Article

A Modular Framework for the Modelling and Optimization of Advanced Chromatographic Processes

Johannes Schmölder and **Malte Kaspereit** *

Institute of Separation Science & Technology, Friedrich-Alexander University Erlangen-Nürnberg; Egerlandstraße 3, 91052 Erlangen, Germany; johannes.schmoelder@fau.de

* Correspondence: malte.kaspereit@fau.de; Tel.: +49-9131-85-27638

check for updates

Abstract: A framework is introduced for the systematic development of preparative chromatographic processes. It is intended for the optimal design of conventional and advanced concepts that exploit strategies, such as recycling, side streams, bypasses, using single or multiple columns, and combinations thereof. The Python-based platform simplifies the implementation of new processes and design problems by decoupling design tasks into individual modules for modelling, simulation, assertion of cyclic stationarity, product fractionation, and optimization. Interfaces to external libraries provide flexibility regarding the choice of column model, solver, and optimizer. The current implementation, named *CADET-Process*, uses the software *CADET* for solving the model equations. The structure of the framework is discussed and its application for optimal design of existing and identification of new chromatographic operating concepts is demonstrated by case studies.

Keywords: process design; optimization; preparative chromatography; column configuration; *CADET-Process*

1. Introduction

Production processes in chemical, pharmaceutical, or biotechnological industries typically require the separation of products from side products or impurities. Classical separation processes, such as distillation, extraction, or filtration are of limited use when the components are physico-chemically very similar, sensitive to harsh conditions, or part of multi-component separations. In particular for such challenging separations, chromatography is a powerful alternative [1–3].

Most chromatographic separations are performed by the so-called batch chromatography using a single chromatographic column. In conventional elution mode, small amounts of the mixture are injected periodically onto the column and the mixture components elute as separated peaks from its outlet. However, depending on the separation problem at hand, many advanced operating modes exist that can outperform conventional batch chromatography in terms of productivity, solvent consumption, and recovery yield. For example, operating concepts, such as closed-loop recycling [4,5] or steady-state recycling (SSR) [6–8] incorporate different strategies for the recycling of unresolved fractions from the column outlet back to the inlet with the aim of improving yield, solvent consumption and/or productivity [7]. If purity requirements are limited, bypass streams can be advantageous [9]. Moreover, the use of multiple columns gives rise to various concepts ranging from clever series or parallel arrangements of multiple batch columns [10,11], over pseudo-continuous processes, up to the many variants of the powerful continuous simulated moving bed (SMB) concept. Details on such advanced chromatographic operating modes are given, for example, in [2,3,12].

The model-based design and optimization of chromatographic separations is, however, not trivial. This is due to the periodic operation and the distinct non-linear dynamics of chromatographic processes,

as well as the many degrees of freedom involved. For some of the processes there exist simple shortcut methods that deliver rough initial design estimates (see e.g., [13] for batch chromatography, [7,8] for SSR systems, [14] for SMB processes, or [15] for a common treatment). However, rigorous process development requires more detailed models and optimization schemes. Here it should be considered that multiple chromatographic interactions mechanisms can be exploited, such as adsorption, ionic interactions, binding to specific ligands, or size exclusion, to name only a few. Furthermore, depending on the given system, further physical phenomena, such as axial dispersion, film and pore diffusion may have to be accounted for. Consequently, there exists a variety of different modelling approaches in chromatography. For details on this, we refer to [1–3], as well as to the instructive book by Guiochon and Lin [16]. Apart from that, a variety of optimization approaches were also proposed for designing chromatography. An overview is given in Section 2.5. When additionally considering the many advanced operating modes mentioned above, this gives rise to an unmanageable number of specific process models and optimization schemes that may have to be implemented when seeking for an optimal process for a given separation task.

Against this background, a general-purpose tool is needed that allows an efficient and flexible handling of the different subtasks in the development of optimal chromatographic processes. The main requirements are:

- Setting up a model for the desired process structure and the specific chromatographic column(s),
- Solving the model equations for simulating the process,
- Determining process performance by evaluating the outgoing streams/chromatograms,
- Performing optimization of continuous variables, timed events, and potentially process structure.

There are several commercial programs available which provide parts of the aforementioned functionalities, such as Aspen Chromatography [17], ChromX by GoSilico [18], and ChromWorks by YpsoFacto [19], and CADET by the Forschungszentrum Jülich [20]. However, these do not provide for the flexibility required for the ab initio development of novel process concepts.

In this work, we present a new modular framework for the efficient modelling, simulation and optimization of advanced chromatographic processes. The framework decouples the different tasks mentioned above in order to allow for a simple and independent manipulation or exchange of operating concept, modelling depth of the chromatographic column(s), solution of the model equations, performance evaluation, and optimization algorithms. The developed platform performs these tasks in separate corresponding modules, which can be interchanged with other own or third-party modules. The platform is implemented in an object-oriented manner in the programming language Python. The current implementation includes an interface to *CADET* [20], which is an efficient open source simulator for various chromatographic column models, among them the very detailed general rate model. Due to this powerful combination, we denote our framework as *CADET-Process*. The software is open source and distributed under GPL version 3, and thus freely available to academia and industry. The code can be obtained from https://github.com/fau-advanced-separations/CADET-Process.

2. Framework Overview

CADET-Process is written in Python, a free and open-source programming language that gained a lot of popularity in the scientific community in recent years. One of the main advantages of Python is the easy integration of other scientific and numerical packages. This makes it especially useful for a modular approach such as the one presented.

Figure 1 gives a general overview of the program structure and workflow. The *ProcessModel* is an abstract representation of the chromatographic process configuration and the required operational and design parameters. The process is simulated using the *SimulatorAdapter* as an interface to external solver and the resulting *Chromatogram* can be tested for cyclic *Stationarity*. For *ProcessOptimization*, a bi-level problem is formulated to reduce the overall complexity of the optimization problem. On the top level, the variables of the *ProcessModel* are optimized, and on the lower level, the process *Performance* is

determined by calling a *FractionationOptimization* subroutine that optimizes the fractionation times of the simulated *Chromatograms*. In both cases, the *OptimizerAdapter* provides an interface to external *Optimizers*. Below, the different framework modules are described in more detail.
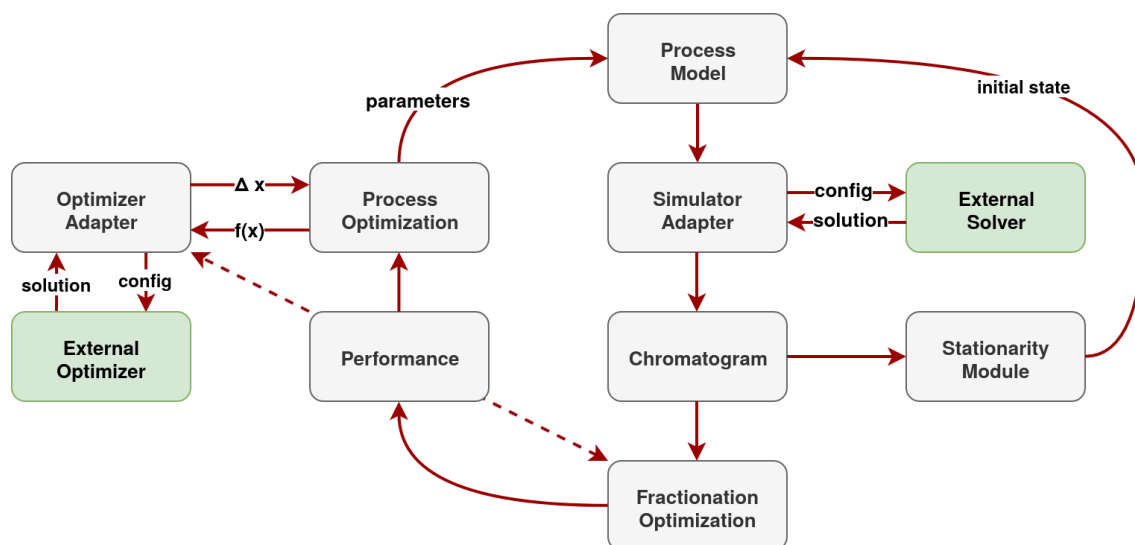

Figure 1. Overview of the framework modules and their relations. For a detailed explanation, see text.

2.1. Process Model

Starting point of process development is the setup of the *ProcessModel* (see top of Figure 1), i.e., the specific configuration of the chromatographic process. This is realized using *UnitOperations* as building blocks. A *UnitOperation* represents the physico-chemical behavior of an apparatus and holds the model parameters. Currently, continuous stirred tank reactors (*CSTRs*), *Columns*, and *TubularReactors* are implemented as *UnitOperations* which can be used to model different parts of a system. In addition, *Sources* and *Sinks* represent the processes' in- and outlets. To model linear gradients, the concentration between *Events* can be interpolated by setting a binary flag in the *Sources* to *True*.

In case of chromatographic *Columns*, different levels of detail can be used for modelling the unit. This provides flexibility regarding the choice of the column model depending on, for example, the purpose of a study and the amount of available information. While there is a large family of chromatographic models, currently the following models are available in *CADET-Process*: the equilibrium model which assumes instantaneous equilibrium and neglects all dispersive effects; the equilibrium-dispersive model which lumps dispersive effects in one apparent dispersion coefficient D_{app}; the transport-dispersive model that accounts for mass transfer limitations; and the general rate model which accounts for external mass transfer, pore diffusion, and a rate-limited binding process. More details on the different models are given in [1–3]. Furthermore, and of obvious relevance for modelling chromatographic columns, *UnitOperations* can be associated with different *BindingModels* that describe the interaction of components with surfaces or chromatographic stationary phases. For this purpose, a variety of equilibrium relations, for example, simple linear adsorption isotherms, competitive forms of the Langmuir and the Bi-Langmuir models, as well as the competitive steric mass action law can be selected.

Importantly, the *ProcessModel* allows connecting multiple *UnitOperations* in a *FlowSheet* to describe more complex operating modes. By this, configurations of several columns or reactor-separator networks can be defined easily. Furthermore, tubing and other external volumes can be modelled using *UnitOperations*. This is made possible by programming the *FlowSheet* as a directed graph that describes the mass transfer between the individual *UnitOperations*. To allow connecting the units correspondingly, every *UnitOperation* can have any number of in- and outputs except for *Sources* that represent streams entering the system, and *Sinks* that represent those exiting. For defining the specific

connections, every *UnitOperation* has an output state that describes the distribution of the outgoing stream to the succeeding *UnitOperations*. Please note that it is straightforward to also include internal recycles in the *FlowSheet*, which is important for systems such as SSR (see example in Section 2.4) or SMB.

Events are used to describe the dynamic operation of the process which is particularly relevant for chromatography. In these processes, dynamic changes occur at the inlets during operational steps such as injection, elution, wash, regeneration or the use of gradients; but also distinct *FlowSheet* actions, such as turning on and off recycles, bypasses, or switching the column configuration have to be considered. Moreover, the fractionation state of products at the outlets of the process changes at specific times (see Section 2.3). By defining *Events*, the state of any attribute of a *UnitOperation* can be changed to a certain value at a given time. To reduce the complexity of the setup, dependencies of *Events* (i.e., conditional events) can be specified that define the time of an *Event* using any linear combination of other *Event* times. Especially for more advanced processes, this reduces the degrees of freedom and facilitates the overall handiness. For this purpose, also *Durations* can be defined to describe the time between the execution of two *Events*. Finally, the cycle time Δt_{cycle} of the *ProcessModel* defines the overall duration of a process after which the *Events* are repeated. If the time of an *Event* is larger than the cycle time, the modulo operation can be applied because of the periodic nature of chromatographic processes. Hence, *Event* is performed at *time modulo* Δt_{cycle}.

Figure 2 shows an example *ProcessModel* illustrating an arbitrary *FlowSheet* with four connected *UnitOperations*. In addition, the figure illustrates the use of regular and conditional *Events*. In the depicted process, at $t = 0$, the output of *Unit*1 is set to $[1, 0]$ which indicates that 100% of the outgoing flow is directed to *output*0 (i.e., *Unit*2). At $t = 1$, the output is changed to $[0.5, 0.5]$ and 50% of the flow is directed to *Unit*3. At the same time, the attribute *attr*0 of *Unit*3 is set to 2.7183.

By default, the SI unit system is applied for all parameters in *CADET-Process*. However, alternative unit systems can be applied, including dimensionless model parameters.

2.2. Process Simulation

The framework does not provide a numerical solver for the model equations, but allows using any external routine for this purpose, provided, a corresponding *SimulatorAdapter* was implemented that translates the abstract *ProcessModel* representation to the application programming interface (API) of the *Solver*. It is important to note that not every *Solver* is capable of simulating every *ProcessModel*. Some are limited with respect to the connectivity of *UnitOperations*, the number of components, or the selected isotherm. The *SimulatorAdapter* provides a unified interface for simulating the *ProcessModel* using such libraries and the concrete implementations of the adapter are responsible for converting the *ProcessModel* to the specific model configuration of the external library. In case of *CADET*, the *SimulatorAdapter* stores the parameters in an HDF5 file, calls a command-line interface (CLI) to run the simulation, and, after reading the results from the same file, converts them to back to the framework's internal format.

While *CADET* is used as the main solver of the framework, other solvers are possible. However, as mentioned, *CADET* is a particularly convenient choice, since it is a fast and accurate solver for column models of various levels of detail, as well as transport and binding mechanisms. Furthermore, *CADET* is also freely available under the terms of the GPLv3. Moreover, in CADET all *UnitOperations* can be solved simultaneously, which is particularly important for dynamic systems with internal recycles.
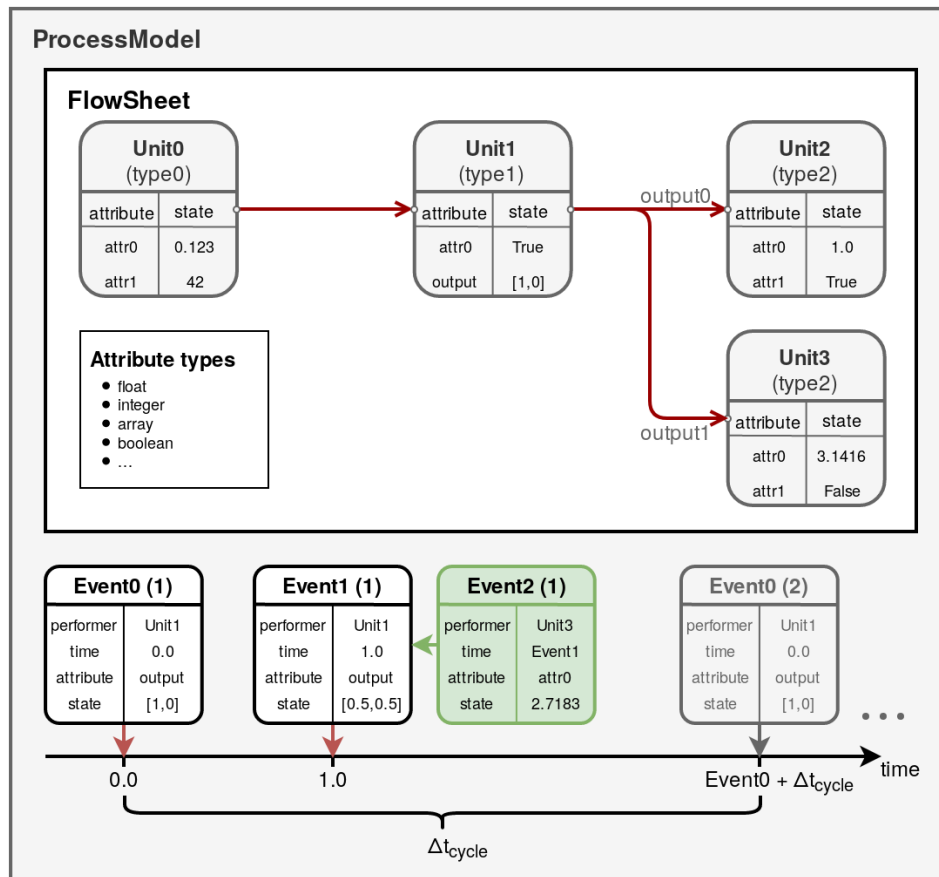
Figure 2. *ProcessModel* of a simple example process represented as a *FlowSheet* of four connected *UnitOperations* and modelling of its dynamic operation using *Events*. The attributes represent the physico-chemical properties of the *UnitOperation*. An *Event* defines when the state of a *UnitOperation* attribute is changed to a certain value. The time at which the *Event* is executed can depend on other *Events* (conditional *Event*, see green box). The cycle time Δt_{cycle} defines the when *Events* are repeated.

2.3. Fractionation

Key information for evaluating the *Performance* (cf. Figure 1) of a chromatographic process are the amounts of the target components in the collected product fractions. To define corresponding fractionation intervals, the *Chromatograms*, i.e., the concentration profiles $c_{i,k}(t)$ at the outlet(s) of the process must be evaluated. (In the strict sense, a *Chromatogram* is given at the outlet of a single column. Please note that here this term is used more generally for the concentration profiles at the outlets (*Sinks*) of a *FlowSheet*, which only accounts for material leaving the process.) The times for the start, $t_{start,j}$, and the end, $t_{end,j}$, of a product fraction j have to be chosen such that constraints on product purity are met. It is important to note, that in advanced chromatographic process configurations, outlet *Chromatograms* can be much more complex than in Figure 3 (left), and that multiple sections of the chromatogram may represent suitable fractions j for collecting one target component i. Moreover, *FlowSheets* can have multiple *Sinks* k that have to be fractionized simultaneously, and the volumetric flow rate Q_k at the outlets may depend on time. These aspects are considered by defining the total product amount of a component i as

$$m_i = \sum_{k=1}^{n_{chrom}} \sum_{j=1}^{n_{frac,k}^i} \int_{t_{start,j}}^{t_{end,j}} Q_k(t) \cdot c_{i,k}(t)dt, \tag{1}$$

where $n_{frac,k}^i$ is the number of fractions considered for component i in *Chromatogram* k, and n_{chrom} is the number of *Chromatograms* that is evaluated. Further *Performance* criteria typically used for evaluation

and optimization of chromatographic performance are the specific productivity, PR_i, the recovery yield, Y_i, and the specific solvent consumption, EC_i, which all depend on the product amounts:

$$PR_i = \frac{m_i}{V_{solid} \cdot \Delta t_{cycle}}, \tag{2}$$

$$Y_i = \frac{m_i}{m_{feed,i}}, \tag{3}$$

$$EC_i = \frac{V_{solvent}}{m_i}, \tag{4}$$

with V_{solid} being the volume of stationary phase, $V_{solvent}$ that of the solvent introduced during a cycle with duration Δt_{cycle}, and m_{feed} the injected amount of mixture to be separated. Multiple *Sources* can be considered for the amounts of consumed feed and solvent,

$$V_{solvent} = \sum_{s=1}^{n_{solvents}} \int_0^{t_{cycle}} Q_s(t)dt, \tag{5}$$

$$m_{feed,i} = \sum_{f=1}^{n_{feeds}} \int_0^{t_{cycle}} Q_f(t) \cdot c_{f,i}(t)dt. \tag{6}$$

For the cumulative product purities PU_i holds

$$PU_i = \frac{m_i^i}{\sum_{l=1}^{n_{comp}} m_l^i}, \tag{7}$$

where n_{comp} is the number of mixture components and m_l^i is the mass of component l in target fraction i.

Several approaches exist for optimizing fractionation times. Typically, the task is formulated as an optimization problem which minimizes an objective that is a relatively simple function of the above performance criteria, while constraints are set on product purity. The optimization problem itself may be solved by various approaches, including screening methods [21], heuristic-based algorithms [22], sophisticated multi-objective methods combining nonlinear programming (NLP) and screening [23], to rigorous optimization based on mixed-integer nonlinear programming (MINLP) [24] or mixed-integer linear programming (MIP) [25].

In this work, the *Fractionation* module automatically sets up an *OptimizationProblem*. For every component, different purity requirements can be specified, and any function may be applied as objective (see Section 2.5 for details). For the objective and constraint functions, fractions are pooled from all *Outlets* of the *FlowSheet* (see Equations (1) and (7)). As initial values for the optimization, areas of the chromatogram with sufficient "local purity" [22] are identified, i.e., intervals where $PU_i(t) = c_i(t) / \sum_j c_j(t) \geq PU_{min,i}$. These initial intervals are then expanded by the optimizer towards regions of lower purity while meeting the cumulative purity constraints. In the current implementation, *COBYLA* [26] of the *SciPy* [27] library is used as *Optimizer*. Yet, any other solver or heuristic algorithm may be implemented.

The strategy is illustrated in Figure 3, which shows a three-component *Chromatogram* with the final determined fractionation intervals (left) together with the local purity profiles used for finding initial values (right). In the example, all components are considered to be target products and a minimum cumulative purity of 95% is required for each component. A ranked product mass was used as objective function (see Equation (9)).
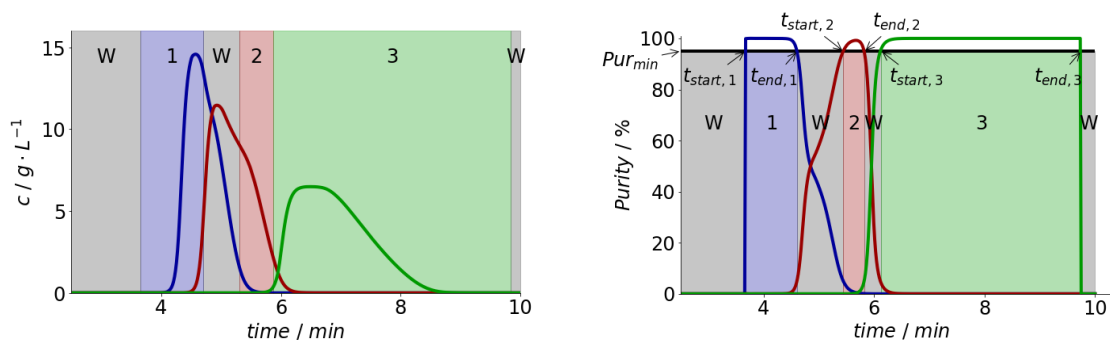
Figure 3. Fractionation of an example *Chromatogram*. **Left**: Product and waste fractions determined by the *FractionationOptimization* module. **Right**: Local purity profiles used for initial values. Please note that the initial waste fraction between fractions 2 and 3 vanishes after optimization. Arbitrary parameters were used for illustration.

2.4. Assertion of Cyclic Stationarity

Preparative chromatographic separations are operated in a repetitive fashion. In particular processes that incorporate the recycling of streams, such as steady-state-recycling (SSR) or simulated moving bed (SMB), have a distinct startup behavior that takes multiple cycles until a periodic steady state is reached. However, in conventional batch chromatography several cycles are also needed to attain stationarity in optimized situations where there is a cycle-to-cycle overlap of the elution profiles of consecutive injections. However, it is not known beforehand how many cycles are required until cyclic stationarity is established.

For this reason, the *Simulator* can simulate a *ProcessModel* for a fixed number of cycles, or continue simulating until the corresponding module in Figure 1 confirms that cyclic *Stationarity* is reached. Different criteria can be specified such as the maximum deviation of the concentration profiles or the peak areas of consecutive cycles [28]. The simulation terminates if the corresponding difference is smaller than a specified value. For the evaluation of the process (see Section 2.3), only the last cycle is examined, as it yields a representative *Performance* of the process in all later cycles. Figure 4 shows the chromatogram of an example SSR process, where the cyclic steady state is reached after 27 cycles. SSR chromatography is a process, in which the unresolved middle fraction is recycled and mixed with fresh feed in a continuously stirred tank (CSTR). Thus, a periodic steady state is developed that can lead to high purities and yields [6]. More details on SSR chromatography can be taken from the Appendix A.
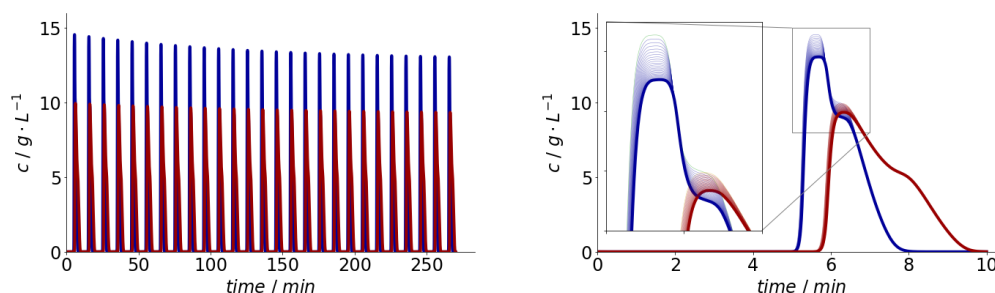

Figure 4. Example SSR process in mixed-recycle operation [7] for the separation of two components (blue and red) reaching cyclic steady state after 27 cycles. **Left**: concentration profiles at the column's outlet. **Right**: overlay of all profiles showing the transient towards stationarity. Arbitrary parameters were used for illustration.

2.5. Process Optimization

The optimization of preparative chromatographic processes is challenging since many degrees of freedom exist that can be manipulated to improve their performance. Besides continuous variables

such as column length and diameter, injection concentration, or flow rate (and many more), the times at which *Events* are performed have to be optimized. Even structural decisions can be subjected to optimization to simultaneously determine the optimal operating concept with the optimal conditions for a given separation.

The performance of chromatographic processes requires the optimization of fractionation times (see Section 2.3). As recommended in [25,29], optimal process design in our framework is treated as a bi-level problem to reduce the overall complexity of the optimization problem. On the top level, the variables of the *ProcessModel* are optimized using one *Optimizer*, and on the lower level, fractionation times of the simulated *Chromatograms* are determined using another *Optimizer* with the same (or a proportional) goal function (see Figure 1).

Simple objective functions are often used that combine criteria such as those in Section 2.3. However, also more detailed cost functions may be applied to directly maximize the profit of a separation. For discussions on useful objective functions see, for example, [2,3,25].

Both gradient-based non-linear programming (NLP) solvers (e.g., interior-point algorithms [30], sequential programming [29], or simplex [11]) and derivative-free approaches (e.g., genetic algorithm [31] and Gaussian process regression [32]) were used successfully for optimal process design. Moreover, mixed-integer non-linear programming (MINLP) was applied to general process optimization using extended cutting plane algorithms [33], as well as for structural decision variables using outer approximation [34], and evolutionary algorithms [35]. The examples show that many different approaches exist for optimizing chromatographic processes. Depending on the problem at hand, different solvers may outperform others. For example, some of the derivative-based solvers can guarantee finding local, but not necessarily global optima. Additionally, they may be insensitive to the change of variables due to a flat objective function. On the other hand, derivative-free solvers can often handle non-convex functions better but require significantly more computational power.

Against this problem, an *OptimizationProblem* class is introduced that decouples the problem formulation from the *Optimizer* used for its solution (see Figure 1), allowing for a simple comparison of different optimization approaches. The *OptimizationVariables x* may refer to any attribute of the *ProcessModel*. This includes model parameters, as well as *FlowSheet-Events*. As for the latter, not only the time when they are executed can be optimized, but also the value to which the attribute is changed when the *Event* is performed, allowing for structural optimization. Bound constraints and linear constraints can limit the parameter space and the user is free to define arbitrary function $f(x)$ as the objective, and add arbitrary nonlinear constraint functions $c(x)$ to the *(OptimizationProblem)*.

The *OptimizerAdapter* provides a unified interface for using external optimization libraries. It is responsible for converting the *OptimizationProblem* to the specific API of the external *Optimizer*. Currently, adapters to DEAP [36], GPyOpt [37], and SciPy's optimization suite [27] are implemented, all of which are published under open source licenses that allow for academic as well as commercial use.

To facilitate the bi-level approach mentioned beforehand, an intermediate *ProcessEvaluator* is implemented. It is responsible for passing the *ProcessModel* to the *Simulator*, checking *Stationarity* requirements and calling the *Fractionation* subroutine. The *ProcessEvaluator* returns the *Performance* to the *OptimizationProblem* which in turn determines the value of the objective function at the current point. The value is passed to the external *Optimizer*, then determines new candidates to be evaluated until a termination criterion is reached.

2.6. User Interface

The structure of *CADET-Process* is reflected by the organization of its source code. The framework is written using an object-oriented approach in which multiple classes are defined. Classes describe data structures and hold methods to modify the data, or to interact with other objects.

The configuration is done by creating instances of classes and configuring them. Physico-chemical parameters of *BindingModels* and *UnitOperations* have to be provided, as well as the *FlowSheet* connectivity and the *Events*. Optionally, numerical parameters for the simulation can be set as

attributes of the concrete *SolverAdapter* implementations. Individual purity requirements and criteria for the assertion of stationarity are specified by configuring instances of the *FractionationOptimization* and *Stationarity*. For process optimization, the *ProcessModel* is passed as input argument to the *OptimizationProblem* in which the variables, objectives, and constraints are defined. The problem is solved by passing it as an input argument to the *OptimizerAdapter* interface. Here, also the termination criteria and other settings of the specific *Optimizer* are stored.

While usually, a single Python script is used for the configuration, the different tasks can also be separated into different modules. For more details, we refer to the online documentation of the framework: https://cadet-process.readthedocs.io. Besides a description of the different modules, tutorials and a collection of examples is given that can be used as the base for more advanced applications.

3. Case Studies

To demonstrate the application and capabilities of the framework, several case studies were conducted. The scripts to recreate the simulations and optimizations reported below can be found in the supplementary material and online: https://github.com/fau-advanced-separations/CADET-Process_case_studies_2019. In the case studies, preparative separations of mixtures with two or three components were considered using processes of different complexity and different optimization variables. In all cases, the equilibrium-dispersive model with competitive Langmuir adsorption isotherm was applied. All processes were run in flow-through mode and the solvent does not have any impact on the binding. The parameters used are summarized in Table 1. With the applied moderately nonlinear conditions and an axial dispersion coefficient that corresponds to several theoretical stages of around 400 and 2000, respectively, the separation difficulty of the examples can be regarded as "modest".

Independent of this, the optimization of such processes is always challenging, even for simple separations. Hence, we do not claim that the obtained results are globally optimal but serve to illustrate that the application of the framework can give insight into interesting operating strategies. For all optimization cases, the DEAP optimizer [36] with a non-dominated sorting genetic algorithm [38] was used with the following objective function,

$$f(x) = \frac{PR_{ranked}(x) \cdot Y_{ranked}(x)}{EC_{ranked}(x)}. \tag{8}$$

This type of equation is commonly used for optimization in chromatography. In addition, we apply here a component-related ranking of the performance criteria according to

$$\eta_{ranked} = \frac{\sum_{i=1}^{n_{comp}} w_i \cdot \eta_i}{\sum_{i=1}^{n_{comp}} w_i}, \quad \eta_i \in \{m_i, PR_i, Y_i, EC_i, ...\}, \tag{9}$$

with the performance criteria for the individual components calculated from Equations (2) through (4). The weighting factors $w_i \geq 0$ introduced in Equation (9) allow a simple ranking of the individual mixture components regarding their value. This provides a simple means for considering components of different economic relevance in the same mixture, but also for switching between different scenarios by exchanging between target products ($w_i > 0$) and undesired waste components ($w_i = 0$).

Table 1. Parameters used in the different case studies.

Parameter			Binary Separation		Ternary Separation		
Column length	L_c	cm	60		60		
Column diameter	D_c	cm	2.4		2.4		
Column porosity	ϵ	-	0.7		0.7		
Apparent axial dispersion	$D_{app,i}$	$m^2 \cdot s^{-1}$	2.4×10^{-6}		4.7×10^{-7}		
Flow rate	Q	$mL \cdot min^{-1}$	1		1		
Feed concentration	$c_{F,i}$	$g \cdot L^{-1}$	10,	10	10,	10,	10
Langmuir parameter	b_i	$L \cdot g^{-1}$	0.02,	0.03	0.01,	0.015	0.03
Saturation capacity	$q_{S,i}$	$g \cdot L^{-1}$	100,	100	100,	100,	200
Minimum product purity	$PU_{min,i}$	-	0.95,	0.95	0.95,	0.95	0.95

3.1. Case 1: Batch Chromatography for a Binary Separation

As an introductory example, we demonstrate the setup of a simple *ProcessModel* and *OptimizationProblem* for the case of a two-component separation by batch chromatography, which is the simplest and most widely used operating concept.

In batch chromatography, at the beginning of a cycle a certain amount of component mixture is injected and transported through the column by an eluent. The components elute as more or less separated peaks from the column and are collected as fractions. This operating mode is modelled here by the simple *FlowSheet* in Figure 5 consisting of a *Column*, two *Sources* for feed and eluent, and a *Sink* as outlet. Its operation is described by defining the following *Events*:

- *FeedOn*: Start of injection at $t = 0$; flow from *Feed* source is turned on;
- *FeedDuration* (auxiliary): Duration of injection;
- *FeedOff* (conditional): End of injection at $t \;\widehat{=}\; FeedOn + FeedDuration$; flow of *Feed* turned off;
- *EluentOn* (conditional): Elution; flow from *Eluent* source turned on at $t \;\widehat{=}\; FeedOff$;
- *EluentOff* (conditional): End of Elution; flow of *Eluent* is turned off at $t \;\widehat{=}\; FeedOn$.
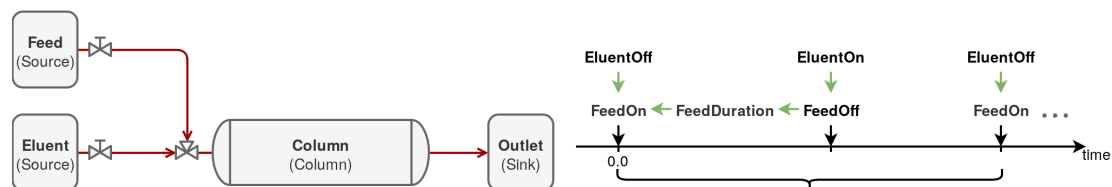

Figure 5. *FlowSheet* and *Events* for modelling conventional batch chromatography. Green arrows indicate the dependencies of conditional *Events*.

Since a preparative separation is operated in a repetitive manner, these *Events* are repeated after the cycle time Δt_{cycle}, defined in the ProcessModel (see Section 2.1), until cyclic *Stationarity* is attained (see Section 2.4).

Figure 6 (left) shows the chromatogram obtained from a simulation which, in fact, corresponds to the optimal scenario obtained after optimization of the *FeedDuration* and the cycle time Δt_{cycle} using Equation (8) with equally important products, $w_1 = w_2 = 1$. The chromatogram corresponds to a so-called stacked-injections scenario where the cycle time is so short that elution profiles from consecutive injections directly follow each other. Moreover, a certain cycle-to-cycle overlap of consecutive profiles is observed. This is the reason attaining cyclic steady-state takes several (here, four) cycles. Overall, the result appears plausible, since it is typical for such optimized scenario with limited purity requirements (see for example [11]). The fractionation of the last cycle of the optimized process is shown in Figure 6 (right). An interesting aspect is that the optimizer automatically allocated not only two product fractions (marked by 1 and 2), but also two waste fractions (*W*). These are located in the mixed band of the given elution profile and in the overlap between consecutive profiles. To our

opinion, the optimal allocation of several waste fractions is so far not sufficiently addressed. Finally, the obtained performance parameters in Table 2 illustrate the known pronounced trade-off in batch chromatography, where good performance in terms of productivity and eluent is at the expense of limited yield.
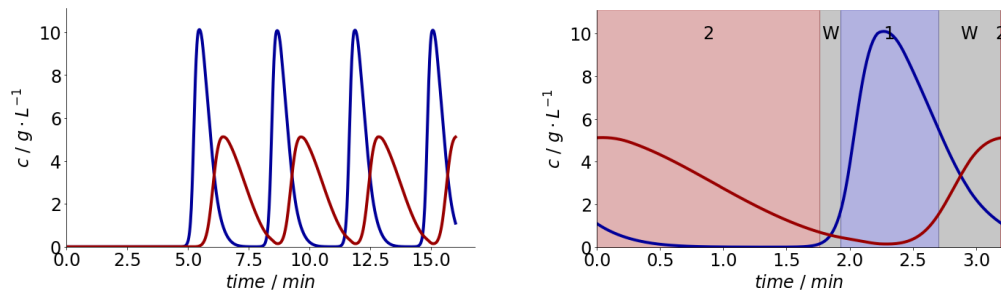

Figure 6. Optimization result for a binary separation by batch chromatography. **Left**: development of chromatogram until cyclic steady state is attained after four cycles. **Right**: determined fractionation intervals shown for the last cycle. Two fractions for the products (1, 2) as well as two waste fractions (*W*) were identified.

Table 2. Performance and values of optimized variables of binary batch separation.

Objective function	$f(x)$	$\mathrm{g}^3 \cdot \mathrm{L}_{\text{eluent}}^{-1} \cdot \mathrm{L}_{\text{solid}}^{-1} \cdot \mathrm{s}^{-1}$	0.038	
Component purity	PU_i	-	0.950	0.950
Recovery yield	Y_i	-	0.776	0.712
Specific productivity	PR_i	$\mathrm{g} \cdot \mathrm{L}_{\text{solid}}^{-1} \cdot \mathrm{s}^{-1}$	0.023	0.021
Specific solvent consumption	EC_i	$\mathrm{L}_{\text{eluent}} \cdot \mathrm{g}^{-1}$	0.409	0.441
Cycle time	Δt_{cycle}	min	03:12	
FeedDuration	Δt_{feed}	min	00:46	

3.2. Case 2: Batch Chromatography for a Ternary Separation

As a slightly more complex application, we consider a ternary separation by batch chromatography. This case is modelled using the same flow sheet and events as in case 1 above by adding a third, significantly stronger adsorbing component to the *ProcessModel* (see Table 1). Three different optimization scenarios are considered to illustrate that the framework can be used to identify optimal operating strategies, and that the latter can be rather different depending on the choice of optimization variables and the objective function.

The first scenario is analogous to case 1 above, i.e., Δt_{cycle} and *FeedDuration* were optimized for all three components being equally relevant products with $w_i = 1$ in Equation (8). Figure 7 shows the optimal chromatogram and the fractionation and their obtained optimization results are summarized in Table 3. As in case 1, a cycle-to-cycle overlap was found optimal. Also a small waste fraction was allocated. Apart from that, closer inspection of the chromatogram in Figure 7 reveals an interesting operating strategy. The optimal cycle time (i.e., the time between consecutive injections) is rather short (7:29 min for a *FeedDuration* of 1:07 min). This leads to a situation where the first and second component peaks of a given cycle overtake the profile of the third component of the previous cycle. This somewhat non-intuitive operating mode is sometimes denoted as "interlocked peaks".
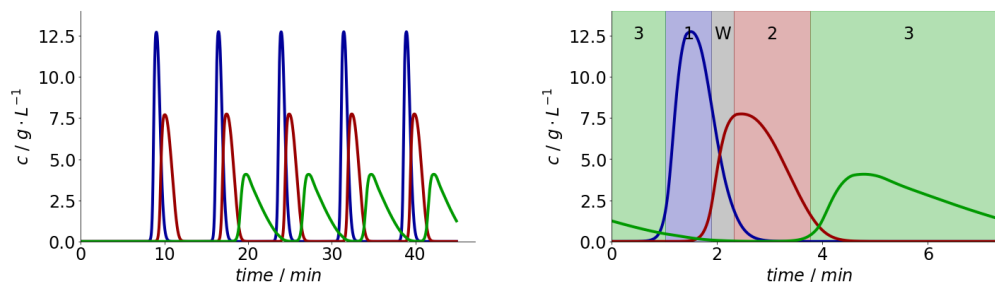
Figure 7. Optimization of a ternary separation by batch chromatography. Feed duration and cycle time were optimized while column length was kept constant. **Left**: development of chromatogram into cyclic steady state. A so-called peak interlocking is observed where the weaker adsorbing compounds 1 and 2 (blue and red) "overtake" the peak of the strongest retained component 3 (green) from the previous cycle. **Right**: determined fractionation intervals for the three products and a waste fraction.

In the second scenario, we consider also the column length as an additional continuous variable to be optimized. All other conditions are identical to the scenario above. Figure 8 (left) shows that the column is operated conventionally (i.e., no peak interlocking as in Figure 7 (left)). As can be seen from the results, the optimal process differs significantly from the previous scenario. It is characterized by a significantly shorter column (10 cm instead of 60 cm), operated with a much shorter cycle time and *FeedDuration* (1:53 min and 31 s, respectively). The fractionation in Figure 8 (right) and data in Table 3 reveal that here a large productivity of component 3 is favored at the expense of a very low yield of components 1 and 2.
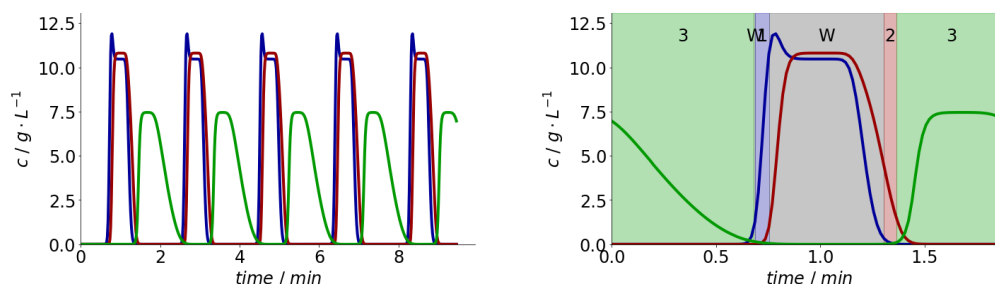

Figure 8. Optimization result for a ternary separation by batch chromatography using the same parameters as for Figure 7, but optimizing in addition to feed duration and cycle time also the column length L_c. **Left**: development of chromatogram into cyclic steady state. **Right**: fractionation intervals. In contrast to Figure 7, the product fractions for components 1 and 2 are small, while that of 3 is large. Two waste fractions were identified, one of which being of significant size.

The previous scenario, which is not resource efficient, is a result of the chosen objective and parameters. To elucidate this, in the last example a different weighting/ranking of the three compounds is assumed such, that only the intermediate component 2 is a product, while compounds 1 and 3 are undesired. This is adjusted by setting in Equation (8) $w_2 = 1$, and $w_1 = w_3 = 0$. As expected, the optimum (see data in Table 3 and Figure 9) is characterized by a large product fraction and, thus, increased yield of component 2, while the whole rest of the elution profile is identified as a waste fraction. Now, as in the first example, interlocked injections are found optimal again, but using a significantly shorter column (38 cm instead of 60 cm) with a cycle time of 2:15 min, and a *FeedDuration* of 16 s.
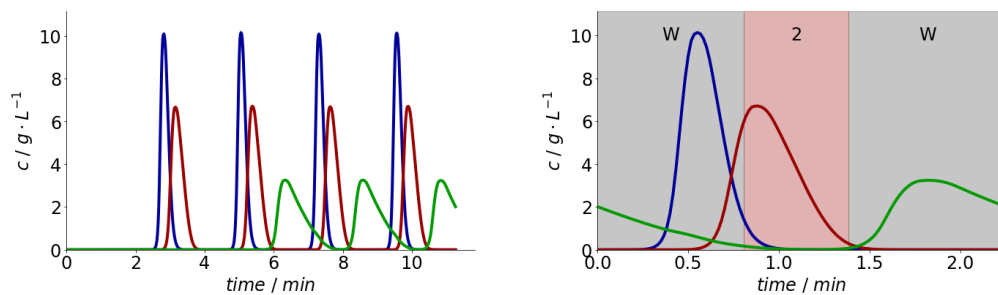
Figure 9. Optimization result for a ternary separation by batch chromatography using the same parameters as for Figure 7, but optimizing in addition to feed duration and cycle time also the column length L_c. **Left**: development of chromatogram into cyclic steady state. **Right**: fractionation intervals. In contrast to Figure 8, only one fraction for component 2 is found. The rest of the profile is identified as waste because components 1 and 3 are not considered to be valuable.

Table 3. Performance and values of optimized variables of ternary batch separation.

		1st Scenario			2nd Scenario			3rd Scenario		
$f(x)$	$g^3 \cdot L_{eluent}^{-1} \cdot L_{solid}^{-1} \cdot s^{-1}$		0.0089			0.0404			0.0165	
PU_i	-	0.950	0.950	0.950	0.950	0.950	0.989	0	0.950	0
Y_i	-	0.771	0.723	0.973	0.077	0.034	1.00	0	0.800	0
PR_i	$g \cdot L_{solid}^{-1} \cdot s^{-1}$	0.007	0.007	0.008	0.016	0.007	0.205	0	0.019	0
EC_i	$L_{eluent} \cdot g^{-1}$	0.743	0.791	0.589	3.337	7.594	0.259	∞	0.914	∞
Δt_{cycle}	min		07:29			01:53			02:15	
FeedDuration	min		01:06			00:31			00:16	
L_c	cm		-			10.0			37.6	

3.3. Case 3: Two Columns in Series for a Ternary Separation

To demonstrate the framework's ability of handling multi-column arrangements, a second column is included in the *FlowSheet* (see Figure 10). Furthermore, the case shows that multiple *Chromatograms* can be fractionized simultaneously in order to determine the process performance.

One approach to increase the productivity of a process is to "shave off" sufficiently separated fractions of the mixture and only let not sufficiently parts of the mixture migrate over an additional column. The following *Events* are characteristic for such a process:

- *FeedOn*: Start of injection at $t = 0$; flow from *Feed* source is turned on;
- *FeedDuration* (auxiliary): Duration of injection;
- *FeedOff* (conditional): End of injection at $t \mathrel{\hat{=}} FeedOn+FeedDuration$; flow of *Feed* turned off;
- *Eluent1On* (conditional): Elution; flow from *Eluent 1* source turned on at $t \mathrel{\hat{=}} FeedOff$;
- *Eluent1Off* (conditional): End of Elution; flow of *Eluent 1* is turned off at $t \mathrel{\hat{=}} FeedOn$.
- *SerialOff*: Output of *Column 1* is redirected to *Outlet 1*;
- *SerialOn*: Output of *Column 1* is connected to input of *Column 2*;
- *Eluent2On* (conditional): Elution of *Column 2*; flow from *Eluent 2* source turned on at $t \mathrel{\hat{=}} SerialOff$;
- *Eluent2Off* (conditional): End of Elution of *Column 2*; flow of *Eluent 2* is turned off at $t \mathrel{\hat{=}} SerialOn$.
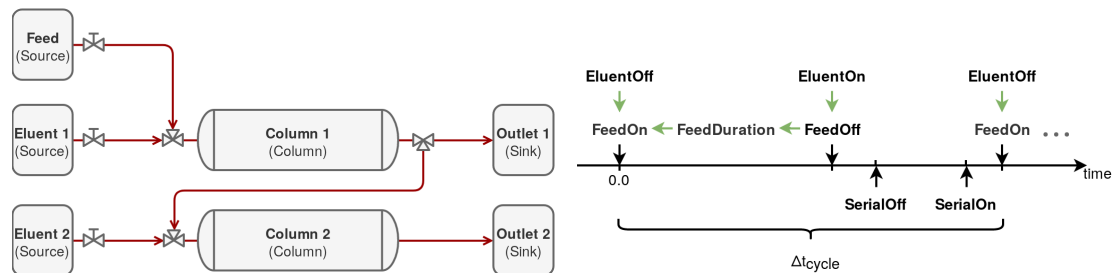
Figure 10. *FlowSheet* and *Events* for modelling columns in series with peak shaving.

Figure 11 depicts the simulation results for the ternary separation in Table 1. The concentration profiles are shown at the outputs of *Column 1* (top left) and *Column 2* (bottom left), and the fractionation of the both outlets (right). As before, subject to optimization were the cycle time and the *FeedDuration*. Furthermore, the event times of *SerialOff* and *SerialOn* were optimized, as well as the length of both columns. In the example, the ranking of the three compounds is assumed such, that only components 1 and 2 are product, and the last component 3 is considered waste by setting $w_1 = w_2 = 1$, and $w_3 = 0$. It is shown that only components 1 and 2 migrate over the second column, and the strongly adsorbing component is removed from the system before entering *Column 2*. This increases the productivity of the system, as the throughput of the process by "blocking" the latter part of the *Column*. Furthermore, it was found, that a short column length for the *Column 1* (28.6 cm) is necessary to achieve this task, whereas *Column 2* is considerably longer (59.3 cm) to sufficiently separate components 1 and 2 (see Table 4). The presence of small amounts of component 3 on the second column, as well as baseline separation between components 2 and 3 in the concentrations profiles after *Column 1* suggest that the process may still be suboptimal. A shorter length of *Column 1* would be sufficient to separate the third component completely and its presence on *Column 2* limits the cycle time. However, an interesting strategy was identified with high yield.
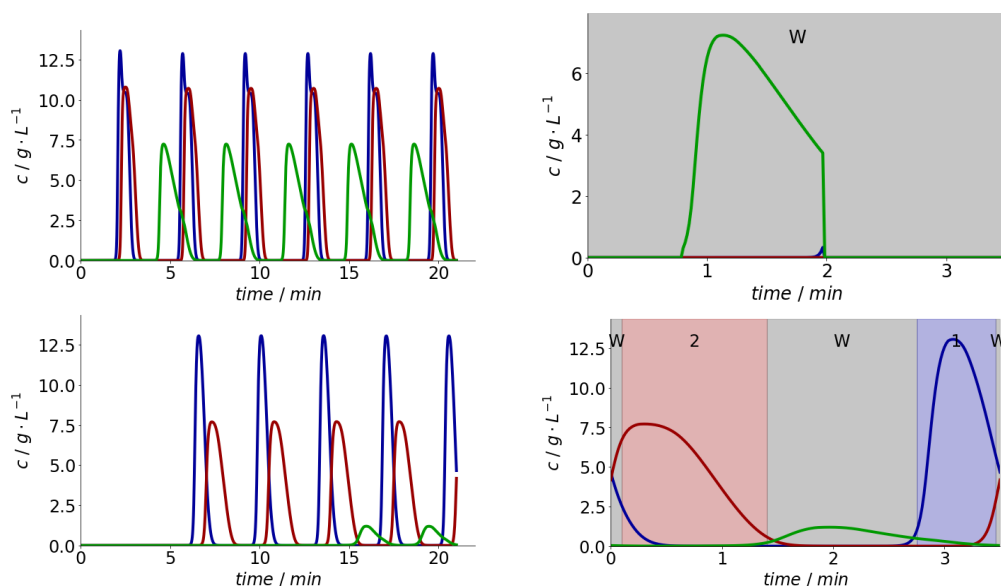

Figure 11. Optimization result for a ternary separation using two columns in series: *FeedDuration*, the times of Δt_{cycle}, *FeedDuration*, *SerialOff* and *SerialOn*, as well as the length of *Column 1* and *Column 2* were optimized. **Left**: development of chromatogram of *Column 1* (top) and *Column 2* (bottom). **Right** (top): no product fraction determined for *Outlet 1*. **Right** (bottom): fraction intervals for the two products (1 and 2) and two waste fractions (*W*) identified.

Table 4. Performance and optimized variables for ternary separation using columns in series.

$f(x)$	$\mathrm{g}^3 \cdot \mathrm{L}_{\text{eluent}}^{-1} \cdot \mathrm{L}_{\text{solid}}^{-1} \cdot \mathrm{s}^{-1}$		0.0240	
PU_i	-	0.950	0.950	0
Y_i	-	0.879	0.875	0
PR_i	$\mathrm{g} \cdot \mathrm{L}_{\text{solid}}^{-1} \cdot \mathrm{s}^{-1}$	0.016	0.016	0
EC_i	$\mathrm{L}_{\text{eluent}} \cdot \mathrm{g}^{-1}$	0.585	0.587	∞
Δt_{cycle}	min		03:30	
$FeedDuration$	min		01:06	
$L_{c,1}$	cm		28.6	
$L_{c,2}$	cm		59.3	

4. Summary and Conclusions

The development and optimal design of advanced chromatographic processes is challenging due to their distinct nonlinear dynamics, periodic operation, and a multitude of design variables. A software platform was developed that greatly simplifies this by decoupling the main design tasks into interchangeable modules for process modelling, simulation, product fractionation, and process optimization. This enables flexible and fast adjustment of process and column configurations, chromatographic interaction mechanisms, and design goals or constraints. The software was implemented using the programming language Python and includes an interface to the open-source simulation tool *CADET*, a fast and accurate numerical solver for chromatographic processes. This makes the program, denoted as *CADET-Process*, directly applicable to developing advanced processes for demanding applications in, for example, the separation of biomolecules, nanoparticles, and the like, where complex interaction mechanisms and various dispersive effects need to be considered.

The *FlowSheet* of the chromatographic systems could be described by connecting *UnitOperations* in a directed graph which represents the material flow between them. The dynamic behavior typical for chromatography, especially for advanced operating modes, was modelled by defining events that change the state of the system at given times. The introduction of dependencies of the event times reduces the complexity of modelling advanced operating concepts. Features, such as the assertion of cyclic stationarity, as well as the automatic determination of optimal fractionation times proved crucial for an efficient implementation of the simulation module and provide standalone functionality that can also be used outside the framework.

Several case studies for the optimization of binary and ternary separations were performed on single- and multicolumn systems with and without recycling. It was shown how the choice of optimization variables and product ranking can lead to different optimal process designs. Although the main purpose of the case studies was to introduce and demonstrate the applicability of the framework, the results feature optimal designs of new operating strategies, such as batch chromatography with optimal cycle-to-cycle overlap and waste fractions, as well as interlocked peaks for separation problems with strongly binding components.

Various extensions of the software can be envisaged, some of them are already in progress. The implementation of more complex gradients is important, for example for biotechnological applications. Furthermore, the normalization of optimization variables with different orders of magnitude can greatly improve the performance of the optimization, as can other, more complex parameter transforms. Also the possibility of defining multiple competing objectives in the *OptimizationProblem* would allow for the analysis of Pareto optimality which enables the study of more interesting problem formulations and cases. As regards processes, *i.a.* SMB chromatography and its multiple variants are of interest.

The presented framework is a fully featured toolbox for modelling and optimizing chromatographic processes, which is the base for more systematic and targeted development of novel chromatographic processes. More operating concepts, optimization variables, and objective functions

can easily be implemented. The framework's modular approach in general enables a straightforward exchange of process models, numeric solvers, and optimizers for future extension of the software.

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

Abbreviations

The following abbreviations are used in this manuscript:

API	Application Programming Interface
CLI	Command-line interface
CSTR	Continuously Stirred Tank Reactor
GPLv3	GNU General Public License version 3
MIP	Mixed-Integer Programming
MINLP	Mixed-Integer Non-Linear Programming
MR-SSR	Mixed-Recycle Steady-State Recycling
SMB	Simulated Moving Bed
SSR	Steady-State Recycling

Appendix A. Mixed-Recycle Steady-State-Recycling

Different operating modes were proposed for steady state recycling (SSR) chromatography. In this study, we will give an example of mixed-recycle steady-state-recycling (MR-SSR) in which the unresolved middle fraction is recycled and mixed with fresh feed in a continuously stirred tank (CSTR). Thus, a periodic steady state is developed that can lead to high purities and yields [6]. Using the flowsheet depicted in Figure A1, the following events are defined:

- *InjectOn*: Start of injection at $t = 0$; flow from *Tank* is turned on;
- *InjectOff* (conditional): End of injection; flow from *Tank* is turned off at $t \cong InjectOn + FeedDuration + RecycleOff - RecycleOn$;
- *EluentOn* (conditional): Elution; flow from *Eluent* source turned on at $t \cong InjectOff$;
- *EluentOff* (conditional): End of Elution; flow of *Eluent* is turned off at $t \cong InjectOn$.
- *FeedOn* (conditional): Application of fresh feed; flow from *Feed* source is turned on;
- *FeedDuration* (auxiliary): Duration of fresh feed application;
- *FeedOff* (conditional): flow of *Feed* is turned off at $t \cong FeedOn + FeedDuration$;
- *RecycleOn*: Output of *Column* is redirected to *Tank*;
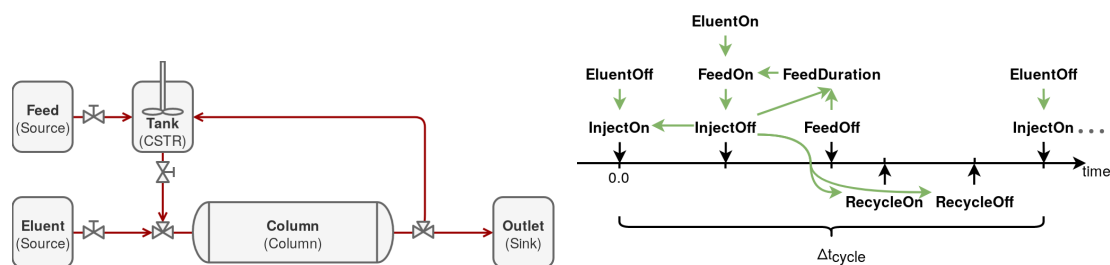- *RecycleOff*: Output of *Column* is connected to *Outlet*.

Figure A1. *FlowSheet* and *Events* for modelling MR-SSR chromatography.

By fixing the start of the injection to $t = 0$, and taking into account the defined event dependencies, four parameters are independent: the cycle time, the *FeedDuration*, and the times when the recycling is turned on and off, respectively.

References

1. Guiochon, G.; Shirazi, D.; Felinger, A.; Katti, A. *Fundamentals of Preparative and Nonlinear Chromatography*; Academic Press: Cambridge, MA, USA, 2006.
2. Schmidt-Traub, H.; Schulte, M.; Seidel-Morgenstern, A. *Preparative Chromatography*; Wiley: Hoboken, NJ, USA, 2012.
3. Nicoud, R.M. *Chromatographic Processes—Modeling, Simulation, and Design*; Cambridge University Press: Cambridge, UK, 2015.
4. Bombaugh, K.; Dark, W.; Levangie, R. High resolution steric chromatography. *J. Chromatogr. Sci.* **1969**, *7*, 42–47. [CrossRef]
5. Heuer, C.; Seidel-Morgenstern, A.; Hugo, P. Experimental investigation and modelling of closed-loop recycling in preparative chromatography. *Chem. Eng. Sci.* **1995**, *50*, 1115–1127. [CrossRef]
6. Bailly, M.; Tondeur, D. Recycle optimization in non-linear productive chromatography—I Mixing recycle with fresh feed. *Chem. Eng. Sci.* **1982**, *37*, 1199–1212. [CrossRef]
7. Sainio, T.; Kaspereit, M. Analysis of steady state recycling chromatography using equilibrium theory. *Sep. Purif. Technol.* **2009**, *66*, 9–18. [CrossRef]
8. Kaspereit, M.; Sainio, T. Simplified design of steady-state recycling chromatography under ideal and nonideal conditions. *Chem. Eng. Sci.* **2011**, *66*, 5428–5438. [CrossRef]
9. Siitonen, J.; Sainio, T.; Rajendran, A. Bypass chromatography—Design and analysis of an improved strategy for operating batch chromatography processes. *J. Chromatogr. A* **2012**, *1230*, 77–92. [CrossRef]
10. Ziomek, G.; Antos, D.; Tobiska, L.; Seidel-Morgenstern, A. Comparison of possible arrangements of five identical columns in preparative chromatography. *J. Chromatogr. A* **2006**, *1116*, 179–188. [CrossRef]
11. García Palacios, J.; Kaspereit, M.; Ziomek, G.; Antos, D.; Seidel-Morgenstern, A. Optimization and Analysis of Possible Column Arrangements for Multicomponent Separations by Preparative Chromatography. *Ind. Eng. Chem. Res.* **2009**, *48*, 11148–11157. [CrossRef]
12. Rodrigues, A.; Pereira, C.; Minceva, M.; Pais, L.; Ribeiro, A.; Ribeiro, A.; Silva, M.; Graca, N.; Santos, J. *Simulated Moving Bed Technology—Principles, Design and Process Applications*; Butterworth-Heinemann, Elsevier: Oxford, UK, 2015.
13. Siitonen, J.; Sainio, T. Explicit equations for the height and position of the first component shock for binary mixtures with competitive Langmuir isotherms under ideal conditions. *J. Chromatogr. A* **2011**, *1218*, 6379–6387. [CrossRef]
14. Mazzotti, M. Equilibrium theory based design of simulated moving bed processes for a generalized Langmuir isotherm. *J. Chromatogr. A* **2006**, *1126*, 311–322, doi:10.1016/j.chroma.2006.06.022. [CrossRef]
15. Siitonen, J.; Sainio, T. Unified design of chromatographic separation processes. *Chem. Eng. Sci.* **2015**, *122*, 436–451. [CrossRef]
16. Guiochon, G.; Lin, B. *Modeling for Preparative Chromatography*; Academic Press: San Diego, CA, USA, 2003.
17. Aspen Chromatography. Aspen Chromatography. Available online: https://www.aspentech.com/products/pages/aspen-chromatography (accessed on 22 October 2019).
18. ChromX. GoSilico ChromX. Available online: https://gosilico.com/chromx/ (accessed on 22 October 2019).

19. ChromWorks. YpsoFacto ChromWorks. Available online: https://www.ypsofacto.com/services-chemical-software-chromworks.php (accessed on 22 October 2019).

20. Leweke, S.; von Lieres, E. Chromatography Analysis and Design Toolkit (CADET). *Comput. Chem. Eng.* **2018**, *113*, 274–294. [CrossRef]

21. Heuer, C.; Küsters, E.; Plattner, T.; Seidel-Morgenstern, A. Design of the simulated moving bed process based on adsorption isotherm measurements using a perturbation method. *J. Chromatogr. A* **1998**, *827*, 175–191. [CrossRef]

22. Shan, Y.; Seidel-Morgenstern, A. Analysis of the isolation of a target component using multicomponent isocratic preparative elution chromatography. *J. Chromatogr. A* **2004**, *1041*, 53–62. [CrossRef]

23. Knutson, H.K.; Holmqvist, A.; Nilsson, B. Multi-objective optimization of chromatographic rare earth element separation. *J. Chromatogr. A* **2015**, *1416*, 57–63. [CrossRef]

24. Sreedhar, B.; Wagler, A.; Kaspereit, M.; Seidel-Morgenstern, A. Optimal cut-times finding strategies for collecting a target component from overloaded elution chromatograms. *Comput. Chem. Eng.* **2013**, *49*, 158–169. [CrossRef]

25. Dienstbier, J.; Schmölder, J.; Burlacu, R.; Liers, F.; Kaspereit, M. Global optimization of batch and steady-state recycling chromatography based on the equilibrium model. *Comput. Chem. Eng.* **2019**, in press. [CrossRef]

26. Powell, M.J. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in Optimization and Numerical Analysis*; Springer: Dordrecht, The Netherlands, 1994; pp. 51–67.

27. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python. *arXiv* **2019**, arXiv:1907.10121.

28. Holmqvist, A.; Andersson, C.; Magnusson, F.; Åkesson, J. Methods and tools for robust optimal control of batch chromatographic separation processes. *Processes* **2015**, *3*, 568–606. [CrossRef]

29. Arkell, K.; Knutson, H.K.; Frederiksen, S.S.; Breil, M.P.; Nilsson, B. Pareto-optimal reversed-phase chromatography separation of three insulin variants with a solubility constraint. *J. Chromatogr. A* **2018**, *1532*, 98–104. [CrossRef]

30. Kawajiri, Y.; Biegler, L.T. Large scale nonlinear optimization for asymmetric operation and design of Simulated Moving Beds. *J. Chromatogr. A* **2006**, *1133*, 226–240. [CrossRef] [PubMed]

31. Heinonen, J.; Kukkonen, S.; Sainio, T. Evolutionary multi-objective optimization based comparison of multi-column chromatographic separation processes for a ternary separation. *J. Chromatogr. A* **2014**, *1358*, 181–191. [CrossRef] [PubMed]

32. Freier, L.; von Lieres, E. Robust Multi-Objective Global Optimization of Stochastic Processes with a Case Study in Gradient Elution Chromatography. *Biotechnol. J.* **2018**, *13*, 1700257. [CrossRef] [PubMed]

33. Emet, S.; Westerlund, T. Solving a dynamic separation problem using MINLP techniques. *Appl. Numer. Math.* **2008**, *58*, 395–406. [CrossRef]

34. Kaspereit, M.; Swernath, S.; Kienle, A. Evaluation of competing process concepts for the production of pure enantiomers. *Org. Process Res. Dev.* **2012**, *16*, 353–363. [CrossRef]

35. García Palacios, J.; Kaspereit, M.; Kienle, A. Integrated Simulated Moving Bed Processes for the Production of Single Enantiomers. *Chem. Eng. Technol.* **2011**, *34*, 688–698. [CrossRef]

36. Fortin, F.A.; De Rainville, F.M.; Gardner, M.A.; Parizeau, M.; Gagné, C. DEAP: Evolutionary Algorithms Made Easy. *J. Mach. Learn. Res.* **2012**, *13*, 2171–2175.

37. GPyOpt: A Bayesian Optimization Framework in Python. 2016. Available online: http://github.com/SheffieldML/GPyOpt (accessed on 22 October 2019).

38. Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput.* **2013**, *18*, 577–601. [CrossRef]