

The Loewner framework for parametric systems: Taming the curse of dimensionality

A. C. Antoulas* I. V. Gosea† C. Poussot-Vassal§

**Department of Electrical and Computer Engineering, Rice University, Houston, TX, USA.*
Email: aca@rice.edu

†*Max Planck Institute for Dynamics of Complex Technical Systems, CSC Group ,
Sandtorstr. 1, 39106 Magdeburg, Germany.*

Email: gosea@mpi-magdeburg.mpg.de, ORCID: [0000-0003-3580-4116](https://orcid.org/0000-0003-3580-4116)

§*DTIS, ONERA, Université de Toulouse, 31000, Toulouse, France.*

Email: charles.poussot-vassal@onera.fr, ORCID: [0000-0001-9106-1893](https://orcid.org/0000-0001-9106-1893)

Abstract: The Loewner framework is an interpolatory framework 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. One main innovation established here is the construction of data-based realizations for any number of parameters. Equally importantly, we show how to alleviate the computational burden, by avoiding the explicit construction of large-scale n -dimensional Loewner matrices of size $N \times N$. This reduces the complexity from $\mathcal{O}(N^3)$ to about $\mathcal{O}(N^{1.4})$, thus taming the curse of dimensionality and making the solution scalable to very large data sets. To achieve this, a new generalized multivariate rational function realization is defined. Then, we introduce the n -dimensional multivariate Loewner matrices and show that they can be computed by solving a coupled set of Sylvester equations. The null space of these Loewner matrices then allows the construction of the multivariate barycentric transfer function. The principal result of this work is to show how the null space of the n -dimensional Loewner matrix can be computed using a sequence of 1-dimensional Loewner matrices, leading to a drastic computational burden reduction. 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.

Keywords: parameterized linear systems, Loewner matrix, multivariate functions, barycentric rational interpolation, frequency response, interpolation methods, multivariate barycentric form, Lagrange polynomial basis, multivariate Loewner matrix, Sylvester equations, curse of dimensionality.

Novelty statement: First, we propose a generalized realization form for rational functions in n -variables (for any n), which are described in the Lagrange basis. Secondly, we show that the corresponding n -dimensional Loewner matrix is the solution of a cascaded Sylvester equation set. Finally, we demonstrate that the barycentric coefficients can be obtained using a sequence of 1-dimensional Loewner matrices instead of the large-scale n -dimensional one, drastically reducing the computational effort.

1 Introduction

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

1.1 Motivation and context: the non-intrusive data-driven model construction

Multivariate rational model interpolation addresses the problem of constructing a reduced order model that well captures the behavior of a potentially large-scale model depending on multiple variables. In the context of dynamical systems governed by differential and algebraic equations, the multivariate nature mainly comes from the parametric dependency of the underlying system or model. These parameters account for physical characteristics such as mass, length, or material property (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 motivations for dynamical multivariate/parametric reduced order model (pROM) construction stem from three inevitable facts about modern computing and actual 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 be always 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 large to be stored or simply inaccessible;
- (iii) Third, a potentially large number of parameters may be necessary, to be used in the next steps of the process.

Often, complex and accurate parametric models are needed by practitioners to perform simulations, forecasting, parametric uncertainty propagation, and optimization in a broad sense. The goals could be 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 multi-query model-based optimization process, the complexity dictates the accuracy, scalability, and applicability of the approach, it is relevant to seek a pROM with low complexity.

1.2 Literature overview

In the last decade, there has been a lot of effort invested into devising reliable and accurate model reduction (intrusive) and reduced-order modeling (non-intrusive) methods, synthesized in a multitude of methodologies continuously developed in the last years [3, 10, 32, 5, 12, 13]. For the class of parametric systems, the comprehensive review contribution in [14] provides an exhaustive account of projection-based methods, from the 2000s until the middle of 2010s.

Additionally, relatively new approaches use time-domain snapshot data to compute reduced-order models, such as operator inference (OpInf) [46] and dynamic mode decomposition (DMD) [50]. Extension of such methods to parameterized dynamical systems have been recently proposed, for OpInf in [57, 40] and also for DMD in [2, 49].

For the class of frequency-domain methods, we mainly concentrate in what follows on interpolation-based methods. For other classes of projection-based methods we refer the reader to the survey [14]. As explained in this review paper, reduced-order models for parametric systems are typically computed by means of projection, using either a local or a global basis for matrix or transfer function interpolation. Here, we mention some of the relevant contributions in the last years [1, 23, 58, 25]. Additionally, optimal and quasi-optimal approaches have also been proposed in [9, 33, 41], which try to impose optimality in certain norms, e.g., in the $\mathcal{H}_2 \otimes \mathcal{L}_2$ norm. Finally, implementation issues for pMOR with software packages in Python programming language were addressed in [42].

Non-intrusive methods based on interpolating or approximately matching (using least squares fitting) of transfer function measurements (of the underlying parameterized rational transfer function) have been quite prolific in the last decades, with the following prominent contributions mentioned below:

- Extensions of the Loewner Framework (LF) to multivariate rational approximation by interpolation: [6, 34, 54] together with the AAA algorithm (Adaptive Antoulas-Anderson) for multivariate functions [48, 24], and an extension of LF with orthogonal polynomials [56].
- Extensions of the Vector Fitting framework to multivariate rational approximation, including the generalized Sanathanan-Koerner iteration in [16, 59]; these works are mostly concerned with imposing stability and passivity guarantees for macro model generation in the field of electronics.

We also mention other related works, such as [19, 8, 45, 18]. Recent trends in harnessing the power of artificial neural networks (ANNs) in connection to deep learning have allowed increased computational power and increased the usability of the methods. Established methods involve incorporating PINNs or DeepONets in the solution or analysis of parametric PDEs, such as in [11, 21, 29].

1.3 Connection to other fields

Tensors represent generalizations of vectors and matrices in multiple dimensions, possessing at least one extra dimension (a minimum of three). Applications include, among others, some from the fields of signal processing (e.g., array processing), scientific computing (e.g., discretization of multivariate functions), and more recently, quantum computing (e.g., simulation of quantum many-body). We refer the reader to [36] for additional information and detailed discussion on the matter. However, explicitly working with tensors, especially of higher dimensions, is not a trivial task. The number of elements in a tensor increases exponentially with the number of dimensions, and so do the computational/memory requirements. The exponential dependency together with the challenges that arise from it, are connected to the curse of dimensionality (**C-o-D**).

Tensor decompositions are particularly important and relevant for several strenuous computational tasks since they can potentially alleviate the curse of dimensionality that occurs when working with high-dimensional tensors, as explained in [53]. 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 [27].

Tensorization and Loewner matrices were recently connected in the contribution [20]. There, a collection of one-dimensional (standard) Loewner matrices is reshaped as a third-dimensional tensor (named "Loewnerization"), for which the block term decomposition (BTD) is applied. The application of interest is blind signal separation. This approach was recently extended in [52], to the MIMO case.

Nonlinear eigenvalue problems (NEPs) represent an established field of study. A NEP can be viewed as a generalization of the (ordinary) eigenvalue problem to equations that depend nonlinearly on the eigenvalue. Linearization techniques allow reformulating any polynomial EP as a larger linear eigenvalue problem and then applying the established (classical) algorithms to solve it. Other linearizations have been proposed involving rational approximation [31], and [37, 30], which involve the usage of the rational Krylov or the AAA algorithms, together with [17] which uses the Loewner and Hankel frameworks in the context of contour integrals.

1.4 Problem statement

A linear-in-state dynamical system parameterized in terms of parameters included in the set $\mathcal{S} = [s_1, \dots, s_n]^\top \subset \mathbb{C}^{n-1}$, is characterized in state-space representation by the following equations:

$$\mathbf{E}(\mathcal{S})\dot{\mathbf{x}}(t; \mathcal{S}) = \mathbf{A}(\mathcal{S})\mathbf{x}(t; \mathcal{S}) + \mathbf{B}(\mathcal{S})\mathbf{u}(t), \quad \mathbf{y}(t; \mathcal{S}) = \mathbf{C}(\mathcal{S})\mathbf{x}(t; \mathcal{S}), \quad (1)$$

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) are as follows: $\mathbf{E}(\mathcal{S}), \mathbf{A}(\mathcal{S}) \in \mathbb{R}^{M \times M}$, $\mathbf{B}(\mathcal{S}) \in \mathbb{R}^{M \times n_u}$, $\mathbf{C}(\mathcal{S}) \in \mathbb{R}^{n_y \times M}$. For simplicity of exposition, we consider only the single-input and single-output (SISO) scenario in

what follows, i.e., $n_u = n_y = 1$. Although extension to multi-input multi-output (MIMO) systems is relevant, it will be the topic of future research, e.g. following the formulation exposed in [54]. In the sequel, particular attention is allocated to the exposition of a solution that **tames** the curse of dimensionality (**C-o-D**).

Remark 1.1 (Focus on the SISO case). *To simplify the exposition, we consider solely the case of single-input and single-output (SISO) systems (i.e. $n_u = n_y = 1$). Although extension to multi-input multi-output (MIMO) systems is relevant, it will be the topic of future research, e.g. following the formulation exposed in [54]. In the sequel, particular attention is allocated to the exposition of a solution that **tames** the curse of dimensionality (**C-o-D**) which is a more challenging and fundamental problem.*

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

$$\mathbf{H}({}^1s, {}^2s, \dots, {}^ns) = \mathbf{C}(\mathcal{S}) [{}^1s\mathbf{E}(\mathcal{S}) - \mathbf{A}(\mathcal{S})]^{-1} \mathbf{B}(\mathcal{S}) \in \mathbb{C}. \quad (2)$$

It involves n variables ${}^js \in \mathbb{C}$, $j = 1, \dots, n$, including the ones in the set \mathcal{S} but also the frequency or Laplace variable, which is denoted by 1s .

We denote the complexity of each variable js with the value d_j (the highest degree in which the variable occurs in both polynomials describing the rational function shown above) and say that $\mathbf{H}({}^1s, {}^2s, \dots, {}^ns)$ in (2) is of complexity (d_1, d_2, \dots, d_n) .

As we are interested in the non-intrusive data-driven setup, let us now consider that the function in (2) is not explicitly known. Instead, one has access to evaluations at (support or interpolatory) points ${}^1\lambda_{j_1}, {}^2\lambda_{j_2}, \dots, {}^n\lambda_{j_n}$ along ${}^1s, {}^2s, \dots, {}^ns$, leading to measurements $\mathbf{w}_{j_1, j_2, \dots, j_n}$, for $j_1, j_2, \dots, j_n = 1, \dots, k_1, k_2, \dots, k_n$. Under some assumptions detailed in the sequel, we seek a reduced multivariate rational model, pROM, $\hat{\mathbf{H}}$ given as

$$\hat{\mathbf{H}}({}^1s, {}^2s, \dots, {}^ns) = \mathbf{G}\Phi({}^1s, {}^2s, \dots, {}^ns)^{-1}\mathbf{W} \in \mathbb{C}, \quad (3)$$

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

In the sequel, we concentrate on continuous-time dynamical systems. Therefore, the first variable 1s will be associated with the dynamic Laplace (frequency) variable, while ${}^2s, \dots, {}^ns$ will stand as non-dynamic parametric variables (typically, they are real-valued but a complex form is also possible). Note that a similar discrete sampled-time model can be obtained using the z -transform (see e.g. [55]).

1.5 Historical notes

The Loewner matrix was introduced by Karl Löwner in the seminal paper published nine decades ago [38]. It has been further studied and used in multiple works dealing with data-driven rational function approximation with application in system theory at large. 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 [39], introducing the two-sided Loewner framework, allowing the construction of a rational model with minimal McMillan degree and embedding a realization with minimal order, directly from the data. Then, [7] provides a comprehensive review of the case of single-variable linear systems, gathering most of the results up to 2017. Later in [6] the one-sided framework is extended to two variables/parameters, and its corresponding Lagrangian realization is derived. Later in [34], the multi-parameter Loewner framework (mpLF) is presented together (for up to three parameters) with the barycentric form, but without the description of a

realization. Recently, handbook and tutorial contributions for LF, with its extensions and many applications, were proposed in [7, 35]. Finally, [26] provides a comprehensive overview including parametric and nonlinear Loewner extensions, and practical applications and test cases from aerospace engineering and computational fluid dynamics.

The AAA algorithm in [43] 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 [15]. In [48], the parametric AAA (p-AAA) algorithm is introduced. This extends the original AAA formulation of [43] 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 multi-dimensional 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 quality is enforced on the available data. Then, extensions of the AAA algorithm to linear systems with quadratic outputs (characterized by transfer functions in one and in two variables) were proposed in [25]. For more extensions and applications of the AAA algorithm, we refer the reader to [44].

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 remarkable to notice that despite all these ramifications, the multivariate versions of Loewner were poorly studied and if so, limited to three variables. This is probably due to the increasing numerical complexity and missing realization form, relevant in system theory. In this paper, we address these two points.

1.6 Contribution and paper organization

Our goal in the sequel, is to provide a complete and scalable solution to the data-driven multivariate reduced order model construction. The results provided in [6] and [34] are extended. The main result consists in taming the complexity issue both in the null space computation and in the realization dimension. More specifically, the contribution is six-fold:

- (i) We propose a multivariate generalized realization allowing to describe with state-space form (with limited complexity), any multivariate rational functions (Section 2 and Theorem 2.1);
- (ii) The n -D multivariate Loewner matrix is introduced, and is shown to be the solution of a cascaded Sylvester equation set (in Section 4);
- (iii) As the dimension N of the n -D Loewner matrix exponentially increases with the number of data, variables, and associated degrees, we demonstrate that the associated null space, required to construct the rational approximant in the Lagrange basis, can be obtained using a collection of 1-D Loewner matrices, leading to a computational complexity reduced from $\mathcal{O}(N^3)$ to about $\mathcal{O}(N^{1.4})$ (Section 5 and, Theorem 5.3 and Theorem 5.4);
- (iv) We detail two data-driven multivariate generalized model construction algorithms (Section 6, and Algorithm 1 and Algorithm 2);
- (v) Stepping back from the dynamic systems perspective, we also note that the proposed approach provides a solution to the tensor approximation problem. Indeed, we approximate any tensorized data set with a rational function. Importantly, this is done by taming the **C-o-D** as pointed out in (iii), thanks to the recursive null space construction in Theorem 5.3.
- (vi) The proposed method has as a by-product the *multi-linearization* of the underlying nonlinear eigenvalue problems (NEP) by means of an interpolatory approach.

Among these contributions, items (i) and (iii) are the main theoretical results towards **taming the curse of dimensionality**, for data-driven multivariate functions and realization construction. More specifically, item (i) provides a new realization structure applicable to any n -dimensional rational function (expressed in the barycentric format) where the complexity (e.g. matrix dimensions) 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 a much lower computational effort. In addition, as pointed in item (v), the proposed iterative null space construction permits the treatment of larger tensor problems with a limited computational complexity.

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 variables case. Then, in [Section 4](#), the data-based n -D **Loewner matrices** are defined, and a connection with cascaded Sylvester equations is made. The relation with the multivariate barycentric/Lagrangian rational form, as well as the multivariate realization is also given. 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. Such a decomposition allows the drastic reduction of the complexity, thus **taming the curse of dimensionality**; it naturally opens the approach to complex cases. From all these contributions, two algorithms are sketched in [Section 6](#), indicating complete procedures for the construction of a non-intrusive multivariate dynamical model realization from input-output data. Numerical examples illustrating the effectiveness of the proposed process are described in [Section 7](#)¹. Finally, [Section 8](#) concludes the paper and provides an outlook to 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 (i.e. state-space equation set) from a n variables transfer functions in the form (2). This is expressed in the Lagrange basis. After some preliminaries, this result is stated in [Theorem 2.1](#). This stands as the first major contribution of this paper.

2.1 Preliminaries

Let us first introduce definitions and intermediate results on which we build the generalized realization. First, we derive the pseudo-companion Lagrange basis, then we provide the multi-row and multi-column indices and coefficient matrices propositions, and finally propositions on the characteristic polynomial.

2.1.1 Pseudo-companion Lagrange matrix

Consider a rational function \mathbf{H} in n variables, namely $^j s$, each of degree d_j ($j = 1, \dots, n$), as in (2). We will consider as basis for polynomials, the **Lagrange basis**. The **Lagrange pseudo-companion matrix** considered here is denoted $^j \mathbb{X}^{\text{Lag}}$ and is defined as follows.

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

$$^j \mathbb{X}^{\text{Lag}} = \left[\frac{\mathbf{X}^{\text{Lag}}(^j s)}{^j \mathbf{q}^{\text{Lag}}} \right] = \begin{bmatrix} ^j \mathbf{x}_1 & -^j \mathbf{x}_2 & 0 & \dots & 0 \\ ^j \mathbf{x}_1 & 0 & -^j \mathbf{x}_3 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ ^j \mathbf{x}_1 & 0 & \dots & 0 & -^j \mathbf{x}_{n_j} \\ \hline ^j q_1 & ^j q_2 & \dots & ^j q_{n_j-1} & ^j q_{n_j} \end{bmatrix} \in \mathbb{C}^{n_j \times n_j} [^j s], \quad (4)$$

with values $^j q_i$, $i = 1, \dots, n_j$ chosen so that $^j \mathbb{X}^{\text{Lag}}$ is unimodular, i.e. $\det(^j \mathbb{X}^{\text{Lag}}) = 1$.

Following the general interpolation framework, the $^j s$ ($j = 1, \dots, n$) variables of \mathbf{H} (2) are split into **left** and **right** variables, or equivalently into **row** and **column** variables. For simplicity

¹An exhaustive set of numerical examples, together with data and numerical results is provided at https://sites.google.com/site/charlespoussotvassal/nd_loew_tcod.

of exposition (and by permutation, if necessary), we assume that $^1s, \dots, ^ks$ are the column (right) variables and $^{k+1}s, \dots, ^ns$ are the row (left) variables ($0 < k < n$, $k \in \mathbb{N}$). Based on this data, we define two **Kronecker products** of the associated pseudo-companion matrices:

Definition 2.2. Consider the column $^1s, \dots, ^ks$ and row $^{k+1}s, \dots, ^ns$ variables, we define the Kronecker products of the pseudo-companion matrices (4) as

$$\begin{aligned} \mathbf{\Gamma}^{\text{Lag}} &= {}^1\mathbb{X}^{\text{Lag}} \otimes {}^2\mathbb{X}^{\text{Lag}} \otimes \dots \otimes {}^k\mathbb{X}^{\text{Lag}} \in \mathbb{C}^{\kappa \times \kappa}[{}^1s, \dots, ^ks], \\ \mathbf{\Delta}^{\text{Lag}} &= {}^{k+1}\mathbb{X}^{\text{Lag}} \otimes {}^{k+2}\mathbb{X}^{\text{Lag}} \otimes \dots \otimes {}^n\mathbb{X}^{\text{Lag}} \in \mathbb{C}^{\ell \times \ell}[{}^{k+1}s, \dots, ^ns], \end{aligned} \quad (5)$$

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 Multi-row/multi-column 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:

$$I_q = [i_{k+1}^q, i_{k+2}^q, \dots, i_n^q], \quad J_r = [j_1^r, j_2^r, \dots, j_k^r], \quad q = 1, \dots, \ell, \quad r = 1, \dots, \kappa.$$

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

Remark 2.1. 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} in the Lagrange basis:

$$\mathbf{H}({}^1s, {}^2s, \dots, {}^ns) = \frac{\sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \dots \sum_{j_n=1}^{k_n} \frac{c_{j_1, j_2, \dots, j_n} \mathbf{w}_{j_1, j_2, \dots, j_n}}{(1s^{-1}\lambda_{j_1})(2s^{-2}\lambda_{j_2}) \dots (ns^{-n}\lambda_{j_n})}}{\sum_{j_1=1}^{k_1} \sum_{j_2=1}^{k_2} \dots \sum_{j_n=1}^{k_n} \frac{c_{j_1, j_2, \dots, j_n}}{(1s^{-1}\lambda_{j_1})(2s^{-2}\lambda_{j_2}) \dots (ns^{-n}\lambda_{j_n})}}, \quad (6)$$

where $c_{j_1, j_2, \dots, j_n} \in \mathbb{C}$ are the barycentric weights and $\mathbf{w}_{j_1, j_2, \dots, j_n} \in \mathbb{C}$ the data evaluated at the barycentric combination $\{^1\lambda_{j_1}, ^2\lambda_{j_2}, \dots, ^n\lambda_{j_n}\} \in \mathbb{C}^n$, or equivalently, following Definition 2.3,

$$\mathbf{H}({}^1s, {}^2s, \dots, {}^ns) = \frac{\sum_{q=1}^{\ell} \sum_{r=1}^{\kappa} \frac{\beta_{I_q, J_r}}{\prod_{a \in I_q} \prod_{b \in J_r} (a s^{-a} \lambda_{j_a}) (b s^{-b} \lambda_{j_b})}}{\sum_{q=1}^{\ell} \sum_{r=1}^{\kappa} \frac{\alpha_{I_q, J_r}}{\prod_{a \in I_q} \prod_{b \in J_r} (a s^{-a} \lambda_{j_a}) (b s^{-b} \lambda_{j_b})}} \quad (7)$$

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

$$\mathbb{A}^{\text{Lag}} = \begin{bmatrix} \alpha_{I_1, J_1} & \alpha_{I_1, J_2} & \dots & \alpha_{I_1, J_\kappa} \\ \alpha_{I_2, J_1} & \alpha_{I_2, J_2} & \dots & \alpha_{I_2, J_\kappa} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{I_\ell, J_1} & \alpha_{I_\ell, J_2} & \dots & \alpha_{I_\ell, J_\kappa} \end{bmatrix}, \quad \mathbb{B}^{\text{Lag}} = \begin{bmatrix} \beta_{I_1, J_1} & \beta_{I_1, J_2} & \dots & \beta_{I_1, J_\kappa} \\ \beta_{I_2, J_1} & \beta_{I_2, J_2} & \dots & \beta_{I_2, J_\kappa} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{I_\ell, J_1} & \beta_{I_\ell, J_2} & \dots & \beta_{I_\ell, J_\kappa} \end{bmatrix}. \quad (8)$$

Notice that \mathbb{A}^{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}^{Lag} , contains the barycentric weights of the numerator, i.e. the product of the denominator barycentric weights with the corresponding value 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 2.1. *Given is the polynomial $\mathbf{p}(s)$ of degree less than or equal to n , expressed in a Lagrange basis as $\mathbf{p}(s) = \pi \left(\frac{\alpha_1}{\mathbf{x}_1} + \cdots + \frac{\alpha_{n+1}}{\mathbf{x}_{n+1}} \right)$ 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 pseudo-companion form matrix as in Definition 2.1, where ${}^j \mathbf{x}_i$ is replaced by \mathbf{x}_j and ${}^j q_1$ by α_j .*

Next, following Proposition 2.1, we consider two-variable polynomials $\mathbf{p}(s, t)$ of degree (at most) n, m in the variables s, t , respectively. Let $\mathbf{x}_i(s) = s - s_i, s_i \in \mathbb{C}, i = 1, \dots, n+1$, and $\mathbf{y}_j(t) = t - t_j, t_j \in \mathbb{C}, j = 1, \dots, m+1$, 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 re-written by highlighting the matrix form of (8), as,

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

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

$$\mathbf{S} = \underbrace{\begin{bmatrix} \mathbf{x}_1 & -\mathbf{x}_2 & 0 & \cdots & 0 \\ \mathbf{x}_1 & 0 & -\mathbf{x}_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{x}_1 & 0 & \cdots & 0 & -\mathbf{x}_{n+1} \\ \epsilon_1 & \epsilon_2 & \cdots & \epsilon_n & \epsilon_{n+1} \end{bmatrix}}_{\in \mathbb{C}^{(n+1) \times (n+1)}[s]}, \quad \mathbf{T} = \underbrace{\begin{bmatrix} \mathbf{y}_1 & -\mathbf{y}_2 & 0 & \cdots & 0 \\ \mathbf{y}_1 & 0 & -\mathbf{y}_3 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{y}_1 & 0 & \cdots & 0 & -\mathbf{y}_{m+1} \\ \zeta_1 & \zeta_2 & \cdots & \zeta_m & \zeta_{m+1} \end{bmatrix}}_{\in \mathbb{C}^{(m+1) \times (m+1)}[t]}, \quad (9)$$

where the constants ϵ_i and ζ_j are chosen so that $\det(\mathbf{S}) = 1$ and $\det(\mathbf{T}) = 1^2$. The coefficients $\alpha_{i,j}$ are arranged in the form of a matrix $\mathbb{A}^{\text{Lag}} \in \mathbb{C}^{(n+1) \times (m+1)}$, as in (8), or in a vectorized version (taken row-wise) $\text{vec}(\mathbb{A}^{\text{Lag}}) \in \mathbb{C}^{1 \times \kappa}$:

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

$$\text{vec}(\mathbb{A}^{\text{Lag}}) = [\alpha_{1,1}, \alpha_{1,2}, \dots, \alpha_{1,m+1} \mid \cdots \mid \alpha_{n+1,1}, \dots, \alpha_{n+1,m+1}],$$

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

$$\mathbf{M}_1 = \underbrace{\begin{bmatrix} (\mathbf{S} \otimes \mathbf{T})(1:\kappa-1, :) \\ \text{vec}(\mathbb{A}^{\text{Lag}}) \end{bmatrix}}_{\in \mathbb{C}^{\kappa \times \kappa}[s,t]} \quad \text{and} \quad \mathbf{M}_2 = \underbrace{\begin{bmatrix} \mathbf{S}(1:n-1, :) & \mathbf{0}_{n-1, m-1} \\ \mathbb{A}^{\text{Lag}} & \mathbf{T}(1:m-1, :)^{\top} \end{bmatrix}}_{\in \mathbb{C}^{(n+m-1) \times (n+m-1)}[s,t]}. \quad (10)$$

Proposition 2.2. *The determinants of \mathbf{M}_1 and \mathbf{M}_2 are both equal to $\mathbf{p}(s, t)$.*

²One may chose $1/\epsilon_i = \prod_{j \neq i} (s_i - s_j)$ and $1/\zeta_i = \prod_{j \neq i} (t_i - t_j)$, for $i, j = 1, \dots, n, m$.

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 2.1](#). \square

Remark 2.2 (Curse of dimensionality). *This result shows that by splitting the variables into left and right, the $\mathbf{C-o-D}$ 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

Now all the preliminary ingredients are in place, the first main result is stated in [Theorem 2.1](#). Its proof is given subsequently.

2.2.1 Main result

The main result given in [Theorem 2.1](#) provides a systematic way to construct a realization as in (3), from a transfer function \mathbf{H} given in a barycentric / Lagrange form (6).

Theorem 2.1. *Given quantities in [Definition 2.1](#) and [Definition 2.2](#), a $2\ell + \kappa - 1 = m$ -th order realization $(\mathbf{G}, \Phi, \mathbf{W})$ of the multivariate function \mathbf{H} (2), in barycentric form (6), satisfying $\mathbf{H}(^1s, \dots, ^ns) = \mathbf{W}\Phi(^1s, ^2s, \dots, ^ns)^{-1}\mathbf{G}$, is given by,*

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

$$\mathbf{G} = \left[\begin{array}{c} \mathbf{0}_{\kappa-1, 1} \\ \mathbf{\Delta}(\ell, :)^{\top} \\ \mathbf{0}_{\ell, 1} \end{array} \right] \in \mathbb{C}^{m \times 1} \text{ and } \mathbf{W} = \left[\mathbf{0}_{1, \kappa} \mid \mathbf{0}_{1, \ell-1} \mid -\mathbf{e}_{\ell}^{\top} \right] \in \mathbb{C}^{1 \times m}, \quad (11)$$

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 $\mathbb{A}^{Lag}, \mathbb{B}^{Lag} \in \mathbb{C}^{\ell \times \kappa}$ are appropriately chosen, according to the chosen pseudo-companion basis.

Proof. See [Section 2.2.2](#). \square

Remark 2.3 (Matrix realization). *From [Theorem 2.1](#) and following (1)'s notations, it follows that $\Phi(^1s, ^2s, \dots, ^ns) = ^1s\mathbf{E}(S) - \mathbf{A}(S)$, $\mathbf{B}(S) = \mathbf{W}$ and $\mathbf{C}(S) = \mathbf{G}$.*

Corollary 2.1. *The realization defined by the tuple $(\mathbf{W}, \Phi, \mathbf{G})$ has dimension $m = 2\ell + \kappa - 1$, and is both R -controllable and R -observable, i.e.*

$$\left[\Phi(^1s, \dots, ^ns) \quad \mathbf{G} \right] \text{ and } \left[\begin{array}{c} \mathbf{W} \\ \Phi(^1s, \dots, ^ns) \end{array} \right] \quad (12)$$

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

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

Remark 2.4 (Connection with NEPs). *The realizations proposed address the problem of linearization in the context of NEPs. Specifically, since we are in the multivariate case, our realizations achieve multi-linearizations 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 .*

2.2.2 Proof of Theorem 2.1

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

$$\mathbf{H} = \mathbf{W} \Phi^{-1} \mathbf{G} = \frac{\mathbf{n}}{\mathbf{d}} = \mathbf{W}_2 \Phi_{22}^{-1} \Phi_{21} \Phi_{11}^{-1} \mathbf{G}_1. \quad (13)$$

The last expression can be expressed explicitly as:

$$\underbrace{[\mathbf{0}_{1,\ell-1} \mid \mathbf{e}_\ell^\top]}_{\mathbf{W}_2} \underbrace{\Delta^{-\top}}_{\Phi_{22}^{-1}} \underbrace{[\mathbb{B}^{\text{Lag}} \mid \mathbf{0}_{\ell,\ell-1}]}_{\Phi_{21}} \underbrace{\left[\begin{array}{c|c} \Gamma(1 : \kappa - 1, 1 : \kappa) & \mathbf{0}_{\kappa-1,\ell-1} \\ \hline \mathbb{A}^{\text{Lag}} & \Delta(1 : \ell - 1, :)^\top \end{array} \right]^{-1}}_{\Phi_{11}^{-1}} \underbrace{\left[\begin{array}{c} \mathbf{0}_{\kappa-1,1} \\ \hline \Delta(\ell, :)^\top \end{array} \right]}_{\mathbf{G}_1} \cdot \left[\begin{array}{c} \mathbf{c}_\kappa \\ ? \end{array} \right] \text{ where } ? \text{ has size } \ell-1$$

The expressions for \mathbf{r}_ℓ^\top and \mathbf{c}_κ are a consequence of Proposition 2.3 below. It follows that $\mathbf{n} = \mathbf{r}_\ell^\top \mathbb{B}^{\text{Lag}} \mathbf{c}_\kappa$. Interchanging \mathbb{A}^{Lag} and \mathbb{B}^{Lag} in (11), amounts to interchanging \mathbf{n} and \mathbf{d} , in \mathbf{H} (13). Consequently, the expression for the denominator is: $\mathbf{d} = \mathbf{r}_\ell^\top \mathbb{A}^{\text{Lag}} \mathbf{c}_\kappa$.

Proposition 2.3. (a) The last row of $\Delta^{-\top}$ is:

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

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

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

Remark 2.5 (Splitting possibilities). The possibility of splitting the variables to **left** variables and **right** variables allows us to pick the splitting that **minimizes** m . For instance, if we have four variables with degrees $(2, 2, 1, 1)$, splitting the variables into $(2, 1)$ – $(2, 1)$ gives $m = 17$, while the splitting (2) – $(2, 1, 1)$ (i.e. one column and three rows variables) gives $m = 26$.

2.3 Comments

In Theorem 2.1, both the $\mathbb{A}^{\text{Lag}}, \mathbb{B}^{\text{Lag}} \in \mathbb{C}^{\ell \times \kappa}$ matrices values and dimensions are directly related to the pseudo-companion basis chosen in Definition 2.1 and on the columns-rows variables split. Without entering into technical considerations (out of the scope of this paper), one may notice the following:

- (i) different pseudo-companion forms (4) can be considered, leading to different structures associated with different polynomial bases such as Lagrange, and monomial basis. Here the Lagrange basis will be considered only³;
- (ii) different permutations and rearrangements of j 's in Definition 2.2 may be considered. This results in different realization order $m = 2\ell + \kappa - 1$. 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. These data can be obtained from any dynamical black box model, simulator, or experiment.

³Other basis may be investigated in future research but this is out of the scope of the paper.

3 Data definitions and description

Following the Loewner philosophy presented in a series of papers [39, 6, 34, 26], let us define P_c , the column (or right) 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, to simplify the exposition of the general n -D case.

3.1 The 1-D case

When single variable functions $\mathbf{H}(^1s)$ are considered, i.e. $n = 1$ in (2), let us define the following column and row data:

$$P_c^{(1)} := \{(^1\lambda_{j_1}; \mathbf{w}_{j_1}), j_1 = 1, \dots, k_1\}, \quad P_r^{(1)} := \{(^1\mu_{i_1}; \mathbf{v}_{i_1}), i_1 = 1, \dots, q_1\}, \quad (14)$$

where $^1\lambda_{j_1}, ^1\mu_{i_1} \in \mathbb{C}$ are disjoint interpolation points (or support points), which evaluation through \mathbf{H} respectively lead to $\mathbf{w}_{j_1}, \mathbf{v}_{i_1} \in \mathbb{C}$. To support our exposition, let the data (14) be represented in the tableau given in Table 1a, where the measurement vectors $\mathbf{W}_{k_1} \in \mathbb{C}^{k_1}$ and $\mathbf{V}_{q_1} \in \mathbb{C}^{q_1}$ indicate the evaluation of \mathbf{H} through the single variable 1s , evaluated at $^1\lambda_{j_1}$ and $^1\mu_{i_1}$ respectively. Table 1a (also called \mathbf{tab}_1) is called a measurement matrix. From \mathbf{tab}_1 , (1,1) block of dimension $k_1 \times 1$ is the column measurements, and (1,2) block of dimension $q_1 \times 1$, is the row measurements.

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

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

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

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

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

Remark 3.1 (Row/column data). *The "column" and "row" denomination comes from the fact that these data refer to the columns or rows of the Loewner matrix, introduced later on this paper. The "right" and "left" terms are also used in the so-called tangential Loewner framework to deal with MIMO data (see [7] or [51]).*

3.2 The 2-D case

Let us define the column and row data:

$$\begin{cases} P_c^{(2)} & := \{(^1\lambda_{j_1}, ^2\lambda_{j_2}; \mathbf{w}_{j_1, j_2}), j_1, j_2 = 1, \dots, k_1, k_2\} \\ P_r^{(2)} & := \{(^1\mu_{i_1}, ^2\mu_{i_2}; \mathbf{v}_{i_1, i_2}), i_1, i_2 = 1, \dots, q_1, q_2\} \end{cases}, \quad (15)$$

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 disjoint interpolation points, for which evaluating $\mathbf{H}(^1s, ^2s)$ respectively at, lead to $\mathbf{w}_{j_1, j_2}, \mathbf{v}_{i_1, i_2} \in \mathbb{C}$. Similarly to the 1-D case, data (15) may be represented in the Table 1b, where $\mathbf{W}_{k_1, k_2} \in \mathbb{C}^{k_1 \times k_2}$ and $\mathbf{V}_{q_1, q_2} \in \mathbb{C}^{q_1 \times q_2}$ are the measurement matrices related to evaluation of \mathbf{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 measurements: $\phi_{rc} \in \mathbb{C}^{q_1 \times k_2}$ and $\phi_{cr} \in \mathbb{C}^{k_1 \times q_2}$. The former resulting from the cross interpolation points evaluation of $\mathbf{H}(^1s, ^2s)$ along $\{^1\mu_{i_1}, ^2\lambda_{j_2}\}$ and the latter along $\{^1\lambda_{j_1}, ^2\mu_{i_2}\}$. It follows that Table 1b (denoted \mathbf{tab}_2), is a measurement matrix.

Remark 3.2 (Cross measurements). *In [34, 54], these cross-measurements are used in the extended Loewner matrix construction for improved accuracy. For clarity of exposition we will not make use these data, i.e. we will consider the first and last blocks of the tableau \mathbf{tab}_n .*

3.3 The n -D case

Now the single variable and two variables cases have been addressed, as well as various relevant notations were introduced, let us present the n variables data case:

$$\begin{cases} P_c^{(n)} := \{({}^1\lambda_{j_1}, {}^2\lambda_{j_2}, \dots, {}^n\lambda_{j_n}; \mathbf{w}_{j_1, j_2, \dots, j_n}), j_1, j_2, \dots, j_n = 1, \dots, k_1, k_2, \dots, k_n\} \\ P_r^{(n)} := \{({}^1\mu_{i_1}, {}^2\mu_{i_2}, \dots, {}^n\mu_{i_n}; \mathbf{v}_{i_1, i_2, \dots, i_n}), i_1, i_2, \dots, i_n = 1, \dots, q_1, q_2, \dots, q_n\} \end{cases} \quad (16)$$

Similarly, one may now derive the n variable measurement matrix called \mathbf{tab}_n , illustrated in the table sequence given in Table 2. Similarly to the expositions made for the single and two variable cases, each sub-table considers frozen configurations of ${}^3s, {}^4s, \dots, {}^ns$, along with the combinations of the support points ${}^3\lambda_{j_3}, {}^4\lambda_{j_4}, \dots, {}^n\lambda_{j_n}$ and ${}^3\mu_{i_3}, {}^4\mu_{i_4}, \dots, {}^n\mu_{i_n}$. More specifically, considering the first sub-tableau, the evaluation is for ${}^3s, {}^4s, \dots, {}^ns = {}^3\lambda_{j_3}, {}^4\lambda_{j_4}, \dots, {}^n\lambda_{j_n}$ while the second is for ${}^3s, {}^4s, \dots, {}^ns = {}^3\mu_{i_3}, {}^4\mu_{i_4}, \dots, {}^n\mu_{i_n}$, and so on. Finally, note that the $\mathbf{w}_{j_1, j_2, \dots, j_n}$ and $\mathbf{v}_{i_1, i_2, \dots, i_n}$ values involved in the data (16) are concentrated in the first and last sub-tableau, in the multi-dimensional (tensor) $\mathbf{W}_{k_1, \dots, k_n} \in \mathbb{C}^{k_1 \times k_2 \times \dots \times k_n}$ and $\mathbf{V}_{q_1, \dots, q_n} \in \mathbb{C}^{q_1 \times q_2 \times \dots \times q_n}$. Then, \mathbf{tab}_n may be viewed as a n -dimensional tensor.

<p>(a) ${}^3s = {}^3\lambda_{j_3}, {}^4s = {}^4\lambda_{j_4}, \dots, {}^ns = {}^n\lambda_{j_n}$.</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 33%;">2s</td> <td style="width: 33%;">${}^2\lambda_{1, \dots, k_2}$</td> <td style="width: 33%;">${}^2\mu_{1, \dots, q_2}$</td> </tr> <tr> <td>1s</td> <td>$\mathbf{W}_{k_1, k_2, \dots, k_n}$</td> <td>$\phi_{crc \dots c}$</td> </tr> <tr> <td>${}^1\lambda_{1, \dots, k_1}$</td> <td>$\phi_{rcrc \dots c}$</td> <td>$\phi_{rrrc \dots c}$</td> </tr> <tr> <td>${}^1\mu_{1, \dots, q_1}$</td> <td>$\phi_{rcrc \dots c}$</td> <td>$\phi_{rrrc \dots c}$</td> </tr> </table>	2s	${}^2\lambda_{1, \dots, k_2}$	${}^2\mu_{1, \dots, q_2}$	1s	$\mathbf{W}_{k_1, k_2, \dots, k_n}$	$\phi_{crc \dots c}$	${}^1\lambda_{1, \dots, k_1}$	$\phi_{rcrc \dots c}$	$\phi_{rrrc \dots c}$	${}^1\mu_{1, \dots, q_1}$	$\phi_{rcrc \dots c}$	$\phi_{rrrc \dots c}$	<p>(b) ${}^3s = {}^3\mu_{i_3}, {}^4s = {}^4\lambda_{j_4}, \dots, {}^ns = {}^n\lambda_{j_n}$.</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 33%;">2s</td> <td style="width: 33%;">${}^2\lambda_{1, \dots, k_2}$</td> <td style="width: 33%;">${}^2\mu_{1, \dots, q_2}$</td> </tr> <tr> <td>1s</td> <td>$\phi_{crc \dots c}$</td> <td>$\phi_{rrrc \dots c}$</td> </tr> <tr> <td>${}^1\lambda_{1, \dots, k_1}$</td> <td>$\phi_{rcrc \dots c}$</td> <td>$\phi_{rrrc \dots c}$</td> </tr> <tr> <td>${}^1\mu_{1, \dots, q_1}$</td> <td>$\phi_{rcrc \dots c}$</td> <td>$\phi_{rrrc \dots c}$</td> </tr> </table>	2s	${}^2\lambda_{1, \dots, k_2}$	${}^2\mu_{1, \dots, q_2}$	1s	$\phi_{crc \dots c}$	$\phi_{rrrc \dots c}$	${}^1\lambda_{1, \dots, k_1}$	$\phi_{rcrc \dots c}$	$\phi_{rrrc \dots c}$	${}^1\mu_{1, \dots, q_1}$	$\phi_{rcrc \dots c}$	$\phi_{rrrc \dots c}$
2s	${}^2\lambda_{1, \dots, k_2}$	${}^2\mu_{1, \dots, q_2}$																							
1s	$\mathbf{W}_{k_1, k_2, \dots, k_n}$	$\phi_{crc \dots c}$																							
${}^1\lambda_{1, \dots, k_1}$	$\phi_{rcrc \dots c}$	$\phi_{rrrc \dots c}$																							
${}^1\mu_{1, \dots, q_1}$	$\phi_{rcrc \dots c}$	$\phi_{rrrc \dots c}$																							
2s	${}^2\lambda_{1, \dots, k_2}$	${}^2\mu_{1, \dots, q_2}$																							
1s	$\phi_{crc \dots c}$	$\phi_{rrrc \dots c}$																							
${}^1\lambda_{1, \dots, k_1}$	$\phi_{rcrc \dots c}$	$\phi_{rrrc \dots c}$																							
${}^1\mu_{1, \dots, q_1}$	$\phi_{rcrc \dots c}$	$\phi_{rrrc \dots c}$																							
<p>(c) ${}^3s = {}^3\lambda_{j_3}, {}^4s = {}^4\mu_{i_4}, \dots, {}^ns = {}^n\lambda_{j_n}$.</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 33%;">2s</td> <td style="width: 33%;">${}^2\lambda_{1, \dots, k_2}$</td> <td style="width: 33%;">${}^2\mu_{1, \dots, q_2}$</td> </tr> <tr> <td>1s</td> <td>$\phi_{ccrc \dots c}$</td> <td>$\phi_{crrc \dots c}$</td> </tr> <tr> <td>${}^1\lambda_{1, \dots, k_1}$</td> <td>$\phi_{ccrc \dots c}$</td> <td>$\phi_{crrc \dots c}$</td> </tr> <tr> <td>${}^1\mu_{1, \dots, q_1}$</td> <td>$\phi_{rcrc \dots c}$</td> <td>$\phi_{rrrc \dots c}$</td> </tr> </table>	2s	${}^2\lambda_{1, \dots, k_2}$	${}^2\mu_{1, \dots, q_2}$	1s	$\phi_{ccrc \dots c}$	$\phi_{crrc \dots c}$	${}^1\lambda_{1, \dots, k_1}$	$\phi_{ccrc \dots c}$	$\phi_{crrc \dots c}$	${}^1\mu_{1, \dots, q_1}$	$\phi_{rcrc \dots c}$	$\phi_{rrrc \dots c}$	<p>(d) ${}^3s = {}^3\mu_{i_3}, {}^4s = {}^4\mu_{i_4}, \dots, {}^ns = {}^n\mu_{i_n}$.</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 33%;">2s</td> <td style="width: 33%;">${}^2\lambda_{1, \dots, k_2}$</td> <td style="width: 33%;">${}^2\mu_{1, \dots, q_2}$</td> </tr> <tr> <td>1s</td> <td>$\phi_{ccr \dots r}$</td> <td>$\mathbf{V}_{q_1, q_2, \dots, q_n}$</td> </tr> <tr> <td>${}^1\lambda_{1, \dots, k_1}$</td> <td>$\phi_{ccr \dots r}$</td> <td>$\mathbf{V}_{q_1, q_2, \dots, q_n}$</td> </tr> <tr> <td>${}^1\mu_{1, \dots, q_1}$</td> <td>$\phi_{rcr \dots r}$</td> <td>$\mathbf{V}_{q_1, q_2, \dots, q_n}$</td> </tr> </table>	2s	${}^2\lambda_{1, \dots, k_2}$	${}^2\mu_{1, \dots, q_2}$	1s	$\phi_{ccr \dots r}$	$\mathbf{V}_{q_1, q_2, \dots, q_n}$	${}^1\lambda_{1, \dots, k_1}$	$\phi_{ccr \dots r}$	$\mathbf{V}_{q_1, q_2, \dots, q_n}$	${}^1\mu_{1, \dots, q_1}$	$\phi_{rcr \dots r}$	$\mathbf{V}_{q_1, q_2, \dots, q_n}$
2s	${}^2\lambda_{1, \dots, k_2}$	${}^2\mu_{1, \dots, q_2}$																							
1s	$\phi_{ccrc \dots c}$	$\phi_{crrc \dots c}$																							
${}^1\lambda_{1, \dots, k_1}$	$\phi_{ccrc \dots c}$	$\phi_{crrc \dots c}$																							
${}^1\mu_{1, \dots, q_1}$	$\phi_{rcrc \dots c}$	$\phi_{rrrc \dots c}$																							
2s	${}^2\lambda_{1, \dots, k_2}$	${}^2\mu_{1, \dots, q_2}$																							
1s	$\phi_{ccr \dots r}$	$\mathbf{V}_{q_1, q_2, \dots, q_n}$																							
${}^1\lambda_{1, \dots, k_1}$	$\phi_{ccr \dots r}$	$\mathbf{V}_{q_1, q_2, \dots, q_n}$																							
${}^1\mu_{1, \dots, q_1}$	$\phi_{rcr \dots r}$	$\mathbf{V}_{q_1, q_2, \dots, q_n}$																							

Table 2: n -D table construction: \mathbf{tab}_n (three first and last sub-tableau).

4 Multivariate Loewner matrices and null spaces

Based on Section 3 (see (16) and \mathbf{tab}_n), we are now ready to present our *main tool*: the **multivariate Loewner matrix**. Following the expository account in the previous section, we first recall the 1-D and 2-D Loewner matrices, before presenting its n -D version. For each case, the Loewner matrix together with its link with Sylvester equations is stressed. Then, the connections between the Loewner null space and the barycentric transfer function are recalled, 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 connection with dynamical systems theory may be found in [7]).

4.1.1 Loewner matrix and the Sylvester equation

Definition 4.1. Given the data described in (14), 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

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

Theorem 4.1. *Considering the data in (14), we define the following matrices:*

$$\mathbf{\Lambda}_1 = \text{diag}({}^1\lambda_1, \dots, {}^1\lambda_{k_1}), \mathbf{M}_1 = \text{diag}({}^1\mu_1, \dots, {}^1\mu_{q_1})$$

$$\mathbb{W}_1 = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{k_1}], \mathbb{V}_1 = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{q_1}]^\top \text{ and } \mathbf{L}_1 = \mathbf{1}_{q_1}, \mathbf{R}_1 = \mathbf{1}_{k_1}^\top.$$

The Loewner matrix, as 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, Lagrangian form, and generalized realization

Computing $\mathbb{L}_1 \mathbf{c}_1 = 0$, the null space of the single-variable Loewner matrix, the following holds: $\mathbf{c}_1 = [c_1 \ c_2 \ \dots \ c_{k_1}]^\top \in \mathbb{C}^{k_1}$ contains the so-called barycentric weights of the single-variable rational function $\mathbf{g}({}^1s)$ given by

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

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

Result 4.1 (1-D realization). *Given [Definition 2.2](#) and following [Theorem 2.1](#), a generalized realization of $\mathbf{g}({}^1s)$ is obtained with the following settings: $\mathbb{A}^{Lag} = \mathbf{c}_1^\top$, $\mathbb{B}^{Lag} = \emptyset$, $\mathbf{\Gamma} = \mathbb{1}^{Lag}$ and $\mathbf{\Delta} = \emptyset$.*

Note that this representation recovers the result already discussed e.g. in [\[7\]](#). A simple example follows below to explicitly illustrate for the reader.

Example 4.1. *Let us consider $\mathbf{H}({}^1s) = \mathbf{H}(s) = (s^2 + 4)/(s + 1)$, being a single-valued rational function of complexity 2 (i.e. dimension 2 along s). By evaluating \mathbf{H} in ${}^1\lambda_{j_1} = [1, 3, 5]$ and ${}^1\mu_{i_1} = [2, 4, 6, 8]$, one gets $\mathbf{w}_{j_1} = [5/2, 13/4, 29/6]$ and $\mathbf{v}_{i_1} = [8/3, 4, 40/7, 68/9]$. Then, we construct the Loewner matrix, its null space ($\text{rank } \mathbb{L}_1 = 2$) and rational function interpolating the data as,*

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

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

$$\mathbf{\Phi}(s) = \begin{bmatrix} s-1 & 3-s & 0 \\ s-1 & 0 & 5-s \\ -\frac{1}{3} & \frac{4}{3} & -1 \end{bmatrix} \text{ and } \begin{cases} \mathbf{W} = [0 \ 0 \ | \ -1], \\ \mathbf{G}^\top = [\frac{5}{6} \ -\frac{13}{3} \ \frac{29}{6}]. \end{cases}$$

4.2 The 2-D case

We now continue the exposition with the 2-D case. This section recovers the results originally given in [\[6\]](#). The reader is invited to refer to this paper for further details and involved derivations.

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 4.2. *Given the data described in (15), the 2-D Loewner matrix $\mathbb{L}_2 \in \mathbb{C}^{q_1 q_2 \times k_1 k_2}$, 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 4.3. Considering the data given in (15), we define the following matrices based on Kronecker products:

$$\begin{aligned}\mathbf{\Lambda}_1 &= \text{diag}({}^1\lambda_1, \dots, {}^1\lambda_{k_1}) \otimes \mathbf{I}_{k_2}, \quad \mathbf{M}_1 = \text{diag}({}^1\mu_1, \dots, {}^1\mu_{q_1}) \otimes \mathbf{I}_{q_2}, \\ \mathbf{\Lambda}_2 &= \mathbf{I}_{k_1} \otimes \text{diag}({}^2\lambda_1, \dots, {}^2\lambda_{k_2}), \quad \mathbf{M}_2 = \mathbf{I}_{q_1} \otimes \text{diag}({}^2\mu_1, \dots, {}^2\mu_{q_2}), \\ \mathbb{W}_2 &= [\mathbf{w}_{1,1}, \mathbf{w}_{1,2}, \dots, \mathbf{w}_{1,k_2}, \mathbf{w}_{2,1}, \dots, \mathbf{w}_{k_1,k_2}], \quad \mathbf{R}_2 = \mathbf{1}_{k_1 k_2}^\top, \\ \mathbb{V}_2 &= [\mathbf{v}_{1,1}, \mathbf{v}_{1,2}, \dots, \mathbf{v}_{1,q_2}, \mathbf{v}_{2,1}, \dots, \mathbf{v}_{q_1,q_2}]^\top \text{ and } \mathbf{L}_2 = \mathbf{1}_{q_1 q_2}.\end{aligned}\tag{17}$$

Theorem 4.2. The 2-D Loewner matrix as defined in Definition 4.2 is solution of the following coupled Sylvester equations:

$$\mathbf{M}_2 \mathbb{X} - \mathbb{X} \mathbf{\Lambda}_2 = \mathbb{V}_2 \mathbf{R}_2 - \mathbf{L}_2 \mathbb{W}_2 \text{ and } \mathbf{M}_1 \mathbb{L}_2 - \mathbb{L}_2 \mathbf{\Lambda}_1 = \mathbb{X}.\tag{18}$$

Proof. We start with the first equation in (18) and note it is exactly the same as the Sylvester equation from the 1-D case. Hence, its solution can be written explicitly as $(\mathbb{X})_{\mathbf{i}, \mathbf{j}} = \frac{\mathbf{e}_{\mathbf{i}}^\top \mathbb{V}_2 - \mathbb{W}_2 \mathbf{e}_{\mathbf{j}}}{\mathbf{e}_{\mathbf{i}}^\top \mathbf{M}_2 \mathbf{e}_{\mathbf{i}} - \mathbf{e}_{\mathbf{j}}^\top \mathbf{\Lambda}_2 \mathbf{e}_{\mathbf{j}}}$, where \mathbf{e}_k is the k -th unit vector of appropriate length (all its elements are equal to 0 but the k -th one). Additionally, we have that: $\mathbf{i} = q_2(i_1 - 1) + i_2$, $1 \leq i_1 \leq q_1$, $1 \leq i_2 \leq q_2$ and $\mathbf{j} = k_2(j_1 - 1) + j_2$, $1 \leq j_1 \leq k_1$, $1 \leq j_2 \leq k_2$. By a slight abuse of notation, we can embed \mathbf{i} with pair (i_1, i_2) and \mathbf{j} with pair (j_1, j_2) . Next, we analyze the second equation in (18). Because of its simplified form, we can explicitly write the (\mathbf{i}, \mathbf{j}) entry of \mathbb{L}_2 as $(\mathbb{L}_2)_{\mathbf{i}, \mathbf{j}} = \frac{(\mathbb{X})_{\mathbf{i}, \mathbf{j}}}{\mathbf{e}_{\mathbf{i}}^\top \mathbf{M}_1 \mathbf{e}_{\mathbf{i}} - \mathbf{e}_{\mathbf{j}}^\top \mathbf{\Lambda}_1 \mathbf{e}_{\mathbf{j}}}$. By substituting in this relation the explicit form of the (\mathbf{i}, \mathbf{j}) element of \mathbb{X} , we get that

$$(\mathbb{L}_2)_{\mathbf{i}, \mathbf{j}} = \frac{\mathbf{e}_{\mathbf{i}}^\top \mathbb{V}_2 - \mathbb{W}_2 \mathbf{e}_{\mathbf{j}}}{(\mathbf{e}_{\mathbf{i}}^\top \mathbf{M}_1 \mathbf{e}_{\mathbf{i}} - \mathbf{e}_{\mathbf{j}}^\top \mathbf{\Lambda}_1 \mathbf{e}_{\mathbf{j}})(\mathbf{e}_{\mathbf{i}}^\top \mathbf{M}_2 \mathbf{e}_{\mathbf{i}} - \mathbf{e}_{\mathbf{j}}^\top \mathbf{\Lambda}_2 \mathbf{e}_{\mathbf{j}})},$$

which after replacing the uni-index notation with \mathbf{i} and \mathbf{j} with that of multi-index form concerning pairs instead, i.e., (i_1, i_2) and (j_1, j_2) , we get that $(\mathbb{L}_2)_{\mathbf{i}, \mathbf{j}}$ above is precisely equal to $\ell_{j_1, j_2}^{i_1, i_2}$ from before. This concludes the proof. \square

Corollary 4.1. 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, Lagrangian form, and generalized realization

Computing $\mathbb{L}_2 \mathbf{c}_2 = 0$, the null space of the bivariate Loewner matrix, we get:

$$\mathbf{c}_2^\top = \left[\underbrace{c_{1,1} \ \dots \ c_{1,k_2}}_{\alpha_1^\top} \ \middle| \ \underbrace{c_{2,1} \ \dots \ c_{2,k_2}}_{\alpha_2^\top} \ \middle| \ \dots \ \middle| \ \underbrace{c_{k_1,1} \ \dots \ c_{k_1,k_2}}_{\alpha_{k_1}^\top} \right] \in \mathbb{C}^{k_1 k_2},$$

and $\mathbf{c}_2^\top \odot \mathbb{W}_2 = [\beta_1^\top \ | \ \beta_2^\top \ | \ \dots \ | \ \beta_{k_1}^\top] \in \mathbb{C}^{k_1 k_2}$. These are the barycentric weights of the bivariate rational function $\mathbf{g}({}^1s, {}^2s)$:

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

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

Result 4.2 (2-D realization). Given Definition 2.2 and following Theorem 2.1, a generalized realization of $\mathbf{g}({}^1s, {}^2s)$ is obtained with the following settings: $\mathbb{A}^{Lag} = [\alpha_1 \ \alpha_2 \ \dots \ \alpha_{k_1}]$, $\mathbb{B}^{Lag} = [\beta_1 \ \beta_2 \ \dots \ \beta_{k_1}]$ and $\mathbf{\Gamma} = {}^1\mathbb{X}^{Lag}$, $\mathbf{\Delta} = {}^2\mathbb{X}^{Lag}$.

Example 4.2. Let us consider $\mathbf{H}({}^1s, {}^2s) = \mathbf{H}(s, t) = (s^2t)/(s - t + 1)$ of complexity $(2, 1)$. By evaluating \mathbf{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 gets the following response tableau \mathbf{tab}_2 :

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

Next, we construct the two-dimensional Loewner matrix and compute its null space, leading to (note that $\mathbf{rank}(\mathbb{L}_2) = 5$),

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

It follows that the rational two-variables function $\mathbf{g}(s, t) =$

$$\frac{\frac{1}{9(s-1)(t+1)} - \frac{1}{3(s-1)(t+3)} - \frac{2}{(s-3)(t+1)} + \frac{6}{(s-3)(t+3)} + \frac{25}{9(s-5)(t+1)} - \frac{25}{3(s-5)(t+3)}}{\frac{1}{3(s-1)(t+1)} - \frac{5}{9(s-1)(t+3)} - \frac{10}{9(s-3)(t+1)} + \frac{14}{9(s-3)(t+3)} + \frac{7}{9(s-5)(t+1)} - \frac{1}{(s-5)(t+3)}}$$

recovering the original function $\mathbf{H}(s, t)$. Then, a realization in the Lagrangian basis may be obtained as $\hat{\mathbf{H}}(s, t) = \mathbf{W}\Phi(s, t)^{-1}\mathbf{G}$, where

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

$$\mathbf{W} = [0 \ 0 \ 0 \ | \ 0 \ 0 \ -1], \quad \mathbf{G}^\top = [0 \ 0 \ | \ 1/2 \ -1/2 \ | \ 0 \ 0].$$

Next, by applying the Schur complement, the realization can be compressed to:

$$\Phi_c(s, t) = \left[\begin{array}{ccc|c} s-1 & 3-s & 0 & 0 \\ s-1 & 0 & 5-s & 0 \\ \hline -\frac{1}{3} & \frac{10}{9} & -\frac{7}{9} & t+1 \\ \frac{5}{9} & -\frac{14}{9} & 1 & -t-3 \end{array} \right] \text{ and}$$

$$\mathbf{W}_c(t) = [-\frac{2t}{9} \quad 4t \quad -\frac{50t}{9} \ | \ 0], \quad \mathbf{G}_c^\top = [0 \ 0 \ | \ \frac{1}{2} \quad -\frac{1}{2}].$$

The corresponding transfer function turns out to be:

$$\hat{\mathbf{H}}_c(s, t) = \mathbf{W}_c(t)\Phi_c(s, t)^{-1}\mathbf{G}_c = \frac{s^2t}{s-t+1}.$$

Therefore, the realization is further reduced. The price to pay with this compression step is the appearance of a parameter-dependent output matrix $\mathbf{W}_c(t)$.

4.3 The n -D case

This section provides the main result by extending the previous cases to multivariate functions in n variables.

4.3.1 Loewner matrices and Sylvester equations

Definition 4.4. Given the data described in (16), 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 matrix entries given by

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

Definition 4.5. Considering the data given in (16), we define the following matrices based on Kronecker products:

$$\begin{aligned} \Lambda_1 &= \text{diag}({}^1\lambda_1, \dots, {}^1\lambda_{k_1}) \otimes \mathbf{I}_{k_2} \otimes \mathbf{I}_{k_3} \otimes \cdots \otimes \mathbf{I}_{k_n}, & \mathbf{M}_1 &= \text{diag}({}^1\mu_1, \dots, {}^1\mu_{q_1}) \otimes \mathbf{I}_{q_2} \otimes \mathbf{I}_{q_3} \otimes \cdots \otimes \mathbf{I}_{q_n} \\ \Lambda_2 &= \mathbf{I}_{k_1} \otimes \text{diag}({}^2\lambda_1, \dots, {}^2\lambda_{k_2}) \otimes \mathbf{I}_{k_3} \otimes \cdots \otimes \mathbf{I}_{k_n}, & \mathbf{M}_2 &= \mathbf{I}_{q_1} \otimes \text{diag}({}^2\mu_1, \dots, {}^2\mu_{q_2}) \otimes \mathbf{I}_{q_3} \otimes \cdots \otimes \mathbf{I}_{q_n} \\ &\dots & &\dots \\ \Lambda_n &= \mathbf{I}_{k_1} \otimes \cdots \otimes \mathbf{I}_{k_{n-1}} \otimes \text{diag}({}^n\lambda_1, \dots, {}^n\lambda_{k_n}), & \mathbf{M}_n &= \mathbf{I}_{q_1} \otimes \cdots \otimes \mathbf{I}_{q_{n-1}} \otimes \text{diag}({}^n\mu_1, \dots, {}^n\mu_{q_n}), \end{aligned}$$

$$\begin{aligned} \mathbb{W}_n &= [\mathbf{w}_{1,1,\dots,1}, \mathbf{w}_{1,1,\dots,2}, \dots, \mathbf{w}_{1,1,\dots,k_n}, \mathbf{w}_{1,\dots,2,1}, \dots, \mathbf{w}_{k_1,k_2,\dots,k_n}], & \mathbf{R}_n &= \mathbf{1}_{k_1 k_2 \cdots k_n}^\top, \\ \mathbb{V}_n &= [\mathbf{v}_{1,1,\dots,1}, \mathbf{v}_{1,1,\dots,2}, \dots, \mathbf{v}_{1,1,\dots,q_n}, \mathbf{v}_{1,\dots,2,1}, \dots, \mathbf{v}_{q_1,q_2,\dots,q_n}]^\top & \text{and } \mathbf{L}_n &= \mathbf{1}_{q_1 q_2 \cdots q_n}. \end{aligned}$$

Theorem 4.3. The n -D Loewner matrix as introduced in Definition 4.4 is the solution of the following set of coupled Sylvester equations:

$$\begin{cases} \mathbf{M}_n \mathbb{X}_1 - \mathbb{X}_1 \Lambda_n &= \mathbb{V}_n \mathbf{R}_n - \mathbf{L}_n \mathbb{W}_n, \\ \mathbf{M}_{n-1} \mathbb{X}_2 - \mathbb{X}_2 \Lambda_{n-1} &= \mathbb{X}_1, \\ &\dots \\ \mathbf{M}_2 \mathbb{X}_{n-1} - \mathbb{X}_{n-1} \Lambda_2 &= \mathbb{X}_{n-2}, \\ \mathbf{M}_1 \mathbb{L}_n - \mathbb{L}_n \Lambda_1 &= \mathbb{X}_{n-1}. \end{cases}$$

Proof. We replicate the train of thought from the proof of Theorem 4.2. By substitution, we resolve the explicit formulation for matrices \mathbb{X}_1 and \mathbb{X}_2 , as:

$$(\mathbb{X}_1)_{i,j} = \frac{\mathbf{e}_i^\top \mathbb{V}_n - \mathbb{W}_n \mathbf{e}_j}{\mathbf{e}_i^\top \mathbf{M}_n \mathbf{e}_i - \mathbf{e}_j^\top \Lambda_n \mathbf{e}_j}, \quad (\mathbb{X}_2)_{i,j} = \frac{\mathbf{e}_i^\top \mathbb{V}_n - \mathbb{W}_n \mathbf{e}_j}{(\mathbf{e}_i^\top \mathbf{M}_{n-1} \mathbf{e}_i - \mathbf{e}_j^\top \Lambda_{n-1} \mathbf{e}_j)(\mathbf{e}_i^\top \mathbf{M}_n \mathbf{e}_i - \mathbf{e}_j^\top \Lambda_n \mathbf{e}_j)}.$$

By mathematical induction, the following relationships hold: $(\mathbb{L}_n)_{i,j} =$

$$\begin{aligned} &\frac{\mathbf{e}_i^\top \mathbb{V}_n - \mathbb{W}_n \mathbf{e}_j}{(\mathbf{e}_i^\top \mathbf{M}_1 \mathbf{e}_i - \mathbf{e}_j^\top \Lambda_1 \mathbf{e}_j)(\mathbf{e}_i^\top \mathbf{M}_2 \mathbf{e}_i - \mathbf{e}_j^\top \Lambda_2 \mathbf{e}_j) \cdots (\mathbf{e}_i^\top \mathbf{M}_n \mathbf{e}_i - \mathbf{e}_j^\top \Lambda_n \mathbf{e}_j)} = \frac{\mathbf{e}_i^\top \mathbb{V}_n - \mathbb{W}_n \mathbf{e}_j}{\prod_{k=1}^n (\mathbf{e}_i^\top \mathbf{M}_k \mathbf{e}_i - \mathbf{e}_j^\top \Lambda_k \mathbf{e}_j)} \\ &= \frac{\mathbf{v}_{i_1, i_2, \dots, i_n} - \mathbf{w}_{j_1, j_2, \dots, j_n}}{({}^1\mu_{i_1} - {}^1\lambda_{j_1})({}^2\mu_{i_2} - {}^2\lambda_{j_2}) \cdots ({}^n\mu_{i_n} - {}^n\lambda_{j_n})} = \ell_{j_1, j_2, \dots, j_n}^{i_1, i_2, \dots, i_n}. \quad \square \end{aligned}$$

Remark 4.1. An alternative definition of the n -D Loewner matrix \mathbb{L}_n can be provided by characterizing its (\mathbf{i}, \mathbf{j}) element explicitly, i.e., as follows:

$$(\mathbb{L}_n)_{i,j} = \frac{\mathbf{e}_i^\top \mathbb{V}_n - \mathbb{W}_n \mathbf{e}_j}{\prod_{k=1}^n (\mathbf{e}_i^\top \mathbf{M}_k \mathbf{e}_i - \mathbf{e}_j^\top \Lambda_k \mathbf{e}_j)},$$

where $\mathbf{i} = q_n \cdots q_2(i_1 - 1) + q_n \cdots q_2(i_2 - 1) + \cdots + q_n(i_{n-1} - 1) + i_n$, $1 \leq i_k \leq q_k$, $1 \leq k \leq n$ and $\mathbf{j} = k_n \cdots k_2(j_1 - 1) + k_n \cdots k_2(j_2 - 1) + \cdots + k_n(j_{n-1} - 1) + j_n$, $1 \leq j_k \leq k_k$, $1 \leq k \leq n$. For example, if $n = 3$, we get that $\mathbf{i} = q_3 q_2(i_1 - 1) + q_3(i_2 - 1) + i_3$, $1 \leq i_k \leq q_k$, $1 \leq k \leq n$ and $\mathbf{j} = k_3 k_2(j_1 - 1) + k_3(j_2 - 1) + j_3$, $1 \leq j_k \leq k_k$, $1 \leq k \leq n$.

4.3.2 Null space, Lagrangian form and generalized realization

Computing $\mathbb{L}_n \mathbf{c}_n = 0$, the null space of the n -variables Loewner matrix, is based on the following relationships: $\mathbf{c}_n^\top = [\alpha_1 \mid \alpha_2 \mid \cdots \mid \alpha_{k_1}] \in \mathbb{C}^{k_1 k_2 \cdots k_n}$ where

$$\begin{aligned} \alpha_1 &= [c_{1,\dots,1,1} \cdots c_{1,\dots,1,k_n} \mid c_{1,\dots,2,1} \cdots c_{1,\dots,2,k_n} \mid \cdots \mid \\ &\quad c_{1,k_2,\dots,k_{n-1},1} \cdots c_{1,k_2,\dots,k_{n-1},k_n}] \\ \alpha_2 &= [c_{2,\dots,1,1} \cdots c_{2,\dots,1,k_n} \mid \cdots] \\ \alpha_{k_1} &= [c_{k_1,k_2,\dots,1} \cdots c_{k_1,k_2,\dots,k_n}] \end{aligned}$$

and $\mathbf{c}_n^\top \odot \mathbb{W}_n = [\beta_1 \ \beta_2 \ \cdots \ \beta_{k_1}] \in \mathbb{C}^{k_1 k_2 \cdots k_n}$, contain the so-called barycentric weights of the n -variables rational function $\mathbf{g}(^1s, ^2s, \dots, ^ns)$ given by

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

which interpolates $\mathbf{H}(^1s, ^2s, \dots, ^ns)$ at the support points $\{^1\lambda_{j_1}, ^2\lambda_{j_2}, \dots, ^n\lambda_{j_n}\}$.

Result 4.3 (n -D realization, for $k = 1$). Given [Definition 2.2](#) and following [Theorem 2.1](#), a generalized realization of $\mathbf{g}(^1s, ^2s, \dots, ^ns)$ is obtained using $\mathbb{A}^{Lag} = [\alpha_1 \ \alpha_2 \ \cdots \ \alpha_{k_1}]$, $\mathbb{B}^{Lag} = [\beta_1 \ \beta_2 \ \cdots \ \beta_{k_1}]$, $\mathbf{\Gamma} = {}^1\mathbb{X}^{Lag}$ and $\mathbf{\Delta} = {}^2\mathbb{X}^{Lag} \otimes \cdots \otimes {}^n\mathbb{X}^{Lag}$.

Example 4.3. Consider the three variable rational function $\mathbf{H}(s, t, p) = (s + pt)/(p^2 + s + t)$ 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 three-dimensional Loewner matrix \mathbb{L}_3 has **rank** $(\mathbb{L}_3) = 11$ and

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

Following [Result 4.3](#), we may obtain the realization $(\mathbf{W}, \mathbf{\Phi}(s, t, p), \mathbf{G})$. By arranging as $(s) - (t, p)$, one obtains a realization of dimension $m = 11$. Instead, by arranging as $(s, t) - (p)$ we get $m = 9$. Then $\hat{\mathbf{H}}(s) = \mathbf{W}\mathbf{\Phi}(s, t, p)^{-1}\mathbf{G}$, where $\mathbf{\Phi}(s, t, p) =$

$$\left(\begin{array}{ccc|ccc|ccc|ccc} (s-2)(t-1) & -(s-2)(t-3) & -(t-1)(s-4) & (s-4)(t-3) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1-s}{2} - \frac{t}{2} & \frac{s}{2} - 1 & \frac{t}{2} - \frac{3}{2} & \frac{s}{2} - 2 & 2 - \frac{s}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} - \frac{t}{2} & \frac{t}{2} - \frac{3}{2} & \frac{t}{2} - \frac{1}{2} & \frac{3}{2} - \frac{t}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \frac{1}{2} & -\frac{15}{28} & -\frac{15}{28} & \frac{4}{7} & p-5 & p-5 & 0 & 0 & 0 & 0 & 0 \\ -\frac{39}{28} & \frac{41}{28} & \frac{41}{28} & -\frac{43}{28} & 6-p & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{13}{14} & -\frac{27}{28} & -\frac{27}{28} & 1 & 0 & 7-p & 0 & 0 & 0 & 0 & 0 \\ \hline \frac{1}{8} & -\frac{17}{56} & -\frac{9}{56} & \frac{19}{56} & 0 & 0 & p-5 & p-5 & \frac{1}{2} & 0 & 0 \\ -\frac{2}{7} & \frac{5}{7} & \frac{5}{7} & -\frac{11}{14} & 0 & 0 & 6-p & 0 & -1 & 0 & 0 \\ \frac{9}{56} & -\frac{23}{56} & -\frac{11}{56} & \frac{25}{56} & 0 & 0 & 0 & 7-p & \frac{1}{2} & 0 & 0 \end{array} \right)$$

and $\mathbf{W} = -\mathbf{e}_9^\top$, $\mathbf{G}^\top = [\mathbf{0}_{1,3} \mid 1/2 \ -1 \ 1/2 \mid \mathbf{0}_{1,3}]$. With the above partitioning, we obtain

$\kappa = 2 \times 2$ (associated to 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 $J_4 = [j_1^4, j_2^4]$.

By applying the Schur complement, the resolvent can be compressed to size 7 as $\mathbf{\Phi}_c(s, t, p) =$

$$\left[\begin{array}{cc|ccccc} s-2 & 4-s & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & -\frac{15}{28} & (p-5)(t-1) & (p-5)(t-1) & \frac{t}{2} - \frac{1}{2} & \frac{5}{2} - \frac{p}{2} & \frac{5}{2} - \frac{p}{2} \\ -\frac{39}{28} & \frac{41}{28} & -(p-6)(t-1) & 0 & 1-t & \frac{p}{2} - 3 & 0 \\ \frac{13}{14} & -\frac{27}{28} & 0 & -(p-7)(t-1) & \frac{t}{2} - \frac{1}{2} & 0 & \frac{p}{2} - \frac{7}{2} \\ -\frac{15}{28} & \frac{4}{7} & -(p-5)(t-3) & -(p-5)(t-3) & \frac{3}{2} - \frac{t}{2} & \frac{p}{2} - \frac{5}{2} & \frac{p}{2} - \frac{5}{2} \\ \frac{41}{28} & -\frac{43}{28} & (p-6)(t-3) & 0 & t-3 & 3 - \frac{p}{2} & 0 \\ -\frac{27}{28} & 1 & 0 & (p-7)(t-3) & \frac{3}{2} - \frac{t}{2} & 0 & \frac{7}{2} - \frac{p}{2} \end{array} \right],$$

then, $\mathbf{G}_c^\top = [0 \mid -1/4 \quad 1/2 \quad -1/4 \quad 1/4 \quad -1/2 \quad 1/4]$ and the output matrix becomes parameter-dependent $\mathbf{W}_c(t, p) = [-\frac{pt}{14} - \frac{1}{7} \quad \frac{pt}{14} + \frac{2}{7} \quad \mathbf{0}_{1,5}]$. The transfer function is also recovered:

$$\hat{\mathbf{H}}_c(s, t, p) = \mathbf{W}_c(t, p) \Phi_c(s, t, p)^{-1} \mathbf{G}_c = \frac{s + pt}{p^2 + s + t}.$$

5 Addressing the curse of dimensionality

From [Definition 4.4](#), it follows that the n -D Loewner matrix \mathbb{L}_n has dimension:

$$\mathbb{L}_n \in \mathbb{C}^{Q \times K}, \text{ where } Q = q_1 q_2 \dots q_n \text{ and } K = k_1 k_2 \dots k_n.$$

Clearly, the dimension increases exponentially with the number of parameters and the corresponding degrees (this is also obvious from the 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. Note also that the computationally most favorable case, $K = Q = N$, and the complexity is $\mathcal{O}(N^3)$ flop.

The need for the full matrix to perform the SVD decomposition renders the process applicability unfeasible for many data sets. Here, the curse of dimensionality (**C-o-D**) is addressed through a tailored n -D Loewner matrix null space decomposition. More specifically, in this section, we suggest an alternate approach allowing 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. Similarly to the previous sections, to illustrate our exposition, 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. Thus, this leads to a series of 1-D Loewner matrix null space computations.

5.1 Null space computation in the 2-D case

Let us first consider the two variables case.

Theorem 5.1. *Let $h_{i,j} \in \mathbb{C}$ be measurements of the transfer function $\mathbf{H}(^1s, ^2s)$, with $^1s_i, i = 1, \dots, n$, and $^2s_j, j = 1, \dots, m$. Let $k_1 = \lfloor n/2 \rfloor$ and $k_2 = \lfloor m/2 \rfloor$ be number of column interpolation points (see $P_c^{(2)}$ in [\(15\)](#)). The null space of the corresponding 2-D Loewner matrix⁴ is spanned by*

$$\mathcal{N}(\mathbb{L}_2) = \text{vec} \left[\mathbf{c}_1^{^2s_1} \cdot \left[\mathbf{c}_1^{^1s_{k_1}} \right]_1, \dots, \mathbf{c}_1^{^2s_{k_2}} \cdot \left[\mathbf{c}_1^{^1s_{k_1}} \right]_{k_2} \right], \quad (19)$$

where $\mathbf{c}_1^{^1s_{k_1}} = \mathcal{N}(\mathbb{L}_1^{^1s_{k_1}})$ is the null space of the 1-D Loewner matrix for frozen $^1s = ^1s_{k_1}$, and $\mathbf{c}_1^{^2s_j} = \mathcal{N}(\mathbb{L}_1^{^2s_j})$ is the j -th null space of the 1-D Loewner matrix for frozen $^2s_j = \{^2s_1, \dots, ^2s_{k_2}\}$.

Proposition 5.1. *Given the setup in [Theorem 5.1](#), the null space computation flop complexity is $k_1^3 + k_1 k_2^3$ or $k_2^3 + k_2 k_1^3$, instead of $k_1^3 k_2^3$.*

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

$$\underbrace{\begin{bmatrix} & t_1 & t_2 & \cdots & t_m \\ s_1 & h_{1,1} & h_{1,2} & \cdots & h_{1,m} \\ s_2 & h_{2,1} & h_{2,2} & \cdots & h_{2,m} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ s_{n-1} & h_{n-1,1} & h_{n-1,2} & \cdots & h_{n-1,m} \\ s_n & h_{n,1} & h_{n,2} & \cdots & h_{n,m} \end{bmatrix}}_{\text{tab}_2}, \quad \underbrace{\begin{bmatrix} & t_1 & t_2 & \cdots & t_m \\ s_1 & \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,m} \\ s_2 & \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,m} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ s_{n-1} & \alpha_{n-1,1} & \alpha_{n-1,2} & \cdots & \alpha_{n-1,m} \\ s_n & \alpha_{n,1} & \alpha_{n,2} & \cdots & \alpha_{n,m} \end{bmatrix}}_{\mathbb{A}^{\text{Lag}}}.$$

⁴The 2-D Loewner matrix is constructed by substituting in [\(15\)](#): $^1\lambda_{j_1} = s_{j_1}, ^1\mu_{i_1} = s_{i_1}$ and $^2\lambda_{j_2} = t_{j_2}, ^2\mu_{i_2} = t_{i_2}$ (where $j_1 = 1, \dots, k_1 \leq \lfloor n/2 \rfloor, j_2 = 1, \dots, k_2 \leq \lfloor m/2 \rfloor, i_1 = k_1 + 1, \dots, n$ and $i_2 = k_1 + 1, \dots, m$), and $\mathbf{w}_{i,j} = h_{j_1, j_2}$ and $\mathbf{v}_{i,j} = h_{i_1, i_2}$. In addition $q_1 = n - k_1$ and $q_2 = m - k_2$.

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

Table 3: 2-D tableau for [Example 5.1](#): \mathbf{tab}_2 .

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

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

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

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

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

Finally, multiplying the j -th column with the j -th entry of the magenta row yields a vector that spans the desired null space of \mathbb{L}_2 . The procedure requires computing the null 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. \square

Example 5.1. Continuing [Example 4.2](#), 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 4.2](#), we invoke [Theorem 5.1](#). We thus construct a sequence of 1-D Loewner matrices as follows⁵:

- First construct a 1-D Loewner matrix along 1s , for ${}^2s = {}^2\lambda_2 = -3$, i.e. considering data of

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

$\mathbf{tab}_2(:, 2)$ (second column). This leads to

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

- 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:

$$\begin{aligned} \mathbb{L}_1^{1\lambda_1} &= \begin{bmatrix} \frac{1}{6} & \frac{1}{10} \\ \frac{1}{9} & \frac{1}{15} \end{bmatrix} \Rightarrow \mathbf{c}_1^{1\lambda_1} = \begin{bmatrix} -\frac{3}{5} \\ 1 \end{bmatrix}, \\ \mathbb{L}_1^{1\lambda_2} &= \begin{bmatrix} \frac{6}{5} & \frac{6}{7} \\ \frac{9}{10} & \frac{9}{14} \end{bmatrix} \Rightarrow \mathbf{c}_1^{1\lambda_2} = \begin{bmatrix} -\frac{5}{7} \\ 1 \end{bmatrix}, \quad \mathbb{L}_1^{1\lambda_3} = \begin{bmatrix} \frac{75}{28} & \frac{25}{12} \\ \frac{15}{7} & \frac{5}{3} \end{bmatrix} \Rightarrow \mathbf{c}_1^{1\lambda_3} = \begin{bmatrix} -\frac{7}{9} \\ 1 \end{bmatrix}. \end{aligned}$$

- Finally, the scaled null space vector $\hat{\mathbf{c}}_2$ reads

$$\hat{\mathbf{c}}_2 = \begin{bmatrix} \mathbf{c}_1^{1\lambda_1} \cdot [\mathbf{c}_1^{2\lambda_2}]_1 & \mathbf{c}_1^{1\lambda_2} \cdot [\mathbf{c}_1^{2\lambda_2}]_2 & \mathbf{c}_1^{1\lambda_3} \cdot [\mathbf{c}_1^{2\lambda_2}]_3 \end{bmatrix}^\top = \mathbf{c}_2,$$

which is equal to the one obtained directly with the 2-D Loewner matrix (see [Example 4.2](#)). 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 $3^3 = 27$ **flop**, and three 2×2 1-D Loewner matrices \rightsquigarrow null space computation is $2^3 = 8$ **flop**. Thus, $27 + 3 \times 8 = 51$ **flop** are needed here while $6^3 = 216$ **flop** were required in [Example 4.2](#), involving \mathbb{L}_2 directly. Note that the very same result may be obtained by first computing $\mathcal{N}(\mathbb{L}_1^{1\lambda_3}) \in \mathbb{R}^2$, then $\mathcal{N}(\mathbb{L}_1^{2\lambda_2}) \in \mathbb{R}^3$. In this case, the computational cost would be $2^3 + 2 \times 3^3 = 62$ **flop**.

5.2 Null space computation in the 3-D case

Let us now consider the three variables case.

Theorem 5.2. Let $h_{i,j,k} \in \mathbb{C}$ be measurements of the response of a transfer function $\mathbf{H}(^1s, ^2s, ^3s)$, along with 1s_i , 2s_j and 3s_k ($i = 1, \dots, n$, $j = 1, \dots, m$ and $k = 1, \dots, p$), and let $k_1 \leq \lfloor n/2 \rfloor$, $k_2 \leq \lfloor m/2 \rfloor$ and $k_3 \leq \lfloor p/2 \rfloor$ be number of column interpolation points (see $P_c^{(3)}$ in (16), $n = 3$). The null space of the corresponding 3-D Loewner matrix is spanned by

$$\mathcal{N}(\mathbb{L}_3) = \mathbf{vec} \begin{bmatrix} \mathbf{c}_2^{1s_1} \cdot [\mathbf{c}_1^{(2s_{k_2}, 3s_{k_3})}]_1, \dots, \mathbf{c}_2^{1s_{k_1}} \cdot [\mathbf{c}_1^{(2s_{k_2}, 3s_{k_3})}]_{k_1} \end{bmatrix},$$

where $\mathbf{c}_1^{(2s_{k_2}, 3s_{k_3})} = \mathcal{N}(\mathbb{L}_1^{(2s_{k_2}, 3s_{k_3})})$ is the null space of the 1-D Loewner matrix for frozen $\{^2s, ^3s\} = \{^2s_{k_2}, ^3s_{k_3}\}$, and $\mathbf{c}_2^{1s_j} = \mathcal{N}(\mathbb{L}_2^{1s_j})$ is the j -th null space of the 2-D Loewner matrix for frozen $^1s_j = \{^1s_1, \dots, ^1s_{k_1}\}$.

Proof. The proof follows the proof 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, which concludes the proof. \square

Remark 5.1 (Toward recursivity). From [Theorem 5.2](#), it follows that the 3-D Loewner matrix null space may be obtained with 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 5.2](#)).

Example 5.2. We continue with continue [Example 4.3](#). We now illustrate how much the complexity and dimensionality issue may be reduced when applying the suggested recursive process. First, remember that the three-dimensional Loewner matrix \mathbb{L}_3 has a dimension of 12 and its null space is $\mathbf{c}_3^\top = [\mathbf{c}_{3,1}^\top \parallel \mathbf{c}_{3,2}^\top]$, given as

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

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 4.3](#), 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 $\mathbf{tab}_3(:, 2, 3)$, leading to

$$\mathbb{L}_1^{(2\lambda_2, 3\lambda_3)} = \begin{bmatrix} \frac{31}{2700} & \frac{31}{2800} \\ \frac{31}{2592} & \frac{31}{2688} \end{bmatrix} \quad \text{and} \quad \mathbf{c}_1^{(2\lambda_2, 3\lambda_3)} = \begin{bmatrix} -\frac{27}{28} \\ 1 \end{bmatrix}.$$

- 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 s_2 and s_3 , 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:

$$\mathbf{c}_2^{^1\lambda_1} = \left[-\frac{14}{27}, \frac{13}{9}, -\frac{26}{27}, \frac{5}{9}, -\frac{41}{27}, 1 \right]^\top, \quad \mathbf{c}_2^{^1\lambda_2} = \left[-\frac{15}{28}, \frac{41}{28}, -\frac{27}{28}, \frac{4}{7}, -\frac{43}{28}, 1 \right]^\top,$$

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

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

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 the one 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 apply to $\mathbf{c}_2^{^1\lambda_2}$).

- First, one constructs the 1-D Loewner matrix along the second variable s_2 for frozen first and third variables, i.e. elements of $\mathbf{tab}_3(1, :, 3)$, leading to

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

- 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 following null spaces,

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

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

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

By 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 now inspecting the complexity, one observes that only a collection of 1-D Loewner matrices needs to be constructed, as well as their null spaces. Here, one (i) 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, being much lower than 1,728 flop for \mathbb{L}_3 . One may also notice that changing the variables 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 multi-variate Loewner matrices are no longer needed and can be replaced by a series of single variables, taming the curse of dimensionality.

5.3 Null space computation in the n -D case

We now state the second main result of this paper: [Theorem 5.3](#) allows to address the **C-o-D** related to the null space computation of the n -D Loewner matrix by splitting a n -D Loewner matrix null space into a 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 5.3. *Being given the tableau \mathbf{tab}_n as in [Table 2](#) in response of the n -variables \mathbf{H} function [\(2\)](#) evaluated at the data set [\(16\)](#), the null space of the corresponding n -D Loewner matrix \mathbb{L}_n , is spanned by*

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

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

Proof. The proof follows the one given for the 2-D and 3-D cases. \square

[Theorem 5.3](#) gives a generic process to compute the null space of an n -D Loewner matrix via a 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. In the next section, we will measure how much this contributes to taming the **C-o-D**, in terms of flop complexity.

5.4 Summary of the complexity

Let us now state the main complexity result, related to [Theorem 5.3](#). This theorem is the major argument for taming the **C-o-D**. It states the complexity for a n -D null space construction.

Theorem 5.4. *The flop number for the recursive approach [Theorem 5.3](#), is:*

$$\mathit{flop}_1(n) = \sum_{j=1}^n \left(n_j^3 \prod_{k=1}^j n_{k-1} \right) \text{ where } n_0 = 1. \quad (20)$$

Proof. Consider now the case of a function in n variables js , of degree $d_j > 1$, $j = 1, \dots, n$. [Table 4](#) presents 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:

$$\begin{aligned} \mathit{flop}_1(n) &= \nu_1^3 + (\nu_1) \nu_2^3 + \dots + (\nu_1 \nu_2 \dots \nu_{n-2}) \nu_{n-1}^3 + (\nu_1 \nu_2 \dots \nu_{n-2} \nu_{n-1}) \nu_n^3 \\ &= \nu_1^3 + \nu_1 \left(\nu_2^3 + \nu_2 \left(\nu_3^3 + \dots + \nu_{n-2} \left(\nu_{n-1}^3 + \nu_{n-1} \left(\nu_n^3 \right) \right) \right) \right). \end{aligned}$$

It readily follows that the number of flop above is minimized, if the variables are arranged so that $\nu_i \geq \nu_{i+1}$, for $i = 1, \dots, n - 1$. This concludes the proof. \square

# of variables of \mathbf{H}	# \mathbb{L}_1 matrix	Size of each \mathbb{L}_1	flop per \mathbb{L}_1
n	$\nu_1 \nu_2 \cdots \nu_{n-2} \nu_{n-1}$	ν_n	ν_n^3
$n-1$	$\nu_1 \nu_2 \cdots \nu_{n-2}$	ν_{n-1}	ν_{n-1}^3
\vdots	\vdots	\vdots	\vdots
3	$\nu_1 \nu_2$	ν_3	ν_3^3
2	ν_1	ν_2	ν_2^3
1	1	ν_1	ν_1^3

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

In [Figure 1](#), we show the result in [Theorem 5.4](#) (cascaded n -D Loewner) and compare it to the reference full \mathbb{L}_n null space computation via SVD⁶, of complexity $\mathcal{O}(N^3)$ (dashed blue line) and with two standard complexity references, namely $\mathcal{O}(N^2)$ (dotted blue line) and $\mathcal{O}(N \log(N))$ (solid blue line). More specifically, in [Figure 1](#), we first compute the number of required flop for different n -D Loewner matrices dimension, constructed with a different number of variables j s (for $j = 1, \dots, 5$) each of varying orders ($d_j = 1, \dots, 12$). In [Figure 1](#) we report all combinations, obtained with optimal ordering, i.e. $\nu_i \geq \nu_{i+1}$ (for $i = 1, \dots, n-1$) (red dots), and all the worst combinations (yellow dots). An estimation of the order of complexity is computed by seeking an upper bound of these point clouds, leading to an upper (solid red line), lower (solid yellow line), and average (solid black line) flop complexity.

From the above considerations, it follows that the proposed null space computation method given in [Theorem 5.3](#), leads to the computational complexity of [Theorem 5.4](#), then leads to a complexity drop from an order 3 to order 1.5 (and at least a factor of 2, when a sub-optimal ordering is chosen), which drastically reduces the data storage, and both the computational burden and time. As illustrated in [Section 7](#), this allows treating problems with a high number of variables, with a reasonable computational time and complexity.

6 Data-driven multivariate model construction

Now the theoretical contributions have been detailed, this section will now focus on the numerical aspects and considerations to construct the realization from data measurements.

6.1 Two algorithms

In what follows, we provide details to the proposed two algorithms. The first one is a direct method (without requiring any sort of iteration) inspired by the procedure in [\[34\]](#), while the second is iterative, inspired by the p-AAA algorithm introduced in [\[48\]](#). These two procedures are outlined in [Algorithm 1](#) and [Algorithm 2](#). For additional, more in-depth details, we refer the reader to the corresponding works [\[34, 48\]](#).

6.2 Discussion

The main difference between the two algorithms is that [Algorithm 1](#) is direct while [Algorithm 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 1](#), the process first needs to estimate the rational order along each variable j s. Then, we construct the interpolation set [\(16\)](#) (here one may shuffle data and interpolate different blocks). From this initial data set, the n -D 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 4.3](#).

By considering [Algorithm 2](#), the process is almost similar. The difference between the two algorithms consists in the absence of the order detection process in the second algorithm. It is

⁶Notice that we consider here $K = Q = N$ to simplify the exposition.

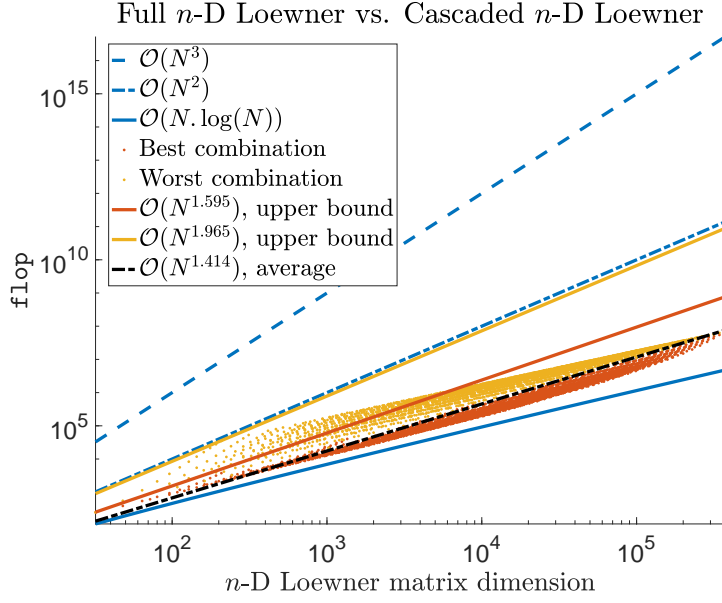


Figure 1: flop comparison. Cascaded n -D Loewner best and worst combination are respectively materialized by red and yellow dots. An upper bound of the best and worst combination is given with the red and yellow lines. The average complexity is materialized by the dash-dotted black line. Full n -D Loewner is $\mathcal{O}(N^3)$ (blue dashed). Comparison with $\mathcal{O}(N^2)$ and $\mathcal{O}(N \log(N))$ references are shown in dash-dotted and solid blue lines.

Require: \mathbf{tab}_n as in Table 2

- 1: Check that interpolation points are disjoint.
- 2: Compute $d_j = \max_k \mathbf{rank} \mathbb{L}_1^{(k)}$, the order along variable j s (k being all possible combinations for frozen the variables $\{1s, \dots, k-1s, k+1s, \dots, ns\}$).
- 3: Construct (16), a sub-selection $P_c^{(n)}$ where $(k_1, k_2, \dots, k_n) = (d_1, d_2, \dots, d_n) + 1$; and $P_r^{(n)}$ where (q_1, q_2, \dots, q_n) gather the rest of the data.
- 4: Compute \mathbf{c}_n , the n -D Loewner matrix null space e.g. using Theorem 5.3.
- 5: Construct \mathbb{A}^{Lag} , \mathbb{B}^{Lag} , $\mathbf{\Gamma}$ and $\mathbf{\Delta}$ as in Result 4.3 with any left/right separation.
- 6: Construct multivariate realization as in Theorem 2.1.

Ensure: $\hat{\mathbf{H}}(1s, \dots, ns) = \mathbf{W}\mathbf{\Phi}(1s, 2s, \dots, ns)^{-1}\mathbf{G}$ interpolates $\mathbf{H}(1s, 2s, \dots, ns)$ along $P_c^{(n)}$.

Algorithm 1: Direct data-driven pROM construction

Require: \mathbf{tab}_n as in Table 2 and tolerance $\text{tol} > 0$

- 1: Check that interpolation points are disjoint.
- 2: **while** $\text{error} > \text{tol}$ **do**
- 3: Search the point indexes with maximal error (first iteration: pick any set).
- 4: Add points in $P_c^{(n)}$ and put the remaining ones in $P_r^{(n)}$, obtain (16).
- 5: Compute \mathbf{c}_n , the n -D Loewner matrix null space e.g. using Theorem 5.3.
- 6: Construct \mathbb{A}^{Lag} , \mathbb{B}^{Lag} , $\mathbf{\Gamma}$ and $\mathbf{\Delta}$ as in Result 4.3 with any left/right separation.
- 7: Construct multivariate realization as in Theorem 2.1.
- 8: Evaluate $\text{error} = \max \|\mathbf{tab}_n - \hat{\mathbf{tab}}_n\|$ where $\hat{\mathbf{tab}}_n$ is the evaluation of $\hat{\mathbf{H}}(1s, \dots, ns)$ along the support points.
- 9: **end while**

Ensure: $\hat{\mathbf{H}}(1s, \dots, ns) = \mathbf{W}\mathbf{\Phi}(1s, 2s, \dots, ns)^{-1}\mathbf{G}$ interpolates $\mathbf{H}(1s, 2s, \dots, ns)$ along $P_c^{(n)}$.

Algorithm 2: Adaptive data-driven pROM construction

instead 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 [43] and its parametric version from [48]; we similarly follow this greedy approach.

6.2.1 Transforming to a real arithmetic setup

All computational steps have been presented using complex data. This is the most straightforward manner to present this generic method. However, in system theory, it is often preferable 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 closed by conjugations. Then, by introducing the projection matrix $\mathbf{J} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -i & i \end{bmatrix}$ and by repeating it as many times as the complex conjugated points are given, allows to construct left and right projectors that preserves the realness of the n -D Loewner matrix and thus of the null space. For space limitations, we keep a rigorous exposition of the details for later communication. For more details on the exact procedure, we refer the reader to [34, Section A.2].

6.2.2 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 / hard point in the proposed process: the **null space computation**. Indeed, either in the full n -D and the recursive 1-D case, a null space must be computed. Numerically, there exist multiple ways to compute it: SVD or QR decomposition, linear resolution... Without entering into details out of the scope of this paper, many tuning variables may be adjusted to improve the accuracy. These elements are crucial to the success of the proposed solution. Investigations are left (and not forgiven) to future works. In the next section, all null spaces have been computed using the standard SVD routine of `Matlab`. For more details, the reader may refer to [28].

7 Numerical experiments

The effectiveness of the numerical procedures sketched in [Algorithm 1](#) and [Algorithm 2](#) is illustrated in this section, through some complex examples, involving multiple variables ranging from two to twenty. In what follows the computations have been performed on an Apple MacBook Air with 512 Go SSD and 16 Go RAM, with a M1 chip. The software used is `Matlab` 2022b.

7.1 A simple synthetic parametric model (2-D)

Let start with the simple example used in [34, Section 5.1] and [48, Section 3.2.1], which transfer function reads: $\mathbf{H}(s, p) = \frac{1}{1+25(s+p)^2} + \frac{0.5}{1+25(s-0.5)^2} + \frac{0.1}{p+25}$. We use the same sampling setting as in the above references. Along the s variable, 21 points linearly spaced points from $[-1, 1]$. For the direct method of [Algorithm 1](#) we alternatively sample the grid as ${}^1\lambda_{j_1} = [-1, -0.8, \dots, 1]$ and ${}^1\mu_{i_1} = [-0.9, -0.7, \dots, 0.9]$; Then, along the p variable, 21 points linearly spaced points from $[0, 1]$. For the direct method of [Algorithm 1](#) we alternatively sample the grid as ${}^2\lambda_{j_2} = [0, 0.1, \dots, 1]$ and ${}^2\mu_{i_2} = [0.05, 0.15, \dots, 0.95]$. First, we apply [Algorithm 1](#) and obtain the single variables singular value decay reported in [Figure 2](#) (left), suggesting approximation orders along (s, p) of $(d_1, d_2) = (4, 3)$, being precisely the one of the equation $\mathbf{H}(s, p)$ 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 2](#). In [Table 5](#), we report the iterations of this algorithm when computing the null space with either the full 2-D version ([Table 5a](#)) or the recursive 1-D one ([Table 5b](#)). 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. $\text{tol}=10^{-6}$. By now comparing the `flop` complexity, the benefit of the proposed recursive 1-D approach with

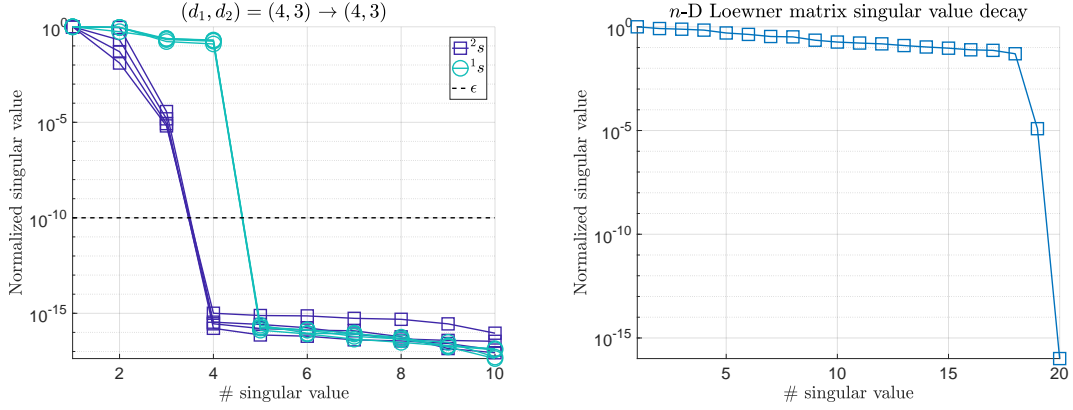


Figure 2: 2-D simple synthetic model: Algorithm 1 normalized singular values of each 1-D (left) and the 2-D (right) Loewner matrices.

respect to the 2-D one is clearly emphasized, even for such a simple setup. Indeed, while the latter is of 9,953 flop ($= 1 + 2^3 + 6^3 + 12^3 + 20^3$), the former leads to 680 flop ($= 2 + 10 + 51 + 172 + 445$), being 14 times smaller.

The mismatch for the three configurations over all the sampling points of \mathbf{tab}_2 data is close to machine precision for all configurations.

(a) Algorithm 2 (full \mathbb{L}_n)				(b) Algorithm 2 (recursive \mathbb{L}_1)				
Iter.	${}^1\lambda_{j_1}$	${}^2\lambda_{j_2}$	(k_1, k_2)	flop	${}^1\lambda_{j_1}$	${}^2\lambda_{j_2}$	(k_1, k_2)	flop
1	0	0	(1, 1)	1^3	0	0	(1, 1)	2
2	-1		(2, 1)	2^3	-1		(2, 1)	10
3	-0.9	0.9	(3, 2)	6^3	0.1	0.05	(3, 2)	51
4	-0.1	0.2	(4, 3)	12^3	-0.9	0.75	(4, 3)	172
5	0.6	1	(5, 4)	20^3	0.7	0.15	(5, 4)	445

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

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 ones with Algorithm 1 and Algorithm 2 (with recursive 1-D null space).

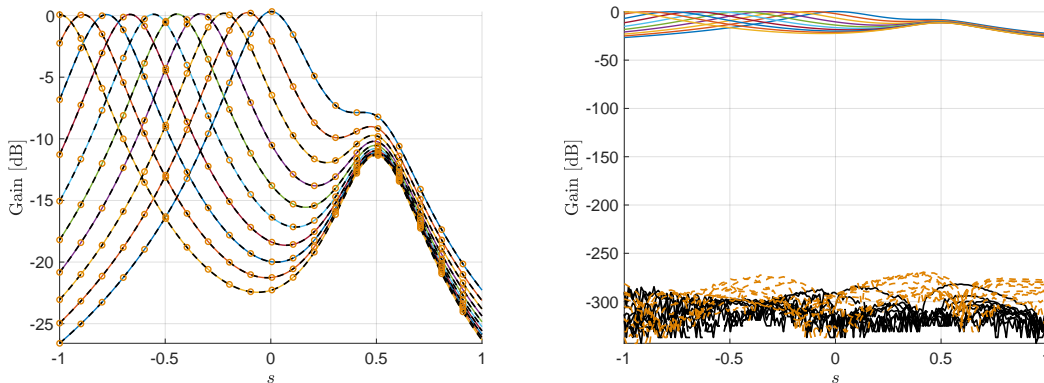


Figure 3: 2-D simple model: frequency responses (left) and errors (right); original vs. Algorithm 1 (black lines), and Algorithm 2 (orange dots and dashed lines).

7.2 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 [22]⁷. 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 dynamic. This $G(s, v)$ is known only at a few sampled frequencies and some true airspeed, *i.e.*, $G(i\omega_i, v_j)$ where $i = 1, \dots, 150$ and $j = 1, \dots, 10$. 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 $i[10, 35]$ and ${}^1\mu_{i_1} = -{}^1\lambda_{j_1}$; Along the v variable, ${}^2\lambda_{j_2}$ are 5 linearly spaced points between $[4.77, 5.21] \cdot 10^3$ and ${}^2\mu_{i_2}$, 5 linearly spaced between $[4.82, 5.27] \cdot 10^3$; Along the m variable, ${}^3\lambda_{j_3}$ are 5 linearly spaced points between $[1.52, 1.66] \cdot 10^3$ and ${}^2\mu_{i_2}$, 5 linearly spaced between $[1.54, 1.68] \cdot 10^3$.

Here, the data is a 3-dimensional tensor $\mathbf{tab}_3 \in \mathbb{C}^{300 \times 10 \times 10}$. By applying Algorithm 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.

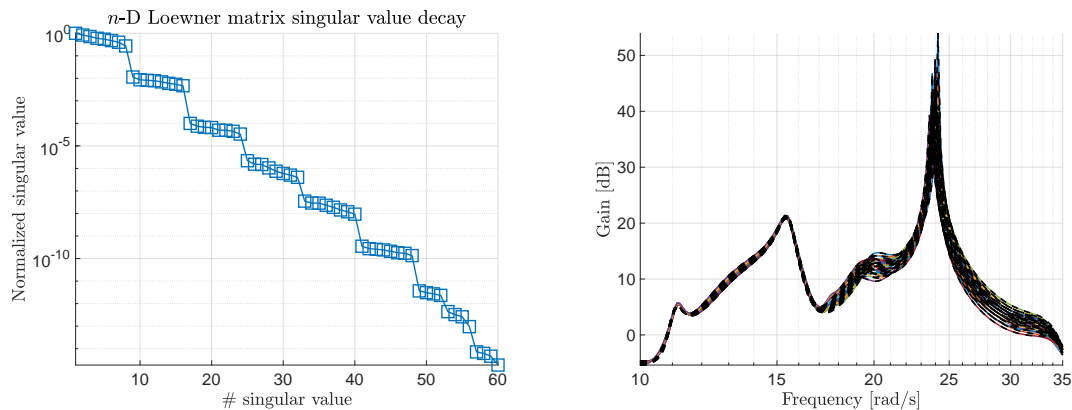


Figure 4: 3-D flutter model: 3-D Loewner matrix singular values (left) and frequency responses (right). Original (solid colored) and pROM (black dashed).

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 [47, 54, 22] for details and industrial applications.

7.3 A multi-variate 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 following rational function in 20 variables

$$\mathbf{H}(s_1, s_2, \dots, s_{20}) = \frac{3 \cdot s_1^3 + 4 \cdot s_8 + 12s + 13s \cdot s_{14} + 15s}{1s + 2s^2 \cdot 3s + 4s + 5s + 6s + 7s \cdot 8s + 9s \cdot 10s \cdot 11s + 13s + 13s^3 \cdot \pi + 17s + 18s \cdot 19s - 20s},$$

with a complexity of (3, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1). By applying Algorithm 1 with the recursive 1-D null space construction, the barycentric coefficients $\mathbf{c}_n \in \mathbb{C}^{6291456}$ are obtained with a computational complexity of 54,263,884 flop, computed in 55 minutes. As provided in the

⁷Acknowledgments to P. Vuillemin for (modified) data generation.

supplementary material, this vector allows reconstructing the original model \mathbf{H} with $\hat{\mathbf{H}}$, with an absolute error $\approx 10^{-11}$, for a random parameter selection.

Applying the full n -D Loewner version instead would have theoretically required the construction of a Loewner matrix of dimension $N = 6,291,456$, which null space computation would cost about $2.49 \cdot 10^{20}$ flops, being prohibitive, at least on a desktop computer. Note that storing such a $N \times N$ n -D Loewner matrix would require $6291456^2 \cdot \frac{8}{2^{30}} = 294,912$ GB in double precision and 147,456 GB in simple precision (where 8 is the number of bytes in double precision and 2^{30} scales the bytes into Giga Bytes). Here, constructing the complete 20-D tensor is a waste of time and energy. Instead, we consider that the unknown function \mathbf{H} can be evaluated when needed.

8 Conclusions

We 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 also showed the relationship between n -D Loewner and Sylvester equations. Then, as the numerical complexity explodes with the number of data and variables, we introduce a recursive 1-D null space procedure, equivalent to the full n -D one. This process reduces the complexity from $\mathcal{O}(N^3)$ to approximately $\mathcal{O}(N^{1.4})$. This becomes a major step toward **taming the curse of dimensionality**. Throughout the paper we apply these results to some numerical examples, demonstrating their effectiveness. Last but not least, we claim that the contributions presented are not limited to the system dynamic and rational approximation fields, but rather apply to many scientific computing domains, including tensor approximation and nonlinear eigenvalue problems, for which dimensionality remains an issue.

Supplementary material

This contribution presents the main theoretical results, emphasizing technical details and procedures supported by some numerical experiments. To provide an exhaustive account of our findings (in the direction of applying the proposed approaches for a multitude of test cases) with more insight for interested readers, additional numerical results are made available at

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

This web page gathers a collection of benchmarks that numerically demonstrate the scalability of the proposed algorithms. It also includes the model description, the associated n -D tensor (if accessible), and the obtained solutions, allowing reproducibility of the results in the paper.

References

- [1] Amsallem, D., Farhat, C.: An online method for interpolating linear parametric reduced-order models. *SIAM J. Sci. Comput.* **33**(5), 2169–2198 (2011). doi:[10.1137/100813051](https://doi.org/10.1137/100813051)
- [2] Andreuzzi, F., Demo, N., Rozza, G.: A dynamic mode decomposition extension for the forecasting of parametric dynamical systems. *SIAM J. Appl. Dyn. Syst.* **22**(3), 2432–2458 (2023). doi:[10.1137/22M1481658](https://doi.org/10.1137/22M1481658)
- [3] Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems, *Adv. Des. Control*, vol. 6. SIAM Publications, Philadelphia, PA (2005). doi:[10.1137/1.9780898718713](https://doi.org/10.1137/1.9780898718713)
- [4] Antoulas, A.C., Anderson, B.D.O.: On the scalar rational interpolation problem. *IMA J. Math. Control. Inf.* **3**(2-3), 61–88 (1986). doi:[10.1093/imamci/3.2-3.61](https://doi.org/10.1093/imamci/3.2-3.61)
- [5] Antoulas, A.C., Beattie, C.A., Gugercin, S.: Interpolatory Methods for Model Reduction. Computational Science & Engineering. Society for Industrial and Applied Mathematics, Philadelphia, PA (2020). doi:[10.1137/1.9781611976083](https://doi.org/10.1137/1.9781611976083)

-
- [6] Antoulas, A.C., Ionita, A.C., Lefteriu, S.: On two-variable rational interpolation. *Linear Algebra and its Applications* **436**(8), 2889–2915 (2012). doi:[10.1016/j.laa.2011.07.017](https://doi.org/10.1016/j.laa.2011.07.017). Special Issue dedicated to Danny Sorensen’s 65th birthday
- [7] Antoulas, A.C., Lefteriu, S., Ionita, A.C.: A tutorial introduction to the Loewner framework for model reduction. In: *Model Reduction and Approximation*, chap. 8, pp. 335–376. SIAM (2017). doi:[doi/10.1137/1.9781611974829.ch8](https://doi.org/10.1137/1.9781611974829.ch8)
- [8] Austin, A.P., Krishnamoorthy, M., Leyffer, S., Mrenna, S., Müller, J., Schulz, H.: Practical algorithms for multivariate rational approximation. *Computer Physics Communications* **261**, 107663 (2021). doi:[10.1016/j.cpc.2020.107663](https://doi.org/10.1016/j.cpc.2020.107663)
- [9] Baur, U., Beattie, C., Benner, P., Gugercin, S.: Interpolatory projection methods for parameterized model reduction. *SIAM J. Comput.* **33**(5), 2489–2518 (2011)
- [10] Baur, U., Benner, P., Feng, L.: Model order reduction for linear and nonlinear systems: A system-theoretic perspective. *Archives of Computational Methods in Engineering* **21**(4), 331–358 (2014). doi:[10.1007/s11831-014-9111-2](https://doi.org/10.1007/s11831-014-9111-2)
- [11] Beltrán-Pulido, A., Bilonis, I., Aliprantis, D.: Physics-informed neural networks for solving parametric magnetostatic problems. *IEEE Transactions on Energy Conversion* **37**(4), 2678–2689 (2022). doi:[10.1109/TEC.2022.3180295](https://doi.org/10.1109/TEC.2022.3180295)
- [12] Benner, P., Grivet-Talocia, S., Quarteroni, A., Rozza, G., Schilders, W., Silveira, L.M.: *Model Order Reduction, Volume 1: System- and Data-Driven Methods and Algorithms*. De Gruyter, Berlin, Boston (2021). doi:[10.1515/9783110498967](https://doi.org/10.1515/9783110498967). URL <https://doi.org/10.1515/9783110498967>
- [13] Benner, P., Grivet-Talocia, S., Quarteroni, A., Rozza, G., Schilders, W., Silveira, L.M.: *Model Order Reduction, Volume 2: Snapshot-Based Methods and Algorithms*. De Gruyter, Berlin, Boston (2021). doi:[10.1515/9783110671490](https://doi.org/10.1515/9783110671490). URL <https://doi.org/10.1515/9783110671490>
- [14] Benner, P., Gugercin, S., Willcox, K.: A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.* **57**(4), 483–531 (2015). doi:[10.1137/130932715](https://doi.org/10.1137/130932715)
- [15] Berrut, J.P., Trefethen, L.N.: Barycentric Lagrange interpolation. *SIAM Review* **46**(3), 501–517 (2004). doi:[10.1137/S00361445024177](https://doi.org/10.1137/S00361445024177)
- [16] Bradde, T., Grivet-Talocia, S., Zanco, A., Calafiore, G.C.: Data-driven extraction of uniformly stable and passive parameterized macromodels. *IEEE Access* **10**, 15786–15804 (2022). doi:[10.1109/ACCESS.2022.3147034](https://doi.org/10.1109/ACCESS.2022.3147034)
- [17] Brennan, M.C., Embree, M., Gugercin, S.: Contour integral methods for nonlinear eigenvalue problems: A systems theoretic approach. *SIAM Rev.* **65**(2), 439–470 (2023). doi:[10.1137/20M1389303](https://doi.org/10.1137/20M1389303)
- [18] Chellappa, S., Feng, L., de la Rubia, V., Benner, P.: Adaptive interpolatory MOR by learning the error estimator in the parameter domain. *Model Reduction of Complex Dynamical Systems* pp. 97–117 (2021). doi:[10.1007/978-3-030-72983-7_5](https://doi.org/10.1007/978-3-030-72983-7_5)
- [19] Constantine, P.G.: Active subspaces: Emerging ideas for dimension reduction in parameter studies. SIAM (2015). doi:[10.1137/1.9781611973860](https://doi.org/10.1137/1.9781611973860)
- [20] Debals, O., Van Barel, M., De Lathauwer, L.: Löwner-based blind signal separation of rational functions with applications. *IEEE Transactions on Signal Processing* **64**(8), 1909–1918 (2015). doi:[10.1109/TSP.2015.2500179](https://doi.org/10.1109/TSP.2015.2500179)
- [21] Demo, N., Strazzullo, M., Rozza, G.: An extended physics informed neural network for preliminary analysis of parametric optimal control problems. *Computers & Mathematics with Applications* **143**, 383–396 (2023). doi:[10.1016/j.camwa.2023.05.004](https://doi.org/10.1016/j.camwa.2023.05.004)

-
- [22] dos Reis de Souza, A., Poussot-Vassal, C., Vuillemin, P., Zucco, J.T.: Aircraft flutter suppression: from a parametric model to robust control. In: Proceedings of European Control Conference. Bucharest, Romania (2023). doi:[10.23919/ECC57647.2023.10178141](https://doi.org/10.23919/ECC57647.2023.10178141)
- [23] Geuss, M., Lohmann, B.: STABLE - a stability algorithm for parametric model reduction by matrix interpolation. *Mathematical and Computer Modelling of Dynamical Systems* **22**(4), 307–322 (2016). doi:[10.1080/13873954.2016.1198383](https://doi.org/10.1080/13873954.2016.1198383)
- [24] Gosea, I.V., Gugercin, S.: Data-driven modeling of linear dynamical systems with quadratic output in the AAA framework. *Journal of Scientific Computing* **91**(1), 16 (2022). doi:[10.1007/s10915-022-01771-5](https://doi.org/10.1007/s10915-022-01771-5)
- [25] Gosea, I.V., Gugercin, S., Unger, B.: Parametric model reduction via rational interpolation along parameters. In: 60th IEEE Conference on Decision and Control (CDC), December 14–17, Austin, TX, USA, pp. 6895–6900 (2021). doi:[10.1109/CDC45484.2021.9682841](https://doi.org/10.1109/CDC45484.2021.9682841)
- [26] Gosea, I.V., Poussot-Vassal, C., Antoulas, A.: Data-driven modeling and control of large-scale dynamical systems in the Loewner framework. *Handbook of Numerical Analysis* **23**(Numerical Control: Part A), 499–530 (2022). doi:[10.1016/bs.hna.2021.12.015](https://doi.org/10.1016/bs.hna.2021.12.015)
- [27] Grasedyck, L., Kressner, D., Tobler, C.: A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen* **36**(1), 53–78 (2013). doi:[10.1002/gamm.201310004](https://doi.org/10.1002/gamm.201310004)
- [28] Guglielmi, N., Overton, M., Stewart, G.: An efficient algorithm for computing the generalized null space decomposition. *SIAM J. Matrix Anal. Appl.* **36**(1), 38–54 (2015). doi:[10.1137/140956737](https://doi.org/10.1137/140956737)
- [29] Guo, M., Manzoni, A., Amendt, M., Conti, P., Hesthaven, J.S.: Multi-fidelity regression using artificial neural networks: Efficient approximation of parameter-dependent output quantities. *Computer methods in applied mechanics and engineering* **389**, 114378 (2022). doi:[10.1016/j.cma.2021.114378](https://doi.org/10.1016/j.cma.2021.114378)
- [30] Güttel, S., Negri Porzio, G.M., Tisseur, F.: Robust rational approximations of nonlinear eigenvalue problems. *SIAM J. Comput.* **44**(4), A2439–A2463 (2022). doi:[10.1137/20M1380533](https://doi.org/10.1137/20M1380533)
- [31] Güttel, S., Van Beeumen, R., Meerbergen, K., Michiels, W.: NLEIGS: A class of fully rational Krylov methods for nonlinear eigenvalue problems. *SIAM J. Comput.* **36**(6), A2842–A2864 (2014). doi:[10.1137/130935045](https://doi.org/10.1137/130935045)
- [32] Hesthaven, J.S., Rozza, G., Stamm, B.: Certified Reduced Basis Methods for Parametrized Partial Differential Equations. *SpringerBriefs in Mathematics*. Springer, Cham (2016). doi:[10.1007/978-3-319-22470-1](https://doi.org/10.1007/978-3-319-22470-1)
- [33] Hund, M., Mitchell, T., Mlinarić, P., Saak, J.: Optimization-based parametric model order reduction via $\mathcal{H}_2 \otimes \mathcal{L}_2$ first-order necessary conditions. *SIAM J. Sci. Comput.* **44**(3), A1554–A1578 (2022). doi:[10.1137/21M140290X](https://doi.org/10.1137/21M140290X)
- [34] Ionita, A., Antoulas, A.: Data-Driven Parametrized Model Reduction in the Loewner Framework. *SIAM J. Sci. Comput.* **36**(3), A984–A1007 (2014). doi:[10.1137/130914619](https://doi.org/10.1137/130914619)
- [35] Karachalios, D.S., Gosea, I.V., Antoulas, A.C.: The Loewner framework for system identification and reduction. In: P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W.H.A. Schilders, L.M. Silveira (eds.) *Methods and Algorithms, Handbook on Model Reduction*, vol. 1. De Gruyter (2021). doi:[10.1515/9783110498967-006](https://doi.org/10.1515/9783110498967-006)
- [36] Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. *SIAM Rev.* **51**(3), 455–500 (2009). doi:[10.1137/07070111X](https://doi.org/10.1137/07070111X)
- [37] Lietaert, P., Meerbergen, K., Pérez, J., Vandereycken, B.: Automatic rational approximation and linearization of nonlinear eigenvalue problems. *IMA Journal of Numerical Analysis* **42**(2), 1087–1115 (2022). doi:[10.1093/imanum/draa098](https://doi.org/10.1093/imanum/draa098)

-
- [38] Loewner, K.: Über monotone Matrixfunktionen. *Mathematische Zeitschrift* **38**(1), 177–216 (1934). doi:[10.1007/BF01170633](https://doi.org/10.1007/BF01170633)
- [39] Mayo, A.J., Antoulas, A.C.: A framework for the solution of the generalized realization problem. *Linear Algebra and Its Applications* **425**(2-3), 634–662 (2007). doi:[10.1016/j.laa.2007.03.008](https://doi.org/10.1016/j.laa.2007.03.008)
- [40] McQuarrie, S.A., Khodabakhshi, P., Willcox, K.E.: Nonintrusive reduced-order models for parametric partial differential equations via data-driven operator inference. *SIAM J. Sci. Comput.* **45**(4), A1917–A1946 (2023). doi:[10.1137/21M1452810](https://doi.org/10.1137/21M1452810)
- [41] Mlinarić, P., Benner, P., Gugercin, S.: Interpolatory necessary optimality conditions for reduced-order modeling of parametric linear time-invariant systems. *arXiv preprint arXiv:2401.10047* (2024). doi:[10.48550/arXiv.2401.10047](https://doi.org/10.48550/arXiv.2401.10047)
- [42] Mlinarić, P., Rave, S., Saak, J.: Parametric model order reduction using pyMOR. *Model Reduction of Complex Dynamical Systems* pp. 357–367 (2021). doi:[10.1007/978-3-030-72983-7_17](https://doi.org/10.1007/978-3-030-72983-7_17)
- [43] Nakatsukasa, Y., Sète, O., Trefethen, L.N.: The AAA algorithm for rational approximation. *SIAM J. Comput.* **40**(3), A1494–A1522 (2018). doi:[10.1137/16M1106122](https://doi.org/10.1137/16M1106122)
- [44] Nakatsukasa, Y., Sete, O., Trefethen, L.N.: The first five years of the AAA algorithm. *ArXiv preprint online* (2023). doi:[10.48550/arXiv.2312.03565](https://doi.org/10.48550/arXiv.2312.03565)
- [45] Nobile, F., Pradovera, D.: Non-intrusive double-greedy parametric model reduction by interpolation of frequency-domain rational surrogates. *ESAIM: Mathematical Modelling and Numerical Analysis* **55**(5), 1895–1920 (2021). doi:[10.1051/m2an/2021040](https://doi.org/10.1051/m2an/2021040)
- [46] Peherstorfer, B., Willcox, K.: Data-driven operator inference for nonintrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering* **306**, 196–215 (2016). doi:[10.1016/j.cma.2016.03.025](https://doi.org/10.1016/j.cma.2016.03.025)
- [47] Quero, D., Vuillemin, P., Poussot-Vassal, C.: A generalized eigenvalue solution to the flutter stability problem with true damping: the p-L method. *Journal of Fluids and Structures* **103**, 103266 (2021). doi:[10.1016/j.jfluidstructs.2021.103266](https://doi.org/10.1016/j.jfluidstructs.2021.103266)
- [48] Rodriguez, A.C., Balicki, L., Gugercin, S.: The p-AAA algorithm for data driven modeling of parametric dynamical systems. *SIAM J. Sci. Comput.* **45**(3), A1332–A1358 (2023). doi:[10.1137/20M1322698](https://doi.org/10.1137/20M1322698)
- [49] Sun, S., Feng, L., Chan, H.S., Miličić, T., Vidaković-Koch, T., Benner, P.: Parametric dynamic mode decomposition for nonlinear parametric dynamical systems. *arXiv preprint arXiv:2305.06197* (2023). doi:[10.48550/arXiv.2305.06197](https://doi.org/10.48550/arXiv.2305.06197)
- [50] Tu, J.H., Rowley, C.W., Luchtenburg, D.M., Brunton, S.L., Kutz, J.N.: On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics* **1**(2), 391–421 (2014). doi:[10.3934/jcd.2014.1.391](https://doi.org/10.3934/jcd.2014.1.391)
- [51] Van Dooren, P., Gallivan, K.A., Absil, P.A.: \mathcal{H}_2 -optimal model reduction of MIMO systems. *Applied Mathematics Letters* **21**(12), 53–62 (2008)
- [52] Verbeke, D., Ishteva, M., Dreesen, P.: MIMO transfer function decoupling by Loewner tensorization. *IFAC-PapersOnLine* **56**(2), 7300–7305 (2023). doi:[10.1016/j.ifacol.2023.10.342](https://doi.org/10.1016/j.ifacol.2023.10.342)
- [53] Vervliet, N., Debals, O., Sorber, L., De Lathauwer, L.: Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis. *IEEE Signal Processing Magazine* **31**(5), 71–79 (2014). doi:[10.1109/MSP.2014.2329429](https://doi.org/10.1109/MSP.2014.2329429)
- [54] Vojkovic, T., Quero, D., Poussot-Vassal, C., Vuillemin, P.: Low-order parametric state-space modeling of MIMO systems in the Loewner framework. *SIAM J. Appl. Dyn. Syst.* **22**(4), 3130–3164 (2023). doi:[10.1137/22M1509898](https://doi.org/10.1137/22M1509898)

-
- [55] Vuillemin, P., Kergus, P., Poussot-Vassal, C.: Hybrid Loewner data driven control. In: Proceedings of the IFAC World Congress. Berlin, Germany (2020). doi:[10.1016/j.ifacol.2020.12.1574](https://doi.org/10.1016/j.ifacol.2020.12.1574)
- [56] Xiao, Y.Q., Grivet-Talocia, S., Manfredi, P., Khazaka, R.: A novel framework for parametric Loewner matrix interpolation. IEEE Transactions on Components, Packaging and Manufacturing Technology **9**(12), 2404–2417 (2019). doi:[10.1109/TCPMT.2019.2948802](https://doi.org/10.1109/TCPMT.2019.2948802)
- [57] Yıldız, S., Goyal, P., Benner, P., Karasözen, B.: Learning reduced-order dynamics for parametrized shallow water equations from data. International Journal for Numerical Methods in Fluids **93**(8), 2803–2821 (2021). doi:[10.1002/fld.4998](https://doi.org/10.1002/fld.4998)
- [58] Yue, Y., Feng, L., Benner, P.: Reduced-order modelling of parametric systems via interpolation of heterogeneous surrogates. Advanced Modeling and Simulation in Engineering Sciences **6**, 10 (2019). doi:[10.1186/s40323-019-0134-y](https://doi.org/10.1186/s40323-019-0134-y)
- [59] Zanco, A., Grivet-Talocia, S., Bradde, T., De Stefano, M.: Enforcing passivity of parameterized LTI macromodels via Hamiltonian-driven multivariate adaptive sampling. IEEE Trans. on Computer-Aided Design of Integrated Circ. and Sys. **39**(1), 225–238 (2018). doi:[10.1109/TCAD.2018.2883962](https://doi.org/10.1109/TCAD.2018.2883962)