ON THE USE OF PYTHON BASED CAPE-OPEN TOOLS AT IFPEN

Pipeline calculations using Python CAPE-OPEN Unit Operation and wax crystallization calculations using Python CAPE-OPEN Thermo Import package

MARTIN GAINVILLE (IFPEN)

CAPE-OPEN 2021 ANNUAL MEETING

OCTOBER 27-28, 2021

CONTENT

- Why using CAPE-OPEN for some developments at IFPEN?
- Application using « Python CAPE-OPEN Thermo Import »
- Application using « Python CAPE-OPEN Unit Operation »
- Conclusions

CAPE-OPEN FOR SOME DEVELOPMENTS AT IFPEN

- Generic/Standard-based developments are always recommended in multiple partner projects

- CAPE-OPEN provides a powerful structuring framework
  - To develop and share modules (Unit Operation, Thermo) that can be used in different CAPE-OPEN based process simulators[1]
  - To prototype some IFPEN advanced solutions delivered in collaborative, academic or EU projects
  - Recently ***Python-based Unit Operation*** and ***Python CAPE-OPEN Thermo Import package*** open new opportunities for fast developments of advanced solutions
  - COFE from amsterCHEM is an efficient backup solution in case some companies / academic organizations do no have already a process simulator or CAPE-OPEN based process simulators[1]

[1] also named Process Modeling Environment (PME)

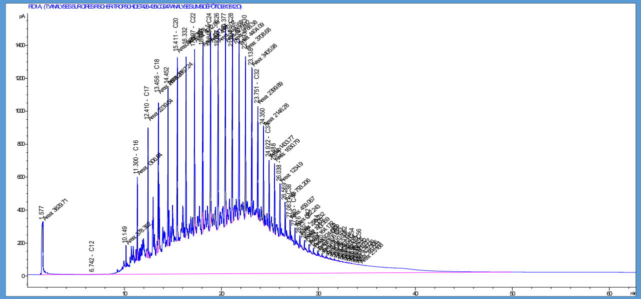
APPLICATION USING « PYTHON CAPE-OPEN THERMO IMPORT »

- Python CAPE-OPEN Thermo Import package developed by amsterCHEM tailor-made engineering software solutions
  - It allows for importing CAPE-OPEN version 1.1 Thermodynamic and Physical Property Packages into Python for Windows
  - It facilitates integration of PVT calculations into Python codes and Jupyter notebooks
    - It provides access to large sets of physical compounds, equations of state, flash types and physical property models
    - It allows to perform easily extensive thermodynamic calculations under various pressures, temperatures and enthalpies into Python advanced physical models

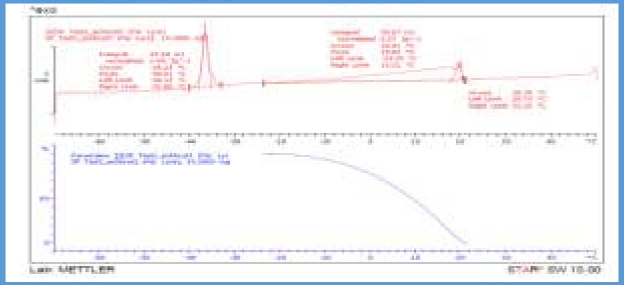
APPLICATION USING « PYTHON CAPE-OPEN THERMO IMPORT »

- Wax crystallization calculations using Multiflash®
  - Multiflash allows to calculate the solubility of n-paraffin compounds in waxy crudes

Gas Chromatography analyses allow to define compound lists and compositions



DSC analyses provide crystallization curves of n-paraffin compounds

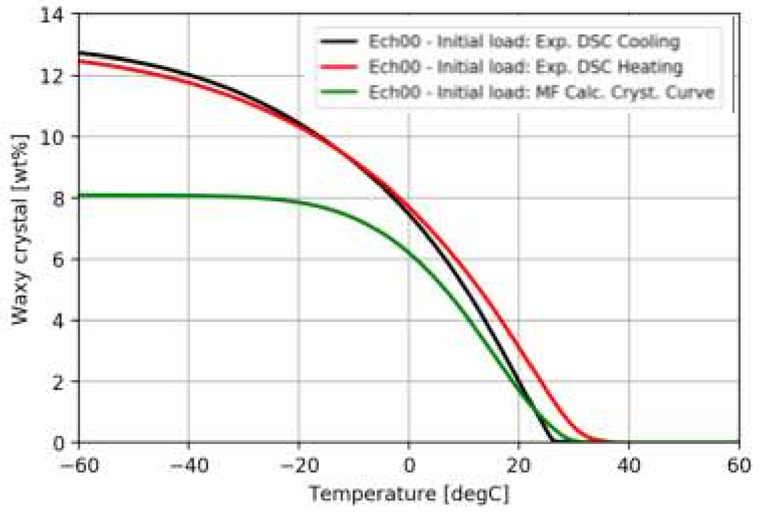


Input data (Compound list + composition)

Thermodynamic package Multiflash®

Simulated crystallization curve (mass_crys = f(T))

Exp. Crystallization curve



APPLICATION USING « PYTHON CAPE-OPEN THERMO IMPORT »

- Wax crystallization calculations using Multiflash®
  - Number of compounds: 121 from C6 to C56
  - Model Type: Wax
  - Thermo Model: RKSA

```python
import capeopen_thermo as co

co.license('XXXYYYY000')
system = co.System('Multiflash Property Package Manager')
pkg=system.create_package('MF_WAXY_BLEND_70_30_VV')
print([c.name for c in pkg.compounds])
T0 = 303.15
P0 = 1000000
nbComp = len(pkg.compounds)
C0 = [(1/nbComp) for i in range(nbComp)]
eq0=pkg.create_phase_equilibrium(C0,'temperature',T0,'pressure',P0)
for k in eq0.phases:
   print('Phase state:',k.state_of_aggregation, ', Fraction:', k.phase_fraction, ', Composition:', k.x)
```

Define property package manager

Load Property Package

Perform P,T equilibrium calculation

Iterate on existing phases

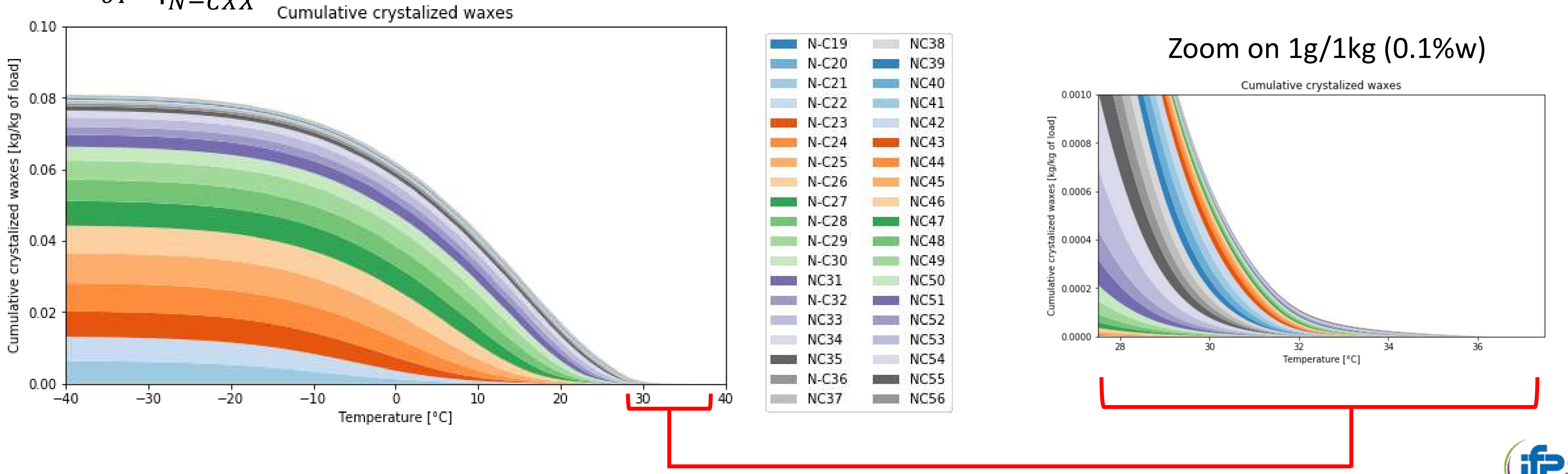
```
['N-HEXANE', 'METHYLCYCLOPENTANE', 'BENZENE', 'CYCLOHEXANE', 'N-
HEPTANE', '2,2-DIMETHYLPENTANE', 'DIMETHYL-CYCLOPENTANE',
'METHYLCYCLOHEXANE_I', '2-METHYLHEXANE', '3-ETHYLPENTANE',
TOLUENE', 'N-OCTANE', '2,2-DIMETHYLHEXANE', '2,5-
DIMETHYLHEXANE', '2,3-DIMETHYLHEXANE', '2-
THYLHEPTANE', '2,3,4-TRIMETHYLPENTANE',
NE', '1,1-DIMETHYLCYCLOHEXANE', 'ETHYLBENZENE',
XYLENE', 'N-NONANE', 'PS9', 'N-DECANE', 'PS10', 'N-
12', 'N-C13', 'PS13', 'N-C14', 'PS14', 'N-C15', 'PS15',
'NC16', 'PS16', 'NC17', 'PS17', 'NC18', 'PS18', 'NC19', 'PS19', 'NC20', 'PS20',
PS21', 'NC22', 'PS22', 'NC23', 'PS23', 'NC24', 'PS24', 'NC25', 'PS25',
S26', 'NC27', 'PS27', 'NC28', 'PS28', 'NC29', 'PS29', 'NC30', 'PS30',
S31', 'NC32', 'PS32', 'NC33', 'PS33', 'NC34', 'PS34', 'NC35', 'PS35',
S36', 'NC37', 'PS37', 'NC38', 'PS38', 'NC39', 'PS39', 'NC40', 'PS40',
S41', 'NC42', 'PS42', 'NC43', 'PS43', 'NC44', 'PS44', 'NC45', 'PS45',
'NC47', 'PS47', 'NC48', 'PS48', 'NC49', 'PS49', 'NC50', 'PS50',
C52', 'PS52', 'NC53', 'PS53', 'NC54', 'PS54', 'NC55', 'PS55',
```

```
liquid , Fraction: 0.7739, Composition:
.01068, 0.01176, ..., 2.091e-07, 0.01068]
Phase state: solid , Fraction: 0.2261, Composition: [0.0,
0.0, 0.0, ..., 0.03654, 0.0]
```

APPLICATION USING « PYTHON CAPE-OPEN THERMO IMPORT »

- Wax crystallization calculations using Multiflash®
  - The individual n-paraffin crystallization within the blend can easily be calculated in Python by calls to Multiflash through CAPE-OPEN interfaces
  - N-paraffin crystallization rate can be used to model wax deposition mechanism in pipe flow modules

$\left.\frac{\partial C_{crys}}{\partial T}\right|_{N-CXX}$ for the load



APPLICATION USING « PYTHON CAPE-OPEN UNIT OPERATION »

- Python Unit Operation developed amsterCHEM tailor-made engineering software solutions
  - It is a CAPE-OPEN based unit operation, for which the calculations are performed with a Python script
  - Easy to develop and to plug into Process Modeling Environment

- Purpose of using Python Unit Operation for pipeline developments
  - IFPEN has an history of developing hydrodynamic models representing multiphase hydrodynamic regimes in pipes carrying gas/oil/water mixtures
  - Python UO permits to experiment with the models in a process simulator like the representation of an O&G production network

- Interoperability tests were performed with a simple pipeline flow model
  - Python Unit Operation (1.0.10.0) with Python (3.8.10 x64)
  - COFE (3.5.0.10 x64) and PRO/II (10.2.3 x64)
  - CO Physical Property Packages
    - TEA and MULTIFLASH (7.1.28 x64)

TEA

Multiflash®

COFE

PRO/II

APPLICATION USING « PYTHON CAPE-OPEN UNIT OPERATION »

- Simple pipeline flow implementation in the Python Unit Operation

1- Get access to inlet Material Object

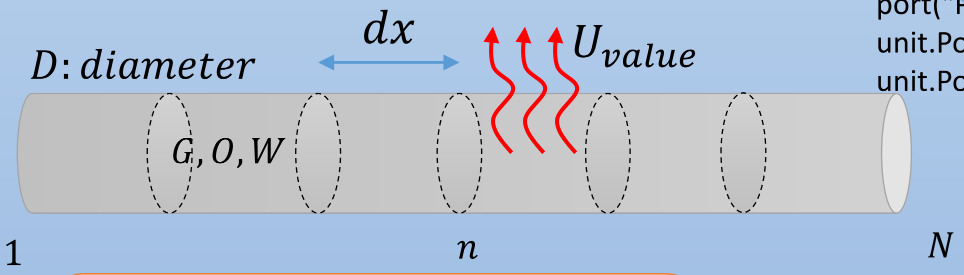
fd = unit.ports["Feed"]

2- Get inlet stream information

$$F_{mol} = \text{fd.flow\_rate}$$
$$\text{Cmol}_{\text{in}} = \text{fd.x}$$
$$\text{Pin} = \text{fd.p}$$
$$\text{hin} = \text{fd.get\_property('enthalpy')}$$

4- Pipe flow sequential calculations along cells

port("Feed",
unit.PortType.MATERIAL,
unit.PortDirection.FEED)



port("Product",
unit.PortType.MATERIAL,
unit.PortDirection.PRODUCT)

$$\frac{\Delta P}{dx} = \frac{1}{2}\,\rho_m V_m^2 \,\frac{f_D\left(Re_m, \frac{\varepsilon}{D}\right)}{D}$$

$$\frac{\Delta h}{dx} = -\pi\, D\, U_{value}\,\frac{(T_n - T_{Ambiant})}{F_{mol}}$$

CAPE-OPEN Parameters: diameter, lengh, …

3- Create new equilibrium (PT, PH, …) and calculate local conditions

eq = fd.create_phase_equilibrium(Cmol,'enthalpy', hin,'pressure',Pin)

$$\text{Molar\_Volume} = \sum_{k\ in\ eq.phases} \frac{\text{k.phase\_fraction}}{\text{k.get\_property('density')}}$$

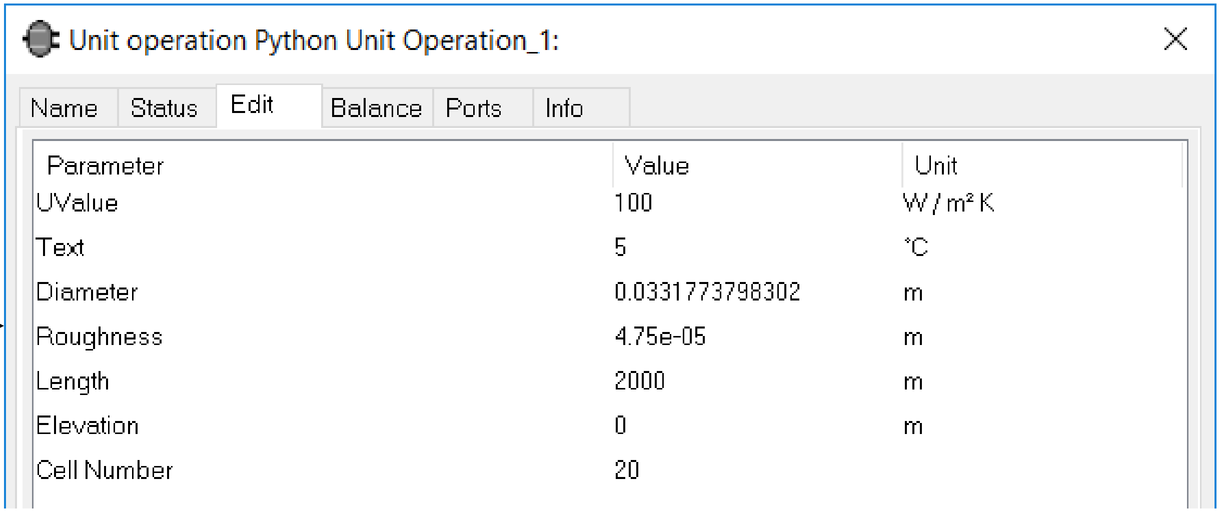
$$\text{Molar\_Mass} = \sum_{k\ in\ eq.phases} \text{k.phase\_fraction}\ \frac{\text{k.get\_property('molecularWeight')}}{1000}$$

$$\rho_m = \frac{Molar\_Mass}{Molar\_Volume}$$

$$V_m = \frac{Molar\_Volume\ \ fd.flow\_rate}{\frac{\pi\, D^2}{4}}$$

APPLICATION USING « PYTHON CAPE-OPEN UNIT OPERATION »

- Preliminaty interoperability tests
  - $CO_2 + H_2O$ + DODECANE mixture
    - Cmol = (0.2, 0.5, 0.3)
  - Pin = 20 bar, Tin = 20 °C, Fin = 1 kg/s
  - Pipe CO parameters


Unit operation Python Unit Operation_1:

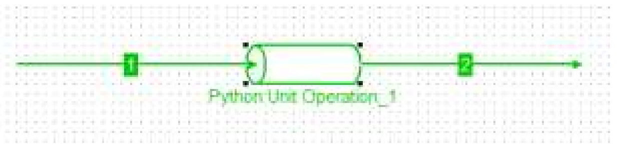
Parameter	Value	Unit
UValue	100	W / m² K
Text	5	°C
Diameter	0.0331773798302	m
Roughness	4.75e-05	m
Length	2000	m
Elevation	0	m
Cell Number	20	

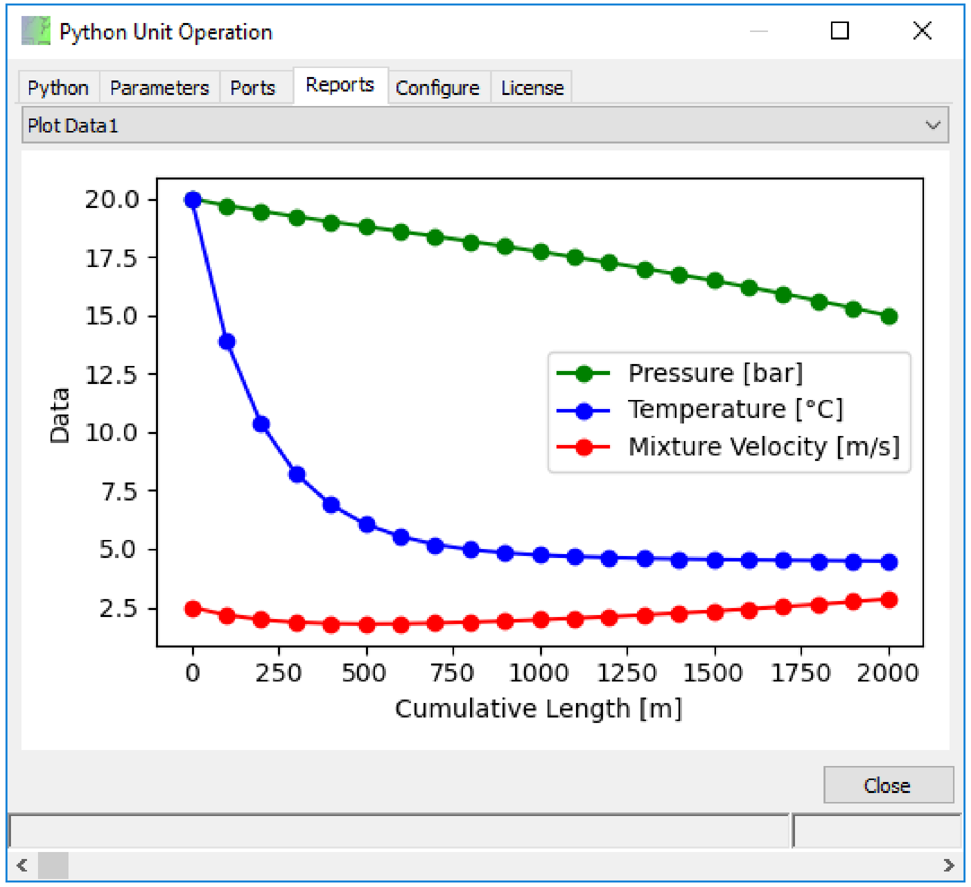
CASE	PME	THERMO PACKAGE	SUPPORTED PHASES	RESULTS
1	COFE	TEA	2: G,O	OK (rmk: only one phase present for PT conditions)
2	COFE	MULTIFLASH	2: G,O	OK (rmk: only one phase present for PT conditions)
2	COFE	MULTIFLASH	3: G,O,W	OK
3	PROII	TEA	2: G,O	OK (results identical to case 1 ✓)
4	PROII	MULTIFLASH	3: G,O,W	NOK with any CO OUs, OK with PROII UOs
5	PROII	MULTIFLASH	2: G,O	OK
6	PROII	PROII Thermo (PR)	2: G,O	OK (issue with Python UO plot report ⚠)

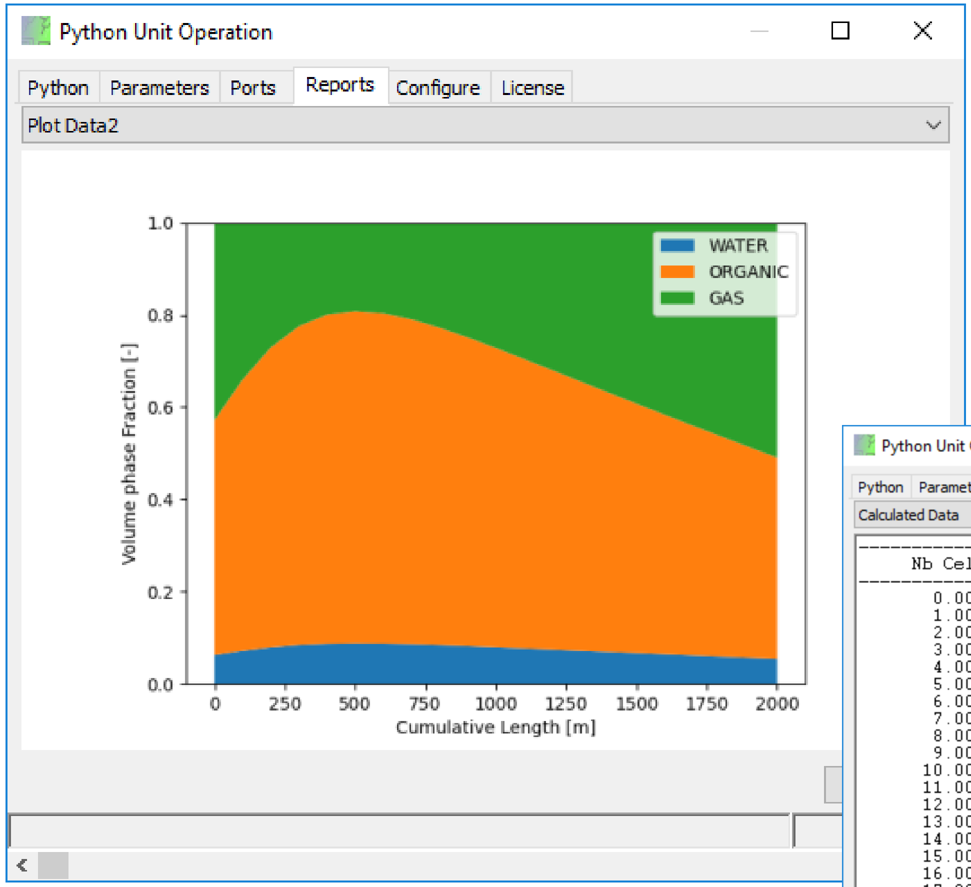
➔ *Preliminary tests, issues need to be further investigated*

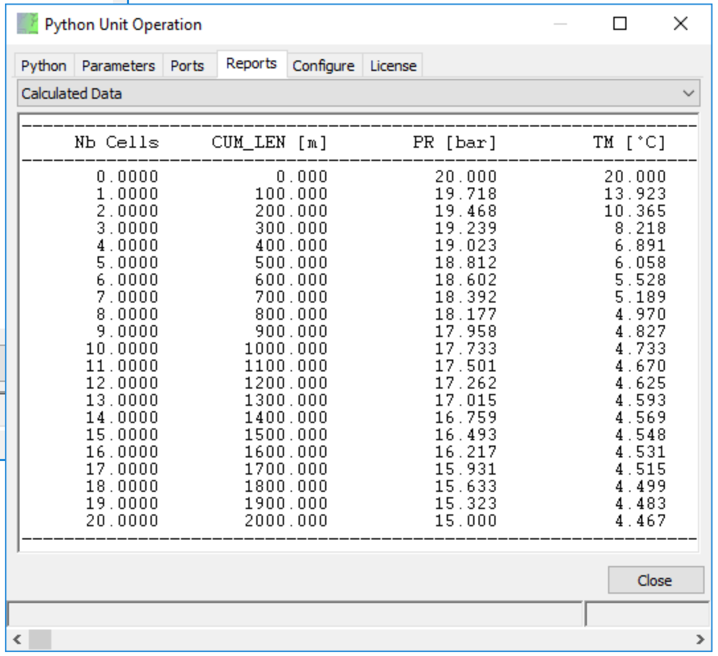
APPLICATION USING « PYTHON CAPE-OPEN UNIT OPERATION »

- Results: Python Pipe OU operation + Multiflash 3Ph in COFE





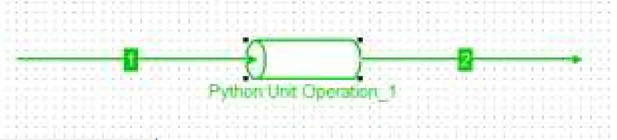




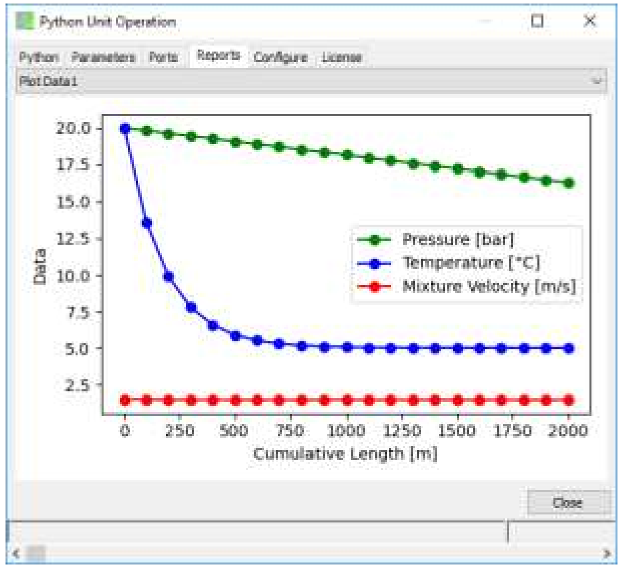
```
----------------------------------------------------------------------
     Nb Cells      CUM_LEN [m]         PR [bar]          TM [°C]
----------------------------------------------------------------------
       0.0000            0.000           20.000           20.000
       1.0000          100.000           19.718           13.923
       2.0000          200.000           19.468           10.365
       3.0000          300.000           19.239            8.218
       4.0000          400.000           19.023            6.891
       5.0000          500.000           18.812            6.058
       6.0000          600.000           18.602            5.528
       7.0000          700.000           18.392            5.189
       8.0000          800.000           18.177            4.970
       9.0000          900.000           17.958            4.827
      10.0000         1000.000           17.733            4.733
      11.0000         1100.000           17.501            4.670
      12.0000         1200.000           17.262            4.625
      13.0000         1300.000           17.015            4.593
      14.0000         1400.000           16.759            4.569
      15.0000         1500.000           16.493            4.548
      16.0000         1600.000           16.217            4.531
      17.0000         1700.000           15.931            4.515
      18.0000         1800.000           15.633            4.499
      19.0000         1900.000           15.323            4.483
      20.0000         2000.000           15.000            4.467
----------------------------------------------------------------------
```

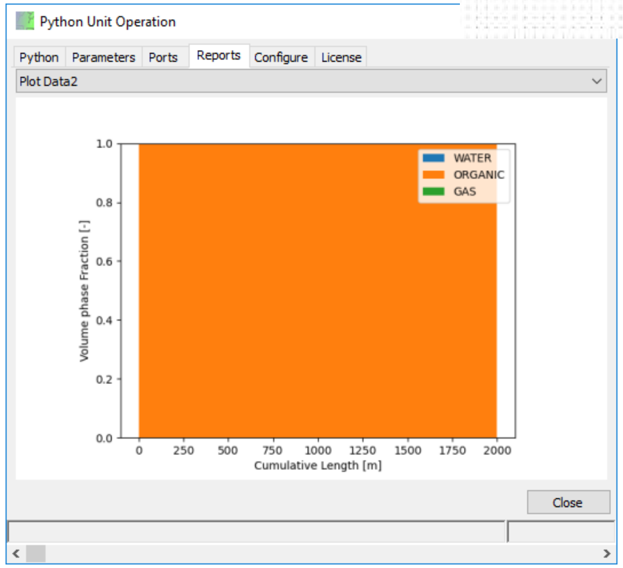
APPLICATION USING « PYTHON CAPE-OPEN UNIT OPERATION »

Results: Python Pipe OU operation + Multiflash 2Ph in COFE

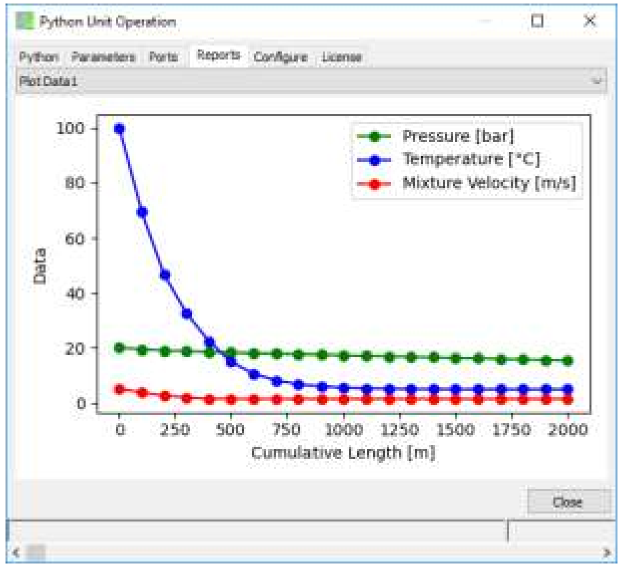


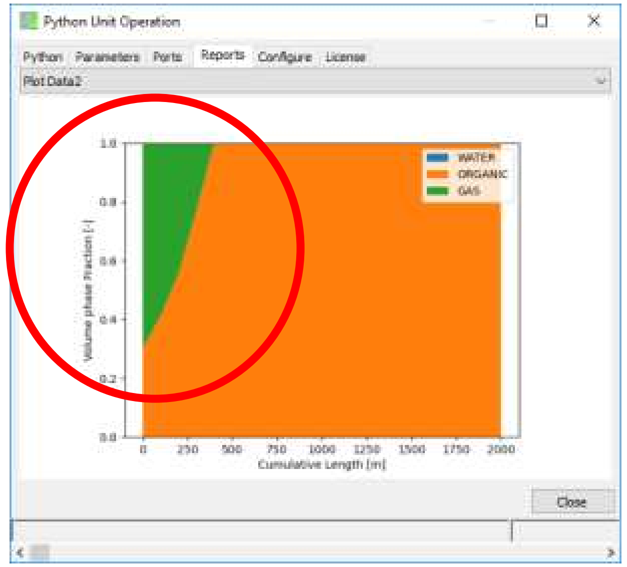
Tin = 20 °C: only organic phase present in the line





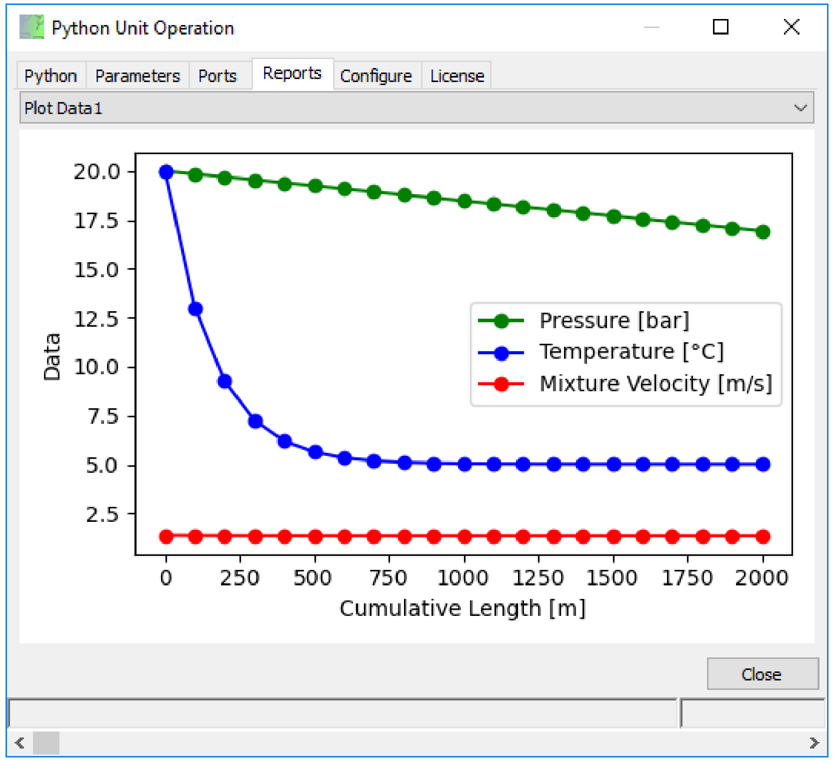
Tin = 100 °C: two-phase conditions at the inlet of the line

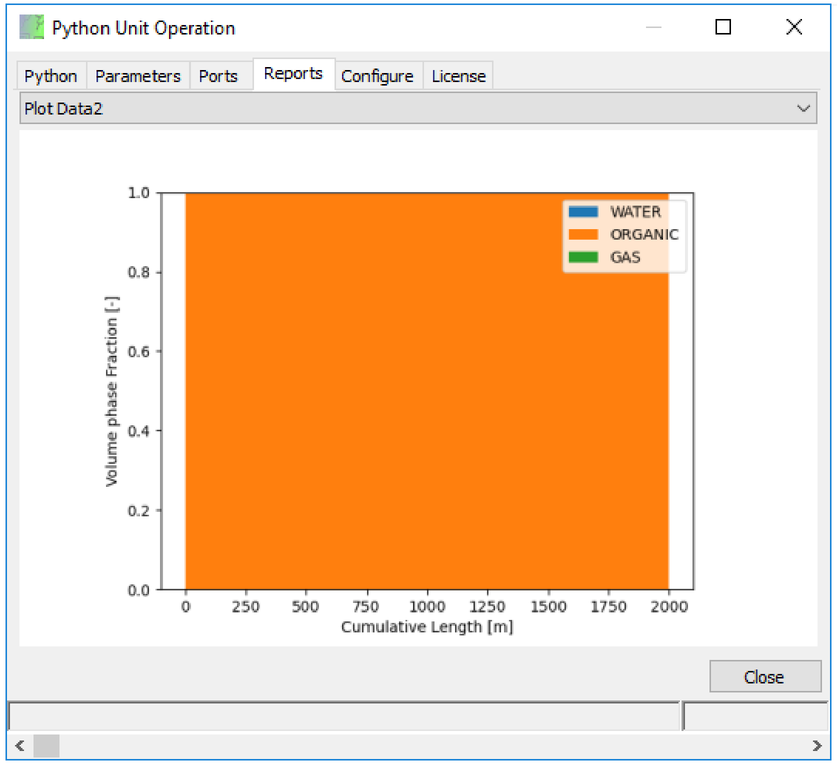




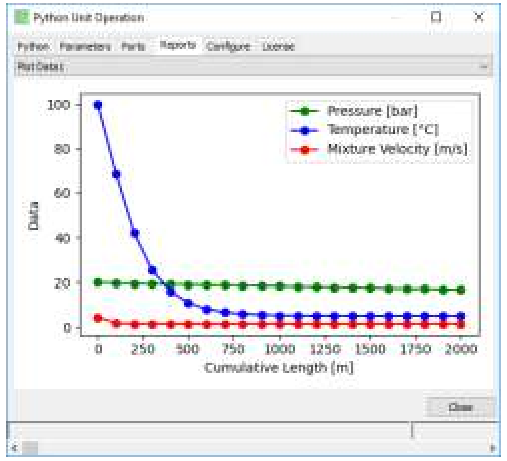
APPLICATION USING « PYTHON CAPE-OPEN UNIT OPERATION »

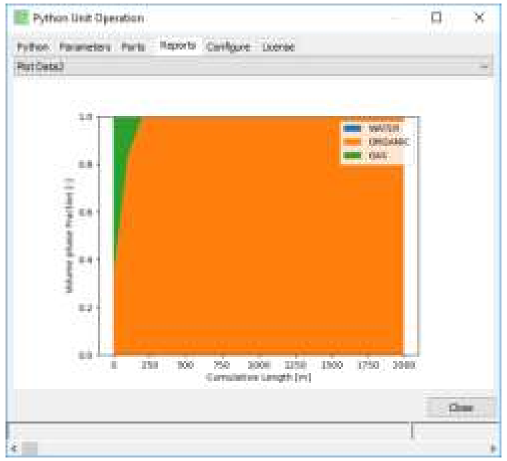
● Results: Python Pipe OU operation + TEA in PROII





Tin = 100 °C: two-phase conditions at the inlet of the line





➔ Identical results as ones obtained with COFE and TEA ✓

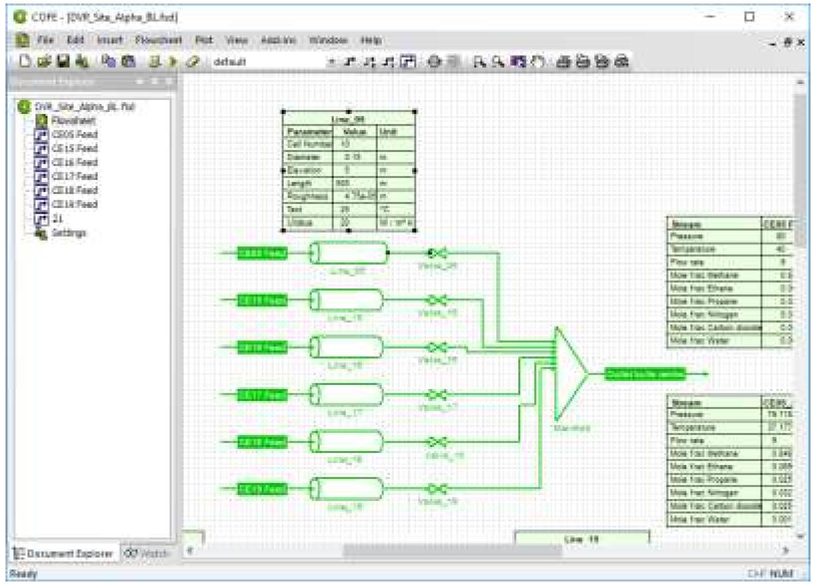
➔ Tests also performed with some PRO/II thermodynamic libraries ✓

👍 Using PRO/II permits to obtain consistency with the thermodynamic libraries usually used at IFPEN

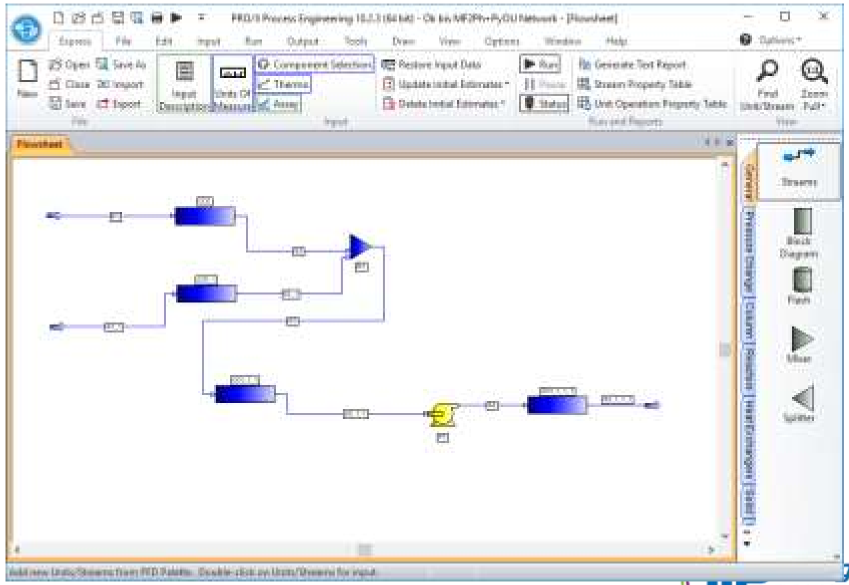
CONCLUSIONS

- ***Python*** is an easy programming language with useful and powerful packages (matplotlib, numpy, scipy, pandas, …)
- ***Python CO Unit Operation*** and ***Python CO Thermo Import package*** allow to develop easily CAPE-OPEN tailor-made components
- Complex systems can be further developed and tested in Process Modeling Environment

Simulation of gas production network using Python CO Unit Operation within COFE



Simulation of pipe network using Python CO Unit Operation within PRO/II and using Multiflash thermodynamics



ACKNOWLEDGMENTS

- Thanks to Jasper van Baten (amsterCHEM) for his support, reactivity and for providing all these useful CAPE-OPEN compliant components

- Links
  - www.amsterchem.com/pythonthermo.html
  - www.amsterchem.com/pythonunitoperation.html

Innovating for energy

Find us on:

🌐 www.ifpenergiesnouvelles.com

@IFPENinnovation