Model Reduction for Linear Simulated Moving Bed Chromatography Systems Using Krylov-Subspace Methods

Suzhou Li, Yao Yue, Lihong Feng, and Peter Benner
Computational Methods in Systems and Control Theory (CSC) group of MPI-Magdeburg, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, D-39106 Magdeburg, Germany

Andreas Seidel-Morgenstern
Physical and Chemical Foundations of Process Engineering (PCF) group of MPI-Magdeburg, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, D-39106 Magdeburg, Germany

Inst. of Process Engineering, Otto-von-Guericke University, Universitätsplatz 2, D-39106 Magdeburg, Germany

DOI 10.1002/aic.14561
Published online July 28, 2014 in Wiley Online Library (wileyonlinelibrary.com)

Simulated moving bed (SMB) chromatography is a well-established technology for separating chemical compounds. To describe an SMB process, a finite-dimensional multistage model arising from the discretization of partial differential equations is typically employed. However, its relatively high dimension poses severe computational challenges to various model-based analysis. To overcome this challenge, two Krylov-type model order reduction (MOR) methods are proposed to accelerate the computation of the cyclic steady states (CSSs) of SMB processes with linear isotherms. A "straightforward method" that carefully deals with the switching behavior in MOR is first proposed. Its improvement, a "subspace-exploiting method," thoroughly exploits each reduced model to achieve further acceleration. Simulation studies show that both methods achieve high accuracy and significant speedups. The subspace-exploiting method turns out to be computationally much more efficient. Two challenging analyses of SMB processes, namely uncertainty quantification and CSS optimization, further demonstrate the accuracy, efficiency, and applicability of the proposed methods.
 AIChE J, 60: 3773–3783, 2014
Keywords: simulated moving bed chromatography, model order reduction, Krylov-subspace method, optimization, uncertainty quantification

Introduction

Simulated moving bed (SMB) chromatography[1] is a continuous multicolumn separation process. After first applications in the petroleum and sugar industries, it has been recognized as a crucial technique for enantiomer separations in the pharmaceutical industry.[2,3] In the last few decades, various methods have been developed for the analysis, design, optimization, and control of SMB processes.[4,5] These methods lay firm theoretical foundations for the successful scale-up and industrial implementation of SMB systems. In the development of these methods, mathematical models of SMB play an extremely important role. Usually, a detailed process model is preferred as it captures more accurate process dynamics and guarantees high reliability of results. However, because of the distributed parameter property of the system, such a model is normally obtained from the discretization of the partial differential algebraic equations (PDAEs) and has a high dimension. The periodic steady state and nonlinear adsorption behavior also complicate its solution. Furthermore, the modeling complexity increases when more sophisticated SMB variants[4,5] are investigated, as they involve more modeling details and dynamics. Therefore, using a detailed model for the aforementioned tasks is computationally expensive, especially for optimization and control issues. A common approach is to use a simplified true moving bed (TMB) model to approximate the SMB system, as it is continuous and has an easily calculable steady state. However, the TMB approximation does not take the periodic port switching into account. Furthermore, as was pointed out by some authors,[6–8] its behavior deviates from the real SMB process seriously if a small number of columns are used, or for cases where operating conditions vary during each switching period. As a result, it is highly desirable to develop reliable and computationally efficient surrogate models that can reproduce the original SMB dynamics. To the best of our knowledge, the first attempt in this aspect is a novel single-column process developed by Mota and Araújo,[9] which uses only one chromatographic column to replicate the periodic state of SMB.[10] Another direction is to use model order reduction (MOR) techniques, which are identified as powerful tools for generating low-dimensional and computationally efficient approximations. Over the last few decades, various MOR algorithms have been developed and applied to many engineering areas. For a comprehensive overview, we refer to Antoulas,[11] Benner et al.,[12] and Baur

Correspondence concerning this article should be addressed to Y. Yue at yue@mpi-magdeburg.mpg.de.

et al.[13] However, very limited research efforts have been devoted to the development of MOR methods in the SMB literature. Erdem et al.[14] applied a balanced truncation method to derive a linearized time-varying reduced-order model (ROM), which renders the online optimization and control of an SMB unit computationally feasible. Vilas and Vande Wouwer[15] built a ROM using the proper orthogonal decomposition (POD) technique and embedded it into a multimodel predictive control framework. Recently, Boulkroune et al.[16] designed a nonlinear SMB state observer based on the ROM created by the same POD scheme. In our previous studies,[17,18] we also constructed POD-based ROMs aiming at accelerating SMB optimization. However, the reduction in computational costs achieved by all methods reviewed above is still not very significant.

To achieve a better speedup, we propose a new Krylov-type MOR framework applicable to SMB processes characterized by linear adsorption isotherms in this article. Within this MOR framework, the straightforward method and the subspace-exploiting method are developed for accelerating the calculation of the cyclic steady state (CSS). These two methods differ in the way of using ROMs derived.

The article is structured as follows. We will begin by presenting the full-order model (FOM), a multistage dynamical system obtained from the spatial discretization of a first-principles SMB model. Then, the standard Krylov-type MOR technique, which only applies to single-stage linear systems, will be briefly introduced. The main part of the article will present the two methods proposed and discuss their properties in detail. Afterward, the performance of these methods for predicting the CSS solution will be evaluated in terms of accuracy and efficiency and two applications will be examined to demonstrate the potential of using ROMs for accelerating the two tasks. The first case study focuses on the uncertainty quantification (UQ) of a glucose-fructose separation in a pilot-scale SMB unit. In a second case study, we will consider the optimization of CSS operating parameters for another example characterized by other geometric, thermodynamic, and kinetic parameters. We will end the article with conclusion and future directions.

SMB Chromatography and Krylov-Type Model Reduction

This section will introduce the essential background knowledge. We will first explain the basic principles of the SMB chromatographic process, and then concentrate on its mathematical modeling. Then, we present the standard Krylov-type MOR.

The SMB process and its mathematical modeling

The SMB process is developed as a practical implementation of the TMB concept. The schematic diagram of a classical four-zone SMB unit is shown in Figure 1. It consists of multiple identical chromatographic columns that are connected to each other to form a closed loop. The two inlets (feed and desorbent) and two outlets (extract and raffinate) divide the unit into four distinct zones playing different roles for the separation of a binary mixture of a and b. The feed and desorbent are supplied continuously, and meanwhile the two product streams are also continuously withdrawn from the raffinate and extract outlets. To simulate the true countercurrent movement in TMB, the four streams are periodically advanced ahead by one column in the direction of the liquid flow after a fixed switching period t_s as is shown in Figure 1. For more specific details regarding SMB chromatography and related topics, we refer the reader to the state-of-art reviews given by Rajendran et al.[4] and Seidel-Morgenstern et al.[5]

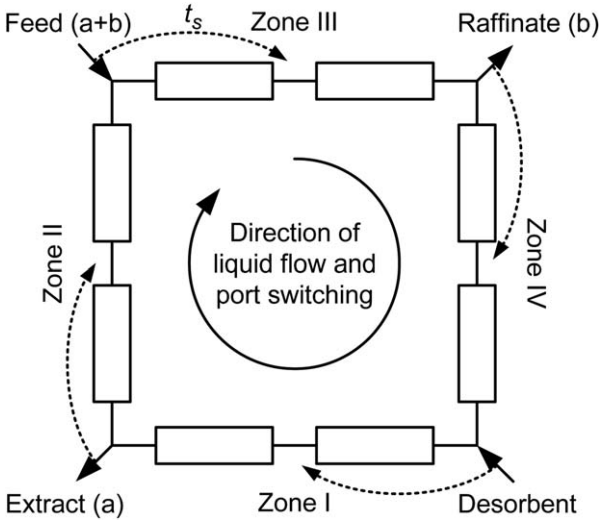

Figure 1. Schematic diagram of a conventional SMB unit with four zones and eight columns.

The remaining part of this section will present a classical first-principles SMB model derived from mass conservation equations.[19] This model consists of single-column models and balance equations around the inlet and outlet nodes. In this work, it is assumed that each chromatographic column is described by an axially dispersed plug-flow model with a limited mass-transfer rate characterized by a linear driving force (LDF) approximation. In this column model, the differential mass balance of component α (α=a, b) in the liquid phase of column η ($\eta=1,2,\ldots,n_{\mathrm{col}}$) can be written as

$$\frac{\partial c_\alpha^\eta}{\partial \tau}+\frac{1-\epsilon}{\epsilon}\frac{\partial q_\alpha^\eta}{\partial \tau}=\frac{t_s Q_\eta}{\epsilon A_c L}\left(\frac{1}{Pe}\frac{\partial^2 c_\alpha^\eta}{\partial z^2}-\frac{\partial c_\alpha^\eta}{\partial z}\right),\ z\in(0,\,1) \tag{1}$$

where c_α^η and q_α^η are the concentrations of solute α in the liquid and solid phases, respectively, Q_η is the volumetric flow rate, ϵ is the column porosity, t_s is the switching period, A_c is the cross-sectional area of the column, L is the column length, n_{col} is the number of columns, Pe is the Péclet number, which is defined as the ratio of the rate of mass transport by advection to that by diffusion,[20] $\tau=t/t_s$ and $z=Z/L$ are the dimensionless time and spatial coordinates, respectively.

The adsorption rate described by the LDF approximation reads as

$$\frac{\partial q_\alpha^\eta}{\partial \tau}=t_s k_\alpha(q_\alpha^{\eta,\mathrm{Eq}}-q_\alpha^\eta),\ z\in[0,\,1] \tag{2}$$

where k_α is the mass-transfer coefficient of component α and $q_\alpha^{\eta,\mathrm{Eq}}$ is the adsorption equilibrium concentration determined by the adsorption isotherm equation for component α

$$q_\alpha^{\eta,\mathrm{Eq}}=f_\alpha(c_a^\eta,c_b^\eta). \tag{3}$$

In this work, we restrict our discussion to the linear isotherm

$$q_\alpha^{\eta,\mathrm{Eq}}=H_\alpha c_\alpha^\eta,\ \alpha=\mathrm{a,b} \tag{4}$$

where H_α is the Henry constant. It is assumed that $H_a > H_b$.

The boundary conditions of Eq. 1 are

$$\frac{\partial c_\alpha^\eta}{\partial z}|_{z=0}=Pe(c_\alpha^\eta|_{z=0}-c_\alpha^{\eta,\text{in}}),\quad \frac{\partial c_\alpha^\eta}{\partial z}|_{z=1}=0 \tag{5}$$

with $c_\alpha^{\eta,\text{in}}$ being the concentration of component α at the inlet of column η.

By considering the mass balances at the inlet and outlet ports, the set of node equations can be readily established:

Desorbent node

$$Q_{\text{IV}}+Q_{\text{D}}=Q_{\text{I}} \tag{6}$$

$$c_\alpha^{1,\text{in}}(\tau)Q_{\text{I}}=c_\alpha^{n_{\text{col}}}(\tau,1)Q_{\text{IV}} \tag{7}$$

Extract node

$$Q_{\text{I}}-Q_{\text{E}}=Q_{\text{II}} \tag{8}$$

$$c_\alpha^{n_{\text{I}}+1,\text{in}}(\tau)=c_\alpha^{n_{\text{I}}}(\tau,1) \tag{9}$$

Feed node

$$Q_{\text{II}}+Q_{\text{F}}=Q_{\text{III}} \tag{10}$$

$$c_\alpha^{n_{\text{I}}+n_{\text{II}}+1,\text{in}}(\tau)Q_{\text{III}}=c_\alpha^{n_{\text{I}}+n_{\text{II}}}(\tau,1)Q_{\text{II}}+c_\alpha^{\text{F}}Q_{\text{F}} \tag{11}$$

Raffinate node

$$Q_{\text{III}}-Q_{\text{R}}=Q_{\text{IV}} \tag{12}$$

$$c_\alpha^{n_{\text{I}}+n_{\text{II}}+n_{\text{III}}+1,\text{in}}(\tau)=c_\alpha^{n_{\text{I}}+n_{\text{II}}+n_{\text{III}}}(\tau,1) \tag{13}$$

where $Q_{\text{I}},\ldots,Q_{\text{IV}}$ are the flow rates in the four zones denoted by $\omega\in\{\text{I},\text{II},\text{III},\text{IV}\}$, Q_{D}, Q_{E}, Q_{F}, Q_{R} are the desorbent, extract, feed, and raffinate flow rates, respectively, n_ω is the number of columns in zone ω, and c_α^{F} is the feed concentration of solute α.

Because the system of PDAEs above is highly coupled, its analytical solution cannot be obtained. Thus, efficient numerical discretization techniques must be used. In this article, we use the orthogonal collocation on finite elements (OCFE) method[21] on a nondimensionalized PDAE because the OCFE method not only deals well with steep profiles in SMB systems,[21] but is also of high accuracy, for example, it is able to capture second-order information and can therefore approximate dispersion with relatively high accuracy, so numerical dispersion is relatively small.[22] The resulting multistage system of differential algebraic equations (DAEs) is

$$\begin{cases} M\dot{x}_i(\tau)=A(p)x_i(\tau)+B(p), & i=1,2,\ldots,\ \tau\in[0,\ 1] \quad (14)\\ x_i(0)=P_{\text{s}}x_{i-1}(1), & i=2,3,\ldots \quad (15)\\ x_1(0)=x^0, & \quad (16)\end{cases}$$

where $x_i(\tau)\in\mathbb{R}^n$ is the state vector representing the concentrations of the liquid and solid phases at the ith period (or stage), $M\in\mathbb{R}^{n\times n}$, $A\in\mathbb{R}^{n\times n}$, $B\in\mathbb{R}^n$ are the coefficient matrices, $p\in\mathbb{R}^{n_p}$ is the vector of system parameters of interest, $P_{\text{s}}:\mathbb{R}^n\mapsto\mathbb{R}^n$ is a permutation operator, x^0 is the initial state for the first period, i is the period index, and the entries in all matrices and vectors above are dimensionless scalars due to the nondimensionalization. Note that in the formulation above, each stage has an identical duration equal to one switching period. The port switching triggers the stage transition with the transition condition stated in Eq. 15. At the transition, only the state variables change and the coefficient matrices remain stage-independent.

Because of the cyclic port shifting along the circularly arranged columns, SMB does not reach a stationary state but a CSS after undergoing the initial start-up phase. Mathematically, if $x_{\text{css}}(\tau)$ is a CSS solution, it satisfies

$$x_{\text{css}}(0)=P_{\text{s}}x_{\text{css}}(1). \tag{17}$$

Depending on how to calculate the CSS solution, two major classes of methods can be distinguished in the literature:[7,23]

- The conventional method integrates Eqs. 14–16 for a number of periods until the CSS is reached. This method is also referred to as the dynamic integration or simulation scheme.
- The full-discretization approach further discretizes Eq. 14 with respect to the temporal coordinate, thus transforming the original PDAEs into a large-scale sparse system of algebraic equations. Together with the CSS condition Eq. 17, the resulting algebraic system is solved by an appropriate solver specially designed for sparse problems to evaluate the CSS.

In this work, we focus on the conventional method. To evaluate the CSS solution for a given vector of parameters p, the conventional method integrates from a given initial condition x^0 for a number of periods until convergence to the CSS has been achieved. To judge whether x_i is a good approximation of the true CSS defined in Eq. 17, we can check the following stopping criterion

$$\|x_i(0)-P_{\text{s}}x_i(1)\|_2\le\epsilon_{\text{css}},(i\ge 2) \tag{18}$$

where ϵ_{css} is a specified CSS tolerance. Due to Eq. 15, the stopping criterion Eq. 18 is equivalent to the following form that will be used in our numerical case studies

$$\|x_i(0)-x_{i-1}(0)\|_2\le\epsilon_{\text{css}},(i\ge 2). \tag{19}$$

For convenience, this FOM-based method for CSS calculation is referred to as FOM-CSS throughout the work.

To guarantee the approximation to the original PDAE system to be accurate, a fine spatial discretization is often needed in the OCFE scheme, rendering Eq. 14 to be of high dimension. For the cases with steeper concentration profiles, finer grids must be imposed to alleviate physically unrealistic oscillations, which causes the system dimension to further increase. For clarity, we refer to this large-scale system as the FOM throughout the work. Its high dimension poses a significant challenge to the CSS evaluation. For the integration method, the integration of the FOM must be performed repeatedly, which is an expensive task. Although the more efficient full-discretization method avoids such a step, the computational effort is highly sensitive to the spatial dimension. Furthermore, to achieve an accuracy comparable to that of the integration scheme, a high-resolution grid for the temporal domain is usually required. This enlarges the dimension of the resulting algebraic system, thereby placing a high requirement on the performance of sparse solvers. MOR techniques are efficient tools for dealing with the challenge arising from the large-scale state equations. In the next section, we will detail the Krylov-type MOR for the FOM system. ROMs derived will replace the FOM in the integration method to accelerate the CSS calculation. The combination of the full-discretization method with MOR will be investigated in our future work.

An introduction to Krylov-type MOR

This section begins with the definition of Krylov subspaces and then discusses Krylov-type MOR for linear dynamical systems in both frequency and time domains.

Krylov Subspace. Given a matrix $A\in\mathbb{R}^{n\times n}$ and a vector $B\in\mathbb{R}^n$, the order-k Krylov subspace is defined by

$$\mathcal{K}_k(A,B)=\text{span}\{B,AB,A^2B,\ldots,A^{k-1}B\} \tag{20}$$

where $k \le n$ and for applications, mostly $k \ll n$. Krylov subspace methods play an indispensable role in solving linear equations and algebraic eigenvalue problems, building ROMs, and so forth. They are very suitable for large-scale problems because only matrix-vector multiplication is required. However, explicitly computing the Krylov vectors $A^jB\,(0 \le j \le k-1)$ to build the Krylov subspace is numerically unstable because these vectors tend very quickly to become almost linearly dependent. The most commonly used methods to generate an orthonormal basis of $\mathcal{K}_k(A,B)$ include the Arnoldi process[24] and the Lanczos method.[25] Since for the SMB system described in Eq. 14, we need to deal with nonsymmetric matrices, we choose the Arnoldi process for better numerical stability.

Algorithm 1 (The Arnoldi process to compute a basis matrix V of $\mathcal{K}_k(A,B)$).
1. Set the initial vector $v_1=\frac{B}{||B||_2}$ *and the iteration index* $j=1$.
2. While $j \le k-1$, *do:*
2.1. Compute $\tilde{w}_j=Av_j$.
2.2. Orthogonalize $\tilde{w}_j$ *to span*$\{v_1,v_2,\ldots,v_j\}$ *to get* w_j.
2.3. Normalization: if $||w_j||_2=0$, *stop; otherwise, compute* $v_{j+1}=\frac{w_j}{||w_j||_2}$ *and assign* $j=j+1$.
End do.
3. Assign $V=[v_1,v_2,\ldots,v_j]$.

In Step 2.2, orthogonalization could be implemented with various methods including the Gram–Schmidt method, the modified Gram–Schmidt method, the Gram–Schmidt method with reorthogonalization, and so forth. We will use the Gram–Schmidt method with reorthogonalization in our numerical tests for higher numerical stability.[26]

Krylov-Type MOR for Linear Systems in the Frequency Domain. Originally, Krylov-type MOR techniques were developed for dynamical systems in the frequency domain. To reduce the first-order linear system

$$(sM-A)X=B \tag{21}$$

where $M,A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^n$, and $s \in \mathbb{C}$, Krylov methods build the following Krylov subspace

$$\mathcal{K}_k(A^{-1}M,A^{-1}B). \tag{22}$$

A basis matrix V of the subspace above can be built via the Arnoldi process detailed in Algorithm 1. Approximating X in the range of V by

$$X \approx V\hat{X},\ \hat{X} \in \mathbb{R}^k \tag{23}$$

and forcing the residual $(sM-A)V\hat{X}-B$ orthogonal to colspan$\{V\}$ leads to the following ROM

$$(s\hat{M}-\hat{A})\hat{X}=\hat{B} \tag{24}$$

where $\hat{M}=V^TMV$, $\hat{A}=V^TAV$, and $\hat{B}=V^TB$. It is well known that the Krylov-type ROM in Eq. 24 has the moment matching property, which means that the first Taylor coefficients of $V\hat{X}$ match those of X.[27,28]

In the discussion above, we actually choose the frequency interpolation point s_0 to be $s_0=0$. To achieve moment matching at an arbitrary frequency interpolation point s_0, we can simply rewrite Eq. 21 into

$$((s-s_0)M-(A-s_0M))X=B \tag{25}$$

and use the method described above to reduce it using the substitutions $M \leftarrow M$, $A \leftarrow A-s_0M$, and $B \leftarrow B$. However, our numerical results show that for the SMB processes considered in this article, the choice $s_0=0$ works quite well. Therefore, for simplicity, we assume $s_0=0$ in the rest of the article.

Krylov-Type MOR for Linear Systems in the Time Domain: The Shift-Reduce-Reproduce Scheme. Krylov-type MOR can also be used to reduce time-domain systems. As the Laplace transform of the following time-domain system with zero initial condition

$$\begin{cases} M\dot{x}(\tau)=Ax(\tau)+B, \\ x(0)=0, \end{cases} \tag{26}$$

is just Eq. 21, where (x, X) is a Laplace transform pair, the Krylov subspace in Eq. 22 can also be used to reduce Eq. 26.[29] To reduce the general time-domain linear system

$$\begin{cases} M\dot{x}(\tau)=Ax(\tau)+B, \\ x(0)=x^0, \qquad x^0 \in \mathbb{R}^n \text{ is an arbitrary vector} \end{cases} \tag{27}$$

a common strategy[29] is to conduct the following shift

$$\xi(\tau)=x(\tau)-x^0 \tag{28}$$

to obtain an equivalent system with zero initial state, which we will call Σ

$$\Sigma: \quad \begin{cases} M\dot{\xi}(\tau)=A\xi(\tau)+\tilde{B}, \\ \xi(0)=0, \end{cases} \tag{29}$$

where

$$\tilde{B} \triangleq Ax^0+B. \tag{30}$$

Then, we can reduce Eq. 29 with the Krylov subspace

$$\mathcal{K}_k(A^{-1}M,A^{-1}\tilde{B}). \tag{31}$$

We denote the resulting ROM by

$$\hat{\Sigma}: \quad \begin{cases} \hat{M}\dot{\hat{\xi}}(\tau)=(\hat{A}\hat{\xi}\tau)+\hat{\tilde{B}}, \\ \hat{\xi}(0)=0, \end{cases} \tag{32}$$

where $\hat{M}$, $\hat{A}$ are defined as in Eq. 24 and $\hat{\tilde{B}}=V^T\tilde{B}$.
Using the resulting ROM, we can approximate the evolution of $\xi(\tau)$ by $V\hat{\xi}(\tau)$, from which we can compute an approximation of $x(\tau)$, namely $\tilde{x}(\tau)$ defined by

$$\tilde{x}(\tau)=V\hat{\xi}(\tau)+x^0. \tag{33}$$

We call the procedure above that uses MOR to obtain a state vector approximation for Eq. 27 the shift-reduce-reproduce scheme for the rest of the article.

Krylov-Type MOR for SMB Models

Following the shift-reduce-reproduce scheme described earlier, we present two methods for computing the CSS of

the SMB process in this section: the "straightforward method" (SROM-CSS) and the "subspace-exploiting method" (SEROM-CSS). Although in using the simulation scheme to compute the CSS of the SMB process for a given p, M, $A(p)$, and $B(p)$ in Eq. 14 do not change throughout, both methods have to use a series of ROMs to compute the CSS because a ROM also depends on the (approximated) initial state vector that normally varies among different stages. Before introducing the two methods, we define the notation. Assume that we reduce an order-n FOM to an order-k ROM. For a variable in the full-order space, say $x \in \mathbb{R}^n$ (or $M \in \mathbb{R}^{n\times n}$), we use $\hat{x} \in \mathbb{R}^k$ (or $\hat{M} \in \mathbb{R}^{k\times k}$) to denote the corresponding variable in the ROM, while we use $\tilde{x} \in \mathbb{R}^n$ to denote the approximation of x computed via the ROM. For algorithms containing only one loop, we use $\cdot_i$ to denote the variable $\cdot$ for the ith (outer) loop; while for algorithms containing two layers of loops, $\cdot_{i,j}$ denotes the variable $\cdot$ for the jth inner loop within the ith outer loop. For example, $\tilde{x}_{i,j}$ represents an approximation of $x_{i,j}$, which is the state vector for the jth inner loop within the ith outer loop.

The "straightforward method" for computing the CSS

Recall that given a parameter vector p in Eq. 14, we normally reach a CSS corresponding to p after simulating the FOM for a sufficiently large number of periods, which is computationally expensive. The goal of SROM-CSS is twofold:

1. using the shift-reduce-reproduce scheme to accelerate the simulation above;

2. following the evolution of the FOM to compute the CSS, which means that we need to capture both the evolution trajectory within each period and the state transition between any two successive periods accurately.

As the shift-reduce-reproduce scheme can only reduce single-stage systems, we use it for the multistage SMB model in an "on the fly" manner. More specifically, we first use the shift-reduce-reproduce scheme for the first period with the initial condition x^0; for the ith period ($i \geq 2$), we use the end state approximation of the $(i-1)$-st period $\tilde{x}_{i-1}(1)$ to compute a good initial state approximation $\tilde{x}_i(0)=P_s\tilde{x}_{i-1}(1)$ and then apply the shift-reduce-reproduce scheme.

Let x_i be the state vector of the FOM for the ith period, $\tilde{x}_i$ be the computed approximation of x_i, ξ_i be a shifted state vector corresponding to $\tilde{x}_i$ having zero initial state

$$\xi_i(\tau)=\tilde{x}_i(\tau)-\tilde{x}_i(0). \tag{34}$$

Further, let V_i be the matrix built by the Arnoldi process within whose range we approximate ξ_i and $\hat{\xi}_i$ be the state vector of the ROM. Then, we have

$$\xi_i(\tau)=V_i\hat{\xi}_i(\tau) \tag{35}$$

and

$$\tilde{x}_i(\tau)=V_i\hat{\xi}_i(\tau)+\tilde{x}_i(0). \tag{36}$$

The initial state approximation $\tilde{x}_i(0)$ in Eq. 34 is then computed by

$$\tilde{x}_i(0)=\begin{cases} x_1(0), & \text{for } i=1, \\ P_s\tilde{x}_{i-1}(1), & \text{for } i \geq 2. \end{cases} \tag{37}$$

An algorithm for SROM-CSS is outlined as follows.

Algorithm 2 (SROM-CSS).

1. Set the initial state $x_1(0)$, the initial state approximation $\tilde{x}_1(0)=x_1(0)$, the error tolerance ϵ_{css}, and the iteration index $i=1$.
2. Do:
  2.1. ***Shift****. Conduct the shift in Eq. 34 to get a shifted system Σ_i in the form of Eq. 29.*
  2.2. ***Reduce & simulate****. Build the ROM $\hat{\Sigma}_i$ for Σ_i in the form of Eq. 32 using the method presented in the previous section, and then simulate $\hat{\xi}_i$ from $\tau=0$ to $\tau=1$.*
  2.3. ***Update****. Set $i=i+1$ and compute $\tilde{x}_i(0)$ using Eq. 37.*
until $||\tilde{x}_i(0)-\tilde{x}_{i-1}(0)||_2 \leq \epsilon_{\text{css}}$.

SROM-CSS leads to a significant speedup due to the analysis below. The main computational cost contains three parts:

1. LU factorization: Throughout the whole algorithm, only one large-scale LU factorization is required: the LU factorization of A.

2. Linear system solves: For each stage, k linear solves with A are conducted to build the ROM. Although a single linear system solve is cheaper than an LU factorization, linear system solves serve as the computational dominant part of the whole algorithm due to the large number of solves.

3. ROM simulation: Due to the low order of the ROMs, it is very cheap to simulate ROMs.

Compared to FOM-CSS, in which simulating a single stage requires solving the large-scale FOM at each integration step, the computational cost is drastically reduced.

The "Subspace-Exploiting Method" for Computing the CSS (SEROM-CSS). Although SROM-CSS is already much more efficient than FOM-CSS in computing the CSS, it does not fully exploit the ROMs since in order to follow the evolution strategy of FOM-CSS, a ROM can be used to simulate only one period. Actually, a ROM contains much richer information because the corresponding subspace allows for many other trajectories, and there may exist trajectories that are closer to the true CSS than the trajectory computed by SROM-CSS. If we can locate such a trajectory cheaply for each subspace, we expect further speedup in computing the CSS because for convergence we should require less subspaces, or less ROMs, the constructions of which are the computationally dominant part. Therefore, the goal of the subspace-exploiting method is to find "the best" trajectory within the subspace, which is the trajectory closest to the CSS in some sense and at the same time cheap to compute.

In a subspace, a true CSS is normally not well-defined. Suppose that the FOM has only one CSS. Then, a subspace has a CSS only when the CSS of the FOM lies within this subspace. Now, we analyze this in more detail. Following the notation used for the straightforward method, assume that the ROM $\hat{\Sigma}_i$ has a state trajectory $\hat{x}_{\text{css},i}(\tau)$, whose corresponding state vector approximation $\tilde{x}_{\text{css},i}(\tau)$ satisfies the true CSS condition Eq. 17, which means

$$\tilde{x}_{\text{css},i}(0)=P_s\tilde{x}_{\text{css},i}(1). \tag{38}$$

Note that the shift $\tilde{x}_i(0)$, which is computed from the end state of the $(i-1)$-st iteration, is normally different from the initial CSS state $\tilde{x}_{\text{css},i}(0)$, which is not known before we build the ROM. Therefore, the relationships corresponding to Eqs. 34 and 35 for $\tilde{x}_{\text{css},i}$ are

$$\xi_{\text{css},i}(\tau)=\tilde{x}_{\text{css},i}(\tau)-\tilde{x}_i(0) \tag{39}$$

and

$$\xi_{\mathrm{css},i}(\tau)=V_i\hat{\xi}_{\mathrm{css},i}. \tag{40}$$

Therefore, $\tilde{x}_{\mathrm{css},i}$ is computed via the ROM by

$$\tilde{x}_{\mathrm{css},i}(\tau)=\tilde{x}_i(0)+V_i\hat{\xi}_{\mathrm{css},i}(\tau). \tag{41}$$

Using this relationship, we rewrite the CSS condition Eq. 38 into

$$V_i\hat{\xi}_{\mathrm{css},i}(0)=(P_s-I)\tilde{x}_i(0)+P_sV_i\hat{\xi}_{\mathrm{css},i}(1), \tag{42}$$

which implies that

$$(P_s-I)\tilde{x}_i(0)+P_sV_i\hat{\xi}_{\mathrm{css},i}(1)\in \mathrm{colspan}\{V_i\}, \tag{43}$$

a relationship that does not hold in general. This means that the ROM normally does not have a CSS approximation satisfying the true CSS condition Eq. 38. An intuitive explanation of this phenomenon is that, we impose too many CSS constraints on the ROM. Remind that for the FOM of order n, the system usually has a unique CSS solution if we impose n CSS constraints. Therefore, for a ROM of order k, we should impose k constraints to compute a unique solution rather than n constraints as what we did in Eq. 38. To deal with this problem, we define a so-called pseudo-CSS $\hat{\xi}_{\mathrm{pcss},i}$ that satisfies

$$\hat{\xi}_{\mathrm{pcss},i}(0)=V_i^T(P_s-I)\tilde{x}_i(0)+V_i^TP_sV_i\hat{\xi}_{\mathrm{pcss},i}(1), \tag{44}$$

which means that Eq. 43 is satisfied after applying the orthonormal projector $V_iV_i^T$ on both side, or equivalently, the residual between the two sides in Eq. 43 is orthogonal to $\mathrm{colspan}\{V_i\}$. Imposing the k constraints in Eq. 44, we expect that the k-dimensional subspace usually has a pseudo-CSS.

Like in FOM-CSS that integrates the FOM repeatedly to compute the CSS, we can introduce an inner loop to compute the pseudo-CSS $\hat{\xi}_{\mathrm{pcss},i}$ in an iterative manner using the following transition condition

$$\hat{\xi}_{i,j+1}(0)=V_i^TP_sV_i\hat{\xi}_{i,j}(1)+V_i^T(P_s-I)\tilde{x}_{i,1}(0), \tag{45}$$

where $j\in\mathbb{N}^+$ is the index for the inner loop and $\tilde{x}_{i,1}(0)$ serves as both the initial point for the inner loop and the shift in Eq. 34. This iterative scheme must converge to $\hat{\xi}_{\mathrm{pcss},i}$ as long as it converges. In Eq. 45, both $V_i^TP_sV_i$ and $V_i^T(P_s-I)\tilde{x}_{i,1}(0)$ do not change with j. Therefore, they need to be computed only once at the beginning of the ith outer iteration.

We summarize SEROM-CSS in Algorithm 3.

Algorithm 3 (SEROM-CSS).
1. Set the initial state $x_{1,1}(0)$, the initial state approximation $\tilde{x}_{1,1}(0)=x_{1,1}(0)$, the error tolerance ϵ_{css}, and the iteration index $i=1$.
2. Do:
2.1. ***Shift****. Conduct the shift in Eq. 34 to get a shifted system Σ_i in the form of Eq. 29.*
2.2. ***Reduce & simulate****. Build the ROM $\hat{\Sigma}_i$ for Σ_i in the form of Eq. 32 using the shift-reduce-reproduce scheme. Set $j=1$.*
Do:
2.2.1. Simulate $\hat{\Sigma}_i$ from $\tau=0$ to $\tau=1$ using the initial state $\hat{\xi}_{i,j}(0)$.
2.2.2. Use Eq. 45 to compute $\hat{\xi}_{i,j+1}(0)$. Set $j=j+1$.
until $||\hat{\xi}_{i,j}(0)-\hat{\xi}_{i,j-1}(0)||_2\le\epsilon_{\mathrm{css}}$.
2.3. ***Update****. Compute $\tilde{x}_{i+1,1}(0)=P_s(V_i\hat{\xi}_{i,j-1}(1)+\tilde{x}_{i,1}(0))$ and set $i=i+1$.*
until $||\tilde{x}_{i,1}(0)-\tilde{x}_{i-1,1}(0)||_2\le\epsilon_{\mathrm{css}}$.

In Algorithm 3, we use the stopping criterion $||\hat{\xi}_{i,j}(0)-\hat{\xi}_{i,j-1}(0)||_2\le\epsilon_{\mathrm{css}}$ for the inner loop of the subspace-exploiting method, which is cheaper to check than the original condition $||\tilde{x}_{i,j}(0)-\tilde{x}_{i,j-1}(0)||_2\le\epsilon_{\mathrm{css}}$. Actually, these two conditions are equivalent as we have

$$||\hat{\xi}_{i,j}(0)-\hat{\xi}_{i,j-1}(0)||_2=||\xi_{i,j}(0)-\xi_{i,j-1}(0)||_2=||\tilde{x}_{i,j}(0)-\tilde{x}_{i,j-1}(0)||_2 \tag{46}$$

thanks to the same shift and the same V_i used within the ith outer loop.

The speedup of SEROM-CSS is expected to be higher than that of SROM-CSS. Compared to SROM-CSS, SEROM-CSS reduces the number of (outer) stages in achieving convergence, which equals the total number of ROMs required. Therefore, SEROM-CSS builds much fewer ROMs than SROM-CSS, and therefore, drastically reduces this computationally dominant part, at a very low expense of conducting a larger number of computationally much cheaper ROM simulations for each single stage. The computational cost of SEROM-CSS also contains three main parts like that of SROM-CSS:

1. LU factorization: Like SROM-CSS, SEROM-CSS also requires only one large-scale LU factorization, namely the LU factorization of A, throughout the whole algorithm.

2. Linear system solves: Like in SROM-CSS, k linear solves with A are conducted to build the ROM for each stage and they serve as the computationally dominant part. However, due to the more thorough exploitation of each ROM, fewer stages should be required for convergence to the CSS, which means that fewer linear system solves should be conducted.

3. ROM simulation: Like in SROM-CSS, simulating ROMs is very cheap. For a single stage, more computational effort is spent on ROM simulation. However, the total computational cost of ROM simulation in SEROM-CSS is not necessarily higher than that in SROM-CSS as the total number of stages is expected to decrease.

Implementation details

This section discusses some implementation details of the two methods proposed earlier.

Determining the Order of the ROMs. For the proposed methods to work, it is very important to choose appropriate orders for the ROMs. If they are too small, the ROMs are not accurate enough, or even lead to completely wrong results. If they are too large, the ROMs become more expensive to build and evaluate, which means that the total computational effort increases. An ideal approach would be to use a good error bound to help choose the order of ROMs. However, finding a tight and computationally cheap error bound for Krylov-type ROMs is still an open question.[30,31] A common strategy used in engineering is to check the residual.[32–34] If a dynamical system is not too ill-conditioned, a small residual means a small error. For the system in Eq. 26, we define the relative residual of the ROM by

$$r(\tilde{x}(V,\tau))=\frac{||M\dot{\tilde{x}}(V,\tau)-A\tilde{x}(V,\tau)-B||_2}{||B||_2} \tag{47}$$

where we replace $\tilde{x}(\tau)$ by $\tilde{x}(V,\tau)$ to emphasize the fact that $\tilde{x}$ also depends on V. If we enlarge the Krylov subspace,

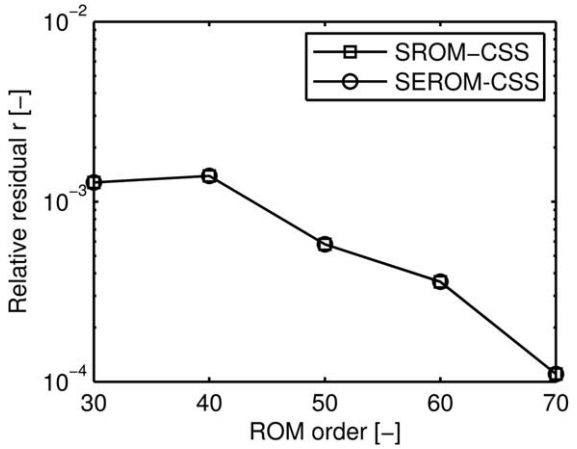

Figure 2. Development of relative residual of the CSS solution with respect to the ROM order for SROM-CSS and SEROM-CSS. The CSS solution at $\tau=1$ is used for residual calculation.

which means that V contains more columns, r normally decreases as is shown in Figure 2, the physical setting of which is shown in Table 1. This allows us to build the Krylov subspace in a dynamical manner: if the residual is above a predefined tolerance ϵ_r, we can increase the order of the ROM until this residual is below ϵ_r. However, relying too much on this dynamical scheme brings in an extra computational cost and implementation complexity. In addition, to check $r(\tilde{x}(V,\tau))$ for all τ in each stage further increases the computational cost. Therefore, we advise: (1) checking $r(\tilde{x}(V,\tau))$ only for important τ's, for example, checking them at $\tau=0$ and/or $\tau=1$ for each stage or even just for the CSS; (2) using a relatively high order at first to avoid increasing the order of ROMs dynamically.

The Stopping Criterion for the Inner Loop of the Subspace-Exploiting Method. In Algorithm 3, the subspace-exploiting method uses the same stopping criterion $||\hat{\tilde{\xi}}_{i,j}(0)-\hat{\tilde{\xi}}_{i,j-1}(0)||_2 \leq \epsilon_{\text{css}}$ for all inner loops. In a more general formulation, we can use different tolerances for different inner loops, or more specifically, use the pseudo-CSS tolerance $\epsilon_{\text{css},i}$ for the inner loop in the ith outer iteration. We can even use other types of stopping criteria such as posing a small relative error or fixing the number of iterations. Here, we focus on the stopping criterion $||\hat{\tilde{\xi}}_{i,j}(0)-\hat{\tilde{\xi}}_{i,j-1}(0)||_2 \leq \epsilon_{\text{css},i}$ because thanks to the fact that it is of the same form as the stopping criterion of the outer loop $||\tilde{x}_{i,1}(0)-\tilde{x}_{i-1,1}(0)||_2 \leq \epsilon_{\text{css}}$, we have a clue in determining how strict a stopping criterion we should impose. If $\epsilon_{\text{css},i}$ is too large, the subspace is not exploited well and the speedup of the subspace-exploiting method is similar to that of the straightforward method. Conversely, $\epsilon_{\text{css},i}$ should not be too small. If $\epsilon_{\text{css},i} \ll \epsilon_{\text{css}}$, too much computational effort is spent on the inner loop as the outer loop does not require such accuracy. Therefore, a reasonable $\epsilon_{\text{css},i}$ should have the same order of magnitude as ϵ_{css}. For all our numerical tests, we will set $\epsilon_{\text{css},i}=\epsilon_{\text{css}}$.

Table 1. Model Parameters and Operating Conditions for the First Example, which was Described by Azevedo and Rodrigues[35]

Number of columns n_{col}	12	Henry constant H_{a}	0.53
Column configuration	3-3-3-3	Henry constant H_{b}	0.27
Column dimensions (cm)	2.6 × 30	Q_{I} (mL/s)	0.567
Column porosity ϵ	0.4	Q_{II} (mL/s)	0.465
Péclet number Pe	500	Q_{III} (mL/s)	0.521
Mass-transfer coefficient K_{a} (s^{-1})	0.0218	Q_{IV} (mL/s)	0.400
Mass-transfer coefficient K_{b} (s^{-1})	0.0310	t_{s} (s)	198
Feed concentrations (mol/L)	0.2222		

Subscripts a and b represent fructose and glucose, respectively.

Table 2. Comparison of the Three Methods for CSS Calculation of the First Example

Method	Model Order	CPU Time (s)	Ni[a]	Pu_{a} (%)	Pu_{b} (%)
FOM-CSS	1008	612	292	95.276	86.068
SROM-CSS	50	192	277	95.271	86.066
SEROM-CSS	50	13	8	95.271	86.065

[a]Ni denotes the number of periods required to reach the CSS.

Results and Discussion

In this part, we will first evaluate the performance of both SROM-CSS and SEROM-CSS for CSS calculation. Then, we will examine two SMB case studies in two different analyses: UQ and CSS optimization, aiming at demonstrating the accuracy, efficiency, and practical applicability of the proposed methods. In ROM-based analysis, SROM-CSS or SEROM-CSS replaces FOM-CSS for CSS computation.

Evaluation of straightforward and subspace-exploiting methods

A glucose-fructose separation performed on a pilot-scale SMB unit reported by Azevedo and Rodrigues[35] was used as our test bench. This sugar separation is a representative example that exhibits linear adsorption behavior. The model parameters and operating conditions are summarized in Table 1. In this example, we use FOM-CSS, SROM-CSS, and SEROM-CSS to calculate the CSS solution. Both ROM-based methods will be compared to FOM-CSS in terms of the model order, CPU time, number of periods required to reach the CSS, and product purity. The DASPK3.1 package[36] was used as the integrator with an absolute error tolerance of 1.0×10^{-6} and a relative error tolerance of 1.0×10^{-5}. The concentration profiles normalized with respect to the feed concentrations were used to check CSS numerically with the tolerance $\epsilon_{\text{css}}=1.0\times10^{-5}$. For the SEROM-CSS method, the pseudo-CSS tolerance was specified the same as ϵ_{css}. All computations were performed on a Linux machine with an Intel 3.0 GHz Pentium D processor and 2 GB RAM, and the code was implemented in Fortran 90. The FOM was constructed by combining five finite elements and three internal collocation points for the spatial discretization of the PDAE model. We found that the choice of this spatial mesh is sufficient for the case study. For SROM-CSS and SEROM-CSS, the residual-based criterion described above was used to help choose the orders. For this purpose, we calculated the relative residual at the CSS solution ($\tau=1$) predicted by each method using Eq. 47 and show their development with respect to the order in Figure 2. As is expected, the residual rapidly decreases as the model order increases, indicating that the solution becomes more accurate. Finally, we fixed the order at 50 for both methods in

Table 3. Comparison of Results of Uncertainty Analysis for the Glucose-Fructose Separation

Method	Model Order	CPU Time (min)	$E[\mathrm{Pu_a}]$ (%)	$E[\mathrm{Pu_b}]$ (%)	$\sigma[\mathrm{Pu_a}]$ (%)	$\sigma[\mathrm{Pu_b}]$ (%)
FOM-CSS	1008	363.8	94.484	85.644	3.251	1.644
SEROM-CSS	50	11.4	94.480	85.641	3.250	1.643

the following simulation studies, which guarantees the residual below the specified tolerance $\epsilon_r = 1.0 \times 10^{-3}$.

The CSS simulation results are presented in Table 2, where subscripts a and b represent fructose and glucose, respectively. The FOM consists of 1008 DAEs for each stage and FOM-CSS converges after 292 periods and its computational time is 612 s. The purities of glucose and fructose at the CSS are 95.276% and 86.068%, respectively, which are very close to those reported by Azevedo and Rodrigues.[35] By contrast, both ROM-based methods build ROMs of order 50, which is only about 1/20 of that of the FOM, and result in significant acceleration of CSS evaluation. In particular, SEROM-CSS takes only 8 periods to converge and the calculation is sped up by a factor of 47. Furthermore, the product purities predicted by the ROM-based schemes are accurate up to four digits, which is in accordance with the prescribed error tolerance and perfectly acceptable for various purposes.

To explain why SEROM-CSS is more efficient, we further show in Figure 3 the comparison of time consumed for the ROM construction and integration for SROM-CSS and SEROM-CSS. We note that for SROM-CSS, the time spent for constructing ROMs is 186.3 s, which is more than 97% of the total CPU time. The ROM integration is much cheaper. As is expected, SEROM-CSS needs only 5.2 s in building ROMs. This tremendous time reduction is achieved at the very low expense of more ROM integration.

The results presented above validate that both methods produce highly accurate ROMs of significantly lower orders for CSS computation. SEROM-CSS has the advantage of being computationally more efficient. Thus, we consider only this method in the following case studies.

Uncertainty quantification

UQ aims at quantifying the impact of uncertainties in input parameters upon outputs of a system. It remains a largely unexplored area for continuous chromatographic processes. For the sugar separation problem whose physical settings is shown in Table 1, Kurup et al.[37] combined a Monte Carlo-based method with random sampling to quantify the effect of the isotherm uncertainty on the separation performance. In their work, the input uncertainties were propagated through an expensive full-order SMB model. In the Monte Carlo method, a large number of FOM simulations are often required to yield accurate results, rendering the analysis procedure time-consuming. In this section, we will use SEROM-CSS for fast and accurate uncertainty analysis.

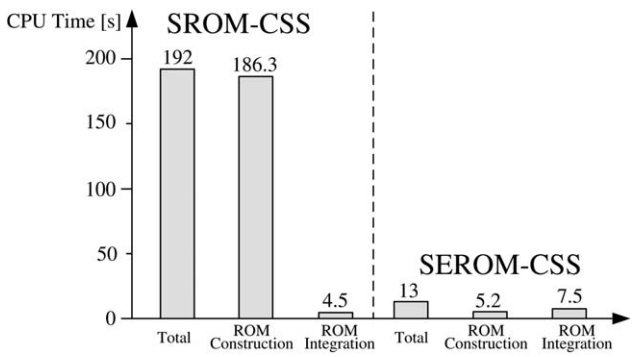

Figure 3. Detailed comparison of CPU time consumed by ROM construction and integration for SROM-CSS and SEROM-CSS.

We follow the same assumption made in Kurup et al.[37] that the two isotherm parameters, namely the Henry constants H_a and H_b, are random input variables with uniform distribution. Due to inevitable measurement errors, these parameters are subject to a $\pm 10\%$ deviation from the nominal values given in Table 1. The product purities were chosen as the output variables. The Latin hypercube sampling approach was used to generate the sample set. To avoid increasing the ROM order dynamically, we also chose a relatively high order of 50 for ROMs before performing the analysis. As was shown in Figure 2, this order allows the residual at the nominal point significantly lower than the tolerance $\epsilon_r = 1.0 \times 10^{-3}$. We thus expect that it is conservative enough and should be appropriate for other sampling points. In fact, with this choice we found that the residual values at all sampling points satisfy the required tolerance.

The probabilistic metrics of the output parameters in terms of the mean value $E[\cdot]$ and the standard deviation $\sigma[\cdot]$ are presented in Table 3. We observe that the ROM-based analysis produces almost the same results as the FOM-based scheme. The mean values and standard deviations are correct up to four and three significant digits, respectively. The total CPU time is shortened from more than 6 h to only 11.4 min. The analysis process is accelerated by a factor of over 31. We also note that the purity of fructose $\mathrm{Pu_a}$ shows more obvious deviation from its mean value than the purity of glucose $\mathrm{Pu_b}$. This indicates that $\mathrm{Pu_a}$ is more sensitive to the isotherm uncertainty. Furthermore, the effect of variations in the isotherm parameters on the purities is shown in Figure 4 in a scatter plot. Clearly, the ROM-based UQ captures a trend nearly identical to that predicted by the FOM-based analysis. With an increase in H_b, $\mathrm{Pu_a}$ sharply decreases but does not follow a clear trend with respect to the change in H_a. On the contrary, $\mathrm{Pu_b}$ is more sensitive to the variation in H_a rather than in H_b. The observations made here are quantitatively identical to those reported by Kurup et al.[37]

Optimization of the CSS

Optimization of the SMB process based on its FOM is often an expensive task. The sequential method[38,39] and the simultaneous method[7,40] have been developed for solving the CSS optimization problem. In the sequential method, the integration of the FOM system must be performed at each iteration of the optimization procedure in order to fulfill the CSS condition. The simultaneous scheme, on the other hand, further discretizes the temporal dimension of the FOM and treats the CSS condition as a set of equality constraints of the optimization problem. The sequential method usually leads to a small-size nonlinear programming (NLP) problem

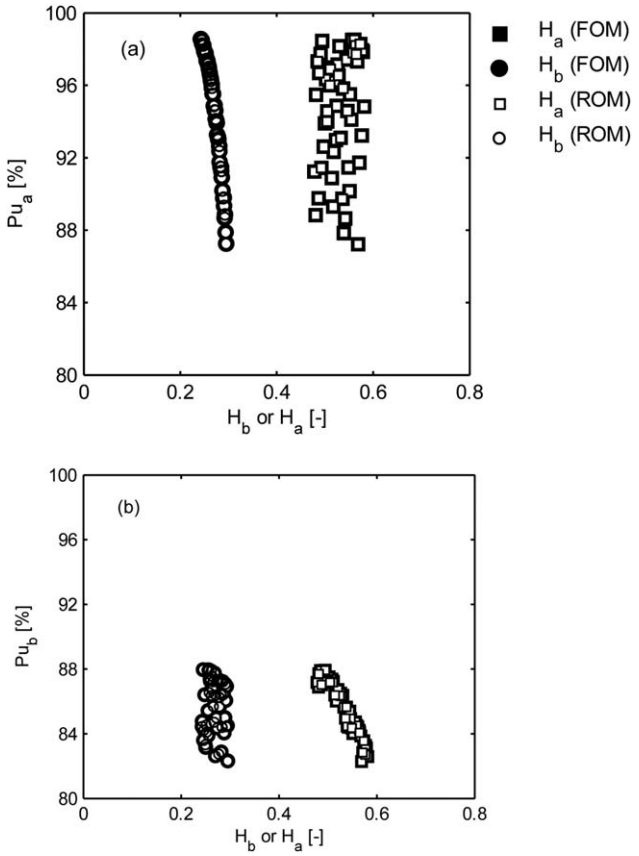

Figure 4. Effect of variations in H_a and H_b on the purity of fructose (a) and glucose (b).

and is thus easy to implement. However, the time-consuming CSS integration renders it less efficient. The simultaneous strategy avoids the integration loop but results in a large-scale sparse NLP problem, for which specially tailored optimization algorithms must be used to ensure satisfactory solution efficiency. MOR can serve as a potential method for alleviating the computational difficulties arising in both solution strategies. In this section, we examine particularly the potential of embedding SEROM-CSS into the sequential method to accelerate the CSS optimization.

A second literature example with linear isotherm, which was reported by Araújo et al.,[10] was selected as the test case. The different geometric, thermodynamic, and kinetic parameters and the operating conditions are listed in Table 4. The system is optimized for maximal feed throughput while fulfilling the purity specifications and operating constraints. This optimization problem is formulated mathematically as follows

$$\min_{p} \; -Q_{\mathrm{F}}$$

such that

Table 4. Model Parameters and Operating Conditions for the Second Example, Which was Described by Araújo et al.[10]

Number of columns n_{col}	8	Mass-transfer coefficient K_a (s^{-1})	0.1
Column configuration	2-2-2-2	Mass-transfer coefficient K_b (s^{-1})	0.1
Column dimensions (cm)	2.6 × 11	Feed concentrations (g/L)	2.9
Column porosity ϵ	0.4	Henry constant H_a	3.86
Péclet number Pe	1000	Henry constant H_b	2.72

$$\mathrm{Pu_a}(p)=\frac{\int_0^1 c^{\mathrm{E}}_{\mathrm{a,css}}(\tau,p)d\tau}{\int_0^1 (c^{\mathrm{E}}_{\mathrm{a,css}}(\tau,p)+c^{\mathrm{E}}_{\mathrm{b,css}}(\tau,p))d\tau}\geq \mathrm{Pu_{a,min}},$$

$$\mathrm{Pu_b}(p)=\frac{\int_0^1 c^{\mathrm{R}}_{\mathrm{b,css}}(\tau,p)d\tau}{\int_0^1 (c^{\mathrm{R}}_{\mathrm{a,css}}(\tau,p)+c^{\mathrm{R}}_{\mathrm{b,css}}(\tau,p))d\tau}\geq \mathrm{Pu_{b,min}},$$

$$Q_{\mathrm{I}}(p)\leq Q_{\max},$$

$$m_{\mathrm{I}}-m_{\mathrm{II}}>0, m_{\mathrm{I}}-m_{\mathrm{IV}}>0, m_{\mathrm{III}}-m_{\mathrm{II}}>0, m_{\mathrm{III}}-m_{\mathrm{IV}}>0. \tag{48}$$

In Problem (48), $\mathrm{Pu_a}$ and $\mathrm{Pu_b}$ are the extract and raffinate product purities, respectively. Variables $c^{\mathrm{E}}_{\alpha,\mathrm{css}}(\tau,p)$ and $c^{\mathrm{R}}_{\alpha,\mathrm{css}}(\tau,p)$ denote the CSS concentrations withdrawn from the extract and raffinate outlets for component α, respectively. The product purities must fulfill the minimal acceptable purity values $\mathrm{Pu_{a,min}}$ and $\mathrm{Pu_{b,min}}$ to ensure the on-spec production. The other inequality constraints impose the flow rate restrictions that must be respected during the operation of the unit. The CSS operating conditions $p=[m_{\mathrm{I}}, m_{\mathrm{II}}, m_{\mathrm{III}}, m_{\mathrm{IV}}, t_s]^T\in\mathbb{R}^5$ consists of the switching period t_s and four dimensionless m-factors introduced by the triangle theory[41]

$$m_\omega=\frac{Q_\omega t_s-V_{\mathrm{col}}\epsilon}{V_{\mathrm{col}}(1-\epsilon)}, \omega=\mathrm{I,II,III,IV} \tag{49}$$

where V_{col} is the column volume.

During the solution of the NLP problem above, we use both FOM-CSS and SEROM-CSS to calculate the CSS solution. For each column, five finite elements combining three internal collocation points were used to discretize the axial domain of the PDAE model, and the resulting FOM system consists of 672 DAEs for each stage. The dimensionless concentration profiles were used to check CSS numerically with the tolerance $\epsilon_{\mathrm{css}}=1.0\times10^{-4}$. Both $\mathrm{Pu_{a,min}}$ and $\mathrm{Pu_{b,min}}$ were specified as 99%. The maximal allowable flow rate $Q_{\max}$ was fixed at 30 mL/min. The optimization started from the initial guess $p_0=[4.5, 2.9, 3.5, 2.5, 7.0\ \mathrm{min}]^T$, whose elements represent the unknown four dimensionless flow rate ratios m_ω (ω=I, II, III, IV) and the switching period, respectively. The

Table 5. Comparison of CSS Optimization Results Obtained with the FOM-Based and ROM-Based Methods for the Second Example

Method	Model Order	Q_{I} (mL/min)	Q_{II} (mL/min)	Q_{III} (mL/min)	Q_{IV} (mL/min)	t_s (min)	Ni	CPU Time (min)
FOM-CSS	672	30.00	21.28	25.89	18.40	5.828	7	88.2
SEROM-CSS	50	30.00	21.26	25.88	18.41	5.834	6	5.3

Ni represents the number of iterations required to achieve convergence.

gradients of the objective function and constraints required by the optimizer E04UCF from the NAG library were calculated analytically except for those of the purity constraints. In this work, they were approximated by a finite difference method. Finally, we used a strategy similar to that discussed in the previous section to choose the ROM order. In this example, the initial order was fixed at 50. With this choice, the required residual criterion $\epsilon_r = 1.0 \times 10^{-3}$ was always satisfied during the course of optimization, and thus it was not necessary to update the order dynamically.

The optimization results are summarized in Table 5. The FOM-based optimization takes seven iterations to locate the optimum and the total computational time is more than 1 h. By contrast, with the help of the ROM-based scheme, the optimizer converges to almost the same solution in only 5.3 min after six iterations. The optimization process is sped up by a factor of more than 16. This solution time achieved with the sequential method is quite short and is already comparable to that using the very efficient simultaneous strategy.[40] We also calculated the product purities at the optimum found by the ROM-based optimization with the FOM simulation to validate if they really fulfill the purity specifications. It is confirmed that the purity constraints are satisfied within the allowed tolerance of 1.0×10^{-4}. The case study clearly demonstrates the great potential of SEROM-CSS for accelerating CSS optimization.

Conclusions

In this article, a Krylov-subspace-based MOR technique was suggested to reduce a classical high-dimensional multistage model of the SMB chromatography. A straightforward method (SROM-CSS) and a subspace-exploiting method (SEROM-CSS) were proposed. To follow the evolution trajectory in the FOM-based scheme (FOM-CSS), SROM-CSS builds a ROM for each stage and integrates each ROM sequentially until the CSS is reached. SEROM-CSS, on the other hand, introduces an additional inner loop for each ROM. An "optimal" trajectory, namely the so-called pseudo-CSS, can be located by integrating this ROM repeatedly. By thoroughly exploiting each ROM, SEROM-CSS requires to build much fewer reduced models in computing the CSS solution. As a result, the cost for building ROMs that is the computationally most dominant part in both strategies is significantly reduced. The performance of the proposed methods was compared to that of FOM-CSS using a sugar separation as a numerical example. The simulation results showed that both methods allow using ROMs of significantly lower order to obtain a highly accurate CSS prediction. SEROM-CSS is proven to be the most efficient scheme, although SROM-CSS also clearly outperforms FOM-CSS. Two typical model-based applications in analyzing SMB processes, namely UQ and CSS optimization, illustrated that the ROM-based solution approach not only computes almost the same results as the FOM-based scheme, but also results in a remarkable speedup of 16–47 in computation time depending on different cases and analyses.

It should be pointed out that although SROM-CSS is less efficient than SEROM-CSS in computing CSS, it can be easily adapted to quickly reproduce transient behavior of SMB. Thus, the adapted version can be valuable for various non-CSS-based issues, such as online control[14] and optimal transient operation such as start-up and shutdown.[42]

In future work, we will explore the potential of combining the Krylov-type MOR method with the full-discretization scheme for CSS evaluation. Extension of our work to SMB processes with nonlinear isotherms, for example, using a trajectory piecewise-linear MOR method,[43] serves as another challenging research direction.

Literature Cited

1. Broughton DB, Gerhold CG. Continuous sorption process employing fixed beds of sorbent moving inlets and outlets. U.S. Patent 2985589. 1961.
2. Ganetsos G, Barker P, editors. *Preparative and Production Scale Chromatography, Volume 61 of Chromatographic Science Series*. New York: CRC Press, 1992.
3. Negawa M, Shoji F. Optical resolution by simulated moving-bed adsorption technology. *J Chromatogr A*. 1992;590(1):113–117.
4. Rajendran A, Paredes G, Mazzotti M. Simulated moving bed chromatography for the separation of enantiomers. *J Chromatogr A*. 2009;1216(4):709–738.
5. Seidel-Morgenstern A, Keßler LC, Kaspereit M. New developments in simulated moving bed chromatography. *Chem Eng Technol*. 2008; 31(6):826–837.
6. Kloppenburg E, Gilles ED. A new concept for operating simulated moving-bed processes. *Chem Eng Technol*. 1999;22(10):813–817.
7. Araújo JMM, Rodrigues RCR, Mota JPB. Optimal design and operation of a certain class of asynchronous simulated moving bed processes. *J Chromatogr A*. 2006;1132(1–2):76–89.
8. Toumi A, Engell S, Diehl M, Bock HG, Schlöder J. Efficient optimization of simulated moving bed processes. *Chem Eng Process*. 2007; 46:1067–1084.
9. Mota JPB, Araújo JMM. Single-column simulated-moving-bed process with recycle lag. *AIChE J*. 2005;51(6):1641–1653.
10. Araújo JMM, Rodrigues RCR, Mota JPB. Use of single-column models for efficient computation of the periodic state of a simulated moving-bed process. *Ind Eng Chem Res*. 2006;45(15):5314–5325.
11. Antoulas AC. Approximation of Large-Scale Dynamical Systems, Volume 6 of Advances in Design and Control. Philadelphia, PA: SIAM, 2005.
12. Benner P, Gugercin S, Willcox K. A survey of model reduction methods for parametric systems. Max Planck Institute Magdeburg Preprint MPIMD/13–14. MPI-Magdeburg, Magdeburg, 2013. Available at http://www.mpi-magdeburg.mpg.de/preprints/. Last accessed on July 22, 2014.
13. Baur U, Benner P, Feng L. Model order reduction for linear and nonlinear systems: a system-theoretic perspective. Technical Report MPIMD/14-07. Max Planck Institute Magdeburg, Magdeburg, 2014. Accessed on July 22, 2014.
14. Erdem G, Abel S, Morari M, Mazzotti M, Morbidelli M, Lee JH. Automatic control of simulated moving beds. *Ind Eng Chem Res*. 2004;43(2):405–421.
15. Vilas C, Vande Wouwer A. Combination of multi-model predictive control and the wave theory for the control of simulated moving bed plants. *Chem Eng Sci*. 2011;66(4):632–641.
16. Boulkroune B, Kinnaert M, Zemouche A. POD-based state estimation of simulated moving bed chromatographic processes. Proceedings of the 12th European Control Conference. Zurich: EUCA, 2013:3408–3414.
17. Li S, Feng L, Benner P, Seidel-Morgenstern A. Efficient optimization of simulated moving bed chromatographic processes using reduced order models. *Comput Aided Chem Eng*. 2012;30:1232–1236.
18. Li S, Feng L, Benner P, Seidel-Morgenstern A. Using surrogate models for efficient optimization of simulated moving bed chromatography. *Comput Chem Eng*. 2014;14(4):121–132.
19. Guiochon G, Felinger A, Shirazi DG. *Fundamentals of Preparative and Nonlinear Chromatography*, 2nd ed. Amsterdam: Academic Press, 2006.
20. Thibodeaux LJ, Mackay D, editors. *Handbook of Chemical Mass Transport in the Environment*. Boca Raton: CRC Press, 2010.
21. Carey GF, Finlayson BA. Orthogonal collocation on finite elements. *Chem Eng Sci*. 1975;30(5–6):587–596.
22. Ma Z, Whitley RD, Wang NHL. Pore and surface diffusion in multicomponent adsorption and liquid chromatography systems. *AIChE J*. 1996;42(5):1244–1262.
23. Minceva M, Pais LS, Rodrigues AE. Cyclic steady state of simulated moving bed processes for enantiomers separation. *Chem Eng Process*. 2003;42(2):93–104.
24. Arnoldi WE. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Q Appl Math*. 1951;9(17):17–29.
25. Lanczos C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J Res Natl Bur Stand*. 1950;45(4):255–282.

26. Daniel J, Gragg W, Kaufman L, Stewart G. Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization. *Math Comput*. 1976;30(136):772–795.
27. Bai Z. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Appl Numer Math*. 2002;43(1–2):9–44.
28. Salimbahrami B, Lohmann B. Order reduction of large scale second-order systems using Krylov subspace methods. *Linear Algebra Appl*. 2006;415(2–3):385–405.
29. Feng L, Koziol D, Rudnyi EB, Korvink JG. Parametric model reduction for fast simulation of cyclic voltammograms. *Sens Lett*. 2006; 4(2):165–173.
30. Bai Z, Ye Q. Error estimation of the Padé approximation of transfer functions via the Lanczos process. *ETNA*. 1998;7:1–17.
31. Meerbergen K. The solution of parametrized symmetric linear systems. *SIAM J Matrix Anal Appl*. 2003;24(4):1038–1059.
32. Simoncini V, Perotti F. On the numerical solution of $(\lambda^2 A+\lambda B+C)x=b$ and application to structural dynamics. *SIAM J Sci Comput*. 2002;23(6):1875–1897.
33. Meerbergen K. Fast frequency response computation for Rayleigh damping. *Int J Numer Methods Eng*. 2008;73(1):96–106.
34. Meerbergen K, Bai Z. The Lanczos method for parameterized symmetric linear systems with multiple right-hand sides. *SIAM J Matrix Anal Appl*. 2010;31(4):1642–1662.
35. Azevedo DCS, Rodrigues AE. Fructoseglucose separation in a SMB pilot unit: modelling, simulation, design and operation. *AIChE J*. 2001;47(9):2042–2051.
36. Li S, Petzold L. Design of new DASPK for sensitivity analysis. Technical Report. California: Department of Computer Science, University of California Santa Barbara, 1999.
37. Kurup AS, Subramani HJ, Harris MT. A Monte Carlo-based error propagation analysis of simulated moving bed systems. *Sep Purif Technol*. 2008;62(3):582–589.
38. Dünnebier G, Klatt KU. Optimal operation of simulated moving bed chromatographic processes. *Comput Chem Eng*. 1999;23(Suppl 1): S195–S198.
39. Dünnebier G, Fricke J, Klatt KU. Optimal design and operation of simulated moving bed chromatographic reactors. *Ind Eng Chem Res*. 2000;39(7):2290–2304.
40. Kawajiri Y, Biegler LT. Optimization strategies for simulated moving bed and PowerFeed processes. *AIChE J*. 2006;52(4):1343–1350.
41. Mazzotti M, Storti G, Morbidelli M. Optimal operation of simulated moving bed units for nonlinear chromatographic separations. *J Chromatogr A*. 1997;769(1):3–24.
42. Li S, Kawajiri Y, Raisch J, Seidel-Morgenstern A. Optimization of startup and shutdown operation of simulated moving bed chromatographic processes. *J Chromatogr A*. 2011;1218(25):3876–3889.
43. Rewienski M, White J. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Trans Comput Aided Design Integr Circuits Syst*. 2003;22(2):155–170.

Manuscript received May 15, 2014, and revision received July 12, 2014.