ENS Rennes and STJU Paris Elite Institute of Technology

Nov 2025 - Apr 2026

Internship thesis

Nonlinear Compressed Reduced Basis Method for Data Assimilation




Author
Raphaël LECOQ

Supervised by
Helin GONG, Yvon MADAY

Abstract

Reduced-order models are widely used to accelerate the simulation and assimilation of parametrized partial differential equations, but their performance is fundamentally limited by the linear approximation capabilities of classical reduced basis methods. This work proposes a nonlinear compressed reduced basis framework for data assimilation that combines Proper Orthogonal Decomposition (POD), the Empirical Interpolation Method (EIM), and machine learning. We first revisit the theoretical foundations of reduced basis methods, establishing the connection between POD approximation errors and the Kolmogorov N-width of the solution manifold. Motivated by the observation that the intrinsic dimension of the parameter manifold is often much smaller than the number of POD modes required for accurate reconstruction, we decompose the reduced coordinates into a low-dimensional linear component and a nonlinear residual. The latter is approximated by a Gaussian Process Regression model that learns the mapping between low order and high order POD coefficients. This nonlinear closure is incorporated into several data assimilation strategies, including direct optimization, gradient-based reconstruction, and fixed-point iterations, to recover the full state from sparse and noisy sensor observations. Numerical experiments on analytical benchmark problems and nonlinear transport equations demonstrate that the proposed approach accurately learns the latent nonlinear structure of the reduced coordinates and improves state reconstruction while reducing the number of optimization variables. The framework aims to provide a flexible and computationally efficient extension of classical POD-EIM techniques for nonlinear reduced-order modeling and sensor-based data assimilation.

Acknowledgment :
This internship was made possible thanks to Helin GONG[1]. I am sincerely grateful to him for welcoming me to Shanghai and into his research team at the Shanghai Paris Elite Institute of Technology. His guidance, support, and availability throughout this internship were invaluable.

I would also like to express my sincere thanks to Yvon MADAY[2] for his supervision from Paris and for the many insightful discussions we had during the internship. His advice and perspective greatly contributed to this work.

Most of the ideas presented in this report originated from our joint discussions. I am especially grateful for the time both Helin GONG and Yvon MADAY devoted to understanding my results, helping me interpret them, and suggesting new directions. Their scientific insight and encouragement were instrumental in shaping this work.

Finally, I would like to warmly thank Xinyue GUO[3]. When I arrived in Shanghai, she had already laid the foundations of the project and established an efficient workflow. Throughout the internship, we worked closely together to develop the code and produce the results presented in this report. Her dedication, technical skills, and collaborative spirit made this project both productive and enjoyable.

Find the code on Github: https://github.com/RaphaeLcq/SJTU_NL_ROM_project.

For any questions, typo or corrections:

raphael.lecoq@ens-rennes.fr

[1]Associate Professor, Paris Elite Institute of Technology, Shanghai Jiao Tong University, Shanghai, China.
[2]Professor, Sorbonne University, Laboratoire Jacques-Louis Lions, Paris, France.
[3]Bachelor student, Paris Elite Institute of Technology, Shanghai Jiao Tong University, Shanghai, China.

Contents

Part 1
Reduced Basis Method

I - Approximation of the solution manifold

1) Physic state

Let X be a Hilbert[4] of functions over a space-time domain $\Omega\times]0;T[$, with $\Omega \subset \mathbb{R}^d$ $(d \geq 1)$ open[5] space domain, endowed with the inner product $\langle\cdot|\cdot\rangle_X$ and its induced norm $\|\cdot\|_X$. Let $P \in \mathbb{N}^\star$, $\mathbb{P} \subset \mathbb{R}^P$ be a compact parameter space. Let $F : X \times \mathbb{P} \longmapsto \mathbb{R}$ a functional.
We consider the physical field parameterized by $\boldsymbol{\mu} \in \mathbb{P}$:

$$\text{Find } u = u(\cdot,\cdot\,;\boldsymbol{\mu}) \in X, \quad F(u,\boldsymbol{\mu}) = 0. \tag{1.1}$$

In what follows we suppose the physic system admits an unique solution $u(\cdot\,;\boldsymbol{\mu})$ for each $\boldsymbol{\mu} \in \mathbb{P}$.
Such physical system usually derives from a parameterized partial differential equation (PPDE):

Example 1: Parameterized partial differential equation [RAU91; HRS16]

Let $L_{\boldsymbol{\mu}} : X \to X'$ be a linear operator that includes the linear differential operators[a].
Let $\mathcal{G}_{\boldsymbol{\mu}} : X \mapsto X'$ be a nonlinear operator that includes the nonlinear differential operators[b].
A *Parameterized Partial Differential Equation* can be written :

$$\text{Find } u = u(\cdot,\cdot\,;\boldsymbol{\mu}) \in X, \quad \begin{cases} \dfrac{\partial u}{\partial t} + L_{\boldsymbol{\mu}}(u) + \mathcal{G}_{\boldsymbol{\mu}}(u) = 0, \\ \text{Initial conditions}, \\ \text{Boundary conditions}, \\ \text{Physical constraints}. \end{cases} \tag{1.2}$$

Here the PPDE is supposed to have (in the time-dependant case) only order one time derivative, a linear part $L_{\boldsymbol{\mu}}$ and a nonlinear part $\mathcal{G}_{\boldsymbol{\mu}}$.

[a] See [LAN96; KYT96] for definition.
[b] See [LE 13; TAY91; KYT96] for definition.

We define the manifold described in X by the solution of (1.1) when the parameter $\boldsymbol{\mu}$ describes $\mathbb{P}$:

Definition 1.1: Solution manifold

We define the *solution manifold* [RHE88] as:

$$\mathcal{M} := \{u(\cdot,\cdot\,;\boldsymbol{\mu}) \;/\; \boldsymbol{\mu} \in \mathbb{P}, \quad F(u,\boldsymbol{\mu}) = 0\} \subset X.$$

This is the quantity of interest that we want to approximate in what follow.

Such system of equations is hard to describe in a closed analytical form, thus the use of numerical methods to obtain an approximation of the solution. Some methods include Galerkin elements methods [EG04; DE12; FIL24; QSS06].

In practice, these methods discretize the space X (in space and time) into a linear space $\boldsymbol{X}_\delta$ of finite dimension denoted $n_d \geq 1$.

[4] See [BRE11] for functional analysis theory.
[5] See [MEN90] for topology theory.

In the time-dependent case, $\boldsymbol{u}_{n_d} \in \mathbb{R}^{n_d}$ will be isomorph to a matrix where the coefficient u_{ij} corresponds to the i-th spatial degree of freedom at the j-th timestep.

For the sake of simplicity we assume that we are working in a conforming Galerkin framework [AG21, Section 18 and 19], where $\boldsymbol{X}_\delta \subset X$ such that $F(\boldsymbol{u}_{n_d}, \boldsymbol{\mu})$ is well defined. For $\boldsymbol{\mu} \in \mathbb{P}$, the *truth problem* then reads :

$$\text{Find } \boldsymbol{u}_{n_d} = \boldsymbol{u}_{n_d}(\boldsymbol{\mu}) \in \boldsymbol{X}_\delta, \quad F(\boldsymbol{u}_{n_d}, \boldsymbol{\mu}) = 0. \tag{1.3}$$

We call *truth solution* a solution $\boldsymbol{u}_{n_d}$ of the truth problem. From the truth problem, one can construct the truth solution manifold :

Definition 1.2: Truth solution manifold

We define the *truth solution manifold* as

$$\mathcal{M}_\delta := \{\boldsymbol{u}_{n_d}(\boldsymbol{\mu}) \ / \ \boldsymbol{\mu} \in \mathbb{P}, \ F(\boldsymbol{u}_{n_d}, \boldsymbol{\mu}) = 0\} \subset \boldsymbol{X}_\delta.$$

For $\delta > 0$ and for n_d big enough[a] it is expected that :

$$\forall \boldsymbol{\mu} \in \mathbb{P}, \ \|\boldsymbol{u}_{n_d}(\boldsymbol{\mu}) - u(\boldsymbol{\mu})\|_X \leq \delta. \tag{1.4}$$

[a] Strongly depends on the numerical method, e.g. for order 1 Lagrange finite elements $n_d \propto \frac{1}{\delta}$, [EG04; QSS06].

Remark :

The issue with the truth problems is the computational complexity, e.g. for finite elements it usually scales like $(n_d)^d$ [HRS16, Section 3.5].[a] In particular, this not tractable for high precision simulations or high dimensional PDEs.

[a] Recall that $d \leq 1$ is the spatial dimension and n_d is the discrete space-time dimension.

In the case of conforming Galerkin finite element methods, Cea's lemma [CEA64] give an a priori error control :

$$\exists C > 0, \ \forall \boldsymbol{\mu} \in \mathbb{P}, \qquad \|u(\boldsymbol{\mu}) - \boldsymbol{u}_{n_d}(\boldsymbol{\mu})\|_X \leq C \inf_{\boldsymbol{v}_{n_d} \in \boldsymbol{X}_\delta} \|u(\boldsymbol{\mu}) - \boldsymbol{v}_{n_d}\|_X. \tag{1.5}$$

The approximation is said to be *quasi-optimal* if it satisfies an estimation equivalent to Eq.(1.5). In what follow, we suppose we have such property of quasi-optimality of the truth problem.

2) Approximation by a reduced basis

To answer the computational issue we consider a method with reduced linear dimension. We look for a reference function $\boldsymbol{u}_{\text{ref}} \in \boldsymbol{X}_\delta$ and a reduced basis $\mathcal{B}_N = \left[\boldsymbol{\zeta}^1, \dots, \boldsymbol{\zeta}^N\right] \in (\mathcal{M}_\delta - \boldsymbol{u}_{\text{ref}})^N$, $N \geq 1$ such that

$$\boldsymbol{V}_N := \text{Span}\left\{(\boldsymbol{\zeta}^i)_{1 \leq i \leq N}\right\} + \boldsymbol{u}_{\text{ref}} \subset \mathcal{M}_\delta$$

is a reduced space of dimension N that aims to represent $\mathcal{M}_\delta$ with high fidelity and dimension $N \ll n_d$.

We call the linear space $\boldsymbol{V}_N$ a *Reduced Basis* (RB) of the PPDE. Its goal is to approximate $\mathcal{M}_\delta$ as well as possible while having the smallest dimension allowing it.
Such method of constructing a RB to approximate a infinite-dimensional solution manifold is the so called *Reduced Basis Method* [OS16].
Searching an approximation of the physical field $u \in X$ by $\boldsymbol{u}_N \in \boldsymbol{V}_N$ then read as :

$$\text{For } \boldsymbol{\mu} \in \mathbb{P}, \ \text{find } \boldsymbol{u}_N(\boldsymbol{\mu}) = \boldsymbol{u}_{\text{ref}} + \sum_{i=1}^{N} \hat{u}_i(\boldsymbol{\mu}) \boldsymbol{\zeta}^i \in \boldsymbol{V}_N, \ F(\boldsymbol{u}_N, \boldsymbol{\mu}) = 0. \tag{1.6}$$

The formulation of the equation is the same as before yet the computational cost of the reduced basis is of the order $\mathcal{O}(N^d)$ which can be several amplitudes lesser than $\mathcal{O}(\ (n_d)^d\ )$.

Since the reduced basis is a linear subset of $\boldsymbol{X}_\delta$, we recover quasi-optimality [HRS16, Eq.(3.7)] :

$$\exists C>0,\ \forall \boldsymbol{\mu}\in\mathbb{P},\qquad \|u(\boldsymbol{\mu})-\boldsymbol{u}_N(\boldsymbol{\mu})\|_X \leq C \inf_{\boldsymbol{v}_N\in\boldsymbol{V}_N} \|u(\boldsymbol{\mu})-\boldsymbol{v}_N\|_X. \tag{1.7}$$

To compare $\boldsymbol{V}_N$ with $\mathcal{M}$ we consider their Hausdorff distance[6,7] in the X-norm :

$$E(\mathcal{M},V_N\ ;\|\cdot\|_X) := \sup_{u\in\mathcal{M}} \inf_{\boldsymbol{v}_N\in V_N} \|u-\boldsymbol{v}_N\|_X = \sup_{\boldsymbol{\mu}\in\mathbb{P}} \min_{\boldsymbol{v}_N\in V_N} \|u(\boldsymbol{\mu})-\boldsymbol{v}_N\|_X. \tag{1.8}$$

The $\|\cdot\|_X$ can be omitted if there's no ambiguity on the considered norm. The inf is attained because X is a Hilbert [BRE11, Theorem 5.2] and $\boldsymbol{V}_N$ a finite dimensional linear subset.
We then focus on the optimal distance one can expect with such a approximation of $\mathcal{M}$.

Definition 1.3: Kolmogorov N-width [KOL36]

The *Kolmogorov N-width* measures the optimal Hausdorff distance between the approximation of a manifold $\mathcal{M}$ and a N-dimensional linear space included in $\mathcal{M}$:

$$d_N(\mathcal{M}\ ;\|\cdot\|_X) := \inf_{\substack{\boldsymbol{V}_N\subset\mathcal{M}\\ \dim\ (\boldsymbol{V}_N)=N}} E(\mathcal{M},\boldsymbol{V}_N\ ;\|\cdot\|_X).$$

The $\|\cdot\|_X$ can be omitted if there's no ambiguity on the considered norm.

Because of the orthogonality property Eq.(1.8) and denoting by $\Pi_{\boldsymbol{V}_n}$ the orthogonal projection from X into $\boldsymbol{V}_N$ we can rewrite this quantity as :

$$d_N(\mathcal{M}) = \inf_{\substack{\boldsymbol{V}_N\subset\mathcal{M}\\ \dim\ (\boldsymbol{V}_N)=N}} \|u-\Pi_{\boldsymbol{V}_n}(u)\|_{\mathrm{L}^\infty(\mathbb{P},\boldsymbol{V}_N)}. \tag{1.9}$$

Assuming the numerical solving method of the physical field is as precise as we want, we can write :

$$\forall\boldsymbol{\mu}\in\mathbb{P},\ \|u(\boldsymbol{\mu})-\boldsymbol{u}_{n_d}(\boldsymbol{\mu})\|_X\leq\delta \implies |d_N(\mathcal{M}_\delta)-d_N(\mathcal{M})|\leq\delta. \tag{1.10}$$

It is equivalent to estimate Kolmogorov N-width with either the truth solutions or the continuous ones. Thanks to Eq.(1.7) we get the following a priori control :

$$\exists C>0,\ \forall\boldsymbol{\mu}\in\mathbb{P},\ \|u(\boldsymbol{\mu})-\boldsymbol{u}_N(\boldsymbol{\mu})\|_X\leq C d_N(\mathcal{M}). \tag{1.11}$$

In that sense, equivalence holds between the error of the reduced basis representation of the analytical solution and the optimal distance between the solution manifold and a reduced basis model.

PDE	Equation	Kolmogorov N-width decay	Reference
Diffusion/Heat equation	$\partial_t u+\partial_x(\mu\partial_x u)=f$	$d_N(\mathcal{M})\lesssim\exp(-\alpha N)$	[MPT02]
Linear advection	$\partial_t u+\mu\partial_x u=f$	$d_N(\mathcal{M})\gtrsim N^{-1/2}$	[OS16]
Wave equation	$\partial_t^2 u-\mu^2\partial_x^2 u=f$	$d_N(\mathcal{M})\gtrsim N^{-1/2}$	[GU19]
Stokes equation	$-\nu\,\Delta\boldsymbol{u}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}+\nabla p = f$ $\mathrm{div}(\boldsymbol{u}) = 0$	$d_N(\mathcal{M})\lesssim\exp(-N/3)$	[MS20]

Table 1: Kolmogorov N-width of some partial differential equations.

These estimations are only true for some specific situations since there the Kolmogorov N-width decay depends on the data of the PDE and the dimension of the parameter space [AGU23].

[6] [SEN90], note this is the full Hausdorff distance since $\boldsymbol{V}_N\subset\mathcal{M}_\delta$.
[7] Also called the *angle*, *deviation* or *discrepancy* between $\mathcal{M}$ and $\boldsymbol{V}_N$ [QMF16, Section 5.4].

II - Generation of the Reduced Basis by Proper Orthogonal Decomposition

A N-dimensional Proper Orthogonal Decomposition (POD) [LIA+02] space of X is a N-dimensional linear subset $\boldsymbol{V}_N \subset X$ that minimizes their distance in the least-square sense [LIA+02; BUL12; HRS16].

Let $N_s \geq 1$. Let $\mathbb{P}_h = \{\boldsymbol{\mu}^1, \ \dots \ , \boldsymbol{\mu}^{N_s}\} \subset \mathbb{P}$ be a discrete and finite point-set. Define a set of snapshots of the truth solutions of cardinality N approximating the truth manifold as :

$$\mathcal{M}_\delta(\mathbb{P}_h) := \{\boldsymbol{u}_{n_d}(\boldsymbol{\mu}) \ / \ \boldsymbol{\mu} \in \mathbb{P}_h\}.$$

Define the centered snapshots matrix $\boldsymbol{S} = \begin{bmatrix} \boldsymbol{u}_{n_d}(\boldsymbol{\mu}^1) & \dots & \boldsymbol{u}_{n_d}(\boldsymbol{\mu}^{N_s}) \end{bmatrix} \in \mathbb{R}^{n_d \times N_s}$. To construct the reduced basis $\boldsymbol{V}_N$ we apply a Singular Vector Decomposition (SVD) [LIA+02; BUL12] to $\boldsymbol{S}$.
Let $r = \text{rank}(\boldsymbol{S}) \leq \min(n_d, N)$. There exists $\boldsymbol{V} \in \mathbb{R}^{n_d \times r}$, $\boldsymbol{D} = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$ and $\boldsymbol{W} \in \mathbb{R}^{N_s \times r}$ such that [BHK20] :

$$\boldsymbol{S} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{W}^T, \quad \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_r, \quad \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_r. \tag{1.12}$$

The diagonal values of $\boldsymbol{D}$, $\sigma_i > 0$, are such that σ_i^2 is an eigenvalue of $\boldsymbol{S}^T\boldsymbol{S}$ and :

$$\sigma_1 \geq \dots \geq \sigma_r > 0. \tag{1.13}$$

Remark :

Some SVD [LIA+02; BUL12; QMF16] considers $\boldsymbol{U} \in \mathbb{R}^{n_d \times n_d}$, $\boldsymbol{D} \in \mathbb{R}^{n_d \times N_s}$ and $\boldsymbol{W} \in \mathbb{R}^{N_s \times N_s}$ and $p = \min(N_s, n_d)$ such that :

$$\sigma_1 \geq \dots \geq \sigma_r > 0 = \sigma_{r+1} = \dots = \sigma_p.$$

This holds no difference since the $p - r$ last columns of $\boldsymbol{V}$ are associated to $\ker(\boldsymbol{S}^T\boldsymbol{S})$.

The best N-rank approximation of $\boldsymbol{S}$ in the least square sense is [BHK20, Section 3.5] :

$$\boldsymbol{S}_N = \boldsymbol{V}_N\boldsymbol{D}_N\boldsymbol{W}_N^T. \tag{1.14}$$

where $\boldsymbol{V}_N \in \mathbb{R}^{n_d \times N}$ is obtained by the first N columns of $\boldsymbol{Q}$, $\boldsymbol{D}_N = \text{diag}(\sigma_1, \dots, \sigma_n) \in \mathbb{R}^{N \times N}$ and $\boldsymbol{W}_N \in \mathbb{R}^{N \times N}$ is obtained by the first N columns of $\boldsymbol{W}$.

Theorem 1.4: SVD error estimation

We have the following operator least square error [BHK20, Lemma 3.8 and Theorem 3.9] :

$$\|\boldsymbol{S} - \boldsymbol{S}_N\|_2^2 = \min_{\substack{\boldsymbol{A} \in \mathbb{R}^{n_d \times N} \\ \text{rank}(A) = N}} \|\boldsymbol{S} - \boldsymbol{A}\|_2^2 = \sigma_{N+1}^2.^a$$

Schmidt-Eckart-Young theorem [QMF16, Theorem 6.1] gives the Frobenius least-square error :

$$\|\boldsymbol{S} - \boldsymbol{S}_N\|_F^2 = \min_{\substack{\boldsymbol{A} \in \mathbb{R}^{n_d \times N} \\ \text{rank}(A) = N}} \|\boldsymbol{S} - \boldsymbol{A}\|_F^2 = \sum_{k=N+1}^{r} \sigma_k^2.^b$$

[a] Spectral norm on linear operators of its argument [HJ12, Example 5.6.6].
[b] Frobenius norm on matrices of its argument [HJ12, Eq.(5.6.0.2)].

Remark :

> The notation with $r = \text{rank}(\boldsymbol{S})$ can be replaced by $p = \min(n_D, N_s)$ since p is the maximum rank that $\boldsymbol{S} \in \mathbb{R}^{n_d \times N_s}$ can admit and defining $\sigma_i = 0,\ \forall p \geq i > r$.
> Hence both notations for the SVD are equivalents, and the dimension of the SVD **must be** lower than r or $\boldsymbol{S}_N = \boldsymbol{S}$.

The SVD orthogonal basis is constructed by taking the N first left singular vectors of $\boldsymbol{S}$ i.e. the N first columns of $\boldsymbol{Q}$, that is $\boldsymbol{V}_N$.
We recall that the projection operator from $\mathbb{R}^{n_d}$ to $\text{Span}\{\boldsymbol{V}_N\}$ is :

$$\Pi_{\boldsymbol{V}_N} : \boldsymbol{x} \in \mathbb{R}^{n_d} \longmapsto \boldsymbol{V}_N \boldsymbol{V}_N^T x. \tag{1.15}$$

We have the following optimal property [QMF16, Proposition 6.1] :

$$\sum_{i=1}^{N} \left\| \boldsymbol{u}_{n_d}(\boldsymbol{\mu}^i) - \boldsymbol{V}_N \boldsymbol{V}_N^T \boldsymbol{u}_{n_d}(\boldsymbol{\mu}^i) \right\|_2^2 = \min_{\substack{\boldsymbol{A} \in \mathbb{R}^{n_d \times N} \\ \boldsymbol{A}^T \boldsymbol{A} = \boldsymbol{I}_N}} \sum_{i=1}^{N} \left\| \boldsymbol{u}_{n_d}(\boldsymbol{\mu}^i) - \boldsymbol{A}\boldsymbol{A}^T \boldsymbol{u}_{n_d}(\boldsymbol{\mu}^i) \right\|_2^2 \tag{1.16}$$

$$= \sum_{i=N+1}^{r} \sigma_i^2. \tag{1.17}$$

where here $\|\cdot\|_2$ refers to the euclidean 2-norm on $\mathbb{R}^{n_d}$ [HJ12, Eq.(5.2.1)].
Thus $\boldsymbol{V}_N$ columns are the N-optimal orthogonal vectors in the sense of the l^2-projection [COL12, Section 6.2] from any snapshot taken in $\boldsymbol{S}$ to $\text{Span}(\boldsymbol{V}_n)$.

To get an optimal distance over the X-norm with elements of $\boldsymbol{X}_\delta$, we consider a symmetric definite positive matrix $\boldsymbol{M}_\delta$ such that for $\boldsymbol{u}_\delta, \boldsymbol{v}_\delta \in \boldsymbol{X}_\delta$, it holds $\langle \boldsymbol{u}_\delta | \boldsymbol{v}_\delta \rangle_X = \boldsymbol{u}_\delta^T \boldsymbol{M}_\delta \boldsymbol{v}_\delta$ and $\|\boldsymbol{v}_\delta\|_X^2 = \boldsymbol{v}_\delta^T \boldsymbol{M}_\delta \boldsymbol{v}_\delta$.
In such framework, the orthogonal projections into a space spanned by a matrix $\boldsymbol{V}$ becomes :

$$\Pi_{\boldsymbol{V}_N}(\boldsymbol{x}) = \boldsymbol{V}_N \boldsymbol{V}_N^T \boldsymbol{M}_\delta \boldsymbol{x}. \tag{1.18}$$

Then by [QMF16, Proposition 6.2] and [HRS16, Section 3.2.1] :

Theorem 1.5: POD basis in the X-norm

Consider $\tilde{\boldsymbol{S}} = \boldsymbol{M}_\delta^{1/2} \boldsymbol{S}$, $\tilde{\boldsymbol{S}} = \tilde{\boldsymbol{V}} \tilde{\boldsymbol{D}} \tilde{\boldsymbol{W}}$ its full rank SVD and $\tilde{\boldsymbol{S}_N} = \tilde{\boldsymbol{V}_N} \tilde{\boldsymbol{D}_N} \tilde{\boldsymbol{W}_N}$ its N-dimensional reduced SVD.
Let $\boldsymbol{V}_N = \boldsymbol{M}_\delta^{1/2} \tilde{\boldsymbol{V}_N}$.
Then all the precedent results hold in the norm $\|\cdot\|_X$ (or $\|\cdot\|_{\boldsymbol{X}_\delta}$), in particular :

$$\sum_{i=1}^{N} \left\| \boldsymbol{u}_{n_d}(\boldsymbol{\mu}^i) - \boldsymbol{V}_N \boldsymbol{V}_N^T \boldsymbol{M}_\delta \boldsymbol{u}_{n_d}(\boldsymbol{\mu}^i) \right\|_X^2 = \min_{\substack{\boldsymbol{A} \in \mathbb{R}^{n_d \times N} \\ \boldsymbol{A}^T \boldsymbol{M}_\delta \boldsymbol{A} = \boldsymbol{I}_N}} \sum_{i=1}^{N} \left\| \boldsymbol{u}_{n_d}(\boldsymbol{\mu}^i) - \boldsymbol{A}\boldsymbol{A}^T \boldsymbol{M}_\delta \boldsymbol{u}_{n_d}(\boldsymbol{\mu}^i) \right\|_X^2$$

$$= \sum_{i=N+1}^{r} \sigma_i^2.$$

Since we are interested in a POD that approximates entirely $\mathcal{M}_\delta$ and not only a discrete subset $\mathcal{M}_\delta(\mathbb{P}_h)$, we consider the minimization problem [QMF16, Remark 6.3] and [HRS16, Eq.(3.6)] :

$$\delta_{N,2}(\mathcal{M}_\delta) = \min_{\substack{\boldsymbol{A} \in \mathbb{R}^{n_d \times N} \\ \boldsymbol{A}^T \boldsymbol{M}_\delta \boldsymbol{A} = \boldsymbol{I}_N}} \left(\int_{\mathbb{P}} \left\| \boldsymbol{u}_{n_d}(\boldsymbol{\mu}) - \boldsymbol{A}\boldsymbol{A}^T \boldsymbol{M}_\delta \boldsymbol{u}_{n_d}(\boldsymbol{\mu}) \right\|_X^2 \, d\boldsymbol{\mu} \right)^{1/2}. \tag{1.19}$$

The minimum is attained [QMF16, Proposition 6.3] and Eq.1.19 is analogous to the Kolmogorov N-width for the $\mathrm{L}^2(\mathbb{P}, \boldsymbol{X}_\delta)$ norm induced with $\|\cdot\|_X$ [QMF16, Remark 6.3] :

$$\delta_{N,2}(\mathcal{M}_\delta) = \min_{\substack{\boldsymbol{A}\in\mathbb{R}^{n_d\times N}\\ \boldsymbol{A}^T\boldsymbol{M}_\delta\boldsymbol{A}=\boldsymbol{I}_N}} \left\|\boldsymbol{u}_{n_d} - \boldsymbol{A}\boldsymbol{A}^T\boldsymbol{M}_\delta\boldsymbol{u}_{n_d}\right\|_{(\mathrm{L}^2(\mathbb{P},\boldsymbol{X}_\delta);\ \|\cdot\|_X)} \leq |\mathbb{P}|^{1/2}\, d_N(\mathcal{M}_\delta).$$

We need to bridge between the parameter-continuous framework and the parameter-discrete one. This is done [QMF16, Section 6.5].
Using quadrature formula we can write [QSS06, Section 9.1] :

$$\int_{\mathbb{P}} \left\|\boldsymbol{u}_{n_d}(\boldsymbol{\mu}) - \boldsymbol{V}_N\boldsymbol{V}_N^T\boldsymbol{M}_\delta\boldsymbol{u}_{n_d}(\boldsymbol{\mu})\right\|_X^2 d\boldsymbol{\mu} \simeq \sum_{i=1}^{N} w^i \left\|\boldsymbol{u}_{n_d}\left(\boldsymbol{\mu}^i\right) - \boldsymbol{V}_N\boldsymbol{V}_N^T\boldsymbol{M}_\delta\boldsymbol{u}_{n_d}\left(\boldsymbol{\mu}^i\right)\right\|_X^2. \tag{1.20}$$

where $w^i > 0$ are suitable quadrature weights and $\boldsymbol{\mu}^i \in \mathbb{P}$ are quadrature points in $\mathbb{P}$.
This approximation holds an error that we call *sampling error* :

$$\varepsilon^2_{\substack{\text{sampling}\\ \text{error}}} := \left| \sum_{i=1}^{N} \left\|\boldsymbol{Q}^{1/2}\boldsymbol{u}_{n_d}\left(\boldsymbol{\mu}^i\right) - \boldsymbol{Q}^{1/2}\boldsymbol{V}_N\boldsymbol{V}_N^T\boldsymbol{M}_\delta\boldsymbol{u}_{n_d}\left(\boldsymbol{\mu}^i\right)\right\|_X^2 - \int_{\mathbb{P}} \left\|\boldsymbol{u}_{n_d}(\boldsymbol{\mu}) - \boldsymbol{V}_N\boldsymbol{V}_N^T\boldsymbol{M}_\delta\boldsymbol{u}_{n_d}(\boldsymbol{\mu})\right\|_X^2 d\boldsymbol{\mu} \right|.$$

The choice of the quadrature points depends on the dimension of the parameters space and define the sampling $\mathbb{P}_h = \left(\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^{N_s}\right)$ to construct $\mathcal{M}_\delta(\mathbb{P}_h)$.
We denote by $\boldsymbol{Q} = \mathrm{diag}(w_1, \ldots, w_N) \in \mathbb{R}^{N_s\times N_s}$ the diagonal weight matrix and define an SVD on $\tilde{\boldsymbol{S}} = \boldsymbol{M}_\delta^{1/2}\boldsymbol{S}\boldsymbol{Q}^{1/2}$, then the N first singular vectors $\boldsymbol{V}_N$ are such that [QMF16, Algorithm 6.3] :

$$\begin{aligned}
&\sum_{i=1}^{N} w_i \left\|\boldsymbol{u}_{n_d}\left(\boldsymbol{\mu}^i\right) - \boldsymbol{V}_N\boldsymbol{V}_N^T\boldsymbol{M}_\delta\boldsymbol{u}_{n_d}\left(\boldsymbol{\mu}^i\right)\right\|_X^2 \\
= &\min_{\substack{\boldsymbol{A}\in\mathbb{R}^{n_d\times N}\\ \boldsymbol{A}\boldsymbol{M}_\delta\boldsymbol{A}^T=\boldsymbol{I}_N}} \left\|\tilde{\boldsymbol{S}} - \boldsymbol{A}\boldsymbol{A}^T\tilde{\boldsymbol{S}}\right\|_F^2 \\
= &\sum_{i=N+1}^{r} \sigma_i^2.
\end{aligned}$$

Note the $\sigma_i > 0$ are the eigenvalues for the considered sampling. The approximation of $\mathcal{M}_\delta$ by a weighted POD is [QMF16, Eq.(6.35)] :

$$C d_n(\mathcal{M}_\delta) \geq \delta_{N,2}(\mathcal{M}_\delta) \sim \varepsilon^2_{\text{sampling error}} + \underbrace{\sum_{i=N+1}^{r} \sigma_i^2}_{\text{POD error}} + \varepsilon^2_{\text{numerical error}} \geq d_N(\mathcal{M}_\delta). \tag{1.21}$$

For a parameter $\boldsymbol{\mu} \in \mathbb{P}$, we expect the error between the reduced solution $\boldsymbol{u}_N(\boldsymbol{\mu})$ and the truth solution $\boldsymbol{u}_{n_d}(\boldsymbol{\mu})$ to be equivalent to $d_N(\mathcal{M}_\delta)$ up to the sampling error $\varepsilon_{\text{sampling error}}$ and the numerical error $\varepsilon_{\text{numerical error}} \sim 10^{-12}$ [QSS06, Section 2.4].

Remark :

Let $N \geq 1$. Let $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ be symmetric positive definite (SPD) matrix. Its norm induced by $\langle \boldsymbol{M}\boldsymbol{u}|\boldsymbol{u}\rangle = \|u\|_{\boldsymbol{M}}^2$, $u \in \mathbb{R}^N$, is such that if $\lambda_1 = \max \mathrm{Sp}(\boldsymbol{M})$ and $\lambda_n = \min \mathrm{Sp}(\boldsymbol{M})$:

$$\lambda_n \|\boldsymbol{u}\|_2^2 \leq \|u\|_{\boldsymbol{M}}^2 \leq \lambda_1 \|\boldsymbol{u}\|_2^2.$$

Thus a POD on the 2-norm is equivalent in the sense of the norms to a POD on the X-norm if the matrix $\boldsymbol{M}_\delta$ is SPD.
Hence it is enough to do a sampled POD with the 2-norm in that case.

We conclude by the following statement that summarizes the method :

Theorem 1.6: Weighted POD in energy norm

Let $N_s \geq 1$, $(w_1, \dots, w_N) \in (\mathbb{R}_+^*)^{N_s}$ be quadrature weights and $(\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N) \in \mathbb{P}^{N_s}$ be quadrature points. Define $\boldsymbol{Q} = \mathrm{diag}(w_1, \dots, w_N) \in \mathbb{R}^{N_s \times N_s}$.

Consider the matrix of truth solutions snapshots $\boldsymbol{S} = \begin{bmatrix} \boldsymbol{u}_N(\boldsymbol{\mu}^1) & \dots & \boldsymbol{u}_N(\boldsymbol{\mu}^{N_s}) \end{bmatrix} \in \mathbb{R}^{n_d \times N}$.
Let $\boldsymbol{M}_\delta \in \mathbb{R}^{n_d \times n_d}$ be the SPD matrix that defines $\boldsymbol{X}_\delta$ inner-product.
Let $\tilde{\boldsymbol{S}} = \boldsymbol{M}_\delta^{1/2} \boldsymbol{S} \boldsymbol{Q}^{1/2}$.
Then the rank N matrix $\boldsymbol{V}_N$ defined by the N first left singular vectors of $\tilde{\boldsymbol{S}}^T \tilde{\boldsymbol{S}}$ form a POD basis such that :

$$\sum_{i=1}^{N} w_i \left\| \boldsymbol{u}_{n_d}(\boldsymbol{\mu}^i) - \boldsymbol{V}_N \boldsymbol{V}_N^T \boldsymbol{M}_\delta \boldsymbol{u}_{n_d}(\boldsymbol{\mu}^i) \right\|_X^2$$

$$= \min_{\substack{\boldsymbol{A} \in \mathbb{R}^{n_d \times N} \\ \boldsymbol{A} \boldsymbol{M}_\delta \boldsymbol{A}^T = \boldsymbol{I}_N}} \left\| \tilde{\boldsymbol{S}} - \boldsymbol{A} \boldsymbol{A}^T \tilde{\boldsymbol{S}} \right\|_F^2 = \sum_{i=N+1}^{r} \sigma_i^2.$$

By Eq.(1.21) we note that the decrease rate of this method is directly linked to the Kolmogorov N-width, and if the latter has a very slow decay rate then a large number of mode is needed which also results in a large number of snapshots needed. However large number of modes can make POD-based ROMs slower than full order models.

III - Error modeling

1) Model error

Let $\boldsymbol{V}_N$ be a reduced space generated by POD. Recalling the form of the reduced solution $\boldsymbol{u}_N \in \boldsymbol{V}_N$:

$$\boldsymbol{u}_N = \boldsymbol{u}_{\mathrm{ref}} + \sum_{k=1}^{N} \hat{u}_k(\boldsymbol{\mu}) \boldsymbol{\zeta}^k = \boldsymbol{u}_{\mathrm{ref}} + \boldsymbol{V}_N \hat{\boldsymbol{u}}. \tag{1.6}$$

If $\boldsymbol{u}_{n_d} \in \boldsymbol{X}_\delta$, then its approximation is :

$$\boldsymbol{u}_{n_d} \simeq \boldsymbol{u}_{\mathrm{ref}} + \boldsymbol{V}_N \hat{\boldsymbol{u}}.$$

We define the *model error* (or *bias*) of the ROM as

$$\boldsymbol{\eta} = \boldsymbol{u}_{n_d} - \boldsymbol{u}_N \in \boldsymbol{V}_N^\perp, \tag{1.22}$$

such that

$$\boldsymbol{u}_{n_d}(\boldsymbol{\mu}) = \boldsymbol{u}_N(\boldsymbol{\mu}) + \boldsymbol{\eta}(\boldsymbol{\mu}) = \boldsymbol{u}_{\mathrm{ref}} + \boldsymbol{V}_N \hat{\boldsymbol{u}}(\boldsymbol{\mu}) + \boldsymbol{\eta}(\boldsymbol{\mu}). \tag{1.23}$$

The goal of what follows is to find a way to estimate the error $\boldsymbol{\eta}$. Toward that goal, we'll make use of the Talyor expansion [AGH+26].

Let $\boldsymbol{\mu} \in \mathbb{P}$, let $u(\boldsymbol{\mu}) \in X$ be the solution of Eq.(1.1). Assuming the map $\boldsymbol{\mu} \in \mathbb{P} \longmapsto u(\cdot, \cdot\,; \boldsymbol{\mu})$ is smooth enough, we can apply a 1^{st} order Taylor expansion with respect to $\boldsymbol{\mu}$:

$$u(\boldsymbol{\mu}) \underset{\|\boldsymbol{\mu} - \boldsymbol{\mu}_0\|_{\mathbb{R}^P} \to 0}{=} u(\boldsymbol{\mu}_0) + \nabla_{\boldsymbol{\mu}}(u)(\boldsymbol{\mu}_0) \cdot (\boldsymbol{\mu} - \boldsymbol{\mu}_0) + o(\|\boldsymbol{\mu} - \boldsymbol{\mu}_0\|_{\mathbb{R}^P}). \tag{1.24}$$

where $\nabla_{\boldsymbol{\mu}}(\cdot)$ is the differentiation with respect to the parameter dependence.
We get the following analogy between the approximation Eq.(1.6) and Eq.(1.24) :

$$\begin{aligned} \boldsymbol{u}_{\text{ref}} &\equiv \boldsymbol{u}(\boldsymbol{\mu}_0), && (1.25)\\ \boldsymbol{V}_N\hat{\boldsymbol{u}}(\boldsymbol{\mu}) &\equiv \nabla_{\boldsymbol{\mu}}(u)(\boldsymbol{\mu}_0)\cdot(\boldsymbol{\mu}-\boldsymbol{\mu}_0), && (1.26)\\ \boldsymbol{\eta}(\boldsymbol{\mu}) &\equiv \underset{\|\boldsymbol{\mu}-\boldsymbol{\mu}_0\|_{\mathbb{R}^P}\to 0}{o}(\|\boldsymbol{\mu}-\boldsymbol{\mu}_0\|_{\mathbb{R}^P}). && (1.27)\end{aligned}$$

This is proven to be true locally [AGH+26]. Since $\nabla_{\boldsymbol{\mu}}(u)(\boldsymbol{\mu}_0)\in\mathcal{L}(\mathbb{R}^P,X)$, we expect a reduced basis approximating $u(\boldsymbol{\mu})$ for $\boldsymbol{\mu}\simeq\boldsymbol{\mu}_0$ to need $N\simeq P$ vectors to reconstruct, at least locally around $\boldsymbol{\mu}_0$, the truth solution.
Hence it is wise to consider $N\gtrsim P$ modes for a reduced basis to approximate $u(\boldsymbol{\mu})$ if $\boldsymbol{\mu}\simeq\boldsymbol{\mu}_0$.
It is possible to go further with Taylor's expansion to get a better intuition of what $\boldsymbol{\eta}$ can be :

$$u(\boldsymbol{\mu}) \underset{\|\boldsymbol{\mu}-\boldsymbol{\mu}_0\|_{\mathbb{R}^P}\to 0}{=} u(\boldsymbol{\mu}_0)+\nabla_{\boldsymbol{\mu}}(u)(\boldsymbol{\mu}_0)\cdot(\boldsymbol{\mu}-\boldsymbol{\mu}_0)+\frac{1}{2}\,\nabla^2_{\boldsymbol{\mu}}(u)(\boldsymbol{\mu}_0)(\boldsymbol{\mu}-\boldsymbol{\mu}_0)\otimes(\boldsymbol{\mu}-\boldsymbol{\mu}_0)+o(\|\boldsymbol{\mu}-\boldsymbol{\mu}_0\|^2_{\mathbb{R}^P}). \tag{1.28}$$

Thus, we can expect for $\boldsymbol{\mu}\simeq\boldsymbol{\mu}_0$:

$$\boldsymbol{\eta}(\boldsymbol{\mu})=\frac{1}{2}\,\nabla^2_{\boldsymbol{\mu}}(u)(\boldsymbol{\mu}_0)(\boldsymbol{\mu}-\boldsymbol{\mu}_0)\otimes(\boldsymbol{\mu}-\boldsymbol{\mu}_0)+\underset{\|\boldsymbol{\mu}-\boldsymbol{\mu}_0\|_{\mathbb{R}^P}\to 0}{o}(\|\boldsymbol{\mu}-\boldsymbol{\mu}_0\|^2_{\mathbb{R}^P})\simeq\frac{1}{2}\,\nabla^2_{\boldsymbol{\mu}}(u)(\boldsymbol{\mu}_0)(\boldsymbol{\mu}-\boldsymbol{\mu}_0)\otimes(\boldsymbol{\mu}-\boldsymbol{\mu}_0) \tag{1.29}$$

If $u\in X$ with X an Hilbert, then we can write for $(\boldsymbol{\zeta}^i)_i\in X^{\mathbb{N}}$:

$$u(\boldsymbol{\mu})=\sum_{i\in\mathbb{N}}\hat{u}_i(\boldsymbol{\mu})\zeta^i.$$

We can also write in the discrete case :

$$\boldsymbol{u}_{n_d}(\boldsymbol{\mu})=\boldsymbol{u}_{\text{ref}}+\sum_{1\le i\le n_d}\hat{u}_i(\boldsymbol{\mu})(\boldsymbol{\mu})\boldsymbol{\zeta}^i.$$

Thus we can write the equivalence :

$$\boldsymbol{\eta}(\mu)=\frac{1}{2}\,\nabla^2_{\boldsymbol{\mu}}(u)(\boldsymbol{\mu}_0)(\boldsymbol{\mu}-\boldsymbol{\mu}_0)\otimes(\boldsymbol{\mu}-\boldsymbol{\mu}_0)+\underset{\|\boldsymbol{\mu}-\boldsymbol{\mu}_0\|_{\mathbb{R}^P}\to 0}{o}(\|\boldsymbol{\mu}-\boldsymbol{\mu}_0\|^2_{\mathbb{R}^P})\equiv\sum_{i>P}\hat{u}_i(\boldsymbol{\mu})\boldsymbol{\zeta}^i. \tag{1.30}$$

We observe that the coefficients appearing in $\boldsymbol{\eta}(\boldsymbol{\mu})$ are mixed term of the P firsts coefficients. The equivalence with the reduced basis coefficients suggests the same dependence of the last coefficients from the first ones. This is proven in [AGH+26] :

Theorem 1.7: True manifold dimension of the coefficients space

For $\boldsymbol{\mu}_0\in\mathbb{P}\subset\mathbb{R}^P$, there exist $r>0$ such that for any $\mu\in\mathcal{B}_{\|\cdot\|_2}(\boldsymbol{\mu}_0,r)$, the coefficients $\hat{u}_i(\boldsymbol{\mu})$, $i>n$ are function of $\hat{u}_1(\boldsymbol{\mu}_0),\dots,\hat{u}_P(\boldsymbol{\mu}_0)$, i.e. for any $i\in\mathbb{N}$, there exists $h_i:\mathbb{R}^P\to\mathbb{R}$ such that $\hat{u}_i=h_i(\hat{u}_1,...,\hat{u}_P)$.

We consider the following conjecture :
There exists $n\simeq P$ such that the coefficients $\hat{u}_i$, $i>n_0$ are function of $\hat{u}_1,\dots,\hat{u}_{n_0}$, i.e. for any $N\in\mathbb{N}$, there exists $h_n:\mathbb{R}^{n_0}\to\mathbb{R}$ such that $\hat{u}_n=h_n(\hat{u}_1,...,\hat{u}_{n_0})$.

Some efforts have already been made in that direction, e.g. using a quadratic approximation [SP24; JAI+17] or using encoding-decoding framework [BNS25; ARE+26]. Following [ARE+26], our goal is to approximate h_n with Machine Learning methods.

2) Decomposition in linear and nonlinear basis

Choose $n \in \mathbb{N}^*$ and $\overline{n} \in \mathbb{N}^*$ such that $N = n + \overline{n}$ is big enough to have :

$$\|\boldsymbol{S} - \boldsymbol{S}_N\|_F \leq \epsilon,$$

where $\boldsymbol{S}$ and $\boldsymbol{S}_N$ are defined by Theorem 1.6. Recall that $\hat{\boldsymbol{u}} = \boldsymbol{V}_N^T u \in \mathbb{R}^N$ was defined as the N coefficients in the POD basis.
We define for $1 \leq i \leq n$ and $1 \leq j \leq \overline{n}$:

$$\alpha_i = \hat{u}_i, \quad \beta_j = \hat{u}_{j+N}.$$

The hypothesis of Theorem 1.7 suggests that there exists $n_0 \simeq P$ such that for any $n \geq n_0$ we can construct h which depends on n with

$$\boldsymbol{\beta} = h(\boldsymbol{\alpha}) \in R^{\overline{n}},$$

where

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \quad \text{and} \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{\overline{n}} \end{bmatrix}.$$

Suppose knowing only the first $n \simeq P$ coefficients $(\alpha_i)_{1 \leq i \leq n}$ is enough to reconstruct the following $\overline{n}$ coefficients $(\beta_i)_{1 \leq i \leq \overline{n}}$.
To highlights this separation of the coefficients, we consider two parts for the POD basis :

- The linear part $\boldsymbol{V} = \begin{bmatrix} \boldsymbol{\zeta}^1 & \dots & \boldsymbol{\zeta}^n \end{bmatrix}$ consists of the n first columns of $\boldsymbol{V}_N$.
- The nonlinear part $\overline{\boldsymbol{V}} = \begin{bmatrix} \boldsymbol{\zeta}^{n+1} & \dots & \boldsymbol{\zeta}^{\overline{n}} \end{bmatrix}$ consists of the columns $n+1$ to $N = \overline{n} + n$ of $\boldsymbol{V}_N$.

Thus if $\boldsymbol{u}_N - \boldsymbol{u}_{\text{ref}} \in \boldsymbol{V}_N =: [\boldsymbol{V}\ \overline{\boldsymbol{V}}]$;

$$\boldsymbol{u}_N = \boldsymbol{u}_{\text{ref}} + \boldsymbol{V}\boldsymbol{\alpha} + \overline{\boldsymbol{V}}\boldsymbol{\beta} = \boldsymbol{u}_{\text{ref}} + \underbrace{\boldsymbol{V}\boldsymbol{\alpha}}_{\text{linear coefficients}} + \underbrace{\overline{\boldsymbol{V}} h(\boldsymbol{\alpha})}_{\text{nonlinear in } \boldsymbol{\alpha}}.$$

From now on, $\boldsymbol{\alpha}$ will always denote the linear coefficients in front on the linear basis $\boldsymbol{V}$, and $\boldsymbol{\beta}$ will denote the coefficients that can be represented by a function $h(\boldsymbol{\alpha})$ in front of $\overline{\boldsymbol{V}}$.

The error between u_{n_d} and this u_N is as follow :

$$\begin{aligned}
\|\boldsymbol{u}_{n_d} - \boldsymbol{u}_N\|_X^2 &= \left\| \underbrace{\boldsymbol{u}_{n_d} - \boldsymbol{u}_{\text{POD}}}_{\in \boldsymbol{V}_N^\perp} + \underbrace{\boldsymbol{u}_{\text{POD}} - \boldsymbol{u}_N}_{\in \boldsymbol{V}_N} \right\|_X^2 \\
&= \|\boldsymbol{u}_{n_d} - \boldsymbol{u}_{\text{POD}}\|_X^2 + \left\| \boldsymbol{V}\boldsymbol{\alpha} + \overline{\boldsymbol{V}}\boldsymbol{\beta} - \boldsymbol{V}\boldsymbol{\alpha} - \overline{\boldsymbol{V}} h(\boldsymbol{\alpha}) \right\|_X^2 \\
&= \|\boldsymbol{\eta}_{\text{POD}}\|_X^2 + \left\| \overline{\boldsymbol{V}} \left(\boldsymbol{\beta} - h(\boldsymbol{\alpha}) \right) \right\|_X^2 \\
&= \|\boldsymbol{\eta}_{\text{POD}}\|_X^2 + \|\boldsymbol{\beta} - h(\boldsymbol{\alpha})\|_2^2 .
\end{aligned}$$

Thus the learning error will be controlled by the model error of the POD $\boldsymbol{\eta}_{\text{POD}}$ and the learning error between $h(\boldsymbol{\alpha})$ and $\boldsymbol{\beta}$:

$$\|\boldsymbol{u}_{n_d} - \boldsymbol{u}_N\|_X^2 = \|\boldsymbol{\eta}_{\text{POD}}\|_X^2 + \|\boldsymbol{\beta} - h(\boldsymbol{\alpha})\|_2^2 . \tag{1.31}$$

Part 2
Nonlinear compressed reduced basis method for EIM

I - Sensor-observed system

1) Observation function

We actually do not have access to the continuous solution $u \in X$ but only of observations of the system. We set $m \in \mathbb{N}^*$ the number of sensors used to observe the system and assume that we directly observe u° ; i.e. there exists a function

$$H : u \in X \to \mathbb{R}^m,$$

called *observation function* such that any observation $\boldsymbol{y}^\circ \in \mathbb{R}^m$ of the system can be described as

$$\boldsymbol{y}^\circ = H(u^\circ)$$[8]

for a certain $u^\circ \in \mathcal{M}$. In what follow, we suppose $H|_{\mathbb{R}^{n_d}} = \boldsymbol{P}$ is well-defined and linear.
In our case, we consider that H is a linear pointwise observation such that for $\boldsymbol{u} \in \mathbb{R}^{n_d}$:

$$H(\boldsymbol{u}) := \boldsymbol{P}\boldsymbol{u} := \begin{bmatrix} u(\boldsymbol{x}_{i_1}) \\ \vdots \\ u(\boldsymbol{x}_{i_m}) \end{bmatrix} = \begin{bmatrix} u_{i_1} \\ \vdots \\ u_{i_m} \end{bmatrix}.$$

Where for $x_0 \leq \boldsymbol{x}_{i_1} \leq ... \leq \boldsymbol{x}_{i_m} \leq \boldsymbol{x}_{n_d}$, $\boldsymbol{x}_{i_j}$ is the location of the $j-th$ sensor in the discretized space.
$\boldsymbol{P} : \mathbb{R}^{n_d} \to \mathbb{R}^m$ is a $m \times n_d$ matrix that selects the coefficients $(i_k)_{1 \leq k \leq m}$ of a vector in $\mathbb{R}^{n_d}$.
To account for measurements errors, we define the noisy observation

$$\boldsymbol{y}^\circ_\varepsilon := \boldsymbol{y}^\circ + \boldsymbol{e} \in \mathbb{R}^m. \tag{2.1}$$

The error is supposed to be **relative error** modeled as a gaussian noise of scale $\sigma > 0$ that we assume to be centered and independent between each sensors. At sensor location i, the error is

$$\boldsymbol{e}_i = (\boldsymbol{y}^\circ)_i \cdot \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2).$$

Thus the noisy observation at location i is :

$$(\boldsymbol{y}^\circ_\varepsilon)_i = (\boldsymbol{y}^\circ)_i + (\boldsymbol{y}^\circ)_i \cdot \varepsilon_i = (\boldsymbol{P}\boldsymbol{u}^\circ)_i + \boldsymbol{e}_i. \tag{2.2}$$

We can define the error vector as a gaussian vector :

$$\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}_{R^m}, \sigma^2 \boldsymbol{I}_m),$$

This can be rewritten with the Hadamard product :

$$\boldsymbol{y}^\circ_\varepsilon = \boldsymbol{y}^\circ + \boldsymbol{y}^\circ \odot \boldsymbol{\varepsilon} = \boldsymbol{P}\boldsymbol{u}^\circ + \boldsymbol{e}. \tag{2.3}$$

where

$$\boldsymbol{e} := \boldsymbol{P}\boldsymbol{u}^\circ \odot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}_{R^m}, \sigma^2 \boldsymbol{I}_m).$$

From that definition we have the relative error :

$$\mathbb{E}\left[\frac{\|\boldsymbol{e}\|_2^2}{\|\boldsymbol{y}^\circ\|_2^2}\right] = \sigma^2. \tag{2.4}$$

[8] Note we suppose here that u models exactly the real observed system and admits no modeling error.

2) Field reconstruction from partial observations

The EIM [BAR+04; MM13] propose a greedy selection of the sensors locations and an optimal reconstruction of the observed values.

a) Empirical Interpolation Method

The original EIM [BAR+04] method identifies iteratively couples of parameters and (magic) points : $(\boldsymbol{\mu}^1, \boldsymbol{x}^1)$, $(\boldsymbol{\mu}^2, \boldsymbol{x}^2)$,...,$(\boldsymbol{\mu}^N_{\text{EIM}}, \boldsymbol{x}^N_{\text{EIM}})$ in a recursive manner as follows :

Assume $\left\{(\boldsymbol{\mu}^i, \boldsymbol{x}^i)\right\}_{i=1,...,n}$ are determined, and for any $\boldsymbol{\mu}$ we can define the interpolation operator $\mathcal{I}_n$ such that

$$\mathcal{I}_n[u(.;\boldsymbol{\mu})] = \sum_{i=1}^{n} \gamma_i u(.;\boldsymbol{\mu}^i) \tag{2.5}$$

with

$$\forall k = 1, ..., n, \quad \mathcal{I}_n[u(.;\boldsymbol{\mu})](\boldsymbol{x}^k) = u(.;\boldsymbol{\mu})(\boldsymbol{x}^k) \tag{2.6}$$

then $\boldsymbol{\mu}^{n+1}$ is defined as

$$\boldsymbol{\mu}^{n+1} = \text{argmax}_{\boldsymbol{\mu}} \|u(.;\boldsymbol{\mu}) - \mathcal{I}_n[u(.;\boldsymbol{\mu})]\|_X \tag{2.7}$$

and

$$\boldsymbol{x}^{n+1} = \text{argmax}_{\boldsymbol{x}} |u(\boldsymbol{x};\boldsymbol{\mu}^{n+1}) - \mathcal{I}_n[u(.;\boldsymbol{\mu}^{n+1})](\boldsymbol{x})|. \tag{2.8}$$

Remark :

Note that it is interesting to introduce the basis

$$\xi^i = \frac{u(.;\boldsymbol{\mu}^i) - \mathcal{I}_{i-1}[u(.;\boldsymbol{\mu}^i)]}{u(\boldsymbol{x}^i;\boldsymbol{\mu}^i) - \mathcal{I}_{i-1}[u(.;\boldsymbol{\mu}^i)](\boldsymbol{x}^i)} \tag{2.9}$$

since then, for any $\boldsymbol{\mu}$,

$$\mathcal{I}_n[u(.;\boldsymbol{\mu})] = \mathcal{I}_{n-1}[u(.;\boldsymbol{\mu})] + \left[u(\boldsymbol{x}^n;\boldsymbol{\mu}) - \mathcal{I}_{n-1}[u(.;\boldsymbol{\mu})](\boldsymbol{x}^n)\right]\xi^n. \tag{2.10}$$

Aside from this original EIM method, was also proposed an alternative approach when an ordered appropriate reduced basis $\{\zeta^i\}_{i=1}^N$ is given. There is no point in identifying the $\boldsymbol{\mu}^j$, and only the "magic point" are determined through the following recursive method :

$$\boldsymbol{x}^1 = \text{argmax}_{\boldsymbol{x}} |\zeta^1(\boldsymbol{x})| \tag{2.11}$$

then

$$\xi^1 = \frac{\zeta^1}{\zeta^1(\boldsymbol{x}^1)}$$

and

$$\mathcal{I}_1[u(.;\boldsymbol{\mu})] = u(\boldsymbol{x}^1;\boldsymbol{\mu})\xi^1. \tag{2.12}$$

Next, assuming that $(\xi^1, \boldsymbol{x}^1)$, $(\xi^2, \boldsymbol{x}^2)$,...,$(\xi^n, \boldsymbol{x}^n)$ have been built and, for any $\boldsymbol{\mu}$:

$$\mathcal{I}_n[u(.;\boldsymbol{\mu})] = \mathcal{I}_{n-1}[u(.;\boldsymbol{\mu})] + \left[u(\boldsymbol{x}^n;\boldsymbol{\mu}) - \mathcal{I}_{n-1}[u(.;\boldsymbol{\mu})](\boldsymbol{x}^n)\right]\xi^n. \tag{2.13}$$

Hence

$$\mathcal{I}_n[u(.;\boldsymbol{\mu})] = \mathcal{I}_1[u(.;\boldsymbol{\mu})] + \sum_{i=1}^{n} \left[u(\boldsymbol{x}^i;\boldsymbol{\mu}) - \mathcal{I}_{i-1}[u(.;\boldsymbol{\mu})](\boldsymbol{x}^i)\right]\xi^i. \tag{2.14}$$

We construct

$$\boldsymbol{x}^{n+1} = \text{argmax}_{\boldsymbol{x}} [\zeta^{n+1}(\boldsymbol{x}) - \mathcal{I}_n[\zeta^{n+1}](\boldsymbol{x})| \tag{2.15}$$

and

$$\xi^{n+1} = \frac{\zeta^{n+1} - \mathcal{I}_n[\zeta^{n+1}]}{\zeta^{n+1}(\boldsymbol{x}^{n+1}) - \mathcal{I}_n[\zeta^{n+1}](\boldsymbol{x}^{n+1})} \tag{2.16}$$

b) General Empirical Interpolation Method

The original General Empirical Interpolation method (GEIM) [MM13] identifies iteratively couples of parameters and linear forms : $(\boldsymbol{\mu}^1, \boldsymbol{\sigma}^1)$, $(\boldsymbol{\mu}^2, \boldsymbol{\sigma}^2)$,...,$(\boldsymbol{\mu}^N_{\text{EIM}}, \boldsymbol{\sigma}^N_{\text{EIM}})$ in a recursive manner as follows : Assume $\left\{(\boldsymbol{\mu}^i, \boldsymbol{\sigma}^i)\right\}_{i=1,...,n}$ are determined, and for any $\boldsymbol{\mu}$ we can define the interpolation operator $\mathcal{J}_n$ such that

$$\mathcal{J}_n[u(.;\boldsymbol{\mu})] = \sum_{i=1}^{n} \gamma_i u(.;\boldsymbol{\mu}^i) \tag{2.17}$$

with

$$\forall k = 1, ..., n, \quad \boldsymbol{\sigma}^k\Big[\mathcal{J}_n[u(.;\boldsymbol{\mu})]\Big] = \boldsymbol{\sigma}^k\big[u(.;\boldsymbol{\mu})\big]. \tag{2.18}$$

Then $\boldsymbol{\mu}^{n+1}$ is defined as

$$\boldsymbol{\mu}^{n+1} = \text{argmax}_{\boldsymbol{\mu}} \|u(.;\boldsymbol{\mu}) - \mathcal{J}_n[u(.;\boldsymbol{\mu})]\|_X, \tag{2.19}$$

and

$$\boldsymbol{\sigma}^{n+1} = \text{argmax}_{\boldsymbol{\sigma}} |\boldsymbol{\sigma}\big[u(.;\boldsymbol{\mu}^{n+1}) - \mathcal{J}_n[u(.;\boldsymbol{\mu}^{n+1})]\big]|. \tag{2.20}$$

Remark :

Note that it is interesting to introduce the basis

$$\xi^i = \frac{u(.;\boldsymbol{\mu}^i) - \mathcal{J}_{i-1}[u(.;\boldsymbol{\mu}^i)]}{\boldsymbol{\sigma}^i\big[u(\boldsymbol{x}^i;\boldsymbol{\mu}^i) - \mathcal{J}_{i-1}[u(.;\boldsymbol{\mu}^i)]\big]} \tag{2.21}$$

since then, for any $\boldsymbol{\mu}$,

$$\mathcal{J}_n[u(.;\boldsymbol{\mu})] = \mathcal{J}_{n-1}[u(.;\boldsymbol{\mu})] + \boldsymbol{\sigma}^n\big[u(\boldsymbol{x}^n;\boldsymbol{\mu}) - \mathcal{J}_{n-1}[u(.;\boldsymbol{\mu})]\big]\xi^n \tag{2.22}$$

Aside from this original GEIM method, was also proposed an alternative approach when an ordered appropriate reduced basis $\{\zeta^i\}_{i=1}^N$ is given. There is no point in identifying the $\boldsymbol{\mu}^j$, and only the "magic moments" are determined through the following recursive method

$$\boldsymbol{\sigma}^1 = \text{argmax}_{\boldsymbol{\sigma}} |\boldsymbol{\sigma}\big[\zeta^1\big]| \tag{2.23}$$

then

$$\xi^1 = \frac{\zeta^1}{\boldsymbol{\sigma}^1\big[\zeta^1\big]}$$

and

$$\mathcal{J}_1[u(.;\boldsymbol{\mu})] = \boldsymbol{\sigma}^1 u(\boldsymbol{x}^1;\boldsymbol{\mu})\xi^1. \tag{2.24}$$

Next, assuming that $(\xi^1, \boldsymbol{x}^1)$, $(\xi^2, \boldsymbol{x}^2)$,...,$(\xi^n, \boldsymbol{x}^n)$ have been built and, for any $\boldsymbol{\mu}$,

$$\mathcal{J}_n[u(.;\boldsymbol{\mu})] = \mathcal{J}_{n-1}[u(.;\boldsymbol{\mu})] + \boldsymbol{\sigma}^n\Big[u(.;\boldsymbol{\mu}) - \mathcal{J}_{n-1}[u(.;\boldsymbol{\mu})]\Big]\xi^n. \tag{2.25}$$

Hence

$$\mathcal{J}_n[u(.;\boldsymbol{\mu})] = \mathcal{J}_1[u(.;\boldsymbol{\mu})] + \sum_{i=1}^{n} \boldsymbol{\sigma}^i\Big[u(.;\boldsymbol{\mu}) - \mathcal{J}_{i-1}[u(.;\boldsymbol{\mu})]\Big]\xi^i. \tag{2.26}$$

We construct

$$\boldsymbol{\sigma}^{n+1} = \text{argmax}_{\boldsymbol{\sigma}} |\boldsymbol{\sigma}\big[\zeta^{n+1} - \mathcal{J}_n[\zeta^{n+1}]\big]| \tag{2.27}$$

and

$$\xi^{n+1} = \frac{\zeta^{n+1} - \mathcal{J}_n[\zeta^{n+1}]}{\boldsymbol{\sigma}^{n+1}\big[\zeta^{n+1} - \mathcal{J}_n[\zeta^{n+1}]\big]}. \tag{2.28}$$

Following the *gappy EIM* [PD20] we construct $\boldsymbol{P} : \mathbb{R}^{n_d} \to \mathbb{R}^m$, $m \geq N_{\text{EIM}}$ using the EIM to add more observations than there are basis functions.

3) Succinct analysis of the noisy EIM error

Let n_d be the discrete spacetime dimension of the full-order problem, and let X be the associated state space equipped with the norm $\|\cdot\|_X$.
We recall that the POD basis $(\boldsymbol{\xi}_i)_{i=1}^{n_d}$ fully captures the discrete vector $\boldsymbol{u}^\circ \in \mathbb{R}^{n_d}$.
Let $N_{\text{EIM}} \geq 1$ be the number of EIM functions $(\boldsymbol{\psi}^i)_{i=1}^{N_{\text{EIM}}}$ who are the columns of $\boldsymbol{V}_{\text{EIM}} \in \mathbb{R}^{n_d \times N_{\text{EIM}}}$.
A function $\boldsymbol{u}^\circ \in \mathbb{R}^{n_d}$ deviated from a known reference state $\boldsymbol{u}_{\text{ref}} \in \mathbb{R}^{n_d}$ can be described by :

$$\boldsymbol{u}^\circ - \boldsymbol{u}_{\text{ref}} = \sum_{i=1}^{N_{\text{EIM}}} \gamma_i^\circ \boldsymbol{\zeta}^i + \sum_{i=N_{\text{EIM}}+1}^{n_d} \alpha_i^\circ \boldsymbol{\xi}^i = \boldsymbol{V}_{\text{EIM}} \boldsymbol{\gamma}^\circ + \boldsymbol{\eta},$$

where $\boldsymbol{\eta} = \sum_{i=N_{\text{EIM}}+1}^{n_d} \alpha_i^\circ \boldsymbol{\xi}^i$ represents the model error such that $\boldsymbol{\eta} \perp \boldsymbol{V}_{\text{EIM}}$.
Let $\boldsymbol{P} \in \mathbb{R}^{m \times n_d}$ be the sensor placement matrix sampling $m \geq n_{\text{EIM}}$ locations. The noisy sensor observation vector $\boldsymbol{y}_\varepsilon^\circ \in \mathbb{R}^m$ is modeled as :

$$\boldsymbol{y}_\varepsilon^\circ = \boldsymbol{P}\boldsymbol{u}^\circ + \boldsymbol{e},$$

where $\boldsymbol{e} \in \mathbb{R}^m$ is a relative measurement noise vector defined by $\boldsymbol{e} = \sigma(\boldsymbol{P}\boldsymbol{u}^\circ) \odot \boldsymbol{\epsilon}$, with noise level $\sigma > 0$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_m)$.
The gappy EIM reconstruction operator $\boldsymbol{M} \in \mathbb{R}^{n_d \times m}$ is defined as $\boldsymbol{M} = \boldsymbol{V}_{\text{EIM}}(\boldsymbol{P}\boldsymbol{V}_{\text{EIM}})^\dagger$, which satisfies the fundamental interpolation identity $\boldsymbol{M}\boldsymbol{P}\boldsymbol{V}_{\text{EIM}} = \boldsymbol{V}_{\text{EIM}}$. The reconstructed field is given by :

$$\boldsymbol{u}_{\text{EIM}} = \boldsymbol{M}(\boldsymbol{y}_\varepsilon^\circ - \boldsymbol{P}\boldsymbol{u}_{\text{ref}}) + \boldsymbol{u}_{\text{ref}}.$$

By substituting the observation model into the reconstruction equation, we isolate the total approximation error:

$$\begin{aligned}
\boldsymbol{u}^\circ - \boldsymbol{u}_{\text{EIM}} &= (\boldsymbol{u}^\circ - \boldsymbol{u}_{\text{ref}}) - \boldsymbol{M}\left(\boldsymbol{P}(\boldsymbol{u}^\circ - \boldsymbol{u}_{\text{ref}}) + \boldsymbol{e}\right) \\
&= (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{P})(\boldsymbol{u}^\circ - \boldsymbol{u}_{\text{ref}}) - \boldsymbol{M}\boldsymbol{e} \\
&= (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{P})(\boldsymbol{V}_{\text{EIM}}\boldsymbol{\gamma}^\circ + \boldsymbol{\eta}) - \boldsymbol{M}\boldsymbol{e}.
\end{aligned}$$

Using the property $(\boldsymbol{I} - \boldsymbol{M}\boldsymbol{P})\boldsymbol{V}_{\text{EIM}} = \boldsymbol{0}$, the projection error cleanly simplifies to :

$$\boldsymbol{u}^\circ - \boldsymbol{u}_{\text{EIM}} = (\boldsymbol{I} - \boldsymbol{M}\boldsymbol{P})\boldsymbol{\eta} - \boldsymbol{M}\boldsymbol{e}.$$

Taking the total norm and applying the triangle inequality yields :

$$\|\boldsymbol{u}^\circ - \boldsymbol{u}_{\text{EIM}}\|_X \leq \|\boldsymbol{I} - \boldsymbol{M}\boldsymbol{P}\|_2 \, \|\boldsymbol{\eta}\|_X + \|\boldsymbol{M}\boldsymbol{e}\|_X \, .$$

Squaring both sides and taking the expectation with respect to the random noise field $\boldsymbol{\epsilon}$, we evaluate the scaling of the stochastic term :

$$\begin{aligned}
\mathbb{E}_\epsilon\left[\|\boldsymbol{M}\boldsymbol{e}\|_X^2\right] &= \mathbb{E}_\epsilon\left[\|\boldsymbol{M}(\boldsymbol{P}\boldsymbol{u}^\circ \odot \boldsymbol{\epsilon})\|_X^2\right] \\
&\leq \|\boldsymbol{M}\|_2^2 \, \mathbb{E}_\epsilon\left[\|\boldsymbol{P}\boldsymbol{u}^\circ \odot \boldsymbol{\epsilon}\|_2^2\right] \\
&= \|\boldsymbol{M}\|_2^2 \sum_{i=1}^{m} |(\boldsymbol{P}\boldsymbol{u}^\circ)_i|^2 \, \mathbb{E}_\epsilon[\epsilon_i^2] \\
&= \sigma^2 \, \|\boldsymbol{M}\|_2^2 \sum_{i=1}^{m} |(\boldsymbol{P}\boldsymbol{u}^\circ)_i|^2 \\
&\leq \sigma^2 \, \|\boldsymbol{M}\|_2^2 \, \|\boldsymbol{P}\|_2^2 \, \|\boldsymbol{u}^\circ\|_X^2 \, .
\end{aligned}$$

Normalizing by the energy of the true state $\|\boldsymbol{u}^\circ\|_X^2$ and using Young's inequality for products :

$$\frac{\mathbb{E}_\epsilon\left[\|\boldsymbol{u}^\circ - \boldsymbol{u}_{\text{EIM}}\|_X^2\right]}{\|\boldsymbol{u}^\circ\|_X^2} \leq 2\,\|\boldsymbol{I} - \boldsymbol{M}\boldsymbol{P}\|_2^2\,\frac{\|\boldsymbol{\eta}\|_X^2}{\|\boldsymbol{u}^\circ\|_X^2} + 2\sigma^2\,\|\boldsymbol{M}\|_2^2\,\|\boldsymbol{P}\|_2^2\,. \tag{2.29}$$

Equation (2.29) highlights the fundamental EIM trade-off: increasing the number of modes N_{EIM} minimizes the relative truncation error $\|\boldsymbol{\eta}\|_X / \|\boldsymbol{u}^\circ\|_X$, but increases the operator norm bounds $\|\boldsymbol{M}\|_2$, thereby magnifying the sensor noise sensitivity proportional to σ^2.
Bounds of $\|\boldsymbol{I} - \boldsymbol{M}\boldsymbol{P}\|_2$ and $\|\boldsymbol{M}\|_2$ exist with respect to the number of EIM basis functions N_{EIM} and the number of observations $m \geq N_{\text{EIM}}$ [PD20] .

4) Data assimilation methods

Figure 1 summarizes the workflow that will be described in the next part.
The goal is to build in an offline part the needed POD basis, the EIM basis and magic points and an encoder h trained via a Gaussian Process Regressor (GPR). Then, from a noisy partial observation of the system, we try different methods to recover the reduced POD basis coefficients

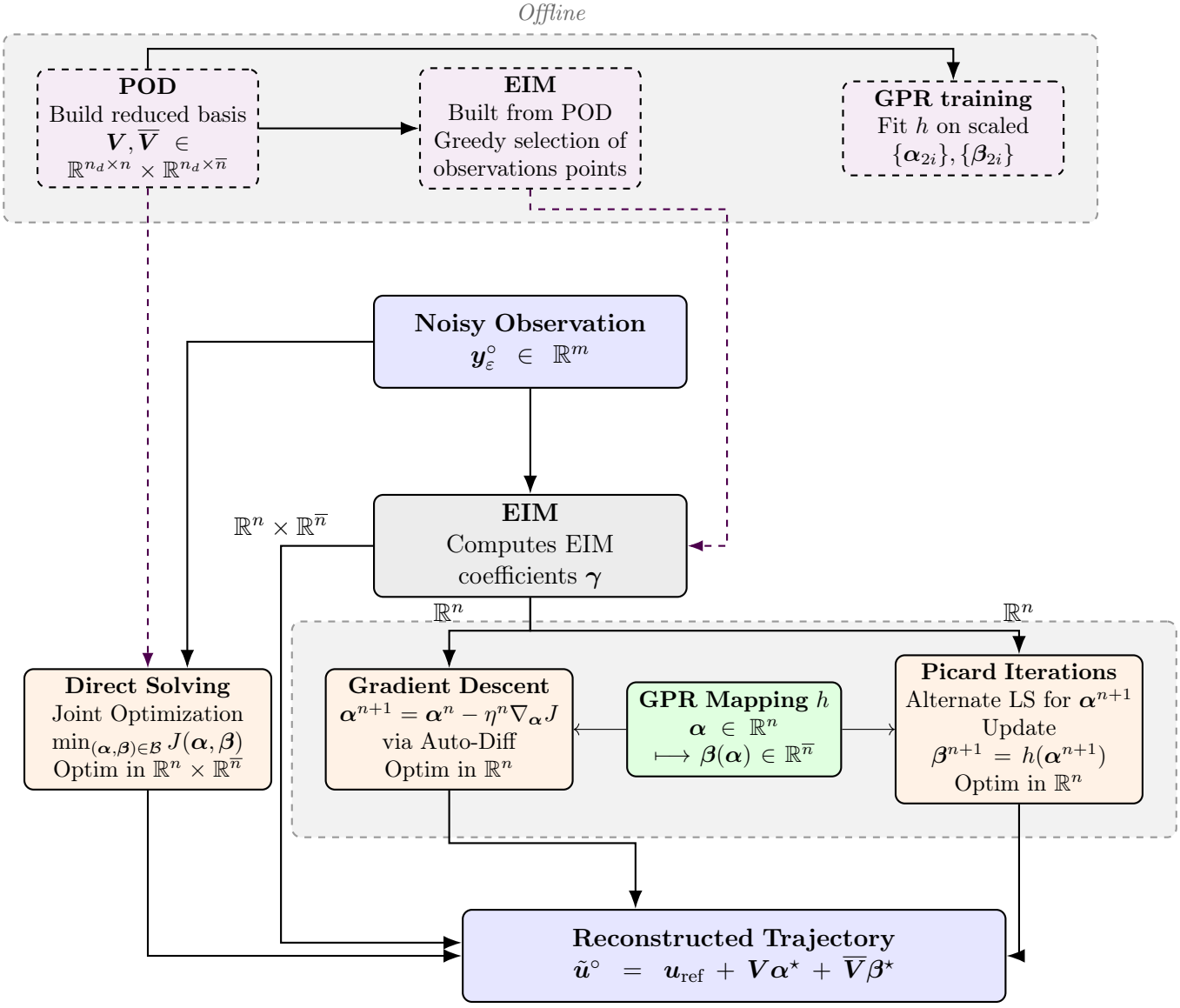


Figure 1: Architecture of the different proposed methods of assimilation.

We focus on retrieving the full trajectory $\boldsymbol{u}_{n_d}$ from observations $\boldsymbol{y}_\varepsilon^\circ$ using the POD basis.

This has already been addressed with the Generalized Empirical Interpolation Method (EIM) [BAR+04; MM13; GON+19], GEIM with ML closure [NIU+25] and Parameterized Best Knowledge Model (PBDW) [MAD+15; GON+19] if accounting for modeling error.

Let $\boldsymbol{V}_N$ be a reduced basis. Recall that $\Pi_{\boldsymbol{V}_N}$ is the orthogonal projection onto $\boldsymbol{V}_N$.
Let $\boldsymbol{P} : \mathbb{R}^{n_d} \to \mathbb{R}^m$ be a linear observation operator.
For an noisy observation $\boldsymbol{y}_\varepsilon^\circ$ find POD coefficients $(\boldsymbol{\alpha}^\circ, \boldsymbol{\beta}^\circ) \in \mathbb{R}^n \times \mathbb{R}^{\overline{n}}$ that are the POD coefficients of the true trajectory $\boldsymbol{u}^\circ \in \mathbb{R}^{n_d}$:

$$\begin{cases} \boldsymbol{y}_\varepsilon^\circ &= \boldsymbol{P}\boldsymbol{u}^\circ + \boldsymbol{e}, \\ \tilde{\boldsymbol{u}}^\circ &= \boldsymbol{u}_{\text{ref}} + \Pi_{\boldsymbol{V}_N}(\boldsymbol{u}^\circ - \boldsymbol{u}_{\text{ref}}) \quad = \quad \boldsymbol{u}_{\text{ref}} + \boldsymbol{V}\boldsymbol{\alpha}^\circ + \overline{\boldsymbol{V}}\boldsymbol{\beta}^\circ. \end{cases} \tag{2.30}$$

Let $\boldsymbol{\alpha}_{\min}, \boldsymbol{\alpha}_{\max} \in \mathbb{R}^n$ and $\boldsymbol{\beta}_{\min}, \boldsymbol{\beta}_{\max} \in \mathbb{R}^{\overline{n}}$ be respectively the **componentwise** minimum and maximum values of $\boldsymbol{\alpha}, \boldsymbol{\beta}$ for all the snapshots in $\mathbb{P}_h$. Let $\delta_\alpha, \delta_\beta > 0$ be some box tolerance.
In what follow, we will use the following box constraint during optimizations [GON+19] :

$$\mathcal{B} := \prod_{i=1}^{n} [\alpha_{\min,i} - \delta_\alpha; \alpha_{\max,i} + \delta_\alpha] \times \prod_{i=1}^{\overline{n}} [\beta_{\min,i} - \delta_\beta; \beta_{\max,i} + \delta_\beta] \tag{2.31}$$

or

$$\tilde{\mathcal{B}} = \prod_{i=1}^{n} [\alpha_{\min,i} - \delta_\alpha; \alpha_{\max,i} + \delta_\alpha]. \tag{2.32}$$

When there is no ambiguity, we will use the notation $\mathcal{B}$ even when optimizing only on $\boldsymbol{\alpha}$.

We propose several methods that can be written in two forms for a cost function $J > 0$:

$$\text{(a)} \quad \text{Find } (\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star) := \underset{(\boldsymbol{\alpha},\boldsymbol{\beta})\in\mathcal{B}}{\operatorname{argmin}} J(\boldsymbol{\alpha}, \boldsymbol{\beta}), \tag{2.33}$$

$$\text{(b)} \quad \text{Find } \boldsymbol{\alpha}^\star := \underset{\boldsymbol{\alpha}\in\mathcal{B}}{\operatorname{argmin}} J(\boldsymbol{\alpha}), \quad \boldsymbol{\beta}^\star = h(\boldsymbol{\alpha}^\star). \tag{2.34}$$

a) Direct solving with full POD model

For an observed $\boldsymbol{y}_\varepsilon^\circ \in \mathbb{R}^m$, we solve the linear least square problem :

$$(\boldsymbol{\alpha}^\star_{\text{POD}}, \boldsymbol{\beta}^\star_{\text{POD}}) := \underset{(\boldsymbol{\alpha},\boldsymbol{\beta})\in\mathcal{B}}{\operatorname{argmin}} \frac{1}{2} \left\| \boldsymbol{y}_\varepsilon^\circ - \boldsymbol{P}(\boldsymbol{V}\boldsymbol{\alpha} + \overline{\boldsymbol{V}}\boldsymbol{\beta} + \boldsymbol{u}_{\text{ref}}) \right\|^2_{\mathbf{L}^2(\mathbb{R}^m)}. \tag{2.35}$$

b) Gradient descent with ML model

We let $(\boldsymbol{\alpha}_0, \boldsymbol{\beta}_0) = \text{EIM}(\boldsymbol{y}_\varepsilon^\circ)$. Suppose the ML function h is differentiable, we apply a gradient descent to :

$$J(\boldsymbol{\alpha}) = \frac{1}{2} \left\| \boldsymbol{y}_\varepsilon^\circ - \boldsymbol{P}(\boldsymbol{V}\boldsymbol{\alpha} + \overline{\boldsymbol{V}}h(\boldsymbol{\alpha}) + \boldsymbol{u}_{\text{ref}}) \right\|^2_{\mathbf{L}^2(\mathbb{R}^m)}. \tag{2.36}$$

$$\boldsymbol{\alpha}^\star_{\text{GD}} := \underset{\boldsymbol{\alpha}\in\mathcal{B}}{\operatorname{argmin}} J(\boldsymbol{\alpha}). \tag{2.37}$$

The gradient with respect to the input $\boldsymbol{\alpha}$ is :

$$\nabla_{\boldsymbol{\alpha}} J = \left(\boldsymbol{P}\boldsymbol{V} + \boldsymbol{P}\overline{\boldsymbol{V}} \frac{\partial h}{\partial \boldsymbol{\alpha}} \right)^T (\boldsymbol{y}_\varepsilon^\circ - \boldsymbol{P}(\boldsymbol{V}\boldsymbol{\alpha} + \overline{\boldsymbol{V}}h(\boldsymbol{\alpha}) + \boldsymbol{u}_{\text{ref}}) \ ) \in \mathbb{R}^n. \tag{2.38}$$

The gradient $\dfrac{\partial h}{\partial \boldsymbol{\alpha}} \in \mathbb{R}^{\overline{n}\times n}$ is done by automatic differentiation [NEI10; BAY+18] using the Tensorflow library[9].
Let $\eta^n > 0$ be the learning rate defined in [BB88]. We update $\boldsymbol{\alpha}^n$ as :

$$\boldsymbol{\alpha}^{n+1} = \boldsymbol{\alpha}^n - \eta^n \nabla_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}^n).$$

The process stops when $\|\nabla_\alpha J(\boldsymbol{\alpha}^n)\| < \text{tol}$ or $n > n_{\max \text{ iter}}$ where tol and $n_{\max \text{ iter}}$ are arbitrary constants.
We denote the solution $\boldsymbol{\alpha}^\star_{\text{GD}}$ at the end of the process and set $\boldsymbol{\beta}^\star_{\text{GD}} = h(\boldsymbol{\alpha}^\star_{\text{GD}})$ such that the final solution is given by $(\boldsymbol{\alpha}^\star_{\text{GD}}, \boldsymbol{\beta}^\star_{\text{GD}})$.

c) Picard iterations

Let $\boldsymbol{\gamma}_0 = \text{EIM}(\boldsymbol{y}^\circ_\varepsilon)$. Let $\boldsymbol{\alpha}_0 = \Pi_{\boldsymbol{V}}(\boldsymbol{V}_{\text{EIM}}\boldsymbol{\gamma}_0)$. We set $\boldsymbol{\beta}_0 = h(\boldsymbol{\alpha}_0)$.
Suppose we computed $\boldsymbol{\alpha}_n, \beta_n$. We solve the linear least square problem :

$$\boldsymbol{\alpha}^{n+1} := \underset{\boldsymbol{\alpha}\in\mathcal{B}}{\text{argmin}} \frac{1}{2} \left\| \boldsymbol{y}^\circ_\varepsilon - \boldsymbol{P}(\boldsymbol{V}\boldsymbol{\alpha} + \overline{\boldsymbol{V}}\boldsymbol{\beta}^{\boldsymbol{n}} + \boldsymbol{u}_{\text{ref}}) \right\|^2_{\mathbf{L}^2(\mathbb{R}^m)}. \tag{2.39}$$

Once computed, we set $\boldsymbol{\beta}^{n+1} := h(\boldsymbol{\alpha}^{n+1})$.
The optimization process is repeated until $\dfrac{\left\|\boldsymbol{\alpha}^{n+1} - \boldsymbol{\alpha}^n\right\|_{\mathbb{R}^n}}{\|\boldsymbol{\alpha}^n\|_{\mathbb{R}^n}} < \text{tol}$ or until the maximum number of iterations $n_{\max}$ has been reached.
We denote the solution at the end of the process $(\boldsymbol{\alpha}^\star_{\text{Picard}}, \boldsymbol{\beta}^\star_{\text{Picard}}) = (\boldsymbol{\alpha}^{n_{\text{final}}+1}, \boldsymbol{\beta}^{n_{\text{final}}+1})$ (or $\boldsymbol{\beta}^{n_{\text{final}}}$).

5) Learning the mapping between coefficients

We use Gaussian Process Regression (GPR) [RW05] models to learn the mapping $\boldsymbol{\alpha} \in \mathbb{R}^n \longmapsto \boldsymbol{\beta}(\boldsymbol{\alpha})$.
We use different kernels [RW05, Section 4.2] to learn the function :

- Constant kernel of scale ℓ :
$$\text{Cste}(\ell)(x_i, x_j) = \ell^2.$$
- Dot product kernel of scale ℓ :
$$\text{Dot}(\ell)(x_i, x_j) = \ell^2 + x_i x_j.$$
- Radial Basis Function kernel (RBF) with scale ℓ :
$$\text{RBF}(\ell)(x_i, x_j) = \exp(-\left|x_i - x'_j\right|^2 / 2\ell^2).$$
- Matern kernel with scale ℓ and regularity ν :
$$\text{Matern}(\nu)(x_i, x_j) = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\sqrt{2\nu}}{\ell} |x_i - x_j| \right)^\nu K_\nu(x_i, x_j).$$
Here K_ν is a modified Bessel function, see [RW05, Chapter 4, Section 4.2] .
- White noise kernel of scale ϵ (here δ_0 is the Dirac operator at 0) :
$$\text{WhiteNoise}(\epsilon)(x_i, x_j) = \epsilon\ \delta_0(x_i, x_j).$$

[9] www.tensorflow.org/

The sum, dot product and power of kernels are allowed operations that are aready implementend in the used libraries. The hyperparameters (scale, noise level, constants, ...) are automaticaly optimized within the fitting part. The GPR is trained on scaled input.
We use the scikit-learn[10] gaussian process library for the training over the data.
If needed (e.g. for automatic differentiation), the sklearn GPR will be converted into tensorflow[11] using GPflow[12] GPR.
We suppose $\boldsymbol{\mu}_1 < \boldsymbol{\mu}_2 < ... < \boldsymbol{\mu}_{N_{\text{samples}}}$. The GPR is trained on $\mathbb{P}_{h,\text{train}} := \{\boldsymbol{\mu}_{2i}\}_i$ then tested on $\mathbb{P}_{h,\text{test}} := \{\boldsymbol{\mu}_{2i+1}\}_i$.

II - Numerical results

1) Test cases

a) Orthogonal modes, $P = 1$, $N = 2$

Let $I = [0;\pi] \subset \mathbb{R}$, $\mathbb{P} = [0;1] \subset \mathbb{R}$ ($P = 1$). Here, the parameter space dimension is 1. Let $\lambda = 0.1$.
We consider a function u parameterized by $\mu \in \mathbb{P}$:

$$u(x\,;\mu) = \hat{u}_1(\mu)\Phi_1(x) + \hat{u}_2(\mu)\Phi_2(x) \tag{2.40}$$

$$= \sqrt{12}\left(\mu - \frac{1}{2}\right)\sqrt{\frac{2}{\pi}}\sin(x) + \lambda\sqrt{180}\left(\mu^2 - \mu + \frac{1}{6}\right)\sqrt{\frac{2}{\pi}}\cos(x). \tag{2.41}$$

Following Appendix B, the function u is its own exact POD approximation i.e. $\boldsymbol{u}_{n_d} = \boldsymbol{u}_n$ for $N = 2$ with modes :

$$\begin{cases} \Phi_1 &= \sqrt{\frac{2}{\pi}}\sin, \\ \Phi_2 &= \sqrt{\frac{2}{\pi}}\cos. \end{cases} \tag{2.42}$$

And coefficients :

$$\begin{cases} \alpha(\mu) := \hat{u}_1(\mu) &= \sqrt{12}\left(\mu - \frac{1}{2}\right), \\ \beta(\mu) := \hat{u}_2(\mu) &= \lambda\sqrt{180}\left(\mu^2 - \mu + \frac{1}{6}\right). \end{cases} \tag{2.43}$$

We discretize $[0;\pi]$ in $n_d = 4096$ points $x_0 = 0 \leq x_i = \frac{i}{4095} \leq x_{4095} = \pi$. The samples are $\boldsymbol{u}_{n_d} = \begin{bmatrix} u(x_0) & ... & u(x_{4095}) \end{bmatrix}^T \in \mathbb{R}^{4096}$.
We construct $\mathbb{P}_h$ by uniformly sampling 2000 values in in $[0;1]$. We define $\mathbb{P}_{h,\text{train}} = \{\mu_{2i} \;/\; i \in [\![0;2000]\!]\}$.

Up to sampling error, the discrete POD modes (Φ_1, Φ_2) will be :

$$\Phi_1^h = \pm\sqrt{2/\pi}\sin, \qquad \Phi_2^h = \pm\sqrt{2/\pi}\cos.$$

We have the relation of dependance between α and β :

$$\beta(\alpha) = \lambda\frac{\sqrt{180}}{12}((\alpha)^2 - 1). \tag{2.44}$$

Thus we will set $n = 1$, $\overline{n} = 1$ and $N = n + \overline{n} = 2$.

[10] https://scikit-learn.org/
[11] www.tensorflow.org/
[12] gpflow.github.io

b) Advection of a gaussian bump, $P = 1$, $N = \infty$

Let $I = [-0.1; 1.1] = [a; b]$, $\mathbb{P} = [0; 1]$ $(P = 1)$. Let $s > 0$.
We consider an accelerating advection of a gaussian bump :

$$u(x\ ; \mu) = \frac{1}{(2\pi s^2)^{1/2}} \exp\left(-\frac{(x-\mu^2)^2}{2s^2}\right). \tag{2.45}$$

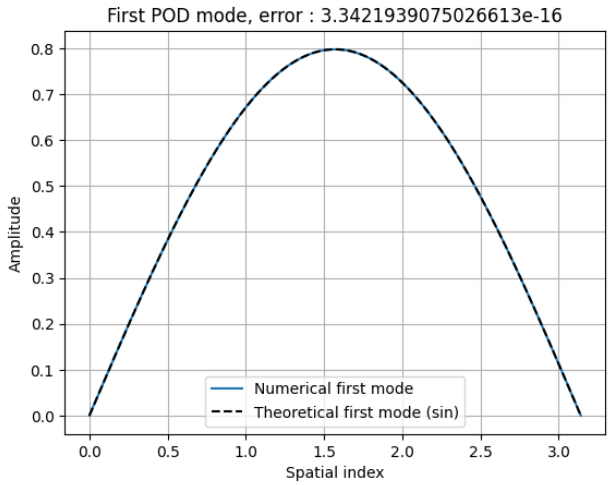
We choose $s^2 = 0.01$ such that $u(\boldsymbol{x} = -0.1\ ; \mu = 0) = u(\boldsymbol{x} = 1.1\ ; \mu = 1) \simeq 0$.
We discretize $[-0.1; 1.1]$ in $n_d = 4096$ points $x_0 = -0.1 \leq x_i = \frac{i}{4095} \leq x_{4095} = 1.1$. The samples are $\boldsymbol{u}_{n_d} = \begin{bmatrix} u(x_0) & \dots & u(x_{4095}) \end{bmatrix}^T \in \mathbb{R}^{4096}$.
We construct $\mathbb{P}_h$ by uniformly sampling 2000 values with the function $\mu \in [0; 1] \mapsto \mu^2$ which is equivalent to uniformly sampling $[0; 1]$ and consider $\exp(-(\boldsymbol{x}-\mu)^2)$. We define $\mathbb{P}_{h,\text{train}} = \{\mu_{2i}\ /\ i \in [\![0; 2000]\!]\}$.

Appendix B shows that it is not possible to exactly recover $\mathcal{M} = \{u(\cdot\ ; \mu)\ /\ \mu \in \mathbb{P}\}$ with a finite dimensional space, but it can be done up to any numerical precision.

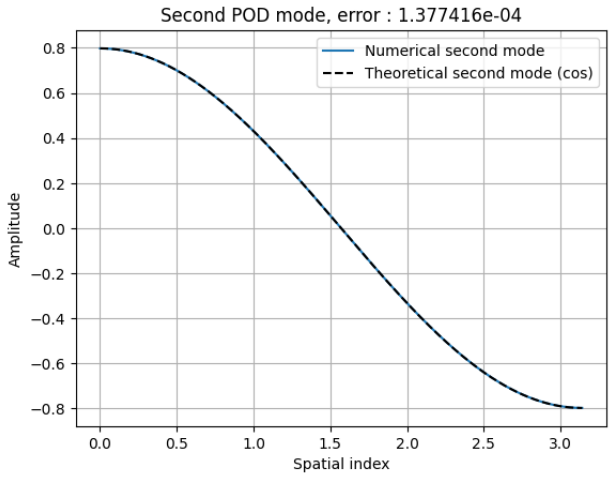
2) Orthogonal modes results

a) POD and EIM analysis

We benchmark the POD implementation on the Orthogonal modes Eq.(2.41) for $N_{\text{train}} = 2000$.
Recall $\Phi_1 = \sqrt{\frac{2}{\pi}} \sin$ and $\Phi_2 = \sqrt{\frac{2}{\pi}} \cos$.



(a) First mode of the numerical POD Φ_1^h and the theoretical first mode Φ_1.



(b) Second mode of the numerical POD Φ_2^h and the theoretical second mode Φ_2.

Figure 2: Modes of the numerical POD and their $\|\cdot\|_X^2$-error with respect to the theoretical mode.

The POD mode error is computed by $\left\|\Phi_i^h - \Phi_i\right\|_X$ following Eq.(3.3) of Appendix A.

The energy is $\mathcal{E}_1 = \sigma_1^2 = 1 - \lambda^2 = 0.99$ and $\mathcal{E}_2 = \sigma_2^2 = \lambda^2 = 0.01$, as stated in Appendix B :

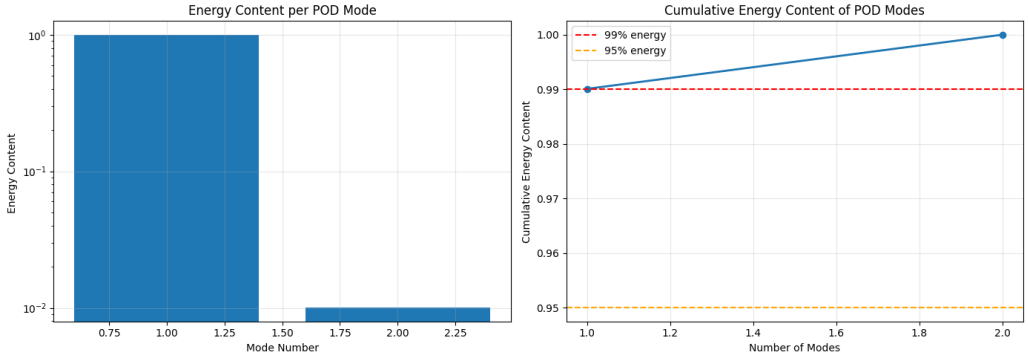


Figure 3: Energy histogram per modes (left). Cumulative energy $I = \sum_{i \leq n} \mathcal{E}_i / \sum_i \mathcal{E}_i$ (right).

We can verify Eq.(2.43) and Eq.(2.44) :

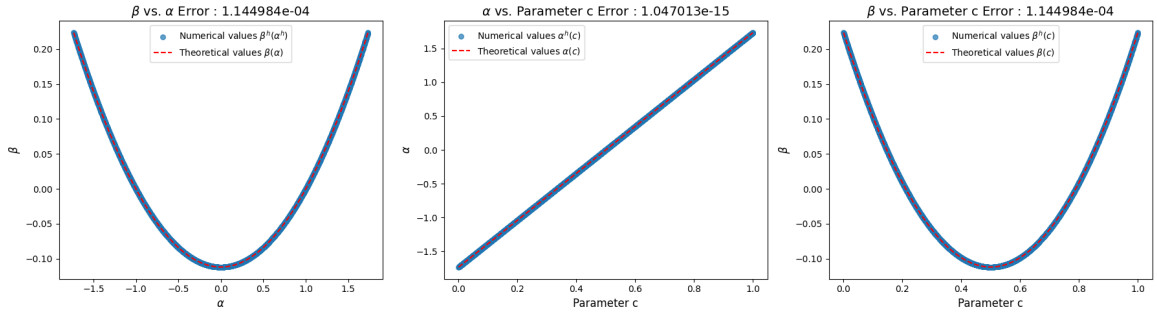


Figure 4: Comparison of the numerical coefficients with the theory of the orthogonal test case.

The error is computed as the Mean Relative Error Eq.(3.10) of the $\|\cdot\|_{\mathrm{L}^2(\mathbb{P}_h)}$ norm with uniform sampling, see Appendix A.
Here, $c = \mu \in \mathbb{P}$ is the parameter, $N_s = N_{\text{train}} = 2000$, $\alpha = \alpha(c)$ and $\beta = \beta(c)$ are defined Eq.(2.43).

The error values are numerical precision on α and small enough on β to consider that the POD is well-implemented up to sampling error.
We check that the selected magic points of the EIM are $x_1 \in \frac{\pi}{2}$ and $x_2 \in \{0, \pi\}$ (Appendix B) :

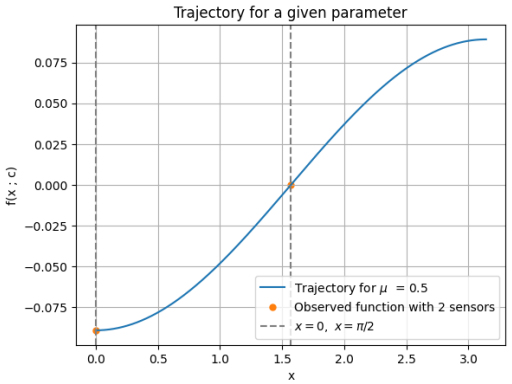


Figure 5: Trajectory $u^\circ(x\,;\mu)$ for a given $\mu = 0.5$, the observed function $\boldsymbol{y}^\circ$ and the sensors location.

We compare the results of EIM and POD in the case where there's no noise ($\sigma = 0$).

	Number of Modes	MRE	RMSE	Relative L^∞ error
POD	$n = 2$	7.78×10^{-16}	7.04×10^{-16}	2.20×10^{-14}
EIM	$m = n = 2$	1.99×10^{-14}	7.61×10^{-13}	2.19×10^{-14}

Table 2: Error metrics in the $\|\cdot\|_X$ norm.

Following Appendix A the MRE is done for the $\mathbf{L}^2(\mathbb{P}_h)$-norm with $y = \boldsymbol{M}_\delta^{1/2}\boldsymbol{u}_{n_d}(\boldsymbol{\mu})$ and $v = \boldsymbol{M}_\delta^{1/2}\boldsymbol{u}_n(\boldsymbol{\mu})$. The RMSE is done for the same y, v.
The relative L^∞ error is done on the space $\mathbf{L}^2(\mathbb{P}_h, \mathbf{L}^2(\Omega, \mathbb{R}^{n_d}))$.

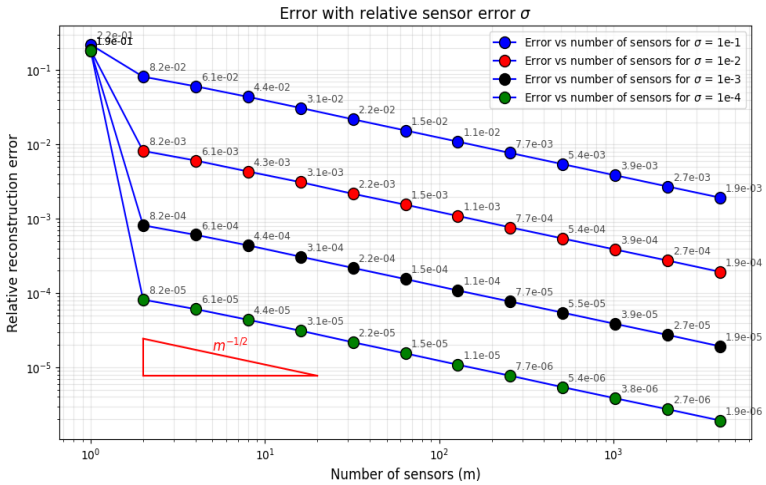


Figure 6: Relative error $\|\boldsymbol{u}^\circ - \boldsymbol{u}_{\text{EIM}}\|_X / \|\boldsymbol{u}^\circ\|_X$ over $\mathbb{P}_{h,\text{test}}$ for different value of the noise scale σ when the number of sensors m varies and $n_{\text{EIM}} = 2$ if $m > 1$, $n_{\text{EIM}} = 1$ if $m = 1$.

Note that the slope is $m^{-1/2}$ for each error scale and the error for $m = N = 2$ is of the scale of σ which is what the theory predict [PD20] and Eq.(2.29).

b) Learning the coefficients map

Recall we are trying to learn the mapping Eq.(2.44) :

$$\beta(\alpha) = \lambda \frac{\sqrt{180}}{12}((\alpha)^2 - 1).$$

Kernel	$(n, \overline{n})$	MRE	RMSE	Relative L^∞ error
RBF + Cste**2	(1,1)	1.77×10^{-07}	5.14×10^{-09}	4.28×10^{-08}
Matern(nu=1.5) + Cste**2	(1,1)	4.22×10^{-08}	3.56×10^{-08}	4.89×10^{-06}
Matern(1, nu=5) + Cste**2 + White	(1,1)	2.85×10^{-09}	1.96×10^{-09}	1.66×10^{-07}
Dot ** 2	(1,1)	3.80×10^{-16}	5.22×10^{-16}	1.19×10^{-14}

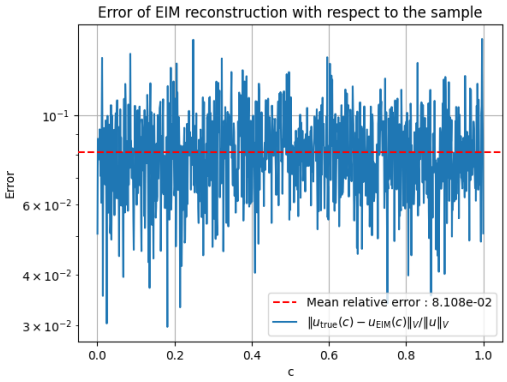
Table 3: Error metrics of the GPR output $h(\alpha(\mu))$ with respect to the target value $\beta(\mu)$ for different kernels and $\mu \in \mathbb{P}_{h,\text{test}}$.

Note that DotProduct optimized its scale to $\ell = 0.999$ which implies the optimized kernel is a polynomial of the form $x^2 + \ell^2$ up to a scaling. That correspond almost exactly to $\alpha^2 + (-1)^2$.
With not much surprises, the polynomial kernel performs the best.

c) Online data assimilation

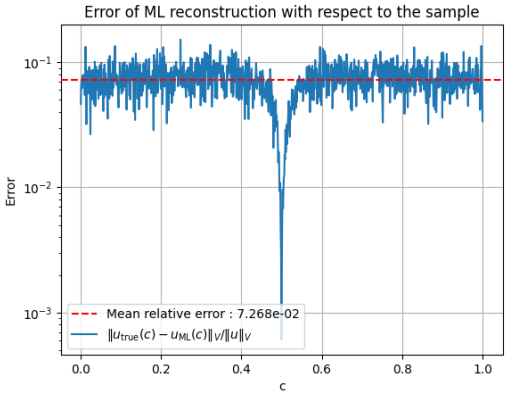
For each $\mu \in \mathbb{P}_{h,\text{test}}$ we generate $N_{\text{real}} = 10$ trajectories $\boldsymbol{u}_i^\circ(\mu)$ and a noisy observation $\boldsymbol{y}_i^\circ(\mu) = \boldsymbol{P}\boldsymbol{u}_i^\circ + \boldsymbol{e}_i$ with independent realization of the noise $\boldsymbol{e}$. We consider a relative scale $\sigma = 10^{-1}$. This construct $N_{\text{real}} \times n_{\text{test samples}} = 10^4$ noisy observations.

For $\mu \in \mathbb{P}$, $\epsilon_i(\mu) = \left\|\boldsymbol{u}_i^\circ(\mu) - \boldsymbol{u}_i^{\text{reconstructed}}\right\|_X / \left\|\boldsymbol{u}_i^\circ(\mu)\right\|_X$. Figure 7 plot $\epsilon(\mu) = \frac{1}{N_{\text{real}}} \sum_{i=1}^{N_{\text{real}}} \epsilon_i(\mu)$.



(a) Empirical Interpolation Method.

(b) Proper Orthogonal Decomposition.



(c) Gradient descent.

(d) Picard iteration method.

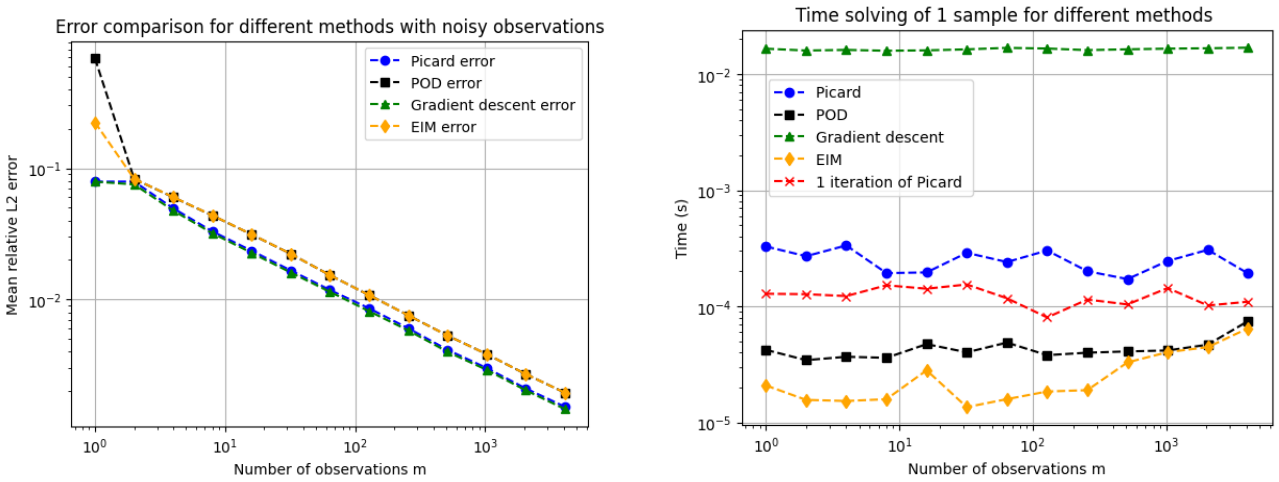
Figure 7: Error comparison for $\mu \in \mathbb{P}_{h,\text{test}}$, $m = N_{\text{modes}} = 2$, $\sigma = 10^{-1}$.

The Gradient Descent with Machine Learning and Picard iterations perform slightly better with a huge improvement for $\mu = 1/2$ (which is the root of the function $\alpha(\mu)$).
In Table 4, we showcase the average time taken to reconstruct 1 sample from $\boldsymbol{y}^\circ$.

Method	POD	Picard	EIM	Gradient Descent
Time (s/sample)	3.35×10^{-5}	3.83×10^{-4}	1.66×10^{-5}	3.27×10^{-2}.

Table 4: Computation time per sample of different methods for orthogonal modes.

We note that EIM is the fastest because the interpolating matrix is cached, and $n = \overline{n} = 1$. We think that POD or Picard iterations can be faster for $n \gg \overline{n}$. The real bottleneck seems to be the Gradient Descent which is slowed by the GPflow auto-differentiation of the GPR model.



(a) Reconstruction error for $\sigma = 10^{-1}$.

(b) Computation time for 1 sample.

Figure 8: Performance comparison of different methods when the number of sensors $m = 2^i$, $i = 0, ..., 12$ and $n = \overline{n} = 1$.

In Figure 8a we observe similar performances of all the methods, with slight advantages for methods using the GPR model. Figure 8b highlights faster performances for EIM up to a certain number of sensors m from which the EIM is not faster anymore.
The methods using the GPR seems have constant solving time for m if $m \leq 4096$, $n = \overline{n} = 1$.
We note that for $m = 1$ (which implies that only 1 mode is used for EIM), the GPR closure slightly outperforms the EIM and POD method while still being fast in the Picard case.

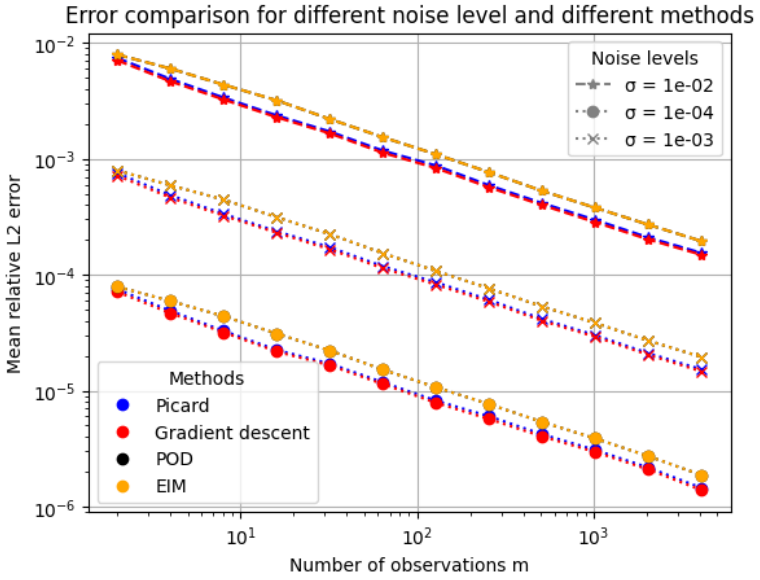


Figure 9: Relative error of reconstruction $\|\boldsymbol{u}_{n_d}(\mu) - \boldsymbol{u}^\star(\mu)\|_X / \|\boldsymbol{u}_{n_d}\|_X$ depending on the noise level σ when $m = 2^i$, $i = 1, ..., 12$ and $n = \overline{n} = 1$.

Figure 9 showcases the similar behavior regarding the relative noise level σ and the number of sensors m. Note we discarded $m = 1$ since it is a special case here.

3) Advection of a gaussian bump

a) Analysis of the POD learnability

We note that this test case is advection dominated, thus we expect the cumulative energy $\mathcal{E} = \sum_{i=1}^{\infty} \sigma_i^2$ to have slow decaying terms :

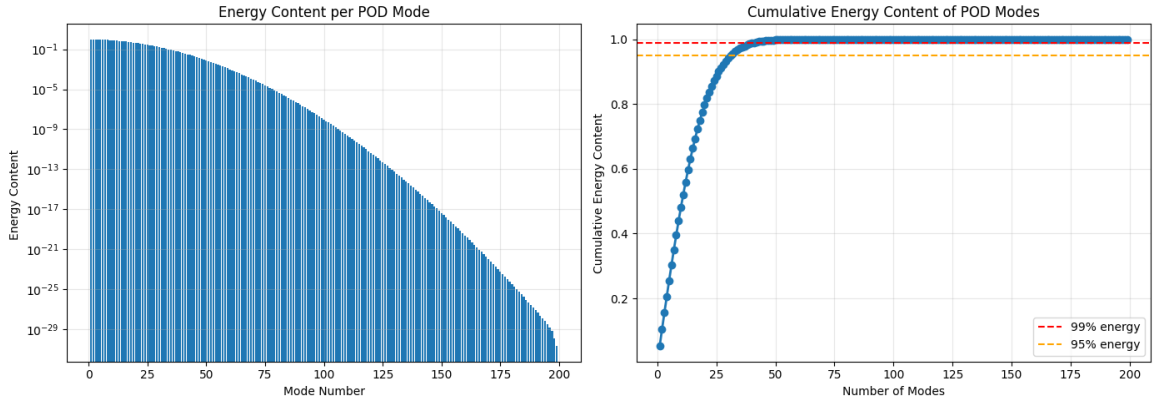


Figure 10: Energy histogram per modes (left). Cumulative energy $I = \sum_{i \leq n} \mathcal{E}_i / \sum_i \mathcal{E}_i$ (right).

We select value of the singular values σ_N is $\sigma_N \lesssim 10^{-16}$ which is numerical noise. We thus select $N = 199$.

	Number of Modes	MRE	RMSE	Relative L^∞ error
POD	$n = 199$	6.25×10^{-15}	3.08×10^{-14}	5.51×10^{-14}
EIM	$m = 199$	9.78×10^{-15}	2.88×10^{-12}	6.24×10^{-14}

Table 5: Error metrics of the EIM for different test cases in the $\|\cdot\|_X$ norm.

The following table compares in the higher part the error for different values of n with the same kernel initialisation.
The lower part compares different kernel that has been tried. Only Matern and RBF have been able to showcase significant learning results for the whole set of coefficients.

Kernel	$(n, \overline{n})$	MRE	RMSE	Relative L^∞ error
RBF + Cste**2 + White	(1,100)	1.06×10^{-00}	3.04×10^{-00}	8.13×10^{-01}
RBF + Cste**2 + White	**(2,100)**	8.03×10^{-08}	9.72×10^{-06}	4.83×10^{-05}
RBF + Cste**2 + White	(3,100)	9.42×10^{-08}	2.85×10^{-06}	5.69×10^{-05}
RBF + Cste**2 + White	**(2,125)**	2.62×10^{-06}	3.00×10^{-04}	1.04×10^{-03}
RBF + Cste**2 + White	(3,125)	2.82×10^{-06}	3.23×10^{-04}	1.13×10^{-03}
RBF + Cste2 + White**	(2,197)	5.98×10^{-05}	5.18×10^{-03}	2.98×10^{-02}
Matern(ν=2.5) + Cste**2 + White	(2,197)	1.24×10^{-04}	4.96×10^{-04}	1.68×10^{-02}
Matern(ν=1.5) + Cste**2 + White	(2,197)	1.56×10^{-03}	1.01×10^{-02}	3.02×10^{-02}

Table 6: Error metrics of the GPR output $h(\boldsymbol{\alpha}(\mu))$ with respect to the target value $\boldsymbol{\beta}(\mu)$ for different kernels and $\mu \in \mathbb{P}_{h,\text{test}}$.

The value coefficients α_i are expected to be regular in Appendix B.
Since RBF = Matern($\nu = \infty$) where ν controls the regularity of Matern functions, it is no surprise that the best results are obtained with RBF kernel or Matern with large ν.
Following Table 6, it seems that the α_i, β_i are indeed regular and the real dimension of the coefficients manifold is $n = 2$.

We also note that more modes to learn ($\overline{n}$ increases) translates into more to learn GPR learning error, which is no surprise and should be fixed with more numerical samples.

In the following graphs, we compare the error of the POD $\|\boldsymbol{u}_{n_d} - \boldsymbol{u}_{\text{POD}}\|_X = \|\boldsymbol{\eta}_{\text{POD}}\|_X$, the error of the ML constructed function $\|\boldsymbol{u}_{n_d} - \boldsymbol{u}_N\|_X$ and the learning error $\|\boldsymbol{\beta} - h(\boldsymbol{\alpha})\|_2$ for different trained GPR.
We use the Greedy quadratic error closure $\boldsymbol{u}_{n_d} - \boldsymbol{u}_{n,\text{POD}} = \boldsymbol{\eta}_{\text{GQuadMani}}$ proposed in [SP24; JAI+17] as a reference comparison for error modeling. We also verify Eq.(1.31) for $n = 2$, $\overline{n} = 123, 148$:

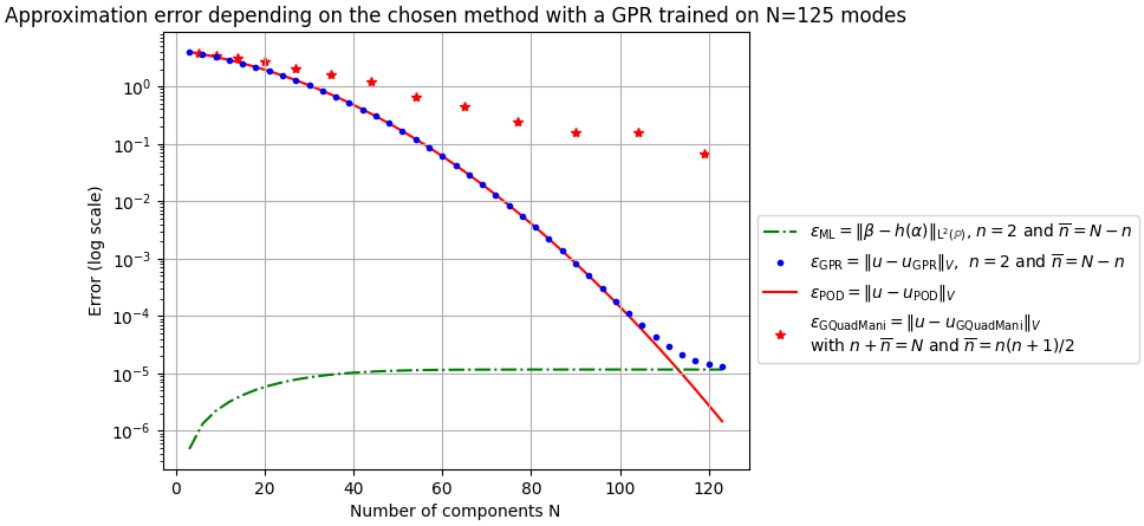


Figure 11: Error on reconstructing $\boldsymbol{u}_{n_d}$ with a GPR trained on $n = 2, \overline{n} = 125$.

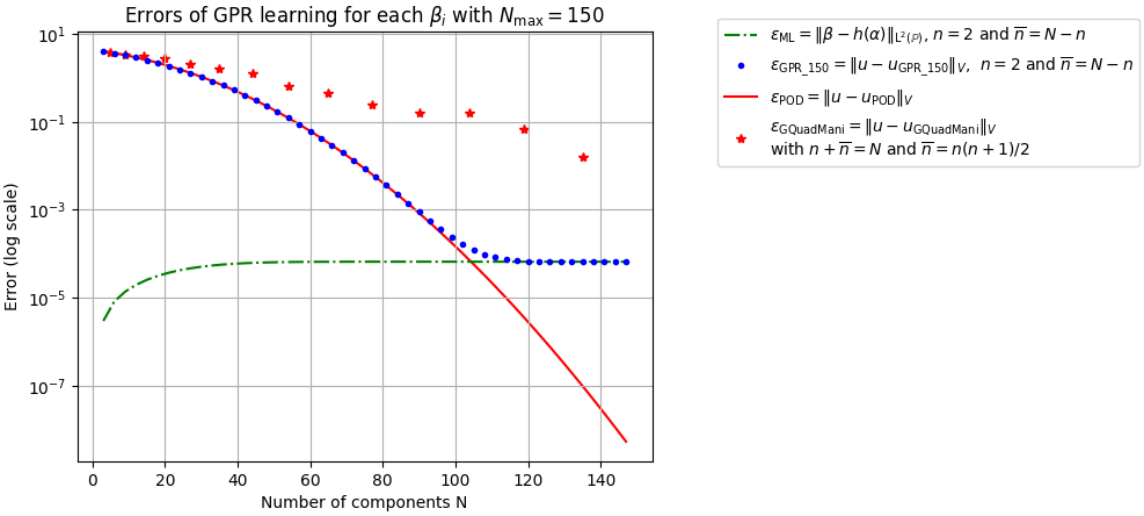


Figure 12: Error on reconstructing $\boldsymbol{u}_{n_d}$ with a GPR trained on $n = 2, \overline{n} = 150$.

The reconstruction error attains a plateau : this is due to the decay of the norms ($\|\beta_i\|_2 = \sigma_i$). Thus even if the relative error increases, the reconstruction error stays stable :

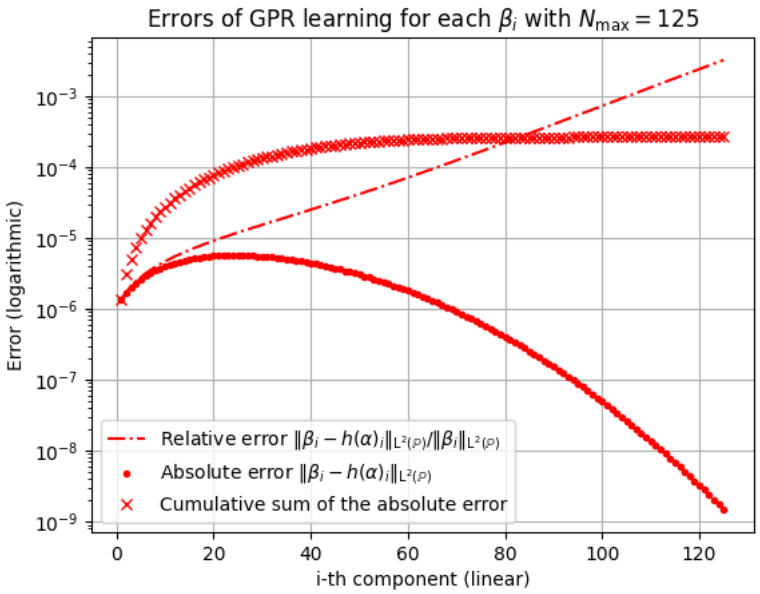


Figure 13: Different learning error metrics for a GPR trained with $n = 2$, $\overline{n} = 125$.

The relative error is linear in the loglog scale, translating the difficulty to learn the high frequencies of the POD coefficients with the low frequencies ones, see Figure 14 and Figure 15.

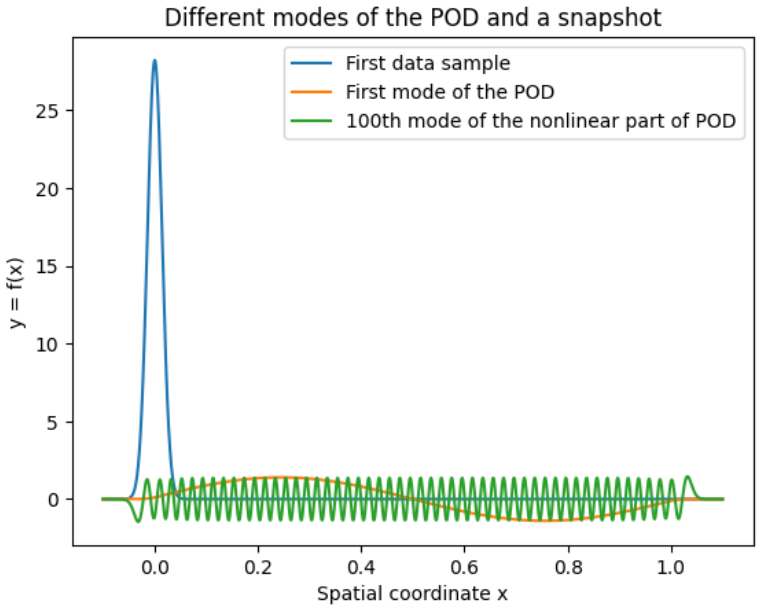


Figure 14: The first trajectory of the dataset, the first mode of the POD basis and the 100-th mode of the POD basis.

It is clear that the lower order modes are slowly oscillating while the higher order modes are fast oscillating corrections. Nyquist-Shannon sampling theorem states that we will not be able to recover highly oscillating coefficients by only sampling the first mode. Thus we expect $n > 1$.

Figure 15 shows that β_{100} is hard to describe only from α_1. The trajectory $\beta_{100}(\mu)$ seems to be in a 2D space as the GPR learning showcased.

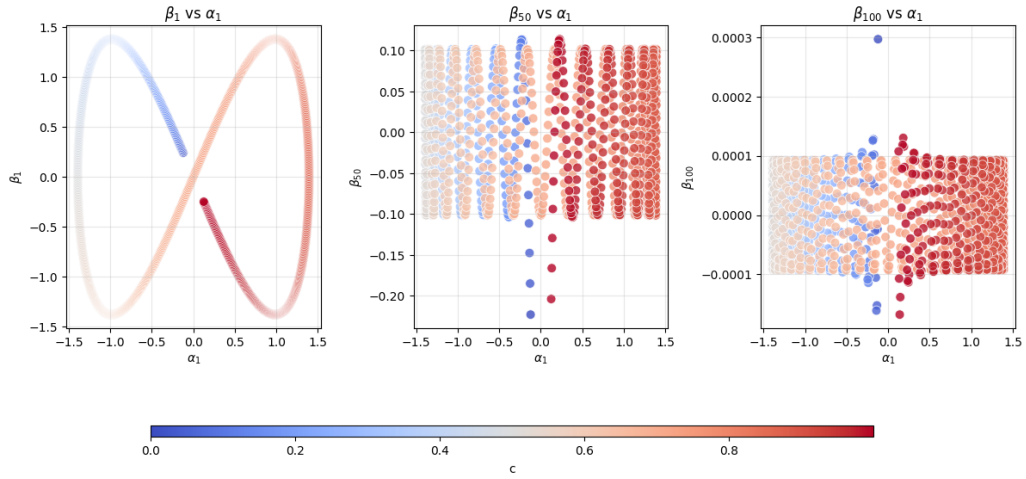


Figure 15: $\beta_i(\alpha_1)$ plotted as function of $\alpha_1(c)$ when the parameter $c \in \mathbb{P}$ varies and $i = 1, 50, 100$.

This can be verified plotting $\beta_{100} = \beta_{100}(\alpha_1, \alpha_2)$ in Figure 16 :

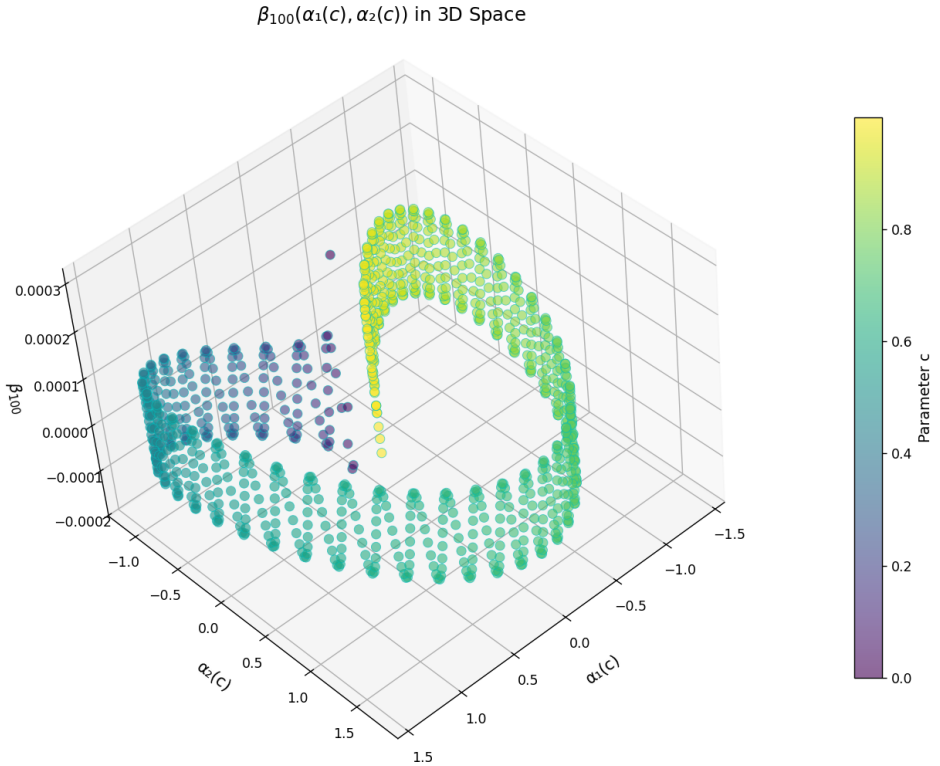


Figure 16: Coefficient $\beta_{100}(\alpha_1, \alpha_2)$ plotted for the different values of $\alpha_1(c), \alpha_2(c)$ when the parameter $c \in \mathbb{P}$ varies.

The results of learning of Table 6 are satisfying considering the measurements error level considered in industry range from $\sigma = 10^{-1}$ to $\sigma = 10^{-4}$ thus we need the ML model to allow representation up to a tolerance of 10^{-4}.

b) Online data assimilation

We consider $n = 2$ and $\overline{n} = 150$. The GPR Kernel is RBF(ℓ=0.0223) + 0.00188**2 + White(ϵ=1e-12).

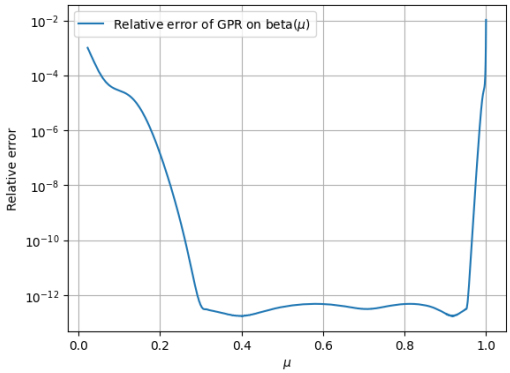
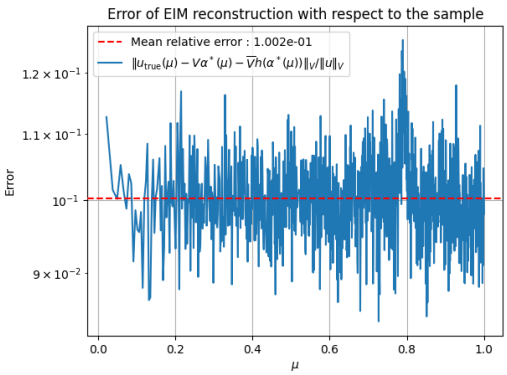
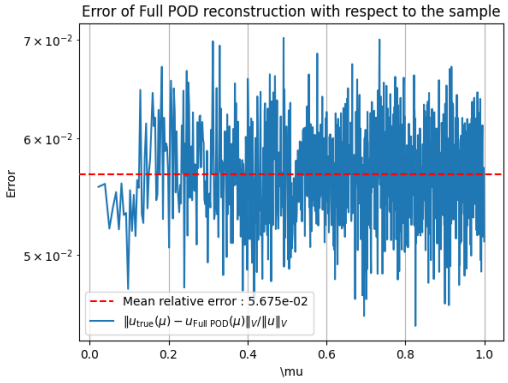


Figure 17: Relative error of the GPR over the samples $\mu \in \mathbb{P}_{h,\text{test}}$. Mean relative $\mathbf{L}^2$-error : 5.98×10^{-05}.

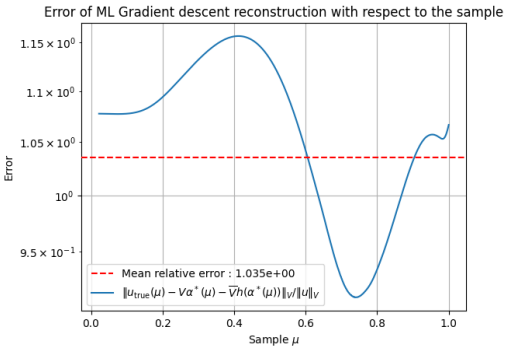
Figure 18 showcases the error on each samples for $\sigma = 10^{-1}$.



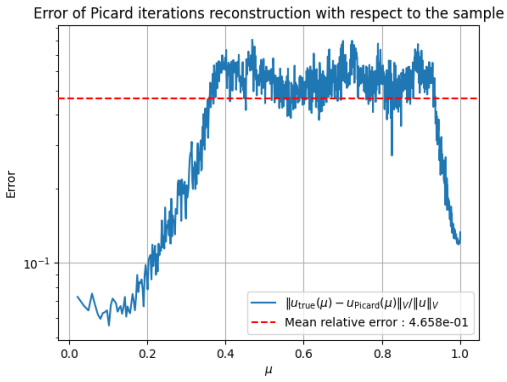
(a) Empirical Interpolation Method.



(b) Proper Orthogonal Decomposition.



(c) Gradient descent with Machine Learning.



(d) Picard iteration method.

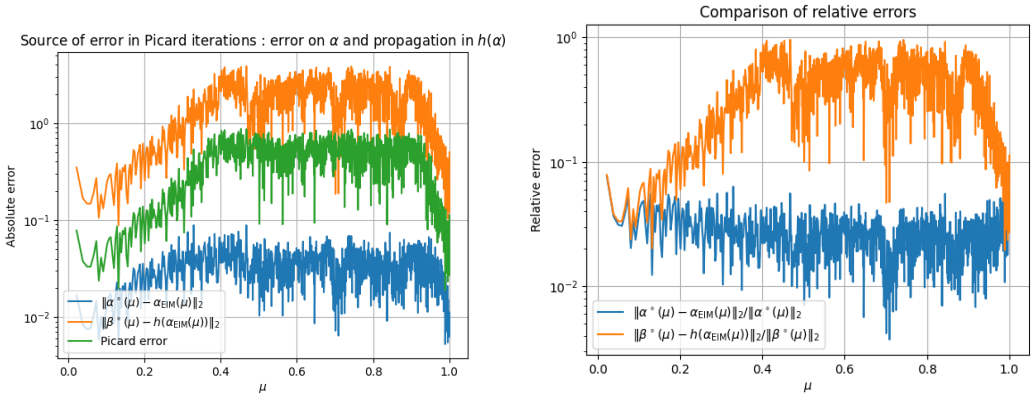
Figure 18: Error comparison for $\mu \in \mathbb{P}_{h,\text{test}}$, $m = N_{\text{modes}} = 2$, $\sigma = 10^{-1}$. Advection case.

For the error, a plain Least Square on the POD coefficients seems to perform the best, while Picard or Gradient Descent don't seem to perform better. This may be explain in the further discussion. Note that Picard iterations converge usually after two or three iterations at most.

Method	POD	Picard	EIM	Gradient Descent
Time (s/sample)	6.67×10^{-4}	1.63×10^{-3}	5.56×10^{-4}	1.31×10^{0}.
MRE	5.69×10^{-2}	4.27×10^{-1}	1.00×10^{-1}	1.035×10^{0}.

Table 7: Comparison of different methods for advection.

The time for POD and EIM are alike in term of computation time while Picard corresponds to a few EIM computation. This is no surprise since Picard is an iteration of Least Square + ML evaluation. The bottleneck is on the Gradient Descent : the GPflow implementation applies autodifferation on 150 independant GPR with a cost of evaluation of ~ 1 seconds per sample. This is not tractable.

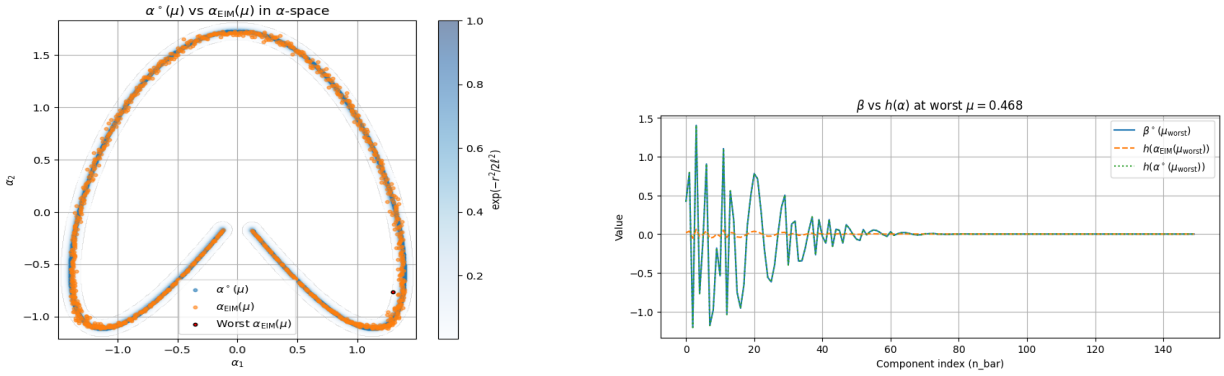


(a) Absolute error of EIM, Picard and GPR.

(b) Relative error of EIM and GPR.

Figure 19: $\mathbf{L}^2$-error for each samples of the GPR, EIM and Picard data assimilation method.

Figure 19 highlights the propagation of the error on $\boldsymbol{\alpha}^\circ$ from $\boldsymbol{\alpha}_{\text{EIM}}$ through the GPR model h.



(a) The trajectory of $\boldsymbol{\alpha}^\circ$ and the value of $\boldsymbol{\alpha}_{\text{EIM}}$ for the test samples. In blue, the output of $\exp(\|\boldsymbol{\alpha}^\circ - \boldsymbol{\alpha}\|_2^2 / 2\ell^2)$ where $\ell = 0.0223$ is the optimized GPR length scale.

(b) Value of the $\boldsymbol{\beta}_i^\circ$, $h(\boldsymbol{\alpha}^\circ)_i$ and $h(\boldsymbol{\alpha}_{\text{EIM}})_i$ for $i = 1, \ldots, 199$.

Figure 20: Graphic analysis of the GPR sensitivity to noise.

Figure 20 shows that a GPR trained on non noisy $\boldsymbol{\alpha}$ with high confidence may return a GPR with length scale ℓ too small compared to the scale of the noise. It results as a GPR returning almost null values for noisy observations!

Some conclusions can be drawn out of these results :

1. The GPR **must** be trained on a noisy dataset or with length scale adapted to the noise.
2. Considering more coefficients β_i to learn does not necessarily mean a better approximation.
3. The method of optimization must be **gradient-free** or parallel evaluations of the GPR should be used.
4. For $m = n_{\text{EIM}}$, the Picard fixed-point and Gradient Descent procedures did **NOT** improve much the accuracy with respect to the initialization, thus no need for online optimization after evaluation the first coefficients.

Answering the first conclusion will not be an issue and only needs a retraining of the GPR.
Answering the third conclusion has already being done with Picard iteration. We can also consider setting $m = N_{\text{EIM}}$ and consider, having evaluated $\boldsymbol{\alpha} \in \mathbb{R}^n$,

$$\tilde{\boldsymbol{u}} = \boldsymbol{u}_{\text{ref}} + \boldsymbol{V}\boldsymbol{\alpha} + \overline{\boldsymbol{V}} h(\boldsymbol{\alpha})$$

with no optimization process.
It is also possible to consider new maps $y \mapsto (\gamma_i)_{1\leq i\leq n}, (\hat{u}_i)_{1\leq i\leq n} \mapsto (\gamma_i)_{i>n}, (\hat{u}_i)_{i>n}$.
For the fourth conclusion, it is actually easier to only evaluate the mapping once instead of optimizing over it.
For the second conclusion, it is important to take into consideration that the number of POD modes N depends on the sampling i.e. $N = N_s(N_{\text{train}})$. Thus depending on the number of samples considered, the number of physical reduced basis modes could be reduced and the efforts could be concentrated into learning only meaningful coefficients.
It is clear that the lower order modes are slowly oscillating while the higher order modes are fast oscillating corrections. Nyquist-Shannon sampling theorem states that we will not be able to recover highly oscillating coefficients by only sampling the first mode. Thus we expect $n > 1$.

Conclusion

In this work, the encoding of reduced basis methods using Gaussian Process Regressors for data assimilation was studied.
It was demonstrated the capacity of Machine Learning to discover the latent space dimension of the reduced basis coefficients and the possibility to encode a full reduced basis into only a few modes, around the number of parameters.

Several data assimilation methods have been proposed to reconstruct the full field from partial observations. The Empirical Interpolation Method has been identified as a good initialization point when needed, and has also served as benchmark to compare the results.
Gradient methods were shown to slow down the data assimilation process for slight to no improvements while fixed-points methods or direct evaluation of the encoder could prove to beat the EIM.
It was also highlighted that if the latent dimension of the reduced basis was small enough with respect to the number of basis functions and the number of observations, the evaluation time are equivalent between EIM and the fixed-point method.

Last it is needed to construct the encoder with scales comparable to the relative noise scale and not enough to just learn the exact mapping.
Sampling considerations were done and need to be further studied in order to improve the selection of the reduced basis modes, thus the final reconstruction precision.

Some perspectives consist of focusing on the different mappings learning from the observation to the reconstructed fields by learning e.g. the map from EIM coefficients to POD, or EIM to the difference between EIM and POD coefficients and the last POD coefficients.
These could be used to reconstruct the field without minimization on α but just reconstructing the full field directly from the first coefficients and the evaluation of the ML model.

Part 3
Appendix

A Error metrics

1) Parameter norm

Let $P \geq 1$ be the parameter dimension. Let $\mathbb{P} \subset \mathbb{R}^P$.
Let N_s be the number of samples used to represent the parameter space $\mathbb{P}$.
Let $n \geq 1$ be the dimension of an output of interest. Let $g : \boldsymbol{\mu} \in \mathbb{P} \longmapsto g(\boldsymbol{\mu}) \in \mathbb{R}^n$ be the output function. For $\boldsymbol{\mu} \in \mathbb{P}$, the 2-norm of $g(\boldsymbol{\mu}) \in \mathbb{R}^n$ is :

$$\|g(\boldsymbol{\mu})\|^2_{2,\mathbb{R}^n} = \sum_{i=1}^{n} |\ (g(\boldsymbol{\mu}))_i\ |^2 .$$

Following [QSS06, Section 9.1], let $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_{N_s}$ be quadrature points in $\mathbb{P}$ and let $w_1, ..., w_{N_s} > 0$ be quadrature weights.
The discrete $\mathbf{L}^2(\mathbb{P})$ norm is :

$$\|g(\boldsymbol{\mu})\|^2_{\mathbf{L}^2(\mathbb{P}),h} = \sum_{i=1}^{N_s} w_i \|g(\boldsymbol{\mu}_i)\|^2_{2,\mathbb{R}^n} . \tag{3.1}$$

When the quadrature is uniform, this reduce to $w_i = \frac{1}{N_s}$:

$$\|g(\boldsymbol{\mu})\|^2_{\mathbf{L}^2(\mathbb{P}),h} = \frac{1}{N_s} \sum_{i=1}^{N_s} \|g(\boldsymbol{\mu}_i)\|^2_{2,\mathbb{R}^n} .$$

For $\boldsymbol{\mu} \in \mathbb{P}$, the supremum norm of $g(\boldsymbol{\mu}) \in \mathbb{R}^n$ is :

$$\|g(\boldsymbol{\mu})\|_{\infty,\mathbb{R}^n} = \max_{1 \leq i \leq n} |g(\boldsymbol{\mu})| .$$

The supremum norm over the parameters is defined as :

$$\|g(\boldsymbol{\mu})\|_{\mathbf{L}^\infty(\mathbb{P})} = \max_{\boldsymbol{\mu}_i \in \mathbb{P}_h} \|g(\boldsymbol{\mu}_i)\|_{\infty,\mathbb{R}^n} . \tag{3.2}$$

2) Timespace norm

a) Spatial norm

Let $d \geq 1$ be the space dimension. Let $\Omega \subset \mathbb{R}^d$ be an open domain. Let $n_d \geq 1$ be the number of nodes discretizing Ω. Let $V \subset \mathbf{L}^2(\Omega, \mathbb{R}^{n_d})$ be an Hilbert.
Let $\boldsymbol{M}_\delta$ be a mass matrix that can be constructed either by quadrature points over the spatial domain Ω or the basis functions of the Finite Elements space.
If $\boldsymbol{u}_{n_d} \in V_h$, the 2-norm is defined as :

$$\|\boldsymbol{u}_{n_d}\|^2_{2,\mathbb{R}^{n_d}} = \sum_{i=1}^{n_d} |(\boldsymbol{u}_{n_d})_i|^2 .$$

If $\boldsymbol{u}_{n_d} \in V_h$, the energy norm or V-norm is defined as :

$$\|\boldsymbol{u}_{n_d}\|^2_V = \left\|\boldsymbol{M}_\delta^{1/2} \boldsymbol{u}_{n_d}\right\|^2_{2,\mathbb{R}^{n_d}} = \sum_{i=1}^{n_d} \left|(\boldsymbol{M}_\delta^{1/2} \boldsymbol{u}_{n_d})_i\right|^2 . \tag{3.3}$$

The supremum norm of $\boldsymbol{u}_{n_d} \in \mathbb{R}^{n_d}$ is:

$$\|\boldsymbol{u}_{n_d}\|_{\infty,\mathbb{R}^{n_d}} = \max_{1 \leq i \leq n_d} |(\boldsymbol{u}_{n_d})_i| . \tag{3.4}$$

b) Timespace norm

Let $d \geq 1$ be the space dimension. Let $\Omega \subset \mathbb{R}^d$ be an open domain. Let $n_d \geq 1$ be the number of nodes discretizing Ω.
Let $T > 0$, let $I = [0; T]$ be a time interval. Let $V \subset \mathbf{L}^2(\Omega, \mathbb{R}^{n_d})$ be an Hilbert.
Let $\boldsymbol{M}_\delta$ be a mass matrix that can be constructed either by quadrature points over the spatial domain Ω or by the L^2-innerproduct of the Finite Elements basis functions.
If $\boldsymbol{u}_{n_d} \in \mathbf{L}^2(I, V)$, the energy norm or V-norm is defined as :

$$\|\boldsymbol{u}_{n_d}\|_V^2 = \int_I \left\| \boldsymbol{M}_\delta^{1/2} \boldsymbol{u}_{n_d}(t) \right\|_{2,\mathbb{R}^{n_d}}^2 \, dt. \tag{3.5}$$

If I is uniformly discretized in $N_{\text{timesteps}} \geq 1$, $0 = t_0 < t_1 < ... < t_{N_{\text{timesteps}}} = T$, this reduce to :

$$\|\boldsymbol{u}_{n_d}\|_V^2 = \frac{1}{N_{\text{timesteps}} + 1} \sum_{i=0}^{N_{\text{timesteps}}} \left\| \boldsymbol{M}_\delta^{1/2} \boldsymbol{u}(t_i) \right\|_{2,\mathbb{R}^{n_d}}^2 . \tag{3.6}$$

The supremum norm of $\boldsymbol{u}_{n_d} \in \mathbf{L}^2(I, \mathbb{R}^{n_d})$ is :

$$\|\boldsymbol{u}_{n_d}\|_{\infty,\mathbf{L}^2(I,\mathbb{R}^{n_d})} = \max_{1 \leq i \leq N_{\text{timesteps}}} \|\boldsymbol{u}_{n_d}(t_i)\|_{\infty,\mathbb{R}^{n_d}} . \tag{3.7}$$

3) Error metrics

Let $(E, \|\cdot\|_E)$ be a normed space. Let $y \in E$ be a target value. Let $v \in E$ be an approximation of y.
The error in norm $\|\cdot\|_E$ between y and v is :

$$\mathrm{Err}_E(v, y) := \|y - v\|_E . \tag{3.8}$$

The relative error in norm $\|\cdot\|_E$ between y and v is :

$$\mathrm{RelErr}_E(v, y) := \begin{cases} \|y - v\|_E \,/\, \|y\|_E & \text{if } y \neq 0, \\ \|v\|_E & \text{if } y = 0. \end{cases} \tag{3.9}$$

Let $(y_i)_{1 \leq i \leq N_s} \in (R^n)^{N_s}$ is a sample of N_s target in R^n and $(v_i)_{1 \leq i \leq N_s}$ is a sample of N_s approximations, let $w_i > 0$ be quadrature weights.
The Mean Relative Error (RME) is :

$$\mathrm{MRE}_E(\boldsymbol{y}, \boldsymbol{v}) := \sum_{i=1}^{N_s} w_i \mathrm{RelErr}_E(v_i, y_i) / \sum_i w_i. \tag{3.10}$$

In the uniform sampling case, this reduce to :

$$\mathrm{MRE}_E(\boldsymbol{y}, \boldsymbol{v}) := \frac{1}{N_s} \sum_{i=1}^{N_s} \mathrm{RelErr}_E(v_i, y_i). \tag{3.11}$$

The Root Mean Square Error (RMSE) is their average $\mathbf{L}^2$-norm over the sampling :

$$\mathrm{RMSE}_{2,\mathbb{R}^n}(\boldsymbol{y}, \boldsymbol{v}) := \sqrt{\sum_{i=1}^{N_s} w_i \|y_i - v_i\|_{2,\mathbb{R}^n}^2 / \sum_i w_i}. \tag{3.12}$$

In the uniform sampling case, this reduce to :

$$\mathrm{RMSE}_{2,\mathbb{R}^n}(\boldsymbol{y}, \boldsymbol{v}) := \sqrt{\frac{1}{N_s} \sum_{i=1}^{N_s} \|y_i - v_i\|_{2,\mathbb{R}^n}^2}.$$

B Test cases construction

1) Orthogonal modes

Let $I = [0; \pi]$, $\mathbb{P} = [0; 1]$, $\lambda > 0$.
We consider the function parameterized by $\mu \in \mathbb{P}$:

$$u(x\ ; \mu) = \alpha_1(\mu)\Phi_1(x) + \lambda\alpha_2(\mu)\Phi_2(x). \tag{3.13}$$

Where we construct Φ_1, Φ_2 and α_1, α_2 such that :

$$\begin{aligned} \langle \Phi_i | \Phi_j \rangle_{\mathrm{L}^2(I)} &= \delta_{ij}, \\ \langle \alpha_i | \alpha_j \rangle_{\mathrm{L}^2(\mathbb{P})} &= \delta_{ij}, \\ \textstyle\int_{\mathbb{P}} \alpha_1 = \int_{\mathbb{P}} \alpha_2 &= 0. \end{aligned} \tag{3.14}$$

We want to force $\Phi_1 = \sin / \left\|\sin\right\|_{\mathrm{L}^2(I)}$ and $\Phi_2 = \cos / \left\|\cos\right\|_{\mathrm{L}^2(I)}$.
To force $\Phi_2 = \sqrt{\frac{2}{\pi}}\cos$ to be the 2^{nd} POD mode, we set $\lambda = 0.1$ such that cos will represent exactly $1/100$ of the total energy.
Note that

$$\underset{\varphi \in \mathrm{Span}(\Phi_1, \Phi_2)}{\operatorname{argmax}} \left\|\varphi\right\|_{\mathrm{L}^\infty(\Omega)} = \{\Phi_1, \Phi_2\}$$

and

$$\underset{x \in [0;\pi]}{\operatorname{argmax}} \left|\Phi_1(x)\right| = \left\{\frac{\pi}{2}\right\}, \quad \underset{x \in [0;\pi]}{\operatorname{argmax}} \left|\Phi_2\right| = \{0, \pi\}.$$

Thus expect the EIM to select $x_1 \in \left\{0, \frac{\pi}{2}, \pi\right\}$ as first sensor locations, then choose $x_2 = \frac{\pi}{2}$ if $x_1 \neq \frac{\pi}{2}$ or $x_2 \in \{0, \pi\}$ if $x_1 = \frac{\pi}{2}$.

2) Accelerating advection of a gaussian bump

Let $I = [-0.1; 1.1] = [a; b]$, $\mathbb{P} = [0; 1]$ $(P = 1)$. Let $s > 0$.
We consider an accelerating advection of a gaussian bump :

$$u(x\ ; \mu) = \frac{1}{(2\pi s^2)^{1/2}} \exp\left(-\frac{(x - \mu^2)^2}{2s^2}\right). \tag{3.15}$$

We choose $s^2 = 0.01$ such that $u(x = -0.1\ ; \mu = 0) = u(x = 1.1\ ; \mu = 1) \simeq 0$.
We expand the square to separate the coefficient from the spatial parameter :

$$u(x\ ; \mu) = \frac{1}{(2\pi s^2)^{1/2}} \exp\left(\frac{-x^2}{2s^2}\right) \exp\left(\frac{x\mu}{s^2}\right) \exp\left(\frac{-\mu^4}{2s^2}\right).$$

The middle part can be written with the power serie of exponential :

$$u(x\ ; \mu) = \sum_{n=0}^{\infty} e^{\frac{-\mu^4}{2s^2}} \frac{\mu^{2n} x^n}{s^{2n} n!} e^{\frac{-x^2}{2s^2}} = \sum_{n=0}^{\infty} \alpha_n(\mu) q_n(x).$$

The family of functions $q_n(x) := x^n e^{\frac{-x^2}{2s^2}}$ is linearly independent. Thus $(q_n)_n$ is an infinite dimensional family of $\mathrm{L}^2(I)$. Hence the function u can not be represented by a finite dimensional basis, otherwise there would exist a finite dimensional basis of $\mathrm{L}^2(I)$ that contains an infinite dimensional basis of $\mathrm{L}^2(I)$. We conclude that the numerical POD basis can have infinite dimension up to numerical precision.

3) Burgers equation

This part collects an analysis of Burgers equation made during the internship that we did not have time to try regarding the difficulties encountered on the advection case.

a) The continuous problem

Let $\Omega =]0;1[^2$ be a space domain. Let $I = [0;1]$ be a time interval. Let $\nu \geq 0$ be a viscosity parameter. Let $\delta_1, \delta_2 > 0$ be small enough to respect Eq.(3.35), $\mathbb{P} \subset [1-\delta_1; 1+\delta_1] \times [0.1-\delta_2; 0.1+\delta_2]$. Let $\boldsymbol{\mu} = (a, s) \in \mathbb{P}$.
We note the initial condition u_0 as :

$$u_0(\boldsymbol{x}\ ;\boldsymbol{\mu}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x_1-a)^2+(x_2-a)^2}{2s^2}\right).$$

We consider the solution of the Burgers equation :

$$\text{Find } u(\cdot,\cdot\ ;\boldsymbol{\mu}) \text{ s.t. } \begin{cases} \dfrac{\partial u}{\partial t} + \operatorname{div}\left(\dfrac{u^2}{2}\begin{pmatrix}1\\1\end{pmatrix}\right) - \nu\Delta u &= 0 & \text{in } \Omega, \\ u(\boldsymbol{x},0) &= u_0(\boldsymbol{x}\ ;\boldsymbol{\mu}) & \text{in } \Omega, \\ u(\boldsymbol{x},t) &= 0 & \text{on } \partial\Omega. \end{cases} \tag{3.16}$$

If $\nu = 0$, the expected solution is :

$$u(x,t) = u_0(x + u_0(x,y)t).$$

Since $u_0 > 0$ the solution is a advected gaussian bump along the line $\mathbb{R}\begin{bmatrix}1\\1\end{bmatrix}$ from $(x_0, y_0) = (a, a)$.
If $\nu > 0$, then the solution follow the same advection trajectory with dispersion of the energy.

b) Discrete equation

We rewrite this in variational form.
For $u \in \mathrm{L}^2(\mathbb{R}_+^*; \mathrm{H}_0^1(\Omega))$:

$$\begin{aligned} & \frac{\partial u}{\partial t} + \operatorname{div}\left(\frac{u^2}{2}\begin{pmatrix}1\\1\end{pmatrix}\right) - \nu\Delta u && = 0 \\ \iff\ & \forall v \in \mathrm{H}_0^1(\Omega), \left\langle \frac{\partial u}{\partial t} \middle| v \right\rangle_{\mathrm{L}^2(\Omega)} + \left\langle \operatorname{div}\left(\frac{u^2}{2}\begin{pmatrix}1\\1\end{pmatrix}\right) \middle| v \right\rangle_{\mathrm{L}^2(\Omega)} - \nu \left\langle \Delta u | v \right\rangle_{\mathrm{L}^2(\Omega)} && = 0 \\ \iff\ & \forall v \in \mathrm{H}_0^1(\Omega), \left\langle \frac{\partial u}{\partial t} \middle| v \right\rangle_{\mathrm{L}^2(\Omega)} - \left\langle \frac{u^2}{2}\begin{pmatrix}1\\1\end{pmatrix} \middle| \nabla v \right\rangle_{\mathbf{L}^2(\Omega)} + \nu \left\langle \nabla u | \nabla v \right\rangle_{\mathbf{L}^2(\Omega)} && = 0. \end{aligned}$$

We discretize in time with finite difference using implicit diffusion and semi-implicit advection schemes:

$$\begin{aligned} & \forall v \in \mathrm{H}_0^1(\Omega), \left\langle \frac{u^{n+1}-u^n}{dt} \middle| v \right\rangle_{\mathrm{L}^2(\Omega)} - \left\langle \frac{u^n u^{n+1}}{2}\begin{pmatrix}1\\1\end{pmatrix} \middle| \nabla v \right\rangle_{\mathbf{L}^2(\Omega)} + \nu \left\langle \nabla u^{n+1} \middle| \nabla v \right\rangle_{\mathbf{L}^2(\Omega)} = 0 \\ \iff\ & \forall v \in \mathrm{H}_0^1(\Omega), \left\langle \frac{u^{n+1}}{dt} \middle| v \right\rangle_{\mathrm{L}^2(\Omega)} + \nu \left\langle \nabla u^{n+1} \middle| \nabla v \right\rangle_{\mathbf{L}^2(\Omega)} = \left\langle \frac{u^n}{dt} \middle| v \right\rangle_{\mathrm{L}^2(\Omega)} + \left\langle \frac{u^n u^{n+1}}{2}\begin{pmatrix}1\\1\end{pmatrix} \middle| \nabla v \right\rangle_{\mathbf{L}^2(\Omega)} \end{aligned}$$

We then use the finite element approximation for a FEM basis $(\Phi_i)_i$ such that for all n, $u^n = \sum_i u_i^n \Phi_i$ and let $v = \Phi_j$:

$$\sum_i \left(\frac{u_i^{n+1}}{dt} \langle \Phi_i | \Phi_j \rangle_{\mathrm{L}^2(\Omega)} + u_i^{n+1} \nu \langle \nabla \Phi_i | \nabla \Phi_j \rangle_{\mathbf{L}^2(\Omega)} \right) \tag{3.17}$$

$$= \sum_i \frac{u_i^n}{dt} \langle \Phi_i | \Phi_j \rangle_{\mathrm{L}^2(\Omega)} + \left\langle \frac{(\sum_i u_i^n \Phi_i)(\sum_\ell u_\ell^{n+1} \Phi_i)}{2} \middle| \frac{\partial \Phi_j}{\partial x} + \frac{\partial \Phi_j}{\partial y} \right\rangle_{\mathrm{L}^2(\Omega)}. \tag{3.18}$$

This being true for all j we expand the last term as :

$$\frac{1}{2} \sum_{i\ell j} u_i^{n+1} u_\ell^n \left\langle \phi_i \phi_\ell \middle| \frac{\partial \Phi_j}{\partial x} + \frac{\partial \Phi_j}{\partial y} \right\rangle_{\mathrm{L}^2(\Omega)}. \tag{3.19}$$

We define :

$$\boldsymbol{C}_{i\ell j} = \frac{1}{2} \left\langle \phi_i \phi_\ell \middle| \frac{\partial \Phi_j}{\partial x} + \frac{\partial \Phi_j}{\partial y} \right\rangle_{\mathrm{L}^2(\Omega)}, \tag{3.20}$$

$$\boldsymbol{N}(\boldsymbol{U}^n)_{\ell j} = \sum_i \boldsymbol{C}_{i\ell j} u_i^n. \tag{3.21}$$

Thus the algebraic formulation :

$$\left(\frac{1}{dt} \boldsymbol{M}_\delta + \nu \boldsymbol{K}_\delta + \boldsymbol{N}(\boldsymbol{U}^n) \right) \boldsymbol{U}^{n+1} = \frac{1}{dt} \boldsymbol{M}_\delta \boldsymbol{U}^n. \tag{3.22}$$

$$\tag{3.23}$$

We replace now $\boldsymbol{u}^n$ by its approximation $\overline{\boldsymbol{U}} + \boldsymbol{\Phi}\boldsymbol{\alpha}^n$:

$$\left(\frac{1}{dt} \boldsymbol{M}_\delta + \nu \boldsymbol{K}_\delta + \boldsymbol{N}(\overline{\boldsymbol{U}} + \boldsymbol{\Phi}\boldsymbol{\alpha}^n) \right) (\overline{\boldsymbol{U}} + \boldsymbol{\Phi}\boldsymbol{\alpha}^{n+1}) = \frac{1}{dt} \boldsymbol{M}_\delta (\overline{\boldsymbol{U}} + \boldsymbol{\Phi}\boldsymbol{\alpha}^n) \tag{3.24}$$

$$\iff \left(\frac{1}{dt} \boldsymbol{M}_\delta + \nu \boldsymbol{K}_\delta + \boldsymbol{N}(\overline{\boldsymbol{U}} + \boldsymbol{\Phi}\boldsymbol{\alpha}^n) \right) \boldsymbol{\Phi}\boldsymbol{\alpha}^{n+1} = \frac{1}{dt} \boldsymbol{M}_\delta \boldsymbol{\Phi}\boldsymbol{\alpha}^n - \left(\nu \boldsymbol{K}_\delta - \boldsymbol{N}(\overline{\boldsymbol{U}} + \boldsymbol{\Phi}\boldsymbol{\alpha}^n) \right) \overline{\boldsymbol{U}}. \tag{3.25}$$

$$\iff \left(\frac{1}{dt} \boldsymbol{\Phi}^T \boldsymbol{M}_\delta + \nu \boldsymbol{K}_\delta + \boldsymbol{N}(\overline{\boldsymbol{U}} + \boldsymbol{\Phi}\boldsymbol{\alpha}^n) \right) \boldsymbol{\Phi}\boldsymbol{\alpha}^{n+1} = \frac{1}{dt} \boldsymbol{M}_\delta \boldsymbol{\Phi}\boldsymbol{\alpha}^n - \left(\nu \boldsymbol{K}_\delta - \boldsymbol{N}(\overline{\boldsymbol{U}} + \boldsymbol{\Phi}\boldsymbol{\alpha}^n) \right) \overline{\boldsymbol{U}}. \tag{3.26}$$

We use Eq.(3.26) as the algebraic formulation of the PDE solver.
This formulation is stable under a CFL condition :

$$\forall v \in \mathrm{H}_0^1(\Omega), \left\langle \frac{u^{n+1}}{dt} \middle| v \right\rangle_{\mathrm{L}^2(\Omega)} + \nu \left\langle \nabla u^{n+1} \middle| \nabla v \right\rangle_{\mathbf{L}^2(\Omega)} = \left\langle \frac{u^n}{dt} \middle| v \right\rangle_{\mathrm{L}^2(\Omega)} + \left\langle \frac{u^n u^{n+1}}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \middle| \nabla v \right\rangle_{\mathbf{L}^2(\Omega)} \tag{3.27}$$

$$v = u^{n+1} \implies \left\langle \frac{u^{n+1}}{dt} \middle| u^{n+1} \right\rangle_{\mathrm{L}^2(\Omega)} + \nu \left\langle \nabla u^{n+1} \middle| \nabla u^{n+1} \right\rangle_{\mathbf{L}^2(\Omega)} = \left\langle \frac{u^n}{dt} \middle| u^{n+1} \right\rangle_{\mathrm{L}^2(\Omega)} + \left\langle \frac{u^n u^{n+1}}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \middle| \nabla u^{n+1} \right\rangle_{\mathbf{L}^2(\Omega)} \tag{3.28}$$

$$\iff \left\langle \frac{u^{n+1} - u^n}{dt} \middle| u^{n+1} \right\rangle_{\mathrm{L}^2(\Omega)} + \nu \left\| \nabla u^{n+1} \right\|_{\mathbf{L}^2(\Omega)}^2 = \left\langle \frac{u^n u^{n+1}}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \middle| \nabla u^{n+1} \right\rangle_{\mathbf{L}^2(\Omega)}. \tag{3.29}$$

$$\tag{3.30}$$

We use the following equality :

$$\left\langle \frac{u^{n+1}-u^n}{dt} \middle| u^{n+1} \right\rangle_{\mathrm{L}^2(\Omega)} = \frac{1}{2dt}\left(\left\|u^{n+1}\right\|^2_{\mathrm{L}^2(\Omega)} - \left\|u^n\right\|^2_{\mathrm{L}^2(\Omega)} + \left\|u^{n+1}-u^n\right\|^2_{\mathrm{L}^2(\Omega)} \right).$$

It gives the following estimate :

$$\frac{1}{2dt}\left(\left\|u^{n+1}\right\|^2_{\mathrm{L}^2(\Omega)} - \left\|u^n\right\|^2_{\mathrm{L}^2(\Omega)} + \left\|u^{n+1}-u^n\right\|^2_{\mathrm{L}^2(\Omega)} \right) + \nu \left\|\nabla u^{n+1}\right\|^2_{\mathbf{L}^2(\Omega)} = \left\langle \frac{u^n u^{n+1}}{2} \begin{pmatrix}1\\1\end{pmatrix} \middle| \nabla u^{n+1} \right\rangle_{\mathbf{L}^2(\Omega)}. \tag{3.31}$$

Then we focus on the nonlinear term of the right-hand side. Suppose $u^n \in \mathrm{L}^\infty(\Omega)$:

$$\begin{aligned}
\left\langle \frac{u^n u^{n+1}}{2} \begin{pmatrix}1\\1\end{pmatrix} \middle| \nabla u^{n+1} \right\rangle_{\mathbf{L}^2(\Omega)} &= \left\langle \frac{u^n u^{n+1}}{2} \middle| \begin{pmatrix}1\\1\end{pmatrix}^T \cdot \nabla u^{n+1} \right\rangle_{\mathrm{L}^2(\Omega)} \\
&\leq \frac{1}{2}\left\|u^n\right\|_{\mathrm{L}^\infty(\Omega)} \left\|u^{n+1}\right\|_{\mathrm{L}^2(\Omega)} \underbrace{\left\| \begin{pmatrix}1\\1\end{pmatrix} \right\|_{\mathbb{R}^2,\infty}}_{=1} \left\|\nabla u^{n+1}\right\|_{\mathbf{L}^2(\Omega)}, \quad \text{Cauchy-Schwarz [BRE11]} \\
&\leq \left(\frac{1}{2\sqrt{\nu}} \left\|u^n\right\|_{\mathrm{L}^\infty(\Omega)} \left\|u^{n+1}\right\|_{\mathrm{L}^2(\Omega)} \right) \left(\sqrt{\nu} \left\|\nabla u^{n+1}\right\|_{\mathbf{L}^2(\Omega)} \right) \\
&\leq \frac{1}{4\nu} \left\|u^n\right\|^2_{\mathrm{L}^\infty(\Omega)} \left\|u^{n+1}\right\|^2_{\mathrm{L}^2(\Omega)} + \frac{1}{2}\nu \left\|\nabla u^{n+1}\right\|^2_{\mathbf{L}^2(\Omega)}, \quad \text{Young's inequality [BRE11].}
\end{aligned}$$

Thus we have the following inequality by inserting the last line to Eq.(3.31) :

$$\begin{aligned}
&\frac{1}{2dt}\left(\left\|u^{n+1}\right\|^2_{\mathrm{L}^2(\Omega)} - \left\|u^n\right\|^2_{\mathrm{L}^2(\Omega)} + \left\|u^{n+1}-u^n\right\|^2_{\mathrm{L}^2(\Omega)} \right) + \nu \left\|\nabla u^{n+1}\right\|^2_{\mathbf{L}^2(\Omega)} && (3.32) \\
&\qquad \leq \frac{1}{4\nu} \left\|u^n\right\|^2_{\mathrm{L}^\infty(\Omega)} \left\|u^{n+1}\right\|^2_{\mathrm{L}^2(\Omega)} + \frac{1}{2}\nu \left\|\nabla u^{n+1}\right\|^2_{\mathbf{L}^2(\Omega)}. && (3.33)
\end{aligned}$$

We rearrange the result :

$$\left\|u^{n+1}\right\|^2_{\mathrm{L}^2(\Omega)} \left(1 - \frac{dt}{2\nu}\left\|u^n\right\|^2_{\mathrm{L}^\infty(\Omega)} \right) + \underbrace{\frac{\nu}{2}\left\|\nabla u^{n+1}\right\|^2_{\mathbf{L}^2(\Omega)}}_{0\leq} \leq \left\|u^n\right\|^2_{\mathrm{L}^2(\Omega)} - \underbrace{\left\|u^{n+1}-u^n\right\|^2_{\mathrm{L}^2(\Omega)}}_{\leq 0}.$$

Thus we have the following estimate linking $\left\|u^n\right\|_{\mathrm{L}^2(\Omega)}$ and $\left\|u^{n+1}\right\|_{\mathrm{L}^2(\Omega)}$:

$$\left\|u^{n+1}\right\|^2_{\mathrm{L}^2(\Omega)} \left(1 - \frac{dt}{2\nu}\left\|u^n\right\|^2_{\mathrm{L}^\infty(\Omega)} \right) \leq \left\|u^n\right\|^2_{\mathrm{L}^2(\Omega)}. \tag{3.34}$$

This estimate prove the scheme is stable under the condition :

$$\begin{aligned}
& 0 \leq 1 - \frac{dt}{2\nu}\left\|u^n\right\|^2_{\mathrm{L}^\infty(\Omega)} \\
\iff & \frac{dt}{2\nu}\left\|u^n\right\|^2_{\mathrm{L}^\infty(\Omega)} \leq 1 \\
\Rightarrow & \frac{dt(dx)^2}{2\nu}\left\|u^n\right\|^2_{\mathrm{L}^2(\Omega)} \leq 1, \qquad \text{inverse discrete inequality [AG21].}
\end{aligned}$$

Note that if

$$\frac{dt}{2\nu}\left\|u^n\right\|^2_{\mathrm{L}^\infty(\Omega)} \leq 1,$$

then

$$\left\|u^{n+1}\right\|_{\mathrm{L}^2(\Omega)} \leq \left\|u^n\right\|_{\mathrm{L}^2(\Omega)}.$$

Thus the CFL condition in essential supremum norm $\|\cdot\|_{\mathrm{L}^\infty}$ is :

$$\frac{dt}{2\nu}\left\|u^0\right\|^2_{\mathrm{L}^\infty(\Omega)} \leq 1. \tag{3.35}$$

And the L^2 CFL condition is :

$$\frac{dt(dx)^2}{2\nu}\left\|u^0\right\|^2_{\mathrm{L}^2(\Omega)} \leq 1. \tag{3.36}$$

List of Tables

List of Figures

References

[AG21] ERN A. and J-L. GUERMOND. *Finite Elements I: Approximation and Interpolation*. Springer International Publishing, 2021. DOI: 10.1007/978-3-030-56341-7.

[AGH+26] J. AGHILI et al. "Nonlinear compressive reduced basis approximation : when Taylor meets Kolmogorov". In: (2026). DOI: 10.48550/ARXIV.2601.13712.

[AGU23] F. ARBES, C. GREIF, and K. URBAN. *The Kolmogorov N-width for linear transport: Exact representation and the influence of the data*. 2023. DOI: 10.48550/ARXIV.2305.00066.

[ARE+26] S. ARES DE PARGA et al. "Nonlinear projection-based model order reduction with machine learning regression for closure error modeling in the latent space". In: *Computer Methods in Applied Mechanics and Engineering* 448 (2026). DOI: 10.1016/j.cma.2025.118443.

[BAR+04] M. BARRAULT et al. "An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations". In: *Comptes Rendus. Mathématique* 339 (2004). ISSN: 1778-3569. DOI: 10.1016/j.crma.2004.08.006.

[BB88] JONATHAN BARZILAI and JONATHAN M. BORWEIN. "Two-Point Step Size Gradient Methods". In: *IMA Journal of Numerical Analysis* (1988). ISSN: 1464-3642. DOI: 10.1093/imanum/8.1.141.

[BAY+18] A. G. BAYDIN et al. "Automatic Differentiation in Machine Learning: a Survey". In: *Journal of Machine Learning Research* 18.153 (2018), pp. 1–43. URL: http://jmlr.org/papers/v18/17-468.html.

[BNS25] A. BENSALAH, A. NOUY, and J. SOFFO. *Nonlinear manifold approximation using compositional polynomial networks*. 2025. DOI: 10.48550/ARXIV.2502.05088.

[BHK20] A. BLUM, J. HOPCROFT, and R. KANNAN. *Foundations of Data Science*. Cambridge University Press, 2020.

[BRE11] H. BREZIS. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Springer New York, 2011. DOI: 10.1007/978-0-387-70914-7.

[BUL12] V. BULJAK. *Inverse Analyses with Model Reduction*. Springer Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-22703-5.

[CEA64] J. CEA. "Approximation variationnelle des problèmes aux limites". In: *Annales de l'Institut Fourier* (1964). DOI: 10.5802/aif.181.

[COL12] R. COLEMAN. *Calculus on Normed Vector Spaces*. Springer New York, 2012. DOI: 10.1007/978-1-4614-3894-6.

[DE12] Daniele Antonio DI PIETRO and Alexandre ERN. *Mathematical Aspects of Discontinuous Galerkin Methods*. Springer, 2012. DOI: 10.1007/978-3-642-22980-0.

[EG04] A. ERN and J-L. GUERMOND. *Theory and Practice of Finite Elements*. 2004. DOI: 10.1007/978-1-4757-4355-5.

[FIL24] F. FILBERT. *Analyse numérique : algorithme et étude mathématique*. 2nd edition. Dunod, 2024. ISBN: 9782100873791.

[GON+19] H. GONG et al. "PBDW method for state estimation: error analysis for noisy data and nonlinear formulation". In: 2019. DOI: https://doi.org/10.48550/arXiv.1906.00810.

[GU19] C. GREIF and K. URBAN. *Decay of the Kolmogorov N-width for wave problems*. 2019. DOI: 10.48550/ARXIV.1903.08488.

[HRS16] J. S. HESTHAVEN, G. ROZZA, and B. STAMM. *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer International Publishing, 2016. DOI: 10.1007/978-3-319-22470-1.

[HJ12] R. A. HORN and C. R. JOHNSON. *Matrix Analysis*. Cambridge University Press, 2012. DOI: 10.1017/cbo9781139020411.

[JAI+17] S. JAIN et al. "A quadratic manifold for model order reduction of nonlinear structural dynamics". In: *Computers & Structures* 188 (Aug. 2017), pp. 80–94. ISSN: 0045-7949. DOI: 10.1016/j.compstruc.2017.04.005.

[KOL36] A. KOLMOGOROFF. "Über die beste Annäherung von Funktionen einer gegebenen Funktionenklasse". In: *Annals of Mathematics* 37 (1936), pp. 107–110.

[KYT96] F. K. KYTHE. *Fundamental Solutions for Differential Operators and Applications*. Birkhäuser Boston, 1996. DOI: 10.1007/978-1-4612-4106-5.

[LAN96] C. LANCZOS. *Linear Differential Operators*. Society for Industrial and Applied Mathematics, 1996. DOI: 10.1137/1.9781611971187.

[LE 13] H. LE DRET. *Équations aux dérivées partielles elliptiques non linéaires*. Springer Berlin Heidelberg, 2013. DOI: 10.1007/978-3-642-36175-3.

[LIA+02] Y.C. LIANG et al. "Proper Orthogonal Decomposition and Its Applications—Part I: Theory". In: *Journal of Sound and Vibration* (2002). DOI: 10.1006/jsvi.2001.4041.

[MPT02] Y. MADAY, A. T. PATERA, and G. TURINICI. "Global a priori convergence theory for reduced-basis approximations of single-parameter symmetric coercive elliptic partial differential equations". In: *Comptes Rendus. Mathématique* (2002). DOI: 10.1016/s1631-073x(02)02466-4.

[MAD+15] Y. MADAY et al. "PBDW State Estimation: Noisy Observations; Configuration-Adaptive Background Spaces; Physical Interpretations". In: *ESAIM: Proceedings and Surveys* (2015). DOI: 10.1051/proc/201550008.

[MM13] Yvon Maday and Olga Mula. "A Generalized Empirical Interpolation Method: Application of Reduced Basis Techniques to Data Assimilation". In: *Analysis and Numerics of Partial Differential Equations*. Springer Milan, 2013, pp. 221–235. ISBN: 9788847025929. DOI: 10.1007/978-88-470-2592-9_13.

[MS20] C. MARCATI and C. SCHWAB. "Analytic regularity for the incompressible Navier-Stokes equations in polygons". In: (2020). DOI: 10.48550/ARXIV.2004.11264.

[MEN90] B MENDELSON. *Introduction to Topology*. 3rd ed. Dover Publications, 1990. ISBN: 978-0486663524.

[NEI10] R. D. NEIDINGER. "Introduction to Automatic Differentiation and MATLAB Object-Oriented Programming". In: *SIAM Review* 52 (2010). DOI: 10.1137/080743627.

[NIU+25] T. NIU et al. "Machine Learning-Based Bias Correction for the GEIM Model". In: *Numerical Mathematics: Theory, Methods and Applications* (Oct. 2025). ISSN: 1004-8979. DOI: 10.4208/nmtma.oa-2025-0025.

[OS16] M. OHLBERGER and RAVE S. "Reduced Basis Methods: Success, Limitations and Future Challenges". In: (2016). v2. URL: https://arxiv.org/abs/1511.02021v2.

[PD20] B. PEHERSTORFER and S. DRMAC Z.and GUGERCIN. "Stability of Discrete Empirical Interpolation and Gappy Proper Orthogonal Decomposition with Randomized and Deterministic Sampling Points". In: *SIAM Journal on Scientific Computing* (2020). DOI: 10.1137/19m1307391.

[QMF16] A. QUARTERONI, A. MANZONI, and N. FEDERICO. *Reduced Basis Methods for Partial Differential Equations: An Introduction*. MS&A – Modeling, Simulation and Applications. Springer, 2016. DOI: 10.1007/978-3-319-15431-2.

[QSS06] A. QUARTERONI, R. SACCO, and F. SALERI. *Numerical Mathematics*. 2nd ed. 2006. DOI: 10.1007/b98885.

[RW05] C. E. RASMUSSEN and C. K. I. WILLIAMS. *Gaussian Processes for Machine Learning*. The MIT Press, 2005. DOI: 10.7551/mitpress/3206.001.0001.

[RAU91] J. RAUCH. *Partial Differential Equations*. Springer New York, 1991. DOI: 10.1007/978-1-4612-0953-9.

[RHE88] W. C. RHEINBOLDT. "On the computation of multi-dimensional solution manifolds of parametrized equations". In: *Numerische Mathematik* (1988). DOI: 10.1007/bf01395883.

[SP24] P. SCHWERDTNER and B. PEHERSTORFER. *Greedy construction of quadratic manifolds for nonlinear dimensionality reduction and nonlinear model reduction*. 2024. DOI: 10.48550/ARXIV.2403.06732.

[SEN90] B. SENDOV. *Hausdorff Approximations*. Ed. by Gerald Beer. Springer Netherlands, 1990. DOI: 10.1007/978-94-009-0673-0.

[TAY91] M. E. TAYLOR. "Pseudodifferential Operators and Nonlinear PDE". In: (1991). DOI: 10.1007/978-1-4612-0431-2.