

INTERPOLATION-BASED MODEL ORDER REDUCTION FOR POLYNOMIAL PARAMETRIC SYSTEMS*

PETER BENNER[†] AND PAWAN GOYAL[‡]

Abstract. In this work, we investigate a model order reduction scheme for polynomial parametric systems. We begin with defining the generalized multivariate transfer functions for the system. Based on this, we aim at constructing a reduced-order system, interpolating the defined generalized transfer functions at a given set of interpolation points. Furthermore, we provide a method, inspired by the Loewner approach for linear and (quadratic-)bilinear systems, to determine a good-quality reduced-order system in an automatic way. We also discuss the computational issues related to the proposed method and a potential application of CUR matrix approximation in order to further speed-up simulations of reduced-order systems. We test the efficiency of the proposed methods via several numerical examples.

Key words. Model order reduction, interpolation, tensor algebra, matricization, polynomial dynamical systems, parametric systems, transfer functions.

AMS subject classifications. 15A69, 34C20, 41A05, 49M05, 93A15, 93C10, 93C15

1. Introduction. An accurate solution of time-dependent partial differential equations (PDEs), or ordinary differential equations (ODEs), or a combination of both requires a fine spatial discretization of the governing equations. This leads to a large number of equations, thus a high-dimensional system. This inevitably imposes a huge burden on computational resources, and more often than not, it is *almost* impossible to make use of such high-dimensional systems in engineering problem like, *e.g.*, optimization or control. A way to resolve this issue is to construct a reduced-order system or a low-dimensional model, replicating the important dynamics of the original system.

In this paper, we focus on parametric polynomial systems of the form:

$$(1.1) \quad \begin{aligned} E(\mathbf{p})\dot{x}(t, \mathbf{p}) &= A(\mathbf{p})x(t, \mathbf{p}) + \sum_{\xi=2}^d H_{\xi}(\mathbf{p})x^{\otimes \xi}(t, \mathbf{p}) + \sum_{\eta=1}^d N_{\eta}(\mathbf{p})(u(t) \otimes x^{\otimes \eta}(t, \mathbf{p})) \\ &\quad + B(\mathbf{p})u(t), \quad x(0, \mathbf{p}) = 0, \\ y(t, \mathbf{p}) &= C(\mathbf{p})x(t, \mathbf{p}), \end{aligned}$$

where $E(\mathbf{p}), A(\mathbf{p}) \in \mathbb{R}^{n \times n}$, $B(\mathbf{p}) \in \mathbb{R}^{n \times m}$, $C(\mathbf{p}) \in \mathbb{R}^{q \times n}$, $H_{\xi}(\mathbf{p}) \in \mathbb{R}^{n \times n^{\xi}}$, $\xi \in \{2, \dots, d\}$, $N_{\eta}(\mathbf{p}) \in \mathbb{R}^{n \times m \cdot n^{\eta}}$, $\eta \in \{1, \dots, d\}$; the state, input and output vectors are $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^q$, respectively; $x^{\otimes \xi} := \underbrace{x \otimes \dots \otimes x}_{\xi\text{-times}}$ and the

parameter vector is denoted $\mathbf{p} \in \mathbb{R}^{n_p}$. Since the system (1.1) has polynomial terms of the order up to d , we refer to it as a *d-th order polynomial system*. The system (1.1) lies in n -dimensional Euclidean space and generally, n is in $\mathcal{O}(10^5) - \mathcal{O}(10^6)$. Due to the computational burden mentioned above, we seek to construct a low-dimensional system, having the same structure as (1.1), which captures the dynamics of the original system (1.1) for any given input $u(t)$ and parameter \mathbf{p} in a desired domain.

*Submitted to the editors April 29, 2019.

[†]Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany (benner@mpi-magdeburg.mpg.de).

[‡]Corresponding author. Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany (goyalp@mpi-magdeburg.mpg.de).

Many of the widely used methods in model order reduction (MOR) to construct low-dimensional models for (parametric) nonlinear systems are based on *snapshots*. This means that the state vector x at time t needs to be evaluated for a given input and parameter. In this category, proper orthogonal decomposition is arguably the most favored method. This relies on determining the dominant subspace for the state vectors through singular value decomposition (SVD) of the collected snapshots, which is generally followed by computing a reduced-order system via Galerkin projection. For more details, we refer to [17]. For nonlinear systems, it is often combined with hyper-reduction methods such as EIM [6] and DEIM [13] to further reduce computational costs related to the reduced nonlinear terms. Another widely known method in this category is the *trajectory piecewise linear* method, in which a nonlinear system is approximated by a weighted sum of linearized systems (linearized along the trajectory). Then, each linear system is reduced using popular methods for linear systems such as balanced truncation or iterative methods, see, e.g. [3, 9, 12, 18]. Moreover, reduced basis methods, which are also snapshots-based methods, have been successfully applied to several nonlinear parametric systems, see, e.g., [25]. Although these methods have been very successful, they share a common drawback of being dependent on snapshots, or in other words, simulations for given inputs and parameters. Hence, it may become harder to obtain a reduced-order system to use e.g., in control.

In this work, we rather focus on MOR methods, allowing us to determine reduced-order systems without any prior knowledge of inputs. There are, broadly speaking, two types of such methods, namely interpolation-based approaches and balanced truncation. Recently, there have been significant efforts to extend these methods from linear to special classes of non-parametric polynomial systems, namely bilinear systems, and quadratic-bilinear systems, see, e.g., [4, 7, 8, 10, 11, 15]. For parametric nonlinear systems, there has been a very recent work for bilinear parametric systems [26], where the construction of an interpolating reduced system has been proposed for a given set of interpolation points, and such a problem for quadratic-bilinear parametric systems still remains to be studied.

In this paper, we investigate an interpolation-based MOR scheme to obtain a reduced-order system for the parametric system (1.1). For this, we first define generalized transfer functions for the system (1.1). Based on this, we aim at constructing a reduced-order system such that its generalized transfer functions interpolate those of the original system at a given set of interpolation points for the frequency and parameters. Furthermore, we propose a scheme, inspired by the Loewner approach for linear and (quadratic-)bilinear systems [4, 15], thus leading to an algorithm that allows us to construct a good quality reduced-order system in an automatic fashion. Furthermore, we discuss related computational aspects and an application of pseudo-skeletal matrix approximation, the so-called CUR, to further reduce the computational complexity related to the reduced nonlinear terms.

The remaining structure of the paper is as follows. In the following section, we discuss polynomialization of nonlinear systems and recap some basic concepts from tensor algebra. In Section 3, we present the generalized transfer functions corresponding to (1.1) for a fixed parameter vector and discuss the construction of an interpolating reduced-order system using Petrov-Galerkin projection. Based on this, we propose an algorithm which allows us to determine a good quality reduced-order system in an automatic fashion. In Section 4, we discuss the related computational aspects and investigate an application of CUR matrix approximation to further reduce the complexity of the reduced nonlinear terms. In the subsequent section, we extend the proposed method to polynomial parametric systems. In Section 6, we illustrate

the efficiency of the proposed algorithms by means of two benchmark problems and their variants. We conclude the paper with a summary of our contributions and future perspectives.

We make use of the following notation in the paper:

- `orth()`: it returns an orthonormal basis of a given matrix.
- The Hadamard product and Kronecker product are denoted by ‘ \circ ’ and ‘ \otimes ’, respectively.
- Using MATLAB[®] notation, $A(:, 1:r)$ denotes the first r columns of the matrix A , and $A(i, j)$ is the (i, j) th element of the matrix A .
- I_m is the identity matrix of size $m \times m$.
- $\mathcal{V}^{\otimes \xi}$ is a short-hand notation for $\underbrace{\mathcal{V} \otimes \cdots \otimes \mathcal{V}}_{\xi\text{-times}}$, where \mathcal{V} is a vector/matrix.

2. Polynomialization of Nonlinear Systems and Tensor Algebra. In this section, we recap two topics. We begin with the polynomialization of nonlinear systems.

2.1. Polynomialization of nonlinear systems. A class of nonlinear systems, containing nonlinear terms such as exponential, trigonometric, rational, can be rewritten as a polynomial system (1.1), by introducing some auxiliary variables. This process is very closely related to the McCormick relaxation, used in nonconvex optimization [24]. In the recent past, due to advances in the methodologies for MOR for quadratic-bilinear (QB) systems, there has been a substantial focus on rewriting a nonlinear system into the QB form. However, in the subsection, we will illustrate with an example how a polynomialization of a nonlinear system is done by introducing less auxiliary variables as compared to its quadratic-bilinearization.

An illustrative example. Let us consider the following one-dimensional nonlinear ODE:

$$(2.1a) \quad \dot{x}(t) = -x(t) - x^3(t) \cdot e^{-x(t)} + u(t),$$

$$(2.1b) \quad y(t) = x(t).$$

Now, we seek to rewrite the system (2.1) as a polynomial system via polynomialization. For this, we introduce an auxiliary variable as $z(t) := e^{-x(t)}$ and derive the corresponding differential equation. That is

$$\dot{z}(t) = -e^{-x(t)} \dot{x}(t) = -z(t) (-x(t) - x^3(t)z(t) + u(t)).$$

Thus, we can equivalently write the input-output system (2.1) as follows:

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} &= \begin{bmatrix} -x(t) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ x(t)z(t) \end{bmatrix} - \begin{bmatrix} x^3(t)z(t) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ x^3(t)z^2(t) \end{bmatrix} - \begin{bmatrix} 0 \\ z(t) \end{bmatrix} u(t) + \begin{bmatrix} u(t) \\ 0 \end{bmatrix}, \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix}. \end{aligned}$$

As can be seen, the system (2.1), which has cubic exponential nonlinearity, can be rewritten into a polynomial system (3.1) of order 5 by introducing a single variable. However, if one aims at rewriting the system into the QB form, then we need to introduce at least 3 more auxiliary variables, which somehow also makes the resulting system even more complicated, hence, also impeding its model reduction. Therefore, it is advantageous to work with the polynomial system of order 5 and thus, we aim at reducing polynomial systems with MOR schemes for polynomial systems.

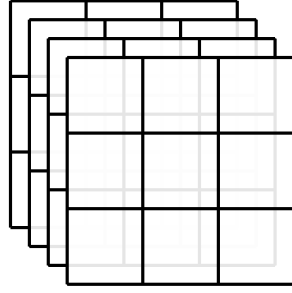


Figure 2.1: Illustration of a third-order tensor.

Furthermore, we emphasize that it still remains an open problem how many auxiliary variables are required minimally in order to rewrite a smooth nonlinear system into a polynomial system, which demands further research. However, we mention that there has been some initial work in [16] in this direction.

2.2. Tensor Algebra. As the nonlinear part of the considered systems are written in Kronecker (tensor) format, we will need a number of tensor based calculations in the remainder of this paper. We will review or introduce the necessary concepts in this subsection. A tensor is a multidimensional or an N -way array. An N th-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_N}$ is an N -dimensional array with entries $\mathcal{X}(i_1, \dots, i_N) \in \mathbb{R}$, where $i_j \in \{1, \dots, n_j\}, j \in \{1, \dots, N\}$. For illustration, in Figure 2.1, we present an illustration of third-order tensor. An important concept of a tensor is the so-called *matricization*. This allows us to unfold a tensor into a matrix, which plays a crucial role in tensor computations. For an N th order tensor, there are N different ways to unfold as a matrix. In the following, we define mode- n matricization of a tensor \mathcal{X} .

DEFINITION 2.1 (e.g., [20]). *The mode- n matricization of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times \dots \times n_N}$, denoted by $\mathbf{X}_{(n)}$, satisfies the following mapping:*

$$\mathbf{X}_{(n)}(i_n, j) = \mathcal{X}(i_1, \dots, i_N),$$

where $j = 1 + \sum_{k=1, k \neq n}^N (i_k - 1)J_k$ with $J_k = \prod_{m=1, m \neq n}^{k-1} n_m$.

Like matrix-vector and matrix-matrix products, tensor-tensor, tensor-matrix and tensor-vector products can be defined; however, the notation becomes quite cumbersome. In the following, we present a connection between the mode- n matricization and Kronecker products. For this, we define the following tensor-matrix product:

$$\mathbf{Y} = \mathcal{X} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \dots \times_N \mathbf{A}^{(N)},$$

where $\mathbf{A}^{(l)} \in \mathbb{R}^{J_l \times n_l}$. Then, we have the following relation:

$$(2.2) \quad \mathbf{Y}_{(n)} = \mathbf{A}^{(n)} \mathbf{X}_{(n)} \left(\mathbf{A}^{(N)} \otimes \dots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \mathbf{A}^{(1)} \right)^T.$$

Of particular interest of the paper, we explicitly note down the results for tensor-vector products as well. For this, let us define the following product:

$$(2.3) \quad \mathbf{Y} = \mathcal{X} \bar{\times}_1 \mathbf{a}_1 \bar{\times}_2 \dots \bar{\times}_M \mathbf{a}_M,$$

where $M \leq N$ and $\mathbf{a}_m \in \mathbb{R}^{j_m}$, $m \in \{1, \dots, M\}$. Then, using [20, Prop. 3.7], we define

$$(2.4) \quad \mathbf{Y}_{(n)} = a_n^T \mathbf{X}_{(n)} (\mathbf{a}_M \otimes \dots \otimes \mathbf{a}_{n+1} \otimes \mathbf{a}_{n-1} \otimes \dots \otimes \mathbf{a}_1), \quad n \in \{1, \dots, M\}.$$

Furthermore, we consider a special case, which is very useful later in the paper, that is when $N = M$. In this case, the quantity

$$\mathcal{X}_{\bar{\times}_1 \mathbf{a}_1 \bar{\times}_2 \dots \bar{\times}_N \mathbf{a}_N} =: \Xi$$

is a scalar. Hence, using (2.4), we obtain the following relation:

$$(2.5) \quad \begin{aligned} a_1^T \mathbf{X}_{(1)} (\mathbf{a}_N \otimes \dots \otimes \mathbf{a}_2) &= a_2^T \mathbf{X}_{(2)} (\mathbf{a}_N \otimes \dots \otimes \mathbf{a}_3 \otimes \mathbf{a}_1) = \dots \\ &= a_N^T \mathbf{X}_{(N)} (\mathbf{a}_{N-1} \otimes \dots \otimes \mathbf{a}_1). \end{aligned}$$

For further details on tensor concepts such as tensor-matrix, and tensor-vector multiplications, and matricization, we refer readers to [20] and references therein.

3. Construction of Interpolating Reduced-Order Systems. In this section, we discuss the construction of interpolating reduced-order systems. For simplicity, we begin with non-parametric polynomial systems as follows:

$$(3.1) \quad \begin{aligned} \dot{x}(t) &= Ax(t) + \sum_{\xi=2}^d H_{\xi} x^{\otimes \xi}(t) + \sum_{\eta=1}^d N_{\eta} (u(t) \otimes x^{\otimes \eta}(t)) + Bu(t), \quad x(0) = 0, \\ y(t) &= Cx(t), \end{aligned}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^q$ are state, input and output vectors respectively, and all other matrices are constants and are of appropriate size. Moreover, the system (3.1) is referred to as a single-input single-output (SISO) system when $q = m = 1$; otherwise, it is referred to as a multi-input multi-output system (MIMO).

3.1. Reduced-order modeling for SISO systems. We begin with considering SISO polynomial systems (3.1). As a first step towards developing a MOR scheme for the system, we aim at defining the generalized multivariate transfer functions. Following the steps as shown in [11] for QB systems, we write the Volterra series corresponding to the system (3.1) as follows:

$$(3.2) \quad \begin{aligned} x(t) &= \int_0^t e^{A\sigma_1} Bu(t_{\sigma_1}) d\sigma_1 + \sum_{\xi=2}^d \int_0^t e^{A\sigma_1} H_{\xi} x^{\otimes \xi}(t_{\sigma_1}) d\sigma_1 \\ &\quad + \int_0^t \sum_{\eta=1}^d e^{A\sigma_1} N_{\eta} x^{\otimes \eta}(t_{\sigma_1}) u(t_{\sigma_1}) d\sigma_1, \end{aligned}$$

where $t_{\sigma_1} := t - \sigma_1$. The above equation also allows us to express $x(t_{\sigma_1})$ as follows:

$$(3.3) \quad \begin{aligned} x(t_{\sigma_1}) &= \int_0^{t_{\sigma_1}} e^{A\sigma_2} Bu(t_{\sigma_1} - \sigma_2) d\sigma_2 + \sum_{\xi=2}^d \int_0^{t_{\sigma_1}} e^{A\sigma_2} H_{\xi} x^{\otimes \xi}(t_{\sigma_1} - \sigma_2) d\sigma_2 \\ &\quad + \sum_{\eta=1}^d \int_0^{t_{\sigma_1}} e^{A\sigma_2} N_{\eta} x^{\otimes \eta}(t_{\sigma_1} - \sigma_2) u(t_{\sigma_1} - \sigma_2) d\sigma_2. \end{aligned}$$

Substituting the expression in (3.3) for $x(t_{\sigma_1})$ in (3.2) and multiplying by C yields

$$\begin{aligned}
y(t) &= \int_0^t C e^{A\sigma_1} B u(t_{\sigma_1}) d\sigma_1 \\
&+ \sum_{\xi=2}^d \int_0^t \underbrace{\int_0^{t_{\sigma_1}} \cdots \int_0^{t_{\sigma_1}}}_{\xi\text{-times}} C e^{A\sigma_1} H_\xi (e^{A\sigma_2} B \otimes \cdots \otimes e^{A\sigma_{\xi+1}} B) d\sigma_1 d\sigma_2 \cdots d\sigma_{\xi+1} \\
&+ \sum_{\eta=1}^d \int_0^t \underbrace{\int_0^{t_{\sigma_1}} \cdots \int_0^{t_{\sigma_1}}}_{\eta\text{-times}} C e^{A\sigma_1} N_\eta (e^{A\sigma_2} B \otimes \cdots \otimes e^{A\sigma_{\eta+1}} B) \\
&\times (u(t_{\sigma_1}) u(t_{\sigma_1} - \sigma_2) \cdots u(t_{\sigma_1} - \sigma_{\eta+1})) d\sigma_1 d\sigma_2 \cdots d\sigma_{\eta+1} + \cdots .
\end{aligned}$$

As the above Volterra series, corresponding to the system (3.1), is cumbersome and contains infinitely many terms, we consider only the leading *kernels* of the series, which are as follows:

$$(3.4a) \quad f_L(t_1) := C e^{At_1} B,$$

$$(3.4b) \quad f_H^{(\xi)}(t_1, \dots, t_{\xi+1}) := C e^{At_1} H_\xi (e^{At_2} B \otimes \cdots \otimes e^{At_{\xi+1}} B),$$

$$(3.4c) \quad f_N^{(\eta)}(t_1, \dots, t_{\eta+1}) := C e^{At_1} N_\eta (e^{At_2} B \otimes \cdots \otimes e^{At_{\eta+1}} B),$$

where $\xi \in \{2, \dots, d\}$ and $\eta \in \{1, \dots, d\}$. Furthermore, taking the multivariate Laplace transform (see, e.g., [27]) of the above kernels, we get the frequency-domain representations of the kernels as follows:

$$(3.5a) \quad \mathbf{F}_L(s_1) := \mathcal{L}(f_L) = C \Phi(s_1) B,$$

$$(3.5b) \quad \mathbf{F}_H^{(\xi)}(s_1, \dots, s_{\xi+1}) := \mathcal{L}(f_H^{(\xi)}) = C \Phi(s_{\xi+1}) H_\xi (\Phi(s_\xi) B \otimes \cdots \otimes \Phi(s_1) B),$$

$$(3.5c) \quad \mathbf{F}_N^{(\eta)}(s_1, \dots, s_{\eta+1}) := \mathcal{L}(f_N^{(\eta)}) = C \Phi(s_{\eta+1}) N_\eta (\Phi(s_\eta) B \otimes \cdots \otimes \Phi(s_1) B),$$

where $\Phi(s) = (sI_n - A)^{-1}$ is the so-called state transition matrix, and $\mathcal{L}(\cdot)$ denotes the multivariate Laplace transform. In the above, we have assumed that the mass matrix in front of $\dot{x}(t)$ in (3.1) is $E = I_n$; however, one can also perform the above algebra to derive the multivariate transfer function for $E \neq I_n$ but invertible. In this case, we can also obtain the multivariate transfer functions as in (3.5), where the matrix $\Phi(s)$ will be $(sE - A)^{-1}$ instead of $(sI - A)^{-1}$. In the rest of the paper, we assume that the matrix E is an invertible matrix. We aim at constructing reduced-order systems, having a similar structure as in (3.1), as follows:

$$\begin{aligned}
(3.6) \quad \widehat{E} \dot{\widehat{x}}(t) &= \widehat{A} \widehat{x}(t) + \sum_{\xi=2}^d \widehat{H}_\xi \widehat{x}^{\otimes \xi}(t) + \sum_{\eta=1}^d \widehat{N}_\eta (u(t) \otimes \widehat{x}^{\otimes \eta}(t)) + \widehat{B} u(t), \quad \widehat{x}(0) = 0, \\
\widehat{y}(t) &= \widehat{C} \widehat{x}(t),
\end{aligned}$$

where $\widehat{x}(t) \in \mathbb{R}^r$, $u(t) \in \mathbb{R}$ and $\widehat{y}(t) \in \mathbb{R}$ are reduced state, input and output vectors, respectively with $r \ll n$, and all other matrices are of appropriate size. To that end, our goal is to construct reduced-order systems (3.6) using Petro-Galerkin projection such that the multivariate transfer functions, as given in (3.5), of the original system match with those of the reduced-order system at a given set of interpolation points.

For this, we essentially require projection matrices $V \in \mathbb{R}^{n \times r}$ and $W \in \mathbb{R}^{n \times r}$, thus leading to the system matrices of (3.6) as follows:

$$(3.7) \quad \begin{aligned} \widehat{E} &= W^T A V, & \widehat{A} &= W^T A V, & \widehat{H}_\xi &= W^T H_\xi V^{\otimes \xi}, & \xi &\in \{2, \dots, d\}, \\ \widehat{B} &= W^T B, & \widehat{C} &= C V, & \widehat{N}_\eta &= W^T N_\eta V^{\otimes \eta}, & \eta &\in \{1, \dots, d\} \end{aligned}$$

with $x(t) \approx V \widehat{x}(t)$. Clearly, the choice of the matrices V and W must ensure the desired interpolating properties of the original and reduced-order systems and also determines the quality of the reduced-order system. Thus, in the following theorem, we reveal the construction of the projection matrices V and W , yielding an interpolating reduced-order system.

THEOREM 3.1. *Consider a SISO system as given in (3.1). Let σ_i and μ_i , $i \in \{1, \dots, \widehat{r}\}$, be interpolation points such that $(sE - A)$ is invertible for all $s = \{\sigma_i, \mu_i\}$, $i \in \{1, \dots, \widehat{r}\}$. Moreover, let the projection matrices V and W be as follows:*

$$\begin{aligned} V_L &= \text{range}(\Phi(\sigma_1)B, \dots, \Phi(\sigma_{\widehat{r}})B), \\ V_N &= \bigcup_{\eta=1}^d \text{range}(\Phi(\lambda_{\eta+1})N_\eta(\Phi(\lambda_\eta)B \otimes \dots \otimes \Phi(\lambda_1)B)), \\ V_H &= \bigcup_{\xi=2}^d \text{range}(\Phi(\lambda_{\xi+1})H_\xi(\Phi(\lambda_\xi)B \otimes \dots \otimes \Phi(\lambda_1)B)), \\ W_L &= \text{range}(\Phi(\mu_1)^T C^T, \dots, \Phi(\mu_{\widehat{r}})^T C^T), \\ W_N &= \bigcup_{\eta=1}^d \text{range}(\Phi(\lambda_1)^T (N_\eta)_{(2)}(\Phi(\lambda_\eta)B \otimes \dots \otimes \Phi(\lambda_2)B \otimes \Phi(\beta)^T C^T)), \\ W_H &= \bigcup_{\xi=2}^d \text{range}(\Phi(\lambda_1)^T (H_\xi)_{(2)}(\Phi(\lambda_\xi)B \otimes \dots \otimes \Phi(\beta)^T C^T)), \\ V &= \text{range}(V_L, V_N, V_H), \\ W &= \text{range}(W_L, W_N, W_H), \end{aligned}$$

where $\Phi(s) := (sE - A)^{-1}$, $\lambda_i \in \{\sigma_1, \dots, \sigma_{\widehat{r}}\}$, $i \in \{1, \dots, d+1\}$, $\beta \in \{\mu_1, \dots, \mu_{\widehat{r}}\}$, and $(H_\xi)_{(2)} \in \mathbb{R}^{n \times n^\xi}$ and $(N_\eta)_{(2)} \in \mathbb{R}^{n \times m \cdot n^\xi}$ are, respectively, the mode-2 matricizations of the $(\xi+1)$ -way tensor $\mathcal{H}_\xi \in \mathbb{R}^{n \times \dots \times n}$ and $(\eta+2)$ -way tensor $\mathcal{N}_\eta \in \mathbb{R}^{n \times \dots \times n}$ whose mode-1 matricizations are H_ξ and N_η , respectively. Assume V and W are of full column rank. If a reduced-order system is computed as shown in (3.7) using the matrices V and W , then the reduced-order system satisfies the following interpolation conditions:

$$\begin{aligned} (3.9a) \quad & \mathbf{F}_L(\lambda_1) = \widehat{\mathbf{F}}_L(\lambda_1), \\ (3.9b) \quad & \mathbf{F}_L(\beta) = \widehat{\mathbf{F}}_L(\beta), \\ (3.9c) \quad & \mathbf{F}_N^{(\eta)}(\lambda_1, \dots, \lambda_{\eta+1}) = \widehat{\mathbf{F}}_N^{(\eta)}(\lambda_1, \dots, \lambda_{\eta+1}), \\ (3.9d) \quad & \mathbf{F}_N^{(\eta)}(\lambda_1, \dots, \lambda_\eta, \beta) = \widehat{\mathbf{F}}_N^{(\eta)}(\lambda_1, \dots, \lambda_\eta, \beta), \\ (3.9e) \quad & \mathbf{F}_H^{(\xi)}(\lambda_1, \dots, \lambda_{\xi+1}) = \widehat{\mathbf{F}}_H^{(\xi)}(\lambda_1, \dots, \lambda_{\xi+1}), \\ (3.9f) \quad & \mathbf{F}_H^{(\xi)}(\lambda_1, \dots, \lambda_\eta, \beta) = \widehat{\mathbf{F}}_H^{(\xi)}(\lambda_1, \dots, \lambda_\eta, \beta). \end{aligned}$$

Proof. The relations, given in (3.9a) and (3.9b) follows directly from the linear case, see, e.g., [3]. Therefore, we omit their proofs for the sake of brevity of the paper. However, for the rest of the proof, we note down intermediate results, which can be obtained while proving (3.9a) and (3.9b):

$$(3.10a) \quad V\widehat{\Phi}(\lambda_1)\widehat{B} = \Phi(\lambda_1)B, \quad \lambda_1 \in \{\sigma_1, \dots, \sigma_{\bar{r}}\}$$

$$(3.10b) \quad W\Phi(\beta)^T\widehat{C} = \Phi(\beta)^TC^T, \quad \beta \in \{\mu_1, \dots, \mu_{\bar{r}}\},$$

where $\Phi(s) = (sE - A)^{-1}$ and $\widehat{\Phi}(s) = (s\widehat{E} - \widehat{A})^{-1}\widehat{B}$. Now, we focus on the relation (3.9c). We begin with

$$\begin{aligned} & V\widehat{\Phi}(\lambda_{\eta+1})\widehat{N}_\eta \left(\widehat{\Phi}(\lambda_\eta)\widehat{B} \otimes \dots \otimes \widehat{\Phi}(\lambda_1)\widehat{B} \right) \\ &= V\widehat{\Phi}(\lambda_{\eta+1})W^T N_\eta V^\circledast \left(\widehat{\Phi}(\lambda_\eta)\widehat{B} \otimes \dots \otimes \widehat{\Phi}(\lambda_1)\widehat{B} \right) \\ & \quad \left(\because \widehat{N}_\eta = W^T N_\eta V^\circledast \right) \\ &= V\widehat{\Phi}(\lambda_{\eta+1})W^T N_\eta \left(V\widehat{\Phi}(\lambda_\eta)\widehat{B} \otimes \dots \otimes V\widehat{\Phi}(\lambda_1)\widehat{B} \right) \\ &= V\widehat{\Phi}(\lambda_{\eta+1})W^T N_\eta (\Phi(\lambda_\eta)B \otimes \dots \otimes \Phi(\lambda_1)B) \\ & \quad \text{(using (3.10a))} \\ &= V\widehat{\Phi}(\lambda_{\eta+1})W^T \underbrace{\Phi(\lambda_{\eta+1})^{-1} \Phi(\lambda_{\eta+1}) N_\eta (\Phi(\lambda_\eta)B \otimes \dots \otimes \Phi(\lambda_1)B)}_{\in V} \\ & \quad \text{(introduction of } I_n = \Phi(\lambda_{\eta+1})^{-1}\Phi(\lambda_{\eta+1}) \text{)} \\ (3.11) \quad &= V\widehat{\Phi}(\lambda_{\eta+1})W^T \Phi(\lambda_{\eta+1})^{-1}Vz, \end{aligned}$$

where the vector z is such that $Vz = \Phi(\lambda_{\eta+1})N_\eta (\Phi(\lambda_\eta)B \otimes \dots \otimes \Phi(\lambda_1)B)$. Additionally, we have

$$\begin{aligned} \widehat{\Phi}(s)W^T \Phi(s)^{-1}V &= (s\widehat{E} - \widehat{A})^{-1}W^T (sE - A)V \\ &= (s\widehat{E} - \widehat{A})^{-1}(sW^T EV - W^T AV) = I_r. \end{aligned}$$

Substituting the above relation in (3.11) and pre-multiplying with C yields the relation (3.9c). Similarly, we can prove the relation (3.9e). Next, we focus on the relation (3.9d). We know that

$$\widehat{N}_\eta = W^T N_\eta V^\circledast.$$

Hence, using (2.2), we obtain

$$(3.12) \quad \left(\widehat{N}_\eta \right)_{(2)} = V^T (N_\eta)_{(2)} \left(V^{\circledast(\eta-1)} \otimes W \right),$$

where $\left(\widehat{N}_\eta \right)_{(2)}$ is the mode-2 matricization of the tensor $\widehat{\mathcal{N}}_\eta$ whose mode-1 matricization is \widehat{N}_η . With the relation (3.12), we now consider

$$\begin{aligned} & W\widehat{\Phi}(\lambda_1)^T \left(\widehat{N}_\eta \right)_{(2)} \left(\widehat{\Phi}(\lambda_\eta)\widehat{B} \otimes \dots \otimes \Phi(\lambda_2)\widehat{B} \otimes \widehat{\Phi}(\beta)^T\widehat{C}^T \right) \\ &= W\widehat{\Phi}(\lambda_1)^T V^T (N_\eta)_{(2)} \left(V^{\circledast(\eta-1)} \otimes W \right) \left(\widehat{\Phi}(\lambda_\eta)\widehat{B} \otimes \dots \right) \end{aligned}$$

$$\begin{aligned}
& \otimes \Phi(\lambda_2) \widehat{B} \otimes \widehat{\Phi}(\beta)^T \widehat{C}^T \\
& \quad \text{(using (3.12))} \\
& = W \widehat{\Phi}(\lambda_1)^T V^T (N_\eta)_{(2)} \left(V \widehat{\Phi}(\lambda_\eta) \widehat{B} \otimes \cdots \otimes V \widehat{\Phi}(\lambda_2) \widehat{B} \otimes W \widehat{\Phi}(\beta)^T \widehat{C}^T \right) \\
& = W \widehat{\Phi}(\lambda_1)^T V^T (N_\eta)_{(2)} (s \Phi(\lambda_\eta) B \otimes \cdots \otimes \Phi(\lambda_2) B \otimes \Phi(\beta)^T C^T) \\
& \quad \text{(using (3.10))} \\
& = W \widehat{\Phi}(\lambda_1)^T V^T \Phi(\lambda_1)^{-T} \\
& \quad \underbrace{\times \Phi(\lambda_1)^T (N_\eta)_{(2)} (\Phi(\lambda_\eta) B \otimes \cdots \otimes \Phi(\lambda_2) B \otimes \Phi(\beta)^T C^T)}_{\in W(\cdot := Wq)} \\
& = W \widehat{\Phi}(\lambda_1)^T V^T \Phi(\lambda_1)^{-T} Wq = Wq \\
& = \Phi(\lambda_1)^T (N_\eta)_{(2)} (\Phi(\lambda_\eta) B \otimes \cdots \otimes \Phi(\lambda_2) B \otimes \Phi(\beta)^T C^T).
\end{aligned}$$

Next, we multiply both sides by B^T to get

$$\begin{aligned}
& \widehat{B}^T \widehat{\Phi}(\lambda_1)^T (\widehat{N}_\eta)_{(2)} \left(\widehat{\Phi}(\lambda_\eta) \widehat{B} \otimes \cdots \otimes \Phi(\lambda_2) \widehat{B} \otimes \widehat{\Phi}(\beta)^T \widehat{C}^T \right) \\
& \quad = B \Phi(\lambda_1)^T (N_\eta)_{(2)} (\Phi(\lambda_\eta) B \otimes \cdots \otimes \Phi(\lambda_2) B \otimes \Phi(\beta)^T C^T).
\end{aligned}$$

Using the matricization property of tensor-vector multiplications (2.4), we get

$$\widehat{C} \widehat{\Phi}(\beta) \widehat{N}_\eta \left(\widehat{\Phi}(\lambda_\eta) \widehat{B} \otimes \cdots \otimes \widehat{\Phi}(\lambda_1) \widehat{B} \right) = C \Phi(\beta) N_\eta (s \Phi(\lambda_\eta) B \otimes \cdots \otimes \Phi(\lambda_1) B),$$

which is nothing but the relation given in (3.9d). Using similar steps, we can prove (3.9f); thus, for the sake of brevity, we skip it. This concludes the proof. \square

3.2. Tangential-interpolating ROMs for MIMO systems. In this subsection, we discuss a construction of an interpolating reduced-order systems for MIMO polynomial systems. Similar to the SISO case, the leading generalized transfer functions for a MIMO polynomial system are given as follows:

$$(3.14a) \quad \mathbf{F}_L(s_1) = C \Phi(s_1) B,$$

$$(3.14b) \quad \mathbf{F}_H^{(\xi)}(s_1, \dots, s_{\xi+1}) = C \Phi(s_{\xi+1}) H_\xi (\Phi(s_\xi) B \otimes \cdots \otimes \Phi(s_1) B),$$

$$(3.14c) \quad \mathbf{F}_N^{(\eta)}(s_1, \dots, s_{\eta+1}) = C \Phi(s_{\eta+1}) N_\eta (I_m \otimes \Phi(s_\eta) B \otimes \cdots \otimes \Phi(s_1) B),$$

where $\Phi(s) = (sI_n - A)^{-1}$. In Theorem 3.1, we have provided a general interpolation framework for SISO polynomial systems, which can be extended to the MIMO case. However, a straightforward extension of the interpolation idea for the MIMO case might lead to projection matrices V and W with an unmanageable number of columns for the MIMO case. Therefore, we make use of the tangential interpolation concept from the linear case for MIMO systems [14]. Furthermore, for ease of practical implementation, we avoid vectors in the projection matrices V and W corresponding to cross frequencies. This means that we set $\lambda_1 = \lambda_2 = \cdots = \lambda_{\{\eta, \xi\}} = \beta$. As a result, we propose the following lemma that is arguably of more importance from a practical point of view.

LEMMA 3.2. *Consider the original system as given in (3.1). Let $\sigma_i \in \mathbb{C}$, $i \in \{1, \dots, \tilde{r}\}$, be interpolation points such that $sE - A$ is invertible for all $s \in \{\sigma_1, \dots, \sigma_{\tilde{r}}\}$,*

and $b_i \in \mathbb{C}^m$ and $c_i \in \mathbb{C}^q$ $i \in \{1, \dots, \tilde{r}\}$ be right and left tangential directions corresponding to σ_i , respectively. Let V and W be defined as follows:

$$V_L = \bigcup_{i=1}^{\tilde{r}} \text{range}(\Phi(\sigma_i)Bb_i),$$

$$V_N = \bigcup_{\eta=1}^d \bigcup_{i=1}^{\tilde{r}} \text{range}(\Phi(\sigma_i)N_\eta(I_m \otimes \Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i)),$$

$$V_H = \bigcup_{\xi=2}^d \bigcup_{i=1}^{\tilde{r}} \text{range}(\Phi(\sigma_i)H_\xi(\Phi(\sigma_i)Bb_i \otimes \cdots \otimes \Phi(\sigma_i)Bb_i)),$$

$$W_L = \bigcup_{i=1}^{\tilde{r}} \text{range}(\Phi(\sigma_i)^T C^T c_i),$$

$$W_N = \bigcup_{\eta=1}^d \bigcup_{i=1}^{\tilde{r}} \text{range}\left(\Phi(\sigma_i)(N_\eta)_{(2)}(I_m \otimes \Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\sigma_i)^T C^T)\right),$$

$$W_H = \bigcup_{\xi=2}^d \bigcup_{i=1}^{\tilde{r}} \text{range}\left(\Phi(\sigma_i)(H_\xi)_{(2)}(\Phi(\sigma_i)B \otimes \cdots \otimes \Phi(\sigma_i)B \otimes \Phi(\beta)^T C^T)\right),$$

$$V = \text{range}(V_L, V_N, V_H),$$

$$W = \text{range}(W_L, W_N, W_H).$$

If a reduced-order system is computed as shown in (3.7) using the projection matrices V and W , where we assume V and W to be of full rank, then the following interpolation conditions are fulfilled:

$$(3.16a) \quad \mathbf{F}_L(\sigma_i)b_i = \widehat{\mathbf{F}}_L(\sigma_i)b_i,$$

$$(3.16b) \quad c_i^T \mathbf{F}_L(\sigma_i) = c_i^T \widehat{\mathbf{F}}_L(\sigma_i),$$

$$(3.16c) \quad \frac{d}{ds_1} c_i^T \mathbf{F}_L(\sigma_i)b_i = \frac{d}{ds_1} c_i^T \widehat{\mathbf{F}}_L(\sigma_i)b_i,$$

$$(3.16d) \quad \mathbf{F}_N^{(\eta)}(\sigma_i, \dots, \sigma_i) \left(I_m \otimes b_i^{\textcircled{\eta}} \right) = \widehat{\mathbf{F}}_N^{(\eta)}(\sigma_i, \dots, \sigma_i) \left(I_m \otimes b_i^{\textcircled{\eta}} \right),$$

$$(3.16e) \quad c_i^T \mathbf{F}_N^{(\eta)}(\sigma_i, \dots, \sigma_i) \left(I_m^{\textcircled{\eta}} \otimes b_i^{\textcircled{\eta-1}} \right) = c_i^T \widehat{\mathbf{F}}_N^{(\eta)}(\sigma_i, \dots, \sigma_i) \left(I_m^{\textcircled{\eta}} \otimes b_i^{\textcircled{\eta-1}} \right)$$

$$(3.16f) \quad \frac{d}{ds_j} c_i^T \mathbf{F}_N^{(\eta)}(\sigma_i, \dots, \sigma_i) \left(I_m \otimes b_i^{\textcircled{\eta}} \right) = \frac{d}{ds_j} c_i^T \widehat{\mathbf{F}}_N^{(\eta)}(\sigma_i, \dots, \sigma_i) \left(I_m \otimes b_i^{\textcircled{\eta}} \right),$$

$$(3.16g) \quad \mathbf{F}_H^{(\xi)}(\sigma_i, \dots, \sigma_i)b_i^{\textcircled{\xi}} = \widehat{\mathbf{F}}_H^{(\xi)}(\sigma_i, \dots, \sigma_i)b_i^{\textcircled{\xi}},$$

$$(3.16h) \quad c_i^T \mathbf{F}_H^{(\xi)}(\sigma_i, \dots, \sigma_i) \left(I_m \otimes b_i^{\textcircled{\xi-1}} \right) = c_i^T \widehat{\mathbf{F}}_H^{(\xi)}(\sigma_i, \dots, \sigma_i) \left(I_m \otimes b_i^{\textcircled{\xi-1}} \right),$$

$$(3.16i) \quad \frac{d}{ds_j} c_i^T \mathbf{F}_H^{(\xi)}(\sigma_i, \dots, \sigma_i)b_i^{\textcircled{\xi}} = \frac{d}{ds_j} c_i^T \widehat{\mathbf{F}}_H^{(\xi)}(\sigma_i, \dots, \sigma_i)b_i^{\textcircled{\xi}}$$

where $i \in \{1, \dots, \tilde{r}\}$, $\xi \in \{2, \dots, d\}$, $\eta \in \{1, \dots, d\}$ and $\frac{d}{ds_j}$ denotes the partial derivative with respect to s_j of a given function.

Proof. The proof of (3.16a), (3.16b), (3.16d), (3.16e), (3.16g), and (3.16h) exactly follows the proof of [Theorem 3.1](#). Using very similar steps and simple algebra, one can easily prove the rest of the conditions. \square

3.3. Connection to the Loewner Approach. In recent years, Loewner-based MOR has received a lot of attention. For linear systems, the authors in [\[23\]](#) have discussed the Loewner approach to construct reduced-order systems using transfer function data. Later on, the Loewner approach has been extended to other classes of nonlinear systems, namely bilinear and QB systems in [\[4, 15\]](#), where data related to generalized transfer functions is required to obtain a reduced-order system.

An important ingredient in the Loewner approach is the construction of the Loewner matrix (\mathbb{L}) and the shifted Loewner matrix (\mathbb{L}_s). One way to construct the matrices \mathbb{L} and \mathbb{L}_s is either by using an experimental set-up or by using numerical evaluations of the generalized transfer functions, which is the primary inspiration of the method. However, there is a strong connection with interpolation of (generalized-) transfer functions, corresponding to a given system. As a result, we, alternatively, can construct the latter matrices by projection for a given realization of a system, ensuring the interpolation conditions.

For an example, let us consider 4 frequency measurements $H(\sigma_1), H(\sigma_2), H(\mu_1)$ and $H(\mu_2)$, where $H(s) := C(sE - A)^{-1}B \in \mathbb{C}$ is the transfer function of a linear SISO system with the system matrices (E, A, B, C) . As shown e.g., in [\[5\]](#), the matrices \mathbb{L} and \mathbb{L}_s , using the data points and letting $\sigma_{\{1,2\}}$ and $\mu_{\{1,2\}}$ to be the right and left interpolation points, can be constructed as follows:

$$(3.17) \quad \mathbb{L}(i, j) = \frac{H(\mu_i) - H(\sigma_j)}{\mu_i - \sigma_j}, \quad \mathbb{L}_s(i, j) = \frac{\mu_i H(\mu_i) - \sigma_j H(\sigma_j)}{\mu_i - \sigma_j},$$

where $i, j \in \{1, 2\}$. Moreover, if the matrices V and W are given as in [Theorem 3.1](#), i.e.,

$$\begin{aligned} V &= [(\sigma_1 E - A)^{-1}B, \quad (\sigma_2 E - A)^{-1}B], \\ W &= [(\mu_1 E - A)^{-T}C^T, \quad (\mu_2 E - A)^{-T}C^T], \end{aligned}$$

then the matrices \mathbb{L} and \mathbb{L}_s , shown in (3.17), can also be constructed as

$$(3.18) \quad \mathbb{L} = -W^T E V, \quad \mathbb{L}_s = -W^T A V.$$

A similar analogy can also be seen for bilinear and QB systems [\[4, 15\]](#). It is preferable to construct \mathbb{L} and \mathbb{L}_s using the data if the data corresponding to the transfer function can be either computed cheaply by its explicit expression or determined by an experimental setup. However, in the case nonlinear systems, it is not straightforward to determine the generalized transfer function by an experiment, which is mainly due to not having a clear interpretation of generalized transfer functions of nonlinear systems as in the case of linear systems. Thus, the method to determine \mathbb{L} and \mathbb{L}_s by projection shown in (3.18), can be of greater use when measurement data is not available but instead, we have a system realization.

In this paper, we assume that a realization of the polynomial systems (3.1) is given and thus focus on constructing the matrices \mathbb{L} and \mathbb{L}_s using projection (3.18), and the rest of the system matrices using the same projection matrices V and W are given as follows:

$$\begin{aligned} \mathbb{B} &= W^T B, & \mathbb{H}_\xi &= W^T H_\xi V^\otimes, & \xi &\in \{2, \dots, d\}, \\ \mathbb{C} &= C V, & \mathbb{N}_\eta &= W^T N_\eta (I_m \otimes V^\otimes), & \eta &\in \{1, \dots, d\}. \end{aligned}$$

By [Theorem 3.1](#), it is clear that the systems $\mathcal{S}_1 : (E, A, H_\xi, N_\eta, B, C)$ and $\mathcal{S}_2 : (\mathbb{L}, \mathbb{L}_s, \mathbb{H}_\xi, \mathbb{N}_\eta, \mathbb{B}, \mathbb{C})$ are interpolating at the considered frequency points. However, \mathcal{S}_2 can be singular, meaning that it may contain a lot of redundant information which can be compressed. Thus, inspired by the Loewner approach for linear, bilinear, and QB systems, we remove the redundancy by compressing the information using an SVD of the following matrices, composed of \mathbb{L} and \mathbb{L}_s :

$$(3.19) \quad \begin{bmatrix} \mathbb{L} & \mathbb{L}_s \end{bmatrix} = Y_1 \Sigma_1 X_1^T,$$

$$(3.20) \quad \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} = Y_2 \Sigma_2 X_2^T,$$

where the diagonal entries of Σ_1 and Σ_2 are in non-increasing order. Based on the first r columns of Y_1 and X_2 , denoted by Y_r and X_r , we can determine a compressed $\widehat{\mathcal{S}}_2$, compressing the information of \mathcal{S}_2 , as follows:

$$\begin{aligned} \widehat{E} &= Y_r^T \mathbb{L} X_r, & \widehat{A} &= Y_r^T \mathbb{L}_s X_r^T, & \widehat{H}_\xi &= Y_r^T \mathbb{H}_\xi X_r^\circledast, & \xi &\in \{2, \dots, d\}, \\ \widehat{B} &= Y_r^T \mathbb{B}, & \widehat{C} &= \mathbb{C} X_r, & \widehat{N}_\eta &= Y_r^T \mathbb{N}_\eta (I_m \otimes X_r^\circledast), & \eta &\in \{1, \dots, d\}. \end{aligned}$$

There are essentially two steps involved in order to get $\widehat{\mathcal{S}}_2$. In the first step, we require matrices such as \mathbb{H}_ξ and \mathbb{N}_η , which are generally dense, hence unmanageable. This is followed by compressing these matrices by using X_r and Y_r . However, upon closer inspection, we can determine $\widehat{\mathcal{S}}_2$ without completely forming \mathcal{S}_2 , or matrices \mathbb{H}_ξ and \mathbb{N}_η , but we can rather determine $\widehat{\mathcal{S}}_2$ by directly projecting the original system matrices using appropriate projection matrices. If we define the *effective* projection matrices as follows:

$$(3.21) \quad V_{\text{eff}} := V X_r, \quad \text{and} \quad W_{\text{eff}} := W Y_r,$$

then $\widehat{\mathcal{S}}_2$ can be determined in a traditional projection framework of the original system (3.1) as follows:

$$(3.22) \quad \begin{aligned} \widehat{E} &= W_{\text{eff}}^T E V_{\text{eff}}, & \widehat{A} &= W_{\text{eff}}^T A V_{\text{eff}}, & \widehat{H}_\xi &= W_{\text{eff}}^T H_\xi V_{\text{eff}}^\circledast, \\ \widehat{B} &= W_{\text{eff}}^T B, & \widehat{C} &= C V_{\text{eff}}, & \widehat{N}_\eta &= V_{\text{eff}}^T N_\eta (I_m \otimes V_{\text{eff}}^\circledast), \end{aligned}$$

where $\xi \in \{2, \dots, d\}$ and $\eta \in \{1, \dots, d\}$. We point out that it is advantageous to determine reduced system matrices, or the system $\widehat{\mathcal{S}}_2$ as shown in (3.22); this way, we are not required to form large dense matrices such as \mathbb{H}_ξ and \mathbb{N}_η . We can rather compute reduced matrices by multiplying efficiently the sparse and super-sparse¹ original matrices H_ξ and N_η with V_{eff} and W_{eff} . Having all these results, we briefly sketch the steps to determine reduced-order systems in [Algorithm 3.1](#). However, an important computational aspect related to tensor computations such as $W_{\text{eff}}^T H_\xi V_{\text{eff}}^\circledast$ still remains, which is discussed in the next section.

4. Computational Aspects and Application of CUR. In this section, we discuss two important computational aspects which are related to evaluating the non-linear terms of the reduced-order systems (3.22) and the use of the CUR matrix approximation in order to accelerate simulations of the reduced-order systems.

¹super-sparsity of a matrix is defined as a ratio of the number of non-zero distinct numbers to the total number of non-zero elements.

Algorithm 3.1 MOR for Non-Parametric Polynomial Systems (LbNPS-*Algo*).

Input: The system matrices $E, A, H_\xi, N_\eta, B, C$, $\xi \in \{2, \dots, d\}$, $\eta \in \{1, \dots, d\}$ and a set of interpolation points σ_i and corresponding tangential directions b_i and c_i , the reduced order r .

Output: The reduced system matrices $\hat{E}, \hat{A}, \hat{H}_\xi, \hat{N}_\eta, \hat{B}, \hat{C}$, $\xi \in \{2, \dots, d\}$, $\eta \in \{1, \dots, d\}$.

- 1: Determine V and W as shown in [Lemma 3.2](#).
- 2: Define Loewner and shifted Loewner matrices as follows:

$$\mathbb{L} = -W^T E V, \quad \mathbb{L}_s = -W^T A V,$$

- 3: Compute SVD of the matrices:

$$\begin{aligned} [\mathbb{L}, \mathbb{L}_s] &= Y_1 \Sigma_1 X_1^T, \\ \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} &= Y_2 \Sigma_2 X_2^T. \end{aligned}$$

- 4: Define $Y_r := Y_1(:, 1:r)$ and $X_r := X_2(:, 1:r)$.
- 5: Determine compact projection matrices:

$$V_{\text{eff}} := \text{orth}(V X_r) \text{ and } W_{\text{eff}} := \text{orth}(W Y_r).$$

- 6: Determine the reduced-order system as follows:

$$\begin{aligned} \hat{E} &= W_{\text{eff}}^T E V_{\text{eff}}, & \hat{A} &= W_{\text{eff}}^T A V_{\text{eff}}, & \hat{H}_\xi &= W_{\text{eff}}^T H_\xi V_{\text{eff}}^{\otimes \xi}, & \xi &\in \{2, \dots, d\}, \\ \hat{B} &= W_{\text{eff}}^T B, & \hat{C} &= C V_{\text{eff}}, & \hat{N}_\eta &= V_{\text{eff}}^T N_\eta \left(I_m \otimes V_{\text{eff}}^{\otimes \eta} \right), & \eta &\in \{1, \dots, d\}. \end{aligned}$$

4.1. Efficient evaluation the nonlinear terms of the ROMs. Let us begin with the computational effort related to evaluating, e.g., $\hat{H}_\xi := W_{\text{eff}}^T H_\xi V_{\text{eff}}^{\otimes \xi}$. It can be noticed that a direct computation of the above terms requires the computation of $V_{\text{eff}}^{\otimes \xi}$. Generally, the matrix V_{eff} is a dense matrix; thus, the computation related to $V_{\text{eff}}^{\otimes \xi}$ is of complexity $\mathcal{O}((n \cdot r)^\xi)$, which easily becomes an unmanageable task. For $\xi = 2$, the authors in [8] have proposed a method using tensor algebra to compute \hat{H}_2 without explicitly forming $V_{\text{eff}} \otimes V_{\text{eff}}$. On the other hand, the authors in [11] have aimed at exploiting the structure of the nonlinear operators, typically arising in PDEs/ODEs, thus also leading to an efficient method to compute \hat{H}_2 .

In this paper, we focus on the latter approach, where the explicit nonlinear operator of the PDEs is utilized, to compute \hat{H}_ξ . Extending the discussion in [11], in principle, we can write the term $H_\xi x^{\otimes \xi}$ in the system (3.1) in the Hadamard product form as follows:

$$(4.1) \quad H_\xi x^{\otimes \xi} = \mathcal{A}_1 x \circ \dots \circ \mathcal{A}_\xi x,$$

where \circ denotes the Hadamard product and $\mathcal{A}_i \in \mathbb{R}^{n \times n}$ are the constant matrices depending on the nonlinear operator in a governing equation. In order to reduce these nonlinear terms, resulting in a reduced-order system, we proceed as follows. Firstly, we substitute $x(t) \approx V_{\text{eff}} \hat{x}(t)$, where $x(t) \in \mathbb{R}^n$ and $\hat{x}(t) \in \mathbb{R}^r$ are the original and reduced state vectors, respectively, and then multiply W_{eff}^T from the left-hand side, thus leading to the corresponding nonlinear term:

$$\hat{H}_\xi \hat{x}^{\otimes \xi} = W_{\text{eff}}^T \left(\left(\hat{\mathcal{A}}_1 \hat{x} \right) \circ \dots \circ \left(\hat{\mathcal{A}}_\xi \hat{x} \right) \right),$$

where $\widehat{\mathcal{A}}_i = \mathcal{A}_i V_{\text{eff}}$, $i \in \{1, \dots, \xi\}$. Next, we use the relation between a Hadamard product and the Kronecker product, that is

$$\mathcal{P}p \circ \mathcal{Q}q = \begin{bmatrix} \mathcal{P}(1, :) \otimes \mathcal{Q}(1, :) \\ \vdots \\ \mathcal{P}(n, :) \otimes \mathcal{Q}(n, :) \end{bmatrix} (p \otimes q).$$

Thus, we get

$$(4.2) \quad W_{\text{eff}}^T \left((\widehat{\mathcal{A}}_1 \widehat{x}) \circ \dots \circ (\widehat{\mathcal{A}}_\xi \widehat{x}) \right) = W_{\text{eff}}^T \underbrace{\begin{bmatrix} \widehat{\mathcal{A}}_1(1, :) \otimes \dots \otimes \widehat{\mathcal{A}}_\xi(1, :) \\ \vdots \\ \widehat{\mathcal{A}}_1(n, :) \otimes \dots \otimes \widehat{\mathcal{A}}_\xi(n, :) \end{bmatrix}}_{=:\widehat{\mathcal{A}}} \widehat{x}^{\otimes}.$$

It can be seen that $W_{\text{eff}}^T \widehat{\mathcal{A}} = \widehat{H}_\xi$. Summarizing, we can perform computations related to \widehat{H}_ξ efficiently by utilizing the particular structure of the nonlinear terms in PDEs/ODEs, without explicitly forming V_{eff}^{\otimes} . We illustrate the procedure for a typical nonlinear PDE term in [Subsection 4.3](#).

4.2. CUR matrix approximation and ROMs. Next, we discuss another computational issue, due to which we may not achieve the desired reduction in the simulation time even after reducing the original system (3.1). Explaining this issue further, the reduced matrices such as $\widehat{H}_\xi \in \mathbb{R}^{r \times r^\xi}$ are generally dense matrices which are multiplied with \widehat{x}^{\otimes} . Thus, the computation $\widehat{H}_\xi \widehat{x}^{\otimes}$ is in $\mathcal{O}(r^{2\xi+1})$, which increases rapidly with the order of the reduced system or polynomial system (3.1). As a remedy, in this paper, we propose a new procedure to approximate $\widehat{H}_\xi \widehat{x}^{\otimes}$, which can be computed cheaply. For this, we make use of the *CUR matrix approximation*, see, e.g. [22, 28, 29]. Using this, we can approximate the matrix $\widehat{\mathcal{A}}$, defined in (4.2), as follows:

$$(4.3) \quad \widehat{\mathcal{A}} \approx CUR,$$

where $\mathcal{C} \in \mathbb{R}^{n \times n_c}$ and $\mathcal{R} \in \mathbb{R}^{n_r \times r^l}$ contain wisely chosen n_c columns and n_r rows of the matrix $\widehat{\mathcal{A}}$, respectively, and $\mathcal{U} \in \mathbb{R}^{n_c \times n_r}$ is determined such that it minimizes $\|\widehat{\mathcal{A}} - CUR\|$ in an appropriate norm. There has been a significant research how to choose columns and rows appropriately, leading to a good or even optimal in some sense, approximation of a matrix. We refer the reader to [22, 28, 29] and references therein for more details. Substituting the relation (4.3) in (4.2) results in

$$(4.4) \quad W_{\text{eff}}^T \widehat{\mathcal{A}} \widehat{x}^{\otimes} \approx W_{\text{eff}}^T CUR \widehat{x}^{\otimes}.$$

Next, we closely look at the term $\mathcal{R} \widehat{x}^{\otimes}$, whose columns are given as

$$(4.5) \quad \left(\widehat{\mathcal{A}}_1(i_r, :) \otimes \dots \otimes \widehat{\mathcal{A}}_\xi(i_r, :) \right) \widehat{x}^{\otimes},$$

where i_r belongs to the columns chosen by the CUR matrix approximation. We know that $\widehat{\mathcal{A}}_1(i_r, :) = \mathcal{A}_1(i_r, :)V$. Substituting this relation and $x \approx V\widehat{x}$, we get

$$\begin{aligned} & \widehat{\mathcal{A}}_1(i_r, :) \otimes \dots \otimes \widehat{\mathcal{A}}_\xi(i_r, :) \widehat{x}^{\otimes} \\ &= (\mathcal{A}_1(i_r, :) \otimes \dots \otimes \mathcal{A}_\xi(i_r, :)) V^{\otimes} \widehat{x}^{\otimes} \\ &\approx (\mathcal{A}_1(i_r, :) \otimes \dots \otimes \mathcal{A}_\xi(i_r, :)) x^{\otimes} := \text{NL}_{i_r}. \end{aligned}$$

Comparing the above quantity with (4.1), it can be noticed that the quantity \mathbf{NL}_{i_r} is nothing but the computation of the corresponding nonlinearity of the original system at a particular grid point. Furthermore, the term $W_{\text{eff}}^T \mathbf{C} \mathbf{U} \in \mathbb{R}^{r \times n_r}$ can be precomputed. This idea is very closely related to empirical interpolation methods, which are commonly used in reduced basis methods or proper orthogonal decomposition for nonlinear systems to reduce the computational cost related to nonlinear terms [6, 13].

4.3. An illustration using Chafee-Infante equation. In the following, we illustrate the computation of the reduced nonlinear term \widehat{H}_ξ and the usage of the CUR decomposition with the help of the Chafee-Infante equation. At this stage, we avoid describing the governing PDEs of the Chafee-Infante equation; we provide a detailed description of it in the numerical section. However, at the moment, we just note that it has cubic nonlinearity, i.e., $-v^3$, where v is the dependent variable. Hence, if the system is written in the form given in (3.1), we have the following nonlinear term:

$$H_3 x^\otimes := -x \circ x \circ x.$$

If the above term is reduced using the projection matrices V_{eff} and W_{eff} as shown in (3.22), then we obtain

$$\begin{aligned} (4.6) \quad W_{\text{eff}}^T H_3 V_{\text{eff}}^\otimes \widehat{x}^\otimes &= \widehat{H}_3 \widehat{x}^\otimes \\ &= W_{\text{eff}}^T (V_{\text{eff}} \widehat{x} \circ V_{\text{eff}} \widehat{x} \circ V_{\text{eff}} \widehat{x}) \\ &= W_{\text{eff}}^T \underbrace{\begin{bmatrix} V_{\text{eff}}(1, :) \otimes V_{\text{eff}}(1, :) \otimes V_{\text{eff}}(1, :) \\ \vdots \\ V_{\text{eff}}(n, :) \otimes V_{\text{eff}}(n, :) \otimes V_{\text{eff}}(n, :) \end{bmatrix}}_{=: \mathcal{V}_{\text{eff}}} (\widehat{x} \otimes \widehat{x} \otimes \widehat{x}). \end{aligned}$$

Equation (4.6) shows that instead of explicitly forming V_{eff}^\otimes to determine \widehat{H}_3 , we can rather compute it by a smart choice of rows and perform the Kronecker products as shown in (4.6). Furthermore, as discussed earlier, the evaluation of the term $\widehat{H}_3 \widehat{x}^\otimes$, in general, is of complexity $\mathcal{O}(r^7)$, which might be expensive if the order of the reduced system (r) is notable. Also, we stress that the term $\widehat{H}_3 \widehat{x}^\otimes$ needs to be computed at each time step for every simulation. To ease that we aim at further approximating $\widehat{H}_3 \widehat{x}^\otimes$. Thus, we first apply the CUR matrix approximation to the matrix \mathcal{V}_{eff} , defined in (4.6), to approximate it by using selected columns and rows, that is

$$(4.7) \quad \mathcal{V}_{\text{eff}} \approx \mathcal{C}_v \mathcal{U}_v \mathcal{R}_v,$$

where $\mathcal{C}_v \in \mathbb{R}^{r \times n_c}$ and $\mathcal{R}_v \in \mathbb{R}^{n_r \times r^3}$ consist of columns and rows of \mathcal{V}_{eff} , respectively. Let us assume that $\mathcal{I}_{\mathbf{R}} \subseteq \{1, \dots, n\}$ denotes the indices, leading to the construction of the matrix \mathcal{R}_v in (4.7), i.e.,

$$\mathcal{R}_v = \mathcal{V}_{\text{eff}}(\mathcal{I}_{\mathbf{R}}, :).$$

As a result, we use the relation (4.7) in (4.6) to obtain

$$\begin{aligned} W_{\text{eff}}^T \mathcal{V}_{\text{eff}} \widehat{x}^\otimes &\approx W_{\text{eff}}^T \mathcal{C}_v \mathcal{U}_v \mathcal{R}_v \widehat{x}^\otimes \\ &\approx \underbrace{W_{\text{eff}}^T \mathcal{C}_v \mathcal{U}_v}_{=: \Psi} (\mathcal{V}_{\text{eff}}(\mathcal{I}_{\mathbf{R}}, :) \widehat{x}^\otimes). \end{aligned}$$

Now, it can be noticed that the term $\mathcal{V}_{\text{eff}}(\mathcal{I}_{\mathbf{R}}, \cdot) \hat{x}^{\otimes 3}$ is nothing but evaluating the nonlinearity (in this case, it is a cubic nonlinearity) at indices $\mathcal{I}_{\mathbf{R}}$. As a result, we need to determine the nonlinearity at n_r points. Moreover, the matrix $\Psi \in \mathbb{R}^{r \times r_v}$ can be precomputed. This is exactly the idea of hyper-reduction methods such as (D)EIM proposed in [6, 13] in the case of nonlinear MOR. However, a major difference between the methodology in this paper and (D)EIM is that we do not require time-domain snapshots of the nonlinearity as needed in the case of DEIM, but we rather approximate the nonlinear terms in the reduced-order systems. Summarizing, for the Chafee-Infante equation in the end, we have

$$(4.8) \quad W_{\text{eff}}^T \mathcal{V}_{\text{eff}}(\hat{x} \otimes \hat{x} \otimes \hat{x}) \approx \Psi(\tilde{x} \circ \tilde{x} \circ \tilde{x}),$$

where $\tilde{x} = V_{\text{eff}}(\mathcal{I}_{\mathbf{R}}, \cdot) \hat{x}$, which is of complexity $\mathcal{O}(r \cdot n_r^2)$.

REMARK 4.1. *In the above, we have focused on the computational aspect related to \hat{H}_ξ and $\hat{H}_\xi \hat{x}^{\otimes 3}$. However, analogously, a complexity reduction can be performed for computing \hat{N}_η and $\hat{N}_\eta \hat{x}^{\otimes 3}$.*

5. Parametric Polynomial Systems. Until now, we have discussed the non-parametric case, i.e., all the system matrices are assumed to be constant. In this section, we briefly present an extension of the idea presented in the previous sections to parametric polynomial system (1.1). Similar to the non-parametric case, we can derive generalized transfer functions for the parametric case, which are given as follows:

$$(5.1a) \quad \mathbf{F}_L(s_1, \mathbf{p}) = C(\mathbf{p})\Phi(s_1, \mathbf{p})B(\mathbf{p}),$$

$$(5.1b)$$

$$\mathbf{F}_H^{(\xi)}(s_1, \dots, s_{\xi+1}, \mathbf{p}) = C(\mathbf{p})\Phi(s_{\xi+1}, \mathbf{p})H_\xi(\Phi(s_\xi, \mathbf{p}) \otimes \dots \otimes \Phi(s_1, \mathbf{p}))B(\mathbf{p})^{\otimes \xi},$$

$$(5.1c)$$

$$\mathbf{F}_N^{(\eta)}(s_1, \dots, s_{\eta+1}, \mathbf{p}) = C(\mathbf{p})\Phi(s_{\eta+1}, \mathbf{p})N_\eta(I_m \otimes \Phi(s_\eta, \mathbf{p}) \otimes \dots \otimes \Phi(s_1, \mathbf{p}))B(\mathbf{p})^{\otimes \eta},$$

where $\Phi(s, \mathbf{p}) = (sE(\mathbf{p}) - A(\mathbf{p}))^{-1}$. In the following, we present an extension of Lemma 3.2 to the parametric case.

THEOREM 5.1. *Consider the original parametric polynomial system as given in (3.1). Let σ_i, \mathbf{p}_i , $i \in \{1, \dots, \tilde{r}\}$ be interpolation points such that $sE(\mathbf{p}) - A(\mathbf{p})$ is invertible for all $s \in \{\sigma_1, \dots, \sigma_{\tilde{r}}\}$, $\mathbf{p} \in \{\mathbf{p}_1, \dots, \mathbf{p}_{\tilde{r}}\}$, and let $b_i \in \mathbb{R}^m$ and $c_i \in \mathbb{R}^q$ $i \in \{1, \dots, \tilde{r}\}$ be tangential directions. Moreover, let V and W be defined as follows:*

$$V_L = \bigcup_{i=1}^{\tilde{r}} \text{range}(\Phi(\sigma_i, \mathbf{p}_i)B(\mathbf{p}_i)b_i),$$

$$V_N = \bigcup_{\eta=1}^d \bigcup_{i=1}^{\tilde{r}} \text{range}(\Phi(\sigma_i, \mathbf{p}_i)N_\eta(I_m \otimes \Phi(\sigma_i, \mathbf{p}_i)B(\mathbf{p}_i)b_i \otimes \dots \otimes \Phi(\sigma_i, \mathbf{p}_i)B(\mathbf{p}_i)b_i)),$$

$$V_H = \bigcup_{\xi=2}^d \bigcup_{i=1}^{\tilde{r}} \text{range}(\Phi(\sigma_i, \mathbf{p}_i)H_\xi(\Phi(\sigma_i, \mathbf{p}_i)B(\mathbf{p}_i)b_i \otimes \dots \otimes \Phi(\sigma_i, \mathbf{p}_i)B(\mathbf{p}_i)b_i)),$$

$$W_L = \bigcup_{i=1}^{\tilde{r}} \text{range}(\Phi(\sigma_i, \mathbf{p}_i)^T C(\mathbf{p}_i)^T c_i)$$

$$W_N = \bigcup_{\eta=1}^d \bigcup_{i=1}^{\tilde{r}} \text{range}(\Phi(\sigma_i, \mathbf{p}_i)(N_\eta)_{(2)}(I_m \otimes \Phi(\sigma_i, \mathbf{p}_i)B(\mathbf{p}_i)b_i \otimes \dots$$

$$\begin{aligned}
& \otimes \Phi(\sigma_i, \mathbf{p}_i) B(\mathbf{p}_i) b_i \otimes \Phi(\sigma_i, \mathbf{p}_i)^T C(\mathbf{p}_i)^T c_i), \\
W_H = & \bigcup_{\xi=2}^d \bigcup_{i=1}^{\tilde{r}} \text{range} \left(\Phi(\sigma_i, \mathbf{p}_i) (H_\xi)_{(2)} (\Phi(\sigma_i, \mathbf{p}_i) B(\mathbf{p}_i) b_i \otimes \cdots \right. \\
& \left. \otimes \Phi(\sigma_i, \mathbf{p}_i) B(\mathbf{p}_i) b_i \otimes \Phi(\sigma_i, \mathbf{p}_i)^T C(\mathbf{p}_i)^T c_i \right),
\end{aligned}$$

$$V = \text{range} (V_L, V_N, V_H),$$

$$W = \text{range} (W_L, W_N, W_H).$$

If a reduced-order system is computed using the projection matrices V and W given above, assuming V and W are full rank matrices, as follows:

$$\begin{aligned}
(5.3) \quad \widehat{E}(\mathbf{p}) &= W^T E(\mathbf{p}) V, \quad \widehat{A}(\mathbf{p}) = W^T A(\mathbf{p}) V, \quad \widehat{H}_\xi(\mathbf{p}) = W^T H_\xi(\mathbf{p}) V^{\otimes \xi}, \quad \xi \in \{2, \dots, d\}, \\
\widehat{B}(\mathbf{p}) &= W^T B(\mathbf{p}), \quad \widehat{C}(\mathbf{p}) = C(\mathbf{p}) V, \quad \widehat{N}_\eta(\mathbf{p}) = W^T N_\eta(\mathbf{p}) V^{\otimes \eta}, \quad \eta \in \{1, \dots, d\}.
\end{aligned}$$

then the following interpolation conditions are fulfilled:

$$\begin{aligned}
\mathbf{F}_L(\sigma_i, \mathbf{p}_i) b_i &= \widehat{\mathbf{F}}_L(\sigma_i, \mathbf{p}_i) b_i, \\
c_i^T \mathbf{F}_L(\sigma_i, \mathbf{p}_i) &= c_i^T \widehat{\mathbf{F}}_L(\sigma_i, \mathbf{p}_i), \\
\frac{d}{ds_1} c_i^T \mathbf{F}_L(\sigma_i, \mathbf{p}_i) b_i &= \frac{d}{s_1} c_i^T \widehat{\mathbf{F}}_L(\sigma_i, \mathbf{p}_i) b_i, \\
\frac{d}{d\mathbf{p}} c_i^T \mathbf{F}_L(\sigma_i, \mathbf{p}_i) b_i &= \frac{d}{d\mathbf{p}} c_i^T \widehat{\mathbf{F}}_L(\sigma_i, \mathbf{p}_i) b_i, \\
\mathbf{F}_N^{(\eta)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) \left(I_m \otimes b_i^{\otimes \eta} \right) &= \widehat{\mathbf{F}}_N^{(\eta)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) \left(I_m \otimes b_i^{\otimes \eta} \right), \\
c_i^T \mathbf{F}_N^{(\eta)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) \left(I_m^{\otimes \eta} \otimes b_i^{\otimes (\eta-1)} \right) &= c_i^T \widehat{\mathbf{F}}_N^{(\eta)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) \left(I_m^{\otimes \eta} \otimes b_i^{\otimes (\eta-1)} \right), \\
\frac{d}{ds_j} c_i^T \mathbf{F}_N^{(\eta)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) \left(I_m \otimes b_i^{\otimes \eta} \right) &= \frac{d}{ds_j} c_i^T \widehat{\mathbf{F}}_N^{(\eta)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) \left(I_m \otimes b_i^{\otimes \eta} \right), \\
\frac{d}{d\mathbf{p}} c_i^T \mathbf{F}_N^{(\eta)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) \left(I_m \otimes b_i^{\otimes \eta} \right) &= \frac{d}{d\mathbf{p}} c_i^T \widehat{\mathbf{F}}_N^{(\eta)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) \left(I_m \otimes b_i^{\otimes \eta} \right), \\
\mathbf{F}_H^{(\xi)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) b_i^{\otimes \xi} &= \widehat{\mathbf{F}}_H^{(\xi)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) b_i^{\otimes \xi}, \\
c_i^T \mathbf{F}_H^{(\xi)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) \left(I_m \otimes b_i^{\otimes (\xi-1)} \right) &= c_i^T \widehat{\mathbf{F}}_H^{(\xi)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) \left(I_m \otimes b_i^{\otimes (\xi-1)} \right), \\
\frac{d}{ds_j} c_i^T \mathbf{F}_H^{(\xi)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) b_i^{\otimes \xi} &= \frac{d}{ds_j} c_i^T \widehat{\mathbf{F}}_H^{(\xi)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) b_i^{\otimes \xi}, \\
\frac{d}{d\mathbf{p}} c_i^T \mathbf{F}_H^{(\xi)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) b_i^{\otimes \xi} &= \frac{d}{d\mathbf{p}} c_i^T \widehat{\mathbf{F}}_H^{(\xi)}(\sigma_i, \dots, \sigma_i, \mathbf{p}_i) b_i^{\otimes \xi},
\end{aligned}$$

where $F_L, F_N^{(\eta)}, F_H^{(\xi)}, \widehat{F}_L, \widehat{F}_N^{(\eta)}, \widehat{F}_H^{(\xi)}$ are also assumed to be differentiable with respect to s_j and \mathbf{p} .

Proof. The proof of the lemma can be proven along the lines of the proof of [Theorem 3.1](#). Therefore, for the sake of brevity of the paper, we skip the proof. \square

In the above, we have assumed a general parametric structure for the system matrices, e.g., $E(\mathbf{p}), A(\mathbf{p})$, and the corresponding reduced-order system can be computed as

shown in (5.3). However, if we assume an affine parametric structure of the system matrices as follows:

$$(5.5) \quad \begin{aligned} E(\mathbf{p}) &= \sum_{i=1}^{t_e} \alpha_e^{(i)}(\mathbf{p}) E^{(i)}, & A(\mathbf{p}) &= \sum_{i=1}^{t_a} \alpha_a^{(i)}(\mathbf{p}) A^{(i)}, & H_\xi(\mathbf{p}) &= \sum_{i=1}^{t_{h_\xi}} \alpha_{h_\xi}^{(i)}(\mathbf{p}) H_\xi^{(i)}, \\ B(\mathbf{p}) &= \sum_{i=1}^{t_b} \alpha_b^{(i)}(\mathbf{p}) B^{(i)}, & C(\mathbf{p}) &= \sum_{i=1}^{t_c} \alpha_c^{(i)}(\mathbf{p}) C^{(i)}, & N_\eta(\mathbf{p}) &= \sum_{i=1}^{t_{n_\eta}} \alpha_{n_\eta}^{(i)}(\mathbf{p}) N_\eta^{(i)}, \end{aligned}$$

then the resulting reduced-order system can be determined, having the same structure

$$(5.6) \quad \begin{aligned} \widehat{E}(\mathbf{p}) &= \sum_{i=1}^{t_e} \alpha_e^{(i)}(\mathbf{p}) \widehat{E}^{(i)}, & \widehat{A}(\mathbf{p}) &= \sum_{i=1}^{t_a} \alpha_a^{(i)}(\mathbf{p}) \widehat{A}^{(i)}, & \widehat{H}_\xi(\mathbf{p}) &= \sum_{i=1}^{t_{h_\xi}} \alpha_{h_\xi}^{(i)}(\mathbf{p}) \widehat{H}_\xi^{(i)}, \\ \widehat{B}(\mathbf{p}) &= \sum_{i=1}^{t_b} \alpha_b^{(i)}(\mathbf{p}) \widehat{B}^{(i)}, & \widehat{C}(\mathbf{p}) &= \sum_{i=1}^{t_c} \alpha_c^{(i)}(\mathbf{p}) \widehat{C}^{(i)}, & \widehat{N}_\eta(\mathbf{p}) &= \sum_{i=1}^{t_{n_\eta}} \alpha_{n_\eta}^{(i)}(\mathbf{p}) \widehat{N}_\eta^{(i)}, \end{aligned}$$

where the original matrices are projected by the standard projection for given projection matrices; for example:

$$(5.7) \quad \begin{aligned} \widehat{E}^{(1)} &= W^T E^{(1)} V, & \widehat{A}^{(1)} &= W^T A^{(1)} V, & \widehat{H}_\xi^{(1)} &= W^T H_\xi^{(1)} V^\otimes, & \xi &\in \{2, \dots, d\}, \\ \widehat{B}^{(1)} &= W^T B^{(1)}, & \widehat{C}^{(1)} &= C^{(1)} V, & \widehat{N}_\eta^{(1)} &= W^T N_\eta^{(1)} V^\oplus, & \eta &\in \{1, \dots, d\}. \end{aligned}$$

Furthermore, like in the non-parametric case, we can easily develop the connection to the Loewner and shifted-Loewner type system, assuming we have sufficient interpolation points for frequency and parameter variables. In Algorithm 5.1, we outline all the steps to construct reduced-order systems for the parametric case, which is again inspired by the Loewner-type MOR.

6. Numerical Results. In this section, we illustrate the efficiency of the proposed methods by means of two nonlinear PDE examples and their variants. All the simulations were done on an Intel[®] Core™i7-6700 CPU@3.40GHz, 8 MB cache, 8 GB RAM, Ubuntu 16.04, MATLAB Version 9.1.0.441655(R2016b) 64-bit(glnxa64). In the following, we note some details used in the numerical simulations:

- All original and reduced-order systems are integrated by the routine `ode15s` in MATLAB with relative error and absolute error tolerances of 10^{-10} .
- We measure the output at 500 equidistant points within the time interval $[0, T]$, where T is the end time.
- We choose interpolation points for the frequency (s) in a logarithmic scale for a given frequency range, and interpolation points for parameters (\mathbf{p}) are chosen randomly using the `rand` command. To ensure reproducibility, we use `randn('seed', 0)` to initialize a random number generator.

6.1. Chafee-Infante equation. In our first example, we deal with a widely considered one-dimensional Chafee-Infante equation. Its governing equation and boundary conditions are given as follows:

$$(6.1) \quad \begin{aligned} \dot{v}(t) &= v_{xx} + v(1 - v^2), & x &\in (0, L) \times (0, T), & v(0, \cdot) &= u(t), & (0, T), \\ v_x(L, \cdot) &= 0, & (0, T), & & v(x, 0) &= 0, & (0, L). \end{aligned}$$

MOR of this example has been considered in various papers [8, 10, 11, 15], where the authors have proposed different methods to reduce it. The governing equation

Algorithm 5.1 Model Reduction for Parametric Polynomial Systems (1.1).

Input: The original system (1.1) with the affine structure as in (5.5), and a set of interpolation points for the frequency and parameters, i.e., σ_i and \mathbf{p}_i , $i \in \{1, \dots, \tilde{r}\}$.

Output: A reduced-order system given as in (5.6).

- 1: Determine V and W as shown in Theorem 5.1.
- 2: Based on $E(\mathbf{p})$ and $A(\mathbf{p})$ in (5.5), define

$$\mathbb{L}^{(i)} = W^T E^{(i)} V, \quad i \in \{1, \dots, t_e\}, \quad \mathbb{L}_s^{(i)} = W^T A^{(i)} V, \quad i \in \{1, \dots, t_a\}$$

$$(5.8) \quad \left[\mathbb{L}^{(1)}, \dots, \mathbb{L}^{(t_e)}, \mathbb{L}_s^{(1)}, \dots, \mathbb{L}_s^{(t_a)} \right] = Y_1 \Sigma_1 X_1^T$$

$$(5.9) \quad \begin{bmatrix} \mathbb{L}^{(1)} \\ \vdots \\ \mathbb{L}^{(t_e)} \\ \mathbb{L}_s^{(1)} \\ \vdots \\ \mathbb{L}_s^{(t_a)} \end{bmatrix} = Y_2 \Sigma_2 X_2^T.$$

- 3: Define $Y_r := Y_1(:, 1:r)$ and $X_r := X_2(:, 1:r)$.
- 4: Determine the compact projection matrices:
 $V := \text{orth}(V X_r)$ and $W := \text{orth}(W Y_r)$.
- 5: Determine a reduced-order system as shown in (5.6).

has cubic nonlinearity. In the literature, a common approach to reduce such a cubic nonlinear system via system-theoretic MOR is twofold. First, it is to rewrite the cubic system into a QB system by introducing auxiliary variables. Thereafter, one can reduce it by employing a MOR scheme for QB systems such as balanced truncation [10], and interpolation based approaches, e.g., [2, 8, 11]. However, in this process, we lose the original cubic nonlinearity structure in the reduced-order system. On the other hand, the proposed method, in this paper, allows us to reduce a cubic system directly, having preserved the nonlinearity in the reduced-order system.

We set the domain length $L = 1$. The system of equations (6.1) is discretized using a finite-difference method by taking $k = 500$ grid points. Next, we aim at constructing a reduced cubic system by applying Algorithm 3.1. For this purpose, we consider the frequency range $[10^{-3}, 10^3]$. For comparison, we also rewrite the cubic system into the QB form, which results in an equivalent QB system of order $n_{qb} = 2 \cdot 500 = 1000$. We consider the same frequency range in order to employ Algorithm 3.1 to construct a reduced QB system.

First, in Figure 6.1, we observe the decay of the singular values, obtained from the Loewner pencil $(\mathbb{L} - s\mathbb{L}_s)$. As one can expect, the singular values related to the original cubic system decay faster as compared to its equivalent QB form. Hence, for the same order of the reduced-order system, we can anticipate a better quality reduced system. Next, we construct the reduced cubic and QB systems of order $r = 10$. To test the quality of both reduced cubic and QB systems, we perform time-domain simulations using control inputs $u^{(1)}(t) = 10(\sin(\pi t) + 1)$ and $u^{(2)}(t) = 5(te^{-t})$, which

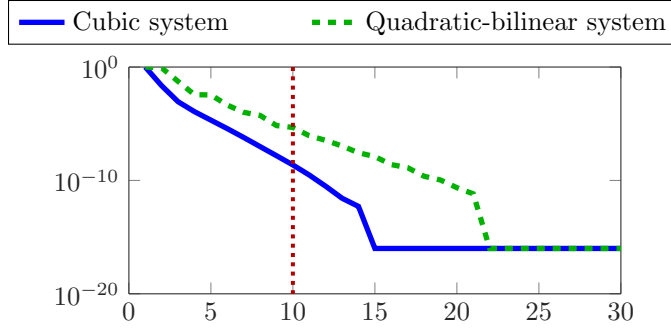


Figure 6.1: Relative decay of singular values based on the Loewner pencils, obtained using the original cubic system and its equivalent transformed QB system.

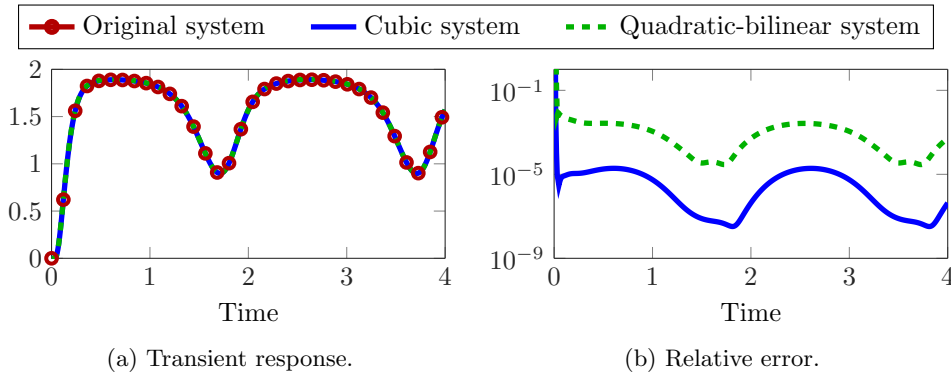


Figure 6.2: Chafee-Infante equation: a comparison of the original and reduced-order systems for the input $u^{(1)} = 10(\sin(\pi t) + 1)$.

are compared in Figures 6.2 and 6.3 by showing the responses and relative errors. As can be seen from these figures, the cubic reduced-order system captures the dynamics of the original system much better as compared to the QB system; precisely, we gain up to 3 orders of magnitude better accuracy using the new method.

6.2. The FitzHugh-Nagumo(FHN) model. As a second non-parametric example, we consider the FHN system, which describes basic neuronal dynamics. This is a coupled cubic nonlinear system, whose governing equations and boundary conditions are as follows:

$$(6.2) \quad \begin{aligned} \epsilon v_t &= \epsilon^2 v_{xx} + v(v - 0.1)(1 - v) - w + q, \\ w_t &= hv - \gamma w + q \end{aligned}$$

with boundary conditions

$$\begin{aligned} v(x, 0) &= 0, & w(x, 0) &= 0, & x &\in (0, L), \\ v_x(0, t) &= i_0(t), & v_x(1, t) &= 0, & t &\geq 0, \end{aligned}$$

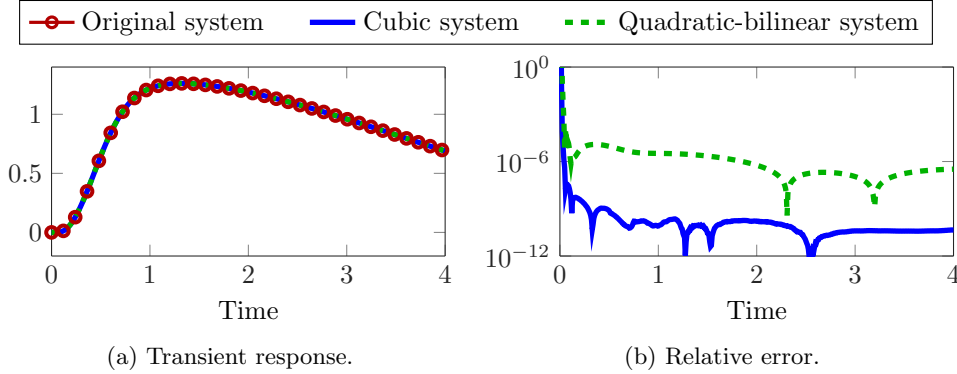


Figure 6.3: Chafee-Infante equation: a comparison of the original and reduced-order systems for the input $u^{(2)} = 5(te^{-t})$.

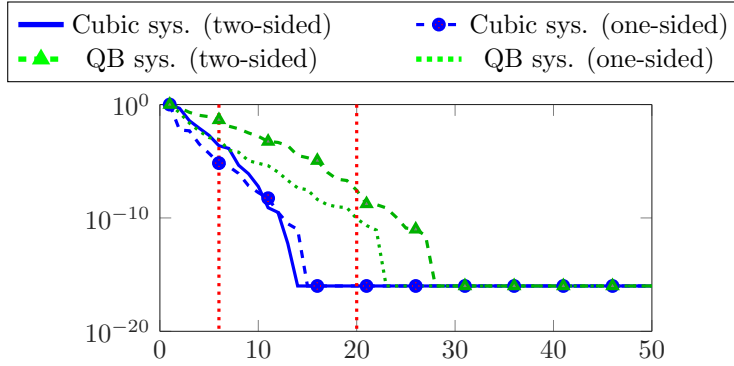


Figure 6.4: FHN model: relative decay of singular values based on the Loewner pencils, obtained via one-sided and two-sided projection of the corresponding systems.

where $h = 0.05$, $\gamma = 2$, $q = 0.05$, $L = 0.1$ and i_0 acts an actuating control input which takes the values $5 \cdot 10^4 t^3 e^{-t}$, and briefly mentioning, the variables v and w denote the activation and de-activation of a neuron, respectively. We discretize the governing equation using a finite difference method, having taken 100 grid points. This leads to a cubic system of order $n = 200$. We use the same output setting as used, e.g., in [11]. The system has two inputs and two outputs, thus, is a MIMO system. The MOR problem related to the FHN system has been considered by several researchers, see, e.g., [10, 11, 13]. Similar to the previous example, system-theoretic MOR of the FHN system has also been considered by first rewriting it into a QB system and employing MOR schemes such as interpolation-based and balanced truncation to reduce it. Thus, we obtain an equivalent QB system of order $n_{qb} = 300$. However, by doing so, we lose the original nonlinear structure.

In order to apply [Algorithm 3.1](#) to obtain reduced-order systems for the original cubic and its equivalent QB systems, we choose 200 points in the frequency range $[10^{-2}, 10^2]$. In [Figure 6.4](#), we first show the relative decay of the singular values for the Loewner pencils (denoted by cubic sys. (two-sided) and QB sys. (two-sided)).

Therein, as expected, we observe that the singular values of the cubic system decay faster as compared to its equivalent QB system. Next, we construct reduced cubic and QB systems of order $r = 20$. To determine the quality of the reduced systems, we perform time-domain simulations and plot the result in [Figure 6.5](#). We observe that the obtained cubic reduced system captures the dynamics of the original system very well, whereas the reduced QB system is unstable. This illustrates a common shortcoming of [Algorithm 3.1](#) (the Loewner approach) that it does not always result in a stable reduced system.

As a remedy, we propose to obtain a reduced-order system using Galerkin (one-sided) projection. For this, we determine the matrix V at Step 1 in [Algorithm 3.1](#) and set $W = V$. This is followed by determining X_r as shown in Step 4 of the algorithm and determine the projection matrix V_{eff} . Subsequently, we set $W_{\text{eff}} = V_{\text{eff}}$ and compute a reduced-order system. As a result, we have a reduced-order system by Galerkin projection instead of Petrov-Galerkin projection. An advantage of doing Galerkin projection is the (local) stability of the reduced-order system in some cases. Next, we compute reduced systems of order $r = 20$ using the cubic and its equivalent QB form, using Galerkin projection.

First, we observe the decay of singular values in [Figure 6.4](#), showing that for Galerkin projection as well, the decay is faster for cubic systems as compared to the equivalent QB systems. Furthermore, we compare the transient response of the reduced-order systems obtained from Galerkin projection in [Figure 6.5](#), which shows that the cubic reduced-order system performs much better as compared to the QB reduced systems. Interestingly, we also observe that the reduced cubic systems, using Petrov-Galerkin and Galerkin projection, tend to perform equally good as the time progresses but in the beginning, the reduced system, obtained using Petrov-Galerkin projection, performs better.

Furthermore, we mention that the same order of accuracy as the reduced QB system (one-sided) of order $r = 20$ can be obtained from the reduced cubic order $r = 6$ only. Surprisingly, we also observe that the typical limit-cycles behavior of the system can be captured by the reduced cubic system of order as low as $r = 2$. On the other hand, the reduced QB could not capture the typical limit cycles behavior below the order $r = 15$. This is a profound observation, which illustrates that keeping the original structure of nonlinearity can lead to much better reduced-order systems.

6.3. Parametric Chafee-Infante equation. As our parametric example, we consider the following parametric Chafee-Infante equation:

$$(6.3) \quad \dot{v}(t) = v_{xx} + v(\mathbf{p} - v^2), \quad x \in (0, L) \times (0, T),$$

where $\mathbf{p} \in [0.25, 2]$. The boundary and initial conditions are the same as given in [Subsection 6.1](#), and the domain length and discretization scheme are also chosen the same as in [Subsection 6.1](#). Next, we aim at constructing a reduced cubic parametric system using [Algorithm 5.1](#). For this, we take 200 points in the frequency range $[10^{-3}, 10^3]$ and the equal number of points for the parameter in the considered interval.

First, in [Figure 6.6](#), we plot the decay of the singular values based on the Loewner pencil, which decays exponentially. Subsequently, we determine a reduced parametric system of order $r = 5$. To compare the quality of the reduced-order system, we simulate for the same inputs as used in [Subsection 6.1](#) and for $\mathbf{p} = \{0.25, 1, 2\}$. We plot the transient response and relative errors in [Figures 6.7](#) and [6.8](#), illustrating that the reduced parametric system can capture the dynamics of the system for different inputs and different parameter.

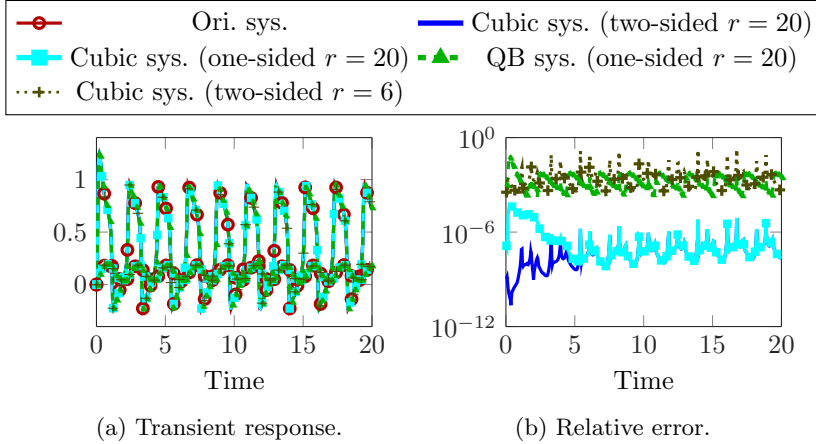


Figure 6.5: FHN model: a comparison of the original and reduced cubic and QB systems using one-sided and two-sided projections, having employed [Algorithm 3.1](#).

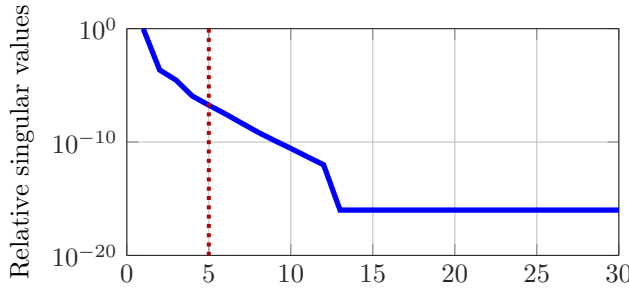


Figure 6.6: Parametric Chafee-Infante equation: relative decay of singular values based on the Loewner pencil.

6.4. Usage of CUR in ROM. In this section, we illustrate the usage of the CUR matrix approximation to further approximate the nonlinear reduced terms. For this, we again consider the Chafee-Infante equation as considered in [Subsection 6.1](#). Using the same setting as shown in [Subsection 6.1](#), we aim at determining reduced cubic systems using one-sided and two-sided projections. First, in [Figure 6.9](#), we plot the relative decay of the singular values based on the Loewner pencil, obtained using the one-sided and two-sided projection matrices. We observe that the singular values based on the two-sided projection decay fast as compared to the one-sided projection.

Next, we construct reduced-order systems of order $r = 10$ using one-sided and two-sided projections using [Algorithm 3.1](#), preserving the cubic nonlinear terms. As discussed in [Section 4](#), we can approximate these terms by making use of CUR matrix approximation. For CUR matrix approximation, we choose 60 rows and 60 columns of \mathcal{V} (defined in (4.6)), which are chosen based on an adaptive sampling proposed in [\[29\]](#). We would like to mention that the number 60 for row and columns is determined based on a trial and error method. An appropriate automatic method for CUR matrix approximation, being suitable for MOR, needs further research. To this end, we have

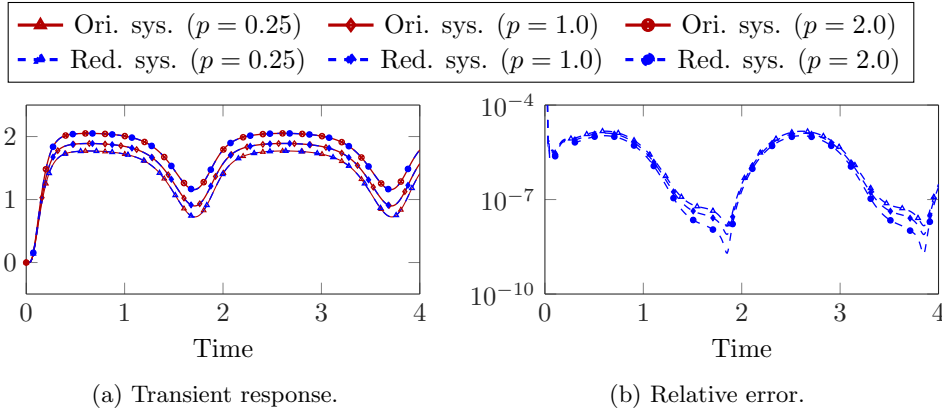


Figure 6.7: Parametric Chafee-Infante equation: a comparison of the original and reduced-order systems for the input $u^{(1)} = 10(\sin(\pi t) + 1)$ and for different parameter values.

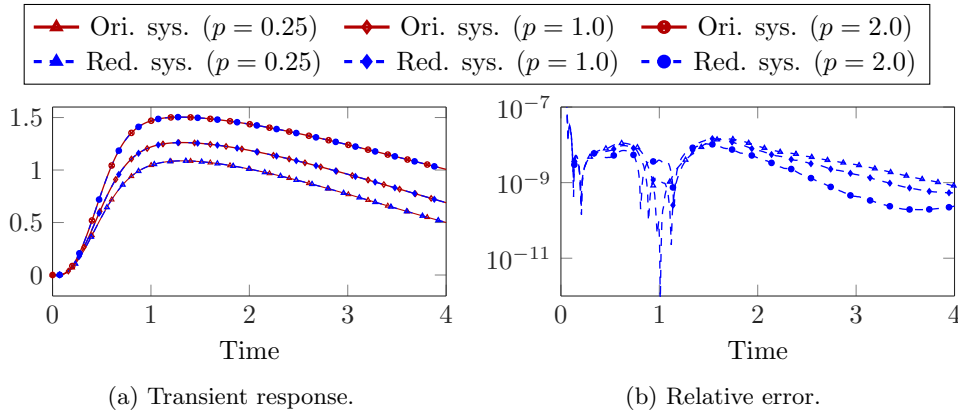


Figure 6.8: Parametric Chafee-Infante equation: a comparison of the original and reduced-order systems for the input $u^{(2)} = 5(e^{-t}t)$ and for different parameter values.

four reduced systems as follows:

- One-sided projection (**OneSProj**)
- One-sided projection, combined with CUR matrix approximation
(**OneSProj + CUR**)
- Two-sided projection (**TwoSProj**)
- Two-sided projection, combined with CUR matrix approximation
(**TwoSProj + CUR**)

To compare the quality of these reduced-order systems, we perform the time-domain simulations of these systems with the original systems for two control inputs, the same as considered in [Subsection 6.1](#). We observe that two-sided projection yields the best reduced-order systems among the four reduced-order systems. Furthermore, when

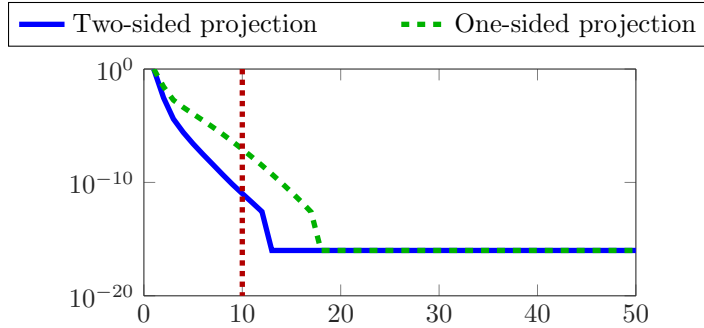


Figure 6.9: Chafee-Infante equation: relative decay of singular values using the Loewner pencil, obtained via one-sided and two-sided projections.

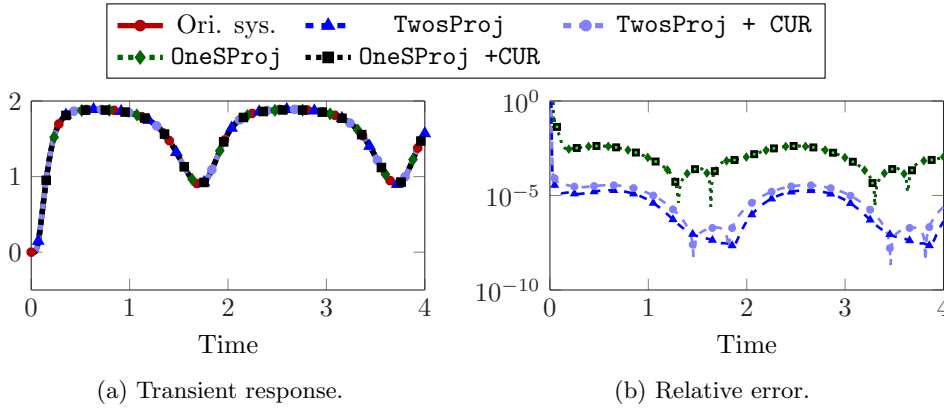


Figure 6.10: Chafee-Infante equation: a comparison of the original and (CUR combined) reduced-order systems for the input $u^{(1)} = 10(\sin(\pi t) + 1)$.

the two-sided reduced-order system is combined with CUR matrix approximation, then we notice that the quality of the reduced-order system decreases a little but still provides a very good approximation of the original system. Interestingly, we also notice that CUR matrix approximation applied to the one-sided reduced-order system also performs really good, where the reduction in quality of OsP is very slim.

7. Conclusions. In this paper, we have discussed the construction of interpolating reduced-order systems for a parametric polynomial system. For this purpose, we have introduced generalized multi-variate transfer functions for the systems. Furthermore, we have proposed algorithms, inspired by the Loewner approach, to generate good quality reduced-order systems in an automatic way. Furthermore, we have discussed related computational issues and also the usage of the CUR matrix approximation in the simulations of reduced systems which may be helpful in some case. We have illustrated the efficiency of the approaches via several numerical experiments, where we have observed that preserving the original structure of the nonlinearity in reduced-order systems can lead to much better reduced-order systems.

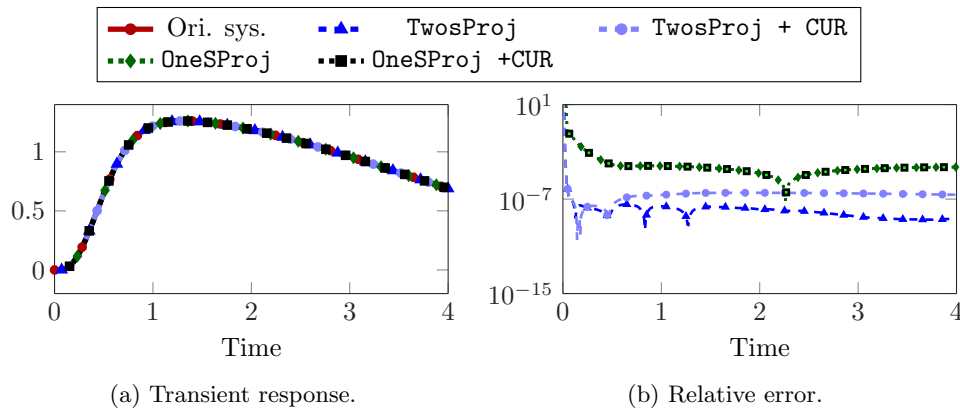


Figure 6.11: Chafee-Infante equation: a comparison of the original and (CUR combined) reduced-order systems for the input $u^{(2)} = 5(e^{-t}t)$.

So far, in our numerical experiments, we have logarithmically or randomly chosen interpolation points in given intervals for frequency and parameters in order to apply Algorithms 3.1 and 5.1. However, choosing these interpolation points wisely, instead of logarithmically or randomly, can ease the computational burden. In this direction, \mathcal{H}_2 -optimal framework [11] and an adaptive choice of interpolation points based on an error estimate [1] can be extended from quadratic-bilinear systems to polynomial systems. Moreover, in Section 4, we have discussed the computational aspect related to $\hat{H}_\xi \hat{x}^\ominus$ which can be eased with the help of CUR matrix approximation. However, we do not take the projection matrix W into account for an approximation of the latter term. Thus, it would be valuable to employ the projection matrix W as well, which could improve the approximation quality. Also, an appropriate choice of CUR matrix approximation for MOR still demands some more investigation. Last but not least, it will be of a great interest to the MOR community to extend the proposed methodology to other classes of nonlinear systems such as rational nonlinear systems, i.e., those systems containing nonlinear functions e.g., $\frac{1}{1+x}$ or $e^{-1/x}$. Such systems, for example, arise, in batch chromatography reactors [19] and reactor models [21]. However, we remark that such systems can be rewritten as a polynomial system by introducing auxiliary variables as discussed in Subsection 2.1, but the goal would be to preserve the original structure of the nonlinearity in the reduced-order systems.

REFERENCES

- [1] M. I. AHMAD, P. BENNER, AND L. FENG, *Interpolatory model reduction for quadratic-bilinear systems using error estimators*, *Engrg. Comput.*, 36 (2018), pp. 25–44, <https://doi.org/10.1108/EC-04-2018-0162>.
- [2] M. I. AHMAD, P. BENNER, AND I. JAIMOUKHA, *Krylov subspace projection methods for model reduction of quadratic-bilinear systems*, *IET Control Theory & Applications*, 10 (2016), pp. 2010–2018, <https://doi.org/10.1049/iet-cta.2016.0415>.
- [3] A. C. ANTOULAS, *Approximation of Large-Scale Dynamical Systems*, vol. 6 of *Adv. Des. Control*, SIAM Publications, Philadelphia, PA, 2005, <https://doi.org/10.1137/1.9780898718713>.
- [4] A. C. ANTOULAS, I. V. GOSEA, AND A. C. IONITA, *Model reduction of bilinear systems in the Loewner framework*, *SIAM J. Sci. Comput.*, 38 (2016), pp. B889–B916.

- [5] A. C. ANTOULAS, S. LEFTERIU, AND A. C. IONITA, *A tutorial introduction to the Loewner framework for model reduction*, in Model Reduction and Approximation: Theory and Algorithms, P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, eds., SIAM, 2017, pp. 335–376, <https://doi.org/10.1137/1.9781611974829.ch8>.
- [6] M. BARRAULT, Y. MADAY, N. C. NGUYEN, AND A. T. PATERA, *An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations*, C. R. Math. Acad. Sci. Paris, 339 (2004), pp. 667–672.
- [7] P. BENNER AND T. BREITEN, *Interpolation-based \mathcal{H}_2 -model reduction of bilinear control systems*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 859–885.
- [8] P. BENNER AND T. BREITEN, *Two-sided projection methods for nonlinear model order reduction*, SIAM J. Sci. Comput., 37 (2015), pp. B239–B260, <https://doi.org/10.1137/14097255X>.
- [9] P. BENNER, A. COHEN, M. OHLBERGER, AND K. WILLCOX, eds., *Model Reduction and Approximation: Theory and Algorithms*, Computational Science & Engineering, SIAM Publications, Philadelphia, PA, 2017, <https://doi.org/10.1137/1.9781611974829>.
- [10] P. BENNER AND P. GOYAL, *Balanced truncation model order reduction for quadratic-bilinear systems*, e-prints 1705.00160, arXiv, 2017, <https://arxiv.org/abs/1705.00160>.
- [11] P. BENNER, P. GOYAL, AND S. GUGERCIN, *\mathcal{H}_2 -quasi-optimal model order reduction for quadratic-bilinear control systems*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 983–1032, <https://doi.org/10.1137/16M1098280>.
- [12] P. BENNER, V. MEHRMANN, AND D. C. SORENSEN, *Dimension Reduction of Large-Scale Systems*, vol. 45 of Lect. Notes Comput. Sci. Eng., Springer-Verlag, Berlin/Heidelberg, Germany, 2005.
- [13] S. CHATURANTABUT AND D. C. SORENSEN, *Nonlinear model reduction via discrete empirical interpolation*, SIAM J. Sci. Comput., 32 (2010), pp. 2737–2764, <https://doi.org/10.1137/090766498>.
- [14] K. GALLIVAN, A. VANDENDORPE, AND P. VAN DOOREN, *Model reduction of MIMO systems via tangential interpolation*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 328–349.
- [15] I. V. GOSEA AND A. C. ANTOULAS, *Data-driven model order reduction of quadratic-bilinear systems*, Numer. Lin. Alg. Appl., 25 (2018), p. e2200, <https://doi.org/10.1002/nla.2200>.
- [16] C. GU, *QLMOR: A projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems*, IEEE Trans. Comput. Aided Des. Integr. Circuits. Syst., 30 (2011), pp. 1307–1320, <https://doi.org/10.1109/TCAD.2011.2142184>.
- [17] M. GUBISCH AND S. VOLKWEIN, *Proper orthogonal decomposition for linear-quadratic optimal control*, in Model Reduction and Approximation: Theory and Algorithms, P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, eds., SIAM, 2017, pp. 3–63, <https://doi.org/10.1137/1.9781611974829.ch1>.
- [18] S. GUGERCIN, A. C. ANTOULAS, AND C. BEATTIE, *\mathcal{H}_2 model reduction for large-scale linear dynamical systems*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 609–638, <https://doi.org/10.1137/060666123>.
- [19] G. GUIOCHON, A. FELINGER, AND D. G. SHIRAZI, *Fundamentals of preparative and nonlinear chromatography*, Academic Press, Boston, 2006.
- [20] T. G. KOLDA, *Multilinear operators for higher-order decompositions*, tech. report, Technical report SAND2006-2081, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, 2006, <https://www.sandia.gov/~tgkolda/pubs/pubfiles/SAND2006-2081.pdf>.
- [21] B. KRAMER AND K. WILLCOX, *Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition*, AIAA J., (2019), <https://doi.org/10.2514/1.J057791>.
- [22] M. W. MAHONEY AND P. DRINEAS, *CUR matrix decompositions for improved data analysis*, Proc. of the National Academy of Sciences, (2009), pp. 697–702.
- [23] A. J. MAYO AND A. C. ANTOULAS, *A framework for the solution of the generalized realization problem*, Linear Algebra Appl., 425 (2007), pp. 634–662.
- [24] G. P. MCCORMICK, *Computability of global solutions to factorable nonconvex programs: Part I-Convex underestimating problems*, Mathematical programming, 10 (1976), pp. 147–175.
- [25] A. QUARTERONI, A. MANZONI, AND F. NEGRI, *Reduced Basis Methods for Partial Differential Equations*, vol. 92 of La Matematica per il 3+2, Springer International Publishing, 2016. ISBN: 978-3-319-15430-5.
- [26] A. C. RODRIGUEZ, S. GUGERCIN, AND J. BORGGAARD, *Interpolatory model reduction of parameterized bilinear dynamical systems*, Adv. Comput. Math., 44 (2018), pp. 1887–1916, <https://doi.org/10.1007/s10444-018-9611-y>.
- [27] W. J. RUGH, *Nonlinear system theory*, Johns Hopkins University Press Baltimore, 1981.
- [28] D. C. SORENSEN AND M. EMBREE, *A DEIM induced CUR factorization*, SIAM J. Sci. Comput., 38 (2016), pp. A1454–A1482.

- [29] S. WANG AND Z. ZHANG, *Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling*, The Journal of Machine Learning Research, 14 (2013), pp. 2729–2769.