



NOVÉ TECHNOLOGIE
VÝZKUMNÉ CENTRUM
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

ODBOR MODELOVÁNÍ DEFORMAČNÍCH A DYNAMICKÝCH PROCESŮ

AUTORIZOVANÝ SOFTWARE

PERM_IDENT
SOFTWARE PRO IDENTIFIKACI PERMEABILITY
FILTRAČNÍHO KOLÁČE

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

Číslo projektu: FR-TI 1/369

Číslo výsledku: *NTC-SW-03-10*

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

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

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

PLZEŇ, LISTOPAD 2010

Jazyk výsledku: CZE

Hlavní obor: JR

Uplatněn: ANO

Název výsledku česky:

PERM_IDENT Software pro identifikaci permeability filtračního koláče

Název výsledku anglicky:

PERM_IDENT Software for identification of filtration cake permeability

Abstrakt k výsledku česky:

PERM_IDENT (verze 1.0) je program v jazyce Python, který byl vytvořen pro automatickou identifikaci permeability filtračního koláče. Program je schopný analyzovat sadu experimentálních dat oddělujícího lisování v lineární lisovací komoře a identifikovat permeabilitu koláče podle daného modelu. Obecná koncepce navíc dovoluje identifikovat některé další neznámé parametry modelu, jako například viskozitu pevné fáze. Pro vyjádření nelineární závislosti permeability na porózitě je použit přístup Tiller-Yeh a pro difúzní proudění intersticiální tekutiny je předpokládána platnost Darcyho zákona.

Abstrakt k výsledku anglicky:

A software tool PERM_IDENT (version 1.0) was developed for computer-aided identification of filtration cake permeability. Program is capable to analyze complete set of experimental data from cake filtration and consolidation tests according to a given mathematical model. Additionally, the general concept of the software tool allows identification of other unknown material parameters like solid phase viscosity. Nonlinear relation of the permeability and porosity is modeled using the Tiller-Yeh approach and the (intrinsic) liquid flow within pores of the solid matrix is assumed to follow the Darcy's law.

Klíčová slova česky:

permeabilita; filtrace; konsolidace; porézní médium; Darcyho zákon

Klíčová slova anglicky:

permeability; filtration; consolidation; porous media; Darcy's law

Vlastník výsledku: *Západočeská univerzita v Plzni*

IČ vlastníka výsledku: *49777513*

Stát: *Česká republika*

Lokalizace: http://www.zcu.cz/ntc/vysledky/sw/ntc_sw_03_10.html

Licence: *ANO*

Licenční poplatek: *NE*

Ekonomické parametry: *Efektivní identifikace materiálových parametrů přispívá k zvýšení kvality výroby, hodnoty produktů a ekonomických zisků*

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*

PERM_IDENT Dokumentace

Vydání 1.0

Jan Očenášek

03.01.2011

Obsah

1 Úvod	3
1.1 Teoretický úvod	3
1.2 Instalace	4
2 Tutorial	5
2.1 run_IDENT	5
3 Dokumentace	9
3.1 assemble.py	9
3.2 solvers.py	10
3.3 fortran_test.py	11
3.4 optfun.py	11
3.5 parametry.py	11
3.6 fortran.t_diag_sym_f	12
4 Indices and tables	15
Literatura	17
Index	19

PERM_IDENT - SOFTWARE PRO IDENTIFIKACI PERMEABILITY FILTRAČNÍHO KOLÁČE

je program v jazyce **Python**, který byl vytvořen pro automatickou identifikaci permeability filtračního koláče. Program je schopný analyzovat sadu experimentálních dat oddělujícího lisování v lineární lisovací komoře a identifikovat permeabilitu koláče podle daného modelu.

Obecná koncepce navíc dovoluje identifikovat některé další neznámé parametry modelu, jako například viskozitu pevné fáze.

Použitý matematický model je popsán 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

PERM_IDENT je komplexní program, jehož hlavní (spustitelný) soubor je

- `run_IDENT.py`

Program umožňuje zpracování experimentálních dat z oddělujícího lisování v lineární lisovací komoře. Je možné analyzovat ustálené i dynamické zkoušky nebo jejich kombinace a automaticky identifikovat zvolené materiálové parametry. Přesto, že původní účel programu bylo identifikovat pouze permeabilitu, je možné identifikovat i jiné parametry, např. viskozitu skeletu. Jádrem programu je iterativní multiparametrický optimalizační proces, tj. je možné identifikovat více parametrů zároveň, ovšem vždy s ohledem na charakter experimentálních dat.

Část kódu, která provádí eliminaci pásové matice, je navíc k dispozici také v jazyce *fortran*. Tento kód je připojen v binární kompilované podobě a umožňuje přibližně trojnásobnou rychlost řešení problému. Ovšem při přechodu na jiný operační systém není zaručena kompatibilita kompilovaného kódu a kompilaci je třeba provést znovu nebo použít kód v pythonu.

1.1 Teoretický úvod

Matematický model lisovacího procesu je popsán soustavou parciálních diferenciálních rovnic pro objemovou deformaci pevné fáze $\epsilon = \epsilon(t, \omega)$ a tlak intersticiální tekutiny $p_f = p_f(t, \omega)$

$$\eta \frac{\partial \epsilon}{\partial t} + \mathcal{F}(\epsilon) - \frac{\gamma}{\Phi(\epsilon)} p_f = -P(t) + p_f \quad (1.1)$$

$$-\frac{\partial}{\partial \omega} \left(\frac{K(\Phi(\epsilon))}{\mu} (1 - \Phi(\epsilon)) \frac{\partial p_f}{\partial \omega} \right) + \frac{1 + \gamma/\Phi(\epsilon)}{\eta(1 - \Phi(\epsilon))} p_f = \frac{1}{\eta} \frac{P(t) + \mathcal{F}(\epsilon)}{1 - \Phi(\epsilon)}.$$

Pro odvození a podrobný popis viz rovnice (13) v [i] nebo rovnice (6.10) v [ii].

Při řešení je možné zvolit mezi dvěma typy okrajových podmínek.

1. Jednodušší podmínka (pro ideální filtr) uvažuje nulový tlak v místě síta $p_f(t, 0) = 0$. Dále $\frac{\partial p_f}{\partial \omega}(t, (1 - \Phi_0)h_0) = 0$ a $\epsilon(0, \omega) = 0$.
2. Pro reálnou propustnost filtru Λ je podmínka $\frac{\partial p_f}{\partial \omega}(t, 0) = \Lambda(p_f - p_{okolí})$. Dále stejně $\frac{\partial p_f}{\partial \omega}(t, (1 - \Phi_0)h_0) = 0$ a $\epsilon(0, \omega) = 0$.

Pozn.: Propustnost filtru Λ je v programu *LLSID* značena jako *A*, ve vstupním souboru jako *Alpha*.

Soustava je řešena Galerkinovou metodou s lineárním prvkem a explicitní integrací v čase. Pro časovou integraci je k dispozici Newtonova metoda 1. řádu a Runge-Kutta 4. řádu.

1.2 Instalace

Pro správnou funkci programu je doporučena instalace Python verze 2.6.

Závislost na dalších modulech:

- numpy
- scipy
- re
- wx
- matplotlib

Instalaci pro Windows lze realizovat například pomocí distribuce [Enthought Python Distribution - EPD6](#), nebo [PythonXY](#) (Scientific-oriented Python Distribution based on Qt and Eclipse). .

Tutorial

2.1 run_IDENT

Program *run_IDENT* je určen k identifikaci jednoho nebo více materiálových parametrů z experimentálních měření. Iterativním algoritmem se hledají hodnoty parametrů, pro které se výsledky modelu nejlépe přiblíží experimentálním hodnotám. Úloha je formulována jako hledání minima chybové funkce (multiparametrická optimalizace), uživatel tedy musí definovat chybovou funkci pomocí *optfun.error_func*.

error_func (x0, parms, output file) : ()

Parametry

- *x0* – hodnoty optimalizovaných parametrů z poslední iterace
- *parms* – hodnoty parametrů (ze vstupního souboru)
- *outputfile* – jméno výstupního souboru

Typ navracené hodnoty reálné číslo

Spuštění programu se provede příkazem (z příkazové řádky):

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

nebo krátce:

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

bez parametrů (defaultní vstupní a výstupní soubor):

```
python run_IDENT.py
```

je stejné jako:

```
python run_IDENT.py --input input.txt --output results_ident.txt
```

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. Symbol * před jménem parametru označuje, že tento parametr bude identifikován. Zadaná hodnota přitom určuje počáteční odhad.

Příklad vstupního souboru pro identifikaci parametrů *eta*, *alpha0* a *C* je následující:

```
#
#   Vstupni soubor pro program 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 konstitutivniho vztahu pro permeabilitu [m/kg]
*C=     27.      # exponent C konstitutivniho 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 pro vypis prubehu vypoctu
```

2.1.2 Popis parametrů

- **D** [m], **nel** [1]- Velikost elementu, počet elementů. Počáteční tloušťka filtračního koláče je tedy $h_0 = D * nel$.
- **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 (skeletu)
- **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 Tiller a Yeha, viz [ii] rov. (7.7). Změnu konstitutivního vztahu lze provést v procedure *props* (assembly.py).
- **P** [Pa] - Zadává v čase konstantní zatížení na píst. Pokud bude třeba, v proceduře *p_load* (assembly.py) lze předepsat 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 = 0` bez výpisů.

2.1.3 Výstup

Grafický nebo textový výstup zásadně závisí na formě experimentálních dat. Proto je formát výstupu definován také v *optfun.error_func*.

Defaultně je přednastaven výstup do textového souboru, který ukazuje v každé iteraci odhad parametrů i hodnotu definované chybové funkce. Následuje příklad výstupu:

```

Marked variables: ['C', 'alpha0', 'eta']
Initial guess: [27.0, 30000000000.0, 1800000000.0]
Iterations:
[ 2.70000000e+01  3.00000000e+10  1.80000000e+09]  ERR:  1.22956
[ 2.83500000e+01  3.00000000e+10  1.80000000e+09]  ERR:  1.42456
[ 2.70000000e+01  3.15000000e+10  1.80000000e+09]  ERR:  1.26836
[ 2.70000000e+01  3.00000000e+10  1.89000000e+09]  ERR:  1.22740
[ 2.56500000e+01  3.10000000e+10  1.86000000e+09]  ERR:  1.05116
[ 2.43000000e+01  3.15000000e+10  1.89000000e+09]  ERR:  0.85460
[ 2.52000000e+01  2.95000000e+10  1.92000000e+09]  ERR:  0.94180
[ 2.40000000e+01  3.06666667e+10  2.00000000e+09]  ERR:  0.78583
    
```

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 ($parms$)

Hlavni resic soustavy pde

Parameter $parms$ – obsahuje vsechny parametry ulohy (viz class parametry.Parametry)

Vraci

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

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 optfun.py

error_func (*x0, parms, outputfile*)

Definuje funkci chyby (odchylky) experimentálních dat od simulace. Minimalizaci této funkce se identifikují parametry modelu. Viz *run_FIT*.

Parametry

- *x0* – hodnoty optimalizovaných parametru z poslední iterace
- *parms* – hodnoty parametru (ze vstupního souboru)
- *outputfile* – jméno výstupního souboru

Typ navrácené hodnoty reálné číslo (velikost chyby)

3.5 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': 'lienar'}
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

3.6 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 fuction 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

assemble (module), 9

bands_k() (v modulu assemble), 9
bands_m() (v modulu assemble), 9

clear_values() (metoda parametry.Parametry), 12

epsdot() (v modulu assemble), 9
error_func() (v modulu optfun), 11

f_test() (v modulu fortran_test), 11
fortran (module), 12
fortran_test (module), 11

get() (metoda parametry.Parametry), 12
get_empty() (metoda parametry.Parametry), 12
get_flag() (metoda parametry.Parametry), 12
get_marked() (metoda parametry.Parametry), 12
get_marked_names() (metoda parametry.Parametry),
12

optfun (module), 5, 11

p_load() (v modulu assemble), 10
parametry (module), 11
Parametry() (třída v parametry), 11
pde_iter() (v modulu solvers), 10
pde_iter_err() (v modulu solvers), 10
pde_solver() (v modulu solvers), 10
pf_solver() (v modulu solvers), 11
props() (v modulu assemble), 10

read() (metoda parametry.Parametry), 12
Rside() (v modulu assemble), 9

set_marked() (metoda parametry.Parametry), 12
solvers (module), 10

t_diag_sym() (v modulu solvers), 11
t_diag_sym_f() (v modulu fortran), 12

write() (metoda parametry.Parametry), 12