Stiefel manifold interpolation for non-intrusive model reduction of parameterized fluid flow problems

Achraf El Omari, Mohamed El Khlifi, Laurent Cordier

Highlights

Stiefel Manifold Interpolation for Non-Intrusive Model Reduction of Parameterized Fluid Flow Problems

Achraf El Omari, Mohamed El Khlifi, Laurent Cordier

- First application of interpolation on the tangent space of the Stiefel manifold to engineering data.
- Non-intrusive method for interpolating physical solutions at a new operating condition.
- Direct interpolation method for spatial and temporal modes. No post-interpolation operations required.

Stiefel Manifold Interpolation for Non-Intrusive Model Reduction of Parameterized Fluid Flow Problems

Achraf El Omari, Mohamed El Khlifi, Laurent Cordier*

[a]Hassan II University of Casablanca, Faculty of Sciences and Techniques, Mohammedia, Morocco
[b]Hassan II University of Casablanca, Faculty of Sciences and Techniques, Mohammedia, Morocco
[c]Institut Pprime, CNRS, Université de Poitiers, ISAE-ENSMA, UPR 3346, France

Abstract

Many engineering problems are parameterized. In order to minimize the computational cost necessary to evaluate a new operating point, the interpolation of singular matrices representing the data seems natural. Unfortunately, interpolating such data by conventional methods usually leads to unphysical solutions, as the data live on manifolds and not vector spaces. An alternative is to perform the interpolation in the tangent space to the Grassmann manifold to obtain interpolated spatial modes. Temporal modes are afterwards determined via the Galerkin projection of the high-fidelity model onto the interpolated spatial basis. This method, which is known for some fifteen years, is intrusive. Recently, Oulghelou and Allery (JCP, 2021) have proposed a non-intrusive approach (equation-free), but requiring the resolution of two low-dimensional optimization problems after interpolation. In this paper, a non-intrusive alternative based on Interpolation on the Tangent Space of the Stiefel Manifold (ITSSM) is presented. This approach

*Corresponding author

has the advantage of not requiring a calibration phase after interpolation. To assess the method, we compare our results with those obtained using global POD on the one hand, and two methods based on Grassmann interpolation on the other. These comparisons are performed for two classical configurations encountered in fluid dynamics. The first corresponds to the one-dimensional non-linear Burgers' equation. The second example is the two-dimensional cylinder wake flow. We show that the proposed strategy can accurately reconstruct the physical quantities associated with a new operating point. Moreover, the estimation is fast enough to allow real-time computation.
Keywords: Manifold interpolation, Stiefel manifold, non-intrusive model reduction, fluid dynamics, parameterized model reduction

1. Introduction

The vast majority of engineering problems encountered in automotive, aeronautical and biomedical applications are parametric in nature [1]. The configurations of interest depend on so-called *control parameters* such as Reynolds or Mach numbers, or more generally on geometric design parameters or external forces. To improve our physical understanding of these domains and/or to solve optimization problems, numerous numerical resolutions of *high-fidelity* models are required. Despite continuous progress in numerical methods, the resolutions of these systems, often consisting of Partial Differential Equations (PDEs) derived from first principles, are still too demanding in terms of computation. This is all the more true as applications become increasingly complex at the same time. This high computational cost is due to the huge number of degrees of freedom arising from the spatial discretisation of the models and to the small

time steps required to solve them accurately. To overcome these difficulties, Reduced-Order Models (ROMs) have been developed extensively over the last twenty years [e.g., 2]. The purpose of these simplified dynamical models is to represent the most important information content of the high-fidelity model in a low dimensional subspace. The final objective is to enable ROMs to operate in near real-time computations and to be integrated in optimal control procedures. ROMs are divided into two main families [3, Chap. 1]: *intrusive methods*, in which the reduced solutions are determined by solving dynamical models obtained by projecting the high-fidelity model onto the reduced space, and *non-intrusive methods*, in which only the data collected from numerical simulations or experimental measurements are considered. The main drawback of intrusive methods is that they require access to the high-fidelity model that produced the data.

Proper Orthogonal Decomposition (POD) is one of the most popular model reduction methods in fluid dynamics. Given a collection of high-dimensional input data, called snapshots, the POD algorithm [see 4, 5] determines an orthogonal basis whose elements are solutions of an integral eigenvalue problem [6]. By definition, these eigenfunctions, also called POD modes, are representative of the most dominant (in a given sense) realizations of the input data. Model reduction techniques generally lack robustness to parameter variation. To address this issue, a single global projection basis for the entire parametric domain is used successfully in Bergmann et al. [7] for optimizing an aerodynamic configuration. This global approach might however become computationally inefficient if each new parametric configuration is significantly different in terms of information content. In Bergmann and Cordier [8], a trust-region optimization algorithm is coupled to a POD ROM. Inside the trust-region, the POD model is used for

predicting the dynamics. When the optimization leads to parameters outside the domain of validity of the trust-region, the POD basis has to be determined again, and it is necessary to solve the high-fidelity model, which can be very expensive. Despite the interest of these methods, it seems relevant to seek to develop fast and robust algorithms for adapting a set of pre-computed POD bases to variations in parameter values.

From a mathematical point of view, POD subspaces can be interpreted as special cases of *matrix manifolds* [for the definition, see Chap. 3 of 9, or later in Section 3.1]. Matrix manifolds are characterized by some properties of linear algebra, such as orthogonality, full-column rank or positiveness. For instance, POD basis can be represented by an orthogonal, full-column rank matrix (see Section 2). The problem of interpolating POD bases can therefore be formally reduced to that of interpolating data on matrix manifolds. Since a manifold is a space of non-zero curvature, the direct interpolation of points contained on the manifold does not necessarily produce a point on the manifold (see Section 3). For example, the interpolation of orthogonal matrices does not necessarily produce an orthogonal matrix. To reproduce the dynamics of the system for an "unseen" configuration, Amsallem and Farhat [10] proposed, in their seminal work, to interpolate local basis on the *tangent space* to a matrix manifold (see Section 3.2). The algorithm can be summarized as follows. First, local bases determined by POD are mapped to the tangent space of a matrix manifold of choice (see the discussion in Section 3.3). Then, the mapped data is interpolated with a conventional interpolation algorithm. Finally, the interpolated result is mapped back to the manifold yielding the sought basis for the "unseen" parameter. Essentially, this algorithm exploits the fact that the tangent space of

a differential manifold is a "flat" Euclidean vector space in which interpolation can be performed in the usual way. Considering that the POD basis can be represented naturally on a Grassmann manifold (see Section 3.1 for the definition), Amsallem and Farhat [10] called their approach *Interpolation on the Tangent Space of the Grassmann Manifold* (ITSGM). This interpolated basis is then used in a classical Galerkin projection to predict the temporal dynamics of the system for unseen configuration. Amsallem et al. [11] extended their early work to the interpolation of the linear structural dynamics of reduced-order models. To take into account the physical constraints of the problem, they chose to perform the interpolation on the tangent space to the manifold of symmetric positive definite (SPD) matrices. The work of Amsallem and Farhat [10] has received a great deal of attention. As a consequence, interpolation on the Grassmann manifold was used repeatedly in the literature to study other engineering configurations.

In Oulghelou and Allery [12], the authors used interpolation on the tangent space of the Grassmann manifold to optimize by optimal control the flow in a lid-driven cavity at low Reynolds number. In Oulghelou and Allery [13] and Oulghelou and Allery [14], the same authors solved several 2D optimal control problems (Burgers, nonlinear reaction diffusion, nonlinear heat) using concurrently two interpolation methods: ITSGM and Proper Generalized Decomposition. More recently, Vlachas et al. [15] proposed an extension of interpolation on the tangent space of the Grassmann manifold, using a preprocessing that consists in partitioning the parameter space into several subregions, thus facilitating interpolation. These studies have shown that ITSGM is well suited to tackling parametric problems. However, in all the cited cases, the method is intrusive, the temporal modes being obtained by Galerkin projection. Naively, we might think

that it is possible to use the same method of interpolation on the tangent space of the Grassman manifold for the right singular matrix of the snapshot matrix. However, this is in principle impossible, as the ITSGM algorithm determines a solution up to right multiplication with an orthogonal matrix (this is referred to as an equivalence class, see Section 3.1). For the spatial modes determined by ITSGM and used in the Galerkin projection, this does not pose any difficulties, as we can show (Section 2.2) that the reduced-order model determined by Galerkin projection remains unchanged by using spatial POD modes known to within one orthogonal matrix. In that perspective, Oulghelou and Allery [16] recently proposed a non-intrusive two-step extension, referred as Bi-CITSGM (Bi-Calibrated ITSGM). In the first step, the left and right singular matrices obtained by Singular Value Decomposition of the snapshot matrices are interpolated by ITSGM and the singular value matrix are interpolated by spline cubic. The second step consists in solving two auxiliary optimization problems aiming to calibrate the interpolated modes by determining two small orthogonal matrices. This method was used successfully to estimate a wake flow past a circular cylinder at low Reynolds number (195).

In cases, often encountered in industrial applications, where commercial software is used in the numerical process, it is clearly not possible to use an intrusive method based on the integration of a reduced-order model derived from Galerkin projection. In this context, Friderikos et al. [17] recently proposed a non-intrusive method, based on a mechanism specific to the commercial finite element software used for simulation (Abaqus), for imposing internal constraints. In the same paper, the authors investigate the necessary stability conditions for applying POD bases interpolation on the tangent space of the Grassmann manifold. They

showed that these stability conditions help to explain some numerical difficulties encountered with Grassmann interpolation.

Recently, Oulghelou et al. [18] proposed a new interpolation strategy to accelerate updates of Galerkin projections. Instead of using the Grassmann manifold as in their previous work, the POD subspaces are represented[1] on the quotient space of the set of full-rank matrices by the orthogonal group. This approach exploits the reduced computational cost associated with the simple explicit expressions of the exponential and logarithmic mappings determined in Massart and Absil [19]. The parametric interpolation phase is performed using a weighted Riemannian barycenter method or Karcher barycenter (see Zimmermann and Bergmann [20] for instance). The advantage of this interpolation method is to do not require the introduction of a reference point in the tangent space. However, the interpolant is searched iteratively as the existence and uniqueness of the solution to the minimization problem associated with the Karcher barycenter can only be guaranteed locally. This interpolation step of the spatial POD subspace is followed by the resolution of a Galerkin system in order to evaluate the temporal dynamics. In a later work, Oulghelou et al. [21] opted to interpolate both the spatial and temporal bases without a calibration phase. According to the authors, calibration between the columns of the spatial and temporal bases is naturally ensured by barycentric interpolation, unlike interpolation on the Grassmann manifold where calibration is lost.

In this paper, an alternative non-intrusive interpolation method is assessed

[1] In contrast to the interpolation on the Stiefel manifold, the orthogonality of the matrix representing the POD subspace is not considered.

on configurations typically encountered in fluid mechanics. Inspired by previous studies, we propose an original method called Interpolation on the Tangent Space of the Stiefel Manifold (ITSSM). As it will be shown, the main advantage of this approach is that it does not require calibration phases, as interpolation can be performed directly on the left and right singular matrices. Interpolation on the Stiefel manifold has not been treated in the literature, except in a short 8-page article [22] in which the authors made two numerical experiments on a financial model called Black-Scholes equation, very far from those generally encountered in physics.

The first objective of this paper is to present in more depth the previous ITSSM method, by testing it on two examples issued from physical modelling in fluid mechanics (Burger's equation and wake flow downstream a circular cylinder). Another objective is to help popularize the use of interpolation on the tangent space of the Stiefel manifold. In Section 3.3, we will return to the advantages provided by the ITSSM method compared to the ITSGM algorithm used so far. As Zimmermann and Debrabant note in their article, it is necessary for performing interpolation on the Stiefel manifold to have efficient projection algorithms on the manifold and its tangent space. While an explicit formula existed for about twenty years for projecting to the Stiefel manifold a point belonging to its tangent space (the so-called exponential mapping, see Section 3.1 and Alg. 2), an efficient algorithm for calculating the inverse projection operator (denoted logarithm mapping, see Alg. 3) was developed very recently by Zimmermann [23].

The paper is organized as follows. In Section 2, we present POD-based reduced-order modelling. Section 3 begins with reminders of differential geome-

try, essential for understanding interpolation methods on tangent spaces of matrix manifold, then presents in detail the Stiefel and Grassmann manifolds. In Section 3.2, the general process of interpolation on a tangent space to a matrix manifold is reviewed. Subsequently, Section 3.3 briefly presents the advantages of using the Stiefel manifold for interpolation. Section 4 describes the interpolation algorithm on the tangent space to the Stiefel manifold, as well as its numerical complexity. Finally, in Section 5, the proposed method is assessed by comparing results with the global POD and two interpolation methods on the Grassmann manifold. Two test configurations of increasing complexity are analyzed. The first one corresponds to the one-dimensional non-linear Burgers' problem (Section 5.1), while the second one is the two-dimensional cylinder wake flow (Section 5.2). Conclusions are drawn in Section 6.

2. Reduced-order modelling based on Proper Orthogonal Decomposition

2.1. Proper Orthogonal Decomposition

Let $\mathcal{U} = \{\boldsymbol{u}(\boldsymbol{x}, t_m)\}_{m=1}^{N_t}$ be a set of N_t real[2] snapshots of size $N_{xc} = N_x\, N_c$ taken over a time interval $[0, T]$, with $\boldsymbol{x} \in \Omega$ the spatial domain, N_x the number of points of a spatial grid and N_c the dimension of the system state (for example, the three components of the fluid velocity). We introduce H a Hilbert space with inner product $(\cdot, \cdot)_H$ and induced norm $\|\cdot\|_H$. The aim of the POD [24, for instance] is to find a subspace $\mathcal{S}_k$ of dimension $k \ll N_t$, such that $\mathcal{S}_k =$

[2]Similarly, we could consider complex-valued snapshots. We would then have to modify accordingly the inner product.

span $(\boldsymbol{\Phi}_1, \cdots, \boldsymbol{\Phi}_k)$ where $\{\boldsymbol{\Phi}_i\}_{i=1}^k$ are solutions of the constrained optimization problem given by:

$$\max_{\Pi_{\mathsf{POD}}} \sum_{m=1}^{N_t} \|\Pi_{\mathsf{POD}} \boldsymbol{u}(\boldsymbol{x}, t_m)\|_H^2 \quad \text{s.t.} \quad \|\boldsymbol{\Phi}_k\|_H^2 = 1\,, \tag{1}$$

where Π_{POD} is the orthogonal projector on $\mathcal{S}_k$, *i.e.*

$$\Pi_{\mathsf{POD}} \boldsymbol{u}(\boldsymbol{x}, t_m) = \sum_{i=1}^{k} \left(\boldsymbol{u}(\boldsymbol{x}, t_m), \boldsymbol{\Phi}_i(\boldsymbol{x})\right)_H \boldsymbol{\Phi}_i(\boldsymbol{x})\,. \tag{2}$$

Using, for instance, a Lagrange multiplier technique, we can show that the solution of (1) is given by the following eigenvalue/vector problem:

$$\left(\tilde{S}\tilde{S}^{\mathsf{T}}\right) \tilde{\boldsymbol{\Phi}}_i = \lambda_i \tilde{\boldsymbol{\Phi}}_i, \quad i = 1, \cdots, k\,, \tag{3}$$

where $\tilde{S} = W^{1/2} S$, with S the snapshot matrix of size $N_{xc} \times N_t$, $W^{1/2}$ the square root of the weighting matrix W representing the inner product $(\cdot, \cdot)_H$ and $\tilde{\boldsymbol{\Phi}}_i = W^{1/2} \boldsymbol{\Phi}_i$. The matrix W is assumed symmetric, positive semidefinite. If H corresponds to $L^2(\Omega)$, then W is the mass matrix of the FEM. If H is the vector space of real-valued vectors of size N_{xc} ($\mathbb{R}^{N_{xc}}$) with the Euclidean inner product, then W is the identity matrix.

The solution of (3) can be determined via the singular value decomposition of $\tilde{S}$, *i.e.*

$$\tilde{S} = \tilde{U} \tilde{\Sigma} \tilde{V}^{\mathsf{T}}\,,$$

where $\tilde{U} \in \mathbb{R}^{N_{xc} \times N_{xc}}$ and $\tilde{V} \in \mathbb{R}^{N_t \times N_t}$ are the left and right singular matrix, respectively, and $\Sigma \in \mathbb{R}^{N_{xc} \times N_t}$ is the singular value matrix. Both $\tilde{U}$ and $\tilde{V}$ are orthogonal matrices: $\tilde{U}\tilde{U}^{\mathsf{T}} = \tilde{U}^{\mathsf{T}}\tilde{U} = I_{N_{xc}}$ and $\tilde{V}\tilde{V}^{\mathsf{T}} = \tilde{V}^{\mathsf{T}}\tilde{V} = I_{N_t}$. Finally, $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_p, 0, \cdots, 0)$ with $p = \min(N_{xc}, N_t)$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_p = 0$ where $r = \mathsf{rank}(\tilde{S}) \leq p$.

Precisely, we have span $\left(\tilde{\boldsymbol{\Phi}}_1, \cdots, \tilde{\boldsymbol{\Phi}}_k\right) = \tilde{U}_k$ where $\tilde{U}_k$ corresponds to the first k columns of $\tilde{U}$. POD therefore reduces to the singular value decomposition of a snapshot matrix, which is function of the chosen inner product. The orthogonal projector on $\mathcal{S}_k$ is given by $\Pi_{\mathsf{POD}} = \tilde{U}_k \tilde{U}_k^{\mathsf{T}}$. It is worth noting that if the matrix $\tilde{U}_k$ is known to within one orthogonal matrix Q, then the projector is not modified:

$$\Pi_{\mathsf{POD}} = \left(\tilde{U}_k Q\right)\left(\tilde{U}_k Q\right)^{\mathsf{T}} = \tilde{U}_k \underbrace{Q Q^{\mathsf{T}}}_{I} \tilde{U}_k^{\mathsf{T}} = \tilde{U}_k \tilde{U}_k^{\mathsf{T}} .$$

2.2. POD reduced-order model

The main aim of model reduction is to represent the high-fidelity dynamics derived from physical principles by a low-dimensional dynamical system consisting of a few degrees of freedom in effective interaction [6, for instance]. Applications are aimed at improving understanding of physical phenomena via simplified phenomenological models, or at parametric studies requiring numerous resolutions of the high-fidelity system, as in the case for continuation methods or optimization problems [25, and in many others]. Let's consider a spatio-temporal semi-discretized dynamical model given by

$$\frac{d\boldsymbol{v}(t;\boldsymbol{\lambda})}{dt} = \boldsymbol{f}\left(\boldsymbol{v}(t;\boldsymbol{\lambda})\right) , \tag{4}$$

where $\boldsymbol{v} \in \mathbb{R}^{N_{xc}}$ corresponds to the discretized – in space – state vector $\boldsymbol{u}$, $\boldsymbol{\lambda} \in \mathbb{R}^{N_\lambda}$ to the parameters of the model, and $\boldsymbol{f}$ to the full-order dynamics discretized in space. Following the results presented in Section 2.1, the full state vector $\boldsymbol{v}(t)$ can be approximated as

$$\boldsymbol{v}(t) \approx \tilde{U}_k \boldsymbol{a}(t) \quad \text{or} \quad \boldsymbol{a}(t) \approx \tilde{U}_k^{\mathsf{T}} \boldsymbol{v}(t) \in \mathbb{R}^k, \quad \text{with} \quad k \ll N_{xc} . \tag{5}$$

By projecting the dynamics onto the subspace generated by POD, the dimension of the problem is considerably reduced. We would like to replace (4) with a simplified dynamical model. To do this, we can perform a Galerkin projection, by expanding $\boldsymbol{v}$ with (5) and projecting (4) onto the k-dimensional subspace spanned by $\tilde{U}_k$. Multiplying both sides of (4) by $\tilde{U}_k^\mathsf{T}$, we obtain:

$$\tilde{U}_k^\mathsf{T} \frac{d\boldsymbol{v}(t;\boldsymbol{\lambda})}{dt} = \tilde{U}_k^\mathsf{T} \boldsymbol{f}\left(\boldsymbol{v}(t;\boldsymbol{\lambda})\right) \quad \text{or} \quad \frac{d\boldsymbol{a}(t;\boldsymbol{\lambda})}{dt} = \tilde{U}_k^\mathsf{T} \boldsymbol{f}\left(\boldsymbol{v}(t;\boldsymbol{\lambda})\right) . \tag{6}$$

A similar procedure is described in Cordier et al. [26] for the incompressible Navier-Stokes equations (see also Appendix A). Let's imagine for a moment that the matrix $\tilde{U}_k$ is known to within one orthogonal matrix Q. Replacing $\tilde{U}_k$ by $\tilde{U}_k Q$, we find again model (6), confirming that the POD subspace is invariant via multiplication by an orthogonal matrix. This result is at the basis of the intrusive method generally used for interpolation on the tangent space of the Grassmann manifold (see Section 3.3).

3. Interpolation in the tangent space of a matrix manifold

This section deals with interpolation in the tangent space of a matrix manifold. It begins (Section 3.1) with a reminder of essential notions of differential geometry and, in particular, recalls the definition and important properties of Stiefel and Grassmann manifolds. Essentially, these reminders come from Absil et al. [9], Bendokat et al. [27], Zimmermann [28], Boumal [29]. Readers familiar with differential geometry or interpolation on Grassmann manifolds might want to skip this section at first reading. Section 3.2 describes the algorithm of the interpolation in the tangent space of a matrix manifold. Finally, Section 3.3 presents the interest of interpolating on the Stiefel manifold compared with that on the Grassmann manifold.

3.1. Some elements of differential geometry

3.1.1. Matrix manifold

A *manifold* is a topological space[3] such that each point has a neighbourhood which is homeomorphic[4] to an open subset of an Euclidean space [31]. After Absil et al. [9, Chap. 3], a *matrix manifold* is a manifold for which there is a natural representation of elements in the form of matrices. This matrix representation allows to use linear algebra to tackle problems of differential geometry. In this way, it is easier, for example, to solve optimization problems, constrained by non-Euclidean data.

3.1.2. Matrix manifolds of interest

The matrix manifolds of interest for practical applications in mechanical engineering in general, and in particular in model reduction, are [28]:

- The general linear group $\mathrm{GL}(n)$ of invertible square matrices, *i.e.*

$$\mathrm{GL}(n) = \left\{ A \in \mathbb{R}^{n \times n} \mid \det(A) \neq 0 \right\} .$$

[3] A topological space is a set endowed with a structure called a topology. The most commonly used definition of a topology is via *open sets*. A topology on a set X may be defined [30, Section 1.2] as a collection τ of subsets of X, called open sets, satisfying the following axioms: 1. Both X and the empty set belong to τ. 2. Any (finite or infinite) union of members of τ belongs to τ. 3. Any finite intersection of members of τ belongs to τ. An essential aspect is the ability to define *closeness* without necessarily considering a distance.

[4] Two spaces with a homeomorphism between them are called homeomorphic. A homeomorphism is a bijective and continuous function between topological spaces that has a continuous inverse function.

Regular matrices appear in Linear Time Invariant (LTI) and discretized PDE systems.

- The orthogonal group $\mathrm{O}(n)$, *i.e.*

$$\mathrm{O}(n) = \left\{ Q \in \mathbb{R}^{n \times n} \mid QQ^{\mathsf{T}} = I_n = Q^{\mathsf{T}}Q \right\} .$$

- The matrix manifold of symmetric positive definite matrices $\mathrm{SPD}(n)$, *i.e.*

$$\mathrm{SPD}(n) = \left\{ A \in \mathsf{sym}(n) \mid x^{\mathsf{T}}Ax > 0, \, \forall x \in \mathbb{R}^n \setminus \{0\} \right\} ,$$

  where $\mathsf{sym}(n)$ is the set of real, symmetric n-by-n matrices.

- The Stiefel manifold $\mathcal{S}(n,r) \subset \mathbb{R}^{n \times r}$ of rectangular column-orthogonal n-by-r matrices with $r \leq n$, *i.e.*

$$\mathcal{S}(n,r) = \left\{ U \in \mathbb{R}^{n \times r} \mid U^{\mathsf{T}}U = I_r \right\} .$$

- The Grassmann manifold $\mathcal{G}(n,r) \subset \mathbb{R}^{n \times r}$ of r-dimensional subspaces $\mathcal{U}$ of $\mathbb{R}^n$ for $r \leq n$, *i.e.*

$$\mathcal{G}(n,r) = \left\{ \mathcal{U} \subset \mathbb{R}^n \mid \dim(\mathcal{U}) = r \right\} .$$

  Every point $\mathcal{U} \in \mathcal{G}(n,r)$ may be represented by selecting a basis $\{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_r\}$ such that $\mathcal{U} = \mathsf{span}\,(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_r)$. In numerical schemes, orthonormal bases are exclusively used. Hence, points $\mathcal{U}$ on the Grassmann manifold are represented by points $U \in \mathcal{S}(n,r)$ via $\mathcal{U} = \mathsf{span}(U)$. Such U is called a matrix representative of a subspace $\mathcal{U}$.

Stiefel and Grassmann manifolds play a fundamental role in interpolating POD bases. We therefore detail their most important properties in Section 3.1.2.1 and 3.1.2.2, respectively.

3.1.2.1. The Stiefel manifold. An alternative but equivalent mode of describing the Stiefel manifold is to consider a *quotient space* representation of $\mathcal{S}(n,r)$, see Example 7.4 of Zimmermann [28] or Section 2.3.1 of Edelman et al. [32]. Let $U \in \mathcal{S}(n,r)$, *i.e.* such that $U \in \mathbb{R}^{n\times r}$ and $U^{\mathsf{T}}U = I_r$. We extend the columns of $U = (u^1, \cdots, u^r)$ to an orthogonal matrix $Q = (u^1, \cdots, u^r, u^{r+1}, \cdots, u^n) \in \mathrm{O}(n)$. Let us define the Lie group $I_r \times \mathrm{O}(n-r) \subset \mathrm{O}(n)$ as

$$I_r \times \mathrm{O}(n-r) = \left\{ \begin{pmatrix} I_r & 0 \\ 0 & R \end{pmatrix} \mid R \in \mathrm{O}(n-r) \right\} .$$

The action $\tilde{Q} = Q\Phi$ with any orthogonal matrix $\Phi \in I_r \times \mathrm{O}(n-r)$ preserves the first r columns of Q. Hence, U may be identified with the equivalence class

$$[Q] = \{Q\Phi \mid \Phi \in I_r \times \mathrm{O}(n-r)\} .$$

From this, we deduce that the Stiefel manifold may be also defined as the quotient space $\mathcal{S}(n,r) = \mathrm{O}(n)/\left(I_r \times \mathrm{O}(n-r)\right)$.

Tangent space For any matrix $U \in \mathcal{S}(n,r)$, the tangent space of $\mathcal{S}(n,r)$ at U is given by

$$\mathrm{T}_U\mathcal{S}(n,r) = \{\Delta \in \mathbb{R}^{n\times r} \mid U^{\mathsf{T}}\Delta = -\Delta^{\mathsf{T}}U\} \subset \mathbb{R}^{n\times r} .$$

Every tangent vector $\Delta \in \mathrm{T}_U\mathcal{S}(n,r)$ may be written as:

$$\Delta = UA + U^{\perp}B, \quad A \in \mathbb{R}^{r\times r} \text{ skew}, \quad B \in \mathbb{R}^{(n-r)\times r} \text{ arbitrary} ,$$

where $U^{\perp} \in \mathcal{S}(n, n-r)$ is such that $(U, U^{\perp}) \in \mathrm{O}(n)$. The dimension of $\mathrm{T}_U\mathcal{S}(n,r)$ and $\mathcal{S}(n,r)$ is $nr - \frac{1}{2}r(r+1) = \frac{r(r-1)}{2} + r(n-r)$.

Distances Let $U \in \mathcal{S}(n,r)$ be a point and $\Delta = UA + U^{\perp}B$ and $\tilde{\Delta} = U\tilde{A} + U^{\perp}\tilde{B}$ be two tangent vectors belonging to $\mathrm{T}_U\mathcal{S}(n,r)$. Two standard metrics can be defined on the Stiefel manifold:

- The *Euclidean metric* that is the one inherited from the representation of $\mathcal{S}(n,r)$ in $\mathbb{R}^{n\times r}$:

$$\langle \Delta, \tilde{\Delta} \rangle_0 = \mathsf{trace}\left(\Delta^{\mathsf{T}}\tilde{\Delta}\right) = \mathsf{trace}\, A^{\mathsf{T}}\tilde{A} + \mathsf{trace}\, B^{\mathsf{T}}\tilde{B}\,.$$

- The *canonical metric* derived from the quotient representation of $\mathcal{S}(n,r)$:

$$\langle \Delta, \tilde{\Delta} \rangle_U = \mathsf{trace}\left(\Delta^{\mathsf{T}}\left(I - \frac{1}{2}UU^{\mathsf{T}}\right)\tilde{\Delta}\right) = \frac{1}{2}\,\mathsf{trace}\, A^{\mathsf{T}}\tilde{A} + \mathsf{trace}\, B^{\mathsf{T}}\tilde{B}\,.$$

Geodesics In differential geometry, the shortest path between two points in a manifold $\mathcal{M}$ is called a geodesic. This path can be uniquely represented by a twice differentiable function $c(t)$, with $0 \leq t \leq 1$, solution of a second order differential equation. For the Stiefel manifold, we can consider $c(0) = U \in \mathcal{S}(n,r)$ and $\dot{c}(0) = \Delta \in \mathrm{T}_U\mathcal{S}(n,r)$ as initial point of the geodesic and initial derivative condition, while the terminal point at $t = 1$ is $c(1) = \tilde{U} \in \mathcal{S}(n,r)$ (see Figure 1). The exponential mapping gives the final extremity point of the geodesic. The logarithmic mapping is the inverse of the exponential mapping when such an inverse can be defined.

The Stiefel exponential The Stiefel exponential at a base point $U \in \mathcal{S}(n,r)$ sends a Stiefel tangent vector Δ to the endpoint $\tilde{U} \in \mathcal{S}(n,r)$ of a geodesic that starts from U with velocity vector Δ. An efficient algorithm for computing the Stiefel exponential map with respect to the canonical metric was derived in [32]. This algorithm is reported in Appendix B.

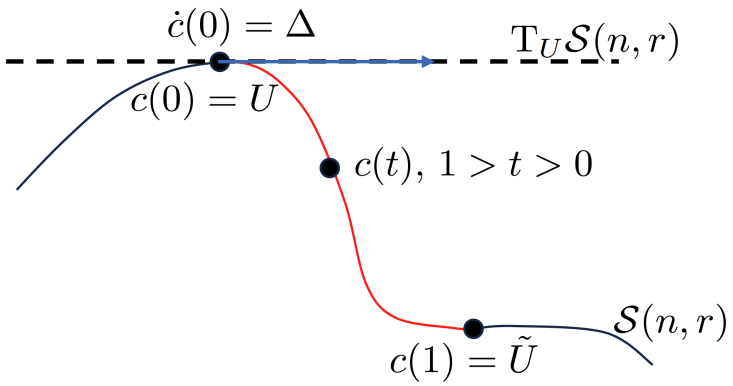


Figure 1: Visual illustration of the geodesic path $c(t)$ on the Stiefel manifold.

The Stiefel logarithm The Stiefel logarithm at a base point $U \in \mathcal{S}(n,r)$ finds for another point $\tilde{U} \in \mathcal{S}(n,r)$ a Stiefel tangent vector Δ such that the geodesic that starts from U with velocity Δ reaches $\tilde{U}$ after an arc length of $\|\Delta\|_U = \sqrt{\langle \Delta, \Delta \rangle_U}$.

A matrix-algebraic algorithm for approximating the Stiefel logarithm map with respect to the canonical metric was recently developed in [23]. Other variants may be found in [33]. This algorithm is reported in Appendix C.

3.1.2.2. The Grassmann manifold. Similar to what was done for the Stiefel manifold (see Section 3.1.2.1), the Grassmann manifold may be viewed as the quotient space $\mathcal{G}(n,r) = \mathrm{O}(n)/\left(\mathrm{O}(r) \times \mathrm{O}(n-r)\right)$ where the Lie subgroup $\mathrm{O}(r) \times \mathrm{O}(n-r) \subset \mathrm{O}(n)$ is defined as

$$\mathrm{O}(r) \times \mathrm{O}(n-r) = \left\{ \begin{pmatrix} S & 0 \\ 0 & R \end{pmatrix} \mid S \in \mathrm{O}(r), R \in \mathrm{O}(n-r) \right\}.$$

An alternative representation of the Grassmann manifold as a quotient man-

ifold of the Stiefel manifold is

$$\mathcal{G}(n,r) = \mathcal{S}(n,r)/\mathrm{O}(r) = \{[U] \mid U \in \mathcal{S}(n,r)\} \ ,$$

where the equivalence class is given by $[U] = \{UR \mid R \in \mathrm{O}(r)\}$.

Tangent space For any matrix representative $U \in \mathcal{S}(n,r)$ of $\mathcal{U} \in \mathcal{G}(n,r)$, the tangent space of $\mathcal{G}(n,r)$ at $\mathcal{U}$ is given by

$$\mathrm{T}_{\mathcal{U}}\mathcal{G}(n,r) = \{\Delta \in \mathbb{R}^{n\times r} \mid U^{\mathsf{T}}\Delta = 0\} \subset \mathbb{R}^{n\times r} \, .$$

Every tangent vector $\Delta \in \mathrm{T}_{\mathcal{U}}\mathcal{G}(n,r)$ may be written as:

$$\Delta = U^{\perp}B, \quad B \in \mathbb{R}^{(n-r)\times r} \text{ arbitrary} \, ,$$

where $U^{\perp} \in \mathcal{S}(n, n-r)$ is such that $(U, U^{\perp}) \in \mathrm{O}(n)$. The dimension of $\mathrm{T}_{\mathcal{U}}\mathcal{G}(n,r)$ and $\mathcal{G}(n,r)$ is $nr - r^2$.

Distances Let Δ and $\tilde{\Delta}$ be two tangent vectors belonging to $\mathrm{T}_{\mathcal{U}}\mathcal{G}(n,r)$. A metric on $\mathrm{T}_{\mathcal{U}}\mathcal{G}(n,r)$ is defined as

$$\langle \Delta, \tilde{\Delta} \rangle_{\mathcal{U}} = \mathrm{trace}\left(\Delta^{\mathsf{T}}\tilde{\Delta}\right) = \langle \Delta, \tilde{\Delta} \rangle_0 \, ,$$

where $\langle \cdot, \cdot \rangle_0$ stands for the standard inner matrix product. In contrast to the Stiefel manifold, the canonical metric is equivalent to the Euclidean metric [32, §2.5].

The Grassmann exponential The Grassmann exponential at a base point $\mathcal{U} \in \mathcal{G}(n,r)$ sends a Grassmann tangent vector Δ to the endpoint $\tilde{\mathcal{U}} \in \mathcal{G}(n,r)$ of a geodesic that starts from $\mathcal{U}$ with velocity vector Δ. An efficient algorithm for computing the Grassmann exponential was derived in [32]. This algorithm is reported in Appendix D.

The Grassmann logarithm The Grassmann logarithm at a base point $\mathcal{U} \in \mathcal{G}(n,r)$ finds for another point $\tilde{\mathcal{U}} \in \mathcal{G}(n,r)$ a Grassmann tangent vector Δ such that the geodesic that starts from $\mathcal{U}$ with velocity Δ reaches $\tilde{\mathcal{U}}$ after an arc length of $\|\Delta\|_{\mathcal{U}} = \sqrt{\langle \Delta, \Delta \rangle_{\mathcal{U}}}$.

An algorithm for computing the Grassmann logarithm is reported in Appendix E.

3.2. Principle of tangent space interpolation

The method of Interpolation on a Tangent Space to a Matrix Manifold (ITSMM) was proposed by Amsallem and Farhat [10] to interpolate bases obtained by POD (see Section 2.1 for the definition of POD and its link with the singular value decomposition of snapshot matrices). Indeed, despite the optimality of POD bases and their interest in modelling, they are generally highly dependent on the parameters used to determine them. Interpolating these bases was therefore quickly studied. However, since the space of POD subspaces is not flat, it is impossible to interpolate bases directly. The concept of differential geometry and matrix manifold has overcome these difficulties. Hereafter, we present interpolation on tangent space in a generic way, without focusing on Stiefel or Grassmann manifolds.

Let $\{\mathcal{Y}_i = \mathcal{Y}(\lambda_i)\}_{i=1}^{N_p}$ denote a set of elements of a manifold $\mathcal{M}$ and $\Lambda = \{\lambda_i\}_{i=1}^{N_p}$ be the corresponding set of operating conditions. We consider that each element $\mathcal{Y}_i$ is represented by a matrix $\mathrm{Y}_i \in \mathbb{R}^{n \times r}$ that belongs to a matrix manifold $\mathcal{M}'$. The goal is to compute an element $\widehat{\mathcal{Y}} \in \mathcal{M}$ associated with a new value of the parameter $\widehat{\lambda} \notin \Lambda$ and its matrix representative $\widehat{\mathrm{Y}} \in \mathcal{M}'$.

The interpolation algorithm on the tangent space consists of 4 steps (see Figure 2 for a graphical representation of the general operation):

- *Step 1:* Choose $\mathcal{Y}_{i_{\mathsf{Ref}}} \in \{\mathcal{Y}_i\}_{i=1}^{N_p}$ as a reference point. This point serves as the origin for the tangent interpolation space $\mathrm{T}_{\mathcal{Y}_{i_{\mathsf{Ref}}}}\mathcal{M}$.

- *Step 2:* Map each of the elements $\{\mathcal{Y}_i\}_{i=1}^{N_p}$ that are sufficiently close[5] to $\mathcal{Y}_{i_{\mathsf{Ref}}}$ to the tangent space $\mathrm{T}_{\mathcal{Y}_{i_{\mathsf{Ref}}}}\mathcal{M}$. For that, determine

$$\mathcal{X}_i = \mathsf{Log}^{\mathcal{M}}_{\mathcal{Y}_{i_{\mathsf{Ref}}}}(\mathcal{Y}_i)$$

  and the corresponding matrix representative $X_i \in \mathbb{R}^{n\times r}$.

- *Step 3:* Interpolate with any linear[6] multi-variate interpolation method the matrices X_i to obtain the matrix $\widehat{X}$ associated with the target value $\widehat{\lambda}$. A complete overview of the different interpolation scenarios is made in Section 7.3 of Benner et al. [34]. A commonly used method is to perform Lagrangian interpolation, *i.e.*

$$\widehat{X}(\widehat{\lambda}) = \sum_{j=1}^{N_p} \left(\prod_{i=1, i\neq j}^{N_p} \frac{\widehat{\lambda} - \lambda_i}{\lambda_j - \lambda_i} \right) X_j \,. \tag{7}$$

- *Step 4:* Map the element $\widehat{\mathcal{X}} \in \mathrm{T}_{\mathcal{Y}_{i_{\mathsf{Ref}}}}\mathcal{M}$ represented by the matrix $\widehat{X}$ to an element $\widehat{\mathcal{Y}} \in \mathcal{M}$ represented by a matrix $\widehat{\mathrm{Y}} \in \mathcal{M}'$. For that, determine

$$\widehat{\mathcal{Y}} = \mathsf{Exp}^{\mathcal{M}}_{\mathcal{Y}_{i_{\mathsf{Ref}}}}(\widehat{\mathcal{X}}) \,.$$

[5]The logarithm mapping $\mathrm{Log}^{\mathcal{M}}_{\mathcal{Y}_{i_{\mathrm{Ref}}}}$ is only defined in a neighbourhood of $\mathcal{Y}_{i_{\mathrm{Ref}}}$. If the neighbourhood is sufficiently small then the interpolation method should not be sensitive to the choice of the reference point $\mathcal{Y}_{i_{\mathrm{Ref}}}$.

[6]Because of the vector space structure of the tangent space of $\mathcal{M}$, the interpolation method must be linear with the data. Linear interpolation, Lagrange and Hermite interpolation, spline interpolation and interpolation via radial basis may be used.

When $\mathcal{M}$ is an embedded manifold[7] of $\mathbb{R}^{n\times r}$, then $\mathcal{M} = \mathcal{M}'$ and $\mathcal{Y}_i = \mathrm{Y}_i$ for $i = 1, \cdots, N_p$.

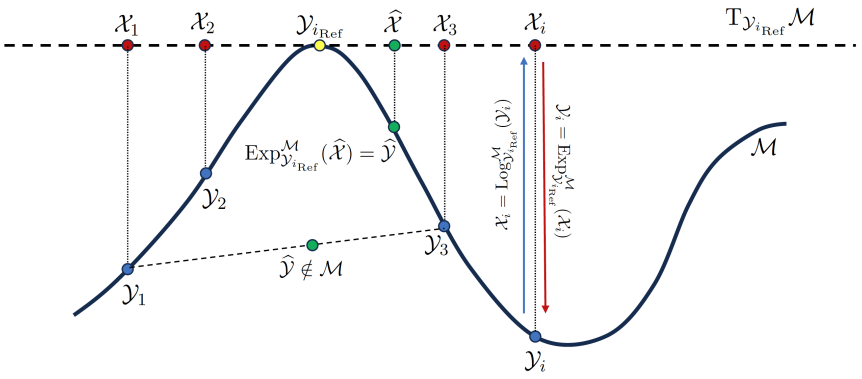


Figure 2: Graphical description of the interpolation on the tangent space of a matrix manifold $\mathcal{M}$. Each point $\mathcal{Y}_i$ of the manifold $\mathcal{M}$ is first projected via logarithm mapping onto the tangent space in $\mathcal{Y}_{i_{\mathrm{Ref}}}$ of $\mathcal{M}$. The points $\mathcal{X}_i$ which are obtained, are then interpolated to determine $\widehat{\mathcal{X}}$ corresponding to the solution for $\lambda = \widehat{\lambda}$. Finally, $\widehat{\mathcal{X}}$ is then projected back onto the manifold $\mathcal{M}$ with the exponential mapping, leading to $\widehat{\mathcal{Y}}$. Linear interpolation of points $\mathcal{Y}_1$ and $\mathcal{Y}_3$ does not lead in general to a point $\widehat{\mathcal{Y}}$ on the manifold $\mathcal{M}$.

3.3. Advantage of working with the Stiefel manifold

From a mathematical point of view, the Grassmann manifold can be represented (see Section 3.1.2.2) as a quotient manifold of the Stiefel manifold, *i.e.*

$$\mathcal{G}(n,r) = \mathcal{S}(n,r)/\mathrm{O}(r) = \{[U] \mid U \in \mathcal{S}(n,r)\} \ ,$$

where the points of $\mathcal{G}(n,r)$ are given by an equivalence class defined as $[U] = \{UR \mid R \in \mathrm{O}(r)\}$. The Grassmann elements are therefore known up to an orthogonal matrix. This result explains why, until the introduction of the method

[7] This is the case for $\mathrm{GL}(n)$, $\mathrm{O}(n)$, $\mathrm{SPD}(n)$ and $\mathcal{S}(n,r)$.

recently proposed by Oulghelou and Allery [16], interpolation on Grassmann was only carried out in an intrusive manner.

The main advantage of the Stiefel manifold is that it is an embedded manifold. Therefore, each point of the Stiefel manifold is represented by itself, rather than an equivalence class as in the case of quotient manifolds such as the Grassmann manifold. It is then possible to determine directly the interpolated solution without the need of solving additional optimization problems to calibrate the solution.

We present the algorithm in the next section and perform numerical tests on two generic configurations in Section 5.

4. Interpolation on the Stiefel manifold

4.1. Description of the ITSSM method

We consider a set of parameter values $\Lambda = \{\lambda_i\}_{i=1}^{N_p}$ where[8] $\lambda_i \in \mathbb{R}$ are physical and/or modelling parameters. For each value λ_i, we dispose of $\tilde{S}_i = \tilde{S}(\lambda_i)$ a set of snapshots, solutions of a parametrized non-linear space-time dependent problem of interest. Following the notations introduced in Section 2.1, each column of $\tilde{S}_i$ contains the solution to the problem at a given time. So we have $\tilde{S}_i \in \mathbb{R}^{N_{xc} \times N_t}$ where $N_{xc} = N_x\, N_c$ is the dimension of the solution state vector and N_t is the number of time samples.

The objective is to efficiently compute the solution snapshot matrix $\widehat{S} = S(\widehat{\lambda})$ for a new value $\widehat{\lambda} \notin \Lambda$ of the parameter without returning to numerical

[8]For the sake of simplicity, the algorithm is presented for $\lambda \in \mathbb{R}$. The extension to the case where we have $\boldsymbol{\lambda} \in \mathbb{R}^{N_\lambda}$ is immediate.

simulations. To resolve this problem, we propose in this paper a strategy based on Interpolation on a Tangent Space to the Stiefel Manifold (ITSSM).

The first step is to perform truncated Singular Value Decompositions (SVD) of order $k \leq N_t$ to the snapshot matrices (see Section 2.1), *i.e.*

$$\tilde{S}_i = \tilde{U}_i \tilde{\Sigma}_i \tilde{V}_i^{\mathsf{T}} \quad i = 1, \cdots, N_p \,,$$

where the index k is chosen via a decay analysis of the set of singular values $\tilde{\Sigma}_i$. A frequently used criterion is the relative information content (see Section 5.2.3). By definition of the SVD, the matrices $\tilde{U}_i$ and $\tilde{V}_i$ are orthogonal. We therefore have $\tilde{U}_i \in \mathcal{S}(N_{xc}, k)$ and $\tilde{V}_i \in \mathcal{S}(N_t, k)$.

The next step is to apply successively to $\tilde{U}_i$ and $\tilde{V}_i$ the ITSSM algorithm described in Section 3.2 with $\mathcal{M}$ a Stiefel manifold. This gives us $\widehat{U}$ and $\widehat{V}$, the left and right singular matrices of $\widehat{S}$. Similarly, the matrices of singular values $\tilde{\Sigma}_i$ are interpolated using a standard interpolation method such as Lagrange, Spline, Radial Basis Functions, Inverse Distance Weighted,... The result is $\widehat{\Sigma}$. Finally, we determine $\widehat{S}$ as $\widehat{U}\widehat{\Sigma}\widehat{V}$. The procedure is summarized in Alg. 1. An enhanced version of this algorithm that also uses derivative information with respect to the parameters was recently described in Zimmermann [35].

4.2. Computational complexity

The ITSSM algorithm can be divided into an off-line phase and an on-line phase. In the off-line phase, all SVD decompositions and projections onto the tangent space of the Stiefel manifold can be performed (Stiefel logarithm map). To do this, a reference point must be chosen to determine the tangent space. In the on-line phase, interpolations are performed and the interpolated matrices

Algorithm 1 Interpolation on the Tangent Space of the Stiefel Manifold (ITSSM). All the operations are done for $i = 1, \cdots, N_p$.

Input: $\tilde{S}_i = \tilde{S}(\lambda_i) \in \mathbb{R}^{N_{xc} \times N_t}$.

1: Apply the SVD $\tilde{S}_i = \tilde{U}_i \tilde{\Sigma}_i \tilde{V}_i^{\mathsf{T}}$.
   Choose k, order of the truncated SVD.

2: Apply successively to $\tilde{U}_i$ and $\tilde{V}_i$ the ITSSM algorithm described in Section 3.2 with $\mathcal{M}$ a Stiefel manifold.
   Determine $\widehat{U}$ and $\widehat{V}$.

3: Interpolate $\tilde{\Sigma}_i$ with a standard interpolation method.
   Determine $\widehat{\Sigma}$.

4: Compute $\widehat{S} = \widehat{U}\widehat{\Sigma}\widehat{V}$.

Output: $\widehat{S} = S(\widehat{\lambda})$.

projected back onto the Siefel space (Stiefel exponential map). The solution is determined by matrix multiplication of the various interpolated quantities.

The computational complexity of the different phases are as follows:

- N_p truncated SVD of the snapshot matrices. The total asymptotic computational cost is equal to $N_p \times \mathcal{O}(N_{xc} k^2)$.
- $2 \times (N_p - 1)$ Stiefel Logarithm mappings. The total asymptotic computational cost is equal to $2 \times (N_p - 1) \times (\mathcal{O}(N_{xc} k^2) + \mathcal{O}(N_t k^2))$.
- Assuming that Lagrange is chosen as univariate interpolation method, the interpolation of initial velocities and singular values matrices is executed for an asymptotic computational cost equal to $\mathcal{O}(N_{xc} k) + \mathcal{O}(N_t k) + \mathcal{O}(k^2)$.

– 2 Stiefel Exponential mappings. The total asymptotic computational cost is equal to $2 \times (\mathcal{O}(N_{xc}k^2) + \mathcal{O}(N_t k^2))$.

– The two matrix multiplications needed for $\widehat{S}$ are computed at an asymptotic computational cost equal to $\mathcal{O}(N_{xc}kN_t)$.

To analyse the numerical complexity, let's consider the case often encountered in numerical applications, where $N_{xc} \gg N_t \geq k$. The off-line steps are performed once and for all with an asymptotic computational cost equal to $(3N_p - 2) \times \mathcal{O}(N_{xc}k^2) + 2 \times (N_p - 1) \times \mathcal{O}(N_t k^2)$. On the other hand, a computational cost equal to $2 \times \mathcal{O}(N_{xc}k^2) + \mathcal{O}(N_{xc}N_t k) + \mathcal{O}(N_{xc}k) + 2 \times \mathcal{O}(N_t k^2) + \mathcal{O}(N_t k) + \mathcal{O}(k^2)$ is expected for the on-line phase. Hence, the total computational cost of the proposed algorithm grows linearly with respect to N_{xc}, the number of degrees of freedom of the high-fidelity model. Therefore, the method ITSSM is computationally efficient for on-line computations.

5. Applications

5.1. Application: 1D Burgers' equation

As a first configuration, we consider the one-dimensional non-linear Burgers' problem. These equations share many similarities with the Navier-Stokes equations, and have often been used as a model system for Turbulence. In the following, we analyse the Burgers' equations with periodic initial disturbance and homogeneous boundary conditions, as we know in this case an analytical solution u_a (see Appendix F for the expression), enabling interpolation methods to

be validated. The underlying problem is governed by the set of equations

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} - u \frac{\partial u}{\partial x}, \qquad x \in [0, L] \quad \text{and} \quad t \in [0, T] \ , \tag{8a}$$

$$u(x, 0) = u_0 \sin(\frac{\pi x}{L}) \qquad x \in [0, L] \ , \tag{8b}$$

$$u(0, t) = u(L, t) = 0 \qquad t \in [0, T] \ , \tag{8c}$$

where u, x, t and ν are the solution field, spatial coordinate, time and kinematic viscosity, respectively.

This problem is clearly parametrized by the kinematic viscosity ν. We therefore solve (8) for a set of viscosity values ranging from 0.01 to 0.2 in steps of 0.01. This set of numerical solutions will later be used to perform interpolations in Stiefel tangent spaces. In a classical Machine Learning approach, this set corresponds to the training data set. Two further numerical simulations are performed for the target values of kinematic viscosity considered: $\widehat{\nu} = 0.075$ and $\widehat{\nu} = 0.145$. This set acts as a testing data set. The other simulation parameters are fixed and arbitrarily given by $u_0 = 1$, $L = 1$ and $T = 0.1$. The solution field corresponding to a target value $\widehat{\nu}$ of kinematic viscosity is estimated using the ITSSM algorithm (see Alg. 1) for[9] $k = N_t = 1001$. The Lagrange polynomial method (see (7)) is chosen to interpolate both the singular values and the initial velocities in the tangent space to the Stiefel manifold. All calculations are performed in Matlab.

We first analyse the influence of numerical parameters characterizing the interpolation on the results. We call $\Delta\nu$, the difference in ν values between

[9] In this first, rather simple case study, there is no need to determine the dimension of the projection subspace via a criterion of relative information content, for instance. Such a study will be performed in Section 5.2.3 for the cylinder wake flow.

two points used for interpolation, and N_{IP}, the number of interpolation points used to apply the ITSSM algorithm. In the following, we consider $\Delta\nu \in \{0.01,\ 0.02,\ 0.03,\ \ 0.04,\ 0.05,\ 0.06\}$ and $N_{\mathsf{IP}} \in \{2, 3, 4\}$. The aim of these numerical tests is to challenge the interpolation method by considering a given number of interpolation points and increasing the value of $\Delta\nu$, *i.e.* by enlarging the space over which the data are taken. In Tables 1 and 2, we give the values of ν used for interpolation. Reference points are shown in bold. According to Amsallem and Farhat [10] (see Remark II, Section IV. B.), interpolation on the Grassmann manifold is not very sensitive to the choice of reference points as long as the bases to be interpolated are chosen in the neighbourhood of the reference point. We consider that the same result holds for interpolation on the Stiefel manifold, and choose by convention that

- for $N_{\mathsf{IP}} = 2$, the reference point is always the first point (left point) ;
- for $N_{\mathsf{IP}} = 3$, the reference point is always the second point (middle point) and
- for $N_{\mathsf{IP}} = 4$, the reference point is always the second point.

The quality of the estimation at $\widehat{\nu}$ will be measured on the basis of different relative error criteria. We first consider the 2-norm of the solution matrices obtained analytically (U_{a}) and by ITSSM (U_{ITSSM}). We thus define [10]

$$E_u(\nu = \widehat{\nu}) = \frac{\|U_{\mathsf{a}}(\nu = \widehat{\nu}) - U_{\mathsf{ITSSM}}(\nu = \widehat{\nu})\|_2}{\|U_{\mathsf{a}}(\nu = \widehat{\nu})\|_2}\,, \tag{9}$$

[10] By extension, we keep the same notation E_u and e_u when the approximation is made by global POD or by interpolation on the Grassmann manifold. The context should prevent any confusion.

where

$$\|A\|_2 = \sup_{\boldsymbol{x}\neq 0} \frac{\|A\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}\,,$$

for any $m \times n$ matrix A and $\boldsymbol{x} \in \mathbb{R}^n$.

We also analyse interpolated solutions using relative instantaneous error defined as

$$e_u(t;\nu=\widehat{\nu}) = \frac{\|u_{\mathsf{a}}(x,t;\nu=\widehat{\nu}) - u_{\mathsf{ITSSM}}(x,t;\nu=\widehat{\nu})\|_2}{\|u_{\mathsf{a}}(x,t;\nu=\widehat{\nu})\|_2}\,, \tag{10}$$

where $u_{\mathsf{ITSSM}}(x,t;\nu=\widehat{\nu})$ are obtained using the method described in Section 4.1.

$\Delta\nu$	$N_{\mathrm{IP}}=2$	$N_{\mathrm{IP}}=3$	$N_{\mathrm{IP}}=4$
0.01	**0.07** ; 0.08	0.06 ; **0.07** ; 0.08	0.05 ; **0.06** ; 0.07-0.08
0.02	**0.07** ; 0.09	0.05 ; **0.07** ; 0.09	0.07 ; **0.09** ; 0.11-0.13
0.03	**0.07** ; 0.1	0.07 ; **0.1** ; 0.13	0.01 ; **0.04** ; 0.07-0.1
0.04	**0.05** ; 0.09	0.01 ; **0.05** ; 0.09	0.01 ; **0.05** ; 0.09-0.13
0.05	**0.06** ; 0.11	0.01 ; **0.06** ; 0.11	0.01 ; **0.06** ; 0.11-0.16
0.06	**0.07** ; 0.13	0.01 ; **0.07** ; 0.13	0.01 ; **0.07** ; 0.13-0.19

Table 1: Interpolation points used for the prediction at $\widehat{\nu} = 0.075$ of the 1D Burgers' equation. Reference points are denoted in bold.

To evaluate the method proposed in Alg. 1, we compare the results of ITSSM with three approaches:

– The global POD approach, hereafter referred to as GPOD. This approach determines a global POD basis by including in the snapshot matrix multiple data sets obtained for several values of the control parameter [36] or from

$\Delta\nu$	$N_{\text{IP}} = 2$	$N_{\text{IP}} = 3$	$N_{\text{IP}} = 4$
0.01	**0.14** ; 0.15	0.13 ; **0.14** ; 0.15	0.13 ; **0.14** ; 0.15-0.16
0.02	**0.13** ; 0.15	0.13 ; **0.15** ; 0.17	0.13 ; **0.15** ; 0.17-0.19
0.03	**0.13** ; 0.16	0.10 ; **0.13** ; 0.16	0.10 ; **0.13** ; 0.16-0.19
0.04	**0.13** ; 0.17	0.09 ; **0.13** ; 0.17	0.05 ; **0.09** ; 0.13-0.17
0.05	**0.11** ; 0.16	0.06 ; **0.11** ; 0.16	0.01 ; **0.06** ; 0.11-0.16
0.06	**0.13** ; 0.19	0.07 ; **0.13** ; 0.19	0.01 ; **0.07** ; 0.13-0.19

Table 2: Interpolation points used for the prediction at $\widehat{\nu} = 0.145$ of the 1D Burgers' equation. Reference points are denoted in bold.

forced transients of the flow [7]. The snapshot matrix to be considered is therefore written as

$$S_{\text{GPOD}}(\widehat{\lambda}) = [S(\lambda_1) \dots S(\lambda_{N_{\text{IP}}})] \ ,$$

with $\lambda_1 < \widehat{\lambda} < \lambda_{N_{\text{IP}}}$. To ensure fair comparisons, the global POD basis is determined by retaining the snapshots corresponding to the control parameter values used for interpolation (see Tables 1 and 2). The same procedure will be followed in Section 5.2.4 for the wake flow (see Table 6). The number of columns in the snapshot matrix is therefore given by $N_{\text{IP}}\, N_t$. We then determine the temporal correlation matrix of size $N_{\text{IP}}\, N_t \times N_{\text{IP}}\, N_t$ and solve the eigenvalue problem. One advantage of GPOD is that the modes are calculated once and for all for all the values $\widehat{\lambda}$. However, the method is very costly in terms of calculation time (assembling the temporal correlation matrix and calculating the eigenelements), especially as the number of interpolation points increases. While good prediction properties can be obtained over the entire parameter space [see 36, for instance],

the global modes are no longer optimal for any of the control parameter values. We thus clearly lose any physical consideration. Furthermore, we have no mathematical results to guide our choice of snapshots to introduce into S_{GPOD}. Finally, the computational cost is still too high to make this approach a model reduction method in itself (see Table 10 for a comparison of the CPU time).

– A non-intrusive method similar to that of Alg. 1 in which the left and right singular matrices are interpolated on the Grassmann manifold. To make the distinction with the name introduced by Amsallem [37], this method is hereafter referred to as ITSGM$_{\text{ni}}$ (ni for non-intrusive). This method has the advantage of keeping the structure of Alg. 1. However, there is no guarantee that the modes are ordered correctly. It is therefore sometimes necessary to introduce a specific procedure (Bi-CITSGM algorithm introduced by Oulghelou and Allery [16] for instance) to rearrange the modes. In our applications, the ITSGM$_{\text{ni}}$ modes were correctly ordered. We therefore did not resort to such post-interpolation procedures. When using POD basis interpolations to solve optimal control problems, it is obviously not possible to rely on a posteriori checks. It will then be necessary to systematically rearrange the modes. This method has also been used in Friderikos et al. [38] to determine a space-time POD basis for parametric simulations of rigid-viscoplastic.

– Finally, we also use interpolation on the Grassmann manifold as originally introduced by Amsallem [37]. The spatial basis is thus interpolated on the Grassman manifold, the temporal evolution being obtained by integration

of a POD reduced-order model (see Appendix A). The results are hereafter referred to as ITSGM.

A comparison of the relative error values E_u obtained by ITSSM, GPOD, ITSGM$_{\text{ni}}$ and ITSGM is shown in Figures 3 and 4 for $\widehat{\nu} = 0.075$ and $\widehat{\nu} = 0.145$, respectively. Numerical values of interpolation errors are also given in Tables 3 and 4. We observe that the relative errors obtained by GPOD are 6 orders of magnitude lower than those from ITSSM, ITSGM$_{\text{ni}}$ and ITSGM. The order of magnitude of the error is 10^{-9} to 10^{-7} in the case of GPOD, 10^{-6} to 10^{-3} in the case of ITSSM, 10^{-3} to 10^{-2} in the case of ITSGM$_{\text{ni}}$ and ITSGM. We observe that the error obtained by ITSGM remains constant whatever the values of $\Delta\nu$ and N_{IP}. This result can be explained by the fact that the error increases gradually in time during the temporal integration of the POD reduced-order model (see Fig. 6). This phenomenon is classically encountered in model reduction and explains why data assimilation methods must sometimes be used to limit the amplification of the error [39]. In the vast majority of tested configurations, the error obtained by ITSSM is smaller than the errors obtained by interpolation on the Grassmann manifold.

For a given value of N_{IP}, the minimum reconstruction error determined by ITSSM is generally obtained for the lowest value of $\Delta\nu$ (0.01). For $N_{\text{IP}} = 4$, we do not observe this behaviour, the minimum value being found for $\Delta\nu = 0.04$. Furthermore, the evolution of the error is generally not monotonic with $\Delta\nu$ for a fixed value of N_{IP}. This result is not entirely surprising, since we do not know a priori the topology of the manifold representing the solution. We note, however, that the relative error does not tend to increase disproportionately with the value of $\Delta\nu$.

$\Delta\nu$	$N_{\text{IP}} = 2$			
	GPOD	ITSSM	ITSGM_{ni}	ITSGM
0.01	2.1402×10^{-8}	2.62×10^{-4}	2.7×10^{-3}	1.90×10^{-2}
0.02	2.1755×10^{-8}	1.7×10^{-3}	2.7×10^{-3}	1.90×10^{-2}
0.03	2.2455×10^{-8}	1.7×10^{-3}	2.7×10^{-3}	1.90×10^{-2}
0.04	2.3955×10^{-9}	4.7×10^{-3}	1.49×10^{-2}	1.90×10^{-2}
0.05	2.1577×10^{-9}	2.4×10^{-3}	8.5×10^{-3}	1.90×10^{-2}
0.06	2.6514×10^{-8}	1.2×10^{-3}	2.7×10^{-3}	1.90×10^{-2}

(a) $N_{\text{IP}} = 2$

$\Delta\nu$	$N_{\text{IP}} = 3$			
	GPOD	ITSSM	ITSGM_{ni}	ITSGM
0.01	1.8784×10^{-9}	2.15×10^{-4}	2.7×10^{-3}	1.90×10^{-2}
0.02	4.5266×10^{-9}	1.1×10^{-3}	2.7×10^{-3}	1.90×10^{-2}
0.03	2.4660×10^{-9}	1.6×10^{-3}	1.16×10^{-2}	1.90×10^{-2}
0.04	3.7987×10^{-7}	4.1×10^{-3}	1.75×10^{-2}	1.90×10^{-2}
0.05	4.5378×10^{-9}	5.7×10^{-3}	1.28×10^{-2}	1.90×10^{-2}
0.06	7.0529×10^{-9}	5.1×10^{-3}	4.5×10^{-3}	1.90×10^{-2}

(b) $N_{\text{IP}} = 3$

$\Delta\nu$	$N_{\text{IP}} = 4$			
	GPOD	ITSSM	ITSGM_{ni}	ITSGM
0.01	2.4190×10^{-9}	1.3×10^{-3}	8.5×10^{-3}	1.90×10^{-2}
0.02	2.2713×10^{-8}	3.1×10^{-3}	7.4×10^{-3}	1.90×10^{-2}
0.03	4.4338×10^{-8}	4.6×10^{-3}	2.27×10^{-2}	1.90×10^{-2}
0.04	6.5561×10^{-9}	7.23×10^{-4}	1.54×10^{-2}	1.90×10^{-2}
0.05	7.2344×10^{-9}	5.3×10^{-3}	9.7×10^{-3}	1.90×10^{-2}
0.06	7.0180×10^{-9}	1.6×10^{-3}	3.4×10^{-3}	1.90×10^{-2}

(c) $N_{\text{IP}} = 4$

Table 3: Relative interpolation errors E_u for different values of $\Delta\nu$ and N_{IP}. Prediction at $\widehat{\nu} = 0.075$ for the 1D Burgers' equation. Comparison of ITSSM with GPOD, ITSGM_{ni} and ITSGM results.

$\Delta\nu$	$N_{\mathrm{IP}} = 2$			
	GPOD	ITSSM	$\mathrm{ITSGM_{ni}}$	ITSGM
0.01	5.3168×10^{-9}	3.6×10^{-5}	1.5×10^{-3}	2.8×10^{-3}
0.02	6.8492×10^{-9}	1.18×10^{-4}	4.6×10^{-3}	2.8×10^{-3}
0.03	7.0203×10^{-9}	2×10^{-3}	4.6×10^{-3}	2.8×10^{-3}
0.04	7.3939×10^{-9}	2.4×10^{-3}	4.6×10^{-3}	2.8×10^{-3}
0.05	9.9095×10^{-9}	8.8×10^{-3}	1.14×10^{-2}	2.8×10^{-3}
0.06	8.1988×10^{-9}	3.2×10^{-3}	4.6×10^{-3}	2.8×10^{-3}

(a) $N_{\mathrm{IP}} = 2$

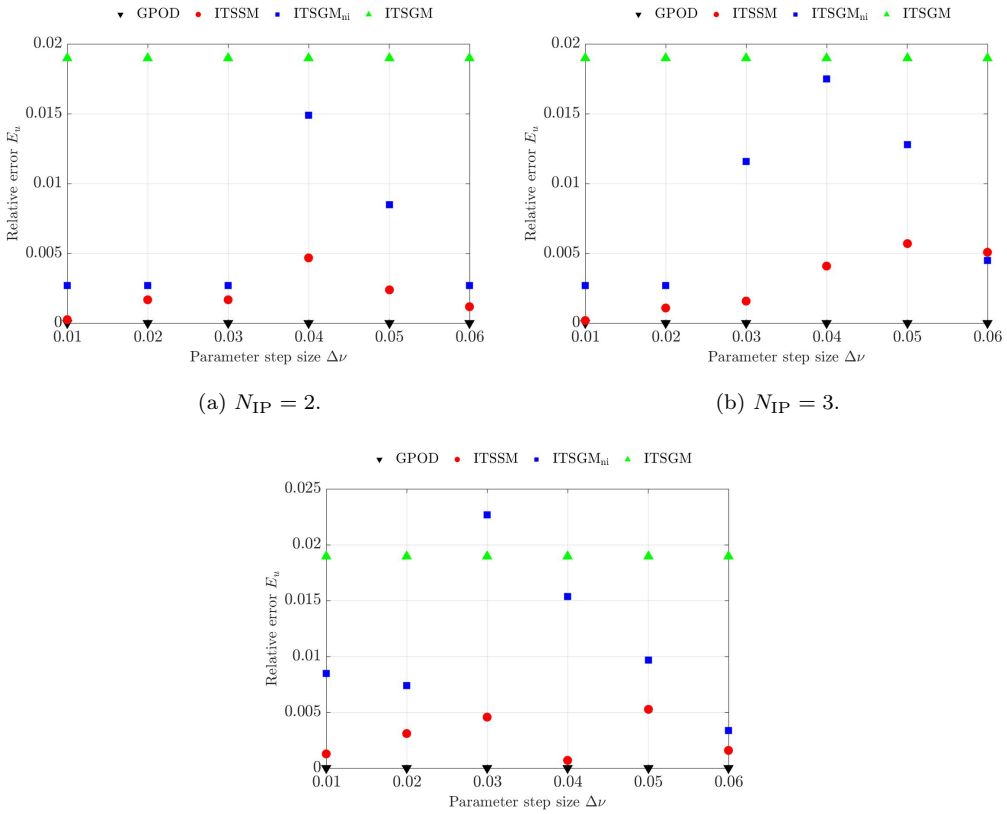
$\Delta\nu$	$N_{\mathrm{IP}} = 3$			
	GPOD	ITSSM	$\mathrm{ITSGM_{ni}}$	ITSGM
0.01	6.5573×10^{-9}	8×10^{-6}	1.5×10^{-3}	2.8×10^{-3}
0.02	6.7900×10^{-9}	1.49×10^{-4}	1.5×10^{-3}	2.8×10^{-3}
0.03	1.2009×10^{-8}	5.18×10^{-4}	4.6×10^{-3}	2.8×10^{-3}
0.04	1.5251×10^{-8}	2.7×10^{-4}	4.6×10^{-3}	2.8×10^{-3}
0.05	1.8883×10^{-9}	1.3×10^{-3}	1.14×10^{-2}	2.8×10^{-3}
0.06	2.6390×10^{-8}	1.4×10^{-3}	4.6×10^{-3}	2.8×10^{-3}

(b) $N_{\mathrm{IP}} = 3$

$\Delta\nu$	$N_{\mathrm{IP}} = 4$			
	GPOD	ITSSM	$\mathrm{ITSGM_{ni}}$	ITSGM
0.01	6.3958×10^{-9}	3.0×10^{-5}	1.5×10^{-3}	2.8×10^{-3}
0.02	7.0333×10^{-9}	6.14×10^{-4}	1.5×10^{-3}	2.8×10^{-3}
0.03	1.1475×10^{-8}	1.4×10^{-3}	4.6×10^{-3}	2.8×10^{-3}
0.04	2.3649×10^{-9}	1.2×10^{-3}	1.94×10^{-2}	2.8×10^{-3}
0.05	8.8879×10^{-9}	5.08×10^{-4}	3.46×10^{-2}	2.8×10^{-3}
0.06	8.8169×10^{-9}	6.77×10^{-4}	2.92×10^{-2}	2.8×10^{-3}

(c) $N_{\mathrm{IP}} = 4$

Table 4: Relative interpolation errors E_u for different values of $\Delta\nu$ and N_{IP}. Prediction at $\widehat{\nu} = 0.145$ for the 1D Burgers' equation. Comparison of ITSSM with GPOD, $\mathrm{ITSGM_{ni}}$ and ITSGM results.

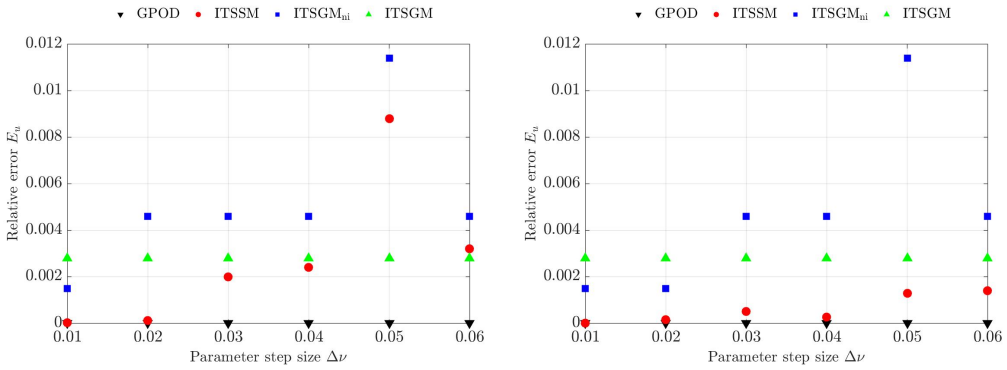


(a) $N_{\mathrm{IP}} = 2$.

(b) $N_{\mathrm{IP}} = 3$.

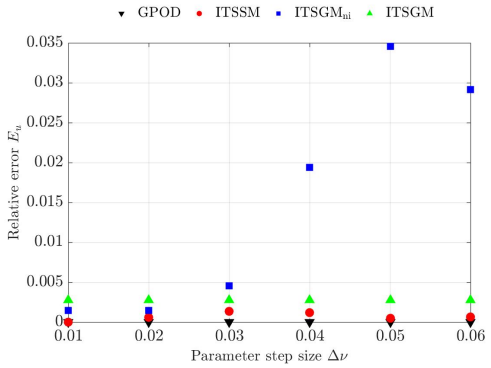
(c) $N_{\mathrm{IP}} = 4$.

Figure 3: Relative interpolation errors E_u for different values of $\Delta\nu$ and N_{IP}. Prediction at $\widehat{\nu} = 0.075$ for the 1D Burgers' equation. Comparison of ITSSM with GPOD, ITSGM$_{\mathrm{ni}}$ and ITSGM results.



(a) $N_{\mathrm{IP}} = 2$.

(b) $N_{\mathrm{IP}} = 3$.



(c) $N_{\mathrm{IP}} = 4$.

Figure 4: Relative interpolation errors E_u for different values of $\Delta\nu$ and N_{IP}. Prediction at $\widehat{\nu} = 0.145$ for the 1D Burgers' equation. Comparison of ITSSM with GPOD, ITSGM$_{\mathrm{ni}}$ and ITSGM results.

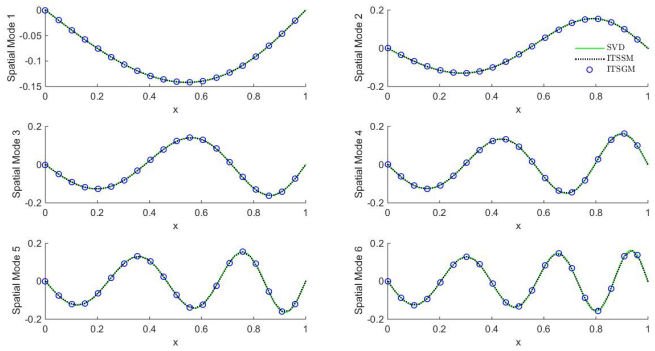


Figure 5: Comparison of the spatial modes obtained by SVD with those determined by interpolation (ITSSM and ITSGM methods) for $\widehat{\nu} = 0.075$, $N_{\mathrm{IP}} = 2$ and $\Delta\nu = 0.06$.

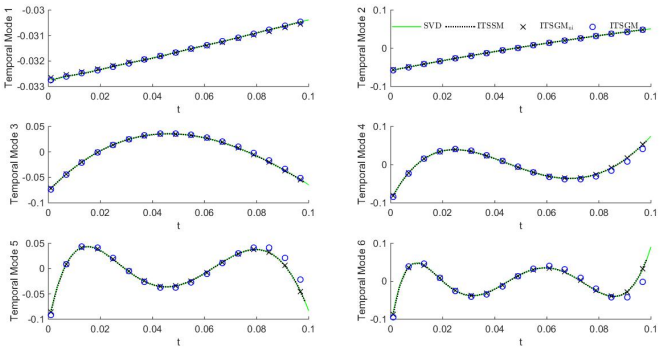
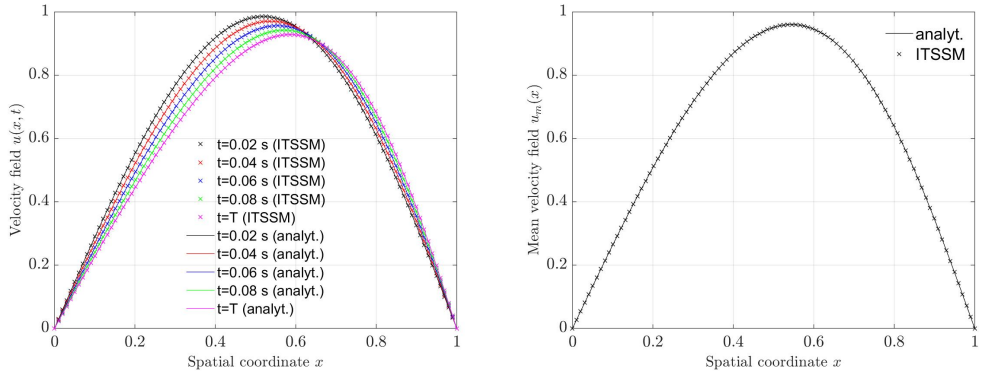


Figure 6: Comparison of the temporal modes obtained by SVD with those determined by interpolation (ITSSM, $\mathrm{ITSGM}_{\mathrm{ni}}$ and ITSGM methods) for $\widehat{\nu} = 0.075$, $N_{\mathrm{IP}} = 2$ and $\Delta\nu = 0.06$.

To go beyond the analysis of the relative error, we now seek to represent the solution at $\widehat{\nu} = 0.075$ from the ITSSM results interpolated under the conditions $N_{\mathsf{IP}} = 2$ and $\Delta\nu = 0.06$. The error associated with these parameters is in the low range of those obtained for all the parameters tested. It will therefore be characteristic of solutions that can be obtained by interpolation. We first compare in Figures 5 and 6, the spatial and temporal modes obtained by the three interpolation methods (ITSSM, ITSGM$_{\mathsf{ni}}$ and ITSGM) to the corresponding modes obtained directly by SVD. We obtain a perfect prediction of these two families of modes. Looking closely at the results obtained by ITSGM, we note, as we pointed out earlier, that the prediction at the end of the time integration domain ($t > 0.08$) and for the high-order modes becomes slightly imprecise. This phenomenon reflects the amplification in time of the error via the integration of the POD reduced-order model. The instantaneous and mean velocities determined by ITSSM are then compared with the analytical solution at various characteristic times in Figure 7. We observe a perfect match between the ITSSM reconstruction and the analytical solution. This result is further confirmed by the time evolution of the relative error corresponding to SVD and ITSSM shown in Figure 8 in semi-logarithmic scale. The error oscillates around 10^{-15} for the SVD results (of the order of magnitude of double precision) and 10^{-4} for ITSSM.

Finally, Figure 9 shows the relative interpolation errors obtained by ITSSM for different target values when $N_{\mathsf{IP}} = 2$ and $\Delta\nu = 0.06$. The target values considered range from 0.015 to 0.195 in 0.01 steps (19 values tested). In all cases, we obtain very low relative error values (at most 1.6%), indicating that ITSSM interpolation produces very good approximations of the solutions. We also note that for $\widehat{\nu} > 0.7$, the relative error values are even lower, below 0.4%.



(a) Instantaneous velocities.

(b) Mean velocity.

Figure 7: Comparison of instantaneous and mean velocities obtained analytically with the ITSSM solution for $\widehat{\nu} = 0.075$, $N_{\mathrm{IP}} = 2$ and $\Delta\nu = 0.06$.

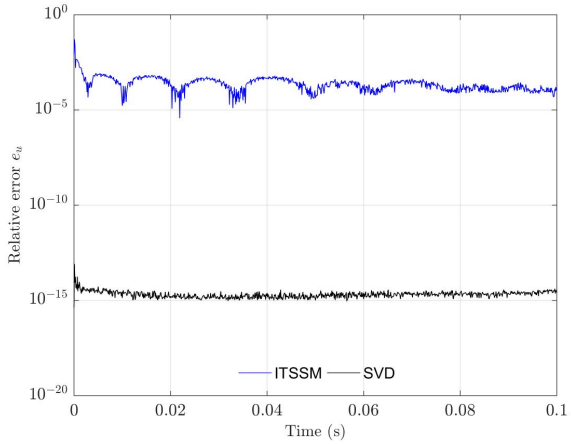


Figure 8: Time evolution of the relative interpolation error e_u for $\widehat{\nu} = 0.075$, $N_{\mathrm{IP}} = 2$ and $\Delta\nu = 0.06$. Comparison of SVD and ITSSM results.

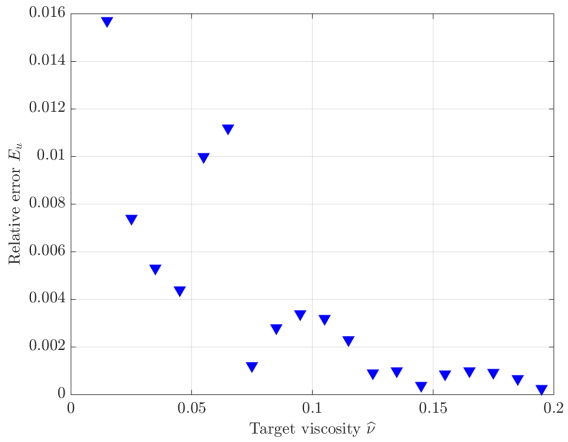


Figure 9: Relative interpolation errors E_u obtained by ITSSM for different target values $\widehat{\nu}$ when $N_{\mathrm{IP}} = 2$ and $\Delta\nu = 0.06$. Interpolation and reference points were chosen following the same rules as in Tables 1 and 2.

5.2. Application: 2D-cylinder wake flow

5.2.1. Flow configuration

We now consider as configuration the incompressible unsteady wake flow behind a circular cylinder of diameter D (see Figure 10). Let Ω be a two-dimensional domain filled with Newtonian viscous fluid of dynamic viscosity μ and density ρ. The boundaries $\partial\Omega$ of Ω are denoted by $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4 \cup \Gamma_c$, where Γ_i, $i = 1, 2, 3, 4$ are the external boundaries of the domain and Γ_c corresponds to the cylinder boundary. The continuity and Navier-Stokes equations are made dimensionless by introducing $L_{\mathrm{ref}} = D$, $u_{\mathrm{ref}} = U_\infty$, $t_{\mathrm{ref}} = D/U_\infty$ and $p_{\mathrm{ref}} = \rho U_\infty^2$ as reference quantities for length, velocity, time and pressure, respectively. The governing equations can then be expressed in the following non-dimensional form:

$$\begin{cases} \dfrac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = \quad -\nabla p + \dfrac{1}{\mathrm{Re}}\Delta \boldsymbol{u} \quad \boldsymbol{x}\in\Omega \quad t\in[0,T]\ , \\ \qquad\qquad \nabla\cdot\boldsymbol{u} = \quad 0\ , \end{cases} \tag{11}$$

where $\boldsymbol{u} = (u, v)$ is the velocity field, p is the pressure field and $\mathrm{Re} = \rho U_\infty D/\mu$ is the Reynolds number. The boundary conditions associated with the problem (11) are defined in Appendix G.1.

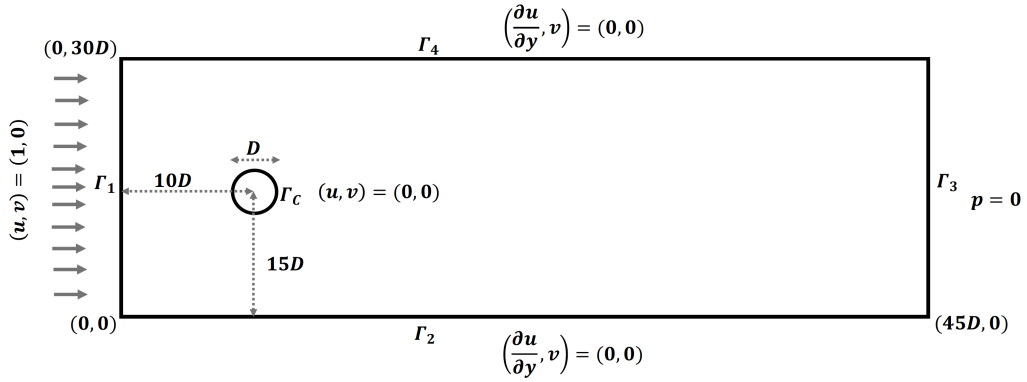


Figure 10: Computational domain and boundary conditions for the 2D-cylinder wake flow simulation.

Numerical simulations were performed using a Finite Element Method (FEM). For this, a variational formulation of the Navier-Stokes equations (11) was first derived (see Appendix G) and then solved with FreeFem++ [40]. For the time discretization, a third-order Backward Differential Formula was used with a constant time step $\Delta t = 0.01$. Non-linear terms were discretized semi-implicitly using an optimized Newton method. Concerning the spatial discretization, Taylor-Hood finite elements $\mathbb{P}2-\mathbb{P}1$ ($\mathbb{P}2$ for velocity fields and $\mathbb{P}1$ for pressure) were employed. The unstructured mesh (see Figure G.19 in Appendix G) contains 10798 DOFs for velocity and 8170 DOFs for pressure.

5.2.2. Numerical validation

To validate the numerical developments, we compare our results to those obtained in the 2D-2 benchmark of Schäfer et al. [41] for the same boundary conditions and flow parameters. We consider a flow at Re $= 100$, and evaluate the drag (C_D) and lift (C_L) coefficients, the Strouhal number characterizing the vortex shedding frequency (St $= Df_{\mathsf{vs}}/U_\infty$), and finally the pressure drop across the cylinder (Δp). The drag and lift coefficients are given by $C_D = F_D/\left(\frac{1}{2}\rho U_\infty^2\right)$ and $C_L = F_L/\left(\frac{1}{2}\rho U_\infty^2\right)$ where F_D and F_L are the drag and lift forces exerted by the fluid on the cylinder boundary Γ_c. This reference numerical simulation is randomly initialized in velocity and pressure. It is therefore necessary to wait for the decay of the transients (see Fig. 11) before analysing the characteristic flow quantities. The corresponding values are reported in Table 5. The results of our numerical simulations match well with those of the benchmark, validating our numerical developments.

Aerodynamic coefficients	Current results	Schäfer et al. [41]
$C_{D,\max}$	3.221	3.220 – 3.440
$C_{L,\max}$	1.011	0.990 - 1.010
$\Delta p_{\max}$	2.495	2.460 - 2.500
St	0.293	0.295 - 0.305

Table 5: Coefficients $C_{D,\max}$, $C_{L,\max}$, $\Delta p_{\max}$ and St for the 2D-cylinder wake flow at Re $= 100$. Comparison of our results with those obtained by Schäfer et al. [41]. The ranges of values given for Schäfer et al. correspond to the results obtained for different simulation parameters (mesh characteristics and time steps).

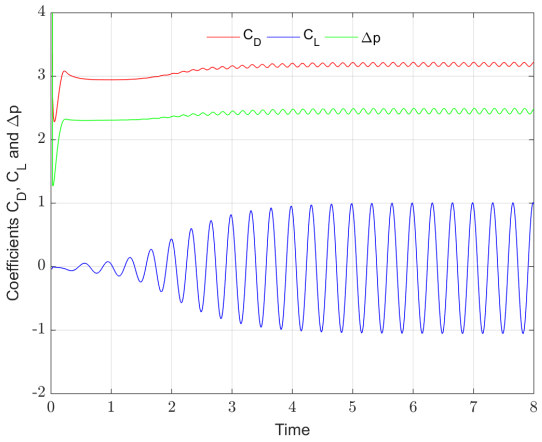


Figure 11: Time evolution of the aerodynamic coefficients (C_D and C_L) and pressure drop (Δp) along the cylinder for the 2D-cylinder wake flow at Re = 100. These results correspond to the benchmark described in Schäfer et al. [41]. The dimensions of the computational box are much smaller than those of Figure 10, which explains why the mean value of the drag coefficient and the maximum value of the lift coefficient are different from the values shown in Figure 12.

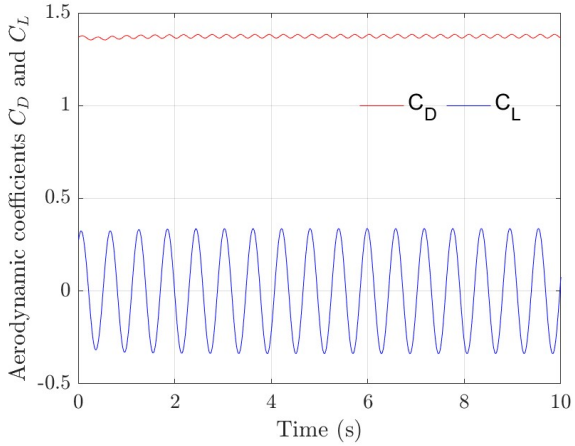
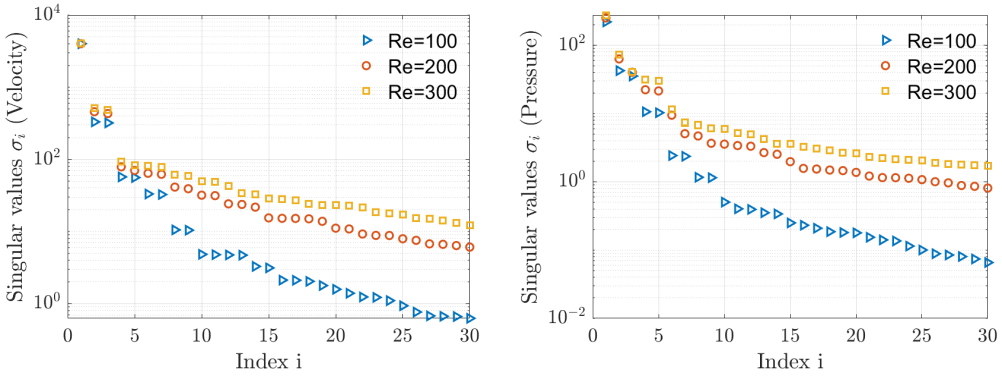


Figure 12: Time evolution of the aerodynamic coefficients C_D and C_L for the 2D-cylinder wake flow at Re = 100. The dimensions of the computational box are those shown in Figure 10.

5.2.3. SVD analysis

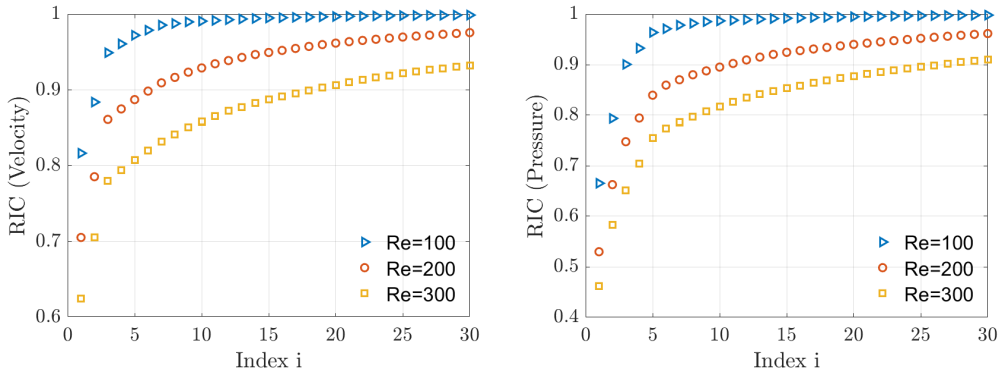
In Section 5.2.4, we will apply the ITSSM method to estimate both the velocity and the pressure fields in the wake flow at $\widehat{Re} = 195$. To do this, we will perform various interpolations (N_{IP} ranging from 2 to 4, as in Section 5.1) for Reynolds numbers going from 100 to 300. In this section, we therefore determine the order of the reduced-order models needed to represent with sufficient accuracy velocity and pressure fields over this Reynolds number range. To initialize all the numerical simulations performed for Reynolds numbers strictly greater than 100, we consider a solution snapshot obtained in the periodic flow regime at Re $= 100$ (see Figure 12, for example, for the time evolution of aerodynamic coefficients).

The SVD is performed on $N_t = 500$ snapshots taken evenly between $t_i = 2$ and $t_f = 7$, *i.e.* in the established dynamical regime. Figures 13a and 13b show the singular value spectrum obtained from the SVD of the velocity and pressure fields, respectively, at Re $= 100, 200$ and 300. The singular value spectrum plotted on a semi-log scale decreases quickly. It is therefore conceivable to neglect the contribution of the small singular values and to approximate the velocity and pressure fields by a low-rank approximation obtained with the first modes of the right and left SVD matrices. To go further in the analysis, the relative information content, defined as $\text{RIC}(k) = \frac{\sum_{i=1}^{k} \sigma_i}{\sum_{i=1}^{N_t} \sigma_i}$, is plotted in Figures 14a and 14b for both the velocity and pressure fields. Convergence to 1 is very fast for both velocity and pressure, with a slight advantage for velocity. In the following, we choose to keep the first 10 SVD modes for approximating the velocity and pressure fields.



(a) Velocity. (b) Pressure.

Figure 13: Singular value spectrum of the velocity and pressure fields for the 2D-cylinder wake flow (Re $= 100, 200$ and 300).



(a) Velocity. (b) Pressure.

Figure 14: Relative information content of the velocity and pressure fields for the 2D-cylinder wake flow (Re $= 100, 200$ and 300).

5.2.4. Prediction by ITSSM of the flow at $\widehat{Re} = 195$

The ITSSM procedure is applied to approximate both the velocity and the pressure fields by interpolating solutions snapshots associated to a variety of Reynolds number values. As a training database, we consider a set of numerical simulations, performed at Reynolds numbers ranging from 100 to 300 in steps of 10, *i.e.* 12 simulations. We choose as target Reynolds number $\widehat{\mathrm{Re}} = 195$ located roughly in the middle of the range defined above. As in Section 5.1, the Lagrange method is used to interpolate the singular values as well as the initial velocities on the tangent space. An investigation is done in order to choose at best the number of neighbouring interpolation points N_{IP} and the parameter step size ΔRe. In Table 6, we give the values of Reynolds numbers used for interpolation. Reference points, shown in bold, are determined similarly as in Section 5.1. Interpolation quality is assessed using the relative error criteria E_u and E_p (see (9) for the definition) for the velocity and pressure fields, respectively. These values are reported in Tables 7 and 8 where ITSSM results are compared with those of GPOD, ITSGM$_{\mathrm{ni}}$ and ITSGM. Let's start with some general observations. Firstly, since we do not derive a POD reduced-order model for the pressure coefficients, we cannot apply interpolation on the Grassmann manifold for pressure and therefore determine relative errors E_p in Table 8 for ITSGM. The relative errors obtained by GPOD are two orders of magnitude smaller than those associated with all the interpolation methods (ITSSM, ITSGM$_{\mathrm{ni}}$ and ITSGM.) The order of magnitude of the error in the case of GPOD is 10^{-3}, 10^{-2} to 10^{-1} in the case of ITSSM, 10^{-2} in the case of ITSGM$_{\mathrm{ni}}$ and finally 10^{-1} in the case of ITSGM. The error associated with the ITSGM method is nearly independent of the parameters ΔRe and N_{IP} for the same reasons as in Section 5.1 (see dis-

cussion of the results given in Tables 3 and 4). For the approximation of velocity fields, the ITSSM method always performs better than ITSGM. Surprisingly, the ITSGM_{ni} method remains competitive with ITSSM in terms of relative error. As for pressure fields, the relative error level is in favor of ITSSM, for $N_{\text{IP}} = 2$ when $\Delta\text{Re} = 30$, 40 or 50, for $N_{\text{IP}} = 3$ in the case of $\Delta\text{Re} = 30$ and finally for $N_{\text{IP}} = 4$ when $\Delta\text{Re} = 50$.

ITSSM interpolation gives the lowest relative errors for $N_{\text{IP}} = 3$ and $\Delta\text{Re} = 30$. Hereafter, we will use these values of parameters to evaluate the method on other physical criteria.

ΔRe	$N_{\text{IP}} = 2$	$N_{\text{IP}} = 3$	$N_{\text{IP}} = 4$
10	**190** ; 200	190 ; **200** ; 210	190 ; **200** ; 210 ; 220
20	**190** ; 210	170 ; **190** ; 210	180 ; **200** ; 220 ; 240
30	**190** ; 220	160 ; **190** ; 220	160 ; **190** ; 220 ; 250
40	**190** ; 230	150 ; **190** ; 230	160 ; **200** ; 240 ; 280
50	**190** ; 240	150 ; **200** ; 250	150 ; **200** ; 250 ; 300
100	**190** ; 290	100 ; **200** ; 300	See caption.

Table 6: Interpolation points used for the prediction at $\widehat{Re} = 195$ of the cylinder wake flow. Reference points are denoted in bold. In our training database, we do not have four numerical simulations separated by 100 in terms of Reynolds numbers. Hence, we cannot evaluate the case $N_{\text{IP}} = 4$ and $\Delta Re = 100$. For the same reasons, there will be no results in the corresponding cells of Tables 7 and 8.

In Figure 15, we compare the first 5 longitudinal modes obtained by SVD and ITSSM for $N_{\text{IP}} = 3$ and $\Delta\text{Re} = 30$. Figure 16 shows the corresponding representations for the transverse spatial modes. We obtain a perfect match between the two families of modes, showing that ITSSM interpolation works.

ΔRe	$N_{\text{IP}} = 2$			
	GPOD	ITSSM	ITSGM_{ni}	ITSGM
10	2.1×10^{-3}	1.595×10^{-1}	2.82×10^{-2}	1.709×10^{-1}
20	1.6×10^{-3}	8.91×10^{-2}	2.9×10^{-2}	1.708×10^{-1}
30	1.6×10^{-3}	1.6×10^{-2}	2.91×10^{-2}	1.708×10^{-1}
40	1.5×10^{-3}	1.51×10^{-2}	2.9×10^{-2}	1.707×10^{-1}
50	1.2×10^{-3}	4.6×10^{-2}	2.88×10^{-2}	1.707×10^{-1}
100	1.5×10^{-3}	3.8×10^{-2}	2.88×10^{-2}	1.703×10^{-1}

(a) $N_{\text{IP}} = 2$

ΔRe	$N_{\text{IP}} = 3$			
	GPOD	ITSSM	ITSGM_{ni}	ITSGM
10	1.6×10^{-3}	1.259×10^{-1}	2.7×10^{-2}	1.709×10^{-1}
20	1.6×10^{-3}	6.02×10^{-2}	2.85×10^{-2}	1.708×10^{-1}
30	1.6×10^{-3}	1.07×10^{-2}	2.84×10^{-2}	1.706×10^{-1}
40	1.6×10^{-3}	1.7×10^{-2}	2.86×10^{-2}	1.704×10^{-1}
50	1.4×10^{-3}	3.43×10^{-2}	2.73×10^{-2}	1.713×10^{-1}
100	1.8×10^{-3}	2.96×10^{-2}	2.85×10^{-2}	1.726×10^{-1}

(b) $N_{\text{IP}} = 3$

ΔRe	$N_{\text{IP}} = 4$			
	GPOD	ITSSM	ITSGM_{ni}	ITSGM
10	1.6×10^{-3}	1.063×10^{-1}	2.85×10^{-2}	1.708×10^{-1}
20	1.0×10^{-3}	5.68×10^{-2}	2.8×10^{-2}	1.708×10^{-1}
30	1.2×10^{-3}	2.01×10^{-2}	2.92×10^{-2}	1.707×10^{-1}
40	1.2×10^{-3}	1.51×10^{-2}	2.76×10^{-2}	1.710×10^{-1}
50	1.3×10^{-3}	1.99×10^{-2}	2.73×10^{-2}	1.711×10^{-1}
100	—	— 47	—	—

(c) $N_{\text{IP}} = 4$

Table 7: Relative errors E_u of the velocity field for the 2D-cylinder wake flow at $\widehat{\text{Re}} = 195$ for different values of ΔRe and N_{IP}. Comparison of ITSSM with GPOD, ITSGM_{ni} and ITSGM results.

ΔRe	$N_{\mathrm{IP}} = 2$		
	GPOD	ITSSM	$\mathrm{ITSGM_{ni}}$
10	3.1×10^{-3}	1.838×10^{-1}	6.67×10^{-2}
20	2.9×10^{-3}	1.385×10^{-1}	6.69×10^{-2}
30	2.9×10^{-3}	5.84×10^{-2}	6.72×10^{-2}
40	2.7×10^{-3}	5.67×10^{-2}	6.87×10^{-2}
50	2.8×10^{-3}	5.12×10^{-2}	6.97×10^{-2}
100	3.1×10^{-3}	8.39×10^{-2}	7.01×10^{-2}

(a) $N_{\mathrm{IP}} = 2$

ΔRe	$N_{\mathrm{IP}} = 3$		
	GPOD	ITSSM	$\mathrm{ITSGM_{ni}}$
10	2.9×10^{-3}	1.676×10^{-1}	6.5×10^{-2}
20	3.0×10^{-3}	1.048×10^{-1}	6.68×10^{-2}
30	3.3×10^{-3}	5.08×10^{-2}	6.69×10^{-2}
40	3.1×10^{-3}	6.95×10^{-2}	6.74×10^{-2}
50	3.1×10^{-3}	7.77×10^{-2}	6.56×10^{-2}
100	3.1×10^{-3}	7.14×10^{-2}	6.85×10^{-2}

(b) $N_{\mathrm{IP}} = 3$

ΔRe	$N_{\mathrm{IP}} = 4$		
	GPOD	ITSSM	$\mathrm{ITSGM_{ni}}$
10	3.0×10^{-3}	1.458×10^{-1}	6.5×10^{-2}
20	2.4×10^{-3}	1.682×10^{-1}	6.55×10^{-2}
30	2.8×10^{-3}	7.89×10^{-2}	6.75×10^{-2}
40	2.7×10^{-3}	1.074×10^{-1}	6.71×10^{-2}
50	2.8×10^{-3}	5.55×10^{-2}	6.73×10^{-2}
100	—	48 —	—

(c) $N_{\mathrm{IP}} = 4$

Table 8: Relative errors E_p of the pressure field for the 2D-cylinder wake flow at $\widehat{\mathrm{Re}} = 195$ for different values of ΔRe and N_{IP}. Comparison of ITSSM with GPOD and $\mathrm{ITSGM_{ni}}$ results.

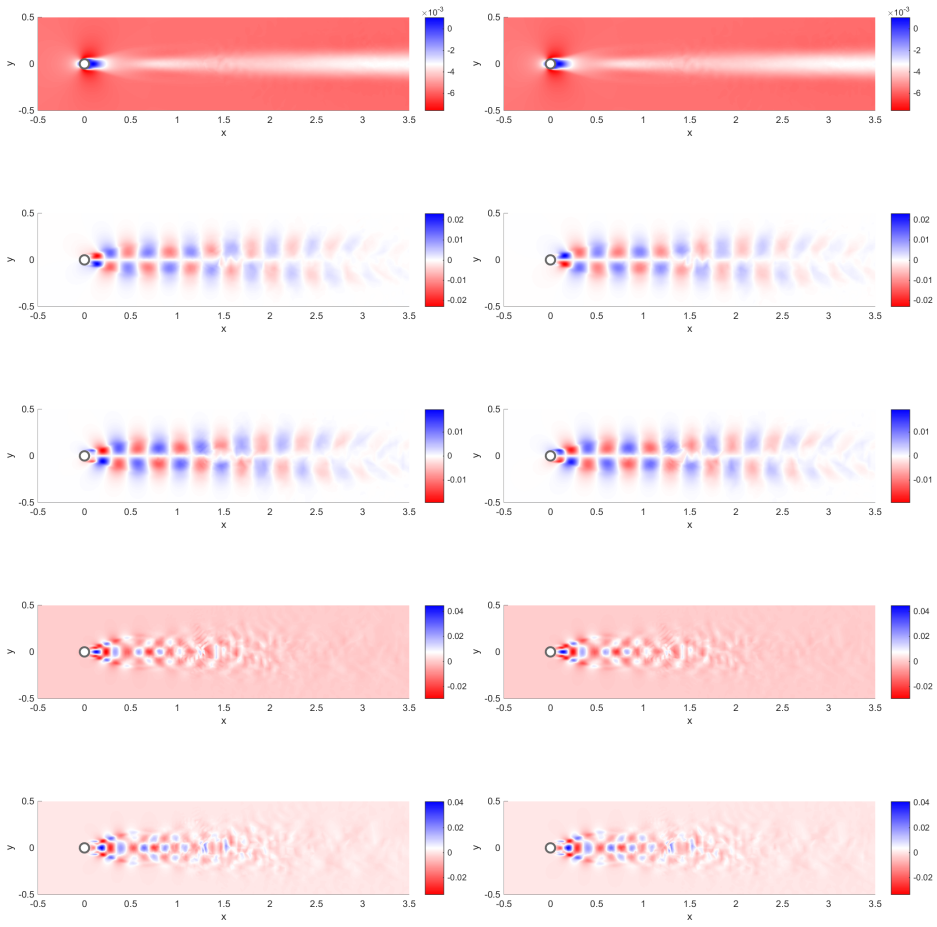


Figure 15: Comparison of longitudinal spatial modes obtained by SVD (left) and ITSSM (right). Mode 1 to 5 from top to bottom.

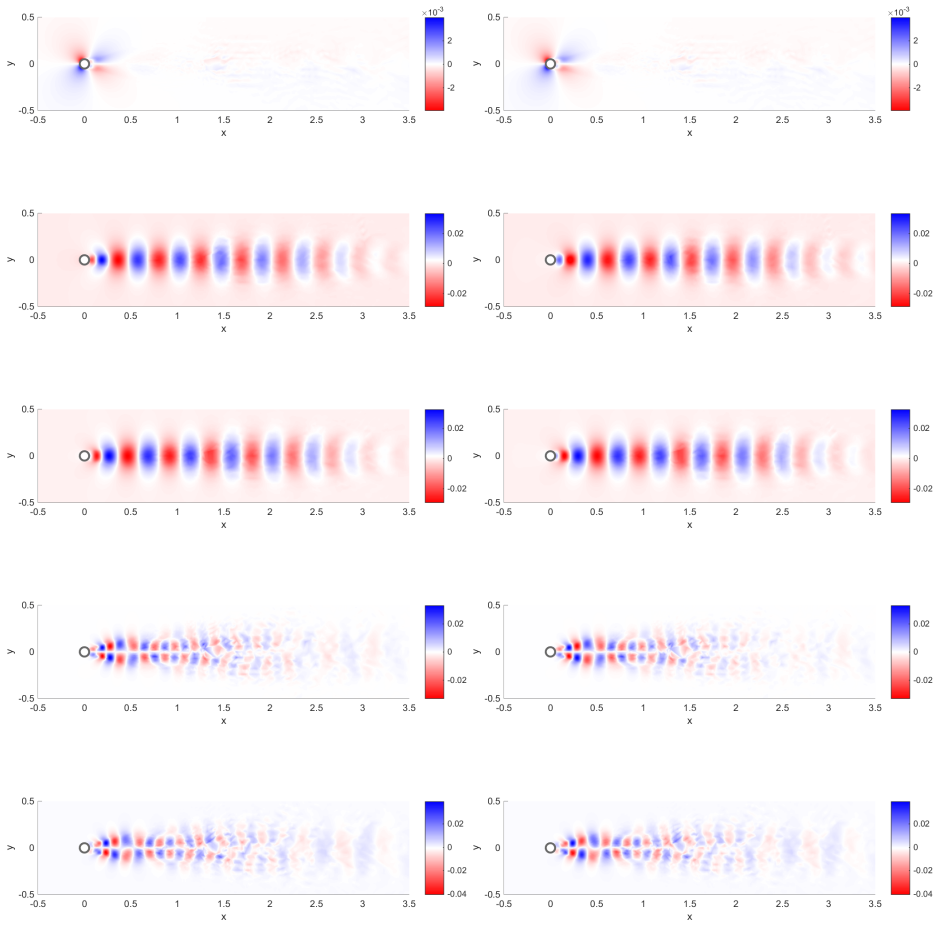


Figure 16: Comparison of transversal spatial modes obtained by SVD (left) and ITSSM (right). Mode 1 to 5 from top to bottom.

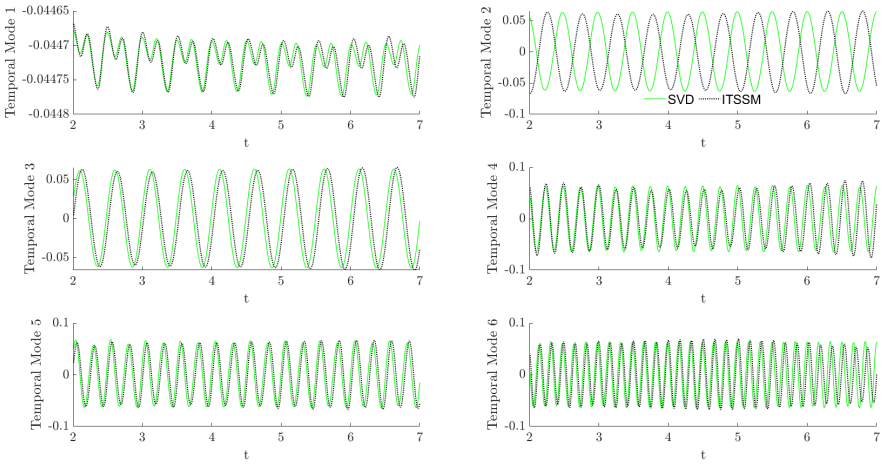


Figure 17: Comparison of the first 6 temporal modes obtained by SVD and ITSSM.

Figure 17 represents the temporal evolution of the first 6 modes obtained by POD (SVD of the snapshot matrix at $\widehat{\mathrm{Re}} = 195$) and by interpolation on the tangent space of the Stiefel manifold. As expected, the amplitude variation of the first POD mode is very small, as this component corresponds, to a first approximation, to the mean field calculated as the average over all the snapshots used for POD. For the other modes, we find that the solutions determined by ITSSM are very accurate representation of the targeted field. However, we note a slight phase shift in time, mainly on modes 2 and 3. We also note that mode 2 is obtained to within -1 coefficient, which is reflected in the isovalues shown in Figures 15 and 16.

To further assess the ITSSM method, we compare the results obtained by finite element simulations and ITSSM in terms of average drag coefficient ($\overline{C_D}$), maximum lift coefficient ($C_{L,\mathsf{max}}$) and Strouhal number (St), see Table 9 en-

hanced[11] with results of GPOD and ITSGM$_{\text{ni}}$. For the Strouhal number, all the reduced-order methods predict the exact value. The GPOD algorithm gives a better prediction of $\overline{C_D}$ and $C_{L,\text{max}}$ than the ITSGM$_{\text{ni}}$ and ITSSM methods. The prediction of $\overline{C_D}$ is better by ITSSM than by ITSGM$_{\text{ni}}$. Regarding to the prediction of the value of $C_{L,\text{max}}$, ITSSM results in an underestimate, while ITSGM$_{\text{ni}}$ results in an overestimate, with a very slight advantage in favor of ITSGM$_{\text{ni}}$.

The results of the ITSSM method converge to those obtained with the FEM method. This result is confirmed in Figures 18a and 18b where the time evolution of C_D and C_L is plotted. The results obtained by ITSSM are in perfect agreement with those of FEM simulations. In terms of CPU time, it is shown in Table 10 that the ITSSM approach is[12] 525 times faster than the FEM method, 195 times faster than the POD approach which is the classical model reduction technique in fluid mechanics and 11932 times faster than the GPOD. The CPU times of the two non-intrusive interpolation methods are of the same order of magnitude, with the ITSGM$_{\text{ni}}$ method being twice as fast as the ITSSM algorithm. The ITSGM method takes 58 times longer than the ITSSM, the former being penalized by the calculation of the coefficients of the POD reduced-order model and its time integration. In addition, the CPU time corresponding to the online phase of the ITSSM method is very small (0.0469 s) and thus could be perfectly executed in near-real time computations [13].

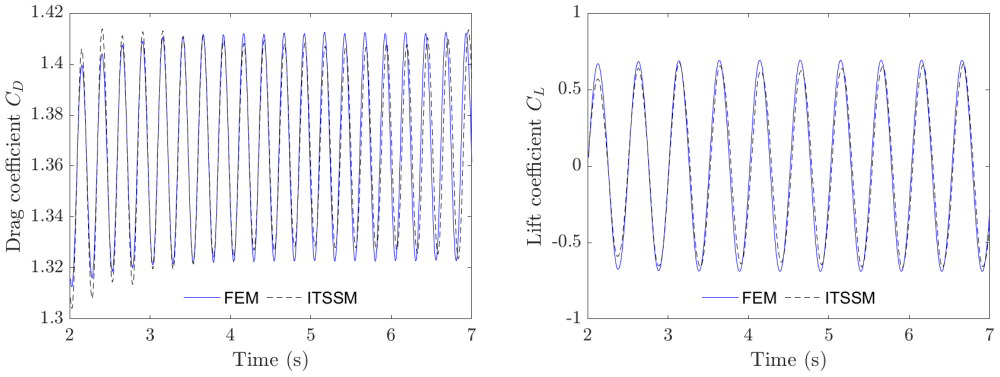
[11] For the same reasons as above (see analysis of Table 8), we did not use ITSGM to estimate the pressure field, and were therefore unable to assess its contribution to the drag and lift coefficients.

[12] These CPU time ratios are evaluated by considering the sum of the offline and online phases.

[13] Here, we use "real time" in the sense described by Chinesta et al. [42], *i.e.* as an

	$\overline{C_D}$	$C_{L,\max}$	St
FEM	1.3666	0.6928	0.1953
GPOD	1.3677	0.6957	0.1953
ITSGM_{ni}	1.3659	0.7154	0.1953
ITSSM	1.3661	0.6772	0.1953

Table 9: Comparison of coefficients $\overline{C_D}$, $C_{L,\max}$ and St obtained by FEM, GPOD, ITSGM_{ni} and ITSSM for the 2D-cylinder wake flow at $\widehat{\text{Re}} = 195$ with $N_{\text{IP}} = 3$ and $\Delta\text{Re} = 30$.



(a) C_D. (b) C_L.

Figure 18: Comparison of the time evolution of C_D and C_L obtained by FEM and ITSSM for the 2D-cylinder wake flow at $\widehat{\text{Re}} = 195$ with $N_{\text{IP}} = 3$ and $\Delta\text{Re} = 30$.

	CPU time	
	Offline	Online
FEM	47.7 min	
POD	17.7 min	0.4 s
GPOD	18.05 h	2.8 s
ITSGM_{ni}	2.6 s	0.0247 s
ITSGM	2.5 s	5.22 min
ITSSM	5.4 s	0.0469 s

Table 10: Comparison of CPU times associated to FEM, POD, GPOD, ITSGM_{ni}, ITSGM and ITSSM methods for the 2D-cylinder wake flow at $\widehat{\text{Re}} = 195$ with $N_{\text{IP}} = 3$ and $\Delta\text{Re} = 30$. The POD is computed at $\text{Re} = 195$ with 500 snapshots. For the POD application, the offline phase involves calculating the temporal correlation matrix and solving the eigenvalue problem. The online phase corresponds to the reconstruction of the physical fields from the selected modes (6 for POD and 42 for GPOD). For the application of Grassmann or Stiefel interpolation methods, the offline phase corresponds to the application of SVD and the calculation of Logarithm mappings. For ITSGM, the online phase involves calculating the coefficients of the POD reduced-order model, its time integration and the reconstruction of the physical fields. The calculation of the coefficients of the POD reduced-order model (terms A_i, B_{ij} and C_{ijk} in Appendix A) is the most penalizing step.

6. Conclusion

In this paper, we have proposed a new non-intrusive interpolation strategy for non-linear, parameterized, time-dependent fluid flow problems such as those encountered with the Navier-Stokes equations. The aim of the proposed method, called Interpolation on the Tangent Space of the Stiefel Manifold (ITSSM), is to adapt an existing set of experimental or numerical data to estimate the solution of the problem for a new value of control parameters. This method operates on a set of solution snapshots and not on the underlying high-fidelity model. It is therefore perfectly suited for real-time computations.

Up to now, the model reduction community has used interpolation on the Grassmann manifold to approximate solutions for a new parameter value. In the intrusive approach originally proposed by Amsallem [37], the spatial modes were interpolated on the Grassmann manifold, and the temporal modes determined using a reduced-order model derived by Galerkin projection of the high-fidelity model onto the previous spatial modes. It was therefore necessary to know the high-fidelity model in order to perform the interpolation. However, for some complex physical problems, such as brain function, there is as yet no model. Very recently, Oulghelou and Allery [16] proposed a non-intrusive method based on a calibration phase performed in a corrective manner after an initial stage of interpolation on the Grassmann manifold. Intrinsically, these difficulties stem from the quotient structure of the Grassmann manifold, which leads to the determination of a solution up to right multiplication with an orthogonal matrix. After the interpolation on Grassmann, it is therefore necessary to solve auxiliary optimiza-

approach that is fast enough to be embedded in real systems requiring decision-making.

tion problems in order to determine two orthogonal matrices of small dimension leading to physical solution. The advantage of interpolation on the tangent space of the Stiefel manifold is that it does not require calibration phases. This comes from the fact that the Stiefel manifold is an embedded manifold, where each element is uniquely represented by itself rather than by an equivalence class as in the case of the Grassmann manifold. Clearly, the interpolation on the Stiefel manifold was made possible thanks to the efficient algorithm developed by Zimmermann [23] in order to evaluate the logarithm mapping on the Stiefel manifold.

Interpolation on the tangent space of the Stiefel manifold was tested on two academic fluid dynamics configurations. The first corresponds to the one-dimensional non linear Burgers' problem, where the parameter is the kinematic viscosity. The second configuration is the two-dimensional cylinder wake flow, where the parameter is given by the Reynolds number. The obtained results show that the ITSSM algorithm is efficient in terms of both reconstruction accuracy and computational time. ITSSM results were compared to Global POD and to two interpolation methods based on Grassmann interpolation (ITSGM_{ni} and ITSGM). The Global POD was more accurate but too expensive in terms of CPU time. For both tested configurations, ITSSM was more accurate than ITSGM. We also found that in the case of the cylinder wake flow, ITSSM performs better than ITSGM_{ni} for several values of N_{IP} and ΔRe. The results of Section 5.2.4 showed that the ITSSM method perfectly predicted the flow at $\text{Re} = 195$.

Following this initial work, many perspectives open up. Obviously, we will now be able to solve optimal control problems in fluid dynamics. We will also be able to revisit our algorithm by focusing on the stability conditions recently highlighted in the work of Friderikos et al. [17]. Another direction of interest is to study

the influence of the chosen interpolation method on the interpolated results. An interesting direction would be to use a weighted barycentric interpolation method as in Mosquera et al. [43], Oulghelou et al. [18], Zimmermann and Bergmann [20]. It would be also interesting to perform a data clustering step before interpolating by ITSSM. Very likely, this preprocessing step should improve on-line interpolation performance in terms of precision. Finally, the ITSSM algorithm being very versatile, we can imagine applying it in many disciplinary fields of engineering and more broadly in many disciplines where data-driven approaches make full sense.

Appendix A. POD reduced-order models

In this appendix, we give the expressions of the POD reduced-order models for the Burgers' equation (Section 5.1) and for the cylinder wake flow (Section 5.2). For that, we apply the procedure described in Section 2.2, considering for decomposition on the POD basis of a velocity field $\boldsymbol{u}$, the expression:

$$\boldsymbol{u}(\boldsymbol{x},t) = \boldsymbol{u}_m(\boldsymbol{x}) + \sum_{j=1}^{q} a_j(t)\boldsymbol{\Phi}_j(\boldsymbol{x}) \tag{A.1}$$

where $\boldsymbol{u}_m$ is the mean field, q the number of modes kept in the expansion and a_j, the $j-$th coordinate of $\boldsymbol{a}(t)$.

The reduced order models are given by

$$\sum_{j=1}^{q} M_{ij}\frac{da_j(t)}{dt} = A_i + \sum_{j=1}^{q} B_{ij}a_j(t) + \sum_{j=1}^{q}\sum_{k=1}^{q} C_{ijk}a_j(t)a_k(t) \tag{A.2}$$

$$\sum_{j=1}^{q} M_{ij}a_j(0) = \left(\boldsymbol{\Phi}_i, \boldsymbol{u}(\boldsymbol{x},0) - \boldsymbol{u}_m(\boldsymbol{x})\right)_\Omega \qquad i = 1, \cdots q$$

where $M_{ij} = (\boldsymbol{\Phi}_i, \boldsymbol{\Phi}_j)_\Omega$ is the mass matrix for the $L^2(\Omega)$ inner product. The coefficients A_i, B_{ij} and C_{ijk} depend on the equations governing the case in question. Their expressions are given in the next two sections.

Appendix A.1. Coefficients for the Burgers' equation

$$A_i = -\left(\Phi_i, u_m \frac{\partial u_m}{\partial x}\right)_\Omega + \frac{1}{\mathsf{Re}}\left(\Phi_i, \frac{\partial^2 u_m}{\partial x^2}\right)_\Omega \tag{A.3}$$

$$B_{ij} = -\left(\Phi_i, u_m \frac{\partial \Phi_j}{\partial x}\right)_\Omega - \left(\Phi_i, \Phi_j \frac{\partial u_m}{\partial x}\right)_\Omega + \frac{1}{\mathsf{Re}}\left(\Phi_i, \frac{\partial^2 \Phi_j}{\partial x^2}\right)_\Omega \tag{A.4}$$

$$C_{ijk} = -\left(\Phi_i, \Phi_j \frac{\partial \Phi_k}{\partial x}\right)_\Omega \tag{A.5}$$

Appendix A.2. Coefficients for the cylinder wake flow

$$\begin{aligned} A_i = &-\left(\boldsymbol{\Phi}_i, (\boldsymbol{u}_m \cdot \boldsymbol{\nabla})\, \boldsymbol{u}_m\right)_\Omega - \frac{1}{\mathsf{Re}}\left(\overline{\overline{\boldsymbol{\nabla}}}\boldsymbol{\Phi}_i, \overline{\overline{\boldsymbol{\nabla}}}\boldsymbol{u}_m\right)_\Omega \\ &+ \frac{1}{\mathsf{Re}}\left[\left(\overline{\overline{\boldsymbol{\nabla}}}\boldsymbol{u}_m\right)^{\mathsf{T}} \boldsymbol{\Phi}_i\right]_{\partial\Omega} \end{aligned} \tag{A.6}$$

$$\begin{aligned} B_{ij} = &-\left(\boldsymbol{\Phi}_i, (\boldsymbol{u}_m \cdot \boldsymbol{\nabla})\, \boldsymbol{\Phi}_j\right)_\Omega - \left(\boldsymbol{\Phi}_i, (\boldsymbol{\Phi}_j \cdot \boldsymbol{\nabla})\, \boldsymbol{u}_m\right)_\Omega \\ &- \frac{1}{\mathsf{Re}}\left(\overline{\overline{\boldsymbol{\nabla}}}\boldsymbol{\Phi}_i, \overline{\overline{\boldsymbol{\nabla}}}\boldsymbol{\Phi}_j\right)_\Omega + \frac{1}{\mathsf{Re}}\left[\left(\overline{\overline{\boldsymbol{\nabla}}}\boldsymbol{\Phi}_j\right)^{\mathsf{T}} \boldsymbol{\Phi}_i\right]_{\partial\Omega} \end{aligned} \tag{A.7}$$

$$C_{ijk} = -\left(\boldsymbol{\Phi}_i, (\boldsymbol{\Phi}_j \cdot \boldsymbol{\nabla})\, \boldsymbol{\Phi}_k\right)_\Omega \tag{A.8}$$

where

$$\left(\overline{\overline{P}}, \overline{\overline{Q}}\right)_\Omega = \int_\Omega \overline{\overline{P}} : \overline{\overline{Q}}\, \mathrm{d}\boldsymbol{x} = \sum_{i,j=1}^{N_c} \int_\Omega P_{ij} Q_{ij}\, \mathrm{d}\boldsymbol{x} \quad \text{and} \quad [\boldsymbol{u}]_{\partial\Omega} = \int_{\partial\Omega} \boldsymbol{u} \cdot \boldsymbol{n}\, \mathrm{d}\boldsymbol{x}$$

with $\boldsymbol{n}$, the outward unit normal vector.

Algorithm 2 Stiefel Exponential map [32]

Input: Base point $U \in \mathcal{S}(n,r)$, tangent vector $\Delta \in \mathrm{T}_U\mathcal{S}(n,r)$

1: $\mathrm{A} := U^{\mathsf{T}}\Delta$ horizontal component, skew

2: $\mathrm{QR} := \Delta - U\mathrm{A}$ Thin qr-decomposition of normal component of Δ

3: $\begin{pmatrix} \mathrm{A} & -\mathrm{R}^{\mathsf{T}} \\ \mathrm{R} & 0 \end{pmatrix} = \mathrm{T}\Lambda\mathrm{T}^{H} \in \mathbb{R}^{2r\times 2r}$ EVD of skew-symmetric matrix

4: $\begin{pmatrix} \mathrm{M} \\ \mathrm{N} \end{pmatrix} := \mathrm{T}\exp_m(\Lambda)\,\mathrm{T}^{H} \begin{pmatrix} \mathrm{I}_r \\ \mathbf{0} \end{pmatrix} \in \mathbb{R}^{2r\times r}$

Output: $\tilde{U} := \mathrm{Exp}_U^{\mathcal{S}}(\Delta) = U\mathrm{M} + \mathrm{QN} \in \mathcal{S}(n,r)$

Appendix B. Stiefel exponential map

The algorithmic procedure is summarized in Alg. 2. EVD stands for Eigen-Value Decomposition and $\exp_m$ denotes the standard matrix decomposition. The computational effort is $\mathcal{O}(nr^2)$.

Appendix C. Stiefel logarithm map

The algorithmic procedure is summarized in Alg. 3. After Zimmermann [23], this algorithm is guaranteed to converge if the input data points U and $\tilde{U}$ are at most of a Euclidean distance $\|U - \tilde{U}\|_2 \leq 0.09$ apart. The computational effort is $\mathcal{O}(nr^2)$.

Appendix D. Grassmann exponential map

The algorithmic procedure is summarized in Alg. 4. The computational effort is $\mathcal{O}(nr^2)$. By omitting the rightmost V^{T} in line 2 of 4, we still have a representative of the same equivalence class [32]. This is the expression that is used in Amsallem [37] and in many papers using Grassmann interpolation.

Algorithm 3 Stiefel Logarithm map [23]

Input: Base point $U \in \mathcal{S}(n,r)$, $\tilde{U} \in \mathcal{S}(n,r)$ in the neighbourhood of U, $\tau > 0$ convergence threshold

1: $\mathrm{M} := U^{\mathsf{T}}\tilde{U} \in \mathbb{R}^{r\times r}$

2: $\mathrm{QN} := \tilde{U} - U\mathrm{M} \in \mathbb{R}^{n\times r}$ Thin qr-decomposition of normal component of $\tilde{U}$

3: $\mathrm{V}_0 := \begin{pmatrix} \mathrm{M} & \mathrm{X}_0 \\ \mathrm{N} & \mathrm{Y}_0 \end{pmatrix} \in O(2r)$ Compute orthogonal completion of $\begin{pmatrix} \mathrm{M} \\ \mathrm{N} \end{pmatrix}$

4: **for** $k = 0, 1, 2, \cdots$ **do**

5: $\begin{pmatrix} \mathrm{A}_k & -\mathrm{B}_k^{\mathsf{T}} \\ \mathrm{B}_k & \mathrm{C}_k \end{pmatrix} := \log_m(\mathrm{V}_k)$ Matrix logarithm of orthogonal matrix

6: **if** $\|\mathrm{C}_k\|_2 \leq \tau$ **then** break

7: **end if**

8: $\Phi_k := \exp_m(-\mathrm{C}_k)$ matrix exponential of skew matrix

9: $\mathrm{V}_{k+1} := \mathrm{V}_k\mathrm{W}_k$ where $\mathrm{W}_k := \begin{pmatrix} \mathrm{I}_r & 0 \\ 0 & \Phi_k \end{pmatrix}$

10: **end for**

Output: $\Delta := \mathrm{Log}_U^{\mathcal{S}}(\tilde{U}) = U\mathrm{A}_k + \mathrm{QB}_k \in \mathrm{T}_U\mathcal{S}(n,r)$

Algorithm 4 Grassman Exponential map [see 32, Section 2.5.1]

Input: Base point $\mathcal{U} = [U] \in \mathcal{G}(n,r)$, where $U \in \mathcal{S}(n,r)$, tangent vector $\Delta \in \mathrm{T}_U\mathcal{G}(n,r)$

1: $\mathrm{Q}\Sigma\mathrm{V}^{\mathsf{T}} \overset{SVD}{:=} \Delta$ with $Q \in \mathcal{S}(n,r)$ thin SVD of tangent vector

2: $\tilde{U} := U\mathrm{V}\cos(\Sigma)\mathrm{V}^{\mathsf{T}} + \mathrm{Q}\sin(\Sigma)\mathrm{V}^{\mathsf{T}}$ cos and sin act only on diagonal entries

Output: $\tilde{\mathcal{U}} := \mathrm{Exp}_{\mathcal{U}}^{\mathcal{G}}(\Delta) = [\tilde{U}] \in \mathcal{G}(n,r)$

Appendix E. Grassmann logarithm map

Algorithm 5 Grassman Logarithm map [see 44, §3.8, p. 210]

Input: Base point $\mathcal{U} = [U] \in \mathcal{G}(n,r)$, with $U \in \mathcal{S}(n,r)$, $\tilde{\mathcal{U}} = [\tilde{U}] \in \mathcal{G}(n,r)$ with $\tilde{U} \in \mathcal{S}(n,r)$

1: $\mathrm{M} := U^{\mathsf{T}}\tilde{U}$

2: $\mathrm{L} := \left(I - UU^{\mathsf{T}}\right)\tilde{U}\mathrm{M}^{-1} = \tilde{U}\mathrm{M}^{-1} - U$

3: $\mathrm{Q}\Sigma\mathrm{V}^{\mathsf{T}} \overset{SVD}{:=} L$ thin SVD

4: $\Delta := \mathrm{Q}\arctan(\Sigma)\mathrm{V}^{\mathsf{T}}$ arctan acts only on diagonal entries

Output: $\Delta := \mathrm{Log}^{\mathcal{G}}_{\mathcal{U}}(\tilde{\mathcal{U}}) \in \mathrm{T}_{\mathcal{U}}\mathcal{G}(n,r)$

The algorithmic procedure is summarized in Alg. 5. The composition $\mathsf{Exp}^{\mathcal{G}}_{[U]} \circ \mathsf{Log}^{\mathcal{G}}_{[U]}$ is the identity on $\mathcal{G}(n,r)$, wherever it is defined. However, on the level of the actual matrix representatives, the operation

$$\left(\mathsf{Exp}^{\mathcal{G}}_{[U]} \circ \mathsf{Log}^{\mathcal{G}}_{[U]}\right)\left([\tilde{\mathcal{U}}_{\mathsf{in}}]\right) = [\tilde{\mathcal{U}}_{\mathsf{out}}]$$

produces a matrix $\tilde{\mathcal{U}}_{\mathsf{out}} \neq \tilde{\mathcal{U}}_{\mathsf{in}}$. The input matrix can be achieved via a so-called Procrustes-type preprocessing step [see 28, for more information and an algorithm using it].

Appendix F. Analytical solutions of the 1D Burgers' equation

The analytical solution of the 1D Burgers' equation (8) is given [see 45, §4.1] by

$$u_a(x,t) = \frac{2\pi\nu}{L}\frac{\sum_{n=1}^{\infty} a_n n \sin(n\pi x/L)\exp(-n^2\pi^2\nu t/L^2)}{a_0 + \sum_{n=1}^{\infty} a_n \cos(n\pi x/L)\exp(-n^2\pi^2\nu t/L^2)},$$

where the coefficients a_n are:

$$a_0 = \frac{1}{L}\int_0^L \exp\left[-\frac{u_0 L}{2\pi\nu}\left(1-\cos(\frac{\pi x}{L})\right)\right]\,\mathrm{d}x,$$
$$a_n = \frac{2}{L}\int_0^L \exp\left[-\frac{u_0 L}{2\pi\nu}\left(1-\cos(\frac{\pi x}{L})\right)\right]\times\cos(\frac{n\pi x}{L})\,\mathrm{d}x.$$

Appendix G. Cylinder wake flow

Appendix G.1. Boundary conditions

The problem (11) is solved with the following boundary conditions (see Fig. 10). At the left boundary of the domain, an inlet condition is applied:

$$(u, v) = (1, 0) \;\text{ on }\; \Gamma_1\,.$$

At the side-walls of the channel, zero shear stress conditions are imposed:

$$\frac{\partial u}{\partial y} = 0,\; v = 0 \;\text{ on }\; \Gamma_2,\; \Gamma_4\,.$$

On the surface of the cylinder, no-slip boundary condition is considered:

$$(u, v) = (0, 0) \;\text{ on }\; \Gamma_c\,.$$

Provided that the right boundary is placed at a sufficient distance downstream of the cylinder, the condition

$$p = 0 \;\text{ on }\; \Gamma_3$$

is found to be an acceptable approximation for conditions at the outflow.

Appendix G.2. Variational formulation

The variational formulation of the Navier-Stokes equations require the definition of different function spaces. Let $L^2(\Omega)$ denote the space of square integrable function over the domain Ω, we introduce successively:

$$L_0^2(\Omega) = \left\{ v \in L^2(\Omega) \ / \ \int_\Omega v \ \mathrm{d}x = 0 \right\} ,$$
$$H^1(\Omega) = \left\{ v \in L^2(\Omega) \ / \ \nabla v \in L^2(\Omega) \right\} ,$$

and

$$H_0^1(\Omega, \Gamma) = \left\{ v \in H^1(\Omega) \ / \ v = 0 \text{ on } \Gamma \right\} .$$

$H^1(\Omega)$ is the Hilbert space of vector functions with first derivatives in $L^2(\Omega)$ and $H_0^1(\Omega, \Gamma)$ is the set of $H^1(\Omega)$ functions with zero values on the domain boundary Γ.

Hereafter, we denote $\boldsymbol{H} = H_0^1(\Omega, \Gamma)^2$ and $\boldsymbol{L} = L_0^2(\Omega)$. Let $\boldsymbol{v}$ and q be two test functions for velocity and pressure fields, respectively. The variational formulation consists in finding $\boldsymbol{u}^{n+1}$ and p^{n+1}, velocity and pressure at the time instant $n+1$, such that:

$$\begin{aligned} \int_\Omega \Bigg[& \frac{\alpha_0^r}{\Delta t} \left(\boldsymbol{u}^{n+1} \cdot \boldsymbol{v} \right) + \frac{1}{\mathsf{Re}} \left(\boldsymbol{\nabla} \boldsymbol{u}^{n+1} : \boldsymbol{\nabla} \boldsymbol{v} \right) - q \boldsymbol{\nabla} \cdot \boldsymbol{u}^{n+1} - p^{n+1} \boldsymbol{\nabla} \cdot \boldsymbol{v} \\ & + \boldsymbol{v} \cdot \left(\left(\boldsymbol{u}^{n+1} \cdot \boldsymbol{\nabla} \right) \boldsymbol{u}^n + \left(\boldsymbol{u}^n \cdot \boldsymbol{\nabla} \right) \boldsymbol{u}^{n+1} \right) - \epsilon \, p^{n+1} q \Bigg] \mathrm{d}x \\ = \int_\Omega & \left[\boldsymbol{v} \cdot \left(\left(\boldsymbol{u}^n \cdot \boldsymbol{\nabla} \right) \boldsymbol{u}^n \right) + \boldsymbol{f} \cdot \boldsymbol{v} - \sum_{i=1}^{r} \frac{\alpha_i^r}{\Delta t} \left(\boldsymbol{u}^{n+1-i} \cdot \boldsymbol{v} \right) \right] \mathrm{d}x, \quad \forall \boldsymbol{v} \in \boldsymbol{H}, \forall q \in \boldsymbol{L} , \end{aligned} \tag{G.1}$$

where Δt is the time step of integration and ϵ the coefficient of the penalty method. The coefficients of the BDF scheme of order $r = 1$, 2 or 3 are given by:

$$\boldsymbol{\alpha}^1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \quad ; \quad \boldsymbol{\alpha}^2 = \begin{pmatrix} \frac{3}{2} \\ -2 \\ \frac{1}{2} \end{pmatrix} \quad ; \quad \boldsymbol{\alpha}^3 = \begin{pmatrix} \frac{11}{6} \\ -3 \\ \frac{3}{2} \\ -\frac{1}{3} \end{pmatrix} .$$

The unstructured mesh used to solve the variational formulation (G.1) is shown in Figure G.19.

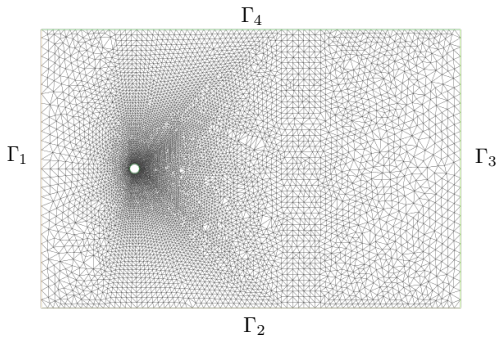


Figure G.19: Spatial mesh used for the 2D-cylinder wake flow simulation. The mesh is built using FreeFem++'s `builmesh` command. The number of mesh points on the different boundaries is: 30 on Γ_1 and Γ_3, 75 on Γ_2 and Γ_4, 45 on Γ_c (cylinder). The cylinder diameter is $D = 0.1$.

References

[1] P. Benner, S. Gugercin, K. Willcox, A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems, SIAM Review 57 (2015) 483–531. URL: http://dx.doi.org/10.1137/130932715. doi:10.1137/130932715. arXiv:http://dx.doi.org/10.1137/130932715.

[2] K. Taira, S. L. Brunton, S. T. M. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, L. S. Ukeiley, Modal analysis of fluid flows: An overview, AIAA Journal 55 (2017) 4013–4041. doi:10.2514/1.J056060. arXiv:1702.01453.

[3] G. Rozza, G. Stabile, F. Ballarin, Advanced Reduced Order Methods and Applications in Computational Fluid Dynamics, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2022. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611977257. doi:10.1137/1.9781611977257. arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611977257.

[4] J. L. Lumley, The structure of inhomogeneous turbulent flows, Atmospheric Turbulence and Radio Wave Propagation (1967). URL: https://cir.nii.ac.jp/crid/1571980075051475712.

[5] L. Sirovich, Turbulence and the dynamics of coherent structures part iii: Dynamics and scaling, Quarterly of Applied Mathematics 45 (1987) 583–590. URL: http://www.jstor.org/stable/43637459.

[6] L. Cordier, M. Bergmann, Proper Orthogonal Decomposition: an overview, in: Lecture series 2002-04, 2003-03 and 2008-01 on post-processing of experimental and numerical data, Von Kármán Institute for Fluid Dynamics, 2008, pp. 1–46. ISBN 978-2-930389-80-X.

[7] M. Bergmann, L. Cordier, J.-P. Brancher, Optimal rotary control of the cylinder wake using POD Reduced Order Model, Phys. Fluids 17 (2005) 097101:1–21.

[8] M. Bergmann, L. Cordier, Optimal control of the cylinder wake in the laminar regime by Trust-Region methods and POD Reduced Order Models, J. Comp. Phys. 227 (2008) 7813–7840.

[9] P.-A. Absil, R. Mahony, R. Sepulchre, Optimization Algorithms on Matrix Manifolds, Princeton University Press, USA, 2007.

[10] D. Amsallem, C. Farhat, Interpolation method for adapting reduced-order models and application to aeroelasticity, AIAA JOURNAL 46 (2008) 1803–1813. doi:10.2514/1.35374.

[11] D. Amsallem, J. Cortial, K. Carlberg, C. Farhat, A method for interpolating on manifolds structural dynamics reduced-order models, International Journal for Numerical Methods in Engineering 80 (2009) 1241–1258. doi:10.1002/nme.2681.

[12] M. Oulghelou, C. Allery, Optimal control based on adaptive model reduction approach to control transfer phenomena, in: S. Sivasundaram (Ed.), ICNPAA 2016 World Congress: 11th International Conference on Mathematical Problems in Engineering, Aerospace and

Sciences, volume 1798 of *AIP Conference Proceedings*, 2017, p. 020119. URL: https://doi.org/10.1063/1.4972711. doi:10.1063/1.4972711. arXiv:https://pubs.aip.org/aip/acp/article-pdf/doi/10.1063/1.4972711/13734799/

[13] M. Oulghelou, C. Allery, Adaptative reduced order model to control non linear partial differential equations, in: M. Papadrakakis, E. Onate, B. Schrefler (Eds.), Proceedings of the VII International Conference on Computational Methods for Coupled Problems in Science and Engineering, 2017, pp. 800–811. URL: http://hdl.handle.net/2117/190688.

[14] M. Oulghelou, C. Allery, A fast and robust sub-optimal control approach using reduced order model adaptation techniques, Applied Mathematics and Computation 333 (2018) 416–434. URL: https://www.sciencedirect.com/science/article/pii/S0096300318302728. doi:https://doi.org/10.1016/j.amc.2018.03.091.

[15] K. Vlachas, K. Tatsis, K. Agathos, A. R. Brink, E. Chatzi, A local basis approximation approach for nonlinear parametric model order reduction, Journal of Sound and Vibration 502 (2021) 116055. URL: https://www.sciencedirect.com/science/article/pii/S0022460X21001279. doi:https://doi.org/10.1016/j.jsv.2021.116055.

[16] M. Oulghelou, C. Allery, Non intrusive method for parametric model order reduction using a bi-calibrated interpolation on the Grassmann manifold, Journal of Computational Physics 426 (2021) 109924. URL: https://www.sciencedirect.com/science/article/pii/S0021999120306987. doi:https://doi.org/10.1016/j.jcp.2020.109924.

[17] O. Friderikos, E. Baranger, M. Olive, D. Neron, On the stability of POD basis interpolation on Grassmann manifolds for parametric model order reduction, Computational Mechanics 70 (2022) 181–204. URL: https://doi.org/10.1007/s00466-022-02163-0. doi:10.1007/s00466-022-02163-0.

[18] M. Oulghelou, C. Allery, R. Mosquera, Parametric reduced order models based on a riemannian barycentric interpolation, International Journal for Numerical Methods in Engineering 122 (2021) 6623–6640. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6805. doi:https://doi.org/10.1002/nme.6805. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.6805.

[19] E. Massart, P.-A. Absil, Quotient geometry with simple geodesics for the manifold of fixed-rank positive-semidefinite matrices, SIAM Journal on Matrix Analysis and Applications 41 (2020) 171–198. doi:10.1137/18M1231389.

[20] R. Zimmermann, R. Bergmann, Multivariate hermite interpolation on riemannian manifolds, SIAM Journal on Scientific Computing 46 (2024) A1276–A1297. URL: https://doi.org/10.1137/22M1541071. doi:10.1137/22M1541071. arXiv:https://doi.org/10.1137/22M1541071.

[21] M. Oulghelou, C. Beghein, C. Allery, A surrogate optimization approach for inverse problems: Application to turbulent mixed-convection flows, Computers & Fluids 241 (2022) 105490. URL: https://www.sciencedirect.com/science/article/pii/S0045793022001311. doi:https://doi.org/10.1016/j.compfluid.2022.105490.

[22] R. Zimmermann, K. Debrabant, Parametric model reduction via interpolating orthonormal bases, in: F. A. Radu, K. Kumar, I. Berre, J. M. Nordbotten, I. S. Pop (Eds.), Numerical Mathematics and Advanced Applications ENUMATH 2017, Springer International Publishing, Cham, 2019, pp. 683–691.

[23] R. Zimmermann, A Matrix-Algebraic Algorithm for the Riemannian Logarithm on the Stiefel Manifold under the Canonical Metric, SIAM Journal on Matrix Analysis and Applications 38 (2017) 322–342. URL: https://doi.org/10.1137/16M1074485. doi:10.1137/16M1074485. arXiv:https://doi.org/10.1137/16M1074485.

[24] S. Volkwein, Model Reduction Using Proper Orthogonal Decomposition, 2011. URL: https://www.math.uni-konstanz.de/numerik/personen/volkwein/teaching/POD-Vorlesung.pdf.

[25] L. Cordier, M. Bergmann, Two typical applications of POD: coherent structures eduction and reduced order modelling, in: Lecture series 2002-04, 2003-03 and 2008-01 on post-processing of experimental and numerical data, Von Kármán Institute for Fluid Dynamics, 2008, pp. 1–60. ISBN 978-2-930389-80-X.

[26] L. Cordier, B. Abou El Majd, J. Favier, Calibration of POD Reduced-Order models using Tikhonov regularization, Int. J. Numer. Meth. Fluids 63 (2010).

[27] T. Bendokat, R. Zimmermann, P. A. Absil, A grassmann manifold

handbook: Basic geometry and computational aspects, arXiv (2020). arXiv:2011.13699.

[28] R. Zimmermann, Manifold interpolation and model reduction, in: P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, L. M. Silveira (Eds.), System- and Data-Driven Methods and Algorithms, volume 1 of *Model Order Reduction*, De Gruyter, Berlin, Boston, 2021, pp. 229–274. URL: https://doi.org/10.1515/9783110498967. doi:doi:10.1515/9783110498967.

[29] N. Boumal, An Introduction to Optimization on Smooth Manifolds, Cambridge University Press, 2023. URL: https://www.nicolasboumal.net/book.

[30] W. Basener, Topology and Its Applications, Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts, Wiley, 2013.

[31] B. Mendelson, Introduction to Topology: Third Edition, Dover Books on Mathematics, Dover Publications, 2012. URL: https://books.google.fr/books?id=FWFmoEUJSwkC.

[32] A. Edelman, T. A. Arias, S. T. Smith, The Geometry of Algorithms with Orthogonality Constraints, SIAM Journal on Matrix Analysis and Applications 20 (1998) 303–353. URL: https://doi.org/10.1137/S0895479895290954. doi:10.1137/S0895479895290954. arXiv:https://doi.org/10.1137/S0895479895290954.

[33] R. Zimmermann, K. Hüper, Computing the Riemannian Logarithm on the Stiefel Manifold: Metrics, Methods, and Perfor-

mance, SIAM Journal on Matrix Analysis and Applications 43 (2022) 953–980. URL: https://doi.org/10.1137/21M1425426. doi:10.1137/21M1425426. arXiv:https://doi.org/10.1137/21M1425426.

[34] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, L. M. Silveira (Eds.), Model Order Reduction, De Gruyter, Berlin, Boston, 2021. URL: https://doi.org/10.1515/9783110498967. doi:doi:10.1515/9783110498967.

[35] R. Zimmermann, Hermite Interpolation and Data Processing Errors on Riemannian Matrix Manifolds, SIAM Journal on Scientific Computing 42 (2020) A2593–A2619. URL: https://doi.org/10.1137/19M1282878. doi:10.1137/19M1282878. arXiv:https://doi.org/10.1137/19M1282878.

[36] J. F. Céspedes Moreno, J. P. Murcia León, S. J. Andersen, Convergence and efficiency of global bases using proper orthogonal decomposition for capturing wind turbine wake aerodynamics, Wind Energy Science Discussions 2024 (2024) 1–21. URL: https://wes.copernicus.org/preprints/wes-2024-81/. doi:10.5194/wes-2024-81.

[37] D. Amsallem, Interpolation on manifolds of CFD-based fluid and finite element-based structural reduced-order models for on-line aeroelastic predictions, Ph.D. thesis, Stanford university, 2010.

[38] O. Friderikos, M. Olive, E. Baranger, D. Sagris, C. N. David, A Space-Time POD Basis Interpolation on Grassmann Manifolds for Parametric Simulations of Rigid-Viscoplastic FEM, MATEC Web Conf. 318 (2020) 01043.

URL: https://doi.org/10.1051/matecconf/202031801043. doi:10.1051/matecconf/202031801043.

[39] L. Cordier, B. R. Noack, G. Tissot, G. Lehnasch, J. Delville, M. Balajewicz, G. Daviller, R. K. Niven, Identification strategies for model-based control, Exp. Fluids, special issue Flow Control 54 (2013) 1580.

[40] F. Hecht, New development in FreeFem++, J. Numer. Math. 20 (2012) 251–265. URL: https://freefem.org/.

[41] M. Schäfer, S. Turek, F. Durst, E. Krause, R. Rannacher, Benchmark Computations of Laminar Flow Around a Cylinder, in: E. H. Hirschel (Ed.), Flow Simulation with High-Performance Computers II: DFG Priority Research Programme Results 1993–1995, Vieweg+Teubner Verlag, Wiesbaden, 1996, pp. 547–566. URL: https://doi.org/10.1007/978-3-322-89849-4_39. doi:10.1007/978-3-322-89849-4_39.

[42] F. Chinesta, A. Huerta, G. Rozza, K. Willcox, Model Reduction Methods, John Wiley & Sons, Ltd, 2017, pp. 1–36. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119176817.ecm2110. doi:https://doi.org/10.1002/9781119176817.ecm2110. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119176817.ecm2110.

[43] R. Mosquera, A. Hamdouni, A. El Hamidi, C. Allery, POD basis interpolation via Inverse Distance Weighting on Grassmann manifolds, Discrete and Continuous Dynamical Systems - S 12 (2019) 1743–1759. URL: /article/id/8d6c05af-52ae-486f-bc89-083656124e62. doi:10.3934/dcdss.2019115.

[44] P.-A. Absil, R. Mahony, R. Sepulchre, Riemannian geometry of grassmann manifolds with a view on algorithmic computation, Acta Applicandae Mathematicae 80 (2004) 199–220. URL: http://www.montefiore.ulg.ac.be/services/stochastic/pubs/2004/AMS04.

[45] M. P. Bonkile, A. Awasthi, C. Lakshmi, V. Mukundan, V. S. Aswin, A systematic literature review of Burgers' equation with recent advances, Pramana 90 (2018). URL: https://doi.org/10.1007/s12043-018-1559-4. doi:10.1007/s12043-018-1559-4.

Contents