Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

A Framework Utilizing a Seamless Integration of Python with AspenPlus® for a Multi-Criteria Process Evaluation

Simon Maier[a,*], Julia Weyand[a], Ginif Kaur[a], Oliver Erdmann[a], and Ralph-Uwe Dietrich[a]

[a] German Aerospace Center, Institute of Engineering Thermodynamics, Pfaffenwaldring 38-40, 70569 Stuttgart, Germany
* Corresponding Author: simon.maier@dlr.de

ABSTRACT

Detailed assessment of fuel production processes at an early stage of a project is crucial to identify potential technical challenges, optimize efficiency and minimize costs and environmental impact. While process simulations often are either very rigid and accurate or very flexible and unprecise, informed decision making can only be maintained by establishing a detailed process model as early as possible within the project lifecycle while keeping relevant aspects of the process flexible enough. In this work, we present the development of a framework based on a dynamic interface between AspenPlus® process simulations and Python, enabling enhanced flexibility and automation for process modeling and optimization. This integration leverages the powerful simulation capabilities of AspenPlus® with the versatility of Python for data analysis and optimization, delivering significant improvements in workflow efficiency and process control. By utilizing the dynamic simulation data exchange with Python, extensive parameter studies can be conducted and post-processed for techno-economic and environmental analyses. Furthermore, the interface allows the implementation of complex kinetic models or optimization routines for single process units. An additional extension for heat integration ensures the technical viability of the process route for reliable comparisons of different routes and process configurations. The functionalities are applied to a biomass- and power-based methanol production process including various process designs and operating conditions. To keep the level of detail at a high level, additional Python scripts are implemented securing a proper scaling of process units such as the methanol synthesis reactor system. The process configurations are assessed technically, economically and environmentally.

Keywords: Modelling and Simulations, Aspen Plus, Technoeconomic Analysis, Life Cycle Analysis.

INTRODUCTION

Aspen Plus is a widely used software tool for process flow sheeting and simulation in chemical engineering. Its robust modeling capabilities enable the design, optimization, and analysis of complex chemical processes across a variety of industries. By providing access to comprehensive thermodynamic models and extensive unit operation libraries, Aspen Plus facilitates efficient process development and performance evaluation.

Despite its strengths, Aspen Plus has limitations that may hinder advanced customization and holistic process analysis. The integration of user-defined scripts, while possible through external tools such as Aspen Custom Modeler or Excel, can be cumbersome and lacks native support, limiting flexibility in certain specialized scenarios. Furthermore, the platform's techno economic analysis (TEA) and life cycle assessment (LCA) features often require external tools or manual data extraction and integration. This lack of transparency and seamless workflow can pose challenges for researchers seeking integrated, transparent assessments that combine process simulation, economic evaluation, and environmental impact analysis.

METHODOLOGY

A simplified illustration of the cost and environmental impact estimation is depicted in Figure 1. Both estimations are based on a detailed process simulation using the commercial AspenPlus® software and Python. Their base functionality is relying on a direct connection with

the simulation via aspenparserDLR [1] as a python-AspenPlus interface. This requires an active AspenPlus® process and hence longer loading times as well as the requirement of AspenPlus® being available on the machine on which the evaluation is conducted. A description of the functionality of pyteea_tea and pyteea_lca can be found elsewhere [2, 3].

Parameter variation framework

To split the data generation from the evaluation of different process concepts, the tool pyteea_sc, of which the 'sc' stands for simulation control, is developed to conduct externally-controlled parameter variations and extract the simulation data required for subsequent analyses after each run. The major communication in between the tools is shown in Figure 1.

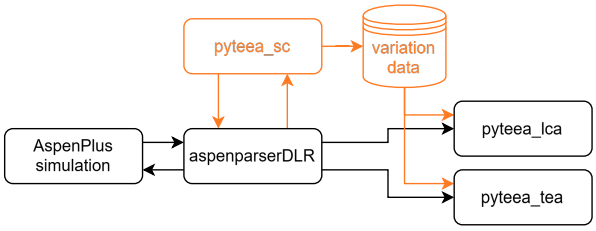

Figure 1. Simplified schematic illustration of the utilized tools and their interaction.

Together with the functionality of varying process parameters externally, the tool allows to include and execute user scripts within the parameter runs. This allows to enhance the level of detail and the complexity of certain process steps or to set up own optimization routines which would require a lot of work-arounds in AspenPlus®.

The main program routine of pyteea_sc can be seen in Figure 2. After a main configuration file is read, the major object `SimControl` is initiated. In dependence on the chosen complexity, multiple further configuration files are read to fill the subobjects for the variation run, such as the `linked_objects` which is either an object which communicates with AspenPlus® or a look-up object of a previous variation run. Furthermore, the `process_vars` are initiated, containing a list of process variables and their variation ranges, their paths within the AspenPlus®, as well as their dependencies within each other. The `additional_features` are a list of user defined functions which are being executed with the variation runs. In the `run_options` settings for the data extraction is defined. nformation about the run configuration, as well as information about the variation matrix and the run status:

```
class SimControl:
    linked_objects: list
    process_vars: list
    additional_features: list
    run_options: RunOptions
    var_structure: VarStructure
    var_status: VarStatus

    def run_simulation(self):
        ...
    def export_data(self):
        ...
    def close_simulation(self):
        ...
```

SimControl also allows to define connections between streams from two or more different AspenPlus® files or look-up objects. This functionality comes with a speed advantage since, e.g., a simulation model of an electrolyser requires much less components than a simulation of a hydrocracking process. Hence, the electrolysis can be calculated in a model with limited components and only its resulting hydrogen connected to the hydrocracker model.

Since the communication with AspenPlus® consumes the most time of the whole procedure, as it can be seen in the digital supplementary material, there might be the need to reduce the number of accesses to it. In case of a benchmark run using 'Case1.2_v10.apw' of the Biomass-Gas-and-Nuclear-To-Liquids Simulation Files published by J. Scott and T. A. Adams [4], a short parameter variation took 7:42 min, of which 6:48 min were communicating with AspenPlus® [5]. Therefore, the framework allows to define specific blocks and streams of which all or only limited data is retrieved. In this way the execution time can be significantly be lowered. Yet, this limits the potential of a later setup of the results as a look-up object. In the standard mode, the mass flows for each component is recorded, as well as its molar flow, volume flow, temperature, pressure, vapour fraction, mass density, path, source, and destination. For the different block types, the list differs a lot, since the data from e.g. a Mixer is only its split ratio, and ports, while for a RadFrac unit, also reflux ratio etc. is read.

To ensure a stable execution of the conducted parameter run, the run status after each execution is tracked and the data is only retrieved when it is actually available while a logger records the run status. Nevertheless, in case of unconverged simulation, the data retrieval is skipped the run number is recorded. The list of unconverged simulations can then be executed in a subsequent variation run in order to complete the desired variation matrix. Often, the convergence issues can be overcome in a second run and are originating from an unfavorable order of varied parameters.

After the connection to all simulations or data objects is set up, potential further configuration files for the additional features are read. Eventually, a variation matrix is generated in dependence on potential primary, secondary, etc. parameter relations. The variation matrix is then exported as well as all relevant run settings to ensure the reusability of the run data.

The modular and object-oriented structure of the tool allows to easily include new functionalities and can flexibly work with one or five process parameters, different recorded process units etc. Furthermore, the tools is set up with a clear deviation between input data and functions and allow no hard-coded input.

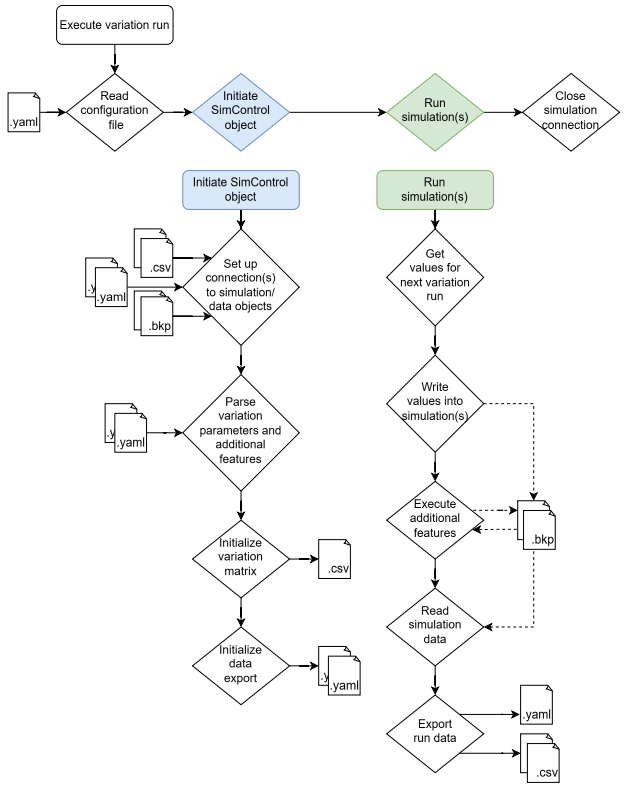

Figure 2. Program routine and its major functions and data in- and output.

For the execution of a parameter variation, the values for each single variation parameter are taken from the variation matrix and written into the simulation(s). In case of using look-up objects, the suitable data is prepared in dependence on the given parameters. Then, any additional features are executed. The additional feature which is included in this work is described in the section below. After all simulations converged the simulation data is exported using the aspenparserDLR. To ensure data retrieval for each complete parameter run as well as for each individual simulation run, md5sums are generated based on the run configuration.

Techno-economic analysis tool

The economic assessment is based on the approach by Peters, Timmerhaus, and West [6] which has been applied in various previous, techno-economic studies [7-9]. A database of reference equipment costs is required to represent any change in the process configurations also in the capital and operational expenditures. Based on the previously described AspenPlus® process model the material and energy flows are used to scale equipment sizes, as well as raw material and by-product streams.

The tool requires a main configuration file which takes plant data input such as lifetime, plant type, operating hours per year, and indirect OPEX cost factors. Together with data coming from various data sources the techno-economic calculations are executed. Currently the tool supports connecting and reading data directly from AspenPlus® simulations, CSV files, YAML files, and JSON files. Once the data is read and parsed from these file input sources, it is further processed in dependence on the reference equipment/raw material/by-products and the costs of the defined product are estimated.

In general, the simulation outputs (material flows, energy flows and equipment dimensions) are employed to calculate capital (CAPEX) and operational expenditures (OPEX) as part of the net production cost (NPC).

Life cycle assessment tool

The same material and energy flows are also utilized as input for the foreground system of the life cycle inventory (LCI). The background system of the LCI is supplied with data from databases like Ecoinvent or ones created from literature or project partner data. With life cycle impact assessment (LCIA) methods, the environmental impacts are then calculated. Details on the python-based life cycle assessment (pyteea_lca) tool can be found elsewhere [3].

The pyteea_lca tool utilizes the open source LCA framework Brightway2 (Python package) [10]. In order to manage and browse through projects and databases easier and in a more intuitive way the Activity Browser (AB) [11], a graphical user interface (GUI) for Brightway2, can be additionally employed.

To perform the LCA, a configuration file must be set up pointing all additional files which are required for the LCA and defining the general information of the LCA evaluation run. Furthermore, the setup requires a series of YAML files that provide the tool with configurations details, such as linked datasets for Brightway2, functional unit, impact categories from LCIA methods and variation run data.

For conducting the LCA of the generation variation run data batch, a look-up object of that data is initiated reading all the configuration information of the previous parameter run and making it available for the ecological evaluation. Hence, all CSV and YAML files are read, including all stream values (heat-, electricity- and mass flows) for each run, and the variation matrix that describes the varied parameters for each run. These files enable the seamless integration of Aspen Plus® results with pyteea_lca for further analysis. Using the ‘get_current_stream_sizes’ function from pyteea_lca, stream values are extracted from the CSV file and linked to their selected aspen id.

The pyteea_lca tool can either iterate over single runs or run through the whole data batch to perform LCA calculations for each run automatically. After calculating the LCIA results, the tool either generates and stores a single or multiple CSV files. The main result file provides environmental impacts per aspen id.

Process configurations

The presented tools within this work are applied to the flowsheet model presented by Maier et al. [12]. The study contains the same four process configurations. Two cases which are purely operating with biomass as feedstock, case 1 using bark and case 2 using straw, respectively. Additionally, for both feedstocks a biomass-and-power-based process concept is assessed. In these cases, a PEM electrolyser is integrated providing hydrogen stoichiometrically to the methanol forming reactions. While in previous work the methanol reactor was modelled as an isothermal fixed-bed reactor, in this study a multi-stage quench reactor system is applied.

Additional feature – Ideal reactor length

As described in section parameter variation framework, another implemented functionality of pyteea_sc is the integration of user scripts which can be included within the parameter variation run.

In this work, an additional feature is implemented that finds the ideal reactor length for each reactor stage in order to prevent over- and underestimation of their size, ensuring that the subsequently calculated costs are properly estimated.

Table 1. Reactor dimensioning parameters.

Parameter	Definition
Control parameter	$T_{eq} - T_{out,Ri} = 5 \pm 0.5$ K
Control variable	Reactor length

The routine is using a second RPlug reactor for each reactor stage which is completely oversized to ensure that it reaches equilibrium. A Transfer block ensures that each stage is calculated with the same gas mixture as the regular reactor stage is working with. The methanol formation of each reactor stage over the temperature together with the equilibrium line is given in Figure 3.

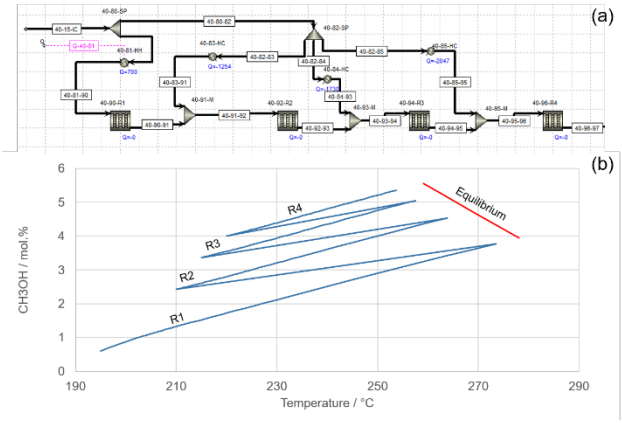

Figure 3. Reactor stages in flowsheeting simulation (a); Reaction profiles for all four reactor stages (b)

A Newton algorithm is applied to find the ideal length of reactor including an exiting function when the composition reaches equilibrium in order to prevent the algorithm from running into an asymptotic regime.

RESULTS & DISCUSSION

Parameter variation framework

For the conducted parameter variation, the execution time was 36:39 min of which 35:02 min where communication between Python and AspenPlus. If compared to previous approaches for coupling AspenPlus® with a programming software like in the case of Lopez et al. [13], the execution time is much higher which limits the framework's application for operation within larger optimizer routines. For usage in this field, the recording options should be set to a minimum. Especially the implemented additional feature can definitely be improved by that and should be seen as a demonstration of the implementation possibility of user script within the variation loop rather than an ideal solution for reactor design.

Since in the case of this study, the intention of the tool is mainly to make AspenPlus® obsolete for subsequent data analyses such as TEA or LCA, a complete data extraction has been conducted.

Techno-economic analysis results

Figure 4 shows the techno-economic analysis results for bark and straw and their different process concepts for 100 MW_{th} biomass input.

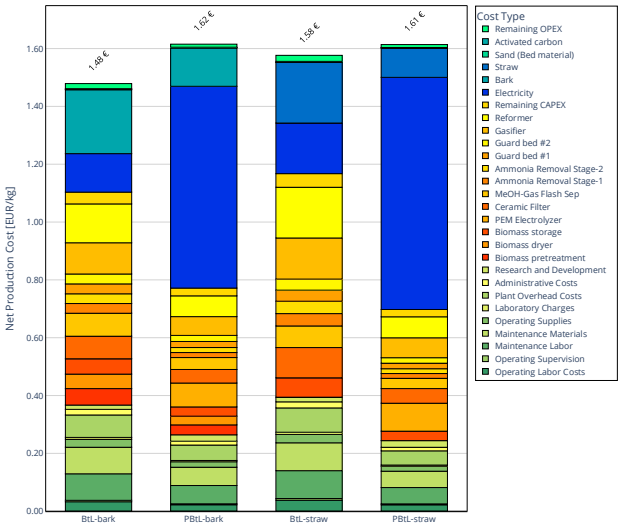

Figure 4. Net production costs for the four process concepts.

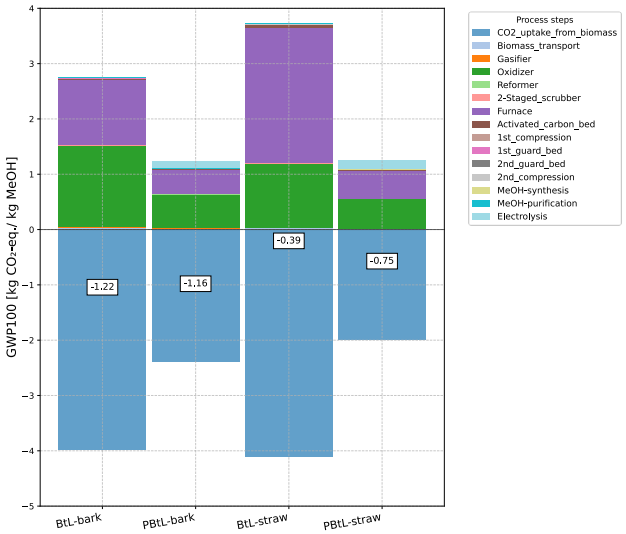

Figure 5. LCA results (Cradle-to-Gate) for the four process concepts. Wind energy is assumed as electricity source.

The main aspects which can be identified in Figure 4 are that the net production costs for the power-and-biomass process concepts are higher than their purely biomass-based representative. Main reason is the production of hydrogen at an electricity price of 105 $€_{2023}$/MWh which makes it more expensive than the hydrogen received by biomass gasification. Nevertheless, the implementation of the additional electrolysers nearly doubles the product output of the processes. Hence, even with the very expensive electricity, the PBtL cases are nearly in the same price range as the biomass-based ones.

Life-cycle analysis results

Using the same simulation data for the ecological analysis of the four process concepts results in the global warming potentials (GWPs) displayed in Figure 5. The figure shows that the processes utilizing bark (BtL-bark & PBtL-bark) have a lower GWP than with straw (BtL-straw & PBtL-straw). Furthermore, when using wind energy as the electricity source, the power-and-biomass based process (PBtL) results in a quite similar GWP in case of bark. In case of straw as a feedstock, the power enhanced process designs results in a 2.2 times higher product yield, which leads to a significant reduction of the process-related emissions since their contribution is divided through a larger amount of product. The gas cleaning steps (reformer, activated carbon bed, 1st guard bed, 2-staged scrubber, 2nd guard bed and 1st compression) and biomass transport (100 km truck transportation) are only contributing 1% for all process concepts and therefore has no significant influence on the GWP. Also, with the MeOH production and purification step (MeOH-synthesis, MeOH-purification and 2nd compression), which are contributing <1%.

The most GWP influencers are the process emissions from the oxidizer flue gas (after biomass gasification) and from furnace (combustion of the purge gas after MeOH-synthesis). These emissions vary depending on the type of biomass. Straw has more emissions from the furnace and bark more from the oxidizer.

CONCLUSIONS & OUTLOOK

The presented work shows the application of a toolbox which allows extensive process analysis together with the implementation of python-based user scripts. Furthermore, the shown tools allow to divide the process analyses from AspenPlus® simulations, which enables extensive analyses without the need of communicating with AspenPlus® simulations. Hence stability issues of the simulation can be taken care of before the data is further processed. Additionally, the introduction of the parameter run data as an intermediate allows to apply more extensive evaluation features on the data batch opening the whole wide world of python-based data analysis tools for the work with rigorous process simulations in AspenPlus®.

Due to the tool's modular programming approach further objects which allow communication with other simulation or modeling tools could be implemented and included in the list of potential `linked_objects`.

Currently, the tool is completely based on YML files and python scripts and its usability could be improved by including a graphical user interface for the input data definition.

DIGITAL SUPPLEMENTARY MATERIAL

Digital supplementary material for the mentioned benchmark case is being made public on LAPSE:2025.0042 [5].

ACKNOWLEDGEMENTS

This research is funded by the German Federal Ministry of Digital Affairs and Transport (BMDV) as part of the Funding Programme Renewable Fuels in the project Methanol-to-Sustainable Aviation Fuels (M2SAF).

REFERENCES

1. Maier, S. and J. Weyand, aspenarserDLR - A Python-AspenPlus interface. zenodo (2025).
2. Maier, S. and J. Weyand, pyteea_tea - A python-based techno-economic assessment tool. zenodo (2025).
3. Weyand, J. and S. Maier, pyteea_lca - A python-based life cycle assessment tool. zenodo (2025).
4. Scott, J.A. and T.A. Adams, Biomass-Gas-and-Nuclear-To-Liquids Simulation Files. (2018), PSE: PSE Community.
5. Maier, S., A Framework Utilizing a Seamless Integration of Python with AspenPlus® for a Multi-Criteria Process Evaluation - Benchmark case. LAPSE:2025.0042. (2025).
6. Peters, M.S., K.D. Timmerhaus, and R.E. West, Plant Design and Economics for Chemical Engineers. 2002).
7. Albrecht, F.G., et al., A standardized methodology for the techno-economic evaluation of alternative fuels – A case study. Fuel 194: 511-526 (2017) 10.1016/j.fuel.2016.12.003.
8. Maier, S., et al., Techno-economically-driven identification of ideal plant configurations for a new biomass-to-liquid process – A case study for Central-Europe. Energy Conversion and Management 247 (2021) 10.1016/j.enconman.2021.114651.
9. Habermeyer, F., et al., Techno-Economic Analysis of a Flexible Process Concept for the Production of Transport Fuels and Heat from Biomass and Renewable Electricity. Frontiers in Energy Research 9 (2021) 10.3389/fenrg.2021.723774.
10. Mutel, C.L., L. de Baan, and S. Hellweg, Two-step sensitivity testing of parametrized and regionalized life cycle assessments: methodology and case study. Environ Sci Technol 47(11): 5660-7 (2013) 10.1021/es3050949.
11. Steubing, B., et al., The Activity Browser — An open source LCA software building on top of the brightway framework. Software Impacts 3 (2020) 10.1016/j.simpa.2019.100012.
12. Maier, S., R.-U. Dietrich, and Y. Rahmat, Identifying the Ideal Process Configuration for a Green Methanol Production Plant Dependent on Economic Boundary Conditions. 36th International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems (ECOS) (2023). 1422-1431.
13. Muñoz López, C.A., et al., A process simulator interface for multiobjective optimization of chemical processes. Computers & Chemical Engineering 109: 119-137 (2018) 10.1016/j.compchemeng.2017.09.014.