Iceland Liechtenstein Norway grants

Norway grants

Kalkulator Otworu Geotermalnego

projekt GeoModel

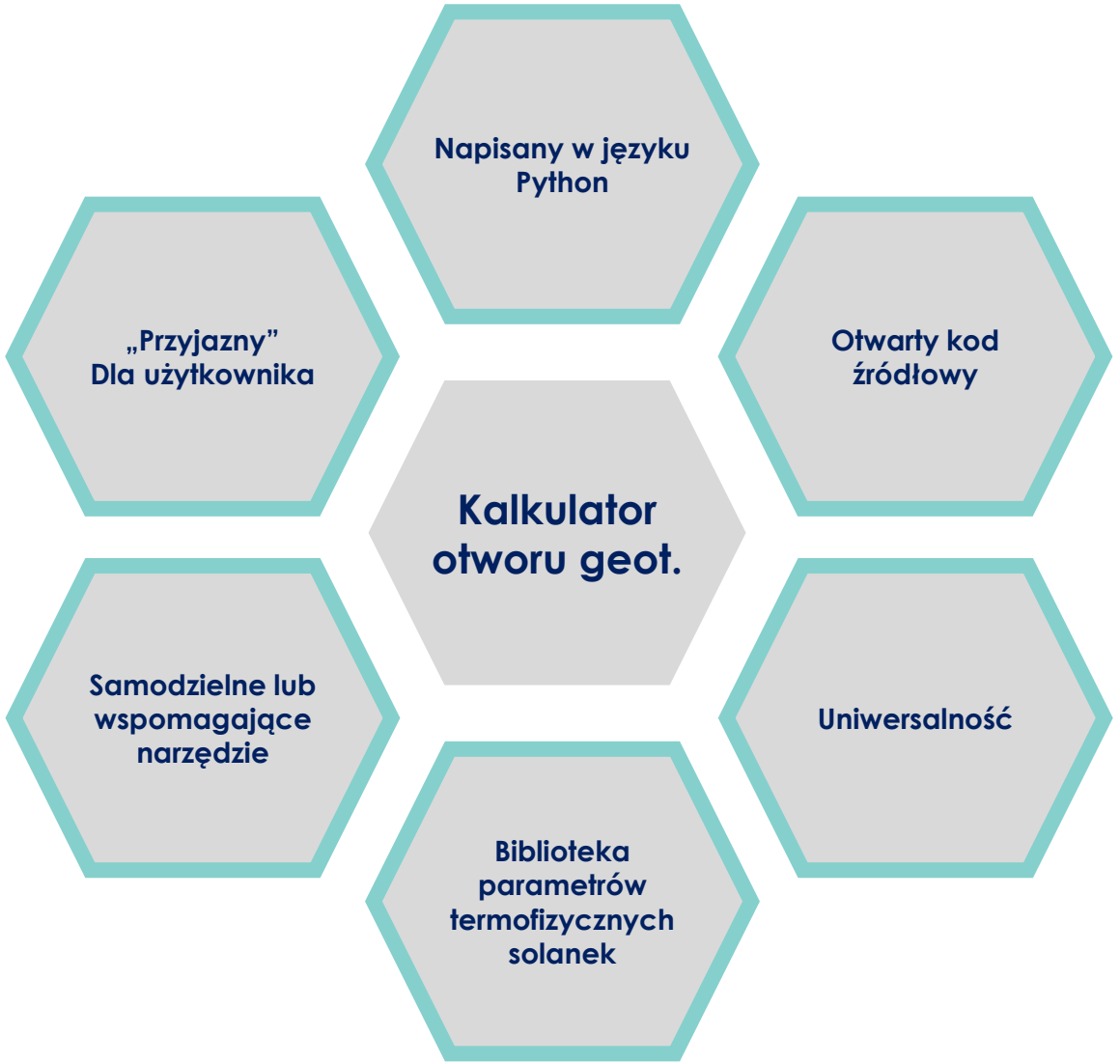
Leszek Pająk, Karol Pierzchała

IGSMiE PAN

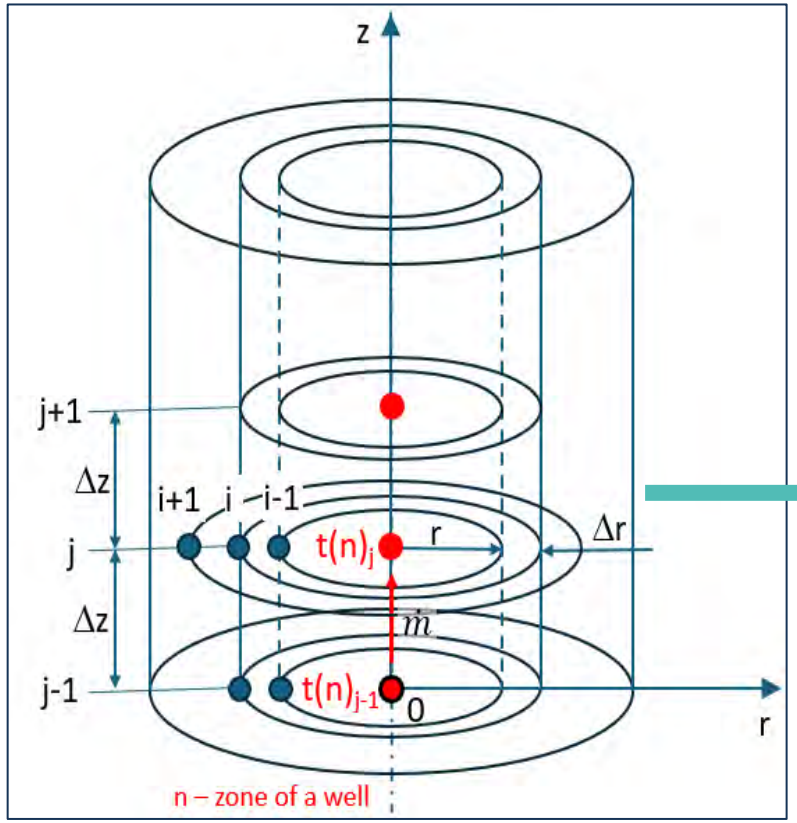
Cel stworzenia kalkulatora?

- Szybkie i wydajne narzędzie do oszacowania temperatury głowicowej solanki geotermalnej
- Możliwość zastosowania kalkulatora do solanek o wysokiej mineralizacji (np. Stargard i Pyrzyce)
- Opcja późniejszej adaptacji na kalkulator otworowego wymiennika ciepła

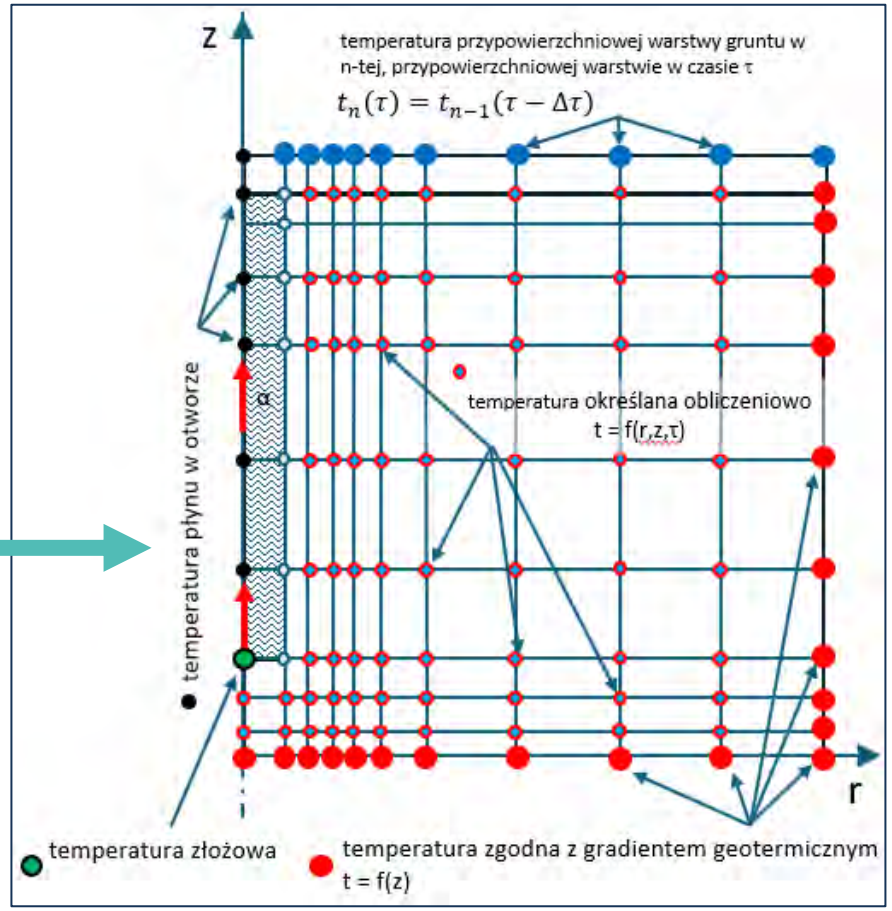
Cechy narzędzia



Metodologia

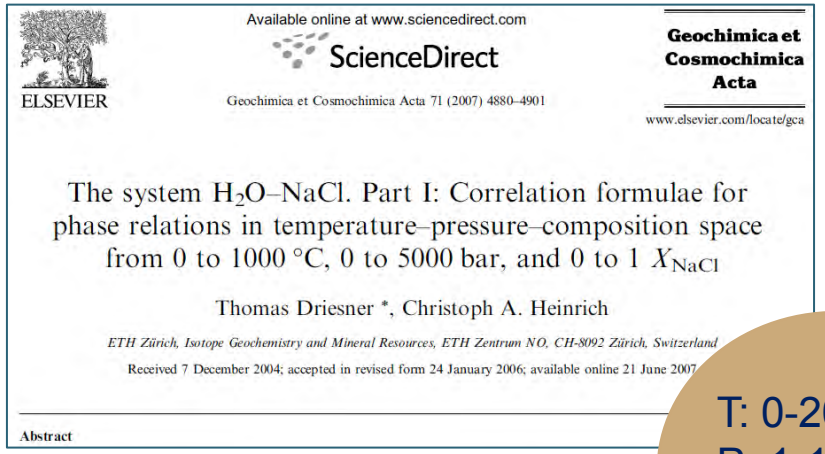


Strefy otworu



Modelowany obszar wraz z warunkami brzegowymi

Korelacje dla solanek


Available online at www.sciencedirect.com
ScienceDirect
Geochimica et Cosmochimica Acta 71 (2007) 4880–4901
Geochimica et Cosmochimica Acta
www.elsevier.com/locate/gca

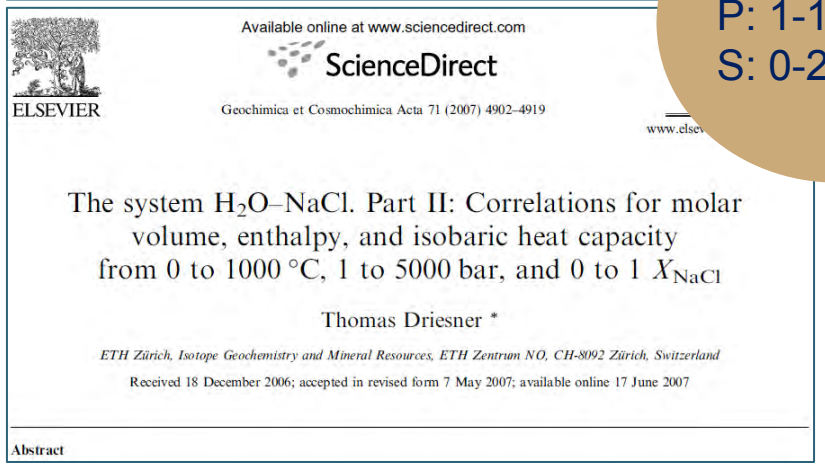
The system H_2O–NaCl. Part I: Correlation formulae for phase relations in temperature–pressure–composition space from 0 to 1000 °C, 0 to 5000 bar, and 0 to 1 X_{NaCl}

Thomas Driesner *, Christoph A. Heinrich

ETH Zürich, Isotope Geochemistry and Mineral Resources, ETH Zentrum NO, CH-8092 Zürich, Switzerland

Received 7 December 2004; accepted in revised form 24 January 2006; available online 21 June 2007

Abstract


Available online at www.sciencedirect.com
ScienceDirect
Geochimica et Cosmochimica Acta 71 (2007) 4902–4919

The system H_2O–NaCl. Part II: Correlations for molar volume, enthalpy, and isobaric heat capacity from 0 to 1000 °C, 1 to 5000 bar, and 0 to 1 X_{NaCl}

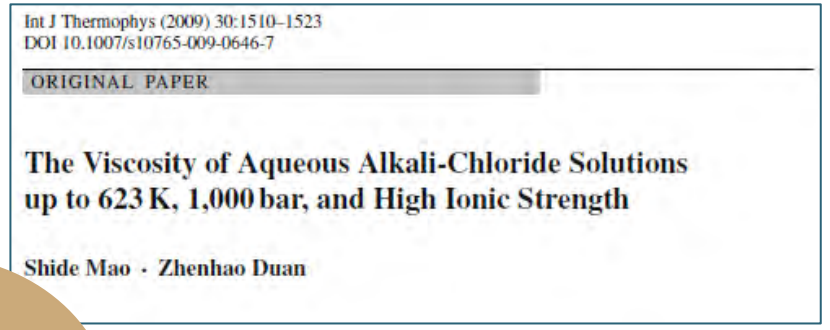
Thomas Driesner *

ETH Zürich, Isotope Geochemistry and Mineral Resources, ETH Zentrum NO, CH-8092 Zürich, Switzerland

Received 18 December 2006; accepted in revised form 7 May 2007; available online 17 June 2007

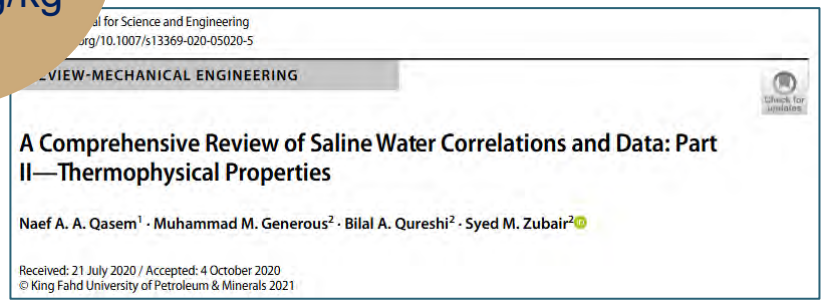
Abstract

T: 0-200°C
P: 1-1000 bar
S: 0-250 g/kg


Int J Thermophys (2009) 30:1510–1523
DOI 10.1007/s10765-009-0646-7
ORIGINAL PAPER

The Viscosity of Aqueous Alkali-Chloride Solutions up to 623 K, 1,000 bar, and High Ionic Strength

Shide Mao · Zhenhao Duan


al for Science and Engineering
org/10.1007/s13369-020-05020-5
VIEW-MECHANICAL ENGINEERING

A Comprehensive Review of Saline Water Correlations and Data: Part II—Thermophysical Properties

Naef A. A. Qasem[1] · Muhammad M. Generous[2] · Bilal A. Qureshi[2] · Syed M. Zubair[2]

Received: 21 July 2020 / Accepted: 4 October 2020
© King Fahd University of Petroleum & Minerals 2021

Tylko do przepływów jednofazowych!

Przygotowanie danych wsadowych

```
4   def main():
5
6
7       if 1:
8           fileName = 'well_prd.xlsx'
9       else:
10          fileName ='well_inj.xlsx'
11
```

Plik Excel → well_prd.xlsx lub well_inj.xlsx

Geometria otworu i strefy przyotworowej (geo)

	A	B	C	D	E	F	G	H	I
1	h1[m]	h2[m]	t1[°C]	t2[°C]	d1[m]	mat	d2[m]	mat	D[m]
2	0	1	8		0.2445	2	0.3	3	6
3	1	50			0.2445	2	0.3	3	6
4	50	2000		60	0.2445	2	0.3	3	6

geo | mat | work | others

Dane materiałowe (mat)

	A	B	C	D	E
1	mat	λ[W/(mK)]	c[J/(kgK)]	r(kg/m3]	type
2	0	0.6	4190	1000	water
3	1	5.00	2000.00	3000.00	air
4	2	4.50	1100.00	2600.00	well surroundings
5	3	4.50	1100.00	2600.00	rock 1

Harmonogram pracy (work)

	A	B	C	D
1	time_[s]	flow_[kg/s]	t_[°C]	pBot[Pa]
2	7200	5	60	20601000
3	14400	10	60	20601000
4	21600	15	60	20601000
5	25200	20	60	20601000

Pozostałe dane (others)

	A	B
1	description	value
2	brine TDS [kg_TDS/kg_pureWater]	0.05
3	The well pipes roughness [-]	1.78
4	Iterative accuracy (suggested 1E-5)	1.00E-05
5	maximal allowed length of section by Z [m]	10
6	maximal allowed value of time step [s]	40

Kod źródłowy

Procedury napisane w Pythonie
Pliki brineProp.py and main.py

biblioteka.py | brineProp.py | wells.py | MRS_RZ.py

brineProp.py > DpFlow

```python
import numpy as np
import CoolProp.CoolProp as cp
import math
#
# # T. Driesner 2007, Procedure for binary H2O-NaCl solutions.
# Scope of procedure: Temperature 0-1000C, pressure 1-5000 bar, molar fraction 0-1 XNaCl.
# Input: Temperature [°C], Pressure [Pa], Salinity [salt mass fraction] ► kg of salt/kg of pure wat
# Result in [J/kg*K]
def cHeat(T,pressure,salt_mass_fraction):
    temperature = T - 273.15
    pbar = pressure / 1e5  # Convert pressure to bar
    molar_mass_H2O = 18.015  # g/mol
    molar_mass_NaCl = 58.443  # g/mol
    n_NaCl = salt_mass_fraction*1000 / molar_mass_NaCl
    n_H2O = 1000 / molar_mass_H2O
    xmol = n_NaCl / (n_NaCl + n_H2O)
    xmol1 = 1 - xmol
    xmol12 = xmol1 * xmol1

    q11 = -32.1724 + 0.0621255 * pbar
    q21 = -1.69513 -(4.52781e-4)*pbar -(6.04279e-8)*(pbar**2)
    q22 = 0.0612567 + (1.88082e-5)* pbar

    q1x1 = 47.9048 -(9.36994e-3)*pbar+(6.51059e-6)*(pbar**2)
    q2x1 = 0.241022+ (3.45087e-5)*pbar-(4.28356e-9)*(pbar**2)

    q10 = q1x1
    q20 = 1 - q21 * np.sqrt(q22)
    q12 = -q11 - q10
    q23 = q2x1 - q20 - q21 * np.sqrt(1 + q22)

    q1 = q10 + q11 * xmol1 + q12 * xmol12 #ok
    q2 = q20 + (q21 * np.sqrt(xmol + q22)) + q23 * xmol #ok

    tstar_h = q1 + q2 * temperature

    cHeat=q2*cp.PropsSI("C","T",tstar_h+273.15,"P",pressure,"Water")
    return cHeat
```

biblioteka.py | brineProp.py | main.py | wells.py | MRS_RZ.py

main.py > ...

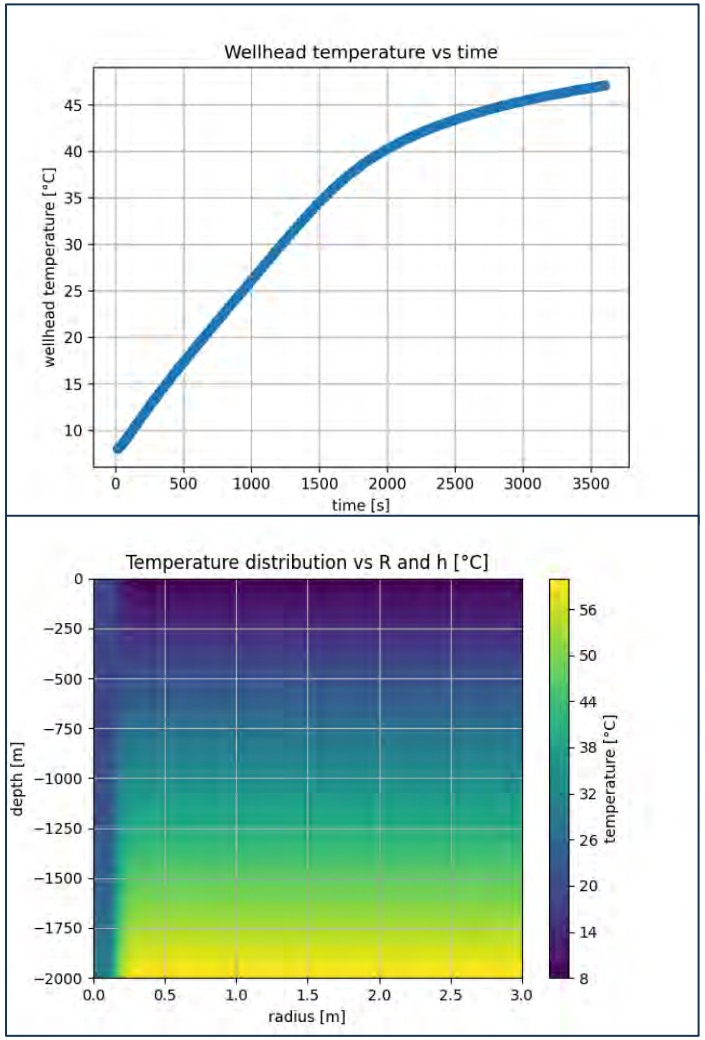
```python
def main():
#
    GeomWell = bib.wellGeometryIN(fileName) # fixing of initial temperatures in the input file - if neccessairy
    WWS = bib.workingScheme(fileName) # a well working scheme, matrix with 3 collumns, 1 - time [s], 2 - mass flow [kg/s]
    Others = bib.ReadOthers(fileName) # reads other important inputs (vactor with rows: 1-TDS [kg_TDS/kgH2O], well casing
    IN = bib.ProfTempStart(fileName, GeomWell) # supplementing of lack of temperatures in the initial state time = 0 sec
    Grid = bib.GridGener(IN, 30.0, 1.5) # Grid generator, IN - input file defined the well construction, (2) maximal allo
    R = Grid[0] # R nodes coordinate [m]
    Z = Grid[1] # Z nodes coordinate [m]
    NC = bib.nodesCoordinates(R, Z) # NC - nodes numbering and coordinates fixing, col. (1) - number of node, (2) - R coo
    t0 = bib.initialTemp(NC, IN) # initial temperature distribution in nodes, vector number of the row = number of node
    N2E = bib.nodes2elements(NC, R, Z) # affiliation of nodes to elements, vector Element[0,3] - 0 local node 0, 1 - loca
    NodesInElements = bib.NodesInElements(N2E)  # to make calculation faster the efect of the findNode(nNode,N2E) is bein
    # nNode - number of the node, output [e0, e2, e3, e4] - information where nNode is the first (e0), second (e1), third
    # Nodes numbering simillar to "nodes2elements(NC, R, Z) - see above"
    #
    am = bib.aMatFix(fileName) # temperature equalization factor [m²/s]
    Mat = bib.checkMat(IN,Grid,N2E,NC) # selecting and linking material to elements, by coordinates R and Z
    nFTBC = bib.FirstTypeBoundary(NC, N2E, Mat) # number of nodes with the first type boundary condition, fixed state nod
    actvN = bib.seekActvN(NC, nFTBC) # from all nodes NC[:,0] (vector contain number of nodes) only those are chosen whic
    NodesBottom = bib.NodesBottom(NC,Z,R) # number of nodes at the botton layer, where the 1st type boundary conditions a
    NodesTop = bib.NodesTop(NC,Z,R) # number of nodes at the Earth's surface, where the 1st type boundary conditions are
    NinWell = bib.nodesZones(Z, NC, N2E, Mat) # nodes number in a zone of a well, the list of lists, where element NinWel
    NfbB = bib.NodesFilledBrine(NinWell,N2E,Mat) # (N)odes(f)illed(b)y(B)rine - find only nodes belonging to elements ful
    DtimeMax = bib.maxDtime(NC,N2E,Mat,am,4.) # maximal alllowed value of time step [s]. The last value coresponding to c
    # Infact due to utilisation of the 3th type boundary condition the length of the timestep is much schorter, and defin
    DtimeMax = 40.
    Dtime = 10.
    t0 = bib.EqTzone(t0,NinWell) # equalize temperatures in the zone of a well. The important is temperature in the node
    twell0 = bib.tempInWell(NinWell, t0, NC) # temperature inside the well tz[n,m] - n the layer of nodes, from the top,
    # print(twell0)
    #
    print('► The suggested length of time step is : ',round(Dtime,2),' sec')
    #
    if (WWS[0,1] >= 0.):
        Result = bib.prodCalculations(WWS,Dtime,DtimeMax,t0,twell0,N2E,NC,Z,R,Mat,am,NinWell,NfbB,actvN,Others,NodesTop,N
    else:
```

Widok terminala podczas symulacji

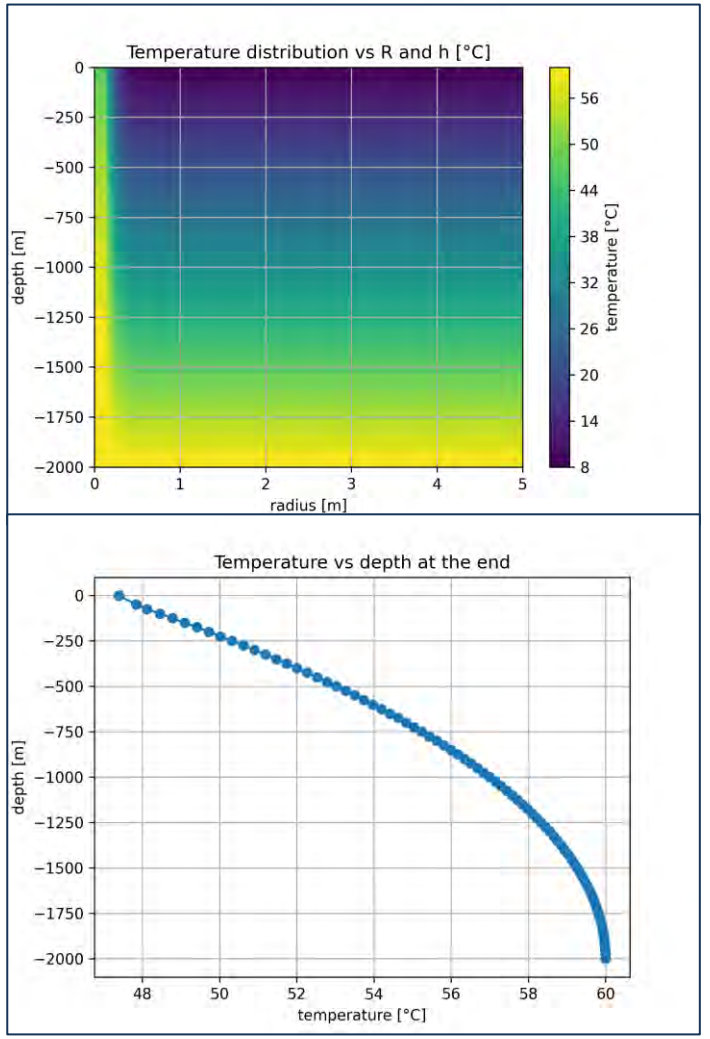
PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

```
-► 16.11 % of time ( 0.04 hr) | time step length:  20.0  sec | wellhead/bottom temp.  20.0 / 50.95 °C
-► 18.33 % of time ( 0.05 hr) | time step length:  20.0  sec | wellhead/bottom temp.  20.0 / 49.76 °C
-► 20.56 % of time ( 0.05 hr) | time step length:  20.0  sec | wellhead/bottom temp.  20.0 / 48.62 °C
-► 22.78 % of time ( 0.06 hr) | time step length:  20.0  sec | wellhead/bottom temp.  20.0 / 47.53 °C
-► 25.0 % of time ( 0.06 hr) | time step length:  20.0  sec | wellhead/bottom temp.  20.0 / 46.47 °C
-► 27.78 % of time ( 0.07 hr) | time step length:  25.0  sec | wellhead/bottom temp.  20.0 / 45.46 °C
```

Wyniki



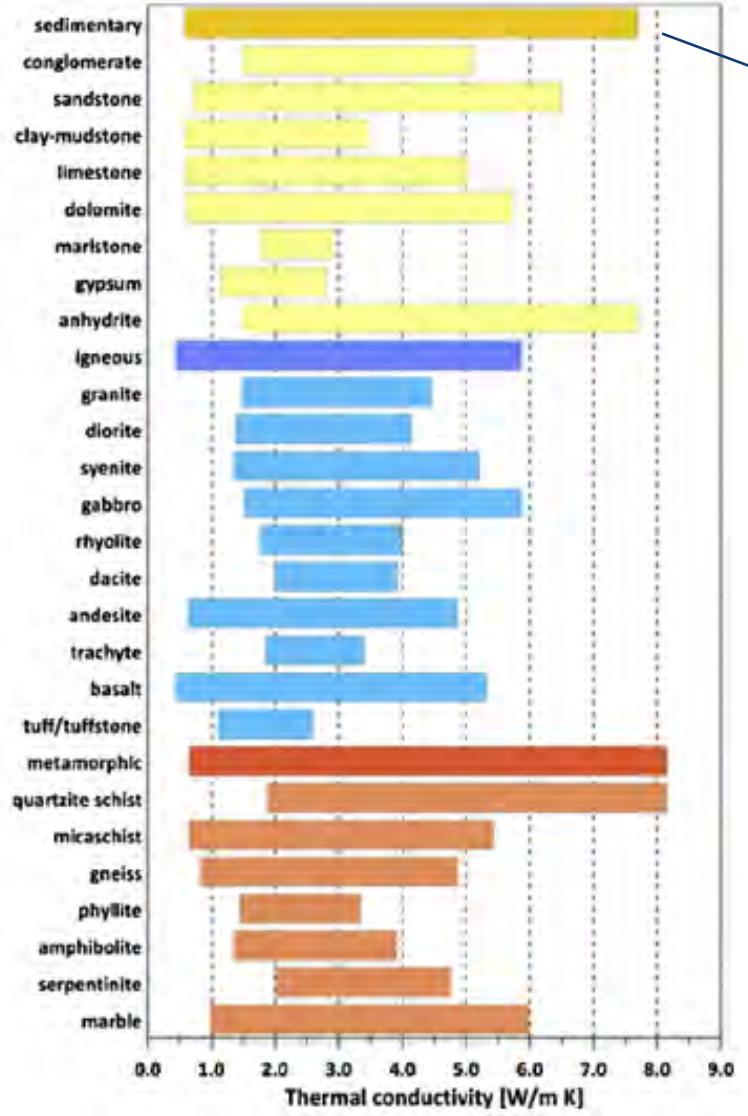
Otwór chłonny



Otwór produkcyjny

► dodatkowo plik excel (out.xlsx) wraz z rozkładem temperatury w węzłach obliczeniowych obszaru

Kalibracja modelu

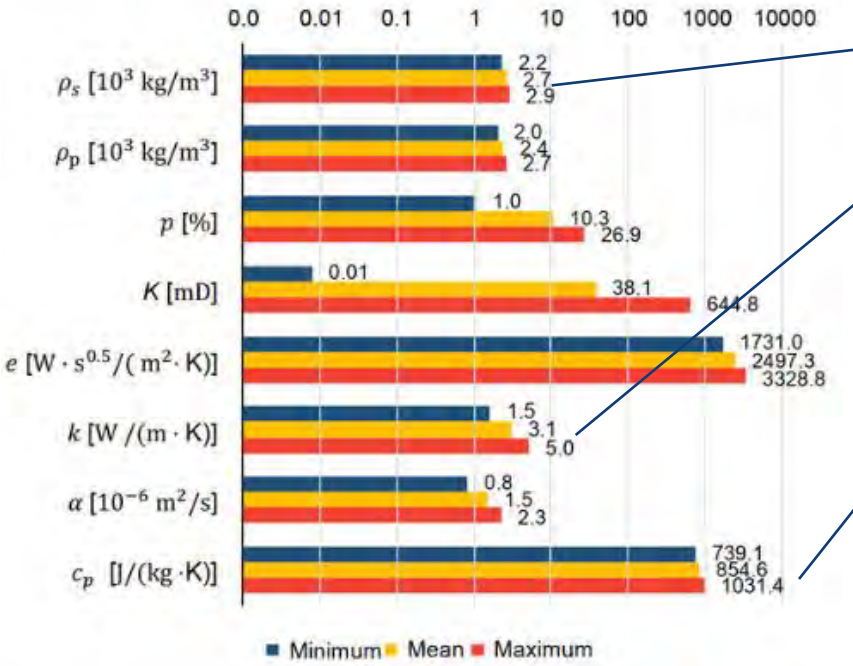


Thermal conductivity of different types of rocks.
Santa G. et al. 2019. An updated ground thermal properties database for GSHP applications

Dane materiałowe (mat)

	A	B	C	D	E
1	mat	λ[W/(mK)]	c[J/(kgK)]	r(kg/m3]	type
2	0	0,6	4190	1000	water
3	1	5,00	2000,00	3000,00	air
4	2	4,50	1100,00	2600,00	well surroundings
5	3	4,50	1100,00	2600,00	rocks 1

mat	λ[W/mK)]	c[J/(kgK)]	r[kg/m3]	type
1				rock 1
2				rock 2
3				rock 3
4				rock 4
5				rock 5
...				...
n				rock n



Descriptive statistics of selected parameters of the analyzed rocks.
Labus K. et al. Thermal properties of rocks from deep boreholes in Poland in terms of obtaining geothermal energy from enhanced geothermal systems

Użyteczność

- Na etapie szacowania zasobów geotermalnych oraz projektowanie ciepłowni geotermalnej, możliwość oszacowania temperatury głowicowej
- Przestój pracy otworu► jak szybko zostanie osiągnięta optymalna temp. głowicowa ►eksploatacja źródeł szczytowych
- Możliwość kalibracji wyników termicznych modelu z rzeczywistą temperaturą osiąganą na głowicy ($\lambda_{skały}$)

Podręcznik użytkownika

Kalkulator otworu geotermalnego
Przewodnik użytkownika

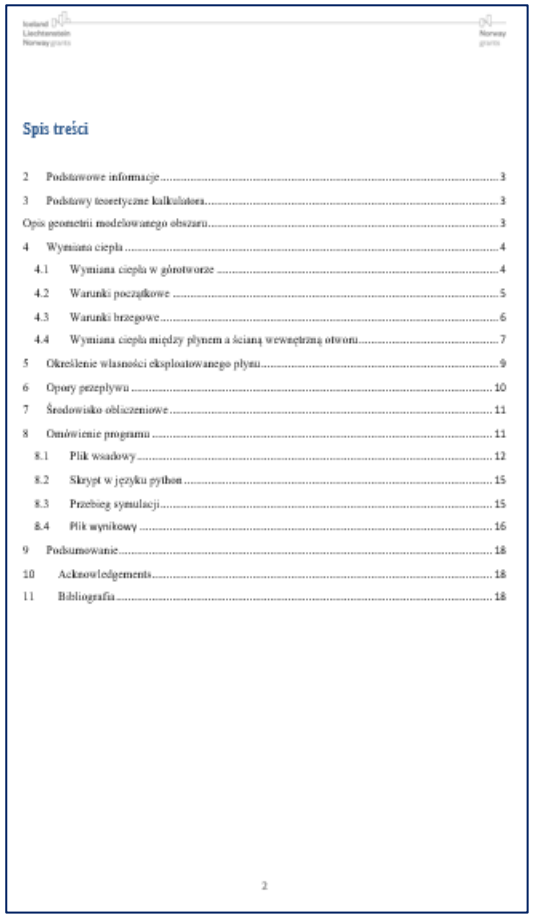
Autorzy:

Leszek Pająk
Maciej Miecznik
Karol Pierzchała

Instytut Gospodarki Surowcami Mineralnymi i Energią Polskiej Akademii Nauk

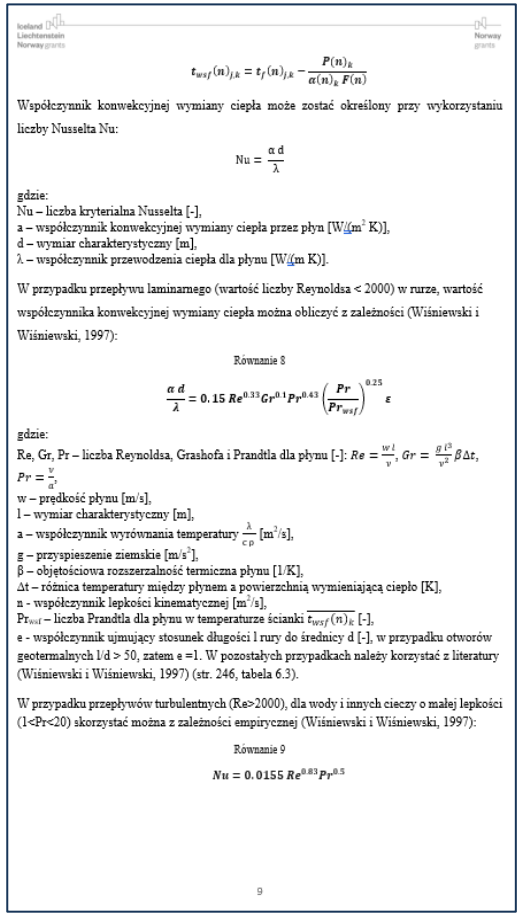
Projekt GeoModel (geomodel.pl)

Grudzień 2024



Spis treści

2 Podstawowe informacje ... 3
3 Podstawy teoretyczne kalkulatora ... 3
Opis geometrii modelowanego obszaru ... 3
4 Wymiana ciepła ... 4
4.1 Wymiana ciepła w górotworze ... 4
4.2 Warunki początkowe ... 5
4.3 Warunki brzegowe ... 6
4.4 Wymiana ciepła między płynem a ścianą wewnętrzną otworu ... 7
5 Określenie własności eksploatowanego płynu ... 9
6 Opory przepływu ... 10
7 Środowisko obliczeniowe ... 11
8 Omówienie programu ... 11
8.1 Plik wsadowy ... 12
8.2 Skrypt w języku python ... 15
8.3 Przebieg symulacji ... 15
8.4 Plik wynikowy ... 16
9 Podsumowanie ... 18
10 Acknowledgements ... 18
11 Bibliografia ... 18



$$t_{wsf}(n)_{j,k} = t_f(n)_{j,k} - \frac{P(n)_k}{\alpha(n)_k F(n)}$$

Współczynnik konwekcyjnej wymiany ciepła może zostać określony przy wykorzystaniu liczby Nusselta Nu:

$$\mathrm{Nu} = \frac{\alpha\, d}{\lambda}$$

gdzie:
Nu – liczba kryterialna Nusselta [-],
a – współczynnik konwekcyjnej wymiany ciepła przez płyn [W/(m^2 K)],
d – wymiar charakterystyczny [m],
λ – współczynnik przewodzenia ciepła dla płynu [W/(m K)].

W przypadku przepływu laminarnego (wartość liczby Reynoldsa < 2000) w rurze, wartość współczynnika konwekcyjnej wymiany ciepła można obliczyć z zależności (Wiśniewski i Wiśniewski, 1997):

Równanie 8

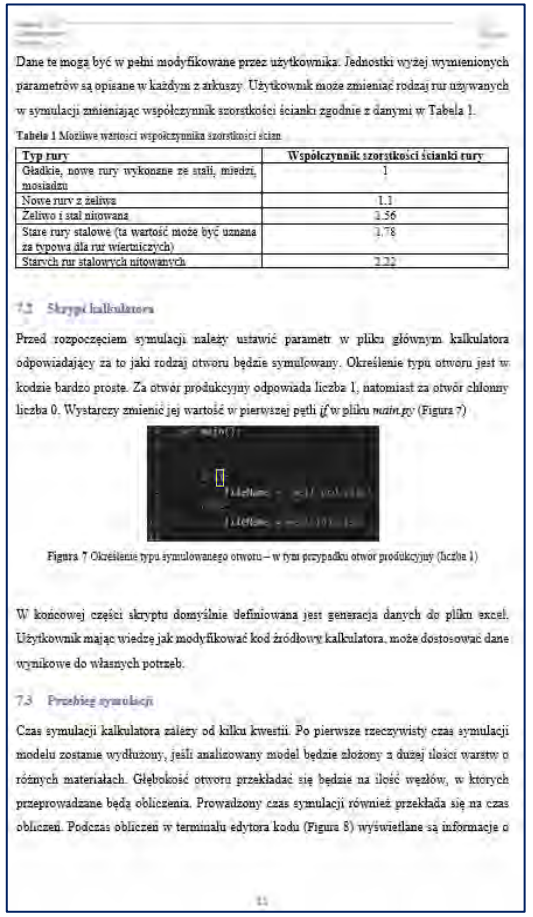
$$\frac{\alpha\, d}{\lambda} = 0.15\, Re^{0.33} Gr^{0.1} Pr^{0.43} \left(\frac{Pr}{Pr_{wsf}}\right)^{0.25} \varepsilon$$

gdzie:
Re, Gr, Pr – liczba Reynoldsa, Grashofa i Prandtla dla płynu [-]: $Re = \frac{w\,l}{\nu}$, $Gr = \frac{g\, l^3}{\nu^2}\beta\Delta t$, $Pr = \frac{\nu}{a^2}$,
w – prędkość płynu [m/s],
l – wymiar charakterystyczny [m],
a – współczynnik wyrównania temperatury $\frac{\lambda}{c\,\rho}$ [m^2/s],
g – przyspieszenie ziemskie [m/s^2],
β – objętościowa rozszerzalność termiczna płynu [1/K],
Δt – różnica temperatury między płynem a powierzchnią wymieniającą ciepło [K],
n - współczynnik lepkości kinematycznej [m^2/s],
Pr_{wsf} – liczba Prandtla dla płynu w temperaturze ścianki $\overline{t_{wsf}(n)_k}$ [-],
e - współczynnik ujmujący stosunek długości l rury do średnicy d [-], w przypadku otworów geotermalnych l/d > 50, zatem e =1. W pozostałych przypadkach należy korzystać z literatury (Wiśniewski i Wiśniewski, 1997) (str. 246, tabela 6.3).

W przypadku przepływów turbulentnych (Re>2000), dla wody i innych cieczy o małej lepkości (1<Pr<20) skorzystać można z zależności empirycznej (Wiśniewski i Wiśniewski, 1997):

Równanie 9

$$Nu = 0.0155\, Re^{0.83} Pr^{0.5}$$



Dane te mogą być w pełni modyfikowane przez użytkownika. Jednostki wyżej wymienionych parametrów są opisane w każdym z arkuszy. Użytkownik może zmieniać rodzaj rur używanych w symulacji zmieniając współczynnik szorstkości ścianki zgodnie z danymi w Tabela 1.

Tabela 1 Możliwe wartości współczynnika szorstkości ścian

Typ rury	Współczynnik szorstkości ścianki rury
Gładkie, nowe rury wykonane ze stali, miedzi, mosiądzu	1
Nowe rury z żeliwa	1.1
Żeliwo i stal nitowana	1.56
Stare rury stalowe (ta wartość może być uznana za typowa dla rur wiertniczych)	1.78
Starych rur stalowych nitowanych	2.22

7.2 Skrypt kalkulatora

Przed rozpoczęciem symulacji należy ustawić parametr w pliku głównym kalkulatora odpowiadający za to jaki rodzaj otworu będzie symulowany. Określenie typu otworu jest w kodzie bardzo proste. Za otwór produkcyjny odpowiada liczba 1, natomiast za otwór chłonny liczba 0. Wystarczy zmienić jej wartość w pierwszej pętli *if* w pliku *main.py* (Figura 7)

Figura 7 Określenie typu symulowanego otworu – w tym przypadku otwór produkcyjny (liczba 1)

W końcowej części skryptu domyślnie definiowana jest generacja danych do pliku excel. Użytkownik mając wiedzę jak modyfikować kod źródłowy kalkulatora, może dostosować dane wynikowe do własnych potrzeb.

7.3 Przebieg symulacji

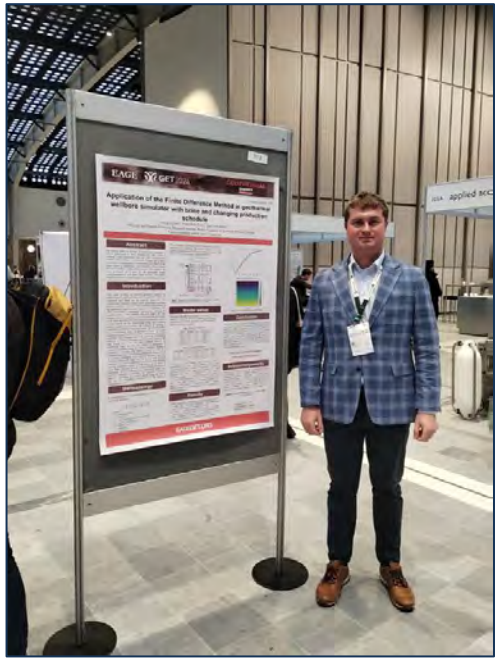
Czas symulacji kalkulatora zależy od kilku kwestii. Po pierwsze rzeczywisty czas symulacji modelu zostanie wydłużony, jeśli analizowany model będzie złożony z dużej ilości warstw o różnych materiałach. Głębokość otworu przekładać się będzie na ilość węzłów, w których przeprowadzane będą obliczenia. Prowadzony czas symulacji również przekłada się na czas obliczeń. Podczas obliczeń w terminalu edytora kodu (Figura 8) wyświetlane są informacje o

+ English version

Prezentacja narzędzia podczas konferencji

European Geothermal Workshops, Stavanger

Geothermal Innovation Days, Monachium



Energy Transition Conference & Exhibition (GET 2024), Rotterdam

Acknowledgements

Projekt Geomodel wraz z opracowanymi narzędziami został dofinansowany ze środków Funduszu Współpracy Dwustronnej Mechanizmu Finansowego EOG na lata 2014–2021 i Norweskiego Mechanizmu Finansowego 2014-2021 w ramach Programu Środowisko, Energia i Zmiany klimatu.

Więcej o projekcie:

https://geomodel.pl

Dziękuję za uwagę

Karol Pierzchała
Instytut Gospodarki Surowcami Mineralnymi i Energią
Polska Akademia Nauk
kpierzchala@min-pan.krakow.pl