Documentation tscw_module

Thomas Simader

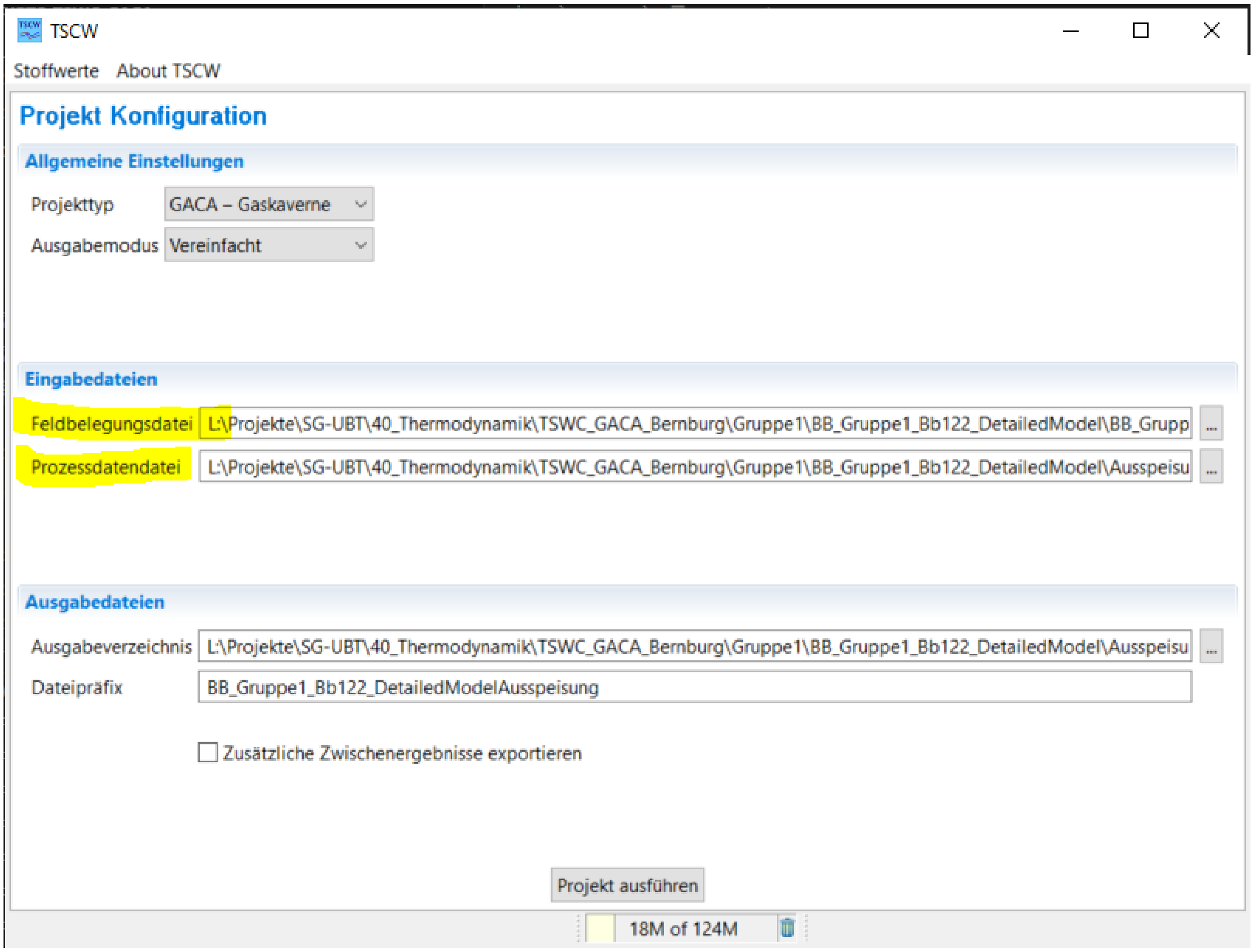
Mar 04, 2024

CONTENTS:

Wilkommen zur Dokumentation des *tscw_module*!

tscw_module ist eine Python-Bibliothek zur Erstellung von Eingabe- und Ausgabedaten aus der TSWC-Software.

Das `tscw_module` ist eine Package, welches in einer python-Umgebung Feldbelegungs- und Prozessdatendateien erstellen kann. Diese dienen als Input für die TSCW - Software. Im ersten Teil wird eine kleine Übersicht über den Workflow zu Erstellung der jeweiligen Dateien gegeben.´ Im zweiten Teil wird vorgestellt, wie TSCW-Output Dateien prozessiert werden können.



CHAPTER

ONE

INSTALLATION

1.1 1. Installation von Python

`tscw_module` ist eine Python Bibliothek. Python ist eine interpretierte, objektorientierte Hochsprache mit dynamischer Semantik. Die integrierten Datenstrukturen, dynamische Typisierung und Bindung machen Python attraktiv für schnelle Anwendungsentwicklung und als Skript- oder Klebesprache. Die einfache Syntax betont Lesbarkeit und reduziert Wartungskosten. Python unterstützt Module und Pakete für Programmmodule und Code-Wiederverwendung. Der Python-Interpreter und die Standardbibliothek sind kostenlos und plattformunabhängig verfügbar und kann über www.python.org/downloads/ heruntergeladen werden. WICHTIG: Es wird empfohlen, bei der Installation die Option `"Add python.exe" to PATH` zu wählen. Dabei wird die ausführbare Datei nach der Installation in die PATH-Variable aufgenommen Python kann direkt von der Windows Konsole aufgerufen werden mit dem Befehl `python` aufgerufen werden. Alternattiv kann Python auch später manuell in die Umgebungsvariable PATH aufnehmen (Admin-Rechte erforderlich). Als Quelltext-Editor empfiehlt sich `Visual Studio Code` von Microsoft (kostenlos), welcher über code.visualstudio.com/download/ heruntergeladen werden kann.

1.2 2. Installation des Moduls

Zu dem gwünschten Arbeitsordner in der Windowskonsole wechseln:

```
$ cd $deinPfad
```

Erstellung einer virtuellen Umgebung mit anschließender Aktivierung:

```
$ python -m venv .tscw_env
$ .tscw_env\Scripts\activate
```

Installieren des `tscw_module`:

```
$ pip install tscw_module
```

Im VS Code mit `Str + P` den Interpreter aufrufen und `.tscw_env` auswählen.

Note: Falls VS Code zukünftig das `tscw_module` nicht erkennen sollte, wird empfohlen, den Interpreter in VS Code zu überprüfen. Zum installieren weiterer Packages via `pip install` muss die virtuelle Umgebung im entsprechen Pfad mit dem Befehl `.tscw_env\Scripts\activate` aktiviert werden. Zum Deaktivieren kann der Befehl `deactivate` benutzt werden.

Note: Es sollte unter https://pypi.org/project/tscw-module/ regelmäßig nach Updates gecheckt werden. Falls es ein Versionsupdate gibt, kann dieses in dem Windos-Terminal mit folgendem Befehl installiert werden:

```
$ .tscw_env\Scripts\activate
```

```
$ pip install tscw_module --upgrade
```

CHAPTER

TWO

TSCW - INPUT - ERSTELLUNG

2.1 Felddaten

Um eine Felddaten-Datei zu erstellen, muss die Klasse `GacaFieldData` erstellt und mit Daten belegt werden.

2.1.1 0. Importiere benötigte Module:

```
import numpy as np
from tscw_module import GacaFieldData, GacaProcessData
import matplotlib.pyplot as plt
```

2.1.2 1. Initialisiere GacaFieldData:

```
n_boreholes        = 1
n_fluid            = 1
delta_z            = 122.5    # [m] Dicke vertikal Schicht (äquidistant)
tvd_max_borehole   = 4*122.5  # [m] RS lzRT ()
medium_type_cavern = 'GAS'    #
medium_id_cavern   = 'Nordverbundgas' # Name der Substanz, welche in TSCW/Stoffwerte/Bibliothek definiert wurde

# Initialisiere GacaFieldData Class
Gaca_field = GacaFieldData(n_boreholes,n_fluid,tvd_max_borehole,delta_z, \
                           medium_type_cavern,medium_id_cavern)
```

2.1.3 2. Definition der radialen Stützstellen um das Bohrloch:

Die radialen Stützstellen werden in diesem Beispiel in dem array `radial_vector_borehole` definiert. Die Unterteilung sollte so erfolgen, dass alle revelanten Installationselemente in radialer Richtung erfasst werden.

```
radial_vector_borehole = np.array([0.0942, 0.1095, 0.1372, 0.1492, 0.2, 0.5, 1, 2, 4, 8, 16, 32, 64, 128]) # [m]
 # Manuell werden beliebig viele Sützstellen mit Agregatzuständen beleget.
 # Die restlichen sind automatisch 'SOLID'
Gaca_field.add_boreholeVector(radial_vector_borehole,['FLUID','SOLID'])
 # OPTIONAL: Kommentare der jeweiligen Stütztstelle beginnend ab dem ersten Element.
Gaca_field.add_radialComment(['GAS','STAHL','RRSF','STAHL','ZEM','Gebirge---->'])
```

2.1.4 3. Definition der Materialeigenschaften um das Bohrloch:

Geometrisch besteht das Bohrlochmodell aus einer Matrix, welche sich aus `Gaca_field.p_borehole` Zeilen und `Gaca_field.m_borehole` Spalten besteht. Diese müssen nun mit Stoffwerten versehen werden über die Funktion `add_materialProperty` hinzugefügt werden.

```
# Material Properties
Gaca_field.add_materialProperty(0,250,  # jeweils UK von Start und Ende angeben (range)
                              np.array([0.0000, 3.6000, 4.2000, 3.6000, 1.6000, 1.9200, 1.9200,   # heat_capacity * rho  [MJ/(m3K)]
                                        1.9200, 1.9200, 1.9200, 1.9200, 1.9200, 1.9200, 1.9200]),
                              np.array([0.000, 50.000, 0.500, 50.000, 1.000, 2.330, 2.330,        # thermal_conductivity [W/(m K)]
                                        2.330, 2.330, 2.330, 2.330, 2.330, 2.330, 2.330]),'Schicht 1') # Name

Gaca_field.add_materialProperty(350,500,
                              np.array([0.0000, 3.6000, 4.2000, 3.6000, 1.6000, 1.9500, 1.9500,
                                        1.9500, 1.9500, 1.9500, 1.9500, 1.9500, 1.9500, 1.9500]),
                              np.array([0.000, 50.000, 0.500, 50.000, 1.000, 5.500, 5.500,
                                        5.500, 5.500, 5.500, 5.500, 5.500, 5.500, 5.500]), 'Schicht 2' )
```

2.1.5 4. Temperatur und Neigung des Bohrlochs

```
temparature_bh = np.transpose(np.tile(np.array([8, 12, 17, 22]),(Gaca_field.m_borehole,1)))
Gaca_field.add_temperature(temparature_bh,'borehole')
Gaca_field.add_boreholeInclination('vertical')
```

2.1.6 5. Kaverneneigenschaften

```
rad_kav           = 30.3      # radius
v_kav             = 464366    # Volumen
# Höhe des Zylinders, wenn das Volumen aus einem Zylinder mit zwei aufgesetzten Halbkugel approximiert wird.
h                 = (v_kav - 4/3*np.pi*rad_kav**3) / (np.pi*rad_kav**2)
density_salt                  = 2170     # [kg/m3]
specific_heat_capacity_salt   = 900      # [J/(kg*K)]
heat_conductivity_salt        = 5.5      # [W/(m*K)]
height_cavern                 = h + 2*rad_kav    # [m]
volume_brine_equivalent       = 30000      # [m3]
radius_brine_level            = rad_kav     # [m]
refdepth_cavern               = tvd_max_borehole     # [m] RS
temperature_brine_equivalent = None # 30
pressure_cavern               =  9.8       # [MPa]
```

(continues on next page)

(continued from previous page)

```


# radiale Stützstellen in der Kaverne
radial_vector_cavern   = np.array([rad_kav, 35.18, 39.51, 43.85, 48.19, 52.52, 56.86, 61.
↪19,
                                  65.53, 69.87, 74.23, 78.54, 82.88, 87.22, 91.56, 95.90,
                                  100.24, 104.58, 109.92, 113.26, 117.60, 121.90, 126.24,
                                  135.00, 140.00, 145.00, 150.00])

# temperatur
temparature_cav = np.array([22.00]*radial_vector_cavern.shape[0])   # const temperature


# Füge die Daten der Klasse hinzu
Gaca_field.add_cavernVector(radial_vector_cavern)
Gaca_field.add_temperature(temparature_cav,'cavern')
Gaca_field.add_cavernCharacteristics(refdepth_cavern, density_salt, specific_heat_
↪capacity_salt,heat_conductivity_salt,
                                    height_cavern, volume_brine_equivalent, radius_brine_
↪level,pressure_cavern, temperature_brine_equivalent)
```

2.1.7 6. Export

```
save_folder     = r'L:\Projekte\SG-UBT\40_Thermodynamik\TSCW_GACA_Bernburg\Gruppe1'
project_name    = 'BB_Bb122'
Gaca_field.export_fieldData(save_folder,project_name, True)
```

Beim exportieren wird in der Konsole folgendes angezeigt:

```
Run sucessfull
L:\Projekte\SG-UBT\40_Thermodynamik\TSCW_GACA_Bernburg\Gruppe1\BB_Gruppe1_Bb122\BB_
↪Gruppe1_Bb122_gaca.fd.txt
Backend TkAgg is interactive backend. Turning interactive mode on.
Exported L:\Projekte\SG-UBT\40_Thermodynamik\TSCW_GACA_Bernburg\Gruppe1\BB_Gruppe1_Bb122\
↪BB_Gruppe1_Bb122_fd.pickle
```

Die Felddaten-Datei `BB_Gruppe1_Bb122_gaca.fd.txt` ist:

```
NUMBER_BOREHOLES     1
N_FLUID      1
M_BOREHOLE     14       # (M)
P_BOREHOLE     4        # (P)
DL    122.5000          # [m]
M_CAVERN      27        # (MK)
MEDIUM_TYPE_CAVERN    GAS
MEDIUM_ID_CAVERN      Nordverbundgas
DEPTH_CAVERN 490.00   # Referenztiefe fuer Druck

RADIAL_VECTOR_BOREHOLE # [m]
#GAS STAHL    RRSF    STAHL   ZEM     Gebirge---->
0.0942        0.1095  0.1372  0.1492  0.2000  0.5000  1.0000  2.0000  4.0000  8.0000  16.
```

(continues on next page)

(continued from previous page)

```
↪0000 32.0000 64.0000 128.0000

COLUMN_CHARACTER_BOREHOLE # [/] der Radialelemente um die Bohrung (M Werte)
FLUID        SOLID    SOLID    SOLID    SOLID    SOLID    SOLID    SOLID    SOLID    SOLID   ␣
↪SOLID    SOLID    SOLID    SOLID

HEAT_CAPACITY_BOREHOLE # [MJ/(K*m3)]  Dichte * spez. Waermekapazitaet der Radialelemente␣
↪um die Bohrung (P*M Werte)
#GAS STAHL    RRSF     STAHL    ZEM      Gebirge---->
0.000         3.600    4.200    3.600    1.600    1.920    1.920    1.920    1.920    1.920    1.
↪920    1.920    1.920    1.920    # UK 122.50m - Schicht 1
0.000         3.600    4.200    3.600    1.600    1.920    1.920    1.920    1.920    1.920    1.
↪920    1.920    1.920    1.920    # UK 245.00m - Schicht 1
0.000         3.600    4.200    3.600    1.600    1.950    1.950    1.950    1.950    1.950    1.
↪950    1.950    1.950    1.950    # UK 367.50m - Schicht 2
0.000         3.600    4.200    3.600    1.600    1.950    1.950    1.950    1.950    1.950    1.
↪950    1.950    1.950    1.950    # UK 490.00m - Schicht 2


THERMAL_CONDUCTIVITY_BOREHOLE # [W/(m*K)]  Waermeleitfaehigkeit der Radialelemente um␣
↪die Bohrung (P*M Werte)
#GAS STAHL    RRSF     STAHL    ZEM      Gebirge---->
0.000         50.000   0.500    50.000   1.000    2.330    2.330    2.330    2.330    2.330    2.
↪330    2.330    2.330    2.330    # UK 122.50m - Schicht 1
0.000         50.000   0.500    50.000   1.000    2.330    2.330    2.330    2.330    2.330    2.
↪330    2.330    2.330    2.330    # UK 245.00m - Schicht 1
0.000         50.000   0.500    50.000   1.000    5.500    5.500    5.500    5.500    5.500    5.
↪500    5.500    5.500    5.500    # UK 367.50m - Schicht 2
0.000         50.000   0.500    50.000   1.000    5.500    5.500    5.500    5.500    5.500    5.
↪500    5.500    5.500    5.500    # UK 490.00m - Schicht 2


TEMPERATURE_BOREHOLE  # [deg C] Temperatur der Radialelemente um die Bohrung (P*M Werte)
#GAS STAHL    RRSF     STAHL    ZEM      Gebirge---->
8.000         8.000    8.000    8.000    8.000    8.000    8.000    8.000    8.000    8.000    8.
↪000    8.000    8.000    8.000    # UK 122.50m - Schicht 1
12.000        12.000   12.000   12.000   12.000   12.000   12.000   12.000   12.000   12.000   12.
↪000   12.000   12.000   12.000   # UK 245.00m - Schicht 1
17.000        17.000   17.000   17.000   17.000   17.000   17.000   17.000   17.000   17.000   17.
↪000   17.000   17.000   17.000   # UK 367.50m - Schicht 2
22.000        22.000   22.000   22.000   22.000   22.000   22.000   22.000   22.000   22.000   22.
↪000   22.000   22.000   22.000   # UK 490.00m - Schicht 2


WELL_VERTICALITY  # [deg] Winkel zwischen Bohrlochachse und Bohrung (P Werte)
0.000        # UK 122.50m - Schicht 1
0.000        # UK 245.00m - Schicht 1
0.000        # UK 367.50m - Schicht 2
0.000        # UK 490.00m - Schicht 2


RADIAL_VECTOR_CAVERN # [m] (MK Werte)
```

(continues on next page)

(continued from previous page)

```
30.3000      35.1800 39.5100 43.8500 48.1900 52.5200 56.8600 61.1900 65.5300 69.8700 74.
↪2300 78.5400 82.8800 87.2200 91.5600 95.9000 100.2400        104.5800        109.9200 ␣
↪      113.2600        117.6000        121.9000        126.2400        135.0000       ␣
↪140.0000        145.0000        150.0000

TEMPERATURE_CAVERN  #  [deg C] Temperatur der Radialelemente um die Kaverne (MK Werte)
22.0000      22.0000 22.0000 22.0000 22.0000 22.0000 22.0000 22.0000 22.0000 22.0000 22.
↪0000 22.0000 22.0000 22.0000 22.0000 22.0000 22.0000 22.0000 22.0000 22.0000 22.0000␣
↪22.0000 22.0000 22.0000 22.0000 22.0000 22.0000

DENSITY_SALT 2170.00 #[kg/m3]
SPECIFIC_HEAT_CAPACITY_SALT  900.00  #[J/(kg*K)]
HEAT_CONDUCTIVITY_SALT       5.50    #[W/(m*K)]
HEIGHT_CAVERN        181.20  #[m]
VOLUME_BRINE_EQUIVALENT      30000.00        #[m3]
RADIUS_BRINE_LEVEL   30.30   #[m]
PRESSURE_CAVERN      9.80    #[MPa] at DEPTH_CAVERN 490.00m
```

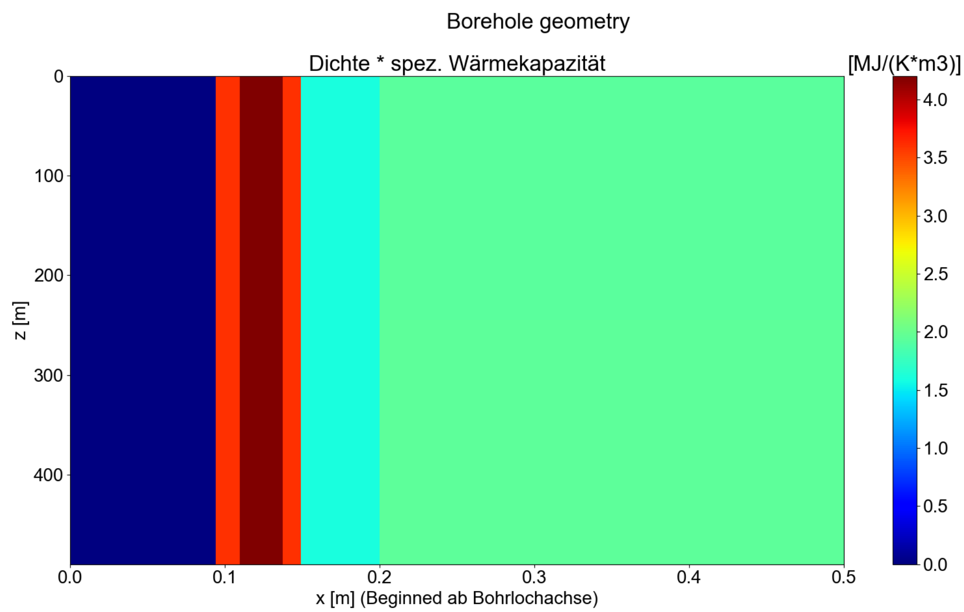
`BB_Gruppe1_Bb122_fd.pickle` ist eine Binär-Datei der exportieren Felddatenklasse. Sie kann mit dem Befehl in ein anderes Skript geladen werden.

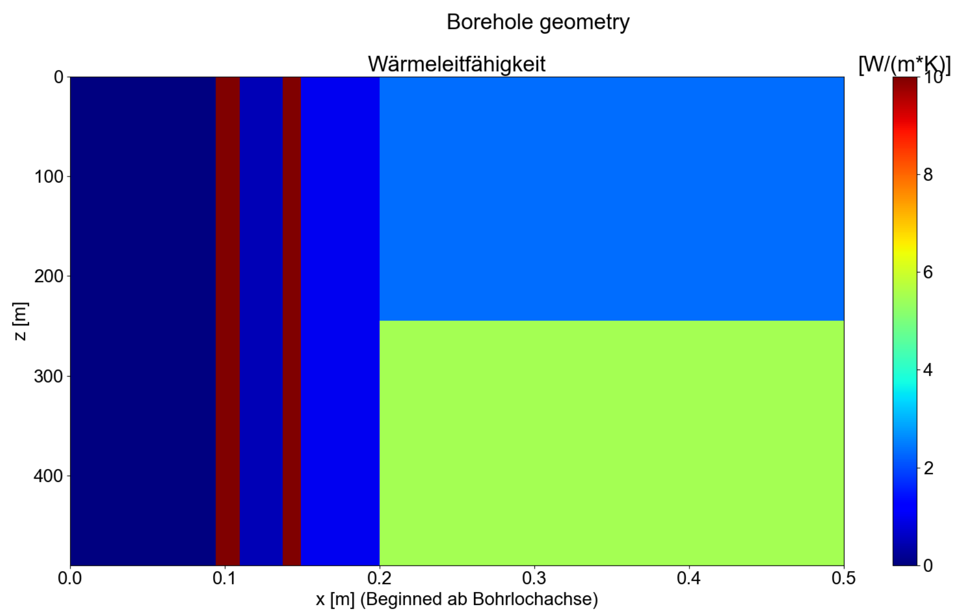
```python
import pickle
with open(field_data_picklePath, 'rb') as f:
   field_data = pickle.load(f)
```

2.1.8 6. Geometrie QC

Es empfiehlt sich, vor dem Starten der Simulation die erstellte txt-Datei zu kontrollieren. Außerdem kann die Geometrie mit folgendem Befehl geplottet werden:

```python
   Gaca_field.plot_geometry([0, 0.5]) # radial range
   plt.show()
```


Borehole geometry
Dichte * spez. Wärmekapazität
[MJ/(K*m3)]
z [m]
x [m] (Beginned ab Bohrlochachse)
0
100
200
300
400
0.0
0.1
0.2
0.3
0.4
0.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0


Borehole geometry
Wärmeleitfähigkeit
[W/(m*K)]
z [m]
x [m] (Beginned ab Bohrlochachse)
0
100
200
300
400
0.0
0.1
0.2
0.3
0.4
0.5
10
8
6
4
2
0

2.2 Prozessdaten

Für die Prozessdaten wird die Klasse `GacaProcessData` erstellt und mit Daten belegt.

2.2.1 1. Initialisierung

```
coupled_annuli  = [1]
medium_type     = ['GAS']
medium_id       = [medium_id_cavern]
description     = 'Gasspeicher Bernburg Bbg 122 Ausspeisung'
PD_Ausspeisung  = GacaProcessData(description, coupled_annuli, medium_type, medium_id)
```

2.2.2 2. Hinzufügen von Etappen

Etappen können chronologisch hinzugefügt werden. Dies hat den Vorteil, dass beispielsweise treppenstufige Aus- oder Einspeisungen in for-loops zu der Klasse hinzugefügt werden können (siehe unten). Beim Export der Klasse werden die Etappen automatisch nummeriert und formatiert. Wichtig ist dass die Parameter in einer 'dict' Klasse erstellt werden und die Schlüsselnamen den Parameternamen aus dem Handbuch entsprechen. Siehe dafür die Dokumentation von :py:func:add_stage

```
# Initialisiere Parameter Dictionary
ausspeisung_param = {
'TERMINATION_ID': 6,
'TERMINATION_QUANTITY': None,
'DT_MAX': 1,
'FLOW_RATE': None,
'K_S': 0.2 ,
'P_BOUNDARY_CONDITION': 'CAVERN',
'BOUNDARY_PRESSURE': 0,
'T_BOUNDARY_CONDITION': 'CAVERN',
'BOUNDARY_TEMPERATURE': 0
}

flow_rate_array            = np.array([2e5, 1.5e5, 1e5, 0.75e5, 0.3e5])  # verschiedene Raten
termination_quantity_array = np.array([9.5, 7.5,6.31,6.03,1.1])          # verschiedene min. Drücke


for flow_rate, termination_quantitiy in zip(flow_rate_array,termination_quantity_array): # for Schleife
   ausspeisung_param['FLOW_RATE']            = flow_rate
   ausspeisung_param['TERMINATION_QUANTITY'] = termination_quantitiy
   PD_Ausspeisung.add_stage(ausspeisung_param)
```

2.2.3 2. Export

```
save_folder     = r'L:\Projekte\SG-UBT\40_Thermodynamik\TSCW_GACA_Bernburg\Gruppe1'
project_name    = 'BB_Bb122'
PD_Ausspeisung.export_processData(save_folder,project_name)
```

In der Konsole wird folgendes angezeigt:

```
L:\Projekte\SG-UBT\40_Thermodynamik\TSCW_GACA_Bernburg\Gruppe1\BB_Gruppe1\Ausspeisung\
↪Bb122_gaca.pd.txt
Run sucessfull
```

Die entsprechende Prozessdatei hat folgende Gestalt:

```
DESCRIPTION  Gasspeicher Bernburg Bbg 122 Ausspeisung
N_FLUID      1
NUMBER_OF_STAGES [/] 5

MEDIUM_TYPE
GAS

MEDIUM_ID
Nordverbundgas

COUPLED_ANNULI       # [integer required!]
1



# ++++++++++++++++++++++++++++++++
STAGE        1

TERMINATION_ID       6
TERMINATION_QUANTITY 9.50
DT_MAX       1
FLOW_RATE    200000.00
K_S   0.20
P_BOUNDARY_CONDITION CAVERN

BOUNDARY_PRESSURE    0
T_BOUNDARY_CONDITION CAVERN

BOUNDARY_TEMPERATURE 0

# ++++++++++++++++++++++++++++++++
STAGE        2

TERMINATION_ID       6
TERMINATION_QUANTITY 7.50
DT_MAX       1
FLOW_RATE    150000.00
K_S   0.20
P_BOUNDARY_CONDITION CAVERN
```

(continues on next page)

(continued from previous page)

```
BOUNDARY_PRESSURE    0
T_BOUNDARY_CONDITION CAVERN

BOUNDARY_TEMPERATURE 0

# ++++++++++++++++++++++++++++++++
STAGE        3

TERMINATION_ID       6
TERMINATION_QUANTITY 6.31
DT_MAX       1
FLOW_RATE    100000.00
K_S  0.20
P_BOUNDARY_CONDITION CAVERN

BOUNDARY_PRESSURE    0
T_BOUNDARY_CONDITION CAVERN

BOUNDARY_TEMPERATURE 0

# ++++++++++++++++++++++++++++++++
STAGE        4

TERMINATION_ID       6
TERMINATION_QUANTITY 6.03
DT_MAX       1
FLOW_RATE    75000.00
K_S  0.20
P_BOUNDARY_CONDITION CAVERN

BOUNDARY_PRESSURE    0
T_BOUNDARY_CONDITION CAVERN

BOUNDARY_TEMPERATURE 0

# ++++++++++++++++++++++++++++++++
STAGE        5

TERMINATION_ID       6
TERMINATION_QUANTITY 1.10
DT_MAX       1
FLOW_RATE    30000.00
K_S  0.20
P_BOUNDARY_CONDITION CAVERN

BOUNDARY_PRESSURE    0
T_BOUNDARY_CONDITION CAVERN

BOUNDARY_TEMPERATURE 0
```

CHAPTER

THREE

TSCW - OUTPUT - PROZESSIERUNG

3.1 TBHC.txt

Druck- und Temperaturergebnisse in Kaverne und im Strömungsraum i Pro Zeitschritt werden hier die Druck- und Temperaturwerte in der Kaverne und für jeden Strömungsraum pro Schicht ausgegeben. Der Index i steht für die Nummer des Strömungsraumes (Tubing bzw. Ringraum), hochgezählt beginnend von der Bohrlochachse. Die Struktur dieser Datei kann in drei Teile gegliedert werden, die durch *** voneinander getrennt sind. Die ersten Zeilen sind im Folgenden exemplarisch dargestellt:

```
FLUID SPACE NO.:                    1
MEDIUM:      Stage 1: GAS.Methan
V_CAVERN:    488925.22
***
STAGE        t_STAGE t_TOTAL T_CAVERN        p_CAVERN        T_BRINE_EQ        T_WH    p_
↪WH  PRODUCT AMOUNT  FLOW_RATE        w_MAX  ***       0.00    25.00    50.00    75.00 ␣
↪100.00  125.00  150.00  175.00  200.00  225.00  250.00  275.00  300.00  325.00  350.00␣
↪ 375.00  400.00  425.00  450.00  475.00  500.00  525.00  550.00  575.00  600.00  625.
↪00  650.00  675.00  700.00  725.00  750.00  775.00  800.00  825.00  850.00  875.00 ␣
↪900.00  925.00  950.00  975.00  1000.00 1100.00 ***       0.00    25.00    50.00    75.00␣
↪ 100.00  125.00  150.00  175.00  200.00  225.00  250.00  275.00  300.00  325.00  350.
↪00  375.00  400.00  425.00  450.00  475.00  500.00  525.00  550.00  575.00  600.00 ␣
↪625.00  650.00  675.00  700.00  725.00  750.00  775.00  800.00  825.00  850.00  875.00␣
↪ 900.00  925.00  950.00  975.00  1000.00 1100.00
1      0.00    0.00    24.26           15.00            24.26        10.00    13.05    ␣
↪80400839.05  100000.00        10.70  ***      10.00    11.00    12.00    13.00    14.00  ␣
↪15.00    16.00    17.00    18.00    19.00    20.00    21.00    22.00    23.00    24.00    25.00 ␣
↪ 26.00    27.00    28.00    29.03    30.00    30.78    31.50    32.25    33.00    33.75    34.50␣
↪  35.28    36.00    36.53    37.00    37.50    38.00    38.50    39.00    39.50    40.00    41.
↪61    41.00    34.85     24.26     24.26 ***      13.05    13.10    13.15    13.19    13.24  ␣
↪13.28    13.33    13.38    13.42    13.47    13.51    13.56    13.61    13.65    13.70    13.74 ␣
↪ 13.79    13.83    13.88    13.93    13.97    14.02    14.06    14.11    14.15    14.20    14.25␣
↪  14.29    14.34    14.38    14.43    14.47    14.52    14.57    14.61    14.66    14.70    14.
↪75    14.79    14.84    14.88    15.00
1      0.00    0.00    24.26           15.00            24.26        10.59    13.05    ␣
↪80400709.24  100000.00        10.71  ***      10.59    11.60    12.60    13.60    14.60  ␣
↪15.60    16.60    17.60    18.60    19.60    20.60    21.60    22.60    23.60    24.61    25.61 ␣
↪ 26.61    27.62    28.61    29.55    30.42    31.19    31.92    32.67    33.43    34.19    34.93␣
↪  35.61    36.24    36.76    37.24    37.74    38.24    38.74    39.24    40.15    40.24    38.
↪19    34.71    30.10     24.26     24.26 ***      13.05    13.10    13.15    13.19    13.24  ␣
↪13.28    13.33    13.38    13.42    13.47    13.51    13.56    13.61    13.65    13.70    13.74 ␣
↪ 13.79    13.83    13.88    13.93    13.97    14.02    14.06    14.11    14.15    14.20    14.25␣
```

(continues on next page)

(continued from previous page)

```
↪  14.29   14.34   14.38   14.43   14.47   14.52   14.56   14.61   14.66   14.70   14.
↪75   14.79   14.84    14.88   15.00
1      0.00    0.00    24.26           15.00              24.26        11.08   13.05    ␣
↪80400579.44  100000.00        10.73  ***       11.08   12.09   13.09   14.09   15.09  ␣
↪16.09   17.09   18.09   19.09   20.09   21.09   22.09   23.10   24.10   25.10   26.10 ␣
↪ 27.10   28.08   29.03   29.93   30.77   31.53   32.27   33.02   33.77   34.50   35.20␣
↪  35.85   36.44   36.95   37.44   37.94   38.44   39.09   39.44   39.17   38.02   35.
↪38   31.95   28.31    24.26   24.26 ***       13.05   13.10   13.15   13.19   13.24  ␣
↪13.28   13.33   13.38   13.42   13.47   13.51   13.56   13.61   13.65   13.70   13.74 ␣
↪ 13.79   13.83   13.88   13.93   13.97   14.02   14.06   14.11   14.15   14.20   14.25␣
↪  14.29   14.34   14.38   14.43   14.47   14.52   14.56   14.61   14.66   14.70   14.
↪75   14.79   14.84    14.88   15.00
1      0.00    0.00    24.26           15.00              24.26        11.51   13.05    ␣
↪80400449.63  100000.00        10.74  ***       11.51   12.51   13.51   14.52   15.52  ␣
↪16.52   17.52   18.52   19.52   20.52   21.52   22.52   23.52   24.53   25.53   26.52 ␣
↪ 27.50   28.46   29.37   30.24   31.07   31.83   32.57   33.31   34.04   34.75   35.41␣
↪  36.03   36.61   37.12   37.61   38.16   38.61   38.89   38.71   37.74   36.05   33.
↪49   30.48   27.45    24.26   24.26 ***       13.05   13.10   13.15   13.19   13.24  ␣
↪13.28   13.33   13.38   13.42   13.47   13.51   13.56   13.61   13.65   13.70   13.74 ␣
↪ 13.79   13.83   13.88   13.93   13.97   14.02   14.06   14.11   14.15   14.20   14.24␣
↪  14.29   14.34   14.38   14.43   14.47   14.52   14.56   14.61   14.66   14.70   14.
↪75   14.79   14.84    14.88   15.00
1      0.01    0.01    24.26           15.00              24.26        11.90   13.05    ␣
↪80400319.83  100000.00        10.75  ***       11.90   12.90   13.91   14.91   15.91  ␣
↪16.91   17.91   18.91   19.91   20.91   21.91   22.91   23.91   24.91   25.91   26.89 ␣
↪ 27.85   28.78   29.68   30.53   31.34   32.10   32.83   33.56   34.27   34.95   35.60␣
↪  36.20   36.76   37.29   37.76   38.20   38.43   38.29   37.68   36.38   34.55   32.
↪18   29.54   26.94    24.26   24.26 ***       13.05   13.10   13.15   13.19   13.24  ␣
↪13.28   13.33   13.38   13.42   13.47   13.51   13.56   13.61   13.65   13.70   13.74 ␣
↪ 13.79   13.83   13.88   13.93   13.97   14.02   14.06   14.11   14.15   14.20   14.24␣
↪  14.29   14.34   14.38   14.43   14.47   14.52   14.56   14.61   14.65   14.70   14.
↪75   14.79   14.84    14.88   15.00
1      0.01    0.01    24.26           15.00              24.26        12.27   13.06    ␣
↪80400190.02  100000.00        10.76  ***       12.27   13.27   14.28   15.28   16.28  ␣
↪17.28   18.28   19.28   20.28   21.28   22.28   23.28   24.28   25.27   26.25   27.22 ␣
↪ 28.16   29.08   29.95   30.79   31.59   32.34   33.06   33.78   34.47   35.14   35.77␣
↪  36.36   36.90   37.40   37.78   38.03   37.99   37.54   36.66   35.22   33.38   31.
↪21   28.87   26.58    24.26   24.26 ***       13.06   13.10   13.15   13.19   13.24  ␣
↪13.28   13.33   13.38   13.42   13.47   13.51   13.56   13.61   13.65   13.70   13.74 ␣
↪ 13.79   13.83   13.88   13.93   13.97   14.02   14.06   14.11   14.15   14.20   14.24␣
↪  14.29   14.34   14.38   14.43   14.47   14.52   14.56   14.61   14.65   14.70   14.
↪75   14.79   14.84    14.88   15.00
(weiter bis zum Simulationsende)
```

Die `*TBHC.txt` Datei kann mit folgendem Befehl eingelesen werden:

```python
from tscw_module import TSCW_TBHC
path = "DeinPfad/*TBHC.txt"
tscw_data = TSCW_TBHC(path)
```

Die einzelenen Sektionen können aufgerufen werden:

```
tscw_data.sr_df  # meta data
tscw_data.vertT_df # temperatur daten
tscw_data.vertP_df # druck daten
tscw_data.df # gesamtes data frame
```

3.2 TFBH.txt

3.3 TFC.txt

CHAPTER

FOUR

TSCW_MODULE

4.1 tscw_module package

4.1.1 Submodules

4.1.2 tscw_module.tscw_DataClassesInput module

class tscw_module.tscw_DataClassesInput.**GacaFieldData**(*n_boreholes*, *n_fluid*, *tvd_max_borehole*, *delta_z*, *medium_type_cavern*, *medium_id_cavern*)

Bases: `object`

Class to create a field data file (*_gaca.fd.txt) as input for TSWC-GACA. Later on the following methods need to be performed for each instance of the class. For each step the according methods will check for correct inputs and prevent error in the exported txt-file.

Workflow for Generating field data.

—- 1. BOREHOLE —-

1. 'add_boreholeVector'
2. 'add_radialComment' (optional)
3. 'add_boreholeInclination'
4. 'add_materialProperty' (depending on vertical discretisation)
5. 'add_temperature' - borehole

—- 2. CAVNERN —-

1. 'add_cavernVector'
2. 'add_temperature' - cavern
3. 'add_cavernCharacteristics'

—- The last step is to export the field data to an txt-file with the method 'export_fieldData'. —-

add_boreholeInclination(*inclination*)

Add inclination for borehole

Parameters

inclination (*array [1xp_borehole] oder 'vertical'*) – either array [1 x p_borehole] with deg data or 'vertical'

add_boreholeVector(*radial_vector*, *aggregate_state*)

Adds a radial vector within the borehole [m] and respective aggregate states ('FLUID' or 'SOLID').

Parameters

- **radial_vector** (*list or array [m]*) – must be strictly increasing, will define self.m_borehole [int].
- **aggregate_state** (*string array*) – list of strings either 'FLUID' or 'SOLID'. Does not need to have the same length as radial_vector. Remaining values will be filled with 'SOLID'.

add_cavernCharacteristics(*refdepth_cavern*, *density_salt*, *specific_heat_capacity_salt*, *heat_conductivity_salt*, *height_cavern*, *volume_brine_equivalent*, *radius_brine_level*, *pressure_cavern*, *temperature_brine_equivalent=None*)

Add cavern characteristics:

Parameters

- **refdepth_cavern** (*float*) – reference depth for cavern pressure and modeling [m]
- **density_salt** (*float*) – [kg/m3]
- **specific_heat_capacity_salt** (*float*) – [J/kgK]
- **heat_conductivity_salt** (*float*) – [W/mK]
- **height_cavern** (*float*) – consists of H_zy + 2*rad_cav [m]
- **volume_brine_equivalent** (*float*) – [m3]
- **radius_brine_level** (*float*) – [m]
- **pressure_cavern** (*float*) – [MPa]
- **temperature_brine_equivalent** (*float, optional*) – [°C], defaults to None

add_cavernVector(*radial_vector*)

Adds a radial vector within the cavern. The first element is the cavern radius.

Parameters

radial_vector (*list or array [m]*) – must be strictly increasing, will define self.m_cavern [int].

add_materialProperty(*top*, *bottom*, *heat_capacity*, *thermal_conductivity*, *name=None*)

Adds material properties to the borehole. The value for the center of gravity of the layer is then modeled, with indication in relation to the bottom edge of the layer. Hierarchical; new values overwrite respective old values at the same intervals.

Parameters

- **top** (*int or float*) – start of layer [m]
- **bottom** (*int or float*) – end of layer [m]
- **heat_capacity** (*array [1 x m_borehole]*) – [MJ/(m3K)]
- **thermal_conductivity** (*[W/(m K)]*) – [1 x m_borehole]
- **name** (*str, optional*) – Name of the layer, will be displayed in .txt file when exported, defaults to None

add_radialComment(*radialComment*)

Add comments for radial borehole vector. Will be displayed lather in txt file (OPTIONAL - for overview purpose only). If the array has a length of e.g. 5, then the comment is valid for the first five radial elements.

Parameters

radialComment (*array*) – E.g. ['ID 858','OD 858','ID 1134', 'ZEMENT', 'FORMATION']

add_temperature(*temperature*, *mode*)

Adds temperature data to the borehole or cavern.

Parameters

- **temperature** (*_type_*) – (p_borehole x m_borehole) array for 'borehole or (m_cavern) for cavern
- **mode** (*str*) – either 'borehole' or 'cavern'

export_fieldData(*save_folder*, *project_name*, *is_binary_export=False*)

Exports class to a txt-File with the suffix "_gaca.fd.txt". Will be exported to a folder named (path, project_name). If it does not exist, it will be created.

Parameters

- **save_folder** (*str*) – path
- **project_name** (*str*) – suffix on path,
- **is_binary_export** (*bool, optional*) – Export class as a binary file (*.pickle). It can be later loaded to display to geometry for example in TSWC_TFBH.create_movie(), defaults to False

plot_geometry(*index_range=None*, *is_export=False*)

Plots geometry.

Parameters

- **index_range** (*list, optional*) – start and end value of radial range, defaults to None
- **is_export** (*bool, optional*) – Export plot data. defaults to False

Returns

Two figures for heat capacity * rho and lambda

Return type

fig_cp, fig_lambda

class tscw_module.tscw_DataClassesInput.**GacaProcessData**(*description*, *coupled_annuli*, *medium_type*, *medium_id*)

Bases: `object`

add_stage(*stage_data*)

Erlaubt flexibles hinzufügen von stages. Stages werden chronologisch hinzugefügt. WICHTIG: Schlüsselnamen aus dict müssen mit Variablenname von TSWC übereinstimmen, erlaubt sind folgende Namen:

- 'TERMINATION_ID',
- 'TERMINATION_QUANTITY',
- 'DT_MAX',
- 'FLOW_RATE',
- 'P_BOUNDARY_CONDITION',

- 'BOUNDARY_PRESSURE',
- 'T_BOUNDARY_CONDITION',
- 'BOUNDARY_TEMPERATURE',
- 'K_S'

'TERMINATION_ID' und 'TERMINATION_QUANTITY' müssen eingegeben werden!

Parameters
stage_data (*dict*) – Enthält alle erforderlichen Parameter in der Form {Schlüssel: Wert}

export_processData(*save_folder*, *project_name*, *suffix=''*)

Exports class to a txt-File with the suffix '_gaca.pd.txt'. Will will be exported to a folder named (path, project_name, suffix). If it does not exist, it will be created.

Parameters

- **save_folder** (*str*) – _description_
- **project_name** (*str*) – _description_
- **suffix** (*str, optional*) – defaults to ''

4.1.3 tscw_module.tscw_DataClassesOutput module

class tscw_module.tscw_DataClassesOutput.**TSCW_Output**(***kwargs*)

Bases: object

Super class for TSWC Output. Other instances are inheritances of it. Must not be called!

calculate_axial_forces_super(*meta_data*, *t_vector*, *p_vector*, *t_total*, *t_etappe*, *T0=None*, *export_dir=None*)

Berechnet Axialkraft bezogen auf Ansatz BB122 aus dem Jahr 2011. Daten und Formeln aus berechnungenbbg_rev2.xlsx.

Parameters

- **meta_data** (*_type_*) – _description_
- **t_vector** (*[1 x n] array*) – temperature in [K]
- **p_vector** (*[1 x n] array*) – pressure in [MPa]
- **t_total** (*[1 x n] array*) – total time in [h]
- **t_etappe** (*[1 x n] array*) – stage time in [h]
- **T0** (*float, optional*) – temperature for reference in delta_t [optional - float], else T = t_total[0]
- **export_dir** (*str, optional*) – absolute file path to export results, needs to end with '.xlsx' , defaults to None

Returns
pandas dataframe containing forces

Return type
pd.Dataframe

create_data_array2plot(*all_plot_points*, *all_data*, *mode='time'*)

Extracts data to plot. Must no be called by user (for intern purposes)

Parameters

- **all_plot_points** (*list*) – which depths to plot [depth_t, depth_p]
- **all_data** (*list*) – [self.vertT_df,self.vertP_df] temperature and pressure data
- **mode** (*str, optional*) – either 'depth' or 'time', defaults to 'time'

Returns

description

Return type

type

create_pT_plot(*data*, *meta*)

creates a temperture - pressure plot against time or depth.

Parameters

- **data** – output from create_data_array2plot
- **meta** (*dict*) – containing information for meta data in plot

Returns

description

Return type

type

class tscw_module.tscw_DataClassesOutput.**TSCW_TBHC**(*path*)

Bases: *TSCW_Output*

Reads data from Projektname_i_pTBHC.txt Important attributes: self.sr_df: - meta data self.vertT_df: - temperature data self.vertP_df: - pressure data

calculate_axial_forces(*meta_data*, *z_ref*, *T0=None*, *is_export=False*)

Calculates resulting axial forces

Parameters

- **meta_data** (*dict*) – dictionary containing meta data.
- **z_ref** (*_type_*) – reference depth, temperature and pressure array will be calculated by the mean at z_ref and z0 = 0m.
- **T0** (*float, optional*) – Initial temperature for reference in delta_T, if None it will be the first element of the temperature array.
- **is_export** (*bool, optional*) – _description_, defaults to False

Returns

pandas dataframe containing forces

Return type

pd.Dataframe

export_csv(*depths*)

Exports data to a xlsx- file.

Parameters

depths (*list*) – which depths to export (alogrithm looks for closest match) `extract_max_force`(*i_etappe*, *mode*, *min_time=0*)

Extracts min or max Fz_ges for a selected Stage.

Parameters

- **i_etappe** (*int*) – Stage number
- **mode** (*str*) – either 'min' or 'max'
- **min_time** (*float, optional*) – float - minimum time that has passed after the value is selected, put in +inf to select end of stage, 0 by default, defaults to 0

Returns

pd.Dataframe containing relevant parameters, respective index in self.df

Return type

type

`plot_axial_forces`(*is_export=False*)

Plots axial forces of forces_df.

Parameters

is_export (*bool, optional*) – Export as png, defaults to False

Return type

figure

`plot_cavern_pt_development`(*is_export=False*)

Plots cavern pressure - temperature development over time.

Parameters

is_export (*bool, optional*) – defaults to False

`plot_forces_difference`(**args*)

Plot difference between calculated forces. Pass other TSWC_TBHC instances as input (comma separated). :type args: :param args: Other instances of TSWC_TBHC. :rtype: figure

`plot_pt_difference`(*depths*, **args*)

`plot_tp_vs_depth`(*depth_t=None*, *depth_p=None*, *is_export=False*)

Plots borehole temperature - pressure development over depth.

Parameters

- **depth_t** (*_type_, optional*) – which time intervals for Temperature, by default all
- **depth_p** (*_type_, optional*) – which time intervals for Pressure, by default all
- **is_export** (*bool, optional*) – defaults to False

`plot_tp_vs_time`(*depth_t=None*, *depth_p=None*, *is_export=False*)

summary

Parameters

- **depth_t** (*_type_, optional*) – which depth intervals for Temperature, by default all
- **depth_p** (*which depth intervals for Pressure, by default all*) – _description_, defaults to None
- **is_export** (*bool, optional*) – defaults to False

class tscw_module.tscw_DataClassesOutput.**TSCW_TFBH**(*path*)

Bases: *TSCW_Output*

Reads data from /*_TFBH.TXT (Radial temperature along depth in borehole). Stores data in self.data: - data

create_movie(*range_radial=None*, *is_export=False*, *n_levels=100*, *field_data_picklePath=None*)

Generates a time lapse of the simulated temperature.

Parameters

- **range_radial** (*list, optional*) – [x0, x1] range of radial start and end point (no exact match needed), defaults to None
- **is_export** (*bool, optional*) – export movie as .mp4 into the same folder of current instance, defaults to False
- **n_levels** (*int, optional*) – how many levels for plt.contourf, defaults to 150
- **field_data_picklePath** (*str, optional*) – path to (/*.pickle) of FieldData class (if it has been exported). If loaded, the geometry will be displayed in the background., defaults to None

Returns

animation

Return type

animation.FuncAnimation

plot_temp_distribution(*times*, *depths=None*, *range_radial=None*, *is_colormap=True*, *n_levels=150*, *is_export=False*)

The values given in all input lists do not need to match exactly with the simulation result. The algorithm automatically finds the nearest neighbor.

Parameters

- **times** (*list*) – which time points
- **depths** (*list, optional*) – which depth points, defaults to None
- **range_radial** (*list, optional*) – [x0, x1] beginning and end of radial range, defaults to None
- **is_colormap** (*bool, optional*) – plot as a colormap or line plot, defaults to True
- **n_levels** (*int, optional*) – how many levels in pcolormesh plot, defaults to 150
- **is_export** (*bool, optional*) – export figure into parent folder of file, defaults to False

Return type

figure

class tscw_module.tscw_DataClassesOutput.**TSCW_TFC**(*path*)

Bases: *TSCW_Output*

Reads data from /*_TFC.TXT (cavern temperature over time) Stores data in self.data: - data

plot_temp_distribution(*times*, *range_radial=None*, *is_export=False*)

Plot radial temperature development in cavern over time.

Parameters

- **times** (*list*) – which time points
- **range_radial** (*list, optional*) – [x0, x1] beginning and end of radial range, defaults to None

4.1.4 tscw_module.thermreg_DataClasses module

class `tscw_module.thermreg_DataClasses.`**`ThermregData`**(*path*, *n_depths=None*)

Bases: `object`

Read in and process Thermreg output data.

`calculate_axial_forces`(*meta_data*, *T0=None*, *vectors=<class 'tuple'>*, *is_export=False*)

Calculates resulting axial forces INPUT: meta_data: dictionary containing meta data. T0: Initial temperature for reference in delta_T vectors: tuple - (temperature, pressure) [K, MPa] is_export: true or false - as a xlsx file OUTPUT Dataframe

`export_csv`()

Export Thermreg p-T data into a xlsx file.

`extract_max_force`(*i_etappe*, *mode*, *min_time=0*)

Extracts min or max Fz_ges for a selected Stage. INPUT: i_etappe: int - Stage number mode: either 'min' or 'max' min_time: float - minimum time that has passed after the value is selected, put in +inf to select end of stage, 0 by default OUTPUT: filtered_df: pd.Dataframe containing relevant parameters df_index: int - index of total Dataframe df of respective class.

`interpolate_pt`(*time_array*)

Interpolates pressure and temperature for a new time_array INPUT: time_array: list - new time array OUTPUT thermreg_temp_inter - interpolated temperature array thermreg_pres_inter - interpolated pressure array

`plot_all_tempfields`(*depths=None*, *radial_vector=None*, *x2_limit=None*, *is_export=False*)

`plot_axial_forces`(*is_export=False*)

Plots axial forces of forces_df. INPUT: is_export: bool - Export as png [optional] OUTPUT figure

`plot_tp_vs_depth`(*depths*, *times*, *is_export=False*)

`plot_tp_vs_time`(*depths*, *is_export=False*)

Plots a temperature - pressure diagram vs Time. INPUT: depths: array of depths (must match n_depths) is_export: true or false [optional] OUTPUT: figure

4.1.5 Module contents

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

t

INDEX