

Dimitrios S. Karachalios, Ion Victor Gosea, and
Athanasios C. Antoulas

6 The Loewner framework for system identification and reduction

Abstract: One of the main approaches to model reduction of both linear and nonlinear dynamical systems is by means of interpolation. This approach seeks reduced models whose transfer function matches that of the original system at selected interpolation points. Data-driven methods constitute an important special case. We start with an account of the Loewner framework in the linear case [52]. It constructs models from given data in a straightforward manner. An important attribute is that it provides a trade-off between accuracy of fit and complexity of the model. We compare this approach with other approximation methods and apply it to different test-cases. One of the case studies to which we apply the aforementioned methods is defined by the inverse of the Bessel function. We then turn our attention to the approximation of an Euler–Bernoulli beam model with Rayleigh damping. Further case studies include the approximation of two real valued functions with specific difficulties (discontinuity, sharp peaks). One computational tool is the SVD; its complexity is cubic in the number of data points. For large data sets the CUR factorization is a viable alternative. Note that its complexity is cubic as well but with respect to the dimension of the reduced order model (ROM). Another option is to use stochastic procedures such as randomized singular value decomposition (r-SVD) [41].

Keywords: Loewner framework, rational interpolation, model order reduction, data-driven, system identification, infinite dimensional systems


MSC 2010: 15-00, 37-00, 41-00, 49-00, 65-00, 93-00

6.1 Introduction

A challenging problem that computational linear algebra deals with is that of big data modeling. The problem consists mainly in constructing reduced complexity systems from input/output data. This contribution focuses on reduction via interpolation. The

Dimitrios S. Karachalios, Ion Victor Gosea, Data-Driven System Reduction and Identification (DRI) group, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstrasse 1, 39106 Magdeburg, Germany

Athanasios C. Antoulas, Data-Driven System Reduction and Identification (DRI) group, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstrasse 1, 39106 Magdeburg, Germany; and Electrical and Computer Engineering Department, Rice University Houston, 6100 Main St., Houston, TX 77005, USA; and Baylor College of Medicine, 1 Baylor Plaza, Houston, TX 77030, USA

Open Access. © 2021 Dimitrios S. Karachalios et al., published by De Gruyter.  This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

Loewner framework is a data-driven approach which can construct low order models from measurements. It can be applied to both frequency and time-domain data [56]. Here we will concentrate on frequency domain data. The Loewner framework will be implemented using (a) the SVD (singular value decomposition), (b) the CUR factorization, (c) randomized SVD (r-SVD). Its performance will be compared with that of the recently developed AAA algorithm see [53], the Vector Fitting approach [21, 40] and the IRKA algorithm [13].

The paper is composed of three sections. The first one covers the fundamentals of the Loewner framework starting from left and right interpolatory reduction. It concludes (a) by describing an interpolation property satisfied by reduced systems and (b) by making the procedure of obtaining real reduced models (despite complex interpolation points and values) explicit. Next the description of two algorithms namely, Loewner-SISO and Loewner-MIMO, is given. Finally two simple examples are presented and the role of generalized inverses outlined.

The second chapter describes methods for implementing the Loewner reduction, namely the SVD, the CUR factorization and the role of splitting the interpolation point in left and right sets. The third chapter illustrates the main features of the Loewner approach by means of seven case studies, namely, (a) the CD player, (b) an oscillating function, (c) the inverse of a Bessel function, (d) an Euler–Bernoulli beam, (e) a heat equation, (f) a function with two sharp peaks, and (g) the sign function. An epilogue and references conclude the presentation.

6.2 The Loewner framework and moment matching

The Loewner framework has attracted increased attention of researchers from various fields of applied mathematics and control engineering in the last 13 years. Consequently, a fair amount of contributions that are now available, deal with various aspects on further extending the framework and with its application to different test-cases. Below we provide an account of some of the work related to or inspired by the “Loewner framework” (see Table 6.1).

Consider linear, time-invariant systems with m inputs, n internal variables (states if \mathbf{E} is non-singular) and p outputs:

$$\Sigma : \begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), & \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), & \text{where} \\ \mathbf{E}, \mathbf{A} \in \mathbb{R}^{n \times n}, & \mathbf{B} \in \mathbb{R}^{n \times m}, & \mathbf{C} \in \mathbb{R}^{p \times n}. \end{cases} \quad (6.1)$$

We will denote this realization of the system by means of the quadruple $\Sigma = (\mathbf{C}, \mathbf{E}, \mathbf{A}, \mathbf{B})$. The associated transfer function is

$$\mathbf{H}(s) = \mathbf{C}\Phi(s)\mathbf{B} \quad \text{where } \Phi(s) = (s\mathbf{E} - \mathbf{A})^{-1} \in \mathbb{C}^{n \times n}. \quad (6.2)$$

Table 6.1: A collection of contributions related to the Loewner Framework.

Original paper [52] & tutorial paper [6]	Chapters 4 and 7 in the book [9]
Extension to parametrized linear systems [4, 42]	Application to modeling multi-port linear systems [48]
bilinear systems [5, 45, 46]	preserving the stability of the ROM [30]
quadratic systems [29, 36]	the Burgers equations [8]
quadratic-bilinear systems [32]	the Oseen equations [10]
linear switched systems [34]	preserving the structure of DAE systems [37]
polynomial systems [11, 16]	systems with delay [35, 59]
modeling from noisy data [20, 50]	approximating functions [31, 33, 43, 44]
modeling from time-domain data [56]	singular/rectangular systems [3]
	genes oscillations [7] and biological rhythms [68]
Perspective based on duality and application to bilinear differential [57, 58]	Interpretation based on interconnection and application to LTV systems [60, 61]

A common way to reduce the complexity of a system is by means of *Petrov–Galerkin projections*. Such projections are defined by means of two matrices $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times k}$, $k < n$, satisfying the condition that $\mathbf{W}^T \mathbf{V} \in \mathbb{R}^{k \times k}$ is invertible.¹

Definition 6.1. Consider $\mathbf{v}_i, \mathbf{w}_i \in \mathbb{R}^n$, $i = 1, \dots, k$, and let $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_k]$, $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_k] \in \mathbb{R}^{n \times k}$. The map defined by $\mathbf{\Pi}_1 = \mathbf{V}(\mathbf{V}^T \mathbf{V})^{-1} \mathbf{V}^T$, is an orthogonal projection onto the span of the columns of \mathbf{V} . If $\mathbf{W}^T \mathbf{V}$ is non-singular, $\mathbf{\Pi}_2 = \mathbf{V}(\mathbf{W}^T \mathbf{V})^{-1} \mathbf{W}^T$, is an oblique projector onto the span of the columns of \mathbf{V} , along the columns of \mathbf{W} . $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$ are usually referred to in the model reduction literature as **Galerkin** and **Petrov–Galerkin** projectors, respectively.

Reducing the system $\Sigma = (\mathbf{C}, \mathbf{E}, \mathbf{A}, \mathbf{B})$ defined above, by means of a *Petrov–Galerkin projection*, we obtain the reduced system $\hat{\Sigma} = (\hat{\mathbf{C}}, \hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{B}})$ with the reduced order matrices given by

$$\hat{\mathbf{C}} = \mathbf{C}\mathbf{V} \in \mathbb{R}^{p \times k}, \quad \hat{\mathbf{E}} = \mathbf{W}^T \mathbf{E}\mathbf{V}, \quad \hat{\mathbf{A}} = \mathbf{W}^T \mathbf{A}\mathbf{V} \in \mathbb{R}^{k \times k}, \quad \hat{\mathbf{B}} = \mathbf{W}^T \mathbf{B} \in \mathbb{R}^{k \times m}. \quad (6.3)$$

There are many ways of choosing Petrov–Galerkin projectors in order to achieve various goals. Here we will restrict our attention to *interpolatory* projections. Such projectors yield reduced models which match moments of the original system. These moments are values of transfer functions at selected frequencies, referred to as *interpolation points*.

Remark 6.1. The **D**-term. In the system representations to follow no explicit **D** terms will be considered. The reason is that such terms can be incorporated in the remaining

¹ The notation $(\cdot)^T$ indicates transposition of (\cdot) , while the notation $(\cdot)^*$ indicates transposition of (\cdot) followed by complex conjugation.

matrices of the realization, thus yielding what is known as a descriptor representation. Consider a rank-revealing factorization

$$\mathbf{D} = \mathbf{D}_1 \mathbf{D}_2 \quad \text{where } \mathbf{D}_1 \in \mathbb{R}^{p \times p}, \quad \mathbf{D}_2 \in \mathbb{R}^{p \times m},$$

and $\rho = \text{rank } \mathbf{D}$. It readily follows that

$$\hat{\mathbf{E}} = \begin{bmatrix} \mathbf{E} & \\ & \mathbf{0}_{\rho \times \rho} \end{bmatrix}, \quad \hat{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \\ & -\mathbf{I}_\rho \end{bmatrix}, \quad \hat{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{D}_2 \end{bmatrix}, \quad \hat{\mathbf{C}} = [\mathbf{C} \quad \mathbf{D}_1],$$

is a descriptor realization of the same system with no explicit \mathbf{D} -term. The reason for not considering explicit \mathbf{D} -terms, comes from the fact that the Loewner framework yields descriptor realizations with the \mathbf{D} -term incorporated in the rest of the realization.

6.2.1 Moments of a system

Given a matrix-valued function of time $\mathbf{h} : \mathbb{R} \rightarrow \mathbb{R}^{p \times m}$, its k th moment is

$$\eta_k = \int_0^\infty t^k \mathbf{h}(t) dt, \quad k = 0, 1, 2, \dots$$

If this function has a *Laplace transform* defined by $\mathbf{H}(s) = \mathcal{L}(\mathbf{h})(s) = \int_0^\infty \mathbf{h}(t) e^{-st} dt$, the k th moment of \mathbf{h} is, up to a sign, the k th derivative of \mathbf{H} evaluated at $s = 0$:

$$\eta_k = (-1)^k \frac{d^k}{ds^k} \mathbf{H}(s) |_{s=0} \in \mathbb{R}^{p \times m}, \quad k = 0, 1, 2, \dots$$

In the sequel, we will also make use of a generalized notion of moments, namely the *moments of \mathbf{h} around the (arbitrary) point $s_0 \in \mathbb{C}$* :

$$\eta_k(s_0) = \int_0^\infty t^k \mathbf{h}(t) e^{-s_0 t} dt.$$

These generalized moments turn out to be (up to a sign) the derivatives of $\mathbf{H}(s)$ evaluated at $s = s_0$:

$$\eta_k(s_0) = (-1)^k \frac{d^k}{ds^k} \mathbf{H}(s) |_{s=s_0} \in \mathbb{R}^{p \times m}, \quad k = 0, 1, 2, \dots$$

In this context, assuming for simplicity that $\mathbf{E} = \mathbf{I}$, the moments of \mathbf{h} at $s_0 \in \mathbb{C}$ are

$$\eta_k(s_0) = -k \mathbf{C} (s_0 \mathbf{I} - \mathbf{A})^{-(k+1)} \mathbf{B}, \quad k = 0, 1, 2, \dots,$$

provided that s_0 is not an eigenvalue of \mathbf{A} .

Notice that the moments determine the coefficients of the Laurent series expansion of the transfer function $\mathbf{H}(s)$ in the neighborhood of $s_i \in \mathbb{C}$; in particular

$$\begin{aligned}\mathbf{H}(s) &= \mathbf{H}(s_0) + \mathbf{H}^{(1)}(s_0) \frac{(s-s_0)}{1!} + \cdots + \mathbf{H}^{(k)}(s_0) \frac{(s-s_0)^k}{k!} + \cdots \\ &= \eta_0(s_0) + \eta_1(s_0) \frac{(s-s_0)}{1!} + \cdots + \eta_k(s_0) \frac{(s-s_0)^k}{k!} + \cdots.\end{aligned}$$

Approximation by moment matching

Given $\Sigma = (\mathbf{C}, \mathbf{E}, \mathbf{A}, \mathbf{B})$, consider the expansion of the transfer function around s_i , $i = 1, \dots, r$, as above. Approximation by moment matching consists in finding

$$\hat{\Sigma} = (\hat{\mathbf{C}}, \hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{B}}), \quad \hat{\mathbf{E}}, \hat{\mathbf{A}} \in \mathbb{R}^{k \times k}, \quad \hat{\mathbf{B}} \in \mathbb{R}^{k \times m}, \quad \hat{\mathbf{C}} \in \mathbb{R}^{p \times k}, \quad (6.4)$$

such that the expansion of the transfer function

$$\hat{\mathbf{H}}(s) = \hat{\eta}_0(s_i) + \hat{\eta}_1(s_i) \frac{(s-s_i)}{1!} + \hat{\eta}_2(s_i) \frac{(s-s_i)^2}{2!} + \hat{\eta}_3(s_i) \frac{(s-s_i)^3}{3!} + \cdots,$$

for appropriate $s_i \in \mathbb{C}$, and $\ell_i, r \in \mathbb{N}$, satisfies

$$\eta_j(s_i) = \hat{\eta}_j(s_i), \quad j = 1, 2, \dots, \ell_i \text{ and } i = 1, \dots, r.$$

This problem is also known as *rational interpolation*.

6.2.2 Rational interpolation by Petrov–Galerkin projection

Rational interpolation by projection was originally proposed in the work of Skelton and co-workers; see [65, 66, 67]. Contributions were also made by Grimme, Gallivan and van Dooren [23, 24, 38].

Suppose that we are given a system $\Sigma = (\mathbf{C}, \mathbf{E}, \mathbf{A}, \mathbf{B})$, assumed SISO (single-input single-output, i. e., $m = p = 1$) for simplicity. We wish to find lower dimensional models $\hat{\Sigma} = (\hat{\mathbf{C}}, \hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{B}})$, defined as in (6.3), $k < n$, such that $\hat{\Sigma}$ approximates the original system in an appropriate way.

Consider k distinct points $s_j \in \mathbb{C}$. Define \mathbf{V} as a *generalized controllability matrix*:

$$\mathbf{V} = [(s_1 \mathbf{E} - \mathbf{A})^{-1} \mathbf{B}, \dots, (s_k \mathbf{E} - \mathbf{A})^{-1} \mathbf{B}] \in \mathbb{C}^{n \times k}, \quad (6.5)$$

and let \mathbf{W}^* be any left inverse of \mathbf{V} . Then we have the following.

Proposition 6.1. $\hat{\Sigma}$ interpolates the transfer function of Σ at the points s_j , that is,

$$\mathbf{H}(s_j) = \mathbf{C}(s_j \mathbf{E} - \mathbf{A})^{-1} \mathbf{B} = \hat{\mathbf{C}}(s_j \hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1} \hat{\mathbf{B}} = \hat{\mathbf{H}}(s_j), \quad j = 1, \dots, k.$$

Proof. Denoting by $\mathbf{e}_j = [0 \cdots 0 \underbrace{1}_j 0 \cdots 0]^T$, the j th unit vector, we obtain the string of equalities below which lead to the desired result:

$$\begin{aligned}
 \hat{\mathbf{C}}(s_j \hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1} \hat{\mathbf{B}} &= \mathbf{C}\mathbf{V}(s_j \mathbf{W}^* \mathbf{E}\mathbf{V} - \mathbf{W}^* \mathbf{A}\mathbf{V})^{-1} \mathbf{W}^* \mathbf{B} \\
 &= \mathbf{C}\mathbf{V}(\mathbf{W}^* (s_j \mathbf{E} - \mathbf{A})\mathbf{V})^{-1} \mathbf{W}^* \mathbf{B} \\
 &= \mathbf{C}\mathbf{V}([\ast \cdots \ast \mathbf{W}^* \mathbf{B} \ast \cdots \ast])^{-1} \mathbf{W}^* \mathbf{B} \\
 &= [\mathbf{C}(s_1 \mathbf{E} - \mathbf{A})^{-1} \mathbf{B}, \dots, \mathbf{C}(s_k \mathbf{E} - \mathbf{A})^{-1} \mathbf{B}] \mathbf{e}_j \\
 &= \mathbf{C}(s_j \mathbf{E} - \mathbf{A})^{-1} \mathbf{B}.
 \end{aligned}$$

□

Next, we are concerned with matching the value of the transfer function at a given point $s_0 \in \mathbb{C}$, together with $k - 1$ derivatives. We define

$$\mathbf{V} = [(s_0 \mathbf{E} - \mathbf{A})^{-1} \mathbf{B}, (s_0 \mathbf{E} - \mathbf{A})^{-2} \mathbf{B}, \dots, (s_0 \mathbf{E} - \mathbf{A})^{-k} \mathbf{B}] \in \mathbb{C}^{n \times k}, \quad (6.6)$$

together with any left inverse \mathbf{W} . The following holds.

Proposition 6.2. $\hat{\Sigma}$ interpolates the transfer function of Σ at s_0 , together with $k-1$ derivatives at the same point:

$$\frac{(-1)^j}{j!} \frac{d^j}{ds^j} \mathbf{H}(s)|_{s=s_0} = \mathbf{C}(s_0 \mathbf{E} - \mathbf{A})^{-(j+1)} \mathbf{B} = \hat{\mathbf{C}}(s_0 \hat{\mathbf{E}} - \hat{\mathbf{A}})^{-(j+1)} \hat{\mathbf{B}} = \frac{(-1)^j}{j!} \frac{d^j}{ds^j} \hat{\mathbf{H}}(s)|_{s=s_0},$$

for $j = 0, 1, \dots, k - 1$.

Proof. Let \mathbf{V} be as defined in (6.6), and \mathbf{W} be such that $\mathbf{W}^T \mathbf{V} = \mathbf{I}_k$. It readily follows that the ℓ th power of the projected matrix $s_0 \hat{\mathbf{E}} - \hat{\mathbf{A}}$ satisfies

$$(s_0 \hat{\mathbf{E}} - \hat{\mathbf{A}})^\ell = \left[\underbrace{\ast \cdots \ast}_{\ell-1} \ast \mathbf{W}^* \mathbf{B} \underbrace{\ast \cdots \ast}_{k-\ell} \right].$$

Consequently $[\mathbf{W}^T (s_0 \mathbf{E} - \mathbf{A})\mathbf{V}]^{-\ell} \mathbf{W}^T \mathbf{B} = \mathbf{e}_\ell$, which finally implies

$$\hat{\mathbf{C}}(s_0 \hat{\mathbf{E}} - \hat{\mathbf{A}})^{-\ell} \hat{\mathbf{B}} = \mathbf{C}\mathbf{V}[\mathbf{W}^T (s_0 \mathbf{E} - \mathbf{A})\mathbf{V}]^{-\ell} \mathbf{W}^T \mathbf{B} = \mathbf{C}\mathbf{V}\mathbf{e}_\ell = \mathbf{C}(s_0 \mathbf{E} - \mathbf{A})^{-\ell} \mathbf{B},$$

for $\ell = 1, 2, \dots, k$. □

Since any $\bar{\mathbf{V}}$ that spans the same column space as \mathbf{V} achieves the same objective, projectors composed of combinations of the cases (6.5) and (6.6) achieve matching of an appropriate number of moments. To formalize this we will make use of the following matrices:

$$\mathcal{R}_k(\mathbf{E}, \mathbf{A}, \mathbf{B}; \sigma) = [(\sigma \mathbf{E} - \mathbf{A})^{-1} \mathbf{B} \quad (\sigma \mathbf{E} - \mathbf{A})^{-2} \mathbf{B} \quad \cdots \quad (\sigma \mathbf{E} - \mathbf{A})^{-k} \mathbf{B}].$$

Corollary 6.1.

- (a) If \mathbf{V} as defined above is replaced by $\bar{\mathbf{V}} = \mathbf{R}\mathbf{V}$, $\mathbf{R} \in \mathbb{R}^{k \times k}$, $\det \mathbf{R} \neq 0$, and \mathbf{W} by $\bar{\mathbf{W}} = \mathbf{R}^{-T}\mathbf{W}$, the same matching results hold true.
- (b) Let \mathbf{V} be such that

$$\text{span col } \mathbf{V} = \text{span col } [\mathcal{R}_{m_1}(\mathbf{E}, \mathbf{A}, \mathbf{B}; \sigma_1) \cdots \mathcal{R}_{m_\ell}(\mathbf{E}, \mathbf{A}, \mathbf{B}; \sigma_\ell)],$$

and \mathbf{W} be any left inverse of \mathbf{V} . The reduced system matches m_i moments at $\sigma_i \in \mathbb{C}$, $i = 1, \dots, \ell$.

6.2.3 Two-sided projections

The above results can be strengthened if the row span of the matrix \mathbf{W}^T is chosen to match the row span of a generalized observability matrix. In such a case twice as many moments can be matched with a reduced system of the same dimension. Here, in addition to points s_1, \dots, s_k , and the associated (6.5), we are given k additional distinct points s_{k+1}, \dots, s_{2k} . These points are used to define a *generalized observability matrix*:

$$\mathbf{W} = [(s_{k+1}\mathbf{E}^T - \mathbf{A}^T)^{-1}\mathbf{C}^T \cdots (s_{2k}\mathbf{E}^T - \mathbf{A}^T)^{-1}\mathbf{C}^T] \in \mathbb{C}^{n \times k}. \quad (6.7)$$

Proposition 6.3. Assuming that $\mathbf{W}^T\mathbf{V}$ has full rank, the projected system $\hat{\Sigma}$, interpolates the transfer function of Σ at the points s_i , $i = 1, \dots, 2k$.

Proof. The string of equalities that follows proves the desired result:

$$\begin{aligned} \hat{\mathbf{C}}(s_i\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1}\hat{\mathbf{B}} &= \mathbf{C}\mathbf{V}(s_i\mathbf{W}^T\mathbf{E}\mathbf{V} - \mathbf{W}^T\mathbf{A}\mathbf{V})^{-1}\mathbf{W}^T\mathbf{B} \\ &= \mathbf{C}\mathbf{V}(\mathbf{W}^T(s_i\mathbf{E} - \mathbf{A})\mathbf{V})^{-1}\mathbf{W}^T\mathbf{B} \\ &= \mathbf{C}\mathbf{V}(\mathbf{W}^T[\cdots \mathbf{B} \cdots])^{-1}\mathbf{W}^T\mathbf{B} \\ &= \mathbf{C}\mathbf{V}\mathbf{e}_i = \mathbf{C}(s_i\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}, \quad \text{for } i = 1, \dots, k, \\ &= \mathbf{C}\mathbf{V} \left(\begin{bmatrix} \vdots \\ \mathbf{C} \\ \vdots \end{bmatrix} \mathbf{V} \right)^{-1} \mathbf{W}^T\mathbf{B} \\ &= \mathbf{e}_i^T \mathbf{W}^T\mathbf{B} = \mathbf{C}(s_i\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}, \quad \text{for } i = k+1, \dots, 2k. \quad \square \end{aligned}$$

The projectors (see [62]) discussed in the previous section satisfy the *Sylvester equations* as shown next.

Proposition 6.4. With $\mathbf{\Lambda} = \text{diag}[\lambda_1, \dots, \lambda_k]$, $\mathbf{M} = \text{diag}[\mu_1, \dots, \mu_q]$, where λ_i and μ_j are mutually distinct, $\mathbf{R} = [1 \cdots 1] \in \mathbb{R}^k$, and $\mathbf{L} = [1 \cdots 1]^T \in \mathbb{R}^q$, the matrices \mathbf{V} and \mathbf{W} satisfy the Sylvester equations:

$$\mathbf{E}\mathbf{V}\mathbf{\Lambda} - \mathbf{A}\mathbf{V} = \mathbf{B}\mathbf{R} \quad \text{and} \quad \mathbf{M}\mathbf{W}^T\mathbf{E} - \mathbf{W}^T\mathbf{A} = \mathbf{L}\mathbf{C}. \quad (6.8)$$

6.2.4 Interpolatory model reduction for MIMO systems

In the general case of MIMO (multi-input multi-output) systems, the moments are $p \times m$ matrices. So, in the case of rational matrix interpolation the most appropriate way to proceed is to interpolate along certain directions. This leads to the so-called *tangential interpolation problem* (see e. g. [6, 21, 25]).

More precisely, we are given a set of *input/output* response measurements specified by left driving frequencies $\{\mu_i\}_{i=1}^q \subset \mathbb{C}$, using left input or tangential directions: $\{\boldsymbol{\ell}_i\}_{i=1}^q \subset \mathbb{C}^p$, producing left responses: $\{\mathbf{v}_i\}_{i=1}^q \subset \mathbb{C}^m$, and right driving frequencies: $\{\lambda_i\}_{i=1}^k \subset \mathbb{C}$, using right input or tangential directions: $\{\mathbf{r}_i\}_{i=1}^k \subset \mathbb{C}^m$, producing right responses: $\{\mathbf{w}_i\}_{i=1}^k$. We are thus given the *left data*: $(\mu_j; \boldsymbol{\ell}_j^T, \mathbf{v}_j^T)$, $j = 1, \dots, q$, and the *right data*: $(\lambda_i; \mathbf{r}_i, \mathbf{w}_i)$, $i = 1, \dots, k$. The problem is to find a rational $p \times m$ matrix $\mathbf{H}(s)$, such that

$$\mathbf{H}(\lambda_i)\mathbf{r}_i = \mathbf{w}_i, \quad i = 1, \dots, k, \quad \boldsymbol{\ell}_j^T \mathbf{H}(\mu_j) = \mathbf{v}_j^T, \quad j = 1, \dots, q. \quad (6.9)$$

The *left data* is rearranged compactly as

$$\mathbf{M} = \begin{bmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_q \end{bmatrix} \in \mathbb{C}^{q \times q}, \quad \mathbf{L} = \begin{bmatrix} \boldsymbol{\ell}_1^T \\ \vdots \\ \boldsymbol{\ell}_q^T \end{bmatrix} \in \mathbb{C}^{q \times p}, \quad \mathbf{V} = \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_q^T \end{bmatrix} \in \mathbb{C}^{q \times m}, \quad (6.10)$$

while the *right data* is rearranged as

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_k \end{bmatrix} \in \mathbb{C}^{k \times k}, \quad \mathbf{R} = [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \cdots \quad \mathbf{r}_k] \in \mathbb{C}^{m \times k}, \quad (6.11)$$

$$\mathbf{W} = [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \cdots \quad \mathbf{w}_k] \in \mathbb{C}^{p \times k}.$$

Interpolation points and tangential directions are determined by the problem or are selected to realize given model reduction goals. For SISO systems, i. e., $m = p = 1$, left and right directions can be taken equal to one ($\boldsymbol{\ell}_j = 1, \mathbf{r}_i = 1$) and hence the conditions above become

$$\left. \begin{aligned} \hat{\mathbf{H}}(\mu_j) &= \mathbf{H}(\mu_j) \Rightarrow \hat{\mathbf{H}}(\mu_j) = \mathbf{v}_j, \quad j = 1, \dots, q, \\ \hat{\mathbf{H}}(\lambda_i) &= \mathbf{H}(\lambda_i) \Rightarrow \hat{\mathbf{H}}(\lambda_i) = \mathbf{w}_i, \quad i = 1, \dots, k. \end{aligned} \right\} \quad (6.12)$$

6.2.5 The Loewner framework

Given a row array of complex numbers (μ_j, \mathbf{v}_j) , $j = 1, \dots, q$, and a column array, $(\lambda_i, \mathbf{w}_i)$, $i = 1, \dots, k$, (with λ_i and the μ_j mutually distinct) the associated *Loewner matrix* is

$$\mathbb{L} = \begin{bmatrix} \frac{\mathbf{v}_1 - \mathbf{w}_1}{\mu_1 - \lambda_1} & \cdots & \frac{\mathbf{v}_1 - \mathbf{w}_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{v}_q - \mathbf{w}_1}{\mu_q - \lambda_1} & \cdots & \frac{\mathbf{v}_q - \mathbf{w}_k}{\mu_q - \lambda_k} \end{bmatrix} \in \mathbb{C}^{q \times k}.$$

Definition 6.2. If \mathbf{g} is rational, i. e., $\mathbf{g}(s) = \frac{\mathbf{p}(s)}{\mathbf{q}(s)}$, for appropriate polynomials \mathbf{p} , \mathbf{q} , the McMillan degree or the complexity of \mathbf{g} is $\deg \mathbf{g} = \max\{\deg(\mathbf{p}), \deg(\mathbf{q})\}$.

Now, if $\mathbf{w}_i = \mathbf{g}(\lambda_i)$, and $\mathbf{v}_j = \mathbf{g}(\mu_j)$, are *samples* of a rational function \mathbf{g} , the *main property* of Loewner matrices asserts the following.

Theorem 6.1 ([52]). *Let \mathbb{L} be as above. If $k, q \geq \deg \mathbf{g}$, then $\text{rank } \mathbb{L} = \deg \mathbf{g}$. In other words the rank of \mathbb{L} encodes the complexity of the underlying rational function \mathbf{g} . Furthermore, the same result holds for matrix-valued functions \mathbf{g} .*

6.2.5.1 The Loewner pencil and interpolatory projectors

In the sequel we denote the *tangential* versions of (6.5) and (6.7) by \mathcal{R} , \mathcal{O} , respectively. For arbitrary k and q , these are defined as

$$\mathcal{R} = [(\lambda_1 \mathbf{E} - \mathbf{A})^{-1} \mathbf{B} \mathbf{r}_1, \dots, (\lambda_k \mathbf{E} - \mathbf{A})^{-1} \mathbf{B} \mathbf{r}_k] \in \mathbb{C}^{n \times k}, \quad (6.13)$$

$$\mathcal{O}^T = [(\mu_1 \mathbf{E}^T - \mathbf{A}^T)^{-1} \mathbf{C}^T \ell_1 \quad \dots \quad (\mu_q \mathbf{E}^T - \mathbf{A}^T)^{-1} \mathbf{C}^T \ell_q] \in \mathbb{C}^{n \times k}. \quad (6.14)$$

It readily follows that the reduced quantities $\hat{\mathbf{E}}$ and $\hat{\mathbf{A}}$ form a *Loewner pencil*:

$$\hat{\mathbf{E}} = -\mathcal{O} \mathbf{E} \mathcal{R} = - \begin{bmatrix} \frac{\mathbf{v}_1^T \mathbf{r}_1 - \ell_1^T \mathbf{w}_1}{\mu_1 - \lambda_1} & \dots & \frac{\mathbf{v}_1^T \mathbf{r}_k - \ell_1^T \mathbf{w}_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{v}_q^T \mathbf{r}_1 - \ell_q^T \mathbf{w}_1}{\mu_q - \lambda_1} & \dots & \frac{\mathbf{v}_q^T \mathbf{r}_k - \ell_q^T \mathbf{w}_k}{\mu_q - \lambda_k} \end{bmatrix} = -\mathbb{L} \in \mathbb{C}^{q \times k}, \quad (6.15)$$

$$\hat{\mathbf{A}} = -\mathcal{O} \mathbf{A} \mathcal{R} = - \begin{bmatrix} \frac{\mu_1 \mathbf{v}_1^T \mathbf{r}_1 - \ell_1^T \mathbf{w}_1 \lambda_1}{\mu_1 - \lambda_1} & \dots & \frac{\mu_1 \mathbf{v}_1^T \mathbf{r}_k - \ell_1^T \mathbf{w}_k \lambda_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mu_q \mathbf{v}_q^T \mathbf{r}_1 - \ell_q^T \mathbf{w}_1 \lambda_1}{\mu_q - \lambda_1} & \dots & \frac{\mu_q \mathbf{v}_q^T \mathbf{r}_k - \ell_q^T \mathbf{w}_k \lambda_k}{\mu_q - \lambda_k} \end{bmatrix} = -\mathbb{L}_s \in \mathbb{C}^{q \times k}, \quad (6.16)$$

$$\hat{\mathbf{B}} = \mathcal{O} \mathbf{B} = \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_q^T \end{bmatrix} = \mathbb{V} \in \mathbb{C}^{q \times m}, \quad \hat{\mathbf{C}} = \mathbf{C} \mathcal{R} = [\mathbf{w}_1 \quad \dots \quad \mathbf{w}_k] = \mathbb{W} \in \mathbb{C}^{p \times k}. \quad (6.17)$$

The resulting quadruple $(\mathbb{W}, \mathbb{L}, \mathbb{L}_s, \mathbb{V})$ is called the *Loewner quadruple*.

Lemma 6.1. *Upon multiplication of the first equation in (6.8) with \mathcal{O} on the left and the second by \mathcal{R} on the right we obtain*

$$\mathbb{L}_s - \mathbb{L} \mathbf{A} = \mathbb{V} \mathbf{R} \quad \text{and} \quad \mathbb{L}_s - \mathbf{M} \mathbb{L} = \mathbf{L} \mathbb{W}. \quad (6.18)$$

By adding/subtracting appropriate multiples of these expressions it follows that the Loewner quadruple satisfies the Sylvester equations

$$\mathbf{M} \mathbb{L} - \mathbb{L} \mathbf{A} = \mathbb{V} \mathbf{R} - \mathbf{L} \mathbb{W} \quad \text{and} \quad \mathbf{M} \mathbb{L}_s - \mathbb{L}_s \mathbf{A} = \mathbf{M} \mathbb{V} \mathbf{R} - \mathbf{L} \mathbb{W} \mathbf{A}. \quad (6.19)$$

Theorem 6.2. *Assume that the pencil $(\mathbb{L}_s, \mathbb{L})$ is regular.² Then $\mathbf{H}(s) = \mathbf{W}(\mathbb{L}_s - s\mathbb{L})^{-1}\mathbf{V}$, satisfies the tangential interpolation condition (6.9).*

Proof. Multiplying the first Sylvester equation by s and subtracting it from equation the second one, we get

$$\mathbf{M}(\mathbb{L}_s - s\mathbb{L}) - (\mathbb{L}_s - s\mathbb{L})\mathbf{A} = (\mathbf{M} - s\mathbf{I})\mathbf{V}\mathbf{R} - \mathbf{L}\mathbf{W}(\mathbf{A} - s\mathbf{I}).$$

Multiplying this equation by \mathbf{e}_i on the right and setting $s = \lambda_i$, we obtain

$$\begin{aligned} (\mathbf{M} - \lambda_i\mathbf{I})(\mathbb{L}_s - \lambda_i\mathbb{L})\mathbf{e}_i &= (\mathbf{M} - \lambda_i\mathbf{I})\mathbf{V}\mathbf{r}_i \Rightarrow (\mathbb{L}_s - \lambda_i\mathbb{L})\mathbf{e}_i = \mathbf{V}\mathbf{r}_i \\ &\Rightarrow \mathbf{W}\mathbf{e}_i = \mathbf{W}(\mathbb{L}_s - \lambda_i\mathbb{L})^{-1}\mathbf{V}\mathbf{r}_i. \end{aligned}$$

Thus $\mathbf{w}_i = \mathbf{H}(\lambda_i)\mathbf{r}_i$. Next, we multiply the above equation by \mathbf{e}_j^T on the left and set $s = \mu_j$:

$$\begin{aligned} \mathbf{e}_j^T(\mathbb{L}_s - \mu_j\mathbb{L})(\mathbf{A} - \mu_j\mathbf{I}) &= \mathbf{e}_j^T\mathbf{L}\mathbf{W}(\mathbf{A} - \mu_j\mathbf{I}) \Rightarrow \mathbf{e}_j^T(\mathbb{L}_s - \mu_j\mathbb{L}) = \mathbf{e}_j^T\mathbf{W} \\ &\Rightarrow \mathbf{e}_j^T\mathbf{V} = \mathbf{e}_j^T\mathbf{W}(\mathbb{L}_s - \mu_j\mathbb{L})^{-1}\mathbf{V}. \end{aligned}$$

Thus $\mathbf{v}_j^T = \mathbf{e}_j^T\mathbf{H}(\mu_j)$. □

Remark 6.2 (Parametrization of all interpolants of complexity equal to the size of \mathbb{L}). With $\mathbf{K} \in \mathbb{C}^{p \times m}$, the Sylvester equations can be rewritten as

$$\begin{aligned} \mathbf{M}\mathbb{L} - \mathbb{L}\mathbf{A} &= (\mathbf{V} - \mathbf{L}\mathbf{K})\mathbf{R} - \mathbf{L}(\mathbf{W} - \mathbf{K}\mathbf{R}) \quad \text{and} \\ \mathbf{M}(\mathbb{L}_s + \mathbf{L}\mathbf{K}\mathbf{R}) - (\mathbb{L}_s + \mathbf{L}\mathbf{K}\mathbf{R})\mathbf{A} &= \mathbf{M}(\mathbf{V} - \mathbf{L}\mathbf{K})\mathbf{R} - \mathbf{L}(\mathbf{W} - \mathbf{K}\mathbf{R})\mathbf{A}. \end{aligned}$$

These equations imply that $(\bar{\mathbf{W}}, \mathbb{L}, \bar{\mathbb{L}}_s, \bar{\mathbf{V}})$ is an interpolant for all $\mathbf{K} \in \mathbb{C}^{p \times m}$, where $\bar{\mathbb{L}}_s = \mathbb{L}_s + \mathbf{L}\mathbf{K}\mathbf{R}$, $\bar{\mathbf{V}} = \mathbf{V} - \mathbf{L}\mathbf{K}\mathbf{R}$ and $\bar{\mathbf{W}} = \mathbf{W} - \mathbf{K}\mathbf{R}$.

6.2.5.2 Construction of interpolants

If the pencil $(\mathbb{L}_s, \mathbb{L})$ is regular, then $\mathbf{E} = -\mathbb{L}$, $\mathbf{A} = -\mathbb{L}_s$, $\mathbf{B} = \mathbf{V}$, $\mathbf{C} = \mathbf{W}$, is a minimal interpolant of the data, i. e., $\mathbf{H}(s) = \mathbf{W}(\mathbb{L}_s - s\mathbb{L})^{-1}\mathbf{V}$, interpolates the data. Otherwise, as shown in [52], problem (6.9) has a solution provided that

$$\text{rank}[s\mathbb{L} - \mathbb{L}_s] = \text{rank}[\mathbb{L}, \mathbb{L}_s] = \text{rank} \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} = r,$$

for all $s \in \{\lambda_j\} \cup \{\mu_i\}$. Consider then the short SVDs:

$$[\mathbb{L}, \mathbb{L}_s] = \mathbf{Y}\hat{\Sigma}_r\tilde{\mathbf{X}}^*, \quad \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} = \tilde{\mathbf{Y}}\Sigma_r\mathbf{X}^*,$$

where $\hat{\Sigma}_r, \Sigma_r \in \mathbb{R}^{r \times r}$, $\mathbf{Y} \in \mathbb{C}^{q \times r}$, $\mathbf{X} \in \mathbb{C}^{k \times r}$, $\tilde{\mathbf{Y}} \in \mathbb{C}^{2q \times r}$, $\tilde{\mathbf{X}} \in \mathbb{C}^{r \times 2k}$.

² The pencil $(\mathbb{L}_s, \mathbb{L})$ is called regular if there is at least one value of $\lambda \in \mathbb{C}$ such that $\det(\mathbb{L}_s - \lambda\mathbb{L}) \neq 0$.

Remark 6.3. r can be chosen as the *numerical rank* (as opposed to the *exact rank*) of the Loewner pencil. For issues related to the rank, we refer the reader to [2], page 50, for details.

Theorem 6.3. *The quadruple $(\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C})$ of size $r \times r, r \times r, r \times m, p \times r$, given by*

$$\mathbf{E} = -\mathbf{Y}^T \mathbb{L} \mathbf{X}, \quad \mathbf{A} = -\mathbf{Y}^T \mathbb{L}_s \mathbf{X}, \quad \mathbf{B} = \mathbf{Y}^T \mathbf{V}, \quad \mathbf{C} = \mathbf{W} \mathbf{X},$$

is a descriptor realization of an (approximate) interpolant of the data with McMillan degree $r = \text{rank } \mathbb{L}$.

Remark 6.4.

- (a) The Loewner approach constructs a descriptor representation $(\mathbf{W}, \mathbb{L}, \mathbb{L}_s, \mathbf{V})$, of an underlying dynamical system exclusively from the data, with no further manipulations involved (i. e., matrix factorizations or inversions). In general, the pencil $(\mathbb{L}_s, \mathbb{L})$ is singular and needs to be projected to a regular pencil (\mathbf{A}, \mathbf{E}) . However, as shown in the *mass–spring–damper* example in equation (6.22), inversion can be replaced by generalized inversion.
- (b) As already mentioned, in the Loewner framework, by construction, \mathbf{D} terms are absorbed in the other matrices of the realization. Extracting the \mathbf{D} term involves an eigenvalue decomposition of $(\mathbb{L}_s, \mathbb{L})$.

6.2.5.3 Interpolation property of reduced systems

Given a Loewner quadruple and the projection matrices³ $\mathbf{X}, \mathbf{Y} \in \mathbb{C}^{n \times k}$, let the reduced quantities be

$$\hat{\mathbb{L}} = \mathbf{X}^* \mathbb{L} \mathbf{Y}, \quad \hat{\mathbb{L}}_s = \mathbf{X}^* \mathbb{L}_s \mathbf{Y}, \quad \hat{\mathbf{V}} = \mathbf{X}^* \mathbf{V}, \quad \hat{\mathbf{W}} = \mathbf{W} \mathbf{Y}.$$

We also consider the projected \mathbf{L} and \mathbf{R} matrices, namely $\hat{\mathbf{L}} = \mathbf{X}^* \mathbf{L}$, $\hat{\mathbf{R}} = \mathbf{R} \mathbf{Y}$. The question which arises is whether these reduced quantities satisfy interpolation conditions as well. The answer is affirmative and to show this we proceed as follows.

The associated $\hat{\mathbf{A}}$ and $\hat{\mathbf{M}}$ must satisfy the *projected equations* resulting from (6.18), i. e.

$$\hat{\mathbb{L}}_s - \hat{\mathbf{L}} \hat{\mathbf{A}} = \hat{\mathbf{V}} \hat{\mathbf{R}} \quad \text{and} \quad \hat{\mathbb{L}}_s - \hat{\mathbf{M}} \hat{\mathbf{L}} = \hat{\mathbf{L}} \hat{\mathbf{W}}. \quad (6.20)$$

Notice that the projected Loewner pencil is not in Loewner form. To achieve this we proceed as follows. We need to diagonalize $\hat{\mathbf{A}}$ and $\hat{\mathbf{M}}$. For this purpose we compute the following two generalized eigenvalue decompositions:

$$[\mathbf{D}_{\hat{\mathbf{A}}}, \mathbf{T}_{\hat{\mathbf{A}}}] = \text{eig}(\hat{\mathbb{L}}_s - \hat{\mathbf{V}} \hat{\mathbf{R}}, \hat{\mathbf{L}}) \quad \text{and} \quad [\mathbf{D}_{\hat{\mathbf{M}}}, \mathbf{T}_{\hat{\mathbf{M}}}] = \text{eig}(\hat{\mathbb{L}}_s - \hat{\mathbf{L}} \hat{\mathbf{W}}, \hat{\mathbf{L}}).$$

³ We call $\mathbf{X}, \mathbf{Y} \in \mathbb{C}^{n \times k}$ *projection matrices* as they are used for defining the *projector*: $\mathbf{X}(\mathbf{Y}^* \mathbf{X})^{-1} \mathbf{Y}^*$.

These decompositions imply

$$\hat{\Lambda} = \mathbf{T}_{\hat{\Lambda}} \mathbf{D}_{\hat{\Lambda}} \mathbf{T}_{\hat{\Lambda}}^{-1} \quad \text{and} \quad \hat{\mathbf{M}} = \mathbf{T}_{\hat{\mathbf{M}}} \mathbf{D}_{\hat{\mathbf{M}}} \mathbf{T}_{\hat{\mathbf{M}}}^{-1}, \quad (6.21)$$

where for simplicity, it is assumed that the matrices $\hat{\Lambda}$ and $\hat{\mathbf{M}}$ are diagonalizable.

It follows that the (diagonal) entries of $\mathbf{D}_{\hat{\Lambda}}$ and $\mathbf{D}_{\hat{\mathbf{M}}}$ are the right frequencies and the left frequencies of the reduced system, respectively. Furthermore, straightforward calculations imply that the remaining quantities are as follows:

$$\begin{cases} \bar{\mathbb{L}}_s = \mathbf{T}_{\hat{\mathbf{M}}}^{-1} \hat{\mathbb{L}}_s \mathbf{T}_{\hat{\Lambda}}, & \bar{\mathbb{L}} = \mathbf{T}_{\hat{\mathbf{M}}}^{-1} \hat{\mathbb{L}} \mathbf{T}_{\hat{\Lambda}}, \\ \bar{\mathbb{V}} = \mathbf{T}_{\hat{\mathbf{M}}}^{-1} \hat{\mathbb{V}}, & \bar{\mathbb{L}} = \mathbf{T}_{\hat{\mathbf{M}}}^{-1} \hat{\mathbb{L}}, \\ \bar{\mathbb{W}} = \hat{\mathbb{W}} \mathbf{T}_{\hat{\Lambda}}, & \bar{\mathbf{R}} = \hat{\mathbf{R}} \mathbf{T}_{\hat{\Lambda}}. \end{cases}$$

Conclusion: the right/left data triples for the reduced system are $(\mathbf{D}_{\hat{\Lambda}}, \bar{\mathbb{W}}, \bar{\mathbf{R}})$, and $(\mathbf{D}_{\hat{\mathbf{M}}}, \bar{\mathbb{V}}, \bar{\mathbb{L}})$, respectively, while the associated Loewner pencil is $(\bar{\mathbb{L}}_s, \bar{\mathbb{L}})$.

6.2.5.4 Real interpolants and reduced models

Most often the data are collected from real systems. In these cases if (s_i, ϕ_i) $s_i, \phi_i \in \mathbb{C}$, is a measurement pair, in order for the interpolants/reduced models to be real, the complex conjugate pair $(\bar{s}_i, \bar{\phi}_i)$, should also be included. Thus the left/right frequencies besides real quantities contain complex ones appearing in complex conjugate pairs. For instance, in the SISO (single-input single-output) case, let the real measurement frequencies be $\sigma_i \in \mathbb{R}$, and the complex ones $\hat{\sigma}_i + j \cdot \hat{\omega}_i$ where j denotes the imaginary unit. We split them in two sets, the left and the right ones, respectively, making sure that each set is closed under complex conjugation:

$$\begin{aligned} \mathbf{M} &= \{\sigma_i, i = 1, \dots, r_1; \hat{\sigma}_i \pm j \cdot \hat{\omega}_i, i = 1, \dots, r_3\}, \\ \mathbf{A} &= \{\sigma_i, i = r_1 + 1, \dots, r_1 + r_2; \hat{\sigma}_i \pm j \cdot \hat{\omega}_i, i = r_3 + 1, \dots, r_3 + r_4\}. \end{aligned}$$

Thus the left set has r_1 real frequencies and r_3 complex frequencies together with their complex conjugates (total $r_1 + 2r_3$ numbers). Similarly the numbers for the right set are r_2 and r_4 , i. e., it consists of $r_2 + 2r_4$ numbers. The quantities \mathbb{W} and \mathbb{V} are assembled in accordance with \mathbf{M} and \mathbf{A} . In addition let us define the matrices:

$$\begin{aligned} \mathbf{J}_\mu &= \text{blkdiag}[\overbrace{\mathbf{I}_{r_1}, \mathbf{J}, \dots, \mathbf{J}}^{r_3 \text{ terms}}] \in \mathbb{C}^{(r_1+2r_3) \times (r_1+2r_3)}, \\ \mathbf{J}_\lambda &= \text{blkdiag}[\overbrace{\mathbf{I}_{r_2}, \mathbf{J}, \dots, \mathbf{J}}^{r_4 \text{ terms}}] \in \mathbb{C}^{(r_2+2r_4) \times (r_2+2r_4)}, \end{aligned}$$

where $\mathbf{J} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -j \\ 1 & j \end{bmatrix}$, where $\text{blkdiag}[\cdot]$ (following Matlab notation) denotes the block diagonal structure. A simple calculation shows then that the matrices

$$\mathbf{M}_R = \mathbf{J}_\mu^* \mathbf{M} \mathbf{J}_\mu, \quad \mathbb{V}_R = \mathbf{J}_\mu^* \mathbb{V}, \quad \mathbf{L}_R = \mathbf{J}_\mu^* \mathbf{L},$$

have real entries. The same happens with the matrices

$$\mathbf{\Lambda}_R = \mathbf{J}_\lambda^* \mathbf{\Lambda} \mathbf{J}_\lambda, \quad \mathbf{W}_R = \mathbf{W} \mathbf{J}_\lambda, \quad \mathbf{R}_R = \mathbf{R} \mathbf{J}_\lambda.$$

Recall equations (6.18). If we now solve the transformed equations for $\mathbb{L}^R, \mathbb{L}_S^R$:

$$\mathbb{L}_S^R - \mathbb{L}^R \mathbf{\Lambda}_R = \mathbb{V}_R \mathbf{R}_R \quad \text{and} \quad \mathbb{L}_S^R - \mathbf{M}_R \mathbb{L}^R = \mathbf{L}_R \mathbb{W}_R,$$

the resulting pencil $(\mathbb{L}_S^R, \mathbb{L}^R)$ has real entries. Hence the algorithms based on \mathbb{L}^R and \mathbb{L}_S^R described below yield *real* reduced order models.

6.2.5.5 The Loewner algorithms for scalar and matrix rational approximation

Next, two algorithms (see Algorithms 6.1 and 6.2) for computing a strictly rational real interpolant for both, the scalar and the matrix interpolation problem are presented.

Algorithm 6.1: Loewner-SISO (Scalar rational approximation) [49].

Input: $\mathbf{S} = [s_1, \dots, s_N] \in \mathbb{C}^N, \mathbf{F} = [\phi_1, \dots, \phi_N] \in \mathbb{C}^N, N \in \mathbb{N}$.

Output: $\hat{\mathbf{E}} \in \mathbb{R}^{r \times r}, \hat{\mathbf{A}} \in \mathbb{R}^{r \times r}, \hat{\mathbf{B}} \in \mathbb{R}^{r \times 1}, \hat{\mathbf{C}} \in \mathbb{R}^{1 \times r}$ with $r \ll N$.

1. Partition the measurements into two disjoint sets and form left and right set as $(\mu_j, \mathbf{v}_j), j = 1, \dots, q$ and $(\lambda_i, \mathbf{w}_i), i = 1, \dots, k$.

$$\text{frequencies : } [s_1, \dots, s_N] \rightarrow [\lambda_1, \dots, \lambda_k], [\mu_1, \dots, \mu_q], \quad k + q = N,$$

$$\text{values : } [\phi_1, \dots, \phi_N] \rightarrow [\mathbf{w}_1, \dots, \mathbf{w}_k] = \mathbb{W}, [\mathbf{v}_1, \dots, \mathbf{v}_q] = \mathbb{V}^T.$$

2. Construct the **Loewner pencil** as

$$\mathbb{L} = \begin{pmatrix} \mathbf{v}_i - \mathbf{w}_j \\ \mu_i - \lambda_j \end{pmatrix}_{i=1, \dots, q}^{j=1, \dots, k}, \quad \mathbb{L}_S = \begin{pmatrix} \mu_i \mathbf{v}_i - \lambda_j \mathbf{w}_j \\ \mu_i - \lambda_j \end{pmatrix}_{i=1, \dots, q}^{j=1, \dots, k}.$$

3. It follows that the **complex raw model** is

$$\{\mathbb{W}, \mathbb{L}, \mathbb{L}_S, \mathbb{V}\}.$$

4. Transform all the complex data to real and there follows the **raw real model**:

$$\{\mathbb{W}_R, \mathbb{L}^R, \mathbb{L}_S^R, \mathbb{V}_R\}.$$

5. Compute the rank-revealing SVDs: $[\mathbf{Y}_1, \Sigma_1, \mathbf{X}_1] = \mathbf{SVD}([\mathbb{L}^R \mathbb{L}_S^R])$ and $[\mathbf{Y}_2, \Sigma_2, \mathbf{X}_2] = \mathbf{SVD}([\mathbb{L}_S^R; \mathbb{L}_S^R])$; the decay of the singular values, leads to the choice of the order r of the approximant.

6. The reduced real model is obtained by **projecting** the *raw real model* with $\mathbf{Y} = \mathbf{Y}_1^{n \times r}$ and $\mathbf{X} = \mathbf{X}_2^{n \times r}$ as

$$\underbrace{\{\mathbb{W}_R, \mathbb{L}^R, \mathbb{L}_S^R, \mathbb{V}_R\}}_{\text{singular}} \xrightarrow{\text{SVD}} \underbrace{\{\mathbb{W}_R \mathbf{X}, \mathbf{Y}^T \mathbb{L}^R \mathbf{X}, \mathbf{Y}^T \mathbb{L}_S^R \mathbf{X}, \mathbf{Y}^T \mathbb{V}_R\}}_{\text{regular}} = \{\hat{\mathbf{C}}, -\hat{\mathbf{E}}, -\hat{\mathbf{A}}, \hat{\mathbf{B}}\}.$$

7. A real approximant of the data is then

$$\hat{H}(s) = \hat{\mathbf{C}}(s\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1}\hat{\mathbf{B}} \approx \phi(s). \quad \square$$

Algorithm 6.2: Loewner-MIMO (Matrix rational approximation).

Input: $\mathbf{S} = [s_1, \dots, s_N] \in \mathbb{C}^N$, $\mathbf{F} = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_N] \in \mathbb{C}^{N \times p \times m}$, $N \in \mathbb{N}$.

Output: $\hat{\mathbf{E}} \in \mathbb{R}^{r \times r}$, $\hat{\mathbf{A}} \in \mathbb{R}^{r \times r}$, $\hat{\mathbf{B}} \in \mathbb{R}^{r \times m}$, $\hat{\mathbf{C}} \in \mathbb{R}^{p \times r}$ with $r \ll N$.

1. Partition the measurements into two disjoint sets:

Left data:

$$\mathbf{M} = \begin{bmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_q \end{bmatrix} \in \mathbb{C}^{q \times q}, \quad \mathbf{L} = \begin{bmatrix} \boldsymbol{\ell}_1^T \\ \vdots \\ \boldsymbol{\ell}_q^T \end{bmatrix} \in \mathbb{C}^{q \times p}, \quad \mathbf{V} = \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_q^T \end{bmatrix} \in \mathbb{C}^{q \times m}.$$

Right data:

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_k \end{bmatrix} \in \mathbb{C}^{k \times k}, \quad \mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_k] \in \mathbb{C}^{m \times k}, \\ \mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k] \in \mathbb{C}^{p \times k}.$$

2. Construct the **Loewner pencil** as

$$\mathbb{L} = \left(\frac{\mathbf{v}_i^T \mathbf{r}_i - \boldsymbol{\ell}_j^T \mathbf{w}_j}{\mu_i - \lambda_j} \right)_{i=1, \dots, q}^{j=1, \dots, k}, \quad \mathbb{L}_S = \left(\frac{\mu_i \mathbf{v}_i^T \mathbf{r}_i - \lambda_j \boldsymbol{\ell}_j^T \mathbf{w}_j}{\mu_i - \lambda_j} \right)_{i=1, \dots, q}^{j=1, \dots, k}.$$

3. It follows that the **complex raw model** is

$$\{\mathbb{W}, \mathbb{L}, \mathbb{L}_S, \mathbb{V}\}.$$

4. Transform all the complex data to real and there follows the **real raw model**:

$$\{\mathbb{W}_R, \mathbb{L}^R, \mathbb{L}_S^R, \mathbb{V}_R\}.$$

5. Compute the rank-revealing SVDs: $[\mathbf{Y}_1, \Sigma_1, \mathbf{X}_1] = \text{SVD}([\mathbb{L}^R; \mathbb{L}_S^R])$ and $[\mathbf{Y}_2, \Sigma_2, \mathbf{X}_2] = \text{SVD}([\mathbb{L}^R; \mathbb{L}_S^R])$; the decay of the singular values, lead to the choice of r .

6. The reduced real model is obtained by **projecting** the *raw real model* with $\mathbf{Y} = \mathbf{Y}_1^{n \times r}$ and $\mathbf{X} = \mathbf{X}_2^{n \times r}$.

$$\underbrace{\{\mathbb{W}_R, \mathbb{L}^R, \mathbb{L}_S^R, \mathbb{V}_R\}}_{\text{singular}} \xrightarrow{\text{SVD}} \underbrace{\{\mathbb{W}_R \mathbf{X}, \mathbf{Y}^T \mathbb{L}^R \mathbf{X}, \mathbf{Y}^T \mathbb{L}_S^R \mathbf{X}, \mathbf{Y}^T \mathbb{V}_R\}}_{\text{regular}} = \{\hat{\mathbf{C}}, -\hat{\mathbf{E}}, -\hat{\mathbf{A}}, \hat{\mathbf{B}}\}.$$

7. A real approximant of the data is

$$\hat{\mathbf{H}}(s) = \hat{\mathbf{C}}(s\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1}\hat{\mathbf{B}} \approx \boldsymbol{\phi}(s). \quad \square$$

6.2.6 Examples

In this section the theory will be illustrated by means of simple examples.

Example 6.1 (A spring–mass–damper system). Let m , d , and k denote the mass, damping, and stiffness of the spring as in Figure 6.1; let also $x(t)$ denote the displacement and $F(t)$ the force applied; the associated differential equation is

$$m\ddot{x}(t) + d\dot{x}(t) + kx(t) = F(t).$$

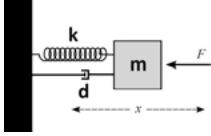


Figure 6.1: A spring–mass–damper system.

This is a SISO (single-input single-output) system. By introducing the state variables $x_1 = x, x_2 = \dot{x}$, the input $u = F$, and as output the velocity $y = \dot{x}$, the following state equations result:

$$\dot{x}_1(t) = x_2(t), \quad m\dot{x}_2(t) = -kx_1 - dx_2(t) + u(t), \quad y(t) = x_2(t).$$

The system matrices are thus

$$\mathbf{E} = \begin{bmatrix} 1 & 0 \\ 0 & m \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & 1 \\ -k & -d \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{C} = [0 \quad 1],$$

and the resulting transfer function is

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} = \frac{s}{ms^2 + ds + k}.$$

In the sequel we will assume for simplicity that all parameters have value one. We now wish to recover state equations equivalent to the ones above from measurements

of the transfer function. Toward this goal we evaluate the transfer function at the real frequencies: $\lambda_1 = \frac{1}{2}, \lambda_2 = 1$ (right frequencies), as well as $\mu_1 = -\frac{1}{2}, \mu_2 = -1$ (left frequencies). The corresponding values of \mathbf{H} are collected in the matrices

$$\mathbf{W} = \left(\begin{array}{cc} \frac{2}{7} & \frac{1}{3} \end{array} \right), \quad \mathbf{V} = \left(\begin{array}{cc} -\frac{2}{3} & -1 \end{array} \right)^T.$$

Furthermore, with $\mathbf{R} = [1 \ 1] = \mathbf{L}^T$, we construct the *Loewner pencil*:

$$\mathbb{L} = \left[\begin{array}{cc} \frac{20}{21} & \frac{2}{3} \\ \frac{6}{7} & \frac{2}{3} \end{array} \right], \quad \mathbb{L}_s = \left[\begin{array}{cc} -\frac{4}{21} & 0 \\ -\frac{4}{7} & -\frac{1}{3} \end{array} \right].$$

Since the pencil $(\mathbb{L}_s, \mathbb{L})$ is regular, we recover the original transfer function:

$$\mathbf{H}(s) = \mathbf{W}\Phi(s)^{-1}\mathbf{V} = \frac{s}{s^2 + s + 1}, \quad \text{where } \Phi(s) = \mathbb{L}_s - s\mathbb{L}.$$

Hence, the measurements above yield a minimal (descriptor) realization of the system in terms of the (state) variables $\mathbf{z}_1, \mathbf{z}_2$:

$$\begin{aligned} \frac{20}{21}\dot{\mathbf{z}}_1(t) + \frac{2}{3}\dot{\mathbf{z}}_2(t) &= \frac{4}{21}\mathbf{z}_1(t) + \frac{2}{3}\mathbf{u}(t), \\ \frac{6}{7}\dot{\mathbf{z}}_1(t) + \frac{2}{3}\dot{\mathbf{z}}_2(t) &= -\frac{4}{7}\mathbf{z}_1(t) - \frac{1}{3}\mathbf{z}_2(t) + \mathbf{u}(t), \quad \mathbf{y}(t) = \frac{2}{7}\mathbf{z}_1(t) + \frac{1}{3}\mathbf{z}_2(t), \end{aligned}$$

with

$$\tilde{\mathbf{E}} = \left[\begin{array}{cc} \frac{20}{21} & \frac{2}{3} \\ \frac{6}{7} & \frac{2}{3} \end{array} \right], \quad \tilde{\mathbf{A}} = \left[\begin{array}{cc} -\frac{4}{21} & 0 \\ -\frac{4}{7} & -\frac{1}{3} \end{array} \right], \quad \tilde{\mathbf{B}} = \left[\begin{array}{c} \frac{2}{3} \\ 1 \end{array} \right], \quad \tilde{\mathbf{C}} = \left[\begin{array}{cc} \frac{2}{7} & \frac{1}{3} \end{array} \right].$$

By multiplying with $\tilde{\mathbf{E}}^{-1}$, it yields (id: identified system in state-space form)

$$\tilde{\mathbf{A}}_{id} = \left[\begin{array}{cc} 4 & \frac{7}{2} \\ -6 & -5 \end{array} \right], \quad \tilde{\mathbf{B}}_{id} = \left[\begin{array}{c} -\frac{7}{2} \\ 6 \end{array} \right], \quad \tilde{\mathbf{C}}_{id} = \left[\begin{array}{cc} \frac{2}{7} & \frac{1}{3} \end{array} \right].$$

Coordinate transformation Let the state vector \mathbf{x} be transformed to the new state vector \mathbf{z} by the non-singular transformation matrix

$$\Psi = \left[\begin{array}{c} \mathbf{C} \\ \mathbf{CA} \end{array} \right]^{-1} \left[\begin{array}{c} \tilde{\mathbf{C}}_{id} \\ \tilde{\mathbf{C}}_{id}\tilde{\mathbf{A}}_{id} \end{array} \right],$$

of dimension 2×2 . Then the following hold:

$$\mathbf{z} = \Psi^{-1}\mathbf{x}, \quad \tilde{\mathbf{A}}_{id} = \Psi^{-1}\mathbf{A}\Psi, \quad \tilde{\mathbf{B}}_{id} = \Psi^{-1}\mathbf{B}, \quad \tilde{\mathbf{C}}_{id} = \mathbf{C}\Psi;$$

e. g., $\Psi\tilde{\mathbf{A}}_{id}\Psi^{-1} = \left[\begin{array}{cc} 0 & 1 \\ -1 & -1 \end{array} \right] = \mathbf{A}$.

Remark 6.5. The above result ensures that the Loewner framework constitutes a data-driven system identification method which constructs a realization only from measurements. It is important to mention that under a coordinate transformation, both systems, initial and identified are identical. At the same time, the underlying dynamics is recovered exactly while the corresponding revealing transfer function remains invariant under such a transformation.

The question now arises: what happens if we collect more data than necessary? Let us consider

$$\mathbf{A} = \text{diag} \left(\frac{1}{2} \quad 1 \quad \frac{3}{2} \quad 2 \right), \quad \mathbf{M} = \text{diag} \left(-\frac{1}{2} \quad -1 \quad -\frac{3}{2} \quad -2 \right).$$

In this case, the associated measurements are

$$\mathbf{W} = \left(\frac{2}{7} \quad \frac{1}{3} \quad \frac{6}{19} \quad \frac{2}{7} \right), \quad \mathbf{V} = \left(-\frac{2}{3} \quad -1 \quad -\frac{6}{7} \quad -\frac{2}{3} \right)^T,$$

and with $\mathbf{R} = [1 \ 1 \ 1 \ 1] = \mathbf{L}^T$, the Loewner pencil is

$$\mathbb{L} = \left[\begin{array}{cc|cc} \frac{20}{21} & \frac{2}{3} & \frac{28}{57} & \frac{8}{21} \\ \frac{6}{7} & \frac{2}{3} & \frac{10}{19} & \frac{3}{7} \\ \hline \frac{4}{7} & \frac{10}{21} & \frac{52}{133} & \frac{16}{49} \\ \frac{8}{21} & \frac{1}{3} & \frac{16}{57} & \frac{5}{21} \end{array} \right], \quad \mathbb{L}_s = \left[\begin{array}{cc|cc} -\frac{4}{21} & 0 & \frac{4}{57} & \frac{2}{21} \\ -\frac{4}{7} & -\frac{1}{3} & -\frac{4}{19} & -\frac{1}{7} \\ \hline -\frac{4}{7} & -\frac{8}{21} & -\frac{36}{133} & -\frac{10}{49} \\ -\frac{10}{21} & -\frac{1}{3} & -\frac{14}{57} & -\frac{4}{21} \end{array} \right].$$

It turns out that we can choose *arbitrary* matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{4 \times 2}$, provided that $\det(\mathbf{Y}^T \mathbf{X}) \neq 0$, e. g.

$$\mathbf{X} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ -2 & 1 \end{bmatrix}, \quad \mathbf{Y}^T = \begin{bmatrix} 0 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix},$$

so that the projected quantities

$$\widehat{\mathbf{W}} = \mathbf{W}\mathbf{X} = \left[-\frac{6}{7} \quad -\frac{1}{21} \right], \quad \widehat{\mathbb{L}} = \mathbf{Y}^T \mathbb{L}\mathbf{X} = \begin{bmatrix} -\frac{6}{7} & -\frac{1}{7} \\ \frac{18}{49} & \frac{1}{147} \end{bmatrix},$$

$$\widehat{\mathbb{L}}_s = \mathbf{Y}^T \mathbb{L}_s \mathbf{X} = \begin{bmatrix} 0 & \frac{1}{21} \\ -\frac{48}{49} & -\frac{19}{147} \end{bmatrix}, \quad \widehat{\mathbf{V}} = \mathbf{Y}^T \mathbf{V} = \begin{bmatrix} -\frac{1}{3} \\ \frac{11}{21} \end{bmatrix},$$

constitute a minimal realization of $\mathbf{H}(s)$:

$$\mathbf{H}(s) = \widehat{\mathbf{W}}(\widehat{\mathbb{L}}_s - s\widehat{\mathbb{L}})^{-1}\widehat{\mathbf{V}} = \frac{s}{s^2 + s + 1}.$$

It should be stressed that this holds for *arbitrary* projection matrices \mathbf{X}, \mathbf{Y} .

6.2.6.1 The generalized inverse approach

There is another way to express the above relationship avoiding arbitrary projectors. Basic ingredients are *generalized inverses*. This approach will be demonstrated only for the spring–mass–damper example. However, it holds in general (see, e. g., [3]).

In the sequel, we will make use (only) of the *Moore–Penrose* generalized inverse, which is defined as follows. Given the (rectangular) matrix $\mathbf{M} \in \mathbb{R}^{q \times k}$, the Moore–Penrose generalized inverse, denoted by $\mathbf{M}^{\text{MP}} \in \mathbb{R}^{k \times q}$, satisfies:

- (a) $\mathbf{M}\mathbf{M}^{\text{MP}}\mathbf{M} = \mathbf{M}$,
- (b) $\mathbf{M}^{\text{MP}}\mathbf{M}\mathbf{M}^{\text{MP}} = \mathbf{M}^{\text{MP}}$,
- (c) $[\mathbf{M}\mathbf{M}^{\text{MP}}]^T = \mathbf{M}\mathbf{M}^{\text{MP}}$,
- (d) $[\mathbf{M}^{\text{MP}}\mathbf{M}]^T = \mathbf{M}^{\text{MP}}\mathbf{M}$.

This generalized inverse always exists and is unique.

In the sequel we will be concerned with rectangular $q \times k$ *polynomial matrices* which have an explicit (rank-revealing) factorization as follows:

$$\mathbf{M} = \mathbf{X}\mathbf{\Delta}\mathbf{Y}^T,$$

where \mathbf{X} , $\mathbf{\Delta}$, \mathbf{Y} have dimension $q \times n$, $n \times n$, $k \times n$, $n \leq q, k$, and all have full rank n . In such cases, the **Moore–Penrose** generalized inverse is

$$\mathbf{M}^{\text{MP}} = \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{\Delta}^{-1}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T.$$

Mass–spring–damper example (continued). The quantity needed is the generalized inverse of

$$\mathbf{\Phi}(s) = \mathbb{L}_s - s\mathbb{L} = \begin{bmatrix} -\frac{20s}{21} - \frac{4}{21} & -\frac{2s}{3} & \frac{4}{57} - \frac{28s}{57} & \frac{2}{21} - \frac{8s}{21} \\ -\frac{6s}{7} - \frac{4}{7} & -\frac{2s}{3} - \frac{1}{3} & -\frac{10s}{19} - \frac{4}{19} & -\frac{3s}{7} - \frac{1}{7} \\ -\frac{4s}{7} - \frac{4}{7} & -\frac{10s}{21} - \frac{8}{21} & -\frac{52s}{133} - \frac{36}{133} & -\frac{16s}{49} - \frac{10}{49} \\ -\frac{8s}{21} - \frac{10}{21} & -\frac{s}{3} - \frac{1}{3} & -\frac{16s}{57} - \frac{14}{57} & -\frac{5s}{21} - \frac{4}{21} \end{bmatrix}. \quad (6.22)$$

We first notice that $\mathbf{\Phi}(s) = \mathbf{X}\mathbf{\Delta}(s)\mathbf{Y}^T$, where \mathbf{X} and \mathbf{Y} can be chosen as follows:

$$\mathbf{X} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\frac{3}{7} & \frac{8}{7} \\ -\frac{1}{2} & 1 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} 1 & 0 & -\frac{7}{19} & -\frac{1}{2} \\ 0 & 1 & \frac{24}{19} & \frac{9}{7} \end{bmatrix} \Rightarrow \det(\mathbf{YX}) \neq 0.$$

Thus by taking the 2×2 upper-left block as $\Delta(s) = \Phi(1 : 2, 1 : 2)(s)$, it follows that $\Phi(s)^{\text{MP}} = \frac{1}{80989667} \frac{1}{s^2+s+1} \mathbf{Z}(s)$, where

$$\mathbf{Z}(s) = \begin{bmatrix} -28(11610185s + 7274073) & 14(3558666s - 5604037) \\ 294(225182s + 281171) & (-147)(192415s - 19668) \\ 3724(54617s + 48189) & (-1862)(29046s - 17485) \\ 98(2527157s + 2123670) & -49(1250553s - 876439) \\ 6076(32301s - 391) & 14(15168851s + 1670036) \\ -2058(29494s + 15609) & -147(417597s + 261503) \\ -26068(5715s + 1523) & -1862(83663s + 30704) \\ -98(1797669s + 409322) & -49(3777710s + 1247231) \end{bmatrix}$$

In the *rectangular case*, where there are two right measurements less, i. e., we only have $\bar{\mathbf{L}} = \text{diag}[\frac{1}{2}, 1]$, while \mathbf{M} remains the same, the right values are $\bar{\mathbf{W}} = \mathbf{W}(:, 1 : 2)$; hence

$$\tilde{\Phi}(s) = \tilde{\mathbf{L}}_s - s\tilde{\mathbf{L}} = \begin{bmatrix} -\frac{20s}{21} - \frac{4}{21} & -\frac{2s}{3} \\ -\frac{6s}{7} - \frac{4}{7} & -\frac{2s}{3} - \frac{1}{3} \\ -\frac{4s}{7} - \frac{4}{7} & -\frac{10s}{21} - \frac{8}{21} \\ -\frac{8s}{21} - \frac{10}{21} & -\frac{s}{3} - \frac{1}{3} \end{bmatrix} = \mathbf{X}\Delta(s)\tilde{\mathbf{Y}}^T,$$

has dimension 4×2 , where $\tilde{\mathbf{Y}} = \mathbf{Y}(1 : 2, 1 : 2)$. In this case the Moore–Penrose inverse is

$$\tilde{\Phi}(s)^{\text{MP}} = \frac{1}{737(s^2 + s + 1)} \times \begin{bmatrix} -4767s - 3402 & \frac{1827}{2}s - \frac{2037}{2} & 3087s + 294 & 3297s + \frac{1365}{2} \\ 5838s + 5250 & -1596s + 903 & -4326s - 1218 & -4515s - 1722 \end{bmatrix},$$

which implies the desired equality

$$\Rightarrow \mathbf{W}\Phi(s)^{\text{MP}}\mathbf{V} = \bar{\mathbf{W}}\tilde{\Phi}(s)^{\text{MP}}\mathbf{V} = \mathbf{H}(s).$$

Conclusion: the Loewner framework allows the definition of rectangular and singular systems. \square

Example 6.2 (Reduction of a 10th order band-stop filter). The system has two inputs and two outputs (MIMO), state-space dimension 10, and a \mathbf{D} term of rank 2. A state-space representation is as follows:

$$\Sigma : \quad \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \quad \text{where}$$

$$\mathbf{A} = \left[\begin{array}{ccccc|ccccc} -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -1 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 0 & -1 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & -1 & 0 & 0 \\ -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & -1 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & -1 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array} \right], \quad \mathbf{B} = \left[\begin{array}{cc|cc} \frac{1}{2} & -\frac{1}{2} & & \\ \frac{1}{2} & -\frac{1}{2} & & \\ \frac{1}{2} & \frac{1}{2} & & \\ \frac{1}{2} & \frac{1}{2} & & \\ \frac{1}{2} & \frac{1}{2} & & \\ \hline 0 & 0 & & \\ 0 & 0 & & \\ 0 & 0 & & \\ 0 & 0 & & \\ 0 & 0 & & \end{array} \right],$$

$$\mathbf{C} = \left[\begin{array}{ccccc|ccccc} -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \end{array} \right], \quad \mathbf{D} = \left[\begin{array}{cc} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{array} \right].$$

The transfer function is a 2×2 rational matrix given by

$$\mathbf{H}(s) = \frac{1}{\mathbf{d}(s)} \begin{bmatrix} \mathbf{n}_1(s) & \mathbf{n}_2(s) \\ -\mathbf{n}_2(s) & -\mathbf{n}_1(s) \end{bmatrix} + \mathbf{D}, \quad \text{where}$$

$$\mathbf{n}_1(s) = s(s^8 + 7s^6 + 13s^4 + 7s^2 + 1),$$

$$\mathbf{n}_2(s) = s(5s^8 + 6s^7 + 25s^6 + 20s^5 + 41s^4 + 20s^3 + 25s^2 + 6s + 5),$$

$$\mathbf{d}(s) = 2(s^4 + s^3 + 3s^2 + 2s + 1)(2s^6 + 3s^5 + 7s^4 + 7s^3 + 7s^2 + 3s + 2).$$

It readily follows that $\lim_{s \rightarrow \infty} \mathbf{H}(s) = \mathbf{D}$. We take $N = 100$ samples of the transfer function on the imaginary axis (frequency response measurements) between 10^{-1} and 10^1 rad/sec. Figure 6.3 (left) shows the first 20 normalized singular values of the resulting real Loewner pencil (the rest are numerically zero). The rank of \mathbb{L} is 10 (the McMillan degree of the system) while the rank of \mathbb{L}_s is 12 (= rank \mathbb{L} + rank \mathbf{D}). The right pane in Figure 6.2 shows that we can obtain a perfect fit (total recovery of the model) with the Loewner framework for this MIMO example only by sampling the transfer function. As both Gramians are $\mathcal{P} = \mathcal{Q} = \frac{1}{2}\mathbf{I}_{10}$, i. e., they are equal and a multiple of the identity matrix, the Hankel singular values (see [2]) are all equal; this makes reduction with balanced truncation not feasible.

The right pane in Figure 6.3 shows the poles of the system obtained by means of the Loewner framework along with the zeros for every entry. The right pane in Figure 6.2 shows the band-stop character around frequency $\omega_0 = 1$ rad/s, of entries (1, 2) and (2, 1).

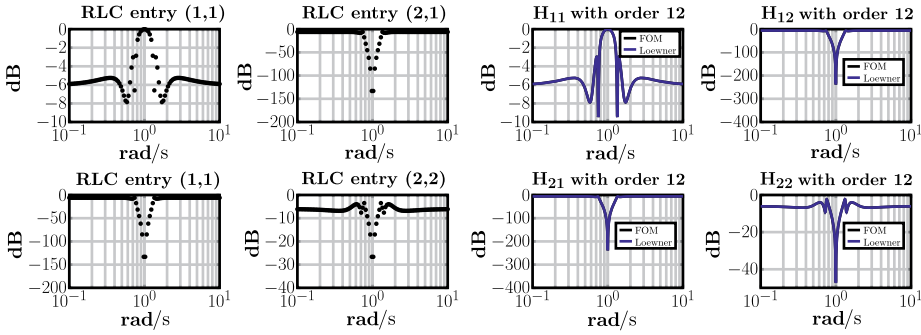


Figure 6.2: Left pane: Shows the 100 measurements sampled with DNS(Direct Numerical Simulations) of the theoretical (2 × 2)-matrix transfer function. Right pane: Loewner approximants.

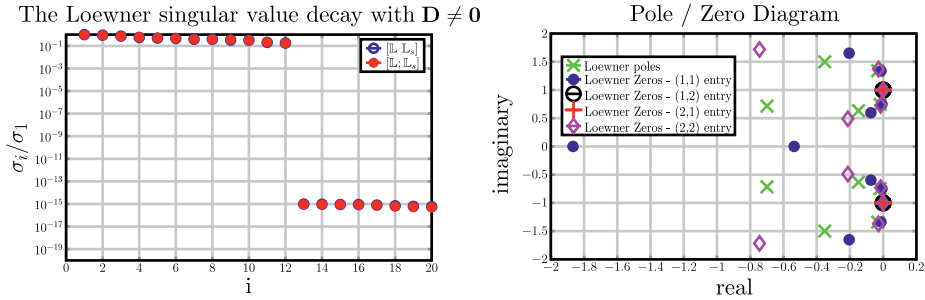


Figure 6.3: Left pane: Shows the first 12 singular values while the rest are numerically zero. Right pane: Pole/Zero diagram.

Computing the poles of the Loewner model confirms the accuracy of the approach. Consider the following:

eig(A)	eig(A _r , E _r)
-0.0181885913675508 - 0.745231200229 i	-0.0181885913675508 - 0.745231200229 i
-0.0181885913675508 + 0.745231200229 i	-0.0181885913675508 + 0.745231200229 i
-0.148402943598342 - 0.632502179219046 i	-0.148402943598342 - 0.632502179219046 i
-0.148402943598342 + 0.632502179219046 i	-0.148402943598342 + 0.632502179219046 i
-0.699080475814867 - 0.715042997542469 i	-0.699080475814867 - 0.715042997542469 i
-0.699080475814867 + 0.715042997542469 i	-0.699080475814867 + 0.715042997542469 i
-0.0327309328175858 - 1.34106659803138 i	-0.0327309328175858 - 1.34106659803138 i
-0.0327309328175858 + 1.34106659803138 i	-0.0327309328175858 + 1.34106659803138 i
-0.351597056401658 - 1.49852758300335 i	-0.351597056401658 - 1.49852758300335 i
-0.351597056401658 + 1.49852758300335 i	-0.351597056401658 + 1.49852758300335 i
	∞
	∞

As can be observed from this table, the Loewner method computes, besides the finite poles, two poles at infinity. This happens because in the Loewner framework the D-term is incorporated in the remaining matrices of the realization.

6.2.7 Summary

Recall Section 6.2.5. The following result summarizes the cases which arise in the Loewner framework, depending on the amount of data available.

Lemma 6.2. *Given is a scalar transfer function of McMillan degree n .*

1. *Amount of data less than $2n$. For $q = k \leq n$, define the transfer function $\hat{\mathbf{H}}(s) = \hat{\mathbf{C}}(s\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1}\hat{\mathbf{B}}$, by means of the Loewner procedure. The interpolation conditions below are satisfied:*

$$\hat{\mathbf{H}}(\mu_i) = \mathbf{H}(\mu_i) \quad \text{and} \quad \hat{\mathbf{H}}(\lambda_j) = \mathbf{H}(\lambda_j) \quad \text{for } i = 1, \dots, k.$$

If $k = q = n$, the Loewner quadruple is equivalent to the original one $(\mathbf{C}, \mathbf{E}, \mathbf{A}, \mathbf{B})$.

2. *Arbitrary amount of data, no reduction. For arbitrary k and q (i. e. $k, q \leq n$ or $k, q \geq n$) the Loewner quadruple interpolates the data, even if the pencil $(\mathbb{L}_s, \mathbb{L})$ is singular. This is to be interpreted as follows:*

$$(\mathbb{L}_s - \lambda_i \mathbb{L})\mathbf{e}_i = \mathbb{V} \quad \text{and} \quad \mathbf{e}_j^T (\mathbb{L}_s - \mu_j \mathbb{L}) = \mathbb{W}.$$

Hence $\mathbb{W}\mathbf{e}_i = \mathbf{w}_i$, $i = 1, \dots, k$, and $\mathbf{e}_j^T \mathbb{V} = \mathbf{v}_j$, $j = 1, \dots, q$. Therefore the transfer function of the Loewner pencil interpolates $\mathbf{H}(s)$ at the left and right interpolation points.

3. *Arbitrary amount of data, followed by reduction. If $k, q \geq n$, consider the rank-revealing SVD decompositions:*

$$[\mathbb{L} \quad \mathbb{L}_s] = \hat{\mathbf{Y}}_r \hat{\Sigma}_r \mathbf{X}_r^T \quad \text{and} \quad \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} = \mathbf{Y}_r \tilde{\Sigma}_r \tilde{\mathbf{X}}_r^T,$$

where $\mathbf{Y}_r \in \mathbb{R}^{q \times r}$, $\mathbf{X}_r \in \mathbb{R}^{k \times r}$, and $r \leq k, q$, is the exact or the numerical rank of the Loewner pencil involved. Let

$$\tilde{\mathbf{E}} = \mathbf{Y}_r^T \mathbb{L} \mathbf{X}_r, \quad \tilde{\mathbf{A}} = \mathbf{Y}_r^T \mathbb{L}_s \mathbf{X}_r \in \mathbb{C}^{r \times r}, \quad \tilde{\mathbf{B}} = \mathbf{Y}_r^T \mathbb{V} \in \mathbb{C}^r, \quad \tilde{\mathbf{C}} = \mathbb{W} \mathbf{X}_r \in \mathbb{C}^{1 \times r}.$$

Then the following approximate interpolation conditions are satisfied:

$$\tilde{\mathbf{H}}(\mu_i) \approx \mathbf{H}(\mu_i), \quad i = 1, \dots, q, \quad \text{and} \quad \tilde{\mathbf{H}}(\lambda_j) \approx \mathbf{H}(\lambda_j), \quad j = 1, \dots, k.$$

In addition, the reduced system satisfies (exact) interpolation conditions as shown in Section 6.2.5.3.

6.3 Practical considerations

This section deals with some key aspects of the Loewner framework, through a practical point of view.

- The factorization of the Loewner matrix into low-rank factor matrices:
 1. Through the singular value decomposition (SVD).
 2. Through the CUR decomposition.
- The choices involving the measurements used in the framework:
 1. Distribution of the interpolation points.
 2. Partition of the interpolation points.

6.3.1 The singular value decomposition

The SVD is arguably one of the most useful and commonly used tools in numerical linear algebra. It is listed as one of the main matrix decompositions and can be efficiently computed through various numerically stable algorithms. It is widely used for different high dimension reduction and approximation methods.

Any complex-valued matrix $\mathbf{A} \in \mathbb{C}^{n \times m}$ has a singular value decomposition given by $\mathbf{A} = \mathbf{Y}\mathbf{\Sigma}\mathbf{X}^*$ where $\mathbf{Y} \in \mathbb{C}^{n \times n}$, $\mathbf{X} \in \mathbb{C}^{m \times m}$ are unitary matrices, i. e., $\mathbf{Y}^*\mathbf{Y} = \mathbf{I}_n$ and $\mathbf{X}^*\mathbf{X} = \mathbf{I}_m$. The left and right singular vectors are collected as columns of matrices \mathbf{X} , and \mathbf{Y} , respectively.

Additionally, the matrix $\mathbf{\Sigma} \in \mathbb{C}^{n \times m}$ is defined as $\Sigma_{i,i} = \sigma_i$ and zero elsewhere. Here, the ordered non-negative scalars $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$ are the singular values (for $n \leq m$).

In what follows, it is assumed that matrix \mathbf{A} has low rank, i. e., $\text{rank}(\mathbf{A}) = r \leq n \leq m$. Let k be a positive integer so that $k < r$. The singular value decomposition of matrix \mathbf{A} can be additively split as follows:

$$\mathbf{A} = \mathbf{Y} \cdot \mathbf{\Sigma} \cdot \mathbf{X}^* = \underbrace{(\mathbf{Y}_k \quad \mathbf{Y}_{n-k})}_{n \times n} \cdot \underbrace{\begin{pmatrix} \mathbf{\Sigma}_k & \mathbf{0}_{k,m-k} \\ \mathbf{0}_{n-k,k} & \mathbf{\Sigma}_{n-k,m-k} \end{pmatrix}}_{n \times m} \cdot \underbrace{\begin{pmatrix} \mathbf{X}_k^* \\ \mathbf{X}_{m-k}^* \end{pmatrix}}_{m \times m} \quad (6.23)$$

$$= \underbrace{\mathbf{Y}_k \mathbf{\Sigma}_k \mathbf{X}_k^*}_{:= \mathbf{A}_k} + \mathbf{Y}_{n-k} \mathbf{\Sigma}_{n-k,m-k} \mathbf{X}_{m-k}^* \quad (6.24)$$

where $\mathbf{Y}_k \in \mathbb{C}^{n \times k}$, $\mathbf{\Sigma}_k \in \mathbb{C}^{k \times k}$ and $\mathbf{X}_k \in \mathbb{R}^{m \times k}$. Note that the matrix $\mathbf{A}_k := \mathbf{Y}_k \mathbf{\Sigma}_k \mathbf{X}_k^* \in \mathbb{C}^{m \times n}$ can be written in terms of the truncated dyadic decomposition, i. e., $\mathbf{A}_k = \sum_{i=1}^k \sigma_i y_i x_i^*$, where y_i and x_i are the i th column of matrices \mathbf{Y} , and \mathbf{X} , respectively.

A problem of interest is to approximate the original matrix \mathbf{A} with a rank k matrix \mathbf{T} , so that the approximation error is minimal with respect to the 2-induced norm or to the Frobenius norm.

From the Schmidt–Eckart–Young–Mirsky theorem (see Theorem 3.6 in [2]), it follows that (given $\sigma_k > \sigma_{k+1}$)

$$\min_{\mathbf{T} \in \mathbb{R}^{n \times m}, \text{rank}(\mathbf{T}) \leq k} \|\mathbf{A} - \mathbf{T}\|_2 = \sigma_{k+1}. \quad (6.25)$$

Moreover, it turns out that the unique solution to the minimization problem in (6.25) is given by $\mathbf{T} = \mathbf{A}_k$. If we replace the 2-induced norm with the Frobenius norm, it follows

that

$$\min_{\mathbf{T} \in \mathbb{R}^{n \times m}, \text{rank}(\mathbf{T}) \leq k} \|\mathbf{A} - \mathbf{T}\|_F = \left(\sum_{i=k+1}^n \sigma_i^2 \right)^{\frac{1}{2}}. \quad (6.26)$$

As before, the unique solution to the minimization problem in (6.26) is again given by $\mathbf{T} = \mathbf{A}_k$. For more details on the singular value decomposition (SVD) we refer the reader to [2], pages 31–41.

The advantage of the SVD is that it offers optimal low-rank solutions in both the 2-induced and the Frobenius norms. At the same time, one disadvantage is given by the fact that the method (full SVD) has cubic complexity with respect to $\min(m, n)$ (in the classical set-up when applied to dense matrices). Taking into account this possible downside, we seek ways of circumventing the usage of the classical SVD and investigate other matrix decompositions. It is worth mentioning that SVD complexity can be faster than cubic for a low-rank approximation with iterative algorithm. In the latter, a randomized version of SVD (r-SVD) will reveal this robust behavior.

6.3.2 The CUR decomposition

A challenging aspect of data-driven approximation methods is the choice of a relevant and meaningful low dimensional subset of the (usually large-scale) data set. In some cases, this subset can be used to preserve relevant characteristics of the dynamics for the model described by the original data. In this framework, it is of interest to devise procedures that can extract relevant information from large-scale sets of measurements. The end goal is to construct reduced order models suitable for tasks such as control, design, and simulation.

Nowadays, the dimension of data sets for various applications can easily reach $\approx \mathcal{O}(10^6)$. In such cases, computing the SVD of large and full matrices becomes prohibitive.

One appealing alternative is the so-called CUR decomposition. As before, the goal is to approximate the original matrix $\mathbf{A} \in \mathbb{C}^{n \times m}$, by a product of three low-rank matrices $\hat{\mathbf{A}} = \mathbf{C}\mathbf{U}\mathbf{R}$. Here, the columns of the matrix $\mathbf{C} \in \mathbb{C}^{n \times c}$ are represent a subset of the columns of \mathbf{A} while the rows of the matrix $\mathbf{R} \in \mathbb{C}^{r \times m}$ form a subset of the rows of \mathbf{A} . Finally, the matrix $\mathbf{U} \in \mathbb{C}^{c \times r}$ is constructed such that the factorization $\hat{\mathbf{A}} = \mathbf{C}\mathbf{U}\mathbf{R}$ holds.

In this new set-up, the left and right singular vectors appearing in the SVD are replaced by columns and rows of the initial matrix \mathbf{A} . Hence, the CUR factorization provides a way of identifying important sets of rows and columns of a matrix \mathbf{A} .

For more details on the CUR decomposition and some of its applications, we refer the reader to [19, 26, 27, 28, 47, 51, 54, 63].

The CUR factorization is hence an important tool for analyzing large-scale data sets which offers the following advantages over SVD:

1. If the matrix \mathbf{A} is sparse, then the matrices \mathbf{C} and \mathbf{R} are also sparse (unlike the matrices \mathbf{X} and \mathbf{Y} in the SVD approach).
2. The CUR factorization computes an approximation of \mathbf{A} in terms of some of the rows and some of the columns of \mathbf{A} . In contrast, the SVD computes approximants in terms of linear combinations of orthonormal bases generated by the rows and columns of \mathbf{A} .
3. Consider $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m > n$. The complexity for computing the full SVD of \mathbf{A} is $O(n^3)$ flops, using for instance the QR factorization, $O(mn^2)$ flops, using iterative methods as in ARPACK, and $O((m+n)k)$ flops per iteration, for approximate incremental methods where the k dominant singular triples are determined approximately (for details see [12]). On the other hand the CUR factorization of order k requires $O(k^3 + k^2(m+n))$ flops per iteration (for details see [47]).

6.3.2.1 CUR approximation of the Loewner matrix

In this section, we apply the CUR factorization to the Loewner matrix. We follow [47], where CUR is applied to Hankel matrices instead.

Definition 6.3. With $\mathbb{L} \in \mathbb{R}^{n \times n}$, let $\mathcal{I} = \{i_1, \dots, i_r\}$ and $\mathcal{J} = \{j_1, \dots, j_r\}$ denote the r -subsets ($r \ll n$) of row and column indices, respectively. If $(\cdot)^{\text{MP}}$ denotes the pseudo inverse, then the CUR factorization of the Loewner matrix \mathbb{L} is given by

$$\mathbb{L}_r := \underbrace{\mathbb{L}(:, \mathcal{J})}_{\mathcal{J}\text{-columns}} \cdot \mathbb{L}(\mathcal{I}, \mathcal{J})^{\text{MP}} \cdot \underbrace{\mathbb{L}(\mathcal{I}, :)}_{\mathcal{I}\text{-rows}}. \quad (6.27)$$

In practical applications, large-scale data matrices are only approximately of low rank (when data can be for instance corrupted by noise). In this case, the sets \mathcal{I} and \mathcal{J} need to be chosen in such a way that the approximation error $\|\mathbb{L} - \mathbb{L}_r\|$ is small. Many approaches for selecting the sets of rows and columns have been proposed. In the following we mention only some of them.

1. Selection based on a maximum volume sub-matrix in [54].
2. Selection based on minimizing the approximation error in the Chebyshev norm (“skeleton” approximation) in [26, 27].
3. Procedure based on the “cross-approximation” algorithm in [55].
4. Selection based on a discrete empirical interpolation method (DEIM) approach in [63].

6.3.2.2 The Loewner CUR algorithm

We introduce a data-driven approximation algorithm for the SISO case based on CUR approach. This constructs a reduced order model by means of an adaptive selection of the rows and columns via the cross-approximation algorithm in [55]. The steps of the procedure are included in Algorithm 6.3.

Algorithm 6.3: Loewner CUR-cross-approximation based – SISO [44].

Input: $\mathbf{S} = [s_1, \dots, s_N] \in \mathbb{C}^N$, $\mathbf{F} = [\phi_1, \dots, \phi_N] \in \mathbb{C}^N$ with $N, r \in \mathbb{N}$, and tolerance values δ, ϵ .

Output: $\hat{\mathbf{E}} \in \mathbb{R}^{r \times r}$, $\hat{\mathbf{A}} \in \mathbb{R}^{r \times r}$, $\hat{\mathbf{B}} \in \mathbb{R}^{r \times 1}$, $\hat{\mathbf{C}} \in \mathbb{R}^{1 \times r}$ with $r \ll N$.

1. Form the left and right sets as (μ_j, \mathbf{v}_j) , $j = 1, \dots, q$ and $(\lambda_i, \mathbf{w}_i)$, $i = 1, \dots, k$
2. Form the Loewner matrices \mathbb{L} and \mathbb{L}_s as in Algorithm 6.1 and step 2.
3. Transform all the complex data to real as explained in Section 6.2.5.4.
4. $\mathcal{J}_0 = [j_1, \dots, j_r] \subset \mathcal{J}_n$ an initial set of column indices.
5. $[\mathcal{I}_r, \sim, \sim] = \text{crossapprox}([\mathbb{L} \ \mathbb{L}_s], \mathcal{J}_0, \delta, \epsilon)$.
6. $[\sim, \mathcal{J}_r, \sim] = \text{crossapprox}([\frac{\mathbb{L}}{\mathbb{L}_s}], \mathcal{I}_r, \delta, \epsilon)$.
7. $\hat{\mathbf{E}} = -\mathbb{L}(\mathcal{I}_r, \mathcal{J}_r)$, $\hat{\mathbf{A}} = -\mathbb{L}_s(\mathcal{I}_r, \mathcal{J}_r)$, $\hat{\mathbf{B}} = \mathbb{V}(\mathcal{I}_r)$, $\hat{\mathbf{C}} = \mathbb{W}(\mathcal{J}_r)$.
8. The rational approximant is given by

$$\mathbf{H}_r(s) = \hat{\mathbf{C}}(s\hat{\mathbf{E}} - \hat{\mathbf{A}})^{-1}\hat{\mathbf{B}}. \quad \square$$

For the practical implementation of the function “*crossapprox*”, used in steps 5 and 6 of the above algorithm, we refer the reader to Algorithm 1 in [47], or to the original reference [55].

Remark 6.6. Instead of using the cross-approximation algorithm, one can use the DEIM (Discrete Empirical Interpolation Method) algorithm from [63]. Hence, steps 5 and 6 in Algorithm 6.3 need to be modified accordingly. As a result, singular value decompositions are performed in order to construct left and right singular vector matrices (for which the DEIM procedure is applied to). In order to avoid the SVD, an incremental QR factorization can be instead used, as proposed in [63].

Remark 6.7. The CUR factorization directly reveals the dominant rows/columns of the data, while the SVD does not. More precisely, the leading singular vectors give only linear combinations of the underlying features. Whereas, with the CUR one gets an actual subset of the initial features (columns) together with the corresponding rows. Consequently, a first benefit of the CUR is that it preserves the physical meaning and structure of the initial data. Additionally, another advantage is that the sparsity is preserved.

6.3.3 Choice of left and right interpolation points

This section deals with the problem of selecting the initial interpolation points in the Loewner framework. More specifically, we investigate how the choice of the initial interpolation points affects the quality of the reduced order model. We take into consideration different point distributions in 1D or in 2D.

Moreover, several splitting techniques are analyzed. These are related to the partition of the data set into two disjoint subsets, which is performed in the beginning of the algorithms in the Loewner framework.

6.3.3.1 Distribution of the interpolation points

We present various distributions of the initial interpolation points for the one-dimensional case (1D) as well as for the two-dimensional case (2D).

1D interpolation grid	2D interpolation grid
equispaced;	equispaced (“same areas”);
logarithmic spaced;	logarithmic spaced;
Chebyshev nodes;	Padua points;
Uniformly random	Uniformly random

In Figure 6.4, we depict different distributions of initial interpolation points. One way of selecting points is that of equispaced or linearly spaced points, commonly used for Fourier analysis. This represents a natural choice because of the usage of trigonometric periodic functions.

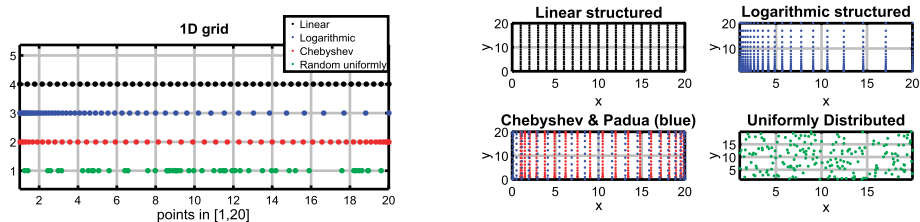


Figure 6.4: A visual representation of different interpolation grids.

In some practical applications, under the assumption that the energy decreases exponentially as time or frequency approach infinity (on an unbounded domain), the choice of logarithmic distributed points is more appropriate.

Naturally, a dense sampling grid can be used in the beginning of the experiment (e. g., for a lower frequency range or for small time instances). The motivation for this approach stems from the assumption that the meaningful quantities (with high energy or with relevant oscillations) appear in the beginning, hence requiring more samples. Afterwards, a more sparse distribution grid of points can be instead chosen as the energy level decays (or as relevant oscillations decay in time).

Additionally, the choice of Chebyshev-type points is motivated by their usage in polynomial-based interpolation on bounded domains due to for example, the elimination of the Runge phenomenon⁴ (high degree polynomials are generally unsuitable for interpolation with equispaced points).

Finally, randomly distributed sampling points often appear in stochastic experiments that are characterized by randomness.

6.3.3.2 Partition of the data points and values

Data splitting is one of the first steps in the classical Loewner algorithm (presented in Section 6.2). In this section, we mention various splitting schemes and how they affect the Loewner matrix singular value decay and also the approximation quality of the Loewner interpolants.

The data set ($n = \text{even}$) is composed of

$$\begin{cases} \text{Sample points : } \mathbf{S} = [\omega_1, \omega_2, \dots, \omega_n] \in \mathbb{R}^n, & \text{with } \omega_1 < \omega_2 < \dots < \omega_n, \\ \text{Sample values : } \mathbf{H} = [H(\omega_1), H(\omega_2), \dots, H(\omega_n)] \in \mathbb{C}^n. \end{cases} \quad (6.28)$$

We analyze four different types of data splitting that are mentioned in the following.

1. **First type:** disjoint splitting.
 - $\boldsymbol{\mu} = [\omega_1, \dots, \omega_{n/2}]$ and $\mathbb{V} = [H(\omega_1), \dots, H(\omega_{n/2})]$,
 - $\boldsymbol{\lambda} = [\omega_{n/2+1}, \dots, \omega_n]$ and $\mathbb{W} = [H(\omega_{n/2+1}), \dots, H(\omega_n)]$.
2. **Second type:** alternate splitting.
 - $\boldsymbol{\mu} = [\omega_1, \omega_3, \dots, \omega_{n-1}]$ and $\mathbb{V} = [H(\omega_1), H(\omega_3), \dots, H(\omega_{n-1})]$,
 - $\boldsymbol{\lambda} = [\omega_2, \omega_4, \dots, \omega_n]$ and $\mathbb{W} = [H(\omega_2), H(\omega_4), \dots, H(\omega_n)]$.
3. **Third type:** magnitude splitting (in this case the set \mathbf{S} is first sorted with respect to the magnitude of the set \mathbf{H}).
 - $\boldsymbol{\mu} = [\omega_1, \dots, \omega_{n/2}]$ and $\mathbb{V} = [H(\omega_1), \dots, H(\omega_{n/2})]$,
 - $\boldsymbol{\lambda} = [\omega_{n/2+1}, \dots, \omega_n]$ and $\mathbb{W} = [H(\omega_{n/2+1}), \dots, H(\omega_n)]$.
4. **Fourth type:** magnitude alternate splitting (in this case, the set \mathbf{S} is first sorted with respect to the magnitude of the set \mathbf{H} and then alternating splitting is applied).
 - $\boldsymbol{\mu} = [\omega_1, \omega_3, \dots, \omega_{n-1}]$ and $\mathbb{V} = [H(\omega_1), H(\omega_3), \dots, H(\omega_{n-1})]$,
 - $\boldsymbol{\lambda} = [\omega_2, \omega_4, \dots, \omega_n]$ and $\mathbb{W} = [H(\omega_2), H(\omega_4), \dots, H(\omega_n)]$.

As observed in practice, when splitting the data as for the first type, the Loewner matrix has a very fast decay of the singular values. Moreover, in this case, the computed reduced models usually provide low approximation quality.

⁴ Runge's phenomenon is a problem of oscillation at the edges of an interval that occurs when using polynomial interpolation with polynomials of high degree over a set of equispaced interpolation points.

On the other hand, for the second separation type (alternate splitting), the left and right sets of sample points can be chosen ϵ -close to one another (element-wise). Hence, as $\epsilon \rightarrow 0$, Hermitian interpolation conditions are enforced (which involve matching the first derivative at those points).

Other observations that hold in the case of second type splitting are that the numerical rank of the Loewner matrix is usually larger than that of the Loewner matrix constructed based on the first type. Additionally, for the second type, the condition number is smaller than that computed for the first type. For the above-mentioned cases, bounds on the singular value decay of the Loewner matrix are provided in [15].

6.4 Case studies

In this section we illustrate the concepts developed in the preceding sections by means of examples. In particular the following seven examples will be analyzed.

1. The benchmark CD player ($n = 120$).
2. The function $f(x) = \exp(-x) \sin(10x)$, $x \in [-1, 1]$.
3. The inverse of the Bessel function of the first kind, in $[0, 10] \times [-1, 1]j$.
4. An Euler–Bernouli beam.
5. A heat equation with transfer function $\mathbf{H}(s) = \exp(-\sqrt{s})$, $s \in [0.01, 100]j$.
6. Approximation of $f = y/\sinh(y)$, $y(x) = 100\pi(x^2 - 0.36)$, $x \in [-1, 1]$.
7. The sign function in the interval $[-b, -a]$ and $[a, b]$, $a > b > 0$.

6.4.1 The CD player

Consider the CD player benchmark example which is a MIMO dynamical system of dimension 120 with 2 inputs and 2 outputs. Here we will consider the (2, 1) sub-system, i. e. the SISO system from the first input to the second output.

The goal is to approximate the transfer function in the Loewner framework. We start by considering 400 interpolation points $\pm j\omega_i$, $i = 1, \dots, 200$, where ω_i are logarithmically spaced in the interval $\Omega = [10^{-1}, 10^5]$. Thus $\Omega = \{\omega_1, \omega_2, \dots, \omega_{200}\}$, where $\omega_i < \omega_{i+1}$, for all i . We now define the left/right interpolation points in four different ways as explained in section 6.3.3.2 and depicted in Figure 6.5 (up).

As can be seen in Figure 6.5 (down), the decay of the Loewner matrix singular values is faster for “half-half” (disjoint) splitting than for “alternating” splitting.

The next step is to choose the truncation order and to determine the level of approximation. We propose two different ways for this purpose.

1. By choosing equal truncation orders r .
2. By choosing for each separation the maximum truncation order so that $\frac{\sigma_r}{\sigma_1}$, is equal to a fixed tolerance value.

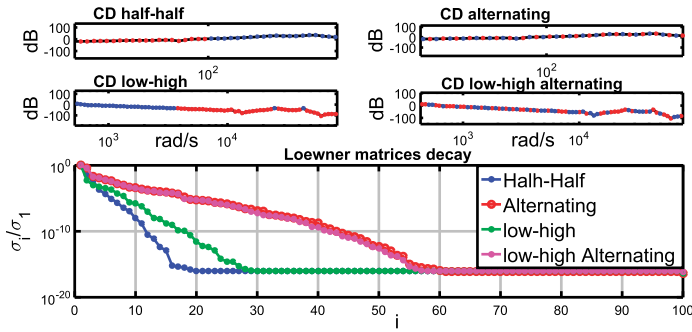


Figure 6.5: The four different splitting schemes (up) and the decay of the singular values ($\frac{\sigma_i}{\sigma_1}, i = 1, \dots, 100$) of the Loewner matrix for each type (down).

First experiment: equal truncation orders

Here, we fix the truncation order to $r = 10$, and compute $\frac{\sigma_r}{\sigma_1}$. The results are presented in Table 6.2.

Table 6.2: Normalized singular values corresponding to $r = 10$ for each splitting.

Case	1st	2nd	3rd	4th
r	10	10	10	10
$\frac{\sigma_r}{\sigma_1}$	$1e - 8$	$1e - 6$	$1e - 4$	$1e - 4$

The frequency response of the original system with those of the four reduced systems (corresponding to each different splitting) is shown in Figure 6.6. Note that all methods produce similar approximation quality.

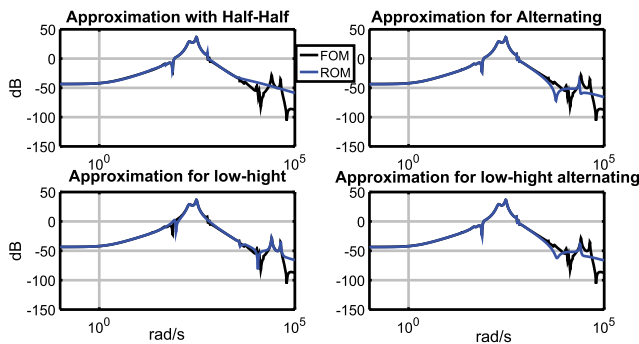


Figure 6.6: Frequency response comparison: original system vs. the reduced ones with equal truncation orders ($r = 10$).

Next, the approximation error for each reduced systems is depicted in Figure 6.7. For the first partition type, the error curve displays a ‘V’ shape form near the middle of the sampling interval. This is where the left and right sampling points are very close to each other.

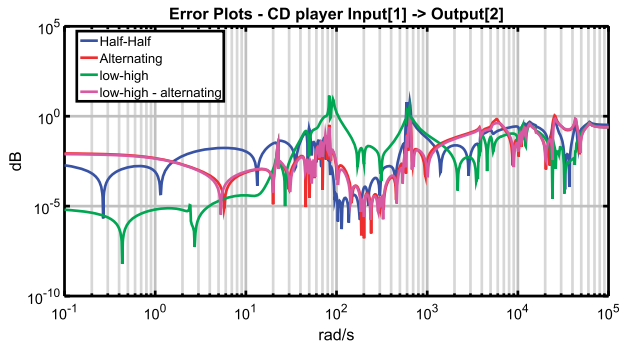


Figure 6.7: Approximation error with the four splitting schemes.

Second experiment: reaching machine precision⁵

The tolerance of normalized singular value $\frac{\sigma_r}{\sigma_1}$ is now fixed (e. g. 10^{-14}). This implies the truncation order r . The results are presented in Table 6.3

Table 6.3: Different truncation orders for all splitting schemes and for a fixed tolerance.

Case	1st	2nd	3rd	4th
r	16	51	23	48
$\frac{\sigma_r}{\sigma_1}$	$1e-14$	$1e-14$	$1e-14$	$1e-14$

The truncation order for the first splitting type is more than three times smaller than that for the second splitting type (16 vs 51).

The frequency response of the original systems with the four reduced systems in depicted in Figure 6.8. All methods produce good approximation quality, with a slight deviation in the high frequency range observed for the first splitting type.

Finally, Figure 6.9 shows the approximation error for each reduced system.

Notice that the blue curve in Figure 6.9 has a ‘V’ shape in the middle of the sampling interval. The lowest approximation error is recorded for the second splitting type (alternate selection).

⁵ Machine precision is the smallest number ϵ such that the difference between 1 and $1 + \epsilon$ is nonzero. This is approximately 10^{-16} .

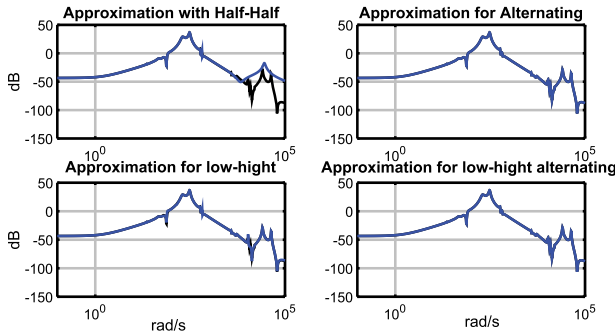


Figure 6.8: Frequency response comparison: original system vs. the reduced ones by reaching machine precision.

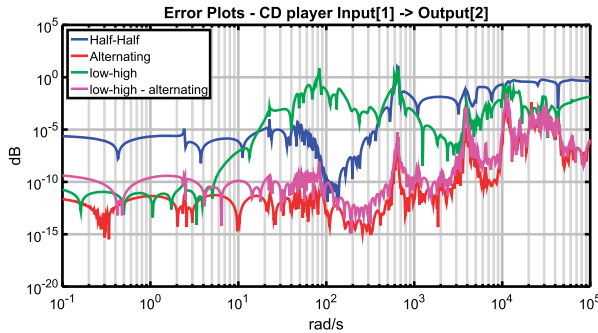


Figure 6.9: Approximation error for the four splitting schemes.

6.4.2 Approximation of an oscillating function

We collect $N = 4,000$ measurements $\{(s_k, \phi_k) : k = 1, \dots, N\}$ of the following function:

$$\phi(x) = e^{-x} \sin(10x), \quad x \in [-1, 1]. \quad (6.29)$$

Assume that the interpolation points $\mathbf{s} = [s_1, s_2, \dots, s_{4000}] \subset [-1, 1]$ are *equispaced*; next we remain with two types of splitting.

1. **First type:** disjoint splitting.
 - Left: $\boldsymbol{\mu} = [s_1, s_2, \dots, s_{2000}] \subset [-1, 0]$
 - Right: $\boldsymbol{\lambda} = [s_{2001}, s_{2002}, \dots, s_{4000}] \subset [0, 1]$

We construct the Loewner pencil and the underlying rank is 11.

2. **Second type:** alternate splitting.
 - Left: $\boldsymbol{\mu} = [s_1, s_3, \dots, s_{3999}] \subset [-1, 1]$
 - Right: $\boldsymbol{\lambda} = [s_2, s_4, \dots, s_{4000}] \subset [-1, 1]$.

We construct the Loewner pencil and the underlying rank is 15.

Figure 6.10 shows the entries of the Loewner matrix in logarithmic scale for the two ways of sampling point separation. Next, the interpolation data is compressed, making use of the following methods: (a) the singular value decomposition SVD, (b) the randomized version rSVD, (c) CUR, implemented with DEIM and (d) CUR implemented with cross approximation. The parameters for the latter two methods are: $\epsilon = 0.001$ and $\delta = 0.01$.

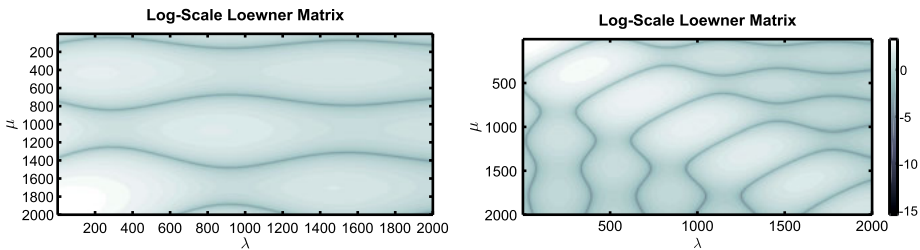


Figure 6.10: Entries of the Loewner matrix for the first splitting (left) and the second splitting (right).

Table 6.4: Results for the first splitting type (disjoint) with an i5-CPU 2.60 GHz.

Reduction – r for $\approx \mathbb{L}$	$\text{rank}(\mathbb{L}_{r \times r})$	$\text{cond}(\mathbb{L}_{r \times r}) = \frac{\sigma_{\max}}{\sigma_{\min}}$	Error $\ \cdot\ _F$	Time (s)
SVD	11	$9.7313e + 10$	$6.7367e - 10$	4.166029
CUR-CrossApprox	11	$7.6582e + 10$	$1.5621e - 09$	0.528352
CUR-DEIM	11	$1.3898e + 11$	$2.2283e - 09$	4.101303
randomized SVD	11	$9.7314e + 10$	$1.1281e - 10$	0.030148

In Figure 6.11 the error curves for the first splitting are shown. The red Xs indicate the selected points with CUR-cross-approximation method while the green crosses (+) indicate the selected points with CUR-DEIM method. In Figure 6.12 the error curves for the second splitting are shown. As opposed to the previously shown results (in Figure 6.11), the error in this case (Figure 6.12) is distributed more uniformly. Additional qualitative measures (e.g., the condition number) under different splitting schemes with the same reduced-order are presented in Tables 6.4 and 6.5.

Table 6.5: Results for the second splitting type (alternate) with an i5-CPU 2.60 GHz.

Reduction – r for $\approx \mathbb{L}$	$\text{rank}(\mathbb{L}_{r \times r})$	$\text{cond}(\mathbb{L}_{r \times r}) = \frac{\sigma_{\max}}{\sigma_{\min}}$	Error $\ \cdot\ _F$	Time (s)
SVD	11	$8.8199e + 4$	0.0020	4.261075
CUR-CrossApprox	11	$1.0228e + 5$	0.0062	0.563411
CUR-DEIM	11	$9.3343e + 4$	0.0245	4.152420
randomized SVD	11	$8.8199e + 4$	0.0020	0.024586

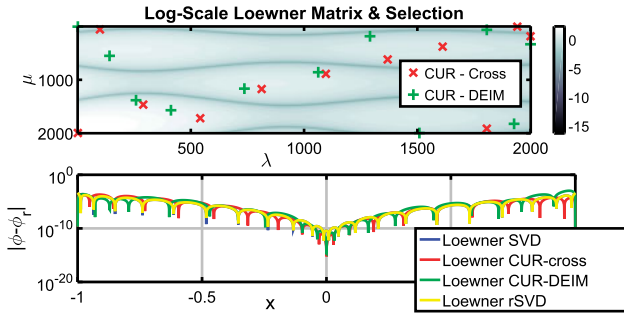


Figure 6.11: Selected points and approximation error for the disjoint splitting.

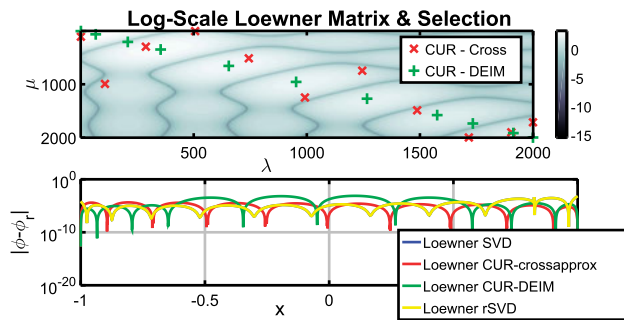


Figure 6.12: Selected points and approximation error for the alternate splitting.

As seen in the above experiments, the splitting of the data influences both the Loewner singular value decay and the quality of approximation. In most of the experiments that follow, we choose the *alternate* way of splitting the data.

6.4.3 Approximation of a Bessel function

In this section we investigate the approximation of the inverse of a Bessel function in a domain in the complex plane. If this function is considered to be the transfer function of a dynamical system, this system is infinite dimensional; furthermore it is not stable as there are poles in the right-half of the complex plane.

In particular we consider the inverse of the Bessel function of the first kind and order $n \in \mathbb{N}$. It is defined by the following contour integral:

$$J_n(s) = \frac{1}{2\pi i} \oint e^{(\frac{s}{2})(t-\frac{1}{t})} t^{-n-1} dt. \tag{6.30}$$

Here, we consider only the case $n = 0$. Our aim is to approximate $H(s) = \frac{1}{J_0(s)}$, $s \in \mathbb{C}$, inside the rectangle $\Omega = [0, 10] \times [-1, 1] \subset \mathbb{C}$. In Figure 6.13 (left pane) the function $H(s)$

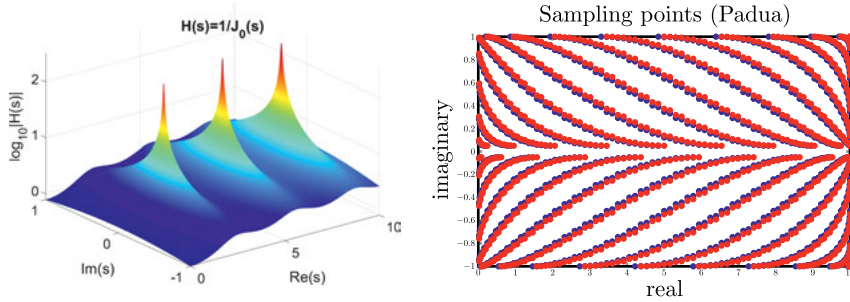


Figure 6.13: Left pane: The inverse of the Bessel function of the 1st kind. Right pane: A subset of 10,000 Padua point grid over $\Omega = [0, 10] \times [-1, 1]$ domain are shown.

is shown in the domain Ω . The three spikes correspond to the unstable poles of the underlying system. These are three of the zeros of the Bessel function. Here we construct approximants $H_r(s)$, of order r , of $H(s)$, using the interpolation points as shown in Figure 6.13 on the right pane. The distribution of the two-dimensional initial grids is 5,000 Padua points with the conjugates. This grid is used to reduce the Runge phenomenon. For more details in approximation theory (i. e. Runge phenomenon, Padua points, barycentric interpolation, etc.), we refer the reader to [64]. In [43, 44], the same experiment with other types of grids (random uniformly, structured) is presented.

In the Loewner framework, the singular value decomposition (SVD) plays a key role. This factorization allows us to extract the numerical order of the rational model which approximates the original non-rational one.

In Figure 6.14 (left pane), we show the distribution of the normalized singular values $\frac{\sigma_j}{\sigma_1}$, $j = 1, \dots, N$, of the augmented matrices $[\mathbb{L} \ \mathbb{L}_s]$ and $[\mathbb{L} \ \mathbb{L}_s]$.

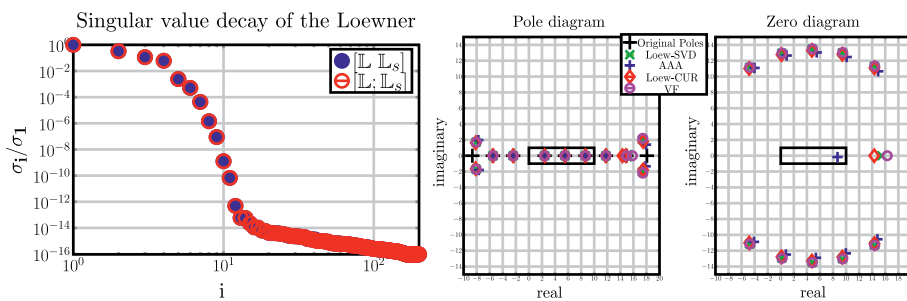


Figure 6.14: Left pane: Singular value decay of 10,000 values. Right pane: Pole/zero diagram with the three original poles (zeros of Bessel) which recovered with 15 digits accuracy.

By taking measurements as in Figure 6.13 (right pane) the decay of the singular values Figure 6.14 – left pane, leads to a reduced order $r = 12$ with $\frac{\sigma_{12}}{\sigma_1} = 4.887 \cdot 10^{-13}$. In Figure 6.14 on the right pane the pole/zero diagram is presented which includes the

results from all methods. Methods VF, and Loewner(SVD or CUR) construct real strictly rational models with degree $(11, 12)^6$ with $D = 0$, as opposed to AAA algorithm which constructs complex proper rational model of degree $(12, 12)$ with a non-zero D term.

By using the methods LoewCUR-cross and AAA, points from the sampling grid are selected. Applying the LoewSVD method the point selection is obtained by compressing the initial grid. This can be achieved by using the first r columns (r : singular vectors) of the singular matrices as projection matrices and by solving two $(r \times r$ -dimension) generalized eigenvalue problems as explained in Section 6.2.5.3. Under this way, we compress the original grid with $N = 10,000$ points into a much smaller set of only $2r = 24$ points which are exact interpolation points for the approximant. As it turns out, the projected points lie in the domain Ω ; see also left pane in Figure 6.15.

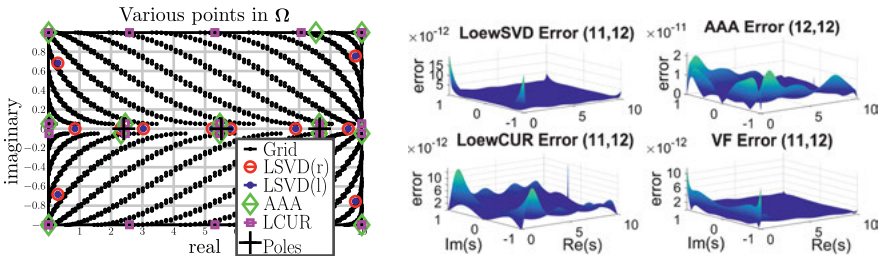


Figure 6.15: Left pane: Support and compressed points for every method over Ω domain with LSVD(r) \rightarrow LoewSVD projected right points, LSVD(l) \rightarrow LoewSVD projected left points. Right pane: The error for every method.

The LoewCUR-cross and AAA methods select points among the initial interpolation points but with different criteria. The AAA algorithm selects support points by minimizing the mean squared error with the rest of the measurements while LoewCUR uses cross approximation, which maximizes the absolute value of the determinant (maximum volume) of the sub-matrix of dimension $(r \times r)$.

In Figure 6.15 on the right pane, the error for each method is shown. The normalized error is computed as $\frac{|H(s) - H_r(s)|}{|H(s)|}$ with 25,000 evaluation points in Ω . It should be mentioned that the above special choice of the original interpolation grid as Padua points, indeed reduced the Runge phenomenon.

Next we wish to visualize the approximation error outside Ω . Towards this goal we chose 25,000 equispaced evaluation points inside the domain $[-3, 13] \times [-3, 3]$. Results with log-contour level error of increasing order $10^{-16}, \dots, 10^{-4}$ are presented in Figure 6.16.

6 The notation (m, n) indicates that the order of the numerator polynomial is m and the order of denominator polynomial is n .

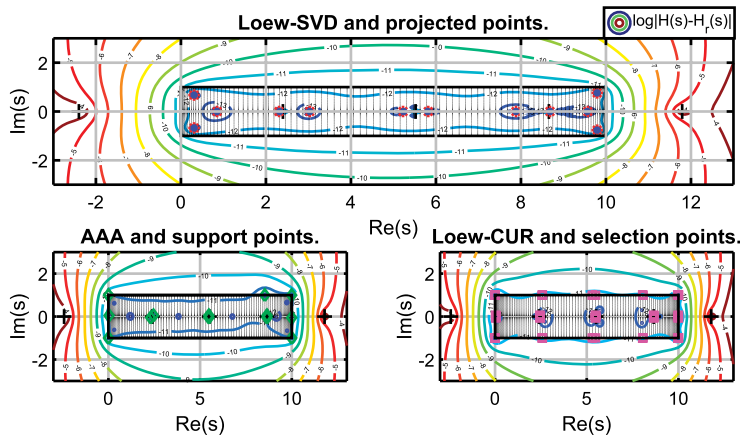


Figure 6.16: Extrapolation error as $\log |H(s) - H_r(s)|$ in $[-3, 13] \times [-3, 3] \subset \mathbb{C}$. The symbol ‘+’ is for the original poles.

All methods constructed accurate rational approximants. Notice, however, that the Loewner approach reaches similar precision with AAA without performing any optimization step. Finally, in terms of computational complexity, the CUR method performed the best.

6.4.4 An Euler–Bernoulli beam

In this subsection we analyse the approximation of an Euler–Bernoulli clamped beam [18]. The underlying PDE describes the oscillation of the free end. As shown in [18], the non-rational transfer function is given by

$$\begin{aligned}
 H(s) &= \frac{sn(s)}{(EI + sc_d I)m^3(s)d(s)}, \quad \text{where} \\
 m(s) &= \left[\frac{-s^2}{EI + c_d I s} \right]^{\frac{1}{4}}, \quad d(s) = 1 + \cosh(Lm(s)) \cos(Lm(s)), \\
 n(s) &= \cosh(Lm(s)) \sin(Lm(s)) - \sinh(Lm(s)) \cos(Lm(s)).
 \end{aligned} \tag{6.31}$$

Usually, the next step consists of a discretization of the PDE involved. We bypass this step and instead take frequency response measurements making use of the transfer function above. The parameter specification is as in [18].⁷ Thus, we have the frequency response of the beam as in Figure 6.17 and on the left pane.

⁷ Young’s modulus (elasticity constant): $E = 69 \text{ GPa} = 6.9 \cdot 10^{10} \text{ N/m}^2$, moment of inertia: $I = 3.58 \cdot 10^{-9} \text{ m}^4$, damping constant: $c_d = 5 \cdot 10^{-4}$, length: $L = 0.7 \text{ m}$, base: $b = 0.07 \text{ m}$, height: $h = 0.0085 \text{ m}$.

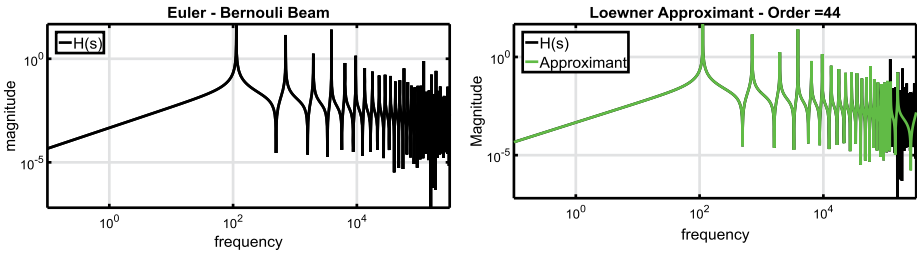


Figure 6.17: Left pane: Original frequency response of the beam. Right pane: The approximant which constructed with the Loewner framework.

The next step is to collect 2,000 measurements on the imaginary axis (frequencies $j\omega_i, i = 1, \dots, 2000$), spaced logarithmically from 1 rad/s to 10^5 rad/s. These points are depicted in the left pane of Figure 6.18.

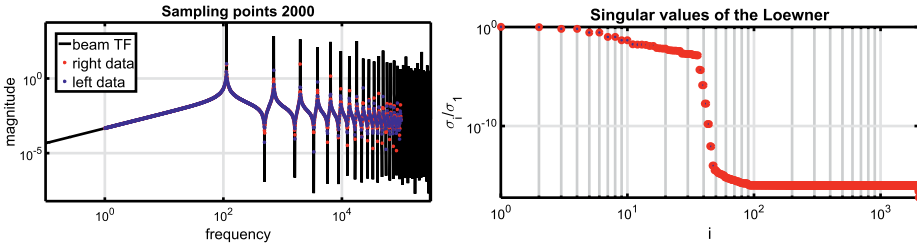


Figure 6.18: Left pane: 2,000 sampling points alternating as left and right. Right pane: The singular value decay.

The singular value of the Loewner matrices decay is as shown in Figure 6.18 on the right pane. Thus, we construct a reduced model with dimension $r = 44$ and the Loewner approximant in Figure 6.17 (right pane) is depicted.

Finally, the poles and zeros for every method are presented in Figure 6.19. The quality of the approximation is given for each method in Figure 6.20 where the evaluation is in the frequency range from 1 to $10^{5.5}$. The error outside the sampling domain increases thus indicating the difficulty of approximation outside of the sampling domain for infinite dimensional systems.

6.4.5 Heat equation

Next, we investigate an one-dimensional heat equation [13]. The corresponding PDE describing the diffusion of heat leads to the following non-rational transfer function:

$$H(s) = e^{-\sqrt{s}}, \quad s \in \mathbb{C}. \tag{6.32}$$

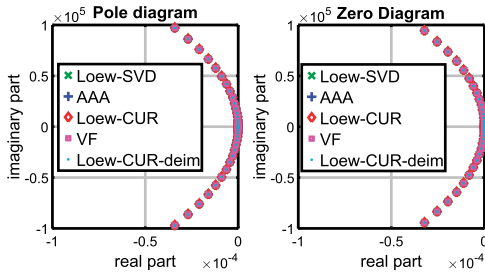


Figure 6.19: Pole/Zero diagram for every method (LoewSVD, LoewCUR-cross, LoewCUR-DEIM, VF and AAA).

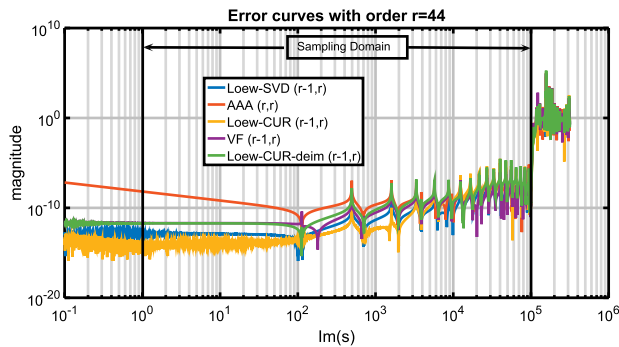


Figure 6.20: The error distribution with 8,000 evaluation points grid.

The aim is to construct reduced models by means of the Loewner framework and compare the results with the TF-IRKA used in [13]. Iterative Rational Krylov Algorithm - IRKA [14] builds optimal reduced models by minimizing the \mathcal{H}_2 error [39].

By collecting 1,000 values of the transfer function on the imaginary axis, the resulting reduced order was chosen to be $r = 6$ (as in [13]). For this truncation order, $\frac{\sigma_6}{\sigma_1} \approx 6 \cdot 10^{-3}$. In Figure 6.21c, the pole/zero distribution for every method is depicted; in Figure 6.21d, the selected points are shown. It is worth mentioning that the Loewner SVD method produced poles near to the optimal set computed by means of IRKA; see Figure 6.21c. Approximation results are in Figure 6.22.

6.4.6 Approximation of a two-peak function

In this section we present an example involving a hyperbolic sine from [22]. The difficulty here results from the two differentiable peaks. More precisely, the function is

$$f(x) = \frac{100\pi(x^2 - 0.36)}{\sinh(100\pi(x^2 - 0.36))}, \quad x \in [-1, 1],$$

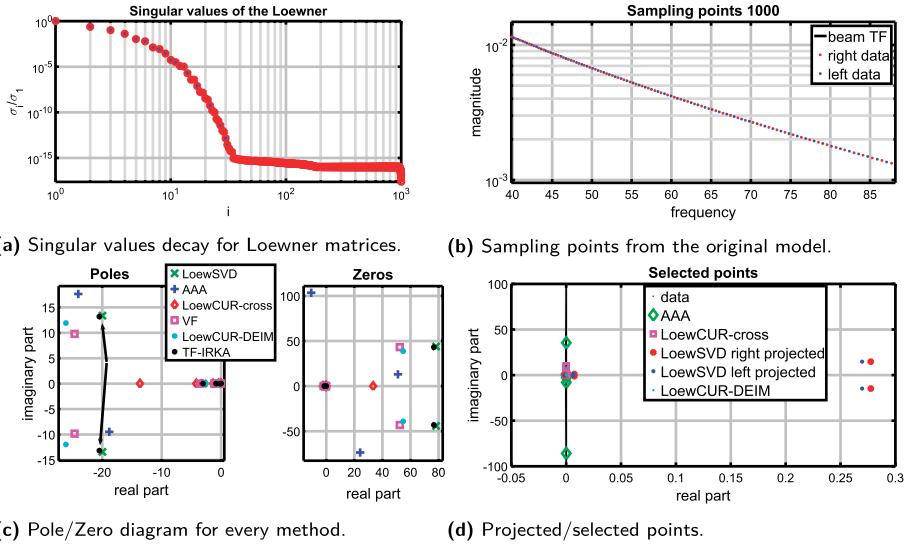


Figure 6.21: Approximation of the heat equation with LoewSVD, LoewCUR, VF, AAA, TF-IRKA.

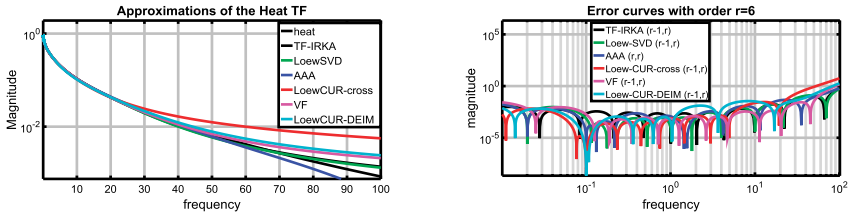


Figure 6.22: Approximation results for the heat equation with various interpolation methods.

and is shown Figure 6.23 (left pane). We approximate this function by choosing 1,000 equispaced points in $[-1, 1]$ as on the right pane in Figure 6.23. The singular values of the Loewner matrix are shown in Figure 6.24 on the left pane while the selected points are shown on the right pane of the same figure. The order is selected to be $r = 38$ with $(\frac{\sigma_{38}}{\sigma_1} \approx 10^{-12})$. In Figure 6.25, the distribution of the poles and zeros for each method is shown. On the other hand, AAA looks quite different because it does not impose real symmetry.

Remark 6.8. In Figure 6.24, right pane, the different supports points are shown. In the case of the LoewSVD method two almost pure imaginary projected points are obtained even if the initial sampling points were real.

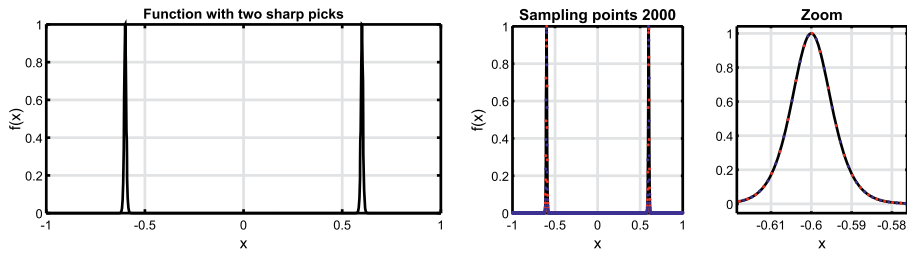


Figure 6.23: Left pane: The function f with two very sharp differentiable picks. Right pane: 1,000 sampling points and zoom in close to one pick.

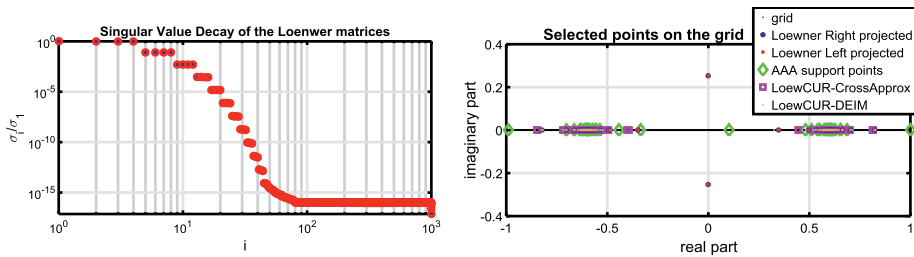


Figure 6.24: Left pane: Singular values decay. Right pane: Various points for every method and the projected points from the Loewner framework.

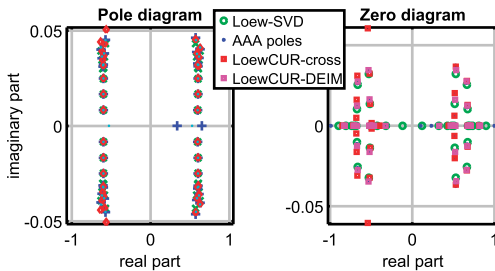


Figure 6.25: The pole/zero diagram.

Finally we observe a good fit for every method, with slightly better performance attained for the Loewner SVD method (see the error plot in Figure 6.26).

6.4.7 Approximation of the sign function

Our final case study problem concerns the approximation of the sign function, known as Zolotarev's fourth problem. Here, we compare the approximation obtained using the Loewner SVD with the optimal solution that is explicitly known [1]. Given two disjoint closed complex sets E and F , Zolotarev's fourth problem is to find the rational

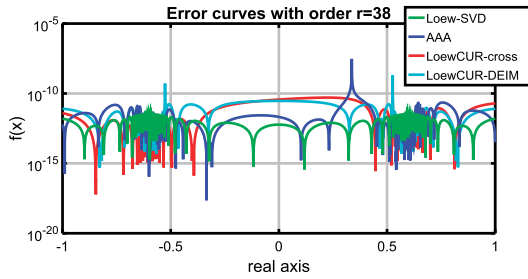


Figure 6.26: The error profile with 5,000 evaluation points over $[-1, 1]$.

function $r(x) = \frac{p(x)}{q(x)}$, where p, q are polynomials of degree k , that deviates least from the sign function

$$\text{sign}(x) = \begin{cases} -1, & x \in E, \\ +1, & x \in F, \end{cases}$$

on $E \cup F$. For general sets E and F , the solution to Zolotarev’s fourth problem is not known; however, there are special cases where the rational function can be given explicitly. For the real disjoint intervals, $E = [-b, -1]$, and $F = [1, b]$ with $b > 1$, an explicit (optimal) solution to Zolotarev’s fourth problem is known [1]. Here, we investigated how well the Loewner framework can approximate this discontinuous function in two symmetric real intervals. We choose $b = 3$ and $N = 2,000$ initial interpolation points from $[-3, -1] \cup [1, 3]$. We perform two experiments. Firstly, we choose initial interpolation points as equispaced and secondly, as Chebyshev nodes. For each choice, we split the data as “half-half” and “alternating” as discussed previously. The left pane in Figure 6.27 shows the plot of the sign function.

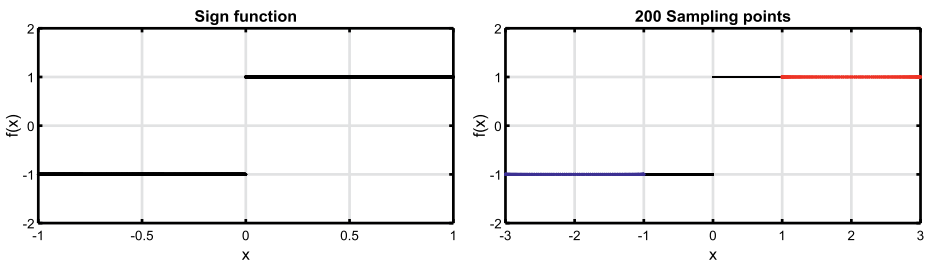


Figure 6.27: Left pane: The sign function. Right pane: 200 Chebyshev points in $[-3, -1] \cup [1, 3]$.

In [17] the explicit solution of this optimization problem is computed. We start by taking $N = 2,000$ measurements as Chebyshev nodes as in Figure 6.27 on the right pane. The above sampling way leads to the following singular value decay of the Loewner matrices as in Figure 6.28 on the left pane.

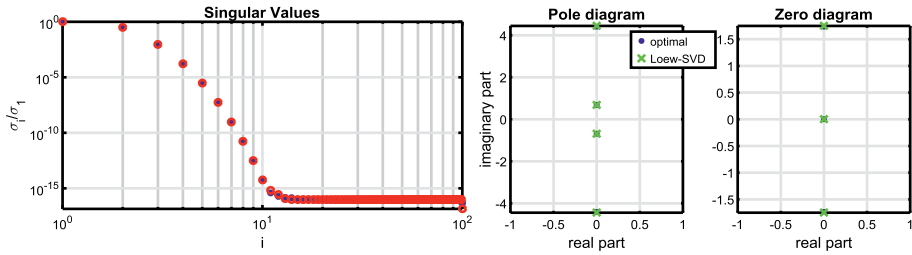


Figure 6.28: Left pane: The singular value decay of the Loewner pencil. Right pane: Pole/Zero diagram for the Loewner and the optimal approximant with order $r = 4$.

From the rank-revealing factorization in the left pane in Figure 6.28, we chose $r = 4$ with $\frac{\sigma_4}{\sigma_1} = 1.657 \cdot 10^{-4}$. In Figure 6.28 on the right pane is the distribution of the pole/zero diagram which is derived from the Loewner SVD method, in comparison with the optimal set is presented.

In Figure 6.29 (left) the Loewner approximant is shown. It is quite close to the optimal one by choosing the Chebyshev nodes and splitting the left and right points as “half–half”. Indeed, the error distribution as presented in the optimal interpolant in Figure 6.30 with the blue line has the equioscillation property of the optimal approximant in the *infinity norm* - $\|\mathbf{x}\|_{\infty} = \max(|x_1|, \dots, |x_n|)$. Thus the equioscillation of the error $|\text{sign}(x) - r(x)|$ on both intervals shows the optimality of the approximant. The Loewner framework succeeds in constructing an approximant very close to the optimal. Another aspect is shown in Figure 6.29 (right pane). More specifically, note that the projected points are indeed interpolation points.

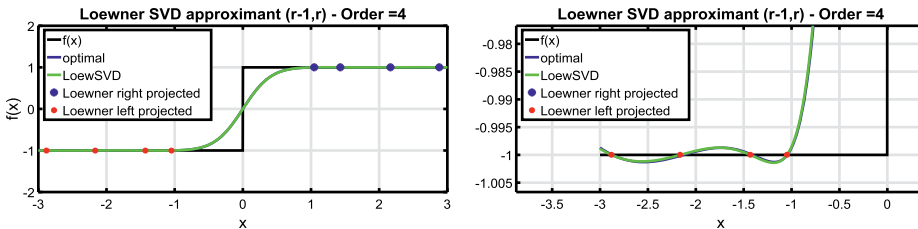


Figure 6.29: Left pane: A comparison between the Loewner approximant with the optimal one order $r = 4$. Right pane: The projected points are approximated interpolation points.

Remark 6.9. If the choice of the splitting is disjoint—“half–half” as in this experiment, the constructed approximant interpolates the data as in Figure 6.29(right pane). If the choice is “alternating” by mixing left and right, then the projected low order model approximates the values and the derivatives at the interpolation points as in Figure 6.31.

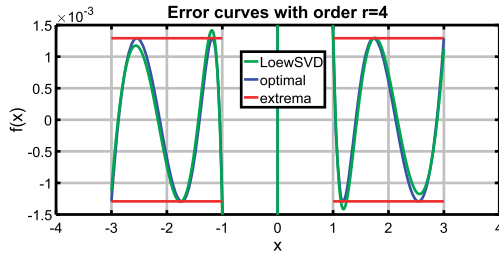


Figure 6.30: Error plot with the Loewner approximant and the optimal solution as well with order $r = 4$.

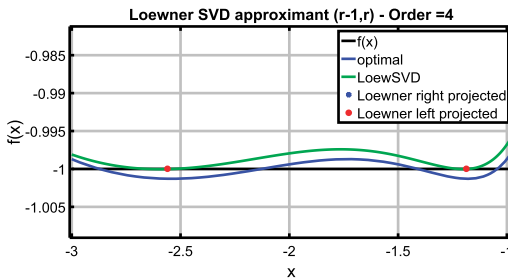


Figure 6.31: By splitting the data as “alternating”, the projected Loewner model approximates the first derivative as well (*Hermite interpolation conditions*).

6.5 Epilogue

Interpolatory methods for model identification and reduction were studied in this contribution. The main focus was on the Loewner framework. The aim was to introduce the Loewner framework by providing results which connect this rational interpolation tool with system theory. At the same time, algorithms that make the Loewner framework a complete numerical tool for approximation with ease of implementation are offered. Several case studies illustrate the effectiveness of the method. Implementation issues like the splitting of the data in left and right were addressed. Finally, connections with the SVD, the r-SVD, CUR, VF and IRKA have been detailed.

Bibliography

- [1] N. I. Akhiezer. *Elements of the theory of elliptic functions. Translations of Mathematical Monographs*, volume 79. American Mathematical Society, Providence, RI, 1990. Translated from the second Russian edition by H. H. McFaden.
- [2] A. C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. Society for Industrial and Applied Mathematics, 2005.

- [3] A. Antoulas. The Loewner framework and transfer functions of singular/rectangular systems. *Appl. Math. Lett.*, 54:36–47, 2016.
- [4] A. Antoulas, A. Ionita, and S. Lefteriu. On two-variable rational interpolation. *Linear Algebra Appl.*, 436(8):2889–2915, 2012. Special Issue dedicated to Danny Sorensen’s 65th birthday.
- [5] A. C. Antoulas, I. V. Gosea, and A. C. Ionita. Model reduction of bilinear systems in the Loewner framework. *SIAM J. Sci. Comput.*, 38(5):B889–B916, 2016.
- [6] A. C. Antoulas, S. Lefteriu, and A. C. Ionita. *A Tutorial Introduction to the Loewner Framework for Model Reduction*, pages 335–376. Society for Industrial and Applied Mathematics for Computational Science & Engineering, 2017, Ch. 8.
- [7] A. C. Antoulas, B. Zhu, Q. Zhang, B. York, B. W. O’Malley, and C. C. Dacso. A novel mathematical method for disclosing oscillations in gene transcription: A comparative study. *PLoS ONE*, 13:1–20, 2018.
- [8] A. C. Antoulas, I. V. Gosea, and M. Heinkenschloss. On the Loewner framework for model reduction of Burgers’ equation. In R. King, editor, *Active Flow and Combustion Control 2018*, pages 255–270. Springer, Cham, 2019.
- [9] A. C. Antoulas, C. A. Beattie, and S. Güğercin. *Interpolatory Methods for Model Reduction*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2020.
- [10] A. Antoulas, I. Gosea, and M. Heinkenschloss. Data-driven model reduction of the Oseen equations using the Loewner framework. In S. Grundel, T. Reis and S. Schöps, editors, *Progress in Differential-Algebraic Equations II. Differential-Algebraic Equations Forum*. Springer, 2020, accepted for publication. <https://doi.org/10.1109/ECC.2015.7330568>.
- [11] A. Antoulas, I. Gosea, and M. Heinkenschloss. Reduction of systems with polynomial nonlinearities in the Loewner framework. In *Book of Abstracts of XXI Householder Symposium on Numerical Linear Algebra*, Selva di Fasano, Italy, June 14–19, 2020.
- [12] C. Baker, K. Gallivan, and P. V. Dooren. Low-rank incremental methods for computing dominant singular subspaces. *Linear Algebra Appl.*, 436(8):2866–2888, 2012. Special Issue dedicated to Danny Sorensen’s 65th birthday.
- [13] C. Beattie and S. Gugercin. Realization-independent H₂-approximation. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 4953–4958, 2012.
- [14] C. Beattie and S. Gugercin. *Model Reduction by Rational Interpolation*, pages 297–334. Society for Industrial and Applied Mathematics for Computational Science & Engineering, 2017, Ch. 7.
- [15] B. Beckermann and A. Townsend. On the singular values of matrices with displacement structure. *SIAM J. Matrix Anal. Appl.*, 38(4):1227–1248, 2017.
- [16] P. Benner and P. Goyal. Interpolation-based model order reduction for polynomial parametric systems, Tech. rep., arXiv preprint available at <https://arxiv.org/abs/1904.11891>, 2019. <https://doi.org/10.1137/19M1259171>.
- [17] M. Berljafa and S. Güttel. A rational Krylov toolbox for Matlab, 2014.
- [18] R. Curtain and K. Morris. Transfer functions of distributed parameter systems: A tutorial. *Automatica*, 45(5):1101–1116, 2009.
- [19] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM J. Matrix Anal. Appl.*, 30(2):844–881, 2008.
- [20] Z. Drmač and B. Peherstorfer. Learning low-dimensional dynamical-system models from noisy frequency-response data with Loewner rational interpolation, Tech. rep., arXiv preprint available at <https://arxiv.org/abs/1910.00110>, 2019.
- [21] Z. Drmač, S. Gugercin, and C. Beattie. Vector fitting for matrix-valued rational approximation. *SIAM J. Sci. Comput.*, 37(5):A2346–A2379, 2015.
- [22] S.-I. Filip, Y. Nakatsukasa, L. N. Trefethen, and B. Beckermann. Rational minimax approximation via adaptive barycentric representations. *SIAM J. Sci. Comput.*, 40(4):A2427–A2455, 2018.

- [23] K. Gallivan, A. Vandendorpe, and P. V. Dooren. On the generality of multipoint Padé approximations. *IFAC Proc. Vol.*, 35(1):331–336, 2002. 15th IFAC World Congress.
- [24] K. Gallivan, A. Vandendorpe, and P. V. Dooren. Sylvester equations and projection-based model reduction. *J. Comput. Appl. Math.*, 162(1):213–229, 2004. Proceedings of the International Conference on Linear Algebra and Arithmetic 2001.
- [25] K. Gallivan, A. Vandendorpe, and P. Van Dooren. Model reduction of MIMO systems via tangential interpolation. *SIAM J. Matrix Anal. Appl.*, 26(2):328–349, 2004.
- [26] S. Goreinov and E. Tyrtyshnikov. Quasioptimality of Skeleton approximation of a matrix in the Chebyshev norm. In *Doklady Mathematics*, volume 83, pages 374–375. Springer, 2011.
- [27] S. Goreinov, E. Tyrtyshnikov, and N. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra Appl.*, 261(1):1–21, 1997.
- [28] S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, and N. Zamarashkin. *How to find a good submatrix*, 2010.
- [29] I. V. Gosea and A. C. Antoulas. Model reduction of linear and nonlinear systems in the Loewner framework: A summary. In *2015 European Control Conference (ECC)*, pages 345–349, 2015.
- [30] I. V. Gosea and A. C. Antoulas. Stability preserving post-processing methods applied in the Loewner framework. In *2016 IEEE 20th Workshop on Signal and Power Integrity (SPI)*, pages 1–4, 2016.
- [31] I. V. Gosea and A. C. Antoulas. Approximation of a damped Euler-Bernoulli beam model in the Loewner framework, Tech. rep., arXiv preprint available at <https://arxiv.org/abs/1712.06031>, 2017.
- [32] I. V. Gosea and A. C. Antoulas. Data-driven model order reduction of quadratic-bilinear systems. *Numer. Linear Algebra Appl.*, 25(6):e2200, 2018.
- [33] I. V. Gosea and A. C. Antoulas. Rational approximation of the absolute value function from measurements: a numerical study of recent methods, Tech. rep., arXiv preprint available at <https://arxiv.org/abs/2005.02736>, 2020.
- [34] I. V. Gosea, M. Petreczky, and A. C. Antoulas. Data-driven model order reduction of linear switched systems in the Loewner framework. *SIAM J. Sci. Comput.*, 40(2):B572–B610, 2018.
- [35] I. V. Gosea, I. P. Duff, P. Benner, and A. C. Antoulas. Model order reduction of bilinear time-delay systems. In *2019 18th European Control Conference (ECC)*, pages 2289–2294, 2019.
- [36] I. Gosea, D. Karachalios, and A. C. Antoulas. Learning reduced-order models of quadratic control systems from input-output data. In *European Control Conference (ECC21), June 29–July 2, 2021, Rotterdam, The Netherlands (Virtual Conference)*, 2021, accepted for publication, arXiv preprint available at <https://arxiv.org/abs/2012.02075>, 2020.
- [37] I. V. Gosea, Q. Zhang, and A. C. Antoulas. Preserving the DAE structure in the Loewner model reduction and identification framework. *Adv. Comput. Math.*, 46(3), 2020.
- [38] E. Grimme, K. A. Gallivan, and P. V. Dooren. On some recent developments in projection-based model reduction. In *ENUMATH 97*, Heidelberg, pages 98–113. World Sci. Publishing, River Edge, NJ, 1998.
- [39] S. Gugercin, A. C. Antoulas, and C. Beattie. H2 model reduction for large-scale linear dynamical systems. *SIAM J. Matrix Anal. Appl.*, 30(2):609–638, 2008.
- [40] B. Gustavsen and A. Semlyen. Rational approximation of frequency domain responses by vector fitting. *IEEE Trans. Power Deliv.*, 14:1052–1061, 1999.
- [41] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, 2011.
- [42] A. C. Ionita and A. C. Antoulas. Data-driven parametrized model reduction in the Loewner framework. *SIAM J. Sci. Comput.*, 36(3):A984–A1007, 2014.
- [43] D. S. Karachalios, I. V. Gosea, Q. Zhang, and A. C. Antoulas. Case study: Approximations of the Bessel function, Tech. rep., arXiv preprint available at <https://arxiv.org/abs/1801.03390>, 2017.

- [44] D. S. Karachalios, I. V. Gosea, and A. C. Antoulas. Data-driven approximation methods applied to non-rational functions. *PAMM*, 18(1), e201800368, 2018.
- [45] D. S. Karachalios, I. V. Gosea, and A. C. Antoulas. A bilinear identification-modeling framework from time domain data. *PAMM*, 19(1):e201900246, 2019.
- [46] D. Karachalios, I. Gosea, and A. C. Antoulas. On bilinear time domain identification and reduction in the Loewner framework. In P. Benner et al., editors, *Model Reduction of Complex Dynamical Systems. International Series of Numerical Mathematics*. Birkhäuser/Springer Nature, Chur, Switzerland, 2021. https://doi.org/10.1007/978-3-030-72983-7_1
- [47] B. Kramer and A. A. Gorodetsky. System identification via CUR-factored hankel approximation. *SIAM J. Sci. Comput.*, 40(2):A848–A866, 2018.
- [48] S. Lefteriu and A. C. Antoulas. A new approach to modeling multiport systems from frequency-domain data. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 29(1):14–27, 2010.
- [49] S. Lefteriu and A. C. Antoulas. A new approach to modeling multiport systems from frequency-domain data. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 29(1):14–27, 2010.
- [50] S. Lefteriu, A. C. Ionita, and A. C. Antoulas. In *Modeling Systems Based on Noisy Frequency and Time Domain Measurements*, pages 365–378. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [51] M. W. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *Proc. Natl. Acad. Sci.*, 106(3):697–702, 2009.
- [52] A. Mayo and A. Antoulas. A framework for the solution of the generalized realization problem. *Linear Algebra Appl.*, 425(2):634–662, 2007. Special Issue in honor of Paul Fuhrmann.
- [53] Y. Nakatsukasa, O. Sète, and L. N. Trefethen. The AAA algorithm for rational approximation. *SIAM J. Sci. Comput.*, 40(3):A1494–A1522, 2018.
- [54] V. Olshevsky, editor. In *Structured matrices in mathematics, computer science, and engineering. I. Contemporary Mathematics*, volume 280. American Mathematical Society, Providence, RI, 2001.
- [55] I. Oseledets and E. Tyrtshnikov. TT-cross approximation for multidimensional arrays. *Linear Algebra Appl.*, 432(1):70–88, 2010.
- [56] B. Peherstorfer, S. Gugercin, and K. Willcox. Data-driven reduced model construction with time-domain Loewner models. *SIAM J. Sci. Comput.*, 39, 2017.
- [57] P. Rapisarda and A. C. Antoulas. A duality perspective on Loewner rational interpolation and state-space modelling of vector-exponential trajectories. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 2096–2100, 2015.
- [58] P. Rapisarda and A. C. Antoulas. Bilinear differential forms and the Loewner framework for rational interpolation. In M. N. Belur, M. K. Camlibel, P. Rapisarda and J. M. Scherpen, editors, *Mathematical Control Theory II: Behavioral Systems and Robust Control*, volume 462, pages 23–43. Springer, 2015.
- [59] P. Schulze and B. Unger. Data-driven interpolation of dynamical systems with delay. *Syst. Control Lett.*, 97:125–131, 2016.
- [60] J. D. Simard and A. Astolfi. An interconnection-based interpretation of the Loewner matrices*. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 7788–7793, 2019.
- [61] J. D. Simard and A. Astolfi. Loewner functions for linear time-varying systems with applications to model reduction. In *21st IFAC World Congress*, Berlin, Germany, 2020, accepted for publication. <https://doi.org/10.1016/j.ifacol.2020.12.1578>
- [62] D. Sorensen and A. Antoulas. The Sylvester equation and approximate balanced reduction. *Linear Algebra Appl.*, 351–352:671–700, 2002. Fourth Special Issue on Linear Systems and Control.

- [63] D. C. Sorensen and M. Embree. A deim induced CUR factorization. *SIAM J. Sci. Comput.*, 38(3):A1454–A1482, 2016.
- [64] L. N. Trefethen. *Approximation Theory and Approximation Practice (Other Titles in Applied Mathematics)*. Society for Industrial and Applied Mathematics, USA, 2012.
- [65] C. D. Villemagne and R. E. Skelton. Model reductions using a projection formulation. In *26th IEEE Conference on Decision and Control*, volume 26, pages 461–466, 1987.
- [66] A. Yousuff and R. Skelton. Covariance equivalent realizations with application to model reduction of large-scale systems. In C. Leondes, editor, *Decentralized/Distributed Control and Dynamic Systems, Part 1 of 3. Control and Dynamic Systems*, volume 22, pages 273–348. Academic Press, 1985.
- [67] A. Yousuff, D. Wagie, and R. Skelton. Linear system approximation via covariance equivalent realizations. *J. Math. Anal. Appl.*, 106(1):91–115, 1985.
- [68] B. Zhu, Q. Zhang, Y. Pan, E. M. Mace, B. York, C. A. Antoulas, C. C. Dacso, and B. W. O’Malley. A cell-autonomous mammalian 12 hr clock coordinates metabolic and stress rhythms. *Cell Metab.*, 6(25):1305–1319, 2017.