Computers and Chemical Engineering 111 (2018) 183–198

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Efficient numerical simulation of simulated moving bed chromatography with a single-column solver

Qiao-Le He, Samuel Leweke, Eric von Lieres*

Forschungszentrum Jülich GmbH, IBG-1: Biotechnology, Wilhelm-Johnen-Str. 1, 52425 Jülich, Germany

ARTICLE INFO

Article history:
Received 1 August 2017
Revised 17 November 2017
Accepted 28 December 2017
Available online 11 January 2018

Keywords:
Simulated moving bed chromatography
One-column analog
Operator-splitting
Cyclic steady state
System trajectory

ABSTRACT

We present four different numerical methods for the numerical simulation of simulated moving bed chromatography. Two approaches use fixed-point iteration for computing cyclic steady states, and two other approaches use operator splitting for computing complete system trajectories. All methods are based on weak coupling of individual column models and can easily be implemented using any existing single-column solver. Simulation software is implemented based on the CADET project and published as open source code.

The numerical performance is compared using five case studies. For both fixed-point iteration and operator-splitting, an alternative approach is found to be more efficient than the standard approach. Namely, the one-column analog saves time in computing the cyclic steady state, while lag-aware operator-splitting yields more detailed information on the system trajectory. The presented methods can be combined with other models, for example to consider hold-up volumes, and have applications beyond simulated bed chromatography.

1. Introduction

1.1. Preparative chromatography

Packed bed chromatography is widely applied for separating and purifying chemical substances in liquid solution. Depending on the task, individual chromatography columns are typically operated in flow-through or bind-elute mode. In flow-through mode, the components of a mixture migrate through the column at different velocities caused by varying binding affinities to the functionalized resin. Typical retention mechanisms are affinity, ion-exchange, hydrophobic interaction, or mixed mode. In bind-elute mode, all components are first adsorbed to the resin and then successively desorbed by adding a mobile phase modifier. Both modes are batch processes, i.e., only a finite load can be processed in one run. The throughput can be increased by performing further runs after regenerating the column. In practice, a sequence of several chromatography steps is often required in order to achieve specific product qualities.

1.2. Simulated moving bed chromatography

Continuous processes often have higher productivities than batch processes (Engell and Toumi, 2004). Hence, true moving bed (TMB) chromatography has been introduced as a theoretical concept for continuous chromatography, and simulated moving bed (SMB) chromatography as a practical concept for its technical implementation (Broughton and Gerhold, 1961). In TMB chromatography, the packed bed and the liquid solution are moved through a circular column in opposite directions. For suitable operating conditions, i.e., bed and liquid velocities, more strongly binding components are traveling in the direction of the bed, and less strongly binding components are traveling in the direction of the liquid. Consequently, these components can be collected at either side of the feed injection point. A desorbent is injected to the circular column across the feed in order to fully elute the components and regenerate the bed.

In SMB chromatography, the circular column is replaced by a finite number of conventional columns (Faria and Rodrigues, 2015; Rajendran et al., 2009; Seidel-Morgenstern et al., 2008; Silva et al., 2015). The bed movement is implemented by periodically shifting the position of these columns in the network. Physical movement of the columns is avoided by switching the network connectivity through a complex system of valves and tubes. The four-zone SMB setup, as illustrated in Fig. 1, facilitates binary separations. As four ports are present in the system, namely feed, desorbent, extract and raffinate, the physical SMB process requires at least four columns. Suitable operating conditions can be determined by triangle theory (Rajendran et al., 2009).

The SMB process approximates the TMB process, and the discrepancy between them can be reduced by using more columns

* Corresponding author.
E-mail address: e.von.lieres@fz-juelich.de (E. von Lieres).

https://doi.org/10.1016/j.compchemeng.2017.12.022
0098-1354/

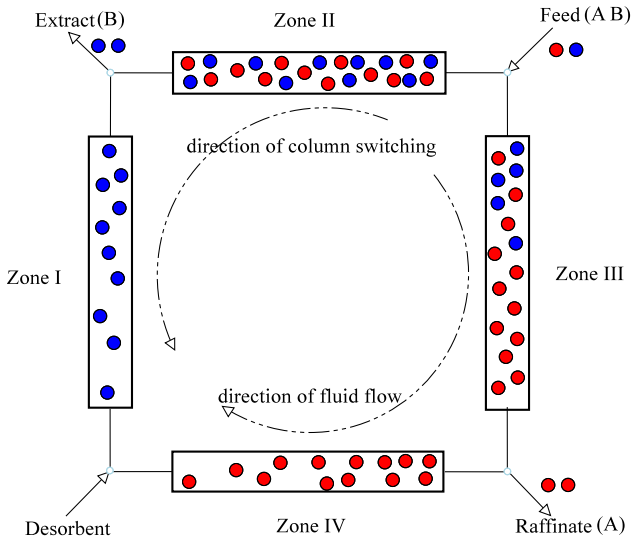

Fig. 1. Schematic of four-zone SMB chromatography for binary separations. Column positions are periodically switched in opposite direction of liquid flow.

that are shorter and proportionally increasing the switching frequency. Nonetheless, SMB chromatography is only a pseudo-continuous process that enters, after a ramp-up phase, a cyclic steady state (CSS). Before entering the CSS, the system follows a dynamic trajectory, which depends on initial conditions (i.e., initial loading of the columns), on boundary conditions (e.g., concentrations at the feed and desorbent ports), and on operating conditions (e.g., flowrates and switching frequency).

1.3. One-column analog

The previously introduced SMB process requires at least four equal columns, one in each zone. The columns should be identically packed, which can be challenging in practice (Abunasser et al., 2003). In the CSS, the system then repeats the same trajectory after each switch, that is, all columns have the same state but with a different time-lag. This fact has been utilized for introducing the one-column analog (OCA) to SMB chromatography (Abunasser and Wankat, 2004; Mota and Araújo, 2005; Rodrigues et al., 2007). In the OCA process, a single column is cycled through all positions of the network, as if it was one column in the SMB process. The remaining columns are replaced by plug-flow devices in order to provide the influx and collect the outflux of that column (Mota and Araújo, 2005). The direction of flow in these plug-flow devices is reversed between loading and unloading. The process can be further simplified by replacing the plug-flow devices with holding tanks, at the cost of increased mixing and decreased purity.

1.4. Ternary separation

The four-zone SMB process can fully separate only binary mixtures. However, many applications require separating a target component from both stronger and weaker retained components. Ternary separations, also referred to as center-cut separations, can be achieved by cascading two four-zone SMB units (Nicolaos et al., 2001; Nowak et al., 2012; Wooley et al., 1998), as illustrated in Fig. 2. In the presented example, the most retained component is collected at the extract port of the first unit, while the raffinate stream is fed to the second unit in order to separate the remaining mixture. In an alternative setup, the least retained component is collected at the raffinate port of the first unit, while the extract stream is fed to the second unit (schematic not shown).

Cascade schemes require designing and operating two four-zone SMB processes, which can be laborious and costly. Hence, integrated schemes have been developed using a single multi-port valve and fewer pumps than cascade schemes. Moreover, process design is simplified due to simultaneous switching of all columns. The left panel of Fig. 3 shows an integrated eight-zone scheme, in which the most retained component is collected at the extract I port, while the raffinate I stream is reinjected across the feed port (Seidel-Morgenstern et al., 2008; da Silva and Seidel-Morgenstern, 2016). The working principle is very similar to the cascade scheme, as the columns are regenerated in zones I and V. The right panel of Fig. 3 shows an alternative eight-zone scheme, in which the least retained component is collected at the raffinate I port, while the extract I stream is reinjected across the feed port.

Ternary separations can also be achieved by integrated five-zone schemes with two extract ports or with two raffinate ports (Abunasser and Wankat, 2005; Jo et al., 2007; Khan and Younas, 2011; Kim et al., 2003; Mun, 2011; Nicolaos et al., 2001; Wang and Ching, 2005; Yu et al., 2015), as illustrated in Fig. 4. Five-zone schemes are particularly effective when the target component is present in much higher abundance than the more strongly retained component.

1.5. Hold-up volumes

In addition to the interstitial volume of the packed bed, SMB units comprise various hold-up volumes, e.g., in the frits, tubing, valve and pumps. The optimal design and performance of SMB processes has been observed to be particularly sensitive to the presence and the size of such hold-up volumes (Grosfils et al., 2010; Katsuo et al., 2009; Lim, 2012; Lim and Bhatia, 2011; Silva et al., 2016). For SMB processes with a linear isotherm, the impact of hold-up volumes can be theoretically analyzed using triangle theory (Katsuo et al., 2009). In the general case, the impact of hold-up volumes is to increase retention time and dispersion.

The impact of hold-up volumes can be described by a network of plug-flow reactors (PFR), stirred tank reactors (CSTR) and/or dispersive plug flow reactors (DPFR) (Grosfils et al., 2010; Lim, 2012; Silva et al., 2016). Fig. 5 illustrates the positions of the most important hold-up volumes within typical SMB units. The hold-up volume on either side of a column, i.e., tubing and frits, can be described by a CSTR that is moved through the network together with that column. Additional CSTR and/or (D)PFR models can be required for describing hold-up volumes in tubing, pumps and detectors outside of the ports.

1.6. Numerical solution

SMB processes and other network configurations are ideally modeled by strongly coupling individual models of the involved columns and hold-up volumes, i.e., one large equation system is set up and simultaneously solved for the entire process. However, there are many options for modeling mass transfer in a chromatography column, e.g., the equilibrium-dispersive model (EDM), the transport-dispersive model (TDM) and the general rate model (GRM) that can be combined with a multitude of binding models (Felinger and Guiochon, 2004). The choice of an appropriate model is highly application specific. In many cases not only the model parameters but also the structure of the model itself need to be determined from measurement data. Hence, a simulator should not only support different SMB schemes but also allow choosing between various modeling options for the columns and hold-up volumes. However, combinatorial complexity can make the implementation of such numerical simulation code very challenging.

In this contribution, we present and compare alternative numerical solution methods that are based on weak coupling of the

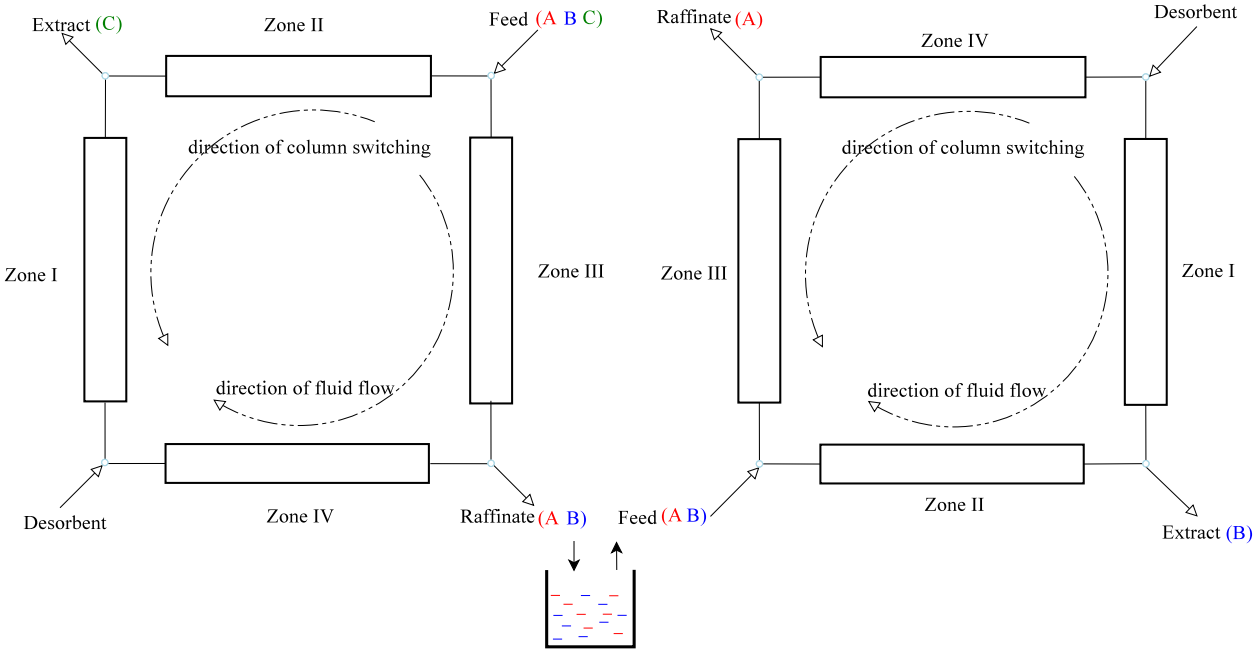

Fig. 2. Cascade scheme of two four-zone SMB units for ternary separations. Alternatively, extract stream of first unit can be fed to second unit.

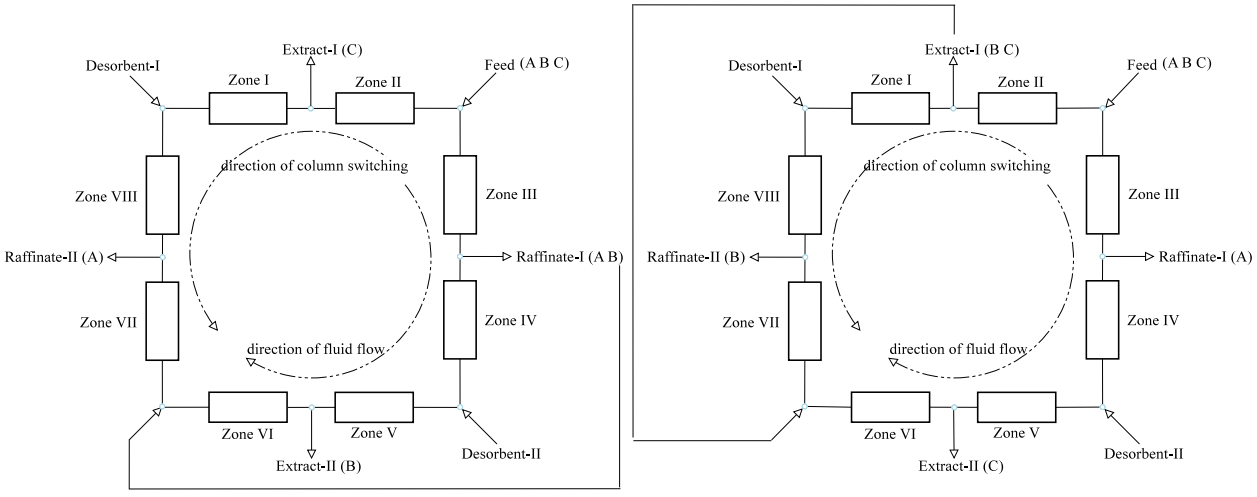

Fig. 3. Integrated eight-zone SMB schemes for ternary separations. Raffinate I (left) or extract I (right) stream is reinjected across the feed port.

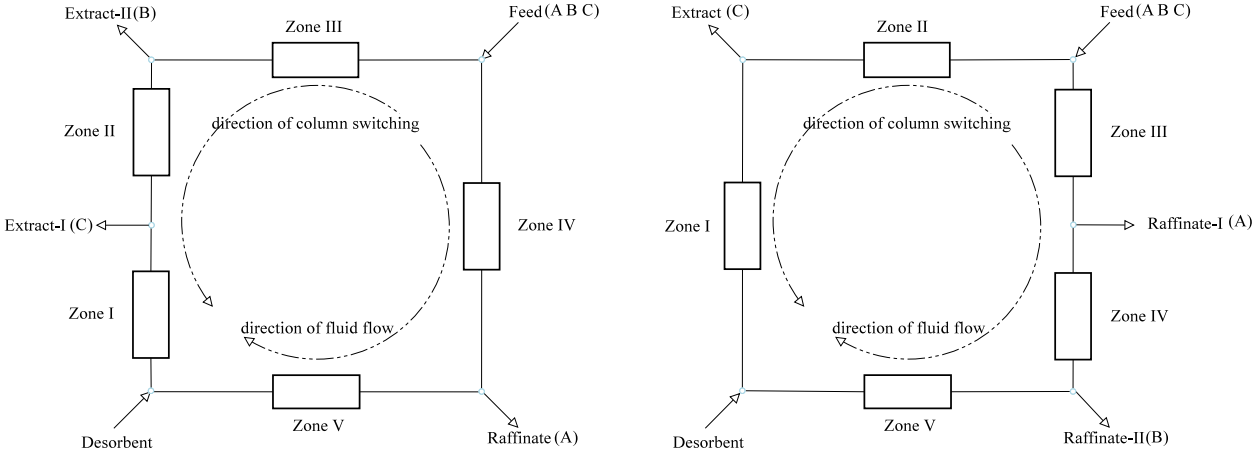

Fig. 4. Integrated five-zone SMB schemes with two extract ports (left) or two raffinate ports (right) for ternary separations.

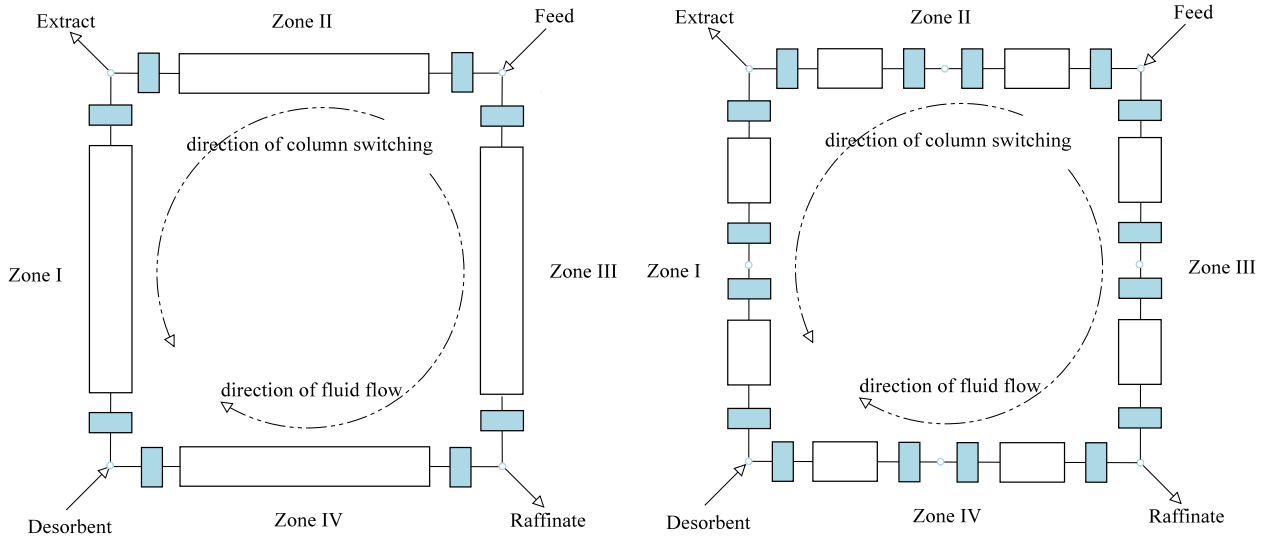

Fig. 5. Four-zone SMB schemes with four (left) and eight (right) columns indicating positions of the associated hold-up volumes.

individual models. Weak coupling refers to numerical approaches in which the coupled subsystems are not simultaneously solved at the same time but iteratively solved one after another. It is computationally less efficient than strong coupling, but can deliver equally accurate results. The major advantage of weak coupling is that existing simulators for single chromatography columns and hold-up volumes can be interfaced without modifying their respective computational codes. Four different solution approaches are introduced and compared: the first two approaches use fixed point iteration (FPI) for computing the CSS, based on the standard SMB process (STD-FPI) and on the OCA process (OCA-FPI). The latter will be shown to be computationally advantageous, in particular for systems with many columns. Two more solution approaches use operator splitting (OSP) for computing the dynamic system trajectory (DST), starting from arbitrary initial values, in full detail. In addition to the standard approach (STD-OSP), we present a numerically more efficient variant, lag-aware operator splitting (LAW-OSP). This approach appears to be novel, at least in the context of SMB chromatography.

Other authors have suggested to apply surrogate models (Li et al., 2014a) and model reduction (Li et al., 2014b; Zhang et al., 2017) for reducing the computational burden of simulating SMB chromatography, which can be particularly useful for parameter estimation and process optimization. However, building surrogate or reduced models also requires a certain number of fully detailed simulations. The methods described in this study can be used for efficiently computing the such detailed simulations with a single-column solver.

2. Mathematical model

The mathematical model of SMB chromatography combines different equations for describing mass transfer in the column, boundary conditions, binding of solute molecules, the impact of hold-up volumes, and network connectivity. Moreover, purity and productivity are defined as performance indicators for the SMB process.

2.1. Mass transfer

The numerical methods presented in this contribution are introduced using the GRM but can as well be applied to any other mass transfer models, e.g., the TDM or EDM. In the GRM, concentration gradients are accounted for along the axial column and radial particle dimensions. Consider an SMB process with N columns of length L. All columns are assumed to be identically packed with particles of radius r_p. The mass balances in the interstitial volume and in the particle pores are given by

$$\frac{\partial c_i^j}{\partial t} = -u_{\text{int}}^j \frac{\partial c_i^j}{\partial z} + D_{\text{ax}}^j \frac{\partial^2 c_i^j}{\partial z^2} - \frac{1-\varepsilon_c}{\varepsilon_c} \frac{3}{r_p} k_{f,i}^j \left(c_i^j - c_{p,i}^j (r = r_p) \right) \quad (1a)$$

$$\frac{\partial c_{p,i}^j}{\partial t} = D_{p,i} \left(\frac{\partial^2 c_{p,i}^j}{\partial r^2} + \frac{2}{r} \frac{\partial c_{p,i}^j}{\partial r} \right) - \frac{1-\varepsilon_p}{\varepsilon_p} \frac{\partial q_i^j}{\partial t} \quad (1b)$$

In Eq. (1), c_i^j, $c_{p,i}^j$ and q_i^j denote the interstitial, pore and stationary phase concentrations of component $i \in \{1, \dots, M\}$ in column $j \in \{1, \dots, N\}$. Furthermore, z denotes the axial position, r radial position, t time, ε_c column porosity, ε_p particle porosity, u_{int}^j interstitial velocity, D_{ax}^j is the axial dispersion coefficient, $D_{p,i}$ the effective pore diffusion coefficient, and $k_{f,i}^j$ the film mass transfer coefficient.

Danckwerts boundary conditions are applied at the column inlet and outlet (Barber et al., 1998):

$$\left. \frac{\partial c_i^j}{\partial z} \right|_{z=0} = \frac{u_{\text{int}}^j}{D_{\text{ax}}^j} \left(c_i^j (z = 0) - c_{\text{in},i}^j \right) \quad (2a)$$

$$\left. \frac{\partial c_i^j}{\partial z} \right|_{z=L} = 0 \quad (2b)$$

Here, $c_{\text{in},i}^j$ is the inlet concentration of component i in column j. The boundary conditions at the particle surface and center are given by

$$\left. \frac{\partial c_{p,i}^j}{\partial r} \right|_{r=r_p} = \frac{k_{f,i}^j}{\varepsilon_p D_{p,i}} \left(c_i^j - c_{p,i}^j (r = r_p) \right) \quad (3a)$$

$$\left. \frac{\partial c_{p,i}^j}{\partial r} \right|_{r=0} = 0 \quad (3b)$$

In the TDM, Eq. (1) is replaced by Eq. (4), and Eq. (3) is not required,

$$\frac{\partial c_i^j}{\partial t} = -u_{\text{int}}^j \frac{\partial c_i^j}{\partial z} + D_{\text{ax}}^j \frac{\partial^2 c_i^j}{\partial z^2} - \frac{1-\varepsilon_c}{\varepsilon_c} \frac{\partial q_i^j}{\partial t}. \quad (4)$$

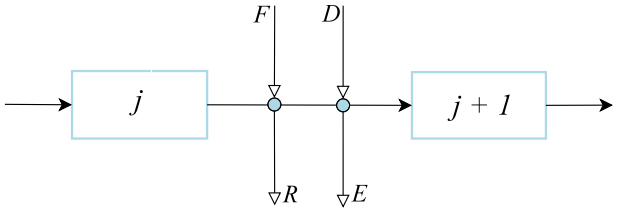

Fig. 6. Schematic of connection between adjacent columns.

2.2. Adsorption and desorption

The GRM, TDM and EDM can be combined with a variety of binding models for describing adsorption and desorption of solute molecules. We here consider the linear isotherm Eq. (5a) and the competitive Langmuir isotherm Eq. (5b):

$$q^j_{\mathrm{eq},i} = H_i c^j_i \tag{5a}$$

$$q^j_{\mathrm{eq},i} = \frac{H_i c^j_i}{1 + \sum_{k=1}^{N} b_k c^j_k} \tag{5b}$$

In Eq. (5), $q^j_{\mathrm{eq},i}$ is the stationary phase concentration in equilibrium, H_i the Henry coefficient, and b_i the equilibrium constant.

Kinetic binding can be described by using the linear driving force (LDF) model:

$$\frac{\partial q^j_i}{\partial t} = k_{\mathrm{kin},i}(q^j_{\mathrm{eq},i} - q^j_i) \tag{6}$$

The parameter $k_{\mathrm{kin},\,i}$ quantifies how fast the equilibrium is reached. In case $k_{\mathrm{kin},\,\mathrm{i}}$ is not specified, Eq. (6) is replaced by $q^j_i = q^j_{\mathrm{eq},i}$ and the TDM formally reduces to the EDM.

2.3. Network connectivity

In the SMB model, two adjacent columns, j and $j+1$ with $j = 1, \ldots, N$ are connected by node j as illustrated in Fig. 6. The cycle is closed by identifying column $j = N+1$ with column $j = 1$, i.e., both of these indices point at the same physical column. More generally, the notation is simplified by identifying columns $j = N+2$ and $j = 2$, $j = N+3$ and $j = 3$, and so on. By definition, node j is located upstream of column j and downstream of column $j+1$. At each node only one or none of the feed (F), desorbent (D), raffinate (R), or extract (E) fluxes is nonzero at a time. The inlet concentration of component i in column $j+1$ is determined by the mass balance of node j:

$$c^{j+1}_{\mathrm{in},i} = \frac{c^j_{\mathrm{out},i} Q^j + \delta^j_i}{Q^{j+1}} \tag{7}$$

In the above equation, $c^j_{\mathrm{out},i}$ denotes the outlet concentration of component i in column j, $Q^j = \varepsilon_c A u^j_{\mathrm{int}}$ is the volumetric flowrate, and A the column cross-section area. The term δ^j with $j = 1, \ldots, N$ determines the current role of node j. Column shifting is implemented by periodically permuting these terms, that is, the terms δ are piecewise constant functions of time:

$$\delta^j_i = \begin{cases} c^F_{\mathrm{in,i}} Q^F & \text{feed} \\ c^D_{\mathrm{in,i}} Q^D & \text{desorbent} \\ -c^j_{\mathrm{out,i}} Q^R & \text{raffinate} \\ -c^j_{\mathrm{out,i}} Q^E & \text{extract} \\ 0 & \text{none of the above} \end{cases} \tag{8}$$

In Eq. (8), c^F_{in} and c^D_{in} are the feed and desorbent concentrations, and Q^F, Q^D, Q^R, Q^E the volumetric flowrates at the feed, desorbent, raffinate, and extract ports, respectively. A straightforward extension of the indexing scheme is required for describing SMB processes with multiple feed, desorbent, raffinate, and/or extract ports (e.g., the cascade or integrated schemes with five and eight zones). The case $\delta^j = 0$ applies to nodes that are currently not connected to a port, that is, in the interior of a zone. This occurs when more columns than zones are present, such as in a four-zone scheme with eight columns.

2.4. Hold-up volumes

The hold-up volumes in a typical SMB process can be classified in four categories: (1) tubing between multi-port valve and column inlet plus frit before packed bed, (2) frit after packed bed plus tubing between column outlet and multi-port valve, (3) tubing between injection point and multi-port valve, and (4) tubing between multi-port valve and detector. Each of these categories can be modeled as one or more PFR, CSTR and/or DPFR in series. For categories 1 and 2, Fig. 5 illustrates the positions of hold-up volumes in the column network. These hold-up volumes before and behind the packed bed are associated with each column, and their network positions are switched together with that column.

The PFR model describes a simple time-lag:

$$c^j_{\mathrm{PFR},i}(t) = c^j_{\mathrm{in},i}\left(t - \tau^j_{\mathrm{PFR}}\right) \tag{9}$$

Here, $c^j_{\mathrm{PFR},i}$ is the concentration of component i in PFR j and τ^j_{PFR} the corresponding time-lag.

The CSTR model describes mixing in a stirred tank:

$$\frac{\mathrm{d}c^j_{\mathrm{CSTR},i}}{\mathrm{d}t} = \frac{c^j_{\mathrm{in},i} - c^j_{\mathrm{CSTR},i}}{\tau^j_{\mathrm{CSTR}}} \tag{10}$$

In Eq. (10), $c^j_{\mathrm{CSTR},i}$ denotes the concentration of component i in CSTR j and τ^j_{CSTR} the corresponding residence time.

The DPFR model describes convection and axial dispersion in a tube of length L:

$$\frac{\partial c^j_{\mathrm{DPFR},i}}{\partial t} = -u_{\mathrm{DPFR},i}\frac{\partial c^j_{\mathrm{DPFR},i}}{\partial z} + D^j_{\mathrm{DPFR},i}\frac{\partial^2 c^j_{\mathrm{DPFR},i}}{\partial z^2} \tag{11}$$

In the above equation, $c^j_{\mathrm{DPFR},i}$ is the concentration of component i in DPFR j, $u_{\mathrm{DPFR},\,i}$ the velocity, and $D^j_{\mathrm{DPFR},i}$ the axial dispersion coefficient. The DPFR model also requires inlet and outlet boundary conditions:

$$\left.\frac{\partial c^j_{\mathrm{DPFR},i}}{\partial z}\right|_{z=0} = \frac{u^j_{\mathrm{DPFR},i}}{D^j_{\mathrm{DPFR},i}}\left(c^j_{\mathrm{DPFR},i}(z=0) - c^j_{\mathrm{in},i}\right) \tag{12a}$$

$$\left.\frac{\partial c^j_{\mathrm{DPFR},i}}{\partial z}\right|_{z=L} = 0 \tag{12b}$$

2.5. Purity and productivity

Purity and productivity are commonly applied indicators for evaluating the performance of SMB processes. Both indicators are determined at a time $t = k t_s$, with $k \in \mathbb{N}$, when the system is in CSS. Their definition is based on concentrations $\bar{c}^j_{\mathrm{out},i}$ of component i at node j, averaged over one switching time t_s. The purity Pu^j_i of a component i withdrawn at node j is the average concentration of this component relative to the average total concentra-

tion of all components; the productivity $\mathrm{Pr}_{i,i'}^{j,j'}$ of a binary separation with components i and i' withdrawn at nodes j and j' is the combined withdrawn mass of these components at their respective ports relative to the total volume of the utilized packed bed in all columns:

$$\bar{c}_{\mathrm{out},i}^{j}(k) = \frac{1}{t_s}\int_{t=(k-1)t_s}^{kt_s} c_{\mathrm{out},i}^{j}(t)\,\mathrm{d}t \tag{13a}$$

$$\mathrm{Pu}_{i}^{j}(k) = \frac{\bar{c}_{\mathrm{out},i}^{j}}{\sum_{k=1}^{N}\bar{c}_{\mathrm{out},k}^{j}} \tag{13b}$$

$$\mathrm{Pr}_{i,i'}^{j,j'}(k) = \frac{Q^{j}\bar{c}_{\mathrm{out},i}^{j} + Q^{j'}\bar{c}_{\mathrm{out},i'}^{j'}}{(1-\varepsilon_c)NAL} \tag{13c}$$

An extension to ternary separations is straightforward.

3. Numerical solution

We present and compare four different numerical approaches for simulating SMB processes with a single-column solver. The FPI based methods numerically approximate the CSS, while the OSP based methods compute the full system trajectory from any initial state. Mathematical details of the presented methods are discussed in Appendix A.

3.1. Standard fixed-point iteration (STD-FPI)

STD-FPI requires an initial guess of the sought CSS that must include the current state in all columns and the chromatogram at the outlet of each column for one switching interval $[t, t+t_s]$ of the SMB unit. The initial guess does not necessarily have to be identical with the real system state at initial time or at any other time of the SMB process. If no other information is available, all zero concentrations are usually the most convenient choice.

The STD-FPI method starts with simulating an arbitrarily selected column from time $t=0$ to $t=t_s$, using the chromatogram at the outlet of the upstream column, and updating the chromatogram at the outlet of the simulated column. Depending on the column position in the network, feed or desorbent streams have to be considered in the mass balance of the upstream node, according to Eq. (7). Next, the adjacent column in downstream direction is simulated for the same time interval, and so on. After the round is completed, i.e., when all N columns have been simulated exactly once, the column positions are switched according to the modeled SMB scheme.

In the second round, each column is successively simulated from time $t=t_s$ to $t=2t_s$, following the same procedure as described in the previous paragraph. Generally, the column to be simulated first can be arbitrarily selected in each round. However, it is most common either to start each round at the same network position, or to follow the initially selected column and start each round at the current position of that column. After each round, the column positions are switched and the next round is simulated. A full cycle of the SMB unit (i.e., N such rounds) constitutes one iteration of the STD-FPI method.

Due to unavoidable numerical errors (e.g., round-off) the CSS will not be reached exactly in a simulation, but only approximately. Hence, the STD-FPI method is stopped when the difference between the results of two successive iterations falls below a prespecified threshold $\varepsilon > 0$. This difference can either be measured using the column states at specific times Eq. (14a) or the chromatograms at raffinate and extract ports Eq. (14b):

$$\begin{aligned}\Delta_z(k) = &\max_{j\in\{1,\dots,N\}}\sum_{i=1}^{M} \\ &\times\left(\int_0^L \left|c_i^j(t=kt_s,z) - c_i^j(t=(k-N)t_s,z)\right|^n \mathrm{d}z\right)^{\frac{1}{n}} \\ &\leqslant \varepsilon_z\end{aligned} \tag{14a}$$

$$\begin{aligned}\Delta_t(k) = &\max_{j\in\{R,E\}}\sum_{i=1}^{M} \\ &\times\left(\int_{(k-1)t_s}^{kt_s} \left|c_i^j(t,z=L) - c_i^j(t-Nt_s,z=L)\right|^n \mathrm{d}t\right)^{\frac{1}{n}} \\ &\leqslant \varepsilon_t\end{aligned} \tag{14b}$$

Here, $j=R$ and $j=E$ indicate the columns just upstream of the raffinate and extract ports, respectively, and n determines the type of the norm. When the GRM is used, Eq. (14a) can be extended by accounting for the pore and stationary phase concentrations in the particles $c_{p,i}^j$ and q_i^j.

STD-FPI requires to create and maintain N instances of the single-column solver, as the current state of all columns needs to be preserved while the other columns are simulated. Alternatively, these states can be stored for re-initializing a single solver instance. However, this comes at the cost of an increased computational overhead.

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

In CSS, all columns experience exactly the same system state, but with a time lag, that means, column j is in the same state at time τ as column $j+1$ at time $\tau+t_s$, or column $j+2$ at time $\tau+2t_s$, and so on. Note that the indices are attached to the columns which cycle through the different positions in the zones. Similar to the physical one-column analog, this fact can be utilized for determining the CSS using only one column state and, hence, only one column simulator instance.

OCA-FPI also requires an initial guess of the CSS, but the current state of only one arbitrarily selected column is required. The initial guess still has to include the initial chromatogram at the outlet of each column for one switching interval $[t, t+t_s]$ of the SMB unit. The OCA-FPI method starts by simulating the selected column at its initial position from time $t=0$ to $t=t_s$. As in the STD-FPI method, the chromatogram at the outlet of the upstream column is used, considering feed or desorbent streams where necessary, and the chromatogram at the outlet of the simulated column is updated. Next, the column position is switched according to the modeled SMB scheme.

Then, the same column, at its new position, is simulated from time $t=t_s$ to $t=2t_s$. Due to the periodicity of the sought CSS, the required chromatogram at the outlet of the upstream column is assumed to be identical to the one between time $t=0$ and $t=t_s$, and the same is stipulated for the updated chromatogram at the outlet of the simulated column. The procedure is repeated, as illustrated in Fig. 7, until the selected column has been simulated at each of the N positions in the network. This constitutes one cycle of the physical OCA process and one iteration of the associated numerical method (Abunasser and Wankat, 2004; Abunasser et al., 2003; Mota and Araújo, 2005).

OCA-FPI computes the same CSS but is easier to implement than STD-FPI, as it eliminates one loop and requires only one instance of the single-column solver. In addition, the simulation time for each iteration is reduced by the factor N, which can be significant for systems with many columns. However, OCA-FPI may

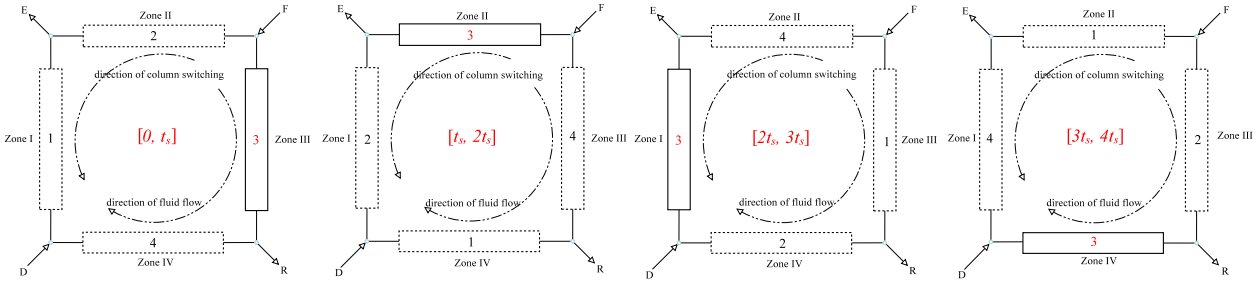

Fig. 7. Schematic of OCA-FPI method applied to four-zone SMB process with four columns. Only the third column is simulated in this example. Periodicity of t_s is assumed for all chromatograms.

require more iterations than STD-FPI for approximating the CSS with the same accuracy. The numerical OCA-FPI method resembles a physical OCA process with ideal plug-flow devices (i.e., without dispersion). The real OCA process with dispersion can be modeled by adding CSTR models to the column network. OCA realizations with holding tanks can be simulated by averaging the inlet boundary conditions over each switching interval, which obviously deteriorates separation performance of the SMB unit.

3.3. Standard operator-splitting (STD-OSP)

OSP allows to simulate the full trajectory of SMB units from any initial system state. This is particularly important for studying ramp-up and shut-down phases. OSP can also handle process variations, e.g., in feed and/or desorbent composition, that prevent the system from entering a CSS. Similar to FPI, OSP is based on weak coupling of individual column models, but the time coordinate is partitioned into much smaller intervals, $t_{\mathrm{OSP}} = \frac{t_s}{m}$ with $1 < m \in \mathbb{N}$, and one of the node mass balances is handled differently. An initial guess is not required, only the initial state of all columns at time $t = 0$ needs to be known.

STD-OSP starts with simulating an arbitrarily chosen column at position j^* from time $t = 0$ to $t = t_{\mathrm{OSP}}$. The required outlet concentrations of the upstream column are assumed to be zero in this time interval. This assumption is clearly justified when all columns are empty at time $t = 0$, which is the most common initial condition. Otherwise, the first column downstream of the desorbent port is a good choice to start with, as the adjacent column upstream of that port has just been regenerated. In contrast to FPI, the simulated chromatograms are not overwritten but continuously recorded.

The recorded chromatogram is used for simulating the next column in downstream direction, and this procedure is repeated until all columns have been simulated from time $t = 0$ to $t = t_{\mathrm{OSP}}$. Now, the next column in downstream direction is where the method has started, and this column is simulated from time t_{OSP} to $2t_{\mathrm{OSP}}$. However, at this (and only this) network position j^*, the previously recorded chromatogram of the upstream column between time $t = 0$ and $t = t_{\mathrm{OSP}}$ is used, that is, Eq. (7) is replaced by

$$c_{\mathrm{in},i}^{j^*}(t) = \frac{c_{\mathrm{out},i}^{j^*-1}(t - t_{\mathrm{OSP}})Q^{j^*-1}(t) + \delta^{j^*-1}(t)}{Q^{j^*}(t)} \tag{15}$$

This time-shift obviously introduces an error that decreases with increasing m. The remaining $N - 1$ columns in downstream direction are successively simulated from time t_{OSP} to $2t_{\mathrm{OSP}}$, using the chromatogram of the respective upstream column in the same time interval, that is, as specified by Eq. (7). This procedure is repeated until the switching time t_s is reached for all columns. After the columns are switched, the STD-OSP method continues with simulating the column at the initially selected network position, using Eq. (15) for the node mass balance, and so on.

In case a CSS is sought, the stopping criterion is again based on Eq. (14). STD-OSP requires a separate solver instance for each column, which are successively started and stopped without re-initialization. The computational effort of the STD-OSP method is comparable to the STD-FPI method, since the same number of columns is simulated over the same time intervals in each iteration (i.e., a full cycle of the SMB unit). However, the STD-OSP method yields much more comprehensive information, that is, the entire trajectory instead of only the CSS.

3.4. Lag-aware operator-splitting (LAW-OSP)

The LAW-OSP method takes advantage of the time-lag between changes at a column inlet and responses at the respective column outlet. In an ideal column without dispersion, this would be the residence time of a non-binding tracer, $t_r = \frac{L}{u_{\mathrm{int}}}$. Even with dispersion, the simulated chromatogram can be assumed to be practically unaffected by the inlet concentration between time $t = 0$ and $t = t_{\mathrm{OSP}} \leq \frac{t_r}{2}$. However, the internal state of that column is affected. Hence, in LAW-OSP the column j^* that is simulated first in each round is simulated again at the end of that round for correcting the internal state. As both simulations must start from the same internal column state, this state has to be saved before the first simulation is performed and restored after that.

For example, consider that column j^* is selected to be simulated first. LAW-OSP starts with storing the internal state of column j^*. Then, column j^* is simulated from time $t = 0$ to t_{OSP} using the same inlet profile as in STD-OSP. Actually, the inlet profile hardly matters, as the column state is reset to the stored values after that simulation and only the simulated chromatogram is recorded, which is practically unaffected by the inlet profile. Next, the other $N - 1$ columns are successively simulated from time $t = 0$ to t_{OSP}, as in STD-OSP. The round is completed by restoring the state of column j^* from memory and simulating it again, this time using the previously recorded chromatogram of the adjacent upstream column $j^* - 1$. Since this simulation is based on the correct inlet profile, it yields the correct internal column state.

After switching column positions, the following round is computed analogously, and so on. In LAW-OSP, the impact of using Eq. (15) instead of Eq. (7) practically vanishes for $m = \lceil 2\frac{t_s}{t_r} \rceil$, which is the smallest integer larger than $2\frac{t_s}{t_r}$. Consequently, the limit $m \to \infty$ is obviated at the cost of an additional column simulation, that is, $N + 1$ instead of N simulations in each round. Smaller values of m help to reduce the computational overhead of repeatedly starting the simulator instances for very small time intervals. Furthermore, LAW-OSP eliminates the need to determine suitable values of m for approximating the sought system trajectories with sufficient accuracy.

Note that LAW-OSP would be exact for models with finite speed of information propagation (e.g., pure convection) provided exact solvers for the partial differential equations are given. Since LAW-

OSP requires the initial values of the first stage in addition to its results for solving the first unit operation again in the second stage, a straight-forward formulation of the method in the operator splitting framework is not possible.

3.5. Simulation software

We have implemented the different solution approaches in a comprehensive simulation software, CADET-SMB, that is based on the Chromatography Analysis and Design Toolkit (CADET). CADET provides a fast and accurate solver for various single column models (von Lieres and Andersson, 2010; Püttmann et al., 2013). CADET and CADET-SMB are both published as open-source software and freely distributed under the GPL license (http://github.com/modsim/cadet, http://github.com/modsim/cadet-smb). CADET-SMB is readily set up for simulating the previously introduced SMB processes with four, five, and eight zones. Moreover, the code can be adapted to any network configuration, including applications beyond SMB chromatography. CADET-SMB can be interfaced to any transport and binding model provided in CADET, which is continuously being extended. Hold-up volumes between the columns and at the ports can optionally be considered.

CADET-SMB also includes routines for maximizing performance indicators, such as purity and productivity. As triangle theory is limited to linear binding models, fast and accurate numerical simulations are required in the general case (Agrawal and Kawajiri, 2015; Agrawal et al., 2014; Engell and Toumi, 2004; Kazi et al., 2012; Lim, 2004; Sreedhar et al., 2014). Both continuous variables (e.g., concentrations, flowrates, switching times) and discrete variables (e.g., network connectivity) can be optimized using different algorithms. Details will be subject of a separate publication.

4. Case studies

Five different case studies are used for demonstrating and comparing the introduced numerical solution methods. In all these studies, the columns are assumed to be initially empty. Moreover, the outlet concentrations are assumed to be zero in the initial time intervals $[0, t_s]$ or $[0, t_{OSP}]$, respectively. Table 1 shows the complete model parameters for all case studies. In the following, the index j can either indicate a specific column ($j \in \{1, \dots, N\}$) or the zone a column is currently in ($j \in \{\mathrm{I}, \dots, \mathrm{VIII}\}$). If not specified otherwise, the same parameters are used for each component in each column. Pure buffer is used as desorbent in all case studies, i.e., $c^D_{\mathrm{in}} = 0\,\mathrm{g\,cm^{-3}}$. All parameter values have been rounded to three significant digits.

The case studies are based on the EDM. CADET has been originally designed for solving the GRM but can be adapted for solving the EDM as follows: Eq. (1b) is spatially discretized using the finite volume method with only one radial cell, $N_r = 1$. A very small particle porosity is used, $\varepsilon_p = 10^{-5}$. Consequently, the column porosity is practically equal to the total porosity, $\varepsilon_c = \frac{\varepsilon_t - \varepsilon_p}{1-\varepsilon_p}$. The effective pore diffusion coefficient and the film mass transfer coefficient are made sure not to be rate limiting, $D_p = 5 \cdot 10^{-5}\,\mathrm{m^2\,s^{-1}}$ and $k_f = 1.6 \cdot 10^4\,\mathrm{m\,s^{-1}}$.

The axial column dimension is discretized using $N_z = 40$ axial cells. The resulting system of ordinary differential equations is solved using an absolute tolerance of 10^{-10}, relative tolerance of 10^{-6}, an initial step size of 10^{-14} and a maximal step size of $5 \cdot 10^6$.

4.1. Case study I

The first case study is taken from Klatt et al. (2002) and describes the separation of fructose ($i = 1$) and glucose ($i = 2$) using a standard four-zone scheme with eight columns and a linear isotherm. The inlet concentrations are converted from $c^F_{\mathrm{in}} = 0.55\,\mathrm{g\,cm^{-3}}$, assuming that fructose and glucose have the same molar mass of $180\,\mathrm{g\,mol^{-1}}$. The cross-sectional column area, $A = \frac{\pi d^2}{4}$, is calculated from the column diameter, $d = 2.6 \cdot 10^{-2}$ m. The axial dispersion coefficient, D_{ax}, is not given by Klatt et al. (2002), and hence a typical value is used. The volumetric flowrates in zones II–IV are calculated from the volumetric flowrates in zone I and at the feed, desorbent, raffinate, and extract ports. The interstitial flowrates in zones I–IV are calculated from the respective volumetric flowrates, $u^j_{\mathrm{int}} = \frac{Q^j}{\varepsilon_c A}$.

4.2. Case study II

The second study is taken from Lübke et al. (2007). The standard four-zone scheme with four columns and a linear isotherm is applied for separating a hypothetic binary mixture. The particle radius, r_p, is not given by Lübke et al. (2007), and hence a typical value is used. In Lübke et al. (2007), the axial dispersion coefficient, D_{ax}, depends on the linear flowrate, u_{int}; however, in case II the same axial dispersion coefficient is used for all columns. A molar mass of $10^3\,\mathrm{g\,mol^{-1}}$ is assumed in order to get $1\,\mathrm{g\,cm^{-3}} = 10^3\,\mathrm{mol\,m^{-3}}$. The inlet concentrations are converted from $c^F_{\mathrm{in}} = 0.55\,\mathrm{g\,cm^{-3}}$. The column diameter is $d = 2 \cdot 10^{-2}$ m.

4.3. Case study III

The third case study is taken from Mun (2011). This study describes the ternary separation of 2'-deoxycytidine ($i = 1$), 2'-deoxyguanosine ($i = 2$), and 2'-deoxyadenosine ($i = 3$) using a five-zone scheme. The Henry coefficients of the second and third component are sufficiently different to allow the desired ternary separation with a five-zone scheme. Mun (2011) applies partial-feeding and partial-closing strategies; however, case III is based on the standard five-zone scheme. The inlet concentrations are converted from $c^F_{\mathrm{in}} = 1.0\,\mathrm{g\,cm^{-3}}$ using the molar masses of $227\,\mathrm{g\,mol^{-1}}$, $267\,\mathrm{g\,mol^{-1}}$ and $251\,\mathrm{g\,mol^{-1}}$ for deoxycytidine, deoxyguanosine, and deoxyadenosine, respectively. The Henry coefficients are from the second ternary-separation system and the volumetric flowrates from point a in Mun (2011). The column diameter is $d = 0.01$ m. The axial dispersion coefficient, D_{ax}, is not explicitly given by Mun (2011), and hence a typical value is used.

4.4. Case study IV

The fourth case study is taken from Wang and Ching (2002, 2005) and describes a separation of the most active enantiomer of nadolol from a racemic mixture, which has been reduced from a quaternary to a ternary system in this study. Wang and Ching (2005) compared five-zone schemes with two extract ports and with two raffinate ports. The former is reported to have a higher yield than the latter; while both schemes yield high purities of the desired enantiomer. Case IV is based on the scheme with two extract ports. The axial dispersion coefficient is $D_{ax} = 0.0060\,V = 1.18 \cdot 10^{-7}\,\mathrm{m^2\,s^{-1}}$, where $V = 1.97 \cdot 10^{-5}\,\mathrm{m^3}$ is the column volume. The volumetric flowrates are from case H in Wang and Ching (2005). The inlet concentrations are converted from $c^F_{\mathrm{in}} = 1.0\,\mathrm{g\,cm^{-3}}$, assuming that nadolols have the same molar mass of $309\,\mathrm{g\,mol^{-1}}$.

4.5. Case study V

The fifth case study is a hypothetical variant of case I for demonstrating a ternary separation with an eight-zone scheme. From zone I to IV, the volumetric flowrates are kept same with case I. In zones V, VI and VIII, the same volumetric flowrates are used as in zones I, II and IV, respectively. In zone VII, a volumetric

Table 1
Complete set of model parameters for case studies I–V.

Scheme		Case I	Case II	Case III	Case IV	Case V	Unit
Number of zones		4	4	5	5	8	
Number of columns	N	8	4	5	10	8	
Number of components	M	2	2	3	3	3	
Columns							
Column length	L	$5.36 \cdot 10^{-1}$	$2.50 \cdot 10^{-1}$	$1.50 \cdot 10^{-1}$	$2.50 \cdot 10^{-1}$	$5.36 \cdot 10^{-1}$	m
Column diameter	d	$2.60 \cdot 10^{-2}$	$2.00 \cdot 10^{-2}$	$1.00 \cdot 10^{-2}$	$1.00 \cdot 10^{-2}$	$2.60 \cdot 10^{-2}$	m
Cross section	A	$5.31 \cdot 10^{-4}$	$3.14 \cdot 10^{-4}$	$7.85 \cdot 10^{-5}$	$7.85 \cdot 10^{-5}$	$5.31 \cdot 10^{-4}$	m^2
Column porosity	ε_c	0.38	0.83	0.80	0.47	0.38	
Particle radius	r_p	$1.63 \cdot 10^{-3}$	$5.00 \cdot 10^{-4}$	$1.50 \cdot 10^{-5}$	$7.50 \cdot 10^{-6}$	$1.63 \cdot 10^{-3}$	m
Transport							
Axial dispersion	D_{ax}^{j}	$3.81 \cdot 10^{-6}$	$3.30 \cdot 10^{-26}$	$3.81 \cdot 10^{-10}$	$1.18 \cdot 10^{-7}$	$3.81 \cdot 10^{-6}$	$m^2\ s^{-1}$
Interstitial velocity	u_{int}^{I}	$6.91 \cdot 10^{-4}$	$3.69 \cdot 10^{-3}$	$4.70 \cdot 10^{-3}$	$2.60 \cdot 10^{-3}$	$6.91 \cdot 10^{-4}$	$m\ s^{-1}$
	u_{int}^{II}	$5.19 \cdot 10^{-4}$	$3.10 \cdot 10^{-3}$	$1.70 \cdot 10^{-3}$	$1.60 \cdot 10^{-3}$	$5.19 \cdot 10^{-4}$	$m\ s^{-1}$
	u_{int}^{III}	$6.18 \cdot 10^{-4}$	$3.47 \cdot 10^{-3}$	$9.28 \cdot 10^{-4}$	$1.30 \cdot 10^{-3}$	$6.18 \cdot 10^{-4}$	$m\ s^{-1}$
	u_{int}^{IV}	$4.86 \cdot 10^{-4}$	$2.94 \cdot 10^{-3}$	$1.20 \cdot 10^{-3}$	$1.40 \cdot 10^{-3}$	$4.86 \cdot 10^{-4}$	$m\ s^{-1}$
	u_{int}^{V}			$9.26 \cdot 10^{-4}$	$1.00 \cdot 10^{-3}$	$6.91 \cdot 10^{-4}$	$m\ s^{-1}$
	u_{int}^{VI}					$5.19 \cdot 10^{-4}$	$m\ s^{-1}$
	u_{int}^{VII}					$6.51 \cdot 10^{-4}$	$m\ s^{-1}$
	u_{int}^{VIII}					$4.86 \cdot 10^{-4}$	$m\ s^{-1}$
Binding							
Henry coefficient	H_1	0.54	5.72	3.15	5.34	0.26	
	H_2	0.28	7.70	7.40	6.80	0.43	
	H_3			23.0	11.2	0.60	
Process							
Feed concentration	$c_{in,1}^{F}$	$2.78 \cdot 10^{3}$	$5.50 \cdot 10^{2}$	$4.41 \cdot 10^{3}$	$3.24 \cdot 10^{3}$	$2.78 \cdot 10^{3}$	$mol\ m^{-3}$
	$c_{in.2}^{F}$	$2.78 \cdot 10^{3}$	$5.50 \cdot 10^{2}$	$3.75 \cdot 10^{3}$	$3.24 \cdot 10^{3}$	$2.78 \cdot 10^{3}$	$mol\ m^{-3}$
	$c_{in.3}^{F}$			$3.98 \cdot 10^{3}$	$3.24 \cdot 10^{3}$	$2.78 \cdot 10^{3}$	$mol\ m^{-3}$
Volumetric flowrates	Q^{I}	$1.40 \cdot 10^{-7}$	$9.62 \cdot 10^{-7}$	$2.92 \cdot 10^{-7}$	$9.73 \cdot 10^{-8}$	$1.40 \cdot 10^{-7}$	$m^3\ s^{-1}$
	Q^{II}	$1.05 \cdot 10^{-7}$	$8.08 \cdot 10^{-7}$	$1.05 \cdot 10^{-7}$	$5.95 \cdot 10^{-8}$	$1.05 \cdot 10^{-7}$	$m^3\ s^{-1}$
	Q^{III}	$1.25 \cdot 10^{-7}$	$9.06 \cdot 10^{-7}$	$5.83 \cdot 10^{-8}$	$4.82 \cdot 10^{-8}$	$1.25 \cdot 10^{-7}$	$m^3\ s^{-1}$
	Q^{IV}	$9.81 \cdot 10^{-8}$	$7.66 \cdot 10^{-7}$	$7.50 \cdot 10^{-8}$	$5.23 \cdot 10^{-8}$	$9.81 \cdot 10^{-8}$	$m^3\ s^{-1}$
	Q^{V}			$5.82 \cdot 10^{-8}$	$4.03 \cdot 10^{-8}$	$1.40 \cdot 10^{-7}$	$m^3\ s^{-1}$
	Q^{VI}					$1.05 \cdot 10^{-7}$	$m^3\ s^{-1}$
	Q^{VII}					$1.31 \cdot 10^{-7}$	$m^3\ s^{-1}$
	Q^{VIII}					$9.81 \cdot 10^{-8}$	$m^3\ s^{-1}$
Volumetric flowrates	Q^{F}	$2.00 \cdot 10^{-8}$	$0.98 \cdot 10^{-7}$	$1.67 \cdot 10^{-8}$	$4.17 \cdot 10^{-9}$	$2.00 \cdot 10^{-8}$	$m^3\ s^{-1}$
	Q^{R}	$2.66 \cdot 10^{-8}$	$1.40 \cdot 10^{-7}$	$1.68 \cdot 10^{-8}$	$1.20 \cdot 10^{-8}$	$2.66 \cdot 10^{-8}$	$m^3\ s^{-1}$
	Q^{D}	$4.14 \cdot 10^{-8}$	$1.96 \cdot 10^{-7}$	$2.34 \cdot 10^{-7}$	$5.70 \cdot 10^{-8}$	$4.14 \cdot 10^{-8}$	$m^3\ s^{-1}$
	Q^{E}	$3.48 \cdot 10^{-8}$	$1.54 \cdot 10^{-7}$	$1.88 \cdot 10^{-7}$	$3.78 \cdot 10^{-8}$	$3.48 \cdot 10^{-8}$	$m^3\ s^{-1}$
	$Q^{F\text{-II}}$					$2.66 \cdot 10^{-8}$	$m^3\ s^{-1}$
	$Q^{R\text{-II}}$					$3.32 \cdot 10^{-8}$	$m^3\ s^{-1}$
	$Q^{D\text{-II}}$					$4.14 \cdot 10^{-8}$	$m^3\ s^{-1}$
	$Q^{E\text{-II}}$			$4.64 \cdot 10^{-8}$	$1.13 \cdot 10^{-8}$	$3.48 \cdot 10^{-8}$	$m^3\ s^{-1}$
Switching time	t_s	1552	180	264	1394.4	1552	s

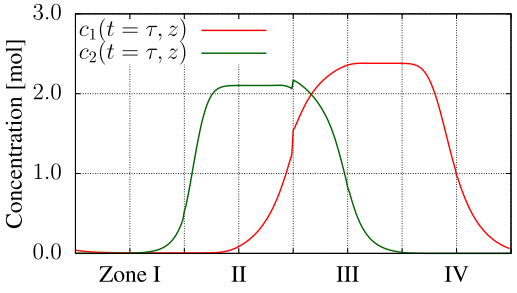

Fig. 8. Axial concentration profiles of case study I in CSS.

flowrate of $Q_{VII} = 1.31 \cdot 10^{-7} m^{-3}\,s^{-1}$ is calculated from the mass balance of the process. The Henry coefficients of all three components are designed using triangle theory such as to facilitate the desired separation.

5. Results and discussion

All benchmarks were computed on an Intel Xeon system with 16 compute cores running at 2.93 GHZ. The numerical solution methods were validated against each other and compared with respect to numerical accuracy and computational efficiency. For instance, Fig. 8 shows the CSS of case I computed by STD-FPI. The combined axial concentration profiles in all columns are displayed at multiples of the switching time. In this example, a total of 104 switches was required to fall below an error of $\varepsilon_t = 10^{-4}$. Fig. 9 shows the corresponding trajectory for the first 40 switching times as computed by LAW-OSP with a time interval of $t_{OSP} = \frac{1552}{5} = 310.4$ s.

5.1. Numerical accuracy

The difference between two numerical approximations $\tilde{c}$ and $\hat{c}$ to the same but unknown solution c can be measured in analogy to Eq. (14). The estimate $\tilde{\Delta}_z(k)$ quantifies the integral difference between axial concentration profiles in all columns at the kth switching time, Eq. (16a), whereas the estimates $\tilde{\Delta}_t(k)$ and $\tilde{\Delta}'_t(k)$ quantify the integral difference between chromatograms at raffinate and extract ports over the kth switching interval, Eq. (16b), or the past k switching intervals, Eq. (16c). In this study, the Euclidean distance is used, $n = 2$. One iteration always refers to a full cycle of

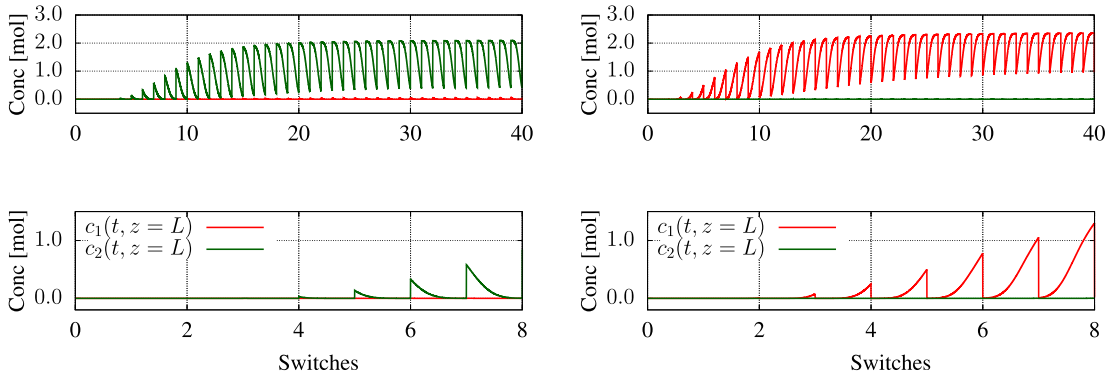

Fig. 9. Trajectories of case study I at extract (left) and raffinate (right) ports. Upper panels show the first 40 switching times (five iterations), lower panels show the first 8 switching times (one iteration).

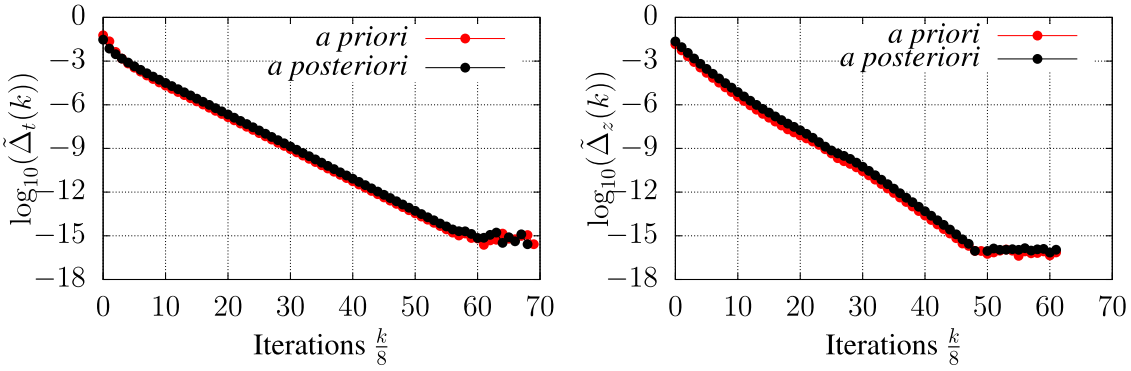

Fig. 10. *A priori* and *a posteriori* error estimates based on chromatograms (left) and axial concentration profiles (right) over number of iterations for STD-FPI and case study I.

the SMB unit,

$$\tilde{\Delta}_z(k) = \sum_{j=1}^{N}\sum_{i=1}^{M}\left(\int_0^L \left|\tilde{c}_i^j(t = k\,t_s, z) - \hat{c}_i^j(t = k\,t_s, z)\right|^n \mathrm{d}z\right)^{\frac{1}{n}} \tag{16a}$$

$$\tilde{\Delta}_t(k) = \max_{j\in\{R,E\}} \sum_{i=1}^{M}\left(\int_{(k-1)t_s}^{kt_s} \left|\tilde{c}_i^j(t, z = L) - \hat{c}_i^j(t, z = L)\right|^n \mathrm{d}t\right)^{\frac{1}{n}} \tag{16b}$$

$$\tilde{\Delta}_t'(k) = \max_{j\in\{R,E\}} \sum_{i=1}^{M}\left(\int_{0}^{kt_s} \left|\tilde{c}_i^j(t, z = L) - \hat{c}_i^j(t, z = L)\right|^n \mathrm{d}t\right)^{\frac{1}{n}} \tag{16c}$$

Before comparing the results of different numerical methods, *a priori* and *a posteriori* error estimates of the individual methods are presented. For instance, Fig. 10 shows both error estimates over the number of iterations for case study I as computed by STD-FPI. The *a priori* estimate allows to monitor the convergence by comparing the results of two successive iterations, while the *a posteriori* estimate provides a more accurate analysis by comparing each iteration to the final result. The measures based on chromatograms, $\tilde{\Delta}_t(k)$, and column states, $\tilde{\Delta}_z(k)$, show similar behavior: the *a priori* estimate is dominated by the *a posteriori* estimate, and the estimates are monotonously decreasing, except for the first and last iterations where they can be distorted by ramp-up effects and numerical roundoff errors.

Fig. 11 shows *a priori* and *a posteriori* error estimates over the number of iterations for case study I as computed by all four numerical methods. Convergence of the OSP methods depends not only on the number of iterations but also on the number of time intervals $m = \frac{t_s}{t_{\mathrm{OSP}}}$. The results in Fig. 11 are computed using $m = 100$ intervals in the STD-OSP method and $m = 5$ intervals in the LAW-OSP method. All methods show the same order of convergence but with different constants, that is, some methods are systematically more accurate when compared for the same number of iterations. STD-FPI is the most simple and easy to implement method but evidently less accurate than the other methods, whose results are similarly accurate as shown in the insets. However, this analysis does not yet take compute time into account, as will be discussed in more detail in Section 5.3.

Fig. 12 shows *a posteriori* error estimates of the two OSP methods for different numbers of time intervals as computed for case study I. The reference solution is computed using the LAW-OSP method with $m = 500$ time intervals. The trajectories are compared for $k = 100$ switching times. As expected, the accuracy of the LAW-OSP method is not affected by the number of time intervals, as long as a minimum requirement is met, while the accuracy of the STD-OSP method monotonously decreases with the number of intervals. Depending on the overhead of restarting the simulation code, large numbers of time intervals can potentially cause excessive computation times.

5.2. Cross validation

The numerical methods and their software implementations in CADET-SMB are validated against each other by direct comparison of simulation results for all five cases. Table 2 shows the mutual differences between all four methods. As before, $m = 100$ time intervals were used for STD-OSP and $m = 5$ for LAW-OSP. The simulations were performed with a stopping criterion of $\varepsilon_t = 10^{-4}$. As the number of iterations can vary between the methods, the

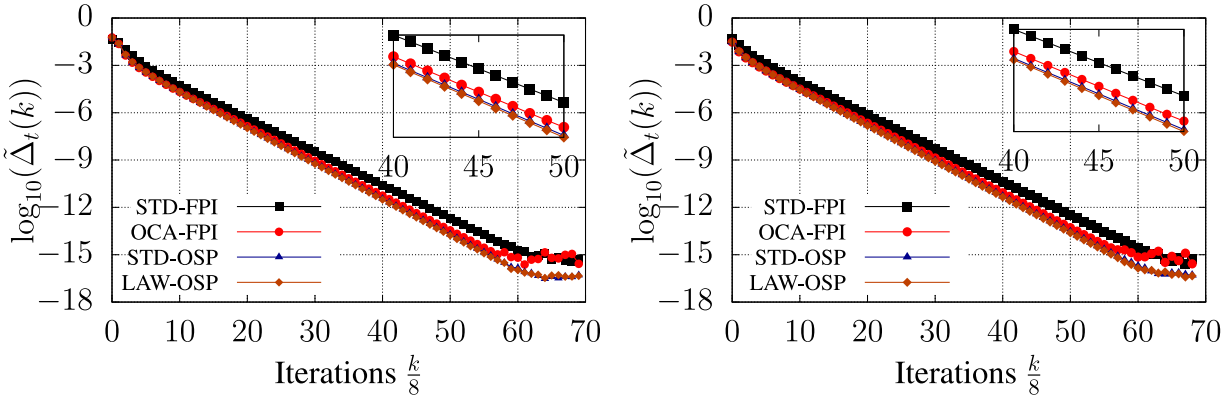

Fig. 11. *A priori* (left) and *a posteriori* (right) error estimates over number of iterations for all four numerical methods and case study I.

Table 2
Differences between the results of all four numerical methods with a stopping criterion of $\varepsilon_t = 10^{-14}$.

	Case I	Case II	Case III	Case IV	Case V
$\tilde{\Delta}_z(k)$					
STD-FPI vs. OCA-FPI	$2.54 \cdot 10^{-16}$	$2.41 \cdot 10^{-15}$	$1.47 \cdot 10^{-11}$	$7.32 \cdot 10^{-10}$	$1.04 \cdot 10^{-14}$
STD-FPI vs. STD-OSP	$6.90 \cdot 10^{-4}$	$2.51 \cdot 10^{-5}$	$8.42 \cdot 10^{-6}$	$3.26 \cdot 10^{-4}$	$2.35 \cdot 10^{-4}$
STD-FPI vs. LAW-OSP	$2.72 \cdot 10^{-7}$	$7.73 \cdot 10^{-8}$	$6.00 \cdot 10^{-8}$	$2.31 \cdot 10^{-7}$	$7.94 \cdot 10^{-8}$
OCA-FPI vs. STD-OSP	$6.90 \cdot 10^{-4}$	$2.51 \cdot 10^{-5}$	$8.42 \cdot 10^{-6}$	$3.26 \cdot 10^{-4}$	$2.35 \cdot 10^{-4}$
OCA-FPI vs. LAW-OSP	$2.72 \cdot 10^{-7}$	$7.73 \cdot 10^{-8}$	$6.00 \cdot 10^{-8}$	$2.31 \cdot 10^{-7}$	$7.94 \cdot 10^{-8}$
STD-OSP vs. LAW-OSP	$6.90 \cdot 10^{-4}$	$2.51 \cdot 10^{-5}$	$8.37 \cdot 10^{-6}$	$3.27 \cdot 10^{-4}$	$2.35 \cdot 10^{-4}$
$\|\|c\|\|_2$	0.12	0.10	0.18	0.10	0.12
$\tilde{\Delta}_t(k)$					
STD-OSP vs. LAW-OSP	$3.02 \cdot 10^{-3}$	$1.58 \cdot 10^{-3}$	$1.47 \cdot 10^{-3}$	$8.11 \cdot 10^{-3}$	$6.05 \cdot 10^{-3}$
$\|\|c\|\|'_2$	1.55	1.18	2.04	1.26	1.25

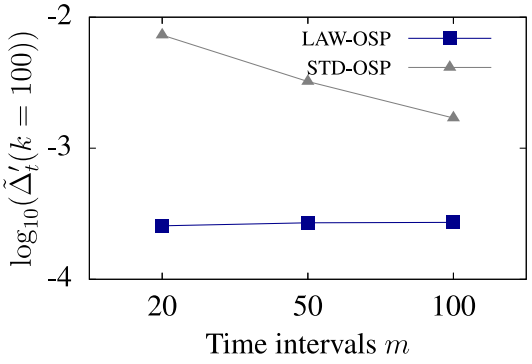

Fig. 12. *A posteriori* error estimates over number of time intervals for both OSP methods and case study I.

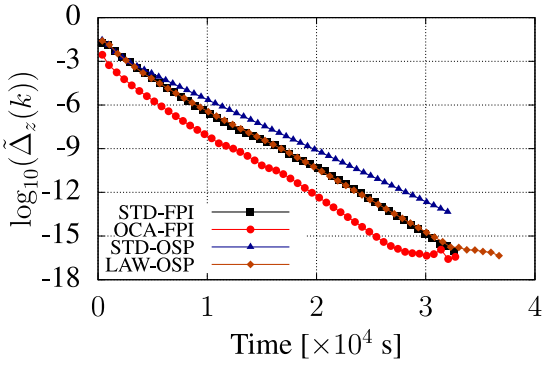

Fig. 13. *A priori* error estimate over computational time for all four numerical methods and case study I.

difference $\tilde{\Delta}_z(k)$ is calculated between the axial concentration profiles at the last switching time, derogation from Eq. (16b). The norm of the true solution, $||c||_2$, is approximated by applying Eq. (16a) to the solution of STD-FPI and zero concentration. It is close to 0.1 for all case studies. Hence, the solutions of all methods match with an accuracy of at least 2–3 significant digits. The accuracy of all methods can be increased by decreasing the stopping criterion ε_t.

In addition, the trajectories $\tilde{\Delta}'_t(k)$ that are calculated by the OSP methods are compared over the first 40 switching times. The respective norm of the true solution, $||c||'_2$, is approximated by applying Eq. (16c) to the solution of STD-OSP and zero concentration. It is close to 1.0 for all case studies. Hence, the solutions of the OSP methods match with an accuracy of at least 3–4 significant digits.

5.3. Computational efficiency

The presented numerical methods differ not only in their accuracy but also in their computational efficiency. Fig. 13 shows numerical accuracy over computational time for case I. Each symbol represents a progression of the respective method by one iteration or eight switching times. Although CADET is capable of parallel computing, only one thread is used here for comparing the different numerical methods. Clearly, OCA-FPI is more efficient than STD-FPI, and LAW-OSP is more efficient than STD-OSP. In other words, both standard methods could be significantly improved.

From another point of view, STD-FPI is more efficient than STD-OSP, and OCA-FPI is more efficient than LAW-OSP. In other words, both FPI methods are more efficient than their OSP counterparts. STD-OSP is outperformed by all other methods. The per-

Table 3
Runtimes of all four numerical methods for different stopping criteria.

	Case I	Case II	Case III	Case IV	Case V
$\varepsilon_t = 10^{-3}$					
STD-FPI	$5.69 \cdot 10^3$	$2.97 \cdot 10^3$	$3.56 \cdot 10^3$	$5.86 \cdot 10^3$	$8.48 \cdot 10^3$
OCA-FPI	$2.81 \cdot 10^3$	$2.24 \cdot 10^3$	$2.59 \cdot 10^3$	$4.77 \cdot 10^3$	$4.29 \cdot 10^3$
STD-OSP	$5.76 \cdot 10^3$	$3.20 \cdot 10^3$	$3.31 \cdot 10^3$	$5.86 \cdot 10^3$	$1.12 \cdot 10^4$
LAW-OSP	$6.28 \cdot 10^3$	$3.54 \cdot 10^3$	$4.11 \cdot 10^3$	$5.83 \cdot 10^3$	$1.07 \cdot 10^4$
$\varepsilon_t = 10^{-4}$					
STD-FPI	$8.38 \cdot 10^3$	$3.67 \cdot 10^3$	$4.35 \cdot 10^3$	$5.90 \cdot 10^3$	$1.31 \cdot 10^4$
OCA-FPI	$5.40 \cdot 10^3$	$3.13 \cdot 10^3$	$3.44 \cdot 10^3$	$5.69 \cdot 10^3$	$6.01 \cdot 10^3$
STD-OSP	$1.02 \cdot 10^4$	$5.19 \cdot 10^3$	$4.11 \cdot 10^3$	$5.85 \cdot 10^3$	$1.55 \cdot 10^4$
LAW-OSP	$9.28 \cdot 10^3$	$4.42 \cdot 10^3$	$5.08 \cdot 10^3$	$5.84 \cdot 10^3$	$1.36 \cdot 10^4$
$\varepsilon_t = 10^{-5}$					
STD-FPI	$1.17 \cdot 10^4$	$4.83 \cdot 10^3$	$5.26 \cdot 10^3$	$7.24 \cdot 10^3$	$1.53 \cdot 10^4$
OCA-FPI	$8.35 \cdot 10^3$	$3.89 \cdot 10^3$	$4.24 \cdot 10^3$	$6.50 \cdot 10^3$	$8.89 \cdot 10^3$
STD-OSP	$1.39 \cdot 10^4$	$6.81 \cdot 10^3$	$4.78 \cdot 10^3$	$7.10 \cdot 10^3$	$1.88 \cdot 10^4$
LAW-OSP	$1.29 \cdot 10^4$	$5.57 \cdot 10^3$	$6.09 \cdot 10^3$	$7.11 \cdot 10^3$	$1.62 \cdot 10^4$
$\varepsilon_t = 10^{-6}$					
STD-FPI	$1.49 \cdot 10^4$	$6.28 \cdot 10^3$	$6.01 \cdot 10^3$	$8.39 \cdot 10^3$	$1.77 \cdot 10^4$
OCA-FPI	$1.01 \cdot 10^4$	$4.72 \cdot 10^3$	$5.05 \cdot 10^3$	$7.62 \cdot 10^3$	$1.08 \cdot 10^4$
STD-OSP	$1.76 \cdot 10^4$	$8.63 \cdot 10^3$	$5.55 \cdot 10^3$	$8.31 \cdot 10^3$	$2.32 \cdot 10^4$
LAW-OSP	$1.59 \cdot 10^4$	$6.47 \cdot 10^3$	$7.08 \cdot 10^3$	$8.44 \cdot 10^3$	$1.70 \cdot 10^4$

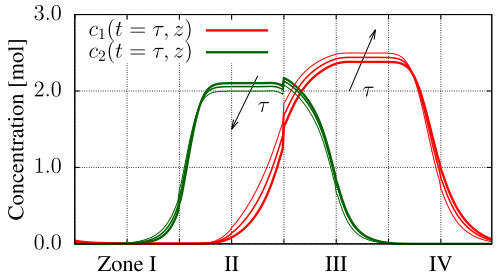

Fig. 14. Impact of hold-up volumes on axial concentration profiles of case study I in CSS.

formance of STD-FPI is similar to LAW-OSP, but the latter yields much richer information by following the correct trajectory from the initial values to the CSS. OCA-FPI outperforms all other methods if the trajectory is not required and only the CSS needs to be known. Table 3 illustrates that these trends are similar for all case studies.

5.4. Hold-up volumes

CADET-SMB allows to consider hold-up volumes in the column network. This is demonstrated by introducing CSTR models, Eq. (10), in case study I as illustrated by Fig. 5. The residence time, τ_{CSTR}, is varied between 0s, 5s and 10s. Fig. 14 shows the impact of these hold-up volumes on the column states in CSS.

6. Conclusions

We have presented and compared four numerical solution methods that are based on weak coupling of single column models. Although weak coupling is usually less efficient than strong coupling, it can deliver equally accurate results. Weak coupling has the major advantage that existing simulators for single chromatography columns can be interfaced without modifying their respective computational codes. Weak coupling can outperform strong coupling, if only the cyclic steady state (CSS) is of interest and the one column analog (OCA) can be applied. Other models can be integrated, for instance to consider the impact of hold-up volumes.

The standard approaches for fixed point iteration (FPI) and operator splitting (OSP) are found to be suboptimal, and more efficient alternatives are presented. The OCA has previously been published, while the lag aware (LAW) approach to OSP appears to be novel, at least in the field of chromatography modeling. The presented non-standard approaches allow to either compute the correct trajectory in the same time as the CSS or to compute the CSS much faster than with STD-FPI. Hence, OCA-FPI should should be preferred if only the CSS is desired, and LAW-OSP should be preferred if the full dynamic trajectory is desired. All methods are implemented in a comprehensive simulation software that is based on the CADET project and published as open source code. The presented case studies are available as part of the CADET-SMB project and can be modified to match custom user requirements that are by far not limited to simulated moving bed (SMB) chromatography.

Acknowledgments

The first author gratefully acknowledges financial support from the Chinese Scholarship Council (CSC) (grant no. 201408310124).

Appendix A. Mathematical details

A1. Standard fixed-point iteration (STD-FPI)

For a formal mathematical description of the FPI-STD method, we first introduce a mapping F_{col} that takes a column state $c(t)$ at some time t and an inlet profile $c_{\text{prof}}([t, t+t_s])$ from time t to $t+t_s$ and advances the column model in time for t_s resulting in an updated state $c(t+t_s)$ and an outlet chromatogram $c_{\text{out}}([t, t+t_s])$:

$$F_{\text{col}}\left(c(t), c_{\text{prof}}([t, t+t_s])\right) = \left(c(t+t_s), c_{\text{out}}([t, t+t_s])\right) \tag{A.1}$$

This mapping represents the action of an arbitrary column simulator with initial data $c(t)$ and boundary data c_{prof}. Simulating one round requires data for all column states $c^j(t)$ and chromatograms $c^j_{\text{out}}([t, t+t_s])$ for $j = 1, \ldots, N$ in order to advance the columns in time using the outlet chromatogram of the previous column as inlet profile via Eq. (7):

$$\begin{aligned} &F_{\text{sw-STD}}\left(c^1(t), c^1_{\text{out}}([t, t+t_s]), \ldots, c^N(t), c^N_{\text{out}}([t, t+t_s])\right) \\ &= \begin{pmatrix} F_{\text{col}}\left(c^1(t), c^1_{\text{in}}([t, t+t_s])\right) \\ \vdots \\ F_{\text{col}}\left(c^N(t), c^N_{\text{in}}([t, t+t_s])\right) \end{pmatrix} \end{aligned} \tag{A.2}$$

Note that all column models can be run simultaneously since there are no dependencies between the models in this approach. The valve switching mapping S_{STD} is defined by the following permutation:

$$\begin{aligned} &S_{\text{STD}}\left(c^1(t), c^1_{\text{out}}([t, t+t_s]), \ldots, c^N(t), c^N_{\text{out}}([t, t+t_s])\right) \\ &= \begin{pmatrix} c^2(t), c^2_{\text{out}}([t, t+t_s]) \\ \vdots \\ c^N(t), c^N_{\text{out}}([t, t+t_s]) \\ c^1(t), c^1_{\text{out}}([t, t+t_s]) \end{pmatrix} \end{aligned} \tag{A.3}$$

The final $F_{\text{STD-FPI}}$ mapping operates one iteration from time t to $t + Nt_s$:

$$F_{\text{STD-FPI}} = \left[S_{\text{STD}} \circ F_{\text{sw-STD}}\right]^N \tag{A.4}$$

In Eq. (A.4), $\circ$ denotes the sequential execution of the mappings from right to left (i.e., composition of the mappings) and exponentiation refers to repetition. In case the CSS would be known

and used as initial guess, and assuming there are no numerical errors, each iteration of the STD-FPI method would result in exactly the same system state. The CSS is a fixed point of the mapping $F_{\text{STD-FPI}}$, Eq. (A.5), hence the name "fixed point iteration". In all other cases, each iteration results in a new approximation of the CSS.

$$\begin{aligned}
&\begin{pmatrix} c^1(t), c^1_{\text{out}}([t, t+t_s]) \\ \vdots \\ c^N(t), c^N_{\text{out}}([t, t+t_s]) \end{pmatrix} \\
&= \begin{pmatrix} c^1(t+Nt_s), c^1_{\text{out}}([t+Nt_s, t+(N+1)t_s]) \\ \vdots \\ c^N(t+Nt_s), c^N_{\text{out}}([t+Nt_s, t+(N+1)t_s]) \end{pmatrix} \\
&= F_{\text{STD-FPI}}\left(c^1(t), c^1_{\text{out}}([t, t+t_s]), \ldots, c^N(t), c^N_{\text{out}}([t, t+t_s])\right) \quad \text{(A.5)}
\end{aligned}$$

The Banach fixed point theorem (Han and Atkinson, 2009) provides a sufficient condition for convergence of the iteration, which essentially consists of a contraction condition on $F_{\text{STD-FPI}}$ in a suitable norm. However, a large number of iterations can be required for determining the CSS with sufficient accuracy, depending on initial, boundary and operating conditions.

A2. One-column analog fixed-point iteration (OCA-FPI)

For a mathematical description of this process, we define a mapping $F_{\text{sw-OCA},\,k}$ that advances a column at position k forward in time. Using the column simulator mapping F_{col}, running a column at position k for one switching period requires data for the state of that column $c(t)$ and all chromatograms $c^j_{\text{out}}([t, t+t_s])$ for $j = 1, \ldots, N$:

$$\begin{aligned}
&F_{\text{sw-OCA},k}\left(c(t), c^1_{\text{out}}([t, t+t_s]), \ldots, c^N_{\text{out}}([t, t+t_s])\right) \\
&= \begin{pmatrix} \left[F_{\text{col}}\left(c(t), c^k_{\text{in}}([t, t+t_s])\right)\right]_{\text{state}} \\ c^1_{\text{out}}([t, t+t_s]) \\ \vdots \\ c^{k-1}_{\text{out}}([t, t+t_s]) \\ \left[F_{\text{col}}\left(c(t), c^k_{\text{in}}([t, t+t_s])\right)\right]_{\text{out}} \\ c^{k+1}_{\text{out}}([t, t+t_s]) \\ \vdots \\ c^N_{\text{out}}([t, t+t_s]) \end{pmatrix} \quad \text{(A.6)}
\end{aligned}$$

In Eq. (A.6), $c^k_{\text{in}}([t, t+t_s])$ is again calculated by Eq. (7), and $[\cdot]_{\text{state}}$ and $[\cdot]_{\text{out}}$ denote the resulting state and chromatogram of the column simulator map, respectively. The full fixed point mapping is given by

$$\begin{aligned}
F_{\text{OCA-FPI}} = {} & F_{\text{sw-OCA},k+1} \circ \cdots \circ F_{\text{sw-OCA},N} \circ F_{\text{sw-OCA},1} \circ \cdots \circ F_{\text{sw-OCA},k-1} \\
& \circ F_{\text{sw-OCA},k} \quad \text{(A.7)}
\end{aligned}$$

This means that first $F_{\text{sw-OCA},\,k}$ is executed followed by $F_{\text{sw-OCA},k-1}$ and so on. By decreasing the index k including the periodic wrap-around from index 1 to N, the column is moved through the different positions against the direction of the fluid flow. Since $F_{\text{sw-OCA},\,k}$ essentially only requires $c^{k-1}_{\text{out}}([t, t+t_s])$ and k is decreased in the $F_{\text{OCA-FPI}}$ mapping, the chromatograms can be updated in-place. One iteration (i.e., one execution of $F_{\text{OCA-FPI}}$) involves N column simulations and traces one column through a full switching cycle. In the CSS the chromatograms, which are taken at fixed positions and not associated to a moving column, do not change and Eq. (A.10) holds,

$$\begin{aligned}
&\left(c(t+Nt_s), c^1_{\text{out}}([t+t_s, t+2t_s]), \ldots, c^N_{\text{out}}([t+t_s, t+2t_s])\right) \quad \text{(A.8)} \\
&= F_{\text{OCA-FPI}}\left(c(t), c^1_{\text{out}}([t, t+t_s]), \ldots, c^N_{\text{out}}([t, t+t_s])\right) \quad \text{(A.9)} \\
&= \left(c(t), c^1_{\text{out}}([t, t+t_s]), \ldots, c^N_{\text{out}}([t, t+t_s])\right). \quad \text{(A.10)}
\end{aligned}$$

As in the STD-FPI approach, convergence of the iteration can be proven using the Banach fixed point theorem, and convergence of the OCA-FPI method is also checked using Eq. (14).

A3. Standard operator-splitting (STD-OSP)

In general, operator splitting is based on a splitting $\mathcal{A} = \mathcal{A}_1 + \mathcal{A}_2$ of an operator $\mathcal{A}$ into two other operators $\mathcal{A}_1$ and $\mathcal{A}_2$ resulting in a decomposition of the full problem, Eq. (A.11a), into two subproblems, Eq. (A.11b) (Holden et al., 2010).

$$\frac{\partial c}{\partial t} = \mathcal{A}c = (\mathcal{A}_1 + \mathcal{A}_2)c \quad \text{(A.11a)}$$

$$\frac{\partial c}{\partial t} = \mathcal{A}_1 c \quad \text{and} \quad \frac{\partial c}{\partial t} = \mathcal{A}_2 c \quad \text{(A.11b)}$$

Such a splitting is often based on physical phenomena that may require substantially different numerical discretization techniques (e.g., diffusion and convection). By solving the subproblems for short time intervals in an interleaved fashion while transferring the intermediate results between them for use as initial or boundary condition, the solution of the full problem is approximated. One example of such a procedure is given by the sequential splitting:

$$\begin{aligned}
c(k\Delta t) \approx {} & \left[\mathcal{F}_2^{\Delta t}((k-1)\Delta t)\mathcal{F}_1^{\Delta t}((k-1)\Delta t)\right] \\
& \cdots \left[\mathcal{F}_2^{\Delta t}(\Delta t)\mathcal{F}_1^{\Delta t}(\Delta t)\right]\left[\mathcal{F}_2^{\Delta t}(0)\mathcal{F}_1^{\Delta t}(0)\right]c_0 \quad \text{(A.12)}
\end{aligned}$$

In Eq. (A.12), $\mathcal{F}_j^{\Delta t}(t_0)c_0$ denotes the solution of the jth subproblem at time $t_0 + \Delta t$ with initial condition c_0 at $t = t_0$ for some time step $\Delta t = T/m$ and $k = 1, \ldots, m$. Eq. (A.12) means that, after the first subproblem with initial condition c_0 has been solved in $[0, \Delta t]$, the second subproblem is solved with the (final time) result as initial condition producing an approximation of the solution at time Δt. The result of the second subproblem, in turn, is used as initial condition for the next round leading to the solution at time $2\Delta t$ and this procedure is iterated k times.

After (spatial) discretization of the SMB problem the fully coupled system can be written as:

$$\frac{\partial c}{\partial t} = \begin{pmatrix} I_1\left(c^1\right) + B_1\left(c^N_{\text{out}}\right) \\ I_2\left(c^2\right) + B_2\left(c^1_{\text{out}}\right) \\ \vdots \\ I_N\left(c^N\right) + B_N\left(c^{N-1}_{\text{out}}\right) \end{pmatrix} + \begin{pmatrix} g_1(t) \\ g_2(t) \\ \vdots \\ g_N(t) \end{pmatrix} \quad \text{(A.13)}$$

Here, I_j denotes the (spatial) operators of the jth unit operation (e.g., GRM, CSTR, etc.) operating on the associated state vector c^j, B_j connects the jth unit operation's inlet with its predecessor's outlet $j-1$, and g_j represents external boundary conditions given by δ^j in Eq. (8). Note that this form can also be achieved for differential-algebraic equations of differential index 1, which arise for quasi-stationary binding models, using the implicit function theorem. We proceed by introducing an auxiliary variable c_{prof} and partitioning the right hand side of Eq. (A.13) into a sum of two operators such that the full coupling is broken up:

$$\begin{pmatrix} \frac{\partial c^1}{\partial t} \\ \frac{\partial c^2}{\partial t} \\ \vdots \\ \frac{\partial c^N}{\partial t} \\ 0 \end{pmatrix} = \underbrace{\begin{pmatrix} I_1\left(c^1\right) + B_1\left(c_{\text{prof}}\right) \\ I_2\left(c^2\right) + B_2\left(c_{\text{out}}^1\right) \\ \vdots \\ I_N\left(c^N\right) + B_N\left(c_{\text{out}}^{N-1}\right) \\ 0 \end{pmatrix} + \begin{pmatrix} g_1(t) \\ g_2(t) \\ \vdots \\ g_N(t) \\ 0 \end{pmatrix}}_{=:\mathcal{I}\left(t, c^1, \ldots, c^N, c_{\text{prof}}\right)} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ c_{\text{out}}^N - c_{\text{prof}} \end{pmatrix}}_{=:\mathcal{C}\left(c_{\text{out}}^N, c_{\text{prof}}\right)} \tag{A.14}$$

Note that Eq. (A.14) is equivalent to Eq. (A.13), since the additional variable c_{prof} can be easily eliminated using the last equation.

Denote by $F_{\text{STD-OSP, 1}}$ a mapping that takes a chromatogram $c_{\text{prof}}([t, t+t_{\text{OSP}}])$ and the states of all columns $c^j(t)$ at time t and solves the subproblem given in Eq. (A.15) with inlet profile c_{in}^1 computed from $c_{\text{prof}}([t, t+t_{\text{OSP}}])$ via Eq. (7) from t to $t+t_{\text{OSP}}$.

$$\begin{pmatrix} \frac{\partial c}{\partial t} \\ 0 \end{pmatrix} = \mathcal{I}\left(t, c^1, \ldots, c^N, c_{\text{prof}}\right) \tag{A.15}$$

This is equivalent to solving a linear sequence of unit operations, as there is no connection from column N to column 1, from start to finish:

$$\begin{aligned} &F_{\text{STD-OSP,1}}\left(c^1(t), \ldots, c^N(t), c_{\text{prof}}([t, t+t_{\text{OSP}}])\right) \\ &\quad = F_{\text{STD-OSP,1}}^N \circ \cdots \circ F_{\text{STD-OSP,1}}^1 \\ &\quad = \begin{pmatrix} c^1(t+t_{\text{OSP}}) \\ \vdots \\ c^N(t+t_{\text{OSP}}) \\ c_{\text{out}}^N([t, t+t_{\text{OSP}}]) \end{pmatrix} \end{aligned} \tag{A.16}$$

The operators $F_{\text{STD-OSP,1}}^j$ are defined as follows:

$$\begin{aligned} &F_{\text{STD-OSP,1}}^j\left(c^1, \ldots, c^N, c_{\text{prof}}([t, t+t_{\text{OSP}}])\right) \\ &\quad = \begin{pmatrix} c^1(t+t_{\text{OSP}}) \\ \vdots \\ c^{j-1}(t+t_{\text{OSP}}) \\ \left[F_{\text{col}}\left(c^j(t), c_{\text{in}}^j([t, t+t_{\text{OSP}}])\right)\right]_{\text{state}} \\ c^{j+1}(t) \\ \vdots \\ c^N(t) \\ \left[F_{\text{col}}\left(c^j(t), c_{\text{in}}^j([t, t+t_{\text{OSP}}])\right)\right]_{\text{out}} \end{pmatrix} \end{aligned} \tag{A.17}$$

The inlet of column j is taken as the chromatogram in the last slot of the vector. Note that the chromatogram c_{prof} has to contain the inlet profile of the first column at the beginning of the procedure.

Given the chromatogram of the last column $c_{\text{out}}^N([t, t+t_{\text{OSP}}])$, the other subproblem, Eq. (A.18a), is solved by a mapping $F_{\text{STD-OSP, 2}}$ that acts as identity on the column states $c^j(t+t_{\text{OSP}})$ and just copies $c_{\text{out}}^N([t, t+t_{\text{OSP}}])$ into $c_{\text{prof}}([t+t_{\text{OSP}}, t+2t_{\text{OSP}}])$, Eq. (A.18b):

$$\begin{pmatrix} \frac{\partial c}{\partial t} \\ 0 \end{pmatrix} = \mathcal{B}\left(c_{\text{out}}^N, c_{\text{prof}}\right) = \begin{pmatrix} 0 \\ c_{\text{out}}^N - c_{\text{prof}} \end{pmatrix} \tag{A.18a}$$

$$F_{\text{STD-OSP,2}}\left(c^1, \ldots, c^N, c_{\text{out}}^N([t, t+t_{\text{OSP}}])\right) = \text{Id} \tag{A.18b}$$

Based on Eq. (A.12), STD-OSP works by iterating the application of $F_{\text{STD-OSP, 2}} \circ F_{\text{STD-OSP, 1}}$, which is equivalent to iteratively advancing the linear sequence of unit operations in time while using Eq. (15) for the first column in the sequence. Using semigroup theory, it can be shown that under mild conditions the result of the STD-OSP method converges to the system trajectory for $m \to \infty$ (Bátkai et al., 2011).

A4. Lag-aware operator-splitting (LAW-OSP)

This approach is very similar to STD-OSP described above. The only difference in the first step, which solves all columns in sequence, is that the state of the first column has to be captured in an auxiliary variable c_{lag}^1. The second step, however, differs significantly since the first column has to be solved again using the previously captured state but the correct inlet profile from the last column.

In order to express this mathematically, an equation for c_{lag}^1 has to be added and the two (split-)operators adapted:

$$\begin{pmatrix} \frac{\partial c^1}{\partial t} \\ \frac{\partial c^2}{\partial t} \\ \vdots \\ \frac{\partial c^N}{\partial t} \\ \frac{\partial c_{\text{lag}}^1}{\partial t} \\ 0 \end{pmatrix} = \underbrace{\begin{pmatrix} I_1\left(c_{\text{lag}}^1\right) + B_1\left(c_{\text{prof}}\right) \\ I_2\left(c^2\right) + B_2\left(c_{\text{out}}^1\right) \\ \vdots \\ I_N\left(c^N\right) + B_N\left(c_{\text{out}}^{N-1}\right) \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} g_1(t) \\ g_2(t) \\ \vdots \\ g_N(t) \\ 0 \\ 0 \end{pmatrix}}_{=:\mathcal{I}_{\text{lag}}\left(t, c^1, \ldots, c^N, c_{\text{lag}}^1, c_{\text{prof}}\right)} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ I_1\left(c_{\text{lag}}^1\right) + B_1\left(c_{\text{out}}^N\right) + g_1(t) \\ c_{\text{out}}^N - c_{\text{prof}} \end{pmatrix}}_{=:\mathcal{C}_{\text{lag}}\left(c_{\text{out}}^N, c_{\text{lag}}^1, c_{\text{prof}}\right)} \tag{A.19}$$

At the beginning of the procedure the auxiliary state c_{lag}^1 is required to equal the state of the first column, i.e., $c_{\text{lag}}^1(0) = c^1(0)$, and the chromatogram c_{prof} has to contain the inlet profile of the first column.

As for STD-OSP, we state the mappings $F_{\text{LAW-OSP, 1}}$ and $F_{\text{LAW-OSP, 2}}$ that constitute one iteration of the LAW-OSP method. The first subproblem is solved by the mapping $F_{\text{LAW-OSP, 1}}$:

$$\begin{aligned} &F_{\text{LAW-OSP,1}}\left(c^1(t), \ldots, c^N(t), c_{\text{lag}}^1(t), c_{\text{prof}}([t, t+t_{\text{OSP}}])\right) \\ &\quad = F_{\text{LAW-OSP,1}}^N \circ \cdots \circ F_{\text{LAW-OSP,1}}^1 \\ &\quad = \begin{pmatrix} c^1(t+t_{\text{OSP}}) \\ \vdots \\ c^N(t+t_{\text{OSP}}) \\ c_{\text{lag}}^1(t) \\ c_{\text{out}}^N([t, t+t_{\text{OSP}}]) \end{pmatrix} \end{aligned} \tag{A.20}$$

For $j = 2, \ldots, N$ the operators $F_{\text{LAW-OSP,1}}^j$ are defined as follows:

$$F_{\text{LAW-OSP,1}}^j\left(c^1, \ldots, c^N, c_{\text{lag}}^1, c_{\text{prof}}([t, t+t_{\text{OSP}}])\right)$$

$$= \begin{pmatrix} c^1(t+t_{\mathrm{OSP}}) \\ \vdots \\ c^{j-1}(t+t_{\mathrm{OSP}}) \\ \left[F_{\mathrm{col}}\left(c^j(t), c^j_{\mathrm{in}}([t,t+t_{\mathrm{OSP}}])\right)\right]_{\mathrm{state}} \\ c^{j+1}(t) \\ \vdots \\ c^N(t) \\ c^1_{\mathrm{lag}}(t) \\ \left[F_{\mathrm{col}}\left(c^j(t), c^j_{\mathrm{in}}([t,t+t_{\mathrm{OSP}}])\right)\right]_{\mathrm{out}} \end{pmatrix} \tag{A.21}$$

The case $j = 1$ is special due to the handling of the lagged state c^1_{lag}:

$$F^1_{\text{LAW-OSP},1}\left(c^1,\ldots,c^N,c^1_{\mathrm{lag}},c_{\mathrm{prof}}([t,t+t_{\mathrm{OSP}}])\right) = \begin{pmatrix} \left[F_{\mathrm{col}}\left(c^1_{\mathrm{lag}}(t), c^1_{\mathrm{in}}([t,t+t_{\mathrm{OSP}}])\right)\right]_{\mathrm{state}} \\ c^2(t) \\ \vdots \\ c^N(t) \\ c^1_{\mathrm{lag}}(t) \\ \left[F_{\mathrm{col}}\left(c^j(t), c^1_{\mathrm{in}}([t,t+t_{\mathrm{OSP}}])\right)\right]_{\mathrm{out}} \end{pmatrix} \tag{A.22}$$

The inlet of column j is taken as the chromatogram in the last slot of the vector and the lagged state of the first column c^1_{lag} is just copied. Note that $c^1_{\mathrm{in}}([t,t+t_{\mathrm{OSP}}])$ uses the chromatogram $c_{\mathrm{prof}}([t,t+t_{\mathrm{OSP}}])$.

Given the chromatogram of the last column $c^N_{\mathrm{out}}([t,t+t_{\mathrm{OSP}}])$ and the lagged state of the first column at the beginning of the iteration c^1_{lag}, the second subproblem is solved by a mapping $F_{\text{LAW-OSP, 2}}$ that acts as identity on the column states $c^j(t+t_{\mathrm{OSP}})$, simulates the first column using the lagged state again and just copies $c^N_{\mathrm{out}}([t,t+t_{\mathrm{OSP}}])$ into $c_{\mathrm{prof}}([t+t_{\mathrm{OSP}},t+2t_{\mathrm{OSP}}])$:

$$F_{\text{LAW-OSP},2}\left(c^1,\ldots,c^N,c^1_{\mathrm{lag}},c^N_{\mathrm{out}}([t,t+t_{\mathrm{OSP}}])\right) = \begin{pmatrix} c^1(t+t_{\mathrm{OSP}}) \\ \vdots \\ c^N(t+t_{\mathrm{OSP}}) \\ \left[F_{\mathrm{col}}\left(c^1_{\mathrm{lag}}(t), c^1_{\mathrm{in}}([t,t+t_{\mathrm{OSP}}])\right)\right]_{\mathrm{state}} \\ c^N_{\mathrm{out}}([t,t+t_{\mathrm{OSP}}]) \end{pmatrix} \tag{A.23}$$

Note that $c^1_{\mathrm{in}}([t,t+t_{\mathrm{OSP}}])$ uses the chromatogram $c_{\mathrm{prof}}([t,t+t_{\mathrm{OSP}}])$ of the very beginning of the iteration instead of the most current $c^N_{\mathrm{out}}([t,t+t_{\mathrm{OSP}}])$ which is copied to $c_{\mathrm{prof}}([t+t_{\mathrm{OSP}},t+2t_{\mathrm{OSP}}])$.

Note that the operator splitting framework is probably inappropriate to analyze LAW-OSP since the continuous problem of LAW-OSP in this setting, Eq. (A.19), is not equivalent to the original problem Eq. (A.13). The validity of this (time-)discretization method fundamentally relies on the specific sequential splitting method applied.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.compchemeng.2017.12.022.

References

Abunasser, N., Wankat, P., 2005. Ternary separations with one-column analogs to SMB. Sep. Sci. Technol. 40 (16), 3239–3259.

Abunasser, N., Wankat, P.C., 2004. One-column chromatograph with recycle analogous to simulated moving bed adsorbers: analysis and applications. Indust. Eng. Chem. Res. 43 (17), 5291–5299.

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. Indust. Eng. Chem. Res. 42 (21), 5268–5279.

Agrawal, G., Kawajiri, Y., 2015. Full superstructure for multiobjective optimization of multicolumn chromatography for ternary separations. Chem. Eng. Technol. 38 (9), 1677–1682.

Agrawal, G., Sreedhar, B., Kawajiri, Y., 2014. Systematic optimization and experimental validation of ternary simulated moving bed chromatography systems. J. Chromatogr. A 1356, 82–95.

Barber, J., Perkins, J., Sargent, R., 1998. Boundary conditions for flow with dispersion. Chem. Eng. Sci. 53 (7), 1463–1464.

Bátkai, A., Csomós, P., Farkas, B., Nickel, G., 2011. Operator splitting for non-autonomous evolution equations. J. Funct. Anal. 260 (7), 2163–2190.

Broughton, D. B., Gerhold, C. G., 1961. Continuous sorption process employing fixed bed of sorbent and moving inlets and outlets. US Patent 2,985,589.

Engell, S., Toumi, A., 2004. Optimization and control of chromatography. Comput. Aided Chem. Eng. 18, 9–22.

Faria, R.P., Rodrigues, A.E., 2015. Instrumental aspects of simulated moving bed chromatography. J. Chromatogr. A 1421, 82–102.

Felinger, A., Guiochon, G., 2004. Comparison of the kinetic models of linear chromatography. Chromatographia 60, S175–S180.

Grosfils, V., Hanus, R., Wouwer, A.V., Kinnaert, M., 2010. Parametric uncertainties and influence of the dead volume representation in modelling simulated moving bed separation processes. J. Chromatogr. A 1217 (47), 7359–7371.

Han, W., Atkinson, K.E., 2009. Theoretical Numerical Analysis. Texts in Applied Mathematics, 39. Springer New York, New York, NY.

Holden, H., Karlsen, K., Lie, K.-A., Risebro, N.H., 2010. Splitting Methods for Partial Differential Equations with Rough Solutions. European Mathematical Society Publishing House, Zuerich, Switzerland.

Jo, S.-H., Kim, J.K., Yoo, C.G., Kim, J.-I., Koo, Y.-M., Mun, S., 2007. Comparative analysis of single-cascade five-zone and two-zone smb systems for the separation of a ternary amino acid mixture. Canad. J. Chem. Eng. 85 (6), 874–882.

Katsuo, S., Langel, C., Schanen, P., Mazzotti, M., 2009. Extra-column dead volume in simulated moving bed separations: theory and experiments. J. Chromatogr. A 1216 (7), 1084–1093.

Kazi, M.-K., Medi, B., Amanullah, M., 2012. Optimization of an improved single-column chromatographic process for the separation of enantiomers. J. Chromatogr. A 1231, 22–30.

Khan, H., Younas, M., 2011. Theoritical analysis and simulation of five-zone simulating moving bed for ternary mixture separation. Canad. J. Chem. Eng. 89 (6), 1480–1491.

Kim, J.K., Zang, Y., Wankat, P.C., 2003. Single-cascade simulated moving bed systems for the separation of ternary mixtures. Indust. Eng. Chem. Res. 42 (20), 4849–4860.

Klatt, K.-U., Hanisch, F., Dünnebier, G., 2002. Model-based control of a simulated moving bed chromatographic process for the separation of fructose and glucose. J. Process Control 12 (2), 203–219.

Li, S., Feng, L., Benner, P., Seidel-Morgenstern, A., 2014. Using surrogate models for efficient optimization of simulated moving bed chromatography. Comput. Chem. Eng. 67, 121–132. doi:10.1016/j.compchemeng.2014.03.024.

Li, S., Yue, Y., Feng, L., Benner, P., Seidel-Morgenstern, A., 2014. Model reduction for linear simulated moving bed chromatography systems using Krylov-subspace methods. AIChE J. 60 (11), 3773–3783. doi:10.1002/aic.14561.

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.

Lim, Y.-I., 2004. An optimization strategy for nonlinear simulated moving bed chromatography: multi-level optimization procedure (MLOP). Korean J. Chem. Eng. 21 (4), 836–852.

Lim, Y.-I., 2012. Optimal flushing flow rates in para-xylene simulated moving-bed considering geometric factor of dead volume. Adsorption 18 (5–6), 469–482.

Lim, Y.-I., Bhatia, S.K., 2011. Effect of dead volume on performance of simulated moving bed process. Adsorption 17 (1), 109–120.

Lübke, R., Seidel-Morgenstern, A., Tobiska, L., 2007. Numerical method for accelerated calculation of cyclic steady state of modicon–smb-processes. Comput. Chem. Eng. 31 (4), 258–267.

Mota, J.P., Araújo, J.M., 2005. Single-column simulated-moving-bed process with recycle lag. AIChE J. 51 (6), 1641–1653.

Mun, S., 2011. Improving performance of a five-zone simulated moving bed chromatography for ternary separation by simultaneous use of partial-feeding and partial-closing of the product port in charge of collecting the intermediate-affinity solute molecules. J. Chromatogr. A 1218 (44), 8060–8074.

Nicolaos, A., Muhr, L., Gotteland, P., Nicoud, R.-M., Bailly, M., 2001. Application of equilibrium theory to ternary moving bed configurations (four+ four, five+ four, eight and nine zones): i. linear case. J. Chromatogr. A 908 (1), 71–86.

Nowak, J., Antos, D., Seidel-Morgenstern, A., 2012. Theoretical study of using simulated moving bed chromatography to separate intermediately eluting target compounds. J. Chromatogr. A 1253, 58–70.

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.

Rajendran, A., Paredes, G., Mazzotti, M., 2009. Simulated moving bed chromatography for the separation of enantiomers. J. Chromatogr. A 1216 (4), 709–738.

Rodrigues, R.C., Araújo, J.M., Eusébio, M.F., Mota, J.P., 2007. Experimental assessment of simulated moving bed and varicol processes using a single-column setup. J. Chromatogr. A 1142 (1), 69–80.

Seidel-Morgenstern, A., Kessler, L.C., Kaspereit, M., 2008. New developments in simulated moving bed chromatography. Chem. Eng. Technol. 31 (6), 826–837.

da Silva, F.V.S., Seidel-Morgenstern, A., 2016. Evaluation of center-cut separations applying simulated moving bed chromatography with 8 zones. J. Chromatogr. A.

Silva, M.S., Rodrigues, A.E., Mota, J.P., 2016. Effect of dead volumes on the performance of an industrial-scale simulated moving-bed parex unit for p-xylene purification. AIChE J. 62 (1), 241–255.

Silva, R.J., Mota, J.P., Peixoto, C., Alves, P.M., Carrondo, M.J., 2015. Improving the downstream processing of vaccine and gene therapy vectors with continuous chromatography. Pharm. Bioprocess. 3 (8), 489–505.

Sreedhar, B., Hobbs, D.T., Kawajiri, Y., 2014. Simulated moving bed chromatography designs for lanthanide and actinide separations using reillex HPQresin. Separ. Purif. Technol. 136, 50–57.

Wang, X., Ching, C.B., 2002. Kinetic and equilibrium study of the separation of three chiral center drug, nadolol, by HPLC on a novel perphenyl carbamoylated β-cyclodextrin bonded chiral stationary phase. Separ. Sci. Technol. 37 (11), 2567–2586.

Wang, X., Ching, C.B., 2005. Chiral separation of β-blocker drug (nadolol) by five–zone simulated moving bed chromatography. Chem. Eng. Sci. 60 (5), 1337–1347.

Wooley, R., Ma, Z., Wang, N.-H., 1998. A nine-zone simulating moving bed for the recovery of glucose and xylose from biomass hydrolyzate. Indust. Eng. Chem. Res. 37 (9), 3699–3709.

Yu, Y., Wood, K.R., Liu, Y., 2015. Simulation and comparison of operational modes in simulated moving bed chromatography. Indust. Eng. Chem. Res. 54 (46), 11576–11591.

Zhang, Y., Feng, L., Seidel-Morgenstern, A., Benner, P., 2017. Accelerating optimization and uncertainty quantification of nonlinear SMB chromatography using reduced-order models. Comput. Chem. Eng. 96, 237–247. doi:10.1016/j.compchemeng.2016.09.017.