EVALUATION OF AN OPEN-SOURCE CHEMICAL PROCESS SIMULATOR USING A PLANT-WIDE OIL AND GAS SEPARATION PLANT FLOWSHEET MODEL AS BASIS

Presentation for Simulate365 user meeting 31-05-2022

Anders Andreasen, Technical Manager, Process

RAMBOLL

Bright ideas. Sustainable change.

DWSIM

OUTLINE

- Short bio
- My history with DWSIM
- Basis of investigated simulation model
- Implementation in DWSIM
- Python interface
- Results
- Other projects with DWSIM

SPEAKER BIO

- M.Sc. Chem. Eng. Aalborg University on Microkinetic modelling
- Ph.D. Chem. Eng. Technical University of Denmark on Hydrogen storage
- 7 years at MAN Energy Solutions in large diesel engine R&D
- 10 years at Ramboll Energy, working with oil & gas –> sustainable energy transition
- Extensive use of commercial simulation software e.g. Design II, Honeywell Unisim, Aspen HYSYS/Aspen Plus, LedaFlow

MY HISTORY WITH DWSIM

https://doi.org/10.3311/PPch.19678 | 503
Creative Commons Attribution
Periodica Polytechnica Chemical Engineering, 66(3), pp. 503–511, 2022

Evaluation of an Open-source Chemical Process Simulator Using a Plant-wide Oil and Gas Separation Plant Flowsheet Model as Basis

Anders Andreasen[1*]

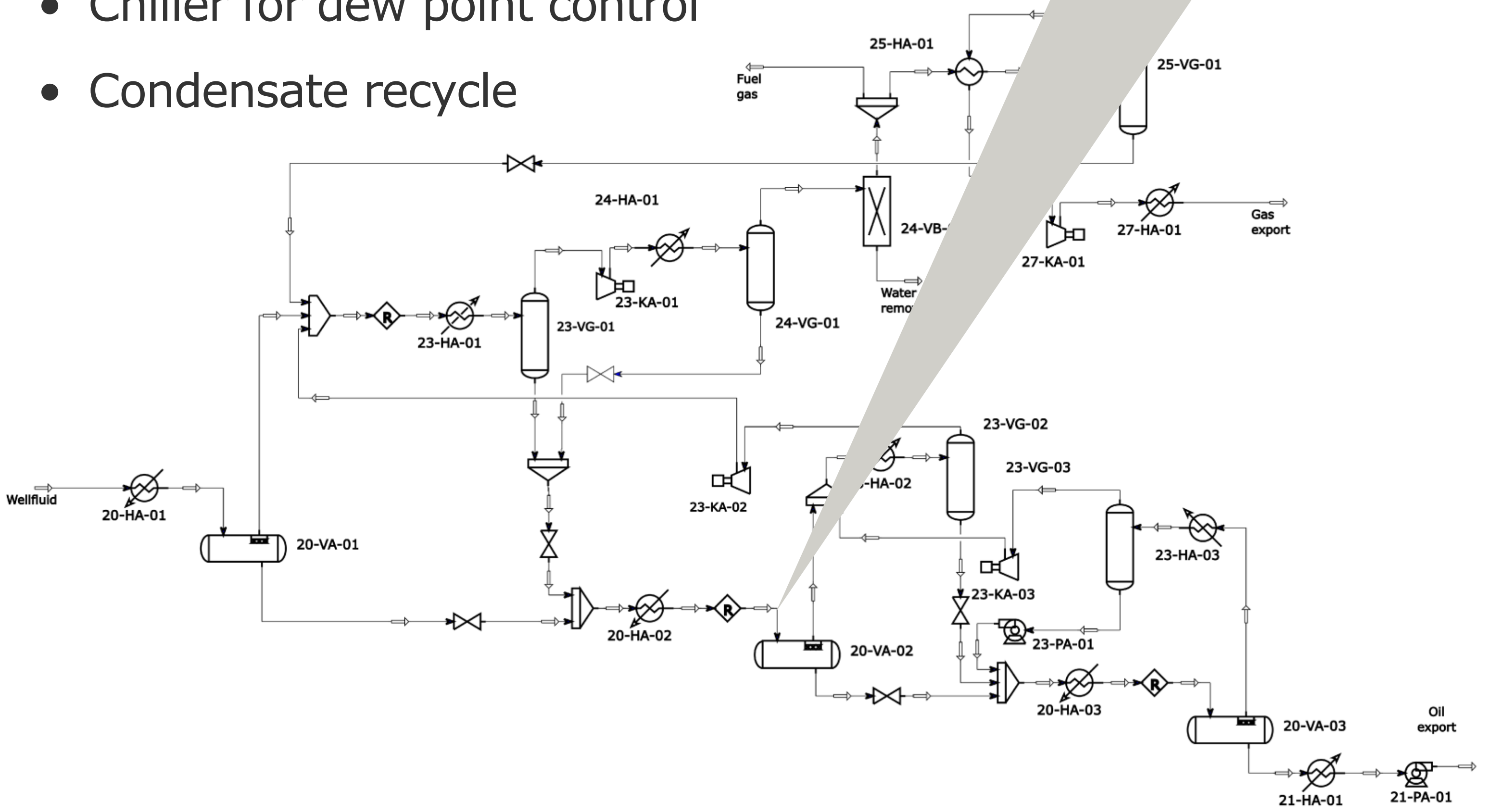
- I have been aware of DWSIM since v4
- I started using DWSIM at v6
- Started doing simple 3ph flash tests
- Daniel made major improvements in the stability of the flash algorithm late 2020 / early 2021
- Made small contributions to the code here and there
- Discussions with Armin led me to the benchmark case
- Simulation benchmark led to the paper being presented today: https://doi.org/10.3311/PPch.19678
- Additional information, simulation files and scripts included:

https://github.com/andr1976/dwsim-paper

SIMULATION MODEL BASIS

- 3 stage Oil & Gas separation
- Flash gas recompression
- Chiller for dew point control
- Condensate recycle

Classical question: Which separator pressures are optimal



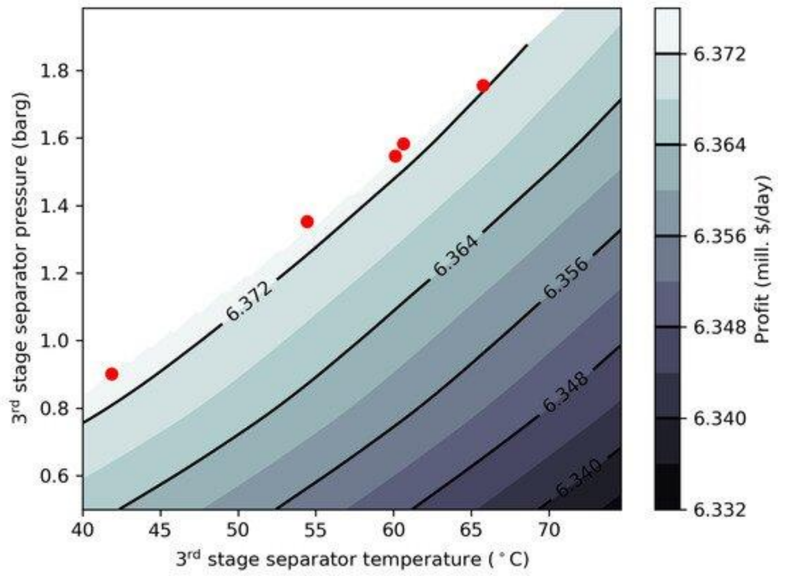
Component	Mole Fraction (%)
H_2O	0.0
N_2	0.0
CO_2	1.5870
CH_4	52.51
C_2H_6	6.24
C_3H_8	4.23
i-C_4H_{10}	0.855
n-C_4H_{10}	2.213
i-C_5H_{12}	1.1240
n-C_5H_{12}	1.271
n-C_5H_{12}	2.2890
C_{7+*}-CUT1	0.8501
C_{7+*}-CUT2	1.2802
C_{7+*}-CUT3	1.6603
C_{7+*}-CUT4	6.5311
C_{7+*}-CUT5	6.3311
C_{7+*}-CUT6	4.9618
C_{7+*}-CUT7	2.9105
C_{7+*}-CUT8	3.0505

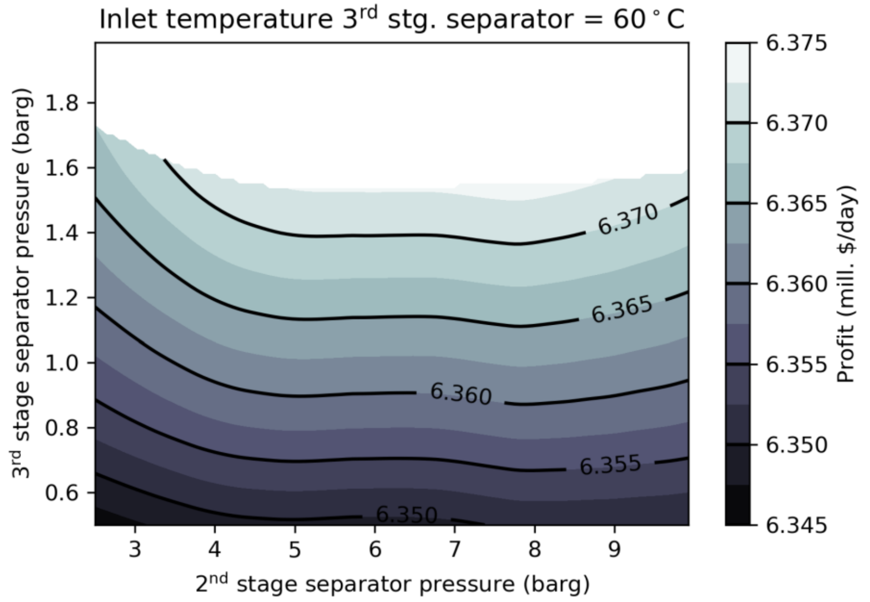
RAMBOLL

Reference case published in: Andreasen, A. Applied Process Simulation-Driven Oil and Gas Separation Plant Optimization Using Surrogate Modeling and Evolutionary Algorithms. *ChemEngineering* **2020**, *4*, 11. https://doi.org/10.3390/chemengineering4010011

SIMULATION MODEL BASIS

- Purpose: find optimal process settings for maximizing operating profit / revenue using global black-box optimisation with non-linear constraints and bounded variables
- Using the NSGA-II evolutionary algorithm (and many others too)
- At the time the model optimisation was state-of-the-art in terms of parameters and the separation plant complexity
- Generally, colder 1. stage and warmer 3. stage maximised profit due to higher oil production
- More interestingly: Some settings had dual/multiple optimal settings





RAMBØLL

SIMULATION DESIGN BASIS

- Peng-Robinson equation of state
- EOS liquid density
- Pseudo components properties taken from HYSYS
- 10 process variables
- Software: DWSIM 6.5.4 (or newer)

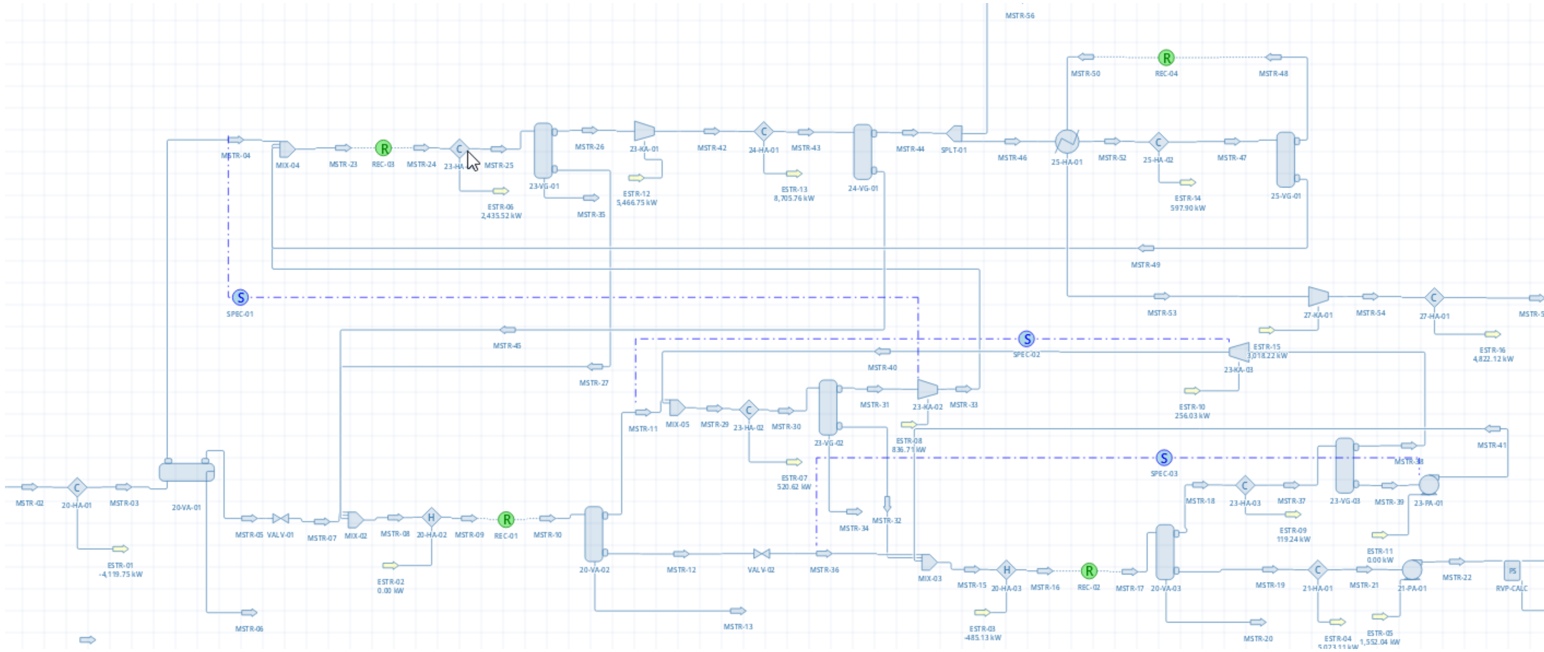
Table 2 Pseudo-component properties

MW kg/kmol	ρ_{liquid} kg/m^3	T_c °C	P_c barg*	V_c m^3/kmol	ω –
108.47	741.1	302.5	26.88	0.4470	0.3265
120.40	755.0	326.3	24.90	0.4940	0.3631
133.63	769.5	351.2	23.04	0.5464	0.4021
164.78	799.0	394.9	20.62	0.6359	0.4654
215.94	838.7	454.0	18.01	0.7636	0.5594
274.34	875.4	517.5	15.33	0.9290	0.6870
334.92	907.3	574.5	13.40	1.0842	0.8157
412.79	957.5	650.2	12.22	1.2285	0.9723

Parameters	Unit	Base Case
$T_{\text{Sep 1}}$	(°C)	70
$P_{\text{Sep 1}}$	(barg)	32
$P_{\text{Sep 2}}$	(barg)	8
$T_{\text{Sep 3}}$	(°C)	65
$P_{\text{Sep 3}}$	(barg)	1.5
$T_{\text{Scrub 1}}$	(°C)	32
$T_{\text{Scrub 2}}$	(°C)	32
$T_{\text{Scrub 3}}$	(°C)	32
$P_{\text{Comp 1}}$	(barg)	90
T_{Refrig}	(°C)	10

DWSIM IMPLEMENTATION

- Separators, H-X, valves, pumps, compressors, recycles, specification block used.
- Spreadsheet used for energy balances
- Python unit-op made to calculate oil RVP
- Python model interface made to enable multi-parameter variation study



PYTHON ADD-ONS

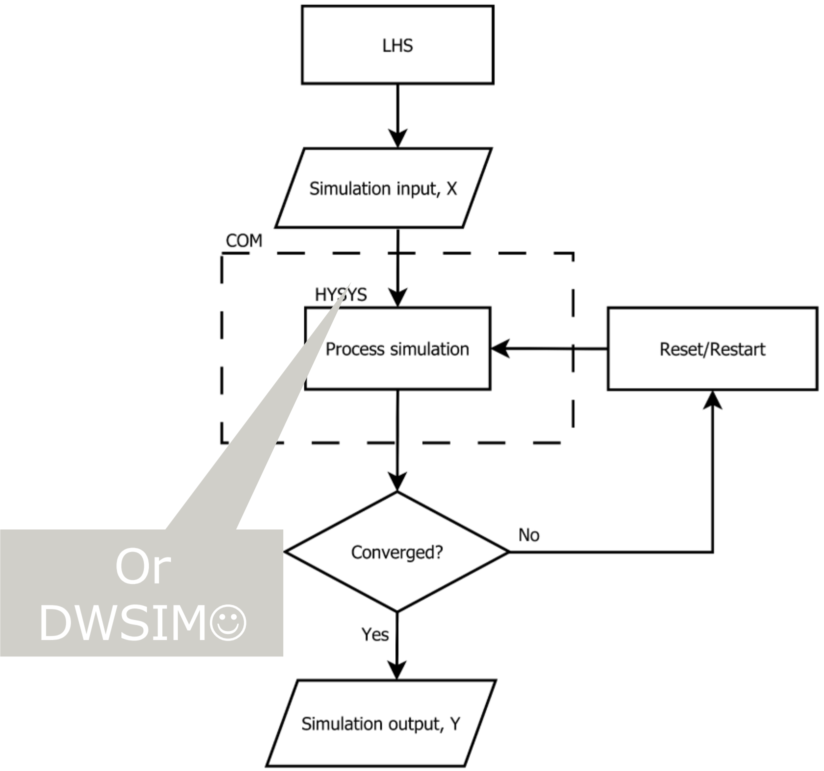
- **RVP code**
- Python interface

```
import clr
import sys

clr.AddReference("DWSIM.Thermodynamics")

from System import *
from DWSIM.Thermodynamics import *

feed = ims1
comp = feed.GetOverallComposition()
W = feed.GetMassFlow()
outflow = oms1

P = 0.5*1.013e5
outflow.Clear()
#outflow.PropertyPackage.DW_CalcEquilibrium(PropertyPackages.FlashSpec.T, PropertyPackages.FlashSpec.P)
outflow.SetTemperature(310.93)
outflow.SetPressure(P)
outflow.SetOverallComposition(comp)
outflow.SetMassFlow(W)
outflow.Calculate(True, True)

#gas = outflow.GetPhase('Vapor')
volflow = outflow.GetVolumetricFlow()
cont = True
count = 1
liq_flow = outflow.Phases[1].Properties.volumetric_flow
gas_flow = outflow.Phases[2].Properties.volumetric_flow
ratio = gas_flow/liq_flow
diff = 4-ratio

while cont:
    if abs(diff) < 0.01 or count > 100:
        break
    if gas_flow:
        if liq_flow:
            ratio = gas_flow/liq_flow
            diff = 4 - ratio
            if ratio > 4:
                P = P + abs(diff)*1e4
            else:
                P = P - abs(diff)*1e4
        else:
            P = P*2
    else:
        P=P*0.5
    count = count +1
    outflow.SetPressure(P)
    outflow.Calculate(True, True)
    liq_flow = outflow.Phases[1].Properties.volumetric_flow
    gas_flow = outflow.Phases[2].Properties.volumetric_flow
```

RAMBOLL

PYTHON ADD-ONS

- RVP code
- **Python interface**

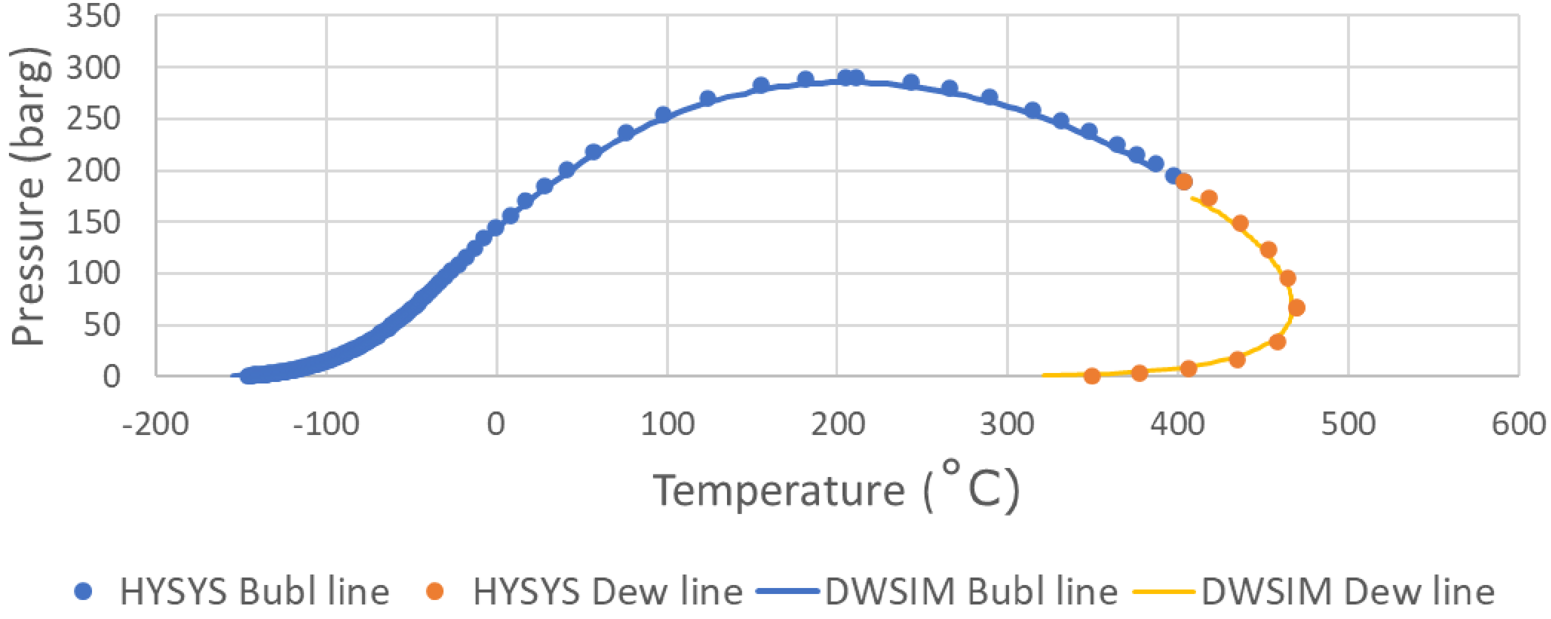



```
class DWSIM:
    def __init__(self, sim_file_path):···

    def __call__(self,x):
        # vars in Pa (abs) and K
        self.sep1t=x[0]+273.15
        self.sep1p= (1.013+x[1])*1e5
        self.sep2p= (1.013+x[2])*1e5
        self.sep3t= x[3]+273.15
        self.sep3p=(1.013+x[4])*1e5
        self.scu1t=x[5]+273.15
        self.scu2t=x[6]+273.15
        self.scu3t=x[7]+273.15
        self.boostp=(1.013+x[8])*1e5
        self.refrig=x[9]+273.15

        self.update_factors()

        err = self.interf.CalculateFlowsheet2(self.sim)
        err = self.interf.CalculateFlowsheet2(self.sim)

        if self.sim.Solved is False:
            err = self.interf.CalculateFlowsheet2(self.sim)
            if self.sim.Solved is False:
                self.load_simulation()
                self.update_factors()
                err = self.interf.CalculateFlowsheet2(self.sim)
                err = self.interf.CalculateFlowsheet2(self.sim)
                if self.sim.Solved is False:
                    self.update_wrong_responses()
            else:
                self.update_responses()
        else:
            self.update_responses()

        print("Errors:",err)
        return np.asarray([self.crude_flow, self.power, self.rvp, self.vap_ratio])

    def load_simulation(self):
        self.sim = self.interf.LoadFlowsheet(self.sim_file_path)

    def update_wrong_responses(self):
        self.rvp = -9999
        self.power = -9999
        self.crude_flow = -9999

    def update_responses(self):···
        #self.crude_flow = self.sim.GetFlowsheetSimulationObject("MSTR-22").GetPhase("OverallLiquid").Properties.volumetric_flow * 3600

    def update_factors(self):···
```

RESULTS: FLUID MODELLING AND PHASE SPLIT

- Comparison between single stage flash at standard conditions
- Good agreement between EOS and flash algorithm implementations
- Phase envelope closely matched
- Conclusion: Equal input gives (almost) same output

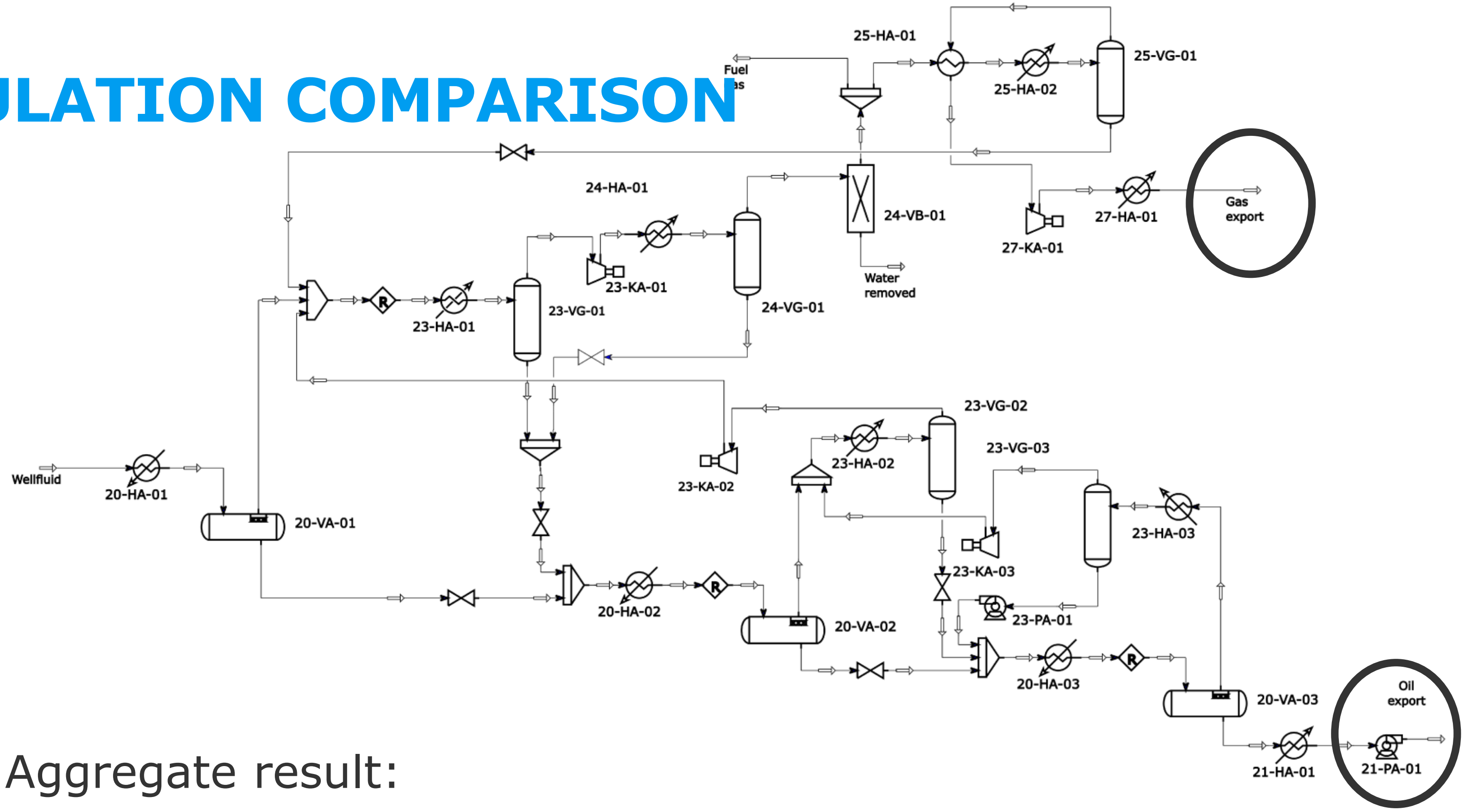
	Unit	HYSYS	DWSIM	Difference (%)
Gas MW	kg/kmol	22.78	22.81	0.114
Gas mole flow	kmol/h	5477.0	5479.8	0.051
Liquid density	kg/m^3	805.4	803.5	−0.244
Liquid MW	kg/kmol	215.3	215.4	0.055
Liquid mole flow	kmol/h	2523.0	2520.2	−0.112
GOR	mol/mol	2.171	2.174	0.163
T_c	°C	402.5	400.8	−0.44
P_c	barg*	191.2	190.4	−0.41



RAMBOLL

RESULTS: FULL SIMULATION COMPARISON

- EOS
- Flash calculations
- Unit operations
  - Valves
  - Heat exchangers
  - Separators
  - Pumps
  - Compressors



Aggregate result:

Table 5 Export stream quality of gas and liquid

	Unit	HYSYS	DWSIM	Difference (%)
Gas export	kmol/h	5102.0	5102.4	0.008
Gas export MW	kg/kmol	20.99	21.02	0.078
Liquid export	kmol/h	2764.3	2763.0	−0.047
Liquid export MW	kg/kmol	201.9	201.9	0.007
Liquid export RVP	psia*	10.1	10.1	0.056

RAMBØLL

RESULTS: FULL SIMULATION COMPARISON - DETAILED

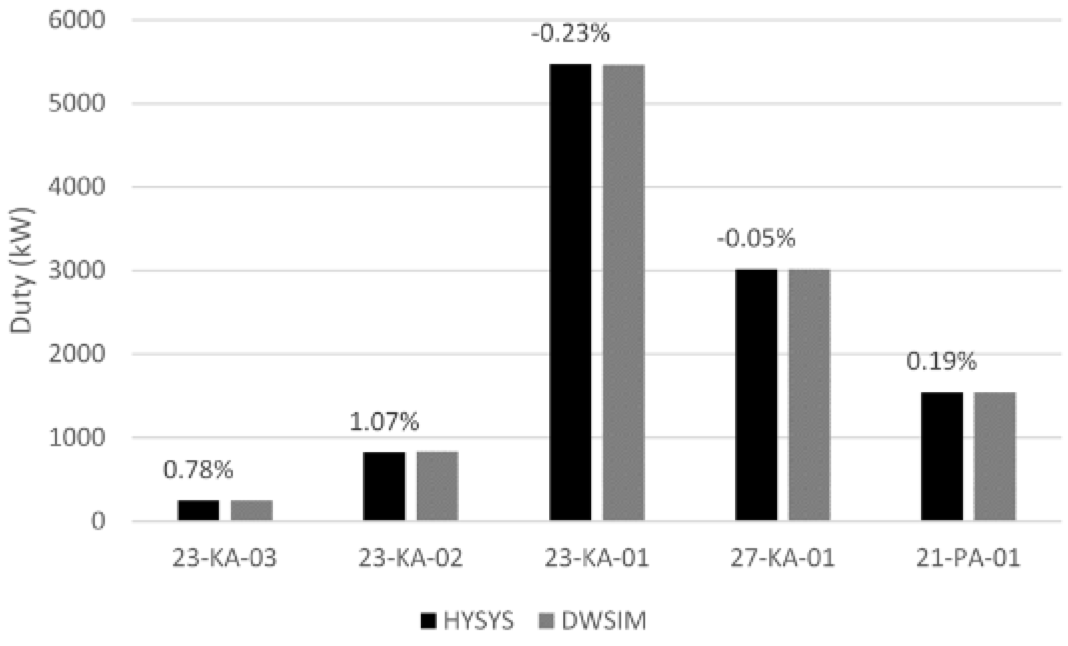


Fig. 3 Main mechanical driver duties calculated with HYSYS and DWSIM. Numbers above the bars are the relative difference

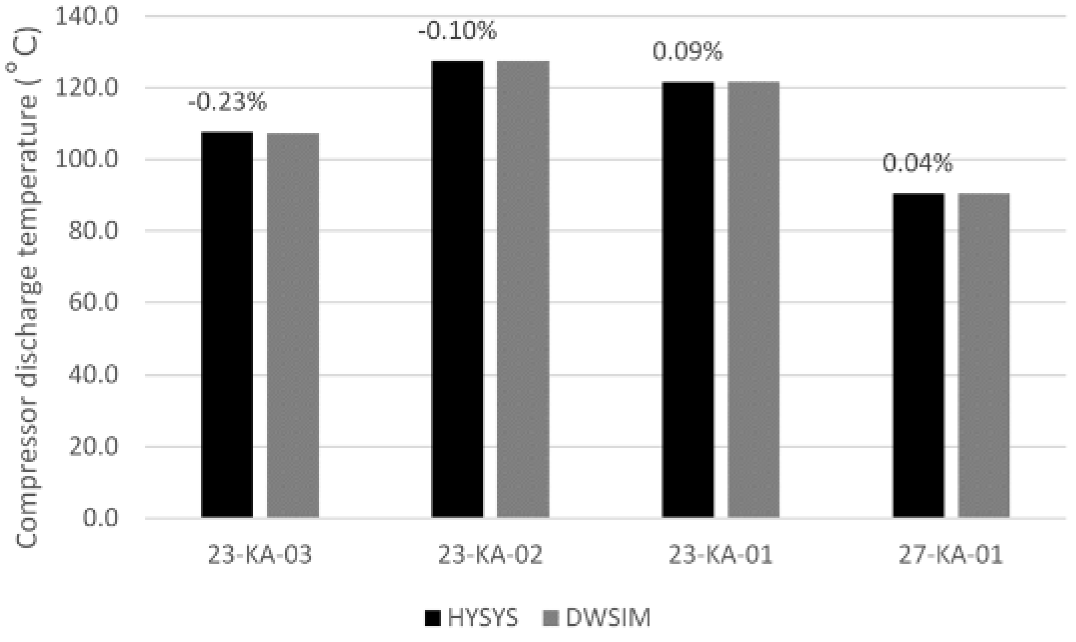
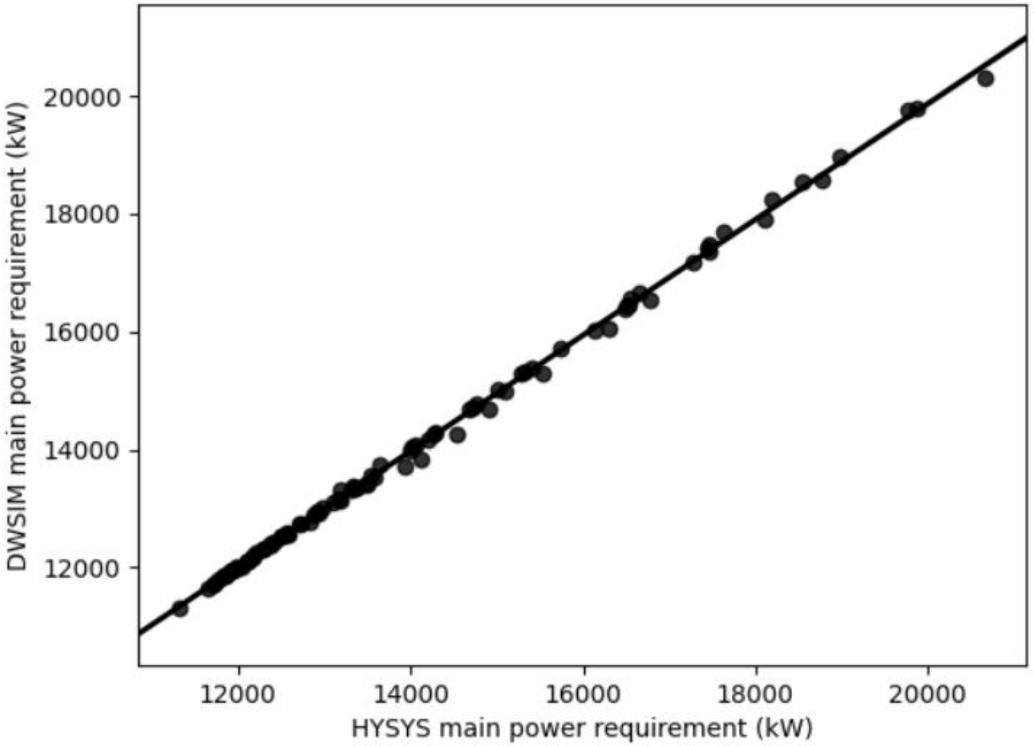
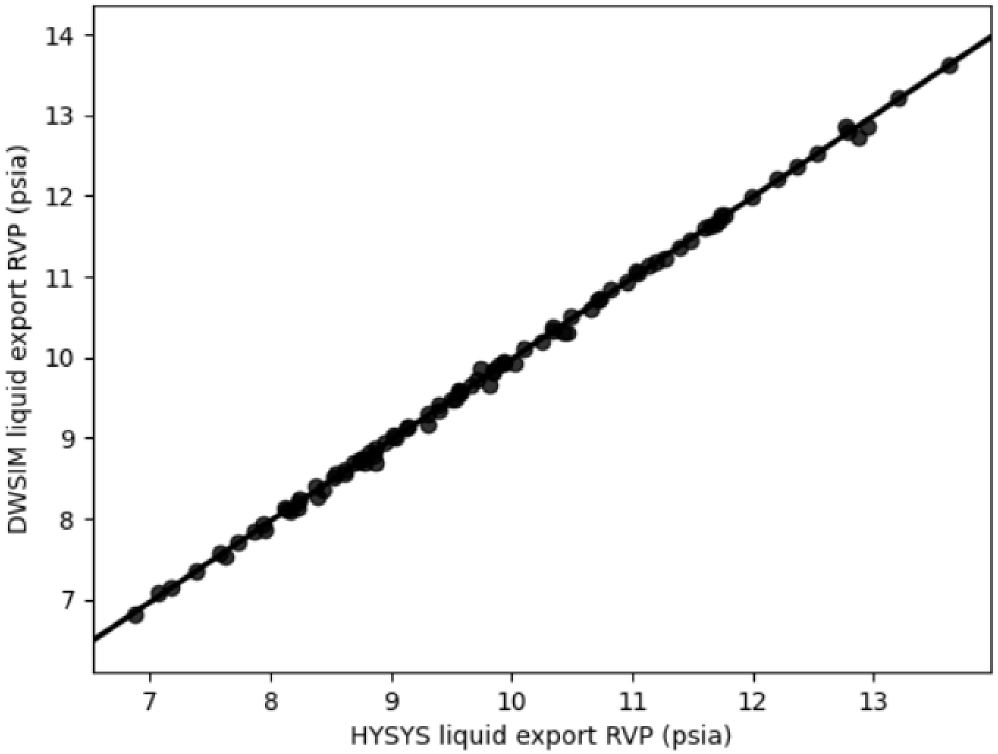


Fig. 4 Comparison of calculated compressor discharge temperatures

RESULTS: MULTIPARAMETER VARIATION STUDY



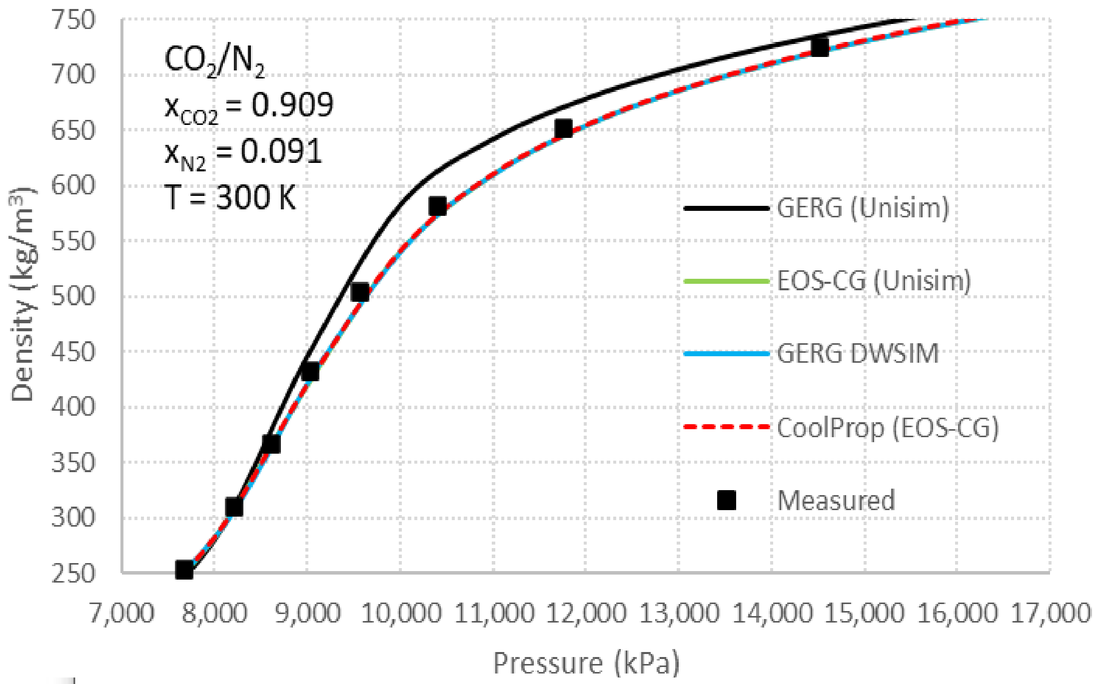


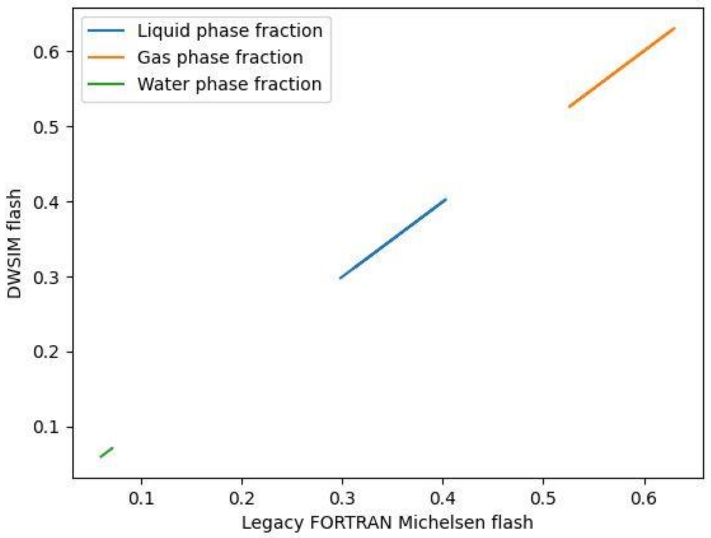
SUMMARY

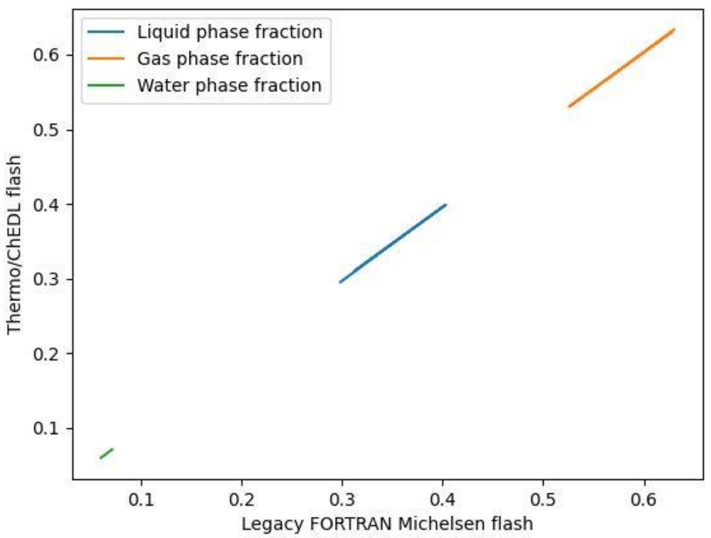
- DWSIM used to model a “complex” oil and gas separation plant
- Results obtained match very closely results from commercial simulator
- Fidelity in DWSIM in an professional and industrial environment is increased
- DWSIM is fairly easy to extend with user models
- Using external python interface opens many interesting applications such as optimisation

POST SCRIPTUM

- DWSIM has advanced EOS included e.g. for CCS applications: Expensive proprietary code is not always better!
- Extensive flash calculation validation made both with python/Thermo (Caleb Bell) and in-house legacy Michelsen flash







RAMBØLL