





Article

Application of Multi-Software Engineering: A Review and a Kinetic Parameter Identification Case Study

Viktória Flóra Csendes ^{1,2}, Attila Egedy ¹, Sébastien Leveneur ² and Alex Kummer ^{1,3,*}

¹ Department of Process Engineering, University of Pannonia, Egyetem st. 10, H-8200 Veszprém, Hungary; viktoria_flora.csendes@insa-rouen.fr (V.F.C.); egedy.attila@mk.uni-pannon.hu (A.E.)

² INSA Rouen, UNIROUEN, Normandie University, LSPC, UR4704, 76000 Rouen, France

³ ELKH-PE Complex Systems Monitoring Research Group, University of Pannonia, Egyetem st. 10, H-8200 Veszprém, Hungary

* Correspondence: kummer.alex@mk.uni-pannon.hu

Abstract: Limitations regarding process design, optimization, and control often occur when using particular process simulators. With the implementation of connection methodologies, integrated tools could be made by coupling popular process simulation software with each other or with programming environments. In the current paper, we summarized and categorized the existing research regarding the application of multi-software engineering in the chemical industry, with an emphasis on software connections. CAPE-OPEN, COM, OPC, and native integration were discussed in detail, with the intention to serve as a guide for choosing the most suitable software combination and connection. These hybrid systems can handle complex user-defined problems and can be used for decision support, performing custom unit operations, operator training, process optimization, building control systems, and developing digital twins. In this work, we proposed the use of process simulator Aspen HYSYS linked together with the numeric computing platform MATLAB to solve a reaction kinetic parameter identification problem regarding the production of γ -valerolactone.

Keywords: CAPE-OPEN; multi-software engineering; software linking; co-simulation; parameter identification; process system engineering



Citation: Csendes, V.F.; Egedy, A.;

Leveneur, S.; Kummer, A.

Application of Multi-Software Engineering: A Review and a Kinetic Parameter Identification Case Study. *Processes* **2023**, *11*, 1503. <https://doi.org/10.3390/pr11051503>

Academic Editor: Fausto Gallucci

Received: 4 April 2023

Revised: 5 May 2023

Accepted: 11 May 2023

Published: 15 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Using process simulation software for designing and optimizing chemical technologies is an active part of developing such processes from the 1980s [1,2]. Several engineering hours and laborious experiments could be spared using computer-aided methods, as they are capable of simulating whole chemical plants in a user-friendly graphical environment. There are many commercial simulation software available on the market today with only a few bigger vendors, but it is mainly individual companies providing that software [3,4]. The variety suggests that those programs are developed for niche parts of the chemical industry, based on valuable information coming from industrial data and experiments provided by the company developing them [5].

Today, a process simulator expected to handle the design of an entire chemical plant has to cover many areas; starting from the reaction stage, it has to include vast databases of components, binary interactions, thermodynamic methods [6], and unit models [7–9]. It should be able to use detailed equipment design and scale up [10] and it should be able to perform parameter estimation [11], data regression, sensitivity analysis [12], optimization [13–15], and the design of control systems [16–18] with the expectation of simulating the real systems precisely.

It is too much of a request for a single tool to provide models of such a large application range and perform all of these requirements properly. Therefore, engineers have to face several limitations regarding their performance and capacity. Individual process simulators

can perform excellently in the field where they were developed, but struggle with complex problems requiring different fields of knowledge to solve them [19].

The use of equation-based numerical environments based on programming languages has the advantage of building custom models, although it often requires advanced programming skills to build user-defined equation systems and utilize algorithms or built-in toolboxes. For example, in performing single unit operations or estimating parameters, user-defined functions come in handy for unique and complex problems, but designing entire chemical plants with user-formulated scripts is unnecessarily laborious work.

Commercial process simulators have these unit operations built-in [1,20,21], where the mathematical equation systems are behind easy-to-use graphical interfaces. That way, the mathematical models are ready-made with the right algorithms already chosen. However, the manipulation or customization of such objects is often restricted. The programming environments have the advantage of containing more complex numerical methods than process simulators, this means that optimization algorithms and response surface methodology can also be used to achieve a wider application range with normal sensitivity or limited optimization [22]. Those limitations led to the demand of utilizing only parts of these environments—which are missing from the others—to solve the problem in cooperation with each other.

Those limitations led to the demand for developing methods that utilize different parts of these environments in cooperation with each other. That way there is no need for new, complex software; engineers can select and work with multiple software that are best for solving the actual problem, without making compromises in limited areas within a single environment. One possibility is to build connections between existing software, to utilize the parts that the problem requires, or to develop simulators by different companies with the same framework to be interchangeable, as the development continues and they have to adapt new features from other software.

Our aim with this study was to summarize the existing research around multi-software engineering and showcase the connection possibilities between different commercial process simulators and numerical tools.

From the early 2000s [6,7] to this day, multi-software engineering developed from the idea of utilizing multiple software for a single problem, to the demand of precisely describing the code and considering different linking methodologies based on complex tasks that cannot be executed with a single tool. Nowadays, the adequate literature focuses on the connection of the simulation software rather than the problem itself and, based on that, differentiates between their performances [13,23,24].

In 2020, Abril [25] published a freeware on interface linking between MATLAB and Aspen Plus that made the connection process more widely available. Therefore, most of the newly-described systems focus on this software and, as Bartolome and Van Gerven [24] mentioned in their work, Aspen Tech is the leading process simulator provider for the chemical industry.

Besides those, another important solution to this software linking problem started to develop in the early 2000s and that is the concept of CAPE (Computer Aided Process Engineering), more specifically CAPE-OPEN [26]. It is based on the fact that the existing process simulators, developed for specific areas of modeling, have limitations regarding capability, although over time they are expanding their simulation packages, mainly with experimental information from companies in the field or from other simulation environments [27]. The success of adapting those changes depends on the simulation environment, how it is built, and how open it is to modifications. The CAPE-OPEN Interface Standard defines a set of software interfaces that allow plug-and-play inter-operability between a given process modeling environment (PME) and a third-party process modeling component (PMC) [26].

In this sense, another computer-aided method has to be mentioned and that is CAMD (Computer-Aided Molecular Design) [6]. The CAPE concept targets mainly the design and modeling of equipment and processes, while CAMD deals with the design of compounds and properties estimation on a molecular level [28]. It is fundamental to study

materials down to the level of atoms and charges, as the more accurate our models are in the earliest stages of design, the closer we are to modeling industry-scale processes that describe real life well. Until the development of rigorous computational resources, the molecular design almost exclusively relied on experimental-based trial-and-error methods and knowledge-based frameworks [29]. With the rise of machine learning, especially deep learning techniques, the optimization and characterization of molecules improved, due to the reduced computational cost, easy incorporation within mathematical models, and invertibility [30]. Today's challenges in molecular design with deep learning techniques include molecular representations, major deep generative architectures, benchmarking, and evaluation metrics [31].

If we take a look at the last decades of published research on the overall topic in the Scopus database [32] from 2000 to 2022, based on the keywords “co-simulation”, “multi-software”, “software linking”, “software coupling” paired with “hybrid model”, and “machine learning”, the following can be seen in Figure 1. The number of published articles in chemical engineering have been exponentially increasing for the last five years based on this data, meaning it is a current topic to study. Moreover, the co-occurrence of keywords in the corresponding articles are multifaceted; several different fields benefit from software connecting methods. Besides the keywords mentioned above, process simulation, optimization, and control appeared the most.

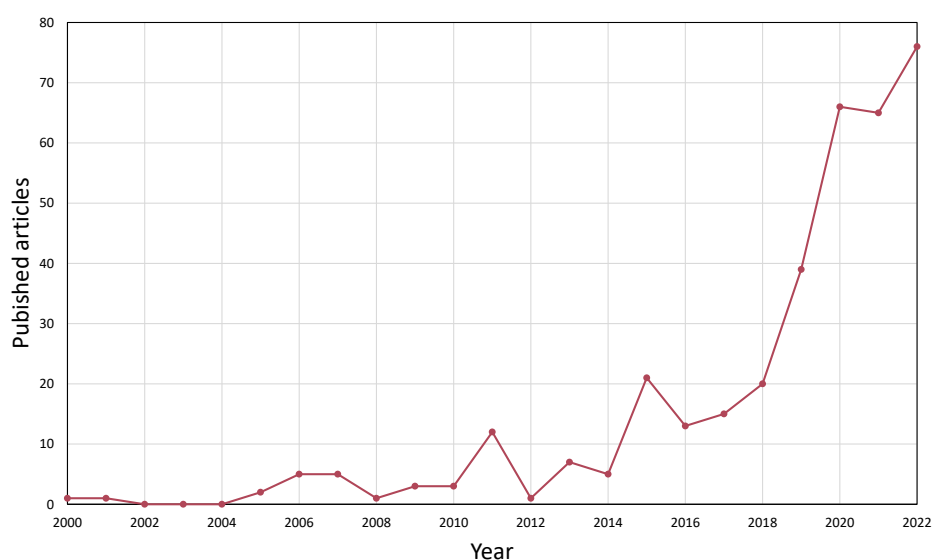


Figure 1. Research interest in multi-software engineering between 2000 and 2022.

It is worth mentioning that, although the ideas behind connecting process simulation software with each other or an external tool have existed for two decades [6,7], there is still no collective name for them. Software linking, software coupling, software integration, co-simulation, hybrid model, or multi-software modeling are often used to describe such processes. Therefore, finding adequate literature on the topic of chemical engineering is quite challenging sometimes; in the current paper we proposed the use of the term multi-software engineering.

We present an application example where the shortcomings of commercial process simulators with restricted reaction kinetic structures can be solved, where we identify simpler reaction kinetics for γ -valerolactone (GVL) production, so the reaction system can be calculated. The product itself was derived from biomass, of which valorization is a highly researched topic in today's era of food, fuel, and economic crisis [33]. γ -valerolactone in this case was produced from butyl-levulinate with catalytic hydrogenation, the kinetics of which are rather complex. For the identification of a directly implementable reaction kinetics, we implemented the dynamic simulator of the GVL production reactor using Aspen HYSYS, and we identified the reaction kinetic parameters using a MATLAB optimization algorithm.

A major shift in process development has been happening for the last few decades with the appearance of reliable data-driven models. Process simulators that are inherently based on first-principle models are reaching a limit of accurately representing more complex problems, not to mention the factor of time and cost. Multi-software engineering is a novel tool for connecting a priori and a posteriori models, in the form of robust process simulators and statistical solvers to keep up with the evolution of information technology. Our contribution to the current state-of-the-art of this research field and the road-map of this paper is as follows:

- We systematically organized the available novel research regarding the application of multi-software engineering in the chemical industry (see Section 2).
- We presented various process simulation software, numerical solver tools, and their connection methodologies applied to a wide range of complex engineering problems to serve as a reference guide for their utilization, suitable both for academic and industrial use.
- We identified future challenges and the fields in which multi-software engineering can be beneficial, but not utilized yet (see Sections 2 and 3).
- We showed the importance of the CAPE-OPEN standard in the interoperability of different process simulator tools (see Section 4).
- Finally, in a case study, we developed a framework linking Aspen HYSYS with the MATLAB environment for kinetic identification, presenting the advantage of building a model in Aspen's graphical interface, but also utilizing the computing capacity and custom operations that a MATLAB environment could provide (see Section 5).

2. Literature Overview

In this section, we attempted to collect some of the most significant research in the field of multi-software engineering. Showing the utilization of various software linking systems to solve complex and diversified problems in the chemical industry, in the hope of shedding light on this developing research area of Computer Aided Process Engineering. The following articles were chosen to showcase the most used simulation environments and numerical tools, their linking possibilities and to highlight the fields of application in four categories; process modeling, process design and optimization, process control and safety, and data-driven methods.

2.1. Process Modeling

If we take a look at the earlier literature on this topic, in the work of Fontalvo et al. [7], multi-software modeling was presented as a tool for calculating the behavior of a membrane unit. An Aspen Plus distillation tower model was connected with a MATLAB subroutine, that was used to attach the model to a membrane unit. No descriptions were made of the connection between these two software or the implementation because the focus was on solving the actual modeling problem, rather than presenting the framework used for the software connection. Later on, one could notice a shift in the literature, where software linking will be more and more showcased, rather than just mentioned as a tool for problem-solving [10,23,24].

As a continuation of the early idea of linking process simulators with coding environments, in Fontalvo's [34], a decade later, the principles of the process were finally described. An Aspen Plus and MATLAB hybrid system was created with an MS Excel connection, that was used in a distillation and vapor permeation process case study to show the protocol of utilization. In this case, the Excel spreadsheet worked as a mediator, which organized and sent information regarding stream results and additional parameters between the two software.

Kiss et al. used MS Excel, not only as a mediator but as a visualization tool. They studied an industrial sulfuric acid plant implemented in a dynamic simulation model with process simulator gPROMS and a graphical user interface created in Microsoft Excel [35]. They could quickly develop an operator screen presenting the process flowsheet in Excel, while the data are exchanged between gPROMS and the spreadsheet. The developed dy-

dynamic model was capable of detecting changes in product quality, convenient for operator training and scenario evaluation.

In earlier software linking approaches, MS Excel was often used as middle-ware; mostly for data reading and organization. In later works, the use of a third software was less desired, as most commercial software vendors improved their connection methods, as could be seen in the following subsections. That way, a third software tool was only limiting the process, giving more space for errors.

In a recent study, Samei and Raisi tried to overcome the scarcity of commercial process simulation software regarding membrane units, with custom gas separation models [36]. A code was first developed for mass transfer calculations, using the solution–diffusion mechanism in the FORTRAN programming language. Integration of this outside calculation to the flowsheeting simulator is made possible by the Aspen Custom Modeler (ACM) unit. This is a built-in feature of the Aspen Tech software, where additional equations could be described and implemented in the software without using outside tools.

Overall, the advantage of the direct relation between two software without the use of middle-ware includes; the reduction in computational resources, because there is no storing limit, avoidance of data loss, and the possibility to use the obtained database for data mining. Although, the use of a middle-ware could be necessary when working with multiple dynamic simulators.

In the works of Mikkonen et al. [37], the dynamic co-simulation of multiple units were studied; a circulated fluidized bed boiler, air separation unit, and a CO₂ purification unit, all communicating through MATLAB. For the boiler and turbine side, the software Apros was used, which is a tool for dynamic modeling power plants and turbine islands. For the air distillation and CO₂ purification side, Aspen Plus Dynamics was used. The challenge in connecting different dynamic models is the delay in their numerical integration times. Therefore, a mediator tool is needed, in this case in the form of MATLAB Simulink. This way a master program keeps the two simulators clock's synchronized. Data transfer between two dynamic simulators is only possible while their clocks are in sync and the models can be reloaded.

However, many more studies can be found in which the reduction of energy consumption [15], equipment comparison [9], or properties estimation [38] was discussed using multi-software tools for modeling.

Another field of software in modeling that often utilizes software linking is computational fluid dynamics (CFD). However, these cannot be labeled as typical process modeling simulators, rather as an expansion to observe and calculate transport phenomena inside a system. Zuhang et al. approached the municipal solid waste incineration (MSWI) process with numerical simulation methods and CFD software with their many difficulties [23]. The investigated process is quite complex due to the different characteristics of the many subprocesses, therefore it is not feasible to perform the modeling with a single simulation tool. They reviewed the current research progress in the field and analyzed the proposed software coupling methods. The two main paths are based on self-developed or commercial software. In the latter, the combinations of FLUENT, Aspen Plus, MATLAB, Phoenix, Visual Studio, Flash, IMMS, Gaebed-ss, and ANSYS CFX can be found. They also mentioned that the development of a digital twin model is a possibility based on these software combination methods.

Da Rosa and Braatz [39] used the open-source CFD software OpenFOAM for the multi-scale modeling of crystallization and the 3D design software SolidWorks was used to generate the computer-aided design (CAD) model and to set and describe boundary conditions.

Egedy et al. [40] used the CFD software COMSOL Multiphysics with a Livelink connection to MATLAB to study spatial and temporal changes of the viscosity and the density along the reactor geometry. Usually, optimization could be interpreted only as a comparison of scenarios [41] in the case of CFD. However, with the use of external numerical environments, many black-box optimization algorithms are available. The connection with MATLAB made

it possible to perform objective function-based optimization on the system to optimize critical parameters; initial catalyst concentration, process temperature, and pressure.

Finally, the most diverse tool for software linking has to be mentioned. Morales-Rodríguez et al. investigated the integration of a set of diverse computational tools employing CAPE-OPEN, combining different Process Modeling Environments (PMEs) and Process Modeling Components (PMCs) to show the interoperability of the different computational tools [42] by at first linking Simulis Thermodynamics as the graphical interface PME with ICAS-MoT as the PMC through a DLL file as the middleware. In another case study, they used a multiscale approach, with built-in unit operations found in the PME—ProSimPlus. The variables were obtained and calculated by the CAPE-OPEN unit operation using the ICAS-MoT solver. COFE as a different PME was also used in the calculations for the comparison of interoperability.

In this sense, another unique tool has to be showcased, which supports the interoperability between different coding environments. MOSAIC is a web-based modeling environment with a unique modular equation-based modeling tool, using the concept of symbolic mathematical language and code generation [43]. It allows the re-usability of models and model parts. The functionality of code generation is that mathematical content is documented with standard and widely used languages (e.g., LaTeX), then the specified model's program code can be directly rendered to different languages. This makes them suitable for integration into many different numerical environments, without prior knowledge of them.

Tolksdorf et al. implemented a model-driven approach for customized code generation for simulation and optimization purposes [8]. The many case studies were carried out in an equation-based flowsheet simulation using CAPE-OPEN interfaces, to highlight the strength and variety of these methods. First, a simplified membrane module was described with MOSAIC modeling and Python code. An integrated mixer and splitter unit operation with two outlets was created with the code generated by MOSAIC modeling and SciLab's "fsolve" function was used as a solver. Using CAPE-OPEN, SciLab unit operations can also be solved inside COFE and Aspen Plus flowsheeting simulators, showing the versatility of the software linking. A final example of a differential-algebraic equation system describing a system of two reactions and three compounds was investigated using MATLAB for code generation and as a solver.

As shown by the collected research data in this section, one of the most applied fields for multi-software engineering regarding the chemical industry is modeling. Software linking is often utilized in modeling complex unit operations, where the model is built in a commercial simulator and an external numerical solver is connected, in which custom equation systems are solved to support the development of an adequate model. In Table 1, we organized and categorized the above-mentioned works by application.

Table 1. Collected and categorised research on modeling regarding multi-software applications.

	Research Focus	Software	Connection	Ref.
Custom unit operations	Membrane unit modeling	Aspen Plus, MATLAB	n.a.	[7]
	Use and comparison of rubbery and glassy membranes in a multi-stage gas separation process for separation of the CO ₂ /CH ₄ mixtures	Aspen Plus, FORTRAN	Aspen Custom Modeler (ACM)	[36]
	Simplified membrane module, integrated mixer and splitter unit operation with two outlets; differential-algebraic equation system describing a system of two reactions and three compounds	CAPE-OPEN, SciLab, Python, MOSAICmodeling, COFE, Aspen Plus, MATLAB (fsolve, fmincon)	n.a.	[8]
	Paraffin wax deposition in longer wells and pipelines	Aspen HYSYS, MATLAB	n.a.	[38]
	Hybrid distillation and vapor permeation system for the partial dehydration of ethanol	Aspen Plus, MATLAB	MATLAB User Model (Mediator: Ms Excel VBA)	[34]
	Heat Integrated Distillation Column	Aspen Plus, MATLAB	ActiveX (Mediator: Ms Excel VBA)	[9]
	Dynamic modeling of Circulated Fluidized Bed Boiler, Air Separation Unit and CO ₂ Purification Unit	Apros, Aspen Plus Dynamics, MATLAB Simulink	OPC (Mediator: MATLAB Simulink)	[37]
CFD	Design of a SIDEM unit; triple effect desalination plant coupled with thermo-vapor compressor	Aspen HYSYS, MATLAB (frnicon)	n.a.	[44]
	Spatial and temporal changes of the viscosity and the density in a Vegetable Oil Carbonation Reactor	COMSOL, MATLAB	LiveLink	[40]
	Combined antisolvent-cooling crystallization of lovastatin with methanol as solvent and water as antisolvent	openFOAM (openCrys), SolidWorks	n.a.	[39]
Kinetic parameter identification	Municipal solid waste incineration process with numerical simulation methods based on mechanical grates	FLUENT, Aspen Plus, MATLAB, Phoenix, Visual Studio, Flash, IMMS, Gaebed-ss and ANSYS CFX	n.a.	[23]
	Fitting of industrial sulfuric acid plant kinetic parameters to the real plant data for modeling	gPROMS, MS Excel	n.a.	[35]

Table 1. Cont.

	Research Focus	Software	Connection	Ref.
Multiscale modeling	Reynolds-Averaged Navier–Stokes model with variable properties for macromixing with a multi-environment PDF model for micromixing; using a spatially varying population balance equation, energy balance and scalar transport equations	openFOAM (openCrys), Solid-Works	n.a.	[39]
	Steam process drive sizing methodology to replace a condensing steam turbine by a backpressure one	Aspen Plus, MATLAB	ActiveX	[10]
	Direct Methanol Fuel Cell modelled through multiscale approach; thermodynamic property model generation, fugacity and activity coefficient calculations for methanol, water and methane	CAPE-OPEN, COFE, ICAS-MoT, ProSimPlus, Simulis Thermodynamics	DLL file	[42]
Scenario modeling	Dynamic modeling of a sulfuric acid production plant	gPROMS, MS Excel	n.a.	[35]
Reduce energy consumption	Propane pre-cooled mixed refrigerant LNG plant	Aspen HYSYS, MATLAB (GA)	ActiveX	[15]
	Separation of a benzene-toluene mixture	Aspen Plus, MATLAB	ActiveX (Mediator: Ms Excel VBA)	[9]

2.2. Process Design and Optimization

In the previous section, we could see a plethora of application examples for multi-software engineering, mostly single-unit modeling and solving custom operational problems. This method is powerful enough to be applied to whole chemical plants and processes with multiple different unit operations. In this subsection, we focused on showing the application to mainly optimization problems, which include multiple equipment and processes.

For example, Bayoumy et al. took a step further and, instead of optimizing a single unit, applied the previously mentioned Aspen HYSYS and MATLAB connection to an extensive green-field saturated gas plant with numerous components and operational units [12]. In their case, a steady-state simulation was optimized using a merge between sensitivity analysis and stochastic optimization techniques.

Zhu et al. took an in-depth look at a multiple gas feed sweetening process and aimed to develop a superstructure using a simulation-based framework [13]. They also collected literature data from the past 5 years of simulation-based optimization methods and software used, focusing on the algorithm in each case.

Aspen HYSYS linked together with MATLAB is often used in various optimization problems regarding the oil industry, namely, genetic algorithm (GA) [14,45] and particle swarm optimization (PSO) [12,46] techniques are used mostly in plant optimization tasks as the following works describe.

For process optimization and structure comparison with the same software framework, a PSO algorithm could be used [47,48], but GA provides greater confidence than PSO in optimization with wider ranges of constrained bounds [12].

Li et al. used the SADDE algorithm in a MATLAB and Aspen Plus environment for an ethylbenzene and styrene separation distillation unit [49]. Ruiz-Femenia et al. developed a Generalized Disjunctive Programming (GDP) framework for a bi-objective problem, which aims to minimize the Total Annual Cost (TAC) and the Dow's Fire and Explosion Index (F&EI) for process safety while finding the optimal values for the continuous variables and the best flowsheet configuration in a superstructure [50]. The synthesis was simulated in Aspen HYSYS and the modeling framework was built in MATLAB and solved by the TOMLAB optimization environment. Superstructure optimization was also investigated for CO₂ capture, linking Unisim Design with MATLAB using a GA algorithm [51]. Multi-objective optimization problems also utilize Aspen Tech software and the MATLAB environment using GA [52,53] and NSGA-II [54] algorithms.

These works are great examples that show the many software that have to be involved in building and optimizing processes with multiple different operational units. One can also notice the importance of choosing the right connection method between software. With the use of metaheuristic algorithms instead of gradient-based ones, the problems of obtaining derivative information through the numerical noise can be avoided [12].

Reducing energy consumption [9,15], improving economic performance [16,55], and optimizing emission values [56–58] are other applications that often utilize multi-software engineering for the optimization task.

Mounaam et al. investigated an industrial sulfuric acid plant with a contact process and developed dynamic models for simulation and optimization purposes [55]. The simulation was provided by Unisim Design; its graphical interface made it possible to use the developed model for operator training or digital twin technology for the real plant. It was connected with the programming language Python, where data reading and writing from the simulation were possible, and performed optimization with a parametric study for cost and SO_x emission minimization.

Brambilla et al. [16] performed an unconventional real-time optimization (RTO) with an offline simulation model built in UniSim Design. The optimal setpoints were obtained analytically, as a function of the feed, products, and energy price fluctuation.

Alabdulkarem and coworkers studied the optimization of a propane pre-cooled mixed refrigerant LNG plant to reduce power consumption [15]. The optimization of the refrigerant mixture was carried out in MATLAB with Genetic Algorithm (GA), that communicated

through the `actxserver` command which was used to create the HYSYS COM server to access the plant model and its variables.

Radó-Fóty et al. used Aspen Plus and MATLAB coupling for the optimization of a coke oven gas purification process based on gas composition requirements [56]. The steady-state model in Aspen Plus sent the results to MATLAB, which was used to provide operational parameters for the technology. The validation of the model was carried out using industrial data.

Briones-Ramírez et al. focused on the procedure of linking Aspen Plus and MATLAB in an example of minimizing the number of stages in a Petlyuk sequence with a detailed multi-objective optimization [59].

Furda et al. developed a steam process drive sizing methodology with the use of Aspen Plus and the MATLAB environment, in a case study to replace a condensing steam turbine with a backpressure one [10]. As they found, previous utility system optimization problems mainly focused on process-side modeling, while the proposed method considered coupled stream- and process-side modeling and also implemented the real operational parameters of the industrial case study system [60–62]. They also mentioned the lack of information regarding the interface linking methodologies and their capabilities and provided a detailed description to set up the Aspen Plus-MATLAB link via a local ActiveX server.

Batolome and Van Gerven also found the lack of description for linking software; therefore, they studied different connection methodologies for the popular process simulator Aspen HYSYS [24]. They described four interconnection approaches for data exchange—direct, indirect, internal spreadsheets, and data tables. A test was carried out with AspenTech example simulation files, both steady state and dynamic, to test the different connection methods to the different external tools—Microsoft Excel VBA, MATLAB, Python, and Unity respectively. Based on the evaluation of the results, the following three criteria were created—average connection time delay, accuracy of input connection, and overall consistency of output. The authors found that the internal spreadsheet connection performed properly in all cases, but found severe limitations in each case.

Optimization algorithms are usually not part of the process simulators, or their capacity is severely limited compared to the wide variety of algorithms implemented in coding environments (such as MATLAB/Python). Applying a suitable connection method, an external solver for these problems can be applied to use an appropriate and reliable optimization algorithm. Table 2 contains the articles collected in this subsection with regard to optimization and sensitivity analysis.

Table 2. Collected and categorized research on design regarding multi-software applications.

	Research Focus	Software	Connection	Ref.
Optimizing operating conditions	Whole green-field saturated gas plant optimization	Aspen HYSYS, MATLAB (GA, PSO)	ActiveX	[12]
	Split ratios of sour gas streams to different feed stages and circulating flow rate of the lean solvent	Aspen HYSYS, MATLAB (PSO)	ActiveX	[13]
	Post-combustion CO ₂ recovery unit utilizing the absorption refrigeration system	Aspen HYSYS, MATLAB (GA)	ActiveX	[14]
	Crude oil distillation systems with preflash units	Aspen HYSYS, MATLAB (GA)	ActiveX	[45]
	Heavy hydrocarbon removal process that reduces the heating value of LNG to meet desired specifications	Aspen HYSYS, MATLAB (PSO)	ActiveX	[46]
	LNG plant optimization on optimum composition of refrigerant mixtures and pinch temperatures	Aspen HYSYS, MATLAB (GA)	ActiveX	[15]
	Coke oven gas purification process based on gas composition requirements; minimizing energy consumption, H ₂ S and NH ₃ content	Aspen Plus, MATLAB (fmincon)	ActiveX	[56]
Multi-objective optimization	Sulfur recovery units using detailed reaction mechanism	Chemkin Pro, Aspen HYSYS, MATLAB (GA)	n.a.	[52]
	Natural gas liquefaction process	Aspen HYSYS, MATLAB (GA)	ActiveX	[53]
	Operation of the product separation process in a methanol to propylene plant	Aspen Plus, MATLAB (NSGA-II)	ActiveX	[54]
	In a Petlyuk sequence minimizing the number of stages in a pre-fractionator, the number of stages and the heat duty in the main column	Aspen Plus, MATLAB (GA)	Local OLE automation server	[59]
	Biobjective optimization of minimizing the Total Annual Cost and the Dow's Fire and Explosion Index in the processes of benzene chlorination and the production of methanol	Aspen HYSYS, MATLAB, GAMS, TOMLAB	n.a.	[50]

Table 2. Cont.

	Research Focus	Software	Connection	Ref.
Process structure optimization	Coupled stream- and process-side modeling for utility system optimization	Aspen Plus, MATLAB	ActiveX	[10]
	Four distillation-based configurations in propylene-propane separation process	Aspen HYSYS, MATLAB (PSO)	ActiveX	[48]
	Cryogenic distillation and membrane separation in a helium extraction processes integrated with nitrogen removal units	Aspen HYSYS, MATLAB (PSO)	ActiveX	[47]
	Double-effect distillation and self-heat recuperation technology in an ethylbenzene/styrene separation process	Aspen Plus, MATLAB (SADDE)	ActiveX	[49]
	Comparison between a classic distillation column and a Heat Integrated Distillation Column	Aspen Plus, MATLAB	ActiveX (Mediator: Ms Excel VBA)	[9]
Superstructure optimization	CO ₂ capture configuration and four different types of structural modifications	UniSim desig, MATLAB (GA)	n.a.	[51]
	Multiple gas feed sweetening process	Aspen HYSYS, MATLAB (PSO)	ActiveX	[13]
	Benzene chlorination and the production of methanol	Aspen HYSYS, MATLAB, GAMS, TOMLAB	n.a.	[50]
Sensitivity analysis	Number of stages of a debutanizer and deethanizer tower; amount of recycled lean oil and split ratio	Aspen HYSYS, MATLAB (GA, PSO)	ActiveX	[12]
	The effect of membrane selectivity on membrane area and reboiler duty for the partial dehydration of ethanol	Aspen Plus, MATLAB	Matlab User Model (Mediator: Ms Excel VBA)	[34]
	Steady state simulation of a sour water stripper and dynamic simulation of a depropanizer; varying the input temperature, pressure and molar flows of three components	Aspen HYSYS, Matlab, Python, Unity	ActiveX, (Mediator: MS Excel)	[24]
	Analysis on parameters that can be changed and that are fixed during operation in a coke oven gas purification process	Aspen Plus, MATLAB	ActiveX	[56]

Table 2. Cont.

	Research Focus	Software	Connection	Ref.
Dynamic optimization	Real time optimization of a propane-propylene critical distillation unit	UniSim Design	n.a.	[16]
	Maximizing the amount of energy produced and the amount of SO ₂ converted into products, minimizing the total amount of SO _x released to the atmosphere	gPROMS, MS Excel	n.a.	[35]
	Industrial sulfuric acid plant with contact process; minimizing emissions, maximizing production performance and revenue	Unisim Design, Python	n.a.	[55]

2.3. Process Control and Safety

In the previous sections, we talked about equipment modeling, design, and process optimization applied to both dynamic and steady-state systems. Dynamic simulation models could be utilized to investigate process control and apply safety measures with the help of multi-software engineering too.

Tuan et al. designed the control system of a depropanizer column using the Aspen HYSYS and MATLAB-SimuLink co-simulation setup, where the control objective was to reduce the variation in product quality [17]. The dynamic model of the distillation column was built in Aspen HYSYS, whereas MATLAB-Simulink was used for designing the model predictive control (MPC) for the multivariable process. The connection was made through MATLAB with the HYSYSLIB toolbox, which uses HYSYS spreadsheets to read and write data for system identification and MPC tuning.

Chinprasit et al. also developed a model predictive control (MPC) for a vinyl chloride monomer process by integrating the concept of plant-wide control and subsystem partitioning to reduce the complexity of the control system optimization [63]. Aspen Plus Dynamics was used to model the thermal cracking, quench, and distillation processes; while in MATLAB-Simulink, the partitioned two squared linear systems were built by the MPC controller block. The connection between the software was built through the AMSimulation block in Simulink which utilizes COM technology.

Other simulators were connected with the use of MATLAB-SimuLink in multi-variable automation problems in the works of by Ryu et al. [18] and Dos Santos Vidal et al. [64]. Brambilla et al. [16] implemented hierarchical controlling using MPC [65] and real-time optimization tools for a distillation process, where the simulator was developed in UniSim Design.

From the above examples, it can be seen that MATLAB-SimuLink is a common tool for solving various control problems, as it includes many built-in toolboxes that can connect different flowsheet simulators with customizable control panels.

The following works are examples of applying multi-software engineering to safety problems. Mizuta et al. performed a detailed safety analysis regarding a runaway reaction during methyl ethyl ketone peroxide (MEKPO) production, where venting from a safety valve was constructed [66]. Dynamic computer simulations were carried out in Aspen Plus Dynamics and the safety valve model was built in the Aspen Custom Modeler block. The model was validated by comparing the results against experimental data, using the Advanced Reactive System Screening Tool (ARSST).

Kummer and Varga demonstrated the use of the multi-software framework, applying dynamic HAZOP to a vacuum distillation column [67]. They used Aspen HYSYS as the dynamic simulator of the process with OPC connection, which allows the reach of model variables and modification of the operating parameters of the process with MATLAB, where the data were being evaluated. The deviations to demonstrate controller failures were sent from MATLAB. The Aspen OTS Framework hosted the OPC server, where MATLAB was connected and the simulator was run. The OPC client used the server to access data from the simulator and to send commands too.

Designing control systems and incorporating process safety is another important area of the application field of multi-software engineering; we grouped these in Table 3. Dynamic process simulators with control systems could be connected to external numerical solvers through built-in toolboxes utilizing COM or OPC connections, where custom algorithms are used for controller tuning and optimization.

Table 3. Collected and categorized research on control and safety regarding multi-software applications.

	Research Focus	Software	Connection	Ref.
Control system design	Design of a condenser with two controllers to set the temperature and the liquid level inside	MOSAIC modeling	n.a	[43]
	MPC control system design and optimization for a propane-propylene super-fractionator	UniSim design	n.a.	[16]
	MPC control system design and tuning for a deprotonizer column	Aspen HYSYS, MATLAB-Simulink	HYSYSLIB toolbox (COM)	[17]
	Partitioned MPC control for a vinyl chloride monomer process, containing cracking, quench and distillation processes	Aspen Plus Dynamic, MATLAB-Simulink	AMSimulation block (COM)	[63]
	MPC based plantwide control structure for the separation of off-gas from a polysilicon plant	Aspen Plus, Aspen Dynamics, MATLAB-Simulink	n.a. (COM)	[18]
	Multivariable fuzzy logic-based control system and a classic multi loop PID applied to a refrigeration system	Aspen Plus, Aspen Dynamics, MATLAB-Simulink	n.a. (COM)	[64]
Implementing control algorithms	Cryogenic NGL recovery unit optimization framework to maximize profit with a stable operation of the process	Aspen HYSYS, Python	n.a.	[68]
Safety-equipment modeling	Safety analysis for vent sizing regarding a runaway reaction; comparison of the method developed to the ISO omega safety valve design method	Aspen Plus Dynamic	Aspen Custom Modeler (ACM)	[66]
Process safety	Generalized Disjunctive Programming framework, that aims to minimize the Total Annual Cost and the Dow's Fire and Explosion Index for process safety	Aspen HYSYS, MATLAB, GAMS, TOMLAB	n.a.	[50]
Dynamic HAZOP	Controller failures of a cumene-hydroperoxide vacuum distillation column to produce phenol	Aspen HYSYS, MATLAB	OPC	[67]

2.4. Data-Driven Methods

Data-driven methods could be applied in any field with large data sets and are developing rapidly. The chemical industry is no exception, as these methods have many benefits compared to the classically used a priori model-driven methods. In the previous section, we showcased various optimization problems; in this section, data-driven machine learning tools are featured, which assist complex optimization and control problems applied to process models. Mushin et al. [69] investigated the optimization of crude oil HDT processes, where Aspen HYSYS models were built to simulate real plant data. The obtained process data were used to develop bootstrap aggregated neural network models. They were used in a multi-objective optimization framework, where the minimization of model prediction confidence bounds as an additional optimization objective was added.

Zhu et al. developed an optimization framework for a cryogenic NGL recovery unit using deep learning-based surrogate models [68]. Aspen HYSYS was used to simulate the cold residue reflux process, which was based on industrial data, to construct a dynamic simulation and implement control strategies. The objective was to maximize profit with a stable operation of the process. An optimization framework, with the use of a differential evolution algorithm, was built in Python to manipulate the operating conditions. For validation, two case studies were created to simulate changing market conditions.

With complex optimization tasks, it is often challenging to maintain computational time at a reasonable speed for obtaining solution convergence between steady-states. To overcome this limitation, surrogate modeling of digital twins could lower the time of convergence in process simulators by implementing machine learning methods in place of model-driven methods [70].

Alhajeri et al. studied the performance of model predictive control (MPC) in an ethylbenzene production process, comparing fully-connected and partially-connected recurrent neural network (RNN) models [71]. The simulation of the process was carried out in Aspen Plus Dynamics, while input signals were generated in MATLAB, linking the two software with a local message passing interface (MPI). Similarly, a Python script was created to send random control actions to the simulation for testing the open-loop prediction.

Reinforcement learning is a type of machine learning paradigm that is based on reward and punishment mechanisms and is often used in optimization problems. Kim et al. considered a deep reinforcement learning approach for the optimization of Single Mixed Refrigerant cycles for the LNG liquefaction process [72]. The optimization objective was the minimization of refrigeration power, subject to the constraint of minimum temperature difference. The Deep Q-Network (DQN) framework was implemented in Python, while the process model was built in UniSim Design and MATLAB was used for delivering decisions. They also reported a similar computational performance between the DQN and GA methods. Hedric et al. also utilized reinforcement learning for the online tuning of MPCs in order to reduce nitrogen oxide emissions in an industrial selective catalytic reduction unit [73]. The dynamic model was developed in the Aspen Custom Modeler (ACM) unit.

However, in order to apply the outcome of the model predictions to the real system, the used model/simulator should be an accurate representation of the fundamental system. Without validating experimental or industrial data, the conceptual model is considered invalid and the simulation model cannot be used reliably.

In this sense, the importance of data reconciliation has to be mentioned, as it could filter out random error and improve the quality of measured data for further usage in, e.g., process simulation. The measuring equipment is inevitably laden with error, which affects the data obtained, causing it to not satisfy balance equations [74].

Farsang et al. studied the online monitoring of catalyst deactivation, where the main idea was to perform the simulator development and data reconciliation simultaneously [11]. Industrial data from an online measuring system were available, but the standard deviation of the data set was large for the observed kinetic values. More accurate parameter

estimation was possible with data reconciliation. The results could also contribute to the validation of online analyzers [75].

Vaccari et al. used a geothermal power plant model built in dynamic simulator UniSim to analyze the behavior of pollutants present in the process and estimate the changes based on the current process conditions [57,58]. The model was developed based on real plant data and data reconciliation was performed via optimization internal to the simulation environments. Seman et al. utilized data reconciliation for the tuning of oil well models [76]. With the proposed methodology, automatic steady-state detection and re-estimating of chosen parameters reflect the current operating conditions. Table 4 contains the above-mentioned research on data-driven methods regarding multi-software applications.

Table 4. Collected and categorized research on machine learning regarding multi-software applications.

	Research Focus	Software	Connection	Ref.
Machine learning	Process profit prediction using deep learning based surrogate models	Aspen HYSYS, Python	n.a.	[68]
	Optimization of crude oil HDT processes to develop bootstrap aggregated neural network models	Aspen HYSYS	n.a.	[69]
	Performance of model predictive control in an ethylbenzene production process, comparing fully-connected and partially-connected recurrent neural network models	Aspen Plus Dynamic, MATLAB, Python	Message passing interface	[71]
	Optimization of the SMR natural gas liquefaction process with Deep Q-Network	UniSim Design, MATLAB, Python	n.a.	[72]
	Controlling nitrogen-oxide emissions in a selective catalytic reduction unit with online dynamic tuning of MPCs	Aspen Dynamics	Aspen Custom Modeler	[73]
Data reconciliation	Online monitoring of catalyst deactivation used for kinetic parameter estimation in C3 hydrogenation system	Aspen Plus	Microsoft Excel VBA	[11]
	Automatic detection of steady-states in oil well models for daily optimization	Marlim simulation model	n.a.	[76]
	Emission control and detection in geothermal power plants	UniSim Design	n.a.	[58]

2.5. Future Challenges of Multi-Software Engineering

There are a variety of engineering problems that multi-software engineering could be applied to, and looking at the last few years of research, the future challenges of the field can be identified. In this regard, the available literature is rather sparse when it comes to equipment sizing and scale-up. It is an aspect that affects operation and controllability, not to mention costs. Performing equipment scale-up is not an obvious operation, but a rather complex problem, that could utilize multi-software modeling, taking advantage of the ready-made unit setups that process simulators provide and the customization that can be achieved with numerical tools.

Sustainable practices are more and more desired in the industry, which means that the development of technologies based on recycled and/or nature-derived feedstocks is in demand. Dealing with such feedstocks requires a more complex molecule system with unique interactions and reactions to use that current process simulators are not ready to handle yet, as our case study (Section 5) shows.

Another important terminology to mention is the Digital Twin technology that is widely developing in the industry for the main purpose of supporting Operator Training Simulations (OTS). A Digital Twin is a replica of a unit or process that can predict design failures if it is applied in early planning. Additionally, a dynamic Digital Twin can be

operated parallel to the real process supporting decision-making and intervention, or simulate start-up and shutdown scenarios that are crucial in a process' lifecycle. It is another example where multi-software engineering tools can be beneficial for building a robust dynamic simulator that is capable of handling such models.

The application of Machine Learning and Artificial Intelligence to chemical engineering problems through different software mediums is also more and more studied. Recently, AspenTech released a new version of the simulation software Aspen Plus V12, which includes basic neural network features. Incorporating data-driven methods inside commercial process simulation environments could significantly reduce computational time.

Although more and more vendors try to accommodate these needs, by releasing new features to their software, these commercially available simulators still cannot provide a full solution for modern engineering problems on their own. Custom modifications are often required by skilled engineers who must have extensive knowledge of other software on the market, as well as programming.

In the following sections, we collected the methods and available software for multi-software engineering use to serve as a guide for software connecting opportunities.

3. Linking Types for Multi-Software Applications

Depending on the application and the type of these flowsheet simulators, one can perform different connections between them or with an external tool. The use of middleware or specific built-in blocks is common. The main approaches are ActiveX (Component Object Model (COM) and Object Linking and Embedding (OLE)) and OPC (Open Platform Communication) connections.

ActiveX is considered an outdated framework for new development in software engineering [77] but several process simulators put it to good use for external software linking purposes [78]. It utilizes the earlier Component Object Model (COM) and Object Linking and Embedding (OLE) technologies for software to communicate with each other using scripting and perform data exchange with each other. It is designed for communication on a single hardware or inside a network. The COM server and its scripting is available for most commercial process simulators, such as Aspen Plus or HYSYS.

With OPC [79], one can create direct simulator-to-simulator connections; that way, run time can be lowered and additional data loss can be avoided. However, a mediator tool is advised to be used, if e.g., one would like to connect dynamic simulators, as in this case the simulation clocks of each software run differently.

With some of the commercial process simulators native integration is also possible. One example of this type of connection is gPROMS Object for MATLAB (gO:MATLAB), which allows solving complex mathematical equations by calling a previously built model from gPROMS within a function in MATLAB [80]. Similarly, gO:Run is a convenient tool for non-advanced users, as it makes it possible to execute previously built gPROMS models behind MS Excel VBA interfaces, where their standard functions can also be used.

Another example is Aspen Plus Dynamics (APD) and MATLAB-Simulink, where one can add the Aspen Modeler Block to the Simulink model to design the control system of a unit. When the APD model information is imported into Simulink, input and output signals can be defined with the same variable names and sequence as in the Control Design Interface (CDI) tool [81].

A similar connection possibility exists in Aspen HYSYS, which is called the Extension Container, with which custom objects can be added to the simulation such as Unit Operations, Reaction Kinetics, and Property Packages. Those extensions become part of the simulation and participate in the simulation calculations akin to regular built-in HYSYS objects [20].

Interfacing COMSOL Multiphysics with several external software is also possible via LiveLink [82].

Integrating the MATLAB environment and its toolboxes enables the use of its functions while modeling. It allows you to call MATLAB functions from the COMSOL Desktop and

import/export data from/to the MATLAB workspace. The MATLAB Simulink add-on with the LiveLink connection can be used for control system design and performing co-simulation of COMSOL Multiphysics models and Simulink diagrams. This kind of connection is also possible with MS Excel, where parameters and variables that are defined and modeled in COMSOL Multiphysics can be synchronized to an Excel worksheet creating data libraries and parameter lists, which later can be used to load in the COMSOL Desktop. For 3D designs, SOLIDWORKS and AutoCAD software can also be connected to COMSOL Multiphysics, where the creation of model geometries and the integration of multiphysics simulation are synchronized between the software for design optimization.

4. CAPE-OPEN Standard

For embedding models from external programs, another important tool has to be mentioned and that is CAPE-OPEN. Computer Aided Process Engineering or Computer Applications in Production and Engineering (CAPE) is a modeling approach to process engineering problems that implies computer-aided methods and tools [83]. The CAPE-OPEN project lists a series of specifications that define software interfaces for process simulation environments. The CAPE-OPEN specifications are supported by the non-profit organization CO-LaN [26]; the standard is uniform and available free of charge. Engineers often come across complex modeling problems and the solutions for them lie in different commercially available process simulation software. One can choose to develop a custom model to solve these problems or try to integrate the existing models into their simulator. Either way, it requires advanced programming skills and takes significant amounts of time and effort, although it provides only a temporary solution. The development of new versions in the modeling environments makes it difficult to integrate the new components and the difficulty of writing software interfaces is particular to the host modeling environment.

The standardization of interfaces that CAPE-OPEN provides for software vendors has the advantage of interoperability. One can differentiate between the two main elements of process simulation software, the Process modeling Environment (PME) and the Process modeling Component (PMC). The PME means a simulation program with a buildable process flowsheet and working thermodynamics. A PME equipped with a CAPE-OPEN plug allows any CAPE-OPEN PMC, such as unit operations or thermodynamic models, to be placed in the modeling environment without further interfacing needed. That means any modeling environment with a CAPE-OPEN interface from any vendor would be able to communicate with any CAPE-OPEN modeling component without the need for a communication channel or additional programming involved.

To allow such accessibility between several users, the CAPE-OPEN project classified standards for Business and Common interfaces as seen in Figure 2. The former are vertical interface specifications, which are domain-specific and contain the main PMCs of a process simulation environment, such as flowsheet monitoring, unit operations, numerical solvers, thermodynamics, and physical properties. The common interface specifications are horizontal and contain basic functions such as utilities, error handling, or identification. They are general-purpose interfaces independent of business interfaces that support basic functions fundamental for CAPE-OPEN development. Another important part of the CAPE-OPEN structure is the communication platform, which allows interoperability between the software components PME and PMC. The object-oriented middleware technology that each object uses to communicate is Microsoft's COM (Component Object Model) or the Object Management Group's (OMG) CORBA (Common Object Request Broker Architecture); these models are described in the implementation specifications.

Figure 3 shows the classification of the assorted PMEs and PMCs that implement the CAPE-OPEN standard. The diagram also helps to distinguish between dynamic and steady-state simulation software. In addition, besides the commercial process simulators, freeware and openware were highlighted. The various in-house thermodynamics engines of bigger commercial vendors were excluded from this figure.

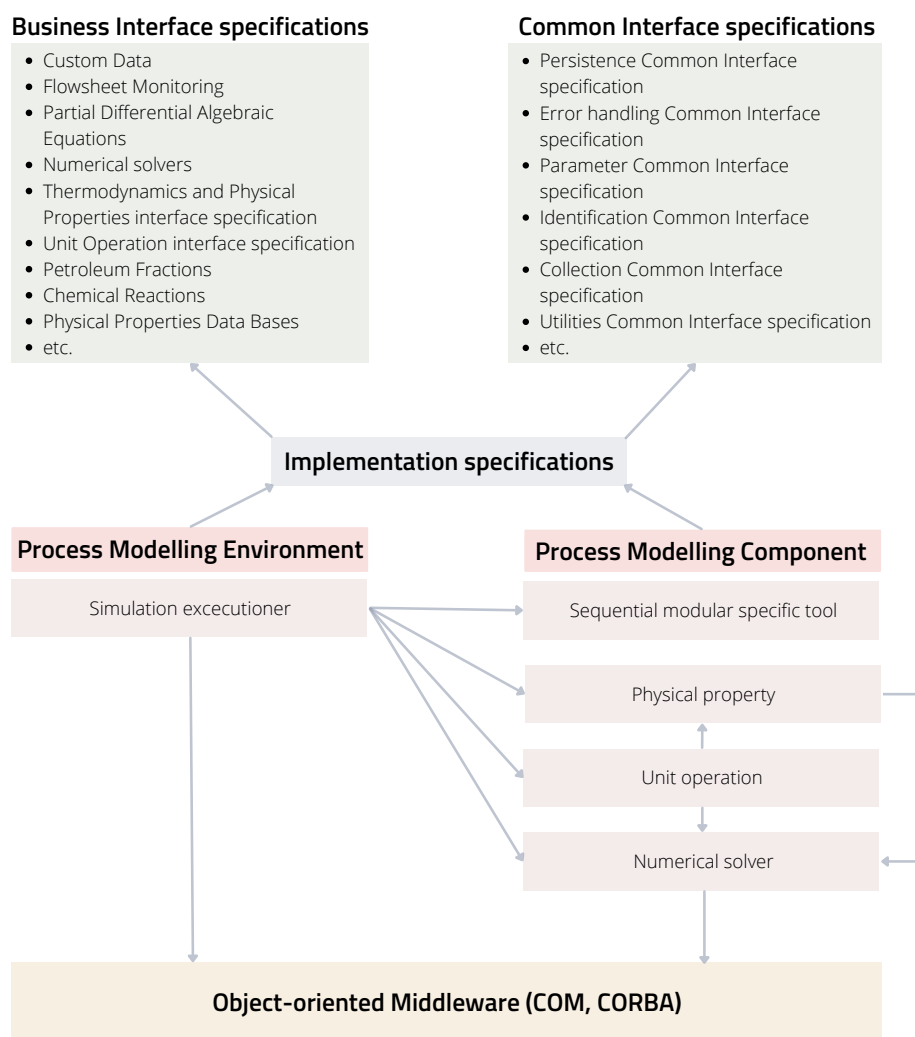


Figure 2. CAPE-OPEN architecture elements and relations (COM: Component Object Model, CORBA: Common Object Request Broker Architecture).

CAPE-OPEN is implemented in programming languages with an open source license, such as SciLab—developed for engineers and scientists—and also one of the most widely used languages in any coding-required field—Python. It is also available in openware such as DWSIM and freeware such as COCO simulator and MOSAIC modeling.

MOSAIC modeling is based on a LaTeX-style entry method for building custom equation systems to use for optimization, steady-state, and dynamic simulation purposes [8,43,84]. It uses symbolic notation for automatic code generation for specific platforms (e.g., C++, FORTRAN, Python, SciLab, MATLAB, gPROMS, Aspen Custom Modeler, GAMS, AMPL, Modelica) that support reusability and portability. It is not designed as a full software; the web-based environment and the available database of models complement solutions for various chemical engineering applications [85].

Another openware is DWSIM, which is a process simulator available on local computers or recently as a web application. A commercial web-based version of DWSIM Pro is also available [21]. It includes solvers for both dynamic and steady-state simulations and various unit operations in both cases. It contains advanced property packages and thermodynamic models, just as commercial simulators, and has the option for Excel and Python plugins interface for thermodynamic calculations.

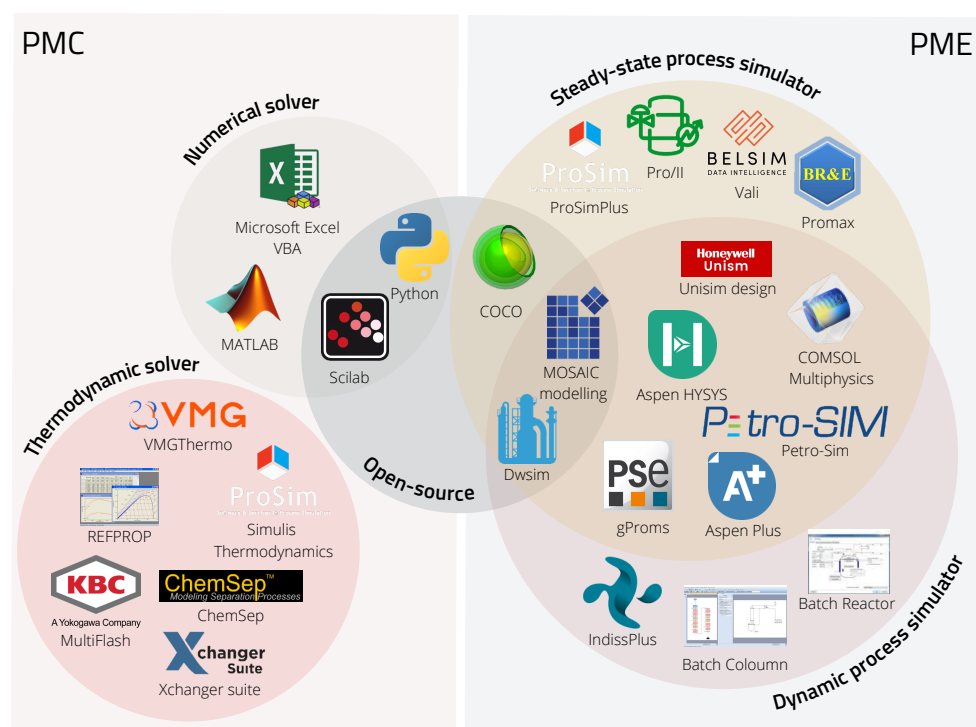


Figure 3. Current process modeling environments with CAPE-OPEN standard specifications implemented.

COCO (CAPE-OPEN to CAPE-OPEN) is a freeware simulation environment that includes a graphical user interface for flowsheeting (COFE—CAPE-OPEN Flowsheet Environment), thermodynamic library and property calculation methods (TEA—Thermodynamics for Engineering Applications), a set of unit operations (COUSCOUS—CAPE-OPEN Unit-operations Simple), and a kinetic and equilibrium reaction package manager (CORN—CAPE-OPEN Reaction Numerics) [86].

The flowsheet simulator COFE is steady-state and belongs in the category of sequential modular flowsheeting engines, which means the sub-models, such as thermodynamic and unit operation models, are considered black boxes and are solved by an iterative function [87]. COFE comes with an Excel template add-in, that deals with the documentation in an Excel workbook. It allows access to all streams and unit operations data, and thermodynamic property calculations can be performed too.

Unlike any other simulation environment, COCO was developed from the beginning as CAPE-OPEN-compliant; it is modeled around CAPE-OPEN interfaces. That means COCO does not use built-in sub-models that use the flowsheet engine's proprietary interface, nor does it provide the option of user-defined models. Its components (TEA, COUSCOUS, COFE) are exchangeable to any other CAPE-OPEN component and work exclusively with such compliant models; additionally, any other third-party CAPE-OPEN compliant PME can utilize COCO's PMCs.

Due to COCO's unique structure and open access, it is often used for interoperability testing against third-party products, which helps with constant development and error detection, as well as the improvement of the CAPE-OPEN standard.

5. Case Study

The original task that inspired this case study was to build a dynamic model of a hydrogenation reactor based on the already existing data of lab-scale experiments from the work of Capecci et al. [88], to further investigate the process. As a process simulation environment that is capable of dynamic simulation, Aspen HYSYS was available to us.

The original kinetic experiments were carried out in a batch reactor under isotherm and isobaric conditions, where non-competitive Langmuir–Hinshelwood kinetics with no dissociation of hydrogen were considered. The described rate expression model is a complex equation [88] and the software is lacking some of the main correlations needed to precisely incorporate all the known data.

That is a common problem engineers and researchers have to face, when some of the key elements are missing to solve even more simple problems with the available software. Therefore, we needed to formulate new kinetics, based on the previous kinetic study, that could be used in the HYSYS environment and perform identification of the already available experimental data. In our case, an external numerical solver (MATLAB) was used to re-identify the reaction kinetics implemented in HYSYS. A co-simulation setup was chosen, as the integration of the two software give us the ability to run several simulation models, perform parameter identification, and validate the data at the same time from a main program, and that way obtain results in one go and organize the data in one place.

5.1. Simulator Development of the Case Study

This case study aimed to show the identification of kinetic parameters with software linking for the production of γ -valerolactone (GVL) from butyl-levulinate (BL) in a two-step hetero-catalytic reaction over Ru/C catalyst.

Figure 4 presents the reaction network of GVL production; the first reaction was a catalytic hydrogenation reaction leading to the intermediate butyl-4-hydroxypentanoate (BHP). The second reaction was the cyclization of this intermediate, leading to the product γ -valerolactone and butanol (BuOH). For the second cyclization step, both non-catalytic and catalytic pathways were possible, meaning the ring closure reaction happened on its own, but in the presence of the catalyst, it was faster. γ -valerolactone was also used as a solvent, as the reaction is considered irreversible in solvent GVL. The hydrogenation is an exothermic reaction with an enthalpy of -38.6 kJ/mole. The catalytic cyclization reaction took place in the liquid phase on a solid catalyst and was endothermic; the enthalpy was $+6.5$ kJ/mole [89]. This cyclization reaction was the rate-determining step.

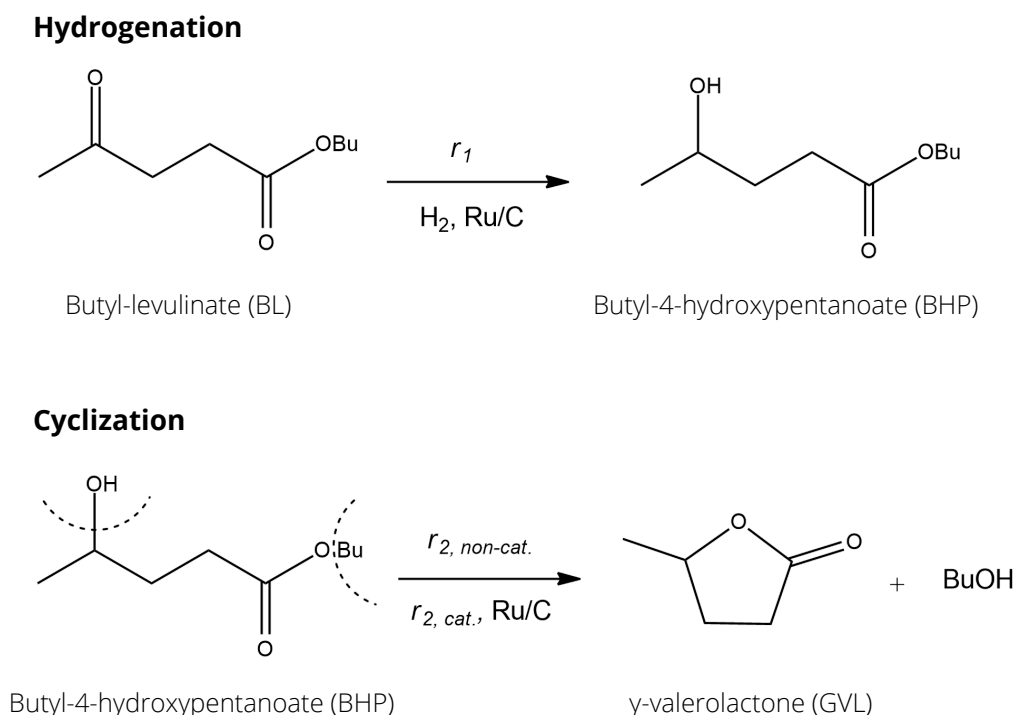


Figure 4. The two-step catalytic hydrogenation reaction of butyl-levulinate to produce γ -valerolactone.

When generating the simulation, kinetics were simplified to Arrhenius type for the three reactions present in the system. Equations (1)–(3) are the new kinetics created to describe the reaction system of the process. Equation (1) represents the rate expression for the catalytic hydrogenation, and Equations (2) and (3) are the reaction rates for the homogeneous catalytic and non-catalytic ring closure reaction, respectively, where k_0 is the pre-exponential factor, E_A is the activation energy in $\left[\frac{\text{J}}{\text{mol}}\right]$, R is the ideal gas constant in $\left[\frac{\text{JK}}{\text{mol}}\right]$, T is the temperature in [K], c is the concentration of the components in $\left[\frac{\text{mol}}{\text{m}^3}\right]$ and $\omega_{cat.}$ is the catalyst loading in $\left[\frac{\text{kg}}{\text{m}^3}\right]$.

$$r_1 = k_{01} \exp\left(-\frac{E_{A1}}{RT}\right) c_{BL} c_{H_2} \omega_{cat.} \quad (1)$$

$$r_{2,cat.} = k_{02,cat.} \exp\left(-\frac{E_{A2,cat.}}{RT}\right) c_{BHP} \omega_{cat.} \quad (2)$$

$$r_{2,ncat.} = k_{02,ncat.} \exp\left(-\frac{E_{A2,ncat.}}{RT}\right) c_{BHP}. \quad (3)$$

The detailed description of the characteristics of the components as well as the setup used for the experiments are written in the works of Capecci and Wang et al. [88–90]. Based on this, four series of experimental data were available for use, where the experiments were conducted under different conditions and the concentration change in time was measured for BL and intermediate BHP. Initial ratios of the mixtures, catalyst loading, temperature, and pressure values were collected in Table 5.

Experiment 2 is different, as there was no catalyst present during the experiment and the intermediate was the starting material to investigate only the non-catalytic ring closure reaction. We chose to include this experiment to improve the fitting of the second ring closure reaction during the identification.

Table 5. Parameters of the experiments used for identification.

Exp.	p_{H_2} [bar]	T [K]	m_{cat} [kg] (50 %wt moisture)	[BL] ₀ [mol/m ³]	[GVL] ₀ [mol/m ³]	[BHP] ₀ [mol/m ³]	[BuOH] ₀ [mol/m ³]
1	22.30	427.15	0.0028	619	8827	0	0
2	23.3	423.15	0	59	6884	1586	189
3	16.30	373.15	0.0005	1893	6719	0	0
4	5.20	393.15	0.0010	1885	6720	0	0

Four dynamic HYSYS simulation files were built according to the experimental setup in the aforementioned article, to be simulated in the HYSYS using the Soave–Redlich–Kwong (SRK) thermodynamic model. Arrhenius kinetics were considered for each reaction as mentioned before, where the pre-exponential factors and the activation energies could be reached via an internal HYSYS spreadsheet. In each case, the input parameters were varied based on the selected experiments in Table 5. Figure 5 shows the unit design that contains a batch reactor equipped with a temperature controller to operate in an isotherm mode. Isobaric conditions were obtained with a high-pressure H_2 inlet stream, similar to the experimental conditions.

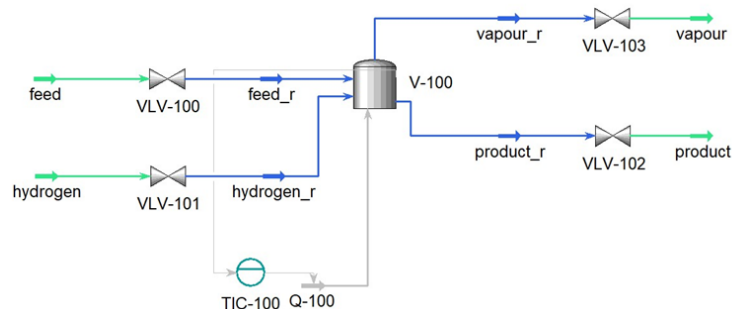


Figure 5. Process flowsheet diagram of the batch reactor built in Aspen HYSYS.

5.2. Multi-Software Based Identification Framework for Kinetic Parameter Identification

The kinetic parameters were identified by linking the HYSYS simulator to the MATLAB environment, where the latter acted as the main program. The connection was made using COM technology through the 'actxserver'. The previously built HYSYS models of the experimental cases were run from MATLAB, where modification to the kinetic parameters was made directly through HYSYS spreadsheets. The schematic identification framework is represented in Figure 6.

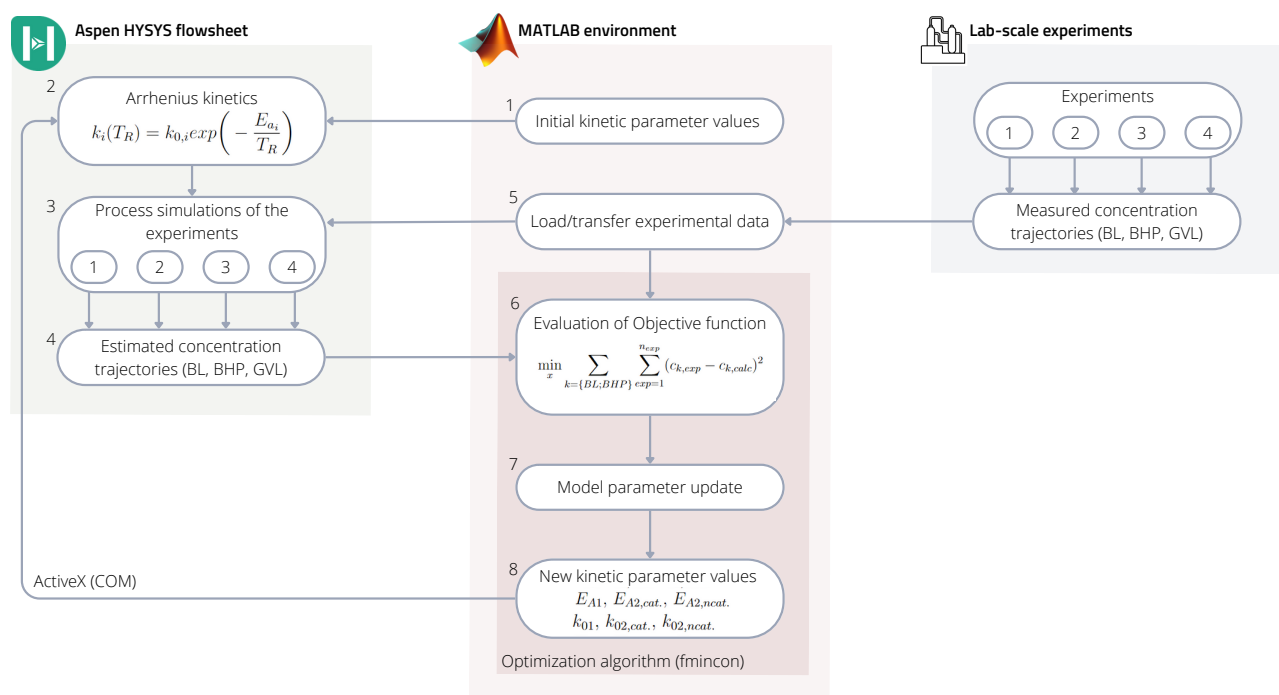


Figure 6. Outline of the identification framework connecting Aspen HYSYS and MATLAB.

The identification framework within the MATLAB environment calculated the kinetic parameters of the HYSYS files, where their value was varied in each iteration step to achieve better fitting to the experimental data sets in each case.

Based on Figure 6 the framework works in the following way:

- (1) The starting point of the sequence is running the MATLAB program, where, as a first step, it sends initial k and E_A values to a previously set up internal HYSYS spreadsheet.
- (2) Values overwritten in the spreadsheet modify the described Arrhenius kinetics in the HYSYS reaction sets.
- (3) Then all four of the HYSYS simulations run with the new kinetic values until a set time (reaction time).

- (4) The simulation results; concentration trajectories of the raw material (BL), intermediate (BHP), and product (BL) are transferred to another internal HYSYS spreadsheet.
- (5) In the meantime, MATLAB reads and organizes the experimental data from text files.
- (6) MATLAB will then import the concentration trajectories from the HYSYS spreadsheets and evaluates the objective function including the estimated concentration trajectories from all four simulations and all four measured concentration values from the experiments.
- (7–8) Based on the normalized error values, in the next iteration step MATLAB will find new kinetic parameter values and send them to HYSYS to run the simulation again.

Equation (4) represents the objective function, which is a minimum search with a quadratic error function including all data sets. To find the optimal point of the objective function, MATLAB's nonlinear multivariable solver 'fmincon' was used.

$$\min_x \sum_{k=\{BL;BHP\}} \sum_{exp=1}^{n_{exp}} (c_{k,exp} - c_{k,calc})^2, \quad (4)$$

where c_k is the concentration of the components BL and BHP in $\left[\frac{\text{mol}}{\text{m}^3}\right]$, indexed exp for the measured data, meaning the experiments 1–4 in Table 5 and $calc$ for the fitted data set from the HYSYS simulations.

5.3. Results and Discussion

The results of the kinetic parameter identification for the process described in Section 5.2 are shown in Figure 7. It can be seen that the calculated concentration trajectories fit the measured data well. The identified kinetic parameters are in Table 6.

Table 6. Results of the kinetic parameter identification.

Pre-exponential factors		
k_{01}	$4.37 \cdot 10^{-03}$	$(\text{m}^3)^3 \text{kg}^{-1} \text{mol}^{-1} \text{s}^{-1}$
$k_{02,cat.}$	$1.01 \cdot 10^{-06}$	$J (\text{m}^3)^2 \text{kg}^{-1} \text{s}^{-1}$
$k_{02,ncat.}$	$1.94 \cdot 10^{-03}$	$\text{m}^3 \text{s}^{-1}$
Activation energies		
E_{A1}	12 304	J mole^{-1}
$E_{A2,cat.}$	0	J mole^{-1}
$E_{A2,ncat.}$	14 490	J mole^{-1}

To compare the performance of our simple kinetic model with Capecci's work, the R^2 of their fitting to the concentration trajectories is 0.976 while ours is 0.952. The predictive ability of our simpler kinetic model is comparable with Capecci's model based on the R^2 indicator.

The program was run with Aspen HYSYS V11 and MATLAB 2021b on an Intel(R) Xeon(R) W-2245 CPU 3.90 GHz, 32 GB; the computation time was 11 h and 15 min. However, it has to be mentioned that the computation time could be lowered with parallel computing.

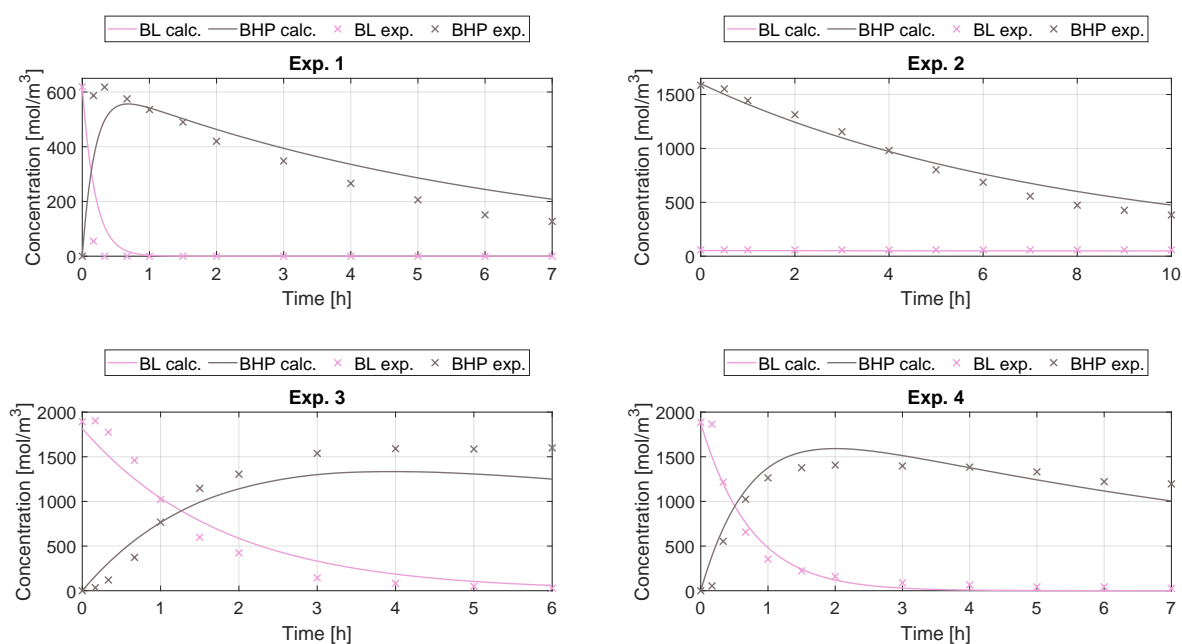


Figure 7. The concentration curve fittings resulting from the identification.

6. Conclusions

In this work, the various applications of multi-software engineering regarding the chemical industry have been highlighted through adequate literature. We organized this prominent research into four categories—process modeling, process design and optimization, process control and safety, and data-driven methods. Custom unit operation modeling and process optimization were found to be the most utilized aspects. The integration methodologies that could link together multiple process simulation software and coding environments were also collected and studied in detail. The CAPE-OPEN standard was also showcased in detail, with a focus on presenting open-source software. Freeware and openware are important parts of the developing process, because of their wide accessibility.

In a case study, parameter identification with multi-software engineering was performed to illustrate one of the many application aspects of the technique. The problem was chosen to showcase the reality when the available process simulator is missing an important element for problem-solving. In our case, it was the lack of kinetics definition, which is crucial for model building inside a simulator. Therefore, a link between the dynamic process simulator Aspen HYSYS and the coding environment MATLAB was made utilizing an ActiveX connection. We found that the method provided good results and performed well in fitting the available experimental data.

Commercial process simulator vendors have to keep up with the rapidly developing information technology and the demand for modeling more complex processes. In the sense of connecting data-driven methods to this software, as for now, two main routes were shown—the possibility of native integration, specific to software, with customizable toolboxes, or improving the plug-ins to accommodate outside environments.

Author Contributions: Conceptualization, A.K. and A.E.; methodology, V.F.C.; software, V.F.C. and A.K.; validation, V.F.C. and A.K.; resources, S.L.; investigation, V.F.C., A.E., S.L. and A.K.; writing—original draft preparation, V.F.C.; writing—review and editing, V.F.C., A.E., S.L. and A.K.; visualization, V.F.C.; supervision, A.E., S.L. and A.K.; funding acquisition, A.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Research, Development and Innovation Fund of Hungary grant number 2019-2.1.11-TÉT-2019-00005.

Data Availability Statement: Data available on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 35 Years of Innovation. Available online: <https://www.aspentech.com/en/about-aspentech/35-years-of-innovation/> (accessed on 19 May 2022).
- Sargent, R. Advances in modelling and analysis of chemical process systems. *Comput. Chem. Eng.* **1983**, *7*, 219–237. [[CrossRef](#)]
- Complete List of Process Simulators. 2017. Available online: <https://simulatelive.com/product-reviews/simulation/complete-list-of-process-simulators-part-1-2> (accessed on 19 May 2022).
- Pistikopoulos, E.N.; Barbosa-Povoa, A.; Lee, J.H.; Misener, R.; Mitsos, A.; Reklaitis, G.V.; Venkatasubramanian, V.; You, F.; Gani, R. Process systems engineering—The generation next? *Comput. Chem. Eng.* **2021**, *147*, 107252. [[CrossRef](#)]
- Tian, Y.; Demirel, S.E.; Hasan, M.F.; Pistikopoulos, E.N. An overview of process systems engineering approaches for process intensification: State of the art. *Chem. Eng. Process.-Process Intensif.* **2018**, *133*, 160–210. [[CrossRef](#)]
- Gani, R. Chemical product design: Challenges and opportunities. *Comput. Chem. Eng.* **2004**, *28*, 2441–2457. [[CrossRef](#)]
- Fontalvo, J.; Cuellar, P.; Timmer, J.; Vorstman, M.; Wijers, J.; Keurentjes, J. Comparing pervaporation and vapor permeation hybrid distillation processes. *Ind. Eng. Chem. Res.* **2005**, *44*, 5259–5266. [[CrossRef](#)]
- Tolksdorf, G.; Esche, E.; Wozny, G.; Repke, J.U. Customized code generation based on user specifications for simulation and optimization. *Comput. Chem. Eng.* **2019**, *121*, 670–684. [[CrossRef](#)]
- Claumann, C.A.; Peruzzo, T.; Felice, V.D.; Marangoni, C.; Machado, R. Modeling and process optimization: An approach using aspen plus and matlab in the energy integration study of distillation columns. In Proceedings of the VIII Congreso Argentino de Ingeniería Química Y, Buenos Aires, Argentina, 2–5 August 2015; Volume 3.
- Furda, P.; Variny, M.; Labovská, Z.; Cibulka, T. Process drive sizing methodology and multi-level modeling linking MATLAB® and Aspen Plus® environment. *Processes* **2020**, *8*, 1495. [[CrossRef](#)]
- Farsang, B.; Németh, S.; Abonyi, J. Online monitoring of catalyst deactivation based on data reconciliation and flowsheeting simulator. *Period. Polytech. Chem. Eng.* **2015**, *59*, 145–150. [[CrossRef](#)]
- Bayoumy, S.H.; El-Marsafy, S.M.; Ahmed, T.S. Optimization of a saturated gas plant: Meticulous simulation-based optimization—A case study. *J. Adv. Res.* **2020**, *22*, 21–33. [[CrossRef](#)]
- Zhu, W.; Ye, H.; Yang, Y.; Zou, X.; Dong, H. Simulation-Based Optimization of a Multiple Gas Feed Sweetening Process. *ACS Omega* **2022**, *7*, 2690–2705. [[CrossRef](#)]
- Shirmohammadi, R.; Soltanieh, M.; Romeo, L.M. Thermo-economic analysis and optimization of post-combustion CO₂ recovery unit utilizing absorption refrigeration system for a natural-gas-fired power plant. *Environ. Prog. Sustain. Energy* **2018**, *37*, 1075–1084. [[CrossRef](#)]
- Alabdulkarem, A.; Mortazavi, A.; Hwang, Y.; Radermacher, R.; Rogers, P. Optimization of propane pre-cooled mixed refrigerant LNG plant. *Appl. Therm. Eng.* **2011**, *31*, 1091–1098. [[CrossRef](#)]
- Brambilla, A.; Vaccari, M.; Pannocchia, G. Analytical RTO for a critical distillation process based on offline rigorous simulation. *IFAC-PapersOnLine* **2022**, *55*, 143–148. [[CrossRef](#)]
- Tuan, T.T.; Tufa, L.D.; Mutalib, M.I.A.; Abdallah, A.F.M. Control of depropanizer in dynamic Hysys simulation using MPC in Matlab-Simulink. *Procedia Eng.* **2016**, *148*, 1104–1111. [[CrossRef](#)]
- Ryu, H.; Lee, J.M. Model predictive control (mpc)-based supervisory control and design of off-gas recovery plant with periodic disturbances from parallel batch reactors. *Ind. Eng. Chem. Res.* **2016**, *55*, 3013–3025. [[CrossRef](#)]
- Dimian, A.C.; Bildea, C.S.; Kiss, A.A. Introduction in Process Simulation. In *Computer Aided Chemical Engineering*; Elsevier: Amsterdam, The Netherlands, 2014; Volume 35, pp. 35–71.
- Aspen HYSYS Customization Guide; Aspen Technology, Inc.: Bedford, MA, USA, 2011; V7.3. Available online: www.aspentech.com (accessed on 19 May 2022).
- DWSIM—The Open Source Chemical Process Simulator. 2022. Available online: <https://dwsim.org/wiki/index.php?title=Literature> (accessed on 19 May 2022).
- Rangaiah, G.P.; Feng, Z.; Hoadley, A.F. Multi-objective optimization applications in chemical process engineering: Tutorial and review. *Processes* **2020**, *8*, 508. [[CrossRef](#)]
- Zhuang, J.; Tang, J.; Aljerf, L. Comprehensive review on mechanism analysis and numerical simulation of municipal solid waste incineration process based on mechanical grate. *Fuel* **2022**, *320*, 123826. [[CrossRef](#)]
- Bartolome, P.S.; Van Gerven, T. A comparative study on Aspen Hysys interconnection methodologies. *Comput. Chem. Eng.* **2022**, *162*, 107785. [[CrossRef](#)]
- Abril, A.F. Aspen Plus-Matlab Link. MATLAB Central File Exchange. 2020. Available online: www.mathworks.com/matlabcentral/fileexchange/69464-aspen-plus-matlab-link (accessed on 19 May 2022).
- CO-LaN. Available online: www.colan.org/general-information-on-co-lan/ (accessed on 19 May 2022).
- Van Baten, J.; Pons, M. CAPE-OPEN: Interoperability in Industrial Flowsheet Simulation Software. *Chem. Ing. Tech.* **2014**, *86*, 1052–1064. [[CrossRef](#)]
- Gani, R. Computer-aided methods and tools for chemical product design. *Chem. Eng. Res. Des.* **2004**, *82*, 1494–1504. [[CrossRef](#)]
- Gani, R. Group contribution-based property estimation methods: Advances and perspectives. *Curr. Opin. Chem. Eng.* **2019**, *23*, 184–196. [[CrossRef](#)]
- Alshehri, A.S.; Tula, A.K.; You, F.; Gani, R. Next generation pure component property estimation models: With and without machine learning techniques. *AIChE J.* **2022**, *68*, e17469. [[CrossRef](#)]

31. Alshehri, A.S.; Gani, R.; You, F. Deep learning and knowledge-based methods for computer-aided molecular design—Toward a unified approach: State-of-the-art and future directions. *Comput. Chem. Eng.* **2020**, *141*, 107005. [[CrossRef](#)]
32. Scopus[®]. Available online: www.scopus.com/ (accessed on 19 May 2022).
33. Di Bucchianico, D.D.M.; Wang, Y.; Buvat, J.C.; Pan, Y.; Moreno, V.C.; Leveneur, S. Production of levulinic acid and alkyl levulinates: a process insight. *Green Chem.* **2022**, *24*, 614–646. [[CrossRef](#)]
34. Fontalvo, J. Using user models in Matlab[®] within the Aspen Plus[®] interface with an Excel[®] link. *Ing. Investig.* **2014**, *34*, 39–43. [[CrossRef](#)]
35. Kiss, A.A.; Bildea, C.S.; Grievink, J. Dynamic modeling and process optimization of an industrial sulfuric acid plant. *Chem. Eng. J.* **2010**, *158*, 241–249. [[CrossRef](#)]
36. Samei, M.; Raisi, A. Multi-stage gas separation process for separation of carbon dioxide from methane: Modeling, simulation, and economic analysis. *Chem. Eng. Process.-Process Intensif.* **2022**, *170*, 108676. [[CrossRef](#)]
37. Mikkonen, H.; Jegoroff, M.; Lappalainen, J.; Savolainen, J. Modelling and dynamic co-simulation studies of oxy-fired power plant. In Proceedings of the Automaatio XXI, Finnish Society of Automation, Helsinki, Finland, 17–18 March 2015.
38. Sousa, A.M.; Matos, H.A.; Pereira, M.J. Modelling paraffin wax deposition using Aspen HYSYS and MATLAB. In *Computer Aided Chemical Engineering*; Elsevier: Amsterdam, The Netherlands, 2019; Volume 46, pp. 973–978.
39. Da Rosa, C.A.; Braatz, R.D. openCrys: Open-source software for the multiscale modeling of combined antisolvent and cooling crystallization in turbulent flow. *Ind. Eng. Chem. Res.* **2018**, *57*, 11702–11711. [[CrossRef](#)]
40. Egedy, A.; Kummer, A.; Leveneur, S.; Varga, T.; Chován, T. Cfd modeling of spatial inhomogeneities in a vegetable oil carbonation reactor. *Processes* **2020**, *8*, 1356. [[CrossRef](#)]
41. Wutz, J.; Waterkotte, B.; Heitmann, K.; Wucherpfennig, T. Computational fluid dynamics (CFD) as a tool for industrial UF/DF tank optimization. *Biochem. Eng. J.* **2020**, *160*, 107617. [[CrossRef](#)]
42. Morales-Rodríguez, R.; Gani, R.; Déchelotte, S.; Vacher, A.; Baudouin, O. Use of CAPE-OPEN standards in the interoperability between modelling tools (MoT) and process simulators (Simulis[®] Thermodynamics and ProSimPlus). *Chem. Eng. Res. Des.* **2008**, *86*, 823–833. [[CrossRef](#)]
43. Kuntsche, S.; Barz, T.; Kraus, R.; Arellano-Garcia, H.; Wozny, G. MOSAIC a web-based modeling environment for code generation. *Comput. Chem. Eng.* **2011**, *35*, 2257–2273. [[CrossRef](#)]
44. Maha, B.; Ali, S.; Ammar, B.B. Modeling and simulation of multi-effect desalination plant (SIDEM unit). In Proceedings of the 2017 International Conference on Green Energy Conversion Systems (GECS), IEEE, Hammamet, Tunisia, 23–25 March 2017; pp. 1–6.
45. Ledezma-Martinez, M.; Jobson, M.; Smith, R. Simulation-Optimization-Based Design of Crude Oil Distillation Systems with Preflash Units. *Ind. Eng. Chem. Res.* **2018**, *57*, 9821–9830. [[CrossRef](#)]
46. Dutta, A.; Karimi, I.A.; Farooq, S. Heating value reduction of LNG (liquefied natural gas) by recovering heavy hydrocarbons: techno-economic analyses using simulation-based optimization. *Ind. Eng. Chem. Res.* **2018**, *57*, 5924–5932. [[CrossRef](#)]
47. Hamed, H.; Karimi, I.A.; Gundersen, T. Optimization of helium extraction processes integrated with nitrogen removal units: A comparative study. *Comput. Chem. Eng.* **2019**, *121*, 354–366. [[CrossRef](#)]
48. Christopher, C.C.E.; Dutta, A.; Farooq, S.; Karimi, I.A. Process synthesis and optimization of propylene/propane separation using vapor recompression and self-heat recuperation. *Ind. Eng. Chem. Res.* **2017**, *56*, 14557–14564. [[CrossRef](#)]
49. Li, X.; Cui, C.; Li, H.; Gao, X. Process synthesis and simulation-based optimization of ethylbenzene/styrene separation using double-effect heat integration and self-heat recuperation technology: A techno-economic analysis. *Sep. Purif. Technol.* **2019**, *228*, 115760. [[CrossRef](#)]
50. Ruiz-Femenia, R.; Fernández-Torres, M.J.; Salcedo-Díaz, R.; Gomez-Rico, M.F.; Caballero, J.A. Systematic tools for the conceptual design of inherently safer chemical processes. *Ind. Eng. Chem. Res.* **2017**, *56*, 7301–7313. [[CrossRef](#)]
51. Oh, S.Y.; Binns, M.; Cho, H.; Kim, J.K. Energy minimization of MEA-based CO₂ capture process. *Appl. Energy* **2016**, *169*, 353–362. [[CrossRef](#)]
52. Rahman, R.K.; Ibrahim, S.; Raj, A. Multi-objective optimization of sulfur recovery units using a detailed reaction mechanism to reduce energy consumption and destruct feed contaminants. *Comput. Chem. Eng.* **2019**, *128*, 21–34. [[CrossRef](#)]
53. Song, R.; Cui, M.; Liu, J. Single and multiple objective optimization of a natural gas liquefaction process. *Energy* **2017**, *124*, 19–28. [[CrossRef](#)]
54. Zhou, L.; Liao, Z.; Wang, L.; Zhang, L.; Ji, X.; Jiao, H.; Wang, J.; Yang, Y.; Dang, Y. Simulation-Based Multiobjective Optimization of the Product Separation Process within an MTP Plant. *Ind. Eng. Chem. Res.* **2019**, *58*, 12166–12178. [[CrossRef](#)]
55. Mounaam, A.; Chhiti, Y.; Souissi, A.; Salouhi, M.; Harmen, Y.; Khouakhi, M.E. Simulation and Optimization of an Industrial Sulfuric Acid Plant with Contact Process Using Python-Unisim Design. In *International Conference on Simulation and Modeling Methodologies, Technologies and Applications, Proceedings of Simulation and Modeling Methodologies, Technologies and Applications, the 10th International Conference, SIMULTECH 2020 Lieusaint, Paris, France, 8–10 July 2020*; Springer: New York, NY, USA, 2020; pp. 64–89.
56. Radó-Fóty, N.; Egedy, A.; Nagy, L.; Hegedűs, I. Investigation and Optimisation of the Steady-State Model of a Coke Oven Gas Purification Process. *Energies* **2022**, *15*, 4548. [[CrossRef](#)]
57. Vaccari, M.; Pannocchia, G.; Tognotti, L.; Paci, M.; Bonciani, R. A rigorous simulation model of geothermal power plants for emission control. *Appl. Energy* **2020**, *263*, 114563. [[CrossRef](#)]

58. Vaccari, M.; Pannocchia, G.; Tognotti, L.; Paci, M. Rigorous simulation of geothermal power plants to evaluate environmental performance of alternative configurations. *Renew. Energy* **2023**, *207*, 471–483. [CrossRef]
59. Briones-Ramírez, A.; Gutiérrez-Antonio, C. Multiobjective Optimization of Chemical Processes with Complete Models using MATLAB and Aspen Plus. *Comput. Syst.* **2018**, *22*, 1157–1170. [CrossRef]
60. Marton, S.; Svensson, E.; Subiaco, R.; Bengtsson, F.; Harvey, S. A steam utility network model for the evaluation of heat integration retrofits—A case study of an oil refinery. *J. Sustain. Dev. Energy Water Environ. Syst.* **2017**, *5*, 560–578. [CrossRef]
61. Sun, W.; Zhao, Y.; Wang, Y. Electro-or turbo-driven?—Analysis of different blast processes of blast furnace. *Processes* **2016**, *4*, 28. [CrossRef]
62. Büttin, H.; Kantor, I.; Maréchal, F. Incorporating location aspects in process integration methodology. *Energies* **2019**, *12*, 3338. [CrossRef]
63. Chinprasit, J.; Panjapornpon, C. Model predictive control of vinyl chloride monomer process by Aspen Plus Dynamics and MATLAB/Simulink co-simulation approach. In *Proceedings of the IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2020; Volume 778, p. 012080.
64. Dos Santos Vidal, S.F.; Schmitz, J.E.; Franco, I.C.; Fileti, A.M.F.; Da Silva, F.V. Fuzzy multivariable control strategy applied to a refrigeration system. *Chem. Prod. Process. Model.* **2017**, *12*, 20160033. [CrossRef]
65. Pannocchia, G.; Brambilla, A. How to use simplified dynamics in model predictive control of superfractionators. *Ind. Eng. Chem. Res.* **2005**, *44*, 2687–2696. [CrossRef]
66. Mizuta, Y.; Sumino, M.; Nakata, H.; Izato, Y.; Miyake, A. Analysis of pressure behavior during runaway reaction with case studies of various depressurization designs. *Process Saf. Environ. Prot.* **2022**, *159*, 96–105. [CrossRef]
67. Kummer, A.; Varga, T. Process simulator assisted framework to support process safety analysis. *J. Loss Prev. Process Ind.* **2019**, *58*, 22–29. [CrossRef]
68. Zhu, W.; Chebeir, J.; Romagnoli, J.A. Operation optimization of a cryogenic NGL recovery unit using deep learning based surrogate modeling. *Comput. Chem. Eng.* **2020**, *137*, 106815. [CrossRef]
69. Muhsin, W.; Zhang, J. Multi-Objective Optimization of a Crude Oil Hydrotreating Process with a Crude Distillation Unit Based on Bootstrap Aggregated Neural Network Models. *Processes* **2022**, *10*, 1438. [CrossRef]
70. Galeazzi, A.; Prifti, K.; Gallo, F.; Manenti, F. A Methodology for The Optimal Surrogate Modelling of Digital Twins Using Machine Learning. In *Computer Aided Chemical Engineering*; Elsevier: Amsterdam, The Netherlands, 2022; Volume 51, pp. 1543–1548.
71. Alhajeri, M.S.; Luo, J.; Wu, Z.; Albalawi, F.; Christofides, P.D. Process structure-based recurrent neural network modeling for predictive control: A comparative study. *Chem. Eng. Res. Des.* **2022**, *179*, 77–89. [CrossRef]
72. Kim, S.; Jang, M.G.; Kim, J.K. Process Design and Optimization of Single Mixed-Refrigerant Processes with the Application of Deep Reinforcement Learning. *Appl. Thermal Eng.* **2023**, *223*, 120038. [CrossRef]
73. Hedrick, E.; Hedrick, K.; Bhattacharyya, D.; Zitney, S.E.; Omell, B. Reinforcement learning for online adaptation of model predictive controllers: Application to a selective catalytic reduction unit. *Comput. Chem. Eng.* **2022**, *160*, 107727. [CrossRef]
74. Farsang, B.; Németh, S.; Abonyi, J. Role of steady state data reconciliation in process model development. *Hung. J. Ind. Chem.* **2013**, *41*, 65–75.
75. Farsanga, B.; Gomorib, Z.; Horvathb, G.; Nagyb, G.; Nemetha, S.; Abonyi, J. Simultaneous validation of online analyzers and process simulators by process data reconciliation. *Chem. Eng.* **2013**, *32*, 1303–1308.
76. Seman, L.O.; Giuliani, C.M.; Camponogara, E.; Müller, E.R.; Vieira, B.F.; Miyatake, L.K.; Medeiros, A.G. Tuning of oil well models with production data reconciliation. *Comput. Chem. Eng.* **2021**, *145*, 107179. [CrossRef]
77. ActiveX Controls on the Internet. 2021. Available online: <https://learn.microsoft.com/en-us/cpp/mfc/activex-controls-on-the-internet?view=msvc-170> (accessed on 19 May 2022).
78. ActiveX Controls on the Internet. Microsoft Docs. 2021. Available online: <docs.microsoft.com/en-us/cpp/mfc/activex-controls-on-the-internet?view=msvc--160> (accessed on 19 May 2022).
79. OPC Foundation. 2022. Available online: <opcfoundation.org/about/what-is-opc/> (accessed on 19 May 2022).
80. The gPROMS Object for MATLAB. Available online: <www.psenterprise.com/products/gproms/objects/gomatlab> (accessed on 19 May 2022).
81. Design and Cosimulate Control of High-Fidelity Distillation Tower with Aspen Plus Dynamics. MathWorks. Available online: <www.mathworks.com> (accessed on 19 May 2022).
82. Interfacing Products with COMSOL Multiphysics® via LiveLink™. Available online: <www.comsol.com/products#modules> (accessed on 19 May 2022).
83. Braunschweig, B.; Gani, R. *Software Architectures and Tools for Computer Aided Process Engineering*; Elsevier: Amsterdam, The Netherlands, 2002.
84. Tolksdorf, G. MOSAIC—A Modeling and Code Generation Tool. 2013. Available online: www.colan.org/wp-content/uploads/2015/09/Y13_COEU_MOSAIC.pdf (accessed on 19 May 2022).
85. MOSAICmodelling. 2022. Available online: <https://mosaic-modeling.de/> (accessed on 19 May 2022).
86. Van Baten, J. An Introduction to Coco. 2007. Available online: www.cocosimulator.org/downloads/AIChE_SaltlakeCity_2007_ExtendedSummary.pdf (accessed on 19 May 2022).
87. Ross Taylor, J.V.B. Flowsheeting with COCO and chemSep. 2010. Available online: www.cocosimulator.org/downloads/FlowsheetingWithCOCOandChemsep_Notes.pdf (accessed on 19 May 2022).

88. Capecci, S.; Wang, Y.; Delgado, J.; Casson Moreno, V.; Mignot, M.; Grénman, H.; Murzin, D.Y.; Leveueur, S. Bayesian Statistics to Elucidate the Kinetics of γ -Valerolactone from n-Butyl Levulinate Hydrogenation over Ru/C. *Ind. Eng. Chem. Res.* **2021**, *60*, 11725–11736. [[CrossRef](#)]
89. Wang, Y.; Plazl, I.; Vernières-Hassimi, L.; Leveueur, S. From calorimetry to thermal risk assessment: γ -Valerolactone production from the hydrogenation of alkyl levulinate. *Process Saf. Environ. Prot.* **2020**, *144*, 32–41. [[CrossRef](#)]
90. Wang, Y.; Cipolletta, M.; Vernières-Hassimi, L.; Casson-Moreno, V.; Leveueur, S. Application of the concept of Linear Free Energy Relationships to the hydrogenation of levulinic acid and its corresponding esters. *Chem. Eng. J.* **2019**, *374*, 822–831. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.