

Accelerating optimization and uncertainty quantification of nonlinear SMB chromatography using reduced-order models

Yongjin Zhang^a, Lihong Feng^{a,*}, Andreas Seidel-Morgenstern^{a,b}, Peter Benner^a

^aMax Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany.

^bInstitute of Process Engineering, Otto von Guericke University, 39106 Magdeburg, Germany.

Abstract

A parametrized reduced-order model is constructed and employed as a surrogate for the full-order model in optimization and uncertainty quantification of nonlinear simulated moving bed chromatography. The reduced-order model is obtained by the reduced basis method using an efficient error estimation. The complexity of the model is reduced by an empirical interpolation method applied to the nonlinear part of the model. Due to the reduced size and complexity of the surrogate model, the processes of optimization and uncertainty quantification are sped up by a factor of 10.

Keywords: reduced basis method, empirical interpolation, cyclic steady state, adaptive snapshot selection, simulated moving bed chromatography, optimization, uncertainty quantification

1. Introduction

Simulated moving bed (SMB) chromatography, a continuous multi-column chromatographic process, has been recognized as a very useful technology for separation processes and is widely used in food, fine chemistry, and pharmaceutical industries at all production scales [44]. The separation process is accomplished through continuous and counter-current movement of the liquid and solid phases driven by periodically switching the positions of the inlet and outlet streams. Due to the periodic switching procedure, the regime of the SMB system never reaches a steady state, but rather a cyclic steady state (CSS). That is, during the CSS period, the concentration profiles are still varying over time, but they are identical between two consecutive switching periods. To make full use of the economic potential of the SMB process, efficient design and optimization of SMB chromatography play an important role and have gained tremendous attention during the past years; see, e.g., [2, 3, 12, 13, 25, 28, 44, 49, 50, 51]. It is worth noting that all the works mentioned are based on high-fidelity models resulting

from discretization of the physical model (partial differential equations). To precisely capture the dynamics of the process, these high-fidelity models are often of large size and high complexity. The main advantage of using such large-scale high-fidelity models is that the accuracy and reliability of the optimization can be guaranteed. Nevertheless, it is time-consuming to solve such high-fidelity models, especially in many-query contexts, e.g., in optimization, uncertainty quantification (UQ), and real-time control settings. To overcome this obstacle, surrogate models via reduced-order modeling have gained increasing attention in the past decades [14, 15, 27, 29, 53].

Model order reduction (MOR) is a useful tool in handling large-scale computations in science and engineering. MOR aims at constructing a low-cost reduced-order model (ROM), which can reproduce the main dynamics of the large-scale high-fidelity model, called the full-order model (FOM) in this work. Till now, various MOR methods have been proposed and successfully applied to different engineering contexts [1, 5, 7, 8, 39, 43, 48]. However, research on MOR for SMB chromatography is limited in the literature. A balanced truncation MOR method was applied to a linear SMB model in [14]. Recently, a Krylov-subspace MOR method was successfully applied, also to linear SMB chromatography in [29]. For nonlinear SMB chromatography, the application of proper orthogonal decomposition (POD) can be found in [27, 53]. In particular, multi-fidelity

*Corresponding author

Email addresses: zhangy@mpi-magdeburg.mpg.de (Yongjin Zhang), feng@mpi-magdeburg.mpg.de (Lihong Feng), seidel-morgenstern@mpi-magdeburg.mpg.de (Andreas Seidel-Morgenstern), benner@mpi-magdeburg.mpg.de (Peter Benner)

surrogate models were discussed in [27]. Nevertheless, the ROM constructed by the POD or Krylov-subspace MOR method is reliable locally, i.e., it is valid only in the neighborhood of the parameter at which the ROM is constructed. As a result, the ROM needs to be updated in many-query contexts, e.g., during the trust-region optimization process in [27].

In this paper, we present a parametric model order reduction (PMOR) method for the nonlinear SMB chromatography, namely, the reduced basis (RB) method [22, 23, 38, 39, 41, 47]. PMOR is designed for a broad class of problems for which the governing equations depend on a set of parameters. The parameter enters the model in many ways, characterizing, e.g., material properties, geometry configurations, initial conditions, boundary conditions, source or force terms, etc. Using PMOR methods, the parameters in the FOM will be kept as symbolic quantities in the ROM so that the resulting ROM is globally reliable, that is, it is able to reproduce the main dynamics of the FOM corresponding to any variation of the parameter in the entire parameter domain. A comprehensive survey of PMOR methods can be found in [7]. In particular, the RB method has been recognized as a robust PMOR method and is widely applied to various problems described by parameterized partial differential equations (PDEs). Using the RB method, the ROM construction and simulation using the ROM can be entirely decoupled into two stages. That is, during the offline stage, all the high-dimension dependent quantities can be precomputed. This process can be expensive, but is performed only once. During the online stage, given any feasible parameter value, the output response can be obtained rapidly by assembling and solving the ROM without resorting to the FOM. This is quite suitable for the aforementioned many-query tasks. In addition, using the RB method, the ROM is constructed iteratively through a greedy algorithm. At each iteration, the parameter that causes the largest error (measured by some error estimator) is selected and used to enrich the current RB. This process continues until the largest error goes below a given error tolerance. As a result, the resulting ROM is accurate enough in the whole parameter domain, and the dimension of the RB (i.e., the order of the ROM) can be kept as small as possible when an efficient error estimator is employed. When the resulting ROM is employed to solve an optimization problem, one single ROM is qualified so that there is no need to update the ROM online, which saves a lot of computational time.

To demonstrate the performance of the parametric ROM, we use it to accelerate the optimization of SMB chromatography and to quantify the robustness of the

optimal solution (alternatively, the product purity) under flow rate uncertainties. As pointed out in [33], uncertainties in the parameters of adsorption isotherm equations, pump stability, extra-column volumes, and packing reproducibility are inevitable in every SMB process. In [33], a robust design of a linear SMB process under flow rate uncertainties was proposed based on the FOM. In this work, the ROM is employed for fast UQ of the product purity of a nonlinear SMB process.

This paper is structured as follows. Section 2 addresses the SMB model, the corresponding discrete system, and the optimization problem. Section 3 shows that the order and the complexity of the FOM are reduced by the RB method and the empirical interpolation method (EIM) [4], respectively. A parametric ROM is derived for the SMB model. To efficiently construct the ROM, an a posteriori error estimation is presented in Section 4. The problems of optimization and UQ are described in Section 5 considering a case study reported in the literature. The efficiency of solving the above problems by using the ROM is presented. Conclusions are drawn in Section 6.

2. SMB chromatography and optimization

2.1. Model description

An SMB unit typically consists of several identical chromatographic columns connected in a series, as shown in Figure 1. Four ports divide the SMB unit into four zones, which play different roles in a separation process. The mixture to be separated and the eluent are fed continuously through the two inlets, and the two purified components are withdrawn also continuously from the two outlets, respectively. The separation regime is accomplished through simulating a counter current movement of the liquid and solid phases by synchronously shifting the inlet and outlet ports one column ahead in the direction of the fluid flow in a certain switching period t_s . For more details and application examples we refer to [3, 19, 37, 44, 45, 49].

The main dynamics of the fluid flow in all columns in an SMB unit are the same except for the node balance relations. We first address the mathematical modeling of the dynamics in one chromatographic column and then describe the node balance equations in between. In this work, we assume that the dynamics of each chromatographic column can be described by an axially dispersed plug flow model with a limited mass-transfer rate quantified by a linear driving force (LDF) approximation. The mass balance in the column k ($k = 1, \dots, N_{\text{col}}$)

can be given by, [19],

$$\frac{\partial c_{z,k}}{\partial t} + \frac{1-\epsilon}{\epsilon} \frac{\partial q_{z,k}}{\partial t} = -\frac{Q_k t_s}{\epsilon A_c L} \left(\frac{\partial c_{z,k}}{\partial x} - \frac{1}{Pe} \frac{\partial^2 c_{z,k}}{\partial x^2} \right), \quad (1)$$

$$0 < x < 1,$$

where $c_{z,k}$, $q_{z,k}$ are the concentrations of the component z ($z = a, b$) in the liquid and solid phases in the k th column, respectively, Q_k the flow rate, A_c the cross-sectional area of the column, L the column length, ϵ the column porosity, t_s the switching period, and Pe the Péclet number. Note that t and x are in the dimensionless form, i.e., $t = \tilde{t}/t_s$, $x = \tilde{x}/L$ (\tilde{t} is the time coordinate and \tilde{x} is the axial coordinate along the column). The adsorption rate is described by the LDF approximation as follows:

$$\frac{\partial q_{z,k}}{\partial t} = t_s \kappa_z (q_{z,k}^{\text{Eq}} - q_{z,k}), \quad 0 \leq x \leq 1, \quad (2)$$

where κ_z is the mass-transfer coefficient and $q_{z,k}^{\text{Eq}}$ is the adsorption equilibrium defined by the adsorption isotherm function

$$q_{z,k}^{\text{Eq}} = f_z(c_{a,k}, c_{b,k}). \quad (3)$$

Different separation processes are described by different adsorption isotherm equations. In this work, the adsorption equilibrium $q_{z,k}^{\text{Eq}}$ is described by the isotherm equations of bi-Langmuir type [19],

$$q_{z,k}^{\text{Eq}} = f_z(c_{a,k}, c_{b,k}) := \frac{H_{z1} c_{z,k}}{1 + K_{a1} c_{a,k} + K_{b1} c_{b,k}} + \frac{H_{z2} c_{z,k}}{1 + K_{a2} c_{a,k} + K_{b2} c_{b,k}}, \quad (4)$$

where H_{zj} and K_{zj} ($j = 1, 2$) are the Henry constants and thermodynamic coefficients, respectively.

Danckwerts-type boundary conditions are imposed to equation (1), i.e.,

$$\begin{cases} \frac{\partial c_{z,k}}{\partial x} \Big|_{x=0} = Pe(c_{z,k}(t, 0) - c_{z,k}^{\text{in}}(t)), \\ \frac{\partial c_{z,k}}{\partial x} \Big|_{x=1} = 0, \end{cases} \quad (5)$$

where $c_{z,k}^{\text{in}}(t)$ is the concentration of component z at the inlet of column k . The system is closed by assuming uniformly pre-equilibrated columns as the initial conditions,

$$\begin{cases} c_{z,k}(0, x) = c_{z,k}^0, \\ q_{z,k}(0, x) = q_{z,k}^0, \end{cases} \quad 0 \leq x \leq 1, \quad k = 1, \dots, N_{\text{col}}.$$

Neglecting dead volumes, the concentrations at the inlet of a column are equal to the concentrations at the outlet of the previous column, i.e.,

$$c_{z,k+1}^{\text{in}}(t) = c_{z,k}(t, 1), \quad (6)$$

except for the feed and desorbent nodes. The mass balances at the inlet and outlet ports can be established as follows:

Desorbent node:

$$\begin{aligned} Q_I &= Q_{IV} + Q_D, \\ Q_I c_{z,1}^{\text{in}}(t) &= Q_{IV} c_{z,N_{\text{col}}}(t, 1). \end{aligned} \quad (7)$$

Extract node:

$$\begin{aligned} Q_{II} &= Q_I - Q_E, \\ c_{z,k_1+1}^{\text{in}}(t) &= c_{z,k_1}(t, 1). \end{aligned} \quad (8)$$

Feed node:

$$\begin{aligned} Q_{III} &= Q_{II} + Q_F, \\ Q_{III} c_{z,k_1+k_{II}+1}^{\text{in}}(t) &= Q_{II} c_{z,k_1+k_{II}}(t, 1) + c_z^F Q_F. \end{aligned} \quad (9)$$

Raffinate node:

$$\begin{aligned} Q_{IV} &= Q_{III} - Q_R, \\ c_{z,k_1+k_{II}+k_{III}+1}^{\text{in}}(t) &= c_{z,k_1+k_{II}+k_{III}}(t, 1). \end{aligned} \quad (10)$$

Here, k_J is the number of columns in the zone J , Q_J is the flow rate in the corresponding zone, $J \in \{I, II, III, IV\}$, c_z^F is the feed concentration of the solute z , and Q_D , Q_E , Q_F , Q_R are the desorbent, extract, feed, and raffinate flow rate, respectively. Note that the flow rates within all columns in a certain zone are assumed to be uniform. For example, for an SMB unit with eight columns and 2-2-2-2 configurations, the flow rates Q_k ($k = 1, \dots, 8$) in the columns satisfy the following relations:

$$\begin{aligned} Q_1 &= Q_2 = Q_I, & Q_3 &= Q_4 = Q_{II}, \\ Q_5 &= Q_6 = Q_{III}, & Q_7 &= Q_8 = Q_{IV}. \end{aligned}$$

2.2. Numerical discretization

We use the finite volume method to discretize the SMB model (1)–(5), where the Lax–Friedrichs flux [26] is used to solve the convection flux, and the central difference scheme is applied to evaluate the diffusion flux. For the temporal discretization, we use the Crank–Nicolson scheme, which yields a second-order accurate evolution scheme.

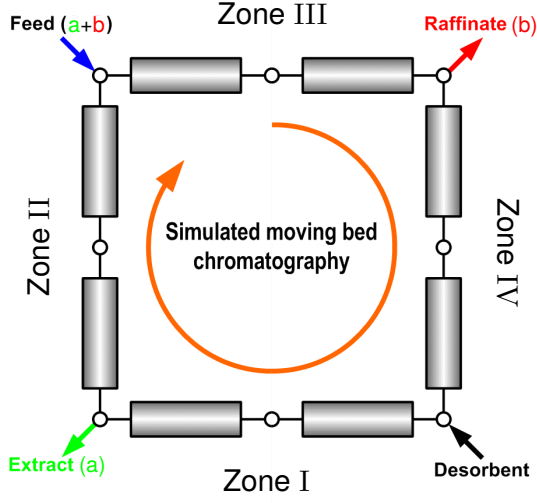


Figure 1: Schematic illustration of an SMB chromatographic process with four zones and eight columns.

Let $\Delta t = 1/K$, $t^n = n\Delta t$, $n \in \mathbb{K} := \{0, 1, \dots, K\}$ be the $K + 1$ time instants over the time interval $[0, 1]$, $\Delta x = 1/\tilde{N}$ be the spatial grid size, and $c_{z,k}^n(\mu)$, $q_{z,k}^n(\mu) \in \mathbb{R}^{\tilde{N}}$ be the numerical approximations of the concentrations ($c_{z,k}$ and $q_{z,k}$, respectively) in the k th column at time instance $t = t^n$. Note that the time interval $[0, 1]$ is actually a dimensionless switching period. The fully discretized finite volume formulation for the k th chromatographic column can be written as

$$A_k(\mu)c_{z,k}^{n+1}(\mu) = B_k(\mu)c_{z,k}^n(\mu) + r_{z,k}^n(c_{z,k}^{\text{in}}(t^n), \mu) - \frac{1-\epsilon}{\epsilon}\Delta th_{z,k}^n(\mu), \quad (11)$$

$$q_{z,k}^{n+1}(\mu) = q_{z,k}^n(\mu) + \Delta th_{z,k}^n(\mu), \quad (12)$$

where $A_k(\mu), B_k(\mu) \in \mathbb{R}^{\tilde{N} \times \tilde{N}}$ are tridiagonal matrices, $h_{z,k}^n(\mu) := f_z(c_{a,k}^n(\mu), c_{b,k}^n(\mu)) - q_{z,k}^n(\mu) \in \mathbb{R}^{\tilde{N}}$ is a nonlinear vector-valued function, and $r_{z,k}^n(c_{z,k}^{\text{in}}(t^n), \mu) = r_{z,k}(c_{z,k}^{\text{in}}(t^n), \mu)[1, 0, \dots, 0]^T \in \mathbb{R}^{\tilde{N}}$ with

$$r_{z,k}(c_{z,k}^{\text{in}}(t^n), \mu) := \left(1 + \frac{Pe}{2\tilde{N}}\right) \frac{\tilde{N}Q_k(\mu)t_s}{\epsilon L^2 A_c} c_{z,k}^{\text{in}}(t^n)\Delta t \in \mathbb{R}. \quad (13)$$

The parameter μ characterizes the operating conditions of the underlying SMB process (e.g., $\mu = [Q_1, \dots, Q_{IV}, t_s]$).

It is noteworthy that $r_{z,k}^n(c_{z,k}^{\text{in}}(t^n), \mu)$ in (11) depends on the information from the connected columns, because

the inflow of the k th column $c_{z,k}^{\text{in}}(t^n)$ is determined by the outflow of the $(k-1)$ st column, $k = 2, \dots, N_{\text{col}}$, and $c_{z,1}^{\text{in}}(t^n)$ is determined by $c_{z,N_{\text{col}}}(t^n, 1)$ —the concentration at the outlet of the N_{col} th column, as detailed in (6)–(10) in the previous subsection. Thus, the system (11)–(12) is coupled with the systems corresponding to the other columns. Assembling all the systems, we obtain the FOM for the SMB unit as follows:

$$A(\mu)c_z^{n+1}(\mu) = B(\mu)c_z^n(\mu) + r_z^n(\mu) - \frac{1-\epsilon}{\epsilon}\Delta th_z^n(\mu), \quad (14)$$

$$q_z^{n+1}(\mu) = q_z^n(\mu) + \Delta th_z^n(\mu), \quad (15)$$

where

$$c_z^n(\mu) = \begin{bmatrix} c_{z,1}^n(\mu) \\ \vdots \\ c_{z,N_{\text{col}}}^n(\mu) \end{bmatrix}, \quad q_z^n(\mu) = \begin{bmatrix} q_{z,1}^n(\mu) \\ \vdots \\ q_{z,N_{\text{col}}}^n(\mu) \end{bmatrix} \in \mathbb{R}^{\mathcal{N}},$$

$$h_z^n(\mu) = \begin{bmatrix} h_{z,1}^n(\mu) \\ \vdots \\ h_{z,N_{\text{col}}}^n(\mu) \end{bmatrix} \in \mathbb{R}^{\mathcal{N}},$$

$A(\mu) = \text{diag}(A_1(\mu), \dots, A_{N_{\text{col}}}(\mu)) \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$, $B(\mu) = \tilde{B}(\mu) + B_r(\mu)$ with a block diagonal matrix $\tilde{B}(\mu) = \text{diag}(B_1(\mu), \dots, B_{N_{\text{col}}}(\mu)) \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ and a (fairly) sparse matrix $B_r(\mu)$ resulting from separating an auxiliary vector

$$r_z^n(c_z^{\text{in}}(t^n), \mu) := \begin{bmatrix} r_{z,1}^n(c_{z,1}^{\text{in}}(t^n), \mu) \\ \vdots \\ r_{z,N_{\text{col}}}^n(c_{z,N_{\text{col}}}^{\text{in}}(t^n), \mu) \end{bmatrix} = B_r(\mu)c_z^n(\mu) + r_z^n(\mu)$$

into two parts depending on the definition of $r_{z,k}(c_{z,k}^{\text{in}}(t^n), \mu)$ in (13), $B_r(\mu) \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$, $r_z^n(\mu) \in \mathbb{R}^{\mathcal{N}}$, and $\mathcal{N} = \tilde{N} \cdot N_{\text{col}}$. The first part, $B_r(\mu)c_z^n(\mu)$, linearly depends on $c_z^n(\mu)$ and is determined by (6)–(10). The second part, $r_z^n(\mu)$, only depends on the parameter μ , which leads to an efficient implementation using the ROM, to be introduced in Section 3.3. Nevertheless, this does not introduce any extra cost for the FOM simulation.

As mentioned earlier, the regime of the SMB is a cyclic steady state, which is characterized by identical transient concentration profiles during two consecutive switching periods. Mathematically, given an initial state, the system (14)–(15) is solved step by step within a switching period; at the end of a period, the state $(c_z^K(\mu), q_z^K(\mu))$ undergoes a shift, and the shifted vectors $(P_s c_z^K(\mu), P_s q_z^K(\mu))$ are used as the new initial state to continue the evolution process until the CSS conditions are satisfied. Here P_s is a permutation matrix. To determine the condition of the CSS, the following two criteria can be used:

1. check whether the concentrations at the beginning of two consecutive periods are identical, i.e.,

$$\begin{aligned} & \max_{z \in \{a,b\}} \max\{\|c_z^0(\mu) - P_s c_z^K(\mu)\|, \|q_z^0(\mu) - P_s q_z^K(\mu)\|\} \\ & < \varepsilon_{\text{CSS}}, \end{aligned}$$

where ε_{CSS} is a user-specified tolerance;

2. check whether the output of interest, e.g., the purity of the product streams defined in (17), in two consecutive periods are equal or not.

2.3. Optimization of SMB chromatography

As mentioned earlier, the optimal operation of SMB chromatography is of practical importance since it allows to exploit the full economic potential of the process and to reduce the separation cost. In this work, we seek the optimal operating conditions that maximize the feed throughput while respecting the purity requirements and the process constraints. The decision variables in the underlying optimization problem are the feed flow rate Q_F and four dimensionless m values m_J , $J = \text{I}, \dots, \text{IV}$, which represent the ratios of the fluid flow rates over the solid flow rate. For notational simplicity, let $\mu := [m_{\text{I}}, \dots, m_{\text{IV}}, Q_F]$. These four m values were introduced in the frame of the triangle theory in [31], which is based on the equilibrium theory of chromatography. These values are closely related to the flow rates within four zones of the SMB unit, Q_J , i.e.,

$$m_J = \frac{t_s Q_J - \varepsilon V_{\text{col}}}{(1 - \varepsilon) V_{\text{col}}}, \quad J = \text{I}, \dots, \text{IV}.$$

Here, V_{col} is the volume of a chromatographic column. Let $\mathcal{P} \subset \mathbb{R}^5$ be an admissible parameter domain, the optimization problem can be formulated as follows:

$$\begin{aligned} & \min_{\mu \in \mathcal{P}} f(\mu) = -Q_F, \\ \text{s.t. } & Pu_{a,\min} - Pu_a(\mu) \leq 0, \\ & Pu_{b,\min} - Pu_b(\mu) \leq 0, \\ & Q_{\text{I}} - Q_{\text{max}} \leq 0, \end{aligned} \quad (16)$$

where

$$\begin{aligned} Pu_a(\mu) &:= \frac{\int_0^1 c_{a,\text{CSS}}^{\text{E}}(t, \mu) dt}{\int_0^1 c_{a,\text{CSS}}^{\text{E}}(t, \mu) dt + \int_0^1 c_{b,\text{CSS}}^{\text{E}}(t, \mu) dt}, \\ Pu_b(\mu) &:= \frac{\int_0^1 c_{b,\text{CSS}}^{\text{R}}(t, \mu) dt}{\int_0^1 c_{a,\text{CSS}}^{\text{R}}(t, \mu) dt + \int_0^1 c_{b,\text{CSS}}^{\text{R}}(t, \mu) dt}, \end{aligned} \quad (17)$$

are the products' purity at the extract and the raffinate outlets, $c_{z,\text{CSS}}^{\text{E}}(t, \mu)$ and $c_{z,\text{CSS}}^{\text{R}}(t, \mu)$ are the CSS concentrations of c_z at the extract and the raffinate outlets, respectively. The constraints $Pu_{a,\min}$, $Pu_{b,\min}$, and Q_{max}

will be specified in the numerical experiments in Section 5. The output is defined as

$$y(t, \mu) := (c_{a,\text{CSS}}^{\text{E}}(t; \mu), c_{a,\text{CSS}}^{\text{R}}(t; \mu), c_{b,\text{CSS}}^{\text{E}}(t; \mu), c_{b,\text{CSS}}^{\text{R}}(t; \mu)). \quad (18)$$

To compute the purity of the products, the large-scale system (14)–(15) must be simulated many times during the optimization process, which is very time-consuming. Instead of using the FOM for the optimization, we employ surrogate ROMs. In what follows, we show how a parametric ROM is generated using the RB method and the EIM.

3. Model order and complexity reduction

In this section, we review the RB PMOR method, which constructs a ROM of significantly reduced size (order). The complexity of the ROM is further reduced using the EIM.

3.1. Order reduction: reduced basis method

The RB method has been recognized as a robust PMOR method and is extensively applied in various application areas for steady and unsteady problems. For more details about RB methods, we refer to [22, 39, 47].

Here we address the framework of the RB method for the SMB model in (14)–(15). We will construct a RB for $c_z^n(\mu)$, $q_z^n(\mu)$, $z = a, b$, respectively. For this, we rewrite each equation in the system in a general form:

$$A(\mu)u^{n+1}(\mu) = B(\mu)u^n(\mu) + g(u^n(\mu), \mu), \quad (19)$$

where $u^n(\mu)$ stands for the unknown vector $c_z^n(\mu)$ in (14) (resp., $q_z^n(\mu)$ in (15)), $A(\mu), B(\mu) \in \mathbb{R}^{N \times N}$ are coefficient matrices (for (15), they are identity matrices), $g(\cdot, \mu)$ stands for the nonlinear terms, e.g., $g(\cdot, \mu) = r_z^n(\mu) - \frac{1-\varepsilon}{\varepsilon} \Delta t h_z^n(\mu)$ in (14) and $g(\cdot, \mu) = \frac{1-\varepsilon}{\varepsilon} \Delta t h_z^n(\mu)$ in (15).

As mentioned earlier, the RB method aims at constructing a parametric ROM that can reproduce the input-output map or the dominant dynamics of the large-scale FOM under parameter variations. A key assumption for the RB method is that the solution $u^n(\mu)$ resides in a low-dimensional subspace \mathcal{V} of \mathbb{R}^N . Let $V \in \mathbb{R}^{N \times N}$ be a RB matrix, i.e., its columns consist of an orthonormal basis of \mathcal{V} . A ROM can be obtained by Galerkin projection as follows:

$$\hat{A}(\mu)u_r^{n+1}(\mu) = \hat{B}(\mu)u_r^n(\mu) + V^T g(Vu_r^n(\mu), \mu), \quad (20)$$

where $\hat{A}(\mu) = V^T A(\mu) V$, $\hat{B}(\mu) = V^T B(\mu) V \in \mathbb{R}^{N \times N}$ are the reduced matrices, and $u_r^n(\mu) \in \mathbb{R}^N$ is the unknown

vector of the ROM. For an efficient offline-online computation, we assume that $A(\mu), B(\mu)$ depend affinely on the parameter μ , i.e., they can be expressed in a separated form,

$$A(\mu) = \sum_{j=1}^{n_a} \xi_j(\mu) A_j, \quad B(\mu) = \sum_{k=1}^{n_b} \zeta_k(\mu) B_k,$$

where A_j, B_k are constant matrices, $\xi_j(\mu), \zeta_k(\mu)$ are the corresponding time and parameter dependent scalar coefficients. This is the case in our applications. The numbers n_a and n_b are expected to be small. Then

$$\begin{aligned} \hat{A}(\mu) &= V^T A(\mu) V = \sum_{j=1}^{n_a} \xi_j(\mu) \hat{A}_j, \\ \hat{B}(\mu) &= V^T B(\mu) V = \sum_{k=1}^{n_b} \zeta_k(\mu) \hat{B}_k, \end{aligned} \quad (21)$$

where $\hat{A}_j = V^T A_j V$ and $\hat{B}_k = V^T B_k V$, $j = 1, \dots, n_a, k = 1, \dots, n_b$. Note that once the projection matrix V is obtained, \hat{A}_j and \hat{B}_k can be precomputed, and in turn, the evaluations of $\hat{A}(\mu)$ and $\hat{B}(\mu)$ for any parameter μ are independent of the dimension N . However, evaluation of the last term in (20), $V^T g(Vu_r^n(\mu), \mu)$, is still N -dependent, due to the nonlinearity of g and/or its non-affine dependence on the parameter μ . At every time step, one must compute a long vector $g(Vu_r^n(\mu), \mu) \in \mathbb{R}^N$, which is of complexity N . Consequently, the computational complexity of solving the ROM is not significantly reduced. To tackle this problem, the EIM or its variants, e.g., the discrete empirical interpolation method [10] or empirical operator interpolation [11], can be employed, which will be addressed in detail in the next subsection.

We now address how to compute the RB V . For parameterized problems, the RB method features in computing V iteratively using a greedy algorithm [52]. A training set $\mathcal{P}_{\text{train}}^{\text{RB}}$ with a finite number of parameter samples is typically chosen a priori as a substitute of the admissible parameter domain \mathcal{P} . At each extension step, a parameter μ_\star that causes the largest error measured by a proper error indicator $\psi(\cdot)$ is chosen from $\mathcal{P}_{\text{train}}^{\text{RB}}$ to enrich the basis. The iteration continues until the error $\psi(\mu_\star)$ goes below the required accuracy ε_{ROM} . For time-dependent problems, the POD-Greedy algorithm [20] is often employed to construct the RB. Algorithm 1 shows the basic steps of the POD-Greedy algorithm. Note that in Step 4, a singular value decomposition (SVD) is performed to compute the first POD mode of \bar{U} (the first column of Q), where Q and Y are unitary matrices, and Σ is a rectangular diagonal matrix with non-negative real numbers on the diagonal.

Algorithm 1 POD-Greedy algorithm for the RB construction

Input: $\mathcal{P}_{\text{train}}^{\text{RB}}, \mu_0, \varepsilon_{\text{ROM}} (< 1)$.

Output: RB $V = [v_1, \dots, v_N]$.

- 1: Initialization: $N = 0, \mu_\star = \mu_0, \eta_N(\mu_\star) = 1, V = []$.
 - 2: **while** $\psi_N(\mu_\star) > \varepsilon_{\text{ROM}}$ **do**
 - 3: Simulate the FOM at μ_\star to get the solution trajectory: $\{u^n(\mu_\star)\}_{n=0}^K$.
 - 4: Perform POD process:
 - Compute $\bar{U} := [\bar{u}^0, \dots, \bar{u}^K]$, $\bar{u}^n = u^n(\mu_\star) - VV^T u^n(\mu_\star)$ ($\bar{u}^n = u^n(\mu_\star)$, if $N = 0$), $n = 0, \dots, K$.
 - Implement SVD of \bar{U} : $\bar{U} = Q\Sigma Y^T$.
 - Take the first column of Q : $v_{N+1} = Q(:, 1)$.
 - 5: Enrich the RB $V := [V, v_{N+1}]$.
 - 6: Update $N = N + 1$.
 - 7: Find $\mu_\star := \arg \max_{\mu \in \mathcal{P}_{\text{train}}^{\text{RB}}} \psi_N(\mu)$.
 - 8: **end while**
-

Remark 3.1. (For Algorithm 1) For an efficient implementation, a proper error indicator $\psi_N(\cdot)$ needs to be computed. Many efforts have been devoted in this area in the last decades; see, e.g., [11, 18, 20, 34, 36, 39, 42, 46, 54]. It remains still an active topic in MOR. In Section 4, we will introduce a recently proposed error estimation from [55], which will be used as the error indicator in the POD-Greedy algorithm to construct the RB V for the SMB model.

Remark 3.2. (For Algorithm 1) **The AdSS technique:** The simulation of the SMB model requires a large number of time steps K in Step 4 of Algorithm 1, leading to a big matrix \bar{U} with many columns, which makes the SVD of \bar{U} expensive. To tackle this problem, the technique called adaptive snapshot selection (AdSS) in [6, 54] can be employed to keep the number of the snapshots and in turn the number of the columns of \bar{U} relatively small. In the meanwhile, almost all the “useful” information is still contained in the selected snapshots so that the cost of RB construction can be largely reduced without loss of accuracy. The idea of the AdSS technique is to filter out the linearly dependent information from the trajectory before performing the POD process in Step 4. The basic idea is as follows: assume that $u^n(\mu)$ is the last selected snapshot, there is no need to include $u^{n+i}(\mu)$ ($i = 1, 2, \dots$) as a new snapshot if it is “almost” linearly dependent on $u^n(\mu)$ (reflected by a suitable indicator); on the other hand, once the correlation between $u^{n+i}(\mu)$ and $u^n(\mu)$ is weak, then $u^{n+i}(\mu)$ is taken as a new snapshot. The indicator can be defined

as, e.g., the angle between the last selected snapshot and the tested vector, i.e., $\angle(u^n(\mu), u^{n+i}(\mu))$. By AdSS, the selected snapshots only consist of the most “representative” vectors of the solution trajectory, and the resulting \bar{U} has much smaller column size. For more details, please refer to [6, 54].

3.2. Complexity reduction: empirical interpolation method

As discussed previously, the EIM can be employed to tackle the nonlinear terms in (20) to reduce the complexity. The idea of the EIM or its variants is to construct an additional basis, say, $W \in \mathbb{R}^{N \times M}$, and approximate $g(Vu_r^n(\mu), \mu)$ by a linear combination of the basis, i.e., $\hat{g}^n(\mu) := W\beta^n(\mu) \approx g(Vu_r^n(\mu), \mu)$, where $\beta^n(\mu) := \beta(u_r^n(\mu), \mu) \in \mathbb{R}^M$ is the vector of coefficients. Then the system in (20) can be rewritten as

$$\hat{A}(\mu)u_r^{n+1}(\mu) = \hat{B}(\mu)u_r^n(\mu) + \hat{G}\beta^n(\mu), \quad (22)$$

where $\hat{G} = V^T W \in \mathbb{R}^{N \times M}$ can be precomputed once the bases W and V are obtained. The computation of $\beta^n(\mu)$ depends only on N and M and can be done cheaply in the online stage. System (22) is cheap to solve since $N, M \ll N$, and it is thus considered as the final ROM.

The EIM, introduced in [4], aims at constructing an affine (namely, parameter-separable) expression to approximate a (nonaffine) function $g(x, \mu)$ with sufficient regularity, i.e., $g(\cdot, \mu) \in L^\infty(\Omega)$, for all $\mu \in \mathcal{P}$. A parameter-independent basis $W := \{w_1(x), \dots, w_M(x)\}$, called collateral reduced basis (CRB), is precomputed based on snapshots of nonlinear function evaluations at a set of properly selected parameter samples. Then an affine approximation is defined by $\hat{g}(\mu) := \sum_{i=1}^M w_i(x)\beta_i(\mu)$, where the coefficients $\beta_i(\mu)$ can be determined by an enforced interpolation condition that $\hat{g}(\mu)$ interpolates $g(x, \mu)$ at a set of carefully selected empirical interpolation (EI) points $X_M := \{x_1^{\text{EI}}, \dots, x_M^{\text{EI}}\}$, i.e.,

$$\sum_{i=1}^M w_i(x_j^{\text{EI}})\beta_i(\mu) = g(x_j^{\text{EI}}, \mu), \quad j = 1, \dots, M. \quad (23)$$

In practical computation, the CRB and the EI points are usually computed in a finite dimensional discrete space, say, \mathbb{R}^N . Let $\mathcal{P}_{\text{train}}^{\text{CRB}} := \{\mu_j \mid \mu_j \in \mathcal{P}, j = 1, \dots, n_{\text{train}}\}$ be a training set with a finite number (n_{train}) of parameter samples, which is chosen as a surrogate of the parameter domain \mathcal{P} . Let $\mathcal{G} := \{g_1, \dots, g_{n_{\text{train}}}\}$ be the set of snapshots, where $g_j := [g(x_1, \mu_j), \dots, g(x_N, \mu_j)]^T \in \mathbb{R}^N$ is the vector of function evaluations of $g(x, \mu)$ at the parameter $\mu_j \in \mathcal{P}_{\text{train}}^{\text{CRB}}$

($j = 1, \dots, n_{\text{train}}$) on the spatial grid $\{x_1, x_2, \dots, x_N\}$. Note that in the discrete space, the basis W is actually a set of constant vectors, $w_1, \dots, w_M \in \mathbb{R}^N$, and the EI points X_M are corresponding to a few indices in the spatial grids, denoted as $\mathcal{I} := \{\varphi_1, \dots, \varphi_M\}$. To use matrix-vector notation, we introduce a vector $\beta(\mu) = [\beta_1(\mu), \dots, \beta_M(\mu)]^T \in \mathbb{R}^M$ and a matrix

$$S = [e_{\varphi_1}, \dots, e_{\varphi_M}],$$

where $e_{\varphi_i} = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^N$ is the φ_i th column of the identity matrix in $\mathbb{R}^{N \times N}$. For any $\mu \in \mathcal{P}$, let $g(\mu) = [g(x_1, \mu), \dots, g(x_N, \mu)]^T \in \mathbb{R}^N$, then the interpolation condition in (23) becomes

$$S^T W \beta(\mu) = S^T g(\mu).$$

Thus, $\beta(\mu) = (S^T W)^{-1} S^T g(\mu)$ and the interpolant defined by the EI basis W reads

$$\hat{g}(\mu) = W\beta(\mu) = W(S^T W)^{-1} S^T g(\mu), \quad \mu \in \mathcal{P}. \quad (24)$$

Note that $S^T W$ is invertible because it is actually a lower triangular matrix with unit diagonal [4, 17]. Given an error tolerance ε_{CRB} , the procedure of constructing the CRB and EI points can be summarized in Algorithm 2.

This interpolant approximation serves to reduce the complexity of the nonlinear parts of the model, which cannot be directly reduced by projection. It is crucial for an efficient offline-online computation for the RB method and other MOR methods. The EIM has been extended in recent works; see, e.g., [9, 10, 11, 16, 17, 30, 35, 40]. For time-dependent problems, we do not treat time as another parameter, but put the time trajectory for all training samples together as the snapshots to construct the CRB. That is, in Algorithm 2, the input snapshot set is redefined as $\mathcal{G} := \{g(u^k(\mu_j), \mu_j) \mid \mu_j \in \mathcal{P}_{\text{train}}^{\text{CRB}}, j = 1, \dots, n_{\text{train}}; k = 0, 1, \dots, K\}$. Note that when the total number (K) of the time steps for one FOM simulation is large, then the number of snapshots will be $(K+1) \cdot n_{\text{train}}$ if no further snapshot selection strategy is employed, and this number can be huge. This may render the computation of the CRB very expensive. Again, the adaptive snapshot selection [6, 54] can be employed to reduce the computational cost, similarly to the implementation for the RB construction; see Remark 3.2.

3.3. RB scheme for the SMB model

For reduced-order modeling of the SMB model in (14)–(15), to reduce the order (size), we construct a RB for each variable, i.e., compute the RB matrices

Algorithm 2 Generation of CRB and EI points (EIM)

Input: $\mathcal{G} := \{g_1, \dots, g_{n_{\text{train}}}\}, \varepsilon_{\text{CRB}} < 1$.

Output: CRB $W = [w_1, \dots, w_M]$ and EI points (indices) $\mathcal{I} := \{\varphi_1, \dots, \varphi_M\}$.

- 1: Initialization: $W = [], S = [], \mathcal{I} = \emptyset$.
- 2: $m = 1, \xi_m = \arg \max_{g_j \in \mathcal{G}} \|g_j\|, \varphi_m = \arg \max_{i \in \{1, \dots, N\}} |\xi_{m,i}|,$
 $w_m = \xi_m / \xi_{m, \varphi_m}, W = [W, w_m], S = [S, e_{\varphi_m}], \mathcal{I} = \mathcal{I} \cup \{\varphi_m\}.$
- 3: **while** $\|\xi_m\| > \varepsilon_{\text{CRB}}$ **do**
- 4: $m = m + 1.$
- 5: For all $g_j \in \mathcal{G}$, compute the interpolant $\hat{g}_j = W(S^T W)^{-1} S^T g_j$ by (24).
- 6: Define $g_m := \arg \max_{g_j \in \mathcal{G}} \|g_j - \hat{g}_j\|$, and the error
 $\xi_m := g_m - \hat{g}_m.$
- 7: **if** $\|\xi_m\| \leq \varepsilon_{\text{CRB}}$ **then**
- 8: Stop and set $M = m - 1.$
- 9: **else**
- 10: Determine the next EI point and basis:

$$\varphi_m = \arg \max_{i \in \{1, \dots, N\}} |\xi_{m,i}|, \quad w_m = \xi_m / \xi_{m, \varphi_m}.$$
- 11: Update $W = [W, w_m], S = [S, e_{\varphi_m}], \mathcal{I} = \mathcal{I} \cup \{\varphi_m\}.$
- 12: **end if**
- 13: **end while**

V_{c_z} and V_{q_z} for the variables c_z and q_z ($z = a, b$), respectively; to reduce the complexity, we construct the CRB matrix W_z for the nonlinear term h_z ($z = a, b$). Let $\hat{c}_z^n(\mu) := V_{c_z} c_{z,r}^n(\mu)$ and $\hat{q}_z^n(\mu) := V_{q_z} q_{z,r}^n(\mu)$ be the approximation to $c_z^n(\mu)$ and $q_z^n(\mu)$, respectively. Applying Galerkin projection, the ROM for the SMB model can be formulated as

$$\hat{A}_{c_z}(\mu) c_{z,r}^{n+1}(\mu) = \hat{B}_{c_z}(\mu) c_{z,r}^n(\mu) + \hat{R}_{c_z} \tilde{r}_z^n(\mu) - \frac{1 - \epsilon}{\epsilon} \Delta t \hat{H}_{c_z} \beta_z^n(\mu), \quad (25)$$

$$q_{z,r}^{n+1} = q_{z,r}^n + \Delta t \hat{H}_{q_z} \beta_z^n(\mu), \quad z = a, b, \quad (26)$$

where $c_{z,r}^n(\mu), q_{z,r}^n(\mu)$ are the reduced state vectors of the ROM, $\beta_z^n(\mu) := \beta(c_{z,r}^n(\mu), q_{z,r}^n(\mu); \mu)$ is the vector of coefficients in the interpolation of $h_z^n(\mu)$ (i.e., $\hat{h}_z^n(\mu) = W_z \beta_z^n(\mu)$), and $\hat{A}_{c_z}(\mu) = V_{c_z}^T A(\mu) V_{c_z}, \hat{B}_{c_z}(\mu) = V_{c_z}^T B(\mu) V_{c_z}, \hat{R}_{c_z} = V_{c_z}^T e_{\varphi_f}, \hat{H}_{c_z} = V_{c_z}^T W_z, \hat{H}_{q_z} = V_{q_z}^T W_z$ are the reduced matrices. Here, $e_{\varphi_f} = [0, \dots, 0, 1, 0, \dots, 0]^T \in \mathbb{R}^N$ is φ_f th column of the identity matrix in $\mathbb{R}^{N \times N}$ (φ_f corresponds to the location of the feed node of the SMB unit in the spatial grid), and $\tilde{r}_z^n(\mu)$ is actually a scalar parameter-dependent coefficient.

cient.

As discussed in Section 3.1, the RB scheme is usually realized by a strategy of offline-online decomposition. That is, the construction of the ROM and the simulation of the ROM can be completely decoupled into two stages. During the offline stage, all terms ($\hat{R}_{c_z}, \hat{H}_{c_z}, \hat{H}_{q_z}$, and the parameter-independent components, e.g., \hat{A}_j, \hat{B}_j in (21), in the affine expressions of $\hat{A}_{c_z}(\mu)$ and $\hat{B}_{c_z}(\mu)$) related to the high dimensional computation are precomputed and stored. This process can be expensive, but needs to be performed only once. During the online stage, given any feasible parameter, the reduced matrices ($\hat{A}_{c_z}(\mu), \hat{B}_{c_z}(\mu), \hat{R}_{c_z}, \hat{H}_{c_z}$, and \hat{H}_{q_z}) can be rapidly assembled using the precomputed data in the offline stage, and a small-sized ROM is solved. This online simulation is independent of the high dimension N , which implies that the offline cost can be paid off by many repeated ROM simulations under parameter variations. For example, in the optimization process, both objective and constraints can be cheaply computed by using the ROM without resorting to the FOM.

In addition, the ROM resulting from the RB method is expected to be reliable in the whole parameter domain. A reliable ROM is guaranteed by an efficient error estimator, telling us up to which stage in the POD-greedy algorithm, a ROM satisfying the accuracy requirement is obtained and the algorithm can be stopped. In the next section, we introduce an error estimation for the ROM, which is used as the error indicator ψ in the POD-greedy algorithm during the construction of the ROM. Once the ROM is constructed, it needs not be updated during the optimization process, unlike the POD based MOR method in [29].

4. Computing the error indicator $\psi(\mu)$

For nonlinear parameterized time-dependent problems with switching procedure, many existing error estimation mentioned earlier [11, 18, 20, 34, 36, 42, 46, 54] are not applicable, since some are only suitable for linear steady problems [34, 42, 46], others are valid only for linear time-dependent systems [18, 20]. While the error estimators from [11, 36, 54] are developed for nonlinear time-dependent problems, they may become inaccurate for systems with switching procedure. In our recent work [55], an efficient output error estimation is derived, and it has been successfully applied to a nonlinear batch chromatographic model and a linear SMB model. In this work, we will use the error estimation proposed in [55] to construct the ROM for the nonlinear SMB chromatography.

Recall that the underlying nonlinear SMB process has four outputs, as defined in (18). Since the error estimation proposed in [55] is derived for a single output, we now compute an error estimator for each output and take their maximum as the error indicator for the whole system. For this, we briefly summarize the main theory in [55] and then focus on the detailed implementation.

Given the time-dependent system in (19), assume that the output of interest is expressed as follows:

$$y^n(\mu) = Pu^n(\mu)$$

where $P^T \in \mathbb{R}^N$ is a constant vector and $u^n(\mu)$ is the solution to (19). Let $\hat{u}^n(\mu) := Vu_r^n(\mu)$ be the approximation to $u^n(\mu)$ and $\hat{y}^n(\mu) = P\hat{u}^n(\mu)$ be the corresponding approximate output. The output error estimation is summarized in the following.

Theorem 4.1. *For the systems (19) and (20), assume that $A(\mu)$ is invertible for all $\mu \in \mathcal{P}$. Then the output error $e_0^{n+1}(\mu) = y^{n+1}(\mu) - \hat{y}^{n+1}(\mu)$ at the time instance t^{n+1} satisfies*

$$\|e_0^{n+1}(\mu)\| \leq \Delta^{n+1}(\mu) := \rho \Phi(\mu) \|r_{\text{pr}}^{n+1}(\mu)\|, \quad (27)$$

where ρ is a computable constant, and

$$\begin{aligned} \Phi(\mu) &= \|(A(\mu))^{-1}\| \|r_{\text{du}}(\mu)\| + \|u_{\text{du},r}(\mu)\|, \\ r_{\text{pr}}^{n+1}(\mu) &= B(\mu)\hat{u}^n(\mu) + g(\hat{u}^n(\mu), \mu) - A(\mu)\hat{u}^{n+1}(\mu). \end{aligned}$$

Here,

$$r_{\text{du}}(\mu) = -P^T - (A(\mu))^T V_{\text{du}} u_{\text{du},r}(\mu) \quad (28)$$

is the residual of the dual system

$$(A(\mu))^T u_{\text{du}}(\mu) = -P^T, \quad (29)$$

and

$$V_{\text{du}}^T (A(\mu))^T V_{\text{du}} u_{\text{du},r}(\mu) = -V_{\text{du}}^T P^T \quad (30)$$

is the corresponding reduced dual system, where V_{du} is the RB for the dual system.

The detailed proof can be found in [55], where $A(\mu)$ and $B(\mu)$ can be time-dependent so that the error estimation is applicable to more general cases. We now focus on the computation of the error estimator $\Delta^{n+1}(\mu)$ in (27). It consists of three parts: $\|r_{\text{pr}}^{n+1}(\mu)\|$, $\Phi(\mu)$, and ρ . For $\|r_{\text{pr}}^{n+1}(\mu)\|$, one can compute it directly using its definition. For $\Phi(\mu)$, the key is to compute $\|u_{\text{du},r}(\mu)\|$ and $\|r_{\text{du}}(\mu)\|$. This needs to compute a RB V_{du} for the dual system in (29). We solve the system in (29) at all the parameters in the training set $\mathcal{P}_{\text{train}}^{\text{RB}} := \{\mu_j \mid \mu_j \in \mathcal{P}, j = 1, \dots, n_{\text{train}}^{\text{RB}}\}$ and collect the solutions as the snapshot matrix $U_{\text{du}} := [u_{\text{du}}(\mu_1), \dots, u_{\text{du}}(\mu_{n_{\text{train}}^{\text{RB}}})]$. The RB

V_{du} for the dual system can be obtained by, e.g., performing an SVD of U_{du} .

To compute the constant ρ , we introduce an auxiliary vector

$$\tilde{r}_{\text{pr}}^{n+1}(\mu) := A(\mu)u^{n+1}(\mu) - A(\mu)\hat{u}^{n+1}(\mu). \quad (31)$$

Then ρ can be estimated, e.g., using the average ratio between the norm of $\tilde{r}_{\text{pr}}^{n+1}(\mu)$ and that of $r_{\text{pr}}^{n+1}(\mu)$ at a specific parameter μ , i.e., $\rho = \frac{1}{K} \sum_{n=1}^K \frac{\|\tilde{r}_{\text{pr}}^{n+1}(\mu_\star)\|}{\|r_{\text{pr}}^{n+1}(\mu_\star)\|}$, where μ_\star is the chosen parameter used to enrich the RB in Algorithm 1, and K is the total number of time steps. Although the detailed solution $u^n(\mu_\star)$ is involved in the computation of $\|r_{\text{pr}}^{n+1}(\mu_\star)\|$, this does not contribute any extra cost, since it has been already computed in Step 3 of Algorithm 1.

As mentioned previously, four outputs are considered in the nonlinear SMB process, i.e.,

$$\begin{aligned} y^n(\mu) &= (c_{a,\text{CSS}}^{\text{E}}(t^n; \mu), c_{a,\text{CSS}}^{\text{R}}(t^n; \mu), \\ &\quad c_{b,\text{CSS}}^{\text{E}}(t^n; \mu), c_{b,\text{CSS}}^{\text{R}}(t^n; \mu)) \\ &=: (y_1^n(\mu), y_2^n(\mu), y_3^n(\mu), y_4^n(\mu)) \\ &=: (P_1 c_a^n(\mu), P_2 c_a^n(\mu), P_3 c_b^n(\mu), P_4 c_b^n(\mu)), \end{aligned} \quad (32)$$

where $P_j = [0, \dots, 0, 1, 0, \dots, 0] \in \mathbb{R}^N$, $j = 1, \dots, 4$, with nonzero entries corresponding to the extract or the raffinate node. Let $\hat{y}_j^n(\mu)$ be the approximation to the output $y_j^n(\mu)$. The task is to compute an estimator $\Delta_j^n(\mu)$ ($= \rho_j \Phi_j(\mu) \|r_{\text{pr},j}^{n+1}(\mu)\|$) for the true error $e_j^n(\mu) := y_j^n(\mu) - \hat{y}_j^n(\mu)$ at the time step n . The computations of $\|r_{\text{pr},j}^{n+1}(\mu)\|$, $\Phi_j(\mu)$, and ρ_j are straightforward, i.e., replacing P with P_j in (27)–(30), and the error estimator $\Delta_j^n(\mu)$ for each output $\hat{y}_j^n(\mu)$ can thus be obtained.

In this work, we use the average output error estimator over all time steps, $\bar{\Delta}_j(\mu) := \frac{1}{K} \sum_{n=1}^K \Delta_j^n(\mu)$, as the error indicator for the j th output, $j = 1, \dots, 4$. For a given parameter μ , the error indicator for the whole system is defined as

$$\psi(\mu) := \max_{j \in \{1, \dots, 4\}} \bar{\Delta}_j(\mu), \quad (33)$$

which is used in Algorithm 1 for the construction of the RB.

5. Implementation and Results

In this paper, we consider as a case study the binary separation of 1, 1'-bi-2-naphthol enantiomers on cellulose triacetate, where a mixture of 72/28 (v/v) heptane/isopropanol is used as eluent [3, 27, 32]. In this

process, the two feed concentrations are identical and fixed at 2.9 g/l. The maximal allowable internal flow rate Q_{\max} is assumed to be restricted due to pressure drop limitations at 1.0 ml/s. The model parameters are given in Table 1, and the constants for the isotherm function in (4) are given in Table 2.

Table 1: Model parameters, configuration considered and column properties.

Number of columns N_{col}	8
Column configuration	2-2-2-2
Column dimensions [cm]	2.6×10.5
Column porosity ϵ [-]	0.4
Péclet number Pe [-]	1000
Mass-transfer coefficients κ_z , $z = a, b$ [s^{-1}]	0.1, 0.1
Feed concentrations c_z^f , $z = a, b$ [g/l]	2.9, 2.9

Table 2: Coefficients of the adsorption isotherm equations.

H_{a1} [-]	2.69	H_{b1} [-]	3.73
H_{a2} [-]	0.1	H_{b2} [-]	0.3
K_{a1} [l/g]	0.0336	K_{b1} [l/g]	0.0446
K_{a2} [l/g]	1.0	K_{b2} [l/g]	3.0

It should be pointed out that a) the orders of magnitude of the used thermodynamic parameters are typical for many chromatographic separation problems and b) the general applicability of the SMB model used was already demonstrated in various case studies, e.g., [45]. We now determine a suitable order \mathcal{N} for the FOM in (14)–(15), based on which the ROM is constructed. As the true solution of the PDEs is not available, we take the solution from a very fine spatial grid ($\mathcal{N} = 3200$) as the reference solution. The runtime and the relative errors of the outputs (i.e., Pu_a, Pu_b) at a random parameter value using different numbers of cells for the discretization are plotted in Figure 2. It is shown that the runtime increases exponentially and the relative error of the outputs decrease as the order of the FOM increases. When $\mathcal{N} = 800$, i.e., one hundred cells are used for each column, the relative error of both products is less than 1% and the computational time is 290 seconds, which is a good trade-off between the accuracy and the computational cost. Thus, we use the system in (14)–(15) with $\mathcal{N} = 800$ as a faithful FOM to construct the ROM in the following.

In this section, we will show the details of the ROM

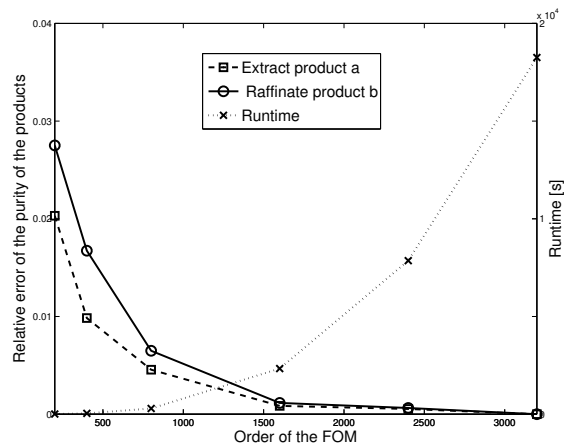


Figure 2: The behavior of the relative errors of the outputs and the runtime corresponding to different orders of the FOM

construction, the error behavior during the basis extension process, and the performance of the ROM in solving the optimization problem. Finally, we analyze the robustness of the optimal solution under flow rate uncertainty based on the resulting ROM. According to the experimental experiences and the triangle theory in [31], we chose the interesting parameter domain as $\mathcal{P} = \{\mu \mid \mu = [m_1, \dots, m_{IV}, Q_F] \in \mathcal{P}_a \subset \mathbb{R}^5\} := [4.30, 4.60] \times [2.40, 2.55] \times [3.05, 3.25] \times [2.10, 2.25] \times [0.1, 0.14]$, which is used as the admissible parameter domain for the ROM construction and the optimization. The tolerances for constructing the ROM are taken as $\varepsilon_{\text{CRB}} = 1.0 \times 10^{-5}$, $\varepsilon_{\text{ROM}} = 5.0 \times 10^{-3}$, and the tolerance for the CSS condition is $\varepsilon_{\text{CSS}} = 5.0 \times 10^{-4}$. All the computations were carried out using C++ code on a Linux machine with an Intel Core 2 Quad CPU 2.83 GHz and 4.00 GB RAM.

5.1. ROM construction

We now implement the RB method presented in Section 3 for the nonlinear SMB chromatography. For an efficient offline-online computation, we first take a training set $\mathcal{P}_{\text{train}}^{\text{CRB}}$ with 150 sample points randomly distributed in the parameter domain \mathcal{P} and use Algorithm 2 to construct the CRB for h_a and h_b , respectively. Then, we take a training set $\mathcal{P}_{\text{train}}^{\text{RB}}$ with 100 random samples and apply Algorithm 1 to compute the RB for each variable. Note that the number of samples in the training set $\mathcal{P}_{\text{train}}^{\text{RB}}$ is smaller than that in the training set $\mathcal{P}_{\text{train}}^{\text{CRB}}$, although the two sets can be taken as the same in principle. For the underlying problem, we found that a (relatively) less number of samples in $\mathcal{P}_{\text{train}}^{\text{RB}}$ is sufficient to

construct a globally accurate ROM. Therefore, we take less samples in $\mathcal{P}_{\text{train}}^{\text{RB}}$ than in $\mathcal{P}_{\text{train}}^{\text{CRB}}$. The error estimator, $\psi(\mu)$ in (33), is employed to guide the parameter sampling during the RB extension process. The corresponding true output error is defined as $e(\mu) := \frac{1}{K} \sum_{n=1}^K e^n(\mu)$, where $e^n(\mu) := \|y^n(\mu) - \hat{y}^n(\mu)\|_\infty$ is the true output error at time step n .

Figure 3 shows the behavior of the average output error estimation over all time steps and the corresponding true output error during the RB construction process. It is seen that the output error estimation bounds the true output error. Moreover, the estimation decays as the RB is enriched, and it goes below the tolerance ε_{ROM} when the number of basis vectors is up to 47. To further assess the reliability of the resulting ROM, we perform full and reduced simulations over a validation set \mathcal{P}_{val} , which consists of 200 random sample points in the parameter domain. The results are summarized in Table 3. It is seen that the maximal true error, $\max_{\mu \in \mathcal{P}_{\text{val}}} e(\mu)$, is 5.6×10^{-4} , which is below the pre-specified tolerance. This demonstrates that the resulting ROM is reliable in the whole parameter domain. Moreover, the average runtime for one FOM simulation is 287.4 s, while it is only 28.7 s for one ROM simulation. The average speedup factor (SpF) is 10.

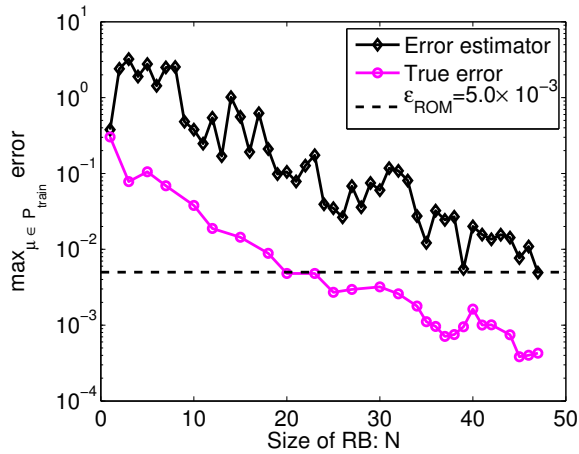


Figure 3: Error estimator decay during the RB extension process.

Table 3: Comparison of runtime for the full and reduced simulations over a validation set with 200 random sample points. $\varepsilon_{\text{ROM}} = 5.0 \times 10^{-3}$.

Model	Maximal error	Average runtime [s]/SpF
FOM	–	287.4 / –
ROM	5.6×10^{-4}	28.7 / 10

5.2. ROM-based optimization

As the ROM is precomputed in the offline stage, it is ready for online computations. When the ROM is employed for the optimization problem in (16), all the quantities computed through model simulations, e.g., the purity Pu_a and Pu_b , will be computed by solving the ROM in (25)–(26). Moreover, we do not modify the ROM during the online stage, since the ROM is reliable and accurate enough in the whole parameter domain.

To show the performance of the ROM, we implement both the FOM-based and the ROM-based optimization. The constraints are specified as $Q_{\text{max}} = 1.0$ ml/s, $Pu_{a,\text{min}} = 95.0\%$, $Pu_{b,\text{min}} = 95.0\%$. The optimization problems are solved by using the optimizer NLOPT_LN_COBYLA from the NLOpt library [24]. This is a local derivative-free optimizer. Different initial guesses may result in slightly different (locally) optimal solutions. Indeed, different initial guesses have been tested, and the differences between the optimal solution of ROM-based optimization and that of the FOM-based one are all sufficiently small. The initial guess used in Table 4 is $\mu_0 = [4.35, 2.42, 3.21, 2.21, 0.11]$. Let μ_k be the parameter chosen by the optimizer at the k th iteration. The iteration continues until the relative variance of the decision variables μ goes below a pre-specified tolerance ε_{opt} , i.e., when $\|\mu_{k+1} - \mu_k\| / \|\mu_k\| \leq \varepsilon_{\text{opt}}$ ($\varepsilon_{\text{opt}} = 1 \times 10^{-3}$ in Table 4). The results are summarized in Table 4.

It is seen that the FOM-based optimization takes 102 iterations to converge and the ROM-based one takes 93 iterations. The difference between the optimal solution to the ROM-based optimization and that of the FOM-based one is acceptable. On the other hand, the runtime for solving the optimization is significantly reduced by using the ROM. Solving the FOM-based optimization takes 8.124 hours, while solving the ROM-based one takes only 0.786 hours. The speedup factor is 10, which is a big progress compared to the recent work in [27]. There, the optimization of a nonlinear SMB model is accelerated using the POD-based ROMs, and the speedup factor is around 2. There are two reasons for the improvement: one is that the ROM in [27] is only locally reliable, while the proposed ROM is globally reliable in the parameter domain. As a result, the ROM in [27] has to be updated during the optimization process, which occupies much computational time. The other is that only the order of the FOM is reduced using the POD Galerkin projection method in [27], no reduction was done for the complexity of the nonlinear parts, which restricts the reduction rate by MOR. In contrast, we employed the RB method and the EIM to reduce both the order and the complexity of the FOM.

Table 4: Comparison of the optimization results based on the FOM and the ROM.

		Initial guess	FOM	ROM
Objective	Q_F [ml/s]	0.11	0.1218	0.1218
Optimal solution	m_I	4.35	4.4467	4.4733
	m_{II}	2.42	2.4936	2.4915
	m_{III}	3.21	3.1163	3.1175
	m_{IV}	2.21	2.1987	2.2000
	Q_F [ml/s]	0.11	0.122	0.122
Constraints	Pu_a	96.1%	95.0%	95.0%
	Pu_b	88.3%	95.0%	95.0%
	Q_I [ml/s]	0.6985	1.0	1.0
# Iterations			102	93
Runtime [h] / SpF			8.124 / -	0.786 / 10.3

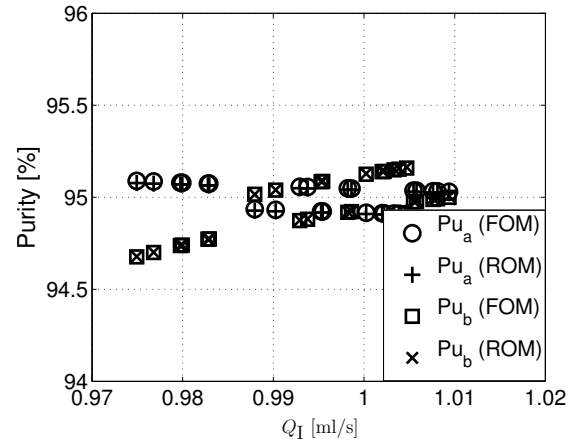
5.3. Uncertainty quantification (UQ)

Uncertainties in the parameters of adsorption isotherm equations, pump stability, extra-column volumes, and packing reproducibility, are inevitable in every SMB process [33]. There, the authors proposed an optimal design method for a linear SMB process under flow rate uncertainty. In this work, we use the standard UQ method: the Monte-Carlo method [21], to analyze the robustness of the product purity under flow rate uncertainty for a nonlinear SMB process.

To analyze the influence of the flow rate uncertainty in each zone upon the product purity, the flow rate in a certain zone is allowed to undertake a $\pm 2\%$ deviation while those in the other zones are fixed. Note that the mean value of the varying flow rate and the fixed value of the flow rate are the optimal solution obtained by the ROM-based optimization, which is $(Q_I, \dots, Q_{IV}) = (0.99, 0.6144, 0.7362, 0.5578)$ [ml/s]. The switching period is fixed at $t_s = 172$ s. For each case, 25 groups of random samples of the parameters are taken for the UQ based on the FOM and the ROM, respectively. More specifically, for the FOM-based UQ, we solve the FOM in (14)–(15) at each group of the sample to compute the output $y(t^n, \mu)$ and in turn the products purity in (17). For the ROM-based UQ, we solve the ROM at the same group of parameter samples to approximately compute the corresponding quantities.

The statistic quantities, e.g., the mean value $E[\cdot]$ and the standard deviation $\sigma[\cdot]$, are presented in Tables 5–8 for the varying flow rate in the four zones, respectively. It is observed that the purity of the component a is more sensitive to Q_{II} , reflected by the larger variance of $\sigma[Pu_a]$ in Table 6. The purity of the component b is more sensitive to Q_{III} , reflected by the larger variance of $\sigma[Pu_b]$ in Table 7. In contrast, Tables 5 and 8 show that the flow rates Q_I and Q_{IV} have less effect on the pu-

rity of both products, since the variances are all much smaller. There is also a clear physical interpretation for this fact. In the separation zones II and III the feed is split into two fractions. A proper selection of Q_{II} and Q_{III} is most crucial for the success of the overall process. Errors would cause that both components move to the same outlet port, which then never can be clean. The two outer zones are responsible for regenerating either the solid or the fluid phase. They also need to be correctly designed, however, this is relatively a simpler task. In addition, the statistic quantities (the mean and the standard derivation of the product purity) obtained from the ROM are almost the same as those obtained from the FOM. The runtime is significantly reduced using the ROM, and the SpF for all cases is around 10.


 Figure 4: Effect of Q_I on the purity of both products Pu_a and Pu_b using the FOM and the ROM, respectively.

Figures 4–7 show the profiles of the product purity corresponding to the varying flow rates in zones I, II,

Table 5: UQ for uncertainty of the flow rate Q_I .

Model	Order	CPU time [h]	$E[Pu_a]$ [%]	$E[Pu_b]$ [%]	$\sigma[Pu_a]$ [%]	$\sigma[Pu_b]$ [%]
FOM	800	2.04	95.00	94.96	0.07	0.15
ROM	47	0.20	94.99	94.96	0.07	0.15

Table 6: UQ for uncertainty of the flow rate Q_{II} .

Model	Order	CPU time [h]	$E[Pu_a]$ [%]	$E[Pu_b]$ [%]	$\sigma[Pu_a]$ [%]	$\sigma[Pu_b]$ [%]
FOM	800	2.00	94.80	94.96	0.65	0.54
ROM	47	0.20	94.80	94.96	0.65	0.54

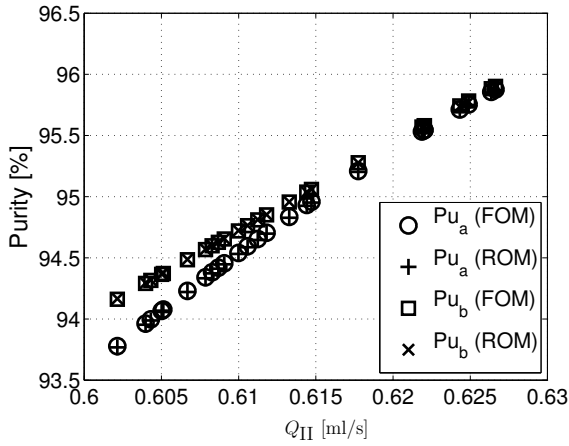


Figure 5: Effect of Q_{II} on the purity of both products Pu_a and Pu_b using the FOM and the ROM, respectively.

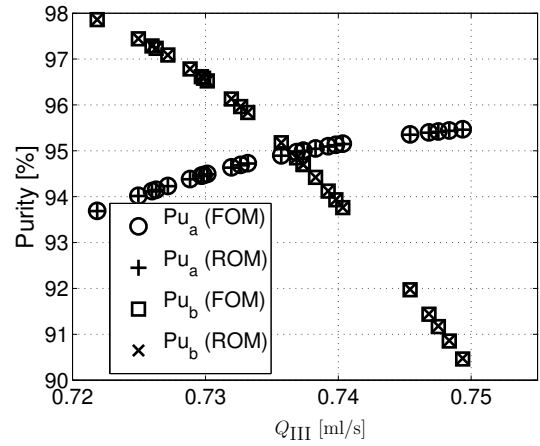


Figure 6: Effect of Q_{III} on the purity of both products Pu_a and Pu_b using the FOM and the ROM, respectively.

III, and IV, respectively. It is shown that the results computed by using the ROM have the same behavior as those of the FOM. Figure 4 shows that the purity of both products does not change monotonically as the flow rate in zone I increases (or decreases), unlike the behavior of a linear SMB model presented in [33], where the changes of the purity with respect to the flow rate are all monotonic. This reflects the complex nonlinear relation between the purity and the flow rate in the nonlinear SMB model. It is also noteworthy from Figure 6 that as the flow rate Q_{III} increases, the purity of the component b becomes higher while the purity of the component a becomes lower. Moreover, only one value of Q_{III} (provided that the other conditions are fixed) satisfies the purity requirement of both components, which is exactly the optimal solution.

In summary, the optimal solution is still reliable if the deviations of the flow rates in zones II and III are relatively small. From Figures 5 and 6, if the purity is allowed to deviate $\pm 0.5\%$ from the mean value 95.0%, the purified products are still acceptable as long as the deviations of the flow rates in zones II and III are less

than 1.0%. Nevertheless, the flow rates in zones I and IV have less influence upon the product purity, as can be seen from Figures 4 and 7. Through the UQ, it is further demonstrated that the parametric ROM is qualified for the many-query task.

6. Conclusions

We have explored using parametric ROMs to accelerate optimization and UQ of the nonlinear SMB chromatography. The parametric ROM is constructed using the RB method, and the nonlinear coupled terms are tackled by the EIM. The order and the complexity of the full-order model are both largely reduced, so the resulting ROM is fairly efficient and globally reliable in the entire parameter domain.

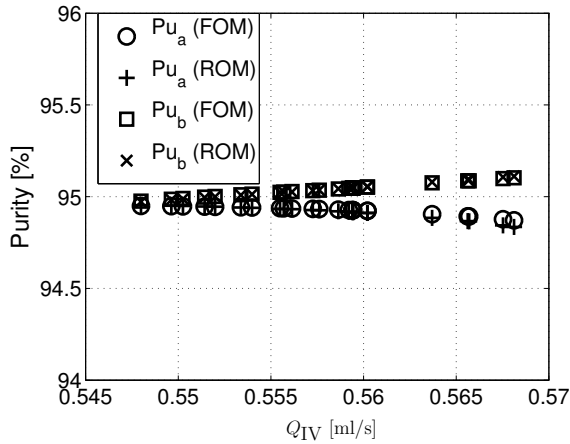
Using the ROM, the optimization problem considered is efficiently solved. The effect on the purity of the products is analyzed under flow rate uncertainty. It is shown that the optimal solution is robust in a wide range of flow rate ratios. The runtime of the UQ is significantly reduced by using the ROM.

Table 7: UQ for uncertainty of the flow rate Q_{III} .

Model	Order	CPU time [h]	$E[Pu_a]$ [%]	$E[Pu_b]$ [%]	$\sigma[Pu_a]$ [%]	$\sigma[Pu_b]$ [%]
FOM	800	1.94	94.78	94.92	0.49	2.20
ROM	47	0.19	94.78	94.92	0.49	2.20

Table 8: UQ for uncertainty of the flow rate Q_{IV} .

Model	Order	CPU time [h]	$E[Pu_a]$ [%]	$E[Pu_b]$ [%]	$\sigma[Pu_a]$ [%]	$\sigma[Pu_b]$ [%]
FOM	800	2.03	94.93	95.04	0.02	0.03
ROM	47	0.20	94.92	95.04	0.03	0.04

Figure 7: Effect of Q_{IV} on the purity of both products Pu_a and Pu_b using the FOM and the ROM, respectively.

It is worth noting that the SMB model considered is involved with nonlinearity, coupling, and periodic switching, which are challenging for MOR. We have shown that the output error estimation derived in [55] is applicable to this nontrivial model. It should also be highlighted that the ROM constructed by the parametric MOR method here produces much higher speedup than that using non-parametric MOR method in [27].

- [1] Antoulas, A. C., 2005. Approximation of Large-Scale Dynamical Systems. SIAM Publications, Philadelphia, PA.
- [2] Araújo, J. M. M., Rodrigues, R. C. R., Mota, J. P. B., 2006. Optimal design and operation of a certain class of asynchronous simulated moving bed processes. *J. Chromatogr. A* 1132 (1), 76–89.
- [3] Araújo, J. M. M., Rodrigues, R. C. R., Mota, J. P. B., 2006. Use of single-column models for efficient computation of the periodic state of a simulated moving-bed process. *Ind. Eng. Chem. Res.* 45 (15), 5314–5325.
- [4] Barrault, M., Maday, Y., Nguyen, N. C., Patera, A. T., 2004. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Math. Acad. Sci. Paris* 339 (9), 667–672.
- [5] Baur, U., Benner, P., Feng, L., 2014. Model order reduction for linear and nonlinear systems: a system-theoretic perspective. *Arch. Comput. Methods Eng.* 21 (4), 331–358.

- [6] Benner, P., Feng, L., Li, S., Zhang, Y., 2015. Reduced-order modeling and ROM-based optimization of batch chromatography. In: Abdulle, A., Deparis, S., Kressner, D., Nobile, F., Picasso, M. (Eds.), *Numerical Mathematics and Advanced Applications-ENUMATH 2013*. Vol. 103 of *Lect. Notes Comput. Sci. Eng.* Springer, Cham, Switzerland, pp. 427–435.
- [7] Benner, P., Gugercin, S., Willcox, K., 2015. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.* 57 (4), 483–531.
- [8] Benner, P., Mehrmann, V., Sorensen, D., 2005. *Dimension Reduction of Large-Scale Systems*. Vol. 45 of *Lect. Notes Comput. Sci. Eng.* Springer-Verlag, Berlin/Heidelberg, Germany.
- [9] Carlberg, K., Bou-Mosleh, C., Farhat, C., 2011. Efficient nonlinear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. *Internat. J. Numer. Methods Engrg.* 86 (2), 155–181.
- [10] Chaturantabut, S., Sorensen, D. C., 2010. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.* 32 (5), 2737–2764.
- [11] Drohmann, M., Haasdonk, B., Ohlberger, M., 2012. Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM J. Sci. Comput.* 34 (2), 937–969.
- [12] Dünnebier, G., Fricke, J., Klatt, K.-U., 2000. Optimal design and operation of simulated moving bed chromatographic reactors. *Ind. Eng. Chem. Res.* 39 (7), 2290–2304.
- [13] Dünnebier, G., Klatt, K.-U., 1999. Optimal operation of simulated moving bed chromatographic processes. *Comput. Chem. Eng.* 23, S195–S198.
- [14] Erdem, G., Abel, S., Morari, M., Mazzotti, M., Morbidelli, M., Lee, J. H., 2004. Automatic control of simulated moving beds. *Ind. Eng. Chem. Res.* 43 (2), 405–421.
- [15] Forrester, A. I., Keane, A. J., 2009. Recent advances in surrogate-based optimization. *Prog. Aerosp. Sci.* 45 (1), 50–79.
- [16] Grepl, M. A., 2012. Certified reduced basis methods for non-affine linear time-varying and nonlinear parabolic partial differential equations. *Math. Models Methods Appl. Sci.* 22 (03).
- [17] Grepl, M. A., Maday, Y., Nguyen, N. C., Patera, A. T., 2007. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *M2AN Math. Model. Numer. Anal.* 41 (3), 575–605.
- [18] Grepl, M. A., Patera, A. T., 2005. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *M2AN Math. Model. Numer. Anal.* 39 (01), 157–181.
- [19] Guiochon, G., Felinger, A., Shirazi, D. G., Katti, A. M., 2006. *Fundamentals of Preparative and Nonlinear Chromatography*. Academic Press.
- [20] Haasdonk, B., Ohlberger, M., 2008. Reduced basis method for finite volume approximations of parametrized linear evolution

- equations. *M2AN Math. Model. Numer. Anal.* 42 (2), 277–302.
- [21] Hammersley, J. M., Handscomb, D. C., 1964. *Monte Carlo Methods*. Methuen.
- [22] Hesthaven, J. S., Rozza, G., Stamm, B., 2016. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. SpringerBriefs in Mathematics. Springer.
- [23] Ito, K., Ravindran, S. S., 1998. A reduced-order method for simulation and control of fluid flows. *J. Comput. Phys.* 143 (2), 403–425.
- [24] Johnson, S. G., 2012. The NLOpt Nonlinear-Optimization Package. <http://ab-initio.mit.edu/nlopt>.
- [25] Kawajiri, Y., Biegler, L. T., 2006. Optimization strategies for simulated moving bed and PowerFeed processes. *AIChE J.* 52 (4), 1343–1350.
- [26] LeVeque, R. J., 2002. *Finite Volume Methods for Hyperbolic Problems*. Vol. 31. Cambridge University Press.
- [27] 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.
- [28] Li, S., Kawajiri, Y., Raisch, J., Seidel-Morgenstern, A., 2010. Optimization of simulated moving bed chromatography with fractionation and feedback: Part I. Fractionation of one outlet. *J. Chromatogr. A* 1217 (33), 5337–5348.
- [29] 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.
- [30] Maday, Y., Nguyen, N. C., Patera, A. T., Pau, G. S. H., 2009. A general multipurpose interpolation procedure: The magic points. *Commun. Pure Appl. Anal.* 8 (1), 383–404.
- [31] Mazzotti, M., Storti, G., Morbidelli, M., 1997. Optimal operation of simulated moving bed units for nonlinear chromatographic separations. *J. Chromatogr. A* 769 (1), 3–24.
- [32] Minceva, M., Pais, L. S., Rodrigues, A. E., 2003. Cyclic steady state of simulated moving bed processes for enantiomers separation. *Chem. Eng. Process.: Process Intensification* 42 (2), 93–104.
- [33] Mota, J. P., Araújo, J. M., Rodrigues, R., et al., 2007. Optimal design of simulated moving-bed processes under flow rate uncertainty. *AIChE J.* 53 (10), 2630–2642.
- [34] Nguyen, N. C., 2007. A posteriori error estimation and basis adaptivity for reduced-basis approximation of nonaffine-parametrized linear elliptic partial differential equations. *J. Comput. Phys.* 227 (2), 983–1006.
- [35] Nguyen, N. C., Patera, A. T., Peraire, J., 2008. A ‘best points’ interpolation method for efficient approximation of parametrized functions. *Internat. J. Numer. Methods Engrg.* 73 (4), 521–543.
- [36] Nguyen, N. C., Rozza, G., Patera, A. T., 2009. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers’ equation. *Calcolo* 46 (3), 157–185.
- [37] Nicoud, R.-M., 2014. The amazing ability of continuous chromatography to adapt to a moving environment. *Industrial & Engineering Chemistry Research* 53 (10), 3755–3765.
- [38] Noor, A. K., Peters, J. M., 1980. Reduced basis technique for nonlinear analysis of structures. *AIAA J.* 18 (4), 145–161.
- [39] Patera, A. T., Rozza, G., 2007. *Reduced Basis Approximation and a Posteriori Error Estimation for Parametrized Partial Differential Equations*. MIT, Cambridge, MA, also available from http://augustine.mit.edu/methodology/methodology_book.htm.
- [40] Peherstorfer, B., Butnaru, D., Willcox, K., Bungartz, H. J., 2014. Localized discrete empirical interpolation method. *SIAM J. Sci. Comput.* 36 (1), A168–A192.
- [41] Porsching, T. A., 1985. Estimation of the error in the reduced basis method solution of nonlinear equations. *Math. Comput.* 45 (172), 487–496.
- [42] Prud’homme, C., Rovas, D. V., Veroy, K., Machiels, L., Maday, Y., Patera, A. T., Turinici, G., 2002. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *J. Fluids Engrg.* 124, 70–80.
- [43] Quarteroni, A., Rozza, G. (Eds.), 2014. *Reduced Order Methods for Modeling and Computational Reduction*. Vol. 9 of Modeling, Simulation & Applications Series. Springer, Cham, Switzerland.
- [44] Rajendran, A., Paredes, G., Mazzotti, M., 2009. Simulated moving bed chromatography for the separation of enantiomers. *J. Chromatogr. A* 1216 (4), 709–738.
- [45] Rodrigues, A. E., Pereira, C., Minceva, M., Pais, L. S., Ribeiro, A. M., Ribeiro, A., Silva, M., Graca, N., Santos, J. C., 2015. Simulated moving bed technology: principles, design and process applications. Butterworth-Heinemann.
- [46] Rozza, G., Huynh, D. B. P., Manzoni, A., 2013. Reduced basis approximation and a posteriori error estimation for Stokes flows in parametrized geometries: roles of the inf-sup stability constants. *Numer. Math.* 125 (1), 115–152.
- [47] Rozza, G., Huynh, D. B. P., Patera, A. T., 2008. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch. Comput. Methods Eng.* 15 (3), 229–275.
- [48] Schilders, W. H. A., van der Vorst, H. A., Rommes, J., 2008. *Model Order Reduction: Theory, Research Aspects and Applications*. Springer-Verlag, Berlin, Heidelberg.
- [49] Seidel-Morgenstern, A., Kessler, L. C., Kaspereit, M., 2008. New developments in simulated moving bed chromatography. *Chem. Eng. Technol.* 31 (6), 826–837.
- [50] Toumi, A., Engell, S., Diehl, M., Bock, H., Schlöder, J., 2007. Efficient optimization of simulated moving bed processes. *Chem. Eng. Process.: Process Intensification* 46 (11), 1067–1084.
- [51] Toumi, A., Engell, S., Ludemann-Hombourger, O., Nicoud, R., Bailly, M., 2003. Optimization of simulated moving bed and Varicol processes. *J. Chromatogr. A* 1006 (1), 15–31.
- [52] Veroy, K., Prudhomme, C., Rovas, D. V., Patera, A. T., 2003. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In: *Proc. of the 16th AIAA computational fluid dynamics conference*. Vol. 3847. pp. 23–26.
- [53] Vilas, C., Vande Wouwer, A., 2011. Combination of multi-model predictive control and the wave theory for the control of simulated moving bed plants. *Chem. Eng. Sci.* 66 (4), 632–641.
- [54] Zhang, Y., Feng, L., Li, S., Benner, P., 2015. Accelerating PDE constrained optimization by the reduced basis method: application to batch chromatography. *Internat. J. Numer. Methods Engrg.* 104, 983–1007.
- [55] Zhang, Y., Feng, L., Li, S., Benner, P., 2015. An efficient output error estimation for model order reduction of parametrized evolution equations. *SIAM J. Sci. Comput.* 37 (6), B910–B936.