ESAIM: M2AN 55 (2021) 1895–1920
https://doi.org/10.1051/m2an/2021040

ESAIM: Mathematical Modelling and Numerical Analysis
www.esaim-m2an.org

NON-INTRUSIVE DOUBLE-GREEDY PARAMETRIC MODEL REDUCTION BY INTERPOLATION OF FREQUENCY-DOMAIN RATIONAL SURROGATES

Fabio Nobile and Davide Pradovera*

Abstract. We propose a model order reduction approach for non-intrusive surrogate modeling of parametric dynamical systems. The reduced model over the whole parameter space is built by combining surrogates in frequency only, built at few selected values of the parameters. This, in particular, requires matching the respective poles by solving an optimization problem. If the frequency surrogates are constructed by a suitable rational interpolation strategy, frequency and parameters can both be sampled in an adaptive fashion. This, in general, yields frequency surrogates with different numbers of poles, a situation addressed by our proposed algorithm. Moreover, we explain how our method can be applied even in high-dimensional settings, by employing locally-refined sparse grids in parameter space to weaken the curse of dimensionality. Numerical examples are used to showcase the effectiveness of the method, and to highlight some of its limitations in dealing with unbalanced pole matching, as well as with a large number of parameters.

Mathematics Subject Classification. 35B30, 35P15, 41A20, 41A63, 93C35, 93C80.

Received February 18, 2021. Accepted July 29, 2021.

1. Introduction

The numerical simulation of dynamical systems in frequency domain is of utmost importance in several engineering fields, among which electronic circuit design, acoustics, resonance modeling and control for large structures, and many others. The computational burden of such simulations has kept increasing in the last decades: on one hand, the problem size has been growing because of the need for higher numerical resolution; on the other hand, the necessity to tune design parameters and model uncertain features has lead researchers to tackle parametric models, possibly with a large number of parameters.

The purpose of *model order reduction* (MOR) in general, and of *parametric MOR* (pMOR) in the specific case of dynamical system in the presence of parameters, is to alleviate this computational load. The main strategy to reach this goal relies on building a surrogate model (*reduced order model*, ROM), which mimics accurately the original problem, but which can be solved at a much reduced cost. In the last two decades, the field of pMOR has thrived, leading to the development, analysis, and application of a wide collection of surrogate modeling strategies. In general, we can assign each of these methods to one of two main categories:

Keywords and phrases. Parametric model order reduction, parametric dynamical systems, non-intrusive method, minimal rational interpolation, greedy algorithm.

Chair of Scientific Computing and Uncertainty Quantification (CSQI), Institute of Mathematics, École Polytechnique Fédérale de Lausanne, Station 8, 1015 Lausanne, Switzerland.
*Corresponding author: davide.pradovera@epfl.ch

© The authors. Published by EDP Sciences, SMAI 2021

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

- **Projection-based pMOR.** The surrogate model is built by restricting the original problem onto a suitable subspace, computed from a set of solutions (most often, snapshots of the system state) of the full problem. This requires access to the operators of the full model, which are not necessarily available in applications, for instance in the case of a black-box solver, or if the system operators never get fully assembled in the solution process. Some subcategories of projective pMOR can be identified depending on whether a global basis (such as POD/Reduced Basis [6] or multi-parameter multi-moment-matching [8, 41]) or a collection of local bases (*e.g.*, manifold interpolation of local bases [2] or of reduced system matrices [3, 27, 31]) are employed.
- **Non-intrusive pMOR.** The surrogate model is constructed by interpolation or regression of a set of solutions (usually, output samples) of the full problem. As long as the system state is not necessary for the application at hand, it is common to work directly with the system output. In this case as well, the methods can be further split into two subgroups, although the boundary between the two is more vague: some approaches set up the surrogate by solving a unique global interpolation problem [20, 24, 26], whereas others build it by first constructing several (rational) models in frequency only, and then combining them over parameter space [17, 43, 44].

Here, we focus on this last class of techniques: non-intrusive approaches which build a surrogate by interpolating local frequency models, *i.e.*, models constructed at some values of the parameters. A discussion of the pros and cons of this local idea, compared to global approaches, can be found in the excellent survey [9] (whose focus is, however, mostly on projective pMOR), as well as in [44]. The latter article is closer in spirit to the present discussion and, as such, we will take it as starting point and main reference for our presentation. Building upon [43, 44], the purpose of this paper is the description of a fully non-intrusive pMOR technique based on parameter interpolation of frequency surrogates. As will be made more clear in the next sections, the objects that are interpolated over parameter space are poles and residues of the Heaviside decomposition of the frequency surrogates. Without going into too much detail here, we summarize briefly the main novelties of our approach, focusing on how it generalizes [43]:

- *Both* frequency space and parameter domain are sampled adaptively, allowing for a better exploration of frequency and parameter domains, as well as for an improved efficiency in the construction of the reduced model. To this aim, we leverage the *minimal rational interpolation* technique as described in [35].
- Our proposed approach for pole/residue matching has polynomial worst-case complexity despite the combinatorial nature of the task. Notably, our approach can also be applied in the case of unbalanced (*i.e.*, with different numbers of poles and residues) frequency surrogates. This is a critical property in view of our adaptive frequency sampling strategy.
- We propose a framework for adaptive parameter sampling in a general *high-dimensional* setting by introducing a hierarchical locally-refined sparse grid structure in parameter space.

Before proceeding, we deem of importance to remark that local approaches based on interpolation of poles and residues, by their very nature, struggle in dealing with poles and residues that do not depend smoothly on the parameters. In such cases, surrogates that avoid the Heaviside decomposition, *e.g.*, by interpolating the rational frequency response directly rather than its poles and residues [17], can have more beneficial properties, since they do not rely on the smoothness of poles and residues. Still, we choose to pursue an Heaviside-based approach rather than a rational-based one for two main reasons:

- By interpolation of poles and residues, one can handle with good flexibility the fairly common case of unbalanced frequency surrogates. More specifically, we will describe how spurious poles may be removed (and missing poles reconstructed) *on the fly* by exploiting the structure of the Heaviside expansion.
- Despite its intrinsic difficulties in approximating non-smooth poles, an adaptive selection of the parameter sample points, as proposed in this work, can recover a good performance, by refining locally at critical parameter locations (*e.g.*, branch points of the poles).

Outline of the paper. We introduce the parametric framework for our approach in Section 2. The ensuing Section 3 contains our main contribution, in the form of a description of the proposed pMOR technique, with

each of the subsections therein examining a different feature of the algorithm. In Section 4, we investigate some of the limitations of our method in the case of crossing and non-smooth poles. We integrate our discussion with two numerical examples in Section 5, showcasing the effectiveness of our method. We conclude with a summary and an outlook for future research in Section 6.

2. Problem framework and notation

The field of pMOR is closely entwined with the study of parametric dynamical systems in the frequency domain. More precisely, one considers the problem:

$$\text{given } (z,\mathbf{p}) \in Z \times \boldsymbol{\mathcal{P}}, \text{ find } Y = Y(z,\mathbf{p}) \text{ such that } \begin{cases} (zE_{\mathbf{p}} - A_{\mathbf{p}})\, X(z,\mathbf{p}) = B_{\mathbf{p}}, \\ Y(z,\mathbf{p}) = C_{\mathbf{p}} X(z,\mathbf{p}), \end{cases} \tag{2.1}$$

where

$$A_{\mathbf{p}}, E_{\mathbf{p}} \in \mathbb{C}^{n_S \times n_S}, \quad X(z,\mathbf{p}), B_{\mathbf{p}} \in \mathbb{C}^{n_S \times n_I}, \quad C_{\mathbf{p}} \in \mathbb{C}^{n_O \times n_S}, \quad \text{and} \quad Y(z,\mathbf{p}) \in \mathbb{C}^{n_O \times n_I}.$$

We call X and Y the *state* and the *output* of the system, respectively. The sets $Z \subset \mathbb{C}$ and $\boldsymbol{\mathcal{P}} \subset \mathbb{C}^d$ ($\mathbb{R}^d$ in most applications) are frequency and parameter domains, respectively. The subscript $\mathbf{p}$ of the system matrices denotes their eventual dependence on the parameters $\mathbf{p}$. We remark that we do not exclude the case of the state being the output of interest: such case can be obtained quite trivially by setting $n_O = n_S$ and $C_{\mathbf{p}}$ equal to the identity matrix.

The behavior of state X and output Y with respect to z only (for fixed $\mathbf{p}$, the non-parametric case) is well understood, and is backed up by an extensive literature in system theory [4]. Among the many properties of these systems, one is crucial to our discussion, namely the Heaviside decomposition: under some quite broad assumptions on the spectral properties of the pencil $(A_{\mathbf{p}}, E_{\mathbf{p}})$, we can write

$$Y(z,\mathbf{p}) = \sum_j (z - \lambda_{\mathbf{p}}^{(j)})^{-m_{\mathbf{p}}^{(j)}} Y_{\mathbf{p}}^{(j)} \quad \text{for } z \in Z \setminus \{\lambda_{\mathbf{p}}^{(j)}\}_j, \tag{2.2}$$

with $\lambda_{\mathbf{p}}^{(j)} \in \mathbb{C} \cup \{\infty\}$, $m_{\mathbf{p}}^{(j)} \in \mathbb{N}$, and $Y_{\mathbf{p}}^{(j)} \subset \mathbb{C}^{n_O \times n_I}$, for all j.

We remark that our discussion relies only on the decomposition (2.2) of the output, together with the following assumptions:

(a) $Y : Z \times \boldsymbol{\mathcal{P}} \to V$, with V a normed vector space (for (2.1), we may set $V = \mathbb{C}^{n_O \times n_I}$ endowed with the Frobenius norm).
(b) In (2.2), $\lambda^{(j)}$ and $Y^{(j)}$ depend continuously on $\mathbf{p}$ for all j, *cf.* Section 4.
(c) In (2.2), $m^{(j)}$ is independent of $\mathbf{p}$ for all j.

As such, our proposed strategy extends further than linear parametric dynamical systems (2.1). Some examples of practical interest are parametric scattering problems in frequency domain [22] and parametric nonlinear eigenproblems [10]. Still, for simplicity of exposition, we will restrict our discussion to the finite-dimensional linear parametric dynamical system (2.1).

In order to simplify the presentation, from here onward we will replace (2) with the stronger assumption that $m^{(j)}$ be equal to 1. This excludes the possibility of degenerate eigenvalues. We remark that, in most practical applications, each multiplicity $m^{(j)}$ is indeed identically equal to 1. We postpone a discussion on this till Sections 4 and 6.

3. The double-greedy pMOR strategy

We start this section by detailing our pMOR technique in its formulation without $\mathbf{p}$-adaptivity. The general structure of the pole-matching-based pMOR algorithm is summarized in Algorithms 3.1 and 3.2, and the different building blocks are discussed more extensively in the following subsections.

Algorithm 3.1 Pole-matching pMOR – Offline phase

function pROM_Train($\mathbf{p}_1, \dots, \mathbf{p}_S$)
  for $k = 1, \dots, S$ **do**
    $\mathrm{ROM}_k \leftarrow$ BuildFrequencyROM($\mathbf{p}_k$) ▷ Section 3.1
  end for
  $J \leftarrow \{1\}$ ▷ Initialize root of search tree
  for iter $= 2, \dots, S$ **do** ▷ Matching loop
    $(k, \ell) \leftarrow \arg\min_{k' \in J,\, \ell' \notin J} \|\mathbf{p}_{k'} - \mathbf{p}_{\ell'}\|$ ▷ Breadth-first search
    permutation $\leftarrow$ ModalMatching(ROM_k, ROM_ℓ) ▷ Section 3.2
    $\mathrm{ROM}_\ell \leftarrow$ ApplyPermutation(ROM_ℓ, permutation)
    $J \leftarrow J \cup \{\ell\}$ ▷ Add to explored set
  end for
  $\boldsymbol{\psi} \leftarrow$ BuildLocalWeightFunctions($\mathbf{p}_1, \dots, \mathbf{p}_S$) ▷ Section 3.3
  return $\{\mathrm{ROM}_k\}_{k=1}^S$, $\boldsymbol{\psi}$
end function

Algorithm 3.2 Pole-matching pMOR – Online phase

function pROM(z, $\mathbf{p}$; $\{\mathrm{ROM}_k\}_{k=1}^S$, $\boldsymbol{\psi}$)
  $\widetilde{Y} \leftarrow 0$, $\mathbf{w} \leftarrow \boldsymbol{\psi}(\mathbf{p})$
  for $j = 1, 2, \dots, \overline{n}$ **do** ▷ Loop over surrogate poles
    $\widetilde{\lambda}^{(j)} \leftarrow 0$, $\widetilde{Y}^{(j)} \leftarrow 0$
    for $k = 1, 2, \dots, S$ **do** ▷ Loop over frequency models
      Extract $\overline{\lambda}_k^{(j)}$ and $\overline{Y}_k^{(j)}$ from ROM_k
      $\widetilde{\lambda}^{(j)} \leftarrow \widetilde{\lambda}^{(j)} + w_k \overline{\lambda}_k^{(j)}$, $\widetilde{Y}^{(j)} \leftarrow \widetilde{Y}^{(j)} + w_k \overline{Y}_k^{(j)}$ ▷ Add local contribution
    end for
    $\widetilde{Y} \leftarrow \widetilde{Y} + \widetilde{Y}^{(j)} / \left(z - \widetilde{\lambda}^{(j)}\right)$ ▷ Add Heaviside term
  end for
  return $\widetilde{Y}$
end function

An online-offline decomposition of the algorithm is performed, in the usual MOR fashion [36]: first, the surrogate model is built in an expensive training phase, which requires solving the original problem at several values of frequency and parameters; the reduced model is then stored and can be evaluated with a (hopefully) much reduced computational cost at arbitrary frequency and parameter values.

The offline phase is shown in Algorithm 3.1. The input of the procedure is a set of parameter values $\boldsymbol{\mathcal{P}}_{\text{train}} = \{\mathbf{p}_1, \dots, \mathbf{p}_S\} \subset \boldsymbol{\mathcal{P}}$, where reduced models *in frequency only* are built: more precisely, for each $k = 1, \dots, S$, we compute the approximation

$$Y(z, \mathbf{p}_k) \approx \mathrm{ROM}_k(z) = \sum_{j=1}^{\overline{n}_k} \frac{\overline{Y}_k^{(j)}}{z - \overline{\lambda}_k^{(j)}}. \tag{3.1}$$

More details on this step are given in Section 3.1. The addends of the sum in (3.1) are then sorted in such a way that poles $\overline{\lambda}_k^{(j)}$ and residues $\overline{Y}_k^{(j)}$ with the same j but different k "correspond to each other". We explain what we mean by this, and how we achieve it, in Section 3.2. In the same section, we also discuss how we deal with the situation where two surrogate models have different amounts of poles, *i.e.*, $\overline{n}_k \neq \overline{n}_\ell$. For the remainder of the present overview, for simplicity we assume $\overline{n}_1 = \ldots = \overline{n}_S =: \overline{n}$.

At this point, it only remains to prescribe a rule to define the reduced model at a new parameter value $\mathbf{p} \in \boldsymbol{\mathcal{P}} \setminus \boldsymbol{\mathcal{P}}_{\text{train}}$. To this aim, we define S weight functions $\boldsymbol{\psi} = (\psi_1, \dots, \psi_S) : \boldsymbol{\mathcal{P}} \to \mathbb{C}^S$, which we employ to

interpolate poles and residues over $\mathbf{p}$-space:

$$\widetilde{\lambda}^{(j)}(\mathbf{p}) = \sum_{k=1}^{S} \psi_k(\mathbf{p}) \overline{\lambda}_k^{(j)} \quad \text{and} \quad \widetilde{Y}^{(j)}(\mathbf{p}) = \sum_{k=1}^{S} \psi_k(\mathbf{p}) \overline{Y}_k^{(j)}. \tag{3.2}$$

The resulting global model can then be evaluated through Algorithm 3.2, as

$$Y(z,\mathbf{p}) \approx \mathrm{pROM}(z,\mathbf{p}) = \sum_{j=1}^{\overline{n}} \frac{\widetilde{Y}^{(j)}(\mathbf{p})}{z - \widetilde{\lambda}^{(j)}(\mathbf{p})} = \sum_{j=1}^{\overline{n}} \frac{\sum_{k=1}^{S} \psi_k(\mathbf{p}) \overline{Y}_k^{(j)}}{z - \sum_{k=1}^{S} \psi_k(\mathbf{p}) \overline{\lambda}_k^{(j)}}. \tag{3.3}$$

Note that, in order for the frequency surrogates to be interpolated exactly, the weights should satisfy the conditions

$$\psi_\ell(\mathbf{p}_k) = \begin{cases} 1 & \text{if } k = \ell, \\ 0 & \text{if } k \neq \ell, \end{cases} \tag{3.4}$$

cf. Section 3.3.

Remark 3.1. Depending on the choice of MOR method in frequency, the Heaviside formulation (3.1) might also include constant, polynomial, or, more generally, smooth terms: for instance,

$$\mathrm{ROM}_k(z) = \sum_{j=1}^{\overline{n}_k} \frac{\overline{Y}_k^{(j)}}{z - \overline{\lambda}_k^{(j)}} + \sum_{\ell=0}^{\overline{m}} \overline{W}_k^{(\ell)} z^\ell. \tag{3.5}$$

If this is the case, we do not need to modify our algorithm, because the additional terms do not require to be matched nor permuted. The resulting global surrogate (after pole matching) is simply

$$\mathrm{pROM}(z,\mathbf{p}) = \sum_{j=1}^{\overline{n}} \frac{\sum_{k=1}^{S} \psi_k(\mathbf{p}) \overline{Y}_k^{(j)}}{z - \sum_{k=1}^{S} \psi_k(\mathbf{p}) \overline{\lambda}_k^{(j)}} + \sum_{\ell=0}^{\overline{m}} \sum_{k=1}^{S} \psi_k(\mathbf{p}) \overline{W}_k^{(\ell)} z^\ell.$$

3.1. Frequency adaptivity *via* Minimal Rational Interpolation

Here we provide some details on the function BuildFrequencyROM in Algorithm 3.1, which encodes the construction of a ROM for a non-parametric dynamical system with respect to a single parameter, namely the frequency z. A number of surrogate modeling strategies for such problems have been proposed in the MOR literature: the most famous are, among projection-based methods, the Proper Orthogonal Decomposition/Reduced Basis [9] and the Krylov/Moment Matching [8] methods, and, among the non-intrusive techniques, the Loewner Framework [24] and the Vector Fitting algorithm [15, 19]. Any of them could be used to supply the surrogate modeling that we require. In fact, different frequency surrogates could even be obtained by different MOR approaches, as long as each reduced model allows for a Heaviside expansion (3.1).

In this work, we consider the Minimal Rational Interpolation (MRI) method proposed in [34], generalizing [11]. MRI is non-intrusive, *i.e.*, it does not require access to the matrices appearing in the dynamical system (2.1), allowing for a wider applicability of the method. In particular, MRI can be used to efficiently build surrogates for vector-valued quantities, *e.g.*, high-(or even ∞-)dimensional states X of dynamical systems (2.1). (In fact, the effectiveness of MRI may even improve if the ambient space, where snapshots of the approximation target are located, is large.) At the same time, one can apply it in a greedy fashion [35], so that the number and location of frequency samples are selected adaptively, in such a way that a prescribed accuracy is attained over the whole frequency domain Z. The specific greedy strategy that we choose has a proper theoretical motivation only if the parametric problem depends on z in a simple way, *e.g.*, linearly, as in (2.1). If this is not the case, or if the dependence on frequency is not known, one may want to consider possible alternatives [35]. More details on the method are provided in [34]. Here we only give a short overview, which is summarized in Algorithm 3.3.

Algorithm 3.3 Greedy Minimal Rational Interpolation

function BUILDFREQUENCYROM($\mathbf{p}$)
  Initialize $Z_{\text{train}} \subset Z$ (very coarse), $Z_{\text{test}} \subset Z$ (fine), tol > 0
  snapshots $\leftarrow \{\,\}$
  for $z \in Z_{\text{train}}$ **do** ▷ Model initialization
    Evaluate $Y(z, \mathbf{p})$ ▷ Full model solve
    snapshots $\leftarrow$ snapshots $\cup \{Y(z, \mathbf{p})\}$
  end for
  repeat ▷ Greedy loop
    ROM$\leftarrow$ BUILDMRI(Z_{train}, snapshots) ▷ SVD of a matrix of size $\#Z_{\text{train}}$[34]
    $z^\star \leftarrow$ GETNEXTSAMPLEPOINT(ROM, Z_{test}) ▷ Maximize (3.7) over Z_{test}
    Evaluate $Y(z^\star, \mathbf{p})$ ▷ Full model solve
    snapshots $\leftarrow$ snapshots $\cup \{Y(z^\star, \mathbf{p})\}$
    Move $z^\star$ from Z_{test} to Z_{train}
  until $\|Y(z^\star, \mathbf{p}) - \text{ROM}(z^\star)\|_V \le \text{tol}\, \|Y(z^\star, \mathbf{p})\|_V$
  Optional: ROM$\leftarrow$ BUILDMRI(Z_{train}, snapshots) ▷ Remark 3.2
  Optional: ROM$\leftarrow$ CLEANUP(ROM) ▷ Remark 3.3
  return ROM
end function

Let a parameter $\mathbf{p} \in \boldsymbol{\mathcal{P}}$ be fixed. Given S distinct frequency sample points $Z_{\text{train}} \subset Z$ and corresponding snapshots $\{Y(z, \mathbf{p})\}_{z \in Z_{\text{train}}}$, the MRI procedure first builds a surrogate denominator Q of degree $\le S-1$ by solving a minimization problem involving the snapshots: in practice, this problem can be solved at $\mathcal{O}(S^3)$ computational cost by SVD. Now the surrogate poles $\{\overline{\lambda}^{(j)}\}_{j=1}^{\overline{n}}$ ($\overline{n} \le S-1$ is the degree of Q) appearing in (3.5) can be extracted from Q through any root-finding algorithm[2]. Then we set $\overline{m} = S - \overline{n} - 1$, and we compute the S terms $\{\overline{Y}^{(j)}\}_{j=1}^{\overline{n}} \cup \{\overline{W}^{(\ell)}\}_{\ell=0}^{\overline{m}}$ in the ROM (3.5) by interpolation:

$$\sum_{j=1}^{\overline{n}} \frac{\overline{Y}^{(j)}}{z - \overline{\lambda}^{(j)}} + \sum_{\ell=0}^{\overline{m}} \overline{W}^{(\ell)} z^\ell = Y(z, \mathbf{p}) \qquad \forall z \in Z_{\text{train}}. \tag{3.6}$$

The procedure just described is encoded by the function BUILDMRI in Algorithm 3.3.

The adaptive selection of frequency sample points is carried out *via* the typical greedy-MOR loop: at each iteration a new sample gets added, at a position selected in the test set Z_{test}, based on the current reduced model. More precisely, the next sample $z^\star$ is selected as the maximizer over Z_{test} of some *a posteriori* indicator, and the algorithm terminates when such indicator is smaller than a prescribed tolerance. However, in our framework, standard residual estimators (*e.g.*, the classic Reduced Basis one [36]) cannot be employed in a non-intrusive manner. Instead, we apply the "look-ahead" idea introduced in [35], to which we refer for more details:

- Exploiting only the current reduced model, we select the next sample point $z^\star$, as the maximizer over Z_{test} of

$$z \mapsto \prod_{z' \in Z_{\text{train}}} |z - z'| \Big/ \prod_{j=1}^{\overline{n}} \left| z - \overline{\lambda}^{(j)} \right|. \tag{3.7}$$

- Since an unknown scaling constant is involved, (3.7) does not tell us whether the prescribed tolerance is satisfied; to obtain this information, we perform an expensive solve of the full model at $z^\star$, and evaluate the approximation error explicitly.

Remark 3.2. As is evident from Algorithm 3.3, the extra expensive solve of the full system does not go to waste, as it is exactly the snapshot which is necessary at the next iteration. Only at the last iteration, when

[2]We are assuming all poles to be simple, *cf.* Section 4.2.

the tolerance is finally satisfied, the additional solution is not included in the surrogate. In practice, to avoid wasting the "free" final snapshot, we can actually add even this last sample to the reduced model by running BUILDMRI an additional time after the greedy loop is completed.

Remark 3.3. As discussed in more detail in Section 3.2, for a reliable matching of poles and residues of different frequency ROMs, it is crucial to remove any unwanted spurious (sometimes referred to as "parasitic" or "non-physical") pole-residue pairs. As such, it is usually worth the effort to add a post-processing "clean-up" step, to remove unwanted pole-residue pairs. The development of reliable strategies to identify spurious effects remains an open problem in rational approximation. However, two quite simple and inexpensive criteria that can be applied are:

- Remove poles which are too far away from Z.
- Remove poles whose residues have small magnitude (possible Froissart doublets [7]).

In addition, it might be possible to remove poles which are provably spurious based on the properties of the problem, *e.g.*, non-real poles for self-adjoint problems, or unstable poles for stable systems.

Any pole removed in this context should be automatically balanced by increasing the number of smooth terms in the ROM, namely $\overline{m}$ in Remark 3.1. We refer to [34] for a more detailed discussion on this procedure, which, employing the notation therein, could be summarized as "choosing $N < S - 1$". Thanks to this strategy, if the eliminated poles are indeed spurious, their removal *does not* impact negatively the accuracy of the frequency surrogate.

At this point, it is important to note that, in general, we cannot guarantee all the frequency ROMs to have the same number of pole-residue pairs $\overline{n}$, even after a perfect removal of all the spurious effects. The main reason for this is that the full model (2.1) may have a different number of poles in Z for different values of $\mathbf{p}$. A numerical example showcasing poles entering and leaving the frequency domain can be found in Section 5.1. This can quickly become problematic, since the greedy MRI algorithm is not guaranteed to identify well poles outside Z (actually, even to capture them at all). One could try to counteract this issue by building frequency models on an enlarged domain $Z' \supset Z$, at the cost of a higher offline time. Still, the same issue of migrating poles might simply arise again at a larger scale.

3.2. Matching frequency models

The offline matching loop in Algorithm 3.1 takes care of permuting the Heaviside addends of the frequency ROMs, see (3.1). The objective of this step is to identify with good accuracy how the location of each pole-residue pair evolves as $\mathbf{p}$ changes: this means that, for each j, we wish $\overline{\lambda}_1^{(j)}, \overline{\lambda}_2^{(j)}, \dots, \overline{\lambda}_S^{(j)}$ (resp. $\overline{Y}_1^{(j)}, \overline{Y}_2^{(j)}, \dots, \overline{Y}_S^{(j)}$) to be approximations of the "same" pole (resp. residue) of the parametric problem for $\mathbf{p} = \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_S$. By "same" pole (resp. residue) we mean some continuous curve $\mathbf{p} \mapsto \lambda_{\mathbf{p}}^{(j')}$ (resp. $\mathbf{p} \mapsto Y_{\mathbf{p}}^{(j')}$) for some fixed j', with $\lambda_{\mathbf{p}}^{(j')}$ and $Y_{\mathbf{p}}^{(j')}$ defined in Section 2. Note that this notion is not unique whenever poles intersect. This case is discussed in Section 4.

The frequency surrogate models are explored in a breadth-first fashion, matching one new ROM at a time to its closest (with respect to $\mathbf{p}$) neighbor. We exclude a global approach, matching all frequency models at the same time, due to its computational unfeasibility: S-partite matching is notoriously an NP-hard problem for $S \geq 3$ [25].

Let us characterize one of this bipartite matching problems, namely the one between ROM_k and ROM_ℓ. For the remainder of this section, we assume that the two models have the same number of poles, *i.e.*, $\overline{n}_k = \overline{n}_\ell =: \overline{n}$. The more general and, due to our choice of frequency surrogate modeling, (potentially) more common case $\overline{n}_k \neq \overline{n}_\ell$ is discussed in Section 3.2.1. We seek $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_{\overline{n}})$ which attains

$$\min_{\boldsymbol{\sigma} \in (1,2,\dots,\overline{n})!} \left(\sum_{j=1}^{\overline{n}} \left| \overline{\lambda}_k^{(j)} - \overline{\lambda}_\ell^{(\sigma_j)} \right| + w \sum_{j=1}^{\overline{n}} \left\| \overline{Y}_k^{(j)} - \overline{Y}_\ell^{(\sigma_j)} \right\|_V \right), \tag{3.8}$$

where $(1,2,\dots,\overline{n})!$ denotes the set of permutations of the tuple $(1,2,\dots,\overline{n})$. We will refer to the minimal value of (3.8) as the *Heaviside distance*[3] between models ROM_k and ROM_ℓ. In (3.8), $w \in [0,\infty]$ denotes a weight, representing the relative importance given to poles and residues in the matching. We note that a cost functional almost identical to (3.8) was first introduced in [43] to tackle the pole-residue matching.

An alternative formulation of the same problem is: given the cost matrix $D \in \mathbb{R}^{\overline{n}\times\overline{n}}$, with entries

$$D_{j,j'} = \left|\overline{\lambda}_k^{(j)} - \overline{\lambda}_\ell^{(j')}\right| + w\left\|\overline{Y}_k^{(j)} - \overline{Y}_\ell^{(j')}\right\|_V, \tag{3.9}$$

we wish to extract exactly one value per row and one value per column, so that the sum of the selected entries is minimized. Despite the combinatorial nature of this optimization problem, there exist polynomial-time algorithms to solve it (*e.g.*, using ideas from maximum flow problems [14]). Here, we solve this problem *via* the `linear_sum_assignment` function in the `scipy.optimize` module [40], which requires as only input the cost matrix D. As shown in [14, Section II.C], this implementation has $\mathcal{O}(\overline{n}^3)$ worst-case complexity, but we note that only $\mathcal{O}(\overline{n}^2)$ operations are necessary if just $\mathcal{O}(1)$ pole swaps are needed, *i.e.*, if the optimal $\boldsymbol{\sigma}$ is "close" to $(1,2,\dots,\overline{n})$.

Remark 3.4. When $\overline{n}_k = \overline{n}_\ell$, the matching optimization problem is symmetric. By this, we mean that, once the optimal permutation has been found, we can either apply it to the "new" model ROM_ℓ, or apply its inverse to the "old" model ROM_k. In practice, in the scope of the breadth-first search in Algorithm 3.1, it is convenient to always rearrange the poles-residues of the new model ℓ, since applying the permutation to model k would require permuting *all* the already explored models $\{\mathrm{ROM}_{k'}\}_{k'\in J}$.

3.2.1. Unbalanced matching

Suppose $\overline{n}_k < \overline{n}_\ell$ (the converse case can be treated analogously by considering a "transposed" matching problem). We can cast the matching optimization problem in rectangular form: find $\boldsymbol{\sigma} = (\sigma_1,\dots,\sigma_{\overline{n}_\ell})$ which attains

$$\min_{\boldsymbol{\sigma}\in(1,2,\dots,\overline{n}_\ell)!}\left(\sum_{j=1}^{\overline{n}_k}\left|\overline{\lambda}_k^{(j)} - \overline{\lambda}_\ell^{(\sigma_j)}\right| + w\sum_{j=1}^{\overline{n}_k}\left\|\overline{Y}_k^{(j)} - \overline{Y}_\ell^{(\sigma_j)}\right\|_V\right). \tag{3.10}$$

(The minimization of (3.10) does not require more computational effort than a balanced problem (3.8) of size $\overline{n}_\ell$ [14].) Then $(\sigma_1,\dots,\sigma_{\overline{n}_k})$ gives the desired permutation, while the indices $\sigma_{\overline{n}_k+1},\dots,\sigma_{\overline{n}_\ell}$ remain unassigned. It remains to choose how to deal with the $\overline{n}_\ell - \overline{n}_k$ extra pole-residue pairs with indices $\sigma_{\overline{n}_k+1},\dots,\sigma_{\overline{n}_\ell}$: this choice depends on whether we think that they are (○) missing from ROM_k or (●) spurious in ROM_ℓ.

The solution for case (●) is quite straightforward: we simply remove the $\overline{n}_\ell - \overline{n}_k$ erroneous poles and residues from ROM_ℓ, as well as from all the surrogates matched to it. Note, however, that the $\overline{n}_k$ remaining residues of ROM_ℓ should be updated after the removal of the spurious poles, in order to guarantee good approximation properties at $\mathbf{p}_\ell$. To this aim, it is quite cheap to recompute the residues of ROM_ℓ by solving an interpolation problem, *cf.* (3.6), exploiting the already available snapshots at $\mathbf{p}_\ell$. In particular, the approach described in Remark 3.3, *i.e.*, the addition of smooth terms to compensate for the removed poles/residues, can be applied here.

Instead, in case (○), the problem is much harder: we wish to reconstruct the poles and residues unaccounted for at $\mathbf{p}_k$ from information at $\mathbf{p}_\ell$. One naive way to achieve this involves "copying" the missing poles and residues from ROM_ℓ to ROM_k, *i.e.*, appending the synthetic terms

$$\frac{\overline{Y}_\ell^{(\sigma_{\overline{n}_k+1})}}{z - \overline{\lambda}_\ell^{(\sigma_{\overline{n}_k+1})}} + \dots + \frac{\overline{Y}_\ell^{(\sigma_{\overline{n}_\ell})}}{z - \overline{\lambda}_\ell^{(\sigma_{\overline{n}_\ell})}} \tag{3.11}$$

[3]When smooth terms are added, see (3.5), or in the unbalanced case $\overline{n}_k \neq \overline{n}_\ell$, see (3.10), this is not an actual distance because it is only semi-positive definite.

Algorithm 3.4 General pole-matching

$J \leftarrow \{1\}$ ▷ Initialize root of search tree, Remark 3.6
for iter $= 2, \dots, S$ **do** ▷ Matching loop
  $(k, \ell) \leftarrow \arg\min_{k' \in J,\, \ell' \notin J} \|\mathbf{p}_{k'} - \mathbf{p}_{\ell'}\|$ ▷ Breadth-first search
  $\boldsymbol{\sigma} \leftarrow$ ModalMatching(ROM$_k$, ROM$_\ell$) ▷ (3.8) or (3.10)
  ROM$_\ell \leftarrow$ApplyPermutation(ROM$_\ell$, $\boldsymbol{\sigma}$) ▷ Possibly using only part of $\boldsymbol{\sigma}$
  if $\overline{n}_k > \overline{n}_\ell$ **then**
    ROM$_\ell \leftarrow$ ROM$_\ell + \sum_{j=\overline{n}_\ell+1}^{\overline{n}_k} \overline{Y}_k^{(j)} / \left(\cdot - \overline{\lambda}_k^{(j)}\right)$ ▷ Extrapolate at index ℓ
    Flag the $\overline{n}_k - \overline{n}_\ell$ poles added to ROM$_\ell$ as "synthetic"
    $\overline{n}_\ell \leftarrow \overline{n}_k$
  else if $\overline{n}_k < \overline{n}_\ell$ **then**
    for $i \in J$ **do** ▷ Extrapolate at indices J
      ROM$_i \leftarrow$ ROM$_i + \sum_{j=\overline{n}_i+1}^{\overline{n}_\ell} \overline{Y}_\ell^{(j)} / \left(\cdot - \overline{\lambda}_\ell^{(j)}\right)$
      Flag the $\overline{n}_\ell - \overline{n}_i$ poles added to ROM$_i$ as "synthetic"
      $\overline{n}_i \leftarrow \overline{n}_\ell$
    end for
  end if
  $J \leftarrow J \cup \{\ell\}$ ▷ Add to explored set
end for
for $j = 1, \dots, \max_{j \in J} \overline{n}_j$ **do**
  synthetic_count $\leftarrow \#\{k = 1, 2, \dots, S$ such that $\overline{\lambda}_k^{(j)}$ is synthetic$\}$
  if synthetic_count $> S(1 - \text{tol}_{\text{synth}})$ **then**
    Remove $\overline{\lambda}^{(j)}$ from all ROMs ▷ Remove poles which are synthetic too many times
  end if
end for
Optional: apply higher-order reconstruction to all remaining synthetic poles ▷ Remark 3.7

to the Heaviside expansion of ROM$_k$. This results in an adjusted surrogate, with a balanced pole matching. Note that, similarly to pole removal, the $\overline{n}_k$ original residues of ROM$_k$ should be updated to account for the added synthetic terms. We remark that more refined (but also potentially less stable) approaches may be based on other forms of extrapolation of the additional Heaviside terms, not only from ROM$_\ell$, but also from all the other frequency surrogates that contain the poles with indices $\sigma_{\overline{n}_k+1}, \dots, \sigma_{\overline{n}_\ell}$, *cf.* Remark 3.7 and Section 3.3.

Remark 3.5. In case (∘), one could impose a balanced matching by retraining the poorer model ROM$_k$, forcing more iterations of greedy MRI until the surrogate has exactly $\overline{n}_\ell$ poles. However, this approach has two clear issues, which might make it disadvantageous in practice:

- If many surrogates need retraining, one may incur in substantial additional computational cost. This is the case especially if **p**-adaptivity is employed, see Section 3.4.
- Instead of identifying correctly the missing poles, MRI could introduce spurious effects, thus interfering with the matching procedure, rather than helping it. This is likely to happen if the missing poles are located outside the frequency domain Z.

At this point, we deem important to give a *caveat*: neither of the two approaches above (pole reconstruction and removal) is, on its own, able to solve adequately all practical situations, as we showcase in a practical example in Section 5.1. Instead, a hybrid version, with some poles getting removed and some reconstructed, has the potential to perform better than either approach. Here, we choose to pursue this third approach, whose effectiveness obviously hinges on how well we can differentiate between "good" and "bad" poles.

We summarize our proposed strategy in Algorithm 3.4, which replaces the simple matching loop in Algorithm 3.1. The main idea is the following: whenever it becomes necessary to match unbalanced models, the naive reconstruction (3.11) is used to augment the less rich surrogate(s); however, the added synthetic poles

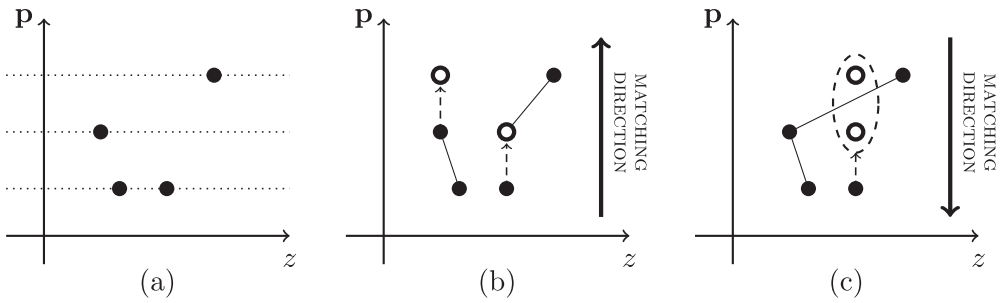

FIGURE 1. Example of history-dependent matching. Poles of the surrogates are denoted by full dots (a). Results by matching bottom-up (b) and top-down (c). Dashed arrows and empty circles are used to denote pole reconstruction steps and synthetic poles, respectively.

are flagged as unreliable. At the end of the matching loop, all the models contain the same number of poles $\overline{n} = \max_{k=1,\dots,S} \overline{n}_k$. At this point, if a pole with a certain index is too often unreliable, it is removed from the pROM. Here, "too often" is determined based on a given tolerance $\mathrm{tol}_{\mathrm{synth}}$ between 0 and 1: the extreme values 0 and 1 correspond to cases (○) and (●), respectively.

The reconstruction of missing poles introduces an asymmetry in the matching procedure, so that the order in which the models are matched matters: for instance, see the situation depicted in Figure 1. As such, it seems important to choose well the root in the breadth-first exploration of $\boldsymbol{\mathcal{P}}_{\mathrm{train}}$. However, choosing optimally this root remains an open problem. From a computational point of view, it makes sense to choose as root the surrogate with the largest number of poles, so that we never have to retrace our steps to add synthetic poles.

Remark 3.7. The constant pole reconstruction (3.11) is quite blunt, especially when the parameter resolution is low. If the poles depend smoothly on $\mathbf{p}$, it is preferable to employ a reconstruction with a larger stencil, for instance global (least squares) polynomial extrapolation, using information from all the surrogates which contain the missing pole, see Section 3.3. However, this is not always viable during the matching loop, since we explore $\boldsymbol{\mathcal{P}}_{\mathrm{train}}$ breadth-first: for instance, if $\overline{n}_k < \overline{n}_\ell$, we are forced to reconstruct poles from the single new model ROM_ℓ (a similar problem may arise in the case $\overline{n}_k > \overline{n}_\ell$ if J is too small). Still, as shown in the last line of Algorithm 3.4, it remains feasible to apply a higher-order pole reconstruction after the matching loop is complete.

3.3. Global *vs.* local interpolation

In this section we describe how one can employ the local pole-residue information (3.1) at $\boldsymbol{\mathcal{P}}_{\mathrm{train}}$ to obtain an approximate Heaviside expansion at a new point $\mathbf{p} \in \boldsymbol{\mathcal{P}} \setminus \boldsymbol{\mathcal{P}}_{\mathrm{train}}$. As shown in (3.2), we rely on an interpolation strategy encoded by the weight functions $\boldsymbol{\psi} : \boldsymbol{\mathcal{P}} \to \mathbb{C}^S$, so that we simply need to prescribe how such weights are constructed. It is important to note that this task, in its natural formulation, is independent of the frequency surrogates, and depends only on the location of the sample parameter points $\boldsymbol{\mathcal{P}}_{\mathrm{train}}$, *cf.* Remark 3.8.

Setting (3.4) as target, we can cast the problem of finding $\boldsymbol{\psi}$ as S independent $\boldsymbol{\mathcal{P}} \to \mathbb{C}$ interpolation problems (one for each ψ_ℓ). To address this problem, an extensive amount of techniques and results are available in the literature [13,33,42]. Here, we consider 3 options:

– **Global.** We can seek weights $\boldsymbol{\psi}$ within some function space with global regularity, *e.g.*, polynomials, or radial basis functions (using a smooth kernel to achieve interpolation). This approach can potentially achieve high accuracy, but relies on some level of smoothness of poles and residues with respect to $\mathbf{p}$. It is important to note that, in this framework, the interpolation condition (3.4) could be weakened and enforced only in a least-squares sense: the resulting surrogate would then require less memory for the storage of approximate poles and residues, *cf.* the "regression" step in [43].

– **Local structured.** To satisfy (3.4), we can employ locally supported basis functions, *e.g.*, piecewise linear "hat functions", or splines. If $d \geq 2$, this approach requires the sample points to be selected in a structured way, for instance using sparse grids, see [5] and Section 3.4, or a mesh-based discretization of $\boldsymbol{\mathcal{P}}$ [17].
– **Local unstructured.** A very simple, but nonetheless practical, way to enforce (3.4) is to construct the weights using a Voronoi tessellation [16] of $\boldsymbol{\mathcal{P}}$ based on the sample points $\boldsymbol{\mathcal{P}}_{\text{train}}$, *i.e.*,

$$\psi_\ell(\mathbf{p}) = \begin{cases} 1 & \text{if } \mathbf{p}_\ell = \arg\min_{\mathbf{p}^\star \in \boldsymbol{\mathcal{P}}_{\text{train}}} \|\mathbf{p} - \mathbf{p}^\star\|, \\ 0 & \text{otherwise.} \end{cases} \tag{3.12}$$

This results in a nearest-neighbor approach, characterized by low accuracy, but also by a great flexibility. In fact, this strategy does not even require the poles to be matched. The main drawback of this approach is that it does not "follow" the evolution of the poles in $\mathbf{p}$-space, resulting in limited predictive capabilities.

Remark 3.8. Depending on the application, the interpolation conditions (3.4) may be complemented by additional constraints to preserve important system properties, like stability, realness, or passivity [17, 19].

3.4. Parameter adaptivity

Until now, we have assumed the parameter sample points $\boldsymbol{\mathcal{P}}_{\text{train}}$ to have been fixed in advance. However, in many situations, it proves extremely useful to have some kind of adaptivity included in the sampling of $\boldsymbol{\mathcal{P}}$, so that samples may be added only where the surrogate model is particularly inaccurate, *e.g.*, in our case, near pole mismatches or where large interpolation errors occur. Still, it is quite difficult to devise adaptive strategies in non-intrusive MOR, especially if the number of parameters d is large, since not much is known about the parametric dependence of the problem. In the context of sampling from high-dimensional parametric spaces, *sparse grids* have been often employed in MOR as a way to alleviate the curse of dimensionality, see, *e.g.*, [6, 12, 21]. Here, the focus is on adaptive sampling, and we propose a technique based on *locally-refined sparse grids*, closely related to that considered in [1], which, in turn, relies on some ideas from [28, 32]. For simplicity, we carry out our construction in the case $\boldsymbol{\mathcal{P}} = [-1, 1]^d$. Generalizations to more complicated parameter domains may be obtained by isomorphism.

Consider the nested ($\Gamma(n) \subseteq \Gamma(n+1)$ for all n) one-dimensional point sets

$$\Gamma(n) = \begin{cases} \emptyset & \text{if } n < 0, \\ \{0\} & \text{if } n = 0, \\ \{2^{1-n} j\}_{j=-2^{n-1}}^{2^{n-1}} & \text{if } n > 0. \end{cases} \tag{3.13}$$

We extend this definition to multiple dimensions by tensorization: for any *level index* $\mathbf{n} = (n_1, \ldots, n_d) \in \mathbb{Z}^d$, we define the corresponding *tensor grid* $\Gamma(\mathbf{n}) = \Gamma(n_1) \times \Gamma(n_2) \times \ldots \times \Gamma(n_d)$. It is useful to define the infinite point set

$$\boldsymbol{\Phi} = \bigcup_{\mathbf{n} \in \mathbb{Z}^d} \Gamma(\mathbf{n}) = \bigcup_{n=0}^{\infty} \Gamma(n)^d,$$

which is dense in $\boldsymbol{\mathcal{P}}$ (it coincides with the dyadic rationals in $\boldsymbol{\mathcal{P}}$) and also a superset of any tensor grid (by construction). We will choose the adaptive sampling points within $\boldsymbol{\Phi}$.

Now, assume that

$$\mathbf{p}^\star \in \Gamma(\mathbf{n}) \setminus \left(\bigcup_{k=1}^{d} \Gamma(n_1, \ldots, n_{k-1}, n_k - 1, n_{k+1}, \ldots, n_d) \right), \tag{3.14}$$

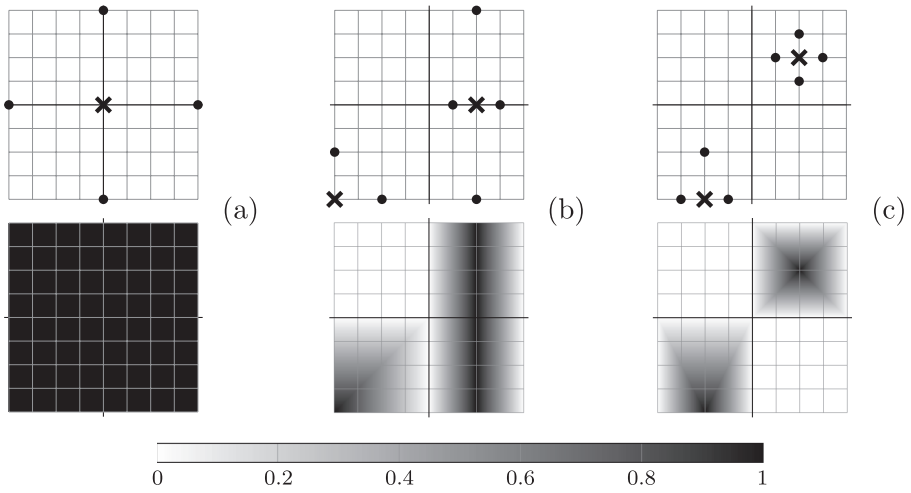

FIGURE 2. In the *top row*, the forward points of $(0,0) \in \Gamma(0,0)$ in (a), $(-1,-1) \in \Gamma(1,1)$ and $(1/2,0) \in \Gamma(2,0)$ in (b), and $(-1/2,-1) \in \Gamma(2,1)$ and $(1/2,1/2) \in \Gamma(2,2)$ in (c). In the bottom row, the corresponding hierarchical hat functions.

i.e., that the coordinates of $\mathbf{p}^\star = \left(j_1/2^{n_1-1}, \ldots, j_d/2^{n_d-1}\right) \in \boldsymbol{\Phi}$ are fractions in lowest terms (with $n_k = 0$ if $j_k = 0$). We define the *forward points* of $\mathbf{p}^\star$ as the $(\leq 2d)$ elements of the discrete neighborhood

$$\begin{aligned} U(\mathbf{p}^\star) &= \bigcup_{k=1}^{d} \left\{ \mathbf{p} \in \Gamma(n_1, \ldots, n_{k-1}, n_k+1, n_{k+1}, \ldots, n_d),\ \|\mathbf{p} - \mathbf{p}^\star\| = 2^{-n_k} \right\} \\ &= \boldsymbol{\mathcal{P}} \cap \bigcup_{k=1}^{d} \left\{ \left(p_1^\star, \ldots, p_{k-1}^\star, p_k^\star \pm 2^{-n_k}, p_{k+1}^\star, \ldots, p_d^\star\right) \right\}. \end{aligned}$$

Moreover, to each $\mathbf{p}^\star$ satisfying (3.14), we associate a *hierarchical hat function* $\varphi_{\mathbf{p}^\star} : \boldsymbol{\mathcal{P}} \to [0,1]$ according to the definition

$$\varphi_{\mathbf{p}^\star}(\mathbf{x}) = \prod_{k=1}^{d} \widehat{\varphi}_{p_k^\star, n_k}(x_k), \tag{3.15}$$

with $\widehat{\varphi}_{p,0}(x) \equiv 1$ and, for $n = 1, 2, \ldots$,

$$\widehat{\varphi}_{p,n}(x) = \begin{cases} 1 - 2^{n-1}\,|x-p| & \text{if } |x-p| < 2^{1-n}, \\ 0 & \text{if } |x-p| \geq 2^{1-n}. \end{cases}$$

By construction, $\varphi_{\mathbf{p}^\star}$ is zero at all $\mathbf{p}' \in \boldsymbol{\Phi}$ of which $\mathbf{p}^\star$ is a forward point (the *backward points* of $\mathbf{p}^\star$), and also at all backward points $\mathbf{p}'' \in \boldsymbol{\Phi}$ of such $\mathbf{p}'$, etc., all the way back to $\mathbf{p} = \mathbf{0}$. We show some two-dimensional examples of forward points and of hierarchical hat functions in Figure 2.

We rely on hierarchical hat functions to cast piecewise-linear interpolation problems over subsets of sparse grids. More precisely, given sample points $\boldsymbol{\mathcal{P}}^\star \subset \boldsymbol{\Phi}$, and data $\{f(\mathbf{p}^\star)\}_{\mathbf{p}^\star \in \boldsymbol{\mathcal{P}}^\star}$, the *piecewise-linear interpolant of f based on samples at $\boldsymbol{\mathcal{P}}^\star$* is the unique element $\widehat{f}_{\boldsymbol{\mathcal{P}}^\star}$ of $\operatorname{span}\{\varphi_{\mathbf{p}^\star}\}_{\mathbf{p}^\star \in \boldsymbol{\mathcal{P}}^\star}$ which interpolates exactly the data: this means that there exist unique coefficients $\{c_{\mathbf{p}^\star}\}_{\mathbf{p}^\star \in \boldsymbol{\mathcal{P}}^\star}$, depending only on $\boldsymbol{\mathcal{P}}^\star$ and $\{f(\mathbf{p}^\star)\}_{\mathbf{p}^\star \in \boldsymbol{\mathcal{P}}^\star}$, such that

$$f(\mathbf{p}^\star) = \widehat{f}_{\boldsymbol{\mathcal{P}}^\star}(\mathbf{p}^\star) = \sum_{\mathbf{p}' \in \boldsymbol{\mathcal{P}}^\star} c_{\mathbf{p}'} \varphi_{\mathbf{p}'}(\mathbf{p}^\star) \qquad \forall \mathbf{p}^\star \in \boldsymbol{\mathcal{P}}^\star. \tag{3.16}$$

The desired Lagrangian basis (3.4) can then be found by setting the data as $\{f(\mathbf{p}^\star) = \delta_{\mathbf{p}^\star \mathbf{p}_l}\}_{\mathbf{p}^\star \in \boldsymbol{\mathcal{P}}^\star}$, with δ the Kronecker delta. We note that the expression of each hierarchical basis function (3.15) depends only on its support point $\mathbf{p}^\star \in \boldsymbol{\mathcal{P}}^\star$, whereas the expression of each Lagrangian basis function (3.4) depends on the whole support set $\boldsymbol{\mathcal{P}}^\star$.

Now we are ready to describe our adaptive technique, which is summarized in Algorithm 3.5. As in the greedy selection of frequency samples (Algorithm 3.3), the adaptivity is achieved through a "look-ahead" idea, although here the approach is rather heuristic: we use the forward points of the current training set as test set, *i.e.*, parameter values at which the accuracy of the current pROM is evaluated. If the surrogate model is too inaccurate at some of the test points, they are added to the training set. This loop is repeated until a specified tolerance is achieved at all current test points.

This approach, differently from the usual *isotropic* adaptive sparse grid sampling [5, 29], in general does not add whole levels $\Gamma(\mathbf{n})$, but only subsets of them. In fact, the training set is not even guaranteed to be *downward-closed*, *i.e.*, a point might be in the training set while some of its backward points are not. The matter of missing backward points is discussed to some detail in [1, Section 3.2]. In the remainder of our presentation and in our numerical experiments, we do *not* require missing backward points to be added to the training set, both for simplicity of exposition and (mainly) to reduce the cost of the offline phase[4]. We remark that we are allowed to work with a non-downward-closed training set because the error estimator driving our **p**-adaptivity does not rely on interpreting the expansion coefficients $c_{\mathbf{p}'}$ in (3.16) as "hierarchical surpluses", as is commonly done in adaptive sparse grids [28]. As a side note, we observe that not including the backward points makes it necessary to recompute the expansion coefficients (3.16) from scratch whenever new training points are added.

Within each iteration, in order to quantify the accuracy of the pROM at a test parameter value **p**, we use the following strategy:

(a) We use the current pROM (whose training set does not include **p**, nor any of the other test points) to predict the frequency response at **p**, using (3.3), where the weight functions ψ_k are hierarchical hat functions (3.15) built by piecewise-linear interpolation (3.16), *cf.* Remark 3.9.
(b) Through Algorithm 3.3, we build a frequency surrogate at **p**, which we take as "truth frequency response" at **p**. This requires solving the full model at **p**, at as many frequency points as required by the z-greedy procedure.
(c) We compare poles and residues of the two models by employing (3.8) as distance; this requires the solution of a pole-matching problem.

For the sake of efficiency, it is crucial to observe that, over the different **p**-greedy iterations, function BUILD-FREQUENCYROM may be called multiple times with the same argument (not only when the pROM is built through PROM_TRAIN, see Algorithm 3.1, but also when evaluating the accuracy of the current model on the test set). As long as memory is not an issue, one should store frequency surrogates built at previous **p**-greedy steps, so that no expensive solve of the full model is wasted.

Remark 3.9. In (3.4), we have forced our pROM technique to reconstruct poles and residues only through piecewise-linear hat functions. However, as discussed towards the end of Section 3.3, in some cases one may want to employ a matching-free nearest-neighbor reconstruction. This is easily achieved by using the piecewise constant basis (3.12) instead of hierarchical hat functions. In this case, to better account for the approximation properties of interpolation basis, it is natural to employ Haar-type sparse grids [13, 28], which can be obtained as in Section 3.4, replacing (3.13) by

$$\Gamma(n) = \begin{cases} \emptyset & \text{if } n \leq 0, \\ \{2^{1-n} j\}_{j=-2^{n-1}+1}^{2^{n-1}-1} & \text{if } n > 0. \end{cases}$$

This essentially corresponds to restricting the sparse grid points to the *interior* of $\boldsymbol{\mathcal{P}}$.

[4] If one can afford a higher offline time, including backward points is advisable. However, we note that the increase in training cost could be significant, since each sparse grid point in $\mathbb{R}^d$ has up to $2d$ backward points, and a (costly) frequency model must be built by MRI at each of them for error estimation.

Algorithm 3.5 Adaptive parameter sampling

function pROM_Adapt
  Initialize $\boldsymbol{\mathcal{P}}_{\text{train}} \subset \boldsymbol{\Phi}$ (with "few" elements), tol > 0
  repeat ▷ Greedy loop
    pROM$\leftarrow$ pROM_Train($\boldsymbol{\mathcal{P}}_{\text{train}}$) ▷ Algorithm 3.1
    $\boldsymbol{\mathcal{P}}_{\text{test}} \leftarrow$ ForwardPointsOf($\boldsymbol{\mathcal{P}}_{\text{train}}$) $\setminus \boldsymbol{\mathcal{P}}_{\text{train}}$ ▷ Update test set
    $\boldsymbol{\mathcal{P}}^{\text{new}}_{\text{train}} \leftarrow \{\ \}$
    for $\mathbf{p} \in \boldsymbol{\mathcal{P}}_{\text{test}}$ **do**
      $\overline{\text{ROM}} \leftarrow$ pROM$(\cdot, \mathbf{p})$ ▷ Predict surrogate at test point
      ROM$\leftarrow$ BuildFrequencyROM($\mathbf{p}$) ▷ Build new surrogate at test point
      if HeavisideDistance(ROM, $\overline{\text{ROM}}$) $>$ tol **then** ▷ Distance defined in (3.8)
        $\boldsymbol{\mathcal{P}}^{\text{new}}_{\text{train}} \leftarrow \boldsymbol{\mathcal{P}}^{\text{new}}_{\text{train}} \cup \{\mathbf{p}\}$
      end if
    end for
    $\boldsymbol{\mathcal{P}}_{\text{train}} \leftarrow \boldsymbol{\mathcal{P}}_{\text{train}} \cup \boldsymbol{\mathcal{P}}^{\text{new}}_{\text{train}}$
  until $\boldsymbol{\mathcal{P}}^{\text{new}}_{\text{train}} = \emptyset$
  Optional: pROM$\leftarrow$ pROM_Train($\boldsymbol{\mathcal{P}}_{\text{train}} \cup \boldsymbol{\mathcal{P}}_{\text{test}}$) ▷ Remark 3.12
  return pROM
end function

Whatever the reconstruction strategy in step (3.4), any of the methods presented in Section 3.3 can still be applied as a post-processing step, at the end of the greedy loop. The reason for this additional computation could be, for instance, a smoother representation of poles and residues, or the removal of eventual noise by regularization. To this aim, we wish to stress that some care should be used when selecting the pole-residue reconstruction strategy. Indeed, due to the local nature of the **p**-refinements, finer and coarser sampling regions may arise, which, if not taken into account, could lead to a poorly-behaving reconstruction. For instance, global polynomial interpolation over sampling points which are "too wild" can be an extremely ill-posed problem, due to a large Lebesgue constant [33], whereas polynomial regression with low enough degree can be expected to behave more nicely.

Remark 3.10. The strategy that we presented is heuristic. In particular, it does not guarantee that, at the end of the greedy loop, the tolerance will be attained over the whole parameter domain, since we are using a relatively small (and sparse) test set to quantify the approximation error. Representing ("sketching") the parameter domain by the test set can be justified only by assuming the resolution of the test set to be sufficiently fine. However, in practice, this is usually computationally unfeasible (especially if the number of parameters is large, due to the curse of dimensionality).

Remark 3.11. In [43], a somewhat similar p-adaptive approach was proposed, which, however, can be applied only to the single-parameter case. While the adaptivity there was essentially "unidirectional", adding samples progressively from one end of $\mathcal{P}$ to the other, here, through sparse grids, our approach acts more "isotropically".

Remark 3.12. As in the frequency-adaptivity, see Remark 3.2, once the greedy iterations are over, we can take advantage of the extra samples taken at test parameter points, and build a much richer pROM than the one that satisfied the tolerance constraint. Here, this idea is even more attractive than for MRI, since the test set can (and usually does) contain quite a large number of parameter values, as opposed to just 1.

4. Remarks on the smoothness of the Heaviside decomposition

Based on how smoothly the system matrices $\{A, B, C, E\}$ in (2.1) depend on $\mathbf{p}$, it is possible for the spectral quantities $\{\lambda^{(j)}, m^{(j)}, Y^{(j)}\}$ in (2.2) to depend smoothly on $\mathbf{p}$ as well. More precisely, continuous dependence is often passed on from matrices to Heaviside terms: small perturbations of the system matrices yield small perturbations of the poles $\lambda^{(j)}$ and of the residues $Y^{(j)}$, at least as long as multiplicities $m^{(j)}$ are independent

of **p** and poles do not cross. However, inheritance of analytic dependence cannot in general be guaranteed, since (polynomial) branches may naturally arise when poles cross. We refer to [37, Chapter 12] for an introductory discussion on the topic, and we report here two simple representative examples.

4.1. A toy example of mode steering

First, we showcase some of the intrinsic difficulties in dealing with crossing or almost crossing poles, even in the absence of bifurcations. The example below was obtained by generalizing a numerical test from [3].

For some fixed $\varepsilon \in \mathbb{R}$, set $d = 1$ and take

$$A_p = \begin{bmatrix} 2p & \varepsilon \\ \varepsilon & 0 \end{bmatrix}, \quad B_p = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \text{and} \quad C_p = E_p = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.1}$$

as the matrices defining a parametric dynamical system of the form (2.1), which depend smoothly on $p \in \mathbb{R}$ (and ε). The system output can be explicitly computed as

$$Y(z,p;\varepsilon) = \begin{bmatrix} z-2p & -\varepsilon \\ -\varepsilon & z \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{z^2 - 2pz - \varepsilon^2} \begin{bmatrix} z \\ p_2 \end{bmatrix} = \frac{Y_p^{(1)}}{z - \lambda_p^{(1)}} + \frac{Y_p^{(2)}}{z - \lambda_p^{(2)}},$$

where the poles and residues are

$$\lambda_p^{(1,2)} = p \pm \sqrt{p^2 + \varepsilon^2} \quad \text{and} \quad Y_p^{(1,2)} = \frac{1}{2} \begin{bmatrix} 1 \pm \frac{p}{\sqrt{p^2+\varepsilon^2}} \\ \pm \frac{\varepsilon}{\sqrt{p^2+\varepsilon^2}} \end{bmatrix}. \tag{4.2}$$

From (4.2), it is not difficult to conclude that, if $\varepsilon = 0$, the two poles coincide for $p = 0$. However, for fixed ε, this degeneracy does not have a negative impact on the smoothness of the Heaviside decomposition[5], since the poles remain simple:

$$Y(z,p;\varepsilon = 0) = \frac{1}{z - 2p} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{z} \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{4.3}$$

We illustrate how the pole-matching algorithm performs in this simple example, using the exact Heaviside expansion (4.2) in place of the frequency surrogates one would obtain, *e.g.*, *via* MRI. First, we fix $\varepsilon = 0$, and consider the two parameter values $p_1 = -1$ and $p_2 = 1$, where the Heaviside expansions of Y are

$$Y(z,p_1;0) = \frac{1}{z+2} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{z} \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \text{and} \quad Y(z,p_2;0) = \frac{1}{z-2} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \frac{1}{z} \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{4.4}$$

We depict poles and residues in Figure 3. If the matching is accurate, pole 0 should be matched with itself, and -2 with 2.

As described in Section 3.2, the matching criterion can be stated in terms of the cost matrix (3.9), which here equals

$$D = \begin{bmatrix} 4 & 2+w \\ 2+w & 0 \end{bmatrix}. \tag{4.5}$$

This means that matching -2 with 2 and 0 with itself has cost 4, whereas matching -2 with 0 and 0 with 2 has cost $4 + 2w$. Hence, as long as $w > 0$, the algorithm performs the correct matching and recovers the exact response. However, if $w = 0$, the two costs are the same, and any matching is allowed.

[5]Interestingly, the *joint* dependence on p and ε is non-smooth. Indeed, by comparing (4.3) and

$$Y(z,0;\varepsilon) = \frac{1/2}{z-\varepsilon} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \frac{1/2}{z+\varepsilon} \begin{bmatrix} 1 \\ -1 \end{bmatrix},$$

we observe that the residues are discontinuous at $(p,\varepsilon) = (0,0)$. However, it is important to note that, while discontinuous, the residues stay uniformly bounded, since the matrix A_p is real symmetric, hence diagonalizable, for all $(p,\varepsilon) \in \mathbb{R}^2$.

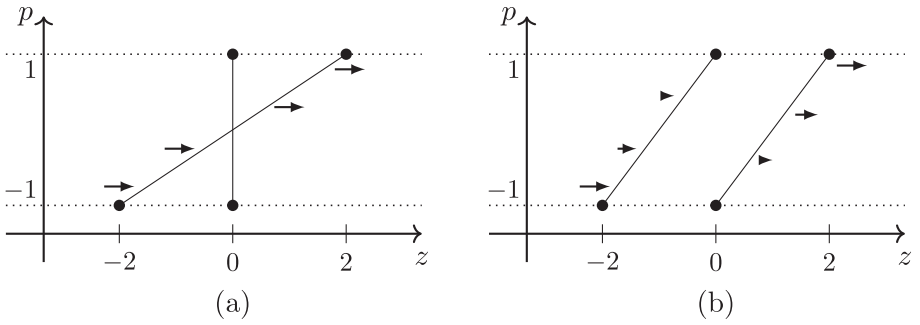

FIGURE 3. Example of pole-matching for $\varepsilon = 0$, with correct (a) and incorrect (b) matching; the residues at a few points, reconstructed by linear interpolation, are represented as small arrows. Missing arrows denote zero residues. Option (a) reconstructs the correct values exactly. Option (b) can happen only if $w = 0$.

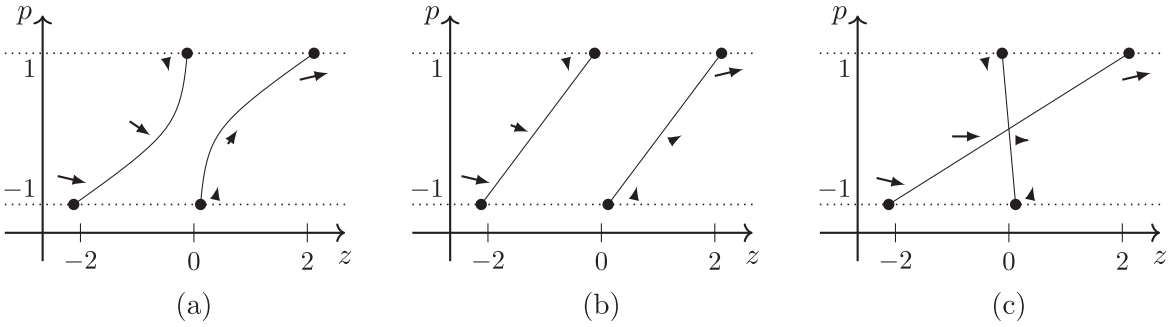

FIGURE 4. Example of pole-matching for $\varepsilon = 1/2$: exact solution (a), and pole-dominated (b) and residue-dominated (c) matching; the residues at a few points, reconstructed by linear interpolation for (b) and (c), are represented as small arrows.

The matching becomes less trivial if $\varepsilon \neq 0$. For instance, let $\varepsilon = 1/2$. We represent graphically poles and residues in Figure 4(a). By building the cost matrix (3.9) in this case (we omit the calculation here), we can conclude that the optimal matching changes depending on whether $w \gtrless 5 - 2\sqrt{5} \simeq 0.53$: if poles have more importance than residues (*i.e.*, w is small), the surrogate poles do not cross, see Figure 4(b), whereas they do if the weight of residues is dominant (*i.e.*, w is large), see Figure 4(c).

At least qualitatively, case (b) appears to be a slightly better approximation of the poles and residues of the system. Still, neither of the surrogates (b) and (c) identifies the pole-residue behavior in a satisfactory way, and only adding more sample points (starting from $p_3 = 0$) will allow for a significant improvement in the quality of the approximation.

4.2. A toy example of bifurcation

In this section we show a deceptively simple example of bifurcation, which, in practical applications, may arise due to unfavorable spectral properties of the problem (*e.g.*, local non-diagonalizability).

Fix a small ε, and consider the scalar problem with a single parameter

$$Y(z, p; \varepsilon) = \left(z^2 - p + \varepsilon\right)^{-1}. \tag{4.6}$$

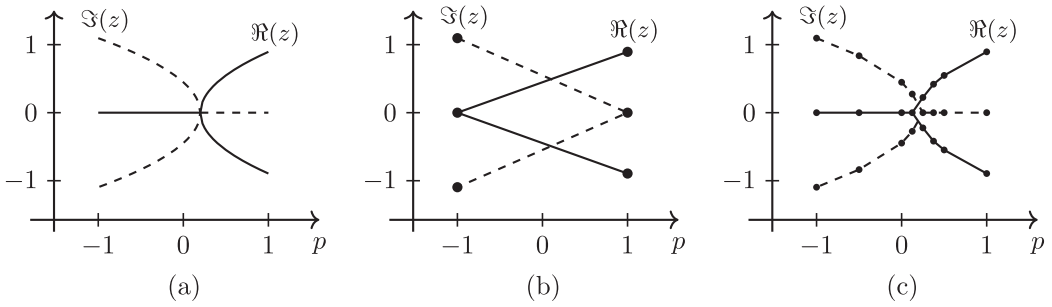

FIGURE 5. Real (full line) and imaginary (dashed line) parts of the exact roots of $z^2 - p + \varepsilon$, for $\varepsilon = 1/5$, with respect to $p \in [-1, 1]$ (a). Surrogate roots from combinations of 2 (b) and 8 (c) z-surrogates.

For $p \neq \varepsilon$, the corresponding Heaviside expansion is readily found:

$$Y(z, p; \varepsilon) = \frac{1/(2\sqrt{p-\varepsilon})}{z - \sqrt{p-\varepsilon}} - \frac{1/(2\sqrt{p-\varepsilon})}{z + \sqrt{p-\varepsilon}}, \tag{4.7}$$

whereas, for $p = \varepsilon$, $Y(z, \varepsilon; \varepsilon) = z^{-2}$, with a double pole at 0. We can see that the poles are non-smooth (as functions of p) at $p = \varepsilon$, where they exhibit a bifurcation of degree 2, see Figure 5(a). Moreover, the residues are not only discontinuous at $p = \varepsilon$, but also unbounded there. This is due to the poles transitioning from single to double.

We apply our pMOR approach in this simple example, assuming, for simplicity, that the surrogate poles and residues coincide with the exact ones. If only the two parameter values $p_1 = -1$ and $p_2 = 1$ are considered, the pROM will inevitably fail to identify $(p, z) = (\varepsilon, 0)$ as a branch point. Indeed, our method "follows" separately the evolution with respect to p of each pole. Consequently, the degree of each denominator in the Heaviside expansion (3.1) is kept equal to 1 even when crossing the singularity. The result, displayed in Figure 5(b), shows two surrogate pole lines which miss the branch point by "twisting" around it in the complex plane. Of course, as soon as more parameter points are added, the approximation quality improves significantly: in particular, if p-adaptivity is applied in this case, a progressively more accurate approximation of the exact poles is built as the greedy iterations proceed, see Figure 5(c).

It is important to remark that the structure of this kind of bifurcation would be much better identified by grouping together all of the involved branches, *i.e.*, by merging several terms of the Heaviside decomposition into a single fraction with denominator degree > 1 (here, 2) and coefficients smoothly dependent on the parameter, *cf.* the exact expression (4.6). This approach is implicitly applied by most methods based on global rational interpolation, *e.g.*, [18, 20, 24], which, in fact, do not rely on the Heaviside decomposition of the output (2.2), but on its rational form

$$Y(z, \mathbf{p}) = C_{\mathbf{p}} \left(z E_{\mathbf{p}} - A_{\mathbf{p}}\right)^{-1} B_{\mathbf{p}} = \frac{P(z, \mathbf{p})}{Q(z, \mathbf{p})},$$

where Q is a polynomial in z of degree $\leq n_S$, with $\mathbf{p}$-dependent coefficients. Procedures to integrate global rational interpolation concepts in our approach are being investigated, with the objective of improving the effectiveness in the approximation of non-smooth poles and residues. A possible idea revolves around a "batch-matching" of poles and residues (matching few-to-few instead of 1-to-1), which would allow an exact recovery of the simple bifurcation in the example above.

Before proceeding, we wish to briefly discuss what happens when a parameter sample point is added exactly at the branch point, which is the case, *e.g.*, if $\varepsilon = 0$ and we place a parameter sample at $p = 0$. In such situations, there are two possibilities:

– The frequency surrogate (built, *e.g.*, by MRI) identifies the double pole correctly. In this case, a simple Heaviside decomposition of the form (3.1) does not exist and our algorithm, as we presented it, fails. The batch-matching approach mentioned above (currently under investigation) would allow dealing in a natural way with this case.
– The frequency surrogate mistakenly identifies the double pole as a couple of very close simple poles, whose residues have a large magnitude as a result, *cf.* (4.7). In this case, we may compute a Heaviside decomposition of the form (3.1), and proceed as usual with the pMRI algorithm. However, due to the unboundedness of the residues, the approximation quality may be locally sub-optimal near the branch point.

We remark that, due to round-off noise (in the computation of the snapshots, in the MRI procedure, and in the Heaviside decomposition), the latter case is much more likely to present itself in practice.

5. Numerical examples

We report in this section two numerical tests as evidence of the effectiveness of our technique. Our simulations were performed on the Helvetios cluster at EPFL [38]. For the sake of reproducibility, the corresponding code has been made available in [30].

5.1. Laplacian eigenvalues on a parametric rectangle

In this section we study a somewhat academic application in the field of PDEs for $d = 1$, which was originally considered in [39]. Given $(k, H) \in \mathbb{C} \times \mathbb{R}^+$, we take the following Helmholtz equation on the rectangle $\Omega_H =]0,1[\times]0,H[$

$$\begin{cases} -\left(\Delta + k^2\right) v_{k,H}(x_1,x_2) = f(x_1, x_2/H) & \text{for } (x_1,x_2) \in \Omega_H, \\ v_{k,H}(x_1,0) = 0 & \text{for } x_1 \in]0,1[\\ \partial_{x_2} v_{k,H}(x_1,H) = \cos(\pi x_1)/H & \text{for } x_1 \in]0,1[\\ \partial_{x_1} v_{k,H}(0,x_2) = \partial_{x_1} v_{k,H}(1,x_2) = 0 & \text{for } x_2 \in]0,H[, \end{cases} \tag{5.1}$$

with $f :]0,1[^2 \to \mathbb{R}$ a piecewise constant forcing term (see [39, Section 4.1] for the exact expression). Given how simply the geometry and the data of the problem depend on the parameter, we can recast (5.1) on the reference domain $\Omega_1 =]0,1[^2$ to make the parametric dependence emerge more clearly:

$$\begin{cases} -\left(\partial^2_{x_1x_1} + H^{-2}\partial^2_{x_2x_2} + k^2\right) u_{k,H}(x_1,x_2) = f(x_1,x_2) & \text{for } (x_1,x_2) \in \Omega_1 \\ u_{k,H}(x_1,0) = 0 & \text{for } x_1 \in]0,1[\\ \partial_{x_2} u_{k,H}(x_1,1) = \cos(\pi x_1) & \text{for } x_1 \in]0,1[\\ \partial_{x_1} u_{k,H}(0,x_2) = \partial_{x_1} u_{k,H}(1,x_2) = 0 & \text{for } x_2 \in]0,1[. \end{cases} \tag{5.2}$$

By inspection of the PDE above, we infer that $(z,p) := (k^2, H^{-2})$ is a good choice of parameters to study (5.2). We set as frequency and parameter ranges $Z = [10,50]$ and $\mathcal{P} = [0.2,1.2]$, respectively.

After spatial discretization by FEM on a regular mesh with n_S degrees of freedom, we obtain an algebraic problem of the form

$$(K_1 + pK_2 - zM)\, U(z,p) = b, \tag{5.3}$$

where K_1, K_2, and M are $n_S \times n_S$ matrices, and $U(z,p)$ and b are vectors of size n_S (here, we choose $n_S = 10201$). We remark that (5.3) is in the form of a dynamical system (2.1) with $B_p = b$. In order to conform to the functional setting of the PDE, we choose as norm over V the functional L^2 norm, which, in the discrete setting, corresponds to the energy norm induced by M: $\|\cdot\|_V = \left\|M^{1/2}\cdot\right\|_2$. The functional H^1_0 or (k-weighted) H^1 norms are also viable options; in our experience, the results of the simulation are barely affected by the choice of the norm.

We set as our target the FEM solution $U(z,p)$, *i.e.*, we fix $C_p = I$, the identity matrix, in (2.1). The poles of the Heaviside decomposition of U gain additional importance as FEM approximations of the eigenvalues of the

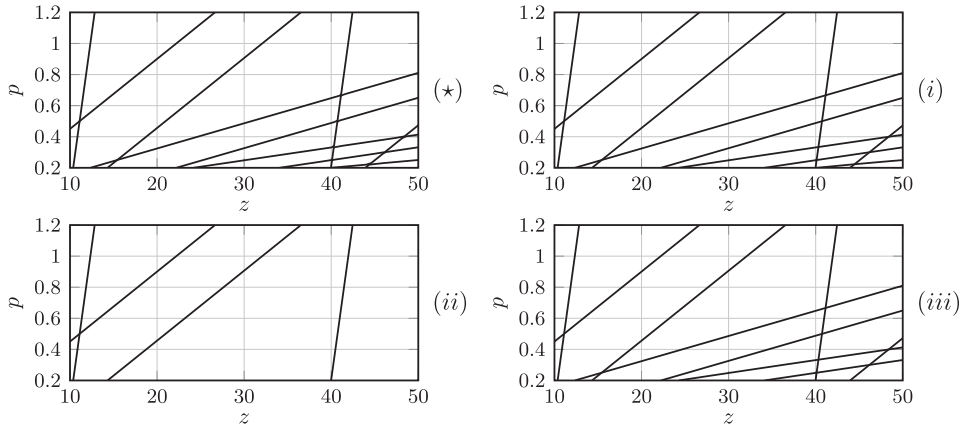

FIGURE 6. Analytic poles (5.4) of problem (5.1) (⋆). Surrogate poles obtained *via* double-greedy pMOR, with $\mathrm{tol}^{(i)}_{\mathrm{synth}}$, $\mathrm{tol}^{(ii)}_{\mathrm{synth}}$, and $\mathrm{tol}^{(iii)}_{\mathrm{synth}}$.

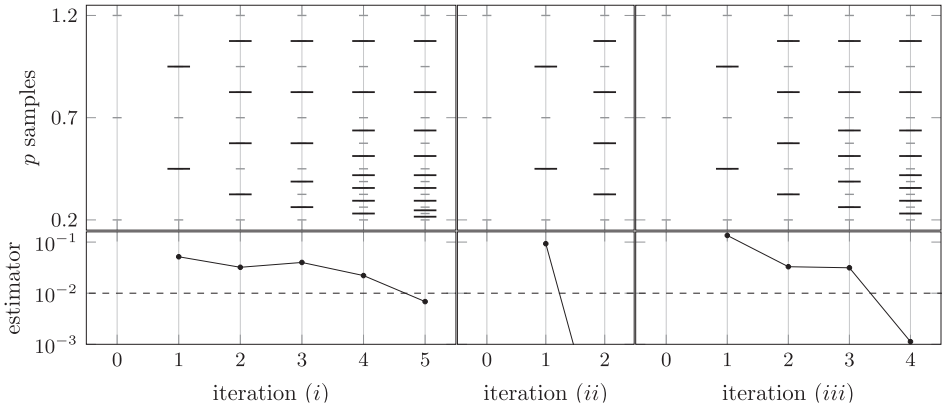

FIGURE 7. In the *top row*, new parameter samples added *via* greedy loop with $\mathrm{tol}^{(i)}_{\mathrm{synth}}$, $\mathrm{tol}^{(ii)}_{\mathrm{synth}}$, and $\mathrm{tol}^{(iii)}_{\mathrm{synth}}$. Thicker lines are used to denote the test set. The maximum error at each greedy iteration (measured by HEAVISIDEDISTANCE in Algorithm 3.5) is shown in the *bottom row*.

Laplace operator on the parametric domain Ω_H. Due to our choice of domain, such eigenvalues[6] are actually available in closed form:

$$\left\{ \pi^2 k^2 + \frac{\pi^2}{H^2} \left(\ell + \frac{1}{2} \right)^2 , \; k, \ell = 0, 1, \dots \right\}, \tag{5.4}$$

see Figure 6 (⋆). We use the exact expression of the poles (5.4) to validate the results obtained by our double-greedy pMOR technique.

We employ the following computational setup:

– The frequency and parameter training sets are initialized as 3 equispaced points in Z and $\mathcal{P}$, respectively, whereas the frequency test set contains 100 equispaced points in Z.

[6]The set (5.4) denotes the exact spectrum of the Laplace operator, without considering the FEM discretization. However, since the mesh is fine enough (the wavenumber k is low and the mesh size is smaller than k^{-2} [23]), we expect the FEM eigenvalues and eigenvectors to be close to the analytic ones.

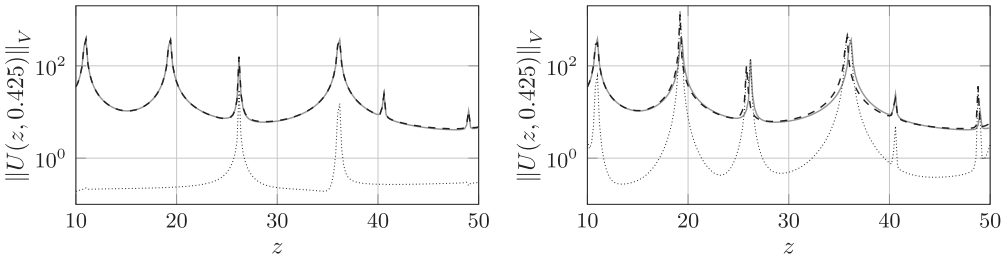

FIGURE 8. On the left, the dashed line represents the surrogate solution norm $\|U\|$ obtained by double-greedy pMOR at $p^\star$. On the right, the dashed line represents the solution norm of the closest z-surrogate (at $p^\star_{\text{close}}$). In both cases, the solid line is the exact norm, whereas the dotted line is the absolute error $\|U - \text{pROM}\|$.

– For MRI, Legendre polynomials are employed, whereas global monomials of degree 2 are used to interpolate (in a least squares sense) poles and residues after the greedy loop, in a post-processing "compression" step.
– After both frequency and parameter greedy loops, the information at the test points is not wasted, but included in the final surrogates, see Remarks 3.2 and 3.12. After the frequency greedy loops, we remove from the frequency surrogates any pole whose distance from Z is larger than 20, see Remark 3.3.
– The matching weight w is set to 1. The frequency and parameter greedy tolerances are set to 10^{-4} and 10^{-2}, respectively.

On top of this, we consider three different choices for the tolerance $\text{tol}_{\text{synth}}$, which is employed to deal with unbalanced matching, see Algorithm 3.4:

(i) $\text{tol}^{(i)}_{\text{synth}} = 0$: all synthetic poles are kept.
(ii) $\text{tol}^{(ii)}_{\text{synth}} = 1$: all synthetic poles are removed.
(iii) $\text{tol}^{(iii)}_{\text{synth}} = 1/2$: synthetic poles are removed only if they account for the majority of the information about a pole.

Also, after the pole-matching has been completed, we improve the synthetic poles by extrapolating *via* global degree 2 monomials the non-synthetic poles, see Remark 3.7. This affects only $\text{tol}^{(i,iii)}_{\text{synth}}$.

We show the surrogate poles in Figure 6. In all cases, we can observe that the pole-crossings are handled well by the matching algorithm. This is likely due to the fact that residue information is taken into account ($w > 0$), *cf.* Section 4.1.

In case (i), one erroneous pole crosses the frequency range Z for small p. This is not caused by spurious poles being present in the frequency surrogates, but by an inaccurate matching of correct poles lying on different sides of Z. Due to the strict tolerance in case (ii), the pROM is blind to most of the poles which are not uniformly inside Z. Instead, the hybrid approach (iii) achieves a good compromise, missing only one pole that leaves Z "too quickly" on the bottom right.

The "hystory" of Algorithm 3.5, namely the location of the new samples and the magnitude of the greedy error indicator, is portrayed in Figure 7. In the less strict cases (i) and (iii), the algorithm correctly identifies the "busiest" region of $\mathcal{P}$, *i.e.*, small values of p, as critical for a good approximation. This results in local refinements near $p = 0.2$. Instead, case (ii) only performs global refinements before terminating. This is actually a symptom of a general (undesirable) property of Algorithm 3.5: if $\text{tol}_{\text{synth}}$ is too large, pole-residue pairs might be removed too "aggressively" from the surrogate, so that the **p**-greedy termination criterion based on the Heaviside distance (3.8) contains only a few terms. This, in turn, yields a smaller Heaviside distance, which is more likely to satisfy the prescribed **p**-greedy tolerance, potentially leading to an early termination. The simplest solution is to reduce $\text{tol}_{\text{synth}}$. Alternatively, one could partition the parametric domain $\boldsymbol{\mathcal{P}} = \bigcup_i \boldsymbol{\mathcal{P}}_i$, and then build a different pMOR

TABLE 1. Simulation details about the example in Section 5.1.

	(i)	(ii)	(iii)
$\mathrm{tol}_{\mathrm{synth}}$	0	1	1/2
p-greedy iterations	5	2	4
p samples	19	9	17
Full model solves	258	109	224
Number of poles (at each p)	14	4	10
Synthetic poles (over all z-surrogates)	81 (out of 266)	0 (out of 36)	22 (out of 170)

surrogate on each parameter sub-domain $\boldsymbol{\mathcal{P}}_i$, see, *e.g.*, [21]. On each sub-domain, it is less likely that a relevant pole will be discarded due to it being "too often" synthetic, *cf.* Section 3.2.1. This second approach is more costly (especially if $\boldsymbol{\mathcal{P}}$ is large and/or high-dimensional), but is particularly advantageous when the poles move very quickly through the parametric domain (*i.e.*, if the gradient $\nabla_{\mathbf{p}}\lambda$ is large).

In Figure 8 (left), we compare exact and surrogate models at the point $p^\star = 0.425$. For simplicity, we only consider the best surrogate, obtained with $\mathrm{tol}_{\mathrm{synth}}^{(iii)}$. The approximation seems of good quality, with pole locations and residue magnitudes being identified extremely well, and the approximation error is small. In particular, the results seem better than those obtained by using the closest frequency surrogate (*i.e.*, the MRI built at the element of $\mathcal{P}_{\mathrm{train}}$ closest to $p^\star$, namely, $p^\star_{\mathrm{close}} = 0.41875$), see Figure 8 (right).

We also report a summary of the execution of the method and of the resulting pROM in Table 1. Our results agree with the main motivation behind the introduction of $\mathrm{tol}_{\mathrm{synth}}$, namely that it should control how to deal with uncertain or missing information, resulting in richer (but also more noise-prone) or poorer (here, insufficient) surrogates.

5.2. Transmission line with high-dimensional parameter space

Our last numerical example concerns the analysis of the admittance parameters of the 3-port transmission tree depicted in Figure 9. A motivation and similar tests can be found, *e.g.*, in [19]. Each branch is composed of a series of unit RLC cells: the "main" branch contains 400 cells, whereas the "up" and "down" branches contain 200 cells each. Resistance, inductance, and capacitance vary between cells: more precisely, if we restrict our focus to the main branch, the values of R, L, and C of the j-th cell are, for all $j = 1, \dots, 400$,

$$\begin{aligned} R^{\mathrm{main},j} &= (1 + \xi_R^{\mathrm{main},j} + p_R^{\mathrm{main}}) \cdot 5\,\mathrm{m\Omega}, \\ L^{\mathrm{main},j} &= (1 + \xi_L^{\mathrm{main},j} + p_L^{\mathrm{main}}) \cdot 0.25\,\mathrm{nH}, \\ C^{\mathrm{main},j} &= (1 + \xi_C^{\mathrm{main},j} + p_C^{\mathrm{main}}) \cdot 0.25\,\mathrm{pF}. \end{aligned}$$

We employ ξ to model random fluctuations of the nominal values in each cell, drawn from a uniform distribution with values between -0.2 and 0.2. Such random values are fixed once and for all during the initialization of the model: we do not consider them as parameters in our analysis. Instead, the parameters p denote branch-wide (independent of j) variations of the nominal values, and are envisioned to vary between -0.1 and 0.1. For $j = 1, \dots, 200$, the values of resistance, inductance, and capacitance in the secondary branches have the same expressions, with the subscript "main" being replaced by "up" and "down" for the top and bottom branches, respectively.

We consider the frequency range $Z = [0, 8]$ GHz, and, as parameters in our pMOR approach, we take the 9-dimensional vector

$$\mathbf{p} = (p_R^{\mathrm{main}}, p_L^{\mathrm{main}}, p_C^{\mathrm{main}}, p_R^{\mathrm{up}}, p_L^{\mathrm{up}}, p_C^{\mathrm{up}}, p_R^{\mathrm{down}}, p_L^{\mathrm{down}}, p_C^{\mathrm{down}}) \in [-0.1, 0.1]^9 = \boldsymbol{\mathcal{P}}.$$

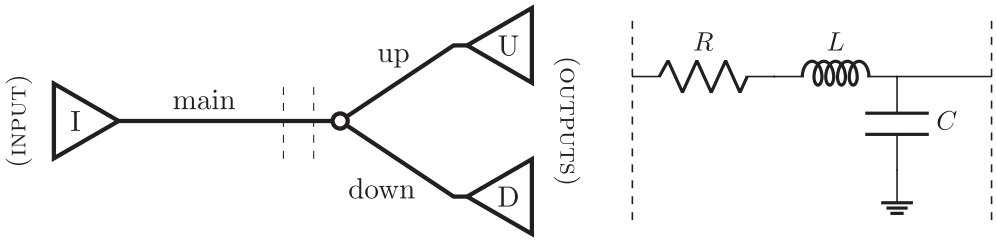

FIGURE 9. Transmission line network diagram (*left*) and circuit representation of a unit cell (*right*).

TABLE 2. Simulation details about the example in Section 5.2.

p-greedy iterations	6
p samples	457
Full model solves	9008
Number of poles (at each **p**)	14
Synthetic poles (over all z-surrogates)	183 (out of 6398)

The admittance parameters can be found by solving a system of the form (2.1), obtained by Modified Nodal Analysis: in particular,

$$A_{\mathbf{p}} = A^0 + p_R^{\text{main}} A^{\text{main}} + p_R^{\text{up}} A^{\text{up}} + p_R^{\text{down}} A^{\text{down}},$$
$$E_{\mathbf{p}} = E^0 + p_L^{\text{main}} E_L^{\text{main}} + p_L^{\text{up}} E_L^{\text{up}} + p_L^{\text{down}} E_L^{\text{down}} + p_C^{\text{main}} E_C^{\text{main}} + p_C^{\text{up}} E_C^{\text{up}} + p_C^{\text{down}} E_C^{\text{down}},$$

while $B_{\mathbf{p}} = B$ and $C_{\mathbf{p}} = C$ are independent of $\mathbf{p}$. In our case, the state X, which contains currents and voltages within the circuit, is a matrix of size 1603×3, whereas the output Y (the admittance matrix) has size 3×3.

Our MOR setup for Algorithm 3.5 is as follows:

- The frequency training set is initialized to the order 10 Chebyshev points of Z, whereas the frequency test set contains 100 equispaced points in Z; the parameter training set is initialized to 19 points in $\boldsymbol{\mathcal{P}}$: the origin $\mathbf{p} = \mathbf{0}$ and its 18 forward points.
- For MRI, Legendre polynomials are employed, whereas piecewise linear hat functions are used to interpolate poles and residues.
- After both frequency and parameter greedy loops, the information at the test points is not wasted, but included in the final surrogates, see Remarks 3.2 and 3.12. After the frequency greedy loops, we remove from the frequency surrogates any pole whose distance from Z is larger than 2 GHz, see Remark 3.3.
- The matching weight w is set to 1; the frequency and parameter greedy tolerances are set to 10^{-3} and 10^{-2}, respectively; the tolerance for unbalanced matching $\text{tol}_{\text{synth}}$ is set to $3/4$.

We show a summary of the results of the offline training in Table 2.

In Figure 10 we compare exact and surrogate models at the randomly chosen point

$$\mathbf{p}^\star = (-0.04, -0.08, -0.03, -0.01, 0.07, -0.09, -0.05, 0.03, 0.02). \tag{5.5}$$

For simplicity, we only show the magnitude of the admittance between ports I and U. The quality of the approximation appears good, with pole locations and residue magnitudes being identified well. In particular, the results seem better than those obtained by using the closest frequency surrogate (*i.e.*, the MRI built at the element of $\boldsymbol{\mathcal{P}}_{\text{train}}$ closest to $\mathbf{p}^\star$, namely, $\mathbf{p}^\star_{\text{close}} = [0, -0.1, 0, 0, 0, -0.1, 0, 0, 0]$).

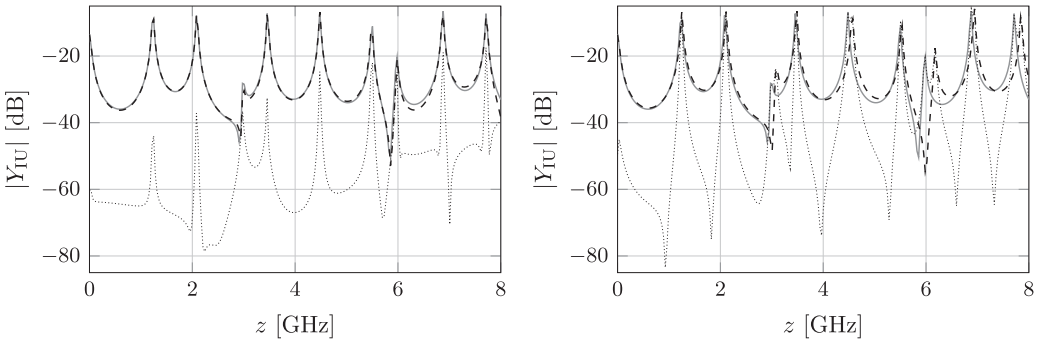

FIGURE 10. On the left, the dashed line represents the surrogate magnitude of the admittance Y_{IU} obtained by double-greedy pMOR at $\mathbf{p}^\star$. On the right, the dashed line represents the admittance magnitude of the closest z-surrogate (at $\mathbf{p}^\star_{\mathrm{close}}$). In both cases, the exact magnitude is denoted by a solid line, and the absolute error by a dotted line.

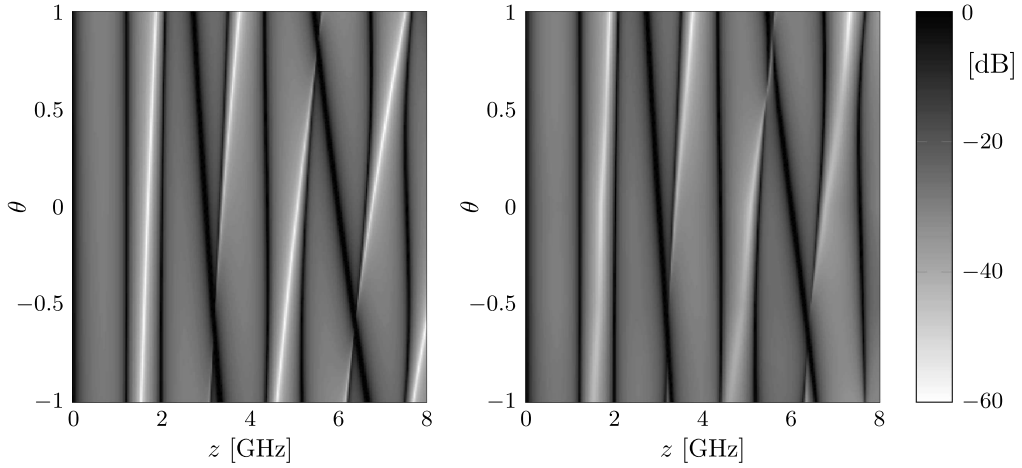

FIGURE 11. Plot of $|Y_{\mathrm{UD}}|$ (in dB) for $\mathbf{p}$ as in (5.6): exact (*left*) and surrogate *via* pMOR (*right*). The color scale is the same for both plots.

We look at a more global picture in Figure 11. We let $\mathbf{p}$ vary along a diagonal line between the two vertices $(0.1, 0.1, 0.1, -0.1, -0.1, \ldots, -0.1)$ and $(-0.1, -0.1, -0.1, 0.1, 0.1, \ldots, 0.1)$:

$$p^{\mathrm{main}}_{R,L,C} = -0.1\theta \quad \text{and} \quad p^{\mathrm{up}}_{R,L,C} = p^{\mathrm{down}}_{R,L,C} = 0.1\theta, \tag{5.6}$$

with θ an auxiliary parameter. We plot the magnitude of the admittance between the two output ports for $z \in Z$ and $\theta \in [-1, 1]$. Two different models are considered:

– The exact full order model (2.1).
– The surrogate obtained *via* double-greedy pMOR.

In Figure 11, we can observe that the exact admittance has 10 poles (darker lines) in the frequency range, some of which cross, and which, overall, create quite an intricate pattern. Still, the double-greedy pMOR surrogate seems to identify well the behavior of the quantity of interest, at least qualitatively. As could be expected, the quality of the approximation degrades slightly around pole intersections. A similar decrease in the accuracy of

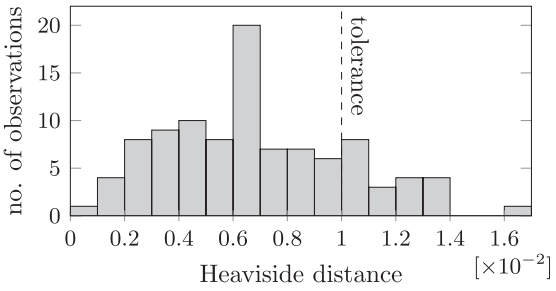

FIGURE 12. Histogram of the **p**-greedy error indicator (measured by HEAVISIDEDISTANCE in Algorithm 3.5) between pMOR surrogate and local (high fidelity) MRI surrogate at 100 verification parameters, not belonging to the training or test sets of the pMOR surrogate.

the surrogate can be observed also near the boundary of the parameter domain, since the two vertices of $\mathcal{P}$ obtained for $\theta = -1$ and $\theta = 1$ are *not* elements of $\mathcal{P}_{\text{train}}$.

To conclude the experiment, we also perform the following verification of the **p**-greedy tolerance employed to build the surrogate: we select 100 quasi-random (Halton) parameter points in $\mathcal{P}$, outside the training and test sets of the pMOR surrogate. At each such point $\mathbf{p}$, we build a reference frequency surrogate by z-adaptive MRI with the same parameters as above (10 Chebyshev points as starting training set, z-greedy tolerance of 10^{-3}). Note that we only use *new* snapshots at $\mathbf{p}$ to build this reference model. Then we compare this model with the prediction given by our pMOR surrogate, obtained by simply plugging the value of $\mathbf{p}$ in (3.3). We make this comparison quantitative through the Heaviside distance (3.8), which is also the metric driving the **p**-adaptivity, *cf.* Algorithm 3.5. We plot the results in the form of a histogram in Figure 12. We can observe that 80% of the verification points lie below the prescribed tolerance and that 99% of them lie below 1.5 times the tolerance. Considering the discrete (sparse grid) nature of the test set, it is reasonable to expect that the tolerance will not be attained everywhere. In this context, we find our results satisfactory since they show that, even in the few cases where the tolerance is not satisfied, the error indicator is still within a small margin of the tolerance.

6. CONCLUSIONS AND OUTLOOK

We have described the double-greedy pMOR approach for non-intrusive surrogate modeling of parametric problems, which samples adaptively in both frequency space and (potentially high-dimensional) parameter domain. In particular, the selection of parameter samples advances by trying to make the surrogate error small over a growing test set. We have illustrated with numerical examples the effectiveness of the method. Notably, we have shown that an accurate identification of number and behavior of poles and residues depends critically on some hyper-parameters (w and $\text{tol}_{\text{synth}}$) and, more generally, on the choice of a good strategy for interpolation over parameter space.

Among the several issues which remain unanswered we can find the following:

- Thanks to the degree of freedom provided by $\text{tol}_{\text{synth}}$, the proposed heuristic strategy for unbalanced matching is reasonably flexible. Still, it is unclear whether an "optimal" choice of this tolerance exists and, if it does, how to find it for a given application, since it depends on quantities unavailable *a priori*.
- In Section 4, we have showcased some of the difficulties related to intersecting poles, in particular the potentially discontinuous behavior of residues. Applying the double-greedy approach in this case may yield inadequate surrogates, with the risk of a large number of iterations of the greedy loop. However, we have observed no such issues in our latter two numerical examples, despite multiple pole intersections. This is likely due to the beneficial spectral properties of the full order problems that we considered.
- Issues similar to those discussed in the previous point are also possible (to a larger degree) in the case of multiple poles, even though, as discussed in Section 4, this case is quite unlikely to present itself thanks

to numerical noise. On one hand, this problem should be partially addressed by MRI: for instance, when building the frequency surrogate, it should be possible to determine whether two poles are a noisy double pole or a couple of single poles. On the other hand, our pMOR algorithm should be extended to allow dealing with multiple poles, without compromising the overall complexity of the algorithm. Both of these directions are object of ongoing research.

Acknowledgements. The authors acknowledge support from the Swiss National Science Foundation through project 182236.

References

[1] F. Alsayyari, Z. Perkó, D. Lathouwers and J. Kloosterman, A nonintrusive reduced order modelling approach using Proper Orthogonal Decomposition and locally adaptive sparse grids. *J. Comput. Phys.* **399** (2019) 108912.

[2] D. Amsallem and C. Farhat, Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity. *AIAA J.* **46** (2008) 1803–1813.

[3] D. Amsallem and C. Farhat, An online method for interpolating linear parametric reduced-order models. *SIAM J. Sci. Comput.* **33** (2011) 2169–2198.

[4] A. Antoulas, Approximation of large-scale dynamical systems, Advances in design and control, SIAM, 2005.

[5] V. Barthelmann, E. Novak and K. Ritter, High dimensional polynomial interpolation on sparse grids. *Adv. Comput. Math.* **12** (2000) 273–288.

[6] U. Baur, C. Beattie, P. Benner and S. Gugercin, Interpolatory Projection Methods for Parameterized Model Reduction. *SIAM J. Sci. Comput.* **33** (2011) 2489–2518.

[7] B. Beckermann, G. Labahn and A. Matos, On rational functions without Froissart doublets. *Numer. Math.* **138** (2018) 615–633.

[8] P. Benner and L. Feng, A robust algorithm for parametric model order reduction based on implicit moment matching, chap. 6 in *Reduced Order Methods for Modeling and Computational Reduction*, edited by A. Quarteroni and R. Gianluigi Rozza. Springer International Publishing (2014) 159–185.

[9] P. Benner, S. Gugercin and K. Willcox, A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Rev.* **57** (2015) 483–531.

[10] T. Betcke, N. Higham, V. Mehrmann, C. Schröder and F. Tisseur, NLEVP: A collection of nonlinear eigenvalue problems. *ACM Trans. Math. Softw.* **39** (2013) 1-28.

[11] F. Bonizzoni, F. Nobile, I. Perugia and D. Pradovera, Fast Least-Squares Padé approximation of problems with normal operators and meromorphic structure. *Math. Comput.* **89** (2020) 1229–1257.

[12] P. Chen and A. Quarteroni, A new algorithm for high-dimensional uncertainty quantification based on dimension-adaptive sparse grid approximation and reduced basis methods. *J. Comput. Phys.* **298** (2015) 176–193.

[13] A. Chkifa, A. Cohen and C. Schwab, High-Dimensional Adaptive Sparse Polynomial Interpolation and Applications to Parametric PDEs. *Found. Comput. Math.* **14** (2014) 601–633.

[14] D. Crouse, On implementing 2D rectangular assignment algorithms. *IEEE Trans. Aerosp. Electron. Syst.* **52** (2016) 1679–1696.

[15] Z. Drmač, S. Gugercin and C. Beattie, Quadrature-Based Vector Fitting for Discretized $\mathcal{H}_2$ Approximation. *SIAM J. Sci. Comput.* **37** (2015) 625–652.

[16] Q. Du and M. Gunzburger, Centroidal Voronoi Tessellation Based Proper Orthogonal Decomposition Analysis. *Control Estimat. Distrib. Parameter Syst.* **143** (2003) 137–150.

[17] F. Ferranti, L. Knockaert and T. Dhaene, Passivity-preserving parametric macromodeling by means of scaled and shifted state-space systems. *IEEE Trans. Microw. Theory Tech.* **59** (2011) 2394–2403.

[18] S. Grivet-Talocia and E. Fevola, Compact Parameterized Black-Box Modeling via Fourier-Rational Approximations. *IEEE Trans. Electromagn. Compat.* **59** (2017) 1133–1142.

[19] S. Grivet-Talocia and B. Gustavsen, Passive Macromodeling: Theory and Applications, Wiley series in microwave and optical engineering, John Wiley & Sons, Inc., Hoboken, New Jersey (2015).

[20] S. Grivet-Talocia and R. Trinchero, Behavioral, Parameterized, and Broadband Modeling of Wired Interconnects with Internal Discontinuities. *IEEE Trans. Electromagn. Compat.* **60** (2018) 77–85.

[21] B. Haasdonk, M. Dihlmann and M. Ohlberger, A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space. *Math. Comput. Model. Dyn. Syst.* **17** (2011) 423–442.

[22] R. Hiptmair, L. Scarabosio, C. Schillings and C. C. Schwab, Large deformation shape uncertainty quantification in acoustic scattering. *Adv. Comput. Math.* **44** (2018) 1475–1518.

[23] F. Ihlenburg and I. Babuška, Finite element solution of the Helmholtz equation with high wave number Part I: The h-version of the FEM. *Comput. Math. Appl.* **30** (1995)9–37.

[24] A. Ionita and A. Antoulas, Data-Driven Parametrized Model Reduction in the Loewner Framework. *SIAM J. Sci. Comput.* **36** (2014) A984–A1007.

[25] R. Karp, Reducibility Among Combinatorial Problems, in *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, edited by R. Miller and J. Thatcher. *The IBM Research Symposia Series*, Plenum Press, New York (1972) 85–103.

[26] S. Lefteriu, A. Antoulas and A. Ionita, Parametric model reduction in the Loewner framework, in Vol. 44 of *IFAC Proceedings Volumes* (2011) 12751–12756.
[27] B. Lohmann and R. Eid, Efficient order reduction of parametric and nonlinear models by superposition of locally reduced models, Methoden und Anwendungen der Regelungstechnik. Erlangen-Münchener Workshops 2007 und 2008 (2009) 1–9.
[28] X. Ma and N. Zabaras, An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *J. Comput. Phys.* **228** (2009) 3084–3113.
[29] F. Nobile, R. Tempone and C. Webster, A Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data. *SIAM J. Numer. Anal.* **46** (2008) 2309–2345.
[30] F. Nobile and D. Pradovera, Non-intrusive double-greedy parametric model reduction by interpolation of frequency-domain rational surrogates-numerical tests (2020). Preprint arXiv:2008.10864v3.
[31] H. Panzer, J. Mohring, R. Eid and B. Lohmann, Parametric model order reduction by matrix interpolation. *At-Automatisierungstechnik* **58** (2010) 475–484.
[32] D. Pflüger, B. Peherstorfer and H. Bungartz, Spatially adaptive sparse grids for high-dimensional data-driven problems, in Vol. 26 *Journal of Complexity*, Academic Press Inc., (2010) 508–522.
[33] G. Phillips, Interpolation and approximation by polynomials, Vol. 14 of *CMS books in mathematics*, Springer, New York (2003).
[34] D. Pradovera, Interpolatory rational model order reduction of parametric problems lacking uniform inf-sup stability. *SIAM J. Numer. Anal.* **58** (2020) 2265–2293.
[35] D. Pradovera and F. Nobile, Frequency-domain non-intrusive greedy model order reduction based on minimal rational approximation (2020). SCEE 2020 Proceedings DOI: 10.5075/epfl-MATHICSE-275533. (to appear).
[36] A. Quarteroni, A. Manzoni and F. Negri, Reduced Basis Methods for Partial Differential Equations: An Introduction, UNITEXT, Springer International Publishing (2015).
[37] M. Reed and B. Simon, Analysis of operators, Vol. 4 of *Methods of modern mathematical physics*, Academic Press, New York, 4th print ed. (1981).
[38] SCITAS, EPFL Helvetios cluster webpage, accessed August 24 (2020).
[39] K. Smetana, O. Zahm and A. Patera, Randomized residual-based error estimators for parametrized equations. *SIAM J. Sci. Comput.* **41** (2019) 900–926.
[40] P. Virtanen, et al., SciPy 1.0: Fundamental algorithms for scientific computing in sython. *Nat. Methods* **17** (2020) 261–272.
[41] D. Weile, E. Michielssen, E. Grimme and K. Gallivan, A method for generating rational interpolant reduced order models of two-parameter linear systems. *Appl. Math. Lett.* **12** (1999) 93–102.
[42] H. Wendland, Scattered Data Approximation, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press (2004).
[43] Y. Yue, L. Feng and P. Benner, An Adaptive Pole-Matching Method for Interpolating Reduced-Order Models (2019). Preprint arXiv:1908.00820.
[44] Y. Yue, L. Feng and P. Benner, Reduced-order modelling of parametric systems via interpolation of heterogeneous surrogates. *Adv. Model. Simul. Eng. Sci.* **6** (2019).

Subscribe to Open (S2O)
A fair and sustainable open access model

This journal is currently published in open access under a Subscribe-to-Open model (S2O). S2O is a transformative model that aims to move subscription journals to open access. Open access is the free, immediate, online availability of research articles combined with the rights to use these articles fully in the digital environment. We are thankful to our subscribers and sponsors for making it possible to publish this journal in open access, free of charge for authors.

Please help to maintain this journal in open access!

Check that your library subscribes to the journal, or make a personal donation to the S2O programme, by contacting subscribers@edpsciences.org

More information, including a list of sponsors and a financial transparency report, available at: https://www.edpsciences.org/en/maths-s2o-programme