni
    POSUVY=[]

#Substituce za zbyte parametry
    C2=float(3/2)
    C4=C2
    C5=C1
    C6=C2

#Pomocna promenna pro sledovani postupu tabulkou
    for m in range(0,2):
        D=0 #Definice promenne parametru D
        k=0

        #Vypocet kumulace poskozeni
        while D<1 and k<(len(PEEQes)-1):

            deps_pl=(PEEQp[m][k+1]-PEEQp[m][k])
            triax=(TRIAxp[m][k]+TRIAxp[m][k+1])*0.5
            lode=(LODE_normp[1][k]+LODE_normp[1][k+1])*0.5

            F1=0.5*(C1*e**(-C2*triax)+C5*e**(-C6*triax))-C3*e**(-C4*triax)
            F2=0.5*(C1*e**(-C2*triax)-C5*e**(-C6*triax))
            F3=C3*e**(-C6*triax)

            eps=F1*lode**2+F2*lode+F3

#Zavedeni parametru poskozeni
        D=D+(deps_pl)/(eps)#pricteni prirustku poskozeni
        k=k+1
        time=TIMEp[m][k]#Urceni inkrementalniho casu poskozeni

#Korekce vypoctoveho casu ve vztahu k Total Time Period
        if PERIODS[m]==1.0:

            posuv=70*time

        elif PERIODS[m]==2.0:

            posuv=70*(time-1)

        POSUVY.append(posuv)

    SUMA=0

```

```
for n in range(0,2):
    SUMA=SUMA+(POSUVY[n]-EXPERIMENTS[n])**2

return (1/m)*SUMA#Navratova hodnota cilove funkce
```

# Příloha C

## Funkce hledání minima

Tato funkce hledá minimum v jednom směru cílového funkcionálu. Je elementární funkcí jeho optimalizačního cyklu.

```
def Fminx(i,k):

    krok=0.5#velikost kroku
    smer=1#+1znamena smer kladneho rustu ve v danem smeru

    minula=0#predchozi funkcní hodnota
    pocetzmen=0#pocet zmen smeru behem iterace
    zjemneni=0.98#míra zjemnení po změně smeru

    while pocetzmen<1000:#v tomto případě byl počet zmen omezen na 1000

        hodnota=F(i,k)#nactení aktualní funkcní hodnoty
        i=i+krok*smer#určení noveho bodu pro zjistení funkcní hodnoty
        nova=F(i,k)#nactení nove funkcní hodnoty

        #pokud je nova funkcní hodnota vetsi, než minula, je nova ulozena jako aktualní a soucasna
        je novym vychozím bodem
        if nova<hodnota:

            nova=hodnota
            i=i+krok*smer

        else:
            smer=smer*(-1)#zajistení zmeny smeru postupu

            krok=krok*zjemneni#zjemnení kroku
            i=i+2*krok*smer#presun do nove souradnice v opacnem smeru
            pocetzmen=pocetzmen+1#zaznamenání toho, že doslo ke změně smeru

    nova=F(i,k)#uložení nove funkcní hodnoty

    minula=nova#v prístím kroku bude "minula" hodnota ta "nova", soucasna
    return i#navrat nove souradnice. Ta odpovida hodnote ladeneho parametru C
```

# Seznam obrázků

2.1	Geometrie konfigurace punch testu . . . . .	6
2.2	Zobrazení křivky regresní závislosti a její porovnání s křivkou získaných dat . .	8
2.3	Průběh síly v pístu testovacího zařízení v závislosti na posuvu kulového razníku získaný analytickou aproximací . . . . .	9
3.1	Schema úlohy Nakazima punch testu . . . . .	13
3.2	Průběh plastické deformace v osově symetrické úloze . . . . .	14
3.3	Porovnání průběhu plastické deformace v radiálním směru od středu plechu . .	14
3.4	Prvotní tvar sítě v úloze cyklické symetrie . . . . .	16
3.5	Rozložení ekvivalentní plastické deformace v úloze s tuhým razníkem . . . . .	16
3.6	Rozložení napětí v místě kontaktu plechu a přípravku . . . . .	17
3.7	Tvar první optimalizace sítě . . . . .	17
3.8	Zobrazení pole napětí HMM po optimalizaci sítě v úloze s deformovatelným razníkem . . . . .	18
3.9	Poddajný razník . . . . .	18
3.10	Nevhodný tvar sítě na středu geometrie analyzovaného plechu . . . . .	19
3.11	Rozdělení geometrie a výsledný tvar sítě . . . . .	20
3.12	Deformovaná 3D geometrie plechu . . . . .	20
3.13	Porovnání sítí ve středové části. Vpravo je ve středu plechu síť tvořena metodou <i>sweep</i> . . . . .	21
3.14	Zobrazení pole napětí HMM na geometrii s použitou metodou <i>sweep</i> . . . . .	21
3.15	Zobrazení pole napětí HMM, síť tvořená pouze jako <i>structured</i> . . . . .	22
3.16	Zobrazení pole napětí HMM, síť tvořená pouze jako <i>structured</i> , použity kvadratické elementy . . . . .	23
3.17	Zobrazení pole Ekvivalentní plastické deformace, síť tvořená pouze jako <i>structured</i> , použity lineární elementy . . . . .	23
3.18	Zobrazení pole Ekvivalentní plastické deformace, střed síťován metodou <i>sweep</i> , použity lineární elementy . . . . .	24
3.19	Zobrazení pole ekvivalentní plastické deformace, střed síťován metodou <i>sweep</i> , použity kvadratické elementy . . . . .	24
4.1	Ukázka Rosenbrockovy metody pro případ optimalizace dvouparametrické funkce [3] . . . . .	31
4.2	Haigh-Westgaardův prostor hlavních napětí [4] . . . . .	32
5.1	Příklad sady výsledků získané z programu Abaqus . . . . .	37
5.2	Import knihoven potřebných pro práci s xlsx soubory a vytvoření polí, do kterých budou načítána data . . . . .	38
5.3	Načítání hodnot do polí proměnných . . . . .	39

5.4	Výpočet hodnoty triaxiality a zdefinování složek tenzoru napjatosti . . . . .	39
5.5	Definice matice $\sigma$ a výpočet Lodeho úhlu . . . . .	40
5.6	Začátek skriptu <b>pole.py</b> , zdefinování 2D polí pro načtení hodnot z více experimentů . . . . .	41
5.7	Definice funkce Bai-Wierzbicki, výpočet momentu porušení . . . . .	42
5.8	Jednosměrná iterace v hledání minima - směr $x$ . . . . .	44
5.9	Jednosměrná iterace v hledání minima - směr $y$ . . . . .	45
5.10	Jednosměrná iterace v hledání minima - směr $y$ . . . . .	46
6.1	Simulace tahové zkoušky s modelem tvárného porušení . . . . .	48
6.2	Zkušební vzorek po Nakazima punch testu . . . . .	49
6.3	Simulace punch testu implementovaným modelem poškození, zobrazení pole Misesova napětí. . . . .	49
6.4	Simulace punch testu implementovaným modelem poškození, pole parametru poškození. . . . .	50
6.5	Porovnání křivek silové odezvy . . . . .	50
6.6	Porovnání křivek silové odezvy pro dvě různé sítě MKP . . . . .	51
6.7	Porovnání křivek silové odezvy v závislosti na posunutí kulového razníku . . . . .	52

# Seznam tabulek

6.1	Přehled provedených úloh a jejich označení . . . . .	47
-----	--	----

# Literatura

- [1] Tomasz Wierzbicki Young-Woong Lee, Jeffrey C. Woertz. Fracture prediction of thin plates under hemi-spherical punch with calibration and experimental verification. *International Journal of Mechanical Sciences*, 31:751–781, 2004.
- [2] Abaqus 6.14 documentation. abaqus documentation web server [online]. dostupné z: <http://abaqus.software.polimi.it/v6.14/index.html>, 2014.
- [3] J. Brousil P. Lederer. *Teorie a optimalizace mechanických systémů II*. 1989.
- [4] Jan Růžička. *Metodika kalibrace nesvázaných modelů a stochastický přístup v problematice tvárného porušování*. 2015.
- [5] F. A. McClintock. *A criterion fo ductile fracture by the growth of holes*. 1968.
- [6] Jiří Plešek. *Mechanika kontinua s příklady*, volume 113. 2015.
- [7] Jan Švec. *Učebnice jazyka Python (aneb Létaující cirkus)*. 2.2 edition, 2002.
- [8] Tomasz Wierzbicki Yingbin Bao. On fracture locus in the equivalent strain and stress triaxiality space. *International Journal of Mechanical Sciences*, 18:81–98, 2002.
- [9] Jan Růžička. Popis programového skriptu pro identifikaci parametrů modelu tvárného porušení, 2012.