

Advanced ROMs in SciML

Projection-based and data-
driven ROMs

Federico Pichi

20 June 2025



Scientific Machine Learning

Many problems can be tackled and solved by **ML** alone (image recognition, text generation).

Many engineering problems can be tackled by **DM** alone (structural mechanics, fluid dynamics).

~~ Some very complex problems can benefit from the interaction between **ML** algorithms and **physics**.

ML can be useful for different tasks in **DM**:

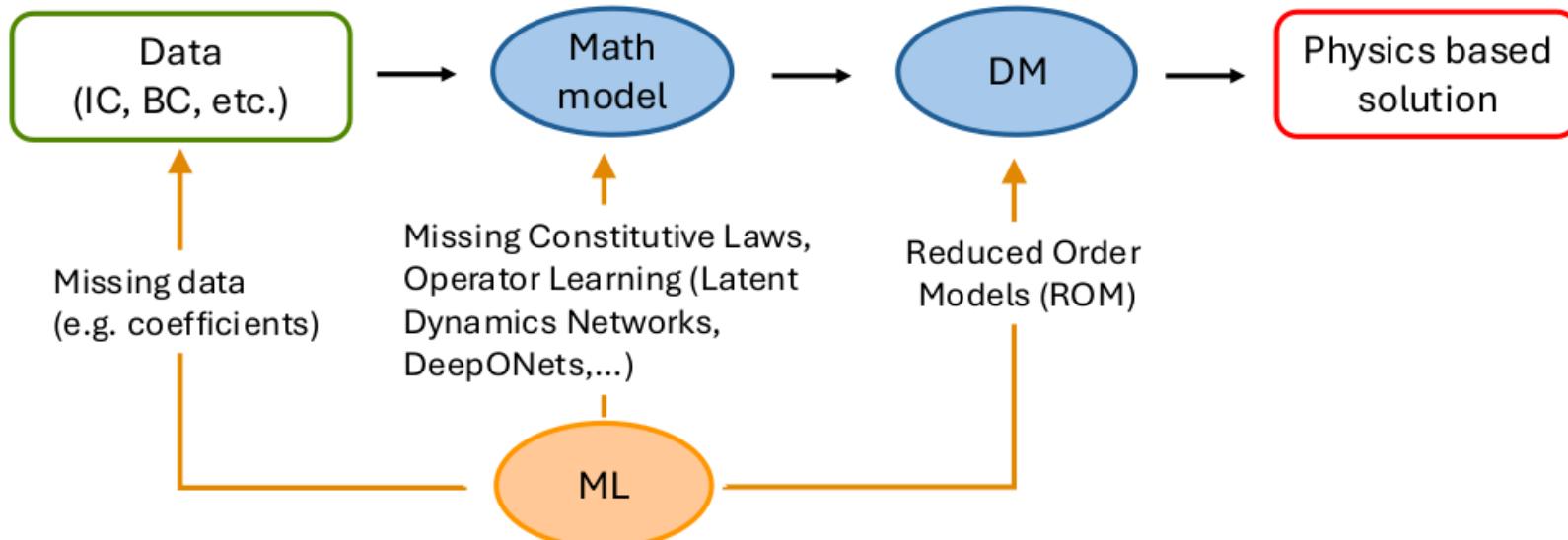
- to recover (unknown) *constitutive laws* by using input-output pairs,
- to estimate and calibrate parameters by solving *multi-query* and *inverse problems*,
- to achieve *model reduction* and improve the efficiency by accelerating solvers.

DM can be useful for different tasks in **ML**:

- to *regularize* ML algorithms by penalizing the cost function and avoid *overfitting*,
- to *augment* data for the (scarce) training set, by adding solutions obtained from numerical algorithms,
- to improve the *analytical* and *predictive* power of ML and maximize its impact on CSE applications.

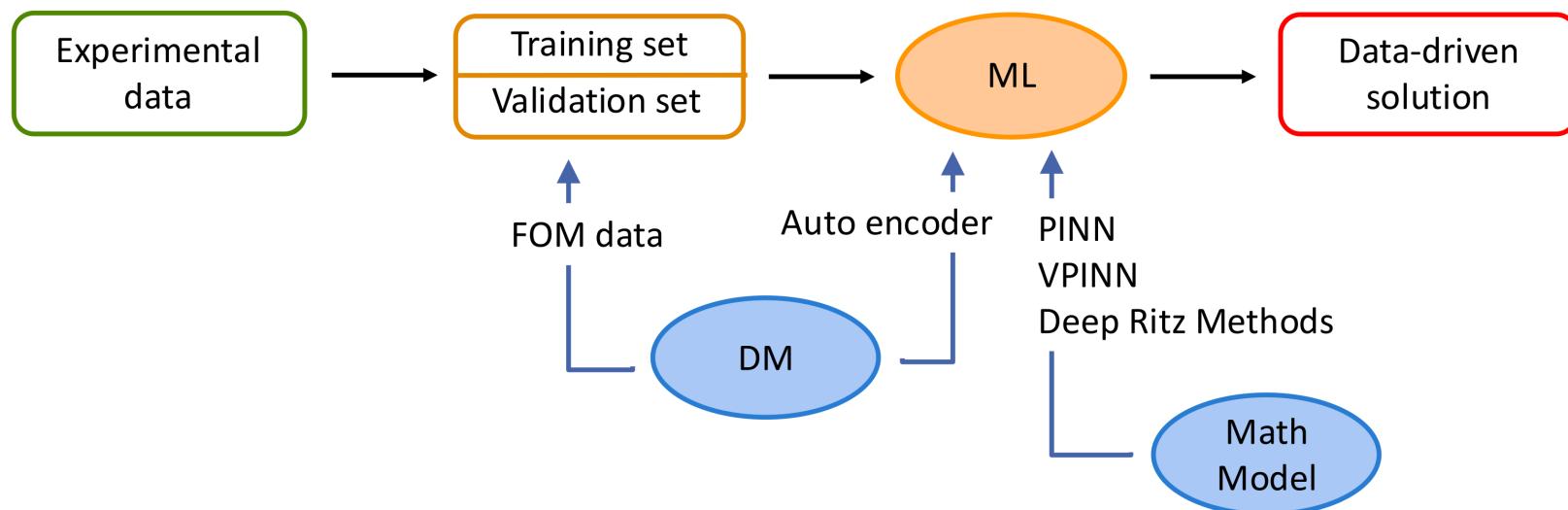
Improving digital models via machine learning

Physics-based approaches are governed by the mathematical models with data (IC, BC, forcing terms)
~~~ ML algorithms can provide missing data and *constitutive laws*, help to learn *differential or integral operators*, identify the *latent dynamics* of phenomena, and provide *ROMs* to improve the efficiency.



# Improving machine learning algorithms via digital models

*Data-driven* approaches (usually) exploit ML algorithms whose learning process is fed by datasets.  
~~ Physics principles and DM can be used to enrich datasets or to replace ML components, e.g. the loss function of the ML algorithm can incorporate information provided by the physical problem



# Domain knowledge

**Scientific knowledge** is essential to describe physical phenomena of a certain complexity and *data-driven* models alone cannot yet replace numerical modelling in these cases, indeed, they may fit training data very well, but could lack in generalization, providing physically unacceptable and meaningless results.

*Domain knowledge* includes physical principles (laws, bcs), constraints (symmetries, invariances), and structures (graph-like, non-smooth data), empowering the learning process to require fewer data points.

Many **scientific areas** are involved in *SciML*:

- *functional analysis, geometry, and linear algebra*, to study the well-posedeness of the models
- *physics*, to formulate constitutive laws
- *numerical analysis* to discretize models
- *probability and statistics* to deal with data assimilation
- *control problems* and optimization to find optimal parameters providing the required output
- *computer science* to implement methods and designing algorithms.

## Computational setting for PDEs

Let  $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^P$  be the parameter, the PDE over the domain  $\Omega(\boldsymbol{\mu}) \subset \mathbb{R}^d$ , can be expressed as

$$G(X(\boldsymbol{\mu}); \boldsymbol{\mu}) = 0 \quad \text{in } \mathbb{X}'_{\boldsymbol{\mu}}, \tag{1}$$

where  $X(\boldsymbol{\mu}) \in \mathbb{X}_{\boldsymbol{\mu}}$  is the solution belonging to some suitable Hilbert space  $\mathbb{X}_{\boldsymbol{\mu}} := \mathbb{X}(\Omega(\boldsymbol{\mu}))$ , and  $G : \mathbb{X}_{\boldsymbol{\mu}} \times \mathcal{P} \rightarrow \mathbb{X}'_{\boldsymbol{\mu}}$  is the parametrized map representing the PDE.

The *weak formulation* of the problem reads as: given  $\boldsymbol{\mu} \in \mathcal{P}$ , find  $X(\boldsymbol{\mu}) \in \mathbb{X}_{\boldsymbol{\mu}}$  such that

$$g(X(\boldsymbol{\mu}), Y; \boldsymbol{\mu}) := \langle G(X(\boldsymbol{\mu}); \boldsymbol{\mu}), Y \rangle_{\mathbb{X}'_{\boldsymbol{\mu}}, \mathbb{X}_{\boldsymbol{\mu}}} = 0 \quad \forall Y \in \mathbb{X}_{\boldsymbol{\mu}}. \tag{2}$$

**Remark:** If  $Z(\boldsymbol{\mu}) \in \mathbb{X}_{\boldsymbol{\mu}}$  is a solution of (1), to ensure the well-posedness of the problem, we need to assume that the Frechét partial derivative  $D_X G(Z; \boldsymbol{\mu}) : \mathbb{X}_{\boldsymbol{\mu}} \rightarrow \mathbb{X}'_{\boldsymbol{\mu}}$  is bijective, meaning that the variational form  $dg[Z](X, Y; \boldsymbol{\mu}) := \langle D_X G(Z; \boldsymbol{\mu}) X, Y \rangle$  is continuous and inf-sup-stable for all  $\boldsymbol{\mu} \in \mathcal{P}$ .

## Parametrized geometries and discretization

To deal with (2) one expresses the weak formulation over a reference domain  $\overline{\Omega} = \Omega(\bar{\boldsymbol{\mu}}) \subset \mathbb{R}^d$ , by defining the (affine) transformation map  $\Phi : \overline{\Omega} \times \mathcal{P} \rightarrow \Omega(\boldsymbol{\mu})$  such that  $\Omega(\boldsymbol{\mu}) = \Phi(\overline{\Omega}, \boldsymbol{\mu})$ , and pull back the problem over  $\overline{\Omega}$  as follows: given  $\boldsymbol{\mu} \in \mathcal{P}$ , find  $X(\boldsymbol{\mu}) \in \mathbb{X}$  such that

$$g(X(\boldsymbol{\mu}), Y; \boldsymbol{\mu}) = 0 \quad \forall Y \in \mathbb{X}.$$

$\rightsquigarrow$  The solution over the original domain  $\Omega(\boldsymbol{\mu})$  can be recovered as  $X(\boldsymbol{\mu}) \circ \Phi(\cdot; \boldsymbol{\mu})$ .

Given a finite dimensional subspace  $\mathbb{X}_{\mathcal{N}} \subset \mathbb{X}$  of dimension  $\mathcal{N}$ , the Galerkin-FE method reads as: given  $\boldsymbol{\mu} \in \mathcal{P}$ , we seek  $X_{\mathcal{N}}(\boldsymbol{\mu}) \in \mathbb{X}_{\mathcal{N}}$  that satisfies

$$g(X_{\mathcal{N}}(\boldsymbol{\mu}), Y_{\mathcal{N}}; \boldsymbol{\mu}) = 0, \quad \forall Y_{\mathcal{N}} \in \mathbb{X}_{\mathcal{N}}.$$

Exploiting the Newton-Kantorovich nonlinear solver: for  $\boldsymbol{\mu} \in \mathcal{P}$  and an initial guess  $X_{\mathcal{N}}^0(\boldsymbol{\mu}) \in \mathbb{X}_{\mathcal{N}}$ , for every  $k = 0, 1, \dots$ , we seek the variation  $\delta X_{\mathcal{N}} \in \mathbb{X}_{\mathcal{N}}$  such that

$$\mathrm{dg}[X_{\mathcal{N}}^k(\boldsymbol{\mu})](\delta X_{\mathcal{N}}, Y_{\mathcal{N}}; \boldsymbol{\mu}) = g(X_{\mathcal{N}}^k(\boldsymbol{\mu}), Y_{\mathcal{N}}; \boldsymbol{\mu}), \quad \forall Y_{\mathcal{N}} \in \mathbb{X}_{\mathcal{N}}, \quad (3)$$

updating the solution as  $X_{\mathcal{N}}^{k+1}(\boldsymbol{\mu}) = X_{\mathcal{N}}^k(\boldsymbol{\mu}) - \delta X_{\mathcal{N}}$  until a suitable stopping criterion is reached.

## Algebraic formulation

From the algebraic point of view, we denote with  $\{E^j\}_{j=1}^{\mathcal{N}}$  a Lagrangian basis for  $\mathbb{X}_{\mathcal{N}}$ , such that, denoting the solution vector as  $\mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}) = \{X_{\mathcal{N}}^{(j)}(\boldsymbol{\mu})\}_{j=1}^{\mathcal{N}}$ , we can write each element  $X_{\mathcal{N}}(\boldsymbol{\mu}) \in \mathbb{X}_{\mathcal{N}}$  as

$$X_{\mathcal{N}}(\boldsymbol{\mu}) = \sum_{j=1}^{\mathcal{N}} X_{\mathcal{N}}^{(j)}(\boldsymbol{\mu}) E^j.$$

Thus, we return to the study of the solution  $\mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}) \in \mathbb{R}^{\mathcal{N}}$  of the system provided by the combination of the Newton-Kantorovich method's  $k$ -th step and the Galerkin-FE discretization: given  $\boldsymbol{\mu} \in \mathcal{P}$  and an initial guess  $\mathbf{X}_{\mathcal{N}}^0(\boldsymbol{\mu}) \in \mathbb{R}^{\mathcal{N}}$ , for every  $k = 0, 1, \dots$ , we seek the variation  $\delta \mathbf{X}_{\mathcal{N}} \in \mathbb{R}^{\mathcal{N}}$  such that

$$\mathbf{J}_{\mathcal{N}}(\mathbf{X}_{\mathcal{N}}^k(\boldsymbol{\mu}); \boldsymbol{\mu}) \delta \mathbf{X}_{\mathcal{N}} = \mathbf{G}_{\mathcal{N}}(\mathbf{X}_{\mathcal{N}}^k(\boldsymbol{\mu}); \boldsymbol{\mu}), \quad (4)$$

where the high-fidelity residual vector in  $\mathbb{R}^{\mathcal{N}}$  and Jacobian matrix in  $\mathbb{R}^{\mathcal{N} \times \mathcal{N}}$  are defined as

$$\begin{aligned} (\mathbf{G}_{\mathcal{N}}(\mathbf{X}_{\mathcal{N}}^k(\boldsymbol{\mu}); \boldsymbol{\mu}))_i &= g(\mathbf{X}_{\mathcal{N}}^k(\boldsymbol{\mu}), E^i; \boldsymbol{\mu}), & \forall i = 1, \dots, \mathcal{N}, \\ (\mathbf{J}_{\mathcal{N}}(\mathbf{X}_{\mathcal{N}}^k(\boldsymbol{\mu}); \boldsymbol{\mu}))_{ij} &= dg[\mathbf{X}_{\mathcal{N}}^k(\boldsymbol{\mu})](E^j, E^i; \boldsymbol{\mu}), & \forall i, j = 1, \dots, \mathcal{N}. \end{aligned}$$

# Reduced Order Modelling

**Goal:** obtain fast and reliable evaluations of the solutions for nonlinear parametrized PDEs

~ Reduced Basis method with offline/online phases, for many-query and real-time computations

- project the high-fidelity model onto a low-dimensional reduced space,
- independence from the degrees of freedom  $\mathcal{N}$  of the underlying high-fidelity method,
- solution as a linear combination of the RB functions which span  $\mathbb{X}_N \subset \mathbb{X}_{\mathcal{N}}$ .

Different ways to construct a basis for the reduced manifold  $\mathbb{X}_N$ :

- POD - Proper Orthogonal Decomposition, a singular value decomposition based algorithm exploited to extract the most meaningful information about the system.
- Greedy approach, iterative strategies exploiting a-posteriori error estimator to sample the worst-approximated parameter, and add it to the reduced space.

# Proper Orthogonal Decomposition

Given  $\mathcal{P}_{\text{train}} = \{\boldsymbol{\mu}^{(i)}\}_{i=1}^{N_{\text{train}}} \subset \mathcal{P}$ , one solves  $N_{\text{train}}$  Galerkin-FE problems associated to values in  $\mathcal{P}_{\text{train}}$ .  
 ↵ the so-called *snapshots* matrix from the high-fidelity solutions:

$$\mathbf{S} = [\mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}^{(1)}) \mid \dots \mid \mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}^{(N_{\text{train}})})] \in \mathbb{R}^{\mathcal{N} \times N_{\text{train}}}.$$

Applying the POD algorithm to  $\mathbf{S}$ , that is performing a Singular Value Decomposition (SVD) of  $\mathbf{S}$ , reads

$$\mathbf{S} = \mathbf{U}\Sigma\mathbf{Z}^T,$$

where  $\mathbf{U} = [\boldsymbol{\zeta}_1 \mid \dots \mid \boldsymbol{\zeta}_{\mathcal{N}}] \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$  and  $\mathbf{Z} = [\boldsymbol{\psi}_1 \mid \dots \mid \boldsymbol{\psi}_{N_{\text{train}}}] \in \mathbb{R}^{N_{\text{train}} \times N_{\text{train}}}$  are orthogonal matrices, and  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{\mathcal{N} \times N_{\text{train}}}$  with  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$ . Then, we can write

$$\begin{aligned} \mathbf{S}\boldsymbol{\psi}_i &= \sigma_i \boldsymbol{\zeta}_i \quad \text{and} \quad \mathbf{S}^T \boldsymbol{\zeta}_i = \sigma_i \boldsymbol{\psi}_i, \quad i = 1, \dots, r \\ \mathbf{S}^T \mathbf{S} \boldsymbol{\psi}_i &= \sigma_i^2 \boldsymbol{\psi}_i \quad \text{and} \quad \mathbf{S} \mathbf{S}^T \boldsymbol{\zeta}_i = \sigma_i^2 \boldsymbol{\zeta}_i, \quad i = 1, \dots, r \end{aligned}$$

The matrix  $\mathbf{C} = \mathbf{S}^T \mathbf{S} \in \mathbb{R}^{N_{\text{train}} \times N_{\text{train}}}$  is called correlation matrix, and its elements are given by

$$C_{ij} = (\mathbf{X}_{\mathcal{N}}^i)^T \mathbf{X}_{\mathcal{N}}^j, \quad 1 \leq i, j \leq N_{\text{train}}.$$

## Proper Orthogonal Decomposition

For any  $N \leq N_{\text{train}}$ , the POD basis  $\mathbf{V} = [\zeta_1, \dots, \zeta_N] \in \mathbb{R}^{\mathcal{N} \times N}$  of dimension  $N$  is defined as the set of the first  $N$  left singular vectors of  $\mathbf{U}$  or, equivalently, the set of vectors

$$\zeta_j = \frac{1}{\sigma_j} S \psi_j, \quad 1 \leq j \leq N$$

obtained from the first  $N$  eigenvectors  $\psi_1, \dots, \psi_N$  of the correlation matrix  $\mathbf{C}$ .

By construction, the POD basis is orthonormal and it minimizes, over all possible  $N$ -dimensional orthonormal bases  $\mathbf{W} = [\mathbf{w}_1 \mid \dots \mid \mathbf{w}_N] \in \mathbb{R}^{\mathcal{N} \times N}$ , the sum of the squares of the errors between each snapshot vector  $\mathbf{X}_{\mathcal{N}}^i$  and its projection onto the subspace spanned by  $\mathbf{W}$ .

**Remark:** the projection  $\Pi_{\mathbf{W}} \mathbf{x}$  of a vector  $\mathbf{x} \in \mathbb{R}^{\mathcal{N}}$  onto  $\text{span}(\mathbf{W})$  is given by

$$\mathbb{P}_{\mathbf{W}} \mathbf{x} = \sum_{j=1}^N (\mathbf{x}, \mathbf{w}_j)_2 \mathbf{w}_j = \mathbf{W} \mathbf{W}^T \mathbf{x},$$

## Projection error

**Theorem.** Let  $\mathbb{V}_N = \{\mathbf{W} \in \mathbb{R}^{N \times N} : \mathbf{W}^T \mathbf{W} = \mathbb{I}_N\}$  be the set of all  $N$ -dim. orthonormal bases.

Denoting by  $\mathbf{V} \in \mathbb{R}^{N \times N}$  the POD basis of dimension  $N$ , the following property holds:

$$\sum_{i=1}^{N_{\text{train}}} \|\mathbf{X}_{\mathcal{N}}^i - \mathbb{P}_{\mathbf{V}} \mathbf{X}_{\mathcal{N}}^i\|_2^2 = \sum_{i=1}^{N_{\text{train}}} \|\mathbf{X}_{\mathcal{N}}^i - \mathbf{V} \mathbf{V}^T \mathbf{X}_{\mathcal{N}}^i\|_2^2 = \min_{\mathbf{W} \in \mathbb{V}_N} \sum_{i=1}^{N_{\text{train}}} \|\mathbf{X}_{\mathcal{N}}^i - \mathbf{W} \mathbf{W}^T \mathbf{X}_{\mathcal{N}}^i\|_2^2 = \sum_{i=N+1}^r \sigma_i^2.$$

The error in the POD basis is equal to the sum of the squares of the singular values corresponding to the neglected POD modes. This result suggests a suitable criterion to select the minimal POD dimension  $N \leq r$  such that the projection error is smaller than a desired tolerance  $\varepsilon_{\text{POD}}$ . Indeed, it is sufficient to choose  $N$  as the smallest integer such that

$$I(N) = \frac{\sum_{i=1}^N \sigma_i^2}{\sum_{i=1}^r \sigma_i^2} \geq 1 - \varepsilon_{\text{POD}}^2,$$

that is the energy retained by the last  $r - N$  modes is equal or smaller than  $\varepsilon_{\text{POD}}^2$ .

$I(N)$  represents the percentage of snapshots energy captured by the first  $N$  POD modes.

## POD with energy inner product

Since the snapshots functions  $\mathbf{X}_{\mathcal{N}}^i$  belong to  $\mathbb{X}_{\mathcal{N}} \subset \mathbb{X}$ , it is natural to seek an alternative POD basis which minimizes the  $\mathbb{X}_{\mathcal{N}}$ -norm, rather than the  $\|\cdot\|_2$  norm, of the projection error of the snapshots vectors  $\mathbf{X}_{\mathcal{N}}^i$ . In particular, denoting by  $\mathbf{X}_{\mathcal{N}}$  the mass matrix of the finite element basis for  $\mathbb{X}_{\mathcal{N}}$  such that  $(\mathbf{X}_{\mathcal{N}})_{i,j} = (E^i, E^j)_{\mathbb{X}}$ , we seek a basis  $\mathbf{W} \in \mathbb{V}_N^{\mathcal{N}}$ , with

$$\mathbb{V}_N^{\mathcal{N}} = \{\mathbf{W} \in \mathbb{R}^{\mathcal{N} \times N} : \mathbf{W}^T \mathbf{X}_{\mathcal{N}} \mathbf{W} = \mathbb{I}_N\},$$

that minimizes the squares of the  $\mathbf{X}_{\mathcal{N}}$ -norm of the error between each snapshot vector  $\mathbf{X}_{\mathcal{N}}^i$  and its  $\mathbb{X}_{\mathcal{N}}$ -orthogonal projection onto the subspace spanned by  $\mathbf{W}$ , i.e.

$$\min_{\mathbf{W} \in \mathbb{V}_N^{\mathcal{N}}} \sum_{i=1}^{N_{\text{train}}} \|\mathbf{X}_{\mathcal{N}}^i - \mathbb{P}_W^{\mathbf{X}_{\mathcal{N}}} \mathbf{X}_{\mathcal{N}}^i\|_{\mathbb{X}_{\mathcal{N}}}^2$$

where the  $\mathbf{X}_{\mathcal{N}}$ -orthogonal projection of  $\mathbf{x} \in \mathbb{R}^{\mathcal{N}}$  onto  $\text{span}(\mathbf{W})$  has been denoted by

$$\mathbb{P}_W^{\mathbf{X}_{\mathcal{N}}} \mathbf{x} = \sum_{j=1}^N (\mathbf{x}, \mathbf{w}_j)_{\mathbf{X}_{\mathcal{N}}} \mathbf{w}_j = \mathbf{W} \mathbf{W}^T \mathbf{X}_{\mathcal{N}} \mathbf{x}.$$

## Projection error with inner product

**Theorem.** Let  $S = [\mathbf{X}_{\mathcal{N}}^1 \mid \dots \mid \mathbf{X}_{\mathcal{N}}^{N_{\text{train}}}] \in \mathbb{R}^{\mathcal{N} \times N_{\text{train}}}$  be a given matrix of rank  $r \leq \min(\mathcal{N}, N_{\text{train}})$ ,  $\mathbf{X}_{\mathcal{N}} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$  a symmetric positive definite matrix,  $\tilde{S} = \mathbf{X}_{\mathcal{N}}^{1/2} S$  and  $\tilde{S} = \tilde{\mathbf{U}} \Sigma \tilde{\mathbf{Z}}^T$  its singular value decomposition, where

$$\tilde{\mathbf{U}} = [\tilde{\boldsymbol{\zeta}}_1 \mid \dots \mid \tilde{\boldsymbol{\zeta}}_{\mathcal{N}}] \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}, \quad \tilde{\mathbf{Z}} = [\tilde{\boldsymbol{\psi}}_1 \mid \dots \mid \tilde{\boldsymbol{\psi}}_{N_{\text{train}}}] \in \mathbb{R}^{N_{\text{train}} \times N_{\text{train}}}$$

are orthogonal matrices and  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{\mathcal{N} \times N_{\text{train}}}$  with  $\sigma_1 \geq \dots \geq \sigma_r$ .

Then, for  $N \leq r$ , the POD basis  $\mathbf{V} = [\mathbf{X}_{\mathcal{N}}^{-1/2} \tilde{\boldsymbol{\zeta}}_1 \mid \dots \mid \mathbf{X}_{\mathcal{N}}^{-1/2} \tilde{\boldsymbol{\zeta}}_N]$  is such that

$$\sum_{i=1}^{N_{\text{train}}} \|\mathbf{X}_{\mathcal{N}}^i - \mathbb{P}_{\mathbf{V}}^{\mathbf{X}_{\mathcal{N}}} \mathbf{X}_{\mathcal{N}}^i\|_2^2 = \sum_{i=1}^{N_{\text{train}}} \|\mathbf{X}_{\mathcal{N}}^i - \mathbf{V} \mathbf{V}^T \mathbf{X}_{\mathcal{N}} \mathbf{X}_{\mathcal{N}}^i\|_{\mathbb{X}_{\mathcal{N}}}^2 = \min_{\mathbf{W} \in \mathbb{V}_N^{\mathcal{N}}} \sum_{i=1}^{N_{\text{train}}} \|\mathbf{X}_{\mathcal{N}}^i - \mathbf{W} \mathbf{W}^T \mathbf{X}_{\mathcal{N}} \mathbf{X}_{\mathcal{N}}^i\|_{\mathbb{X}_{\mathcal{N}}}^2 = \sum_{i=N+1}^r \sigma_i^2.$$

## Greedy Algorithm

The **Greedy strategy** is an *iterative sampling strategy* to construct a *RB* space from the parameter space, fulfilling at each step a suitable *optimality criterion* that relies on the *a posteriori error estimate*.

**Issue:** POD entails a severe computational cost for solving  $N_{\text{train}}$  high-fidelity problems.

~~ construction of the reduced space by minimizing the amount of *snapshots* to be evaluated.

**Key:** an efficient *a posteriori error estimate* for the reduced error  $\|\mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}) - \mathbf{X}_N(\boldsymbol{\mu})\|_{\mathbb{X}_{\mathcal{N}}}$  for any  $\boldsymbol{\mu} \in \mathcal{P}$ .

## Greedy Algorithm

**Idea:** Construct a subspace by iteratively adding a new basis vector at each step, instead than optimizing over all possible  $N$ -dimensional subspaces.

At each iteration  $1 \leq n \leq N - 1$ , we assume that we are given a sample set

$$\Xi_n = \{\boldsymbol{\mu}^{(1)}, \dots, \boldsymbol{\mu}^{(n)}\} \subset \mathcal{P},$$

and an orthonormal basis for  $\mathbb{X}_N$

$$\mathbf{V}_n = [\boldsymbol{\zeta}_1 \mid \dots \mid \boldsymbol{\zeta}_n] \in \mathbb{R}^{\mathcal{N} \times n},$$

obtained by orthonormalization of the snapshot set

$$[\mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}^{(1)}) \mid \dots \mid \mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}^{(n)})] \in \mathbb{R}^{\mathcal{N} \times n},$$

where  $(\mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}^i))_j = \mathbf{X}_{\mathcal{N}}^{(j)}(\boldsymbol{\mu}^i)$ , for  $1 \leq j \leq \mathcal{N}, 1 \leq i \leq n$ .

## Greedy Algorithm

The goal is to find and compute the snapshot that is worst approximated by the current RB approximation. Thus, one seek the value of the parameter  $\mu$  such that

$$\mu^{n+1} = \arg \max_{\mu \in \mathcal{P}} \|\mathbf{X}_{\mathcal{N}}(\mu) - \mathbf{X}_N(\mu)\|_{\mathbb{X}_{\mathcal{N}}},$$

where  $\mathbf{X}_{\mathcal{N}}(\mu) \in \mathbb{R}^n$  denotes the solution of the RB problem.

~ $\rightsquigarrow$  a computationally prohibitive procedure requiring the (HF) information over all the parametric space.

### Weak Greedy Algorithm:

- substitute  $\mathcal{P}$  with a fine sample  $\Xi_{\text{train}} \subset \mathcal{P}$  of cardinality  $N_{\text{train}}$ .
- substitute the reduced error  $\|\mathbf{X}_{\mathcal{N}}(\mu) - \mathbf{X}_N(\mu)\|_{\mathbb{X}_{\mathcal{N}}}$  with a-posteriori error estimator  $\Delta_n(\mu)$  s.t.

$$\|\mathbf{X}_{\mathcal{N}}(\mu) - \mathbf{X}_N(\mu)\|_{\mathbb{X}_{\mathcal{N}}} \leq \Delta_n(\mu) \quad \forall \mu \in \mathcal{P}.$$

## Greedy Algorithm

Hence, at each step  $n = 1, \dots, N - 1$  of the (weak) greedy algorithm, we need to:

1. evaluate the a posteriori error bound  $\Delta_n(\mu)$  for the  $n$ -dim. RB approximation for any  $\mu \in \Xi_{\text{train}}$ ,
2. find, by solving an enumeration problem,

$$\mu^{n+1} = \arg \max_{\mu \in \Xi_{\text{train}}} \Delta_n(\mu).$$

3. check if the stopping criterion is reached, and the final size  $N$  of the RB space  $\mathbb{X}_N$  is such that

$$\max_{\mu \in \Xi_{\text{train}}} \Delta_N(\mu) \leq \varepsilon_g$$

where  $\varepsilon_g > 0$  is a prescribed, sufficiently small, stopping tolerance.

# Greedy Algorithm

## Remarks:

- a posteriori error bound must be computable in a very inexpensive way: the (weak) greedy algorithm requires only  $O(N)$  calls to the high fidelity solver, but yields  $O(Nn_{\text{train}})$  evaluations of the a posteriori error bound each one requiring the solution of a RB problem.
- the basis  $\mathbb{V}$  is kept orthonormal with respect to  $(\cdot, \cdot)_{\mathbb{X}_N}$  by iteratively orthonormalizing the new element appended to the existing basis through a Gram-Schmidt procedure.
- other estimators, such as exploiting relative error bounds, goal-oriented error estimator, or energy norms can be used to evaluate the accuracy of the RB space, but must be inexpensive.
- the choice of a good training sample is important, since  $\Xi_{\text{train}}$  should be (i) small for efficiency reasons, but (ii) sufficiently large in order to represent the parameter set.
- a posteriori error estimators not only guarantee the reliability of the reduction process, providing a certified accuracy, but also its efficiency, being sharp, cheap and asymptotically correct.

## POD-Galerkin approximation

Denoting the basis functions as  $\{\Sigma^i\}_{i=1}^N$ , we can define the RB space as  $\mathbb{X}_N = \text{span}\{\Sigma^1, \dots, \Sigma^N\}$ .

To obtain an efficient online phase we seek an approximation to the PDE solution  $\mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu})$  as a linear combination of the RB functions  $\{\Sigma^i\}_{i=1}^N$ , that is

$$\mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}) \approx \mathbf{X}_N(\boldsymbol{\mu}) = \sum_{i=1}^N \mathbf{x}_N^{(i)}(\boldsymbol{\mu}) \Sigma^i,$$

where  $\mathbf{X}_N(\boldsymbol{\mu}) = \{\mathbf{x}_N^{(i)}(\boldsymbol{\mu})\}_{i=1}^N \in \mathbb{R}^N$  is the reduced coefficients vector in the RB expansion.

Here, we denote by  $\boldsymbol{\Sigma}^i \in \mathbb{R}^{\mathcal{N}}$  the nodal values of the basis function  $\Sigma^i$  for all  $i = 1, \dots, N$ , and express with the matrix  $\mathbf{V} = [\boldsymbol{\Sigma}^1 | \dots | \boldsymbol{\Sigma}^N] \in \mathbb{R}^{\mathcal{N} \times N}$  the change-of-variable mapping from the RB basis to the FE basis, i.e. we write  $\mathbf{X}_N = \mathbf{Vx}_N$ .

## POD-Galerkin approximation

During the online phase we project the linearized weak formulation (4) onto the reduced space  $\mathbb{X}_N$ , obtaining the algebraic problem: given  $\boldsymbol{\mu} \in \mathcal{P}$ , and an initial guess  $\mathbf{x}_N^0(\boldsymbol{\mu}) \in \mathbb{R}^N$ , for every  $k = 0, 1, \dots$ , we seek the variation  $\delta\mathbf{x}_N \in \mathbb{R}^N$  such that

$$\mathbf{J}_N(\mathbf{x}_N^k(\boldsymbol{\mu}); \boldsymbol{\mu})\delta\mathbf{x}_N = \mathbf{G}_N(\mathbf{x}_N^k(\boldsymbol{\mu}); \boldsymbol{\mu}), \quad (5)$$

where  $\mathbf{G}_N \in \mathbb{R}^N$  and  $\mathbf{J}_N \in \mathbb{R}^{N \times N}$  are the reduced residual and the reduced Jacobian matrix.

We can rewrite Equation (5) as

$$\mathbf{V}^T \mathbf{J}_{\mathcal{N}}(\mathbf{V}\mathbf{x}_N^k(\boldsymbol{\mu}); \boldsymbol{\mu})\mathbf{V}\delta\mathbf{x}_N = \mathbf{V}^T \mathbf{G}_{\mathcal{N}}(\mathbf{V}\mathbf{x}_N^k(\boldsymbol{\mu}); \boldsymbol{\mu}), \quad (6)$$

updating the solution as  $\mathbf{x}_N^{k+1}(\boldsymbol{\mu}) = \mathbf{x}_N^k(\boldsymbol{\mu}) - \delta\mathbf{x}_N$  until convergence.

## Issues for projection-based ROMs: efficiency and time

- Equation (6) still involves the *degrees of freedom*  $\mathcal{N}$  of the high-fidelity problem, thus a repeated assembly of the system compromises the efficiency of the RB during the online phase.
  - ~~> adopting hyper-reduction techniques, such as the **Empirical Interpolation Method** (EIM) and its variants, to allow a consistent speed-up by recovering the affine dependency w.r.t. the parameter  $\mu$ .
  - **Time-dependent problems:** *causality, integration in time, Hamiltonian systems, moving features.*
    - ~~> POD-Greedy approaches, nested-POD, dynamical low-rank approximations, gradient flow formulation, frequency-based analysis, latent dynamics evolution.

## Issues for projection-based ROMs: Kolmogorov n-width

- $N \ll \mathcal{N}$ , if not one observes a slow decay of the **Kolmogorov n-width**. If the distance between a subspace  $\mathbb{X}_N$  and  $S$  is determined by the worst-case scenario, then finding the best  $N$ -dimensional subspace of  $\mathbb{X}$  for approximating  $S$  determines the minimum of such distance, i.e.

$$d_n(S; \mathbb{X}) = \inf_{\mathbb{X}_N \subset \mathbb{X}} \sup_{\mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}) \in S} \inf_{\mathbf{X}_N \in \mathbb{X}_N} \|\mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}) - \mathbf{X}_N\|_{\mathbb{X}}.$$

$\rightsquigarrow$  many modes to efficiently recover the solution, *shifted-pod, registration methods, optimal transport*.

|                                                  |                                                 |                                             |                            |
|--------------------------------------------------|-------------------------------------------------|---------------------------------------------|----------------------------|
| $INS(f),$                                        | $f \in \mathcal{K}_{\gamma}^{\bar{\omega}, s},$ | $d_n(\mathcal{M})_{L^2} < \exp(-n^{1/3})$   | [Schwab and Suri 1999]     |
| $-\operatorname{div}(\mu \nabla u) = f,$         | $\mu \in \mathbb{P} \subset \mathbb{R}^m,$      | $d_n(\mathcal{M})_{L^2} < \exp(-n)$         | [Babuska et al. 2007]      |
| $\partial_t u - \mu \partial_x u = 0,$           | $(\mu, t) \in [0, 1]^2,$                        | $d_n(\mathcal{M})_{L^2} > n^{-\frac{1}{2}}$ | [Ohlberger and Rave, 2015] |
| $\partial_{tt}^2 u - \mu \partial_{xt}^2 u = 0,$ | $(\mu, t) \in [0, 1]^2,$                        | $d_n(\mathcal{M})_{L^2} > n^{-\frac{1}{2}}$ | [Greif and Urban, 2019]    |

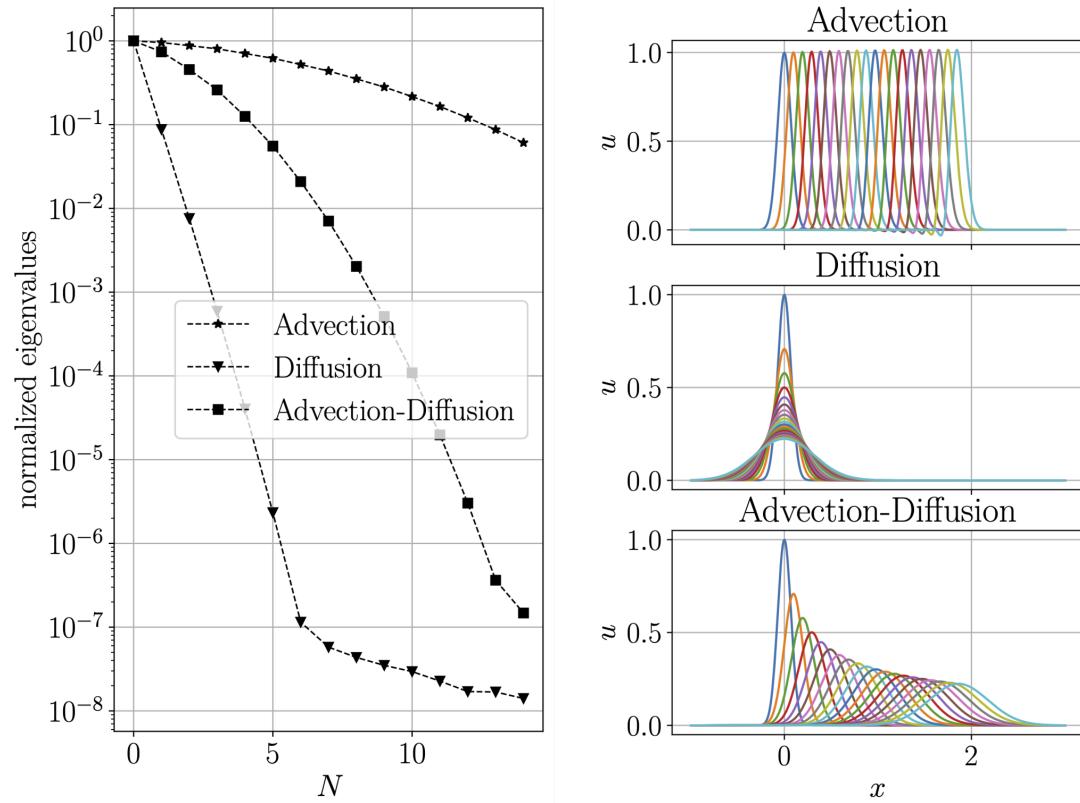
## Slow decay of Kolmogorov n-width

**Advection-dominated** problems suffer from the slow decay of the *Kolmogorov n-width*.

~~ Comparing the decay of PCA eigenvalues of the manifold  $S$  obtained by solving the following *Advection-Diffusion* system for  $t \in \mathcal{P} = [0, 0.5]$

$$\begin{cases} \partial_t u(\mathbf{x}, t) + c_T \nabla_{\mathbf{x}} \cdot u(\mathbf{x}, t) - c_D \Delta_{\mathbf{x}} u(\mathbf{x}, t) = 0 \\ u(\mathbf{x}, 0) = u_0 \end{cases}$$

- for  $c_T = 0, c_D \neq 0$ , diffusive system,
- for  $c_T \neq 0, c_D = 0$ , advective one.

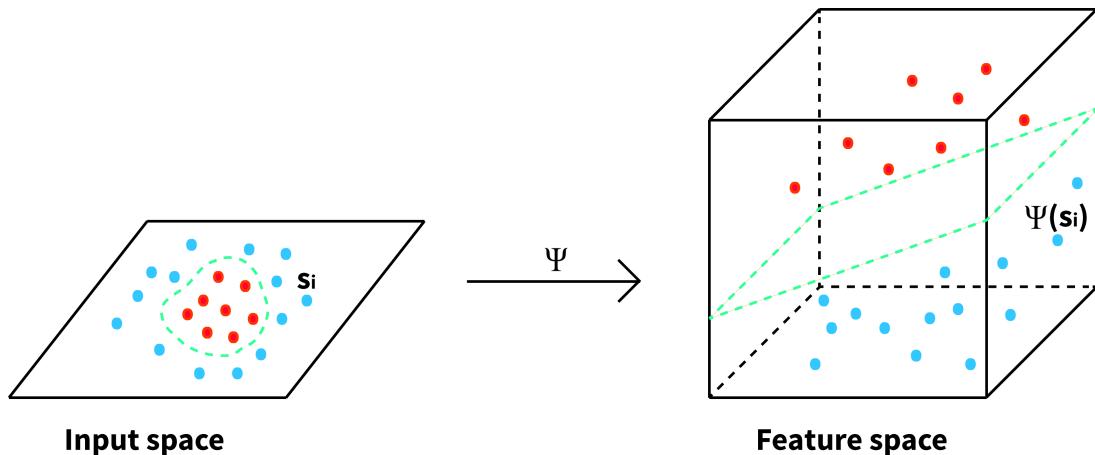


# Kernel Proper Orthogonal Decomposition (kPOD)

The kPOD algorithm is a kernel-based extension of the classical POD algorithm.

**Idea:** kPOD leverages the fact that datasets with nonlinearly separable attributes can be embedded into a higher-dimensional feature space, where the information becomes more spread out and easier to compress through POD.

**Key:** To avoid the explicit calculation of the higher-dimensional coordinates, kPOD uses the kernel function to implicitly map the data to the feature space.



## Kernel Proper Orthogonal Decomposition (kPOD)

We will describe the method starting from the description of POD, which relies on the SVD of the snapshot matrix  $\mathbf{S} \in \mathbb{R}^{\mathcal{N} \times N_{\text{train}}}$  providing a factorization of the form  $\mathbf{S} = \mathbf{U}\Sigma\mathbf{V}^T$ .

Let  $\mathbf{U}^* = [\varphi_1 | \varphi_2 | \dots | \varphi_k] \in \mathbb{R}^{\mathcal{N} \times k}$  be first  $k$  left singular vectors, the low-dimensional representation of a snapshot is given by  $\mathbf{z} = \mathbf{U}^{*\top} \mathbf{X} \in \mathbb{R}^k$ , so that the snapshots are mapped to  $\mathbf{Z} = \mathbf{U}^{*\top} \mathbf{S} \in \mathbb{R}^{k \times N_{\text{train}}}$ .

**Remark:** Notice that the SVD of the snapshot matrix  $\mathbf{S}$  directly provides a diagonalization of the *Gram* or *correlation* matrix  $\mathbf{G} = \mathbf{S}^T \mathbf{S} \in \mathbb{R}^{N_{\text{train}} \times N_{\text{train}}}$  as follows:

$$\mathbf{G} = \mathbf{V}[\Sigma^T \Sigma] \mathbf{V}^T = \mathbf{V} \Lambda \mathbf{V}^T,$$

where  $\Lambda$  is a diagonal matrix with the eigenvalues of  $\mathbf{G}$  on its diagonal, and we can reconstruct  $\mathbf{Z}$  in terms of the Gram matrix as  $\mathbf{Z} = \mathbf{V}^{*\top} \mathbf{G}$ , where  $\mathbf{V}^* \in \mathbb{R}^{N_{\text{train}} \times k}$  stores the first  $k$  columns of  $\mathbf{V}$ .

## Kernel Proper Orthogonal Decomposition (kPOD)

To apply kPOD, we introduce an arbitrary transformation  $\Phi : \mathbb{R}^{\mathcal{N}} \rightarrow \mathbb{R}^{\mathcal{M}}$  that maps the original data to a higher-dimensional feature space, where nonlinearities may become linear.

Let us assume to have  $N_{\text{train}}$  transformed snapshots  $\{\Phi(\mathbf{X}_{\mathcal{N}}^j)\}_{j=1}^{N_{\text{train}}}$ , from which we can define the matrix  $\tilde{\mathbf{S}} = [\Phi(\mathbf{X}_{\mathcal{N}}^1) | \Phi(\mathbf{X}_{\mathcal{N}}^2) | \dots | \Phi(\mathbf{X}_{\mathcal{N}}^{N_{\text{train}}})] \in \mathbb{R}^{\mathcal{N} \times N_{\text{train}}}$ . We aim at performing POD on  $\tilde{\mathbf{S}}$  to extract a low-dimensional linear subspace that captures most of the information in the transformed snapshots.

**Issue:** computing the SVD of  $\tilde{\mathbf{S}}$  directly is computationally prohibitive if  $\mathcal{N}$  is very large, and we do not know the optimal choice of  $\Phi$  that would allow us to untangle the underlying nonlinear manifold.

**Idea:** compute the transformed Gram matrix  $\tilde{\mathbf{G}}$  via a symmetric form  $\kappa(\cdot, \cdot) : \mathbb{R}^{\mathcal{N}} \times \mathbb{R}^{\mathcal{N}} \rightarrow \mathbb{R}^k$  called *kernel function* as a proxy for the inner product in the high-dimensional feature space:

$$[\tilde{\mathbf{G}}]_{ij} = \langle \Phi(\mathbf{X}_{\mathcal{N}}^i), \Phi(\mathbf{X}_{\mathcal{N}}^j) \rangle = \kappa(\mathbf{X}_{\mathcal{N}}^i, \mathbf{X}_{\mathcal{N}}^j).$$

~ this way we can use  $\tilde{\mathbf{V}}^*$  to perform the model order reduction task.

## Kernel Proper Orthogonal Decomposition (kPOD)

In particular, we can introduce the so-called **forward mapping**  $F$ :

$$F : \mathbb{R}^{\mathcal{N}} \rightarrow \mathbb{R}^k,$$
$$\mathbf{Y} \mapsto \mathbf{z} = \tilde{\mathbf{V}}^{*\top} g(\mathbf{Y}), \quad g(\mathbf{Y}) = [\kappa(\mathbf{Y}, \mathbf{X}_{\mathcal{N}}^i)]_{i=1}^{N_{\text{train}}}.$$

**Issue:** no explicit expression for the kPOD modes (unknown matrix  $\tilde{\mathbf{U}}$ ). Having only  $\tilde{\mathbf{V}}^*$  does not allow us to map backward an element  $\mathbf{z} \in \mathbb{R}^k$  into the original field  $\mathbf{X}_{\mathcal{N}} \in \mathbb{R}^{\mathcal{N}}$ .

**Goal:** define the inverse of the forward mapping  $F$  selecting a proper target space  $\mathbb{X} \subset \mathbb{R}^{\mathcal{N}}$  where pre-images are sought. The **backward mapping** is then defined as:

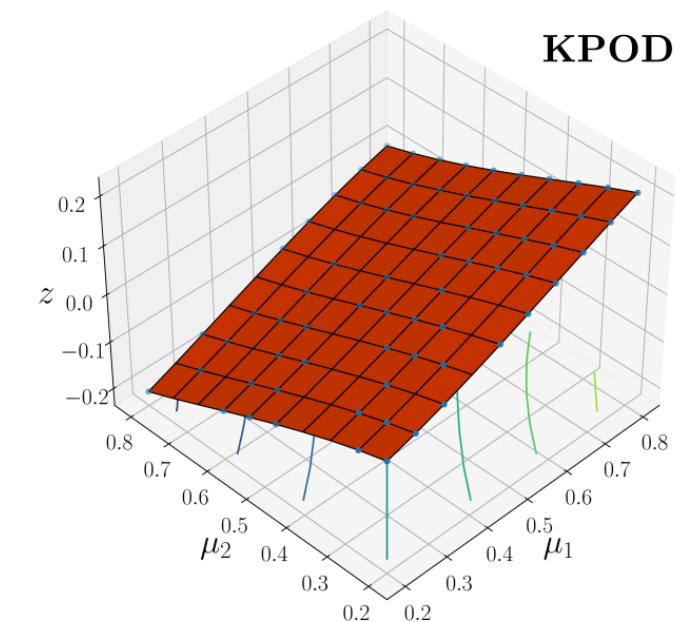
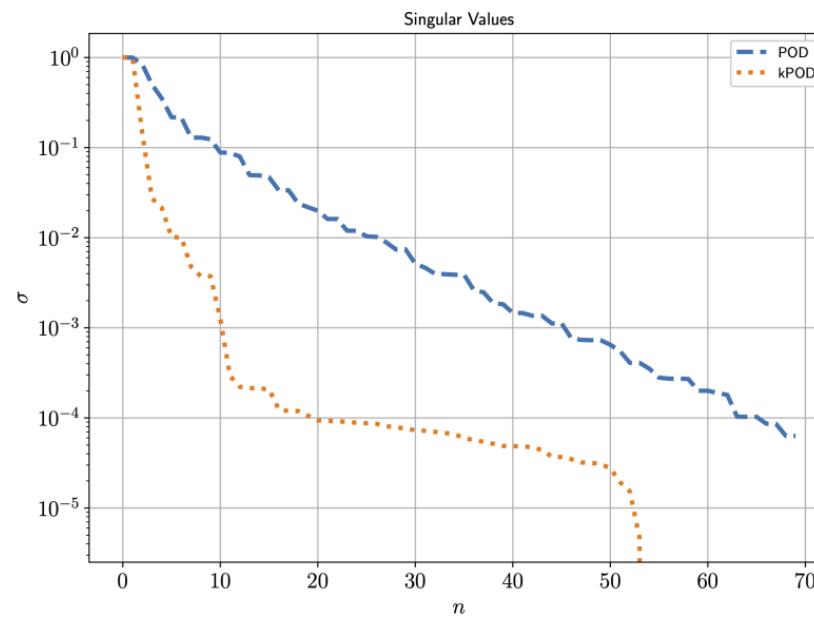
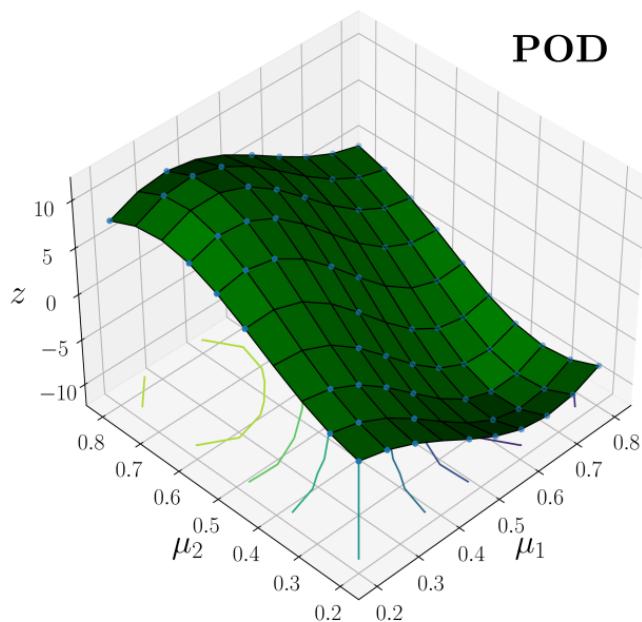
$$\tilde{\mathbf{u}}_h = F^{-1}(\mathbf{z}) \stackrel{\text{def}}{=} \arg \min_{\mathbf{X} \in \mathbb{X}} \|F(\mathbf{X}) - \mathbf{z}\|.$$

**Remark:** since  $F$  is not injective, different  $\mathbf{X}_{\mathcal{N}} \in \mathbb{R}^{\mathcal{N}}$  can have the same image  $\mathbf{z}$ .

# Kernel Proper Orthogonal Decomposition (kPOD)

Poisson equation on the 2D unit square  $\Omega = [0, 1]^2$  and  $\mathcal{P} = [0.2, 0.8]^2$  with a Gaussian source term

$$\begin{cases} \Delta u = \frac{100}{2\sigma} \exp\left(-\frac{(x-\mu_1)^2 + (y-\mu_2)^2}{2\sigma^2}\right) & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases}$$



## Non-intrusive and data-driven reduced order models

- **Linear approaches:** linear expansion with cheap reduced coefficients  $\mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}) \approx \mathbf{V}\mathbf{X}_N(\boldsymbol{\mu})$ .  
~~~ POD+interpolation, POD+Neural Networks, POD+Gaussian Process Regression, POD+PINN
- **Nonlinear approaches:** fully nonlinear mapping of the latent coordinates $\mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}) \approx \psi(\mathbf{X}_N(\boldsymbol{\mu}))$.
~~~ Autoencoders, Neural Operators, Nonlinear Manifold Least Square Petrov Galerkin

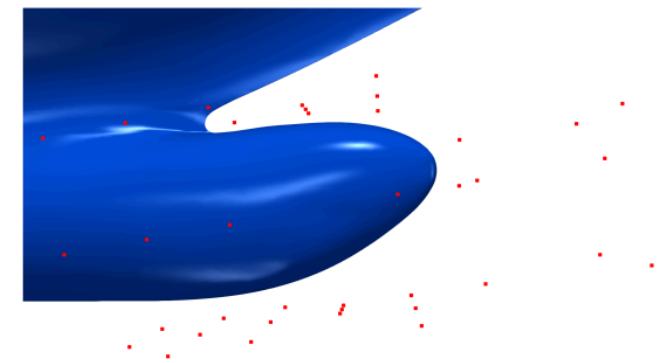
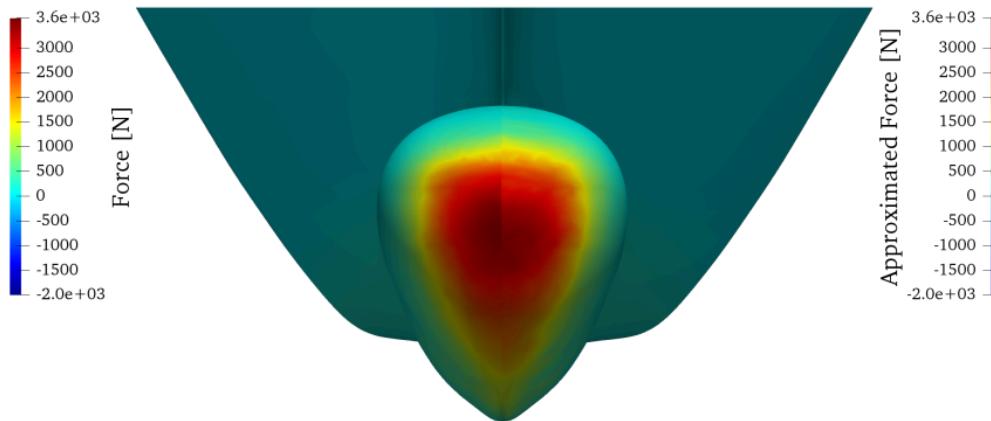
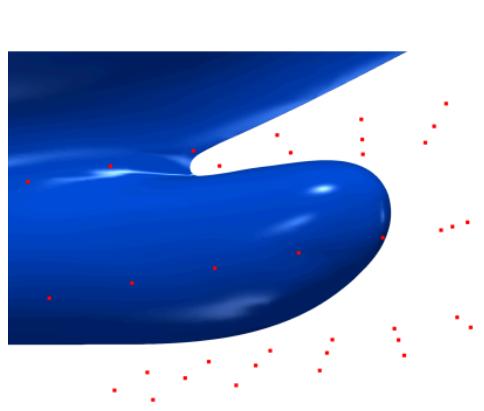
## POD + Interpolation (PODI)

Idea:

1. collect the snapshots data  $\mathbf{S} = [\mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}^{(1)}) \mid \dots \mid \mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}^{(N_{\text{train}})})] \in \mathbb{R}^{\mathcal{N} \times N_{\text{train}}}.$
2. Perform a POD  $\mathbf{S} = \mathbf{U}\Sigma\mathbf{Z}^T$  and construct a reduced basis  $\mathbf{V}$
3. Project the snapshots  $\{(\boldsymbol{\mu}^{(i)}, \mathbf{V}^T \mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}^{(i)}))\}_{i=1}^{N_{\text{train}}}$  to obtain the reduced coefficients.
4. Interpolate the low-dimensional functions  $\mathcal{I} : \boldsymbol{\mu} \mapsto \mathbf{V}^T \mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}) \in \mathbb{R}^N$  (e.g. via lagrange, RBF)
5. Non-intrusive evaluation of the field projecting onto the POD modes as  $\tilde{\mathbf{X}}(\boldsymbol{\mu}) = \mathbf{V}\mathcal{I}(\boldsymbol{\mu}) \in \mathbb{R}^{\mathcal{N}}.$

# POD + Interpolation (PODI)

## PODI reconstruction



**Bulbous bow before**

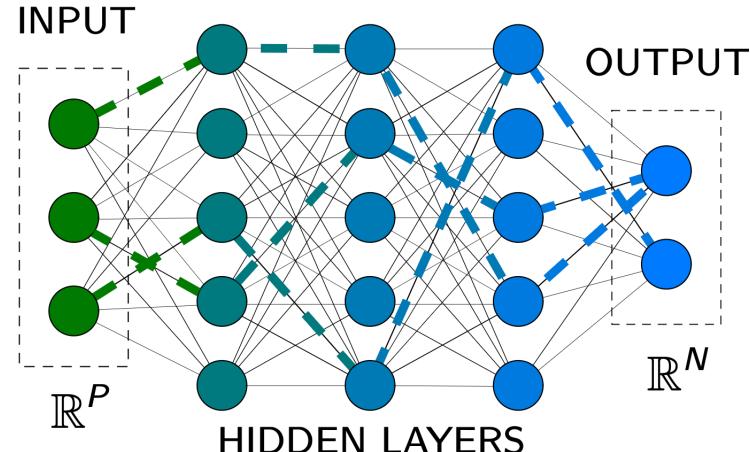
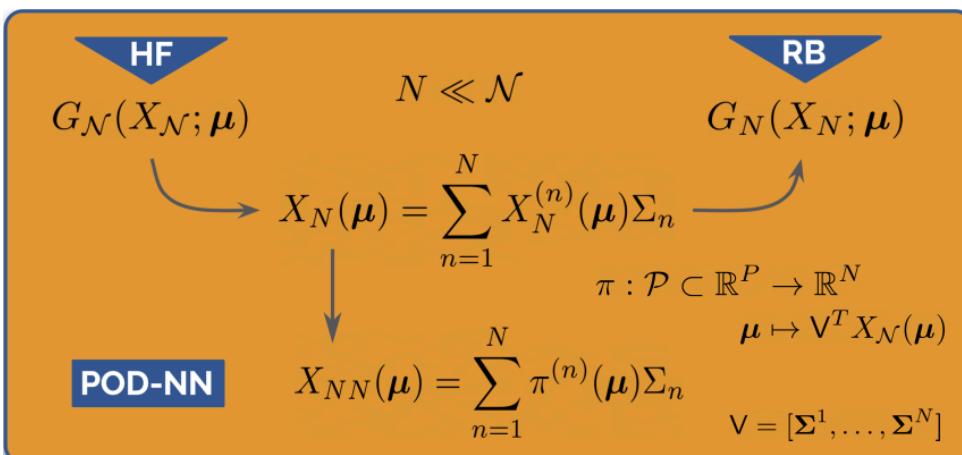
[1] T. Bui-Thanh, M. Damodaran, K. Willcox, Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics, in: 21st AIAA Applied Aerodynamics Conference, 2003

[2] Demo, N., Tezzele, M., Rozza, G., 2019. A non-intrusive approach for the reconstruction of POD modal coefficients through active subspaces. CRAS 347, 873–881. <https://doi.org/10.1016/j.crme.2019.11.012>

# POD + Neural Networks (POD-NN)

POD-NN exploits neural networks to perform a nonlinear regression task in the supervised context.

1. collect the snapshots data  $\mathbf{S} = [\mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}^{(1)}) \mid \dots \mid \mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}^{(N_{\text{train}})})] \in \mathbb{R}^{\mathcal{N} \times N_{\text{train}}}$ .
2. Perform a POD  $\mathbf{S} = \mathbf{U}\Sigma\mathbf{Z}^T$  and construct a reduced basis  $\mathbf{V}$
3. Project the snapshots  $\{(\boldsymbol{\mu}^{(i)}, \mathbf{V}^T \mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}^{(i)}))\}_{i=1}^{N_{\text{train}}}$  to obtain the reduced coefficients.
4. Train a NN to learn the map  $\pi : \boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^P \rightarrow \mathbf{V}^T \mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}) \in \mathbb{R}^N$
5. Non-intrusive evaluation of the field projecting onto the POD modes as  $\tilde{\mathbf{X}}(\boldsymbol{\mu}) = \mathbf{V}\pi(\boldsymbol{\mu}) \in \mathbb{R}^{\mathcal{N}}$ .



# POD + Neural Networks (POD-NN)

Remarks:

- learning procedure may require more snapshots than the generation of the reduced space
- learning from the best possible solution, i.e. the projection on the POD linear subspace
- perform sensitivity analysis of the POD-NN solution, i.e., study the partial derivative  $\frac{\partial \tilde{\mathbf{X}}(\boldsymbol{\mu})}{\partial \mu_i}$ ,  
~~~ the behavior of the solution w.r.t.  $\boldsymbol{\mu}$  can be controlled by the vectors of reduced coefficients.

$$\left\| \frac{\partial \tilde{\mathbf{X}}(\boldsymbol{\mu})}{\partial \mu_i} \right\|_{\mathbb{X}_N} = \left\| \sum_{k=1}^N \frac{\partial \tilde{\mathbf{X}}^{(k)}(\boldsymbol{\mu})}{\partial \mu_i} \Sigma^k \right\|_{\mathbb{X}_N} \leq \sum_{k=1}^N \left\| \frac{\partial \tilde{\mathbf{X}}^{(k)}(\boldsymbol{\mu})}{\partial \mu_i} \right\|,$$

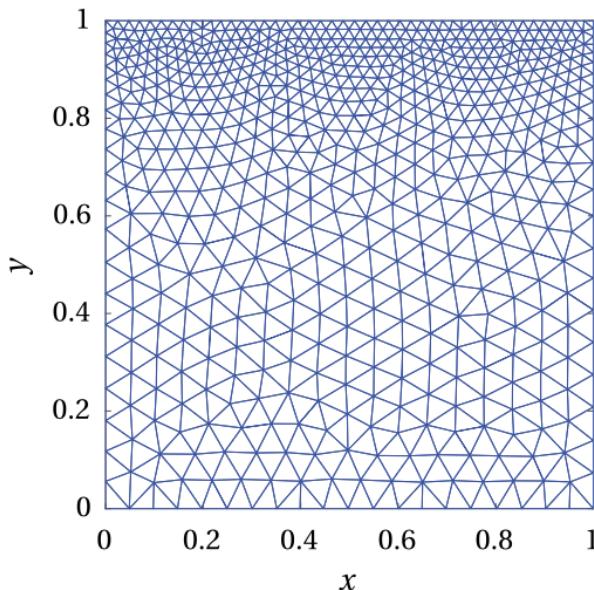
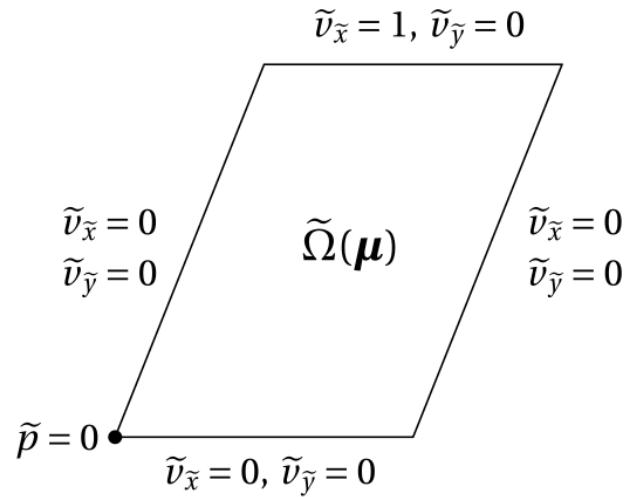
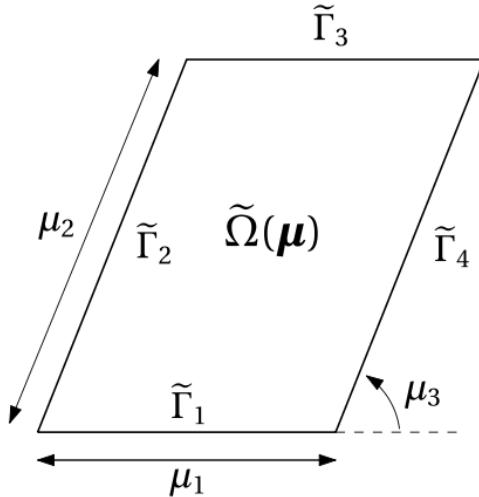
- For any training input $\boldsymbol{\mu}^{(i)} \in \mathcal{P}$, by minimizing the MSE we minimize the distance between the approximation provided by the NN and the projection of the FE solution onto the reduced space \mathbb{X}_N :

$$\text{MSE}(\pi, \mathbf{V}^T \mathbf{X}_N) \propto \|\pi - \mathbf{V}^T \mathbf{X}_N\|_{\mathbb{R}^N}^2 = \|\tilde{\mathbf{X}} - \mathbf{V} \mathbf{V}^T \mathbf{X}_N\|_{\mathbb{R}^N}^2 = \|\tilde{\mathbf{X}} - \mathbb{P}_{\mathbf{V}} \mathbf{X}_N\|_{\mathbb{X}_N}^2$$

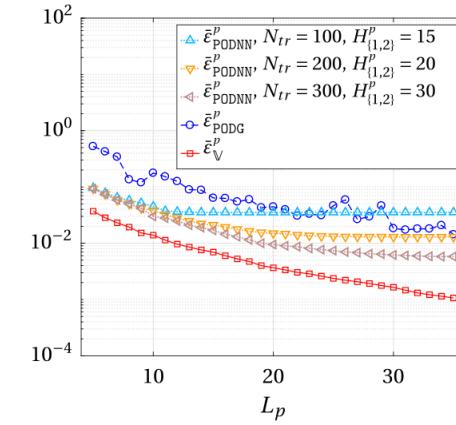
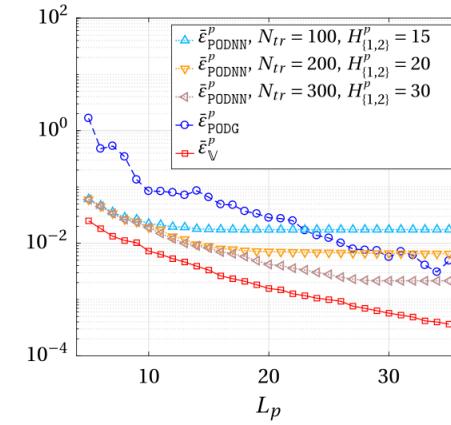
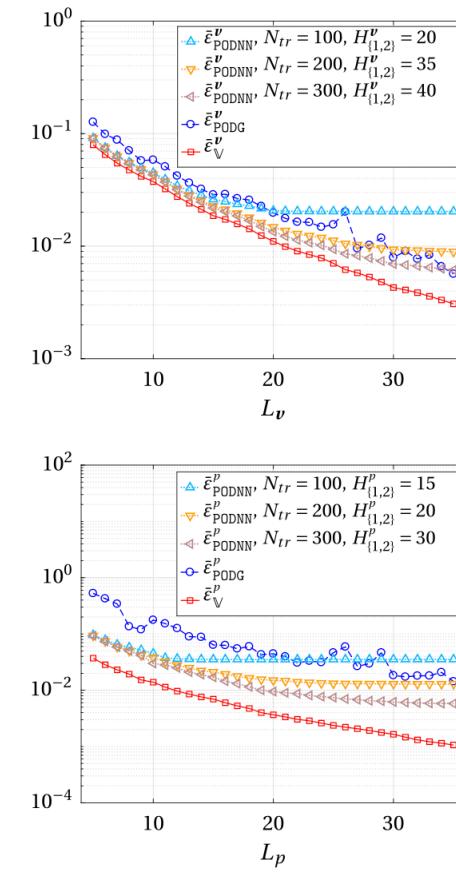
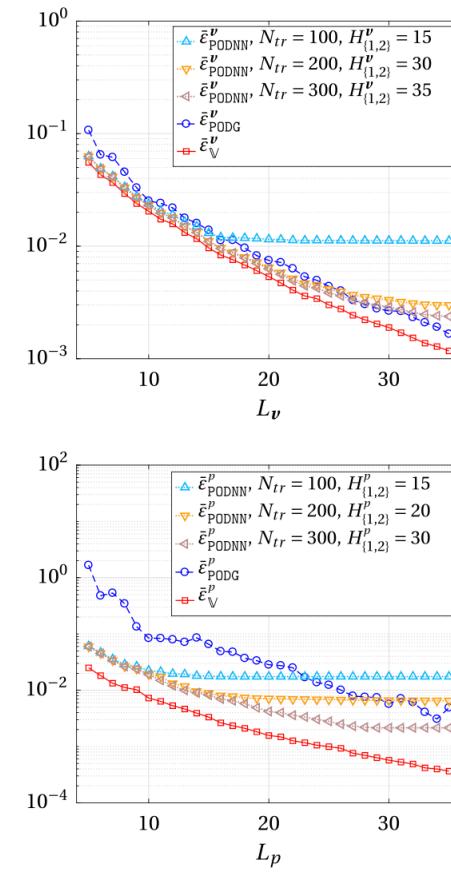
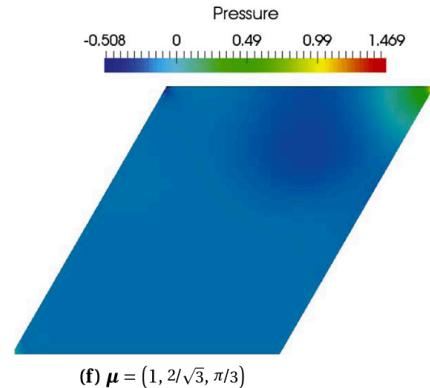
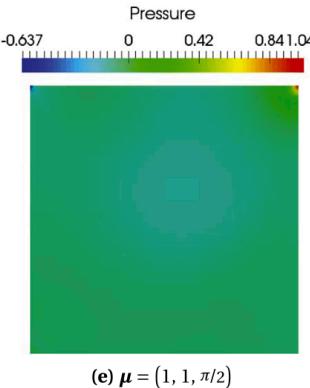
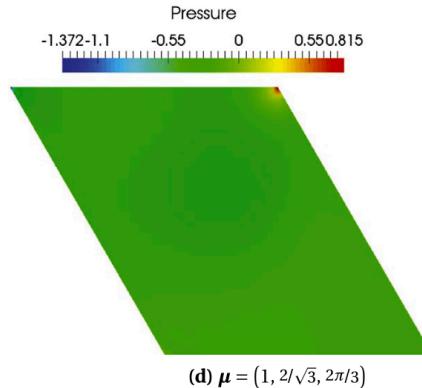
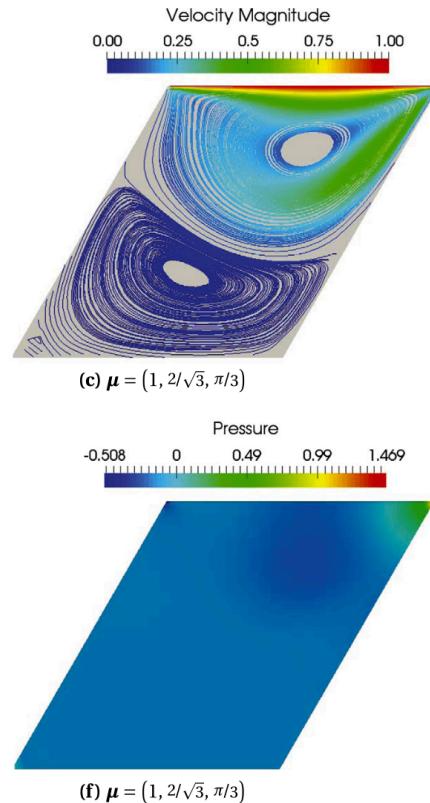
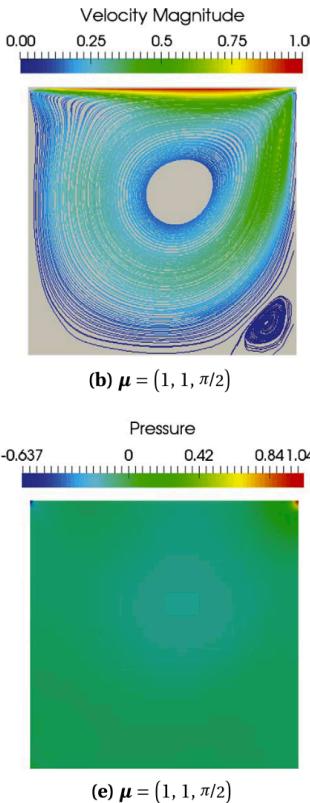
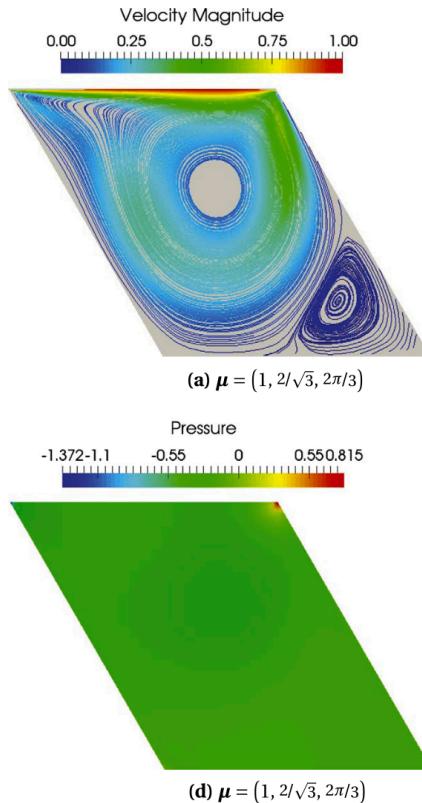
POD + Neural Networks (POD-NN)

Steady incompressible Navier–Stokes equations for a lid-driven cavity problem ($Re = 400$)

$$\begin{cases} \tilde{\nabla} \cdot \tilde{\mathbf{v}}(\boldsymbol{\mu}) = 0 & \text{in } \tilde{\Omega}(\boldsymbol{\mu}), \\ -\nu(\boldsymbol{\mu})\tilde{\Delta}\tilde{\mathbf{v}}(\boldsymbol{\mu}) + (\tilde{\mathbf{v}}(\boldsymbol{\mu}) \cdot \tilde{\nabla})\tilde{\mathbf{v}}(\boldsymbol{\mu}) + \frac{1}{\rho(\boldsymbol{\mu})}\tilde{\nabla}\tilde{p}(\boldsymbol{\mu}) = \mathbf{0} & \text{in } \tilde{\Omega}(\boldsymbol{\mu}), \\ \tilde{\mathbf{v}}(\boldsymbol{\mu}) = \tilde{\mathbf{h}} & \text{on } \tilde{\Gamma}_D(\boldsymbol{\mu}), \\ \tilde{p}(\boldsymbol{\mu})\tilde{\mathbf{n}} - \nu(\boldsymbol{\mu})\tilde{\nabla}\tilde{\mathbf{v}}(\boldsymbol{\mu}) \cdot \tilde{\mathbf{n}} = 0 & \text{on } \tilde{\Gamma}_N(\boldsymbol{\mu}). \end{cases}$$



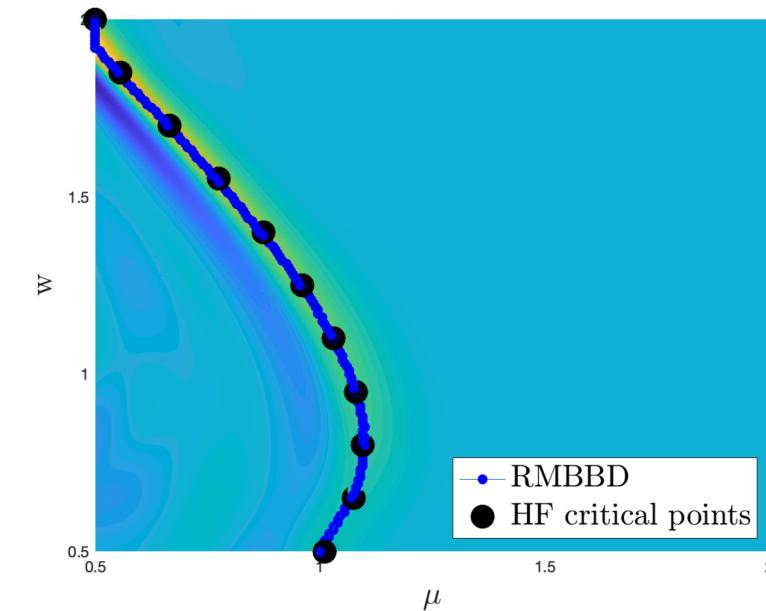
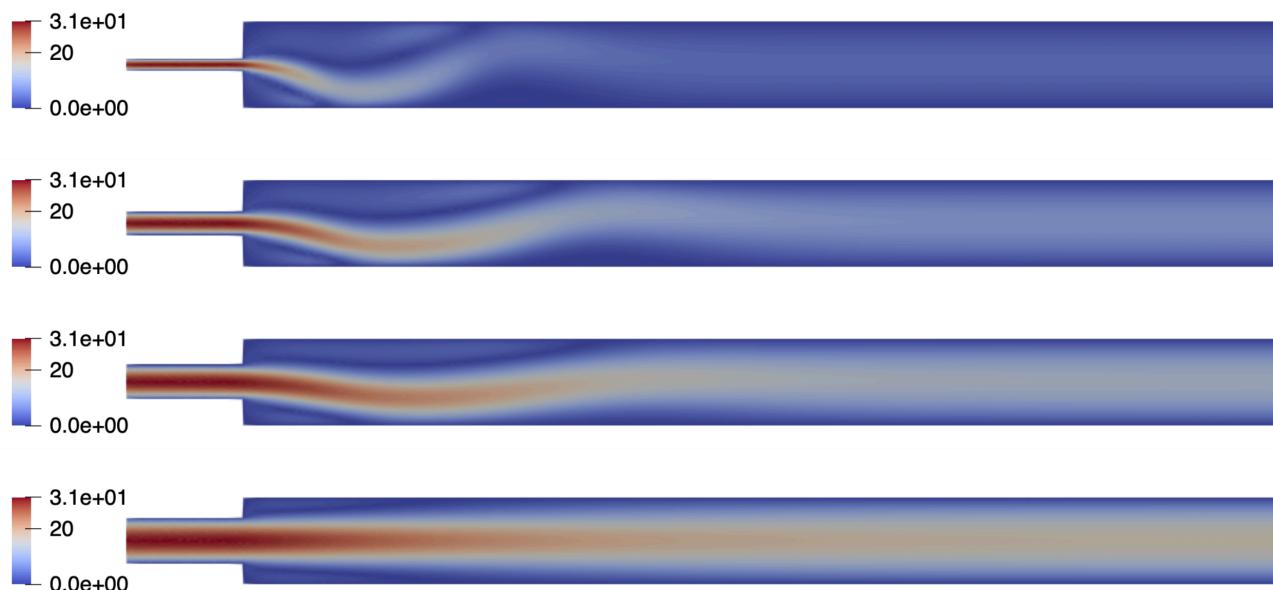
POD + Neural Networks (POD-NN)



[1] Hesthaven, J.S., Ubbiali, S., 2018. Non-intrusive reduced order modeling of nonlinear problems using neural networks. Journal of Computational Physics. <https://doi.org/10.1016/j.jcp.2018.02.037>

POD + Neural Networks (POD-NN)

Navier-Stokes system in a channel, modelling bifurcations with the Coanda effect



[2] Pichi, F., Ballarin, F., Rozza, G., Hesthaven, J.S., 2023. An artificial neural network approach to bifurcating phenomena in computational fluid dynamics. Computers & Fluids. <https://doi.org/10.1016/j.compfluid.2023.105813>

POD + Physics Informed Neural Network (POD-PINN)

In POD-NN, given $\{(\boldsymbol{\mu}^{(i)}, \mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}^{(i)}))\}_{i=1}^{N_{\text{train}}}$, we have $\Xi_{\text{POD-NN}} = \{(\boldsymbol{\mu}^{(i)}, \mathbf{V}^T \mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}^{(i)}))\}_{i=1}^{N_{\text{train}}}$ with the projected solution that we use to train the NN with weights and biases optimized w.r.t. the loss function

$$L_{\text{POD-NN}}(\Xi_{\text{POD-NN}}; \mathbf{w}) = \frac{1}{N_{\text{POD-NN}}} \sum_{i=1}^{N_{\text{POD-NN}}} \|\pi(\boldsymbol{\mu}^{(i)}; \mathbf{w}) - \mathbf{V}^T \mathbf{X}_{\mathcal{N}}(\boldsymbol{\mu}^{(i)})\|_2^2.$$

Issue: This is completely data-driven and has no information regarding the physics of the phenomena

Idea: Embed the physics at the reduced level as loss term, combining intrusive and non-intrusive methods.

Aim: Minimize the mean squared norm of the residual of the reduced-order equation on sampled training points in time-parameter space, so no labeled data is needed for the training.

For a linear problem, given a training set $\Xi_{\text{PINN}} = (\boldsymbol{\mu}^{(1)}, \dots, \boldsymbol{\mu}^{(N_{\text{PINN}})})$ we train a PINN $\pi(\boldsymbol{\mu}; \mathbf{w})$ for the reduced order equations dependent on the weights and biases \mathbf{w} of the network to minimize the loss

$$L_{\text{PINN}}(\Xi_{\text{PINN}}; \mathbf{w}) = \frac{1}{N_{\text{PINN}}} \sum_{i=1}^{N_{\text{PINN}}} \|A_N(\boldsymbol{\mu}^{(i)})\pi(\boldsymbol{\mu}^{(i)}; \mathbf{w}) - \mathbf{f}_N(\boldsymbol{\mu}^{(i)})\|_2^2$$

POD + Physics Informed Neural Network (POD-PINN)

Remark: For complex nonlinear problems, the projection of the high-fidelity solution onto the reduced space is more accurate than the solution of the reduced-order equations, since the latter do not take into account of the impact of the unresolved scales (*truncated modes*) on the resolved scales (*RB modes*).

We train a **Physics-Reinforced Neural Network** (PRNN) $\pi(\mu; \mathbf{w})$, where loss function is defined as the weighted sum of the mean squared norm of the residual of the reduced-order equation and the mean squared error between the network output and the projection coefficients of the high-fidelity solutions:

$$L_{PRNN}(\Xi_{PINN}, \Xi_{PDNN}; \mathbf{w}) = L_{PINN}(\Xi_{PINN}; \mathbf{w}) + \omega L_{PDNN}(\Xi_{PDNN}; \mathbf{w})$$

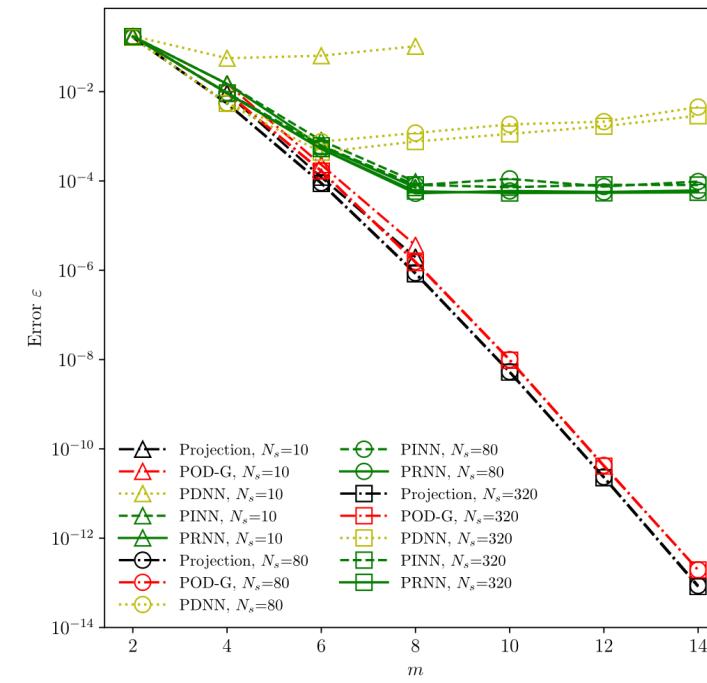
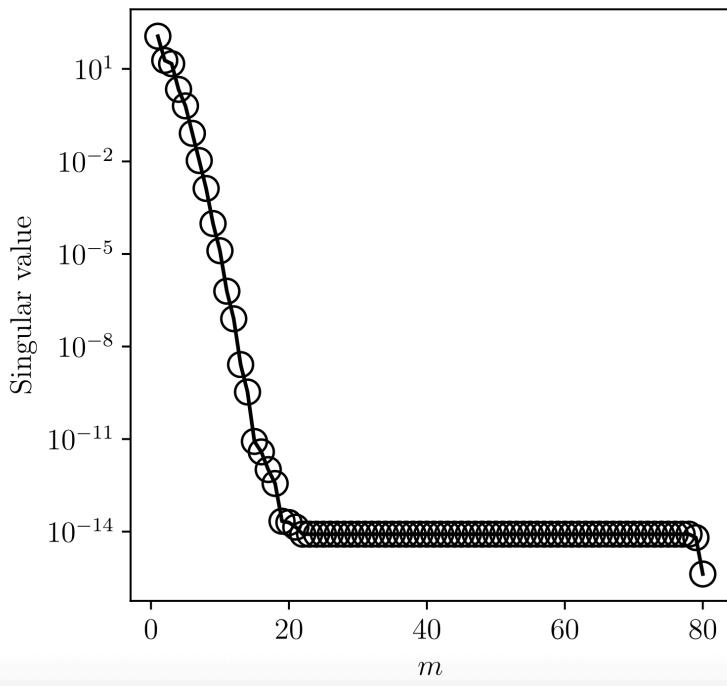
where ω is a scaling parameter which can be chosen freely.

[1] Chen, W., Wang, Q., Hesthaven, J.S., Zhang, C., 2021. Physics-informed machine learning for reduced-order modeling of nonlinear problems. Journal of Computational Physics. <https://doi.org/10.1016/j.jcp.2021.110666>

POD + Physics Informed Neural Network (POD-PINN)

Let's consider the one-dimensional parameterized Burgers' equation with $\mu \in [1, 10]^2$ and s is such that $\phi(x, \mu) = (1 + \mu_1 x) \sin(-\mu_2 x/3)(x^2 - 1)$.

$$\begin{cases} \phi(x, \mu) \frac{\partial \phi(x, \mu)}{\partial x} - \frac{\partial^2 \phi(x, \mu)}{\partial^2 x} = s(x, \mu) & \text{in } -1 \leq x \leq 1, \\ \phi(x = \pm 1, \mu) = 0 & \text{on } \partial\Omega, \end{cases}$$



POD + Physics Informed Neural Network (POD-PINN)

Let's consider now the **nonlinear** parametrized problem

$$L_{PINN}(\Xi_{PINN}; \mathbf{w}) = \frac{1}{N_{PINN}} \sum_{i=1}^{N_{PINN}} \|A_N(\boldsymbol{\mu}^{(i)})\pi(\boldsymbol{\mu}^{(i)}; \mathbf{w}) + M_N(\pi(\boldsymbol{\mu}^{(i)}; \mathbf{w})) - \mathbf{f}_N(\boldsymbol{\mu}^{(i)})\|_2^2$$

where $M_N(\pi(\boldsymbol{\mu}^{(i)}; \mathbf{w})) = \mathbf{V}^T M(\nabla \pi(\boldsymbol{\mu}^{(i)}; \mathbf{w}))$ is the reduced nonlinear term.

Issue: The computational bottleneck is now on the computation of $M_N(\pi(\boldsymbol{\mu}^{(i)}; \mathbf{w}))$ that scales with \mathcal{N} .

~~~A further cost reduction of the nonlinear term is necessary to obtain an efficient reduced-order model.

- Tensor assembly for polynomial nonlinearities
- hyper-reduction as EIM or DEIM
- NN affine approximation of the nonlinear term with NEIM

[2] Hirsch, M., Pichi, F., Hesthaven, J.S., 2024. Neural empirical interpolation method for nonlinear model reduction.

<http://arxiv.org/abs/2406.03562>

## POD + Physics Informed Neural Network (POD-PINN) + NEIM

The Neural Empirical Interpolation Method (NEIM) is a DEIM-inspired neural network based methodology to find an affine approximation of the nonlinear term.

In POD, if  $U_r$  is the basis we assemble the ROM problem as  $U_r^\top A(\mu) U_r \tilde{v} + U_r^\top N(U_r \tilde{v}; \mu) = U_r^\top f(\mu)$ , where the reduced solution is such that  $v(\mu) \approx U_r \tilde{v}(\mu)$ .

In DEIM, one approximates the nonlinear term as  $N(U_r \tilde{v}; \mu) \approx V_k (P^\top V_k)^{-1} P^\top N(U_r \tilde{v}; \mu)$ ,

where  $V_k$  is the matrix of the  $k$  left singular vectors of the matrix of nonlinear snapshots of the HF system  $S_N = [N(v(\mu^{(1)}); \mu^{(1)}), \dots, N(v(\mu^{(N_{\text{train}})}); \mu^{(N_{\text{train}})})]$  and  $P = [e_{p_1}, \dots, e_{p_k}]$  is a matrix with  $e_{p_i} = [0, \dots, 0, 1, 0, \dots, 0]^\top \in \mathbb{R}^{\mathcal{N}}$  the  $p_i$ -th column of the identity matrix for  $i = 1, \dots, k$ .

## NEIM strategy

We find an approximation of the nonlinear term of the form

$$U_r^\top \mathbf{N}(U_r \tilde{v}; \mu) \approx \widehat{\mathbf{N}}(\tilde{v}; \mu) \doteq \sum_{i=1}^k \theta_i(\mu) M_{\mu^{(i)}}(\tilde{v}),$$

where  $M_{\mu^{(i)}} : \mathbb{R}^r \rightarrow \mathbb{R}^r$  is the neural network corresponding to  $\mu^{(i)}$  for  $i = 1, \dots, k$ , and  $\theta_i(\mu)$  is determined through interpolation of some "optimal"  $\theta_i$ .

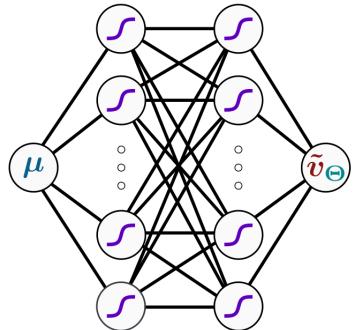
We reinterpret the affine decomposition assumption from the neural network point of view, through a linear combination of parameter dependent coefficients  $\theta_i$  and parameter independent networks  $M_{\mu^{(i)}}$ .

**Remark:** Note that  $\widehat{\mathbf{N}}(\tilde{v}; \mu) \in \mathbb{R}^r$ , while  $\mathbf{N}(U_r \tilde{v}; \mu) \in \mathbb{R}^{\mathcal{N}}$ .

# NEIM architecture

## NEIM for PINN

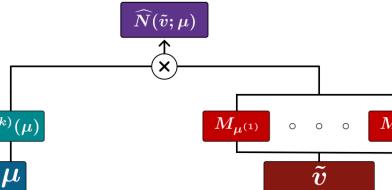
**Goal:** Approximate parameter-to-POD-coefficients map  $\mu \mapsto \tilde{v}_\Theta(\mu)$



- $\mu$  parameters
- $\Theta$  neural network weights
- $\text{activation}$  on hidden layer nodes
- $\tilde{v}_\Theta$  solution coefficients

### 1 Compute NEIM approximation

$$(\tilde{v}, \mu) \mapsto \hat{N}(\tilde{v}; \mu) \approx U_r^\top f(U_r \tilde{v}; \mu)$$



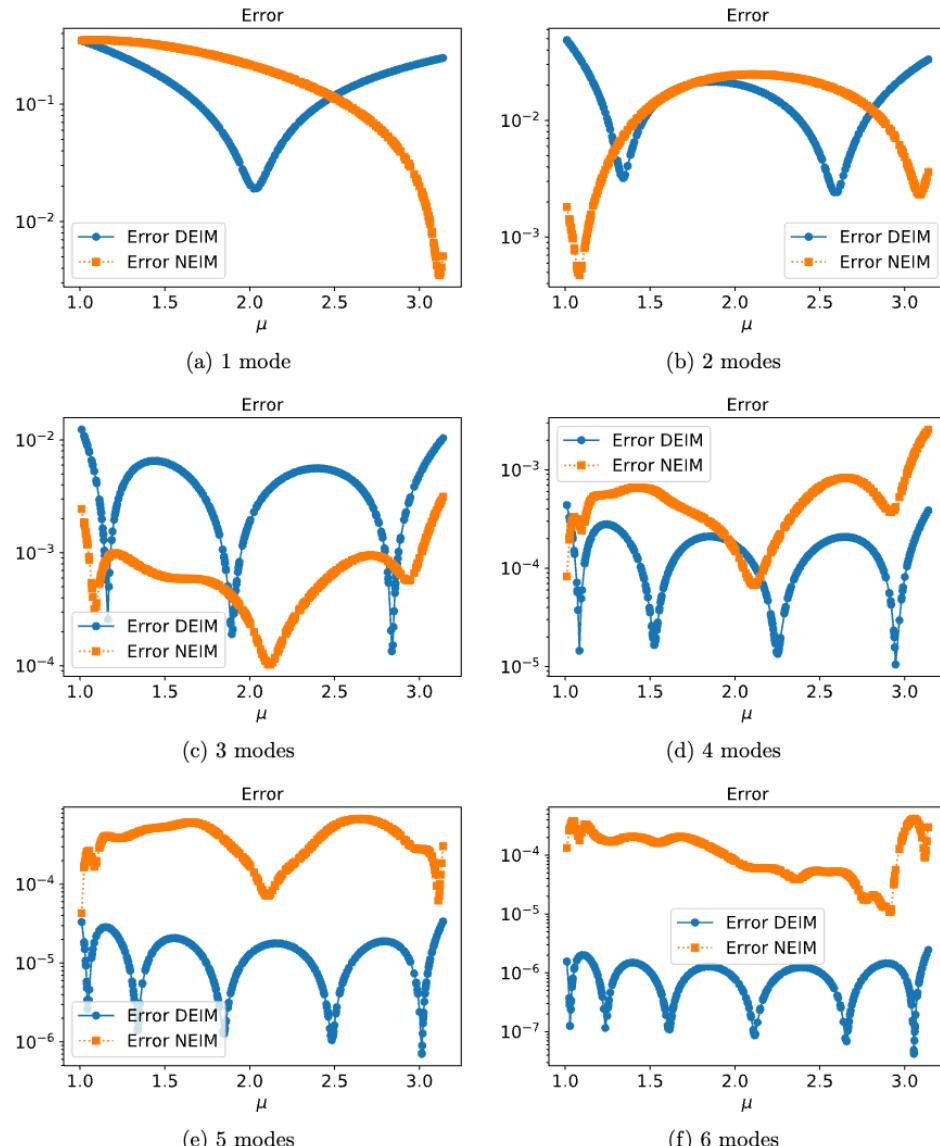
### 2 Exploit NEIM approximation in PINN loss

Find  $\Theta^*$  such that  $\mathcal{J}(\Theta^*) = \min \mathcal{J}(\Theta)$ , where

$$\mathcal{J}(\Theta) = \sum_{\mu \in \Upsilon} \|A\tilde{v}_{\Theta(\mu)} + \hat{N}(\tilde{v}_{\Theta(\mu)}; \mu) - g\|^2$$

with  $g$  reduced forcing term,  $A$  reduced stiffness matrix, and  $\Upsilon$  training set.

### 3 Obtain the approximation $\mu \mapsto \tilde{v}_{\Theta^*(\mu)}$



# NEIM properties

## Remarks:

- In NEIM, we build an approximation of a nonlinearity by greedily choosing parameters corresponding to large error. In DEIM one computes a selection operator for the rows of the nonlinearity, and DEIM interpolates exactly at these selected rows.
- If the nonlinearity acts componentwise on its input, then the complexity of evaluating the nonlinearity depends only on the reduced dimension  $r$ . However, in contrast with NEIM, DEIM is not as efficient if the nonlinearity does not act componentwise on its input.
- NEIM does not rely on the explicit form of the nonlinearity, thus it is also possible to use NEIM in a strictly data-driven context without explicitly evaluating the nonlinearity.
- Being intrusive, DEIM will typically exhibit faster decay of error.
- Each iteration of training is slower for PINN-NEIM than for PINN-DEIM due to the need to backpropagate through the NEIM neural networks, but the loss generally decreases more per iteration for PINN-NEIM.

## References

- Hesthaven, J.S., Rozza, G., Stamm, B., 2015. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, Springer International Publishing AG, Cham. <https://doi.org/10.1007/978-3-319-22470-1>
- Quarteroni, A., Manzoni, A., Negri, F., 2016. *Reduced Basis Methods for Partial Differential Equations: An Introduction*, Springer International Publishing, Cham. <https://doi.org/10.1007/978-3-319-15431-2>
- Rozza, G., Ballarin, F., Scandurra, L., Pichi, F., 2024. *Real Time Reduced Order Computational Mechanics: Parametric PDEs Worked Out Problems*, SISSA Springer Series. <https://doi.org/10.1007/978-3-031-49892-3>
- Hesthaven, J.S., Ubbiali, S., 2018. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics* 363, 55–78. <https://doi.org/10.1016/j.jcp.2018.02.037>
- Hirsch, M., Pichi, F., Hesthaven, J.S., 2024. Neural empirical interpolation method for nonlinear model reduction. <http://arxiv.org/abs/2406.03562>
- Pichi, F., Ballarin, F., Rozza, G., Hesthaven, J.S., 2023. An artificial neural network approach to bifurcating phenomena in computational fluid dynamics. *Computers & Fluids* 254, 105813. <https://doi.org/10.1016/j.compfluid.2023.105813>
- Brivio, S., Fresca, S., Franco, N.R., Manzoni, A., 2023. Error estimates for POD-DL-ROMs: a deep learning framework for reduced order modeling of nonlinear parametrized PDEs enhanced by proper orthogonal decomposition. <https://doi.org/10.48550/arXiv.2305.04680>
- Barnett, J., Farhat, C., Maday, Y., 2023. Neural-network-augmented projection-based model order reduction for mitigating the Kolmogorov barrier to reducibility. *Journal of Computational Physics* 492, 112420. <https://doi.org/10.1016/j.jcp.2023.112420>

## References

- Fresca, S., Dede', L., Manzoni, A., 2021. A Comprehensive Deep Learning-Based Approach to Reduced Order Modeling of Nonlinear Time-Dependent Parametrized PDEs. *J Sci Comput* 87, 61. <https://doi.org/10.1007/s10915-021-01462-7>
- Fresca, S., Manzoni, A., 2022. POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering* 388, 114181. <https://doi.org/10.1016/j.cma.2021.114181>
- Chaturantabut, S., Sorensen, D.C., 2010. Nonlinear Model Reduction via Discrete Empirical Interpolation. *SIAM J. Sci. Comput.* 32, 2737–2764. <https://doi.org/10.1137/090766498>
- Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T., 2004. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique* 339, 667–672. <https://doi.org/10.1016/j.crma.2004.08.006>
- Guo, M., Hesthaven, J.S., 2018. Reduced order modeling for nonlinear structural analysis using Gaussian process regression. *Computer Methods in Applied Mechanics and Engineering* 341, 807–826. <https://doi.org/10.1016/j.cma.2018.07.017>
- Chen, W., Wang, Q., Hesthaven, J.S., Zhang, C., 2021. Physics-informed machine learning for reduced-order modeling of nonlinear problems. *Journal of Computational Physics* 446, 110666. <https://doi.org/10.1016/j.jcp.2021.110666>
- Khamlich, M., Pichi, F., Rozza, G., 2023. Optimal Transport-inspired Deep Learning Framework for Slow-Decaying Problems: Exploiting Sinkhorn Loss and Wasserstein Kernel. <https://doi.org/10.48550/arXiv.2308.13840>
- Romor, F., Stabile, G., Rozza, G., 2023. Non-linear Manifold Reduced-Order Models with Convolutional Autoencoders and Reduced Over-Collocation Method. *J Sci Comput* 94, 74. <https://doi.org/10.1007/s10915-023-02128-2>
- Gonnella, I.C., Pichi, F., Rozza, G., 2025. Nonlinear reduction strategies for data compression: a comprehensive comparison from diffusion to advection problems. <https://doi.org/10.48550/arXiv.2501.12816>