Computers and Chemical Engineering 202 (2025) 109295

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/cace

High-Performance C++ and Julia solvers in CADET for weakly and strongly coupled continuous chromatography problems

Jesper Frandsen [a], Jan Michael Breuer [b,c], Johannes Schmölder [b], Jens Abildskov [a], Jakob Kjøbsted Huusom [a], Krist V. Gernaey [a], Eric von Lieres [b,d,*]

[a] Technical University of Denmark, Chemical and Biochemical Engineering, Kongens Lyngby, Denmark
[b] Forschungszentrum Jülich, IBG-1: Biotechnology, Jülich, Germany
[c] University of Cologne, Faculty of Mathematics and Natural Sciences, Cologne, Germany
[d] RWTH Aachen University, Computational Systems Biotechnology, 52074 Aachen, Germany

ARTICLE INFO

Keywords:
Column liquid chromatography
Continuous chromatography
Simulated moving bed
Discontinuous Galerkin Spectral Element Method
CADET

ABSTRACT

This study evaluates two numerical solvers for continuous chromatography: CADET-Core, implemented in C++, and CADET-Julia, implemented in Julia. Both codes can be used to model arbitrary networks of unit operations, including cyclic settings, and to solve the resulting strongly coupled systems accurately. Furthermore, the One-Column Analog (OCA) Fixed Point Iteration (FPI) method was implemented in CADET-Julia as a weak coupling alternative to the strongly coupled approach for Cyclic Steady State (CSS) simulations. We conducted extensive continuous chromatography benchmarks, comparing the C++ implementation of both a Discontinuous Galerkin Spectral Element Method (DGSEM) and a Finite Volume (FV) method with the Julia DGSEM implementation. The case studies encompass both full system dynamics from startup and CSS benchmarks. For the startup benchmarks, the Julia DGSEM implementation performed slightly better than the C++ implementation, and both performed significantly better than the C++ FV implementation. The C++ implementation scales better than the Julia implementation, resulting an computational advantage for large systems with many degrees of freedom. For the CSS benchmarks, the OCA-FPI approach performed best for spatial resolutions above a certain threshold, otherwise the strongly coupled approach using the Julia implementation was slightly faster.

1. Introduction

Conventional methods for large-scale production of high-value biological products predominantly rely on batch processing. However, continuous processing is increasingly replacing this traditional approach, offering several advantages such as improved capacity utilization, reduced utility consumption, and enhanced product quality (Khanal, 2022; Gerstweiler et al., 2021; Schmidt-Traub et al., 2020). Chromatography is crucial in downstream processing of high-value biological products, where workflows typically involve multiple chromatography steps. Despite its widespread application, transitioning chromatography from batch-mode to continuous-mode remains a significant challenge in the purification of biological products (Kumar and Lenhoff, 2020; Schmidt-Traub et al., 2020; Gerstweiler et al., 2021). Modeling can aid this transition by providing deep process insights and enabling process optimization (Matte, 2022; Gerstweiler et al., 2021).

Recently, there has been growing research interest in various continuous and semi-continuous chromatography setups, including Simulated moving bed (SMB), Periodic counter-current chromatography (PCC), and Multi-column solvent gradient purification (MCSGP)(Gerstweiler et al., 2021; Matte, 2022; Khanal, 2022; Rathore et al., 2022; Lothert and Wolff, 2023). A common scheme in these multi-column setups is connecting columns in series, in closed loops and/or in parallel, using multiple different inlets and outlets. After a certain time, the connections between the units change. Simulating systems with multiple columns is computationally more demanding as compared to batch chromatography, because the models for each individual unit must be solved, either simultaneously, which is referred to as strong coupling, or iteratively, one unit at a time, which is referred to as weak coupling (of the equations, see e.g. He et al., 2018). The system of Partial differential equations (PDE) to describe a continuous chromatography process is relatively large and coupled (Wu et al., 2013; He et al., 2020). The computational demands of the corresponding simulations is further increased, particularly for systems involving many columns, such as SMB chromatography, as these systems are usually simulated over many system volumes and until they reach Cyclic steady state (CSS), where their behavior becomes periodic. Compared to batch chromatography,

[*] Correspondence to: Forschungszentrum Jülich, IBG-1: Biotechnology, 52425 Jülich, Germany.
E-mail address: e.von.lieres@fz-juelich.de (E. von Lieres).

https://doi.org/10.1016/j.compchemeng.2025.109295
Received 15 January 2025; Received in revised form 13 July 2025; Accepted 14 July 2025
Available online 5 August 2025
0098-1354/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

List of Acronyms

GRM	General rate model
LRMP	Lumped rate model with pores
LRM	Lumped rate model without pores
IPF	Ideal plug flow
SMA	Steric mass action
PDAE	Partial differential algebraic equations
PDE	Partial differential equations
ODE	Ordinary differential equations
DAE	Differential algebraic equations
MAE	Maximum absolute error
CSTR	Continuously stirred tank reactor
FV	Finite volume
WENO	Weighted essentially non oscillatory
DGSEM	Discontinuous Galerkin spectral element method
DG	Discontinuous Galerkin
GSM	Galerkin spectral method
BDF	Backwards differentiation formula
NDF	Numerical differentiation formula
EOC	Experimental order of convergence
DoF	Degrees of freedom
OCA	One-column analog
FPI	Fixed-point iteration
CSS	Cyclic steady state
SMB	Simulated moving bed
MCSGP	Multi-column solvent gradient purification
PCC	Periodic counter-current chromatography

Nomenclature

ε_c	Column porosity
ε_p	Pore porosity
ε_t	Total porosity
ε_{CSS}	Tolerance of cyclic steady state
A_c	Cross section area
c^b	Concentration in the bulk (mobile) phase
c^p	Concentration in the particle pore phase
c^s	Concentration in the stationary phase
c_{in}	Inlet concentration
c_{out}	Outlet concentration
D_{ax}	Axial dispersion coefficient
D_p	Pore diffusion coefficient
D_s	Surface diffusion coefficient
f_{bind}	Adsorption model
k_f	Film mass transfer coefficient
L	Column length
N_c	Number of components
N_d^b	Bulk phase polynomial degree
N_d^p	Particle pore phase polynomial degree
N_e^b	Number of bulk phase elements
N_{cycle}	Number of cycles
O	Connection matrix
Q	Liquid volumetric flow rate
r	Radial coordinate
R_c	Impenetrable particle core radius
R_p	Particle radius
t	Time
T_{end}	Simulation end time
u	Interstitial velocity
z	Axial coordinate

semi-continuous and continuous chromatography setups have more design variables, such as multiple inlet flow rates, outlet flow rates, connectivity, inlet modifier concentrations and others. These variables must be designed and optimized to achieve the objectives such as sufficient purity, increased product yield, enhanced capacity utilization, lower utility consumption, and more. Solving such optimization problems with numerous design variables is computationally intensive, both due to the high cost of each individual simulation and the large number of simulations required to explore the design space effectively.

Simulating SMB chromatography can be a particular challenge due to many coupled columns and extended runtime time until CSS is reached. While shortcut design methods exist that can guide the design and operation (Frandsen et al., 2023b,a; Migliorini et al., 2000; Lee et al., 2024), they do not account for important phenomena such as axial dispersion, mass transfer resistance and pore diffusion, making them inaccurate for capturing the actual SMB chromatography dynamics. These shortcut design methods can provide helpful initial values for numerical optimization with more comprehensive models (Li et al., 2007). However, fast chromatography model solvers are instrumental for solving the coupled and generally stiff and nonlinear (due to adsorption) PDE.

Even the most simple setup with only a single chromatography column includes additional unit operations such as system dead volumes in tubing, valves, and pumps, as well as back-mixing effects from components like tubing, bends, valves and more (Schultze-Jena et al., 2017; Schweiger and Jungbauer, 2018). A common approach to account for these phenomena is to model them as a network of Ideal plug flow (IPF) and Continuously stirred tank reactor (CSTR), respectively. The IPF introduces a lag to represent the dead volume, while the CSTR models mixing by assigning an appropriate volume (Carta and Jungbauer, 2020; Kumar and Lenhoff, 2020; Shekhawat and Rathore, 2019). Therefore, modeling a single chromatography column process often requires being able to simulate a network of unit operations. For more complicated setups, such as multi-column setups, the modeling approach becomes even more intricate. Each column in a multi-column system may have its own distinct column characteristics such as column porosity, binding capacity as well as different dead-volumes and mixing effects between the columns. Moreover, for some multi-column chromatography setups such as SMB chromatography, the network topology changes during operation (Schmidt-Traub et al., 2020). Therefore, it is essential that suitable numerical simulation software is not only computationally efficient but also supports arbitrary networks of unit operations.

The compute time required to solve chromatography models highly depends on the spatial discretization scheme. Previous works have demonstrated that implementations of the DGSEM (Hesthaven and Warburton, 2002; Cockburn et al., 2000) can very efficiently solve a variety of well-known chromatography models (Meyer et al., 2020; Breuer et al., 2023; Frandsen et al., 2025b; Javeed et al., 2011; Khan et al., 2021; Zafar et al., 2023, 2021). In a recent benchmark study, the DGSEM was implemented in the programming languages C++ and Julia to solve batch chromatography models. In this work, the C++ implementation as part of CADET-Core (Breuer et al., 2023), is referred to as DG-C++, while the Julia implementation, part of the CADET-Julia code base (Frandsen et al., 2025b), is referred to as DG-Julia. The performance of these implementations was compared across various batch chromatography case studies. CADET serves as an umbrella organization for a suite of tools in biotechnology process modeling and simulation. While CADET-Core is a broad and extensive

C++ code base with both the DGSEM and the Finite volume (FV) method (referred to as FV-C++) implemented, CADET-Julia is a slim code base entirely written in the programming language Julia with only the DGSEM implemented (Breuer et al., 2023; Frandsen et al., 2025b). The benchmarks showed that the DG-Julia implementation was faster but DG-C++ scaled better with the DoF (Frandsen et al., 2025b). Both approaches were generally superior to the FV-C++ implementation.

If information on the startup phase of the process in not required, the compute time of continuous chromatography setups such as SMB chromatography can be decreased even further by using Fixed-point iteration (FPI) algorithms for computing the CSS (He et al., 2018; Abunasser et al., 2003; Abunasser and Wankat, 2004; Mota and Araújo, 2005). This weak coupling approach enables the simulation of multi-column setups using a single column solver. Different weak-coupling algorithms for solving the CSS were tested, and it was shown that the One-column analog (OCA) - FPI most efficiently simulates the CSS (He et al., 2018). The OCA-FPI approach is based on the concept that, during CSS, the column experiences the same system state at every position, with a time lag. In the numerical approach, a single column is cycled through all the positions of the chromatography setup while concentration profiles at the inlet and outlet ports are stored and iteratively updated (He et al., 2018).

Despite studies comparing various weak coupling methods (He et al., 2018), no study had yet compared strong coupling with weak coupling approaches. Additionally, the weak coupling approaches have not been tested for more comprehensive models such as the LRMP and GRM. Likewise, the methods have not been implemented using DGSEM as discretization scheme. This study fills the gap by first introducing and verifying arbitrary networks of unit operations as well as investigating the most efficient approach to simulate continuous chromatography setups, both from startup and until CSS. It focuses on SMB chromatography as a multi-column system with particularly high computational demands. Considering different case studies of varying complexity, the computational performance of DG-C++, DG-Julia and FV-C++ is compared in extensive benchmarks. For the startup benchmarks, the setups are simulated for a certain number of cycles at varying discretization settings, comparing DG-C++, DG-Julia and FV-C++. For the CSS benchmarks, the setups are simulated until CSS is achieved, comparing the DG-C++, DG-Julia and FV-C++ as well as the OCA-FPI implementation using the DGSEM in CADET-Julia. Finally, performance differences are rigorously discussed.

2. Modeling

2.1. General rate model

Chromatography models are modular with respect to transport and adsorption (binding), meaning any transport model can be paired with any adsorption model. In this study, we examine the three most commonly used transport models in chromatography: GRM, LRMP, LRM. Note that the LRM corresponds to the IPF without binding and dispersion. Given the extensive documentation of these models in the literature (Guiochon et al., 2006; Gu, 2015; Schmidt-Traub et al., 2020), we only provide a brief overview of the GRM.

Within the interstitial column volume, the GRM accounts for convection and dispersion of the bulk concentrations c_i^b in axial direction $z \in (0, L)$ as well as film diffusion into the particle pores, with particle liquid concentration c_i^p, for each component $i \in \{1, \dots, N_c\}$ where N_c is the number of components. These mass transfer effects are governed by the equation (Guiochon et al., 2006; Frandsen et al., 2025b)

$$\frac{\partial c_i^b}{\partial t} = -u\frac{\partial c_i^b}{\partial z} + D_{ax,i}\frac{\partial^2 c_i^b}{\partial z^2} + \frac{(1-\varepsilon_c)}{\varepsilon_c} k_{f,i}\frac{3}{R_p}(c_i^b - c_i^p|_{r=R_p}) \tag{1}$$

in $(0, T_{\text{end}}) \times (0, L)$ with Danckwerts boundary conditions

$$u c_{\text{in},i} = \left(u c_i^b - D_{\text{ax},i}\frac{\partial c_i^b}{\partial z}\right)\bigg|_{z=0} \qquad \text{on } (0, T_{\text{end}}), \tag{2a}$$

$$0 = -D_{\text{ax},i}\frac{\partial c_i^b}{\partial z}\bigg|_{z=L} \qquad \text{on } (0, T_{\text{end}}), \tag{2b}$$

where $t \in [0, T_{end}]$ is time, T_{end} is the end time, L is the length of the column. The film mass transfer coefficient is denoted by $k_{f,i}$, D_{ax} is the axial dispersion coefficient and $c_{\text{in},i}$ denotes the column inlet concentrations.

In the particles, diffusion–reaction equations hold in $(0, T_{\text{end}}) \times (0, L) \times (R_c, R_p)$:

$$\frac{\partial c_i^p}{\partial t} + \frac{(1-\varepsilon_p)}{\varepsilon_p}\frac{\partial c_i^s}{\partial t} = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 D_{p,i}\frac{\partial c_i^p}{\partial r}\right) + \frac{(1-\varepsilon_p)}{\varepsilon_p}\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 D_{s,i}\frac{\partial c_i^s}{\partial r}\right), \tag{3}$$

$$\frac{\partial c_i^s}{\partial t} = f_{\text{bind}}(c_0^p, \dots, c_{N_c}^p, c_0^s, \dots, c_{N_c}^s) \tag{4a}$$

Here, c_i^s is the stationary particle phase concentration of component i, $r \in [R_c, R_p]$ is the radial particle coordinate, $R_p > R_c$ is the particle radius and $R_c \geq 0$ is the impenetrable particle core radius, ε_p is the particle porosity, D_p is the pore diffusion coefficient, and f_{bind} is defined by the adsorption model. We note that Eq. (4a) is used when adsorption is assumed to be limited by kinetics. This requires that the system is discretized and solved as a system of Ordinary Differential Equations (ODE). If adsorption happens practically instantaneously, rapid equilibrium adsorption can be assumed, given by Eq. (4b), and the problem is solved as a system of Differential Algebraic Equations (DAE).

$$f_{\text{bind}}(c_0^p, \dots, c_{N_c}^p, c_0^s, \dots, c_{N_c}^s) = 0. \tag{4b}$$

The complementing adsorption model defines f_{bind}, and in this work, we considered the Langmuir isotherm, the SMA and the linear isotherm. For description and mathematical formulation of the isotherms, we refer to existing literature, e.g. Schmidt-Traub et al. (2020). The rapid equilibrium can be approximated by multiplying f_{bind} by a large value, e.g. 10^8, which has been used previously and allowed the system to be solved as a system of ODE (Frandsen et al., 2025b).

The boundary conditions for Eq. (3) are given on $(0, T_{\text{end}}) \times (0, L)$ by

$$\left(\varepsilon_p D_{p,i}\frac{\partial c_i^p}{\partial r} + (1-\varepsilon_p) D_{s,i}\frac{\partial c_i^s}{\partial r}\right)\bigg|_{r=R_p} = k_{f,i}\left(c_i^b - c_i^p\big|_{r=R_p}\right), \tag{5a}$$

$$\left(\varepsilon_p D_{p,i}\frac{\partial c_i^p}{\partial r} + (1-\varepsilon_p) D_{s,i}\frac{\partial c_i^s}{\partial r}\right)\bigg|_{r=R_c} = 0. \tag{5b}$$

2.2. Network of unit operations

To simulate arbitrary process setups, the network of unit operations can be represented as a graph. In this representation, the nodes correspond to unit operations, and the edges represent the connections between them. When one unit operation is connected to another, fluid flows between them with a specific flow rate and composition (Leweke, 2021). Consequently, the inlet concentrations and flow rates for any unit operation are determined through mass balances, which additionally account for various sink and source terms from both inlets and outlets.

To account for varying flow rates and inlet concentrations, model setups and their simulation can be partitioned into distinct sections. These sections correspond to different steps of a chromatography process, such as loading, washing and eluting, where the inlet concentrations and the flow rates may vary over time while the unit operation network remains unchanged. However, in some chromatography processes, the network of unit operations, i.e. the connections between the different unit operations, changes dynamically during operation. To manage these changes, the simulation framework must support flexible configurations for connections, as well as variable flow rates and inlet concentrations. To model these dynamic systems, the connections between components of unit operations are represented by a connection matrix O, and each configuration is defined in a network switch. If component i of unit operation j is connected to component i' of unit operation j' during network switch s, then $O_{(i,j,s),(i',j',s)} = 1$ and $O_{(i,j,s),(i',j',s)} = 0$ otherwise. Each connection between two unit operations is associated with a volumetric flow rate stored in a flow rate matrix Q. If there is a connection from unit operation j to unit operation j' during section k, which is part of network switch s, then $Q_{(j,k),(j',k)} \geq 0$ and $Q_{(j,k),(j',k)} = 0$ otherwise.

With that, the inlet concentration of component i' entering a unit operation j' during section k, which is part of network switch s, can be determined via

$$c_{in,i',j',k} = \frac{\sum_{j=1}^{N_j} Q_{(j,k),(j',k)} O_{(i,j,s),(i',j',s)} c_{out,i,j}}{\sum_{j=1}^{N_j} Q_{(j,k),(j',k)}}, \tag{6}$$

where N_j is the number of unit operations in the network and $c_{out,i,j}$ is the outlet concentration of component i from unit operation j. Flow rates within a section and network switch may vary over time, and can be approximated by piecewise polynomials (Leweke and von Lieres, 2018; Leweke, 2021). If there is no flow into a unit operation i.e. $\sum_{j=1}^{N_j} Q_{(j,k),(j',k)} = 0$, then $c_{in,i',j',k} = 0$. The Eq. (6) also accounts for source terms i.e. inlet terms, where $c_{out,i,j}$ represents the source inlet concentration. Finally, for chromatography columns, the interstitial velocity can be determined through

$$u = \frac{\sum_{j=1}^{N_j} Q_{(j,k),(j',k)}}{A_c \varepsilon_c} \tag{7}$$

where A_c is the cross section area of the chromatography column.

3. Methods

To discretize the Partial differential algebraic equations (PDAE) systems defined in Section 2, we apply the so-called method of lines. That is, we separately discretize the equations in space and time, whereby we draw on readily developed and implemented numerical methods detailed in our previous works (Frandsen et al., 2025b; Breuer et al., 2023; Leweke and von Lieres, 2018; von Lieres and Andersson, 2010).

In this work, we compare three implementations from two code bases, each featuring different numerical algorithms. Table 1 summarizes the most important differences between DG-Julia, DG-C++ and FV-C++.

For the sake of completeness, we provide a brief description of the spatial and temporal discretization methods. Subsequently, we detail extensions made in this work.

3.1. Spatial and temporal discretization

DG-C++ and DG-Julia implement the DGSEM variant derived by Breuer et al. (2023) to spatially discretize the bulk phase Eq. (1). The pore phase equation is spatially discretized using the Galerkin spectral method (GSM) derived by Meyer et al. (2020) and extended to allow for impenetrable particle cores (Frandsen et al., 2025b). FV-C++ implements a Weighted essentially non oscillatory (WENO) FV method to spatially discretize the equations (Leweke and von Lieres, 2018).

Additionally, it was shown that it is faster to solve the chromatography models as systems of PDE (as opposed to PDAE) in DG-Julia by approximating the rapid-equilibrium using a large-value kinetic constant (Frandsen et al., 2025b). Thus, this approach of solving the system is used in this work.

To integrate the semi-discretized system in time, we make use of available, efficient and well-established methods: The DG-C++ and FV-C++ implementations use an adaptive time step, variable order Backwards differentiation formula (BDF) method implemented in the IDA solver of the SUNDIALS software package (Hindmarsh et al., 2005; Gardner et al., 2022). In DG-Julia, we have shown that the QNDF solver is the most effective at solving the chromatography problems using the *DifferentialEquations.jl* package (Rackauckas and Nie, 2017; Frandsen et al., 2025b). The QNDF is an adaptive order quasi-constant time step numerical differentiation formula Numerical differentiation formula (NDF) method which uses Shampines accuracy-optimal kappa values (Shampine et al., 1997). To solve the nonlinear system during time integration, the FV-C++ code utilizes a full Newton method, while both DG-C++ and DG-Julia use a pseudo Newton method. These nonlinear solvers prove to be computationally most efficient for the respective discretization methods (Frandsen et al., 2025b).

3.2. One-column analog fixed-point iteration (OCA-FPI)

The OCA-FPI method is a weak-coupling method to determine the CSS of cyclic processes such as SMB chromatography processes (He et al., 2018; Abunasser et al., 2003; Abunasser and Wankat, 2004; Mota and Araújo, 2005). The concept of a SMB chromatography process is to approximate a true moving bed, where the solids and fluids are moving counter-currently (Schmidt-Traub et al., 2020). In a closed-loop SMB chromatography process, at least four columns are connected in a circular arrangement between the four ports of the true moving bed, namely the feed, desorbent, extract and raffinate ports (He et al., 2018). To simulate the counter-current movement of the solids, the positions of the columns in the network are periodically shifted counter-currently to the fluid direction. Practically, the physical movement of the columns is avoided by rotating all positions of the ports in the direction of the fluid flow using special valves (Schmidt-Traub et al., 2020). After several cycles, the SMB chromatography process reaches a CSS. For more details on SMB chromatography, we refer to existing literature, e.g. Schmidt-Traub et al. (2020).

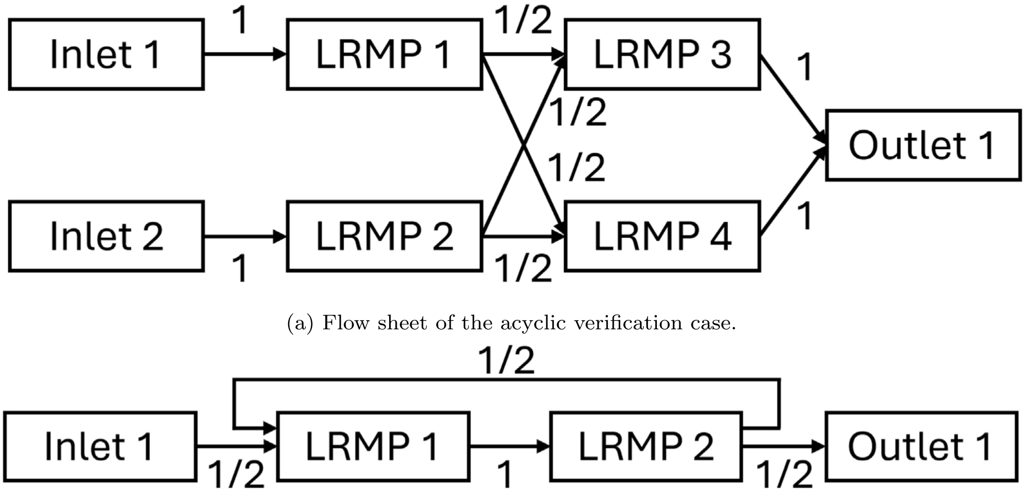
Various weak coupling approaches, that do not follow the physical trajectory to directly compute the CSS of SMB chromatography processes, have already been evaluated (He et al., 2018). Of the tested approaches, He et al. (2018) showed that the OCA-FPI was the most efficient. Therefore, only the OCA-FPI approach was implemented and applied in our benchmarks. The concept of the OCA-FPI approach is that the column will experience the same system state with a time lag at different positions of the network during CSS. This can be used to simulate the CSS using just a single column, while storing inlet and outlet concentration profiles at all ports. The OCA-FPI approach utilizes the fact that the outlet concentration profile of one column equals the inlet concentration profile of the next column in counter-clock wise direction. Thus, the single column is circulated through the column positions such that the initial state (initial conditions) at the new position is the final state at the previous position. Initially, the column is assumed to be filled with pure solvent, i.e., no concentration of solutes in all column positions. The outlet concentration profile of the column is interpolated using piece-wise polynomials and used as inlet boundary condition to simulate the next column position. The inlet and outlet concentration profiles at each column position are stored. Once the column has been simulated in all column positions, this constitutes a whole cycle, and the convergence can be evaluated via the difference of inlet and outlet concentration profiles before and after that cycle. More details can be found in He et al. (2018).

3.3. Evaluation metrics

For each startup and CSS case study, a reference solution was computed using DG-Julia with a very high resolution i.e. sufficiently many DGSEM elements and high-order polynomials. More details can be found in Section 6. From this, we computed the MAE and the L_2 norm for each computational grid refinement. For the startup benchmarks, the MAE and L_2 norm were determined at the end of the simulation, whereas the MAE and L_2 norm were determined at each cycle for

Table 1
Summary of the most important differences between CADET-DG (DG-C++), CADET-FV (FV-C++) and CADET-Julia (DG-Julia).

Code Base	CADET-Core		CADET-Julia
Spatial discretization	DGSEM	FV	DGSEM
Programming language	C++	C++	Julia
Problem type	PDAE	PDAE	PDE
Time integrator	Sundials IDAS	Sundials IDAS	QNDF
Nonlinear solver	Quasi Newton method	Full Newton method	Quasi Newton method
Linear Solver	LU Factorization	LU Factorization	Chosen automatically:Problem dependent
Alias	DG-C++	FV-C++	DG-Julia
Github Repository	cadet/CADET-Core		cadet/CADET-Julia
Git Commit Hash	1572ca6		25ebd2e



(a) Flow sheet of the acyclic verification case.

(b) Flow sheet of the cyclic verification case.

Fig. 1. Flow sheets of the verification cases. The edge weights indicate the flow rate relative to the total flow rate fed into the system. The parameters for the verification studies are defined in Table S1.

the CSS benchmarks. For the CSS case studies, the simulations were stopped when the L_2 norm of the concentrations during one cycle at the raffinate (R) and extract (E) ports went below a specified CSS tolerance ε_{CSS}, compared to the previous cycle i.e.

$$\sum_{\kappa\in\{E,R\}} \sum_{i=0}^{N_c} \|c_{i,\kappa}(t_0 : t_1) - c_{i,\kappa}(t_{-1} : t_0)\|_{\mathcal{L}^2} < \varepsilon_{CSS}. \tag{8}$$

Here, $c(t_0 : t_1)$ are the concentrations in the time interval of t_0 to t_1, $t_0 = (N_{cycle} - 1)t_{cycle}$, $t_1 = N_{cycle}t_s$, $t_{-1} = (N_{cycle} - 2)t_s$ where N_{cycle} is the number of cycles that has been simulated and t_{cycle} is the time it takes to cycle through all the columns.

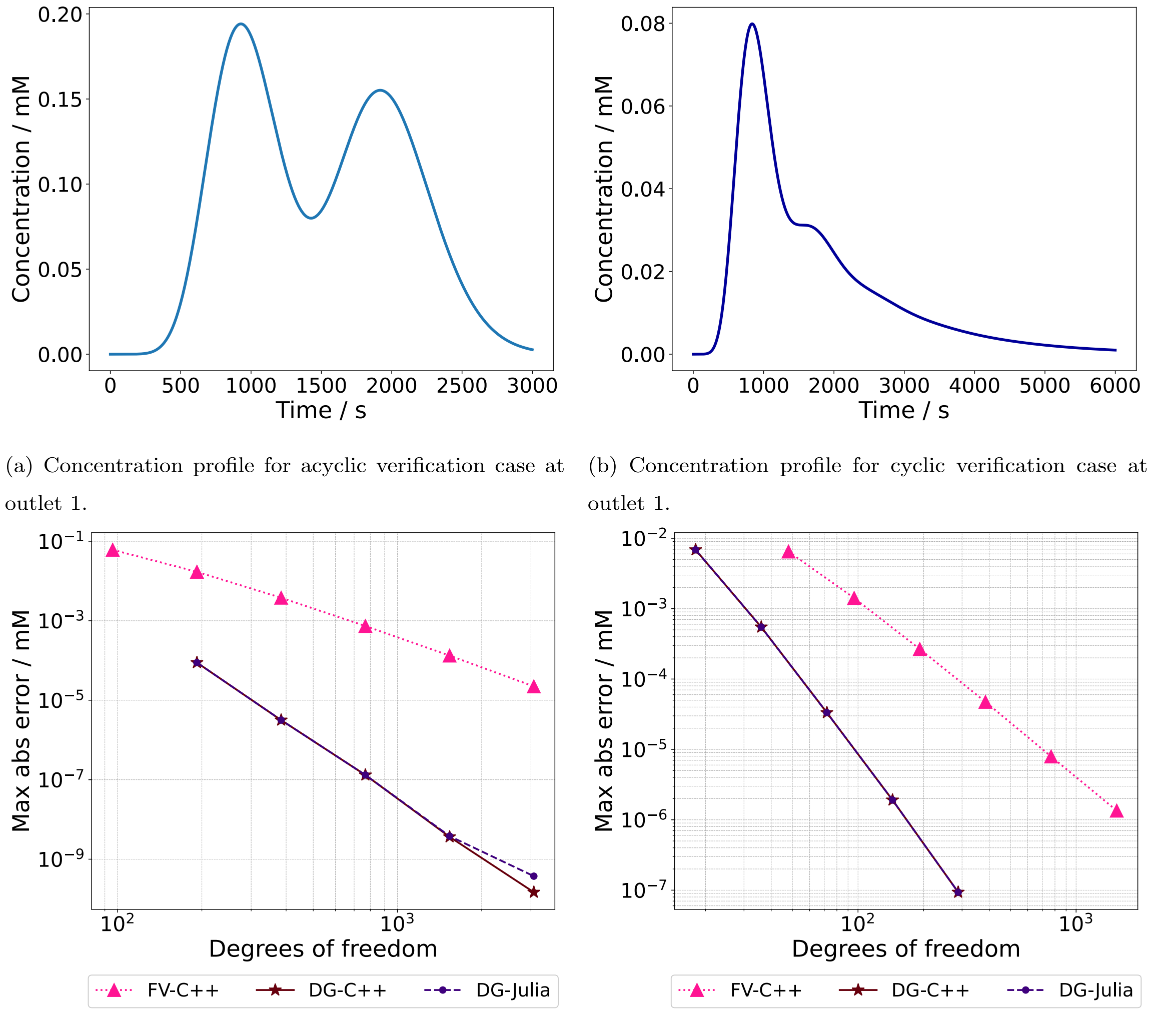
4. Software verification

To verify the modular implementation of arbitrary networks of unit operations, we conduct Experimental order of convergence (EOC) tests against semi-analytical reference solutions. These reference solutions were computed using an independent solver and software, CADET-Semi-Analytic, which is based on numerical inversion of a Hankel–Laplace domain solution (Leweke and von Lieres, 2016; Leweke, 2024). That is, by passing the test, the three codes DG-C++, DG-Julia and CADET-Semi-Analytic mutually verify each other. We consider an acyclic and a cyclic verification case defined by the flow sheets shown in Fig. 1 and parameters shown in Table S1. Since the column models, i.e. the LRM, the LRMP and the GRM, for a single column have already been verified previously (Breuer et al., 2023; Frandsen et al., 2025b), the LRMP was used as column model in order to decrease the computational burden.

For the acyclic case, two concentration pulses of different durations were imposed as inlets to generate a feature-rich output. The inlet 1 used rectangular pulse of 1 mM from 0 to 300 s whereas inlet 2 used a rectangular pulse of 5 mM from 250 to 300 s. The simulation was stopped after 3000 s. For the cyclic case, a 1 mM rectangular concentration pulse lasting 300 s was imposed as inlet, and the simulation was terminated after 6000 s (Leweke, 2021).

For the DGSEM approach in DG-Julia and DG-C++, a third order bulk phase polynomial degree was used with Discontinuous Galerkin (DG) elements gradually refined from 4 to 64 with double increment for the acyclic case. For the cyclic case, a second order bulk phase polynomial degree was used with DG elements gradually refined from 1 to 16 with double increment. For the FV discretization, the elements were gradually refined from 8 to 256 with double increment for both verification cases. The problems were solved using relative and absolute time integrator tolerances of $1{\cdot}10^{-10}$ and $1{\cdot}10^{-12}$, respectively, such that the spatial discretization error dominates. The DoF and MAE for the different spatial methods and implementations are tabulated in Tables 2 and 3 and visualized together with the semi-analytic concentration profiles at outlet 1 in Fig. 2.

Fig. 2 and Tables 2 and 3 demonstrate that both DGSEM and FV methods converge towards the semi-analytic solution and exceed the theoretical order of convergence until the considered numerical tolerance is reached. The DG-C++ implementations converges rapidly with an average EOC of 4.41. As the only discrepancy between MAEs of DG-C++ and DG-Julia occurs in the acyclic case when the MAE approaches the relative integration tolerance of $1 \cdot 10^{-10}$, both implementations of arbitrary networks of unit operations are verified. For the FV-C++ implementation, the EOC is second-order which is expected for the discretization method (Leweke and von Lieres, 2018). The evaluation code for software verification, including model setups and



(a) Concentration profile for acyclic verification case at outlet 1.

(b) Concentration profile for cyclic verification case at outlet 1.

(c) Convergence for the acyclic verification case.

(d) Convergence for the cyclic verification case.

Fig. 2. Reference profiles computed with CADET-Semi-Analytic for outlet 1 of the cyclic and acyclic verification cases in (a)–(b) and convergence tests in terms of MAE and DoF for the verification studies for FV-C++, DG-Julia and DG-C++ in (c)–(d). The number of bulk phase elements was gradually increased. The parameters for the verification studies can be found in Table S1. Note that the DG-C++ and the DG-Julia lines are overlapping.

Table 2
Convergence table of DG-C++ for the acyclic verification case depicted in Fig. 1(a). The bulk phase polynomial degree and number of elements of the DG method are denoted as N_d^b and N_e^z, respectively. The parameters are defined Table S1.

N_d^b	N_e^b	MAE	MAE EOC	L^2 error	L^2 EOC	Bulk DoF
3	4	$8.84 \cdot 10^{-5}$	––	$1.74 \cdot 10^{-3}$	––	192
3	8	$3.16 \cdot 10^{-6}$	4.81	$6.27 \cdot 10^{-5}$	4.79	384
3	16	$1.31 \cdot 10^{-7}$	4.59	$2.65 \cdot 10^{-6}$	4.57	768
3	32	$3.56 \cdot 10^{-9}$	5.21	$7.22 \cdot 10^{-8}$	5.20	1,536
3	64	$1.57 \cdot 10^{-10}$	4.51	$3 \cdot 10^{-9}$	4.59	3,072

convergence analysis, is publicly available in the CADET-Verification test suite (Breuer et al., 2025).

5. Results and discussion

In the following case studies, we investigate SMB chromatography case studies with 4–8 columns, considering the LRM, the LRMP and the GRM with the linear, Langmuir and SMA isotherms to cover a wide range of linear and nonlinear settings for all three transport models. An overview of the case studies is shown in Table 4.

Table 3
Convergence table of DG-C++ for the cyclic verification case depicted in Fig. 1(b). The bulk phase polynomial degree and number of elements of the DG method are denoted as N_d^b and N_e^b, respectively. The parameters are defined in Table S1.

N_d^b	N_e^b	MAE	MAE EOC	L^2 error	L^2 EOC	Bulk DoF
2	1	$6.86 \cdot 10^{-3}$	––	0.14	––	18
2	2	$5.48 \cdot 10^{-4}$	3.65	$9.98 \cdot 10^{-3}$	3.78	36
2	4	$3.33 \cdot 10^{-5}$	4.04	$6.36 \cdot 10^{-4}$	3.97	72
2	8	$1.91 \cdot 10^{-6}$	4.12	$3.72 \cdot 10^{-5}$	4.10	144
2	16	$9.34 \cdot 10^{-8}$	4.35	$1.83 \cdot 10^{-6}$	4.34	288

Table 4
Overview of SMB chromatography case studies.

Case study	Transport model	Isotherm	# Columns	N_c
1	LRM	Linear	8	2
2	LRMP	Langmuir	4	2
3	LRMP	SMA	4	4
4	GRM	SMA	4	4

Case study 1 features an 8-column SMB chromatography system, where the LRM with linear isotherm is incorporated to model the columns. Case study 2 examines a 4-column SMB chromatography system that employs the LRMP and the Langmuir isotherm. Case studies

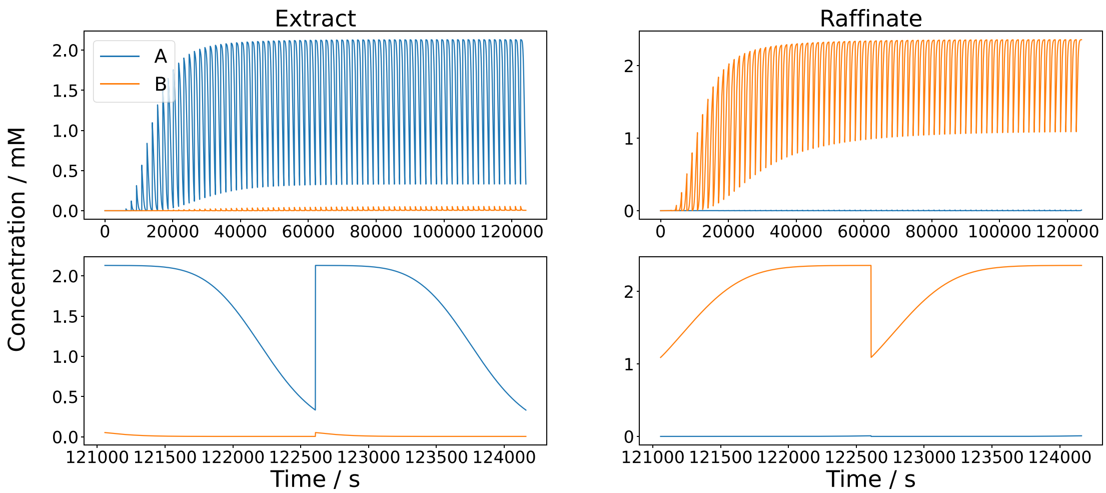


Fig. 3. Concentration profiles for the first 10 cycles (upper figures) and the last two switches (lower figures) for the extract (left figures) and raffinate (right figures) ports of case study 1: LRM with linear binding, parameters are given in Table S2.

3 and 4 both involve a 4-column SMB chromatography system with the SMA isotherm, using the LRMP and the GRM, respectively. The parameters for the case studies are shown in the supplementary material Table S1 (He et al., 2018; Breuer et al., 2023; Püttmann et al., 2013).

As in the verification cases, all problems were solved using relative and absolute time integrator tolerances of $1 \cdot 10^{-10}$ and $1 \cdot 10^{-12}$, respectively. Breuer et al. (2023) showed that the case studies considered here were simulated most efficiently using a minimal polynomial degree of four and the collocation DGSEM variant for the bulk phase. We follow these results for all our case studies and only consider the collocation DGSEM with polynomial degree ≥ 4. Throughout all benchmarks, we vary the number of DGSEM elements, bulk phase polynomial order (≥ 4) and pore phase polynomial order (≥ 4) to investigate numerical convergence and computational performance. The evaluation code was published on Zenodo (Frandsen et al., 2025a) to make our studies and results fully transparent and reproducible. All simulations were conducted on a Dell Latitude 7310 with an Intel(R)-Core TM-i5-10310U 1.70 GHz processor and 16 GB RAM. The reported computational performances were obtained using specific Git commits from codebases that are under active development. Beyond the algorithmic differences, performance can also be influenced by factors such as compiler version, compilation flags (e.g., optimization level), and linked libraries, all of which may vary between builds. Additionally, benchmarking results depend on the underlying hardware; differences in CPU architecture, memory bandwidth, and other system characteristics can lead to variations in compute time.

5.1. Startup SMB chromatography benchmarks

In this section, the SMB chromatography models were solved from startup to a fixed number of cycles. For case study 1, which employs a LRM with linear isotherm considering 8 columns, the model was solved until 10 cycles were completed using different discretization options. In Fig. 3, the concentration profiles are shown for the high-resolution reference solution.

Fig. 3 shows the concentration profiles for the first 10 cycles for case study 1 as well as the last two switches. For the first few switches, the solutes have not yet reached the raffinate and extract outlets, meaning the concentrations of components A and B are zero. As the number of switches increase, the solutes reach the outlets, resulting in concentrations of components A and B in the outlets. The sudden change in outlet concentrations is due to the port switching. After more switches, the concentration profiles of component A and B in the outlets become periodic as they converge towards CSS. In this particular case study, the feed mixture is separated well as component A is primarily in the extract and component B is primarily in the raffinate. The concentration profiles for case study 2, 3 and 4 are shown in Supplementary Material Figures S1 to S3. The compute times and MAEs for case study 1 are shown in Fig. 4.

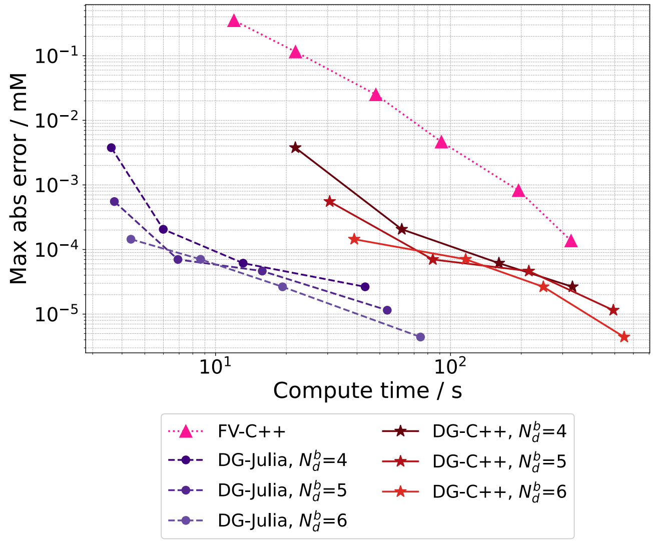


Fig. 4. Compute time and MAE comparison between FV-C++, DG-Julia and DG-C++ evaluated for various bulk phase polynomial degrees, N_d^b. The number of bulk phase elements was gradually increased (doubled, starting with 2 elements). The case study is LRM with the linear isotherm (case study 1) with parameters in Table S2.

The benchmark in Fig. 4 showcases that DG-Julia is significantly faster than FV-C++. Comparing DG-Julia and DG-C++, they reach the same MAEs but DG-Julia is consistently more efficient. This difference in performance for solving the LRM was also observed in previous batch chromatography benchmarks (Frandsen et al., 2025b). Interestingly, the convergence seems to slow down at around MAE less that 10^{-4} mM for both DG-Julia and DG-C++.

For case study 2, the compute times and MAEs are shown in Fig. 5. The corresponding concentration profiles are shown in Figure S1.

The benchmark in Fig. 5 shows that both DG-Julia and DG-C++ are significantly faster that FV-C++. Comparing DG-Julia and DG-C++, they both reach the same MAEs, however, the difference in compute time between DG-Julia and DG-C++ is less predominant. Furthermore,

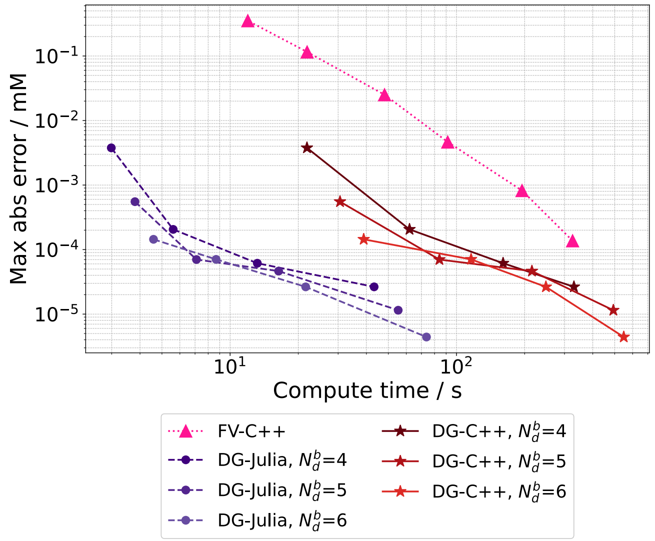


Fig. 5. Compute time and MAE comparison between FV-C++, DG-Julia and DG-C++ evaluated for various bulk phase polynomial degrees, N_d^b. The number of bulk phase elements was gradually increased (doubled, starting with 2 elements). The case study is LRMP with the Langmuir isotherm (case study 2) with parameters in Table S2.

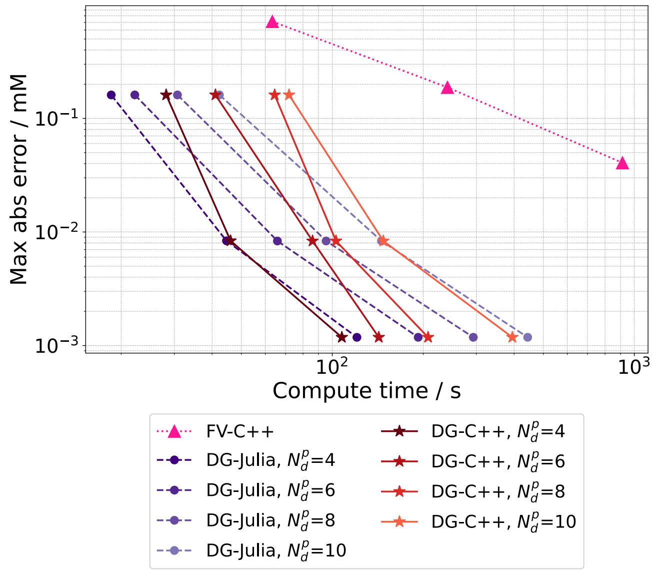


Fig. 6. Compute time and MAE comparison between FV-C++, DG-Julia and DG-C++ evaluated for various particle phase polynomial degrees, N_d^p, and fourth order bulk phase polynomials. The number of bulk phase elements was gradually increased (doubled, starting with 1 element). The case study is GRM with the SMA isotherm (case study 4) with parameters in Table S3.

DG-C++ scales better than DG-Julia as indicated by the slope of the MAE curves. The same trends are observed for case study 3 shown in Supplementary Material Figure S4.

For case study 4, the benchmark are shown in Fig. 6.

The benchmark in Fig. 6 shows that DG-Julia and DG-C++ are significantly faster that FV-C++. Comparing DG-Julia and DG-C++, they both reach the same MAEs, however, DG-Julia is generally faster. Once again, DG-C++ scales better than DG-Julia as indicated by the slope of the MAE curve. As the number of DGSEM elements is increased, the state vector and number of arithmetic operations increase. Here, DG-C++ copes better with the increased state vector and number of computations, resulting in DG-C++ being faster than DG-Julia for some simulations. This scaling advantage of DG-C++ over DG-Julia was also observed in benchmarks of batch chromatography processes (Frandsen et al., 2025b). Furthermore, the MAEs are large compared to case studies 1,2 and 3, which aligns with previous batch chromatography benchmarks where the GRM combined with the SMA isotherm was investigated (Breuer et al., 2023; Meyer, 2020; Frandsen et al., 2025b).

Table 5
Summary of the discretization options for the four different case studies and the average (Avg.) speed up evaluated for all the discretization settings of DG-Julia compared to DG-C++ for the startup SMB chromatography benchmarks. N_e^b represents the number of bulk phase DG elements, N_d^b is the bulk phase polynomials degree, N_d^p is the pore phase polynomial degree and N_{cycles} is the number of cycles.

Case study	N_e^b	N_d^b	N_d^p	N_{cycles}	Avg. Speed up
1	[2,4,8,16]	[4,5,6]	–	10	10
2	[2,4,8,16,32]	[4,5,6]	–	20	1.9
3	[2,4,8]	[4,5,6]	–	20	1.5
4	[1,2,4]	4	[4,6,8,10]	20	1.4

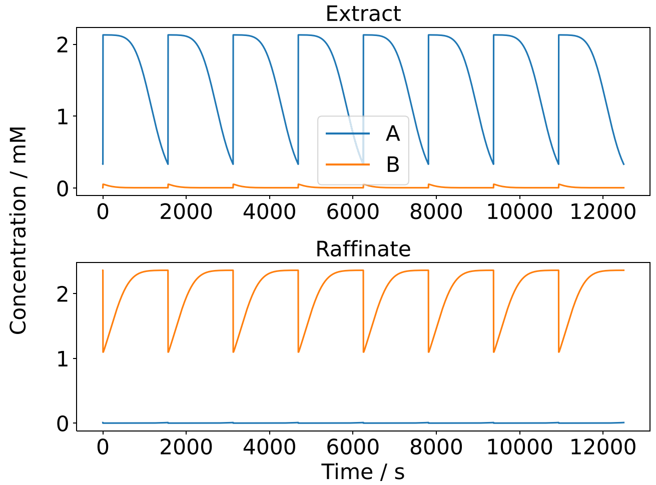


Fig. 7. CSS concentration profiles at the extract (upper figure) and raffinate (lower figure) for one whole cycle. The case study is LRM with the linear isotherm (case study 1) with parameters in Table S2.

A summary of the startup SMB chromatography benchmarks is shown in Table 5.

Table 5 shows that DG-Julia is generally faster at simulating the concentration profiles of the startup phase compared to DG-C++. The performance differences between DG-Julia and DG-C++ matches trends that were previously observed in a batch chromatography benchmarks (Frandsen et al., 2025b).

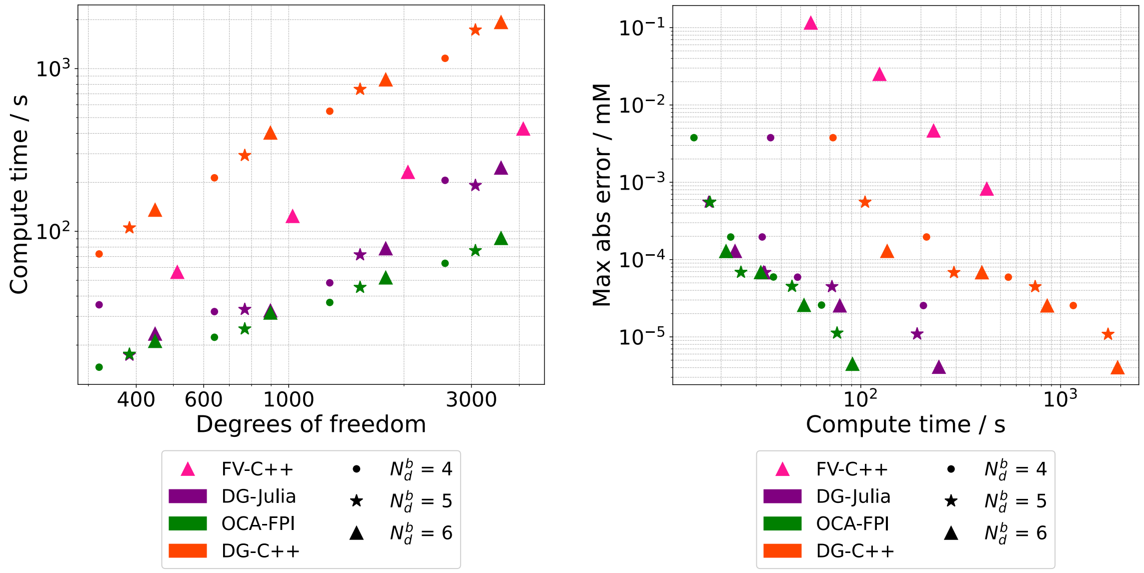
5.2. CSS smb chromatography benchmarks

In this section, simulations are continued until CSS. We note that the number of cycles may depend on the discretization as it directly influences the convergence towards CSS, which we specifically observed when few DGSEM elements were used. The OCA-FPI method was incorporated into the Julia DGSEM implementation and compared to the strongly coupled native approach in C++ and Julia for the DGSEM. For both approaches, benchmarks were evaluated with $\varepsilon_{\text{CSS}} = 1 \cdot 10^{-6}$.

For case study 1 with 8 columns, the CSS profiles at the raffinate and extract ports for one whole cycle are shown in Fig. 7.

The CSS concentration profiles for case studies 2, 3 and 4 can be found in Figures S5 to S7. For case study 1, the DoF, compute times and MAEs are shown in Fig. 8.

Fig. 8 shows that for the simulation with the fewest DoF, the strongly coupled approach using DG-Julia converges slowly to CSS. The reason is that it takes significantly more iterations to reach CSS for that specific simulation compared to the simulations with more



(a) Degrees of freedom and compute time comparison for the strongly coupled FV-C++, DG-Julia and DG-C++ approaches and the OCA-FPI approach in DG-Julia.

(b) Compute time and MAE comparison for the strongly coupled FV-C++, DG-Julia and DG-C++ approaches and the OCA-FPI approach in DG-Julia.

Fig. 8. Benchmarks evaluated for various bulk phase polynomial degrees, N_d^b. The number of bulk phase elements was gradually increased (doubled, starting with 2 elements). The case study is LRM with the linear isotherm (case study 1) with parameters in Table S2.

DoF. Furthermore, for increasing DoF, the advantage of using the OCA-FPI becomes evident as seen in Fig. 8(a) which shows that for larger systems with at least 400 DoF, the OCA-FPI reaches CSS faster than the strongly coupled approach. Note that this DoF threshold is only valid for this case study and might vary for other problems. The faster convergence of the OCA-FPI approach is also reflected in the MAE in Fig. 8(b) which shows that the OCA-FPI approach reaches the same MAEs slightly faster. Compared to FV-C++, the three other approaches are faster. Similar trends are also observed for case study 2, see Fig. 9, where the OCA-FPI approach is consistently faster for DoF larger than 960.

Fig. 9(a) shows that the OCA-FPI method solves the CSS system consistently faster for DoF larger than approximately 960 compared to the strongly coupled approach in DG-Julia. For DoF less than 400, the strongly coupled approach in DG-Julia is only slightly faster than the OCA-FPI approach. For discretization settings resulting in 400 and 960 DoF, the compute times are very similar for this case study. As a result, the CSS is achieved faster at the same MAE for case study 2 as shown in Fig. 9(b). Compared to the strongly coupled DG-C++ approach, OCA-FPI in DG-Julia is faster, which is discussed in detail at the end of this section. Compared to FV-C++, the performance difference for this case study is significant. At comparable MAE of approximately $7 \cdot 10^{-3}$ mM, FV-C++ is 52 times slower than the strongly coupled approach in DG-C++, 97 times slower than the strongly coupled approach in DG-Julia and 99 times slower than the OCA-FPI approach in DG-Julia.

In case studies 3 and 4, where the minimum DoF were 480 and 1760, respectively, the OCA-FPI approach was consistently the fastest method to reach the CSS across all discretization settings. Both the strongly coupled DG-Julia and OCA-FPI in DG-Julia are faster than DG-C++. For case studies 3 and 4, the compute time and MAE comparisons are shown in Figs. 10 and 11.

In Fig. 10, the more refined simulations result in lower MAEs but they consequently result in a higher compute times. The rates at which the problems are solved at the resulting MAEs using different polynomial degrees, i.e. the slopes from the different polynomial degrees in Fig. 10, vary and depend on the specific problem. In Fig. 10, the rates from the different polynomial degrees slightly differ between them and are significantly faster than the rate of the FV-C++.

Table 6
Summary of the discretization options for the four different case studies and the average (Avg.) speed up evaluated for all the discretization settings of DG-Julia using the strongly coupled approach compared to DG-Julia using the OCA-FPI approach (OCA-FPI) and DG-C++ using the strongly coupled approach for the CSS SMB chromatography benchmarks. N_e^b represents the number of bulk phase DG elements, N_d^b is the bulk phase polynomials degree and N_d^p is the pore phase polynomial degree.

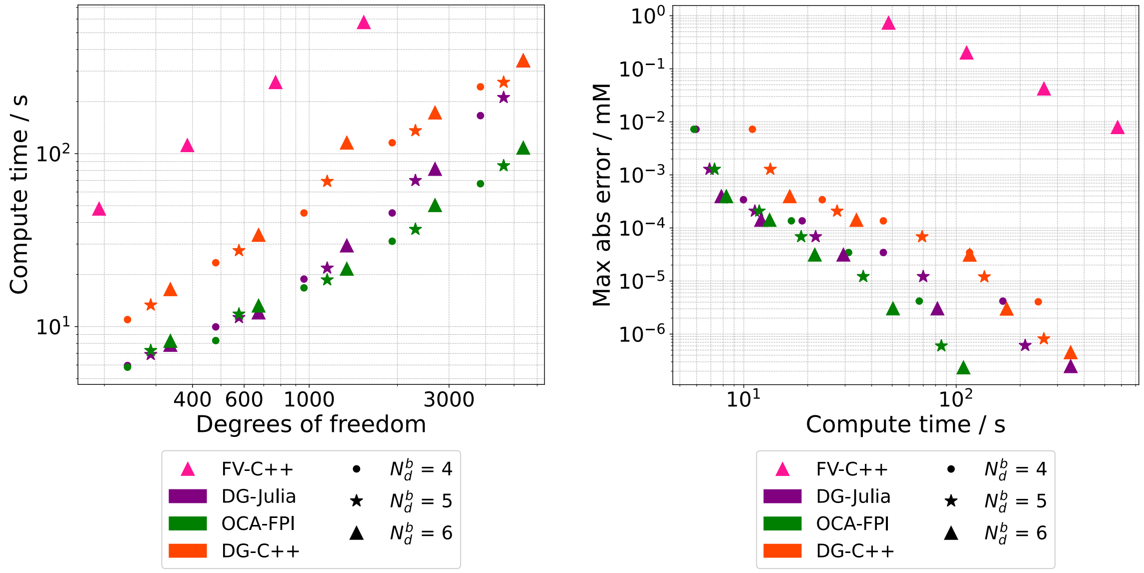
Case study	N_e^b	N_d^b	N_d^p	Avg. DG-Julia Speed up	
				vs. OCA-FPI	vs. DG-C++
1	[2,4,8,16]	[4,5,6]	–	0.66	8.1
2	[2,4,8,16,32]	[4,5,6]	–	0.77	2.2
3	[2,4]	[4,5,6]	–	0.80	1.6
4	[2,4]	4	[4,6]	0.49	1.4

For case study 4 in Fig. 11, the MAE is limited by the number of DGSEM elements which is why increasing the pore phase polynomial degree does not decrease the MAE.

To investigate how the OCA-FPI approach and the strongly coupled approach converge in DG-Julia, the L_2 norm is plotted alongside the compute time for case study 4 in Fig. 12.

In Fig. 12, the dashed and full lines are the strongly coupled and OCA-FPI approaches in DG-Julia, respectively, and the matching colors represent the same discretization options. The figure shows that for the strongly coupled approach, the L_2 norm decreases until the L_2 norm gets to around 10 where it plateaus for some time before the L_2 norm decreases again relatively fast. For the OCA-FPI approach, the L_2 norm decreases significantly faster. As a result, the OCA-FPI approach converges faster towards CSS compared to the strongly coupled approach for this case study. A summary of the CSS SMB chromatography benchmarks for case studies 1, 2, 3 and 4 is shown in Table 6.

Table 6 shows a consistent (when taking the average values over all discretizations) computational advantage of the OCA-FPI approach to reach the CSS compared to the strongly coupled approach in DG-Julia. Generally, this advantage increases with the size of the discretized



(a) Degrees of freedom and compute time comparison for the strongly coupled FV-C++, DG-Julia and DG-C++ approaches and the OCA-FPI approach in DG-Julia.

(b) Compute time and MAE comparison for the strongly coupled FV-C++, DG-Julia and DG-C++ approaches and the OCA-FPI approach in DG-Julia.

Fig. 9. Benchmarks evaluated for various bulk phase polynomial degrees, N_d^b. The number of bulk phase elements was gradually increased (doubled, starting with 2 elements). The case study is LRMP with the Langmuir isotherm (case study 2) with parameters in Table S2.

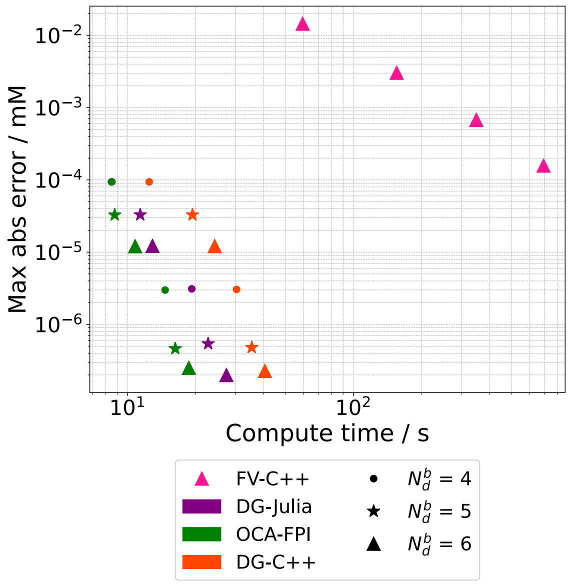


Fig. 10. Compute time and MAE comparison for the strongly coupled FV-C++, DG-Julia and DG-C++ approaches and the OCA-FPI approach in DG-Julia for case study 3 evaluated for various bulk phase polynomial degrees, N_d^b. The number of bulk phase elements was gradually increased (doubled, starting with 2 elements). The case study is the LRMP with the SMA isotherm (case study 3) with parameters in Table S3.

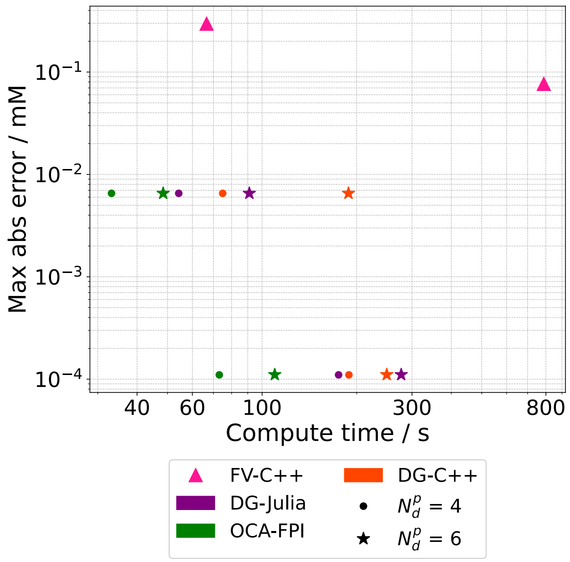


Fig. 11. Compute time and MAE comparison for the strongly coupled FV-C++, DG-Julia and DG-C++ approaches and the OCA-FPI approach in DG-Julia for case study 4 evaluated for various particle phase polynomial degrees, N_d^p, and fourth order bulk phase polynomials. The number of bulk phase elements was gradually increased (doubled, starting with 2 element). The case study is the GRM with the SMA isotherm (case study 4) with parameters in Table S3.

system of equations. Table 6 also shows that the OCA-FPI approach is faster than the strongly coupled approach using DG-C++.

The reason why the OCA-FPI approach is slightly slower than the strongly coupled approach for smaller DoF (less than approximately 400) can be explained by the additional pre-processing between each switch for the OCA-FPI approach. For each switch, the necessity to stop and start the simulation can cause a minor computational overhead. Additionally, the outlet flow must be interpolated using piece-wise polynomials, before it can be used as inlet concentration profile of the next column. These two factors can explain why the OCA-FPI approach slightly slower at low DoF. Contrary, for large systems, it is faster to simulate the CSS using the OCA-FPI approach in DG-Julia, partly because of the relatively poor scaling in DG-Julia. As seen for batch chromatography benchmarks (Frandsen et al., 2025b) as well as for the benchmarks presented in this study, DG-Julia does not scale as well with increasing problem size as DG-C++.

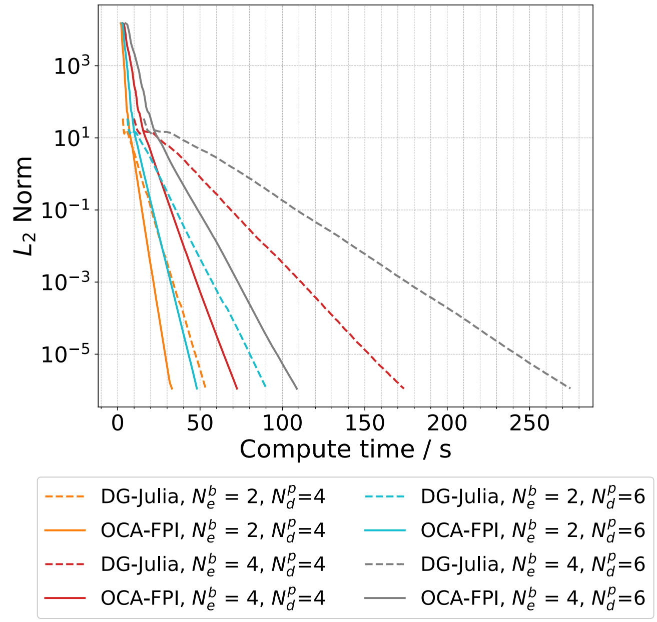


Fig. 12. L_2 norm and compute time comparison for the strongly coupled and OCA-FPI approaches using DG-Julia evaluated for various particle phase polynomial degrees, N_d^p, bulk phase elements and fourth order bulk phase polynomials. Note the y-axis scale is a $\log_{10}$ scale. The case study is the GRM with the SMA isotherm (case study 4) with parameters in Table S3.

Furthermore, the OCA-FPI approach offers a more sparse and smaller system Jacobian. The sparsity pattern of the Jacobian matrices for the strongly coupled approach and for the OCA-FPI approach are shown in Fig. 13 for case study 3.

Simulating multiple coupled columns, the Jacobian matrix becomes less sparse with more elements further away from the main diagonal because of the coupling between the columns which disrupts the sparsity. This coupling is seen in Fig. 13(a) as the four dots that appear on the side of the semi-banded diagonal. The disruption of the sparse Jacobian matrix could increase the compute time as it becomes more difficult for the linear solver to solve in each time step in the ODE integrator. Simulating a single column, the coupling between the columns in the Jacobian matrix is avoided as the inlet is interpolated using a piece-wise polynomial. Therefore, the OCA-FPI system Jacobian equals a sub-block of the Jacobian of the strongly coupled approach, located in the upper left corner and with size 480 × 480.

Comparing the results in Tables 5 and 6, DG-C++ is slightly slower for the CSS evaluations compared to DG-Julia for both the OCA-FPI and the strongly coupled approach. For instance, for the LRMP Langmuir case study, DG-Julia is 1.9 times faster than DG-C++ for the startup benchmarks, whereas DG-Julia is 2.2 times faster than DG-C++ for the CSS benchmarks. However, the number of cycles that DG-Julia and DG-C++ simulated to reach the CSS was approximately the same. We conclude that since the CSS simulations require a higher number of cycles compared to the startup benchmarks, the Sundials IDA DAE solver performs worse at increasing number of cycles compared to the QNDF ODE solver in Julia. This is demonstrated by the results given in Fig. 14 for the LRMP using the Langmuir case study, where the number of cycles was varied at a coarse discretization setting (marked with stars), medium discretization setting (marked with triangles), and fine discretization setting (marked with hexagons).

Fig. 14 shows that for this particular case study, DG-Julia performs better with increasing number of cycles compared to DG-C++. For the fine discretization setting marked with hexagons i.e., $N_d^b = 6$ and $N_e = 32$, DG-C++ solves the problem faster at a lower number of cycles, but beyond 30 cycles, DG-Julia is faster. This performance difference with respect to number of cycles depends on the time integrator. For this specific case study, the QNDF ODE solver used in the DG-Julia implementation is more efficient than the Sundials IDA DAE solver implemented in DG-C++. Thus, the reason why DG-Julia is more efficient at solving until CSS for case study 2 is because it is required to simulate more cycles, which is slightly more efficient in the DG-Julia implementation due to the time integrator, as described before. For case study 4, the advantage of using the OCA-FPI approach is most significant. To investigate this further, the number of cycles was varied at a coarse discretization setting (marked with stars), medium discretization setting (marked with triangles) and fine discretization setting (marked with hexagons) for the GRM case study with the SMA isotherm, shown in Fig. 15.

For this case study, is clear that DG-C++ performs better at fine discretization settings, meaning large systems with many DoF, whereas DG-Julia performs poorly. Conversely, at medium and coarse discretization settings, i.e., smaller systems with fewer DoF, DG-Julia performs better than DG-C++. The fact that DG-Julia performs well with respect to the number of cycles for medium and smaller systems can be a factor in why the OCA-FPI approach in DG-Julia is efficient, though the number of cycles to reach CSS is different using the OCA-FPI and the strongly coupled approaches.

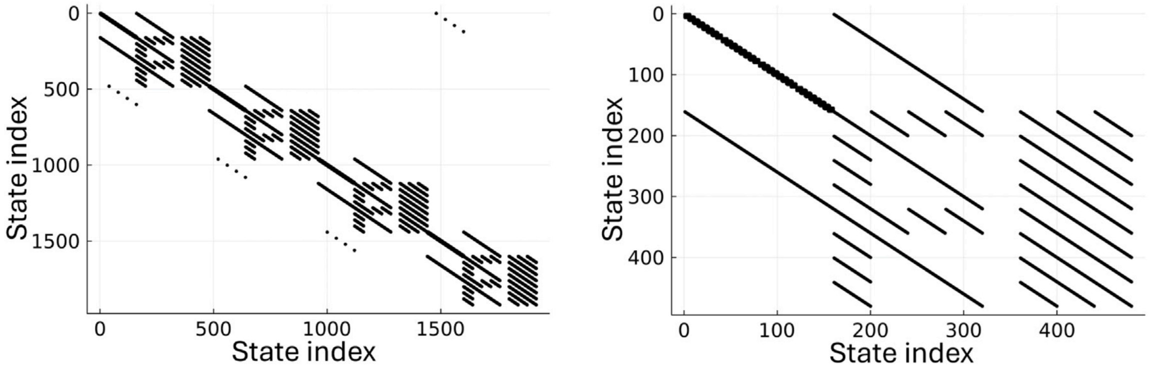
However, the computational benefit of applying the OCA-FPI method is insignificant for systems with a low number of DoF. Therefore, the OCA-FPI approach is especially useful for systems with many columns, components and complex models, i.e., GRM or LRMP, where the number of DoFs is naturally larger.

While the OCA approach is primarily intended for closed-loop SMB chromatography systems, it can also be applied to open-loop SMB chromatography systems such as the 3-column open-loop SMB chromatography system (Li et al., 2007). However, its relevance in open-loop systems may be limited as fewer cycles are required to reach the CSS for open-loop systems compared to closed-loop systems, making the computational advantage less prominent.

Finally, the OCA approach can also be applied to other multi-column setups such as MCSGP or PCC, however, they typically require fewer cycles to reach the CSS and often use fewer columns (Luca et al., 2020; Gomis-Fons et al., 2020). As a result, solving those systems involves fewer DoFs, making the OCA approach less relevant.

6. Conclusions

In this work, we introduced arbitrary networks of unit operations for the CADET-Core and CADET-Julia code bases, and verified both implementations via rigorous order-of-accuracy tests. Further, we conducted extensive continuous chromatography benchmarks, particularly for SMB chromatography, using three different implementations of two different spatial discretization methods: DG-Julia, DG-C++ and FV-C++. The considered benchmarks encompass startup benchmarks and CSS benchmarks. For the startup benchmarks, the DG-Julia implementation demonstrated slightly better performance compared to DG-C++. However, as observed in previous batch chromatography benchmarks (Frandsen et al., 2025b), the DG-C++ implementation scales better compared to DG-Julia, resulting in a performance advantage of DG-C++ over DG-Julia for large systems with many DoF. Both DG-C++ and DG-Julia were significantly more efficient than the FV-C++ implementation. For the CSS benchmarks, the strongly coupled approach of all three implementations was additionally compared with an OCA-FPI approach, which was implemented in Julia utilizing the DGSEM implemented in CADET-Julia. The OCA-FPI approach was the computationally most effective method for simulations of the SMBs until CSS was achieved if the number of DoF were larger than a specific threshold (400 DoF in the considered case studies). Below this threshold, however, the strongly coupled DG-Julia implementation was the fastest. The strong performance of the OCA-FPI approach can be attributed to the simplified system Jacobian, compared to the Jacobian



(a) Sparsity pattern of the Jacobian matrix of the strongly coupled approach.

(b) Sparsity pattern of the Jacobian matrix of the OCA-FPI approach.

Fig. 13. Sparsity pattern of the Jacobian matrices for the LRMP with the SMA isotherm case study (case study 3) using eight DGSEM elements and fourth order bulk phase polynomials.

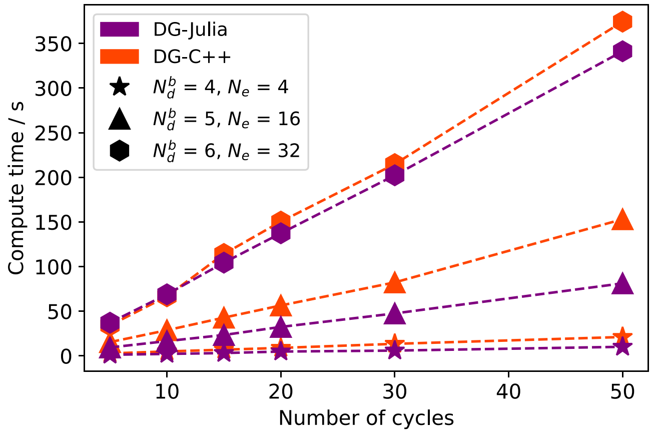


Fig. 14. Number of cycles and compute time comparison for a startup evaluation using the strongly coupled approach in DG-Julia and DG-C++. The simulations were evaluated using a various bulk phase polynomial degree and various number of bulk phase DG elements. The case study is the LRMP with the Langmuir isotherm case study (case study 2) with parameters found in Table S2.

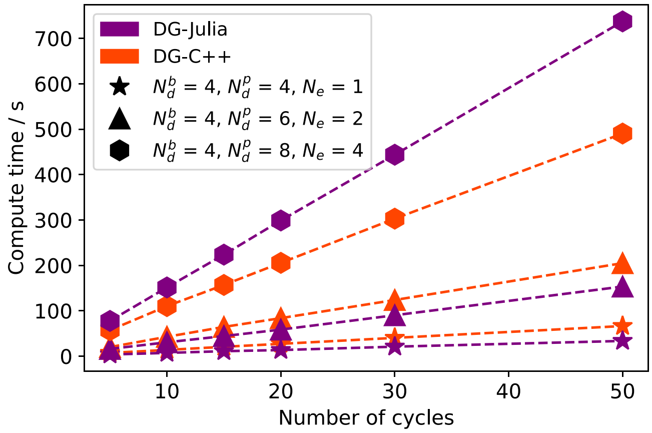


Fig. 15. Number of cycles and compute time comparison for a startup evaluation using the strongly coupled approach in DG-Julia and DG-C++. The simulations were evaluated using various bulk phase polynomial degrees and various numbers of bulk phase DG elements. The case study is the GRM with the SMA isotherm (case study 4) with parameters in Table S3.

of the strongly coupled approach, and the suboptimal scalability of DG-Julia. The DG-C++ implementation performed slightly better for the startup benchmarks compared to some of the CSS benchmarks. This was due to the CSS benchmarks requiring numerous computational cycles, which the QNDF solver in DG-Julia handled slightly more effectively than the specific, fixed BDF DAE time integrator employed in DG-C++.

CRediT authorship contribution statement

Jesper Frandsen: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis. **Jan Michael Breuer:** Writing – original draft, Software, Methodology, Investigation. **Johannes Schmölder:** Writing – review & editing, Software. **Jens Abildskov:** Writing – review & editing, Supervision, Resources, Project administration. **Jakob Kjøbsted Huusom:** Writing – review & editing, Supervision. **Krist V. Gernaey:** Writing – review & editing, Supervision. **Eric von Lieres:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This project has received funding from the Innovative Medicines Initiative 2 Joint Undertaking under grant agreement No 101007799 (Inno4Vac). This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and EFPIA. This communication reflects the author's view and neither IMI nor the European Union, EFPIA, or any associated partners are responsible for any use that may be made of the information contained herein.

Jan M. Breuer has received funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 548805630 (grant no. GA 2160/8-1).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.compchemeng.2025.109295.

Data availability

Data will be made available on request.

References

Abunasser, N., Wankat, P.C., 2004. One-column chromatograph with recycle analogous to stimulated moving bed adsorbers: Analysis and applications. Ind. Eng. Chem. Res. 43, 5291–5299. http://dx.doi.org/10.1021/ie0400346.

Abunasser, N., Wankat, P.C., Kim, Y.S., Koo, Y.M., 2003. One-column chromatograph with recycle analogous to a four-zone simulated moving bed. Ind. Eng. Chem. Res. 42, 5268–5279. http://dx.doi.org/10.1021/ie030283e.

Breuer, J., Jäpel, R., Frandsen, J., 2025. cadet/CADET-Verification: Supplementary to Frandsen et al. 2025. Zenodo, http://dx.doi.org/10.5281/zenodo.14651341.

Breuer, J.M., Leweke, S., Schmölder, J., Gassner, G., von Lieres, E., 2023. Spatial discontinuous Galerkin spectral element method for a family of chromatography models in CADET. Comput. Chem. Eng. 177, 108340. http://dx.doi.org/10.1016/j.compchemeng.2023.108340, URL https://www.sciencedirect.com/science/article/pii/S0098135423002107.

Carta, G., Jungbauer, A., 2020. Protein chromatography : process development and scale-up. vol. 1485, Wiley-VCH, p. 423. http://dx.doi.org/10.1007/978-1-4939-6412-3,

Cockburn, B., Karniadakis, G.E., Shu, C.-W., 2000. The development of discontinuous Galerkin methods. In: Cockburn, B., Karniadakis, G.E., Shu, C.-W. (Eds.), Discontinuous Galerkin Methods. In: Lecture Notes in Computational Science and Engineering, Springer, Berlin, Heidelberg, pp. 3–50. http://dx.doi.org/10.1007/978-3-642-59721-3_1.

Frandsen, J., Breuer, J.M., Schmölder, J., Abildskov, J., Abildskov, J.K.H., Gernaey, K.V., von Lieres, E., 2025a. jespfra/Supplement-to-Frandsen-et-al.-2025a: Supplement-to-Frandsen-et-al.-2025a. Zenodo, http://dx.doi.org/10.5281/zenodo.14651328, URL DOI: 10.5281/zenodo.14651328.

Frandsen, J., Breuer, J.M., Schmölder, J., Huusom, J.K., Gernaey, K.V., Abildskov, J., von Lieres, E., 2025b. CADET-Julia: Efficient and versatile, open-source simulator for batch chromatography in Julia. Comput. Chem. Eng. 192, 108913. http://dx.doi.org/10.1016/j.compchemeng.2024.108913, URL https://www.sciencedirect.com/science/article/pii/S0098135424003314.

Frandsen, J., Huusom, J.K., Gernaey, K.V., Abildskov, J., 2023a. Design of multi-component gradient SMBs. Comput. Aided Chem. Eng. 52, 903–908.

Frandsen, J., Huusom, J.K., Gernaey, K.V., Abildskov, J., 2023b. Shortcut design method for multicomponent gradient simulated moving beds. AIChE J. http://dx.doi.org/10.1002/aic.18304.

Gardner, D.J., Reynolds, D.R., Woodward, C.S., Balos, C.J., 2022. Enabling new flexibility in the SUNDIALS suite of nonlinear and differential/algebraic equation solvers. ACM Trans. Math. Softw. (TOMS) http://dx.doi.org/10.1145/3539801.

Gerstweiler, L., Bi, J., Middelberg, A.P., 2021. Continuous downstream bioprocessing for intensified manufacture of biopharmaceuticals and antibodies. Chem. Eng. Sci. 231, 116272. http://dx.doi.org/10.1016/j.ces.2020.116272.

Gomis-Fons, J., Andersson, N., Nilsson, B., 2020. Optimization study on periodic counter-current chromatography integrated in a monoclonal antibody downstream process. J. Chromatogr. A 1621, 461055. http://dx.doi.org/10.1016/j.chroma.2020.461055.

Gu, T., 2015. Mathematical Modeling and Scale-Up of Liquid Chromatography: With Application Examples. Springer, Google-Books-ID: PiryBwAAQBAJ.

Guiochon, G., Felinger, A., Shirazi, D.G., 2006. Fundamentals of Preparative and Nonlinear Chromatography. Elsevier.

He, Q.L., Leweke, S., von Lieres, E., 2018. Efficient numerical simulation of simulated moving bed chromatography with a single-column solver. Comput. Chem. Eng. 111, 183–198. http://dx.doi.org/10.1016/j.compchemeng.2017.12.022.

He, Q.L., von Lieres, E., Sun, Z., Zhao, L., 2020. Model-based process design of a ternary protein separation using multi-step gradient ion-exchange SMB chromatography. Comput. Chem. Eng. 138, 106851. http://dx.doi.org/10.1016/j.compchemeng.2020.106851.

Hesthaven, J.S., Warburton, T., 2002. Nodal high-order methods on unstructured grids: I. time-domain solution of Maxwell's equations. J. Comput. Phys. 181 (1), 186–221. http://dx.doi.org/10.1006/jcph.2002.7118, URL https://www.sciencedirect.com/science/article/pii/S0021999102971184.

Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., Woodward, C.S., 2005. Sundials. ACM Trans. Math. Softw. (TOMS) http://dx.doi.org/10.1145/1089014.1089020, URL https://dl.acm.org/doi/10.1145/1089014.1089020.

Javeed, S., Qamar, S., Seidel-Morgenstern, A., Warnecke, G., 2011. A discontinuous Galerkin method to solve chromatographic models. J. Chromatogr. A 1218 (40), 7137–7146. http://dx.doi.org/10.1016/j.chroma.2011.08.005, URL https://www.sciencedirect.com/science/article/pii/S0021967311011708.

Khan, A., Perveen, S., Qamar, S., 2021. Discontinuous-Galerkin finite-element method for approximating a model of non-equilibrium liquid chromatography considering Bi-Langmuir isotherm. J. Liq. Chromatogr. Relat. Technol. 44 (5–6), 298–308. http://dx.doi.org/10.1080/10826076.2021.1916526, Publisher: Taylor & Francis.

Khanal, O., 2022. Mathematical modeling and process analytical technology for continuous chromatography of biopharmaceutical products. Curr. Opin. Biotechnol. 78, 102796. http://dx.doi.org/10.1016/j.copbio.2022.102796.

Kumar, V., Lenhoff, A.M., 2020. Mechanistic modeling of preparative column chromatography for Biotherapeutics. Annu. Rev. Chem. Biomol. Eng. 11 (1), 235–255. http://dx.doi.org/10.1146/annurev-chembioeng-102419-125430.

Lee, J.W., Zarei, S., Kienle, A., Seidel-Morgenstern, A., 2024. Short-cut design of double-layer simulated moving bed chromatography for continuous ternary mixture separation: Langmuir isotherms. Ind. Eng. Chem. Res. 63 (39), 16873–16884. http://dx.doi.org/10.1021/acs.iecr.4c01531.

Leweke, S., 2021. Unified Modeling and Efficient Simulation of Chromatographic Separation Processes (Ph.D. thesis). RWTH Aachen University, Aachen, http://dx.doi.org/10.18154/RWTH-2022-01184, Artwork Size: pages 1 Online-Ressource : Illustrationen Pages: pages 1 Online-Ressource : Illustrationen Publication Title: Dissertation Volume: Rheinisch-Westfälische Technische Hochschule Aachen, URL https://publications.rwth-aachen.de/record/840314.

Leweke, S., 2024. CADET-Semi-Analytic: Version 1.2.0. Zenodo, http://dx.doi.org/10.5281/zenodo.14066870.

Leweke, S., von Lieres, E., 2016. Fast arbitrary order moments and arbitrary precision solution of the general rate model of column liquid chromatography with linear isotherm. Comput. Chem. Eng. 84, 350–362. http://dx.doi.org/10.1016/j.compchemeng.2015.09.009, URL https://www.sciencedirect.com/science/article/pii/S009813541500304X.

Leweke, S., von Lieres, E., 2018. Chromatography analysis and design toolkit (CADET). Comput. Chem. Eng. 113, 274–294. http://dx.doi.org/10.1016/j.compchemeng.2018.02.025, URL https://www.sciencedirect.com/science/article/pii/S0098135418300966.

Li, P., Xiu, G., Rodrigues, A.E., 2007. Proteins separation and purification by salt gradient ion-exchange SMB. AIChE J. 59, 215–228. http://dx.doi.org/10.1002/aic.

von Lieres, E., Andersson, J., 2010. A fast and accurate solver for the general rate model of column liquid chromatography. Comput. Chem. Eng. 34 (8), 1180–1191. http://dx.doi.org/10.1016/j.compchemeng.2010.03.008, URL https://www.sciencedirect.com/science/article/pii/S0098135410000992.

Lothert, K., Wolff, M.W., 2023. Chromatographic purification of viruses: State of the art and current trends. Bioprocess Anal. Dev. Virus- Based Adv. Ther. Med. Prod. (Atmps) 145–169. http://dx.doi.org/10.1007/978-3-031-28489-2_7.

Luca, C.D., Felletti, S., Lievore, G., Buratti, A., Vogg, S., Morbidelli, M., Cavazzini, A., Catani, M., Macis, M., Ricci, A., Cabri, W., 2020. From batch to continuous chromatographic purification of a therapeutic peptide through multicolumn countercurrent solvent gradient purification. J. Chromatogr. A 1625, 461304. http://dx.doi.org/10.1016/j.chroma.2020.461304.

Matte, A., 2022. Recent advances and future directions in downstream processing of therapeutic antibodies. In: Poupon, A., Watier, H. (Eds.), Int. J. Mol. Sci. 23 (15), 8663. http://dx.doi.org/10.3390/ijms23158663.

Meyer, K., 2020. Advanced Simulation of Preparative Chromatography Processes (Ph.D. thesis). DTU Chemical Engineering, Kongens Lyngby.

Meyer, K., Leweke, S., von Lieres, E., Huusom, J.K., Abildskov, J., 2020. ChromaTech: A discontinuous Galerkin spectral element simulator for preparative liquid chromatography. Comput. Chem. Eng. 141, 107012. http://dx.doi.org/10.1016/j.compchemeng.2020.107012, URL https://www.sciencedirect.com/science/article/pii/S009813542030329X.

Migliorini, C., Mazzotti, M., Morbidelli, M., 2000. Robust design of countercurrent adsorption separation processes: 5. Nonconstant selectivity. AIChE J. 46, 1384–1399. http://dx.doi.org/10.1002/aic.690460712.

Mota, J.P., Araújo, J.M., 2005. Single-column simulated-moving-bed process with recycle lag. AIChE J. 51, 1641–1653. http://dx.doi.org/10.1002/aic.10426.

Püttmann, A., Schnittert, S., Naumann, U., von Lieres, E., 2013. Fast and accurate parameter sensitivities for the general rate model of column liquid chromatography. Comput. Chem. Eng. 56, 46–57. http://dx.doi.org/10.1016/j.compchemeng.2013.04.021, URL https://www.sciencedirect.com/science/article/pii/S0098135413001440.

Rackauckas, C., Nie, Q., 2017. DifferentialEquations.jl – A performant and feature-rich ecosystem for solving differential equations in Julia. J. Open Res. Softw. 5, 15. http://dx.doi.org/10.5334/jors.151.

Rathore, A.S., Zydney, A.L., Anupa, A., Nikita, S., Gangwar, N., 2022. Enablers of continuous processing of biotherapeutic products. Trends Biotechnol. 40 (7), 804–815. http://dx.doi.org/10.1016/j.tibtech.2021.12.003.

Schmidt-Traub, H., Schulte, M., Seidel-Morgenstern, A., 2020. Preparative Chromatography. 7H_PDwAAQBAJ, John Wiley & Sons, Google-Books-ID.

Schultze-Jena, A., Boon, M.A., Bussmann, P.J., Janssen, A.E., van der Padt, A., 2017. The counterintuitive role of extra-column volume in the determination of column efficiency and scaling of chromatographic processes. J. Chromatogr. A 1493, 49–56. http://dx.doi.org/10.1016/j.chroma.2017.02.068.

Schweiger, S., Jungbauer, A., 2018. Scalability of pre-packed preparative chromatography columns with different diameters and lengths taking into account extra column effects. J. Chromatogr. A 1537, 66–74. http://dx.doi.org/10.1016/j.chroma.2018.01.022.

Shampine, L., Reichelt, M., Shampine, L.F., Reichelt, M.W., 1997. The MATLAB ODE suite. SIAM J. Sci. Comput- Ing http://dx.doi.org/10.1137/S1064827594276424, URL https://hal.science/hal-01333731.

Shekhawat, L.K., Rathore, A.S., 2019. An overview of mechanistic modeling of liquid chromatography. Prep. Biochem. Biotechnol. 49 (6), 623–638. http://dx.doi.org/10.1080/10826068.2019.1615504, Publisher: Taylor & Francis.

Wu, X., Arellano-Garcia, H., Hong, W., Wozny, G., 2013. Improving the operating conditions of gradient ion-exchange simulated moving bed for protein separation. Ind. Eng. Chem. Res. 52, 5407–5417. http://dx.doi.org/10.1021/ie302885f.

Zafar, S., Perveen, S., Qamar, S., 2021. Discontinuous Galerkin scheme for solving non-isothermal and non-equilibrium model of liquid chromatography. J. Liq. Chromatogr. Relat. Technol. 44 (1–2), 52–69. http://dx.doi.org/10.1080/10826076.2020.1867164.

Zafar, S., Perveen, S., Qamar, S., 2023. Discontinuous Galerkin finite element scheme for solving non-linear lumped kinetic model of non-isothermal reactive liquid chromatography. Korean J. Chem. Eng. 40 (3), 555–571. http://dx.doi.org/10.1007/s11814-022-1352-4.