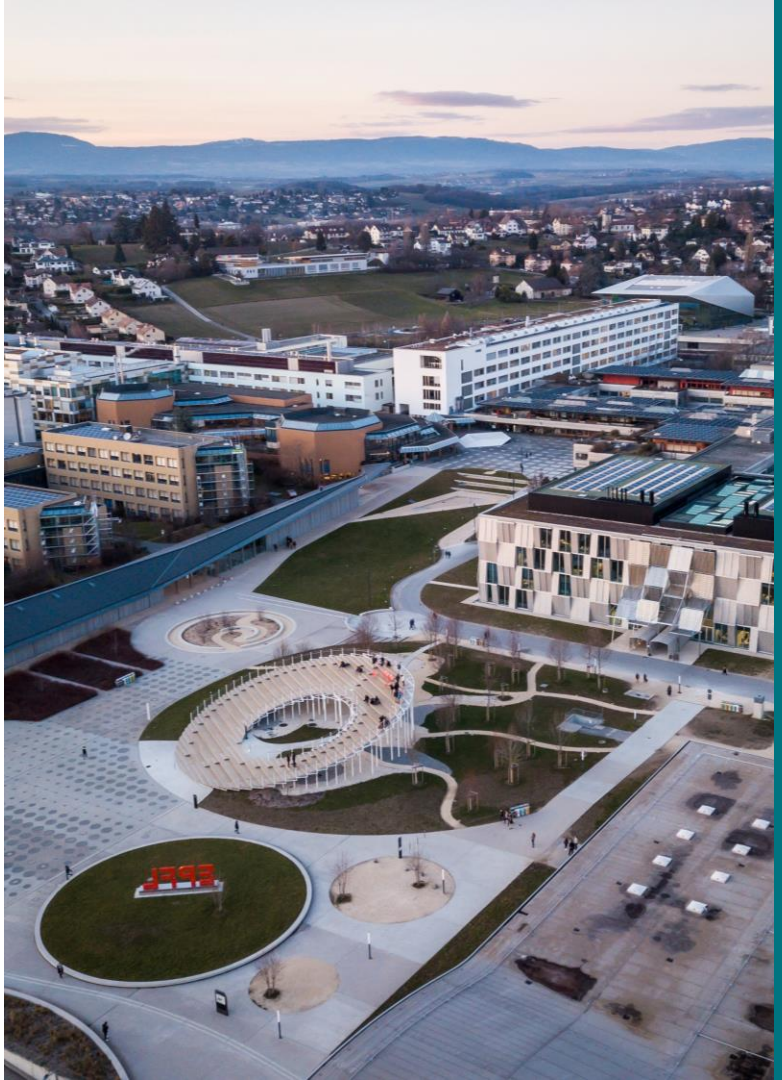
ROSMOSE: A Web-based Tool for Pinch Analysis and Utility Integration

Process development – CHE-459

Soline Corre, Meire Ellen Domingos

Spring 2025

École polytechnique fédérale de Lausanne



OUTLINE

Concept

Functionalities

Case study

Conclusions

ROSMOSE
General information

- ROSMOSE:
  - Newly developed web-based tool using Pinch Analysis methods
  - User friendly interface to simplify utilization of OSMOSE framework
- OSMOSE:
  - Used for over two decades in research projects and teaching activities
  - Metaheuristic and deterministic approaches for combined heat and mass integration

Reference: Open-source scientific publishing system https://quarto.org/ , https://rmarkdown.rstudio.com/

ROSMOSE
General information

- Accessible through dynamic development environments

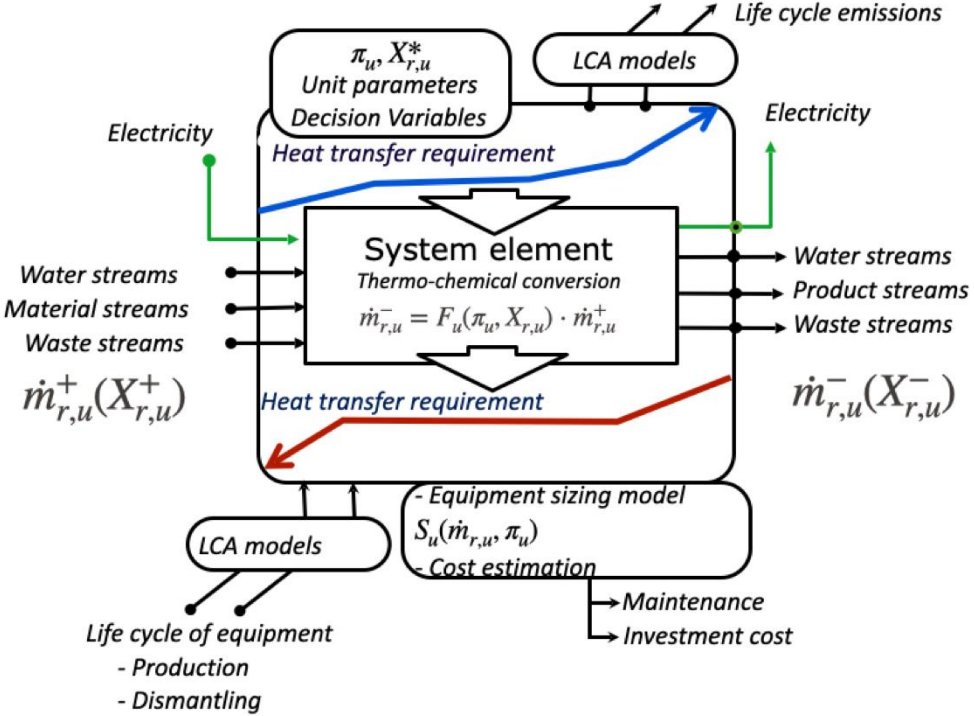
- Flexible in integrating various software and tools

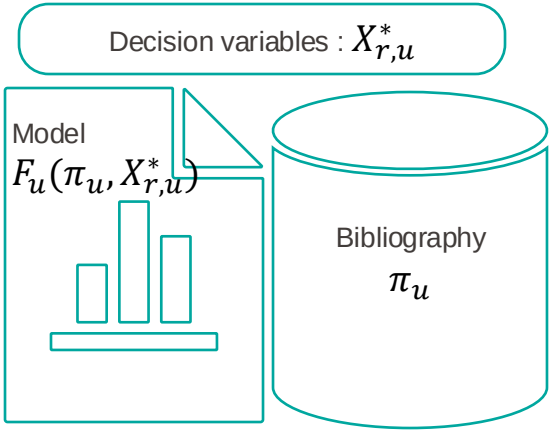
- Many possible output formats for standard and interactive reports

Make Science Reproduceable!

Reference: R Mardown from R Studio, https://rmarkdown.rstudio.com/

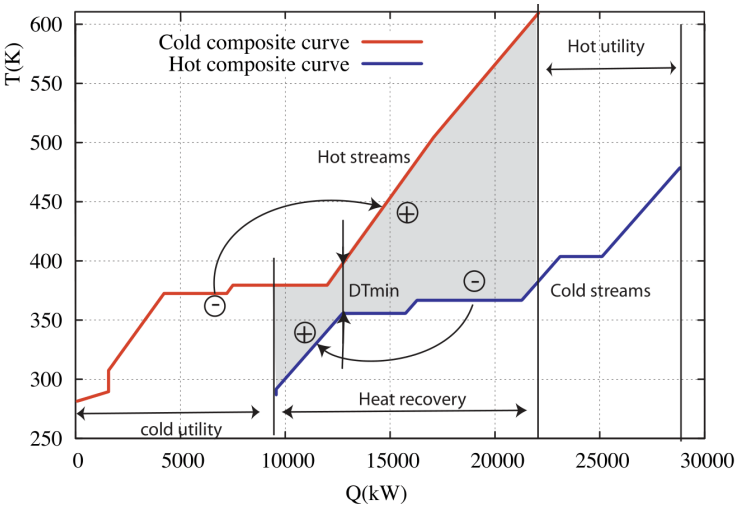
ROSMOSE
Process integration/technology blueprints





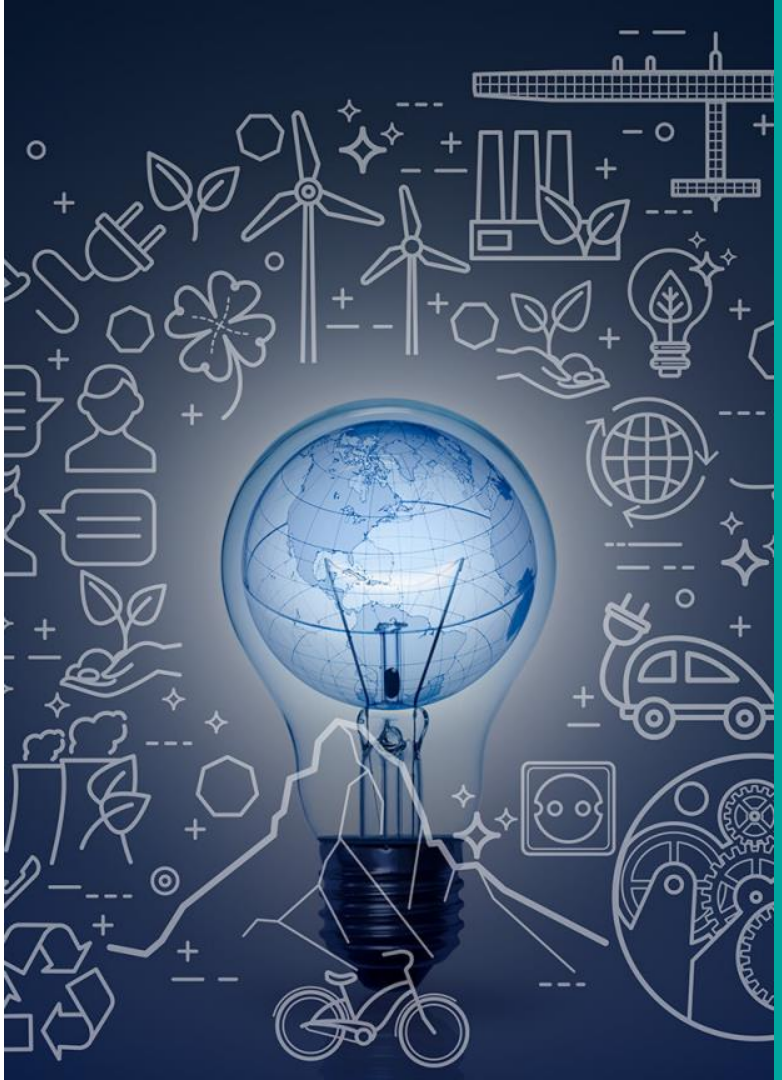
ROSMOSE
General information

- Allows users to:
  - Define systems and energy technologies
  - Call external software (e. g. Aspen Plus)
  - Perform pinch analysis calculations
  - Solve optimisation problems
  - Integrate utilities
  - Compare operational scenarios
  - Produce dynamic reports or presentations



Hot and cold composite curves of a process

Reference: Maréchal, François. (2010). Advanced Energetics Course Notes

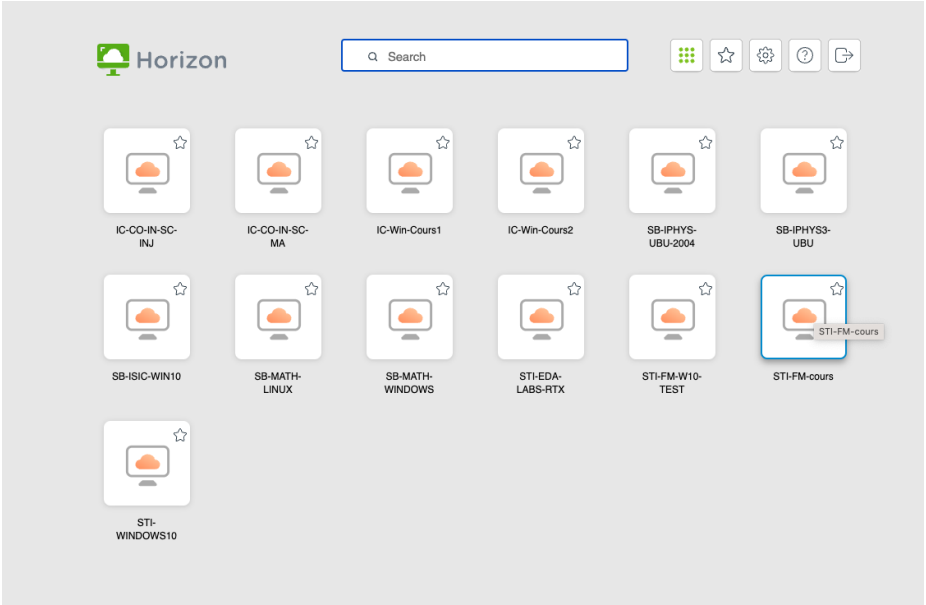


OUTLINE

Overview of the Quarto book and the environment

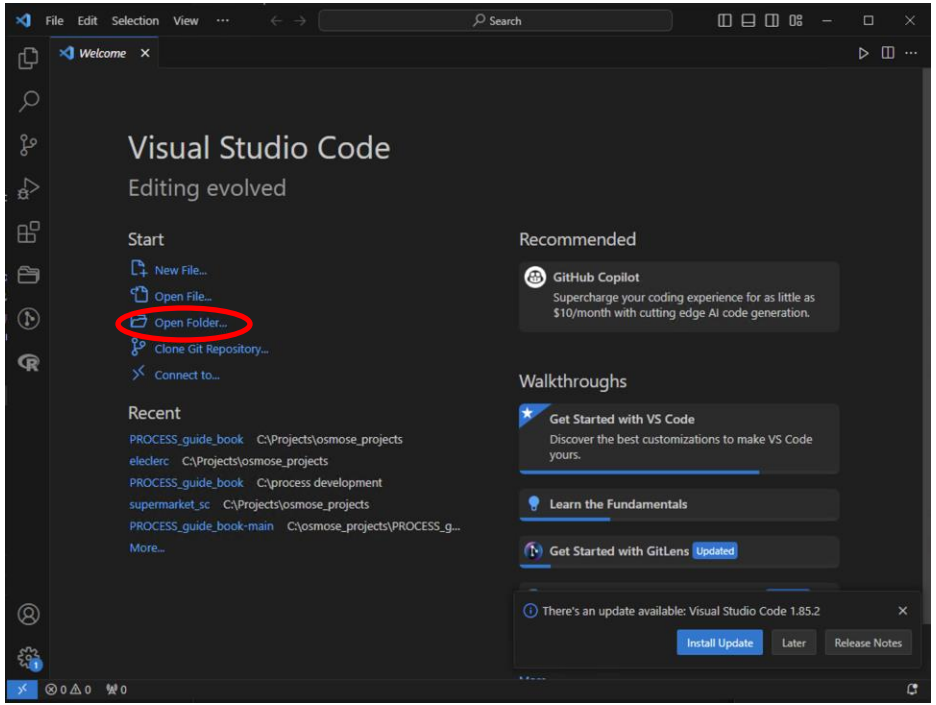
VM Access & File Download

- Access your VM through https://vdi.epfl.ch --> STI-FM-course
- Download the heat integration project files from moodle → "PROCESS_GUIDE_BOOK.zip"
- Move the files to the VM desktop and extract



Open the Project Files

- Open Visual Studio Code
- Open the file using “open folder”

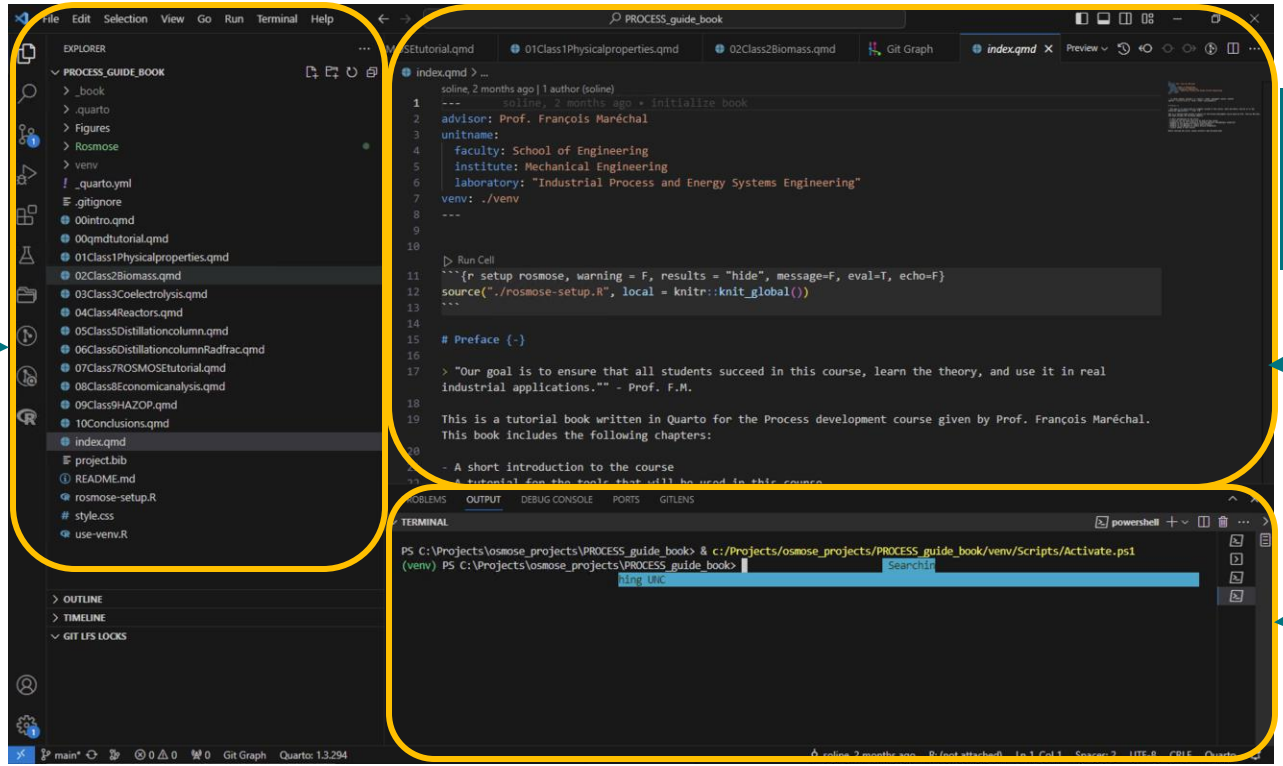


- Then “select folder”

The quarto book is now open in VSCode !

Visual Studio Code environnement

This is where are the files of you model, Aspen files, ROsmose files and results



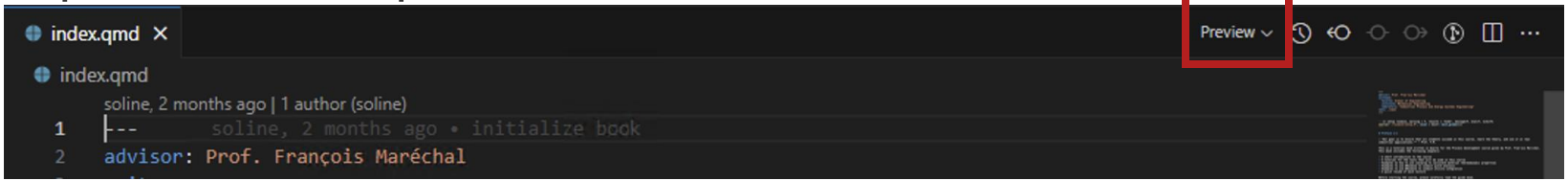
This is the environnement, where you can code

This is the terminal, where you can execute your code

Building the quarto book

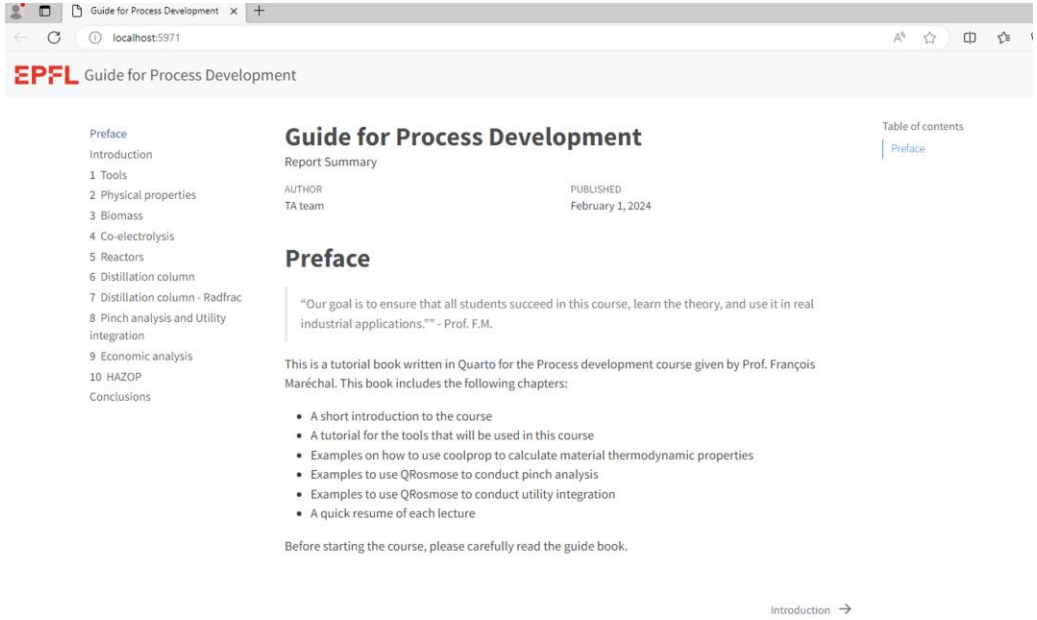
- Open the index.qmd file

Click on “preview”

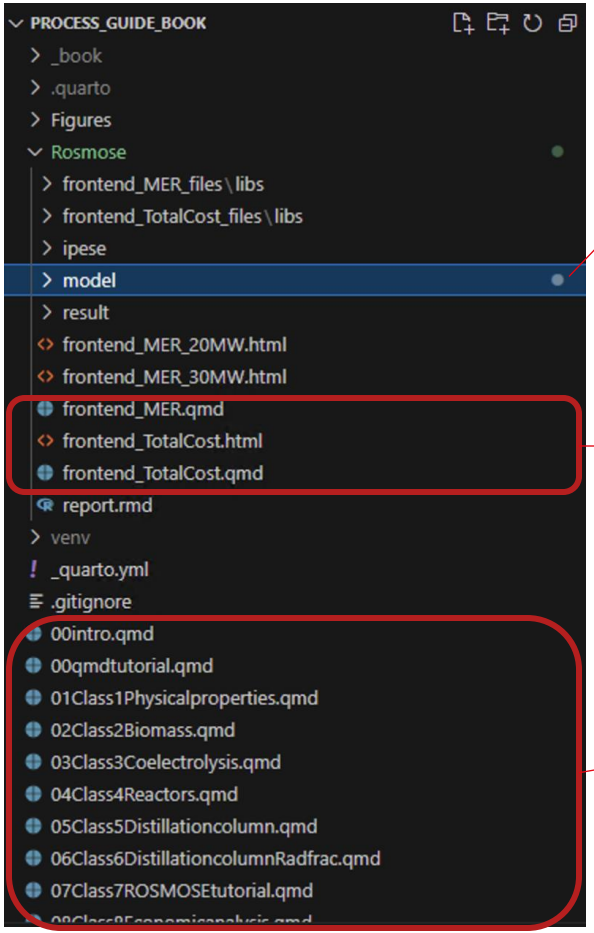


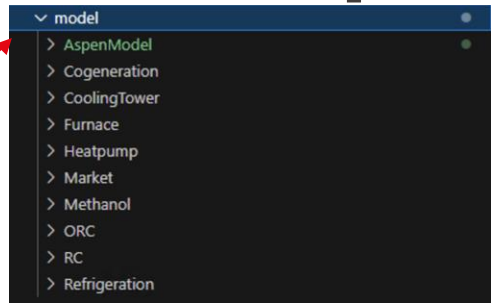
Or write *quarto preview* in the terminal

It creates an html book



Overview of the quarto book - composition





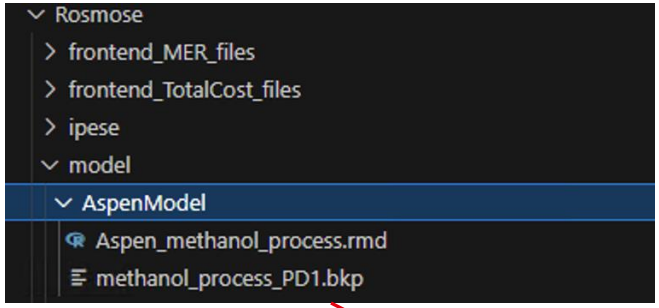
The folder *Model* contains ROsmose and Aspen files

The Frontend: *MER* (minimum energy requirement) and *Total Cost*, they allow you to run the model

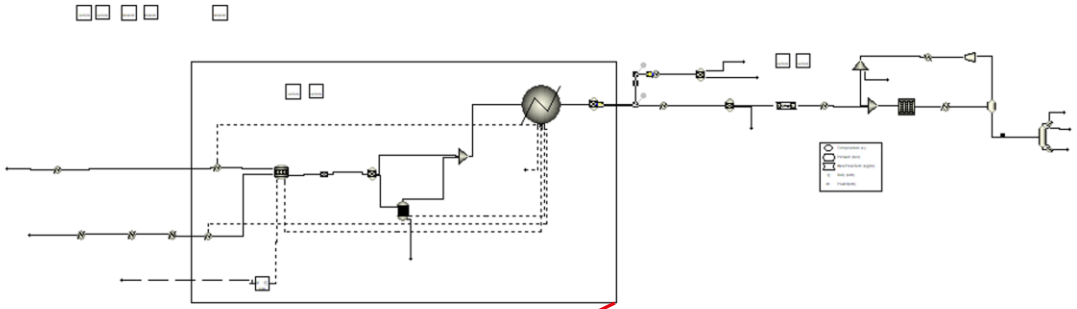
This is the description of the classes. It can be displayed using *Preview* in the index.qmd file

ASPEN Model – methanol production

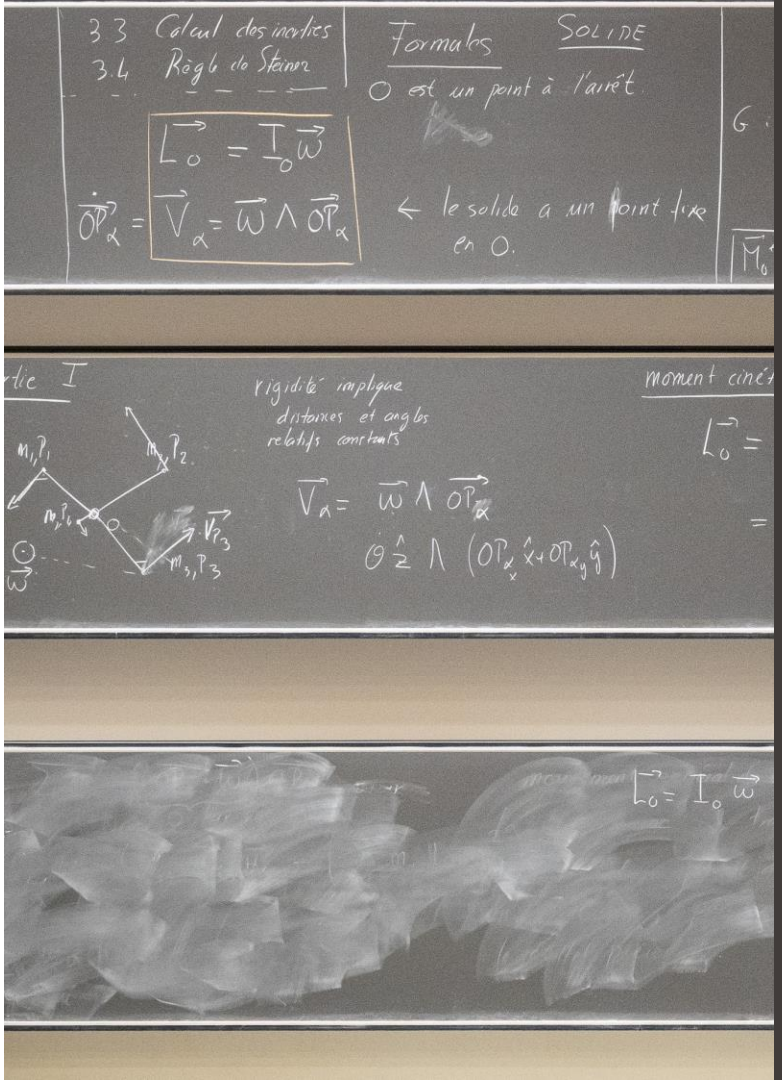
The ASPEN file is located in the guide book, directly linked to ROsmose



It is the same ASPEN file as last class



With Rosmose, we will be doing the heat integration



OUTLINE

Concept

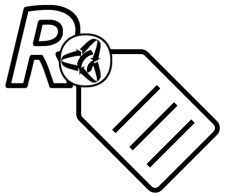
Functionalities

Case study

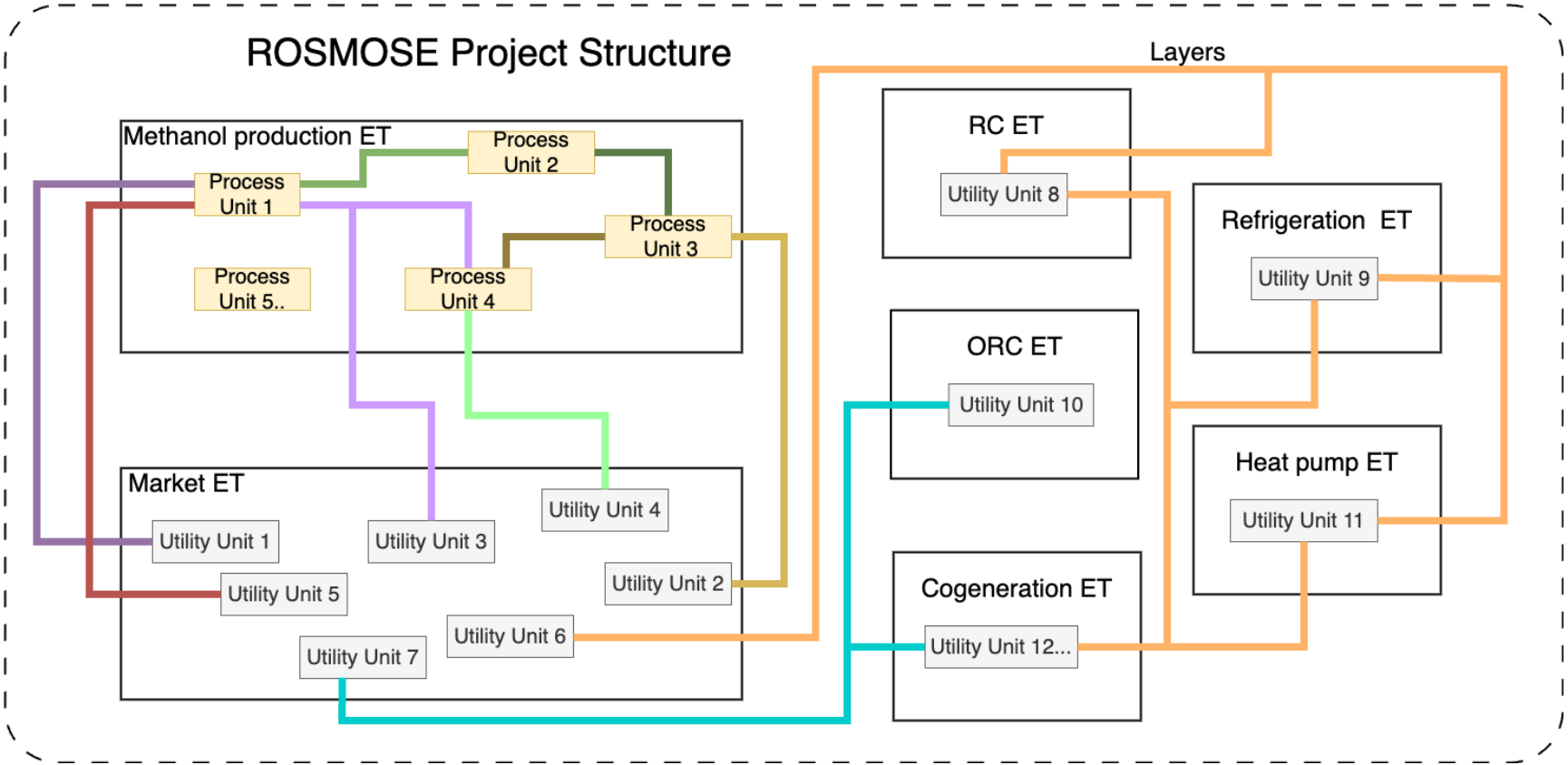
Conclusions

ROSMOSE Functionalities

- ROSMOSE variable is defined as a *tag*, i.e. an object containing:
  - a name,
  - a value,
  - a physical unit, and
  - a description



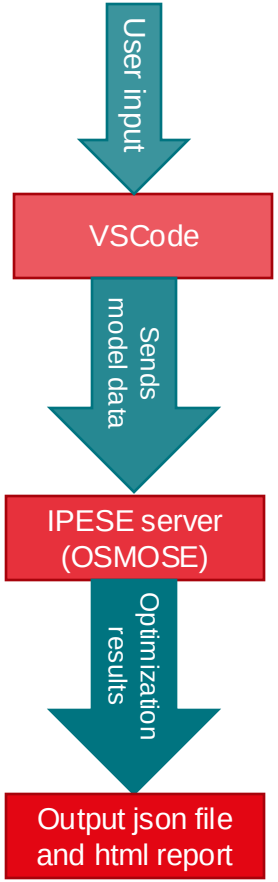
- Tags are used to specify: inputs, outputs, or parameters of a model
- Building blocks of an ROSMOSE project are called *energy technology (ET) models*
- Layers in ROSMOSE can be thought of as the ‘pipes’ through which similar streams flow and are described by a name and a physical unit


ROSMOSE Project Structure
Layers
Methanol production ET
Process Unit 1
Process Unit 2
Process Unit 3
Process Unit 4
Process Unit 5..
Market ET
Utility Unit 1
Utility Unit 3
Utility Unit 4
Utility Unit 5
Utility Unit 2
Utility Unit 6
Utility Unit 7
RC ET
Utility Unit 8
Refrigeration ET
Utility Unit 9
ORC ET
Utility Unit 10
Heat pump ET
Utility Unit 11
Cogeneration ET
Utility Unit 12...

Using Rosmose

- You already have a running example in the folder “Rosmose”

1. User defines the model to optimize
2. User asks the IPESE server to solve the optimization problem
3. IPESE tools process the models and solve the problem
4. If the calculation succeeds, the results are sent in a .JSON file
5. An automatic .html report is also generated summarizing model results and sample plots



ROSMOSE MILP solver

- OSMOSE Lua backend framework handles solving the mixed integer linear programming (MILP) problem

$$min\ Cost_{tot} = \sum_{u \in Units} \left[\sum_{t \in Time} \left(Cop1_u \cdot Y_{u,t} + Cop2_u \cdot M_{u,t} \right) \cdot \Delta t + Cinv1_u \cdot Y_u + Cinv2_u \cdot M_u \right]$$

- Subject to multiple constraints:

$$Y_{u,t} \cdot Fmin_u \leq M_{u,t} \leq Y_{u,t} \cdot Fmax_u \quad ; \forall u \in Units, \quad \forall t \in Time$$

$$\sum Input\ streams = \sum Output\ streams \quad ; \forall\ Layer, \quad \forall\ Time$$

$$\sum_{w=1}^{N_w} M_w \cdot q_{w,r} + \sum_{i=1}^{N} Q_{i,r} + R_{r+1} - R_r = 0 \quad ; \forall\ r = 1 \ldots . N$$

Reference: Yoo M-J, Lessard L, Kermani M, et al. OsmoseLua – An Integrated Approach to Energy Systems Integration with LCIA and GIS. In: *Computer Aided Chemical Engineering*. Elsevier, pp. 587–592

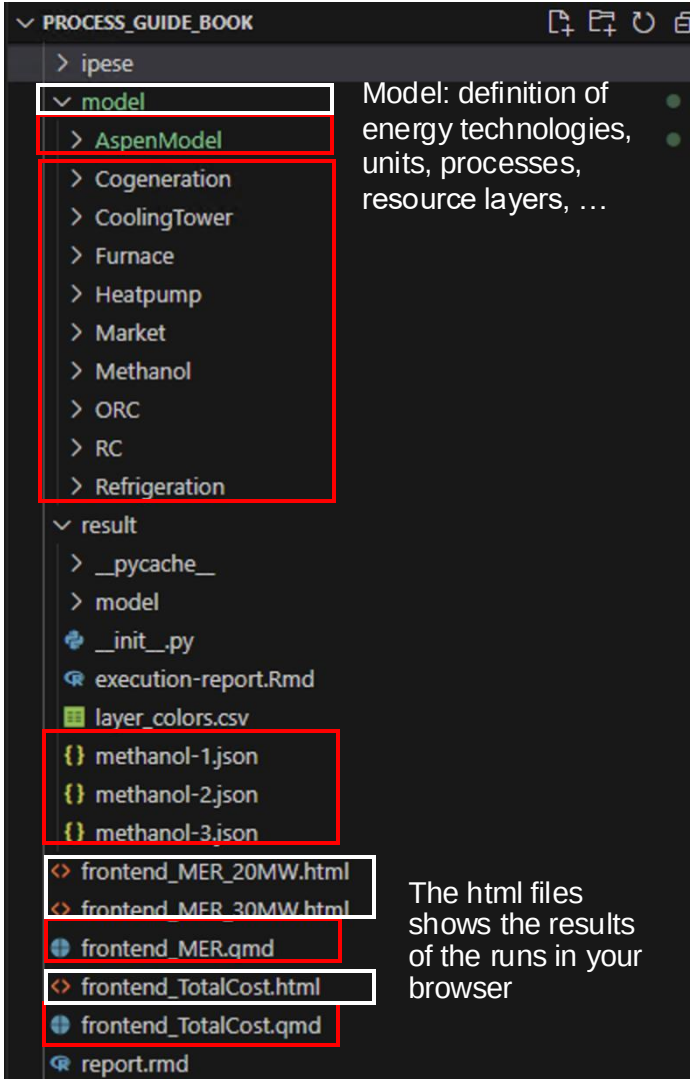
Rosmose folder contents

The Aspen model

The ETs

The results of the runs, stored in a json file

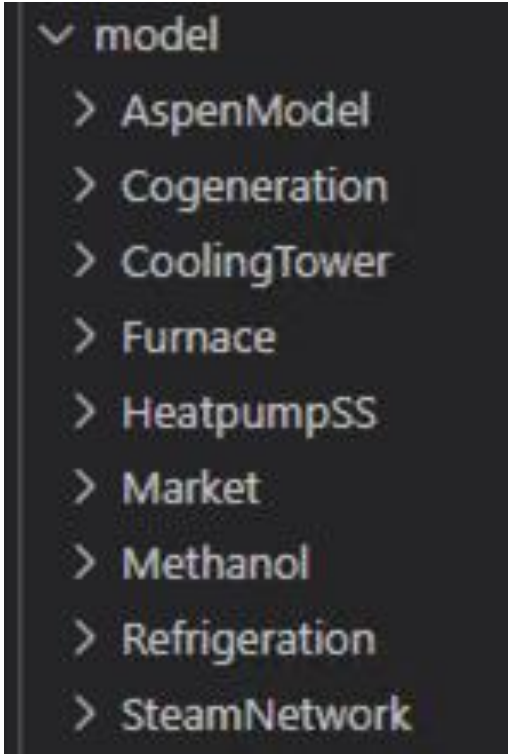
The frontend is the main script that dictates what to include in the optimization and run it. There are two frontends: MER (minimum energy requirement) and total cost (cost optimization)



Using Rosmose – definition of the model

- In the model folder, you define the model for your Energy Technologies (ET). In the example case, we have 9 ETs + the link to the Aspen model:
  - A sample Aspen Model demonstrating how to import flowsheet data from Aspen into ROSMOSE
  - The heat integration ETs *Cogeneration, Cooling Tower*, *Furnace, HeatpumpSS, Methanol, Refrigeration, and Steam Network*
  - The *market,* to close the mass and energy balance with the resources consumed

- For the first step, you will only model your processes to determine the minimum energy requirements and estimated savings → no heat integration ETs needed



Linking Aspen with Rosmose

Rosmose – ASPEN link

- In the folder *AspenModel,* open the file *Aspen_methanol_process.rmd*
- In this file, we can import and export data from our Aspen flowsheet *(methanol_process_PD1.bkp)*
- You can specify conditions that you want to change in the simulation file
- You can import stream/unit data from the aspen flowsheet to be used in the heat integration problem

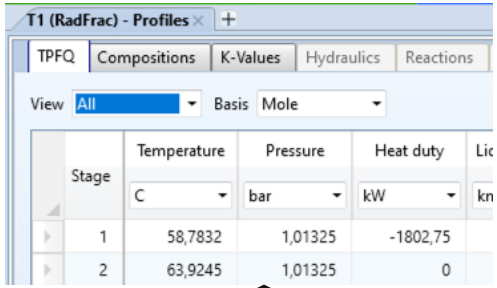
Name and location of the Aspen file

```
```{rosmose}
: MODEL Methanol

|Software|Location| Comment|
|:---|:---|:----|
|ASPEN|model\AspenModel\methanol_process_PD1.bkp||
```
```

Input power of the process

```
```{rosmose}
: MODEL INPUTS Methanol

| Name | Path |Value | Units| Comments |
|:-----|:-----|--------:|:---------|:---------|
|Electrolyzer_size|/Data/Streams/POWER/Input/POWER|-30000|kW||
```
```

Aspen paths of the streams
Follows the syntax Name/path/units

```
```{rosmose}
: MODEL OUTPUTS Methanol

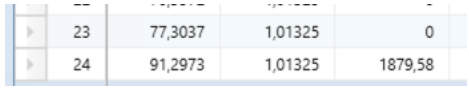
| Name | Path | Units| Comments |
|:-----|:-----|--------:|:---------|
|T_AIR_1|/Data/Streams/AIR1/Output/TEMP_OUT/MIXED|C||
|T_AIR_2|/Data/Streams/AIR2/Output/TEMP_OUT/MIXED|C||
|heater_1|/Data/Blocks/HEATER1/Output/QCALC|kW||
|T_FUEL_1|/Data/Streams/FUEL1/Output/TEMP_OUT/MIXED|C||
|T_FUEL_2|/Data/Streams/FUEL2/Output/TEMP_OUT/MIXED|C||
|heater_3|/Data/Blocks/HEATER3/Output/QCALC|kW||
|T_FUEL_3|/Data/Streams/FUEL3/Output/TEMP_OUT/MIXED|C||
|heater_4|/Data/Blocks/HEATER4/Output/QCALC|kW||
|T_FUEL_4|/Data/Streams/FUEL4/Output/TEMP_OUT/MIXED|C||
|heater_5|/Data/Blocks/HEATER5/Output/QCALC|kW||
|T_AIR_OUT|/Data/Streams/AIROUT/Output/TEMP_OUT/MIXED|C||
|T_AIR_4|/Data/Streams/AIR4/Output/TEMP_OUT/MIXED|C||
|cooler_1|/Data/Blocks/COOLER1/Output/QCALC|kW||
|T_FUEL_OUT|/Data/Streams/FUELOUT/Output/TEMP_OUT/MIXED|C||
|T_LTFUEL|/Data/Streams/LTFUEL/Output/TEMP_OUT/MIXED|C||
```
```

For the distillation columns

- Take the two first steps of the condenser
- And the two last steps of the reboiler

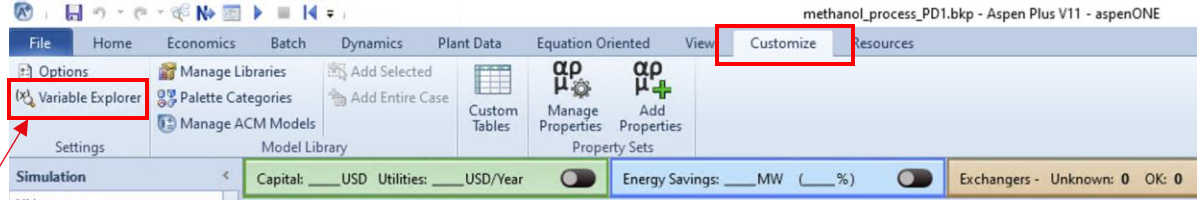


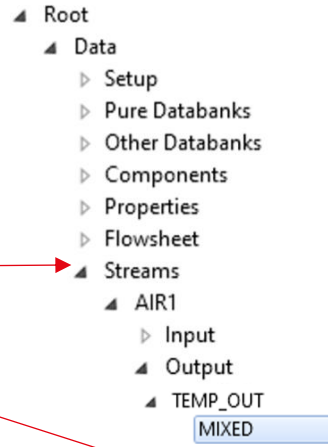

```
|cond_ti|/Data/Blocks/T1/Output/B_TEMP/2|C||
|cond_to|/Data/Blocks/T1/Output/B_TEMP/1|C||
|cond_Q|/Data/Blocks/T1/Output/COND_DUTY|kW||
|reb_ti|/Data/Blocks/T1/Output/B_TEMP/23|C||
|reb_to|/Data/Blocks/T1/Output/B_TEMP/24|C||
|reb_Q|/Data/Blocks/T1/Output/REB_DUTY|kW||
```



Recovering the streams from Aspen

- Open the file *Aspen_methanol_process.rmd* in Aspen Plus
- Go to *Simulation*
- The streams are in *Customize* → *Variable explorer*
- Scroll down the menu until the info you need
- Take the stream path in *Call* (example here for T_AIR_1). When you copy it in Rosmose, **be careful of the slash /** that are backed in Rosmose (\)

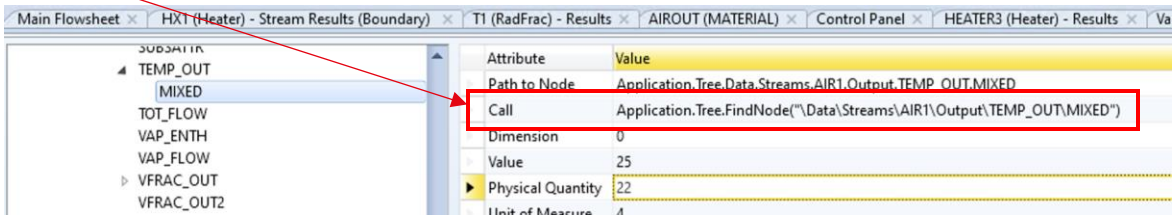





```{rosmose}
: MODEL OUTPUTS Methanol
```

Aspen paths of the stream

Name	Path	Units	Comments
T_AIR_1	/Data/Streams/AIR1/Output/TEMP_OUT/MIXED	C	



Description of the ETs

Definition of the Ets

Name the model

Defining your layers

Defining process units

````
# Biomass ET {-}

```{rosmose Methanol}
! OSMOSE ET Methanol
```
This ET will use the following Layers

```{rosmose Methanol_layers}
: OSMOSE LAYERS Methanol

|Layer      |Display name   |shortname  |Unit       |Color  |
|:----------|:--------------|:----------|:----------|:------|
|ELEC       |Electricity    |elec       |kW         |yellow |

```

The methanol ET contains the following units

```{rosmose Methanol_units}
: OSMOSE UNIT Methanol

|unit name  |type   |
|:----------|:------|
|Methanol   |Process|
```
````

Definition of the ETs

- We have to define some parameters for our unit methanol

```
```{rosmose Methanol_params}
: OSMOSE UNIT_PARAM Methanol

|cost1|cost2|cinv1|cinv2|imp1|imp2|fmin|fmax|
|:----|:----|:----|:----|:---|:---|:---|:---|
|0    |0    |0    |0    |0   |0   |1   |1   |
```
```

- And the hot and cold streams in our processes

Heat Streams

```
```{rosmose Methanol_hs}
: OSMOSE HEAT_STREAMS Methanol

|name         |Tin          |Tout         |Hin  |Hout         |DT min/2|alpha|
|:------------|:------------|:------------|:----|:------------|:-------|:----|
|heater_1     |%T_AIR_1%    |%T_AIR_2%    |0    |%heater_1%   |2.5     |1    |
|heater_3     |%T_FUEL_1%   |%T_FUEL_2%   |0    |%heater_3%   |2.5     |1    |
|heater_4     |%T_FUEL_2%   |%T_FUEL_3%   |0    |%heater_4%   |2.5     |1    |
|heater_5     |%T_FUEL_3%   |%T_FUEL_4%   |0    |%heater_5%   |2.5     |1    |
|cooler_1     |%T_AIR_OUT%  |%T_AIR_4%    |0    |%cooler_1%   |2.5     |1    |
|cooler_2     |%T_FUEL_OUT%|%T_LTFUEL%   |0    |%cooler_2%   |2.5     |1    |
|MSC_cooler_1 |%MSC_cooler_1_Tin% |%MSC_cooler_1_Tout% |0    |%MSC_cooler_1_Duty%  |2.5     |1    |
|MSC_cooler_2 |%MSC_cooler_2_Tin% |%MSC_cooler_2_Tout% |0    |%MSC_cooler_2_Duty%  |2.5     |1    |
|HX_1         |%T_S1%       |%T_S2%       |0    |%HX1_Duty%   |2.5     |1    |
|HX_2         |%T_S4%       |%T_S5%       |0    |%HX2_Duty%   |2.5     |1    |
|HX_3         |%T_S7%       |%T_S8%       |0    |%HX3_Duty%   |2.5     |1    |
|R_1          |%T_S3%       |%T_S4%       |0    |%R1%         |2.5     |1    |
|reb          |%reb_ti%     |%reb_to%     |0    |%reb_Q%      |2.5     |1    |
|cond         |%cond_ti%    |%cond_to%    |0    |%cond_Q%     |2.5     |1    |
```
```

- If your unit is consuming a resource, you can define it as follow (example for the power consumption)

```
```{rosmose}
Power_consump = %MSC_power_tot%-%Electrolyzer_size%
```


```{rosmose Methanol_rs}
: OSMOSE RESOURCE_STREAMS Methanol

|layer      |direction|value                |
|:----------|:--------|:--------------------|
|ELEC       |in       |%Power_consump%      |
```
```

Heat streams

- **Definition of input heat streams**
  - **Tin:** stream inlet temperature [℃]
  - **Tout:** stream outlet temperature [℃]
  - **Hin:** stream inlet enthalpy (0 reference point) [kW]
  - **Hout:** stream outlet enthalpy (Q for hot and cold streams) [kW]
  - **DTmin/2:** Dtmin is the minimum temperature difference allowed for the heat exchange
  - **alpha:** stream heat transfer coefficient [kW/m^2/K]
  - Look at the lecture notes for a guidelines on Dtmin/2 and alpha for different types of streams!
  - Pay attention to only put letters and numbers in the name of the streams. **No special characters (_ - ; …)**

Heat Streams

```{rosmose Methanol_hs}
: OSMOSE HEAT_STREAMS Methanol

|name      |Tin          |Tout          |Hin  |Hout         |DT min/2|alpha|
|:---------|:------------|:-------------|:----|:------------|:-------|:----|
|heater_1  |%T_AIR_1%    |%T_AIR_2%     |0    |%heater_1%   |2.5     |1    |
|heater_3  |%T_FUEL_1%   |%T_FUEL_2%    |0    |%heater_3%   |2.5     |1    |
|heater_4  |%T_FUEL_2%   |%T_FUEL_3%    |0    |%heater_4%   |2.5     |1    |
|heater_5  |%T_FUEL_3%   |%T_FUEL_4%    |0    |%heater_5%   |2.5     |1    |
|cooler_1  |%T_AIR_OUT%  |%T_AIR_4%     |0    |%cooler_1%   |2.5     |1    |
|cooler_2  |%T_FUEL_OUT% |%T_LTFUEL%    |0    |%cooler_2%   |2.5     |1    |
|MSC_cooler_1 |%MSC_cooler_1_Tin% |%MSC_cooler_1_Tout% |0    |%MSC_cooler_1_Duty%  |2.5     |1    |
|MSC_cooler_2 |%MSC_cooler_2_Tin% |%MSC_cooler_2_Tout% |0    |%MSC_cooler_2_Duty%  |2.5     |1    |
|HX_1      |%T_S1%       |%T_S2%        |0    |%HX1_Duty%   |2.5     |1    |
|HX_2      |%T_S4%       |%T_S5%        |0    |%HX2_Duty%   |2.5     |1    |
```

The inputs

You can also define **tags** and use them in your model definitions. Example here for Heatpump1. A tag has a name, a value, a unit and a definition

````
```{rosmose}
Evap_Tin = 54.5 [C] # Evaporator temperature inlet
Evap_Tout = 54.5 [C] # Evaporator temperature outlet
Cond_Tin = 76.5 [C] # Condenser temperature inlet
Cond_Tout = 76.5 [C] # Condenser temperature outlet
Cond_Qmax = 6000 [kW] # Condenser reference heat flow rate
exeff = 0.5 [-] # Second law efficiency
dtmin_2ph = 2 [C] # phase-change delta t minimum
n = 40.0 [yr] #lifetime of a heat pump
i = 0.06 [-] #interest rate
CEPCI_2020 = 596.2 [-] # actual CEPCI
CEPCI_2008 = 575.4 [-] # CEPCI 2008
```

Note: Heat pumping will be profitable only if it allows to transfer heat from below to above the pinch, i.e.

```{rosmose}
COPcarnot = (%Cond_Tin% + 273) / (%Cond_Tin% - %Evap_Tin%) [-] # Carnot COP
COP = %exeff% * %COPcarnot% [-] # Actual COP
W_heatpump = %Cond_Qmax% / %COP% [kW] # Heat pump power consumption
Evap_Qmax = %Cond_Qmax% * (%COP% - 1) / %COP% [kW] # Evaporator heat flow rate
Annuity = (%i%*(1+%i%)**%n%)/((1+%i%)**%n%-1) [-] #annualization factor
Cinv2_HP = 300*%Cond_Qmax%*(%CEPCI_2020%/%CEPCI_2008%)*%Annuity% [Euro/y] #300 Euro/kWth at the condenser
```
````

The frontend and the results

The frontend

- The frontend is the main script that dictates what to include in the optimization and run it.
- It the file that will be compiled by quarto and will launch the optimization
  - It is your optimization scenario
  - You can define multiple frontends corresponding to multiple scenarios (here minimum energy requirement and cost optimization)
  - Generates two outputs: an optimization report in html and a json results file
- To run Rosmose from VSCode you need open the frontend and click on *run code*

- This command will automatically run all lines in your frontend.Rmd file:
  - Importing Rosmose
  - Importing the models of your ETs
  - Plotting your optimization results and generating the output html report

Methanol process in ROsmose

- Osmose-solve options
  - MER: Minimum energy requirements (only process targets)
  - TotalCost: optimal integration of utility units

$$min\ Cost_{tot} = \sum_{u \in Units} \left[\sum_{t \in Time} \left(Cop1_u \cdot Y_{u,t} + Cop2_u \cdot M_{u,t} \right) \cdot \Delta t + Cinv1_u \cdot Y_u + Cinv2_u \cdot M_u \right]$$

  - With an operating time of 8760 h/year → 1 year operating plant without maintenance

```
```{rosmose solve-osmose}
! OSMOSE SOLVE-LOCAL methanol MER [Methanol]


|name    |value|
|:------:|:----:|
|op_time|8760 |
```
```

Minimum energy requirement – MER frontend

- After running the frontend, you can see the results in your browser

- It has plotted the grand composite curve, the composite curves and the MER are given.

Frontend MER in ROSMOSE

Process Development Heat Integration

AUTHOR
Student1, Student2, Student3

PUBLISHED
February 1, 2024

```
source("../rosmose-setup.R", local = knitr::knit_global())
```

Introduction

This report presents the results of your heat integration simulations. Each run will generate an output json file stored in the "Rosmose/result" folder and a "frontend.html" file stored in "Rosmose" folder.

The json files will be numbered as 1, 2, 3,,,,,, for each run. However, the "frontend.html" file is overwritten each time hence to keep previous versions you must rename the file before running your simulation again.

once your models and units are ready you can run the rsomose simulation by pressing the "Preview" button in the top right corner or using the command line "quarto preview" in the terminal at the correct directory location.

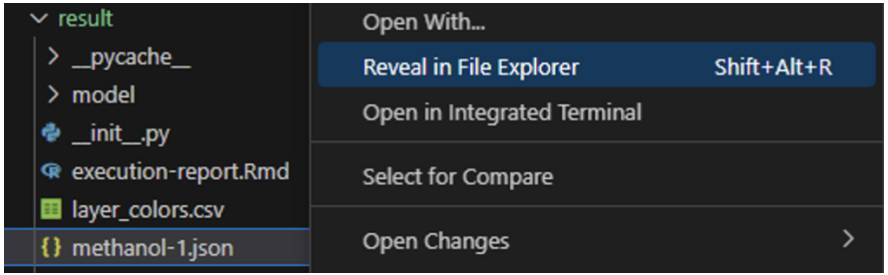
A ready example is given to demonstrate extraction of data from an Aspen flowsheet, changing the process parameters in Aspen through your Rmd file and displaying your integration results.

all Aspen related files are found in this folder "model/AspenModel/."

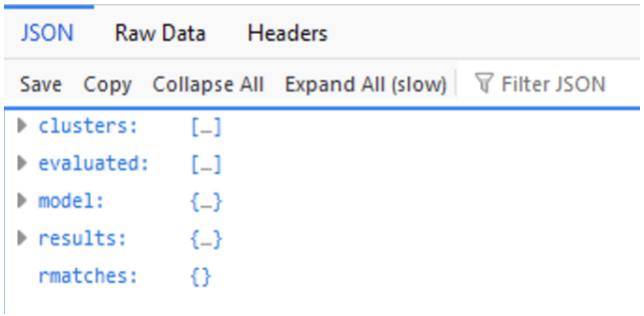
The "Aspen_methanol_process.rmd" file is coupled with the "methanol_process_PD1.bkp" file. The same example Aspen process flowsheet used in all tutorials throughout the course.

Total cost frontend

- The results are in the folder *Results,* in the *methanol.json* file. Click right and open in file explorer

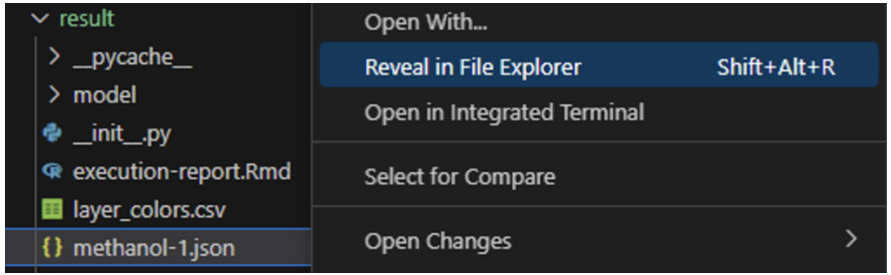


- Open the file with Firefox.

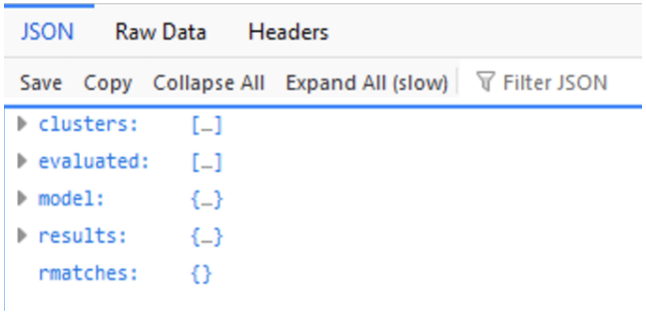


Total cost frontend

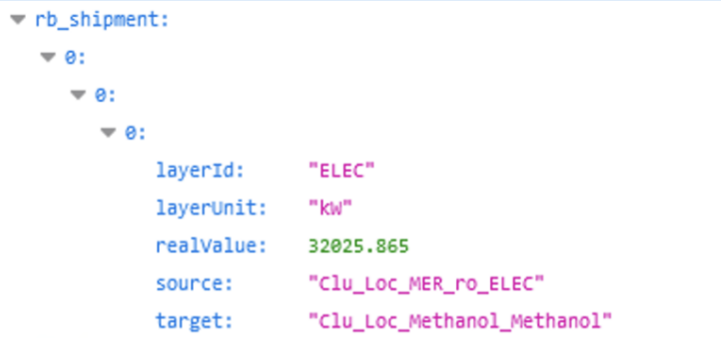
- The results are in the folder *Results*, in the *methanol.json* file. Click right and open in file explorer



- Open the file with Firefox.



- The results are displayed. The main ones you will need are in results > rb_shipment. It is the flows and their value.



- You can also find the KPIs as the capex, the impact, the opex…

OUTLINE

Concept

Functionalities

Case study

Conclusions

Conclusions

- ✓Remember AGIR! (**A**nalyse, **G**enerate, **I**nterpret, **R**eport),
- ✓Find the MER of your process
- ✓Next time we will see how to close the energy balance using various utility systems

For the next class, please look deep into the model and the ETs and prepare your questions about it