Physics-informed cluster analysis and a priori efficiency criterion for the construction of local reduced-order bases

Thomas Daniel, Fabien Casenave, Nissrine Akkari, Ali Ketata, David Ryckelynck

Journal of Computational Physics (2022)

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Physics-informed cluster analysis and a priori efficiency criterion for the construction of local reduced-order bases

Thomas Daniel[a,b], Fabien Casenave[a], Nissrine Akkari[a], Ali Ketata[b], David Ryckelynck[b]

[a] *SafranTech, Rue des Jeunes Bois, Châteaufort, 78114 Magny-les-Hameaux (France).*
[b] *MINES ParisTech, PSL University, Centre des matériaux (CMAT), CNRS UMR 7633, BP 87, 91003 Evry (France).*

ARTICLE INFO

Article history:
Received March 2021

Keywords: local reduced-order bases, cluster analysis, dissimilarity measures, ROM-nets.

ABSTRACT

Nonlinear model order reduction has opened the door to parameter optimization and uncertainty quantification in complex physics problems governed by nonlinear equations. In particular, the computational cost of solving these equations can be reduced by means of local reduced-order bases. This article examines the benefits of a physics-informed cluster analysis for the construction of cluster-specific reduced-order bases. We illustrate that the choice of the dissimilarity measure for clustering is fundamental and highly affects the performances of the local reduced-order bases. It is shown that clustering with an angle-based dissimilarity on simulation data efficiently decreases the intra-cluster Kolmogorov N-width. Additionally, an a priori efficiency criterion is introduced to assess the relevance of a ROM-net, a methodology for the reduction of nonlinear physics problems introduced in our previous work in [T. Daniel, F. Casenave, N. Akkari, D. Ryckelynck, Model order reduction assisted by deep neural networks (ROM-net), *Advanced Modeling and Simulation in Engineering Sciences* 7 (16), 2020]. This criterion also provides engineers with a very practical method for ROM-nets' hyperparameters calibration under constrained computational costs for the training phase. On five different physics problems, our physics-informed clustering strategy significantly outperforms classic strategies for the construction of local reduced-order bases in terms of projection errors.

1. Introduction

Differential equations are widely used for the mathematical modeling of physical phenomena. These differential equations involve boundary conditions, initial conditions, constants and source terms that can be considered as parameters of the physics problem. For well-posed parametrized differential equations, any point of the parameter space is associated to one single solution. Under the third Hadamard well-posedness condition, the solution is a continuous

e-mail: thomas.daniel@mines-paristech.fr (Thomas Daniel)

function of the parameters. Therefore, any connected set in the parameter space defines a connected set in the solution space, called *solution manifold*. This concept can be extended to any quantity of interest, be it an internal variable or a function of the solution. The solution manifold can be interpreted as the support of a probability density function for the solution in uncertainty propagation, when a probabilistic model is given to describe uncertainties on the parameters. The concept of solution manifold also appears in design optimization where the objective is to minimize a solution-dependent cost function by modifying parameters such as material constants, microstructural properties, or geometrical characteristics, for instance. Both uncertainty quantification and design optimization are many-query problems since they require solving the parametrized differential equations for a large number of points in the parameter space, which is sometimes prohibitive. To mitigate the computational cost related to such applications, many *model order reduction* methods [1, 2] have been developed, including methods based on tensor decompositions (e.g. the *Proper Generalized Decomposition* [3, 4]) and projection-based methods (e.g. the *Reduced Basis method* [5, 6] and the *POD Galerkin method* [7, 8]). *Projection-based model order reduction* consists in computing an approximate solution in a low-dimensional subspace of the solution space, which can give accurate predictions provided that the solution manifold is embedded in a low-dimensional space.

The potential of such numerical methods is related to the maximum distance between a point of the solution manifold and its orthogonal projection onto the approximation space. This distance is used to define the *Kolmogorov N-width* measuring the worst-case error for the best N-dimensional approximation space. The behavior of the Kolmogorov width when increasing the dimension N provides information about the reducibility of a given physics problem: slowly decaying Kolmogorov widths indicate that increasing the dimension of the linear approximation subspace does not significantly improve the quality of the approximate solution. The asymptotic behavior of the Kolmogorov width is studied in [9, 10]. Problems combining large values of the Kolmogorov widths with low decay rates are not reducible, which means that one cannot compute accurate approximate solutions at low computational costs. This is the case in particular when considering wave propagation [10] and advection-dominated problems [11, 12, 13, 14, 15, 16]. For such problems, in the projection-based model order reduction community, [17, 18] suggested using multiple *local* approximation spaces dedicated to different subsets of the solution manifold. These local approximation spaces are spanned by local reduced-order bases (ROBs), usually computed with the Proper Orthogonal Decomposition (POD [19, 7]). ROM interpolation techniques have also been extensively studied in [20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]. More recently, [32] introduced two nonlinear model order reduction techniques called *manifold Galerkin method* and *manifold least-squares Petrov-Galerkin method*, where a deep convolutional autoencoder is used for nonlinear dimensionality reduction. In [33], this idea is extended to a hyper-reduction framework, using a shallow masked autoencoder with fully-connected layers.

The present article focuses on the use of multiple local ROBs to tackle the slow decrease of the Kolmogorov N-width, because of the compatibility of this approach with the Galerkin method and its classical extension to hyper-reduction methods [34, 35, 36, 37, 38]. Splitting a non-reducible problem into multiple reducible ones can be achieved with *cluster analysis*. Cluster analysis belongs to unsupervised learning tasks and can be defined as the search of groups (or *clusters*) of similar objects in a database. The choice of the clustering algorithm depends on the underlying motivation and thus on the clusters' topological properties that are expected. Representative-based algorithms [39, 40, 41] use a dissimilarity measure to assign each object of the database to the cluster corresponding to the closest representative object, leading to compact clusters. We refer the reader to the books [42], [40] (chapters 6 and 7), [43] (section 14.3), or articles [44, 45] for more details about clustering algorithms. In cases where efficient error bounds for reduced representations are available, ideal cases being for linear problems with affine parameter dependencies, a powerful framework has been proposed in [46]. Random sketching is used to improve the orthogonal greedy algorithm, and then used in a greedy algorithm where the solution is approximated in a subspace spanned by vectors selected online in a dictionary of candidate basis vectors.

The present work studies *physics-informed* clustering strategies for the construction of dictionaries of local ROBs, for complex nonlinear problems parametrized by a field that may not have known error bounds. Physics-informed cluster analysis consists in clustering the parameter space by means of a dissimilarity measure which involves physical quantities obtained when solving the physics problem. In other words, clusters in the parameter space are implicitly defined as the preimages of clusters found in a database of numerical simulation results. In practice, this means that a clustering algorithm is applied in the solution space, as proposed for the first time in [17, 18]. The focus is on finding a clustering strategy that is appropriate for model order reduction purposes. In [47], it was noticed that clustering based on the Euclidean distance (or L^2 distance) in the solution space was not adapted for the construction of local reduced-order models (ROMs), which led to the definition of *projection-error based local ROMs* (PEBL-ROM)

where the solution space is hierarchically partitioned using the projection error as a dissimilarity criterion. We propose here to work with the sine dissimilarity related to *relative* projection errors instead, giving a symmetric dissimilarity measure that can be plugged into a representative-based clustering method. We show that this dissimilarity measure corresponds to the Hilbert-Schmidt distance between the projections onto the snapshots' approximation spaces, which is of particular interest when using Galerkin projection to solve the governing equations.

Our main contributions are twofold. First, we developed a physics-informed clustering strategy based on the sine dissimilarity and k-medoids clustering, and an automatic snapshot selection procedure for the construction of POD bases. Second, with the practical limitation of the computational budget in industrial contexts, we developed an *a priori* efficiency criterion enabling hyperparameters calibration for dictionary-based ROM-nets [48], where a classifier is trained to automatically recommend the best ROM in the dictionary for a given point in the parameter space without computing the dissimilarity online. Section 2 gives an overview of model order reduction methods and of the techniques that have been developed to deal with non-reducible problems, including dictionaries of local ROMs. Section 3 presents representative-based clustering algorithms, and in particular k-medoids clustering. Our physics-informed clustering strategy and a priori efficiency criterion are introduced in Sections 4 and 5 respectively. Applications to various physics problems are developed in Section 6 and show the importance of choosing an appropriate dissimilarity measure for clustering. Section 4 can be skipped for the first reading, as it contains technical details about our dissimilarity measure.

2. Model order reduction background

Let us consider a physics problem described by the following parametrized differential equation:

$$\mathcal{D}(u; x) = 0, \tag{1}$$

where u is the primal variable belonging to a Hilbert space $\mathcal{H}$ whose inner product is denoted by $\langle ., . \rangle_{\mathcal{H}}$, x denotes the parameters of the problem, and $\mathcal{D}$ is an operator involving a differential operator and operators for initial conditions and/or boundary conditions. Equation (1) can be a system of ordinary differential equations or partial differential equations depending on the physics problem. Let us assume that this physics problem is well-posed in the sense of Hadamard, that is to say that there exists a unique solution $u(x)$ for any parameter x, and that this solution changes continuously with x. Let us introduce the set $\mathcal{X}$ of all the possible parameters x. The *solution manifold* $\mathcal{M}$ is defined by:

$$\mathcal{M} := u(\mathcal{X}) = \{u(x) \mid x \in \mathcal{X}\}. \tag{2}$$

2.1. Introduction to model order reduction

Model order reduction [1, 2] is a discipline in numerical analysis consisting in replacing a computationally expensive high-fidelity model by a fast reduced-order model (ROM) to calculate approximate solutions of some complex physics equations. A ROM can be either a data-driven metamodel (or surrogate model) calibrated with a regression algorithm, or a physics-based model obtained by numerical methods such as the Proper Generalized Decomposition [3, 4], the Reduced Basis method [5, 6], and the POD Galerkin method [7, 8], among others. It is generally used for parametrized equations whose solution must be known for different points in the parameter space. As in machine learning, a model order reduction procedure starts by a *training phase* (or *offline stage*) where the ROM is built from some training data. The ROM is then used on test data in an *exploitation phase* (or *online stage*). In the training phase, high-fidelity solutions, called *snapshots*, are computed with the high-fidelity model for different points of the parameter space to get a sampled representation of the solution manifold. The model order reduction algorithm analyzes these snapshots to learn how the solution is affected by parameter variations. Contrary to usual machine learning problems, the amount of training data is limited because the high-fidelity model giving snapshots is time-consuming and costly. The selection of relevant points in the parameter space can be optimized to ensure that the snapshots are representative of the behavior of the solution, like in the greedy approach of the Reduced Basis method where an a posteriori error estimator is used to select snapshots. Given the cost of computing snapshots in the training phase, a ROM is profitable only if it is extensively used in the exploitation phase. This paper addresses issues that are specific to *projection-based model order reduction* (e.g. POD Galerkin, Reduced Basis method) where the approximate solution is obtained by solving the physics equations with the Galerkin method on a well-chosen reduced-order basis (ROB).

2.2. The Proper Orthogonal Decomposition (POD)

It is now assumed that Equation (1) defines a parametrized partial differential equation whose solution $u(x)$ for a given point $x \in \mathcal{X}$ in the parameter space is a function of space and time defined on $\Omega \times [0; t_f]$, with $\Omega \subset \mathbb{R}^\alpha$, $\alpha = 1$, 2 or 3 and $t_f \in \mathbb{R}_+^*$. Most of the time, the Hilbert space $\mathcal{H}$ is a subspace of the Lebesgue space $L^2(\Omega \times [0; t_f])$ of square-integrable functions. However, parameters x and time t can be considered together in a variable χ called *generalized parameters* living in the set $\tilde{\mathcal{X}} = \mathcal{X} \times [0; t_f]$. Therefore, the solution $u(\chi)$ belongs to the space $L^2(\Omega)$. The Stiefel manifold $V(N, L^2(\Omega))$ represents the set of all orthonormal N-frames in $L^2(\Omega)$. For two square-integrable functions f and g, the notation $\langle f, g \rangle_{L^2(\Omega)}$ stands for the $L^2(\Omega)$ inner product $\int_\Omega f(\boldsymbol{\xi}) g(\boldsymbol{\xi}) d\boldsymbol{\xi}$. The following definition gives a theoretical continuous definition of the proper orthogonal decomposition (POD [19, 7]), also known as the Karhunen-Loève decomposition or principal component analysis:

Definition 2.1 (POD basis)**.** *Let $u : \tilde{\mathcal{X}} \to L^2(\Omega)$. A POD basis $\{\psi_k^*\}_{1 \leq k \leq N} \in V(N, L^2(\Omega))$ of order $N \in \mathbb{N}^*$ of u is a solution of the following optimization problem:*

$$\{\psi_k^*\}_{1 \leq k \leq N} \in \underset{\{\psi_k\}_{1 \leq k \leq N} \in V(N, L^2(\Omega))}{\arg\min} \int_{\chi \in \tilde{\mathcal{X}}} \|u(\chi) - \sum_{k=1}^{N} \langle u(\chi), \psi_k \rangle_{L^2(\Omega)} \psi_k \|_{L^2(\Omega)}^2 \, d\chi. \tag{3}$$

The sum in Equation (3) is the proper orthogonal decomposition of order N of u. When $N \to +\infty$, the approximation error given by the minimum of the cost function in Equation (3) tends towards zero (Theorem 4 in [49]).

Let $\mathcal{H}$ be a Hilbert space with an orthonormal basis $\{e_i\}_{i \in \mathbb{N}^*}$, and $A : \mathcal{H} \to \mathcal{H}$ a linear operator. We define the Hilbert-Schmidt function $\Lambda_{HS(\mathcal{H})}$ as:

$$\Lambda_{HS(\mathcal{H})}(A) := \sqrt{\sum_{i=1}^{\infty} \|A(e_i)\|_{\mathcal{H}}^2}, \tag{4}$$

which can potentially take infinite values. A linear operator A on a Hilbert space $\mathcal{H}$ is a *Hilbert-Schmidt operator* if $\Lambda_{HS(\mathcal{H})}(A)$ is finite. As shown in [50] (Chapter VIII, Theorem 2.3), the set $HS(\mathcal{H})$ of all Hilbert-Schmidt operators on $\mathcal{H}$ is a Hilbert space with respect to the following inner product:

$$\langle A, B \rangle_{HS(\mathcal{H})} := \sum_{i=1}^{\infty} \langle A(e_i), B(e_i) \rangle_{\mathcal{H}}. \tag{5}$$

The Hilbert-Schmidt function $\Lambda_{HS(\mathcal{H})}$ is actually the norm induced by this inner product, and corresponds to the Frobenius norm for matrices when the vector space $\mathcal{H}$ is finite-dimensional. We now use the more conventional notation $\|A\|_{HS(\mathcal{H})} := \Lambda_{HS(\mathcal{H})}(A)$ for the Hilbert-Schmidt norm. The Hilbert-Schmidt inner product and norm are independent of the choice of the basis $\{e_i\}_{i \in \mathbb{N}^*}$, see Proposition 9.18 in Chapter 9 of [51], which will be useful for proofs of some properties of the dissimilarity measure introduced in this paper. The POD is highly related to the theory of Hilbert-Schmidt operators. In [52, 53], it is shown that the POD optimization problem is equivalent to finding the optimal approximation of a Hilbert-Schmidt operator related to u by a finite rank operator in the Hilbert-Schmidt norm. The POD basis functions can also be obtained from the eigenfunctions of the Hilbert-Schmidt integral operator [7] $\mathcal{R}_u$:

$$L^2(\tilde{\mathcal{X}}) \ni \varphi \mapsto \mathcal{R}_u(\varphi) \in L^2(\tilde{\mathcal{X}}), \tag{6}$$

where $\mathcal{R}_u(\varphi)$ is defined by:

$$\tilde{X} \ni \chi \mapsto \mathcal{R}_u(\varphi)(\chi) := \int_{\chi' \in \tilde{\mathcal{X}}} \langle u(\chi), u(\chi') \rangle_{L^2(\Omega)} \varphi(\chi') d\chi' \in \mathbb{R}. \tag{7}$$

In this work, we keep the explicit distinction between the time t and the parameters $x \in \mathcal{X}$ rather than working on the generalized parameters $\chi \in \tilde{\mathcal{X}}$, because we do not consider the time as a clustering variable. Nonetheless, spatio-temporal functions $f \in L^2(\Omega \times [0; t_f])$ are considered as trajectories $(f(., t))_{t \in [0; t_f]}$ in the Hilbert space $L^2(\Omega)$. In other words, such functions are seen as functions defined on Ω and parametrized by the time. For this reason, the manifold $\mathcal{M}$ is rather defined by:

$$\mathcal{M} := \{u(x)(., t) \mid x \in \mathcal{X}, \ t \in [0; t_f]\}, \tag{8}$$

and the approximation spaces are subspaces of $L^2(\Omega)$, leading to an approximate solution expressed as a time-dependent linear combination of basis functions defined on Ω.

In practice, we are given a finite set of m points x_i of the parameter space $\mathcal{X}$, for which high-fidelity solutions $u(x_i)$ are computed in a high-dimensional approximation space whose dimension is denoted by $\mathcal{N}$. These solutions, called snapshots, provide information about the behavior of the physical system and give a sampled version of the solution manifold. The POD is applied as a linear dimensionality reduction technique, processing this information to build a ROB that can be used to accelerate future numerical simulations for new parameters.

Definition 2.2 (POD basis construction)**.** *Given an integer $N \leq \mathcal{N}$, a POD basis $\{\psi_k^*\}_{1\leq k\leq N} \in V(N, L^2(\Omega))$ is computed from the snapshots $\{u(x_i)\}_{1\leq i\leq m}$ as a solution of the following optimization problem:*

$$\{\psi_k^*\}_{1\leq k\leq N} \in \underset{\{\psi_k\}_{1\leq k\leq N}\in V(N,L^2(\Omega))}{\arg\min} \sum_{i=1}^{m} \|u(x_i) - \sum_{k=1}^{N} \langle u(x_i), \psi_k\rangle_{L^2(\Omega)}\psi_k\|^2_{L^2(\Omega\times[0;t_f])}. \tag{9}$$

The uniqueness of the POD basis is obtained by specifying a construction algorithm, such as the Snapshot POD [54, 55] or the singular value decomposition (SVD) for instance. By construction, the subspace spanned by the ROB minimizes the projection errors of the snapshots $u(x_i)$. The optimality of the POD basis is discussed and illustrated in [56]. In practice, when using a numerical procedure to solve Equation (1), for example the finite-element method with a time-stepping scheme, the coordinates of the snapshots in the finite-element basis are stored in columns in a matrix $\mathbf{Q} \in \mathbb{R}^{\mathcal{N}\times mn_t}$ called snapshots matrix, with n_t being the number of time steps. The coordinates of the POD modes ψ_k are given in the N first columns of the matrix $\mathbf{M}^{-1/2}\mathbf{V}$, where $\mathbf{M} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ is the finite-element mass matrix and $\mathbf{V} \in \mathbb{R}^{\mathcal{N}\times\text{rank}(\mathbf{Q})}$ is the matrix of left singular vectors in the SVD of the snapshots matrix $\mathbf{Q}$, when indexing the singular values in decreasing order. The decay rate of the singular values of the snapshots matrix $\mathbf{Q}$ is related to the behavior of the sequence of Kolmogorov widths. It enables evaluating the reducibility of the physics problem. When computing a POD basis for a variable defined at integration points rather than the finite-element mesh nodes, for the purpose of applying Gappy-POD after hyper-reduced simulations, the POD modes are simply given by the N first left singular vectors in the SVD of the corresponding snapshots matrix.

2.3. Non-reducible problems

Approximate solutions of Equation (1) can be obtained by solving the PDEs on a finite-dimensional subspace $\mathcal{H}_N \in \text{Gr}(N, \mathcal{H})$ spanned by a ROB, where the Grassmannian $\text{Gr}(N, \mathcal{H})$ is the set of all N-dimensional subspaces of $\mathcal{H}$. The best approximation in $\mathcal{H}_N$ of the solution $u(x)$ for a given parameter x is the following orthogonal projection $\pi_{\mathcal{H}_N}(u(x))$ onto the approximation space:

$$\pi_{\mathcal{H}_N}(u(x)) = \underset{v\in\mathcal{H}_N}{\arg\min} \|u(x) - v\|_{\mathcal{H}}, \tag{10}$$

with $\|.\|_{\mathcal{H}}$ denoting the norm induced by the inner product of the Hilbert space $\mathcal{H}$.
The Kolmogorov N-width is defined by:

$$d_N(\mathcal{M}) := \inf_{\mathcal{H}_N\in\text{Gr}(N,\mathcal{H})} \sup_{u\in\mathcal{M}} \inf_{v\in\mathcal{H}_N} \|u - v\|_{\mathcal{H}} = \inf_{\mathcal{H}_N\in\text{Gr}(N,\mathcal{H})} \sup_{u\in\mathcal{M}} \|u - \pi_{\mathcal{H}_N}(u)\|_{\mathcal{H}}, \tag{11}$$

and quantifies how well the solution manifold $\mathcal{M}$ can be approximated by searching approximate solutions in a N-dimensional subspace of $\mathcal{H}$. The Kolmogorov N-width corresponds to the worst projection error on the best N-dimensional approximation space. For a fixed solution manifold $\mathcal{M}$, the sequence $(d_N(\mathcal{M}))_{N\in\mathbb{N}}$ is decreasing, which means that approximation errors get lower when increasing the dimension of the approximation space.

For some problems, the Kolmogorov width slowly decays when increasing the dimension N of the approximation space. For these *non-reducible* problems, the dimension N of the linear approximation space giving a sufficiently small Kolmogorov width is generally too high to enable the fast computation of approximate solutions. Qualitatively, the solution manifold $\mathcal{M}$ covers too many independent directions to be embedded in a low-dimensional subspace. To address this issue, several techniques have been developed:

- Problem-specific methods tackle the difficulties of some specific physics problems that are known to be non-reducible, such as advection-dominated problems which have been largely investigated, for instance in [13, 14, 15, 16, 57, 58, 59].

- Online-adaptive model reduction methods update the ROM in the exploitation phase by collecting new information online as explained in [60], in order to limit extrapolation errors when solving the parametrized governing equations in a region of the parameter space that was not explored in the training phase. The ROM can be updated for example by querying the high-fidelity model when necessary for basis enrichment [35, 61, 62, 63, 64]. Other methods propose enrichment procedures that do not require solving the equations with the high-fidelity model, whose complexity scales linearly with ([65, 66]) or is independent of ([67]) the dimension of the high-fidelity model.

- ROM interpolation methods [20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31] use interpolation techniques on Grassmann manifolds or matrix manifolds to adapt the ROM to the parameters considered in the exploitation phase by interpolating between two precomputed ROMs.

- Dictionaries of basis vector candidates enable building a parameter-adapted ROM in the exploitation phase by selecting a few basis vectors. This technique is presented in [68, 69] for the Reduced Basis method.

- Dictionaries of ROMs rely on the construction of several local ROMs adapted to different regions of the solution manifold. These local ROMs can be obtained by partitioning the time interval [70, 71], the parameter space [70, 72, 73, 74, 64, 75, 76], or the solution space [17, 18, 74, 77, 47, 78, 48, 38]. Local ROMs have been used both with the Reduced Basis method and the POD Galerkin method. In the same vein as online-adaptive model reduction methods, local ROBs can be adapted online using for example a low-rank SVD update method, as in [18, 77] when switching from one local ROB to another or in [64] when an error indicator detects extrapolation errors. This concept of local ROMs should not be confused with another type of local (or localized) ROMs described in [79], where the ROMs are associated to subdomains of the computational domain, in the spirit of domain decomposition techniques.

- Nonlinear manifold ROM methods [32, 80, 33] learn a nonlinear embedding and project the governing equations onto the corresponding approximation manifold, by means of a nonlinear function mapping a low-dimensional latent space to the solution space. This function is the decoder of an undercomplete autoencoder trained with the mean squared error loss to compress the snapshots and reconstruct them from their compressed representations. In this way, the nonlinear manifold is approximated with one single nonlinear ROM. Classical linear ROMs are obtained when the autoencoder has only one hidden-layer with linear activation functions. In this case, the decoder simply returns a linear combination of the POD modes.

2.4. Dictionaries of local reduced-order models

This paper focuses on dictionaries of ROMs, where the solution manifold is partitioned to get a collection of subsets $\mathcal{M}_k \subset \mathcal{M}$ that can be covered by a dictionary of low-dimensional subspaces, enabling the use of linear ROMs. If $\{\mathcal{M}_k\}_{k\in[\![1;K]\!]}$ is a partition of $\mathcal{M}$, then:

$$\forall k \in [\![1;K]\!],\ \forall N \in \mathbb{N}^*, \quad d_N(\mathcal{M}_k) \leq d_N(\mathcal{M}). \tag{12}$$

For a given number K of subsets, two partitions can be compared on the basis of the ratios $d_N(\mathcal{M}_k)/d_N(\mathcal{M})$. The idea of clustering training data to define local ROBs traces back to the work of D. Amsallem, K. Washabaug, M.J. Zahr and C. Farhat in 2012, published in [17, 18] and validated on nonlinear problems in computational fluid dynamics and fluid-structure-electric interactions. In these papers, the set of snapshots is partitioned with k-means clustering to define K clusters represented by their means $\{\overline{u}_k\}_{1\leq k\leq K}$. One local ROB is computed for each cluster using the POD. In the exploitation phase, given the solution at the i-th time increment, one looks for the closest mean $\overline{u}_k$ in terms of the norm $\|.\|_{L^2(\Omega)}$ and computes the state of the solution at the $i+1$-th time increment with the corresponding local ROB. This technique has been used more recently in a hyper-reduction framework in [77, 38].

When using a clustering algorithm to partition the solution manifold, the quality of the partition is related to the choice of the clustering method and the dissimilarity measure δ used to group similar solutions on the manifold. Among physics-informed clustering strategies, *i.e.* strategies incorporating simulation data to compute dissimilarities, [17, 18, 77, 78, 38] used k-means with Euclidean distances in the solution space or in a subspace of the solution space found by PCA, [48] used k-medoids with the Grassmann distance between subspaces spanned by the trajectories of the solutions, [47] applied a hierarchical partitioning based on a binary tree structure with the projection error as dissimilarity criterion, and [74] proposed working on the governing equations' nonlinear term, using either a variant

of k-means with the DEIM [81] residual as clustering criterion or k-means on a low-dimensional representation of the governing equations' nonlinear term obtained by a DEIM-based feature selection. It is recalled that k-means is a representative-based clustering algorithm equipped with the Euclidean distance, and that changing this distance leads to other clustering methods. The Local Decomposition Method [82] also relies on a physics-informed clustering strategy even though no dissimilarity measure is used, because a Gaussian mixture model is applied to shock sensors computed from the field of a quantity of interest, which enables separating subsonic and transonic flows in computational fluid dynamics.

2.5. Dictionary-based ROM-nets

The use of ROM dictionaries introduces the need for a model selection method that identifies the most suitable model in the dictionary. In [17, 18, 77, 38], the local ROM is selected by finding the closest cluster representative from the current state of the solution with the Euclidean distance. When model selection is not straightforward and slows down the simulation process, one can use a classifier to learn the model selection task and enable fast model recommendation in the exploitation phase. In [83], global POD-bases for inputs and outputs of a black-box simulation model are constructed, classifiers are trained in the form of self-organizing maps in the POD coefficients space, and local surrogate models are trained – while keeping a reduced representation on a global POD basis. Dictionaries of ROMs with automatic model recommendation made by a classifier can be found in [78, 74, 48, 75, 76]. To our knowledge, the idea of combining physics-informed clustering for the definition of local ROMs with a classifier for model recommendation came from the pioneering works of Peherstorfer, Butnaru, Willcox, and Bungartz on the Localized Discrete Empirical Interpolation Method (LDEIM [74], 2014) and of Nguyen, Barhli, Muñoz and Ryckelynck on computer vision [78] in 2018. When the classification task is performed by deep neural networks and takes the parameters as inputs, this methodology is known as *dictionary-based ROM-net* [48], see Figure 1.

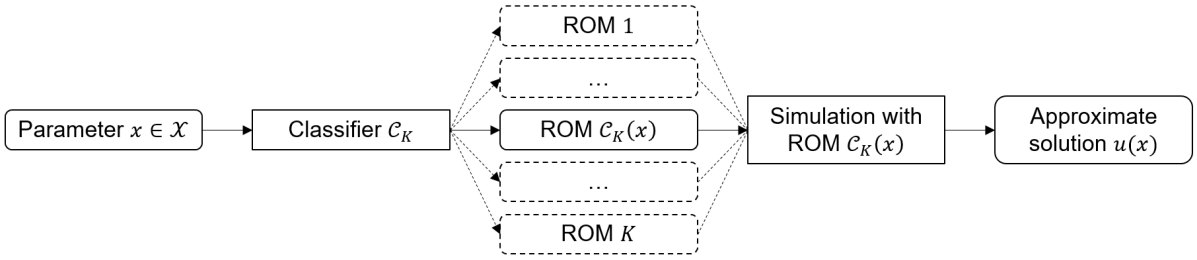

Fig. 1: Exploitation phase of a ROM-net: a dictionary of K local ROMs combined with a classifier $\mathcal{C}_K$ for automatic model recommendation. The classifier takes a point x of the parameter space as an input an returns the label of the most suitable ROM in the dictionary.

Dictionary-based ROM-nets consist in a dictionary of local ROMs and a classifier acting as a model selector, which enables the automatic adaptation of the ROM to the state and the environment of the physical system. The ROM-net's classifier (*real classifier* denoted by $\mathcal{C}_K$ where K is the number of local ROMs) approximates the theoretical *perfect classifier* $\mathcal{K}_K$ returning the index of the best local ROM for a given point in the parameter space. For simplicity, the term *dictionary-based ROM-net* (or simply *ROM-net*) is used throughout this paper to refer to the methodology described by Figures 1 and 2 and Algorithm 1, even when the classifier is not an artificial neural network.

Figure 2 gives the main steps of the training phase of a dictionary-based ROM-net and draws a comparison with the construction of a global ROM benefiting from the ROM-net's physics-informed cluster analysis. The dictionary of local ROMs is built from clusters given by a physics-informed clustering procedure. First, a simplified version of the physics problem is solved for each input example of the training database. The simplified physics problem must be less computationally demanding than the target problem; such simplification is problem-dependent and likely to be constructed by an expert of the considered application. In particular, it can be solved with a coarse mesh to reduce the dimension of the approximation space. The simplified simulations provide what we call *simplified snapshots*: these snapshots cannot be exploited to build ROMs, but they give information about how the physical system reacts to parameter changes. The clustering algorithm finds clusters from the information contained in these simplified snapshots. In light of the clustering results, one must identify a few relevant training examples for which the target problem is solved to get *high-fidelity snapshots*, that is, snapshots that well represent the solution manifold and can then be used for the construction of the local ROMs. The different needs in terms of training data for reduced-order modeling and machine learning can be seen through these two families of snapshots: the clustering and classification

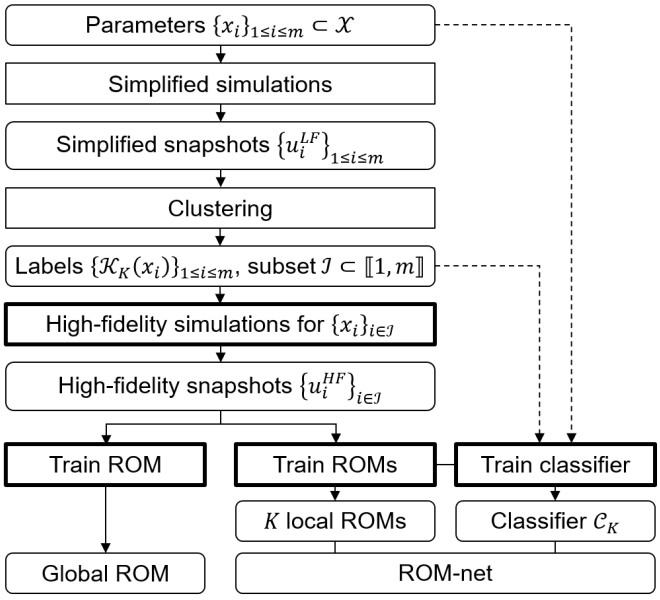

Fig. 2: Training phases of a dictionary-based ROM-net and a global ROM with a physics-informed clustering strategy.

algorithms use information related to the simplified snapshots to get a sufficiently large training set, while the ROMs use a limited number of high-fidelity snapshots in order to learn to make predictions in a physics problem. This distinction between these two types of simulation data is essential when considering complex problems with many degrees of freedom. The three major differences between our work and the seminal works of [78] and [74] are the use of simplified simulations, the clustering strategy with a new ROM-oriented dissimilarity measure, and an a priori efficiency criterion introduced hereinafter. It is noteworthy that, among the four variants of the LDEIM, the parameter-based LDEIM with clustering of snapshots (section 4.2. of [74]) is the one that shares the more similarities with our work, since it applies clustering on simulation data and uses the parameters as inputs for the classifier. The training algorithm is given in Algorithm 1.
When using a dictionary-based ROM-net, two natural questions arise:

- Which dissimilarity measure and clustering algorithm should be used for model order reduction purposes?
- Can the high-fidelity snapshots be automatically selected and how?

After clustering, the training phase of the dictionary-based ROM-net still includes expensive steps corresponding to boxes with thick lines in Figure 2, namely the computation of the high-fidelity snapshots, the construction of the local ROMs (which can involve a hyper-reduction algorithm), and the training of a classifier for automatic model recommendation. Therefore, an evaluation criterion is needed in order to assess the quality of the clusters before continuing the ROM-net's training phase, *i.e.* right after the clustering step, see Figure 2. This criterion should enable the evaluation of the profitability of the ROM-net and the tuning of clustering hyperparameters, using the simplified snapshots only. Put briefly, in addition to the two aforementioned questions, this work must also address the following issues:

- Is it possible to define a simple practical method to select good hyperparameters (number of clusters, number of POD modes, number of high-fidelity snapshots)?
- Can one define an efficiency criterion computable after the clustering step to evaluate the expected performances of the ROM-net with respect to a single global ROM?

Remark 2.3. *Dictionary-based ROM-nets use machine learning to assist model order reduction procedures in the training and exploitation phases. It does not replace physics models by regression models, since numerical predictions made by a ROM are obtained by solving physics equations.*

Remark 2.4. *This paper focuses on the choice of the dissimilarity measure in the clustering task for the construction of the dictionary of ROMs, and on hyperparameters calibration. Our article [84] gives more details about the other important component of a dictionary-based ROM-net, namely the classifier designed for automatic model recommendation.*

Algorithm 1 Training algorithm for dictionary-based ROM-nets

Input: Points $\{x_i\}_{1\leq i\leq m}$ in the parameter space, number of clusters K, number n_s of high-fidelity snapshots per cluster.
Output: Trained dictionary-based ROM-net made of K ROMs and one classifier for automatic model recommendation.

Stage 1 (simplified simulation data generation):
Call the high-fidelity solver to compute solutions of a simplified version of the target physics problem.
Simplified snapshots $\{u_i^{LF}\}_{1\leq i\leq m}$:= solutions for $\{x_i\}_{1\leq i\leq m}$.
Stage 2 (clustering based on simulation data)
Compute sine dissimilarities between simplified snapshots.
Run a k-medoids clustering algorithm using sine dissimilarities as clustering distance, to get K clusters.
$\mathcal{K}_K(x_i)$:= label/cluster of u_i^{LF}.
Stage 3 (snapshots selection)
for $k \in [\![1; K]\!]$ **do**
    Run a k-medoids clustering algorithm on the k-th cluster to get n_s subclusters.
end for
$\mathcal{I}$:= indices of the subclusters' medoids.
Call the high-fidelity solver to compute solutions of the target physics problem for $\{x_i\}_{i\in\mathcal{I}}$.
High-fidelity snapshots $\{u_i^{HF}\}_{i\in\mathcal{I}}$:= solutions for $\{x_i\}_{i\in\mathcal{I}}$.
Stage 4 (ROM dictionary construction):
for $k \in [\![1; K]\!]$ **do**
    Build a local (hyper-)reduced-order model for the k-th cluster using its n_s high-fidelity snapshots.
end for
Stage 5 (classification for automatic model recommendation)
Train a classifier C_K on the labeled dataset $\{(x_i, \mathcal{K}_K(x_i)\}_{1\leq i\leq m}$.

3. Clustering background

3.1. Representative-based clustering

In representative-based clustering algorithms, each cluster is associated to a partitioning representative, *i.e.* a reference point that well represents the cluster's members. Clusters representatives are useful in our case, because they can be used to select the high-fidelity snapshots. Representative-based algorithms generally define the clusters thanks to the Voronoi diagram generated by the representatives, which gives clusters with high cohesion.

Definition 3.1 (Representative-based clustering)**.** *Let us consider a finite set $\{x_i\}_{1\leq i\leq m}$ of elements of a topological space $\mathcal{T}$ endowed with a dissimilarity measure δ. For a given integer $K \in [\![2; m]\!]$, representative-based clustering consists in finding K representatives $\{\tilde{x}_k\}_{1\leq k\leq K} \subset \mathcal{T}$ minimizing the objective function:*

$$\sum_{i=1}^{m} \min_{k\in[\![1;K]\!]} \delta(x_i, \tilde{x}_k)^2. \tag{13}$$

The clusters C_k are given by:

$$C_k := \{x_i \mid \delta(x_i, \tilde{x}_k) \leq \delta(x_i, \tilde{x}_l)\ \forall l \in [\![1; K]\!]\}. \tag{14}$$

When the dissimilarity measure is the Euclidean distance, the optimal representatives are the clusters' means or centroids (see [40], p.162). This problem corresponds to k-means clustering [39], where the cost function in Equation (13) corresponds to the within-cluster variance and is related to clusters inertia.

3.2. K-medoids clustering

In k-medoids, the representatives must be taken among the elements of the dataset. This restriction is particularly useful when functions of the training examples (such as mean and median) do not make sense or cannot be easily computed. It enables working with any type of data with any dissimilarity measure. The next definitions introduce the k-medoids optimization problem:

Definition 3.2 (Binary matrices)**.** *A binary matrix is a matrix whose coefficients are either* 0 *or* 1*. The set of binary matrices of size $m \times n$ is denoted by $\mathcal{B}_{m,n}$.*

Definition 3.3 (K-medoids clustering)**.** *Let us consider a finite set $\{x_i\}_{1\leq i\leq m}$ of elements of a topological space $\mathcal{T}$ endowed with a dissimilarity measure δ. For a given integer $K \in [\![2; m]\!]$, let us introduce the set $\mathcal{Z}_{m,K}$:*

$$\mathcal{Z}_{m,K} := \left\{ \mathbf{Z} \in \mathcal{B}_{m,K} \mid \sum_{k=1}^{K} z_{ik} = 1 \ \forall i \in [\![1; m]\!] \text{ and } \sum_{i=1}^{m} z_{ik} \geq 1 \ \forall k \in [\![1; K]\!] \right\}. \tag{15}$$

K-medoids clustering consists in solving the following optimization problem:

$$\mathbf{Z}^* := \underset{\mathbf{Z}\in\mathcal{Z}_{m,K}}{\arg\min} \sum_{k=1}^{K} \sum_{i=1}^{m} z_{ik} \delta(x_i, \tilde{x}_k)^2, \tag{16}$$

where the medoids $\tilde{x}_k$ are given by:

$$\tilde{x}_k := \underset{x_j\in\{x_l\}_{1\leq l\leq m}}{\arg\min} \sum_{i=1}^{m} z_{ik} \delta(x_i, x_j)^2. \tag{17}$$

This formulation of the k-medoids problem has similarities with the k-means formulation proposed in [85]. With this formulation, the definition of the clusters C_k in Equation (14) is equivalent to:

$$C_k = \{x_i \mid z_{ik}^* = 1\}. \tag{18}$$

Equation (15) defining the set $\mathcal{Z}_{m,K}$ ensures that each point is assigned to one single cluster, and that each cluster contains at least one element. Equation (17) defines the medoid of a cluster as its most central member. K-medoids is a combinatorial optimization problem, for which several heuristic approaches have been proposed to find a suboptimal solution at lower cost. The Partitioning Around Medoids (PAM [41] and Chap. 2 of [86]) is the most known algorithm. It iteratively looks for the best swap between nonmedoid points and medoids. Clustering Large Applications (CLARA [87] and Chap. 3 of [86]) applies PAM on different subsamples to reduce the computational complexity of PAM. As explained in Section 11.2.1 of [42], both PAM and CLARA algorithms can be interpreted as graph-searching problems: PAM explores the entire graph of clustering solutions, while CLARA explores a subgraph only. Clustering Large Applications based on Randomized Sampling (CLARANS [88, 89]) only considers a sample of the neighbors of the current graph node at each iteration, which enables searching over the entire graph as in PAM but at lower cost. These three algorithms have been improved recently in [90] in terms of computational complexity. Apart from these approaches, a simple and fast k-medoids algorithm has been proposed in [91] following the standard implementation of k-means, *i.e.* alternating between a cluster assignment step and updating the medoids with Equation (17). However, as explained in [90], this algorithm does not explore as many configurations as PAM does. For all these algorithms, the dissimilarities $\delta(x_i, x_j)$ are precomputed before looking for clusters.

4. Proposed local ROM approach

4.1. Motivations

Instead of considering the absolute projection error when defining the Kolmogorov N-width, one can use the relative projection error, which leads to the following definition:

Definition 4.1 (Normalized Kolmogorov N-width)**.** *Let $N \in \mathbb{N}^*$. If $\mathcal{M}$ contains at least one nonzero element, the normalized Kolmogorov N-width of the manifold $\mathcal{M}$ in the ambient Hilbert space $\mathcal{H}$ is defined by:*

$$\tilde{d}_N(\mathcal{M}) := \inf_{\mathcal{H}_N\in \mathrm{Gr}(N,\mathcal{H})} \sup_{u\in\mathcal{M}\setminus\{0\}} \inf_{v\in\mathcal{H}_N} \frac{\|u - v\|_{\mathcal{H}}}{\|u\|_{\mathcal{H}}}. \tag{19}$$

Let $\measuredangle_{\mathcal{H}}(u,v) \in [0;\pi/2]$ denote the angle between two nonzero elements u and v of $\mathcal{H}$:

$$\measuredangle_{\mathcal{H}}(u,v) := \arccos\left(\frac{|\langle u,v\rangle_{\mathcal{H}}|}{\|u\|_{\mathcal{H}}\|v\|_{\mathcal{H}}}\right), \tag{20}$$

and let $\measuredangle_{\mathcal{H}}(u,\mathcal{V}) \in [0;\pi/2]$ be the angle between u and a subspace $\mathcal{V} \subset \mathcal{H}$:

$$\measuredangle_{\mathcal{H}}(u,\mathcal{V}) := \inf_{v\in\mathcal{V}} \measuredangle_{\mathcal{H}}(u,v). \tag{21}$$

The normalized Kolmogorov N-width is related to the largest angle between elements of the solution manifold and the approximation space:

Property 4.2. *Let $N \in \mathbb{N}^*$, and suppose that $\mathcal{M}$ contains at least one nonzero element. Then:*

$$\tilde{d}_N(\mathcal{M}) = \inf_{\mathcal{H}_N\in \mathrm{Gr}(N,\mathcal{H})} \sup_{u\in\mathcal{M}\setminus\{0\}} \sin\measuredangle_{\mathcal{H}}(u,\mathcal{H}_N). \tag{22}$$

The proof of this property is given in Appendix A, with another property linking the normalized Kolmogorov width $\tilde{d}_N(\mathcal{M})$ with the absolute Kolmogorov width $d_N(\mathcal{M})$ via an inequality. As suggested by Equation (22) of Property 4.2, the dissimilarity measure should be defined as a function of the angle between elements of the solution manifold in order to focus on the shape of the fields $u \in \mathcal{M}$ rather than their intensities. In this way, clustering would efficiently decrease projection errors by limiting the maximum angular deviations within clusters. The Euclidean distance $\|u-v\|_{\mathcal{H}}$ used in [17, 18, 77, 78, 38] does not always ensure the reduction of projection errors. Indeed, the solution manifold can contain solutions that are relatively close in terms of the Euclidean distance but distributed in many different directions of the space $\mathcal{H}$. On the other hand, having a subset $\mathcal{M}_k$ with a large diameter in terms of the Euclidean distance is not a problem if it is embedded in a low-dimensional space, as indicated by Property 4.2. Let us suppose that the solution manifold contains two elements u and v having disjoint supports $\mathrm{supp}(u)$ and $\mathrm{supp}(v)$ and such that there exists a large real number λ such that λu is still in the solution manifold (see Figure 3). The elements u and λu are aligned in the same direction and could then be obtained with the same 1-dimensional approximation space. However, if λ is large enough, the distance $\|u-\lambda u\|_{\mathcal{H}}$ can be very large with respect to $\|u-v\|_{\mathcal{H}}$. In this case, it is possible to assign u and v to the same cluster while assigning λu to another, whereas u and λu are aligned along a direction that is orthogonal to v. For these reasons, the Euclidean distance does not seem to be adapted, except if the number K of clusters is large enough to get very local subsets $\mathcal{M}_k$ with restricted angular deviations.

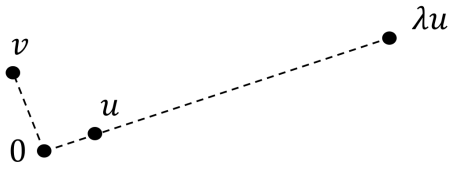

Fig. 3: Clustering with the Euclidean distance would assign u and v to the same cluster and λu to another, whereas u and λu could be computed with the same 1D approximation space.

A more natural and straightforward approach would consist in clustering the parameter space $\mathcal{X}$ to define the subsets $\mathcal{M}_k = u(\mathcal{X}_k)$ for each cluster $\mathcal{X}_k$. Note that the subsets $\mathcal{M}_k$ no longer form a partition of $\mathcal{M}$, although their union still equals to $\mathcal{M}$. This strategy may not be appropriate when u is a nonlinear function of the parameters $x \in \mathcal{X}$. The physics of the underlying problem can also generate situations where small changes of the parameters in some directions of the parameter space totally modifies the shape of the solution in a nonlinear way, while large variations in other directions of the parameter space only imply linear variations. An example is given in [48], where it is shown that clusters identified in the parameter space give subsets $\mathcal{M}_k$ spreading all over the solution manifold $\mathcal{M}$. To avoid this issue, it is preferable to apply a physics-informed clustering strategy by partitioning the solution manifold directly with an appropriate dissimilarity measure δ.

4.2. Proposed dissimilarity measure

This section introduces the dissimilarity measure used in this paper for clustering and gives some of its properties. It is important to stress that this dissimilarity is computed from the simplified snapshots u^{LF} given by the simplified simulations. Hence, in this section, the notation $u^{LF} \in L^2(\Omega \times [0; t_f])$ represents a simplified snapshot.

Definition 4.3 (Principal angles between subspaces)**.** *Let $\mathcal{V}_1$ and $\mathcal{V}_2$ be two subspaces of $L^2(\Omega)$. The principal angles or canonical angles $\theta_k(\mathcal{V}_1, \mathcal{V}_2) \in [0; \pi/2]$ between $\mathcal{V}_1$ and $\mathcal{V}_2$ are defined by:*

$$\forall k \in \mathbb{N}^*, \quad \theta_k(\mathcal{V}_1, \mathcal{V}_2) := \measuredangle(v_1^k, v_2^k), \tag{23}$$

where the angle $\measuredangle := \measuredangle_{L^2(\Omega)}$ is measured in $L^2(\Omega)$ (see Equation (20)), and where the vectors $v_1^k \in \mathcal{V}_1$ and $v_2^k \in \mathcal{V}_2$ are given by the following sequence of optimization problems:

$$\left\{ \begin{array}{rcl} (v_1^1, v_2^1) & \in & \underset{(v_1,v_2)\in\mathcal{V}_1\times\mathcal{V}_2}{\arg\min} \ \measuredangle(v_1, v_2) \\ (v_1^{k+1}, v_2^{k+1}) & \in & \arg\min \left\{ \measuredangle(v_1, v_2) \,\middle|\, v_j \in \mathcal{V}_j \cap \left(\mathrm{span}(\{v_j^i\}_{1\le i\le k})\right)^{\perp}, \ j \in \{1; 2\} \right\} \end{array} \right. \tag{24}$$

with the notation $\mathcal{V}^{\perp}$ denoting the orthogonal complement of $\mathcal{V} \subset L^2(\Omega)$ in $L^2(\Omega)$.

In practice, when the spaces $\mathcal{V}_1$ and $\mathcal{V}_2$ are finite-dimensional, it can be shown (see Theorem 1 of [92]) that the principal angles are given by:

$$\forall k \in [\![1; \min(\dim(\mathcal{V}_1), \dim(\mathcal{V}_2))]\!], \quad \theta_k(\mathcal{V}_1, \mathcal{V}_2) = \arccos \sigma_k, \tag{25}$$

with $\sigma_1 \ge \sigma_2 \ge ... \ge \sigma_{\min(\dim(\mathcal{V}_1),\dim(\mathcal{V}_2))}$ being the singular values of the matrix $\mathbf{C}(\mathcal{V}_1, \mathcal{V}_2) \in \mathbb{R}^{\dim(\mathcal{V}_1)\times\dim(\mathcal{V}_2)}$ defined by:

$$C_{ij}(\mathcal{V}_1, \mathcal{V}_2) := \langle \psi_i^{(1)}, \psi_j^{(2)} \rangle_{L^2(\Omega)}, \tag{26}$$

where the functions $\psi_i^{(1)}$ (resp. $\psi_j^{(2)}$) form an orthonormal basis of $\mathcal{V}_1$ (resp. $\mathcal{V}_2$). The vector $\boldsymbol{\theta}(\mathcal{V}_1, \mathcal{V}_2)$ denotes the vector containing the principal angles between the spaces $\mathcal{V}_1$ and $\mathcal{V}_2$.

Definition 4.4 (n-dimensional elementary basis)**.** *Let $u \in L^2(\Omega \times [0; t_f])$ and $n \in [\![1; \mathcal{N}]\!]$. The n-dimensional elementary basis associated to u is the orthonormal n-frame $\Psi_n(u) \in V(n, L^2(\Omega))$ obtained by solving the POD minimization problem given in Equation (9) with the Snapshot POD algorithm, using the trajectory of u over time as a snapshot.*

Definition 4.5 (n-dimensional elementary approximation space)**.** *Let $u \in L^2(\Omega \times [0; t_f])$ and $n \in [\![1; \mathcal{N}]\!]$. The n-dimensional elementary approximation space $\mathcal{V}_n(u) \in \mathrm{Gr}(n, L^2(\Omega))$ is the subspace spanned by $\Psi_n(u)$.*

In Definition 4.4, the POD basis $\Psi_n(u)$ is used for clustering only, it is not supposed to be used for numerical simulations since it is computed from simplified snapshots. Qualitatively, the subspace $\mathcal{V}_n(u)$ spanned by this POD basis is the best n-dimensional approximation space for the trajectory of $u(., t)$ in $L^2(\Omega)$, that is to say:

$$\mathcal{V}_n(u) = \underset{\mathcal{V}_n\in\mathrm{Gr}(n,L^2(\Omega))}{\arg\min} \int_0^{t_f} \inf_{v\in\mathcal{V}_n} \|u(., t) - v\|^2_{L^2(\Omega)} \, dt. \tag{27}$$

Definition 4.6 (Chordal distance between subspaces [93], p. 140, Section 2)**.** *Let $\mathcal{H}$ be a Hilbert space, and n, m be two integers with $n \le m$. The chordal distance between subspaces $\mathcal{V}_1 \in \mathrm{Gr}(n, \mathcal{H})$ and $\mathcal{V}_2 \in \mathrm{Gr}(m, \mathcal{H})$ is defined by:*

$$d_c(\mathcal{V}_1, \mathcal{V}_2) := \|\sin \boldsymbol{\theta}(\mathcal{V}_1, \mathcal{V}_2)\|_2 = \left(\sum_{k=1}^{n} \sin^2 \theta_k(\mathcal{V}_1, \mathcal{V}_2) \right)^{1/2}. \tag{28}$$

Definition 4.7 (Sine dissimilarity between functions)**.** *Given $n \in [\![1; \mathcal{N}]\!]$, the sine dissimilarity $\tilde{\delta}_n$ between functions u and v in $L^2(\Omega \times [0; t_f])$ is defined by:*

$$\tilde{\delta}_n(u, v) := d_c(\mathcal{V}_n(u), \mathcal{V}_n(v)). \tag{29}$$

Let us now recall the definition of the orthogonal projection $\pi_{\mathcal{H}_n} : L^2(\Omega) \to L^2(\Omega)$ on a n-dimensional subspace $\mathcal{H}_n$ of $L^2(\Omega)$, with an orthonormal basis $\{\psi_k\}_{1\leq k\leq n}$:

$$\forall u \in L^2(\Omega), \quad \pi_{\mathcal{H}_n}(u) = \sum_{k=1}^{n} \langle u, \psi_k \rangle_{L^2(\Omega)} \psi_k. \tag{30}$$

The following properties give interesting interpretations of the sine dissimilarity that motivate its use for the construction of dictionaries of local ROMs. The proofs of these properties are given in Appendix B.

Property 4.8 (Sine dissimilarity and L^2 projection errors)**.** *For all $n \in [\![1; \mathcal{N}]\!]$, the sine dissimilarity is symmetric and satisfies:*

$$\forall (u, v) \in L^2(\Omega \times [0; t_f])^2, \quad \tilde{\delta}_n(u, v) = \left(\sum_{i=1}^{n} \| \psi_i(u) - \pi_{\mathcal{V}_n(v)}(\psi_i(u)) \|^2_{L^2(\Omega)} \right)^{1/2}, \tag{31}$$

with $\pi_{\mathcal{V}_n(v)}$ denoting the orthogonal projection on $\mathcal{V}_n(v)$ and where the functions $\psi_i(u) \in L^2(\Omega)$ for $i \in [\![1; n]\!]$ are the vectors of the elementary basis $\Psi_n(u)$.

Property 4.9 (Sine dissimilarity and Hilbert-Schmidt distance)**.** *For all $n \in [\![1; \mathcal{N}]\!]$, for all $(u, v) \in L^2(\Omega \times [0; t_f])^2$, the sine dissimilarity satisfies:*

$$\tilde{\delta}_n(u, v) = \frac{1}{\sqrt{2}} \| \pi_{\mathcal{V}_n(u)} - \pi_{\mathcal{V}_n(v)} \|_{HS(L^2(\Omega))}. \tag{32}$$

These properties show that the sine dissimilarity can be interpreted either in terms of projection errors, or as the Hilbert-Schmidt distance between the projections onto the elementary approximation spaces. The sine dissimilarity is therefore relevant for model order reduction methods using Galerkin projection for the computation of approximate solutions in low-dimensional approximation spaces. In addition to the proofs of the two aforementioned properties, two other mathematical results on the sine dissimilarity are given in Appendix B: we show that this dissimilarity is a pseudometric on $L^2(\Omega \times [0; t_f])$, and that it is asymptotically equivalent to the Grassmann dissimilarity used in previous papers [48, 94] for small angles. Finally, the next definition introduces the ROM-oriented dissimilarity between parameters as the sine dissimilarity between the corresponding solutions:

Definition 4.10 (ROM-oriented dissimilarity between parameters)**.** *Given $n \in [\![1; \mathcal{N}]\!]$, the ROM-oriented dissimilarity between parameters x and x' in $\mathcal{X}$ is defined by:*

$$\delta_n(x, x') := \tilde{\delta}_n \left(u^{LF}(x), u^{LF}(x') \right), \tag{33}$$

where $u^{LF} : \mathcal{X} \to L^2(\Omega \times [0; t_f])$ is either the primal variable (i.e. the solution of the physics problem) or a dual variable (i.e. an internal variable) defining a quantity of interest.

It is recalled that this dissimilarity is computed from simplified snapshots. Property 7.2 given in Appendix B implies that the ROM-oriented dissimilarity is a pseudometric on $\mathcal{X}$. Several variants of this dissimilarity can be obtained according to the definition of the variable u^{LF}. Using the primal variable should improve the quality of the POD-Galerkin approximation, since the data would be clustered according to the angles between the subspaces spanned by the trajectories of the primal solution. This would give a *method-oriented* dissimilarity, that is, a dissimilarity favoring the accuracy of the numerical method (namely model order reduction) used for numerical simulations. Using a dual variable instead would improve the quality of the Gappy-POD [95] reconstruction for the quantity of interest when hyper-reduction is used. This would define a *goal-oriented* method favoring the accuracy of numerical predictions of a quantity of interest. Of course, one could mix both strategies by taking a weighted average of these two variants of the ROM-oriented dissimilarity.

4.3. Choice of the clustering method

As an unsupervised learning task, clustering has no indisputable evaluation criterion. This is the reason why there is no hierarchy in the large variety of clustering algorithms. The algorithm must be selected according to the purpose. For model order reduction purposes, we have seen that the Kolmogorov N-width relates the physics problem's reducibility to projection errors on the approximation space, which makes the projection error a good candidate for an evaluation criterion:

Definition 4.11 (Relative projection error)**.** *Let $u \in L^2(\Omega)$ be a nonzero square-integrable function, and $\Psi = \{\psi_k\}_{1\leq 1\leq N} \in V(N, L^2(\Omega))$ be an orthonormal reduced-order basis of dimension $N \in \mathbb{N}^*$ in $L^2(\Omega)$. The relative projection error $\eta(u, \Psi)$ of u on* span(Ψ) *is given by:*

$$\eta(u, \Psi) := \frac{\|u - \sum_{k=1}^{N} \langle u, \psi_k \rangle_{L^2(\Omega)} \psi_k\|_{L^2(\Omega)}}{\|u\|_{L^2(\Omega)}}. \tag{34}$$

Remark 4.12. *The relative projection error does not depend on the choice of the orthonormal basis used to represent the subspace* span(Ψ_N)*. Therefore, the notations $\eta(u, \Psi_N)$ and $\eta(u,$* span$(\Psi_N))$ *can be used interchangeably.*

As shown in Equation (22) in Property 4.2, Kolmogorov widths can be decreased by limiting the angular deviation within the clusters. Having defined a dissimilarity measure δ_n based on angles in Definitions 4.7 and 4.10, one must look for compact-shaped clusters in terms of the dissimilarity δ_n. Therefore, we use PAM k-medoids clustering algorithm to reduce the intra-cluster maximum angular deviations as much as possible. Our physics-informed clustering method consists in running simplified simulations and applying PAM to simulation data using the ROM-oriented dissimilarity.

When computed in $L^2(\Omega)$ and therefore with $n = 1$, the sine dissimilarity has a simple formula:

$$\forall (u, v) \in L^2(\Omega)^2, \quad \tilde{\delta}_1(u, v)_{L^2(\Omega)} := \sin \measuredangle_{L^2(\Omega)}(u, v) = \sqrt{1 - \frac{\langle u, v \rangle^2_{L^2(\Omega)}}{\|u\|^2_{L^2(\Omega)} \|v\|^2_{L^2(\Omega)}}}, \tag{35}$$

which gives a direct link with the relative projection error:

$$\tilde{\delta}_1(u, v)_{L^2(\Omega)} = \eta(u, \text{span}(\{v\})). \tag{36}$$

This formula will be used for the computation of the dissimilarity in the applications given at the end of this paper. In this setting, we showed in [96] the following property motivating the choice of k-medoids clustering:

Property 4.13 (Optimality of k-medoids clustering)**.** *The partitions $\mathcal{M}_k$ of $\mathcal{M}$ minimizing the k-medoids cost function with dissimilarity $\tilde{\delta}_1(.,.)_{L^2(\Omega)}$ are exactly the minimizers of a discretized version of the following cost function:*

$$\sum_{k=1}^{K} \mathbb{P}(u \in \mathcal{M}_k)\, \check{d}_1(p_{U|u\in\mathcal{M}_k})^2, \tag{37}$$

where $\check{d}_N$ is a variant of the Kolmogorov width obtained by replacing the worst-case error by the mean squared error as in [97]:

$$\begin{aligned} \check{d}_N(p_U) &:= \left(\inf_{\mathcal{H}_N \in Gr(N, L^2(\Omega))} \mathbb{E}_{U\sim p_U} \left[\eta(U, \mathcal{H}_N)^2 \right] \right)^{1/2} \\ &= \left(\inf_{\mathcal{H}_N \in Gr(N, L^2(\Omega))} \mathbb{E}_{X\sim p_X} \left[\eta(u(X), \mathcal{H}_N)^2 \right] \right)^{1/2}, \end{aligned} \tag{38}$$

with p_X denoting a probability density function in the parameter space and p_U being the resulting probability density function in the solution space.

4.4. Automatic snapshots selection

Once clusters have been identified within the dataset, one must select relevant points for which the entire high-fidelity simulation will be run to provide high-fidelity snapshots for the construction of the local ROMs. For each cluster, the high-fidelity snapshots must be well distributed and representative of the cluster's members. When one wants to use only one snapshot per cluster, then the clusters' medoids are good candidates. For more than one snapshot per cluster, a second k-medoids cluster analysis can be conducted within each cluster, with n_s subclusters where n_s is the desired number of high-fidelity snapshots per cluster, using the same dissimilarity measure as for the first clustering. High-fidelity snapshots can then be computed for the subclusters' medoids. This method corresponds to a two-stage hierarchical k-medoids clustering.

5. ROM-net's efficiency criterion and hyperparameters tuning

5.1. Gain with respect to a global reduced-order model

We recall the following notations: K is the number of clusters (or number of ROMs in the dictionary), N is the number of POD modes, and n_s is the number of snapshots per cluster used for the construction of the local ROMs.

A dictionary-based ROM-net [48] is made of a dictionary of K local ROMs and a classifier C_K which automatically selects the best model from the dictionary for a given point in the parameter space without computing any physics-informed dissimilarity, see Figure 1. The real classifier C_K enables bypassing the simplified simulation that is required to evaluate the perfect classifier $\mathcal{K}_K$, see Figure 2. In this section, it is assumed that all the dictionary's ROMs have the same number of modes, denoted by N, and have been built from the same number of high-fidelity snapshots, denoted by n_s. A dictionary of K ROBs with N modes and n_s high-fidelity snapshots per basis is denoted by $\{\Psi_k^{(K,N,n_s)}\}_{k\in[\![1;K]\!]}$. The objective of this section is to define a practical method for the calibration of the hyperparameters K, N and n_s, based on an evaluation criterion quantifying the ROM-net's profitability with respect to a single global ROM. This criterion must be computable very early in the ROM-net's training phase, right after the physics-informed clustering procedure in Figure 2 and before the computation of high-fidelity snapshots, the construction of the ROMs, and the classifier's training phase. Therefore, the local ROBs $\{\Psi_k^{(K,N,n_s)}\}_{k\in[\![1;K]\!]}$ used in the evaluation criterion are simply built from Kn_s simplified snapshots selected by the two-stage hierarchical k-medoids clustering, instead of the corresponding high-fidelity snapshots that will be computed afterwards. Their performances are compared with the performance of a global ROB $\Psi_g^{(1,N,Kn_s)}$ containing N modes and inferred from the same Kn_s snapshots as $\{\Psi_k^{(K,N,n_s)}\}_{k\in[\![1;K]\!]}$. This global basis thus benefits from the physics-informed clustering procedure for the selection of its snapshots. The ROBs are related to the function $u^{LF}(X) \in L^2(\Omega\times[0;t_f])$ parametrized by the random variable X representing the current point in the parameter space. The following definition introduces the *gain* used in our evaluation criterion:

Definition 5.1 (Gain)**.** *Given integers $K > 1$, $N > 0$ and $n_s > 0$ and a classifier $\mathcal{F}_K : \mathcal{X} \to [\![1;K]\!]$, the gain is defined by:*

$$G(X;K,N,n_s,\mathcal{F}_K) = \frac{\eta\left(u^{LF}(X),\Psi_g^{(1,N,Kn_s)}\right)}{\eta\left(u^{LF}(X),\Psi_{\mathcal{F}_K(X)}^{(K,N,n_s)}\right)}, \tag{39}$$

where $u^{LF}(X)$ results from a simplified simulation. For $K = 1$, the gain equals to 1.

Remark 5.2. *In Definition 5.1, the primal variable u can be replaced by a quantity of interest, depending on the choice made for the definition of the ROM-oriented dissimilarity.*

As a function of X, the gain can be seen as a random variable parametrized by the hyperparameters K, N and n_s and the classifier. The notations $G_C(K,N,n_s)$ and $G_{\mathcal{K}}(K,N,n_s)$ denote $G(X;K,N,n_s,C_K)$ and $G(X;K,N,n_s,\mathcal{K}_K)$ respectively. Right after the physics-informed clustering procedure, the user cannot evaluate the gain $G_C(K,N,n_s)$ since the real classifier C_K has not been trained yet. However, the clusters implicitly define the perfect classifier $\mathcal{K}_K$ and thus the user has access to values of the gain $G_{\mathcal{K}}(K,N,n_s)$. In the next property, the following assumption is made:

[A1] The gain $G_{\mathcal{K}}(K,N,n_s)$ is assumed to be deterministic, which means that it is no longer a random variable but rather a deterministic function of the hyperparameters K, N and n_s. In other words, when the right cluster is chosen, the gain does not depend on X.

Property 5.3 (Gain decomposition)**.** *Under assumption [A1]:*

$$\mathbb{E}[G_C(K,N,n_s)] = p\,G_{\mathcal{K}}(K,N,n_s) + (1-p)\,E(K,N,n_s), \tag{40}$$

where $p = \mathbb{P}(C_K = \mathcal{K}_K)$ is the classification accuracy and $E(K,N,n_s)$ given by:

$$E(K,N,n_s) := \mathbb{E}[G_C(K,N,n_s)\,|\,C_K \neq \mathcal{K}_K] \tag{41}$$

is the conditional expectation of the gain $G_C(K,N,n_s)$ when selecting the wrong ROB.

Proof. The expected gain $\mathbb{E}[G_C(K,N,n_s)]$ satisfies:

$$\mathbb{E}[G_C(K,N,n_s)] = p\,\mathbb{E}[G_C(K,N,n_s)\,|\,C_K = \mathcal{K}_K] + (1-p)\,E(K,N,n_s). \tag{42}$$

If $G_{\mathcal{K}}(K,N,n_s)$ is constant for fixed hyperparameters (K,N,n_s), then:

$$G_{\mathcal{K}}(K,N,n_s) = \mathbb{E}[G_{\mathcal{K}}(K,N,n_s)\,|\,C_K = \mathcal{K}_K] = \mathbb{E}[G_C(K,N,n_s)\,|\,C_K = \mathcal{K}_K], \tag{43}$$

because the gains $G_{\mathcal{K}}(K,N,n_s)$ and $G_C(K,N,n_s)$ return the same values when the real classifier C_K selects the right ROB. Replacing $\mathbb{E}[G_C(K,N,n_s)\,|\,C_K = \mathcal{K}_K]$ by $G_{\mathcal{K}}(K,N,n_s)$ in Equation (42) ends this proof. □

Two additional assumptions are made in what follows:

[A2] The classification accuracy p is modeled as a decreasing function of the number of clusters K defined on a finite interval $[\![1;K_{\max}]\!]$. Indeed, for a fixed number of training examples, increasing the number of classes K makes the classification task more complicated. When the number of classes is too large in comparison with the number of training data, the classifier hardly improves the performance of a random guess classifier.

[A3] The conditional expectation $E(K,N,n_s)$ is constant, meaning that the expected gain when choosing the wrong ROB does not depend on the hyperparameters K, N and n_s. For all K,N,n_s:

$$\begin{aligned} E := E(K,N,n_s) &= \mathbb{E}[G_C(K,N,n_s)\,\big|\,C_K \neq \mathcal{K}_K] \\ &\leq \mathbb{E}[G_{\mathcal{K}}(K,N,n_s)\,\big|\,C_K \neq \mathcal{K}_K] = G_{\mathcal{K}}(K,N,n_s), \end{aligned} \tag{44}$$

so $E \leq G_{\mathcal{K}}(1,N,n_s) = 1$ in particular. In the application presented in the last section of this paper, we take $E = 0.75$.

The next definition introduces the concept of *real profitability* for a dictionary-based ROM-net:

Definition 5.4 (Real ROM-net profitability)**.** *Given integers $K > 1$, $N > 0$ and $n_s > 0$, a dictionary-based ROM-net with classifier C_K and ROM dictionary $\{\Psi_k^{(K,N,n_s)}\}_{k\in[\![1;K]\!]}$ is profitable with a real profit $G_r^* \in \mathbb{R}_+$ if its expected gain satisfies $\mathbb{E}[G_C(K,N,n_s)] \geq G_r^*$.*

This means that, on average, projection errors made by a global ROB are G_r^* times larger than those made by the ROM-net, even when classification errors are taken into account. However, the ROM-net profitability cannot be evaluated a priori on $\mathbb{E}[G_C(K,N,n_s)]$, since the real classifier has not been trained yet and the dictionary of ROBs $\{\Psi_k^{(K,N,n_s)}\}_{k\in[\![1;K]\!]}$ inferred from high-fidelity snapshots have not been computed yet, see Figure 2. For these reasons, the following definition introduces the concept of *perfect profitability*:

Definition 5.5 (Perfect ROM-net profitability)**.** *Given integers $K > 1$, $N > 0$ and $n_s > 0$, a dictionary-based ROM-net with perfect classifier $\mathcal{K}_K$ and ROM dictionary $\{\Psi_k^{(K,N,n_s)}\}_{k\in[\![1;K]\!]}$ is perfectly profitable with a perfect profit $G_p^* \in \mathbb{R}_+$ if $\mathbb{E}[G_{\mathcal{K}}(K,N,n_s)] \geq G_p^*$.*

Property 5.6. *Let $G_r^* \in \mathbb{R}_+$. Let us consider a dictionary-based ROM-net with hyperparameters K, N, n_s. Under assumptions [A1], [A2] and [A3], the dictionary-based ROM-net is profitable with real profit G_r^* if and only if it is perfectly profitable with the following perfect profit:*

$$G_p^*(G_r^*) = \frac{G_r^* - (1 - p(K))E}{p(K)}. \tag{45}$$

Proof. It is a direct consequence of the gain decomposition property (Property 5.3). □

When the gains are computed with the results of the simplified simulations and with ROBs inferred from simplified snapshots, the dictionary-based ROM-net is said to be *a priori profitable* with real profit $G_r^* > 1$ if:

$$\mathbb{E}[G_{\mathcal{K}}(K,N,n_s)] \geq \frac{G_r^* - (1 - p(K))E}{p(K)}. \tag{46}$$

The a priori profitability can be assessed early in the ROM-net training phase, right after the physics-informed clustering procedure.

5.2. Practical method

The number of clusters K, the number of POD modes N and the number of snapshots per cluster n_s are three important hyperparameters when building a dictionary-based ROM-net. Choosing a good number of clusters K may be particularly difficult. The optimal value of K is related to the nonlinearity of the solution manifold: the more curved the solution manifold $\mathcal{M}$ is, the greater K must be to cover $\mathcal{M}$ with several subspaces. It also depends on the number of POD modes N: very fast simulations would require N to be small, which would increase the number of local bases required to cover the solution manifold. Last but not least, K also has an influence on the accuracy of the ROM-net's classifier. In a classification problem, increasing the number of classes while keeping the size of the training set constant makes the learning task tougher. Hence, the performance of a dictionary-based ROM-net does not monotonically increase with K since its classifier may choose the wrong model, leading to inaccurate numerical predictions.
The hyperparameters K, N and n_s must satisfy the following requirements:

[R1] **Limited computational resources:** the total number of high-fidelity snapshots Kn_s is limited by the maximum allowable budget in terms of high-fidelity simulations of the entire physics problem.

[R2] **Speed-up factor requirements:** to effectively reduce the computational cost of high-fidelity simulations, the number N of POD modes per local ROB must not exceed $\mathcal{N}^{1/3}$.

[R3] **Accuracy requirements:** the mean projection error must be lower than a user-defined threshold η^*:

$$\mathbb{E}[\eta\left(u^{LF}(X), \Psi^{(K,N,n_s)}_{\mathcal{K}_K(X)}\right)] \leq \eta^*. \tag{47}$$

[R4] **Gain requirements:** given a user-defined threshold $G_r^* > 1$, Equation (46) for the ROM-net a priori profitability must be satisfied to ensure that the local bases give better performances than a single global ROB.

Remark 5.7 (Concerning requirement R2)**.** *After the Galerkin projection of the governing equations onto a ROB made of N modes, the linear system to be solved at each iteration of the Newton-Raphson is full and thus has a complexity of $O(N^\alpha)$ with $2 \leq \alpha \leq 3$, which must be compared with the complexity $O(\mathcal{N}^\beta)$ of the sparse linear system obtained with the finite-element method, with $1 \leq \beta \leq 2$. The worst case is obtained for $\alpha = 3$ and $\beta = 1$, which gives an upper bound in the order of $\mathcal{N}^{1/3}$ for N.*

Given these constraints, we introduce the definition of *hyperparameters admissible set*:

Definition 5.8 (Hyperparameters admissible set)**.** *The hyperparameters admissible set is defined by:*

$$\mathcal{A} = \{(K, N, n_s) \in (\mathbb{N}^*)^3 \mid \quad \text{[R1], [R2], [R3], [R4] are satisfied.}\}. \tag{48}$$

This definition gives a practical method for the ROM-net profitability analysis and hyperparameters tuning. The hyperparameters admissible set can be identified using simplified snapshots right after the clustering step in the training phase, see Figure 2. If the hyperparameters admissible set is empty, then it is not worth continuing the training phase of the dictionary-based ROM-net given the user-defined thresholds η^* and G_r^* and the maximum number of high-fidelity snapshots $n^{\max}_{\text{snapshots}}$. The user can either build a global ROB using the physics-informed clustering results to identify snapshots, or weaken some of the requirements [R1] to [R4]. The time spent for simplified simulations is not wasted: the user can justify the choice of using a global ROB, and can benefit from these simulations for high-fidelity snapshots selection. On the contrary, if the hyperparameters admissible set is not empty, then there is a benefit in using a dictionary-based ROM-net. The choice of the best hyperparameters configuration among the admissible ones depends on the user's priorities. However, given the cost of the entire training phase, a ROM-net is generally used for applications where the number of test simulations is very high, *e.g.* for parameter optimization or uncertainty quantification. In this case, once accuracy and gain requirements are met, one should take the smallest number of POD modes N to get the highest possible speed-up factor. Among the admissible configurations with the smallest number of modes, it is recommended to choose the value of K minimizing the mean projection error, to get the most accurate dictionary among the fastest admissible ones. The number of high-fidelity snapshots n_s per cluster must be fixed accordingly so that the total number Kn_s of high-fidelity snapshots remains lower than $n^{\max}_{\text{snapshots}}$.

Remark 5.9. *Choosing the smallest possible number of modes N generally implies choosing larger values for K, which usually decreases the performance of the ROM-net's classifier for automatic model recommendation. When interesting values for K are rather large (say greater than* 8*), one can artificially improve the classifier's accuracy by running several reduced simulations in parallel with the models having the highest membership probabilities. An error estimator could then be used to determine which reduced simulation is the most accurate, as proposed in [94]. Such a strategy increases the number of simulations to be run in the exploitation phase, but would enable working with large K's and thus small N's, lessening the computational complexity of online reduced simulations. In addition, when the number of training examples is not large enough compared to the number of clusters K for the classification task, the data augmentation algorithm presented in [84] for the classification of numerical simulations can be applied to reduce the risk of overfitting.*

6. Numerical applications

6.1. 1D steady heat equation

6.1.1. Problem description

Let us consider the following ordinary differential equation:

$$\left\{\begin{array}{rcll} -\left(\lambda u'\right)'(\xi) & = & s(\xi) & \forall \xi \in [0;L] \\ u(0) & = & u_0 & \\ u(L) & = & u_0 & \end{array}\right. \tag{49}$$

where $\lambda \in L^2([0;L])$, $s \in L^2([0;L])$, $u_0 \in \mathbb{R}$ and $u - u_0 \in H_0^1([0;L])$. This equation describes the thermal behavior of an heterogeneous continuous medium of length L with thermal conductivity $\lambda(\xi)$ and temperature $u(\xi)$, in the presence of a heat source $s(\xi)$. We are interested in the behavior of the solution u under variying source terms and conductivity functions. The conductivity function λ is defined by:

$$\lambda(\xi) = \lambda_1 \mathbb{1}_{\{\epsilon(\zeta)L \leq \xi \leq (\epsilon(\zeta)+\zeta)L\}} + \lambda_2 (\mathbb{1}_{\{\xi < \epsilon(\zeta)L\}} + \mathbb{1}_{\{\xi > (\epsilon(\zeta)+\zeta)L\}}), \tag{50}$$

with $\lambda_2 = 1000\lambda_1 \in \mathbb{R}_+^*$ and with ζ being a random variable following the uniform distribution $\mathcal{U}(0.1, 0.5)$. The random variable $\epsilon(\zeta)$ follows the uniform distribution $\mathcal{U}(0, 1-\zeta)$. The source term s is modeled by a zero-mean Gaussian process with an exponential covariance function. The problem described by Equation (49) is therefore parametrized by the heat source distribution s and the microstructural parameters ζ and $\epsilon(\zeta)$. The weak formulation of Equation (49) reads:

$$\int_0^L \lambda(\xi) v'(\xi) u'(\xi) d\xi = \int_0^L s(\xi) v(\xi) d\xi \quad \forall v \in H_0^1([0;L]). \tag{51}$$

The interval $[0;L]$ is discretized into $\mathcal{N} - 1 = 1999$ subdivisions of length $h = L/(\mathcal{N}-1)$. The vertices $\{\xi_i = ih\}_{0 \leq i \leq \mathcal{N}-1}$ define a finite-element mesh whose P1 shape functions are denoted by $\{\phi_i\}_{1 \leq i \leq \mathcal{N}-2}$. The shape functions ϕ_0 and $\phi_{\mathcal{N}-1}$ are not used because of the Dirichlet boundary conditions. The finite-element method computes a high-fidelity approximate solution $u - u_0$ in the space span $(\{\phi_i\}_{1 \leq i \leq \mathcal{N}-2})$, whose coordinates are stored in a vector $\mathbf{q} \in \mathbb{R}^{\mathcal{N}-2}$. This vector is the solution of the following linear system:

$$\mathbf{K}\mathbf{q} = \mathbf{f}, \tag{52}$$

with $\mathbf{K} \in \mathbb{R}^{(\mathcal{N}-2)\times(\mathcal{N}-2)}$ given by:

$$K_{ij} = \int_0^L \lambda(\xi) \phi_i'(\xi) \phi_j'(\xi) d\xi \quad \forall (i,j) \in [\![1; \mathcal{N}-2]\!], \tag{53}$$

and $\mathbf{f} \in \mathbb{R}^{\mathcal{N}-2}$ given by:

$$f_i = \int_0^L s(\xi) \phi_i(\xi) d\xi \quad \forall (i,j) \in [\![1; \mathcal{N}-2]\!]. \tag{54}$$

Both $\mathbf{K}$ and $\mathbf{f}$ are recomputed when considering a new realization of the random source term and conductivity function.

A dataset of 1000 realizations of the random source term and microstructural parameters is generated. For each example in the dataset, the finite-element solution $\mathbf{q}$ is computed with a Python routine. Figure 4 shows the solution's

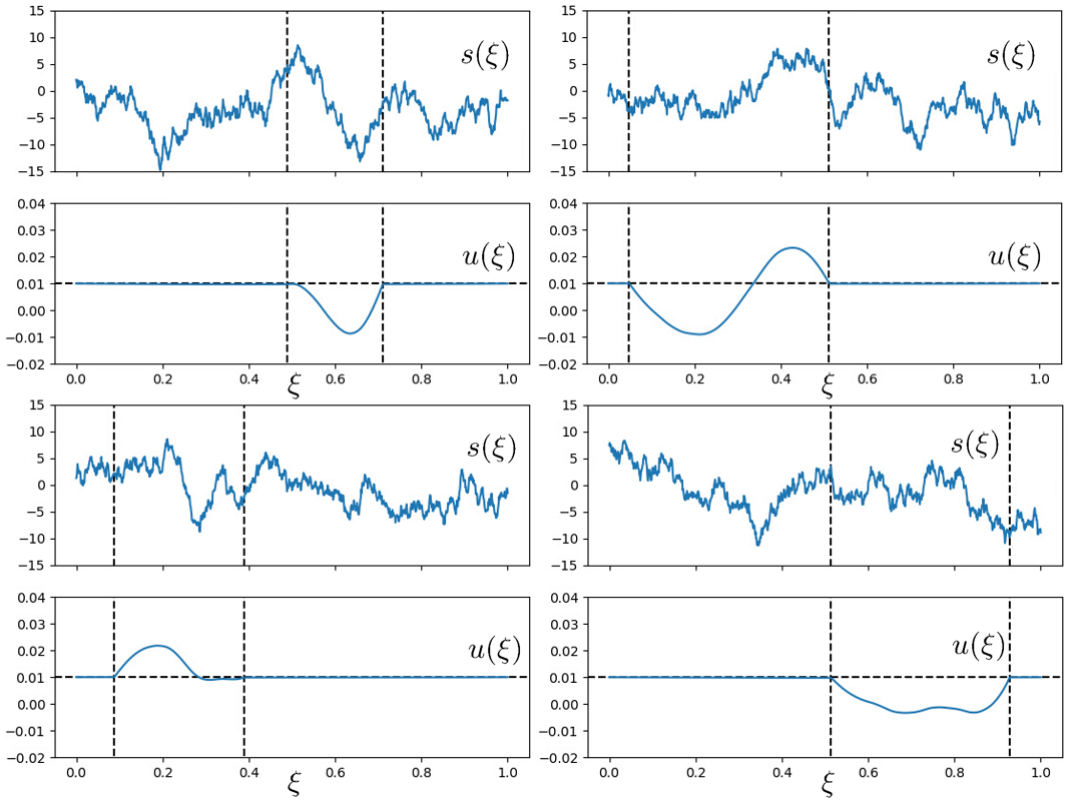

Fig. 4: Examples of solutions $u(\xi)$ for different source terms $s(\xi)$. The vertical dashed lines indicate the locations of the interfaces between the constituents of the bimaterial. Between the two interfaces, the thermal conductivity is λ_1. Outside of this interval, the thermal conductivity is $\lambda_2 = 1000\lambda_1$.

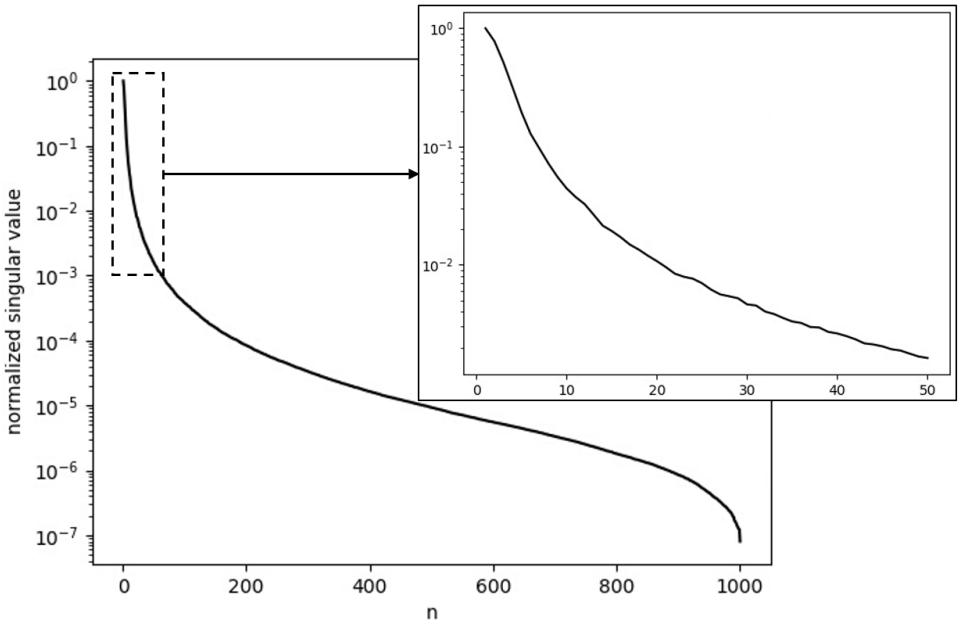

Fig. 5: Decay of the singular values obtained by singular value decomposition on the matrix containing all the examples in the dataset.

behavior for different configurations. One can observe that the solution is not affected by the source term in high-conductivity regions. Figure 5 gives the singular values of the matrix containing the 1000 solutions. It can be observed that the decay of the singular values is rather slow for a 1D problem, meaning that this problem is non-reducible and that a dictionary of local ROBs may be required. The database is splitted into two subsets: a training set and a test set, both containing 500 examples. The training set is used to identify clusters and build the ROBs, while the test set is used for evaluation purposes.

Remark 6.1. *In the training phase of a dictionary-based ROM-net for time-dependent physics problems, the simplified problem that is simulated to provide data for the clustering procedure generally corresponds to a few time steps of the target problem. In this example, Equation* (49) *does not define a time-dependent problem. In this case, the simplified problem can be defined as the target problem solved on a coarse finite-element mesh.*

6.1.2. Hyperparameters calibration

This section deals with the calibration of the hyperparameters (K, N, n_s). K-medoids is applied on the training set with the ROM-oriented dissimilarity δ_n introduced in Equation (33). Since Equation (49) is time-independent, one must take $n = 1$ for the ROM-oriented dissimilarity. We simply use the notation δ instead of δ_1 for the ROM-oriented dissimilarity obtained by computing the sine dissimilarity in the solution space. The ROBs $\Psi_1(u)$ and $\Psi_1(v)$ used to calculate $\delta(u, v)$ are obtained by normalizing the solutions u and v. For clustering, we use our own implementation of PAM [41, 86] k-medoids algorithm, with multiple random initializations for the medoids.

The physics problem considered in this section gives only one field u per set of parameters. Therefore, the number of POD modes N is necessarily lower than or equal to the number of high-fidelity snapshots per cluster n_s. For simplification purposes, we take $N = n_s$. Given that $\mathcal{N} = 2000$, the number of POD modes N must be lower than $\lfloor \mathcal{N}^{1/3} \rfloor = 12$. To effectively reduce the computational cost of high-fidelity simulations, the maximum number of modes considered in this paper is $N = 5$.

Let us say that we are given a budget of 20 high-fidelity simulations. The hyperparameters must satisfy the inequality $Kn_s = KN \leq 20$. Our thresholds for the mean projection error and the mean gain are $\eta^* = 0.35$ and $G_r^* = 2$. A polynomial of degree 2 is considered for the model $p(K)$ for the classification accuracy, and its coefficients are determined by imposing $p(1) = 1$, $p(6) = 0.8$ (value taken from [48]), $p(K_{\max}) = 1/K_{\max}$ (accuracy of a random guess for balanced classes) and $K_{\max} = 20$.

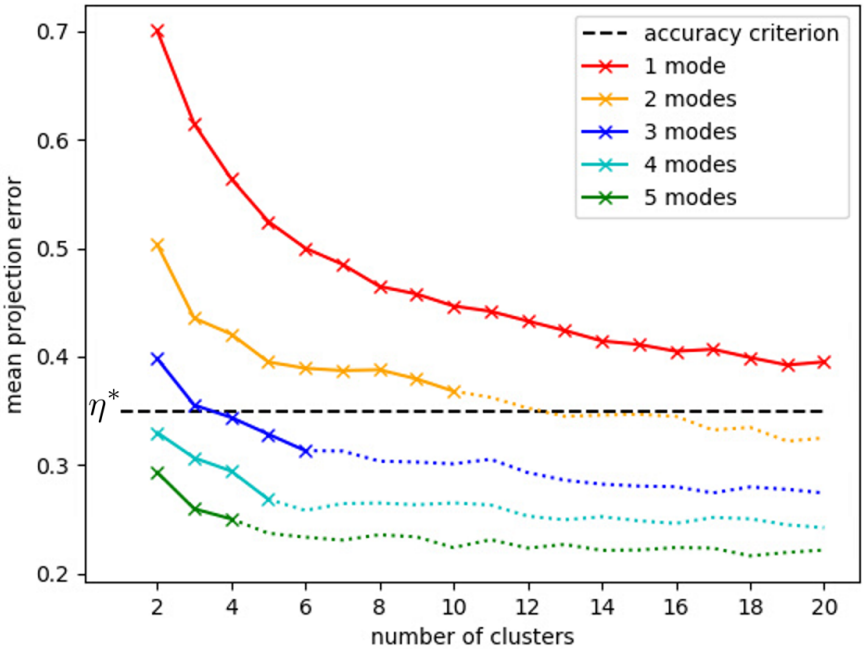

Fig. 6: Mean projection error as a function of the number of clusters K for different number of modes N. Dotted lines indicate the configurations which do not comply with the allocated number of high-fidelity snapshots.

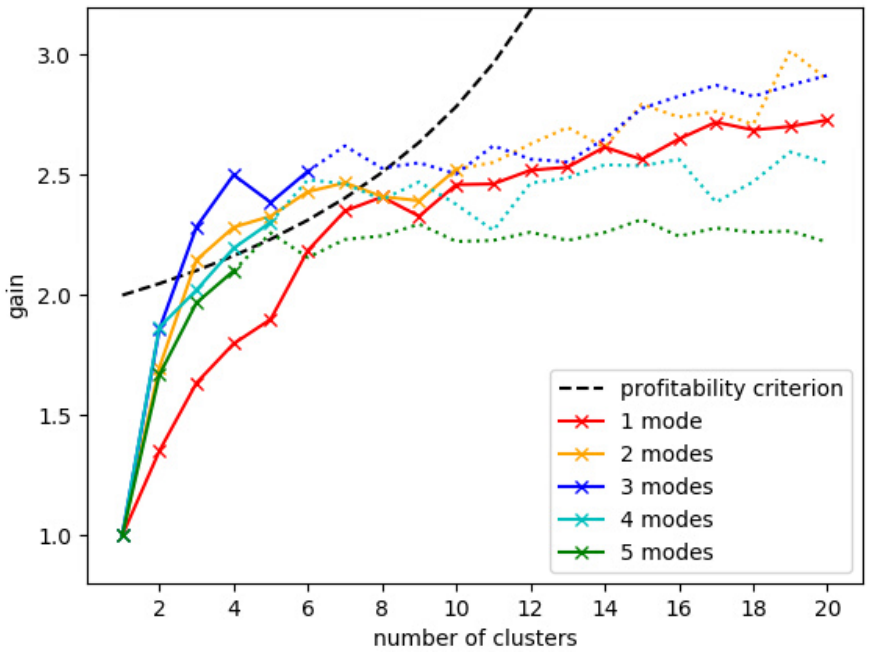

Fig. 7: Gain as a function of the number of clusters K for different number of modes N. Dotted lines indicate the configurations which do not comply with the allocated number of high-fidelity snapshots.

Figure 6 gives the mean projection error as a function of K and N. For $N = 4$ and $N = 5$ modes, the mean projection errors are below the tolerance η^* for all K. For $N = 3$, the accuracy criterion is satisfied for $K \geq 4$. The mean projection error for $N = 2$ modes falls below the tolerance for $K \geq 13$, which does not conform to the constraint $K \leq 10$ imposed by the allocated number of high-fidelity snapshots. With $N = 1$ mode, the mean projection error remains too large, which rejects configurations with $N = 1$. The configurations (K, N) satisfying the accuracy criterion and respecting the budget for high-fidelity snapshots are $K = 4, 5, 6$ for $N = 3$, $K = 2, 3, 4, 5$ for $N = 4$, and $K = 2, 3, 4$ for $N = 5$.

The gain curves are given in Figure 7. The dashed line in black delimits the ROM-net's profitability domain: configurations under this curve are irrelevant, either because the corresponding expected gain is too low, or because misclassification errors would be too frequent because too many classes are considered. The configuration $(K, N) = (5, 5)$ meets both gain and accuracy requirements, but violates the constraint $K \leq 4$ for $N = 5$ and thus requires too many high-fidelity snapshots. For $(K, N) = (3, 3)$, the gain is large enough but the mean projection error is larger than the tolerance, as seen in Figure 6. Finally, the admissible configurations are $K = 4, 5, 6$ for $N = 3$, and $K = 4, 5$ for $N = 4$. The hyperparameters admissible set is represented in Figure 8. Among the admissible configurations, those with $N = 3$ are more interesting in terms of speed of online reduced simulations. The lowest mean projection error is obtained for $K = 6$ when $N = 3$, see Figure 6. Therefore, we choose the hyperparameters $(K, N, n_s) = (6, 3, 3)$, corresponding to the lower right dot in Figure 8.

Remark 6.2. *It has been decided to take the configuration with the best accuracy among the admissible configurations with the smallest value for N, in order to have a simple and systematic approach for hyperparameters calibration. However, in the present example, one could also use the elbow method. The elbow method is commonly used for selecting the number of clusters for k-means clustering or the number of components for a PCA. It consists in choosing the elbow or knee point of the curve of an evaluation criterion. In spite of the difficulties of defining clearly the elbow point in some situations, this method raises interesting questions. In our example, if one uses the elbow method with the error curve, the best number of clusters is still* $K = 6$*: for* $N = 3$*,* $K = 6$ *is the elbow point. When using this method with the gain curve, the best number of clusters turns out to be* $K = 4$*, even when considering a smoothed version of the blue curve in Figure 7 to avoid undesirable fluctuations due to sampling and medoids initializations. Indeed, taking* $K = 5$ *or* $K = 6$ *does not significantly improve the gain when* $N = 3$*, whereas the number of high-fidelity snapshots and the complexity of the classification problem would be increased. The practical method presented in*

this paper can be adapted according to the user's priorities between training cost, online speed, accuracy, and gain.

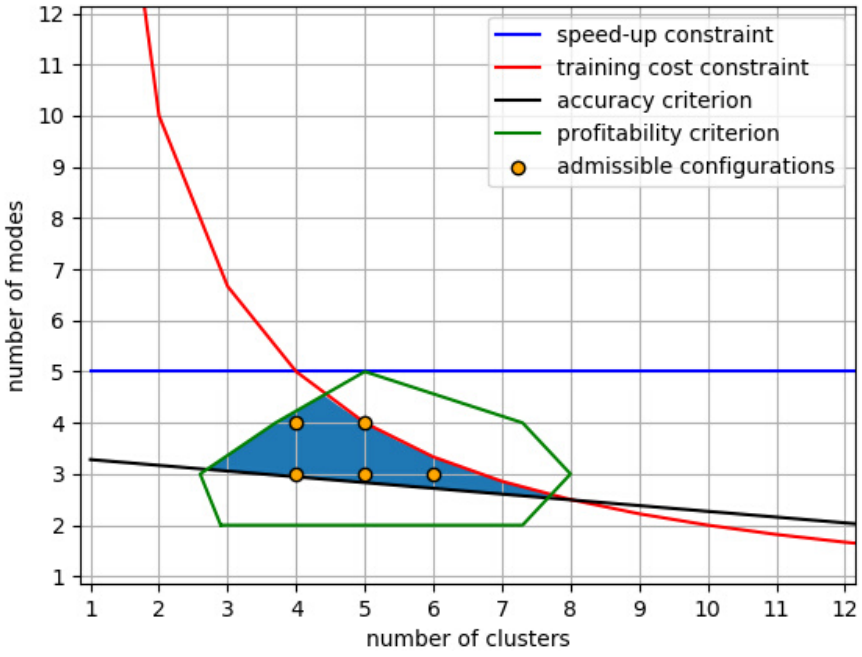

Fig. 8: Hyperparameters admissible set.

6.1.3. Comparison of different model order reduction strategies

Let $x = (s, \zeta, \epsilon(\zeta))$ denote the parameter of the problem. After projection of the source term in the finite-element basis, the parameter x is represented by a $\mathcal{N}+2$-dimensional vector $\mathbf{x}$ whose coordinates are centered and scaled to unit variance. This way, distances in the parameter space can be computed with the Euclidean distance:

$$\delta_{\mathcal{X}}(x, x') = \|\mathbf{x} - \mathbf{x}'\|_2. \tag{55}$$

Introducing the notation $\mathbf{q}(x) \in \mathbb{R}^{\mathcal{N}-2}$ for the solution of Equation (52) for a given parameter x, one can define a physics-informed dissimilarity measure $\delta_{\mathcal{H}}$ using the Euclidean distance in the solution space:

$$\delta_{\mathcal{H}}(x, x') = \|\mathbf{q}(x) - \mathbf{q}(x')\|_2. \tag{56}$$

These dissimilarity measures are compared with the ROM-oriented dissimilarity measure δ introduced in Equation (33), obtained by computing the sine dissimilarity in the solution space. K-medoids clustering is used for both snapshots selection and manifold partitioning in conjunction with one of these three dissimilarity measures. Different model order reduction strategies are compared in terms of projection errors under the following setting:

- **Equivalent number of snapshots:** all the strategies use the same total number of snapshots, which ensures equal budgets for high-fidelity simulations in the training phase. It is recalled that the high-fidelity snapshots are given by high-fidelity simulations that are more expensive than the simplified simulations used to generate the database, find clusters and evaluate their quality.
- **Equivalent number of POD modes:** all the ROBs use the same number of modes, which ensures equivalent speed-ups when exploiting the ROMs.

If a dictionary of K local ROBs is compared with a global ROB made of N modes, then each local ROB must have N modes. For the construction of these local cluster-specific ROBs, $n_s = N$ snapshots are selected in each cluster using the two-stage hierarchical k-medoids clustering procedure. Hence, the total number of snapshots is $Kn_s = KN$. Snapshots for the construction of the global ROB are therefore selected by taking the medoids of a single k-medoids clustering with KN clusters.
Six model order reduction strategies are considered, namely:

- Three global ROBs containing N modes computed from KN snapshots. The snapshots are selected thanks to a k-medoids cluster analysis with KN clusters, using different dissimilarities:
  - **Global ROM 1** uses the dissimilarity $\delta_{\mathcal{X}}$ (Euclidean distance in the parameter space).
  - **Global ROM 2** uses the dissimilarity $\delta_{\mathcal{H}}$ (Euclidean distance in the solution space).
  - **Global ROM 3** uses the ROM-oriented dissimilarity δ (sine dissimilarity in the solution space).
- Three ROM dictionaries consisting of K local ROBs with N modes each. Each local ROB is inferred from N snapshots. Again, k-medoids is applied with different dissimilarity measures:
  - **ROM dictionary 1** uses the dissimilarity $\delta_{\mathcal{X}}$ (Euclidean distance in the parameter space). This strategy is the most natural and simple one among ROM dictionaries.
  - **ROM dictionary 2** uses the dissimilarity $\delta_{\mathcal{H}}$ (Euclidean distance in the solution space) like in [17, 18, 77, 78, 38]. This strategy belongs to physics-informed strategies.
  - **ROM dictionary 3** uses the ROM-oriented dissimilarity δ (sine dissimilarity in the solution space). This is the strategy we have introduced in this paper for dictionary-based ROM-nets. Like ROM dictionary 2, it relies on a physics-informed cluster analysis, but with another dissimilarity.

In this section, the comparison is presented for $K = 6$ and $N = n_s = 3$, the configuration identified in the previous section thanks to the gain curves and the projection error curves. Projection errors as defined in Equation (34) are computed for the 500 test examples for each strategy, which enables estimating their probability density functions using Gaussian kernel density estimation (see section 6.6.1. of [43]). The violin plots of the projection errors are given in Figure 9, and the values of the quartiles and expectations are given in Table 1. The third ROM dictionary using the ROM-oriented dissimilarity clearly outperforms the other strategies. Although using a physics-informed clustering procedure, ROM dictionary 2 fails to improve the performances of global ROMs on this specific example. This result illustrates the fact that the Euclidean distance is not always appropriate for model order reduction purposes. ROM dictionary 1 gives the worst results, showing that integrating physics in cluster analyses is crucial when the final objective is to build local approximation spaces. Interestingly, these results also show that using local ROBs can deteriorate the performances of a global ROB when choosing an improper dissimilarity measure for clustering. In this example, the three global ROMs give approximately the same projection errors. These errors are lower than those obtained with ROM dictionary 1 and ROM dictionary 2 because the global ROMs have more relevant snapshots, since they use KN well-distributed snapshots instead of N badly-distributed snapshots. Hence, the dissimilarities $\delta_{\mathcal{X}}$ and $\delta_{\mathcal{H}}$ both define inefficient notions of locality in this example.

Table 1: Quartiles and means of the projection errors for different model order reduction strategies, with $(K, N, n_s) = (6, 3, 3)$.

Strategy	Dissimilarity	Q_1	Median	Q_3	Mean
Global ROM 1	$\delta_{\mathcal{X}}$	0.3863	0.5735	0.7477	0.5636
Global ROM 2	$\delta_{\mathcal{H}}$	0.3611	0.5427	0.7208	0.5397
Global ROM 3	δ	0.3460	0.5666	0.7346	0.5480
ROM dictionary 1	$\delta_{\mathcal{X}}$	0.5434	0.7412	0.9379	0.7071
ROM dictionary 2	$\delta_{\mathcal{H}}$	0.2874	0.6280	0.8038	0.5586
ROM dictionary 3	δ	0.1482	0.2369	0.4584	0.3132

Remark 6.3. *Figure 9 gives projection errors obtained when choosing the correct cluster and thus the most suitable local ROB. The ROM-net's classification errors would have the effect of moving the distribution of ROM dictionary 3 towards larger errors, reducing the gap between the errors made by the different model order reduction strategies. Therefore, particular attention must be paid to the training of the ROM-net's classifier.*

Figure 10 plots the projection error against the dissimilarity measure $\delta_{\mathcal{X}}$ (left), $\delta_{\mathcal{H}}$ (middle) and δ (right) separating a test example from its closest snapshot. One can clearly see the correlation between the projection error and our ROM-oriented dissimilarity δ, contrasting with the absence of correlations between the projection error and the other dissimilarities. This figure also shows that the dissimilarity with the closest snapshot is generally larger than the

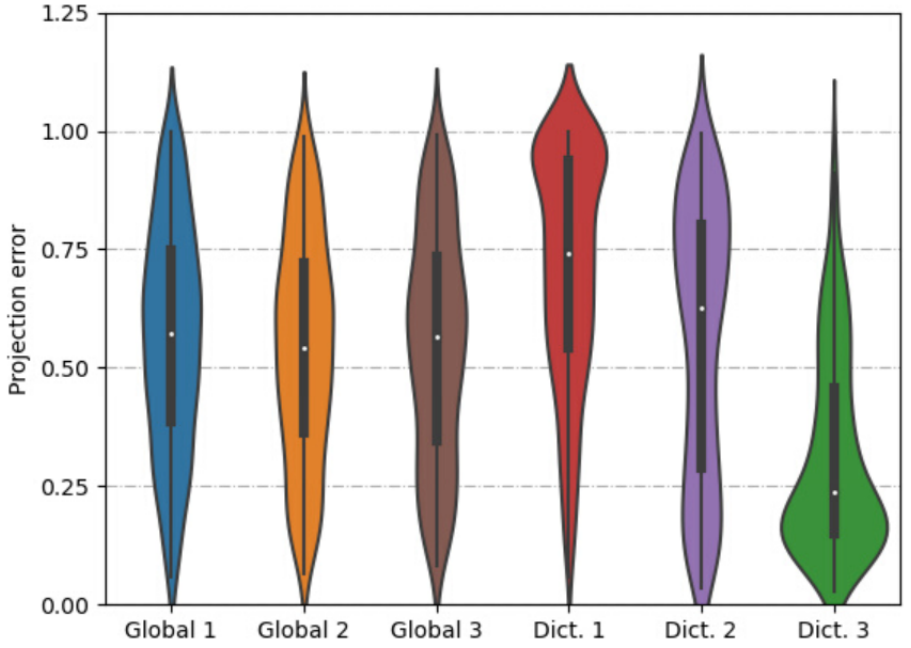

Fig. 9: Violin plots of the projection errors for different model order reduction strategies, with $(K, N, n_s) = (6, 3, 3)$.

projection error onto the POD basis. Indeed, for this time-independent problem, the sine dissimilarity corresponds to the relative projection error; the closest snapshot for this dissimilarity is therefore a better approximation of the solution than the orthogonal projection onto the POD basis, because the POD basis does not perfectly approximate each of its snapshots.

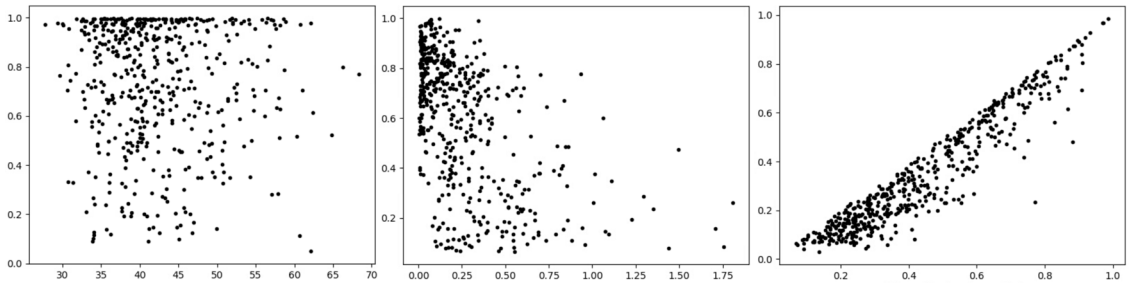

Fig. 10: Scatter plots giving the projection error (y-axis) for test data against the dissimilarity with the closest snapshot (x-axis). From the left to the right: Euclidean distances in the parameter space, Euclidean distances in the solution space, and ROM-oriented dissimilarity.

In the following Sections 6.2-6.5, we compare the performances of dictionaries of local ROMs obtained by clustering using the L^2 and ROM-oriented dissimlarity measures for various physics problems.

6.2. 2D advection equation

We consider the following advection equation:

$$\begin{cases} \dfrac{\partial u}{\partial t} + c\dfrac{\partial u}{\partial \xi_1} = 0 \\ u(\xi_1, \xi_2, t = 0) = U^0 \exp\left(-\dfrac{\xi_1^2 + \left(\xi_2 - \xi_2^0\right)}{l^2}\right), \end{cases} \tag{57}$$

where $(\xi_1, \xi_2) \in [0, 1]^2$ and $t \in [0, 100]$. The analytical solution is known: $u(\xi_1, \xi_2, t) = U^0 \exp\left(-\frac{(\xi_1 - ct)^2 + \left(\xi_2 - \xi_2^0\right)}{l^2}\right)$. The quantities $l = 0.1$ and $c = 1$ are constants, while $U^0 \in \{0.1, 1\}$ and $\xi_2^0 \in \{0, 0.5, 1\}$ are the parameters of the problem. Some snapshots are illustrated on a mesh with 10201 vertices in Figure 11, for various values of U^0, ξ_2^0 and t. A total of 600 snapshots are generated.

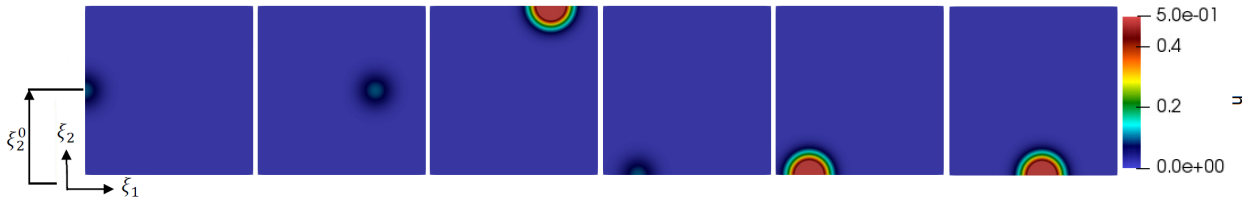

Fig. 11: Snapshots of the solution u of Equation (57) for various values of U^0, ξ_2^0 and t.

Figure 12 shows the number of POD modes for each local basis with respect to the number of clusters, for various accuracy criterions of the POD truncature. For all the considered levels of truncature, the clustering carried out using the sine dissimilarity measure leads to the smallest maximal size of local reduced-order basis.

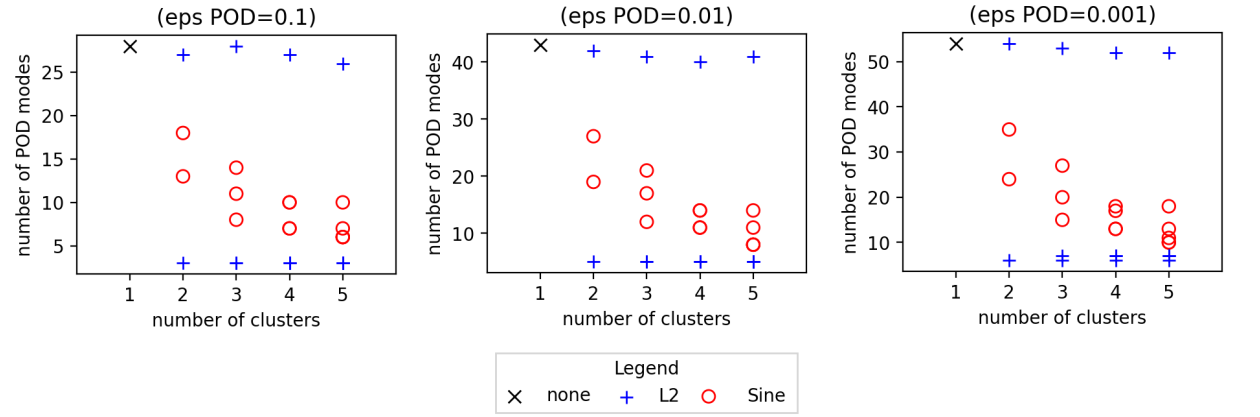

Fig. 12: Number of POD modes for each local basis with respect to the number of clusters, for various accuracy criterions of the POD truncature, applied to the advection problem.

Figure 13 shows the projection errors with respect to the number of clusters, for different size of the local POD basis. For all the considered size of local reduced-order basis, the clustering carried out using the sine dissimilarity measure leads to the smallest projection errors.

Figure 14 shows MultiDimensional Scaling (MDS) representations of L^2 and sine dissimilary measures, with coloring depending on cluster affectation for 5 clusters. A 2-dimensional MDS representation aims to locate points, each representing a solution of the considered physical problem, in such a fashion that the 2D Euclidean distances between each pair of points is as close as possible to the corresponding dissimilarities. In the L^2 dissimilarity case, all the snapshots corresponding to $U^0 = 0.1$ in the MDS representation are very close to each other, which means that their corresponding L^2 pairwise distances are small compared to the rest of the pairwise distances. It is explained by the fact that the L^2 norm quantifies magnitudes. Hence, when applying a k-medoid clustering algorithm, all the snapshots corresponding to $U^0 = 0.1$ are affected to the same cluster. However, these small-magnitude snapshots contain all the independant directions of the solution function space described by the whole snapshot set. As a consequence, the local reduced-order model corresponding to the cluster containing these small-magnitude snapshots has the same reducibility as the complete set, for any accuracy level and even when increasing the number of clusters. This is illustrated in Figure 12, where the L^2 dissimilary case exhibits one local reduced-order model having a number of mode very close to the global reduced-order model. On the contrary, the MDS for the sine dissimilarity in Figure 14 shows three trajectories corresponding to the three different values of ξ_2^0. Actually, each pair of snapshots corresponding of

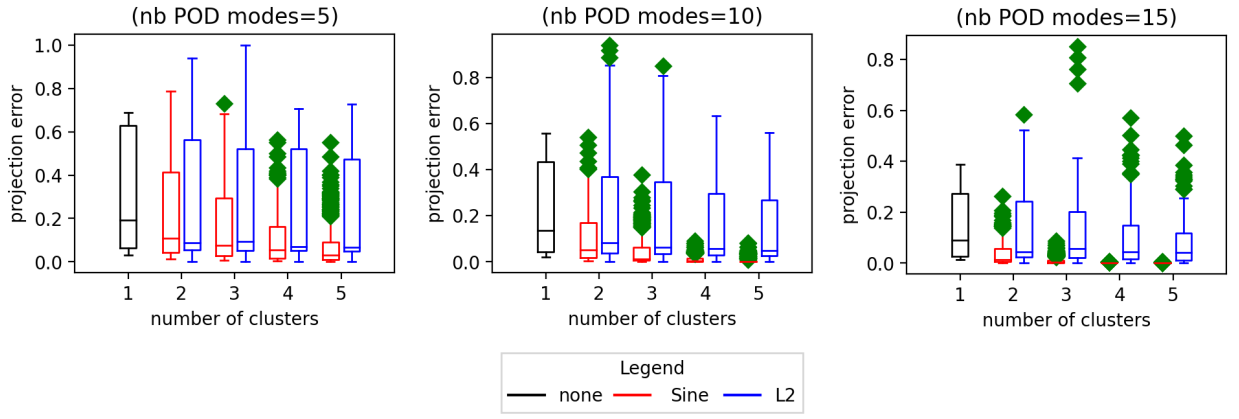

Fig. 13: Box plots for the projection errors with respect to the number of clusters, for different size of the local POD basis, applied to the advection problem. The box plots show the median, first and third quartiles, as well as extremal values (in the form of green dots if outliers).

same values of ξ_2^0 and t, for $U^0 = 0.1$ and 1, are at the same location of the MDS representation, which means that their pairwise sine dissimilarities are zero. Hence, a new snapshot collinear to an existing snapshot do not increase the number of independant directions in the snapshot set. The corresponding clustering produces balanced clusters, having local small-sized reduced-order basis as seen in Figure 12.

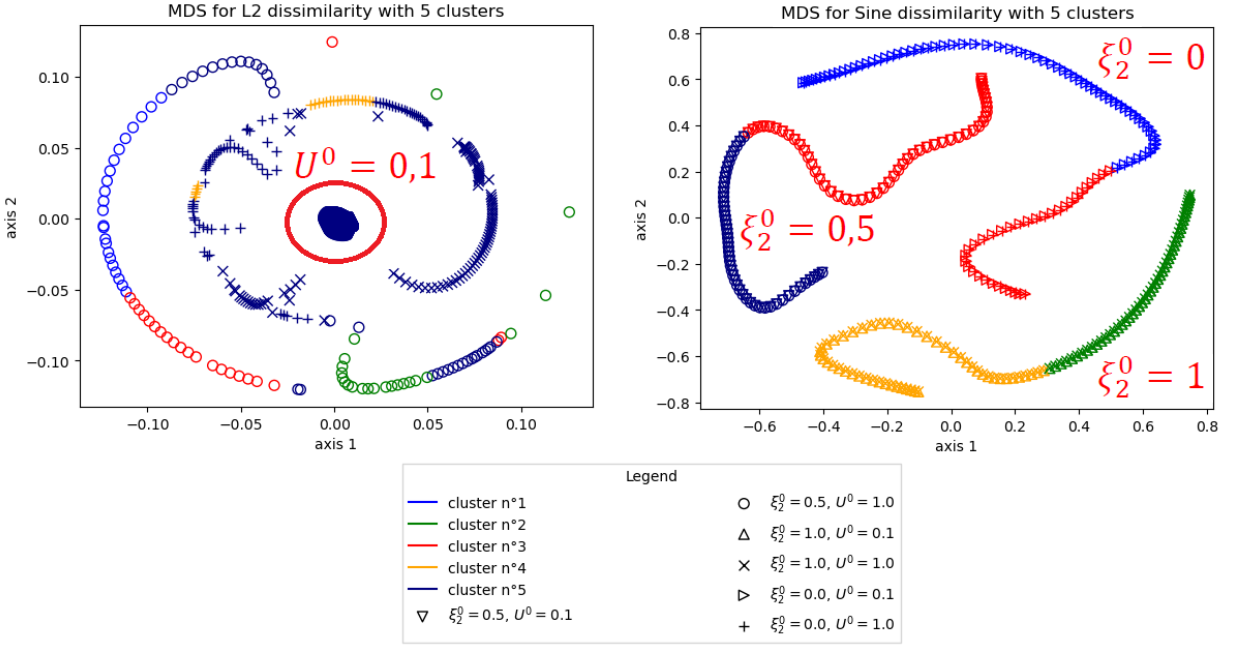

Fig. 14: MDS representations of L^2 and sine dissimilary measures, with coloring depending on cluster affectation for 5 clusters, applied to the advection problem.

Since the same analysis can be done in the following three additional numerical experiments, we do not repeat it and simply explain the new physical settings.

6.3. 2D incompressible Navier-Stokes

We consider the 2D incompressible Navier-Stokes equations in the setting illustrated in Figure 15: the air flows from the left to the right in the rectangular domain, with a uniform Dirichlet boundary condition for the velocity U on Γ_{in}, outflow boundary condition on Γ_{in}, and no-slip boundary condition on the walls Γ_{wall} and on the circular object

Γ_{object}. The mesh is constituted of 19818 vertices; the low-Mach number solver YALES2 [98] for unstructured grids is used. The parameters of the problem are components U_x^0 and U_y^0 of the uniform incoming velocity boundary condition, and we consider 6 temporal simulations for $(U_x^0, U_y^0) \in \{(0.025, 0.025), (0.075, 0.075), (0.25, 0.25), (0.025, -0.025), (0.075, -0.075), (0.25, -0.25)\}$ leading to a total of 600 snapshots. The last time step of each of these simulations is illustrated in Figure 16.

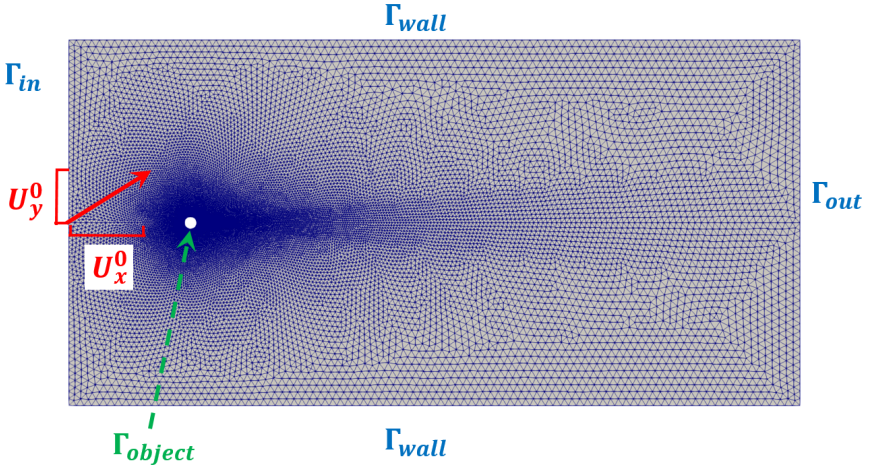

Fig. 15: Setting and mesh for the Navier-Stokes problem.

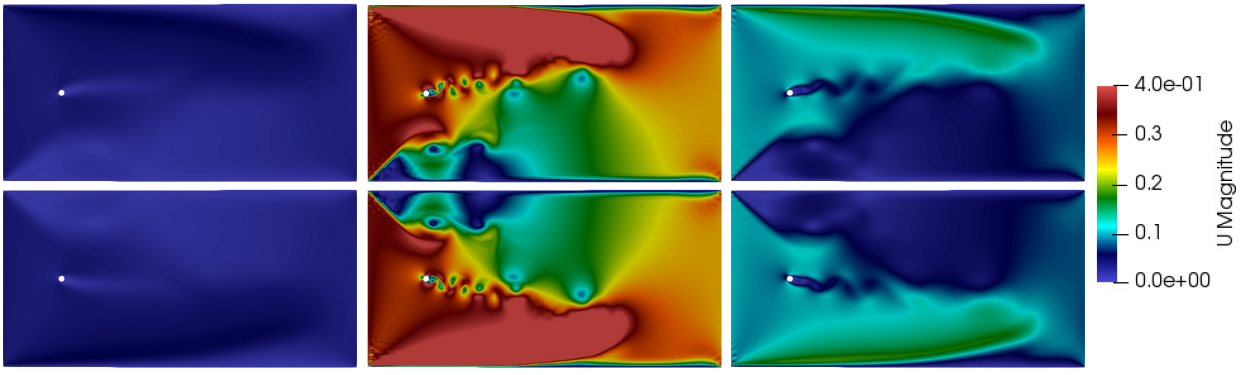

Fig. 16: Snapshots of the solution velocity field U of the Navier-Stokes problem: last time step for $\left(U_x^0, U_y^0\right) \in \{(0.025, 0.025), (0.075, 0.075), (0.25, 0.25), (0.025, -0.025), (0.075, -0.075), (0.25, -0.25)\}$.

The improved performance of the sine dissimilarity based clustering, with respect to the L^2 one, is illustrated in Figures 17-18.

The MDS representations in Figure 19 shows the advantages of the sine dissimilarity in the same fashion as the previous section: with L^2-based clustering, snapshots are grouped by magnitude, where all the small-magnitude ones are tightly packed in a single cluster, whereas with sine-based clustering, snapshots are grouped by direction of the initial condition.

6.4. 2D heat equation

We consider the 2D transient linear heat equation on a square domain, with a localized volumetric heat source term. The parameters of the problem are the location of the source term and its magnitude; only the final time step is kept, see Figure 20 for examples of snapshots. The mesh is composed of 2601 vertices and 500 snapshots for 500 random values of the parameters are computed using the finite-element software *Z-set* [99].

The improved performance of the sine dissimilarity based clustering, with respect to the L^2 one, is illustrated in Figures 21-22.

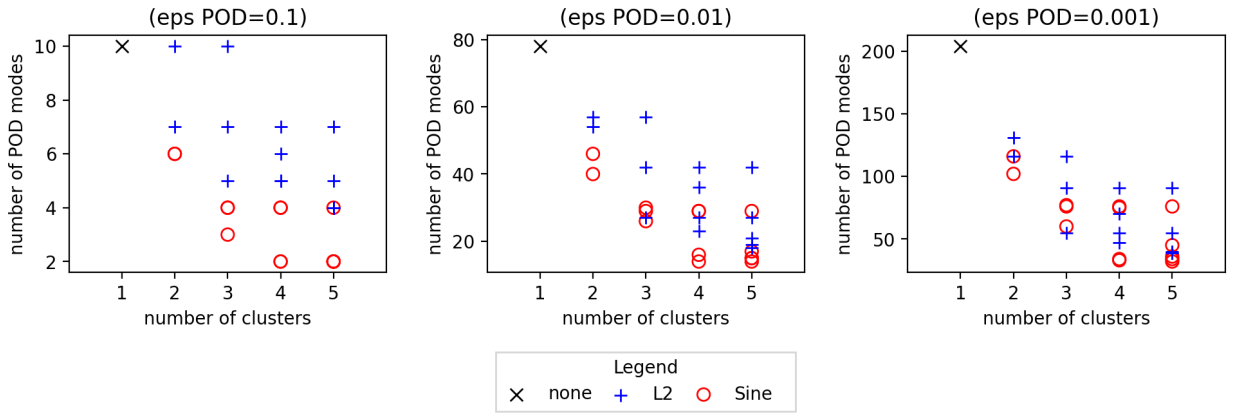

Fig. 17: Number of POD modes for each local basis with respect to the number of clusters, for various accuracy criterions of the POD truncature, applied to the Navier-Stokes problem.

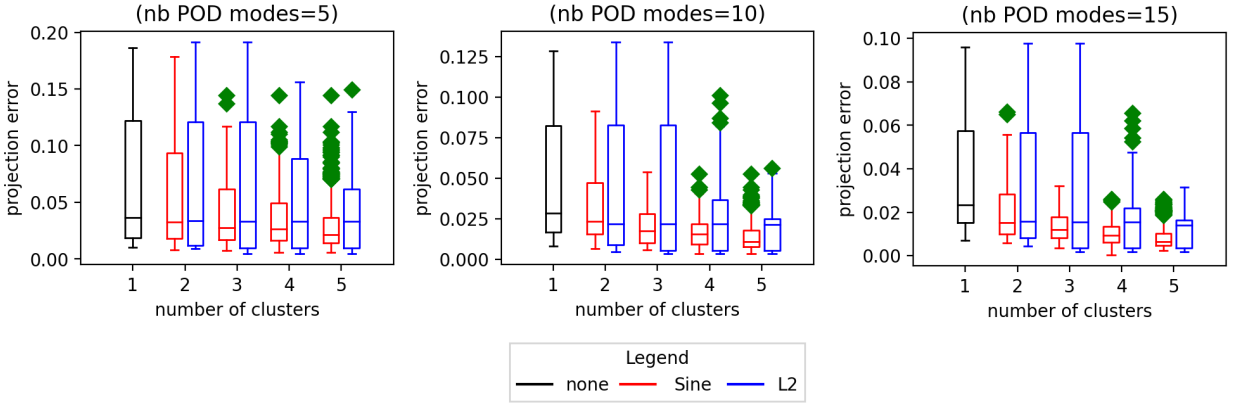

Fig. 18: Box plots for the projection errors with respect to the number of clusters, for different size of the local POD basis, applied to the Navier-Stokes problem. The box plots show the median, first and third quartiles, as well as extremal values (in the form of green dots if outliers).

6.5. 3D nonlinear structural mechanics

We consider a 3D quasistatic nonlinear structural mechanics problem: an object is rotated along an axis intersecting the center of gravity of this object. The material is modeled by a viscoplastic constitutive law with a Von Mises criterion and a Norton flow. The orientation of the axis is the parameter of the problem; only the final time step in kept and the quantity of interest is the accumulated plastic strain, see Figure 23 for examples of snapshots. The mesh is composed of 61741 vertices and 100 snapshots for 100 random values of the parameters are computed using the finite-element software *Z-set* [99].

The improved performance of the sine dissimilarity based clustering, with respect to the L^2 one, is illustrated in Figure 25.

7. Conclusion

Dictionaries of local ROBs are commonly used for nonlinear model order reduction. A natural way of building such dictionaries is to partition the parameter space or the solution space with a clustering algorithm, and then define one local basis per cluster. This article shows that the choice of the dissimilarity measure for clustering is crucial, as it highly affects the performances of the local ROBs. In particular, it is shown that using Euclidean distances in the parameter space or in the solution space can lead to local bases whose performances are worse than those of a global ROB with the same number of modes. To remedy this problem, a ROM-oriented dissimilarity measure involving the principal angles between subspaces spanned by simulation results is introduced. It enables focusing on

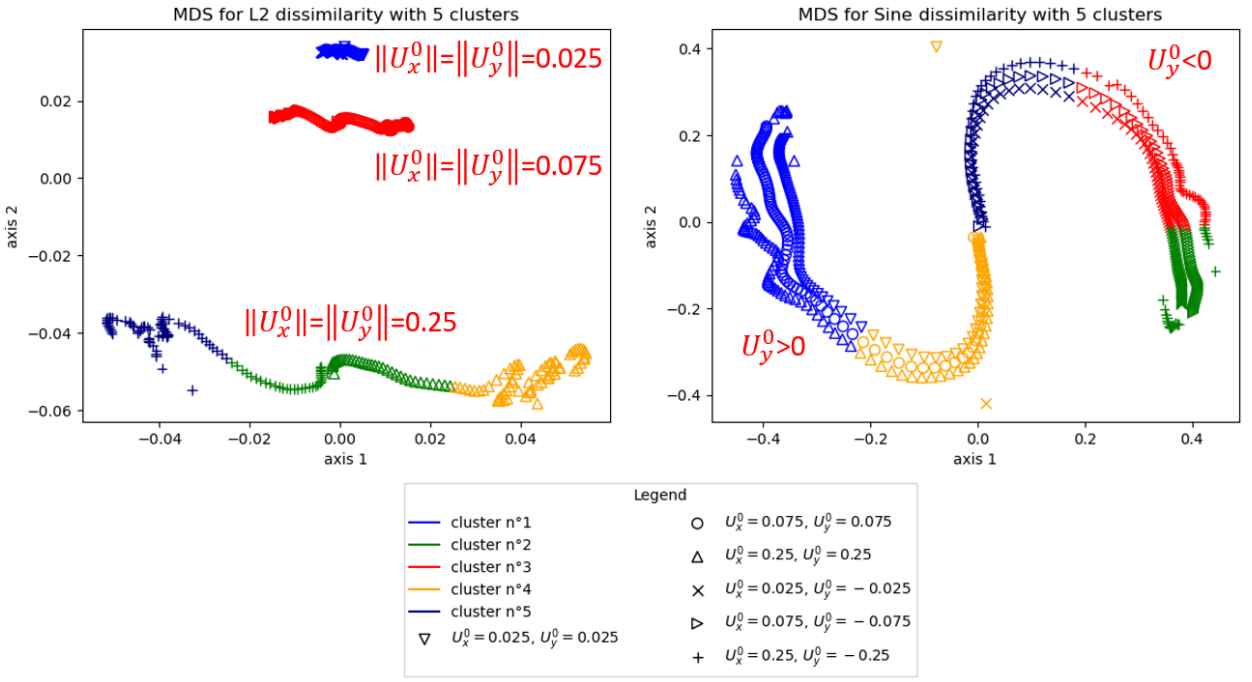

Fig. 19: MDS representations of L^2 and sine dissimilary measures, with coloring depending on cluster affectation for 5 clusters, applied to the Navier-Stokes problem.

Fig. 20: Snapshots of the solution temperature field T of the 2D heat problem for various values of the parameters.

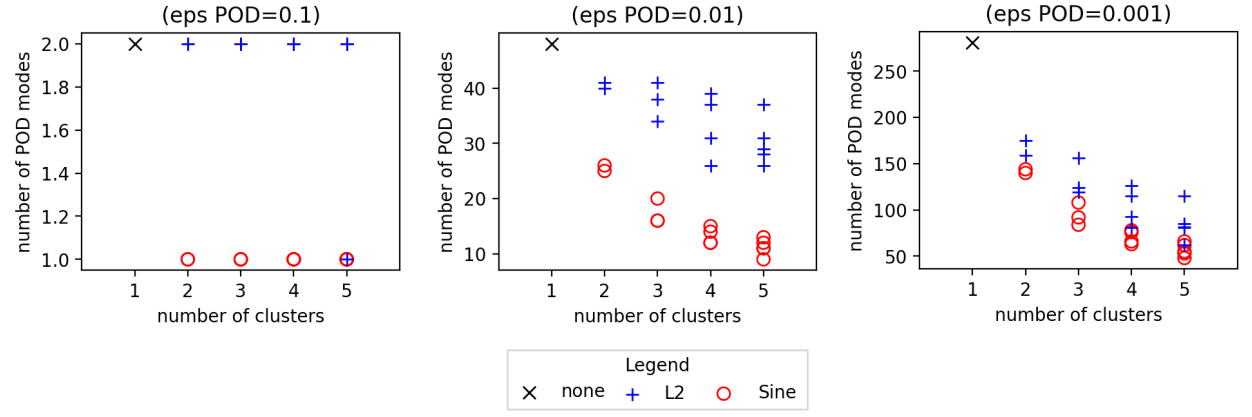

Fig. 21: Number of POD modes for each local basis with respect to the number of clusters, for various accuracy criterions of the POD truncature, applied to the 2D heat problem.

the shape of simulation results rather than their magnitudes. The strength of this dissimilarity comes from its link with the projection error appearing in the definition of the Kolmogorov N-width. The resulting dictionary of ROBs can be integrated in a ROM-net, where a classifier is used in the exploitation phase for fast and automatic model

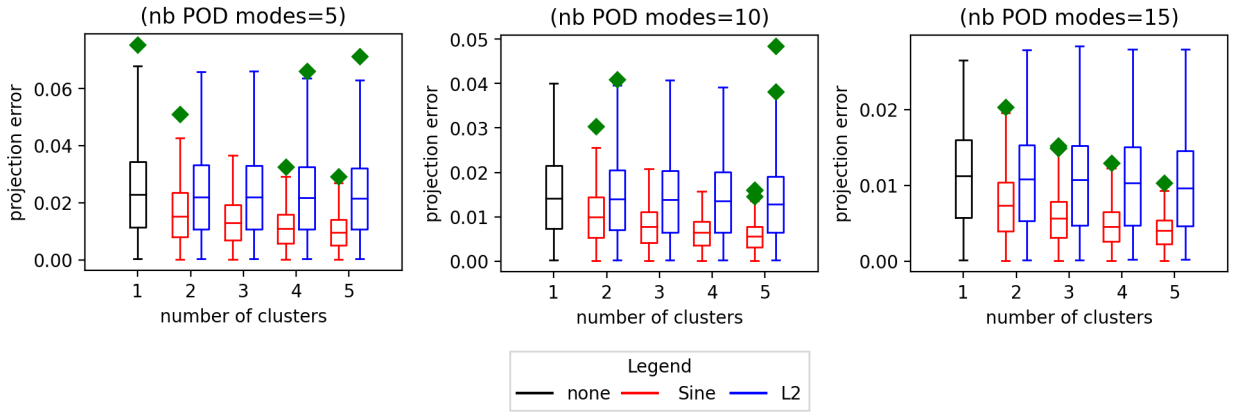

Fig. 22: Box plots for the projection errors with respect to the number of clusters, for different size of the local POD basis, applied to the 2D heat problem. The box plots show the median, first and third quartiles, as well as extremal values (in the form of green dots if outliers).

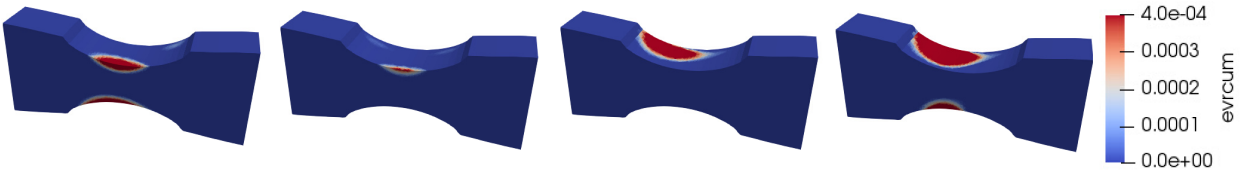

Fig. 23: Snapshots of the accumulated plasticity field *evrcum* of the mechanical problem for various values of the parameters.

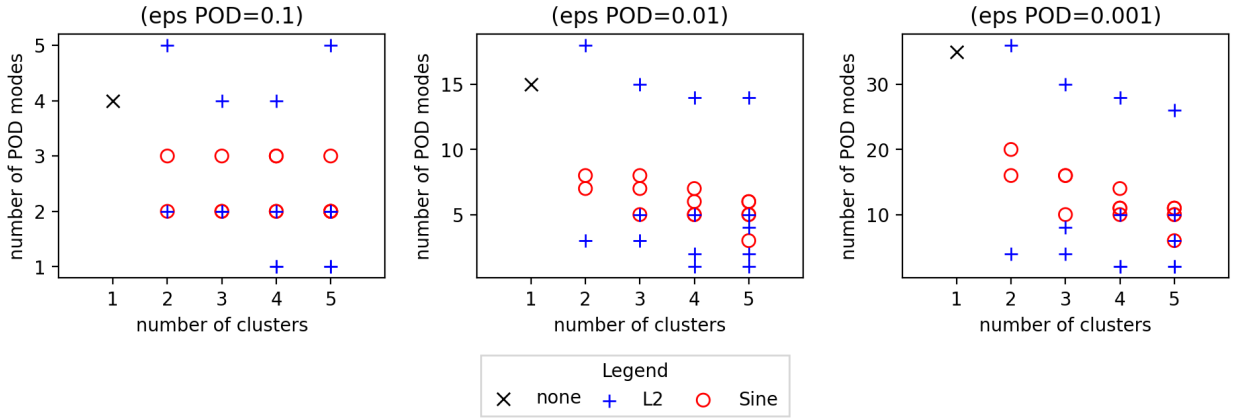

Fig. 24: Number of POD modes for each local basis with respect to the number of clusters, for various accuracy criterions of the POD truncature, applied to the mechanical problem.

recommendation. The present paper gives an a priori efficiency criterion enabling hyperparameters calibration before time-consuming steps of the ROM-net's training phase. Future work will consider the application of this methodology to complex three-dimensional problems and their simulations with local ROMs.

Abbreviations

DEIM: Discrete Empirical Interpolation Method; LDEIM: Localized Discrete Empirical Interpolation Method; PCA: Principal Component Analysis; PDE: Partial Differential Equation; POD: Proper Orthogonal Decomposition; ROB: Reduced-Order Basis; ROM: Reduced-Order Model; SVD: Singular Value Decomposition.

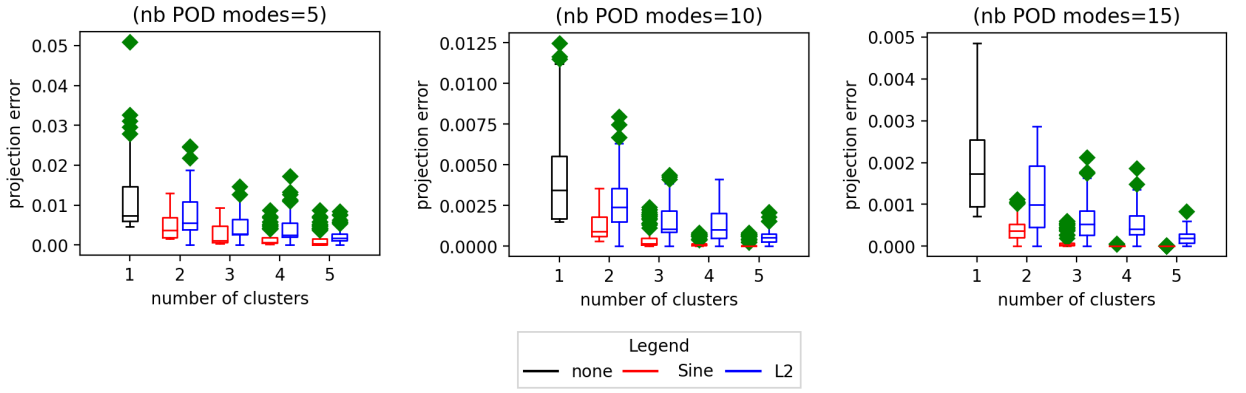

Fig. 25: Box plots for the projection errors with respect to the number of clusters, for different size of the local POD basis, applied to themechanical problem. The box plots show the median, first and third quartiles, as well as extremal values (in the form of green dots if outliers).

Funding

Study funded by Safran and ANRT (Association Nationale de la Recherche et de la Technologie, France).

Appendix A: Properties of the normalized Kolmogorov width

Property 7.1 (Inequalities on Kolmogorov widths)**.** *If $\mathcal{M}$ is bounded and contains at least one nonzero element, then:*

$$\forall N \in \mathbb{N}^*, \quad d_N(\mathcal{M}) \leq \sup_{v\in\mathcal{M}} \|v\|_{\mathcal{H}}\, \tilde{d}_N(\mathcal{M}). \tag{58}$$

Proof. The boundedness of $\mathcal{M}$ implies the existence of $\sup_{v\in\mathcal{M}} \|v\|_{\mathcal{H}}$, thus for all $u \in \mathcal{M} \setminus \{0\}$:

$$\|u - \pi_{\mathcal{H}_N}(u)\|_{\mathcal{H}} \leq \sup_{v\in\mathcal{M}} \|v\|_{\mathcal{H}} \frac{\|u - \pi_{\mathcal{H}_N}(u)\|_{\mathcal{H}}}{\|u\|_{\mathcal{H}}}, \tag{59}$$

which implies Equation (58). □

Proof of Property 4.2. Given that $\measuredangle_{\mathcal{H}}(u, \mathcal{H}_N) \in [0; \pi/2]$, the sine of the angle $\measuredangle_{\mathcal{H}}(u, \mathcal{H}_N)$ satisfies:

$$\begin{aligned} \sin \measuredangle_{\mathcal{H}}(u, \mathcal{H}_N) &= |\sin \measuredangle_{\mathcal{H}}(u, \mathcal{H}_N)| \\ &= \sqrt{1 - \cos^2 \measuredangle_{\mathcal{H}}(u, \mathcal{H}_N)} \\ &= \sqrt{1 - \cos^2 \inf_{v\in\mathcal{H}_N} \arccos\left(\frac{|\langle u, v\rangle_{\mathcal{H}}|}{\|u\|_{\mathcal{H}}\|v\|_{\mathcal{H}}}\right)}, \end{aligned} \tag{60}$$

and, since the function $\alpha \to \sqrt{1 - \cos^2 \alpha}$ is increasing on the interval $[0; \pi/2]$:

$$\sin \measuredangle_{\mathcal{H}}(u, \mathcal{H}_N) = \inf_{v\in\mathcal{H}_N} \sqrt{1 - \cos^2 \arccos\left(\frac{|\langle u, v\rangle_{\mathcal{H}}|}{\|u\|_{\mathcal{H}}\|v\|_{\mathcal{H}}}\right)} = \inf_{v\in\mathcal{H}_N} \sqrt{1 - \frac{\langle u, v\rangle^2_{\mathcal{H}}}{\|u\|^2_{\mathcal{H}}\|v\|^2_{\mathcal{H}}}}. \tag{61}$$

Furthermore:

$$\frac{\|u - \pi_{\mathrm{span}(\{v\})}(u)\|^2_{\mathcal{H}}}{\|u\|^2_{\mathcal{H}}} = \frac{\|u - \langle u, \frac{v}{\|v\|_{\mathcal{H}}}\rangle_{\mathcal{H}} \frac{v}{\|v\|_{\mathcal{H}}}\|^2_{\mathcal{H}}}{\|u\|^2_{\mathcal{H}}} = 1 - \frac{\langle u, v\rangle^2_{\mathcal{H}}}{\|u\|^2_{\mathcal{H}}\|v\|^2_{\mathcal{H}}}, \tag{62}$$

where $\pi_{\mathrm{span}(\{v\})}(u)$ is the orthogonal projection of u on $\mathrm{span}(\{v\})$. Using Equation (62) in Equation (61) yields:

$$\sin \measuredangle_{\mathcal{H}}(u, \mathcal{H}_N) = \inf_{v\in\mathcal{H}_N} \frac{\|u - \pi_{\mathrm{span}(\{v\})}(u)\|_{\mathcal{H}}}{\|u\|_{\mathcal{H}}} = \inf_{v\in\mathcal{H}_N} \inf_{w\in\mathrm{span}(\{v\})} \frac{\|u - w\|_{\mathcal{H}}}{\|u\|_{\mathcal{H}}}. \tag{63}$$

Finally, given that $\underset{v\in\mathcal{H}_N}{\cup}\text{span}(\{v\}) = \mathcal{H}_N$:

$$\sin\measuredangle_{\mathcal{H}}(u,\mathcal{H}_N) = \inf_{v\in\mathcal{H}_N}\frac{\|u-v\|_{\mathcal{H}}}{\|u\|_{\mathcal{H}}}, \tag{64}$$

which ends the proof. □

Appendix B: Properties of the dissimilarity measure

Proof of Property 4.8. Let us first develop the square of the right-hand side of Equation (31), denoted by $f_n(u,v)^2$, using Equation (30), the bilinearity of the L^2 inner product and the orthonormality of the bases $\Psi_n(u)$ and $\Psi_n(v)$:

$$\begin{aligned} f_n(u,v)^2 &= \sum_{i=1}^{n}\|\psi_i(u)\|^2_{L^2(\Omega)} - 2\sum_{i=1}^{n}\sum_{j=1}^{n}\langle\psi_i(u),\psi_j(v)\rangle^2_{L^2(\Omega)} \\ &\quad + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\langle\psi_i(u),\psi_j(v)\rangle_{L^2(\Omega)}\langle\psi_i(u),\psi_k(v)\rangle_{L^2(\Omega)}\langle\psi_j(v),\psi_k(v)\rangle_{L^2(\Omega)} \\ &= n - 2\sum_{i=1}^{n}\sum_{j=1}^{n}\langle\psi_i(u),\psi_j(v)\rangle^2_{L^2(\Omega)} \\ &\quad + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\langle\psi_i(u),\psi_j(v)\rangle_{L^2(\Omega)}\langle\psi_i(u),\psi_k(v)\rangle_{L^2(\Omega)}\delta_{jk} \\ &= n - \sum_{i=1}^{n}\sum_{j=1}^{n}\langle\psi_i(u),\psi_j(v)\rangle^2_{L^2(\Omega)}, \end{aligned} \tag{65}$$

where δ_{jk} is the Kronecker delta function. Let $\mathbf{C}\in\mathbb{R}^{n\times n}$ be the matrix whose entries are the inner products $\langle\psi_i(u),\psi_j(v)\rangle_{L^2(\Omega)}$. Its SVD reads $\mathbf{C} = \mathbf{V}\cos\boldsymbol{\Theta}\,\mathbf{W}^T$ where $\boldsymbol{\Theta}$ is a diagonal matrix containing the principal angles $\theta_k(\mathcal{V}_n(u),\mathcal{V}_n(v))$, and where $\mathbf{V}$ and $\mathbf{W}$ are orthogonal matrices. Then, we obtain:

$$\begin{aligned} f_n(u,v)^2 &= n - \text{tr}(\mathbf{C}^T\mathbf{C}) \\ &= n - \text{tr}\left(\mathbf{W}\cos(\boldsymbol{\Theta})^T\,\mathbf{V}^T\mathbf{V}\cos(\boldsymbol{\Theta})\,\mathbf{W}^T\right) \\ &= n - \text{tr}\left(\mathbf{W}^T\mathbf{W}\cos(\boldsymbol{\Theta})^T\,\mathbf{V}^T\mathbf{V}\cos(\boldsymbol{\Theta})\right) \\ &= n - \text{tr}\left(\cos(\boldsymbol{\Theta})^T\cos(\boldsymbol{\Theta})\right) \\ &= n - \sum_{k=1}^{n}\cos^2\theta_k(\mathcal{V}_n(u),\mathcal{V}_n(v)) \\ &= \sum_{k=1}^{n}1-\cos^2\theta_k(\mathcal{V}_n(u),\mathcal{V}_n(v)) \\ &= \sum_{k=1}^{n}\sin^2\theta_k(\mathcal{V}_n(u),\mathcal{V}_n(v)) \\ &= \tilde{\delta}_n(u,v)^2. \end{aligned} \tag{66}$$

These equations remain true when exchanging u and v, which ends the proof. □

Proof of Property 4.9. Since the Hilbert-Schmidt inner product on $HS(L^2(\Omega))$ does not depend on the choice of the orthonormal basis of $L^2(\Omega)$, let us choose a basis that is relevant for calculations. For $u\in L^2(\Omega\times[0;t_f])$, the n-dimensional elementary basis $\Psi_n(u)$ is completed with an orthonormal basis of the orthogonal complement of $\mathcal{V}_n(u)$ in $L^2(\Omega)$. The resulting orthonormal basis of $L^2(\Omega)$ is denoted by $\{\psi_k(u)\}_{k\in\mathbb{N}^*}$, where the n first basis vectors are those of the basis $\Psi_n(u)$. Let us now expand the term $\|\pi_{\mathcal{V}_n(u)} - \pi_{\mathcal{V}_n(v)}\|^2_{HS(L^2(\Omega))}$:

$$\|\pi_{\mathcal{V}_n(u)} - \pi_{\mathcal{V}_n(v)}\|^2_{HS(L^2(\Omega))} = \|\pi_{\mathcal{V}_n(u)}\|^2_{HS(L^2(\Omega))} + \|\pi_{\mathcal{V}_n(v)}\|^2_{HS(L^2(\Omega))} - 2\langle\pi_{\mathcal{V}_n(u)},\pi_{\mathcal{V}_n(v)}\rangle_{HS(L^2(\Omega))}. \tag{67}$$

Using the definition of the Hilbert-Schmidt inner product given by Equation (5), one has:

$$\begin{aligned}\langle \pi_{\mathcal{V}_n(u)}, \pi_{\mathcal{V}_n(v)} \rangle_{HS(L^2(\Omega))} &= \sum_{i=1}^{\infty} \langle \pi_{\mathcal{V}_n(u)}(\psi_i(u)), \pi_{\mathcal{V}_n(v)}(\psi_i(u)) \rangle_{L^2(\Omega)} \\ &= \sum_{i=1}^{n} \langle \psi_i(u), \pi_{\mathcal{V}_n(v)}(\psi_i(u)) \rangle_{L^2(\Omega)} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{n} \langle \psi_i(u), \psi_j(v) \rangle^2_{L^2(\Omega)},\end{aligned} \tag{68}$$

where the last equality results from the expression of $\pi_{\mathcal{V}_n(v)}(\psi_i(u))$ given by Equation (30). Furthermore:

$$\begin{aligned}\|\pi_{\mathcal{V}_n(u)}\|^2_{HS(L^2(\Omega))} &= \sum_{i=1}^{\infty} \langle \pi_{\mathcal{V}_n(u)}(\psi_i(u)), \pi_{\mathcal{V}_n(u)}(\psi_i(u)) \rangle_{L^2(\Omega)} \\ &= \sum_{i=1}^{n} \langle \psi_i(u), \psi_i(u) \rangle_{L^2(\Omega)} \\ &= n.\end{aligned} \tag{69}$$

Similarly, one can prove that $\|\pi_{\mathcal{V}_n(v)}\|^2_{HS(L^2(\Omega))} = n$. Finally:

$$\|\pi_{\mathcal{V}_n(u)} - \pi_{\mathcal{V}_n(v)}\|^2_{HS(L^2(\Omega))} = 2n - 2\sum_{i=1}^{n}\sum_{j=1}^{n} \langle \psi_i(u), \psi_j(v) \rangle^2_{L^2(\Omega)} = 2f_n(u,v)^2 = 2\tilde{\delta}_n(u,v)^2, \tag{70}$$

where $f_n(u,v)$ was introduced in the proof of Property 4.8. □

Property 7.2. *For all $n \in [\![1; \mathcal{N}]\!]$, the sine dissimilarity is a pseudometric on $L^2(\Omega \times [0; t_f])$.*

Proof. The sine dissimilarity $\tilde{\delta}_n$ is nonnegative and symmetric. Equation (32) implies that for all $u \in L^2(\Omega \times [0; t_f])$, $\tilde{\delta}_n(u,u) = 0$. Equation (32) also yields:

$$\tilde{\delta}_n(u,v) = \frac{1}{\sqrt{2}} \|\pi_{\mathcal{V}_n(u)} - \pi_{\mathcal{V}_n(v)}\|_{HS(L^2(\Omega))} = \frac{1}{\sqrt{2}} \|\pi_{\mathcal{V}_n(u)} - \pi_{\mathcal{V}_n(w)} + \pi_{\mathcal{V}_n(w)} - \pi_{\mathcal{V}_n(v)}\|_{HS(L^2(\Omega))}, \tag{71}$$

so the triangle inequality on the Hilbert-Schmidt norm gives the triangle inequality $\tilde{\delta}_n(u,v) \leq \tilde{\delta}_n(u,w) + \tilde{\delta}_n(w,v)$. □

Note that $\tilde{\delta}_n(u,v) = 0$ does not imply that $u = v$, which is the reason why the sine dissimilarity is not a metric on $L^2(\Omega \times [0; t_f])$. This is not a problem since we want this dissimilarity measure to be zero for all pairs of functions $(u,v) \in L^2(\Omega \times [0; t_f])^2$ whose trajectories over time in $L^2(\Omega)$ give the same POD approximation space. The next property shows the link between the sine dissimilarity and the Grassmann dissimilarity used in [48] for dictionary-based ROM-nets:

Property 7.3 (Equivalence with the Grassmann dissimilarity for small angles)**.** *Given $n \in [\![1; \mathcal{N}]\!]$, let $\boldsymbol{\theta}_n$ denote the vector of principal angles between $\mathcal{V}_n(u)$ and $\mathcal{V}_n(v)$ for two square-integrable functions u and v. As $\|\boldsymbol{\theta}_n\|_2$ tends towards zero, the sine dissimilarity is asymptotically equivalent to the Grassmann dissimilarity $\|\boldsymbol{\theta}_n\|_2$, that is, using Bachmann-Landau notations:*

$$\tilde{\delta}_n(u,v) \underset{\|\boldsymbol{\theta}_n\|_2 \to 0}{\sim} \|\boldsymbol{\theta}_n\|_2. \tag{72}$$

Proof. One must show that:

$$\|\sin \boldsymbol{\theta}_n\|_2 = \|\boldsymbol{\theta}_n\|_2 + o(\|\boldsymbol{\theta}_n\|_2). \tag{73}$$

As $\|\boldsymbol{\theta}_n\|_2$ tends towards zero:

$$\|\sin \boldsymbol{\theta}_n\|_2 = \left(\sum_{i=1}^{n} \sin^2 \theta_{n,k} \right)^{1/2} = \left(\sum_{i=1}^{n} (\theta_{n,k} + o(\theta^2_{n,k}))^2 \right)^{1/2} = \left(\|\boldsymbol{\theta}_n\|_2^2 + o(\|\boldsymbol{\theta}_n\|_2^2) \right)^{1/2}, \tag{74}$$

which gives:

$$\|\sin \boldsymbol{\theta}_n\|_2 = \|\boldsymbol{\theta}_n\|_2 \sqrt{1 + o(1)} = \|\boldsymbol{\theta}_n\|_2 + o(\|\boldsymbol{\theta}_n\|_2). \tag{75}$$

□

References

[1] A. Quarteroni, G. Rozza, Reduced Order Methods for Modeling and Computational Reduction, Springer Publishing Company, Incorporated, 2013.

[2] W. Keiper, A. Milde, S. Volkwein, Reduced-Order Modeling (ROM) for Simulation and Optimization: Powerful Algorithms as Key Enablers for Scientific Computing, Springer International Publishing, 2018. URL: https://books.google.fr/books?id=dHRVDwAAQBAJ.

[3] F. Chinesta, P. Ladeveze, E. Cueto, A short review on model order reduction based on Proper Generalized Decomposition, Archives of Computational Methods in Engineering 18 (2011) 395–404.

[4] F. Chinesta, E. Cueto, PGD-Based Modeling of Materials, Structures and Processes, 2014. doi:10.1007/978-3-319-06182-5.

[5] C. Prud'homme, D. Rovas, K. Veroy, L. Machiels, Y. Maday, A. Patera, G. Turinici, Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods, Journal of Fluids Engineering 124 (2002) 70.

[6] G. Rozza, D. Huynh, A. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, Archives of Computational Methods in Engineering 15 (2007) 1–47.

[7] L. Cordier, M. Bergmann, Proper Orthogonal Decomposition: an overview, in: Lecture series 2002-04, 2003-03 and 2008-01 on post-processing of experimental and numerical data, Von Karman Institute for Fluid Dynamics, 2008., VKI, 2008, p. 46 pages. URL: https://hal.archives-ouvertes.fr/hal-00417819.

[8] C. Rowley, T. Colonius, R. Murray, Model reduction for compressible flow using POD and Galerkin projection, Physica D: Nonlinear Phenomena 189 (2003) 115–129.

[9] A. Cohen, R. DeVore, Kolmogorov widths under holomorphic mappings, IMA Journal of Numerical Analysis 36 (2015) 1–12.

[10] C. Greif, K. Urban, Decay of the kolmogorov n-width for wave problems, Applied Mathematics Letters 96 (2019) 216 – 222.

[11] N. Cagniart, Y. Maday, B. Stamm, Model Order Reduction for Problems with Large Convection Effects, 2019, pp. 131–150. doi:10.1007/978-3-319-78325-3_10.

[12] M. Nonino, F. Ballarin, G. Rozza, Y. Maday, Overcoming slowly decaying kolmogorov n-width by transport maps: application to model order reduction of fluid dynamics and fluid–structure interaction problems, 2019.

[13] W. Dahmen, C. Huang, C. Schwab, G. Welper, Adaptive Petrov–Galerkin methods for first order transport equations, SIAM Journal on Numerical Analysis 50 (2012) 2420–2445.

[14] W. Dahmen, C. Plesken, G. Welper, Double greedy algorithms: Reduced basis methods for transport dominated problems, ESAIM: Mathematical Modelling and Numerical Analysis 48 (2014) 623–663.

[15] D. Rim, B. Peherstorfer, K. Mandli, Manifold approximations via transported subspaces: Model reduction for transport-dominated problems, 2020. URL: https://arxiv.org/abs/1912.13024.

[16] T. Taddei, L. Zhang, Space-time registration-based model reduction of parameterized one-dimensional hyperbolic pdes, 2020. URL: https://arxiv.org/abs/2004.06693.

[17] D. Amsallem, M. Zahr, C. Farhat, Nonlinear model order reduction based on local reduced-order bases, International Journal for Numerical Methods in Engineering (2012) 1–31.

[18] K. Washabaugh, D. Amsallem, M. Zahr, C. Farhat, Nonlinear model reduction for CFD problems using local reduced order bases, 42nd AIAA Fluid Dynamics Conference (2012).

[19] J. Lumley, The structure of inhomogeneous turbulent flows, Atm. Turb. and Radio Wave. Prop. (1967) 166–178.

[20] D. Amsallem, C. Farhat, An online method for interpolating linear parametric reduced-order models, SIAM Journal on Scientific Computing 33(5) (2011) 2169–2198.

[21] T. Lieu, M. Lesoinne, Parameter adaptation of reduced order models for three-dimensional flutter analysis, AIAA Paper 2004-0888 (2004).

[22] T. Lieu, C. Farhat, M. Lesoinne, POD-based aeroelastic analysis of a complete F-16 configuration: ROM adaptation and demonstration, AIAA Paper 2005-2295 (2005).

[23] T. Lieu, C. Farhat, Adaptation of POD-based aeroelastic ROMs for varying Mach number and angle of attack: Application to a complete F-16 configuration, AIAA Paper 2005-7666 (2005).

[24] T. Lieu, C. Farhat, M. Lesoinne, Reduced-order fluid/structure modeling of a complete aircraft configuration, Computer Methods in Applied Mechanics and Engineering 195 (2006) 5730–5742.

[25] T. Lieu, C. Farhat, Adaptation of aeroelastic reduced-order models and application to an F-16 configuration, AIAA Journal 45 (2007) 1244–1257.

[26] D. Amsallem, C. Farhat, Interpolation method for adapting reduced-order models and application to aeroelasticity, AIAA Journal 46(7) (2008) 1803–1813.

[27] D. Amsallem, J. Cortial, C. Farhat, Towards real-time computational-fluid-dynamics-based aeroelastic computations using a database of reduced-order information, AIAA Journal 48 (2010) 2029–2037.

[28] D. Amsallem, R. Tezaur, C. Farhat, Real-time solution of linear computational problems using databases of parametric reduced-order models with arbitrary underlying meshes, Journal of Computational Physics 326 (2016) 373 – 397.

[29] R. Mosquera, A. Hamdouni, A. El Hamidi, C. Allery, POD basis interpolation via Inverse Distance Weighting on Grassmann manifolds, Discrete and Continuous Dynamical Systems, Series S. 12(6) (2018) 1743–1759.

[30] R. Mosquera, A. El Hamidi, A. Hamdouni, A. Falaize, Generalization of the Neville-Aitken Interpolation Algorithm on Grassmann Manifolds : Applications to Reduced Order Model, https://arxiv.org/pdf/1907.02831.pdf, 2019.

[31] Y. Choi, G. Boncoraglio, S. Anderson, D. Amsallem, C. Farhat, Gradient-based constrained optimization using a database of linear reduced-order models, Journal of Computational Physics 423 (2020) 109787.

[32] K. Lee, K. Carlberg, Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders, Journal of Computational Physics 404 (2020) 108973.

[33] Y. Kim, Y. Choi, D. Widemann, T. Zohdi, A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder, 2020.

[34] M. Barrault, Y. Maday, N. Nguyen, A. Patera, An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations, Comptes Rendus Mathematiques 339(9) (2004) 666–672.

[35] D. Ryckelynck, A priori hyperreduction method: an adaptive approach, Journal of Computational Physics, Elsevier 202(1) (2005) 346–366.

[36] C. Farhat, P. Avery, T. Chapman, J. Cortial, Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency, International Journal for Numerical Methods in Engineering 98(9) (2014) 625–662.

[37] J. Hernandez, M. Caicedo, A. Ferrer, Dimensional hyper-reduction of nonlinear finite element models via empirical cubature, Computer methods in applied mechanics and engineering 313 (2017) 687–722.

[38] S. Grimberg, C. Farhat, R. Tezaur, C. Bou-Mosleh, Mesh sampling and weighting for the hyperreduction of nonlinear Petrov-Galerkin reduced-order models with local reduced-order bases (2020).

[39] J. MacQueen, Some methods for classification and analysis of multivariate observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability 1 (1967) 281–297.

[40] C. Aggarwal, Data Mining: The Textbook, Springer International Publishing, 2015. URL: `https://books.google.fr/books?id=cfNICAAAQBAJ`.

[41] L. Kaufmann, P. Rousseeuw, Clustering by means of medoids, Data Analysis based on the L1-Norm and Related Methods (1987) 405–416.

[42] C. Aggarwal, C. Reddy, Data Clustering: Algorithms and Applications, Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, Taylor & Francis, 2013. URL: `https://books.google.fr/books?id=edl7AAAAQBAJ`.

[43] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition, Springer series in statistics, Springer, 2009. URL: `https://books.google.fr/books?id=eBSgoAEACAAJ`.

[44] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: A review, ACM Comput. Surv. 31 (1999) 264–323.

[45] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. Patel, A. Tiwari, M. Er, W. Ding, C. Lin, A review of clustering techniques and developments, Neurocomputing 267 (2017).

[46] O. Balabanov, A. Nouy, Randomized linear algebra for model reduction—part ii: minimal residual methods and dictionary-based approximation, Advances in Computational Mathematics 47 (2021) 1–54.

[47] D. Amsallem, B. Haasdonk, PEBL-ROM: Projection-error based local reduced-order models, Advanced Modeling and Simulation in Engineering Sciences 3 (2016).

[48] T. Daniel, F. Casenave, N. Akkari, D. Ryckelynck, Model order reduction assisted by deep neural networks (ROM-net), Advanced Modeling and Simulation in Engineering Sciences 7 (2020).

[49] T. Henri, J. Yvon, Convergence estimates of POD-Galerkin methods for parabolic problems, volume 166, 2006, pp. 295–306. doi:`10.1007/0-387-23467-5_21`.

[50] I. Gohberg, S. Goldberg, M. Kaashoek, Classes of Linear Operators, number vol. 1 in Classes of Linear Operators, Springer, 1990. URL: `https://books.google.fr/books?id=tCbvAAAAMAAJ`.

[51] C. Cheverry, N. Raymond, Handbook of Spectral Theory, 2019. URL: `https://hal.archives-ouvertes.fr/cel-01587623`, lecture.

[52] S. Djouadi, On the optimality of the proper orthogonal decomposition and balanced truncation, 2009, pp. 4221 – 4226. doi:`10.1109/CDC.2008.4739458`.

[53] S. Djouadi, S. Sahyoun, On a generalization of the proper orthogonal decomposition and optimal construction of reduced order models, in: 2012 American Control Conference (ACC), 2012, pp. 1436–1441. doi:`10.1109/ACC.2012.6315479`.

[54] L. Sirovich, Turbulence and the dynamics of coherent structures, Parts I, II and III, Quarterly of Applied Mathematics XLV (1987) 561 – 590.

[55] A. Chatterjee, An introduction to the proper orthogonal decomposition, Current Science 78 (2000) 808 – 817.

[56] M. Meyer, H. Matthies, Efficient model reduction in non-linear dynamics using the Karhunen-Loève expansion and dual-weighted-residual methods, Computational Mechanics 31 (2003) 179–191.

[57] A. Iollo, D. Lombardi, Advection modes by optimal mass transfer, Phys. Rev. E 89 (2014) 022923.

[58] J. Reiss, P. Schulze, J. Sesterhenn, V. Mehrmann, The Shifted Proper Orthogonal Decomposition: A mode decomposition for multiple transport phenomena, SIAM Journal on Scientific Computing 40 (2018) A1322–A1344.

[59] N. Cagniart, Y. Maday, B. Stamm, Model order reduction for problems with large convection effects, In: Chetverushkin B., Fitzgibbon W., Kuznetsov Y., Neittaanmäki P., Periaux J., Pironneau O. (eds) Contributions to Partial Differential Equations and Applications. Computational Methods in Applied Sciences 47 (2019).

[60] R. Zimmermann, B. Peherstorfer, K. Willcox, Geometric subspace updates with applications to online adaptive nonlinear model reduction, SIAM Journal on Matrix Analysis and Applications 39 (2017).

[61] T. Kim, D. James, Skipping steps in deformable simulation with online model reduction, ACM Trans. Graph. 28 (2009) 1–9.

[62] M. Ohlberger, F. Schindler, Error control for the localized reduced basis multiscale method with adaptive on-line enrichment, SIAM Journal on Scientific Computing 37 (2015) A2865–A2895.

[63] F. Casenave, N. Akkari, An error indicator-based adaptive reduced order model for nonlinear structural mechanics - Application to high-pressure turbine blades, Math. Comput. Appl. 24(2) (2019).

[64] W. He, P. Avery, C. Farhat, In-situ adaptive reduction of nonlinear multiscale structural dynamics models, 2020.

[65] B. Peherstorfer, K. Willcox, Online adaptive model reduction for nonlinear systems via low-rank updates, SIAM Journal on Scientific Computing 37 (2015) A2123–A2150.

[66] B. Peherstorfer, Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling, SIAM Journal on Scientific Computing 42 (2020) A2803–A2836.

[67] P. Etter, K. Carlberg, Online adaptive basis refinement and compression for reduced-order models via vector-space sieving, Computer Methods in Applied Mechanics and Engineering 364 (2020).

[68] Y. Maday, B. Stamm, Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces, SIAM Journal on Scientific Computing 35 (2013) A2417–A2441.

[69] S. Kaulmann, B. Haasdonk, Online greedy reduced basis construction using dictionaries, VI International Conference on Adaptive Modeling and Simulation (ADMOS) (2012).

[70] M. Drohmann, B. Haasdonk, M. Ohlberger, Adaptive reduced basis methods for nonlinear convection–diffusion equations, volume 4, 2010, pp. 369–377. doi:`10.1007/978-3-642-20671-9_39`.

[71] M. Dihlmann, M. Drohmann, B. Haasdonk, Model reduction of parametrized evolution problems using the reduced basis method with adaptive time partitioning, 2011.

[72] J. Eftang, A. Patera, E. Ronquist, An "hp" certified reduced basis method for parametrized elliptic partial differential equations, SIAM J. Scientific Computing 32 (2010) 3170–3200.

[73] B. Haasdonk, M. Dihlmann, M. Ohlberger, A training set and multiple bases generation approach for parametrized model reduction based on adaptive grids in parameter space, Mathematical and Computer Modelling of Dynamical Systems 17 (2011) 423–442.

[74] B. Peherstorfer, D. Butnaru, K. Willcox, H. Bungartz, Localized Discrete Empirical Interpolation Method, SIAM Journal on Scientific Computing 36 (2014).

[75] M. Kapteyn, D. Knezevic, K. Willcox, Toward predictive digital twins via component-based reduced-order models and interpretable machine learning, AIAA Scitech 2020 Forum (2020).

[76] M. Kapteyn, K. Willcox, From physics-based models to predictive digital twins via interpretable machine learning, 2020. URL: `https://arxiv.org/abs/2004.11356`.

[77] D. Amsallem, M. Zahr, K. Washabaugh, Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction, Advances in Computational Mathematics 41 (2015).

[78] F. Nguyen, S. Barhli, D. Muñoz, D. Ryckelynck, Computer vision with error estimation for reduced order modeling of macroscopic mechanical tests, Complexity (2018).

[79] A. Buhr, L. Iapichino, K. Smetana, 6 Localized model reduction for parameterized problems, De Gruyter, 2020, pp. 245–306. URL: `https://doi.org/10.1515/9783110671490-006`. doi:`doi:10.1515/9783110671490-006`.

[80] K. Lee, K. Carlberg, Deep Conservation: A latent-dynamics model for exact satisfaction of physical conservation laws, 2020.

[81] S. Chaturantabut, D. Sorensen, Discrete empirical interpolation for nonlinear model reduction, Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference, CDC/CCC 2009, Proceedings of the 48th IEEE Conference (2010) 4316–4321.

[82] R. Dupuis, J.-C. Jouhaud, P. Sagaut, Surrogate modeling of aerodynamic simulations for multiple operating conditions using machine learning (2019).

[83] L. Mainini, K. Willcox, Surrogate modeling approach to support real-time structural assessment and decision making, AIAA Journal 53 (2015) 1612–1626.

[84] T. Daniel, F. Casenave, N. Akkari, D. Ryckelynck, Data augmentation and feature selection for automatic model recommendation in computational physics, Mathematical and Computational Applications 26 (2021).

[85] A. D. Gordon, J. T. Henderson, An algorithm for euclidean sum of squares classification, Biometrics 33 (1977) 355–362.

[86] L. Kaufman, P. Leonard Kaufman, P. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, A Wiley-Interscience publication, Wiley, 1990. URL: `https://books.google.fr/books?id=Q5wQAQAAIAAJ`.

[87] L. Kaufman, P. Rousseeuw, Clustering Large Data Sets, 1986, pp. 425–437. doi:`10.1016/B978-0-444-87877-9.50039-X`.

[88] R. Ng, J. Han, Efficient and Effective Clustering Methods for Spatial Data Mining, Technical Report, CAN, 1994.

[89] R. Ng, J. Han, CLARANS: A method for clustering objects for spatial data mining, Knowledge and Data Engineering, IEEE Transactions on 14 (2002) 1003– 1016.

[90] E. Schubert, P. Rousseeuw, Faster k-medoids clustering: Improving the PAM, CLARA, and CLARANS algorithms, in: G. Amato, C. Gennaro, V. Oria, M. Radovanović (Eds.), Similarity Search and Applications, Springer International Publishing, Cham, 2019, pp. 171–187.

[91] H. Park, C. Jun, A simple and fast algorithm for k-medoids clustering, Expert Systems with Applications 36 (2009) 3336–3341.

[92] A. Bjorck, G. Golub, Numerical methods for computing angles between linear subspaces, Mathematics of Computation 27 (1973) 123.

[93] J. Conway, R. Hardin, N. Sloane, Packing lines, planes, etc.: Packings in grassmannian space, Experimental Mathematics 5 (1996) 139–159.

[94] D. Ryckelynck, T. Goessel, F. Nguyen, Mechanical dissimilarity of defects in welded joints via Grassmann manifold and machine learning, 2020. URL: `https://hal.archives-ouvertes.fr/hal-02559710`, preprint.

[95] R. Everson, L. Sirovich, Karhunen-Loeve procedure for gappy data, JOSA A 12 (1995).

[96] T. Daniel, F. Casenave, N. Akkari, D. Ryckelynck, Optimal piecewise linear data compression for solutions of parametrized partial differential equations, arXiv preprint: 2108.12291 (2021).

[97] M. Bachmayr, A. Cohen, Kolmogorov widths and low-rank approximations of parametric elliptic PDEs, Mathematics of Computation 86 (2017) 701–724.

[98] V. Moureau, P. Domingo, L. Vervisch, Design of a massively parallel CFD code for complex geometries, Comptes Rendus Mécanique 339 (2011) 141 – 148.

[99] Mines ParisTech and ONERA the French aerospace lab. Z-set: nonlinear material & structure analysis suite, `http://www.zset-software.com`, 1981-present.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Thomas Daniel: Conceptualization, Methodology, Software, Formal analysis, Writing - Original Draft, Writing - Review & Editing. Fabien Casenave: Conceptualization, Methodology, Software, Formal analysis, Writing - Review & Editing, Supervision. Nissrine Akkari: Conceptualization, Formal analysis. Ali Ketata: Conceptualization, Methodology. David Ryckelynck: Conceptualization, Supervision.