Instructions for the folder: https://github.com/hfuselier/PDM/tree/master/python

Héloïse Fuselier
heloise.fuselier@epfl.ch

University of Minnesota — January 15, 2020

Introduction

This document is the instruction manual of the program presented in the folder : https://github.com/hfuselier/PDM/tree/r
Please read carefully before use.

The program provides a fitting solution and representations of the failure surface in a three-dimensional space and in the π-plane for the Paul-Mohr-Coulomb failure criterion. Its development was thought to enable the fitting of the three-parameter, six-parameter and simplified Paul-Mohr-Coulomb criterion. The present document descibes the program and explains its use to evaluate the Paul-Mohr-Coulomb failure criterion and to compare it with the Mohr-Coulomb and Hoek-Brown failure criteria.

1 Files

The files can be divided in the following groups, depending on their use.

- Importation and loading of the rock data:
  - *load_data.py*
  - *convert.py*
- Construction of the planes forming the failure surface:
  - *pmc.py*
  - *mc.py*
  - *hb.py*
- Allocation of data points to P1 or P2 for the Paul-Mohr-Coulomb failure criterion:
  - *brute_force.py*
  - *brute_force_4p.py*
- Computation of the error for data allocation and of the mean standard deviation misfit:
  - *error_computation.py*
  - *error_computation_4p.py*
- Creation of the failure surface graphical respresentations in 3D and in the π-plane:
  - *Plot_3D_6-12-6.ipynb*
  - *Plot_one_plane.ipynb*

2 How to use the files and run the program ?

Depending the fitting we want to perform, a particular combinaison of files is needed. These combinaisons are presented in Table 1. In the next part of this section, the procedure used to complete the different fittings are detailed, to simplify the use of the program. The procedures are already implemented at the beginning of $Plot\_3D\_6-12-6.ipynb$ and $Plot\_one\_plane.ipynb$ files which can be directly used, by following the indications for each fitting made bellow, and the comments.

Table 1: Organization of the program

Application	Associated files
Paul-Mohr-Coulomb with three parameters	$pmc.py$, $brute\_force.py$, $error\_computation.py$, $Plot\_one\_plane.ipynb$
Paul-Mohr-Coulomb with six parameters	$pmc.py$, $brute\_force.py$, $error\_computation.py$, $Plot\_3D\_6-12-6.ipynb$
Simplified Paul-Mohr-Coulomb	$pmc\_4p.py$, $brute\_force\_4p.py$, $error\_computation\_4p.py$, $Plot\_3D\_6-12-6.ipynb$
Mohr-Coulomb fitting with all data	$mc.py$, $error\_computation.py$, $Plot\_one\_plane.ipynb$
Hoek-Brown fitting with all data	$hb.py$, $error\_computation.py$, $Plot\_one\_plane.ipynb$
Mohr-Coulomb fitting with six data	$mc.py$, $error\_computation\_4p.py$, $Plot\_one\_plane.ipynb$
Hoek-Brown fitting with six data	$hb.py$, $error\_computation\_4p.py$, $Plot\_one\_plane.ipynb$

2.1 Paul-Mohr-Coulomb with three parameters

Importation of the files

The following sequence is used to import the required files:

```
import numpy as np
from load import load_data
from convert import convert
```

```
from pmc import *
from error_computation import *
```

Input

The following sequence is used to get rock data (position 13 in *load_data.py* is for Dunnville sandstone) from a ".txt" file:

```
data = load_data(13)
d = convert(data)
```

Fitting of the failure surface

The following sequence creates the fitting plane of the three-parameter Paul-Mohr-Coulomb criterion failure surface and displays the parameter values associate with the provided solution in format "V_0 ϕ_c ϕ_e":

```
P = Plane(data)
print(P.sol)
```

Computation of the mean standard deviation misfit

The following sequence computes the mean standard deviation misfit of the fitted failure surface:

```
S = standard_dev(P,'PMC')
```

Representation in the principal stress space and in the π-plane

The representation of the three-parameter Paul-Mohr-Coulomb failure surface in the principal stress space and the ϕ-plane is obtained by running the *Plot_one_plane.ipynb* file.

2.2 Paul-Mohr-Coulomb with six parameters

Importation of the files

The following sequence sis used to import the required files:

```
import numpy as np
from load import load_data
from convert import convert
from pmc import *
from brute_force import planes_def
```

Input

The following sequence is used to get rock data (position 13 in *load_data.py* is for Dunnville Sandstone) from a ".txt" file:

```
data = load_data(13)
d = convert(data)
```

Fitting of the failure surface

The following sequence creates the fitting planes of the six-parameter Paul-Mohr-Coulomb criterion failure surface, compute the mean standard deviation misfits and displays the parameter values associate with the provided solution for $P1$ and $P2$ in format "$V_0\ \ \phi_c\ \ \phi_e$" :

```
P1, P2, SP1, SP2 = planes_def(data,d)
print(P1.sol)
print(P2.sol)
print(SP1)
print(SP2)
```

Representation in the principal stress space and in the π-plane

The representation of the three-parameter Paul-Mohr-Coulomb failure surface in the principal stress space and the ϕ-plane is obtained by running the $Plot\_3D\_6-12-6.ipynb$ file.

2.3 Simplified Paul-Mohr-Coulomb

Importation of the files

The following sequence is used to import the required files:

```
import numpy as np
from load import load_data
from convert import convert
from pmc_4p import *
from brute_force_4p import planes_def_PMC
```

Input

The following sequence is used to get rock data (position 13 in *load_data.py* is for Dunnville Sandstone) from a ".txt" file:

```
data = load_data(13)
d = convert(data)
```

Fitting of the failure surface

The following sequence creates the fitting planes of the four-parameter Paul-Mohr-Coulomb criterion failure surface, compute the mean standard deviation misfits for the non-fitted data and displays the parameter values associate with the provided solution for $P1$ and $P2$ in format "$V_0\ \ \phi_c\ \ \phi_e$". The selected data to be fitted are stored in the "fit_pts" array:

```
fit_pts = np.concatenate([data[1:4,:],data[5:8,:]])
P1, P2, SP1, SP2 = planes_def_PMC(data,fit_pts)
print(P1.sol)
print(P2.sol)
print(SP1)
print(SP2)
```

Representation in the principal stress space and in the π-plane

The representation of the three-parameter Paul-Mohr-Coulomb failure surface in the principal stress space and the ϕ-plane is obtained by running the $Plot\_3D\_6-12-6.ipynb$ file. It should be noted that some adjusteements shoudl be made on this file to adapt it for the simplified Paul-Mohr-Coulomb criteria. This can be done by following the procedure described by the file comments.

2.4 Mohr-Coulomb failure surface fitting

Importation of the files

The following sequence is used to import the required files:

```
import numpy as np
from load import load_data
from convert import convert
from mc import *
from error_computation import *
```

Input

The following sequence is used to get rock data (position 13 in *load_data.py* is for Dunnville sandstone) from a ".txt" file:

```
data = load_data(13)
d = convert(data)
```

Fitting of the failure surface

The following sequence creates the fitting plane of the Hoek-Brown criterion failure surface and displays the parameter values associate with the provided solution in format "ϕ c C_0 K_p V_0" :

```
P = Plane_MC(data)
print(P.sol)
```

Computation of the mean standard deviation misfit

The following sequence computes the mean standard deviation misfit of the fitted failure surface:

```
S = standard_dev(P,'MC')
```

Representation in the principal stress space and in the π-plane

The representation of the Mohr-Coulomb failure surface in the principal stress space and the ϕ-plane is obtained by running the *Plot_one_plane.ipynb* file. It should be noted that some adjustments should be made on this file to adapt it to create plots of the Mohr-Coulomb failure surface. This can be done by following the procedure described by the file comments.

2.5 Mohr-Coulomb failure surface fitting with a limited amount of data

Importation of the files

The following sequence is used to import the required files:

```
import numpy as np
from load import load_data
from convert import convert
from mc import *
from error_computation_4p import *
```

Input

The following sequence is used to get rock data (position 13 in *load_data.py* is for Dunnville sandstone) from a ".txt" file:

```
data = load_data(13)
d = convert(data)
```

Fitting of the failure surface

The following sequence creates the fitting planes of the Mohr-Coulomb criterion failure surface, compute the mean standard deviation misfits for the non-fitted data and displays the parameter values associate with the provided solution in format "ϕ c C_0 K_p V_0". The selected data to be fitted are stored in the "fit_pts" array:

```
fit_pts = np.concatenate([data[1:4,:],data[5:8,:]])
P,S = planes_def_MC_HB(data,fit_pts,'MC')
print(P.sol)
print(S)
```

Representation in the principal stress space and in the π-plane

See section 2.4

2.6 Hoek-Brown failure surface fitting

Importation of the files

The following sequence is used to import the required files:

```
import numpy as np
from load import load_data
from convert import convert
from hb import *
from error_computation import *
```

Input

The following sequence is used to get rock data (position 13 in *load_data.py* is for Dunnville sandstone) from a ".txt" file:

```
data = load_data(13)
d = convert(data)
```

Fitting of the failure surface

The following sequence creates the fitting plane of the Hoek-Brown criterion failure surface and displays the parameter values associate with the provided solution in format "m C_0 V_0":

```
P = Plane_HB(data)
print(P.sol)
```

Computation of the mean standard deviation misfit

The following sequence computes the mean standard deviation misfit of the fitted failure surface:

```
S = standard_dev(P,'HB')
```

Representation in the principal stress space and in the π-plane

The representation of the Hoek-Brown failure surface in the principal stress space and the ϕ-plane is obtained by running the *Plot_one_plane.ipynb* file. It should be noted that some adjustments should be made on this file to adapt it to create plots of the Hoek-Brown failure surface. This can be done by following the procedure described by the file comments.

2.7 Hoek-Brown failure surface fitting

Importation of the files

The following sequence is used to import the required files:

```
import numpy as np
from load import load_data
from convert import convert
from hb import *
from error_computation_4p import *
```

Input

The following sequence is used to get rock data (position 13 in *load_data.py* is for Dunnville sandstone) from a ".txt" file:

```
data = load_data(13)
d = convert(data)
```

Fitting of the failure surface

The following sequence creates the fitting planes of the Hoek-Brown criterion failure surface, compute the mean standard deviation misfits for the non-fitted data and displays the parameter values associate with the provided solution in format "m C_0 V_0". The selected data to be fitted are stored in the "fit_pts" array:

```
fit_pts = np.concatenate([data[1:4,:],data[5:8,:]])
P,S = planes_def_MC_HB(data,fit_pts,'HB')
print(P.sol)
print(S)
```

Representation in the principal stress space and in the π-plane

See section 2.6