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

Yongjin Zhang, Lihong Feng*, Andreas Seidel-Morgenstern, Peter Benner

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 op-
timization process in [27].

In this paper, we present a parametric model order re-
duction (PMOR) method for the nonlinear SMB chro-
matography, 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 re-
sulting ROM is globally reliable, that is, it is able to re-
produce the main dynamics of the FOM corresponding 
to any variation of the parameter in the entire parame-

ter domain. A comprehensive survey of PMOR meth-
ods 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 pa-

terized partial differential equations (PDEs). Using 
the RB method, the ROM construction and simu-
lation using the ROM can be entirely decoupled into 
two stages. That is, during the offline stage, all the 
high-dimension dependent quantities can be precom-
puted. 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 re-
sorting to the FOM. This is quite suitable for the afore-
mentioned 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 estima-
tor) 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 er-
ror 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], uncer-
tainties in the parameters of adsorption isotherm equa-
tions, pump stability, extra-column volumes, and pack-
ing 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 ad-

dresses the SMB model, the corresponding discrete sys-
tem, 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 liter-
ature. 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 identi-

cal 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 separa-
tion process. The mixture to be separated and the elu-

ten are fed continuously through the two inlets, and the 
two purified components are withdrawn also continu-
ously from the two outlets, respectively. The separation 
regime is accomplished through simulating a counter 
current movement of the liquid and solid phases by syn-
chronously shifting the inlet and outlet ports one col-
umn 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 chro-

matographic column can be described by an axially dis-

persed plug flow model with a limited mass-transfer rate 
quantified by a linear driving force (LDF) approxima-
tion. The mass balance in the column $k (k = 1, \ldots, N_{col})$
can be given by, [19],
\[
\frac{\partial c_{z,k}}{\partial t} + \frac{1 - \epsilon}{\epsilon} \frac{\partial q_{z,k}}{\partial t} = \frac{Q_k t_k}{\epsilon A_k L} \left( \frac{\partial c_{z,k}}{\partial x} - \frac{1}{Pe} \frac{\partial^2 c_{z,k}}{\partial x^2} \right) \hspace{1cm} 0 < x < 1, \tag{1}
\]
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, \(A_k\) the cross-sectional area of the column, \(L\) the column length, \(\epsilon\) the column porosity, \(t_k\) the switching period, and \(Pe\) the Péclet number. Note that \(t\) and \(x\) are in the dimensionless form, i.e., \(t = t/\bar{t}, x = \tilde{x}/L\) (\(\bar{t}\) 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:
\[
\frac{\partial q_{z,k}}{\partial t} = \kappa_z (q_{z,k}^{\text{eq}} - c_{z,k}), \hspace{1cm} 0 \leq x \leq 1, \tag{2}
\]
where \(\kappa_z\) is the mass-transfer coefficient and \(q_{z,k}^{\text{eq}}\) is the adsorption equilibrium defined by the adsorption isotherm function
\[
q_{z,k}^{\text{eq}} = f_z(c_{a,k}, c_{b,k}). \tag{3}
\]

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],
\[
q_{z,k}^{\text{eq}} = f_z(c_{a,k}, c_{b,k}) = \frac{H_{1z} c_{z,k}}{1 + K_{a1} c_{a,k} + K_{b1} c_{b,k}} + \frac{H_{2z} c_{z,k}}{1 + K_{a2} c_{a,k} + K_{b2} c_{b,k}}, \tag{4}
\]
where \(H_{jz}\) and \(K_{jz}\) \(j = 1, 2\) are the Henry constants and thermodynamic coefficients, respectively.

Danckwerts-type boundary conditions are imposed to equation (1), i.e.,
\[
\begin{align*}
\frac{\partial c_{z,k}}{\partial x}_{x=0} &= Pe (c_{z,k}(t, 0) - c_{z,k}^\in(t)), \\
\frac{\partial c_{z,k}}{\partial x}_{x=1} &= 0,
\end{align*} \tag{5}
\]
where \(c_{z,k}^\in(t)\) 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,
\[
\begin{align*}
c_{z,k}(0, x) &= c_{z,k}^0, \hspace{1cm} 0 \leq x \leq 1, \hspace{1cm} k = 1, \ldots, N_{\text{col}}, \\
q_{z,k}(0, x) &= q_{z,k}^0,
\end{align*}
\]
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.,
\[
c_{z,k+1}(t) = c_{z,k}(t, 1), \tag{6}
\]
ecept for the feed and desorbent nodes. The mass balances at the inlet and outlet ports can be established as follows:

Desorbent node:
\[
Q_I = Q_{IV} + Q_D, \\
Q_{I} c_{z,k}^{\text{in}}(t) = Q_{IV} c_{z,N_{\text{col}}}(t, 1). \tag{7}
\]

Extract node:
\[
Q_{II} = Q_I - Q_E, \\
c_{z,k+1}(t) = c_{z,k}(t, 1). \tag{8}
\]

Feed node:
\[
Q_{III} = Q_{II} + Q_F, \\
Q_{III} c_{z,k+1}^{\text{in}}(t) = Q_{IV} c_{z,k+1}^{\text{in}}(t, 1) + c_{z,k}^F Q_F. \tag{9}
\]

Raffinate node:
\[
Q_{IV} = Q_{III} - Q_R, \\
c_{z,k+1}^{\text{in}}(t) = c_{z,k+1}^{\text{in}}(t, 1). \tag{10}
\]

Here, \(k_j\) is the number of columns in the zone \(J\), \(Q_J\) is the flow rate in the corresponding zone, \(J \in \{I, II, III, IV\}\), \(c_{z,k}^F\) 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 \(Q_k\) \((k = 1, \ldots, 8)\) in the columns satisfy the following relations:
\[
Q_1 = Q_2 = Q_3, \hspace{1cm} Q_4 = Q_5 = Q_6, \hspace{1cm} Q_7 = Q_8 = Q_{IV}. \tag{11}
\]

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.
Let Δt = 1/K, τ = nΔt, n ∈ ℕ := {0, 1, ..., K} be the K + 1 time instants over the time interval [0, 1]. Δx = 1/N be the spatial grid size, and c_{i,k}^n(μ), q_{i,k}^n(μ) ∈ ℝ^N are the numerical approximations of the concentrations (c_{i,k} and q_{i,k}, respectively) in the kth column at time instance t = τ. Note that the time interval [0, 1] is actually a dimensionless switching period. The fully discretized finite volume formulation for the kth chromatographic column can be written as

\[ A_k(μ)c_{i,k}^{n+1}(μ) = B_k(μ)c_{i,k}^n(μ) + r_{i,k}(c_{i,k}^n(τ^n), μ), \]
\[ q_{i,k}^{n+1}(μ) = q_{i,k}^n(μ) + Δh_k^c(μ), \]

where \( A_k(μ), B_k(μ) \in ℝ^{N×N} \) are tridiagonal matrices, \( h_k^c(μ) := f_z(c_{i,k}^n(μ), c_{i,k}^n(μ)) - q_{i,k}^n(μ) \in ℝ^N \) is a nonlinear vector-valued function, and \( r_{i,k}(c_{i,k}^n(τ^n), μ) = r_{i,k}(c_{i,k}^n(τ^n), μ)[1, 0, \ldots, 0]^T \in ℝ^N \) with

\[ r_{i,k}(c_{i,k}^n(τ^n), μ) := \left( 1 + \frac{Pe}{2N} \right) \frac{NQ_k(μ)h_{s}}{eL(A_+)c_{i,k}^n(τ^n)Δt} c_{i,k}^n(τ^n)Δt \in ℝ. \]

The parameter μ characterizes the operating conditions of the underlying SMB process (e.g., μ = [Q_k, ..., Q_N, τ_k]).

It is noteworthy that \( r_{i,k}(c_{i,k}^n(τ^n), μ) \) in (11) depends on the information from the connected columns, because the inflow of the kth column \( c_{i,k}^n(τ^n) \) is determined by the outflow of the (k – 1)st column, \( k = 2, \ldots, N_\text{col}, \) and \( c_{i,k}^n(τ^n) \) is determined by \( c_{i,k}^n(τ^1, 1) \)—the concentration at the outlet of the \( N_\text{col} \)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:

\[ A(μ)c_{i}^{n+1}(μ) = B(μ)c_{i}^{n}(μ) + r_{i}(c_{i}^{n}(τ^n), μ) - \frac{1}{ε} Δh_i^c(μ), \]
\[ q_{i}^{n+1}(μ) = q_{i}^{n}(μ) + Δh_i^c(μ), \]

where

\[ c_{i}^{n}(μ) = \begin{bmatrix} c_{i,1}^{n}(μ) \\ \vdots \\ c_{i,\text{N}_\text{col}}^{n}(μ) \end{bmatrix}, \quad q_{i}^{n}(μ) = \begin{bmatrix} q_{i,1}^{n}(μ) \\ \vdots \\ q_{i,\text{N}_\text{col}}^{n}(μ) \end{bmatrix} \in ℝ^N, \]
\[ h_i^c(μ) = \begin{bmatrix} h_{1,1}^c(μ) \\ \vdots \\ h_{\text{N}_\text{col},\text{N}_\text{col}}^c(μ) \end{bmatrix} \in ℝ^N, \]
\[ A(μ) = \text{diag}(A_1(μ), ..., A_{\text{N}_\text{col}}(μ)) \in ℝ^{\text{N}_\text{col}×\text{N}_\text{col}}, \quad B(μ) = \text{diag}(B_1(μ), ..., B_{\text{N}_\text{col}}(μ)) \in ℝ^{\text{N}_\text{col}×\text{N}_\text{col}} \]
with a block diagonal matrix \( B(μ) \) resulting from separating an auxiliary vector

\[ r_{i}^{n}(c_{i}^{n}(τ^n), μ) := \begin{bmatrix} r_{1}^{n}(c_{i,1}^{n}(τ^n), μ) \\ \vdots \\ r_{\text{N}_\text{col}}^{n}(c_{i,\text{N}_\text{col}}^{n}(τ^n), μ) \end{bmatrix} = B_i(μ)c_{i}^{n}(μ) + r_{i}^{n}(μ) \]

into two parts depending on the definition of \( r_{i,k}(c_{i,k}^{n}(τ^n), μ) \) in (13), \( B_i(μ) \in ℝ^{\text{N}_\text{col}×\text{N}_\text{col}}, r_i^{n}(μ) \in ℝ^N, \) and \( \text{N}_\text{col} = \text{N} \cdot \text{N}_\text{col}. \) The first part, \( B_i(μ)c_{i}^{n}(μ) \), linearly depends on \( c_{i}^{n}(μ) \) and is determined by (6)–(10). The second part, \( r_i^{n}(μ) \), 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 \( (c_{i}^{n}(μ), q_{i}^{n}(μ)) \) undergoes a shift, and the shifted vectors \( (P_{i}c_{i}^{n}(μ), P_{i}q_{i}^{n}(μ)) \) are used as the new initial state to continue the evolution process until the CSS conditions are satisfied. Here \( P_{i} \) 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.,
\[
\max_{\varepsilon \in [a, b]} \max \{||c^0_e(\mu) - P_e c^E_e(\mu)||, ||q^0_e(\mu) - P_e q^E_e(\mu)||\} < \epsilon_{CSS},
\]
where \(\epsilon_{CSS}\) 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 \(m_j\), \(J = 1, \ldots, IV\), which represent the ratios of the fluid flow rates over the solid flow rate. For notational simplicity, let \(\mu := [m, \ldots, mV, Q_f]\). 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, \(Q_f\), i.e.,
\[
m_j = \frac{t_e Q_f - \epsilon V_{col}}{(1 - \epsilon)V_{col}}, \quad J = 1, \ldots, IV.
\]

Here, \(V_{col}\) is the volume of a chromatographic column. Let \(P \subset \mathbb{R}^3\) be an admissible parameter domain, the optimization problem can be formulated as follows:
\[
\begin{align*}
\min_{\mu \in P} f(\mu) &= -Q_f, \\
\text{s.t.} & \quad Pu_{a_{min}} - Pu_{a}(\mu) \leq 0, \\
& \quad Pu_{b_{min}} - Pu_{b}(\mu) \leq 0, \\
& \quad Q_f - Q_{max} \leq 0,
\end{align*}
\]
where
\[
Pu_{a}(\mu) := \frac{\int_0^1 c^E_{e,CSS}(t, \mu) \, dt}{\int_0^1 c^E_{a,CSS}(t, \mu) \, dt + \int_0^1 c^E_{b,CSS}(t, \mu) \, dt},
\]
\[
Pu_{b}(\mu) := \frac{\int_0^1 c^R_{e,CSS}(t, \mu) \, dt}{\int_0^1 c^R_{a,CSS}(t, \mu) \, dt + \int_0^1 c^R_{b,CSS}(t, \mu) \, dt},
\]
are the products’ purity at the extract and the raffinate outlets, \(c^E_{e,CSS}(t, \mu)\) and \(c^R_{e,CSS}(t, \mu)\) are the CSS concentrations of \(\epsilon\) 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
\[
y(t, \mu) := (c^E_{e,CSS}(t; \mu), c^R_{e,CSS}(t; \mu), c^E_{b,CSS}(t; \mu), c^R_{b,CSS}(t; \mu)).
\]

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 \(c^E_{\epsilon}(\mu), q^E_{\epsilon}(\mu), z = a, b\), respectively. For this, we rewrite each equation in the system in a general form:
\[
A(\mu)u^{\mu+1}(\mu) = B(\mu)u^{\mu}(\mu) + g(u^{\mu}(\mu), \mu),
\]
where \(u^{\mu}(\mu)\) stands for the unknown vector \(c^E_{\epsilon}(\mu)\) in (14) (resp., \(q^E_{\epsilon}(\mu)\) in (15)), \(A(\mu), B(\mu) \in \mathbb{R}^{N \times N}\) are coefficient matrices (for (15), they are identity matrices), \(g(., \mu)\) stands for the nonlinear terms, e.g., \(g(., \mu) = c^E_{\epsilon}(\mu) - \frac{1}{\epsilon} \Delta t h^2_{\epsilon}(\mu)\) in (14) and \(g(., \mu) = \frac{1}{\epsilon} \Delta t h^2_{\epsilon}(\mu)\) 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 \(u^\mu(\mu)\) resides in a low-dimensional subspace \(V\) 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 \(V\). A ROM can be obtained by Galerkin projection as follows:
\[
\hat{A}(\mu)u^{\mu+1}(\mu) = \hat{B}(\mu)u^{\mu}(\mu) + V^T g(Vu^{\mu}(\mu), \mu),
\]
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 \(u^{\mu}(\mu) \in \mathbb{R}^N\) 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 \( \mu \), 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

\[
\hat{A}(\mu) = V^T A(\mu) V = \sum_{j=1}^{n_a} \xi_j(\mu)\hat{A}_j, \quad \hat{B}(\mu) = V^T B(\mu) V = \sum_{k=1}^{n_b} \zeta_k(\mu)\hat{B}_k,
\]

where \( \hat{A}_j = V^T A_j V \) and \( \hat{B}_k = V^T B_k V \), \( j = 1, \ldots, n_a, k = 1, \ldots, n_b \). 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 \( \mu \) are independent of the dimension \( N \). However, evaluation of the last term in (20), \( V^T g(V\mu)(\mu, \mu) \), is still \( N \)-dependent, due to the nonlinearity of \( g \) and/or its non-affine dependence on the parameter \( \mu \). At every time step, one must compute a long vector \( g(V\mu)(\mu, \mu) \in \mathbb{R}^N \), 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 \( \mathcal{P}^{\text{RB}}_{\text{train}} \) 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 \( \mu^* \) that causes the largest error measured by a proper error indicator \( \psi(\cdot) \) is chosen from \( \mathcal{P}^{\text{RB}}_{\text{train}} \) to enrich the basis. The iteration continues until the error \( \psi(\mu^*) \) goes below the required accuracy \( \epsilon_{\text{ROM}} \). 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 \( \Sigma \) is a rectangular diagonal matrix with non-negative real numbers on the diagonal.

**Algorithm 1** POD-Greedy algorithm for the RB construction

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

**Output:** \( \text{RB } V = [v_1, \ldots, v_N] \).

1. Initialization: \( N = 0, \mu_\star = \mu_0, \eta_N(\mu_\star) = 1, V = [\cdot] \).

2. while \( \psi_N(\mu_\star) > \epsilon_{\text{ROM}} \) do

3. Simulate the FOM at \( \mu_\star \) to get the solution trajectory: \( \{u^i(\mu_\star)\}_{i=0}^K \).

4. Perform POD process:

   Compute \( \bar{U} := [\bar{u}_0^0, \ldots, \bar{u}_K^K] \), \( \bar{u}_i^a = u^a(\mu_\star) - Vv_i^\top \bar{u}^a(\mu_\star) \) (if \( N = 0 \)), \( v_i = 0, \ldots, K \).

   Implement SVD of \( \bar{U} \): \( \bar{U} = Q\Sigma Y^\top \).

   Take the first column of \( Q \): \( v_{N+1} = \bar{Q}(;1) \).

5. Enrich the RB \( V := [V, v_{N+1}] \).

6. Update \( N = N + 1 \).

7. Find \( \mu_\star := \arg \max_{\mu \in \mathcal{P}} \psi(\mu) \).

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 \( u^a(\mu) \) is the last selected snapshot, there is no need to include \( u^a(t_i)(\mu) \) (\( i = 1, 2, \ldots \)) as a new snapshot if it is “almost” linearly dependent on \( u^a(\mu) \) (reflected by a suitable indicator); on the other hand, once the correlation between \( u^a(t_i)(\mu) \) and \( u^a(\mu) \) is weak, then \( u^a(t_i)(\mu) \) 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., \( \angle(u^t(\mu), w^{\nu+1}(\mu)) \). By AdSS, the selected snapshots only consist of the most “representative” vectors of the solution trajectory, and the resulting \( \hat{U} \) 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 complexity 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 \( g(Vu^{\ell}(\mu), \mu) \) by a linear combination of the basis, i.e., \( \hat{g}(\mu) := W\beta(\mu) = g(Vu^{\ell}(\mu), \mu) \), where \( \beta(\mu) = \beta(u^{\ell}(\mu), \mu) \in \mathbb{R}^M \) is the vector of coefficients. Then the system in (20) can be rewritten as

\[
\hat{A}(\mu)u^{\ell+1}(\mu) = \hat{B}(\mu)u^{\ell}(\mu) + \hat{G}\beta(\mu),
\]

where \( \hat{G} = V^TW \in \mathbb{R}^{N \times M} \) can be precomputed once the bases \( W \) and \( V \) are obtained. The computation of \( \beta(\mu) \) 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 \( W := \{w_1(x), \ldots, w_M(x)\} \), 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 \( X_\mu := \{x_1^\mu, \ldots, x_M^\mu\} \), i.e.,

\[
\sum_{i=1}^M w_i(x_j^\mu)\beta_i(\mu) = g(x_j^\mu, \mu), \quad j = 1, \ldots, M.
\]  

In practical computation, the CRB and the EI points are usually computed in a finite dimensional discrete space, say, \( \mathbb{R}^N \). Let \( g_{train}^{\text{CRB}} := \{\mu_j \mid \mu_j \in \mathcal{P}, j = 1, \ldots, n_{\text{train}}\} \) be a training set with a finite number \( n_{\text{train}} \) of parameter samples, which is chosen as a surrogate of the parameter domain \( \mathcal{P} \). Let \( \mathcal{G} := \{g_1, \ldots, g_{\text{max}}\} \) be the set of snapshots, where \( g_j := [g(x_1, \mu_j), \ldots, g(x_N, \mu_j)]^T \in \mathbb{R}^N \) is the vector of function evaluations of \( g(x, \mu) \) at the parameter \( \mu_j \in g_{\text{train}}^{\text{CRB}} \) (\( j = 1, \ldots, n_{\text{train}} \)) on the spatial grid \( \{x_1, x_2, \ldots, x_N\} \). Note that in the discrete space, the basis \( W \) is actually a set of constant vectors, \( w_1, \ldots, w_M \in \mathbb{R}^N \), and the EI points \( X_\mu \) are corresponding to a few indices in the spatial grids, denoted as \( I := \{\nu_1, \ldots, \nu_M\} \). To use matrix-vector notation, we introduce a vector \( \beta(\mu) = [\beta_1(\mu), \ldots, \beta_M(\mu)]^T \in \mathbb{R}^M \) and a matrix

\[
S = [e_{\nu_1}, \ldots, e_{\nu_M}],
\]

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

\[
S^TW\beta(\mu) = S^Tg(\mu).
\]

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

\[
\hat{g}(\mu) = W\beta(\mu) = W(S^TW)^{-1}S^Tg(\mu), \quad \mu \in \mathcal{P}.
\]  

Note that \( S^TW \) is invertible because it is actually a lower triangular matrix with unit diagonal [4, 17]. Given an error tolerance \( \varepsilon_{\text{CRB}} \), 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 \( \mathcal{G} := \{g(\mu_j), \mu_j \mid \mu_j \in g_{\text{train}}^{\text{CRB}}, j = 1, \ldots, n_{\text{train}}\} \). 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: \( \mathcal{G} := \{g_1, \ldots, g_{n_m}\}, \varepsilon_{CRB} < 1 \).
Output: CRB \( W = \{w_1, \ldots, w_M\} \) and EI points (indices) \( I := \{\varphi_1, \ldots, \varphi_M\} \).

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

For nonlinear parameterized time-dependent problems with switching procedure, many existing error estimation schemes 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:

\[ y^n(\mu) = P^T u^n(\mu) \]

where \( P^T \in \mathbb{R}^N \) is a constant vector and \( u^n(\mu) \) is the solution to (19). Let \( \tilde{u}^n(\mu) = V u^n(\mu) \) be the approximation to \( u^n(\mu) \) and \( \tilde{y}^n(\mu) = P \tilde{u}^n(\mu) \) 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 \( e_O^n(\mu) = y^n(\mu) - \tilde{y}^n(\mu) \) at the time instance \( t^n \) satisfies

\[ ||e_O^n(\mu)|| \leq \Delta^n(\mu) := \rho \Phi(\mu)||r^n(\mu)||, \quad (27) \]

where \( \rho \) is a computable constant, and

\[ \Phi(\mu) = ||(A(\mu))^{-1}|| ||r_{du}(\mu)|| + ||u_{du}(\mu)||. \]

\[ r_{du}(\mu) = -P^T - (A(\mu))^T V_{du} u_{du}(\mu) \]

is the residual of the dual system

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

and

\[ V_{du}^T (A(\mu))^T V_{du} u_{du}(\mu) = -V_{du}^T P^T \]

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 \( (A(\mu))^{-1} \) 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 \( \Delta^n(\mu) \) in (27). It consists of three parts: \( ||r^n(\mu)|| \), \( \Phi(\mu) \), and \( \rho \). For \( ||r^n(\mu)|| \) one can compute it directly using its definition. For \( \Phi(\mu) \), the key is to compute \( ||u_{du}(\mu)|| \) and \( ||r_{du}(\mu)|| \). 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 \( \mathcal{P}_{\text{train}}^{R^B} := \{ \mu_j | \mu_j \in \mathcal{P}, j = 1, \ldots, |\mathcal{P}_{\text{train}}^{R^B}| \} \) and collect the solutions as the snapshot matrix \( U_{\text{du}} := [u_{du}(\mu_1), \ldots, u_{du}(\mu_{|\mathcal{P}_{\text{train}}^{R^B}|})]. \) The RB \( V_{du} \) for the dual system can be obtained by, e.g., performing an SVD of \( U_{du} \).

To compute the constant \( \rho \), we introduce an auxiliary vector

\[ \rho^n(\mu) := A(\mu) u^n(\mu) - A(\mu) \tilde{u}^n(\mu). \]

Then \( \rho \) can be estimated, e.g., using the average ratio between the norm of \( \rho^n(\mu) \) and that of \( r^n(\mu) \) at a specific parameter \( \mu \), i.e.,

\[ \rho = \frac{1}{K} \sum_{j=1}^{K} ||\rho^n(\mu_j)|| \] where \( \mu_j \) 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 \( u^n(\mu_j) \) is involved in the computation of \( ||\rho^n(\mu_j)|| \), 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.,

\[ y^n(\mu) = (c_{1,\text{CSS}}(\rho^n; \mu), c_{2,\text{CSS}}(\rho^n; \mu), c_{3,\text{CSS}}(\rho^n; \mu), c_{4,\text{CSS}}(\rho^n; \mu)) \]

\[ := (y_1^n(\mu), y_2^n(\mu), y_3^n(\mu), y_4^n(\mu)) \]

\[ = (P_1 c_{1,\text{CSS}}(\mu), P_2 c_{2,\text{CSS}}(\mu), P_3 c_{3,\text{CSS}}(\mu), P_4 c_{4,\text{CSS}}(\mu)), \]

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

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

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

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 \( Q_{\text{max}} \) 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 \( N_{\text{col}} \) | 8 |
| Column configuration | 2-2-2 |
| Column dimensions [cm] | 2.6 \( \times \) 10.5 |
| Column porosity \( \epsilon \) [-] | 0.4 |
| Péclet number \( Pe \) [-] | 1000 |
| Mass-transfer coefficients \( \kappa_z \), \( z = a, b \) [s \(^{-1}\)] | 0.1, 0.1 |
| Feed concentrations \( c_z \), \( z = a, b \) [g/l] | 2.9, 2.9 |

**Table 2: Coefficients of the adsorption isotherm equations.**

| \( H_{d1} \) [-] | 2.69 | \( H_{b1} \) [-] | 3.73 |
| \( H_{d2} \) [-] | 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., \( P_{\text{ua}}, P_{\text{uh}} \)) 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 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 \( \mathcal{P} = [\mu \mid \mu = [m_1, \ldots, m_{15}, Q]\in \mathcal{P}_0 \subset \mathbb{R}^5] := [4.30, 4.60] \times [2.40, 2.55] \times [3.05, 3.25] \times [2.10, 2.25] \times [0.1, 0.14] \), which is used as the admissible parameter domain for the ROM construction and the optimization.

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

### 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 \( h_0 \) and \( h_0 \), 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 \( \epsilon(\mu) := \frac{1}{N} \sum_{n=1}^{N} \epsilon_n(\mu) \), where \( \epsilon_n(\mu) := \|y_n(\mu) - \hat{y}_n(\mu)\|_\infty \) 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 \( \varepsilon_{\text{ROM}} \) 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 \( \mathcal{P}_{\text{val}} \), 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, \( \max_{\mu \in \mathcal{P}_{\text{val}}} \epsilon(\mu) \), is \( 5.6 \times 10^{-4} \), 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.](image)

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Maximal error</th>
<th>Average runtime [s]/SpF</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOM</td>
<td>( 5.6 \times 10^{-4} )</td>
<td>287.4 / -</td>
</tr>
<tr>
<td>ROM</td>
<td>( 5.6 \times 10^{-4} )</td>
<td>28.7 / 10</td>
</tr>
</tbody>
</table>

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 \( P_{u_0} \) and \( P_{u_1} \), 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 \( Q_{max} = 1.0 \text{ ml/s} \), \( P_{u_0,\text{min}} = 95.0\% \), \( P_{u_1,\text{min}} = 95.0\% \). The optimization problems are solved by using the optimizer NLopt, 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 \( \mu_k \) be the parameter chosen by the optimizer at the \( k \)th iteration. The iteration continues until the relative variance of the decision variables \( \mu \) goes below a prespecified tolerance \( \varepsilon_{\text{opt}} \), 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.

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 ±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 \((Q_1, \ldots, Q_4) = (0.99, 0.6144, 0.7362, 0.5578) [\text{ml/s}]\). The switching period is fixed at \(t_s = 172 \text{ s}\). 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 \(\sigma[P_{ua}]\) in Table 6. The purity of the component \(b\) is more sensitive to \(Q_III\), reflected by the larger variance of \(\sigma[P_{ub}]\) in Table 7. In contrast, Tables 5 and 8 show that the flow rates \(Q_I\) and \(Q_IV\) have less effect on the purity 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 \(Q_1\) on the purity of both products \(P_{ua}\) and \(P_{ub}\) using the FOM and the ROM, respectively.](image)

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 $Q_I$.

<table>
<thead>
<tr>
<th>Model</th>
<th>Order</th>
<th>CPU time [h]</th>
<th>$E[P_{u_a}]$ [%]</th>
<th>$E[P_{u_b}]$ [%]</th>
<th>$\sigma[P_{u_a}]$ [%]</th>
<th>$\sigma[P_{u_b}]$ [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOM</td>
<td>800</td>
<td>2.04</td>
<td>95.00</td>
<td>94.96</td>
<td>0.07</td>
<td>0.15</td>
</tr>
<tr>
<td>ROM</td>
<td>47</td>
<td>0.20</td>
<td>94.99</td>
<td>94.96</td>
<td>0.07</td>
<td>0.15</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Order</th>
<th>CPU time [h]</th>
<th>$E[P_{u_a}]$ [%]</th>
<th>$E[P_{u_b}]$ [%]</th>
<th>$\sigma[P_{u_a}]$ [%]</th>
<th>$\sigma[P_{u_b}]$ [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOM</td>
<td>800</td>
<td>2.00</td>
<td>94.80</td>
<td>94.96</td>
<td>0.65</td>
<td>0.54</td>
</tr>
<tr>
<td>ROM</td>
<td>47</td>
<td>0.20</td>
<td>94.80</td>
<td>94.96</td>
<td>0.65</td>
<td>0.54</td>
</tr>
</tbody>
</table>

Figure 5: Effect of $Q_{II}$ on the purity of both products $P_{u_a}$ and $P_{u_b}$ using the FOM and the ROM, respectively.

Figure 6: Effect of $Q_{III}$ on the purity of both products $P_{u_a}$ and $P_{u_b}$ 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 ±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.
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].


