SIAM Review
Vol. 67, No. 4, pp. 737–770

On the Loewner Framework, the Kolmogorov Superposition Theorem, and the Curse of Dimensionality[*]

Athanasios C. Antoulas[†]
Ion Victor Gosea[‡]
Charles Poussot-Vassal[§]

Abstract. The Loewner framework is an interpolatory approach for the approximation of linear and nonlinear systems. The purpose here is to extend this framework to linear parametric systems with an arbitrary number n of parameters. To achieve this, a new generalized multivariate rational function realization is proposed. We then introduce the n-dimensional multivariate Loewner matrices and show that they can be computed by solving a set of coupled Sylvester equations. The null space of these Loewner matrices allows the construction of multivariate rational functions in barycentric form. The principal result of this work is to show how the null space of n-dimensional Loewner matrices can be computed using a sequence of one-dimensional Loewner matrices. Thus, a decoupling of the variables is achieved, which leads to a drastic reduction of the computational burden. Equally importantly, this burden is alleviated by avoiding the explicit construction of large-scale n-dimensional Loewner matrices of size $N \times N$. The proposed methodology achieves the decoupling of variables, leading (i) to a reduction in complexity from $\mathcal{O}(N^3)$ to below $\mathcal{O}(N^{1.5})$ when $n > 5$ and (ii) to memory storage bounded by the largest variable dimension rather than their product, thus taming the curse of dimensionality and making the solution scalable to very large data sets. This decoupling of the variables leads to a result similar to the Kolmogorov superposition theorem for rational functions. Thus, making use of barycentric representations, every multivariate rational function can be computed using the composition and superposition of single-variable functions. Finally, we suggest two algorithms (one direct and one iterative) to construct, directly from data, multivariate (or parametric) realizations ensuring (approximate) interpolation. Numerical examples highlight the effectiveness and scalability of the method.

Key words. parameterized systems, Loewner matrix, multivariate functions, barycentric forms, rational interpolation, curse of dimensionality, Kolmogorov superposition theorem (KST)

MSC codes. 93A15, 93A30, 93B11, 93B15, 93C05, 93C80

DOI. 10.1137/24M1656657

[*] Received by the editors April 29, 2024; accepted for publication (in revised form) March 21, 2025; published electronically November 6, 2025.
https://doi.org/10.1137/24M1656657

[†] Department of Electrical and Computer Engineering, Rice University, Houston, TX 77251 USA (aca@rice.edu).

[‡] CSC Group, Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, 39106, Germany (gosea@mpi-magdeburg.mpg.de).

[§] DTIS, ONERA, Université de Toulouse, Toulouse, 31055, France (charles.poussot-vassal@onera.fr).

Contents

1. Introduction.

First, the context, motivation, and problem statement are presented. Since it is the principal mathematical tool of the developed method, a brief historical review of Loewner matrix-driven methods is then given. Finally, the contributions and organization of the paper are listed.

1.1. Motivation and Context: Nonintrusive Data-Driven Model Construction.

Rational model approximation addresses the problem of constructing a reduced-order model that accurately captures the behavior of a potentially large-scale model depending on several variables. In the context of dynamical systems governed by differential and algebraic equations, the multivariate nature comes mainly from the parametric dependence of the underlying system or model. These parameters account for physical characteristics such as mass, length, or material properties (in mechanical systems), flow velocity, temperature (in fluid cases), chemical properties (in biological systems), etc. In engineering applications, the parameters are embedded within the model as tuning variables for the output of interest. The challenges and motivation for dynamical multivariate/parametric reduced order model (pROM) construction stem from three inevitable facts about modern computing and engineers' concerns:

(i) First, accurate modeling often leads to large-scale dynamical systems with complex dynamics for which simulation times and data storage needs become prohibitive, or at least impractical for engineers and practitioners.

(ii) Second, the explicit mathematical model describing the underlying phenomena may not always be accessible, while input-output data may be measured either from a computer-based (black-box) simulator or directly from a physical experiment; as a consequence, the internal variables of the dynamical phenomena are usually too numerous to be stored or simply inaccessible.

(iii) Third, a potentially large number of parameters may be necessary for the following steps of the process.

Complex and accurate parametric models are often needed to perform simulations, forecasting, parametric uncertainty propagation, and optimization in a broad sense. The goals are to better understand and analyze the physics, to tune coefficients, to optimize the system, or to construct parameterized control laws. As these objectives often require a multiquery model-based optimization process, the complexity dictates the accuracy, scalability, and applicability of the approach, and it is relevant to seek a pROM or multivariate surrogate with low complexity.

1.2. Literature Overview on Reduced-Order Modeling.

In the last decade, considerable effort has been dedicated to devising reliable and accurate model reduction (intrusive) and reduced-order modeling (ROM) (nonintrusive) methods, synthesized in a multitude of approaches developed in recent years [3, 10, 26, 5, 11, 12]. For the class of parametric systems, the comprehensive review contribution in [13] provides an exhaustive account of projection-based methods from the 2000s up to the middle of the 2010s.

Additionally, relatively new approaches use time-domain snapshot data to compute reduced-order models, such as operator inference (OpInf) [40] and dynamic mode decomposition (DMD) [45]. Extensions of such methods to parameterized dynamical systems have recently been proposed, for OpInf in [49, 36] and for DMD in [2, 44].

For the class of frequency-domain methods, we concentrate on interpolation-based methods. For other classes of projection-based methods, we refer the reader to the survey [13]. As explained in this review paper, reduced-order models for parametric systems are typically computed by employing projection, using either a local or a global basis for matrix or transfer function interpolation. Relevant contributions in past years include [1, 19, 50, 22]. Additionally, (quasi-)optimal approaches were proposed in [9, 28, 37] that try to impose optimality in certain norms, e.g., the $\mathcal{H}_2 \otimes \mathcal{L}_2$ norm.

Nonintrusive methods based on interpolation or approximate matching (using least squares fitting) of transfer function measurements (of the underlying parameterized rational transfer function) have somewhat proliferated in recent decades, with the following prominent contributions. First, extensions of the Loewner Framework (LF) to multivariate rational approximation by interpolation [7, 29, 47] together with the AAA (Adaptive Antoulas–Anderson) algorithm for multivariate functions [43, 21]. Second, extensions of the vector-fitting framework to multivariate rational approximation, including the generalized Sanathanan–Koerner iteration in [15, 51]; these works are mostly concerned with imposing stability and passivity guarantees for macro model generation in the field of electronics.

1.3. Connection with the Kolmogorov Superposition Theorem.

Problem 119 in the book of Polya and Szegö [41] asks the following question: *Are there actually functions of three variables?* Alternatively, is it possible to use compositions of functions of two or fewer variables to express any function of three variables? This question is related to Hilbert's 13th problem [27]: are there any genuinely continuous multivariate functions? Hilbert, in fact, conjectured the existence of a three-variable continuous function that cannot be expressed in terms of the composition and addition of two-variable continuous functions. For a recent overview of this problem, see [38]. The Kolmogorov superposition theorem (KST) answers this question negatively. It shows that continuous functions of several variables can be expressed as the composition and superposition of functions of one variable. Thus, there are no *true* functions of three variables. The present contribution presents connections between the Loewner framework and the KST restricted to rational functions. As a by-product, *taming of the curse of dimensionality (C-o-D),* in terms of computational complexity, storage, and, last but not least, numerical accuracy, is achieved.

1.4. Connection to Other Fields.

Tensors are generalizations of vectors and matrices in multiple dimensions. Applications include, among others, the fields of signal processing (e.g., array processing), scientific computing (e.g., multivariate function discretization), and, more recently, quantum computing (e.g., simulation of quantum many-body problems). We refer the reader to the survey [31] for additional information and a detailed discussion. However, working explicitly with tensors, especially those of higher dimensions, is not a trivial task. The number of elements in a tensor increases exponentially with the number of dimensions, as do the computational/memory requirements. Such exponential dependence, together with the challenges that arise from it, are connected to the **C-o-D**.

Tensor decompositions are particularly important and relevant for several strenuous computational tasks since they can potentially alleviate the C-o-D that occurs when working with high-dimensional tensors, as is explained in [46]. Such a decomposition can accurately represent and substitute the tensor, i.e., one may use it instead of explicitly using the original tensor itself. More details and an extensive literature survey of low-rank tensor approximation techniques, including canonical polyadic decomposition, Tucker decomposition, low multilinear rank approximation, and tensor trains and networks, can be found in [23].

Tensorization and Loewner matrices were recently connected in the contribution [17]. There, a collection of one-dimensional (1-D) (standard) Loewner matrices is reshaped as a three-dimensional (3-D) tensor, to which the block term decomposition (BTD) is applied; the procedure is named "Loewnerization." The application of interest is blind signal separation.

Nonlinear eigenvalue problems (NEPs) can be viewed as a generalization of the (ordinary) eigenvalue problem to equations that depend nonlinearly on the parameters. Linearization techniques allow the reformulation of any polynomial eigenvalue problem as a larger linear eigenvalue problem and then the application of established (classical) algorithms to solve it. Other linearizations involve rational approximation, e.g., [32, 25], which involve the usage of the rational Krylov or the AAA algorithms, together with [16], which uses the Loewner and Hankel frameworks in the context of contour integrals.

1.5. Problem Statement.

A linear-in-state dynamical system parameterized in terms of the parameters of $\mathcal{S} = [{}^2s, \ldots, {}^ns]^\top \subset \mathbb{C}^{n-1}$ is characterized in state-space representation by the equations

$$\mathfrak{E}(\mathcal{S})\dot{\mathbf{x}}(t;\mathcal{S}) = \mathfrak{A}(\mathcal{S})\mathbf{x}(t;\mathcal{S}) + \mathfrak{B}(\mathcal{S})\mathbf{u}(t), \;\; \mathbf{y}(t;\mathcal{S}) = \mathfrak{C}(\mathcal{S})\mathbf{x}(t;\mathcal{S}), \tag{1.1}$$

where $\dot{\mathbf{x}}(t;\mathcal{S})$ refers to the derivative of $\mathbf{x}(t;\mathcal{S}) \in \mathbb{R}^M$ with respect to the time variable t. Additionally, the n_u control inputs are collected in the vector $\mathbf{u}(t) \in \mathbb{R}^{n_u}$, while the n_y outputs are observed in the vector $\mathbf{y}(t;\mathcal{S}) \in \mathbb{R}^{n_y}$. Finally, the dimensions of the system matrices appearing in the state-space realization (1.1) are as follows: $\mathfrak{E}(\mathcal{S}), \mathfrak{A}(\mathcal{S}) \in \mathbb{R}^{M\times M}$, $\mathfrak{B}(\mathcal{S}) \in \mathbb{R}^{M\times n_u}$, $\mathfrak{C}(\mathcal{S}) \in \mathbb{R}^{n_y\times M}$. For simplicity of exposition, we consider only the single-input and single-output (SISO) scenario in what follows, i.e., $n_u = n_y = 1$. The extension to multi-input multioutput (MIMO) systems will be the topic of future research, e.g., based on the formulation exposed in [47]. In what follows, particular attention is paid to the exposition of a solution that **tames** the **C-o-D**.

Remark 1.1 (taming the **C-o-D**). Throughout this work, the expression "taming the C-o-D" will be used to emphasize the decoupling of variables, which drastically reduces both the complexity of computation of barycentric weights in terms of flop, and the memory storage requirements, while at the same time improving numerical accuracy.

Transforming the differential equation in (1.1) using the unilateral Laplace transform, the time variable t becomes ${}^1s$, and solving for the transformed state variable, we have $\mathfrak{X}({}^1s;\mathcal{S}) = [{}^1s\mathfrak{E}(\mathcal{S}) - \mathfrak{A}(\mathcal{S})]^{-1}\mathfrak{B}(\mathcal{S})\mathfrak{U}({}^1s)$. Similarly, transforming the second equation in (1.1) we obtain $\mathfrak{Y}({}^1s;\mathcal{S}) = \mathfrak{C}(\mathcal{S})\mathfrak{X}({}^1s;\mathcal{S})$. These equations yield $\mathfrak{Y}({}^1s;\mathcal{S}) = \mathfrak{C}(\mathcal{S})[{}^1s\mathfrak{E}(\mathcal{S}) - \mathfrak{A}(\mathcal{S})]^{-1}\mathfrak{B}(\mathcal{S})\mathfrak{U}({}^1s)$. The transfer function of the parametric linear time-invariant (pLTI) system in (1.1) is given by

$$\mathfrak{H}({}^1s, {}^2s, \ldots, {}^ns) = \mathfrak{C}(\mathcal{S})\left[{}^1s\mathfrak{E}(\mathcal{S}) - \mathfrak{A}(\mathcal{S})\right]^{-1}\mathfrak{B}(\mathcal{S}) \in \mathbb{C}. \tag{1.2}$$

This is a multivariate rational function involving n variables ${}^ls \in \mathbb{C}$, $l = 1, \ldots, n$.

We denote the complexity of each variable ${}^{l}s$ with the value d_l (the highest degree in which the variable occurs in both polynomials describing the rational function shown above) and say that $\mathfrak{H}({}^{1}s, {}^{2}s, \ldots, {}^{n}s)$ in (1.2) is of complexity $(d_1, d_2, \ldots, d_n)$.

As we are interested in the nonintrusive data-driven setup, let us now consider that the function in (1.2) is not explicitly known. Instead, one has access to evaluations at (support or interpolatory) points ${}^{1}\lambda_{j_1}, {}^{2}\lambda_{j_2}, \ldots, {}^{n}\lambda_{j_n}$ along ${}^{1}s, {}^{2}s, \ldots {}^{n}s$, leading to measurements $\mathbf{w}_{j_1,j_2,\ldots,j_n}$, for $j_l = 1, \ldots, k_l$, where $l = 1, \ldots, n$.

Under some assumptions detailed in what follows, we seek a reduced multivariate rational model, pROM, $\mathbf{H}$, given as

$$\mathbf{H}({}^{1}s, {}^{2}s, \ldots, {}^{n}s) = \mathbf{C}\mathbf{\Phi}({}^{1}s, {}^{2}s, \ldots, {}^{n}s)^{-1}\mathbf{G} \in \mathbb{C}, \tag{1.3}$$

where the vectors $\mathbf{C}^{\top}, \mathbf{G} \in \mathbb{C}^{m}$, and square matrix $\mathbf{\Phi} \in \mathbb{C}^{m\times m}$ define a generalized realization, detailed later. We denote this realization with the triple $(\mathbf{C}, \mathbf{\Phi}, \mathbf{G})$, being the output, the inverse of the resolvent, and the input operators.

In what follows, we concentrate on continuous-time dynamical systems. Therefore, the first variable ${}^{1}s$ will be associated with the dynamic Laplace variable, while ${}^{2}s, \ldots, {}^{n}s$ represent nondynamic parametric variables (in most cases they will be real valued, although a complex form is also possible). Note that a similar discrete sampled-time model can be obtained using the z-transform (see, e.g., [48]). In addition, one might also notice that (1.2) may be any multivariate real- or complex-valued function.

1.6. Historical Notes. The Loewner matrix was introduced by Karl Löwner in his seminal paper published nine decades ago [34] for the study of matrix convexity. It has been further studied and used in multiple works dealing with data-driven rational function approximation with applications in system theory in general. In [4], the Loewner matrix constructed from data is used to compute the barycentric coefficients to obtain the rational approximating function in the Lagrange basis. This is also known as the one-sided Loewner framework. One major update was proposed in 2007 by [35], which introduced the two-sided Loewner framework, constructing a rational model with minimal McMillan degree and also constructing a realization with minimal order, directly from the data. [8] provides a comprehensive review of the case of single-variable linear systems, gathering most of the results up to 2017. In [7], the one-sided framework was extended to two variables/parameters, and its corresponding Lagrange basis realization was derived. Later, in [29], the multiparameter Loewner framework (mpLF) was presented together with (for up to three parameters) the barycentric form, but without the description of a realization. Recently, tutorial contributions for the Loewner framework, with its extensions and applications, were proposed in [8, 30]. [20] provides a comprehensive overview including parametric and nonlinear Loewner extensions, practical applications, and test cases from aerospace engineering and fluid dynamics.

The AAA algorithm in [39] represents an iterative and adaptive version of the method in [4] that makes use of the barycentric representation of rational interpolants. For more details on barycentric forms and connections to Lagrange interpolation, we refer the reader to [14]. In [43], the parametric AAA (p-AAA) algorithm was introduced. This extends the original AAA formulation of [39] to multivariate problems appearing in the modeling of parametric dynamical systems. The p-AAA can be viewed as an adaptation of the mpLF, in that it also uses multidimensional Loewner matrices and computes barycentric forms of the fitted rational functions. The p-AAA algorithm chooses the interpolation points in a greedy manner and enriches the Lagrange bases until an approximation (with desired accuracy) is reached.

In addition, multiple application-oriented research papers utilizing the Loewner framework have been suggested, as well as multiple adaptations of the original version. It is worth noting that the multivariate versions have been poorly studied, and when they were studied, they were limited to three variables. In this paper, we address these two points.

1.7. Contribution and Paper Organization. Our goal is to provide a complete and scalable solution to the data-driven multivariate reduced-order model construction. The results provided in [7, 29] are extended. The main result consists of the decoupling of variables, thus taming the C-o-D. The contribution is fivefold:

(i) We propose a multivariate generalized realization that allows the description with state-space form (with limited complexity) of any multivariate rational function (section 2 and Theorem 2.8).

(ii) The n-dimensional (n-D) multivariate Loewner matrix is introduced and is shown to be the solution of a set of coupled Sylvester equations (section 4 and Theorem 4.13).

(iii) As the dimension N of the n-D Loewner matrix exponentially increases with the amount of data (i.e., variables and associated degrees), we demonstrate that the associated null space can be obtained using a collection of 1-D Loewner matrices; this leads to the reduction of computational complexity from $\mathcal{O}(N^3)$ to less than $\mathcal{O}(N^{1.5})$ when $n > 5$, to a drastic reduction in storage requirements (section 5 and Theorems 5.8, 5.10, and 5.13), and to increased numerical accuracy.

(iv) A connection with Hilbert's 13th problem and the KST is established (first with Theorem 5.9 and then in section 6).

(v) Two data-driven multivariate generalized model construction algorithms (section 7 and Algorithms 7.1 and 7.2) are provided.

Among these contributions, items (i), (iii), and (iv) are the main theoretical results toward *taming the C-o-D* for data-driven multivariate function and realization construction. More specifically, item (i) provides a new realization structure applicable to any n-D rational function expressed in the Lagrange basis, where the complexity (e.g., dimension of matrices) is controlled. Item (iii) shifts the problem of null space computation of a large-scale n-D Loewner matrix to the null space computation of a set of small-scale 1-D Loewner matrices, leading to the very same Lagrange coefficients required in the pROM construction, but with much lower computational effort. Finally, item (iv) links this result to the KST by explicitly detailing the decoupling of variables.

Remark 1.2 (connection to tensors). Stepping back from the dynamical systems perspective, we also note that the proposed approach provides a candidate solution to tensor approximation problems. Indeed, we approximate any problem characterized by tensorized data sets by means of a rational function. This is done by taming the **C-o-D** as pointed out in (iii). Established tensor decompositions may provide a bridge to the philosophy of our proposed method, which requires breaking down the complex problem by eliminating one dimension at every step.

Remark 1.3 (connections to NEPs). The realization proposed addresses the problem of linearization in the context of NEPs. Specifically, our realization achieves multilinearizations of the associated NEPs. Furthermore, in the bivariate case, if we split the two variables, we achieve a linearization. In the case of more than two variables, if we arrange them as the frequency variable ${}^1s$ in the first group (or right variable) and all the other variables (parameters) in the second group (left variables), we achieve a linearization in ${}^1s$.

The remainder of the paper is organized as follows. Section 2 provides the starting point and initial seed by introducing a *generalized multivariate rational functions realization* framework. From this form, a specific structure appropriate to the problem treated here is chosen. Since *data* are the main ingredient of the data-driven framework used, section 3 introduces the data notations in a general n-variable case. Then, in section 4, the data-based *n-D Loewner matrices* are defined and a connection with cascaded Sylvester equations is made. The relationships with the multivariate barycentric rational form (using a Lagrange basis), as well as the multivariate realization, are also established. In section 5, the numerical complexity induced by the n-D null space computation is reduced thanks to the decomposition into a recursive set of 1-D Loewner matrix null space computations instead. This decomposition allows a drastic reduction of the complexity, thus *taming the C-o-D.* Finally, section 6 details the connection with the KST. Based on all these contributions, two algorithms are sketched in section 7 that indicate complete procedures for the construction of a nonintrusive multivariate dynamical model realization from input-output data. Numerical examples that illustrate the effectiveness of the proposed process are described in section 8.[1] Finally, section 9 concludes the paper and provides an outlook on addressing open issues and future research problems.

2. Realizations of Multivariate Rational Functions.

The starting point of this study is the new generalized realization for multivariate rational functions. This leads to the construction of a realization involving internal variable equations from an n-variable transfer function in the form (1.2). This is expressed in the Lagrange basis. After some preliminaries, the result is stated in Theorem 2.8. This stands as the first major contribution of this paper.

2.1. Preliminaries.

First, we derive the pseudocompanion Lagrange basis, then we provide the multirow and multicolumn indices, the coefficient matrices propositions, and, finally, results on the characteristic polynomial.

2.1.1. Pseudocompanion Lagrange Matrix.

Consider a rational function $\mathfrak{H}$ in n variables, namely, ${}^js$, each of degree d_j $(j=1,\dots,n)$, as in (1.2). We will consider the *Lagrange basis* of polynomials. The *Lagrange pseudocompanion matrix* considered here is denoted ${}^j\mathbb{X}^{\rm Lag}$ and is defined as follows.

Definition 2.1. *Let the Lagrange monomials in the variable ${}^js$ be denoted as ${}^j\mathbf{x}_i = {}^js - {}^j\lambda_i$, where $i=1,\dots,n_j$ and ${}^j\lambda_i \in \mathbb{C}$. Associated with the jth variable, we define the pseudocompanion form matrix in the Lagrange basis as*

$$
{}^j\mathbb{X}^{\rm Lag} = \left[\begin{array}{c} \mathbf{X}^{\rm Lag}({}^js) \\ \hline {}^j\mathbf{q}^{\rm Lag} \end{array}\right] = \left[\begin{array}{ccccc} {}^j\mathbf{x}_1 & -{}^j\mathbf{x}_2 & 0 & \cdots & 0 \\ {}^j\mathbf{x}_1 & 0 & -{}^j\mathbf{x}_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ {}^j\mathbf{x}_1 & 0 & \cdots & 0 & -{}^j\mathbf{x}_{n_j} \\ \hline {}^jq_1 & {}^jq_2 & \cdots & {}^jq_{n_j-1} & {}^jq_{n_j} \end{array}\right] \in \mathbb{C}^{n_j\times n_j}[{}^js], \tag{2.1}
$$

with values ${}^jq_i$, $i=1,\dots,n_j$, chosen such that ${}^j\mathbb{X}^{\rm Lag}$ is unimodular, i.e., $\det({}^j\mathbb{X}^{\rm Lag}) = 1$.[2]

[1] An exhaustive account of numerical examples and results, together with all necessary data and codes to reproduce the numerics, can be found in the links to the Supplementary Material at the end of this paper.

[2] One may chose $1/{}^jq_i = \Pi_{k\neq i}(s_i - {}^j\lambda_k)$ for $k=1,\dots,n_j$.

Following the general interpolation framework, the ${}^js$ $(j=1,\ldots,n)$ variables of $\mathfrak{H}$ (1.2) are split into *left* and *right* variables, or equivalently into *row* and *column* variables. For simplicity of exposition (and by permutation if necessary), we assume that ${}^1s,\ldots,{}^ks$ are the column (right) variables and ${}^{k+1}s,\ldots,{}^ns$ are the row (left) variables ($0<k<n$, $k\in\mathbb{N}$). Based on this data, we define two *Kronecker products* of the associated pseudocompanion matrices.

Definition 2.2. *Consider the column* ${}^1s,\ldots,{}^ks$ *and row* ${}^{k+1}s,\ldots,{}^ns$ *variables; we define the Kronecker products of the pseudocompanion matrices* (2.1) *as*

$$\begin{aligned}\mathbf{\Gamma}^{\rm Lag} &= {}^1\mathbb{X}^{\rm Lag}\otimes{}^2\mathbb{X}^{\rm Lag}\otimes\cdots\otimes{}^k\mathbb{X}^{\rm Lag}\in\mathbb{C}^{\kappa\times\kappa}[{}^1s,\ldots,{}^ks],\\ \mathbf{\Delta}^{\rm Lag} &= {}^{k+1}\mathbb{X}^{\rm Lag}\otimes{}^{k+2}\mathbb{X}^{\rm Lag}\otimes\cdots\otimes{}^n\mathbb{X}^{\rm Lag}\in\mathbb{C}^{\ell\times\ell}[{}^{k+1}s,\ldots,{}^ns],\end{aligned}\tag{2.2}$$

where $\kappa=\prod_{j=1}^k n_j$ *and* $\ell=\prod_{j=k+1}^n n_j$. *These matrices are square and unimodular. For brevity, we will now denote them as* $\mathbf{\Gamma}$ *and* $\mathbf{\Delta}$.

2.1.2. Multirow/Multicolumn Indices and the Coefficient Matrices.

We will show how to set up the matrices containing the coefficients of the numerator and denominator polynomials. The key to this goal is an appropriate definition of row/column multi-indices.

Definition 2.3. *Each column of* $\mathbf{\Gamma}$ *and each column of* $\mathbf{\Delta}$ *defines a unique multi-index* I_q, J_r. *We will refer to these indices as* row *and* column multi-indices *(the latter because the* $\mathbf{\Delta}$ *matrix enters in transposed form), as follows:*

$$I_q=\left[i^q_{k+1},\,i^q_{k+2},\,\ldots,\,i^q_n\right]\ ,\ J_r=\left[j^r_1,\,j^r_2,\,\ldots,\,j^r_k\right]\ ,\ q=1,\ldots,\ell,\ r=1,\ldots,\kappa.$$

Each multi-index I_q (J_r) *contains the indices of the Lagrange monomials involved in the qth (rth) column of* $\mathbf{\Delta}$ $(\mathbf{\Gamma})$, *respectively.*

Remark 2.4. The *ordering* of these multi-indices is imposed by the ordering of the Kronecker products in Definition 2.2. More details are available in the examples.

2.1.3. The Coefficient Matrices.

We consider the rational function $\mathbf{H}$ as

$$\mathbf{H}({}^1s,{}^2s,\ldots,{}^ns)=\frac{\sum_{j_1=1}^{k_1}\sum_{j_2=1}^{k_2}\cdots\sum_{j_n=1}^{k_n}\frac{c_{j_1,j_2,\ldots,j_n}\mathbf{w}_{j_1,j_2,\ldots,j_n}}{({}^1s-{}^1\lambda_{j_1})({}^2s-{}^2\lambda_{j_2})\cdots({}^ns-{}^n\lambda_{j_n})}}{\sum_{j_1=1}^{k_1}\sum_{j_2=1}^{k_2}\cdots\sum_{j_n=1}^{k_n}\frac{c_{j_1,j_2,\ldots,j_n}}{({}^1s-{}^1\lambda_{j_1})({}^2s-{}^2\lambda_{j_2})\cdots({}^ns-{}^n\lambda_{j_n})}},\tag{2.3}$$

where $c_{j_1,j_2,\ldots,j_n}\in\mathbb{C}$ are the barycentric weights and $\mathbf{w}_{j_1,j_2,\ldots,j_n}\in\mathbb{C}$ the data evaluated at the combination of interpolation (support) points in $\{{}^1\lambda_{j_1},{}^2\lambda_{j_2},\ldots,{}^n\lambda_{j_n}\}$, or, equivalently, following Definition 2.3, as

$$\mathbf{H}({}^1s,{}^2s,\ldots,{}^ns)=\frac{\sum_{q=1}^{\ell}\sum_{r=1}^{\kappa}\frac{\beta_{I_q,J_r}}{\prod_{a\in I_q}\prod_{b\in J_r}({}^as-{}^a\lambda_{j_a})({}^bs-{}^b\lambda_{j_b})}}{\sum_{q=1}^{\ell}\sum_{r=1}^{\kappa}\frac{\alpha_{I_q,J_r}}{\prod_{a\in I_q}\prod_{b\in J_r}({}^as-{}^a\lambda_{j_a})({}^bs-{}^b\lambda_{j_b})}}.$$

We now define matrices of size $\ell\times\kappa$:

$$\mathbb{A}^{\rm Lag}=\begin{bmatrix}\alpha_{I_1,J_1}&\alpha_{I_1,J_2}&\cdots&a_{I_1,J_\kappa}\\ \alpha_{I_2,J_1}&\alpha_{I_2,J_2}&\cdots&\alpha_{I_2,J_\kappa}\\ \vdots&\vdots&\ddots&\vdots\\ \alpha_{I_\ell,J_1}&\alpha_{I_\ell,J_2}&\cdots&\alpha_{I_\ell,J_\kappa}\end{bmatrix},\ \mathbb{B}^{\rm Lag}=\begin{bmatrix}\beta_{I_1,J_1}&\beta_{I_1,J_2}&\cdots&\beta_{I_1,J_\kappa}\\ \beta_{I_2,J_1}&\beta_{I_2,J_2}&\cdots&\beta_{I_2,J_\kappa}\\ \vdots&\vdots&\ddots&\vdots\\ \beta_{I_\ell,J_1}&\beta_{I_\ell,J_2}&\cdots&\beta_{I_\ell,J_\kappa}\end{bmatrix}.\tag{2.4}$$

Notice that $\mathbb{A}^{\rm Lag}$ contains the appropriately arranged barycentric weights of the denominator of $\mathbf{H}$ (i.e., the entries of a vector in the null space of the associated Loewner matrix), while $\mathbb{B}^{\rm Lag}$, contains the barycentric weights of the numerator, i.e., the product of the denominator barycentric weights with the corresponding values of $\mathbf{H}$.

2.1.4. Characteristic Polynomial in the Barycentric Representation. We consider the single-variable polynomial $\mathbf{p}(s)$ of degree (at most) n in the variable s. For a *barycentric* or *Lagrange representation,* the following holds (by expanding the determinant of $\mathbf{M}$ with respect to the last row).

PROPOSITION 2.5. *Given the polynomial* $\mathbf{p}(s)$ *of degree less than or equal to* n*, expressed in a Lagrange basis as* $\mathbf{p}(s)=\pi\big(\frac{\alpha_1}{\mathbf{x}_1}+\cdots+\frac{\alpha_{n+1}}{\mathbf{x}_{n+1}}\big)$*, where* $\pi=\prod_{i=1}^{n+1}\mathbf{x}_i$*, it follows that* $\det(\mathbf{M})=\sum_{i=1}^{n+1}\alpha_i\prod_{j\neq i}\mathbf{x}_j=\mathbf{p}(s)$*, where* $\mathbf{M}$ *is the pseudocompanion form matrix as in Definition* 2.1*, where* ${}^j\mathbf{x}_i$ *is replaced by* $\mathbf{x}_j$ *and* ${}^jq_1$ *by* α_j*.*

Next, following Proposition 2.5, we consider two-variable polynomials $\mathbf{p}(s,t)$ of degree (at most) n, m in the variables s, t, respectively. Let $\mathbf{x}_i(s)=s-s_i$, $s_i\in\mathbb{C}$, $i=1,\ldots,n+1$, and $\mathbf{y}_j(t)=t-t_j$, $t_j\in\mathbb{C}$, $j=1,\ldots,m+1$, be the monomials constituting a Lagrange basis for two-variable polynomials of degree less than or equal to n, m, respectively. In other words,

$$\mathbf{p}(s,t)=\pi\left[\frac{\alpha_{1,1}}{\mathbf{x}_1\mathbf{y}_1}+\cdots+\frac{\alpha_{1,m+1}}{\mathbf{x}_1\mathbf{y}_{m+1}}+\cdots+\frac{\alpha_{n+1,1}}{\mathbf{x}_{n+1}\mathbf{y}_1}+\cdots+\frac{\alpha_{n+1,m+1}}{\mathbf{x}_{n+1}\mathbf{y}_{m+1}}\right],$$

which can be rewritten by highlighting the matrix form of (2.4) as

$$\mathbf{p}(s,t)=\pi\left[\frac{1}{\mathbf{x}_1},\frac{1}{\mathbf{x}_2},\ldots,\frac{1}{\mathbf{x}_{n+1}}\right]\begin{bmatrix}\alpha_{1,1}&\alpha_{1,2}&\cdots&\alpha_{1,m+1}\\\alpha_{2,1}&\alpha_{2,2}&\cdots&\alpha_{2,m+1}\\\vdots&\vdots&\ddots&\vdots\\\alpha_{n+1,1}&\alpha_{n+1,2}&\cdots&\alpha_{n+1,m+1}\end{bmatrix}\begin{bmatrix}\frac{1}{\mathbf{y}_1}\\\frac{1}{\mathbf{y}_1}\\\vdots\\\frac{1}{\mathbf{y}_{m+1}}\end{bmatrix},$$

where $\pi=\prod_{i=1}^{n+1}\mathbf{x}_i\prod_{j=1}^{m+1}\mathbf{y}_j$. Consider next the pseudocompanion form matrices,

(2.5)
$$\mathbf{S}=\underbrace{\left[\begin{array}{ccccc}\mathbf{x}_1&-\mathbf{x}_2&0&\cdots&0\\\mathbf{x}_1&0&-\mathbf{x}_3&\cdots&0\\\vdots&\vdots&\ddots&\vdots&\vdots\\\mathbf{x}_1&0&\cdots&0&-\mathbf{x}_{n+1}\\\hline\epsilon_1&\epsilon_2&\cdots&\epsilon_n&\epsilon_{n+1}\end{array}\right]}_{\in\mathbb{C}^{(n+1)\times(n+1)}[s]},\ \mathbf{T}=\underbrace{\left[\begin{array}{ccccc}\mathbf{y}_1&-\mathbf{y}_2&0&\cdots&0\\\mathbf{y}_1&0&-\mathbf{y}_3&\cdots&0\\\vdots&\vdots&\ddots&\vdots&\vdots\\\mathbf{y}_1&0&\cdots&0&-\mathbf{y}_{m+1}\\\hline\zeta_1&\zeta_2&\cdots&\zeta_m&\zeta_{m+1}\end{array}\right]}_{\in\mathbb{C}^{(m+1)\times(m+1)}[t]},$$

where the constants ϵ_i and ζ_j are chosen such that $\det(\mathbf{S})=1$ and $\det(\mathbf{T})=1$.[3] The coefficients $\alpha_{i,j}$ are arranged in the form of a matrix $\mathbb{A}^{\rm Lag}\in\mathbb{C}^{(n+1)\times(m+1)}$ as in (2.4),

$$\mathbb{A}^{\rm Lag}=\begin{bmatrix}\alpha_{1,1}&\alpha_{1,2}&\cdots&\alpha_{1,m+1}\\\alpha_{2,1}&\alpha_{2,2}&\cdots&\alpha_{2,m+1}\\\vdots&\vdots&\ddots&\vdots\\\alpha_{n+1,1}&\alpha_{n+1,2}&\cdots&\alpha_{n+1,m+1}\end{bmatrix},$$

[3] One may chose $1/\epsilon_i=\Pi_{j\neq i}(s_i-s_j)$ and $1/\zeta_i=\Pi_{j\neq i}(t_i-t_j)$ for $i,j=1,\ldots,n,m$.

or in a vectorized version (taken rowwise) $\mathrm{vec}\,(\mathbb{A}^{\mathrm{Lag}}) \in \mathbb{C}^{1\times\kappa}$ such that $\mathrm{vec}\,(\mathbb{A}^{\mathrm{Lag}}) = [\alpha_{1,1}, \alpha_{1,2}, \ldots, \alpha_{1,m+1} \mid \cdots \mid \alpha_{n+1,1}, \ldots, \alpha_{n+1,m+1}]$, where $\kappa = (n+1)(m+1)$. Consider also the Kronecker product $\mathbf{S}\otimes\mathbf{T}$, which turns out to be a square polynomial matrix of size κ. We form two matrices

$$\mathbf{M}_1 = \underbrace{\begin{bmatrix} (\mathbf{S}\otimes\mathbf{T})(1{:}\kappa-1,:) \\ \mathrm{vec}\,(\mathbb{A}^{\mathrm{Lag}}) \end{bmatrix}}_{\in\mathbb{C}^{\kappa\times\kappa}[s,t]} \text{ and } \mathbf{M}_2 = \underbrace{\begin{bmatrix} \mathbf{S}(1{:}n-1,:) & \mathbf{0}_{n-1,m-1} \\ \mathbb{A}^{\mathrm{Lag}} & \mathbf{T}(1{:}m-1,:)^\top \end{bmatrix}}_{\in\mathbb{C}^{(n+m-1)\times(n+m-1)}[s,t]}. \tag{2.6}$$

Proposition 2.6. *The determinants of* $\mathbf{M}_1$ *and* $\mathbf{M}_2$ *are both equal to* $\mathbf{p}(s,t)$.

Proof. The first expression follows by expanding the determinant of $\mathbf{M}_1$ with respect to the last row. For the validity of the second expression, see Theorem 2.8. □

Remark 2.7 (C-o-D). This result shows that by splitting the variables into left and right variables, the **C-o-D** is alleviated, as in the former case the dimension is $(n+1)(m+1)$, while in the latter the dimension is $n+m-1$.

2.2. The Multivariate Realization in the Lagrange Basis.

2.2.1. Main Result.

The result provided in Theorem 2.8 yields a systematic way to construct a realization as in (1.3) from a transfer function $\mathbf{H}$ given in a barycentric / Lagrange form (2.3).

Theorem 2.8. *Given quantities in Definition* 2.1 *and Definition* 2.2*, a* $2\ell+\kappa-1 = m$*th-order realization* $(\mathbf{C},\mathbf{\Phi},\mathbf{G})$ *of the multivariate function* $\mathfrak{H}$ *in* (1.2)*, in barycentric form* (2.3)*, satisfying* $\mathbf{H}({}^1s,\ldots,{}^ns) = \mathbf{C}\mathbf{\Phi}({}^1s,\ldots,{}^ns)^{-1}\mathbf{G}$*, is given by*

(2.7)

$$\mathbf{\Phi}({}^1s,\ldots,{}^ns) = \left[\begin{array}{c|c|c} \mathbf{\Gamma}(1:\kappa-1,:) & \mathbf{0}_{\kappa-1,\ell-1} & \mathbf{0}_{\kappa-1,\ell} \\ \hline \mathbb{A}^{\mathrm{Lag}} & \mathbf{\Delta}(1:\ell-1,:)^\top & \mathbf{0}_{\ell,\ell} \\ \hline \mathbb{B}^{\mathrm{Lag}} & \mathbf{0}_{\ell,\ell-1} & \mathbf{\Delta}^\top \end{array}\right] \in \mathbb{C}^{m\times m}[{}^1s,\ldots,{}^ns],$$

$$\mathbf{G} = \left[\begin{array}{c} \mathbf{0}_{\kappa-1,1} \\ \hline \mathbf{\Delta}(\ell,:)^\top \\ \hline \mathbf{0}_{\ell,1} \end{array}\right] \in \mathbb{C}^{m\times 1} \text{ and } \mathbf{C} = \left[\begin{array}{c|c|c} \mathbf{0}_{1,\kappa} & \mathbf{0}_{1,\ell-1} & -\mathbf{e}_\ell^\top \end{array}\right] \in \mathbb{C}^{1\times m},$$

where $\mathbf{e}_r$ *denotes the* r*th unit vector (i.e., all entries are zero except the last one, equal to* 1*) and where* $\mathbb{A}^{\mathrm{Lag}}, \mathbb{B}^{\mathrm{Lag}} \in \mathbb{C}^{\ell\times\kappa}$ *are appropriately chosen according to the pseudocompanion basis used.*

Proof. See subsection 2.2.2. □

Remark 2.9 (matrix realization). From Theorem 2.8 and following (1.1)'s notations, it follows that $\mathbf{\Phi}({}^1s,{}^2s,\ldots,{}^ns) = {}^1s\mathfrak{E}(\mathcal{S}) - \mathfrak{A}(\mathcal{S})$, $\mathbf{G} = \mathfrak{B}(\mathcal{S})$, and $\mathbf{C} = \mathfrak{C}(\mathcal{S})$.

Corollary 2.10. *The realization defined by the tuple* $(\mathbf{C},\mathbf{\Phi},\mathbf{G})$ *has dimension* $m = 2\ell+\kappa-1$*, and it is both R-controllable and R-observable, i.e.,*

$$\begin{bmatrix} \mathbf{\Phi}({}^1s,\ldots,{}^ns) & \mathbf{G} \end{bmatrix} \text{ and } \begin{bmatrix} \mathbf{C} \\ \mathbf{\Phi}({}^1s,\ldots,{}^ns) \end{bmatrix} \tag{2.8}$$

have full rank m *for all* ${}^js \in \mathbb{C}$*. Furthermore,* $\mathbf{\Phi}$ *is a polynomial matrix in the variables* ${}^js$*, while* $\mathbf{C}$ *and* $\mathbf{G}$ *are constant.*

Proof. The result follows by noticing that the expressions in question have full rank for all ${}^js \in \mathbb{C}$ because of the unimodularity of $\mathbf{\Delta}$ and $\mathbf{\Gamma}$. □

2.2.2. Proof of Theorem 2.8.

The Numerator of Realization (2.7). First, partition $\mathbf{\Phi} = \left[\begin{array}{c:c} \mathbf{\Phi}_{11} & \mathbf{0} \\ \hdashline \mathbf{\Phi}_{21} & \mathbf{\Phi}_{22} \end{array}\right]$, where the sizes of the four entries are $(\kappa+\ell-1)\times(\kappa+\ell-1)$, $(\kappa+\ell-1)\times\ell$, $\ell\times(\kappa+\ell-1)$, $\ell\times\ell$, $\mathbf{G} = \left[\begin{array}{c} \mathbf{G}_1 \\ \hdashline \mathbf{0}_{\ell,1} \end{array}\right]$, and $\mathbf{C} = [\mathbf{0}_{1,\kappa+\ell-1}\ \underbrace{-\mathbf{e}_\ell^\top}_{\mathbf{C}_2}]$. It follows that

$$\mathbf{H} = \mathbf{C}\,\mathbf{\Phi}^{-1}\,\mathbf{G} = \frac{\mathbf{n}}{\mathbf{d}} = \mathbf{C}_2\,\mathbf{\Phi}_{22}^{-1}\,\mathbf{\Phi}_{21}\,\mathbf{\Phi}_{11}^{-1}\,\mathbf{G}_1. \tag{2.9}$$

The last expression can be expressed explicitly as

$$\overbrace{\underbrace{[\mathbf{0}_{1,\ell-1} - \mathbf{e}_\ell^\top]}_{\mathbf{C}_2}\underbrace{\mathbf{\Delta}^{-\top}}_{\mathbf{\Phi}_{22}^{-1}}}^{\mathbf{r}_\ell^\top}\underbrace{[\mathbb{B}^{\mathrm{Lag}} \mid \mathbf{0}_{\ell,\ell-1}]}_{\mathbf{\Phi}_{21}}\overbrace{\underbrace{\left[\begin{array}{c:c} \mathbf{\Gamma}(1:\kappa-1,1:\kappa) & \mathbf{0}_{\kappa-1,\ell-1} \\ \hdashline \mathbb{A}^{\mathrm{Lag}} & \mathbf{\Delta}(1:\ell-1,:)^\top \end{array}\right]^{-1}}_{\mathbf{\Phi}_{11}^{-1}}\underbrace{\left[\begin{array}{c} \mathbf{0}_{\kappa-1,1} \\ \hline \mathbf{\Delta}(\ell,:)^\top \end{array}\right]}_{\mathbf{G}_1}}^{\left[\begin{array}{c} \mathbf{c}_\kappa \\ \hline \star \end{array}\right] \text{ where } \star \text{ has size } \ell-1}.$$

The expressions for $\mathbf{r}_\ell^\top$ and $\mathbf{c}_\kappa$ are a consequence of Proposition 2.11. It follows that $\mathbf{n} = \mathbf{r}_\ell^\top\,\mathbb{B}^{\mathrm{Lag}}\,\mathbf{c}_\kappa$. Interchanging $\mathbb{A}^{\mathrm{Lag}}$ and $\mathbb{B}^{\mathrm{Lag}}$ in (2.7) amounts to interchanging $\mathbf{n}$ and $\mathbf{d}$ in $\mathbf{H}$ (2.9); the expression for the denominator is $\mathbf{d} = \mathbf{r}_\ell^\top\,\mathbb{A}^{\mathrm{Lag}}\,\mathbf{c}_\kappa$.

PROPOSITION 2.11. (a) *The last row of* $\mathbf{\Delta}^{-\top}$ *is*

$$\mathbf{r}_\ell^\top = \left[\frac{1}{{}^{k+1}\mathbf{x}_1}, \ldots, \frac{1}{{}^{k+1}\mathbf{x}_{n_{k+1}+1}}\right] \otimes \cdots \otimes \left[\frac{1}{{}^{n}\mathbf{x}_1}, \ldots, \frac{1}{{}^{n}\mathbf{x}_{n+1}}\right].$$

Therefore, $\mathbf{r}_\ell^\top \cdot \mathbb{B}^{\mathrm{Lag}}$ *is a matrix of size* $1\times\kappa$. (b) *The last column of* $\mathbf{\Gamma}^{-1}$ *is*

$$\mathbf{c}_\kappa = \left[\frac{1}{{}^{1}\mathbf{x}_1}, \ldots, \frac{1}{{}^{1}\mathbf{x}_{n_1+1}}\right]^\top \otimes \cdots \otimes \left[\frac{1}{{}^{k}\mathbf{x}_1}, \ldots, \frac{1}{{}^{k}\mathbf{x}_{n_k+1}}\right]^\top.$$

Remark 2.12. The possibility of splitting the variables into *left* and *right* variables allows for the choosing of a splitting that *minimizes* m. For instance, if we have four variables with degrees $(2,2,1,1)$, splitting the variables into $(2,1)$–$(2,1)$ yields $m=17$, while the splitting (2)–$(2,1,1)$ (i.e., one column and three rows) yields $m=26$.

2.3. Comments. In Theorem 2.8, both matrices $\mathbb{A}^{\mathrm{Lag}}, \mathbb{B}^{\mathrm{Lag}} \in \mathbb{C}^{\ell\times\kappa}$ are directly related to the pseudocompanion basis chosen in Definition 2.1 and to the columns-rows variables split. Without entering into technical considerations (which is out of scope for this paper), one may make the following observations. (i) Different pseudocompanion forms (2.1) can be considered, leading to different structures associated with different polynomial bases, such as the Lagrange or the monomial basis. Here, the Lagrange basis will be considered exclusively. (ii) Different permutations and rearrangements of ${}^{j}s$ in Definition 2.2 may be considered. This results in a different realization order with $m = 2\ell+\kappa-1$. Consequently, an adequate choice leads to a reduced order realization, *taming the realization dimensionality* issue.

We are now ready to introduce the main ingredient, namely, the data set. The data can be obtained from any (dynamical) black box model, simulator, or experiment.

3. Definition and Description of the Data.

Following the Loewner philosophy presented in a series of papers [35, 7, 29, 20], let us define P_c, the column (or right)

Table 1 *1-D and 2-D tableau construction.*

(a) 1-D tableau construction: $\mathbf{tab}_1$.

${}^1s$	
${}^1\lambda_{1,\dots,k_1}$	$\mathbf{W}^{\otimes}_{k_1}$
${}^1\mu_{1,\dots,q_1}$	$\mathbf{V}^{\otimes}_{q_1}$

(b) 2-D tableau construction: $\mathbf{tab}_2$.

${}^1s$ \ ${}^2s$	${}^2\lambda_{1,\dots,k_2}$	${}^2\mu_{1,\dots,q_2}$
${}^1\lambda_{1,\dots,k_1}$	$\mathbf{W}^{\otimes}_{k_1,k_2}$	ϕ_{cr}
${}^1\mu_{1,\dots,q_1}$	ϕ_{rc}	$\mathbf{V}^{\otimes}_{q_1,q_2}$

data, and P_r, the row (or left) data. These data will serve the construction of the n-D Loewner matrices in section 4. In what follows, the 1-D and 2-D data cases are first recalled, in preparation for the exposition of the general n-D case.

3.1. The 1-D Case.

When considering single-valued functions $\mathbf{\mathfrak{H}}({}^1s)$, i.e., $n=1$ in (1.2), we define the following column and row data:

$$P_c^{(1)} := \left\{ \left({}^1\lambda_{j_1}; \mathbf{w}_{j_1}\right),\, j_1 = 1,\dots,k_1 \right\},\ P_r^{(1)} := \left\{ \left({}^1\mu_{i_1}; \mathbf{v}_{i_1}\right),\, i_1 = 1,\dots,q_1 \right\}, \tag{3.1}$$

where ${}^1\lambda_{j_1}, {}^1\mu_{i_1} \in \mathbb{C}$ are disjoint interpolation points (or support points) for which the evaluation of $\mathbf{\mathfrak{H}}$, respectively, leads to $\mathbf{w}_{j_1} \in \mathbb{C}$ and $\mathbf{v}_{i_1} \in \mathbb{C}$. To support our exposition, let the data (3.1) be represented in the tableau given in Table 1(a), where the measurement vectors $\mathbf{W}^{\otimes}_{k_1} \in \mathbb{C}^{k_1}$ and $\mathbf{V}^{\otimes}_{q_1} \in \mathbb{C}^{q_1}$ indicate the evaluation of $\mathbf{\mathfrak{H}}$ through the single variable ${}^1s$, evaluated at ${}^1\lambda_{j_1}$ and ${}^1\mu_{i_1}$, respectively. Table 1(a) (also called $\mathbf{tab}_1$) is called a measurement matrix. From $\mathbf{tab}_1$, the (1,1) block of dimension $k_1 \times 1$ contains the column measurements, and the (1,2) block of dimension $q_1 \times 1$ contains the row measurements.

3.2. The 2-D Case.

Let us define the column and row data

$$\left\{ \begin{array}{lll} P_c^{(2)} & := & \left\{ ({}^1\lambda_{j_1}, {}^2\lambda_{j_2}; \mathbf{w}_{j_1,j_2}),\, j_l = 1,\dots,k_l, \quad l = 1,2 \right\}, \\ P_r^{(2)} & := & \left\{ ({}^1\mu_{i_1}, {}^2\mu_{i_2}; \mathbf{v}_{i_1,i_2}),\, i_l = 1,\dots,q_l, \quad l = 1,2 \right\}, \end{array} \right. \tag{3.2}$$

where $\{{}^1\lambda_{j_1}, {}^1\mu_{i_1}\} \in \mathbb{C}^2$ and $\{{}^2\lambda_{j_2}, {}^2\mu_{i_2}\} \in \mathbb{C}^2$ are sets of disjoint interpolation points, for which evaluating $\mathbf{\mathfrak{H}}({}^1s, {}^2s)$, respectively, leads to $\mathbf{w}_{j_1,j_2}, \mathbf{v}_{i_1,i_2} \in \mathbb{C}$. Similar to the 1-D case, data (3.2) may be represented in Table 1(b), where $\mathbf{W}^{\otimes}_{k_1,k_2} \in \mathbb{C}^{k_1 \times k_2}$ and $\mathbf{V}^{\otimes}_{q_1,q_2} \in \mathbb{C}^{q_1 \times q_2}$ are the measurement matrices related to the evaluation of $\mathbf{\mathfrak{H}}$ through the two variables ${}^1s$ and ${}^2s$ evaluated at $\{{}^1\lambda_{j_1}, {}^2\lambda_{j_2}\}$ and $\{{}^1\mu_{i_1}, {}^2\mu_{i_2}\}$.

Compared to the single-variable case, the tableau embeds two additional sets of measurements: $\phi_{rc} \in \mathbb{C}^{q_1 \times k_2}$ and $\phi_{cr} \in \mathbb{C}^{k_1 \times q_2}$. The former results from the cross-interpolation points evaluation of $\mathbf{\mathfrak{H}}({}^1s, {}^2s)$ along $\{{}^1\mu_{i_1}, {}^2\lambda_{j_2}\}$ and the latter from the evaluation along $\{{}^1\lambda_{j_1}, {}^2\mu_{i_2}\}$. It follows that Table 1(b) (denoted $\mathbf{tab}_2$) is a measurement matrix.

Remark 3.1 (cross-measurements). In [29, 47], these cross-measurements are used in the extended Loewner matrix construction for improved accuracy.

3.3. The n-D Case.

Now that the single- and two-variable cases have been reviewed and notations introduced, let us present the n-variable data case:

$$\begin{array}{l} P_c^{(n)} := \left\{ ({}^1\lambda_{j_1}, \dots, {}^n\lambda_{j_n}; \mathbf{w}_{j_1,j_2,\dots,j_n}),\, j_l = 1,\dots,k_l, \quad l = 1,\dots,n \right\}, \\ P_r^{(n)} := \left\{ ({}^1\mu_{i_1}, \dots, {}^n\mu_{i_n}; \mathbf{v}_{i_1,i_2,\dots,i_n}),\, i_l, \dots, q_l, \quad l = 1,\dots,n \right\}. \end{array} \tag{3.3}$$

Similarly, one may derive the n-variable measurement matrix called $\mathbf{tab}_n$, illustrated in the table sequence given in Table 2. Similar to the expositions made for

Table 2 *n-D table construction: $\mathbf{tab}_n$ (some configurations).*

(a) $^3s = {}^3\lambda_1, {}^4s = {}^4\lambda_1, \ldots, {}^ns = {}^n\lambda_1$.

^2s ^1s	$^2\lambda_{k_2}$	$^2\mu_{q_2}$
$^1\lambda_{k_1}$	$\mathbf{W}^{\otimes}_{k_1,k_2,1,\ldots,1}$	$\phi_{crc\cdots c}$
$^1\mu_{q_1}$	$\phi_{rcc\cdots c}$	$\phi_{rrc\cdots c}$

(b) $^3s = {}^3\mu_1, {}^4s = {}^4\mu_1, \ldots, {}^ns = {}^n\mu_1$.

^2s ^1s	$^2\lambda_{k_2}$	$^2\mu_{q_2}$
$^1\lambda_{k_1}$	$\phi_{ccr\cdots r}$	$\phi_{crr\cdots r}$
$^1\mu_{q_1}$	$\phi_{rcr\cdots r}$	$\mathbf{V}^{\otimes}_{q_1,q_2,1,\ldots,1}$

(c) $^3s = {}^3\lambda_{k_3}, {}^4s = {}^4\lambda_{k_4}, \ldots, {}^ns = {}^n\lambda_{k_n}$.

^2s ^1s	$^2\lambda_{k_2}$	$^2\mu_{q_2}$
$^1\lambda_{k_1}$	$\mathbf{W}^{\otimes}_{k_1,k_2,\ldots,k_n}$	$\phi_{crc\cdots c}$
$^1\mu_{q_1}$	$\phi_{rcc\cdots c}$	$\phi_{rrc\cdots c}$

(d) $^3s = {}^3\mu_{q_3}, {}^4s = {}^4\mu_{q_4}, \ldots, {}^ns = {}^n\mu_{q_3}$.

^2s ^1s	$^2\lambda_{k_2}$	$^2\mu_{q_2}$
$^1\lambda_{k_1}$	$\phi_{ccr\cdots r}$	$\phi_{crr\cdots r}$
$^1\mu_{q_1}$	$\phi_{rcr\cdots r}$	$\mathbf{V}^{\otimes}_{q_1,q_2,\ldots,q_n}$

the single- and two-variable cases, each subtable considers frozen configurations of $^3s, {}^4s, \ldots, {}^ns$ along with the combinations of the support points $^3\lambda_{j_3}, {}^4\lambda_{j_4}, \ldots, {}^n\lambda_{j_n}$ and $^3\mu_{i_3}, {}^4\mu_{i_4}, \ldots, {}^n\mu_{i_n}$, thus forming an n-dimensional tensor. In particular, considering the first subtableau, the evaluation is for $^3s, {}^4s, \ldots, {}^ns = {}^3\lambda_1, {}^4\lambda_1, \ldots, {}^n\lambda_1$. The $\mathbf{W}^{\otimes}_{k_1,k_2,j_3,\ldots,j_n} \in \mathbb{C}^{k_1\times k_2}$ and $\mathbf{V}^{\otimes}_{q_1,q_2,i_3,\ldots,i_n} \in \mathbb{C}^{q_1\times q_2}$ entries concatenated form the data tensors $\mathbf{W}^{\otimes} \in \mathbb{C}^{k_1\times k_2\times\cdots\times k_n}$ and $\mathbf{V}^{\otimes} \in \mathbb{C}^{q_1\times q_2\times\cdots\times q_n}$; $\mathbf{tab}_n$ is an n-dimensional tensor.

4. Multivariate Loewner Matrices and Null Spaces.

Based on section 3 (specifically on (3.3) and $\mathbf{tab}_n$), we are now ready to present our *main tool*: the *multivariate Loewner matrix.* Following the exposition in the previous section, we first recall the 1-D and 2-D Loewner matrices before presenting the n-D counterpart. For each dimension, the Loewner matrix is illustrated in close connection to the Sylvester equation that it satisfies. Then, the relationship between the Loewner null space and the barycentric rational function is stated, and the connection with generalized realization is established, linking the data of section 3 with the realization of section 2.

4.1. The 1-D Case.

The single-variable case is briefly mentioned here (more details and connections with dynamical systems theory may be found in [8]).

4.1.1. Loewner Matrix and the Sylvester Equation.

Definition 4.1. *Given the data described in* (3.1)*, the* 1*-D Loewner matrix* $\mathbb{L}_1 \in \mathbb{C}^{q_1\times k_1}$ *has* (i_1, j_1)*th entries equal to*

$$(\mathbb{L}_1)_{i_1,j_1} = \frac{\mathbf{v}_{i_1} - \mathbf{w}_{j_1}}{{}^1\mu_{i_1} - {}^1\lambda_{j_1}},\; i_1 = 1, \ldots, q_1,\; j_1 = 1, \ldots, k_1.$$

Theorem 4.2. *Considering the data in* (3.1)*, we define the following matrices:*

$$\begin{gathered}\mathbf{\Lambda}_1 = \operatorname{diag}\left({}^1\lambda_1, \ldots, {}^1\lambda_{k_1}\right)\;,\; \mathbf{M}_1 = \operatorname{diag}\left({}^1\mu_1, \ldots, {}^1\mu_{q_1}\right),\\ \mathbb{W}_1 = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{k_1}]\;,\; \mathbb{V}_1 = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{q_1}]^\top,\; \textit{and}\; \mathbf{L}_1 = \mathbf{1}_{q_1},\; \mathbf{R}_1 = \mathbf{1}_{k_1}^\top.\end{gathered}$$

The Loewner matrix as defined in Definition 4.1 *is the solution of the Sylvester equation,* $\mathbf{M}_1\mathbb{L}_1 - \mathbb{L}_1\mathbf{\Lambda}_1 = \mathbb{V}_1\mathbf{R}_1 - \mathbf{L}_1\mathbb{W}_1$.

4.1.2. Null Space, Lagrange Basis Form, and Generalized Realization.

Computing $\mathbb{L}_1\mathbf{c}_1 = 0$, the null space of the Loewner matrix $\mathbb{L}_1$, the following holds (with an appropriate number of interpolation points): $\mathbf{c}_1 = \begin{bmatrix} c_1 & \cdots & c_{k_1} \end{bmatrix}^\top \in \mathbb{C}^{k_1}$ contains the so-called barycentric weights of the single-variable rational function $\mathbf{H}({}^1s)$ of degree $(d_1) = (k_1 - 1)$ given by

$$\mathbf{H}({}^1s) = \frac{\sum_{j_1=1}^{k_1} \frac{c_{j_1}\mathbf{w}_{j_1}}{{}^1s - {}^1\lambda_{j_1}}}{\sum_{j_1=1}^{k_1} \frac{c_{j_1}}{{}^1s - {}^1\lambda_{j_1}}} = \frac{\sum_{j_1=1}^{k_1} \frac{\beta_{j_1}}{{}^1s - {}^1\lambda_{j_1}}}{\sum_{j_1=1}^{k_1} \frac{c_{j_1}}{{}^1s - {}^1\lambda_{j_1}}},$$

where $\mathbf{c}_1^\top \odot \mathbb{W}_1 = [\ \beta_1 \quad \beta_2 \quad \cdots \quad \beta_{k_1}\ ] \in \mathbb{C}^{k_1}$ interpolates $\mathfrak{H}({}^1s)$ at points ${}^1\lambda_{j_1}$.

Result 4.3 (1-D realization). *Given Definition* 2.2 *and following Theorem* 2.8, *a generalized realization of* $\mathbf{H}({}^1s)$ *is obtained with the following settings:* $\mathbb{A}^{\mathrm{Lag}} = -\mathbf{c}_1^\top$, $\mathbb{B}^{\mathrm{Lag}} = \emptyset$, $\mathbf{\Gamma} = {}^1\mathbb{X}^{\mathrm{Lag}}$, *and* $\mathbf{\Delta} = \emptyset$.

Note that this representation recovers the result already discussed, e.g., in [8].

Example 4.4. Let us consider $\mathfrak{H}({}^1s) = \mathfrak{H}(s) = (s^2+4)/(s+1)$, a single-valued rational function of complexity 2 (i.e., dimension 2 along s). By evaluating $\mathfrak{H}$ in ${}^1\lambda_{j_1} = [1,3,5]$ and ${}^1\mu_{i_1} = [2,4,6,8]$, one obtains $\mathbf{w}_{j_1} = [5/2, 13/4, 29/6]$ and $\mathbf{v}_{i_1} = [8/3, 4, 40/7, 68/9]$. Then, we construct the Loewner matrix, its null space ($\mathbf{rank}\,\mathbb{L}_1 = 2$), and a rational function interpolating the data as

$$\mathbb{L}_1 = \begin{bmatrix} \frac{1}{6} & \frac{7}{12} & \frac{13}{18} \\ \frac{1}{2} & \frac{3}{4} & \frac{5}{6} \\ \frac{9}{14} & \frac{23}{28} & \frac{37}{42} \\ \frac{13}{18} & \frac{31}{36} & \frac{49}{54} \end{bmatrix}, \ \mathbf{c}_1 = \begin{bmatrix} \frac{1}{3} \\ -\frac{4}{3} \\ 1 \end{bmatrix}, \ \mathbf{H}(s) = \frac{\frac{5}{6(s-1)} - \frac{13}{3(s-3)} + \frac{29}{6(s-5)}}{\frac{1}{3(s-1)} - \frac{4}{3(s-3)} + \frac{1}{s-5}}.$$

Then, $\mathbf{H}(s)$ recovers the original function $\mathfrak{H}(s)$. A realization in the Lagrange basis can be obtained as $\mathbf{H}(s) = \mathbf{C}\mathbf{\Phi}(s)^{-1}\mathbf{G}$, where

$$\mathbf{\Phi}(s) = \left[\begin{array}{ccc} s-1 & 3-s & 0 \\ s-1 & 0 & 5-s \\ \hdashline -\frac{1}{3} & \frac{4}{3} & -1 \end{array}\right] \text{ and } \left\{ \begin{array}{l} \mathbf{C} = \left[\ \frac{5}{6} \quad -\frac{13}{3} \quad \frac{29}{6}\ \right] \\ \mathbf{G}^\top = \left[\begin{array}{cc:c} 0 & 0 & -1 \end{array}\right]. \end{array} \right.$$

4.2. The 2-D Case.

This section recalls the results originally given in [7] for the case of two variables.

4.2.1. The Loewner Matrix and Sylvester Equations.

Similarly to the 1-D case, let us now define the Loewner matrix in the 2-D case.

Definition 4.5. *Given the data described in* (3.2), *the* 2*-D Loewner matrix* $\mathbb{L}_2 \in \mathbb{C}^{q_1q_2 \times k_1k_2}$, *has matrix entries given by*

$$\ell_{j_1,j_2}^{i_1,i_2} = \frac{\mathbf{v}_{i_1,i_2} - \mathbf{w}_{j_1,j_2}}{({}^1\mu_{i_1} - {}^1\lambda_{j_1})({}^2\mu_{i_2} - {}^2\lambda_{j_2})}.$$

Definition 4.6. *Considering the data given in* (3.2), *we define the following matrices based on Kronecker products:*

$$\begin{array}{c} \mathbf{\Lambda}_1 = \mathrm{diag}\left({}^1\lambda_1, \ldots, {}^1\lambda_{k_1}\right) \otimes \mathbf{I}_{k_2}\ ,\ \mathbf{M}_1 = \mathrm{diag}\left({}^1\mu_1, \ldots, {}^1\mu_{q_1}\right) \otimes \mathbf{I}_{q_2}, \\ \mathbf{\Lambda}_2 = \mathbf{I}_{k_1} \otimes \mathrm{diag}\left({}^2\lambda_1, \ldots, {}^2\lambda_{k_2}\right)\ ,\ \mathbf{M}_2 = \mathbf{I}_{q_1} \otimes \mathrm{diag}\left({}^2\mu_1, \ldots, {}^2\mu_{q_2}\right), \\ \mathbb{W}_2 = [\mathbf{w}_{1,1}, \mathbf{w}_{1,2}, \ldots, \mathbf{w}_{1,k_2}, \mathbf{w}_{2,1}, \ldots, \mathbf{w}_{k_1,k_2}]\ ,\ \mathbf{R}_2 = \mathbf{1}_{k_1k_2}^\top, \\ \mathbb{V}_2 = [\mathbf{v}_{1,1}, \mathbf{v}_{1,2}, \ldots, \mathbf{v}_{1,q_2}, \mathbf{v}_{2,1}, \ldots, \mathbf{v}_{q_1,q_2}]^\top,\ \text{and } \mathbf{L}_2 = \mathbf{1}_{q_1q_2}. \end{array} \tag{4.1}$$

THEOREM 4.7. *The* 2*-D Loewner matrix as defined in Definition* 4.5 *is the solution of the following set of coupled Sylvester equations:*

$$\mathbf{M}_2\mathbb{X} - \mathbb{X}\mathbf{\Lambda}_2 = \mathbb{V}_2\mathbf{R}_2 - \mathbf{L}_2\mathbb{W}_2 \text{ and } \mathbf{M}_1\mathbb{L}_2 - \mathbb{L}_2\mathbf{\Lambda}_1 = \mathbb{X}. \tag{4.2}$$

COROLLARY 4.8. *By eliminating the variable* $\mathbb{X}$*, it follows that the* 2*-D Loewner matrix above satisfies the following generalized Sylvester equation:*

$$\mathbf{M}_2\mathbf{M}_1\mathbb{L}_2 - \mathbf{M}_2\mathbb{L}_2\mathbf{\Lambda}_1 - \mathbf{M}_1\mathbb{L}_2\mathbf{\Lambda}_2 + \mathbb{L}_2\mathbf{\Lambda}_1\mathbf{\Lambda}_2 = \mathbb{V}_2\mathbf{R}_2 - \mathbf{L}_2\mathbb{W}_2.$$

4.2.2. Null Space, Lagrange Basis Form, and Generalized Realization.

Computing $\mathbb{L}_2\mathbf{c}_2 = 0$, the null space of the bivariate Loewner matrix, we obtain (using the appropriate number of interpolation points) $\mathbf{c}_2^\top = [\ \underbrace{c_{1,1} \cdots c_{1,k_2}}_{\alpha_1^\top} \mid \underbrace{c_{2,1} \cdots c_{2,k_2}}_{\alpha_2^\top} \mid \cdots \mid \underbrace{c_{k_1,1} \cdots c_{k_1,k_2}}_{\alpha_{k_1}^\top}\ ] \in \mathbb{C}^{k_1k_2}$ and $\mathbf{c}_2^\top \odot \mathbb{W}_2 = [\ \beta_1^\top \mid \beta_2^\top \mid \cdots \mid \beta_{k_1}^\top\ ] \in \mathbb{C}^{k_1k_2}$. These are the barycentric weights of the bivariate rational function $\mathbf{H}({}^1s, {}^2s)$ of degree $(d_1, d_2) = (k_1 - 1, k_2 - 1)$,

$$\mathbf{H}({}^1s, {}^2s) = \frac{\sum_{j_1=1}^{k_1}\sum_{j_2=1}^{k_2} \frac{\beta_{j_1,j_2}}{({}^1s - {}^1\lambda_{j_1})({}^2s - {}^2\lambda_{j_2})}}{\sum_{j_1=1}^{k_1}\sum_{j_2=1}^{k_2} \frac{c_{j_1,j_2}}{({}^1s - {}^1\lambda_{j_1})({}^2s - {}^2\lambda_{j_2})}},$$

which interpolates $\mathfrak{H}({}^1s, {}^2s)$ at the support points $\{{}^1\lambda_{j_1}, {}^2\lambda_{j_2}\}$.

RESULT 4.9 (2-D realization). *Given Definition* 2.2 *and following Theorem* 2.8*, a generalized realization of* $\mathbf{g}({}^1s, {}^2s)$ *is obtained by means of* $\mathbb{A}^{\mathrm{Lag}} = [\ \alpha_1 \quad \alpha_2 \cdots \alpha_{k_1}\ ]$, $\mathbb{B}^{\mathrm{Lag}} = [\ \beta_1 \quad \beta_2 \cdots \beta_{k_1}\ ]$*, and* $\mathbf{\Gamma} = {}^1\mathbb{X}^{\mathrm{Lag}}$, $\mathbf{\Delta} = {}^2\mathbb{X}^{\mathrm{Lag}}$.

Example 4.10. Let us consider $\mathfrak{H}({}^1s, {}^2s) = \mathfrak{H}(s,t) = (s^2t)/(s - t + 1)$ of complexity $(2,1)$. By evaluating $\mathfrak{H}$ in ${}^1\lambda_{j_1} = [1,3,5]$, ${}^1\mu_{i_1} = [0,2,4]$, ${}^2\lambda_{j_2} = [-1,-3]$, and ${}^2\mu_{i_2} = [-2,-4]$, one obtains the response tableau $\mathbf{tab}_2$,

$$\left[\begin{array}{c|c} \mathbf{W}_{k_1,k_2}^{\otimes} & \phi_{cr} \\ \hline \phi_{rc} & \mathbf{V}_{q_1,q_2}^{\otimes} \end{array}\right] = \left[\begin{array}{cc|cc} -\frac{1}{3} & -\frac{3}{5} & -\frac{1}{2} & -\frac{2}{3} \\ -\frac{9}{5} & -\frac{27}{7} & -3 & -\frac{9}{2} \\ -\frac{25}{7} & -\frac{25}{3} & -\frac{25}{4} & -10 \\ \hline 0 & 0 & 0 & 0 \\ -1 & -2 & -\frac{8}{5} & -\frac{16}{7} \\ -\frac{8}{3} & -6 & -\frac{32}{7} & -\frac{64}{9} \end{array}\right].$$

The 2-D Loewner matrix is computed with its null space $(\mathbf{rank}\,(\mathbb{L}_2) = 5)$ as

$$\mathbb{L}_2 = \left[\begin{array}{cc|cc|cc} \frac{1}{3} & -\frac{3}{5} & \frac{3}{5} & -\frac{9}{7} & \frac{5}{7} & -\frac{5}{3} \\ \frac{1}{9} & \frac{3}{5} & \frac{1}{5} & \frac{9}{7} & \frac{5}{21} & \frac{5}{3} \\ \frac{19}{15} & -1 & \frac{1}{5} & -\frac{79}{35} & \frac{23}{35} & -\frac{101}{45} \\ \hline \frac{41}{63} & \frac{59}{35} & -\frac{17}{105} & \frac{11}{7} & \frac{1}{7} & \frac{127}{63} \\ \frac{89}{63} & -\frac{139}{105} & \frac{97}{35} & -\frac{5}{7} & -1 & -\frac{79}{21} \\ \frac{61}{81} & \frac{293}{135} & \frac{239}{135} & \frac{205}{63} & -\frac{223}{189} & \frac{11}{9} \end{array}\right], \ \mathbf{c}_2 = \left[\begin{array}{c} -\frac{1}{3} \\ \frac{5}{9} \\ \hline \frac{10}{9} \\ -\frac{14}{9} \\ \hline -\frac{7}{9} \\ 1 \end{array}\right], \ \mathbb{W}_2^\top = \left[\begin{array}{c} -\frac{1}{3} \\ -\frac{3}{5} \\ \hline -\frac{9}{5} \\ -\frac{27}{7} \\ \hline -\frac{25}{7} \\ -\frac{25}{3} \end{array}\right].$$

It follows that the two-variable rational function $\mathbf{H}(s,t)$, given in the barycentric form, recovers the original rational function $\mathfrak{H}(s,t)$. Then, a realization in the Lagrange basis (with $\{{}^1\lambda_{j_1}, {}^2\lambda_{j_2}\}$) is obtained as $\mathbf{H}(s,t) = \mathbf{C}\mathbf{\Phi}(s,t)^{-1}\mathbf{G}$, where $\mathbf{C} = -\mathbf{e}_6^\top$,

$$\mathbf{\Phi}(s,t) = \left[\begin{array}{ccc|c|cc} s-1 & 3-s & 0 & 0 & 0 & 0 \\ s-1 & 0 & 5-s & 0 & 0 & 0 \\ \hline -\frac{1}{3} & \frac{10}{9} & -\frac{7}{9} & t+1 & 0 & 0 \\ \frac{5}{9} & -\frac{14}{9} & 1 & -t-3 & 0 & 0 \\ \hline \frac{1}{9} & -2 & \frac{25}{9} & 0 & t+1 & \frac{1}{2} \\ -\frac{1}{3} & 6 & -\frac{25}{3} & 0 & -t-3 & -\frac{1}{2} \end{array}\right] \text{ and } \mathbf{G} = \left[\begin{array}{c} 0 \\ 0 \\ \hline 1/2 \\ -1/2 \\ \hline 0 \\ 0 \end{array}\right].$$

By applying the Schur complement the realization can be compressed to dimension 4, at the expense of introducing a parameter-dependent output matrix $\mathbf{C}(t)$.

4.3. The n-D Case.

4.3.1. Loewner Matrices and Sylvester Equations.

DEFINITION 4.11. *Given the data described in* (3.3), *the n-D Loewner matrix $\mathbb{L}_n \in \mathbb{C}^{q_1 q_2 \cdots q_n \times k_1 k_2 \cdots k_n}$, has entries given by*

$$\ell^{i_1,i_2,\ldots,i_n}_{j_1,j_2,\ldots,j_n} = \frac{\mathbf{v}_{i_1,i_2,\ldots,i_n} - \mathbf{w}_{j_1,j_2,\ldots,j_n}}{({}^1\mu_{i_1} - {}^1\lambda_{j_1})\,({}^2\mu_{i_2} - {}^2\lambda_{j_2})\cdots({}^n\mu_{i_n} - {}^n\lambda_{j_n})}.$$

DEFINITION 4.12. *Considering the data given in* (3.3), *we define the following matrices based on Kronecker products:*

$$\begin{array}{ll} \mathbf{\Lambda}_1 = \operatorname{diag}\left({}^1\lambda_1,\ldots,{}^1\lambda_{k_1}\right)\otimes\mathbf{I}_{k_2}\otimes\mathbf{I}_{k_3}\otimes\cdots\otimes\mathbf{I}_{k_n}, & \mathbf{M}_1 = \operatorname{diag}\left({}^1\mu_1,\ldots,{}^1\mu_{q_1}\right)\otimes\mathbf{I}_{q_2}\otimes\mathbf{I}_{q_3}\otimes\cdots\otimes\mathbf{I}_{q_n} \\ \mathbf{\Lambda}_2 = \mathbf{I}_{k_1}\otimes\operatorname{diag}\left({}^2\lambda_1,\ldots,{}^2\lambda_{k_2}\right)\otimes\mathbf{I}_{k_3}\otimes\cdots\otimes\mathbf{I}_{k_n}, & \mathbf{M}_2 = \mathbf{I}_{q_1}\otimes\operatorname{diag}\left({}^2\mu_1,\ldots,{}^2\mu_{q_2}\right)\otimes\mathbf{I}_{q_3}\otimes\cdots\otimes\mathbf{I}_{q_n} \\ \quad\cdots & \quad\cdots \\ \mathbf{\Lambda}_n = \mathbf{I}_{k_1}\otimes\cdots\otimes\mathbf{I}_{k_{n-1}}\otimes\operatorname{diag}\left({}^n\lambda_1,\ldots,{}^n\lambda_{k_n}\right), & \mathbf{M}_n = \mathbf{I}_{q_1}\otimes\cdots\otimes\mathbf{I}_{q_{n-1}}\otimes\operatorname{diag}\left({}^n\mu_1,\ldots,{}^n\mu_{q_n}\right), \end{array}$$

$\mathbb{W}_n = [\mathbf{w}_{1,1,\ldots,1}, \mathbf{w}_{1,1,\ldots,2}, \ldots, \mathbf{w}_{1,1,\ldots,k_n}, \mathbf{w}_{1,\ldots,2,1}, \ldots, \mathbf{w}_{k_1,k_2,\ldots,k_n}]$, $\mathbf{R}_n = \mathbf{1}^\top_{k_1 k_2 \cdots k_n}$, $\mathbb{V}_n = [\mathbf{v}_{1,1,\ldots,1}, \mathbf{v}_{1,1,\ldots,2}, \ldots, \mathbf{v}_{1,1,\ldots,q_n}, \mathbf{v}_{1,\ldots,2,1}, \ldots, \mathbf{v}_{q_1,q_2,\ldots,q_n}]^\top$, *and* $\mathbf{L}_n = \mathbf{1}_{q_1 q_2 \cdots q_n}$.

THEOREM 4.13. *The n-D Loewner matrix as introduced in Definition* 4.11 *is the solution of the following set of coupled Sylvester equations:*

$$\left\{\begin{array}{rcl} \mathbf{M}_n\mathbb{X}_1 - \mathbb{X}_1\mathbf{\Lambda}_n & = & \mathbb{V}_n\mathbf{R}_n - \mathbf{L}_n\mathbb{W}_n, \\ \mathbf{M}_{n-1}\mathbb{X}_2 - \mathbb{X}_2\mathbf{\Lambda}_{n-1} & = & \mathbb{X}_1, \\ & \cdots & \\ \mathbf{M}_2\mathbb{X}_{n-1} - \mathbb{X}_{n-1}\mathbf{\Lambda}_2 & = & \mathbb{X}_{n-2}, \\ \mathbf{M}_1\mathbb{L}_n - \mathbb{L}_n\mathbf{\Lambda}_1 & = & \mathbb{X}_{n-1}. \end{array}\right.$$

4.3.2. Null Space, Lagrange Basis Form, and Generalized Realization. When using an appropriate number of interpolation points, we can compute the null space of the n-variable Loewner matrix $\mathbb{L}_n$, i.e., $\mathbb{L}_n\mathbf{c}_n = 0$. Here, we denote it with $\mathbf{c}_n^\top = [\ \alpha_1 \quad || \, \alpha_2 \quad | \cdots \quad || \, \alpha_{k_1}\ ] \in \mathbb{C}^{k_1 k_2 \cdots k_n}$ written in terms of

$$\begin{array}{l} \alpha_1 = \left[c_{1,\ldots,1,1}\ \ldots\ c_{1,\ldots,1,k_n}\ c_{1,\ldots,2,1}\ \ldots\ c_{1,\ldots,2,k_n} \,|\cdots|\, c_{1,k_2,\ldots,k_{n-1},1}\ \ldots\ c_{1,k_2,\ldots,k_{n-1},k_n}\right], \\ \alpha_2 = \left[c_{2,\ldots,1,1}\ \cdots\ c_{2,\ldots,1,k_n}\ |\ \cdots\right], \\ \alpha_{k_1} = \left[c_{k_1,k_2,\ldots,1}\ \cdots\ c_{k_1,k_2,\ldots,k_n}\right], \end{array}$$

which are vectors that contain the so-called barycentric weights of the n-variable rational function $\mathbf{H}({}^1s, {}^2s, \ldots, {}^ns)$ given by

$$\mathbf{H}({}^1s, {}^2s, \ldots, {}^ns) = \frac{\sum_{j_1=1}^{k_1}\sum_{j_2=1}^{k_2}\cdots\sum_{j_n=1}^{k_n} \frac{\beta_{j_1,j_2,\ldots,j_n}}{\left({}^1s - {}^1\lambda_{j_1}\right)\left({}^2s - {}^2\lambda_{j_2}\right)\cdots\left({}^ns - {}^n\lambda_{j_n}\right)}}{\sum_{j_1=1}^{k_1}\sum_{j_2=1}^{k_2}\cdots\sum_{j_n=1}^{k_n} \frac{c_{j_1,j_2,\ldots,j_n}}{\left({}^1s - {}^1\lambda_{j_1}\right)\left({}^2s - {}^2\lambda_{j_2}\right)\cdots\left({}^ns - {}^n\lambda_{j_n}\right)}},$$

where $\mathbf{c}_n^\top \odot \mathbb{W}_n = [\ \beta_1 \quad \beta_2 \quad \cdots \quad \beta_{k_1}\ ] \in \mathbb{C}^{k_1k_2\cdots k_n}$. By construction, the function $\mathbf{H}({}^1s, {}^2s, \ldots, {}^ns)$ expressed above interpolates $\mathfrak{H}({}^1s, {}^2s, \ldots, {}^ns)$ at the support points $\{{}^1\lambda_{j_1}, {}^2\lambda_{j_2}, \ldots, {}^n\lambda_{j_n}\}$.

RESULT 4.14 (n-D realization for $k=1$). *Given Definition* 2.2 *and following Theorem* 2.8, *a generalized realization of* $\mathbf{H}({}^1s, {}^2s, \ldots, {}^ns)$ *is obtained with the following settings:* $\mathbb{A}^{\text{Lag}} = [\ \alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_{k_1}\ ]$, $\mathbb{B}^{\text{Lag}} = [\ \beta_1 \quad \beta_2 \quad \cdots \quad \beta_{k_1}\ ]$, $\mathbf{\Gamma} = {}^1\mathbb{X}^{\text{Lag}}$, *and* $\mathbf{\Delta} = {}^2\mathbb{X}^{\text{Lag}} \otimes \cdots \otimes {}^n\mathbb{X}^{\text{Lag}}$.

Example 4.15. Consider the three-variable rational function $\mathfrak{H}(s,t,p) = (s+pt)/(p^2+s+t)$ of complexity $(1,1,2)$. It is evaluated at ${}^1\lambda_{j_1} = [2,4]$, ${}^2\lambda_{j_2} = [1,3]$, ${}^3\lambda_{j_3} = [5,6,7]$ and ${}^1\mu_{i_1} = -{}^1\lambda_{j_1}$, ${}^2\mu_{i_2} = -{}^2\lambda_{j_2}$, ${}^3\mu_{i_3} = -{}^3\lambda_{j_3}$. The resulting 3-D Loewner matrix $\mathbb{L}_3$ has $\textbf{rank}\,(\mathbb{L}_3) = 11$ and

$$\begin{aligned} \mathbf{c}_3^\top &= \left[\begin{array}{ccc|ccc|ccc|ccc} \frac{1}{2} & -\frac{39}{28} & \frac{13}{14} & -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & \frac{4}{7} & -\frac{43}{28} & 1 \end{array} \right], \\ \mathbb{W}_3 &= \left[\begin{array}{ccc|ccc|ccc|ccc} \frac{1}{4} & \frac{8}{39} & \frac{9}{52} & \frac{17}{30} & \frac{20}{41} & \frac{23}{54} & \frac{3}{10} & \frac{10}{41} & \frac{11}{54} & \frac{19}{32} & \frac{22}{43} & \frac{25}{56} \end{array} \right]. \end{aligned}$$

Following Result 4.14, we may obtain the realization $(\mathbf{C}, \mathbf{\Phi}(s,t,p), \mathbf{G})$. By arranging as $(s)-(t,p)$, one obtains a realization of dimension $m=13$. Instead, by arranging as $(s,t)-(p)$ we obtain $m=9$. With the latter partitioning, we obtain $\kappa = 2\times 2$ (associated with variables s and t) and $\ell = 3$ (associated with variable p). Thus, with reference to the multi-indices of Definition 2.3, we readily have $I_1 = i_3^1$, $I_2 = i_3^2$ and $I_3 = i_3^3$, and $J_1 = [j_1^1, j_2^1]$, $J_2 = [j_1^2, j_2^2]$, $J_3 = [j_1^3, j_2^3]$, and $J_4 = [j_1^4, j_2^4]$.

5. Variable Decoupling and Addressing the Curse of Dimensionality.

As introduced in Definition 4.11, it follows that the n-D Loewner matrix $\mathbb{L}_n$ is of dimension $Q\times K$, where $Q = q_1q_2\ldots q_n$ and $K = k_1k_2\ldots k_n$. The dimension increases exponentially with the number of parameters and the corresponding degrees (this is also obvious when observing its Kronecker structure). Therefore, computing $\mathbf{c}_n$ results in $\mathcal{O}(QK^2)$ or $\mathcal{O}(KQ^2)$ `flop`, which stands as a limitation of the proposed approach. It is to be noted that the computationally most favorable case is $K=Q=N$, for which the complexity is $\mathcal{O}(N^3)$ `flop`.

The need for the full matrix to perform the `SVD` decomposition renders the process unfeasible in practice for many data sets. Here, the **C-o-D** is addressed through a tailored n-D Loewner matrix null space decomposition which results in the decoupling of the variables. More specifically, in this section we suggest an alternate approach allowing us to construct $\mathbf{c}_n$ without constructing $\mathbb{L}_n$. This approach **tames** the **C-o-D** by constructing a sequence of 1-D Loewner matrices and computing their associated null space instead. Similar to the previous sections, for clarity, we start with the 2-D and 3-D cases before addressing the n-D case. We finally show that in the n-D case, the null space boils down to (i) a 1-D Loewner matrix null space and (ii) multiple $(n-1)$-D Loewner matrix null spaces. With a recursive procedure, $(n-1)$-D becomes $(n-2)$-D, etc. This then leads to a series of 1-D Loewner matrix null space computations. Avoiding the explicit large-scale n-D Loewner matrix construction, which is replaced by small-scale 1-D Loewner matrices, results in drastic `flop` and storage savings.

5.1. Null Space Computation in the 2-D Case.

THEOREM 5.1. *Let* $h_{i,j} \in \mathbb{C}$ *be measurements of the transfer function* $\mathfrak{H}({}^1s, {}^2s)$, *with* ${}^1s_i$, $i=1,\ldots,n$, *and* ${}^2s_j$, $j=1,\ldots,m$. *Let* $k_1 = n/2$ *and* $k_2 = m/2$ *be the numbers of column interpolation points (see* $P_c^{(2)}$ *in* (3.2)*). The null space of the corresponding* 2*-D Loewner matrix is spanned by*[4]

[4] We assume here that $n = k_1+q_1$ and $m = k_2+q_2$, and $k_1 = q_1$ and $k_2 = q_2$. In other configurations, specific treatment is needed.

$$\mathcal{N}(\mathbb{L}_2) = \mathbf{vec}\left[\mathbf{c}_1^{{}^2\lambda_1} \cdot \left[\mathbf{c}_1^{{}^1\lambda_{k_1}}\right]_1, \ldots, \mathbf{c}_1^{{}^2\lambda_{k_2}} \cdot \left[\mathbf{c}_1^{{}^1\lambda_{k_1}}\right]_{k_2}\right], \tag{5.1}$$

where $\mathbf{c}_1^{{}^1\lambda_{k_1}} = \mathcal{N}(\mathbb{L}_1^{{}^1\lambda_{k_1}})$ *is the null space of the* 1*-D Loewner matrix for frozen* ${}^1s = {}^1\lambda_{k_1}$ *and* $\mathbf{c}_1^{{}^2\lambda_j} = \mathcal{N}(\mathbb{L}_1^{{}^2\lambda_j})$ *is the* j*th null space of the* 1*-D Loewner matrix for frozen* ${}^2s_j = \{{}^2\lambda_1, \ldots, {}^2\lambda_{k_2}\}$*.*

Proposition 5.2. *Given the setup in Theorem* 5.1*, the null space computation* `flop` *complexity is* $k_1^3 + k_1k_2^3$ *or* $k_2^3 + k_2k_1^3$*, rather than* $k_1^3k_2^3$*.*

Proof. For simplicity of exposition, let us denote by $h_{i,j} \in \mathbb{C}$ the value of a transfer function $\mathfrak{H}(s_i, t_j)$. We denote the Lagrange monomials by $s - s_i$, $i = 1, \ldots, n$, and by $t - t_j$, $j = 1, \ldots, m$. Then, let the response tableau (the data used for constructing the Loewner matrix) and corresponding barycentric weights $\mathbb{A}^{\mathrm{Lag}}$ be defined as

$$\underbrace{\left[\begin{array}{c|cccc} & t_1 & t_2 & \cdots & t_m \\ \hline s_1 & h_{1,1} & h_{1,2} & \cdots & h_{1,m} \\ s_2 & h_{2,1} & h_{2,2} & \cdots & h_{2,m} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ s_{n-1} & h_{n-1,1} & h_{n-1,2} & \cdots & h_{n-1,m} \\ s_n & h_{n,1} & h_{n,2} & \cdots & h_{n,m} \end{array}\right]}_{\mathbf{tab}_2}, \quad \underbrace{\left[\begin{array}{c|cccc} & t_1 & t_2 & \cdots & t_m \\ \hline s_1 & \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,m} \\ s_2 & \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,m} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ s_{n-1} & \alpha_{n-1,1} & \alpha_{n-1,2} & \cdots & \alpha_{n-1,m} \\ s_n & \alpha_{n,1} & \alpha_{n,2} & \cdots & \alpha_{n,m} \end{array}\right]}_{\mathbb{A}^{\mathrm{Lag}}}.$$

It follows that the denominator polynomial in the Lagrange basis is given as $\mathbf{d}(s,t) = \pi\sum_{i,j=1}^{n,m}\frac{\alpha_{i,j}}{(s-s_i)(t-t_j)}$, where $\pi = \Pi_{i=1}^{n}\Pi_{j=1}^{m}(s - s_i)(t - t_j)$. The coefficients are given by the null space of the associated 2-D Loewner matrix, i.e., $\mathcal{N}(\mathbb{L}_2) = \mathbf{span}\left(\mathbf{vec}\left(\mathbb{A}^{\mathrm{Lag}}\right)\right)$ (where $\mathbb{A}^{\mathrm{Lag}} = [\alpha_{i,j}]$).

If we now set $t = t_j$ $(j = 1, \ldots, m)$, the denominator polynomial becomes $\mathbf{d}(s, t_j) = \pi_{t_j}\sum_i^n \frac{\alpha_{i,j}}{(s-s_i)}$, where $\pi_{t_j} = \Pi_{i=1}^{n}(s - s_i)\Pi_{k\neq j}(t_j - t_k)$. In this case, the coefficients are given by the null space of the associated 1-D Loewner matrix, i.e., $\mathcal{N}(\mathbb{L}_1^{t_j}) = \mathbf{span}\left([\alpha_{1,j}, \ldots, \alpha_{n-1,j}, \alpha_{n,j}]^\top\right)$. Thus, these quantities reproduce the columns of $\mathbb{A}^{\mathrm{Lag}}$, up to a constant, for each column.

Similarly, for $s = s_n$, we obtain $\mathbf{d}(s_n, t) = \pi_{s_n}\sum_j^m \frac{\alpha_{n,j}}{(t-t_j)}$, where $\pi_{s_n} = \Pi_{i=1}^{n-1}(s_n - s_i)\Pi_{j=1}^{m}(t - t_j)$. Again, the coefficients are given by the null space of the associated 1-D Loewner matrix, i.e., $\mathcal{N}(\mathbb{L}_1^{s_n}) = \mathbf{span}\left([\alpha_{n,1}, \ldots, \alpha_{n,m-1}, \alpha_{n,m}]^\top\right)$.

This reproduces the last row of $\mathbb{A}^{\mathrm{Lag}}$, up to a constant. To eliminate these constants, we divide the corresponding vectors by $\alpha_{i,m}$ and obtain the following vectors:

$$\begin{array}{ccccc} \frac{\alpha_{1,1}}{\alpha_{n,1}} & \frac{\alpha_{1,2}}{\alpha_{n,2}} & \cdots & \frac{\alpha_{1,m-1}}{\alpha_{n,m-1}} & \frac{\alpha_{1,m}}{\alpha_{n,m}} \\ \frac{\alpha_{2,1}}{\alpha_{n,1}} & \frac{\alpha_{2,2}}{\alpha_{n,2}} & \cdots & \frac{\alpha_{2,m-1}}{\alpha_{n,m-1}} & \frac{\alpha_{2,m}}{\alpha_{n,m}} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{\alpha_{n-1,1}}{\alpha_{n,1}} & \frac{\alpha_{n-1,2}}{\alpha_{n,2}} & \cdots & \frac{\alpha_{n-1,m-1}}{\alpha_{n,m-1}} & \frac{\alpha_{n-1,m}}{\alpha_{n,m}} \\ 1 & 1 & \cdots & 1 & 1 \\ \hline\hline \frac{\alpha_{n,1}}{\alpha_{n,m}} & \frac{\alpha_{n,2}}{\alpha_{n,m}} & \cdots & \frac{\alpha_{n,m-1}}{\alpha_{n,m}} & 1 \end{array}$$

Finally, multiplying the jth column with the jth entry of the last row yields a vector that spans the desired null space of $\mathbb{L}_2$. The procedure requires computing the null

Table 3 *2-D tableau for Example* 5.4: $\boldsymbol{tab}_2$.

${}^2s$ ${}^1s$	${}^2\lambda_1=-1$	${}^2\lambda_2=-3$	${}^2\mu_1=-2$	${}^2\mu_2=-4$
${}^1\lambda_1=1$	$h_{1,1}=-\frac{1}{3}$	$h_{1,2}=-\frac{3}{5}$	$h_{1,3}=-\frac{1}{2}$	$h_{1,4}=-\frac{2}{3}$
${}^1\lambda_2=3$	$h_{2,1}=-\frac{9}{5}$	$h_{2,2}=-\frac{27}{7}$	$h_{2,3}=-3$	$h_{2,4}=-\frac{9}{2}$
${}^1\lambda_3=5$	$h_{3,1}=-\frac{25}{7}$	$h_{3,2}=-\frac{25}{3}$	$h_{3,3}=-\frac{25}{4}$	$h_{3,4}=-10$
${}^1\mu_1=0$	$h_{4,1}=0$	$h_{4,2}=0$	$h_{4,3}=0$	$h_{4,4}=0$
${}^1\mu_2=2$	$h_{5,1}=-1$	$h_{5,2}=-2$	$h_{5,3}=-\frac{8}{5}$	$h_{5,4}=-\frac{16}{7}$
${}^1\mu_3=4$	$h_{6,1}=-\frac{8}{3}$	$h_{6,2}=-6$	$h_{6,3}=-\frac{32}{7}$	$h_{6,4}=-\frac{64}{9}$

space of m 1-D Loewner matrices of size $n\times n$ and one 1-D Loewner matrix of size $m\times m$. Consequently, the number of flops is mn^3+m^3 instead of n^3m^3, concluding the proof. ∎

Remark 5.3 (normalization with other elements). In the above treatment, we normalize with the last element of the last row. However, it is clear that normalization with other elements can be chosen. This is especially relevant if the last element is zero, i.e., $\alpha_{n,m}=0$. In such a case, if we choose the kth row, we need the barycentric coefficients of the kth first variable.

Example 5.4. Continuing Example 4.10, we construct the tableau with the corresponding values, leading to Table 3. Here, instead of constructing the 2-D Loewner matrix $\mathbb{L}_2$ as in Example 4.10, we invoke Theorem 5.1. We thus construct a sequence of 1-D Loewner matrices as follows:[5]

- First, construct a 1-D Loewner matrix along ${}^1s$ for ${}^2s={}^2\lambda_2=-3$, i.e., considering data of $\mathbf{tab}_2(:,2)$ (second column). This leads to

$$\mathbb{L}_1^{{}^2\lambda_2}=\begin{bmatrix}-\frac{3}{5} & -\frac{9}{7} & -\frac{5}{3}\\ -\frac{7}{5} & -\frac{13}{7} & -\frac{19}{9}\\ -\frac{9}{5} & -\frac{15}{7} & -\frac{7}{3}\end{bmatrix} \text{ and } \mathbf{c}_1^{{}^2\lambda_2}=\begin{bmatrix}\frac{5}{9}\\ -\frac{14}{9}\\ 1\end{bmatrix}.$$

- Then, construct three 1-D Loewner matrices along ${}^2s$ for ${}^1s=\{{}^1\lambda_1,{}^1\lambda_2,{}^1\lambda_3\}$, i.e., considering data of $\mathbf{tab}_2(1,:)$, $\mathbf{tab}_2(2,:)$, and $\mathbf{tab}_2(3,:)$ (first, second, and third rows). This leads to: $\mathbb{L}_1^{{}^1\lambda_1}=\left[\begin{smallmatrix}\frac{1}{6} & \frac{1}{10}\\ \frac{1}{9} & \frac{1}{15}\end{smallmatrix}\right]\Rightarrow\mathbf{c}_1^{{}^1\lambda_1}=\left[\begin{smallmatrix}-\frac{3}{5}\\ 1\end{smallmatrix}\right]$, $\mathbb{L}_1^{{}^1\lambda_2}=\left[\begin{smallmatrix}\frac{6}{5} & \frac{6}{7}\\ \frac{9}{10} & \frac{9}{14}\end{smallmatrix}\right]$ $\Rightarrow\mathbf{c}_1^{{}^1\lambda_2}=\left[\begin{smallmatrix}-\frac{5}{7}\\ 1\end{smallmatrix}\right]$, $\mathbb{L}_1^{{}^1\lambda_3}=\left[\begin{smallmatrix}\frac{75}{28} & \frac{25}{12}\\ \frac{15}{7} & \frac{5}{3}\end{smallmatrix}\right]\Rightarrow\mathbf{c}_1^{{}^1\lambda_3}=\left[\begin{smallmatrix}-\frac{7}{9}\\ 1\end{smallmatrix}\right]$.
- Finally, $\hat{\mathbf{c}}_2=\left[\begin{array}{ccc}\mathbf{c}_1^{{}^1\lambda_1}\cdot[\mathbf{c}_1^{{}^2\lambda_2}]_1 & \mathbf{c}_1^{{}^1\lambda_2}\cdot[\mathbf{c}_1^{{}^2\lambda_2}]_2 & \mathbf{c}_1^{{}^1\lambda_3}\cdot[\mathbf{c}_1^{{}^2\lambda_2}]_3\end{array}\right]^\top$, and the scaled null space vector is equal to $\mathbf{c}_2$, directly obtained with the 2-D Loewner matrix (see Example 4.10). Similarly, the rational function and realization follow.

The corresponding computational cost is obtained by adding the following `flop`: one 1-D Loewner matrix of dimension $3\times 3 \rightsquigarrow$ null space computation takes $3^3=27$ `flop` and three 2×2 1-D Loewner matrices $\rightsquigarrow$ null space computation take $2^3=8$ `flop`. Thus, $27+3\times 8=51$ `flop` are needed here, while $6^3=216$ `flop` were required in Example 4.10, involving $\mathbb{L}_2$ directly. Note that the very same result may be obtained by computing first $\mathcal{N}(\mathbb{L}_1^{{}^1\lambda_3})\in\mathbb{R}^2$ then $\mathcal{N}(\mathbb{L}_1^{{}^2\lambda_1})$, $\mathcal{N}(\mathbb{L}_1^{{}^2\lambda_2})\in\mathbb{R}^3$. In this case, the computational cost would be $2^3+2\times 3^3=62$ `flop`.

[5] Here, $n=6$, $m=4$, $k_1=3$, $k_2=2$.

5.2. Null Space Computation in the 3-D Case.

THEOREM 5.5. *Let $h_{i,j,k} \in \mathbb{C}$ be measurements of the response of a transfer function $\mathfrak{H}({}^1s, {}^2s, {}^3s)$, along with ${}^1s_i$, ${}^2s_j$, and ${}^3s_k$ ($i = 1,\ldots,n$, $j = 1,\ldots,m$, and $k = 1,\ldots,p$), and let $k_1 \leq n/2$, $k_2 \leq m/2$, and $k_3 \leq p/2$ be the number of column interpolation points (see $P_c^{(3)}$ in (3.3), $n = 3$). The null space of the corresponding 3-D Loewner matrix is spanned by*[6]

$$\mathcal{N}(\mathbb{L}_3) = \mathbf{vec}\left[\mathbf{c}_2^{{}^1\lambda_1} \cdot \left[\mathbf{c}_1^{({}^2\lambda_{k_2}, {}^3\lambda_{k_3})}\right]_1, \ldots, \mathbf{c}_2^{{}^1\lambda_{k_1}} \cdot \left[\mathbf{c}_1^{({}^2\lambda_{k_2}, {}^3\lambda_{k_3})}\right]_{k_1}\right], \tag{5.2}$$

where $\mathbf{c}_1^{({}^2\lambda_{k_2}, {}^3\lambda_{k_3})} = \mathcal{N}(\mathbb{L}_1^{({}^2\lambda_{k_2}, {}^3\lambda_{k_3})})$ is the null space of the 1-D Loewner matrix for frozen $\{{}^2s, {}^3s\} = \{{}^2\lambda_{k_2}, {}^3\lambda_{k_3}\}$, and $\mathbf{c}_2^{{}^1\lambda_j} = \mathcal{N}(\mathbb{L}_2^{{}^1\lambda_j})$ is the jth null space of the 2-D Loewner matrix for frozen ${}^1s_j = \{{}^1\lambda_1, \ldots, {}^1\lambda_{k_1}\}$.

Proof. The proof follows that of Theorem 5.1. First, a 1-D null space Loewner matrix is computed for two frozen variables. Then a series of 2-D Loewner matrices are computed along with the two other variables. Scaling is similarly applied. □

Remark 5.6 (toward recursivity). From Theorem 5.5, it follows that the 3-D Loewner matrix null space may be obtained from one 1-D Loewner matrix, followed by multiple 2-D Loewner matrices. Then, invoking Theorem 5.1, these 2-D Loewner matrix null spaces may be split into a sequence of 1-D Loewner matrix null spaces. Therefore, a recursive scheme naturally appears (see Example 5.7).

Example 5.7. We continue with Example 4.15. We now illustrate how much the complexity and dimensionality issue may be reduced when applying the suggested recursive process. First, recall that the 3-D Loewner matrix $\mathbb{L}_3$ has a dimension of 12 and its null space is $\mathbf{c}_3^\top = [\mathbf{c}_{3,1}^\top || \mathbf{c}_{3,2}^\top]$ is given as

$$\mathbf{c}_3 = \left[\begin{array}{cccccccccccc} \frac{1}{2} & -\frac{39}{28} & \frac{13}{14} & | -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & \| -\frac{15}{28} & \frac{41}{28} & -\frac{27}{28} & | \frac{4}{7} & -\frac{43}{28} & 1 \end{array}\right]^\top.$$

Computing such a null space requires an `SVD` matrix decomposition of complexity $12^3 = 1{,}728$ `flop`. Here, instead of constructing the 3-D Loewner matrix $\mathbb{L}_3$ as in Example 4.15, one may construct a sequence of 1-D Loewner matrices using a recursive approach as follows.

- First, a 1-D Loewner matrix along the first variable ${}^1s$ for frozen second and third variables ${}^2\lambda_2 = 3$ and ${}^3\lambda_3 = 7$, i.e., elements of $\mathbf{tab}_3(:, 2, 3)$, is constructed, leading to
$$\mathbb{L}_1^{({}^2\lambda_2, {}^3\lambda_3)} = \begin{bmatrix} \frac{31}{2700} & \frac{31}{2800} \\ \frac{31}{2592} & \frac{31}{2688} \end{bmatrix} \text{ and } \mathbf{c}_1^{({}^2\lambda_2, {}^3\lambda_3)} = \begin{bmatrix} -\frac{27}{28} \\ 1 \end{bmatrix}.$$
- Second, as ${}^1\lambda_j$ is of dimension two ($k_1 = 2$), two 2-D Loewner matrices appear, one for frozen ${}^1\lambda_1$ and one for frozen ${}^1\lambda_2$, along ${}^2s$ and ${}^3s$, i.e., elements of $\mathbf{tab}_3(1, :, :)$ and $\mathbf{tab}_3(2, :, :)$. The first and second 2-D Loewner matrices lead to null spaces spanned by
$$\mathbf{c}_2^{{}^1\lambda_1} = \left[\frac{-14}{27}, \frac{13}{9}, \frac{-26}{27}, \frac{5}{9}, \frac{-41}{27}, 1\right]^\top, \mathbf{c}_2^{{}^1\lambda_2} = \left[\frac{-15}{28}, \frac{41}{28}, \frac{-27}{28}, \frac{4}{7}, \frac{-43}{28}, 1\right]^\top,$$

[6] We assume here that $n = k_1 + q_1$, $m = k_2 + q_2$, and $p = k_3 + q_3$, and $k_1 = q_1$, $k_2 = q_2$, and $k_3 = q_3$. In other configurations, specific treatment is needed.

which can now be scaled by the coefficients of $\mathbf{c}_1^{({}^2\lambda_2,{}^3\lambda_3)}$, leading to

$$\hat{\mathbf{c}}_3 = \left[\begin{array}{cc} \mathbf{c}_2^{{}^1\lambda_1}\cdot[\mathbf{c}_1^{({}^2\lambda_2,{}^3\lambda_3)}]_1 & \mathbf{c}_2^{{}^1\lambda_2}\cdot[\mathbf{c}_1^{({}^2\lambda_2,{}^3\lambda_3)}]_2 \end{array}\right]^\top = \mathbf{c}_3.$$

By considering the first 2-D Loewner matrix $\mathbb{L}_2^{{}^1\lambda_1}$, leading to the null space $\mathbf{c}_2^{{}^1\lambda_1}$, the very same process as that presented in the previous subsection (2-D case) may be performed (to avoid the 2-D matrix construction). In what follows we describe this iteration (for $\mathbf{c}_2^{{}^1\lambda_1}$ only, as it similarly applies to $\mathbf{c}_2^{{}^1\lambda_2}$).

- First, one constructs the 1-D Loewner matrix along the second variable ${}^2s$ for frozen first and third variables, i.e., elements of $\mathbf{tab}_3(1,:,3)$, leading to

$$\mathbb{L}_1^{({}^1\lambda_1,{}^3\lambda_3)} = \left[\begin{array}{cc} \frac{71}{520} & \frac{71}{540} \\ \frac{355}{2496} & \frac{355}{2592} \end{array}\right] \text{ and } \mathbf{c}_1^{({}^1\lambda_1,{}^3\lambda_3)} = \left[\begin{array}{c} -\frac{26}{27} \\ 1 \end{array}\right].$$

- Second, as ${}^2\lambda_{k_2}$ is of dimension two ($k_2 = 2$), two 1-D Loewner matrices appear, one for frozen ${}^2\lambda_1$ and one for frozen ${}^2\lambda_2$, along ${}^3s$ (here again, ${}^1s$ is frozen to ${}^1\lambda_1$). The first and second 1-D Loewner matrices lead to the null spaces

$$\mathbf{c}_1^{({}^1\lambda_1,{}^2\lambda_1)} = \left[\begin{array}{ccc} \frac{7}{13} & -\frac{3}{2} & 1 \end{array}\right]^\top \text{ and } \mathbf{c}_1^{({}^1\lambda_1,{}^2\lambda_2)} = \left[\begin{array}{ccc} \frac{5}{9} & -\frac{41}{27} & 1 \end{array}\right]^\top,$$

which can now be scaled by the coefficients of $\mathbf{c}_1^{({}^1\lambda_1,{}^3\lambda_3)}$, leading to

$$\left[\begin{array}{c} \mathbf{c}_1^{({}^1\lambda_1,{}^2\lambda_1)}\cdot[\mathbf{c}_1^{({}^1\lambda_1,{}^3\lambda_3)}]_1 \\ \mathbf{c}_1^{({}^1\lambda_1,{}^2\lambda_2)}\cdot[\mathbf{c}_1^{({}^1\lambda_1,{}^3\lambda_3)}]_2 \end{array}\right] = \mathbf{c}_2^{{}^1\lambda_1}.$$

Scaling $\mathbf{c}_2^{{}^1\lambda_1}$ with the first element of $\mathbf{c}_1^{({}^2\lambda_2,{}^3\lambda_3)}$ then leads to $\mathbf{c}_{3,1}^\top$.

This step is repeated for $\mathbb{L}_2^{{}^1\lambda_2}$, leading, to the null space $\mathbf{c}_2^{{}^1\lambda_2}$. The later is scaled with the second element of $\mathbf{c}_1^{({}^2\lambda_2,{}^3\lambda_3)}$, leading to $\mathbf{c}_{3,2}^\top$. By checking the complexity, one observes that only a collection of 1-D Loewner matrices needs to be constructed, as well as their null spaces. Here are constructed (i) one 1-D Loewner matrix along ${}^1s$ of dimension 2×2 and (ii) two 2-D Loewner matrices along ${}^2s$ and ${}^3s$, recast as two 1-D Loewner matrices along ${}^2s$ of dimension 2×2 and four 1-D Loewner matrices along ${}^3s$ of dimension 3×3. The resulting complexity is $(1\times 2^3)+(2\times 2^3)+(4\times 3^3) = 132$ `flop`, which is much less than 1,728 `flop` for $\mathbb{L}_3$. One may also notice that changing the variable orders as ${}^1s \leftarrow {}^3s$ and ${}^3s \leftarrow {}^1s$ would lead to $(1\times 3^3)+(3\times 2^3)+(6\times 2^3) = 99$. In both cases, the multivariate Loewner matrices are no longer needed and can be replaced by a series of single variables, taming the C-o-D.

5.3. Null Space Computation in the n-D Case and Variable Decoupling. We now state the second main result of this paper: Theorems 5.8 and 5.9 which allow us to address the **C-o-D** related to the null space computation of the n-D Loewner matrix. This is achieved by splitting an n-D Loewner matrix null space into one 1-D and a collection of $(n-1)$-D null spaces, thus another sequence of 1-D and $(n-2)$-D null spaces, and so on….

Theorem 5.8. *Given the tableau* $\boldsymbol{tab}_n$ *as in Table* 2, *being the evaluation of the* n*-variable* $\mathfrak{H}$ *function* (1.2) *at the data set* (3.3), *the null space of the corresponding* n*-D Loewner matrix* $\mathbb{L}_n$ *is spanned by*

$$\mathbf{vec}\left[\mathbf{c}_{n-1}^{{}^1\lambda_1}\cdot\left[\mathbf{c}_1^{({}^2\lambda_{k_2},{}^3\lambda_{k_3},\ldots,{}^n\lambda_{k_n})}\right]_1,\ldots,\mathbf{c}_{n-1}^{{}^1\lambda_{k_1}}\cdot\left[\mathbf{c}_1^{({}^2\lambda_{k_2},{}^3\lambda_{k_3},\ldots,{}^n\lambda_{k_n})}\right]_{k_1}\right],$$

where $\mathbf{c}_1^{({}^2\lambda_{k_2},{}^3\lambda_{k_3},\ldots,{}^n\lambda_{k_n})}$ *spans* $\mathcal{N}(\mathbb{L}_1^{({}^2\lambda_{k_2},{}^3\lambda_{k_3},\ldots,{}^n\lambda_{k_n})})$, *i.e., the nullspace of the* 1*-D Loewner matrix for frozen* $\{{}^2s,{}^3s,\ldots,{}^ns\}=\{{}^2\lambda_{k_2},{}^3\lambda_{k_3},\ldots,{}^n\lambda_{k_n}\}$, *and* $\mathbf{c}_{n-1}^{{}^1\lambda_j}$ *spans* $\mathcal{N}(\mathbb{L}_{n-1}^{{}^1\lambda_j})$, *i.e., the jth null space of the* $(n-1)$*-D Loewner matrix for frozen* ${}^1s_j=\{{}^1\lambda_1,\ldots,{}^1\lambda_{k_1}\}$.

Proof. The proof follows that given for the 2-D and 3-D cases. □

Theorem 5.8 provides a means to compute the null space of an n-D Loewner matrix via one 1-D and k_1 $(n-1)$-D Loewner matrices. Evidently, the latter $(n-1)$-D Loewner matrix null spaces may also be obtained by k_1 1-D Loewner matrices plus k_1k_2 $(n-2)$-D Loewner matrices. This reveals a recursive scheme that splits the n-D Loewner matrix into a set of 1-D Loewner matrices. As a consequence, the following decoupling theorem holds.

Theorem 5.9. *Given data* (3.3) *and Theorem* 5.8, *the latter achieves decoupling of the variables and the null space can be equivalently written as*

$$\mathbf{c}_n=\underbrace{\mathbf{c}^{{}^ns}}_{\mathbf{Bary}({}^ns)}\odot\underbrace{(\mathbf{c}^{{}^{n-1}s}\otimes\mathbf{1}_{k_n})}_{\mathbf{Bary}({}^{n-1}s)}\odot\underbrace{(\mathbf{c}^{{}^{n-2}s}\otimes\mathbf{1}_{k_nk_{n-1}})}_{\mathbf{Bary}({}^{n-2}s)}\odot\cdots\odot\underbrace{(\mathbf{c}^{{}^1s}\otimes\mathbf{1}_{k_n\ldots k_2})}_{\mathbf{Bary}({}^1s)},\tag{5.3}$$

where $\mathbf{c}^{{}^ls}$ *denotes the vector of barycentric coefficients related to the lth variable.*

As an illustration, in Theorem 5.9, $\mathbf{c}^{{}^1s}=\mathbf{c}_1^{({}^2\lambda_{k_2},{}^3\lambda_{k_3},\ldots,{}^n\lambda_{k_n})}$, while $\mathbf{c}^{{}^2s}$ is the vectorized collection of k_1 vectors $\mathbf{c}_1^{({}^1\lambda_1,{}^3\lambda_{k_3},\ldots,{}^n\lambda_{k_n})},\ldots,\mathbf{c}_1^{({}^1\lambda_{k_1},{}^3\lambda_{k_3},\ldots,{}^n\lambda_{k_n})}$, and so on. In section 6 and (6.1), an illustrative numerical example is given. Next, we assess how much this contributes to taming the **C-o-D**, in terms of both `flop` and memory savings.

5.4. Summary of Complexity, Memory Requirements, and Accuracy. Let us now state the main complexity result, related to Theorem 5.8, which is stated in Theorems 5.10 and 5.13, being the two major justifications for taming the **C-o-D**. They describe the drastic reductions in the computational complexity and the required memory.

Theorem 5.10. *The* `flop` *count for the recursive approach Theorem* 5.8 *is*

$$\texttt{flop}_1=\sum_{j=1}^{n}\left(k_j^3\prod_{l=1}^{j}k_{l-1}\right),\quad \textit{where}\ \ k_0=1.\tag{5.4}$$

Proof. Consider a function in n variables ${}^ls$ of degree $d_l>1$, $l=1,\ldots,n$ (and let $k_l=d_l+1$). Table 4 shows the complexity as a function of the number of variables.

Hence, the total number of `flop` required to compute an element of the null space of the n-D Loewner matrix $\mathbb{L}_n$ is

$$\begin{aligned}\texttt{flop}_1&=k_1^3+(k_1)\,k_2^3+\cdots+(k_1k_2\cdots k_{n-2})\,k_{n-1}^3+(k_1k_2\cdots k_{n-2}k_{n-1})\,k_n^3\\&=k_1^3+k_1\left(k_2^3+k_2\left(k_3^3+\cdots k_{n-2}\left(k_{n-1}^3+k_{n-1}\left(k_n^3\right)\right)\right)\right).\end{aligned}$$ □

Table 4 *Complexity table as a function of the number of variables.*

# of variables of $\mathfrak{H}$	# $\mathbb{L}_1$ matrix	Size of each $\mathbb{L}_1$	`flop` per $\mathbb{L}_1$
n	$k_1\, k_2 \cdots k_{n-2}\, k_{n-1}$	k_n	k_n^3
$n-1$	$k_1\, k_2 \cdots k_{n-2}$	k_{n-1}	k_{n-1}^3
$\vdots$	$\vdots$	$\vdots$	$\vdots$
3	$k_1\, k_2$	k_3	k_3^3
2	k_1	k_2	k_2^3
1	1	k_1	k_1^3

Corollary 5.11. *The variable arrangement that minimizes the* `flop` *cost is the one obtained by reordering the variables* ^{l}*s in decreasing complexity order* d_l*, i.e.,* $d_l \geq d_{l+1}$ *for* $l = 1, \dots, n-1$.

Corollary 5.12. *The most computationally demanding configuration occurs when each* ^{l}*s order satisfies* $d_l = k_l - 1 = k - 1$ $(l = 1, \dots, n)$*, requiring* k *interpolation points each. The worst case* `flop` *is (note that* $N = k^n$*)*

$$\overline{\texttt{flop}_1} = k^3 + k^4 + \cdots + k^{n+2} = k^3 \frac{1-k^n}{1-k} = k^3 \frac{1-N}{1-k}. \tag{5.5}$$

Note that (5.5) is an (n finite) geometric series of ratio k. Consequently, an upper bound of (5.5) can be estimated by considering that $k > 1$ and for a different number of variables n. As an example, for $n = \{1, 2, 3, 4, \dots\}$, the complexity is upper bounded by $\{\mathcal{O}(N^3), \mathcal{O}(N^{2.30}), \mathcal{O}(N^{1.94}), \mathcal{O}(N^{1.73}), \dots\}$, respectively. One can clearly observe that when the number of variables $n > 1$, the flop complexity drops to 2.30, and this decreases as n increases; e.g., for $n = 50$, one obtains $\mathcal{O}(N^{1.06})$.

In Figure 1, we show the result in Theorem 5.10 (cascaded n-D Loewner) and compare it to the reference full $\mathbb{L}_n$ null space computation via `SVD`,[7] of complexity $\mathcal{O}(N^3)$ and with $\mathcal{O}(N^2)$ and $\mathcal{O}(N \log(N))$ references. In the same figure, we evaluate the worst case (5.5) for different numbers of considered variables $n = \{1, 2, \dots, 50\}$ (each is evaluated with complexity $k = 1, \dots, 50$). Then, we evaluate an upper complexity approximate of the form $\mathcal{O}(N^x)$, where $x > 0$ is to be an upper bound of the data set.

With similar importance, the data storage is a key element in the **C-o-D**. In complex and double precision, the construction of the n-D Loewner matrix $\mathbb{L}_n \in \mathbb{C}^{N \times N}$, where $N = k_1 k_2 \cdots k_n$, requires disk storage of $\frac{8}{2^{20}} N^2$ MB. The following theorem states the result in the 1-D case.

Theorem 5.13. *Following the procedure in Theorem* 5.8*, one only needs to sequentially construct single* 1*-D Loewner matrices, each of dimension* $\mathbb{L}_1 \in \mathbb{C}^{k_l \times k_l}$*. The largest stored matrix is* $\mathbb{L}_1 \in \mathbb{C}^{k_{max} \times k_{max}}$*, where* $k_{max} = \max_l k_l$ $(l = 1, \dots, n)$*. In complex and double precision, the maximum disk storage is* $\frac{8}{2^{20}} k_{max}^2$ *MB.*

As an illustration, for a 6-variable problem with complexity $[19, 5, 3, 5, 7, 1]$, one requires $[k_1, k_2, k_3, k_4, k_5, k_6] = [20, 6, 4, 6, 8, 2]$ points, and then $N = 46{,}080$. The n-D Loewner matrix requires 31.64 `GB` of storage, while the 1-D version would require, in the worst-case scenario (i.e., for $k_{\max} = 20$), only 6.25 `KB` of storage.

[7] One should note that we are considering here the case $K = Q = N$ to simplify the exposition.

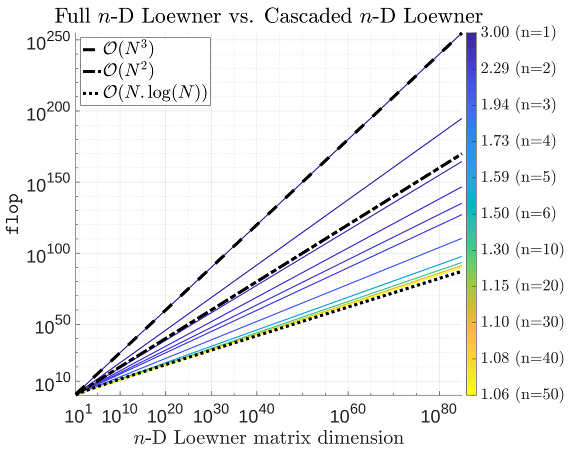

Fig. 1 `flop` *comparison: Cascaded n-D Loewner worst-case upper bounds for a varying number of variables n, while the full n-D Loewner is $\mathcal{O}(N^3)$ (black dashed); comparisons with $\mathcal{O}(N^2)$ and $\mathcal{O}(N\log(N))$ references are shown by dash-dotted and dotted black lines.*

Remark 5.14. In addition to computational complexity and storage, this method improves the numerical accuracy. For instance, in the modest case of a function with complexity $[9,7,2]$, the rank of the 3-D Loewner matrix in floating point is much bigger than one. The method proposed in this paper, therefore, makes the computation of the barycentric weights possible.

From the above considerations, it follows that the proposed null space computation method leads to a drop in not only the computational complexity of the worst-case scenario but also the memory requirements.

As illustrated in section 8, this allows the treatment of problems with a large number of variables, in a reasonable computational time and with manageable complexity, which is the main reason for claiming that the **C-o-D has been "tamed."**

6. Connection to the Kolmogorov Superposition Theorem.

Several researchers have contributed to sharpening Kolmogorov's original result, so currently it is often referred to as the *Kolmogorov, Arnol'd, Kahane, Lorenz, and Sprecher* Theorem (see [38], Theorem 2.1). For simplicity, we will follow [38] and state this result for $n=3$, so that we can compare it with Theorem 5.9.

Theorem 6.1. *Given a continuous function $f:[0,1]^3\to\mathbb{R}$ of three variables, there exist real numbers λ_i, $i=1,2$, single-variable continuous functions $\phi_k:[0,1]\to\mathbb{R}$, $k=1,\ldots,7$, and a single-variable function $g:\mathbb{R}\to\mathbb{R}$ such that*

$$f(x_1,x_2,x_3)=\sum_{k=1}^{7} g(\phi_k(x_1)+\lambda_1\phi_k(x_2)+\lambda_2\phi_k(x_3))\,\forall(x_1,x_2,x_3)\in[0,1]^3\,.$$

In the above result, λ_i and ϕ_k do not depend on f. Thus, for $n=3$, eight functions are needed together with two real scalars λ_i.

The goal of this section is to make contact with KST using a three-variable example.

Example 6.2. Consider the three-variable function $\mathfrak{H}(s,t,x)=\frac{s^2+xs+1}{t+x+st+2}$. Since the degrees in each variable are $(2,1,1)$, we will need the integers $k_1=3$, $k_2=2$, and $k_3=2$. This implies that $N=k_1k_2k_3=12$. The right and left interpolation points are chosen as $s_1 = 1$, $s_2 = 2$, $s_3 = 3$; $t_1 = 4$, $t_2 = 5$; $x_1 = 6$, $x_2 = 7$, and $s_4 = 3/2$, $s_5 = 5/2$, $s_6 = 7/2$; $t_3 = 9/5$, $t_4 = 11/5$; $x_3 = 13/3$, $x_4 = 5$, respectively. Following the theory developed above, the right triples of interpolation points are $\mathbf{S} = [\mathbf{s}_1,\, \mathbf{s}_2,\, \mathbf{s}_3] \otimes \mathbf{1}_2 \otimes \mathbf{1}_2$, $\mathbf{T} = \mathbf{1}_3 \otimes [\mathbf{t}_1,\, \mathbf{t}_2] \otimes \mathbf{1}_2$, $\mathbf{X} = \mathbf{1}_3 \otimes \mathbf{1}_2 \otimes [\mathbf{x}_1,\, \mathbf{x}_2] \in \mathbb{C}^{1\times N}$ (where $\mathbf{s}_i = s - s_i$, $\mathbf{t}_i = t - t_i$, and $\mathbf{x}_i = x - x_i$). Thus, the resulting 3-D Loewner matrix has dimension 12×12, with the 12 barycentric weights given by (for emphasis, we denote by **Bary** what was earlier denoted by $\mathbf{c}$)

$$\mathbf{Bary} = \begin{bmatrix} \frac{16}{29} & -\frac{17}{29} & -\frac{18}{29} & \frac{19}{29} & -\frac{40}{29} & \frac{42}{29} & \frac{46}{29} & -\frac{48}{29} & \frac{24}{29} & -\frac{25}{29} & -\frac{28}{29} & 1 \end{bmatrix}^\top .$$

As already shown, a decomposition of this vector follows in a (pointwise) product of barycentric weights with respect to each variable, separately. Thus, *decoupling* of the problem is achieved (which is one of the important aspects of KST), and the following is obtained: $\mathbf{Bary} = \mathbf{Bary}_x \odot\, \mathbf{Bary}_t \odot \mathbf{Bary}_s$, where $\odot$ denotes the pointwise product. This is (5.3) for $n = 3$ and is the key result that allows the connection with KST and the taming of the C-o-D. We have shown that the 3-D multivariate function can be computed in terms of three 1-D functions (one in each variable). These functions denoted below by $\boldsymbol{\Phi}(x)$, $\boldsymbol{\Psi}(t)$, and $\boldsymbol{\Omega}(s)$ are obtained from a collection of null space computations: 1 along s, 3 along t, and 6 along x. More specifically, following notations of Theorem 5.9,

$$\begin{aligned}
&\mathbf{c}^x = \mathrm{vec}\begin{pmatrix} -\frac{16}{17} & -\frac{18}{19} & -\frac{20}{21} & -\frac{23}{24} & -\frac{24}{25} & -\frac{28}{29} \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}, && \mathbf{Bary}_x = \mathbf{c}^x, \\
&\mathbf{c}^t = \mathrm{vec}\begin{pmatrix} -\frac{17}{19} & -\frac{7}{8} & -\frac{25}{29} \\ 1 & 1 & 1 \end{pmatrix}, && \mathbf{Bary}_t = \mathbf{c}^t \otimes \mathbf{1}_3, \\
&\mathbf{c}^s = \mathrm{vec}\begin{pmatrix} \frac{19}{29} & -\frac{48}{29} & 1 \end{pmatrix}, && \mathbf{Bary}_s = \mathbf{c}^s \otimes \mathbf{1}_{3\cdot 2}.
\end{aligned} \tag{6.1}$$

Furthermore, $\mathbf{Lag}(x)$, $\mathbf{Lag}(t)$, and $\mathbf{Lag}(s)$ are the monomials of the *Lagrange bases components* in each variable. Finally, $\mathbf{W}$ are the right interpolation values for the triples in $\mathbf{S} \times \mathbf{T} \times \mathbf{X}$. The ensuing numerical values are as follows:

$$\underbrace{\begin{bmatrix} -\frac{16}{17} \\ 1 \\ -\frac{18}{19} \\ 1 \\ -\frac{20}{21} \\ 1 \\ -\frac{23}{24} \\ 1 \\ -\frac{24}{25} \\ 1 \\ -\frac{28}{29} \\ 1 \end{bmatrix}}_{\mathbf{Bary}_x}, \underbrace{\begin{bmatrix} -\frac{17}{19} \\ -\frac{17}{19} \\ 1 \\ 1 \\ -\frac{7}{8} \\ -\frac{7}{8} \\ 1 \\ 1 \\ -\frac{25}{29} \\ -\frac{25}{29} \\ 1 \\ 1 \end{bmatrix}}_{\mathbf{Bary}_t}, \underbrace{\begin{bmatrix} \frac{19}{29} \\ \frac{19}{29} \\ \frac{19}{29} \\ \frac{19}{29} \\ -\frac{48}{29} \\ -\frac{48}{29} \\ -\frac{48}{29} \\ -\frac{48}{29} \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}}_{\mathbf{Bary}_s}, \underbrace{\begin{bmatrix} \frac{1}{x-6} \\ \frac{1}{x-7} \\ \frac{1}{x-6} \\ \frac{1}{x-7} \\ \frac{1}{x-6} \\ \frac{1}{x-7} \\ \frac{1}{x-6} \\ \frac{1}{x-7} \\ \frac{1}{x-6} \\ \frac{1}{x-7} \\ \frac{1}{x-6} \\ \frac{1}{x-7} \end{bmatrix}}_{\mathbf{Lag}(x)}, \underbrace{\begin{bmatrix} \frac{1}{t-4} \\ \frac{1}{t-4} \\ \frac{1}{t-5} \\ \frac{1}{t-5} \\ \frac{1}{t-4} \\ \frac{1}{t-4} \\ \frac{1}{t-5} \\ \frac{1}{t-5} \\ \frac{1}{t-4} \\ \frac{1}{t-4} \\ \frac{1}{t-5} \\ \frac{1}{t-5} \end{bmatrix}}_{\mathbf{Lag}(t)}, \underbrace{\begin{bmatrix} \frac{1}{s-1} \\ \frac{1}{s-1} \\ \frac{1}{s-1} \\ \frac{1}{s-1} \\ \frac{1}{s-2} \\ \frac{1}{s-2} \\ \frac{1}{s-2} \\ \frac{1}{s-2} \\ \frac{1}{s-3} \\ \frac{1}{s-3} \\ \frac{1}{s-3} \\ \frac{1}{s-3} \end{bmatrix}}_{\mathbf{Lag}(s)}, \underbrace{\begin{bmatrix} \frac{1}{2} \\ \frac{9}{17} \\ \frac{4}{9} \\ \frac{9}{19} \\ \frac{17}{20} \\ \frac{19}{21} \\ \frac{17}{23} \\ \frac{19}{24} \\ \frac{7}{6} \\ \frac{31}{25} \\ 1 \\ \frac{31}{29} \end{bmatrix}}_{\mathbf{W}},$$

$$\overset{\text{def}}{\Rightarrow} \begin{cases} \boldsymbol{\Phi}(x) = \mathbf{Bary}_x \odot \mathbf{Lag}(x), \\ \boldsymbol{\Psi}(t) = \mathbf{Bary}_t \odot \mathbf{Lag}(t), \\ \boldsymbol{\Omega}(s) = \mathbf{Bary}_s \odot \mathbf{Lag}(s). \end{cases}$$

With the above notation, we can express $\mathbf{H}$ as the quotient of two rational functions:

$$\left.\begin{array}{rcl}\hat{\mathbf{n}}(s,t,x) & = & \sum\limits_{\text{rows}}[\mathbf{W}\odot\mathbf{\Phi}(x)\odot\mathbf{\Psi}(t)\odot\mathbf{\Omega}(s)]\\ \hat{\mathbf{d}}(s,t,x) & = & \sum\limits_{\text{rows}}[\mathbf{\Phi}(x)\odot\mathbf{\Psi}(t)\odot\mathbf{\Omega}(s)]\end{array}\right\}\Rightarrow\frac{\hat{\mathbf{n}}(s,t,x)}{\hat{\mathbf{d}}(s,t,x)}=\mathbf{H}(s,t,x).$$

Consequently, KST for rational functions, as *composition and superposition* of one-variable functions, takes the form

$$\left.\begin{array}{rcl}\hat{\mathbf{n}}(s,t,x) & = & \sum\limits_{\text{rows}}\exp\left[\log\mathbf{W}+\log\mathbf{\Phi}(x)+\log\mathbf{\Psi}(t)+\log\mathbf{\Omega}(s)\right],\\ \hat{\mathbf{d}}(s,t,x) & = & \sum\limits_{\text{rows}}\exp\left[\log\mathbf{\Phi}(x)+\log\mathbf{\Psi}(t)+\log\mathbf{\Omega}(s)\right].\end{array}\right\}\tag{6.2}$$

Similarities and Differences between KST and the Results in (5.3) and (6.2).

(a) While KST refers to continuous functions defined on $[0,1]^n$, (6.2) is concerned with rational functions defined on $\mathbb{C}^n$.

(b) Expressions in (6.2) are valid in a particular basis, namely, the *Lagrange* basis. Multiplication of functions in (6.2) is defined with respect to this basis.

(c) The composition and superposition properties hold for the numerator and denominator. This is important in our case because (6.2) preserves interpolation conditions.

(d) The parameters needed are $n=3$ Lagrange bases (one in each variable) and the barycentric coefficients of the numerator and denominator. Note that in KST, no explicit denominators are considered.

(e) Both KST and (6.2) accomplish the goal of replacing the computation of multivariate functions by means of a series of computations involving single-variable functions, KST for general continuous functions, and (6.2) for rational functions. Notice also that (6.2) provides a different formulation of the problem than KST.

(f) In addition to the Kolmogorov–Arnold neural nets (KANs) [33], our approach provides a new application of KST to the modeling of multiparameter systems.

7. Data-Driven Multivariate Model Approximation.

This section focuses on the numerical aspects of constructing the realization from data measurements.

7.1. Two Algorithms.

In what follows, we detail two algorithms. The first is a direct method extending the algorithm proposed by the authors in [29], while the second is an iterative method inspired by the p-AAA presented in [43]. These procedures are outlined in Algorithms 7.1 and 7.2. For additional details, see also [29, 43].

7.2. Discussion.

The main difference between the two algorithms is that Algorithm 7.1 is direct while Algorithm 7.2 is iterative. Indeed, in the former case, the order is estimated at step 2, while the order is iteratively increased in the latter case until a given accuracy is reached.

By analyzing Algorithm 7.1, the process first needs to estimate the rational order along each variable ${}^{l}s$. Then, we construct the interpolation set (3.3) (here, one may shuffle data and interpolate different blocks). From this initial data set, the n-D

Algorithm 7.1. Direct data-driven pROM construction.

```
Require: $\mathbf{tab}_n$ as in Table 2
  Check that interpolation points are disjoint.
  Compute $d_l = \max_k \mathbf{rank}\, \mathbb{L}_1^{(k)}$, the order along variable ${}^l s$ ($k$ is the number of all
    1D Loewner matrices when fixing variables $\{{}^1 s, \ldots, {}^{k-1} s, {}^{k+1} s, \ldots, {}^n s\}$).
  Construct (3.3), a subselection $P_c^{(n)}$ where $(k_1, k_2, \ldots, k_n) = (d_1, d_2, \ldots, d_n) + 1$,
    and $P_r^{(n)}$, where $(q_1, q_2, \ldots, q_n)$ gather the rest of the data.
  Compute $\mathbf{c}_n$, the $n$-D Loewner matrix null space, e.g., using Theorem 5.8.
  Construct $\mathbb{A}^{\mathrm{Lag}}$, $\mathbb{B}^{\mathrm{Lag}}$, $\mathbf{\Gamma}$, and $\mathbf{\Delta}$ as in Result 4.14 with any left/right
    separation.
  Construct multivariate realization as in Theorem 2.8.
Ensure: $\mathbf{H}({}^1 s, \ldots, {}^n s) = \mathbf{C}\mathbf{\Phi}({}^1 s, {}^2 s, \ldots, {}^n s)^{-1}\mathbf{G}$ interpolates $\mathfrak{H}({}^1 s, {}^2 s, \ldots, {}^n s)$ along
    $P_c^{(n)}$.
```

Algorithm 7.2. Adaptive data-driven pROM construction.

```
Require: $\mathbf{tab}_n$ as in Table 2 and tolerance tol $> 0$
  Check that interpolation points are disjoint.
  while error $>$ tol do
      Search the point indexes with maximal error (first iteration: pick any set).
      Add points in $P_c^{(n)}$ and put the remaining ones in $P_r^{(n)}$, to obtain (3.3).
      Compute $\mathbf{c}_n$, the $n$-D Loewner matrix null space, e.g., using Theorem 5.8.
      Construct $\mathbb{A}^{\mathrm{Lag}}$, $\mathbb{B}^{\mathrm{Lag}}$, $\mathbf{\Gamma}$, and $\mathbf{\Delta}$ as in Result 4.14 with any left/right
        separation.
      Construct multivariate realization as in Theorem 2.8.
      Evaluate error $= \max ||\widehat{\mathbf{tab}_n} - \mathbf{tab}_n||$, where $\widehat{\mathbf{tab}_n}$ is the evaluation of $\mathbf{H}({}^1 s,$
        $\ldots, {}^n s)$ along the support points.
  end while
Ensure: $\mathbf{H}({}^1 s, \ldots, {}^n s) = \mathbf{C}\mathbf{\Phi}({}^1 s, {}^2 s, \ldots, {}^n s)^{-1}\mathbf{G}$ interpolates $\mathfrak{H}({}^1 s, {}^2 s, \ldots, {}^n s)$ along
    $P_c^{(n)}$.
```

Loewner matrix and its null space may be computed using either the full (section 4) or the 1-D recursive (section 5) approach. Based on the barycentric weights, the realization is constructed using Result 4.14.

The difference between the two algorithms consists of the absence of the order detection process in the second algorithm. Instead, it is replaced by an evaluation of the model along the data set at each step until a tolerance is reached. Then, at each iteration, one adds the support points set where the maximal error between the model and the data occurs. This idea is originally exploited in the univariate case of AAA in [39] and its parametric version from [43]; we similarly follow this greedy approach.

7.2.1. Dealing with Real Arithmetic.

All computational steps have been presented using complex data. However, in applications, it is often desirable to deal with real-valued functions in order to preserve the realness of the realization and to allow the time-domain simulations of the differential-algebraic equations. To do so, some assumptions and adaptations must be satisfied. Basically, interpolation points along each variable must be either real or chosen to be closed under conjugations. For more details on the exact procedure, we refer the reader to [29, section A.2].

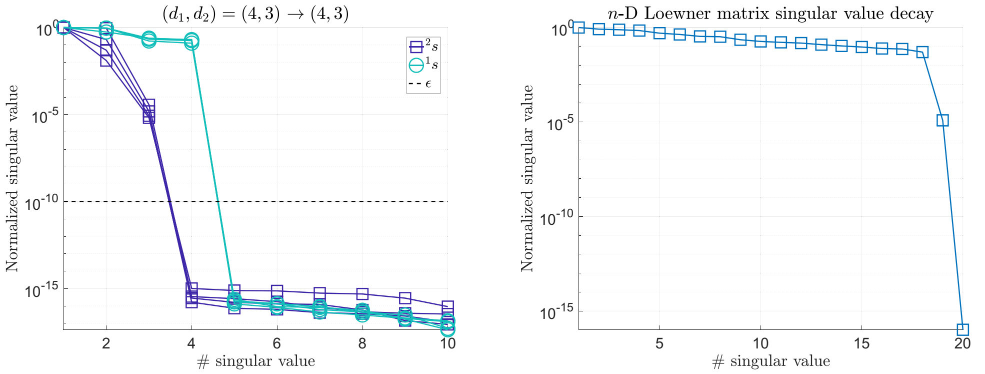

Fig. 2 *2-D simple synthetic model: Algorithm* 7.1 *normalized singular values of each* 1*-D (left) and the* 2*-D (right) Loewner matrices.*

7.2.2. Null Space Computation Remarks.

To apply the proposed methods to a broad range of real-life applications, we want to comment on the major computational effort and hard point in the proposed process: the **null space computation**. Indeed, in both the full n-D and the recursive 1-D cases, a null space must be computed. Numerically, there exist multiple ways to compute it: `SVD` or `QR` decomposition, linear resolution, etc. Without going into detail beyond the scope of this paper, many tuning variables may be adjusted to improve accuracy. These elements are crucial to the success of the proposed solution. In the next section, all null spaces have been computed using the standard `SVD` routine of `MATLAB`. For more detail, refer to [24].

8. Numerical Experiments.

The effectiveness of the numerical procedures sketched in Algorithms 7.1 and 7.2 is illustrated in this section, through examples involving multiple variables ranging from two to twenty. In what follows, the computations were performed on an Apple MacBook Air with 512 `GB SSD` and 16 `GB RAM`, with an `M1` processor. The software used was `MATLAB` 2023b.

8.1. A Simple Synthetic Parametric Model (2-D).

Let us start with the simple example used in [29, section 5.1] and [43, section 3.2.1], whose transfer function reads $\mathfrak{H}(s,p) = \frac{1}{1+25(s+p)^2} + \frac{0.5}{1+25(s-0.5)^2} + \frac{0.1}{p+25}$. We use the same sampling setting as in the above references. Along the s variable, 21 points are linearly spaced from $[-1, 1]$. For the direct method of Algorithm 7.1, we alternatively sample the grid as ${}^1\lambda_{j_1} = [-1, -0.8, \ldots, 1]$ and ${}^1\mu_{i_1} = [-0.9, -0.7, \ldots, 0.9]$; then, along the p variable, there are 21 linearly spaced points from $[0, 1]$. For the direct method of Algorithm 7.1 we alternatively sample the grid as ${}^2\lambda_{j_2} = [0, 0.1, \ldots, 1]$ and ${}^2\mu_{i_2} = [0.05, 0.15, \ldots, 0.95]$. First, we apply Algorithm 7.1 and obtain the single-variable singular value decay reported in Figure 2 (left), suggesting approximation orders along (s, p) of $(d_1, d_2) = (4, 3)$, being precisely that of the equation $\mathfrak{H}(s,p)$ above. Then, the 2-D Loewner matrix is constructed and its associated singular values are reported in Figure 2 (right), leading to the full null space and barycentric weights (results follow next).

Next, we investigate the behavior of Algorithm 7.2. In Table 5, we report the iterations of this algorithm when computing the null space with either the full 2-D version (Table 5(a)) or the recursive 1-D version (Table 5(b)). In both cases, the same order is recovered, i.e., $(4, 3)$. Even if the selected interpolation points are slightly different, the final error is below the chosen tolerance, i.e., `tol`$=10^{-6}$. By now comparing the `flop` complexity, the benefit of the proposed recursive 1-D approach

Table 5 *2-D simple model iterations with different null space computation methods.*

(a) Algorithm 7.2 (full $\mathbb{L}_n$)

Iter.	${}^1\lambda_{j_1}$	${}^2\lambda_{j_2}$	(k_1,k_2)	`flop`
1	0	0	$(1,1)$	1^3
2	-1		$(2,1)$	2^3
3	-0.9	0.9	$(3,2)$	6^3
4	-0.1	0.2	$(4,3)$	12^3
5	0.6	1	$(5,4)$	20^3

(b) Algorithm 7.2 (recursive $\mathbb{L}_1$)

${}^1\lambda_{j_1}$	${}^2\lambda_{j_2}$	(k_1,k_2)	`flop`
0	0	$(1,1)$	2
-1		$(2,1)$	10
0.1	0.05	$(3,2)$	51
-0.9	0.75	$(4,3)$	172
0.7	0.15	$(5,4)$	445

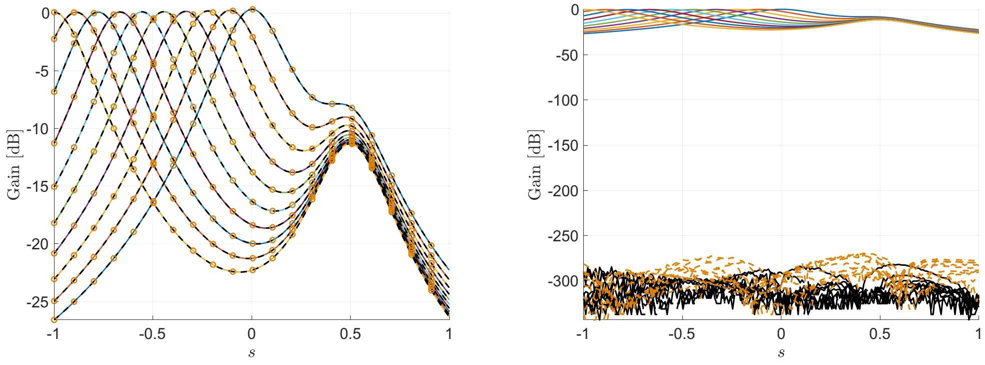

Fig. 3 *2-D simple model: Frequency responses (left) and errors (right); original vs. Algorithm* 7.1 *(black lines) and Algorithm* 7.2 *(orange dots and dashed lines).*

with respect to the 2-D approach is clearly emphasized, even for such a simple setup. Indeed, while the latter is of $= 1+2^3+6^3+12^3+20^3 = 9{,}953$ `flop`, the former leads to $2+10+51+172+445 = 680$ `flop`, which is 14 times smaller. The mismatch in the three configurations over all the sampling points of the $\mathbf{tab}_2$ data is close to machine precision for all configurations.

Finally, to conclude this first example, Figure 3 reports the responses (left) and mismatch (right) along s for different values of p, for the original model and the obtained models with Algorithms 7.1 and 7.2 (with recursive 1-D null space).

8.2. Flutter (3-D). This numerical example is extracted from industrial data and considers a mixed model/data configuration. It represents the flutter phenomena for flexible aircraft as detailed in [18].[8] This model can be described as $s^2M(m)x(s) + sB(m)x(s) + K(m)x(s) - G(s,v) = u(s)$, where $M(m), B(m), K(m) \in \mathbb{R}^{n\times n}$ are the mass, damping, and stiffness matrices, all dependent on the aircraft mass $m \in \mathbb{R}_+$ $(n \approx 100)$. These matrices are constant for a given flight point (but vary for a mass configuration). Then, the generalized aeroelastic forces $G(s,v) \in \mathbb{C}^{n\times n}$ describe the aeroelastic forces exciting the structural dynamics. This $G(s,v)$ is known only at a few sampled frequencies and some true airspeed, i.e., $G(\imath\omega_i, v_j)$, where $i = 1,\ldots,150$ and $j = 1,\ldots,10$. Note that these values are obtained through dedicated high-fidelity numerical solvers. The sampling setup is as follows. Along the s variable, ${}^1\lambda_{j_1}$ are 150 logarithmically spaced points between $\imath[10,35]$ and ${}^1\mu_{i_1} = -{}^1\lambda_{j_1}$. Along the v

[8] We acknowledge Pierre Vuillemin for generating the (modified) data.

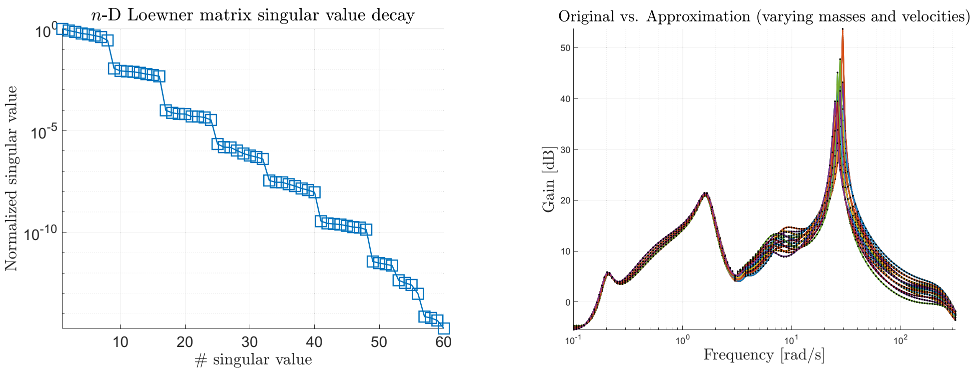

Fig. 4 *3-D flutter model: 3-D Loewner matrix singular values (left) and frequency responses (right). Original (solid colored) and pROM (black dotted).*

variable, ${}^2\lambda_{j_2}$ are five linearly spaced points between $[4.77, 5.21] \cdot 10^3$ and ${}^2\mu_{i_2}$ and five linearly spaced points between $[4.82, 5.27] \cdot 10^3$. Along the m variable, ${}^3\lambda_{j_3}$ are five linearly spaced points between $[1.52, 1.66] \cdot 10^3$ and ${}^2\mu_{i_2}$ and five linearly spaced points between $[1.54, 1.68] \cdot 10^3$.

Here, the data is a 3-D tensor $\mathbf{tab}_3 \in \mathbb{C}^{300\times 10\times 10}$. By applying Algorithm 7.1, an approximation order $(14, 1, 1)$ is reasonable. The singular value decay of the 3-D Loewner matrix is reported in Figure 4 (left). Then, the original and pROM frequency responses are shown in Figure 4 (right), resulting in an accurate model.

One relevant point of the proposed Loewner framework, nicely illustrated in this application, is its ability to construct a realization of a pROM based on a hybrid data set, mixing frequency-domain data and matrices. By connecting this problem to NEPs, the parametric rational approximation allows us to estimate the eigenvalue trajectories; we refer to [42, 47, 18] for details and industrial applications.

8.3. A Multivariate Function with a High Number of Variables (20-D).

To conclude and to numerically demonstrate the scalability features of our process, let us consider the 20-variable rational model $\mathfrak{H}({}^1s, \ldots, {}^{20}s) =$

$$\frac{3 \cdot {}^1s^3 + 4 \cdot {}^8s + {}^{12}s + {}^{13}s \cdot {}^{14}s + {}^{15}s}{{}^1s^{10} + {}^2s^2 \cdot {}^3s + {}^4s + {}^5s + {}^6s + {}^7s \cdot {}^8s + {}^9s \cdot {}^{10}s \cdot {}^{11}s + {}^{13}s + {}^{13}s^3 \cdot \pi + {}^{17}s + {}^{18}s \cdot {}^{19}s - {}^{20}s},$$

with a complexity of $(10, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1)$. By applying Algorithm 7.1 with the recursive 1-D null space construction, the barycentric coefficients $\mathbf{c}_n \in \mathbb{C}^{17301504}$ are obtained with a computational complexity of 149,226,836 `flop`, computed in 4 hours. As explained in the Supplementary Material, this vector allows the reconstruction of the original model with an absolute error $\approx 10^{-7}$ for a random parameter selection. Applying the full n-D Loewner version instead would theoretically require the construction of a Loewner matrix of dimension $N = 17,301,504$, with a null space computation costing about $5.18 \cdot 10^{21}$ `flop`, which is prohibitive on a standard computer. Storing such an $N \times N$ n-D Loewner matrix would require $4,356$ `TB` in double precision, while the 1-D approach needs 1.89 `KB` only (in the worst case).

9. Conclusions.

We have investigated the Loewner framework for linear multivariate/parametric systems and developed a complete methodology (and two algorithms) for data-driven n-variable pROM realization construction in the (n-D)

Loewner framework. We have also shown the relationship between n-D Loewner and Sylvester equations. Then, as the numerical complexity and matrix storage explode with the number of data points and variables, we introduced a recursive 1-D null space procedure, equivalent to the full n-D procedure. This process allows the decoupling of the variables involved and thus provides the effect of drastically reducing (i) the computational complexity and (ii) the matrix storage needs. This becomes a major step toward taming the curse of dimensionality. In addition, we have established a connection between the decoupling result and the Kolmogorov superposition theorem (KST). We have applied these results to numerical examples throughout the paper, demonstrating their effectiveness. Last, we claim that the contributions presented are not limited to the system dynamics and rational approximation fields, but may also apply to many scientific computing areas, including tensor approximation and nonlinear eigenvalue problems, for which dimensionality remains an issue.

Supplementary Material and Software Availability. Additional material to supplement the findings reported in this paper is available at

https://sites.google.com/site/charlespoussotvassal/nd_loew_tcod

and in [6] (where over 30 test cases are analyzed and various methods are compared in detail). Furthermore, the MATLAB code used to generate the figures and illustrations corresponding to the numerical results presented in this paper is available at

https://github.com/cpoussot/mLF

REFERENCES

[1] D. Amsallem and C. Farhat, *An online method for interpolating linear parametric reduced-order models*, SIAM J. Sci. Comput., 33 (2011), pp. 2169–2198, https://doi.org/10.1137/100813051. (Cited on p. 740)

[2] F. Andreuzzi, N. Demo, and G. Rozza, *A dynamic mode decomposition extension for the forecasting of parametric dynamical systems*, SIAM J. Appl. Dyn. Syst., 22 (2023), pp. 2432–2458, https://doi.org/10.1137/22M1481658. (Cited on p. 740)

[3] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, Adv. Des. Control 6, SIAM, Philadelphia, 2005, https://doi.org/10.1137/1.9780898718713. (Cited on p. 739)

[4] A. C. Antoulas and B. D. O. Anderson, *On the scalar rational interpolation problem*, IMA J. Math. Control. Inf., 3 (1986), pp. 61–88, https://doi.org/10.1093/imamci/3.2-3.61. (Cited on p. 742)

[5] A. C. Antoulas, C. A. Beattie, and S. Güğercin, *Interpolatory Methods for Model Reduction*, Comput. Sci. Engrg. 21, SIAM, Philadelphia, 2020, https://doi.org/10.1137/1.9781611976083. (Cited on p. 739)

[6] A. C. Antoulas, I. V. Gosea, C. Poussot-Vassal, and P. Vuillemin, *Tensor-Based Multivariate Function Approximation: Methods, Benchmarking, and Comparison*, preprint, 2025, https://doi.org/10.48550/arXiv.2506.04791. (Cited on p. 768)

[7] A. C. Antoulas, A. C. Ionita, and S. Lefteriu, *On two-variable rational interpolation*, Linear Algebra Appl., 436 (2012), pp. 2889–2915, https://doi.org/10.1016/j.laa.2011.07.017. (Cited on pp. 740, 742, 743, 748, 751)

[8] A. C. Antoulas, S. Lefteriu, and A. C. Ionita, *A tutorial introduction to the Loewner framework for model reduction*, in Model Reduction and Approximation, SIAM, 2017, pp. 335–376, https://doi.org/10.1137/1.9781611974829.ch8. (Cited on pp. 742, 750, 751)

[9] U. Baur, C. Beattie, P. Benner, and S. Gugercin, *Interpolatory projection methods for parameterized model reduction*, SIAM J. Sci. Comput., 33 (2011), pp. 2489–2518, https://doi.org/10.1137/090776925. (Cited on p. 740)

[10] U. Baur, P. Benner, and L. Feng, *Model order reduction for linear and nonlinear systems: A system-theoretic perspective*, Arch. Comput. Meth. Eng., 21 (2014), pp. 331–358, https://doi.org/10.1007/s11831-014-9111-2. (Cited on p. 739)

[11] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira, eds., *Model Order Reduction*, Vol. 1, System- and Data-Driven Methods and Algo-

rithms, De Gruyter, Berlin, Boston, 2021, https://doi.org/10.1515/9783110498967. (Cited on p. 739)

[12] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira, eds., *Model Order Reduction*, Vol. 2, Snapshot-Based Methods and Algorithms, De Gruyter, Berlin, Boston, 2021, https://doi.org/10.1515/9783110671490. (Cited on p. 739)

[13] P. Benner, S. Gugercin, and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Rev., 57 (2015), pp. 483–531, https://doi.org/10.1137/130932715. (Cited on pp. 739, 740)

[14] J.-P. Berrut and L. N. Trefethen, *Barycentric Lagrange interpolation*, SIAM Rev., 46 (2004), pp. 501–517, https://doi.org/10.1137/S0036144502417715. (Cited on p. 742)

[15] T. Bradde, S. Grivet-Talocia, A. Zanco, and G. C. Calafiore, *Data-driven extraction of uniformly stable and passive parameterized macromodels*, IEEE Access, 10 (2022), pp. 15786–15804, https://doi.org/10.1109/ACCESS.2022.3147034. (Cited on p. 740)

[16] M. C. Brennan, M. Embree, and S. Gugercin, *Contour integral methods for nonlinear eigenvalue problems: A systems theoretic approach*, SIAM Rev., 65 (2023), pp. 439–470, https://doi.org/10.1137/20M1389303. (Cited on p. 741)

[17] O. Debals, M. Van Barel, and L. De Lathauwer, *Löwner-based blind signal separation of rational functions with applications*, IEEE Trans. Signal Process., 64 (2015), pp. 1909–1918, https://doi.org/10.1109/TSP.2015.2500179. (Cited on p. 741)

[18] A. dos Reis de Souza, C. Poussot-Vassal, P. Vuillemin, and J. T. Zucco, *Aircraft flutter suppression: From a parametric model to robust control*, in Proceedings of European Control Conference, Bucharest, Romania, 2023, pp.1–6, https://doi.org/10.23919/ECC57647.2023.10178141. (Cited on pp. 766, 767)

[19] M. Geuss and B. Lohmann, *STABLE—A stability algorithm for parametric model reduction by matrix interpolation*, Math. Comput. Model. Dyn. Syst., 22 (2016), pp. 307–322, https://doi.org/10.1080/13873954.2016.1198383. (Cited on p. 740)

[20] I. V. Gosea, C. Poussot-Vassal, and A. C. Antoulas, *Data-driven modeling and control of large-scale dynamical systems in the Loewner framework*, in Handb. Numer. Anal. 23, E. Trelat and E. Zuazua, eds., 2022, pp. 499–530, https://doi.org/10.1016/bs.hna.2021.12.015. (Cited on pp. 742, 748)

[21] I. V. Gosea and S. Gugercin, *Data-driven modeling of linear dynamical systems with quadratic output in the AAA framework*, J. Sci. Comput., 91 (2022), 16, https://doi.org/10.1007/s10915-022-01771-5. (Cited on p. 740)

[22] I. V. Gosea, S. Gugercin, and B. Unger, *Parametric model reduction via rational interpolation along parameters*, in 60th IEEE Conference on Decision and Control (CDC), 2021, pp. 6895–6900, https://doi.org/10.1109/CDC45484.2021.9682841. (Cited on p. 740)

[23] L. Grasedyck, D. Kressner, and C. Tobler, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitt., 36 (2013), pp. 53–78, https://doi.org/10.1002/gamm.201310004. (Cited on p. 741)

[24] N. Guglielmi, M. L. Overton, and G. W. Stewart, *An efficient algorithm for computing the generalized null space decomposition*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 38–54, https://doi.org/10.1137/140956737. (Cited on p. 765)

[25] S. Güttel, G. M. Negri Porzio, and F. Tisseur, *Robust rational approximations of nonlinear eigenvalue problems*, SIAM J. Sci. Comput., 44 (2022), pp. A2439–A2463, https://doi.org/10.1137/20M1380533. (Cited on p. 741)

[26] J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, SpringerBriefs in Math., Springer, Cham, 2016, https://doi.org/10.1007/978-3-319-22470-1. (Cited on p. 739)

[27] D. Hilbert, *Mathematical problems*, Bull. Amer. Math. Soc. (N.S.), 37 (2000), pp. 407–436, https://doi.org/10.1090/S0273-0979-00-00881-8. (Cited on p. 740)

[28] M. Hund, T. Mitchell, P. Mlinarić, and J. Saak, *Optimization-based parametric model order reduction via $\mathcal{H}_2 \otimes \mathcal{L}_2$ first-order necessary conditions*, SIAM J. Sci. Comput., 44 (2022), pp. A1554–A1578, https://doi.org/10.1137/21M140290X. (Cited on p. 740)

[29] A. C. Ionita and A. C. Antoulas, *Data-driven parametrized model reduction in the Loewner framework*, SIAM J. Sci. Comput., 36 (2014), pp. A984–A1007, https://doi.org/10.1137/130914619. (Cited on pp. 740, 742, 743, 748, 749, 763, 764, 765)

[30] D. S. Karachalios, I. V. Gosea, and A. C. Antoulas, *The Loewner framework for system identification and reduction*, in Methods and Algorithms, P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. H. A. Schilders, and L. M. Silveira, eds., De Gruyter, 2021, pp. 181–228, https://doi.org/10.1515/9783110498967-006. (Cited on p. 742)

[31] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500, https://doi.org/10.1137/07070111X. (Cited on p. 740)

[32] P. Lietaert, K. Meerbergen, J. Pérez, and B. Vandereycken, *Automatic rational approximation and linearization of nonlinear eigenvalue problems*, IMA J. Numer. Anal., 42 (2022), pp. 1087–1115, https://doi.org/10.1093/imanum/draa098. (Cited on p. 741)

[33] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljacic, T. Hou, and M. Tegmark, *KAN: Kolmogorov-Arnold Neural Nets*, preprint, arXiv:2404.19756, 2024. (Cited on p. 763)

[34] K. Löwner, *Über monotone Matrixfunktionen*, Math. Z., 38 (1934), pp. 177–216, https://doi.org/10.1007/BF01170633. (Cited on p. 742)

[35] 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, https://doi.org/10.1016/j.laa.2007.03.008. (Cited on pp. 742, 748)

[36] S. A. McQuarrie, P. Khodabakhshi, and K. E. Willcox, *Nonintrusive reduced-order models for parametric partial differential equations via data-driven operator inference*, SIAM J. Sci. Comput., 45 (2023), pp. A1917–A1946, https://doi.org/10.1137/21M1452810. (Cited on p. 740)

[37] P. Mlinarić, P. Benner, and S. Gugercin, *Interpolatory Necessary Optimality Conditions for Reduced-Order Modeling of Parametric Linear Time-Invariant Systems*, preprint, arXiv:2401.10047, 2024. (Cited on p. 740)

[38] S. A. Morris, *Hilbert 13: Are there any genuine continuous multivariate functions?*, Bull. Amer. Math. Soc. (N.S.), 58 (2021), pp. 107–118, https://doi.org/10.1090/bull/1698. (Cited on pp. 740, 761)

[39] Y. Nakatsukasa, O. Sète, and L. N. Trefethen, *The AAA algorithm for rational approximation*, SIAM J. Sci. Comput., 40 (2018), pp. A1494–A1522, https://doi.org/10.1137/16M1106122. (Cited on p. 742, 764)

[40] B. Peherstorfer and K. Willcox, *Data-driven operator inference for nonintrusive projection-based model reduction*, Comput. Methods Appl. Mech Engrg., 306 (2016), pp. 196–215, https://doi.org/10.1016/j.cma.2016.03.025. (Cited on p. 740)

[41] G. Pólya and G. Szegö, *Problems and Theorems in Analysis*, Springer Verlag, 1972 (published in German by Springer in 1924). (Cited on p. 740)

[42] D. Quero, P. Vuillemin, and C. Poussot-Vassal, *A generalized eigenvalue solution to the flutter stability problem with true damping: The p-L method*, J. Fluids Structures, 103 (2021), art. 103266, https://doi.org/10.1016/j.jfluidstructs.2021.103266. (Cited on p. 767)

[43] A. C. Rodriguez, L. Balicki, and S. Gugercin, *The p-AAA algorithm for data-driven modeling of parametric dynamical systems*, SIAM J. Sci. Comput., 45 (2023), pp. A1332–A1358, https://doi.org/10.1137/20M1322698. (Cited on pp. 740, 742, 763, 764, 765)

[44] S. Sun, L. Feng, H. S. Chan, T. Miličić, T. Vidaković-Koch, and P. Benner, *Parametric Dynamic Mode Decomposition for Nonlinear Parametric Dynamical Systems*, preprint, arXiv:2305.06197, 2023. (Cited on p. 740)

[45] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, *On dynamic mode decomposition: Theory and applications*, J. Comput. Dyn., 1 (2014), pp. 391–421, https://doi.org/10.3934/jcd.2014.1.391. (Cited on p. 740)

[46] N. Vervliet, O. Debals, L. Sorber, and L. De Lathauwer, *Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis*, IEEE Signal Process. Mag., 31 (2014), pp. 71–79, https://doi.org/10.1109/MSP.2014.2329429. (Cited on p. 741)

[47] T. Vojkovic, D. Quero, C. Poussot-Vassal, and P. Vuillemin, *Low-order parametric state-space modeling of MIMO systems in the Loewner framework*, SIAM J. Appl. Dyn. Syst., 22 (2023), pp. 3130–3164, https://doi.org/10.1137/22M1509898. (Cited on pp. 740, 741, 749, 767)

[48] P. Vuillemin, P. Kergus, and C. Poussot-Vassal, *Hybrid Loewner data driven control*, in Proceedings of the IFAC World Congress, Berlin, Germany, Elsevier, 2020, pp. 5604–5608, https://doi.org/10.1016/j.ifacol.2020.12.1574. (Cited on p. 742)

[49] S. Yildiz, P. Goyal, P. Benner, and B. Karasözen, *Learning reduced-order dynamics for parametrized shallow water equations from data*, Internat. J. Numer. Methods Fluids, 93 (2021), pp. 2803–2821, https://doi.org/10.1002/fld.4998. (Cited on p. 740)

[50] Y. Yue, L. Feng, and P. Benner, *Reduced-order modelling of parametric systems via interpolation of heterogeneous surrogates*, Adv. Model. Simul. Eng. Sci., 6 (2019), 10, https://doi.org/10.1186/s40323-019-0134-y. (Cited on p. 740)

[51] A. Zanco, S. Grivet-Talocia, T. Bradde, and M. De Stefano, *Enforcing passivity of parameterized LTI macromodels via Hamiltonian-driven multivariate adaptive sampling*, IEEE Trans. Comput.-Aid. Design Integ. Circ. Syst., 39 (2018), pp. 225–238, https://doi.org/10.1109/TCAD.2018.2883962. (Cited on p. 740)