

*AUTORIZOVANÝ SOFTWARE*

***LLIS1D\_Y2P***

SOFTWARE PRO ANALÝZU ODDĚLUJÍCÍHO LISOVÁNÍ V LINEÁRNÍ  
LISOVACÍ KOMOŘE

---

Autor: *Ing. Jan Očenášek, Ph.D.*  
*RNDr. Josef Voldřich, CSc.*

Číslo projektu: FR-TI 1/369

Číslo výsledku: *NTC-SW-04-10*

Odpovědný pracovník: *Ing. Jan Očenášek, Ph.D.*

Vedoucí odboru: *RNDr. Josef Voldřich, CSc.*

Ředitel centra: *doc. Dr. RNDr. Miroslav Holeček*

---

**Jazyk výsledku:** CZE

**Hlavní obor:** JR

**Uplatněn:** ANO

**Název výsledku česky:**

*LLIS1D\_Y2P Software pro analýzu oddělujícího lisování v lineární lisovací komoře*

**Název výsledku anglicky:**

*LLIS1D\_Y2P Software for the analysis of the extruding-expelling process in linear press chamber*

**Abstrakt k výsledku česky:**

*LLIS1D\_Y2P (verze 1.0) je program v jazyce Python pro výpočet zatěžujícího tlakového profilu v závislosti na předepsané časové funkci výtěžnosti či posuvu pístu pro případ oddělujícího lisování v lineární lisovací komoře. Algoritmus je založen na modelu lineární lisovací komory popsaného, který zohledňuje reologické vlastnosti stlačitelného saturovaného porézního skeletu a difúzi tekuté fáze popisuje Darcyho zákon. Rozložení tlaku intersticiální tekutiny je řešeno Galerkinovo metodou a deformace skeletu pomocí explicitní časové integrace. Okrajová podmínka tlaku na píst je identifikována z podmínky posuvu pístu iterativním optimalizačním procesem. Program byl vytvořen jako podpora pro navržení optimálních lisovacích parametrů lisovacího proces.*

**Abstrakt k výsledku anglicky:**

*A software tool LLIS1D\_Y2P (v1.0) is a program in Python language for calculation of the pressure profile when the yield or piston position is known and defined as a function of time during the process of fluid expelling in a linear compression chamber. The model accounts for rheological properties of the deformable solid matrix of compressed seed. The filter cake is modeled as a saturated porous media, where the permeability of solid matrix is described by the Darcy's law. The piecewise linear Galerkin method and the explicit time integration scheme was applied. The boundary condition defining the piston load is identified through iterative bounded minimization routine to match the prescribed piston displacement. This program was developed as a support tool for the design of an optimal filtration process.*

**Klíčová slova česky:**

*Permeabilita;filtrace;konsolidace;porézní médium;Darcyho zákon;výtěžnost*

**Klíčová slova anglicky:**

*Permeability;filtration;consolidation;porous media;Darcy's law;yield*

**Vlastník výsledku:** *Západočeská univerzita v Plzni*

**IČ vlastníka výsledku:** *49777513*

**Stát:** *Česká republika*

**Lokalizace:** [http://www.zcu.cz/ntc/vysledky/sw/ntc\\_sw\\_04\\_10.html](http://www.zcu.cz/ntc/vysledky/sw/ntc_sw_04_10.html)

**Licence:** *ANO*

**Licenční poplatek:** *NE*

**Ekonomické parametry:** *Analýza technologických procesů pomocí počítačových modelů významně přispívá k jejich dalšímu vývoji, zpřesnění, vyšší efektivitě, zlepšení parametrů výroby a snížení nákladů*

**Technické parametry:** *Luděk Hynčík, Západočeská univerzita v Plzni, Nové technologie - Výzkumné centrum v západočeském regionu, Univerzitní 8, 306 14 Plzeň, 377634709, [hyncik@ntc.zcu.cz](mailto:hyncik@ntc.zcu.cz)*

---

# **LLIS1D\_Y2P Dokumentace**

*Vydání 1.0*

**Jan Očenášek**

03.01.2011



---

# Obsah

---

<b>1</b>	<b>Úvod</b>	<b>3</b>
1.1	Teoretický úvod . . . . .	3
1.2	Instalace . . . . .	4
<b>2</b>	<b>Tutorial</b>	<b>5</b>
2.1	run_Y2P . . . . .	5
2.2	run_POST . . . . .	7
<b>3</b>	<b>Dokumentace</b>	<b>13</b>
3.1	assemble.py . . . . .	13
3.2	solvers.py . . . . .	14
3.3	fortran_test.py . . . . .	15
3.4	parametry.py . . . . .	15
3.5	std_report.py . . . . .	16
3.6	xls_export.py . . . . .	17
3.7	fortran.t_diag_sym_f . . . . .	17
<b>4</b>	<b>Literatura</b>	<b>19</b>
<b>5</b>	<b>Indices and tables</b>	<b>21</b>
	<b>Literatura</b>	<b>23</b>
	<b>Index</b>	<b>25</b>



*LLIS1D\_Y2P* - SOFTWARE PRO ANALU ODDUJO LISOVN LINERNISOVACOMO

je program v jazyce **Python** pro výpočet zatěžujícího tlakového profilu v závislosti na předepsané časové závislosti výtěžnosti pro případ oddělujícího filtrování v lineární lisovací komoře. Teorie je popsána v článku [i] nebo výzkumné zprávě [ii].

**Poděkování**

Tento software vznikl za finančního přispění Ministerstva průmyslu a obchodu v rámci podpory výzkumu a vývoje. Jedná se o projekt FR-TI 1/369 “Výzkum a vývoj oddělujícího šnekového lisování”, který je řešen ve spolupráci s firmou FARMET a.s., Česká Skalice.

*obsah:*





---

# Úvod

---

*LLISID\_Y2P* je programový nástroj, jehož hlavní částí je soubor

- `run_Y2P.py`

Pro základní vizualizaci výsledků je přiložen program `run_POST.py`.

Ostatní části programu, které nejsou samostatně spustitelné, jsou v těchto souborech:

*assemble.py, fortran\_test.py, parametry.py, solvers.py, xls\_export.py, std\_report.py*

V adresáři (modulu) *fortran* je dále kód pro řešení soustavy rovnic napsaný ve fortranu a kompilovaný pomocí modulu *F2py*. Program *run\_Y2P* se automaticky pokouší využít tento efektivní kompilovaný řešič. Pokud se to nepodaří, použije o něco pomalejší kód pythonu. Při přechodu na jinou platformu (operační systém) není zaručena kompatibilita kompilovaného kódu a kompilaci je třeba provést znovu.

Program dále umožňuje automatické vytvoření protokolu s výsledky výpočtu ve formátu html, nebo export výsledků ve formátu sešitu MS Excel.

## 1.1 Teoretický úvod

Úloha je modifikovaným problémem oddělujícího lisování v lineární lisovací komoře. Standardní formulace problému (viz. [i] nebo [ii]) předpokládá znalost zatížení na píst lisu a výsledkem řešení je mimo jiné znanost posuvu pístu nebo výtěžnost. Tento program řeší inverzní problém, kdy je znám posuv pístu v čase a síla na píst známá není. Dalšími výsledky řešení jsou síly přenášené skeletem, porozita, vnitřní tlak nekutiny, objemová deformace, nebo posuv skeletu.

Posuv pístu v čase je předepsán pomocí bezrozměrné veličiny  $U_t(t)$  udávající výtěžnost, tj. poměr mezi oddělenou hmotností oleje a počáteční hmotností oleje ve vzorku. Při platnosti předpokladů uvedených v [i] je vztah mezi výtěžností a posuvem pístu  $h_t(t)$  následující:

$$h(t) = h_0(1 - \Phi_0 U_t(t)),$$

kde  $h_0$  je počáteční poloha pístu (vzdálenost od síta) a  $\Phi_0$  je počáteční porozita. Vztah mezi počáteční porositou a hmotnostním poměrem oleje v semenech  $\phi_0$  je

$$\frac{1}{\Phi_0} = 1 + \frac{\rho_f}{\rho_s} \frac{1 - \phi_0}{\phi_0},$$

kde  $\rho_s$  je hustota pevné fáze a  $\rho_f$  hustota oleje.

## 1.2 Instalace

Pro správnou funkci programu je doporučena instalace Python verze 2.6.

Závislost na dalších modulech:

- numpy
- scipy
- re
- wx
- matplotlib
- pickle
- pyExcelerator (potřebný pouze pro export výsledků do .xls)

Pro instalaci v systém Windows doporučujeme distribuci [pythonXY](#) -. Poté je ještě potřeba přidat modul *pyExcelerator* pro možnost exportu výsledků ve formátu *MS-EXCEL*.

Modul *pyExcelerator* se přidává (po instalaci Pythonu) příkazem z instalačního adresáře modulu (viz také *readme.txt* tohoto modulu):

```
python setup.py install
```

---

# Tutorial

---

## 2.1 run\_Y2P

Program `run_Y2P.py` řeší popsaný problém, když jsou dány všechny materiálové a ostatní parametry úlohy vstupním souborem.

Spuštění programu se provede příkazem (z příkazové řádky):

```
python run_Y2P.py --input vstup.txt --output vystup.dat
```

nebo krátce:

```
python run_Y2P.py -ivstup.txt -ovystup.dat
```

bez parametrů (dafaultní vstupní a výstupní soubor):

```
python run_Y2P.py
```

je stejné jako:

```
python run_Y2P.py --input input.txt --output results.dat
```

Parametr `--input` zadává vstupní soubor, který obsahuje všechny parametry výpočtu.

Parametr `--output` určuje jméno standardního výstupního souboru.

### 2.1.1 Formát vstupního souboru

Vstupní soubor je textový soubor, který musí obsahovat všechny předepsané parametry. Seznam parametrů, jejich význam a formát souboru je zřejmý z následujícího příkladu:

```
@#
@#   Vstupni soubor pro program Y2P LLIS1D
@#
D=      4e-5      @# velikost elementu [m]
nel=    250      @# pocet prvku [1]
tsteps= 101      @# pocet casovych kroku [1]
tmax=   600.     @# max. cas [s]
Phi0=   0.5      @# pocatecni porosita [1]
mu=     55e-3    @# dynamicka viskozita tekute faze (oleje) [Pa s]
gamma=  0.       @# parametr analogicky s Biotovym parametrem [1]
E=      117e6    @# objemovy modul pruznosti pevne faze [Pa]
```

```

eta=      1.8e9      @# viskozita pevne faze [Pa s]
alpha0=   3e10      @# parametr alpha0 konstitutivního vztahu pro permeabilitu [m/kg]
C=        27.       @# exponent C konstitutivního vztahu pro permeabilitu [1]
rho_s=    1052.     @# hustota skeletu [kg/m3]
Alpha=    1e-8      @# parametr sita [1/m]
pOkoli=   0.1e6     @# parametr sita [Pa]
P         = 30e6    @# zatizeni na pist [Pa]
method=   2        @# method=2 pro Runge-Kutta, jinak Newton
echo      = 1      @# echo= 1 nebo 2 pro vypis prubehu vypoctu
xls       = 1      @# xls=1 pro vypis do MS-excel souboru
html      = 1      @# html=1 pro vytvoreni predefinovane html zpravy

```

## 2.1.2 Popis parametrů

- **D** [m], **nel** [1]- Velikost elementu, počet elementů. Počáteční tlouška filtračního koláče je tedy  $h_0 = D \cdot nel$ .
- **tsteps** [1], **tmax** [s] - Počet časových kroků, maximální čas. Určují dobu lisování a jemnost časových kroků.
- **Phi0** [1] - počáteční porosita
- **mu** [Pa s] - dynamická viskozita tekuté fáze (oleje)
- **gamma** [1] - parametr analogický s Biotovým parametrem
- **E** [Pa] - objemový modul pružnosti pevné fáze (skletu)
- **eta** [Pa s] - viskozita pevné fáze
- **alpha0** [m/kg], **C** [1] - Parametry konstitutivního vztahu pro permeabilitu. Parametr *alpha0* je počáteční odpor síta a *C* je exponent empirického vztahu Tillerera a Yeha, viz [ii] rov. (7.7). Změnu konstitutivního vztahu lze provést v procedure *props* (assembly.py).
- **P** [Pa] - Zadává v čase konstatní zatížení na píst. Pokud bude třeba, v proceduře *p\_load* (assembly.py) lze předeepsat libovolnou časovou závislost tlaku na píst.
- **method** - Přepínač pro metodu integrace v čase. Hodnota `method = 2` pro Runge-Kutta čtvrtého řádu, jinak Newton prvního řádu. (Pro Newtonovu metodu je potřeba výrazně jemnější časový krok.)
- **echo** - Přepínač pro zobrazení průběhu výpočtu. Hodnota `echo = 1` pro výpis průběhu výpočtu, `echo = 1` pro velmi podrobný výpis a `echo = 0` bez výpisů.
- **xls** - přepínač pro export proměnných do MS-Excell tabulky. Hodnota `xls = 1` zapne export, `xls = 0` bez exportu. Pro tuto možnost je potřeba nainstalovat modul *pyExcelerator* viz. *Instalace*.
- **html** - přepínač pro export výsledků v předdefinované formě do HTML formátu. Hodnota `xls = 1` zapne export, `xls = 0` bez exportu.

## 2.1.3 Výstup

Základním výstupem je binární soubor, který je možné prohlížet pomocí programu *run\_POST*. Obsahuje následující proměnné

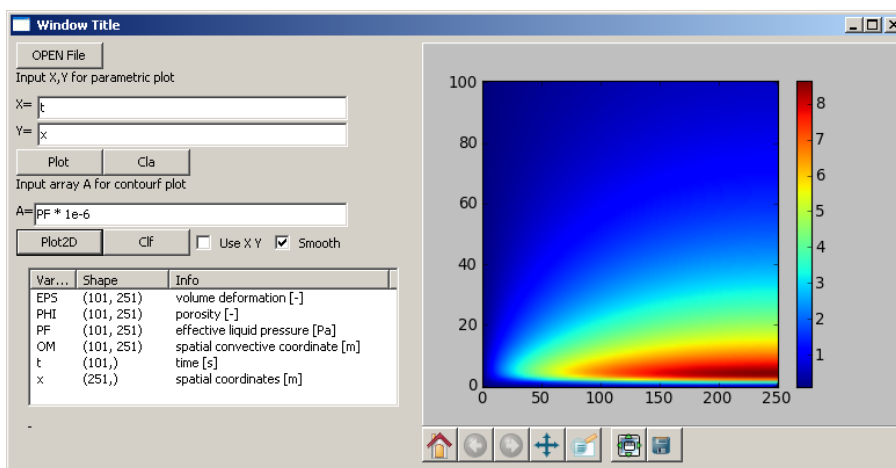
- **EPS** - objemová defomace [1];  $EPS(t, x)$  - pole velikosti (tsteps+1, nel+1)
- **PHI** - porozita [1];  $PHI(t, x)$  - pole velikosti (tsteps+1, nel+1)
- **PF** - efektivní tlak tekuté fáze [Pa];  $PF(t, x)$  - pole velikosti (tsteps+1, nel+1)
- **OM** - zobecněna souřadnice  $\omega$  [m];  $OM(t, x)$  - pole velikosti (tsteps+1, nel+1)
- **t** - čas [s]; pole velikosti (tsteps+1, 1)
- **x** - souřadnice místa (pozice uzlů) [m]; pole velikosti (nel+1, 1)
- **P** - velikost síly zatěžující píst [Pa]; pole velikosti (tsteps+1, 1)

## 2.2 run\_POST

Program *run\_POST* souží k jednoduchému prohlížení výstupního souboru s výsledky. Po spuštění příkazem:

```
python run_POST.py
```

se objeví toto okno s jednoduchým grafickým rozhraním



Obrázek 2.1: Grafické rozhraní *run\_Post.py*

Dialog pro volbu datového souboru se objeví po kliknutí na `OPEN File`. Po úspěšném utvření souboru se objeví informace o načtených proměnných (jméno, rozměr pole, význam).

Var...	Shape	Info
EPS	(101, 251)	volume deformation [-]
PHI	(101, 251)	porosity [-]
PF	(101, 251)	effective liquid pressure [Pa]
OM	(101, 251)	spatial convective coordinate [m]
t	(101,)	time [s]
x	(251,)	spatial coordinates [m]

Obrázek 2.2: Informace o proměnných

### 2.2.1 PLOT XY

Různé křivkové závislosti lze zobrazit pomocí tohoto formuláře:

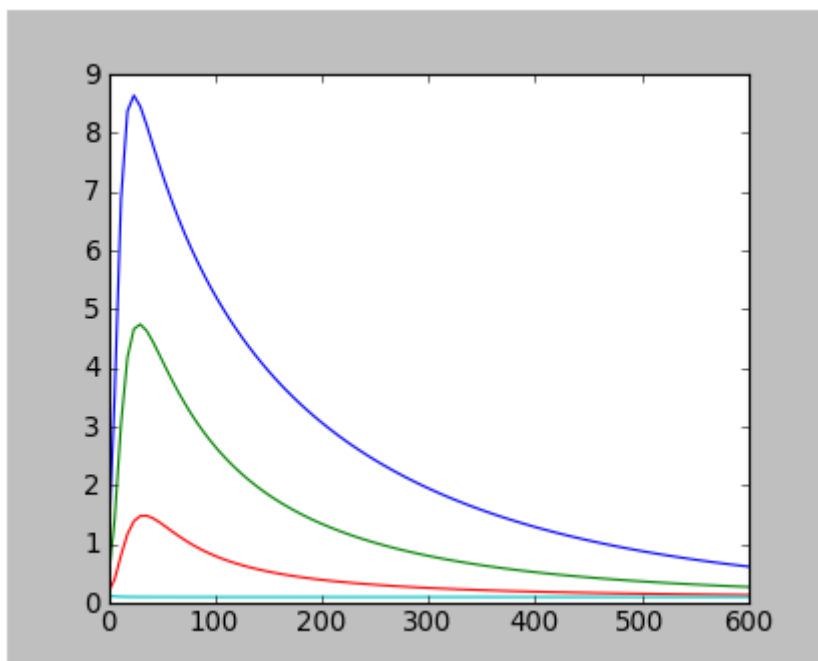
Input X,Y for parametric plot

X=

Y=

Obrázek 2.3: Plot XY - Formulář pro zadání grafu

Syntaxe zadávání proměnných je shodná se syntaxí pythonu, viz Příklady. Je možné zobrazení více křivek. Tlačítka `Plot` a `cla` jsou pro zobrazení (přidání) křivky a smazání grafu.



Obrázek 2.4: Plot XY - Příklad grafu

## 2.2.2 PLOT 2D

Pole, nebo část pole je možné zobrazit v barevných konturách pomocí tohoto formuláře:

Input array A for contourf plot

A=

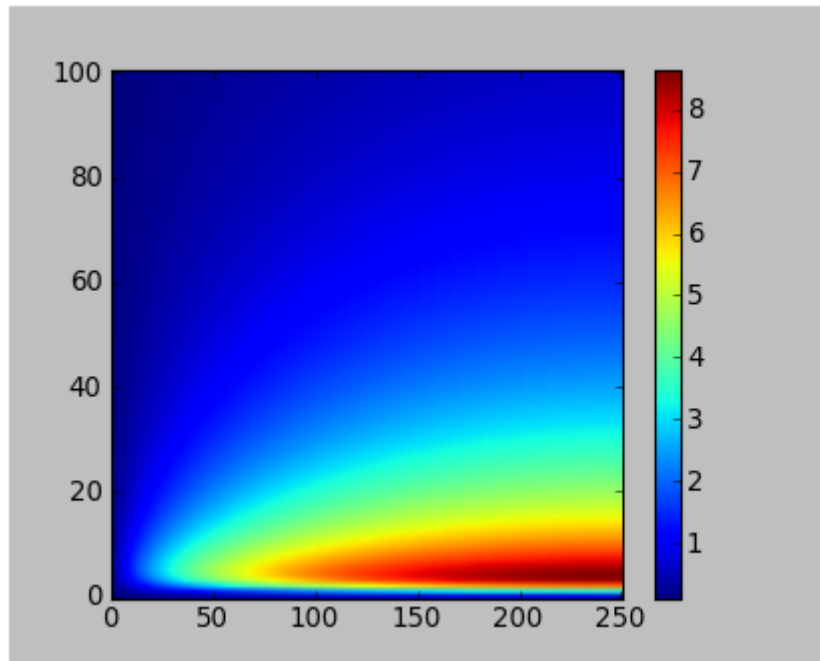
Use X Y
  Smooth

Obrázek 2.5: Plot2D - Formulář pro zadání grafu

Tlačítka `Plot2D` a `cla` jsou pro zobrazení a smazání grafu. Možnost `Smooth` je pro plynulé přechody barev. Pokud není použito volby `Use XY` je na osách zobrazen index pole. Pokud je použita volba `Use XY`, je potřeba zadat do políček `X=` a `Y=` vektory odpovídající velikosti. Jejich hodnoty se zobrazí na osách. Pozor, zadání je poněkud neintuitivní: `X=` pro první index pole - tomu odpovídá svislá osa a `Y=` pro druhý index pole - tomu odpovídá vodorovná osa. V další verzi programu bude tento problém odstraněn.

## 2.2.3 Grafické nástroje

Pro zoom, a uložení grafů lze použít základní nástroje na liště pod grafem.



Obrázek 2.6: Plot2D - Příklad grafu



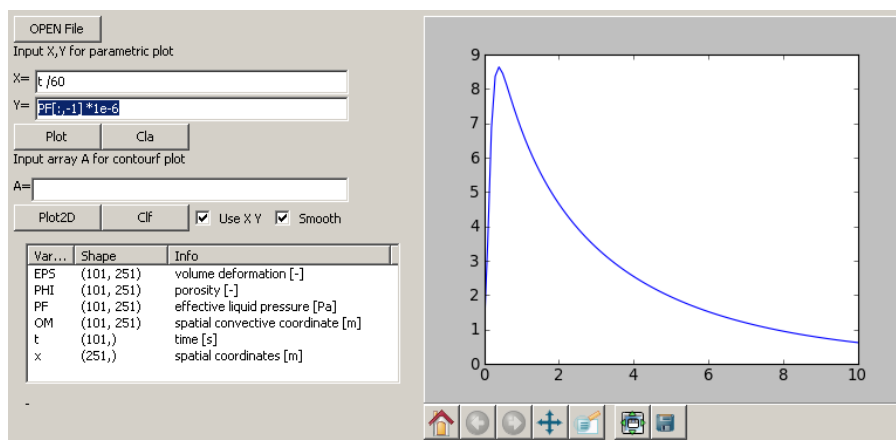
Obrázek 2.7: Grafické nástroje.

## 2.2.4 Příklady

1. Časový průběh (v minutách) tlaku oleje (v MPa) pod pístem.

- $X = t / 60$ .
- $Y = PF[:, -1] * 1e-6$

Dvojtečka : značí všechny řádky pole. Index -1 vybere poslední sloupec, ten odpovídá pozici pístu.



Obrázek 2.8: Příklad 1

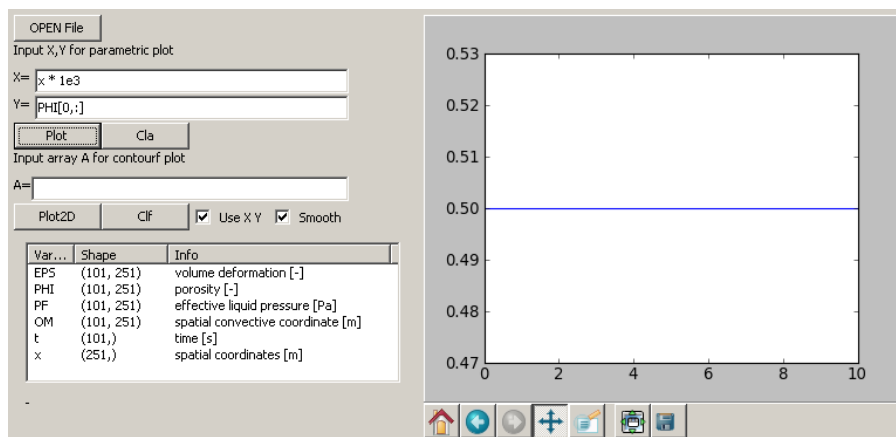
1. Průběh porozity (po délce koláče v mm) v  $t=0$ .

- $X = x * 1e3$



- $Y = \text{PHI}[0, :]$

Index 0 vybere první index (čas  $t=0$ ). Dvojtečka : značí všechny řádky pole.

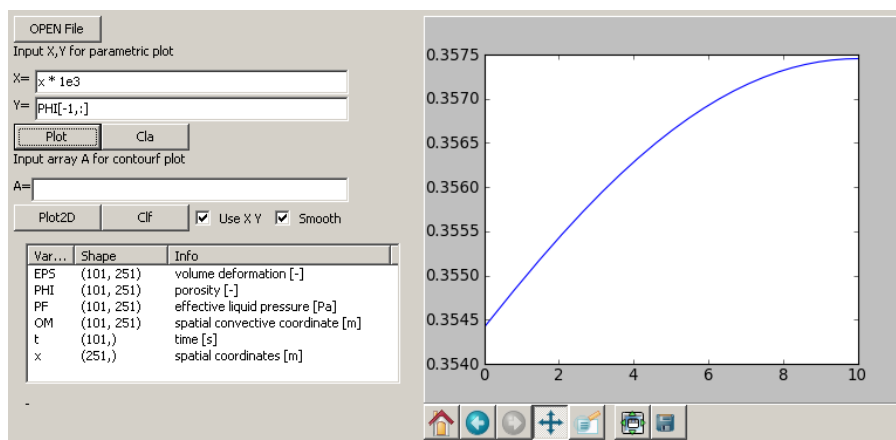


Obrázek 2.9: Příklad 2

1. Průběh porozity ve vzdálenosti od síta (mm) v  $t=t_{\max}$  (na konci simulace). Je zobrazena vzdálenost v nedeformované konfiguraci.

- $X = x * 1e3$
- $Y = \text{PHI}[-1, :]$

Index  $-1$  vybere poslední sloupec, ten odpovídá poslednímu indexu. Dvojtečka : značí všechny sloupce pole.



Obrázek 2.10: Příklad 3

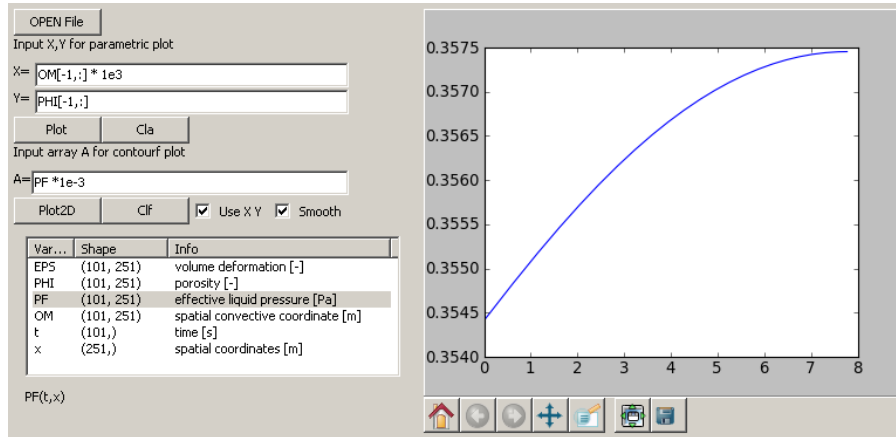
1. Průběh porozity ve vzdálenosti od síta (mm) v  $t=t_{\max}$  (na konci simulace). Je zobrazena vzdálenost v aktuální (deformované) konfiguraci.

- $X = x * 1e3$
- $Y = \text{PHI}[-1, :]$

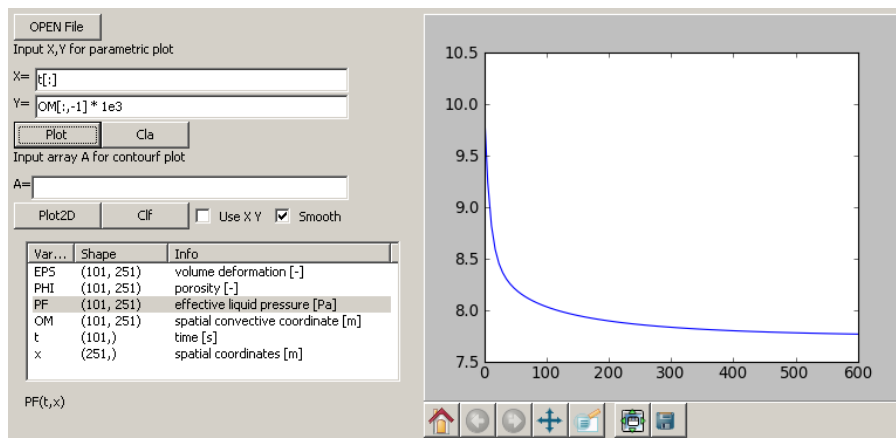
Index  $-1$  vybere poslední sloupec, ten odpovídá poslednímu indexu. Dvojtečka : značí všechny sloupce pole.

1. Pozice pístu (mm) v čase (s).

- $X = t$
- $Y = \text{OM}[:, -1] * 1e3$



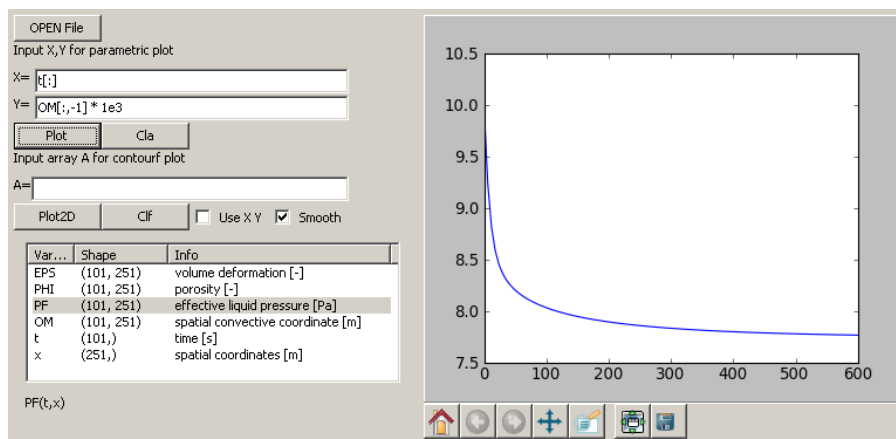
Obrázek 2.11: Příklad 4



Obrázek 2.12: Příklad 5

1. Průběh tlaku oleje. Na svislé ose je čas v [s], na vodorovné vzdálenost od síta v nedeformované konfiguraci (mm).

- $X = t$
- $Y = x * 1e3$
- $A = PF * 1e-3$



Obrázek 2.13: Příklad 6

---

# Dokumentace

---

## 3.1 assemble.py

**Rside** ( $C, D$ )

Sestavení prave strany

**Parametry**

- $C$  – vektor hodnot  $C$  v uzlech
- $D$  – delka mezi uzly

**Vrací Q** vektor prave strany

**bands\_k** ( $Ae, D$ )

Sestavení matice  $K$

**Parametry**

- $Ae$  – vektor hodnot  $A$  pro elementy o rozmeru  $(N,)$
- $D$  – delka mezi uzly

**Vrací Kd, Kv**

**Proměnná Kd** hodnoty na hlavni diagonale

**Proměnná Kv** vedlejsi diagonala

**bands\_m** ( $Be, D$ )

Sestavení matice  $M$

**Parametry**

- $Be$  – vektor hodnot  $A$  pro elementy o rozmeru  $(N,)$
- $D$  – delka mezi uzly

**Vrací Kd, Kv**

**Proměnná Md** hodnoty na hlavni diagonale

**Proměnná Mv** vedlejsi diagonala

**epsdot** ( $parms, eps, pf, t$ )

Casova derivace objemove deformace

**Parametry**

- $eps$  – objemova deformace
- $pf$  – tlak oleje

- $t$  – čas

**Vrací edot** o stejnem rozmeru jako  $eps$  a  $pf$

**p\_load** ( $parms, t$ )

Definuje zatizeni na pist. (Zatizeni lze zadat zavisle na case  $t$ )

**Parameter**  $parms$  – parametry ulohy

**Vrací P** zatizeni na pist [Pa]

**props** ( $parms, eps$ )

Tato procedura vytvori vektory parametru odpovidajici deformovane konfiguraci dane vektorem  $eps$ . Tj. tyto veliciny:

- zakladni materialove parametry
- hodnoty permeability podle empirickeho modelu Tiller-Yeh, tj. podle [ii] rovnice (7.2) a (7.7)
- vztah mezi zatizenim pevne faze a jeho deformaci (generalize Voigt) podle [i] rov. (9) nebo [ii] rov. (5.5)
- parametry sita **Alpha** a **pOkoli**
  - pro hodnotu  $\alpha=0$  je na situ nastavena Dirichletova podminka  $p_f(t, x = 0) = 0$
  - pro hodnotu  $\alpha>0$  je na situ nastaven tok  $Flux = \Lambda * (p_f - p_{Okoli})$

**Vstup** Parametr  $eps$  (objemova deformace) jako radek delky  $N$

**Vystup** Vraci nasledujici parametry (radky stejne delky  $N$ )

- Phi0 [1] ... pocatecni poresita
- mu [Pa s] ... dynamicka viskozita tekute faze (oleje)
- gamma [1] ... parametr analogicky s Biotovym parametrem
- eta [Pa s] ... viskozita pevne faze
- E [Pa] ... objemovy modul pruznosti pevne faze
- Ee [Pa] ... viz clen F(e) rovnice (9) v clanku [1]
- eta [Pa s] ... viskozita pevne faze
- Alpha [?] ... parametr sita
- pOkoli [Pa] ... parametr sita

## 3.2 solvers.py

**pde\_iter** ( $Piter, parms, EPSi, i$ )

**pde\_iter\_err** ( $Piter, parms, EPSi, i, Ut$ )

**pde\_solver\_P** ( $parms$ )

Resic, ktery iterativne hleda odpovidajici profil zatizeni k dane krivce  $Ut$ , ktera definuje vyteznost v zavislosti na case.

$$Ut(t) = 1./Phi0 * (h0-h(t))/h0$$

**Parametry**

- $parms$  – obsahuje vsechny parametry ulohy (viz class parametry.Parametry)
- $Ut$  – vyteznost v case

**Vraci**

- **EPS** - objemova deformace
- **PHI** - porozita
- **PF** - tlak tekutiny
- **X** - materialova souradnice
- **P** - vektor zatizeni

Vsechny vystupni promenne jsou pole o rozmerech  $tsteps+1, nel+1$ . Tj. obsahuji hodnoty v mistech prvko-  
vych uzlu a vsech casovych hladinach vcene  $t=0$ .

**pf\_solver** (*parms, eps, t, D*)

Resic pro rozlozeni tlaku tekute faze (oleje)

**Parametry**

- *eps* – objemova deformace
- *t* – cas

**Param** D velikost elementu

**Vrací pf** plak v [Pa]

Okrajove podminky jsou nastaveny v zavislosti na parametru Alpha.

**t\_diag\_sym** (*d, v, b, n*)

Resi soustavu  $Ax = b$  s tridiagonalni symetrickou matici A a pravou stranou R.

**Parametry**

- *d* – diagonala A
- *v* – vedlejsi diagonala A
- *b* – prava strana

**Vrací x** reseni soustavy

### 3.3 fortran\_test.py

**f\_test** ()

Tato procedura testuje funkcnost rychleho resice soustavy rovnic napsaneho ve fortranu. Vraci hodnotu 1, pokud je vse ok, jinak 0.

### 3.4 parametry.py

**class Parametry** (*parms\_dic*)

This class is used to hold, read from file and save to file parameters given in dictionary *parms\_dic*.

*Example:*

```
p_dic={'Time': 10.0, 'Method': 'linear'}
p= Parametry(p_dic)
p.write('data.txt')
```

This would creat a file *data.txt* containing two lines

```
Time = 10.0 Method = 'linear'
```

To read it back define a dictionary defining parameter names and reload it:

```
p_dic2={'Time':None, 'Method':None}
p2= Parametry(p_dic2)
p2.read('data.txt')
```

To get a parameter value 'Time':

```
t = p.get('Time')
```

Comments in the file:

```
@# Any line starting with @# is ignored
Time = 10.0 @# This would also work
```

### Flags

Variable *self.parms\_flag* is a dictionary holding the same keys like *self.parms\_dic*. Values are False or True. Line of the inputfile starting with asterix will set the flag to True:

```
*Time = 10.0 # Then .get_flag('Time') returnss True
```

#### **clear\_values()**

Set all values to None.

#### **get(key)**

#### **get\_empty()**

Report all keys with None values.

#### **get\_flag(key)**

#### **get\_marked()**

List of parameters values "marked" with asterix

#### **get\_marked\_names()**

List of parameters names "marked" with asterix

#### **read(filename)**

Read values form file

#### **set\_marked(x0)**

List of parameters values "marked" with asterix

#### **write(filename)**

Write values to file

## 3.5 std\_report.py

### **class HTMLreport()**

This class is used to generate simple HTML code containing simulation results.

#### **add\_date(text, tag)**

To add actual date to the report.

*Example:*

```
report.add_date("Date: ", ['code'])
```

#### **add\_fig\_xy(x, y, style)**

To add a simple xy\_plot (figure) to the report.

#### **add\_fig\_xyA(x, y, A, style)**

To add a contour plot as a figure to the report.

**add\_fig\_xy\_leg** (*x, y, style*)

To add a xy\_plot with legend (figure) to the report.

**add\_parameters** (*parms, tag*)

To add complete list of parameters in the report.

**add\_text** (*text, tag*)

To add some text to the report.

#### Parametry

- *text* – string
- *tag* – is a list of html tags (strings) that will bound the text

Example:

```
report.add_text("@textless[]p@textgreater[]", [])
report.add_text("My Text.", [])
report.add_text("@textless[]/p@textgreater[]", [])
```

this would be the same like:

```
report.add_text("My Text.", ['p'])
```

**make\_report** (*htmlfile, title, EPS, PHI, PF, PS, X, x, t, parms*)

In this method is defined the report format.

**save** (*filename*)

To write the file to disk.

## 3.6 xls\_export.py

**class XLSfile** ()

This is a simple class to export numpy.ndarrays in MS-Excel file format. The array must be vector or an matrix. Each vector is placed in separate list. Some information given by *info\_list* is placed at first lines.

**add\_array** (*sheet\_name, array, info\_list*)

This will add an *array* to a new list called *sheet\_name* (Sheet names can not repeat within one file). You can add a list *info\_list* giving information about the exported variable.

**save** (*filename*)

To write the xls file.

## 3.7 fortran.t\_diag\_sym\_f

**t\_diag\_sym\_f** (*d, v, b, n*)

This module 't\_diag\_sym\_f' was auto-generated with f2py (version:2) from the FORTRAN subroutine *t\_diag\_sym\_f.for*. The code function is equal to *solvers.t\_diag\_sym(d, v, b, n)*, i.e. solving eq.  $Ax = b$ .

#### Parametry

- *d* – main diagonal of A
- *v* – upper diagonal (equal to lower diag.)
- *b* – right hand side
- *n* – dimension of A

**Vrací x** solution vector x (length n)





---

# Indices and tables

---

- *Index*
- *Rejstřík modulů*
- *Vyhledávací stránka*



---

# Literatura

---

- [i] Očenášek J., Voldřich J. *Mathematical modeling of a biogenous filter cake and identification of oilseed material parameters*, Applied and Computational Mechanics 2009 (in press).
- [ii] Voldřich J., *Matematické modelování oddělujícího lisování olejin v lisech s lineární a šnekovou geometrií (teorie)*, Výzkumná zpráva NTC ZČU v Plzni, č. NTC 01-05/09, 2009.



---

# Index

---

- add\_array() (metoda xls\_export.XLSfile), 17
- add\_date() (metoda std\_report.HTMLreport), 16
- add\_fig\_xy() (metoda std\_report.HTMLreport), 16
- add\_fig\_xy\_leg() (metoda std\_report.HTMLreport), 16
- add\_fig\_xyA() (metoda std\_report.HTMLreport), 16
- add\_parameters() (metoda std\_report.HTMLreport), 17
- add\_text() (metoda std\_report.HTMLreport), 17
- assemble (module), 13
  
- bands\_k() (v modulu assemble), 13
- bands\_m() (v modulu assemble), 13
  
- clear\_values() (metoda parametry.Parametry), 16
  
- epsdot() (v modulu assemble), 13
  
- f\_test() (v modulu fortran\_test), 15
- fortran (module), 17
- fortran\_test (module), 15
  
- get() (metoda parametry.Parametry), 16
- get\_empty() (metoda parametry.Parametry), 16
- get\_flag() (metoda parametry.Parametry), 16
- get\_marked() (metoda parametry.Parametry), 16
- get\_marked\_names() (metoda parametry.Parametry), 16
  
- HTMLreport() (třída v std\_report), 16
  
- make\_report() (metoda std\_report.HTMLreport), 17
  
- p\_load() (v modulu assemble), 14
- parametry (module), 15
- Parametry() (třída v parametry), 15
- pde\_iter() (v modulu solvers), 14
- pde\_iter\_err() (v modulu solvers), 14
- pde\_solver\_P() (v modulu solvers), 14
- pf\_solver() (v modulu solvers), 15
- props() (v modulu assemble), 14
  
- read() (metoda parametry.Parametry), 16
- Rside() (v modulu assemble), 13
  
- save() (metoda std\_report.HTMLreport), 17
- save() (metoda xls\_export.XLSfile), 17
- set\_marked() (metoda parametry.Parametry), 16
- solvers (module), 14
- std\_report (module), 16
  
- t\_diag\_sym() (v modulu solvers), 15
- t\_diag\_sym\_f() (v modulu fortran), 17
  
- write() (metoda parametry.Parametry), 16
  
- xls\_export (module), 17
- XLSfile() (třída v xls\_export), 17