A Non-Intrusive Space-Time Interpolation from Compact Stiefel Manifolds of Parametrized Rigid-Viscoplastic FEM Problems

Orestis Friderikos, Marc Olive, Emmanuel Baranger, Dimitrios Sagris, Constantine David

myjournal manuscript No.
(will be inserted by the editor)

A Non-Intrusive Space-Time Interpolation from Compact Stiefel Manifolds of Parametrized Rigid-Viscoplastic FEM Problems

Orestis Friderikos[1], Marc Olive[1], Emmanuel Baranger[1], Dimitris Sagris[2], Constantine David[2]

[1] Université Paris-Saclay, ENS Paris-Saclay, CNRS, LMT - Laboratoire de Mécanique et Technologie, 91190, Gif-sur-Yvette, France

[2] Mechanical Engineering Department, Laboratory of Manufacturing Technology & Machine Tools, International Hellenic University, GR-62124 Serres Campus, Greece.

Received: date / Revised version: date

Contents

1 Introduction 4
2 Space–Time POD, Grassmann and compact Stiefel manifolds 10
2.1 Riemannian geometry on Grassmann manifolds 13
2.2 Target Algorithm on compact Stiefel manifolds 19
3 Space-Time Interpolation on compact Stiefel manifolds 23
3.1 Oriented SVD on generic matrices 25

Send offprint requests to: Marc Olive
Correspondence to: marc.olive@math.cnrs.fr

17 3.2 Space–Time interpolation algorithm 29
18 4 Rigid-Viscoplastic FEM Formulation 33
19 4.1 Governing Field Equations 34
20 4.2 Variational form 35
21 4.3 Discretization and iteration 36
22 4.4 Heat Transfer Analysis 38
23 4.5 Computational Procedure for Thermo-Mechanical Analysis 39
24 5 Numerical Investigations 40
25 5.1 Mechanical field 42
26 5.2 Temperature field 45
27 5.3 Computational complexity 46
28 6 Conclusions 47

29 **Abstract** This work aims to interpolate parametrized Reduced Order Model (ROM)
30 basis constructed via the Proper Orthogonal Decomposition (POD) to derive a robust
31 ROM of the system's dynamics for an unseen target parameter value. A novel non-intrusive
32 Space-Time (ST) POD basis interpolation scheme is proposed, for which we define ROM
33 spatial and temporal basis *curves on compact Stiefel manifolds*. An interpolation is finally
34 defined on a *mixed part* encoded in a square matrix directly deduced using the space part,
35 the singular values and the temporal part, to obtain an interpolated snapshot matrix,
36 keeping track of accurate space and temporal eigenvectors. Moreover, in order to establish
37 a well-defined curve on the compact Stiefel manifold, we introduce a new procedure, the so-
38 called oriented SVD. Such an oriented SVD produces unique right and left eigenvectors for
39 generic matrices, for which all singular values are distinct. It is important to notice that the
40 ST POD basis interpolation does not require the construction and the subsequent solution

of a reduced-order FEM model as classically is done. Hence it is avoiding the bottleneck of standard POD interpolation which is associated with the evaluation of the nonlinear terms of the Galerkin projection on the governing equations. As a proof of concept, the proposed method is demonstrated with the adaptation of rigid-thermoviscoplastic finite element ROMs applied to a typical nonlinear open forging metal forming process. Strong correlations of the ST POD models with respect to their associated high-fidelity FEM counterpart simulations are reported, highlighting its potential use for near real-time parametric simulations using off-line computed ROM POD databases.

Notations

$\mathrm{Mat}_{n,p}(\mathbb{R})$	Set of $n \times p$ matrices in $\mathbb{R}$	
$\mathbf{I}_p$	Identity matrix in $\mathrm{Mat}_{p,p}(\mathbb{R})$	
$[\mathbf{y}_1, \ldots, \mathbf{y}_p]$	Matrix in $\mathrm{Mat}_{n,p}(\mathbb{R})$	Matrix with column vectors $\mathbf{y}_i \in \mathbb{R}^n$
$\mathrm{O}(p)$	Orthogonal group on $\mathbb{R}^p$	$\left\{Q \in \mathrm{Mat}_{p,p}(\mathbb{R}), \quad Q^T Q = \mathbf{I}_p\right\}$
$\mathcal{G}(p,n)$	Grassmann manifold	Set of p linear subspaces in $\mathbb{R}^n$
$\pi^{-1}(\mathbf{m})$	Fiber at $\mathbf{m} \in \mathcal{G}(p,n)$	If $\mathbf{y}_1, \ldots, \mathbf{y}_p$ is an orthonormal basis of $\mathbf{m}$ $\pi^{-1}(\mathbf{m}) = \{\mathbf{Y}Q, \quad Q \in \mathrm{O}(p), \quad \mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_p]\}$
$\mathcal{T}_{\mathbf{m}} := \mathcal{T}_{\mathbf{m}}\mathcal{G}(p,n)$	Tangent space of $\mathcal{G}(p,n)$ at $\mathbf{m}$	For $\mathbf{Y} \in \pi^{-1}(\mathbf{m})$, one model of $\mathcal{T}_{\mathbf{m}}$ is $\left\{\mathbf{Z} \in \mathrm{Mat}_{n,p}(\mathbb{R}), \quad \mathbf{Z}^T\mathbf{Y} = 0\right\}$
$\mathcal{S}t(p,n)$	Stiefel manifold	Set of ordered p-tuples independent vectors in $\mathbb{R}^n$
$\mathcal{S}t_c(p,n)$	Compact Stiefel manifold	Set of ordered p-tuples of orthonormal vectors in $\mathbb{R}^n$ $\mathcal{S}t_c(p,n) = \left\{\mathbf{Y} \in \mathrm{Mat}_{n,p}(\mathbb{R}), \quad \mathbf{Y}^T\mathbf{Y} = \mathbf{I}_p\right\}$
$\mathrm{Hor}_{\mathbf{Y}}$	Horizontal space at $\mathbf{Y}$	$\mathrm{Hor}_{\mathbf{Y}} := \left\{\mathbf{Z} \in \mathrm{Mat}_{n,p}(\mathbb{R}), \quad \mathbf{Z}^T\mathbf{Y} = \mathbf{0}\right\}$
$v \in \mathcal{T}_{\mathbf{m}}$	Velocity vector on the tangent plane $\mathcal{T}_{\mathbf{m}}$	Represented by a horizontal lift $\mathbf{Z} \in \mathrm{Hor}_{\mathbf{Y}}$, with $\mathbf{Y} \in \pi^{-1}(\mathbf{m})$
$\mathbf{S}^{(i)}$	Snapshot matrix	$\mathbf{S}^{(i)} \in \mathrm{Mat}_{n,m}(\mathbb{R})$ corresponding to parameter value λ_i

1 Introduction

Computational metal forming has been widely used in academic laboratories and the manufacturing industry over the last decades, becoming nowadays a mature, well established

technology. Nevertheless, new challenging fields are emerging, among others, uncertainty quantification, optimization of processes and parameter identification in design analysis [1, 2]. One of the key challenging topics mentioned in [1] is the introduction of Model Order Reduction (MOR) methods to combat the high computational cost, which is also of paramount interest in the above-mentioned fields. Moreover, due to the multiple sources of strong non-linearities inherent in manufacturing problems, design optimization and multi-parametric studies of large scale models turns out to be prohibitively expensive. Indeed, simulation of complex configurations can be intractable since the computational times can highly increase.

To this end, meta-model techniques are often used to tackle the computational burden. These rely on a manifold learning stage during which we need to capture the original space where the solution of the model problem lies. This data collection consists of solving the full-scale model for an ensemble of training data over the parametric range and is commonly referred to as the offline stage. Even though meta-models can speed up the simulation time, nevertheless their construction with standard computations based on full-order models is expensive.

Closely related to the concept of metamodeling, Reduced Order Models (ROMs) have been chosen to reduce the problem's dimensionality while at the same time maintaining solution accuracy. ROMs can decrease the computational complexity of large-scale systems, solving parametrized problems and offering the potential for near real-time analysis. The methods for building ROMs can be classified into two general families: *a priori* and *a posteriori* ones. The well known a priori MOR includes methods such as the Proper Generalized Decomposition (PGD) [3], and the *a priori* reduction method (APR) [3, 4].

The main characteristic of all these methods is that they do not require any precomputed ROMs. In the second class of methods, the reduced basis is built, *a posteriori*, from the state variable snapshots in the parametric space. One popular method is the POD [5,6,7], also known as Kharhunen-Loève Decomposition (KLD) [8,9], Singular Value Decomposition (SVD) [10] or Principal Component Analysis (PCA) [11,12,13,14].

For nonlinear systems, even though a Galerkin projection reduces the number of unknowns, however, the computational burden for obtaining the solution could still be high due to the computational costs involved in the evaluation of nonlinear terms. Hence, the nonlinear Galerkin projection principally leads to a ROM, but its evaluation could be more expensive than the corresponding one of the original problem. To this effect, to make the resulting ROMs computationally efficient, a sparse sampling method is used, also called hyper reduction, to mention among others, the missing point estimation (MPE) [15], the empirical interpolation method (EIM) [16], the discrete empirical interpolation method (DEIM) [17], the Gappy POD method [18], and the Gauss-Newton with approximated tensors (GNAT) method [19]. Thus, all these methods imply the solution of a new ROM FEM problem.

In the case of a parametric analysis using POD basis interpolation on Grassmann manifolds [20,21], the method starts with a training stage during which the problem is solved for several training points. Then, using the FEM solutions, the full-order field 'snapshots' are compressed using the POD to generate a ROM that is expected to reproduce the most characteristic dynamics of its high-fidelity counterpart solution. However, the relevant information is contained in the vector spaces generated by the (left or right) singular vectors of the snapshot matrices. Now, for a new parameter value, interpolation methods

have to be defined from such relevant *subspaces* spanned by the POD basis vectors [20]. Other approaches obviously could be considered, such as interpolations computed on the space of matrices of a fixed rank, whereby the mechanical origin of the problem imposes to consider the vector subspaces, and not the matrices themselves [21]. Nevertheless, such methods as the one of interpolation between two positive semidefinite matrices of fixed rank [22], can be used in the special case of interpolation between two precomputed ROM basis and a single parameter which is not of general interest in mechanical applications.

To interpolate between different vector spaces of the same dimension (encoded into the mode p of the POD), a Grassmann manifold [23] must be used, which is the set of p-dimensional subspaces of $\mathbb{R}^n$. Such a manifold is in fact a Riemannian manifold [24], so we can construct *geodesics* between two points, and use such geodesics to define a logarithm map to *linearize*, and conversely using the exponential map to return back to the Grassmann manifold. While an interpolation cannot be done directly on Grassmann manifolds, linearization allows computing such an interpolation, at least locally once a reference point has been selected [20,21]. To any new parameter value, thus we get a new subspace obtained from interpolation between all subspaces related to the spatial eigenvectors of the snapshot matrices. Another approach using inverse distance weighting was initiated in [21,25], but it also relies on several choices (as one of the weights). Furthermore, an extension of Neville-Aitken's algorithm to Grassmann manifolds which computes the Lagrange interpolation polynomial in a recursive way from the interpolation of two points was recently presented [26].

In the standard POD interpolation mentioned above [20], the spatial ROM basis corresponding to the target point is used to generate a ROM FEM, which is expected to

have a lower computational cost compared to the high-fidelity problem. The key idea in the *Space-Time* (ST) POD basis interpolation proposed by [27,28], is that the reduced spatial and temporal basis are considered separately, both defining points on two different Grassmann manifolds. However, such points are strongly related: a spatial vector directly corresponds to a temporal vector, and *vice versa*. From this, firstly we need to consider the p-tuples of spatial (and temporal) vectors, instead of the p-dimensional subspace, which defines points on an associated *compact Stiefel manifold*, strongly connected to Grassmann manifolds. Contrary to what is suggested in [28], we propose a different interpolation scheme, as we do not perform interpolation of the singular values, followed by spatial and temporal calibration. Instead, we exploit the dependence between the spatial and temporal parts. Indeed, using an interpolation algorithm defined on a Grassmann manifold, we derive *curves on a compact Stiefel manifold*, which are no longer interpolating, but which nevertheless allow us to obtain new singular vectors for the spatial part, and separately for the temporal part. Such space and temporal singular vectors finally are taken to define a *mixed part* on which a classical interpolation can be computed. In the end, we get in this way a ROM matrix corresponding to a new parameter value. Note that in order to obtain a well-defined curve on compact Stiefel manifolds, we have to introduce a new procedure, the so-called *oriented SVD*. Such an oriented SVD produces unique right and left eigenvectors for snapshot matrices, supposed to be *generic matrices*, for which all non–zero singular values are distinct.

The off-line stage in the ST approach consists of solving FEM problems which are corresponding to the training points of the given parameter. The on-line stage concerns the use of a curve defined on a compact Stiefel manifold to determine the spatial and

temporal ROM basis for the target point, in order to construct the related ROM snapshot matrix. In fact, the ST interpolation offers the advantage of reconstructing a snapshot matrix without relaunching ROM FEM computations. To this end, it results in near-real-time solutions due to direct matrix multiplications in the on-line stage.

We could also mention some other ST approaches [29,30,31,32], where neither Grassmann nor compact Stiefel manifolds are considered. For instance, an approximation of the spatial and temporal basis functions by linear interpolation of their modes is proposed in [29] to study the flow past a cylinder at low Reynolds numbers. A non-intrusive ROM approach for nonlinear parametrized time-dependent PDEs based on a two-level POD method by using Radial Basis Functions interpolation is presented in [30,33].

The method proposed in this work is applied to a coupled thermomechanical rigid visco-plastic (RVP) FEM analysis based on an incremental implicit approach [34,35,36, 37]. Note that the RVP formulation specifically is tailored for metal forming simulations, where the plastic flow is unconstrained and usually of finite magnitude, involving large strain-rates and high temperatures. In the present study, all simulations are performed by using an in-house Matlab code which consists of two independent FEM solvers. A mechanical solver for the viscoplastic deformation analysis [38] and a thermal solver for the heat transfer analysis. A staggered procedure is used to solve the system of coupled equations.

The paper is organized as follows: in section 2, the Proper Orthogonal Decomposition is presented, followed by an introduction to some basic notions about the geometry of the Grassmann and Stiefel manifolds to make the article reasonable self-contained. POD basis interpolation on Grassmannian manifolds is introduced considering the underlying

formulation of the logarithm and the exponential map. The core of this paper is illustrated in section 3, where the computational framework for the ROM adaptation based on a novel non-intrusive Space-Time POD basis interpolation on compact Stiefel manifolds is developed. The following section 4 covers the rigid visco-plastic formulation, the general framework of the thermal field equations, and the thermomechanical coupling. In section 5, the interpolation performance applied to a metal forming process is shown, as well as further computational aspects are discussed. Finally, section 6 highlights the main results and some important outcomes.

2 Space–Time POD, Grassmann and compact Stiefel manifolds

Let us recall here the important link between Proper Orthogonal Decomposition and Grassmann manifold [20,39,40,21,25].

Assume $\mathbf{S} \in \mathrm{Mat}_{n,m}(\mathbb{R})$ to be any real matrix of size $n \times m$ (with $n \geq m$), taken here to be a snapshot matrix with $n = 3N_S$ obtained from the spatial discretization N_s, and $m = N_t$ obtained from the time one. Any spatial POD of mode p leads to a p-dimensional vector space $\mathcal{V}_p \subset \mathbb{R}^m$ such that the Frobenius norm

$$\|\mathbf{S} - \boldsymbol{\Pi}_p \mathbf{S}\|_{\mathrm{F}}^2$$

is minimal, where matrix $\boldsymbol{\Pi}_p$ corresponds to the orthogonal projection on $\mathcal{V}_p$ (see [21] for more details). Such a matrix $\boldsymbol{\Pi}_p$ is directly obtained from a Singular Value Decomposition (SVD) of $\mathbf{S}$. Indeed, let us write a SVD

$$\mathbf{S} = \boldsymbol{\Phi} \boldsymbol{\Sigma} \boldsymbol{\Psi}^T$$

with $\boldsymbol{\Phi} = [\phi_1, \ldots, \phi_r]$ and $\boldsymbol{\Psi} = [\psi_1, \ldots, \psi_r]$, where the columns $\phi_k \in \mathbb{R}^n$ and $\psi_k \in \mathbb{R}^m$ form a set of orthonormal vectors, and $\boldsymbol{\Sigma} \in \mathrm{Mat}_{r,r}(\mathbb{R})$ is a diagonal matrix, where r denotes the rank of $\mathbf{S}$. Then, we can define $\boldsymbol{\Phi}_p := [\phi_1, \ldots, \phi_p] \in \mathrm{Mat}_{n,p}(\mathbb{R})$ and we obtain $\boldsymbol{\Pi}_p = \boldsymbol{\Phi}_p \boldsymbol{\Phi}_p^T$.

In this classical approach, the relevant object is not the reduced matrix $\mathbf{S}_p := \boldsymbol{\Pi}_p \mathbf{S}$, supposed to be of maximal rank, but the p-dimensional vector space $\mathcal{V}_p$ spanned by vectors $\phi_1, \ldots, \phi_p$, and thus the image of the matrix $\boldsymbol{\Phi}_p$. From this, interpolation has to be considered on the set of all p-dimensional vector spaces, that is on the so–called *Grassmann manifold* $\mathcal{G}(p, n)$:

$$\mathcal{G}(p, n) := \{\mathcal{V}_p \subset \mathbb{R}^n, \quad \dim(\mathcal{V}_p) = p\}.$$

Note here that the *point* $\mathbf{m} := \mathcal{V}_p \in \mathcal{G}(p, n)$ defines a vector space spanned by the set $\phi_1, \ldots, \phi_p$ *represented* by matrix $\boldsymbol{\Phi}_p$, however this matrix representation is not unique (see Example 1).

Take now a set $\{\lambda_1, \ldots, \lambda_N\}$ of parameter values leading to snapshot matrices $\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(N)}$ with SVD

$$\mathbf{S}^{(k)} = \boldsymbol{\Phi}^{(k)} \boldsymbol{\Sigma}^{(k)} \boldsymbol{\Psi}^{(k)}, \quad \boldsymbol{\Phi}^{(k)} = [\phi_1^{(k)}, \ldots, \phi_r^{(k)}], \quad \boldsymbol{\Psi}^{(k)} = [\psi_1^{(k)}, \ldots, \psi_r^{(k)}],$$

where $\phi_i^{(k)}$ are orthonormal vectors in $\mathbb{R}^n$ and $\psi_j^{(k)}$ are orthonormal vectors in $\mathbb{R}^m$.

The classical approach [20, 21] then considers the spatial POD of the snapshot matrices $\mathbf{S}_p^{(1)}, \ldots, \mathbf{S}_p^{(N)}$ of mode p, so that we obtain points $\mathbf{m}_i$ $(i = 1, \ldots, N)$ on $\mathcal{G}(p, n)$, respectively represented by the matrices

$$\boldsymbol{\Phi}_p^{(k)} := [\phi_1^{(k)}, \ldots, \phi_p^{(k)}] \in \mathrm{Mat}_{n,p}(\mathbb{R}), \quad \left(\boldsymbol{\Phi}_p^{(k)}\right)^T \boldsymbol{\Phi}_p^{(k)} = \mathbf{I}_p.$$

To any new parameter value $\widetilde{\lambda}$, it is possible to make an interpolation considering the spatial part based on the points $\mathbf{m}_i \in \mathcal{G}(p,n)$, using a local chart given by normal coordinates [20,21,25], in order to obtain a point $\widetilde{\mathbf{m}} \in \mathcal{G}(p,n)$ represented by a matrix $\widetilde{\boldsymbol{\Phi}}$. From such a point $\widetilde{\mathbf{m}} \in \mathcal{G}(p,n)$, we deduce a p-dimensional vector space on which some POD-Galerkin approach [21] can lead to a new ROM model.

On the contrary, we propose another approach as we consider a *Space–Time interpolation*, using both the spatial vector spaces represented by matrices $\boldsymbol{\Phi}_p^{(k)}$ and the temporal vector spaces represented by matrices

$$\boldsymbol{\Psi}_p^{(k)} := [\psi_1^{(k)}, \ldots, \psi_p^{(k)}] \in \mathrm{Mat}_{m,p}(\mathbb{R}), \quad \left(\boldsymbol{\Psi}_p^{(k)}\right)^T \boldsymbol{\Psi}_p^{(k)} = \mathbf{I}_p.$$

An important observation now is that matrices $\boldsymbol{\Phi}_p^{(k)}$ (resp. $\boldsymbol{\Psi}_p^{(k)}$) directly define an *ordered p-tuple of orthonormal vectors* in $\mathbb{R}^n$ (resp. $\mathbb{R}^m$), that is a point on the *compact Stiefel manifold*

$$\mathcal{S}t_c(p,n) := \{\text{Ordered orthonormal } p\text{-tuple of vectors in } \mathbb{R}^n\}.$$

To obtain a Space-Time POD interpolation (instead of a spatial POD interpolation followed by Galerkin approach), we finally adopted the following strategy, when dealing with a parameter value $\widetilde{\lambda}$:

1. Define a *curve* on the compact Stiefel manifold corresponding to the spatial part

$$\lambda \mapsto \boldsymbol{\Phi}(\lambda) \in \mathcal{S}t_c(p,n)$$

   obtained using the already known interpolation algorithm on Grassmann manifold.
2. In the same way, define a *curve* on the compact Stiefel manifold corresponding to the temporal part

$$\lambda \mapsto \boldsymbol{\Psi}(\lambda) \in \mathcal{S}t_c(p,m).$$

3. Construct an *interpolated curve* $\lambda \mapsto \mathbf{S}(\lambda)$ passing through the POD of mode p snapshot matrices $\mathbf{S}_p^{(k)}$, in order to obtain an interpolation of a ROM matrix $\widetilde{\mathbf{S}} := \mathbf{S}(\widetilde{\lambda})$.

In the next subsections, we give all important details to obtain such an interpolated curve $\lambda \mapsto \mathbf{S}(\lambda)$. First, in subsection 2.1 we explain how to compute on Grassmann manifolds using their Riemannian structure to obtain explicit formulae for the *geodesics* defining *normal coordinates*. From this explicit formulae, we can deduce in subsection 2.2 a *target algorithm* in order to define the curves

$$\lambda \mapsto \boldsymbol{\Phi}(\lambda) \in \mathcal{S}t_c(p,n), \quad \lambda \mapsto \boldsymbol{\Psi}(\lambda) \in \mathcal{S}t_c(p,m)$$

on compact Stiefel manifolds. The question on how to define an interpolated curve for matrices $\mathbf{S}_p^{(k)}$ will then be addressed in section 3.

2.1 Riemannian geometry on Grassmann manifolds

We will summarize now some essential results about Grassmann manifolds. Such manifolds are in fact *complete Riemannian manifolds* [24], meaning for instance that we can define the *length* of a curve. Moreover, we can always construct a curve of the shortest length between two points, which is called a *geodesic*, and it will be the starting point to define *normal coordinates* via the exponential and logarithm map (Definition 1 and 2). As we cannot do direct computations on Riemann manifolds, normal coordinates enable us to obtain formulae of curves, such as the Lagrangian polynomials. Note finally that a rigorous mathematical background of all of this is given in [41].

After we give a definition of the Grassmann manifold and how to *represent* its points with matrices, we propose to define the *tangent plane* using matrix representative, to have

formulae for a scalar product, given by (3). From this, we deduce a classical expression for geodesics (Theorem 1).

Let $p \leq n$ be two non-zero integers and $\mathcal{G}(p,n)$ the Grassmann manifold of p-dimensional subspaces in $\mathbb{R}^n$. In fact, Grassmann manifolds are special cases of *quotient manifolds*, meaning that a point on such a manifold can have many *representatives*. Let us consider indeed a p-dimensional linear subspace $\mathcal{V}$ of $\mathbb{R}^n$. Such a subspace can be defined using any ordered set of p independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_p$ in $\mathbb{R}^n$, encoded into a full rank matrix

$$\mathbf{M} := [\mathbf{v}_1, \ldots, \mathbf{v}_p] \in \mathrm{Mat}_{n,p}(\mathbb{R}).$$

Any other basis $\mathbf{v}'_1, \ldots, \mathbf{v}'_p$ of $\mathcal{V}$ will then lead to another full rank matrix

$$\mathbf{M}' := [\mathbf{v}'_1, \ldots, \mathbf{v}'_p] \in \mathrm{Mat}_{n,p}(\mathbb{R}),$$

and we necessary have

$$\mathbf{M}' = \mathbf{M}\mathrm{P}$$

where $\mathrm{P} \in \mathrm{GL}(p)$ is some invertible matrix in $\mathrm{Mat}_{p,p}(\mathbb{R})$. From all this, we deduce that the point $\mathbf{m} := \mathcal{V} \in \mathcal{G}(p,n)$ is represented by the infinite set of matrices

$$\{\mathbf{M}\mathrm{P}, \quad \mathrm{P} \in \mathrm{GL}(p)\}.$$

Now, the ordered set of p independent vectors in $\mathbb{R}^n$ and thus the set of full rank matrices in $\mathrm{Mat}_{n,p}(\mathbb{R})$ define the Stiefel manifold (see Figure 1)

$$\mathcal{S}t(p,n) := \{\mathbf{M} = [\mathbf{v}_1, \ldots, \mathbf{v}_p] \in \mathrm{Mat}_{n,p}(\mathbb{R}), \quad \mathrm{rg}(\mathbf{M}) = p\}$$

so that we obtain a natural map from such Stiefel manifold and the Grassmann manifold $\mathcal{G}(p,n)$ (see Figure 2):

$$\mathbf{M} = [\mathbf{v}_1, \ldots, \mathbf{v}_p] \in \mathcal{S}t(p,n) \mapsto \mathbf{m} = \{\mathbf{M}\mathrm{P}, \quad \mathrm{P} \in \mathrm{GL}(p)\}.$$

In our situation, nevertheless, we will only focus on *orthonormal bases* of p-dimensional subspaces. Doing so, we thus consider matrices defined by orthonormal vectors, leading to the so-called *compact Stiefel manifold*

$$\mathcal{S}t_c(p,n) := \left\{ \mathbf{Y} \in \mathrm{Mat}_{n,p}(\mathbb{R}), \quad \mathbf{Y}^T\mathbf{Y} = \mathbf{I}_p \right\} \tag{1}$$

and any point $\mathbf{m} \in \mathcal{G}(p,n)$ will then be represented by the infinite set

$$\{\mathbf{Y}\mathrm{Q}, \quad \mathrm{Q} \in \mathrm{O}(p)\}$$

where $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_p]$ is defined using an orthonormal basis $\mathbf{y}_1, \ldots, \mathbf{y}_p$ of $\mathbf{m}$. This defines a *surjective* map

$$\pi \ : \ \mathbf{Y} \in \mathcal{S}t_c(p,n) \mapsto \mathbf{m} = \pi(\mathbf{Y}) = \{\mathbf{Y}\mathrm{Q}, \quad \mathrm{Q} \in \mathrm{O}(p)\} \in \mathcal{G}(p,n)$$

and the set of all matrices representing the same point $\mathbf{m} \in \mathcal{G}(p,n)$ is called the *fiber* of π at $\mathbf{m}$ (see Figure 3 for an illustration of a fiber):

$$\pi^{-1}(\mathbf{m}) = \{\mathbf{Y}Q, \quad Q \in \mathrm{O}(p)\}\,.$$

Remark 1 An important point here is that, from now on, any computation on $\mathcal{G}(p,n)$ will be done using a *choice in the fibers*. Nevertheless, for any point $\mathbf{m} \in \mathcal{G}(p,n)$, *there is no canonical* way to choose an element $\mathbf{Y} \in \pi^{-1}(\mathbf{m})$, so any computation has to be independent of that choice.

We need now to define the *geodesics* of Grassmann manifold, which can be done once we have defined the *tangent plane* at each point $\mathbf{m} \in \mathcal{G}(p,n)$ and a Riemaniann metric. Take any point $\mathbf{m} \in \mathcal{G}(p,n)$ represented by a matrix $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_p]$ of orthonormal vectors,

the tangent plane $\mathcal{T}_{\mathbf{m}} := \mathcal{T}_{\mathbf{m}}\mathcal{G}(p,n)$ is then represented by the $p(n-p)$ dimensional vector space

$$\mathrm{Hor}_{\mathbf{Y}} := \left\{ \mathbf{Z} \in \mathrm{Mat}_{n,p}(\mathbb{R}), \quad \mathbf{Z}^T\mathbf{Y} = 0 \right\}, \tag{2}$$

called the *horizontal space*, where $\mathbf{Y} \in \pi^{-1}(\mathbf{m})$. From all this, a vector $v \in \mathcal{T}_{\mathbf{m}}$ will be called a *velocity vector*, which can be *represented* by a matrix $\mathbf{Z} \in \mathrm{Mat}_{n,p}(\mathbb{R})$ such that $\mathbf{Z}^T\mathbf{Y} = 0$, and $\mathbf{Z}$ is called a *horizontal lift* of v.

Example 1 Take here $p = 2$ and $n = 5$, so that $\mathcal{G}(2,5)$ is the set of planes in a five dimensional space. The matrices

$$\mathbf{Y} = \begin{bmatrix} \frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{\sqrt{2}}{2} \\ 0 & 0 \\ \frac{1}{2} & \frac{\sqrt{2}}{2} \\ \frac{1}{2} & 0 \end{bmatrix}, \quad \mathbf{Y}' = \begin{bmatrix} \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} \\ \frac{2-\sqrt{2}}{4} & \frac{2+\sqrt{2}}{4} \\ 0 & 0 \\ \frac{2+\sqrt{2}}{4} & \frac{2-\sqrt{2}}{4} \\ \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} \end{bmatrix}$$

are in the compact Stiefel manifold $\mathcal{S}t_c(2,5)$, representing the same plane $\mathbf{m} \in \mathcal{G}(2,5)$. The horizontal space $\mathrm{Hor}_{\mathbf{Y}}$ defined by (2) is a 6-dimensional vector space of matrices $\mathbf{Z}$, for instance given by

$$\mathbf{Z} = \begin{bmatrix} u_1 & v_1 \\ u_2 & v_2 \\ u_3 & v_3 \\ -u_2 & -v_2 \\ -u_1 + u_2 - u_4 & -v_1 + v_2 - v_4 \end{bmatrix}, \quad u_i, v_i \in \mathbb{R}.$$

Taking now velocity vectors $v_1, v_2 \in \mathcal{T}_{\mathbf{m}}$ with respective horizontal lifts $\mathbf{Z}_1, \mathbf{Z}_2 \in \mathrm{Hor}_{\mathbf{Y}}$ we define the point–wise scalar product [42,40]:

$$\langle v_1, v_2 \rangle_{\mathbf{m}} := \mathrm{tr}\left(\mathbf{Z}_1^T \mathbf{Z}_2\right). \tag{3}$$

Such a Riemannian metric leads to explicit geodesics given by [40,39]:

Theorem 1 *Let $\mathbf{m} \in \mathcal{G}(p,n)$ represented by $\mathbf{Y} \in \mathcal{S}t_c(p,n)$. For any $v \in \mathcal{T}_{\mathbf{m}}$ with horizontal lift given by $\mathbf{Z}$ in $\mathrm{Hor}_{\mathbf{Y}}$, let $\mathbf{Z} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ be a thin SVD of $\mathbf{Z}$. Then*

$$\alpha_v : \ t \in \mathbb{R} \mapsto \alpha_v(t) := \pi\left[\left(\mathbf{Y}\mathbf{V}\cos(t\boldsymbol{\Sigma}) + \mathbf{U}\sin(t\boldsymbol{\Sigma})\right)\mathbf{V}^T\right] \in \mathcal{G}(p,n) \tag{4}$$

is the unique maximal geodesic such that $\alpha_v(0) = \mathbf{m}$ and initial velocity

$$\dot{\alpha_v}(0) := \frac{\partial \alpha_v(t)}{\partial t}\,|_{t=0} = v.$$

Remark 2 Up to our knowledge, there is no proof that

$$\mathbf{Y}(t) := \left(\mathbf{Y}\mathbf{V}\cos(t\boldsymbol{\Sigma}) + \mathbf{U}\sin(t\boldsymbol{\Sigma})\right)\mathbf{V}^T \in \mathcal{S}t_c(p,n). \tag{5}$$

In fact, this follows by direct computation. Indeed, $\mathbf{Z} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ being a thin SVD, we have $\mathbf{V} \in \mathrm{O}(p)$ and

$$\mathbf{Z}^T\mathbf{Y} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{Y} = 0 \implies \boldsymbol{\Sigma}\mathbf{U}^T\mathbf{Y} = 0$$

so that

$$\sin(t\boldsymbol{\Sigma})\mathbf{U}^T\mathbf{Y} = 0 \text{ and } \mathbf{Y}^T\mathbf{U}\sin(t\boldsymbol{\Sigma}) = 0.$$

Finally, we have:

$$\mathbf{Y}^T(t)\mathbf{Y}(t) = \mathbf{V}\left(\cos^2(t\boldsymbol{\Sigma}) + \underbrace{\sin(t\boldsymbol{\Sigma})\mathbf{U}^T\mathbf{Y}}_{=0}\mathbf{V}\cos(t\boldsymbol{\Sigma}) + \cos(t\boldsymbol{\Sigma})\underbrace{\mathbf{V}^T\mathbf{Y}^T\mathbf{U}\sin(t\boldsymbol{\Sigma})}_{=0} + \sin^2(t\boldsymbol{\Sigma})\right)\mathbf{V}^T$$

which concludes the proof.

Remark 3 In many cases, formulas of the geodesic do not use the right multiplication by $\mathbf{V}^T$, as for instance in [40,21]. Of course, as $\mathbf{V}$ being in $\mathrm{O}(p)$ both matrices

$$(\mathbf{Y}\mathbf{V}\cos(t\boldsymbol{\Sigma}) + \mathbf{U}\sin(t\boldsymbol{\Sigma}))\,\mathbf{V}^T \text{ and } \mathbf{Y}\mathbf{V}\cos(t\boldsymbol{\Sigma}) + \mathbf{U}\sin(t\boldsymbol{\Sigma})$$

define the same point on $\mathcal{G}(p,n)$. Now, the choice of such right multiplication in (5) is related to the choice of the horizontal lift $\mathbf{Z} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$. Indeed, taking back the path given by (5), we have

$$\dot{\mathbf{Y}}(t) = (-\mathbf{Y}\mathbf{V}\boldsymbol{\Sigma}\sin(t\boldsymbol{\Sigma}) + \mathbf{U}\boldsymbol{\Sigma}\cos(t\boldsymbol{\Sigma}))\,\mathbf{V}^T \implies \dot{\mathbf{Y}}(0) = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \mathbf{Z}$$

which corresponds to the choice of the horizontal lift for velocity vector $v \in \mathcal{T}_{\mathbf{m}}$.

A consequence of Theorem 1 is an explicit formula for the exponential map [40,21] (see Figure 4):

Definition 1 *Let $\mathbf{m} \in \mathcal{G}(p,n)$ be represented by $\mathbf{Y} \in \mathcal{S}t_c(p,n)$. For any velocity vector $v \in \mathcal{T}_{\mathbf{m}}$ with horizontal lift $\mathbf{Z} \in \mathrm{Hor}_{\mathbf{Y}}$, take $\mathbf{Z} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ to be a thin SVD of $\mathbf{Z}$. Then we define the exponential map*

$$\mathrm{Exp}_{\mathbf{m}} : \mathcal{T}_{\mathbf{m}} \longrightarrow \mathcal{G}(p,n),$$
$$v \mapsto \mathrm{Exp}_{\mathbf{m}}(v) := \pi\left[(\mathbf{Y}\mathbf{V}\cos(\boldsymbol{\Sigma}) + \mathbf{U}\sin(\boldsymbol{\Sigma}))\,\mathbf{V}^T\right] = \alpha_v(1).$$

Now, it is possible to define directly some inverse map of the exponential map, called the logarithm map [40], but only *locally*. For any $\mathbf{m}$ and $\mathbf{Y}$ in its fiber, let us first define the open space

$$\mathrm{U}_{\mathbf{m}} := \{\mathbf{m}_1 \in \mathcal{G}(p,n), \quad \mathbf{Y}^T\mathbf{Y}_1 \text{ is invertible}, \quad \mathbf{Y}_1 \in \pi^{-1}(\mathbf{m}_1)\}. \tag{6}$$

Then we have:

Definition 2 (Logarithm map on Grassmannian manifold) *Let* $\mathbf{m} \in \mathcal{G}(p,n)$ *be represented by a matrix* $\mathbf{Y} \in \mathcal{S}t_c(p,n)$. *For any point* $\mathbf{m}_1$ *in the open space* $\mathrm{U}_{\mathbf{m}}$ *represented by a matrix* $\mathbf{Y}_1 \in \mathcal{S}t_c(p,n)$, *define a thin SVD*

$$\mathbf{Y}_1 \left(\mathbf{Y}^T\mathbf{Y}_1\right)^{-1} - \mathbf{Y} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T.$$

Then the logarithm $\mathrm{Log}_{\mathbf{m}}(\mathbf{m}_1) \in \mathcal{T}_{\mathbf{m}}$ *is the velocity vector in* $\mathcal{T}_{\mathbf{m}}$ *with horizontal lift*

$$\mathbf{Z} = \mathbf{U}\arctan(\boldsymbol{\Sigma})\mathbf{V}^T \in \mathrm{Hor}_{\mathbf{Y}}.$$

Remark 4 The logarithm map is only defined on some *open set* $\mathrm{U}_{\mathbf{m}}$. This means that for any point $\mathbf{m}_1 \notin \mathrm{U}_{\mathbf{m}}$, the associated matrix $\mathbf{Y}^T\mathbf{Y}_1$ is not invertible, so that the computation of

$$\mathbf{Y}_1 \left(\mathbf{Y}^T\mathbf{Y}_1\right)^{-1} - \mathbf{Y}$$

can not be done. Note finally that such an open set is strongly related to the *cut-locus* of a Grassmann manifold [43].

2.2 Target Algorithm on compact Stiefel manifolds

All the mathematical background summarized in subsection 2.1 can be used to obtain an interpolation curve between points $\mathbf{m}_1, \ldots, \mathbf{m}_N$ on Grassmann manifold $\mathcal{G}(p,n)$ [20,21], where each point $\mathbf{m}_i$ corresponds to a parameter value λ_i. Indeed, once a *reference point* $\mathbf{m}_{i_0} \in \{\mathbf{m}_1, \ldots, \mathbf{m}_N\}$ is chosen (see Figure 4):

- We use the logarithm map $\mathrm{Log}_{\mathbf{m}_{i_0}}$ to *linearize*, i.e., meaning we define velocity vectors $v_i := \mathrm{Log}_{\mathbf{m}_{i_0}}(\mathbf{m}_i)$ on the *vector space* $\mathcal{T}_{\mathbf{m}_{i_0}}$.

- We obtain an interpolation curve $\lambda \mapsto v(\lambda)$ between vectors v_i, using for instance Lagrangian polynomial, and thus

$$v(\lambda_i) = v_i, \quad \forall i = 1, \ldots, N.$$

- Taking the exponential map $\mathrm{Exp}_{\mathbf{m}_{i_0}}$, we obtain back an interpolation curve

$$\lambda \mapsto \mathbf{m}(\lambda) := \mathrm{Exp}_{\mathbf{m}_{i_0}}(v(\lambda))$$

between the points $\mathbf{m}_1, \ldots, \mathbf{m}_N$ on $\mathcal{G}(p, n)$, so that

$$\mathbf{m}(\lambda_i) = \mathbf{m}_i, \quad \forall i = 1, \ldots, N.$$

We propose here to define curves on the compact Stiefel manifold $\mathcal{S}t(p, n)$ instead of the ones defined on the Grassmann manifold $\mathcal{G}(p, n)$. The starting point is a set of matrices $\mathbf{Y}_1, \ldots, \mathbf{Y}_N$ in the compact Stiefel manifold $\mathcal{S}t(p, n)$, corresponding to parameter values $\lambda_1, \ldots, \lambda_N$. Once a reference parameter value λ_{i_0} has been chosen, we obtain a curve

$$\lambda \mapsto \mathbf{Y}(\lambda)$$

where in general,

$$\mathbf{Y}(\lambda_i) \neq \mathbf{Y}_i.$$

As a consequence, such a curve will not be an interpolation curve between the matrices $\mathbf{Y}_1, \ldots, \mathbf{Y}_N$ (see Remark 5). Before doing so, and to obtain well-defined curves, we need to make a specific definition:

Definition 3 (Genericity) *A matrix is said to be* generic *if all its non–zero singular values are distinct. The set of generic matrices in* $\mathrm{Mat}_{n,p}(\mathbb{R})$ *is denoted* $\mathrm{Mat}^0_{n,p}(\mathbb{R})$.

For any generic matrix $\mathbf{M} \in \mathrm{Mat}^0_{n,p}(\mathbb{R})$, we know that its thin SVD $\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ is well defined. Indeed, taking $\sigma_1 > \ldots > \sigma_p$ to be its ordered singular values, we can write

$$\mathbf{M} = \sum_{i=1}^{p} \sigma_i \mathbf{u}^i \mathbf{v}_i^T \tag{7}$$

where $\mathbf{u}_i$ (resp. $\mathbf{v}_i$) is a left singular vector associated to σ_i (resp. a right singular vector). All singular values being distinct, the only other possibility is to consider singular vectors $\epsilon_i \mathbf{u}_i$ and $\epsilon_i \mathbf{v}_i$, with $\epsilon_i = \pm 1$, so that the decomposition (7) remains the same. We thus deduce that the target Algorithm below is well defined:

Algorithm 1 (Target algorithm)

- **Inputs**:
  - Matrices $\mathbf{Y}_1, \ldots, \mathbf{Y}_N$ in $\mathcal{S}t_c(p,n)$, corresponding to parameter values $\lambda_1 < \ldots < \lambda_N$.
  - A reference parameter value λ_{i_0} with $i_0 \in \{1, \ldots, N\}$.
  - A parameter value λ.
- **Output**: A matrix $\mathbf{Y}(\lambda) \in \mathcal{S}t_c(p,n)$.

1. Define $\mathbf{Z}_{i_0} := \mathbf{0}$ and for each $k \in \{1, \ldots, N\}$ with $k \neq i_0$ compute a thin SVD of the generic matrix

$$\mathbf{Y}_k \left(\mathbf{Y}_{i_0}^T \mathbf{Y}_k\right)^{-1} - \mathbf{Y}_{i_0} = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^T$$

and define

$$\mathbf{Z}_k := \mathbf{U}_k \arctan(\boldsymbol{\Sigma}_k) \mathbf{V}_k^T, \quad \text{with assumption } \mathbf{Z}_k \in \mathrm{Mat}^0_{n,p}(\mathbb{R}).$$

2. Define an $n \times p$ matrix and compute a thin SVD

$$\mathbf{Z}(\lambda) := \sum_{i=1}^{N} \prod_{i \neq j} \frac{\lambda - \lambda_j}{\lambda_i - \lambda_j} \mathbf{Z}_i = \mathbf{U}(\lambda) \boldsymbol{\Sigma}(\lambda) \mathbf{V}(\lambda)^T,$$

with assumption $\mathbf{Z}(\lambda) \in \mathrm{Mat}^0_{n,p}(\mathbb{R})$.

3. Define the $n \times p$ matrix in $\mathcal{S}t_c(p,n)$ (see Remark 3):

$$\mathbf{Y}(\lambda) := [\mathbf{Y}_{i_0}\mathbf{V}(\lambda)\cos(\boldsymbol{\Sigma}(\lambda)) + \mathbf{U}(\lambda)\sin(\boldsymbol{\Sigma}(\lambda))]\mathbf{V}(\lambda)^T. \tag{8}$$

Note: cos and sin act only on diagonal entries.

In this algorithm, as already noticed and following the assumptions of genericity, the matrices $\mathbf{Z}_k$, $\mathbf{Z}(\lambda)$ and $\mathbf{Y}(\lambda)$ do not depend on the choice of matrices in the associated thin SVD.

Remark 5 Using this target Algorithm to parameter value $\lambda := \lambda_k$ leads to some matrix $\mathbf{Y}(\lambda_k)$ generally different from $\mathbf{Y}_k$ (except for $k = i_0$). Thus, such an algorithm computed on compact Stiefel manifold do not produce an *interpolation* on the points $\mathbf{Y}_1, \ldots, \mathbf{Y}_N$ (see Figure 3). Indeed, to represent an interpolation curve between these points means that if we consider the parameter value $\lambda = \lambda_k$ (with $k \in \{1, \ldots, N\}$) as input in the algorithm, one should expect to return as output $\mathbf{Y}(\lambda_k)$ (given by (8)) the initial matrix $\mathbf{Y}_k$, which is not the case in general.

Nevertheless, matrices $\mathbf{Y}(\lambda_k)$ and $\mathbf{Y}_k$ define the same point on the Grassmann manifold $\mathcal{G}(p,n)$, meaning that they both define an *orthonormal basis of the same subspace* $\mathbf{m}_k$ (see Remark 4). As a consequence, a projection matrix onto the subspace $\mathbf{m}_k$ is given by $\mathbf{Y}(\lambda_k)^T\mathbf{Y}(\lambda_k)$ or equivalently by $\mathbf{Y}_k^T\mathbf{Y}_k$.

Example 2 Take for instance the compact Stiefel manifold $\mathcal{S}t_c(2,5)$, and the three matrices

$$\mathbf{Y}_1 := \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{Y}_2 := \begin{bmatrix} \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ 0 & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} & 0 \\ -\frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ 0 & 0 \end{bmatrix}, \quad \mathbf{Y}_3 := \begin{bmatrix} \frac{\sqrt{3}}{3} & -\frac{\sqrt{6}}{6} \\ 0 & \frac{\sqrt{6}}{4} \\ \frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{12} \\ 0 & \frac{\sqrt{6}}{4} \\ \frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{12} \end{bmatrix}$$

which correspond respectively to $\lambda_1 = 15$, $\lambda_2 = 22$ and $\lambda_3 = 27$. Choosing the reference parameter value to be λ_1 and following the target Algorithm 1 we obtain

$$\mathbf{Y}_2(\mathbf{Y}_1^T\mathbf{Y}_2)^{-1} - \mathbf{Y}_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & -1 \\ -1 & 2 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{Y}_3(\mathbf{Y}_1^T\mathbf{Y}_3)^{-1} - \mathbf{Y}_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}$$

Taking $\lambda = \lambda_2$ and $\lambda = \lambda_3$ as inputs in the algorithm, we finally obtain the matrices (with computation done using 5 digits):

$$\mathbf{Y}(\lambda_2) = \begin{bmatrix} 0.77460 & 0.25820 \\ 0.25820 & 0.51640 \\ 0.51640 & -0.25820 \\ -0.25820 & 0.77460 \\ 0 & 0 \end{bmatrix} \neq \mathbf{Y}_2, \quad \mathbf{Y}(\lambda_3) = \begin{bmatrix} 0.67860 & -0.19876 \\ -0.19876 & 0.57922 \\ 0.47984 & 0.38046 \\ -0.19876 & 0.57922 \\ 0.47984 & 0.38046 \end{bmatrix} \neq \mathbf{Y}_3.$$

3 Space-Time Interpolation on compact Stiefel manifolds

As already noticed, POD is extracting the optimal space structures and the associated temporal modes. An important property is that the spatial and temporal orthogonal modes

are *coupled*: each space component is associated with a temporal component partner and there is a one-to-one correspondence between both spaces. Taking advance of this decomposition into orthogonal modes, it is natural to try a *Space-Time* interpolation on compact Stiefel manifolds based on the target Algorithm 1, instead of an interpolation of the space part alone, followed by a Galerkin approach as is classically done [20,21].

As a starting point, take a set of *snapshot matrices* $\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(N)}$, where each matrix $\mathbf{S}^{(k)} \in \mathrm{Mat}_{n,m}(\mathbb{R})$ corresponds to a given parameter value $\lambda_k \in \mathbb{R}$, with $\lambda_1 < \ldots < \lambda_N$ and $n = 3N_s$ corresponding to the spatial part, while $m = N_t$ corresponds to the temporal part. For a given mode $p \leq N_t$, our goal is to

1. Extract in a unique way a POD of mode p of each matrix $\mathbf{S}^{(k)}$, so that we have a well defined map
$$\mathbf{S}^{(k)} \in \mathrm{Mat}_{n,m}(\mathbb{R}) \mapsto \mathbf{S}_p^{(k)} \in \mathrm{Mat}_{n,m}(\mathbb{R}).$$
2. Obtain for each $\mathbf{S}_p^{(k)} \in \mathrm{Mat}_{n,m}(\mathbb{R})$ a unique matrix $\boldsymbol{\Phi}_p^{(k)} \in \mathcal{S}t_c(p,n)$ for the spatial part and another unique matrix $\boldsymbol{\Psi}_p^{(k)} \in \mathcal{S}t_c(p,m)$ for the temporal part.
3. Use the target Algorithm 1 on matrices $\boldsymbol{\Phi}_p^{(k)}$ first, and then on matrices $\boldsymbol{\Psi}_p^{(k)}$, in order to obtain two curves
$$\lambda \mapsto \boldsymbol{\Phi}(\lambda), \quad \lambda \mapsto \boldsymbol{\Psi}(\lambda) \tag{9}$$
which are not interpolated curves, as in general $\boldsymbol{\Phi}(\lambda_k) \neq \boldsymbol{\Phi}_p^{(k)}$ and $\boldsymbol{\Psi}(\lambda_k) \neq \boldsymbol{\Psi}_p^{(k)}$ (see Remark 5).
4. Define an interpolation curve $\lambda \mapsto \mathbf{S}(\lambda)$ between matrices $\mathbf{S}_p^{(1)}, \ldots, \mathbf{S}_p^{(N)}$, using curves obtained by (9).

We now detail two key points: the first concerns a new type of SVD, called oriented SVD, which allows defining the matrices $\boldsymbol{\Phi}_p^{(k)}$ and $\boldsymbol{\Psi}_p^{(k)}$ in a unique way. Finally, we will explain

in subsection 3.2 how to construct the curve $\lambda \mapsto \mathbf{S}(\lambda)$, which requires the introduction of a mixed part.

3.1 Oriented SVD on generic matrices

As already noticed in section 2, any computation of a POD of mode p of a matrix $\mathbf{S} \in \mathrm{Mat}_{n,m}(\mathbb{R})$ can be obtained from a SVD. Suppose now that $\mathbf{S}$ is of rank $r \geq p$. Any SVD of $\mathbf{S}$ with singular values $\sigma_1 > \ldots > \sigma_r$ leads to *spatial orthonormal vectors* $\phi_1, \ldots, \phi_r$ in $\mathbb{R}^n$ (the left singular vectors) and *temporal orthonormal vectors* $\psi_1, \ldots, \psi_r$ in $\mathbb{R}^m$ (the right singular vectors). A POD of mode p then writes

$$\mathbf{S}_p = \boldsymbol{\Phi}_p \boldsymbol{\Sigma}_p \boldsymbol{\Psi}_p^T, \quad \boldsymbol{\Phi}_p := [\phi_1, \ldots, \phi_p], \quad \boldsymbol{\Sigma}_p := \mathrm{diag}(\sigma_1, \ldots, \sigma_p), \quad \boldsymbol{\Psi}_p := [\psi_1, \ldots, \psi_p]. \tag{10}$$

Now, because of sign indeterminacy of the spatial vectors ϕ_i and temporal vectors ψ_i, the matrices $\boldsymbol{\Phi}_p, \boldsymbol{\Psi}_p$ are not uniquely defined.

To overcome this difficulty, we need to introduce a *new SVD* so that, under the assumption of genericity (see Definition 3), the matrices $\boldsymbol{\Phi}_p$ and $\boldsymbol{\Psi}_p$ given by (10) can be well-defined.

The main idea of the new SVD introduced here is to make an intrinsic choice on the orientation for each space and temporal vector. Indeed, for each spatial vector ϕ, only two choices can occur: ϕ or $-\phi$ (thus inducing a choice on the associated temporal vector). A choice of orientation is then made as follows. Taking the column vectors $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_m]$ and $\mathbf{s}$ to be the first column vector such that the scalar product $\langle \mathbf{s}, \phi \rangle$ is non zero, we impose the sign taking $\langle \mathbf{s}, \phi \rangle > 0$.

Let us now give all details to compute the oriented SVD before obtaining algorithm 2. A first Lemma, obtained by direct computation, allows us to use a column vector of the initial snapshot matrix $\mathbf{S}$ to choose orientation:

Lemma 1 *Let us consider* $\mathbf{s}_1, \dots, \mathbf{s}_m \in \mathbb{R}^n$ *to be the column vectors of* $\mathbf{S} \in \mathrm{Mat}_{n,m}(\mathbb{R})$ *and take* $\phi \in \mathbb{R}^n$ *to be a unit spatial vector of* $\mathbf{S}$*, associated with a non–zero singular value* σ*. Then, there exists* $i \in \{1, \dots, m\}$ *such that* $\langle \mathbf{s}_i, \phi \rangle = \mathbf{s}_i^T \phi \neq 0$.

From this, for any unit spatial vector $\phi \in \mathbb{R}^n$ of $\mathbf{S}$, let us define $\mathbf{s}(\phi)$ to be the first column vector $\mathbf{s}_i$ in $\mathbf{S} = [\mathbf{s}_1, \dots, \mathbf{s}_n]$ such that $\langle \phi, \mathbf{s}_i \rangle \neq 0$:

$$\mathbf{s}(\phi) := \mathbf{s}_i, \quad i := \min \{ j, \quad \langle \mathbf{s}_j, \phi \rangle \neq 0 \} . \tag{11}$$

Any spatial eigenvector can therefore have a specific orientation:

Definition 4 (Oriented eigenvectors) *Let* $\mathbf{S} \in \mathrm{Mat}_{n,m}(\mathbb{R})$ *and* $\phi \in \mathbb{R}^n$ *a unit spatial vector associated to a non–zero singular value* σ*. Then* ϕ *is said to be* oriented *if* $\langle \mathbf{s}(\phi), \phi \rangle > 0$.

From all this, let us now deduce the new SVD:

Lemma 2 (Oriented SVD) *Let* $\mathbf{S} \in \mathrm{Mat}^0_{n,m}(\mathbb{R})$ *(*$m \leq n$*) of rank* r *such that all its non-zero singular values are distinct. Then, there exists one and only one couple of matrices*

$$\boldsymbol{\Phi} = [\phi_1, \dots, \phi_r] \in \mathrm{Mat}_{n,r}(\mathbb{R}), \quad \boldsymbol{\Psi} = [\psi_1, \dots, \psi_r] \in \mathrm{Mat}_{m,r}(\mathbb{R}) \tag{12}$$

such that

$$\langle \phi_i, \phi_j \rangle = \langle \psi_i, \psi_j \rangle = \delta_{ij}, \quad \mathbf{S} = \boldsymbol{\Phi} \boldsymbol{\Sigma} \boldsymbol{\Psi}^T, \quad \boldsymbol{\Sigma} := Diag(\sigma_1, \dots, \sigma_r) \in \mathrm{Mat}_{r,r}(\mathbb{R}) \tag{13}$$

and ϕ_i are oriented spatial unit eigenvectors:

$$\langle \mathbf{s}(\phi_i), \phi_i \rangle > 0 \tag{14}$$

with $\mathbf{s}(\phi_i)$ defined by (11). Such a decomposition is called an oriented SVD.

Proof First, any couple (ϕ, ψ) of spatial–temporal unit eigenvector for $\mathbf{S}$ is defined modulo ± 1, and ψ is obtained in a unique way from ϕ.

Let us suppose now we do not have uniqueness, so that there exist two unit spatial vectors ϕ and ϕ' associated to σ such that

$$\langle \mathbf{s}(\phi), \phi \rangle > 0 \text{ and } \langle \mathbf{s}(\phi'), \phi' \rangle > 0.$$

We necessary have $\phi' = -\phi$ and $\mathbf{s}(\phi) = \mathbf{s}(\phi')$ so we deduce that

$$\langle \mathbf{s}(\phi'), \phi' \rangle >= -\langle \mathbf{s}(\phi), \phi \rangle > 0$$

which is a contradiction, and we can conclude our proof.

We give now an algorithm to obtain such an oriented SVD:

Algorithm 2 (Oriented SVD)

- **Inputs**: $m \leq n$ and $\mathbf{S} \in \mathrm{Mat}^0_{n,m}(\mathbb{R})$ of rank r.
- **Output**: Unique matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ for an oriented SVD of $\mathbf{S}$.

1. Compute a SVD of $\mathbf{S}$ so that to obtain spatial unit vectors $\phi_1, \ldots, \phi_r$ and temporal unit vectors $\psi_1, \ldots, \psi_r$.
2. Consider the column vectors $\mathbf{s}_1, \ldots, \mathbf{s}_m$ of $\mathbf{S}$.

3. For $i = 1, \ldots, r$ define

$$\varepsilon_i := \frac{\langle \mathbf{s}(\phi_i), \phi_i \rangle}{|\langle \mathbf{s}(\phi_i), \phi_i \rangle|}$$

where $\mathbf{s}(\phi_i)$ is the first column vector $\mathbf{s}$ of $\mathbf{S}$ such that $\langle \phi_i, \mathbf{s} \rangle \neq 0$, see (11).

4. For $i = 1, \ldots, r$, make sign replacement

$$\phi_i \leftarrow \varepsilon_i \phi_i, \quad \psi_i \leftarrow \varepsilon_i \psi_i.$$

Example 3 Assume the rank 3 matrix

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \\ 0 & 2 & -1 \\ 0 & -1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = [\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3]$$

where a unit spatial vector corresponding to the largest singular value is given by (with 5 digits)

$$\phi_1 = \begin{bmatrix} -0.31145 \\ 0.41763 \\ 0.74265 \\ -0.28294 \\ -0.31145 \end{bmatrix}$$

and we can check that $\mathbf{s}(\phi_1) = \mathbf{s}_1$ with $\langle \phi_1, \mathbf{s}_1 \rangle < 0$ so that we consider $-\phi_1$ instead of ϕ_1, and so on.

3.2 Space–Time interpolation algorithm

In this subsection, we define a Space–Time interpolation on any family of POD of mode p taken from *generic* snapshot matrices (see an overview in Figure 5). Such interpolation captures both the spatial and temporal part of such matrices, which is necessary from the point of view of mechanical equations, but we will also need to define a specific *mixed part* of each POD (see lemma 3). The method is formalized for the general case of interest in mechanical applications of an arbitrary number of pre-computed POD basis N, and an arbitrary number of parameters N_p per training point. In the following, the method is described for an arbitrary number of basis N and a single parameter $N_p = 1$ by applying a univariate Lagrange interpolation. However, its generalization in a multivariate case is straightforward using an appropriate multivariate interpolation scheme [**?**].

Take back parameter values $\lambda_1 < \ldots < \lambda_N$, corresponding to snapshot matrices $\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(N)}$ in $\mathrm{Mat}_{n,m}(\mathbb{R})$, with $n = 3N_s$ and $m = N_t$. To make use of the oriented SVD, let us suppose:

Genericity assumption: All snapshot matrices $\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(N)}$ have distinct non–zero singular values.

Take now p to be some integer (less or equal than the minimum rank of all matrices $\mathbf{S}^{(k)}$). Using the oriented SVD given by Algorithm 2, we can consider a POD of mode p on each matrix $\mathbf{S}^{(k)}$:

$$\mathbf{S}_p^{(k)} := \boldsymbol{\Phi}_p^{(k)} \boldsymbol{\Sigma}_p^{(k)} {\boldsymbol{\Psi}_p^{(k)}}^T \in \mathrm{Mat}_{n,m}(\mathbb{R}) \tag{15}$$

where $\boldsymbol{\Sigma}_k$ corresponds to singular values, and $\boldsymbol{\Phi}_p^{(k)}$ as well as $\boldsymbol{\Psi}_p^{(k)}$ uniquely define points in a compact Stiefel manifold:

$$\boldsymbol{\Phi}_p^{(k)} := [\phi_1^{(k)}, \ldots, \phi_p^{(k)}] \in \mathcal{S}t_c(p,n), \quad \boldsymbol{\Psi}_p^{(k)} := [\psi_1^{(k)}, \ldots, \psi_p^{(k)}] \in \mathcal{S}t_c(p,m), \tag{16}$$

Recall that in previous equation, $\phi_1^{(k)}, \ldots, \phi_p^{(k)}$ (resp. $\psi_1^{(k)}, \ldots, \psi_p^{(k)}$) correspond to spatial oriented eigenvectors (resp. temporal ones) of $\mathbf{S}^{(k)}$.

Using the target Algorithm 1 first for the spatial matrices $\boldsymbol{\Phi}_p^{(k)}$ and then for the temporal matrices $\boldsymbol{\Psi}_p^{(k)}$, we obtain two curves

$$\lambda \mapsto \boldsymbol{\Phi}(\lambda), \quad \lambda \mapsto \boldsymbol{\Psi}(\lambda).$$

Now, our goal is to produce an interpolation curve between the matrices $\mathbf{S}_p^{(k)}$, taking into account both spatial and temporal curves defined above. Such a curve is given by

$$\lambda \mapsto \mathbf{S}(\lambda) := \boldsymbol{\Phi}(\lambda)\mathbf{M}(\lambda)\boldsymbol{\Psi}(\lambda)^T \text{ with } \mathbf{S}(\lambda_k) = \mathbf{S}_p^{(k)}. \tag{17}$$

Lemma 3 *For a curve defined* (17) *to be an interpolation curve between the matrices* $\mathbf{S}_p^{(k)}$, *we necessary have*

$$\mathbf{M}(\lambda_k) = \boldsymbol{\Phi}(\lambda_k)^T \mathbf{S}_p^{(k)} \boldsymbol{\Psi}(\lambda_k). \tag{18}$$

Proof To satisfy (17), we necessary have

$$\mathbf{S}(\lambda_k) = \boldsymbol{\Phi}(\lambda_k)\mathbf{M}(\lambda_k)\boldsymbol{\Psi}(\lambda_k)^T = \mathbf{S}_p^{(k)}, \tag{19}$$

where, by construction, we have $\boldsymbol{\Phi}(\lambda_k) \in \mathcal{S}t_c(p,n)$ and $\boldsymbol{\Psi}(\lambda_k) \in \mathcal{S}t_c(p,m)$ (see target Algorithm 1). We thus have

$$\boldsymbol{\Phi}(\lambda_k)^T \boldsymbol{\Phi}(\lambda_k) = \boldsymbol{\Psi}(\lambda_k)^T \boldsymbol{\Psi}(\lambda_k) = \mathbf{I}_p,$$

so that by the left and right multiplication of (19) we obtain formula (18) for the mixed part.

Now we have:

Lemma 4 *Let $\lambda \mapsto \mathbf{M}(\lambda) \in \mathrm{Mat}_{p,p}(\mathbb{R})$ be any interpolated curve between the mixed part matrices*

$$\mathbf{M}_k := \boldsymbol{\Phi}(\lambda_k)^T \mathbf{S}_p^{(k)} \boldsymbol{\Psi}(\lambda_k) \in \mathrm{Mat}_{p,p}(\mathbb{R})$$

so that $\mathbf{M}(\lambda_k) = \mathbf{M}_k$ for $k = 1, \ldots, N$. Then, using the curve $\lambda \mapsto \boldsymbol{\Phi}(\lambda)$ (resp. $\lambda \mapsto \boldsymbol{\Psi}(\lambda)$) defined by the target Algorithm 1 applied on the matrices $\boldsymbol{\Phi}_p^{(k)}$ (resp. $\boldsymbol{\Psi}_p^{(k)}$), the curve

$$\kappa \,:\, \lambda \mapsto \boldsymbol{\Phi}(\lambda)\mathbf{M}(\lambda)\boldsymbol{\Psi}(\lambda)^T \tag{20}$$

is an interpolated curve between the matrices $\mathbf{S}_p^{(1)}, \ldots, \mathbf{S}_p^{(N)}$, so that $\kappa(\lambda_k) = \mathbf{S}_p^{(k)}$ for each $k = 1, \ldots, N$.

Proof We need to check that $\kappa(\lambda_k) = \mathbf{S}_p^{(k)}$ for each $k = 1, \ldots, N$. Now:

$$\begin{aligned}
\kappa(\lambda_k) &= \boldsymbol{\Phi}(\lambda_k)\mathbf{M}(\lambda_k)\boldsymbol{\Psi}(\lambda_k)^T = \boldsymbol{\Phi}(\lambda_k)\mathbf{M}_k\boldsymbol{\Psi}(\lambda_k)^T \\
&= \boldsymbol{\Phi}(\lambda_k)\boldsymbol{\Phi}(\lambda_k)^T\mathbf{S}_p^{(k)}\boldsymbol{\Psi}(\lambda_k)\boldsymbol{\Psi}(\lambda_k)^T \\
&= \boldsymbol{\Phi}(\lambda_k)\boldsymbol{\Phi}(\lambda_k)^T \underbrace{\boldsymbol{\Phi}_p^{(k)}\boldsymbol{\Sigma}_p^{(k)}(\boldsymbol{\Psi}_p^{(k)})^T}_{\mathbf{S}_p^{(k)}} \boldsymbol{\Psi}(\lambda_k)\boldsymbol{\Psi}(\lambda_k)^T
\end{aligned}$$

where $\boldsymbol{\Phi}(\lambda_k)\boldsymbol{\Phi}^T(\lambda_k)$ corresponds to the projection matrix on the subspace $\mathbf{m}_k := \pi\left(\boldsymbol{\Phi}(\lambda_k)\right) = \pi\left(\boldsymbol{\Phi}_k\right)$ (see Remark 5) so that

$$\boldsymbol{\Phi}(\lambda_k)\boldsymbol{\Phi}^T(\lambda_k)\boldsymbol{\Phi}_p^{(k)} = \boldsymbol{\Phi}_p^{(k)} \tag{21}$$

and the same being true for the temporal part, we obtain the proof of the lemma.

The Space–Time interpolation algorithm is now given by:

Algorithm 3 (Space–Time interpolation)

- **Inputs**:
  - Generic matrices $\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(N)}$ in $\mathrm{Mat}^0_{n,m}(\mathbb{R})$ $(m \leq n)$, corresponding to parameter values $\lambda_1 < \ldots < \lambda_N$.
  - A reference parameter value λ_{i_0} with $i_0 \in \{1, \ldots, N\}$.
  - A mode $p \leq m$.
  - A parameter value $\widetilde{\lambda}$.
- **Output**: A matrix $\widetilde{\mathbf{S}} \in \mathrm{Mat}_{n,m}(\mathbb{R})$.

1. Compute an oriented SVD on each matrix $\mathbf{S}^{(k)}$ and write a POD of mode p

$$\mathbf{S}_p^{(k)} := \boldsymbol{\Phi}_p \boldsymbol{\Sigma}_p^{(k)} (\boldsymbol{\Psi}_p^{(k)})^T \in \mathrm{Mat}_{n,m}(\mathbb{R})$$

   with $\boldsymbol{\Phi}_p^{(k)} \in \mathcal{S}t_c(p,n)$ and $\boldsymbol{\Psi}_p^{(k)} \in \mathcal{S}t_c(p,m)$ uniquely defined.

2. Consider the target Algorithm 1 applied to the spatial parts $\boldsymbol{\Phi}_p^{(1)}, \ldots, \boldsymbol{\Phi}_p^{(N)}$, reference parameter value λ_{i_0} and each of the $N+1$ parameter values $\lambda_1, \ldots, \lambda_N, \widetilde{\lambda}$, so from (8) we can define matrices in $\mathcal{S}t_c(p,n)$:

$$\boldsymbol{\Phi}(\lambda_k) := \mathbf{Y}(\lambda_k), \quad \boldsymbol{\Phi}(\widetilde{\lambda}) := \mathbf{Y}(\widetilde{\lambda}). \tag{22}$$

3. Consider the target Algorithm 1 applied to the temporal parts $\boldsymbol{\Psi}_p^{(1)}, \ldots, \boldsymbol{\Psi}_p^{(N)}$ reference parameter value λ_{i_0} and each of the $N+1$ parameter values $\lambda_1, \ldots, \lambda_N, \widetilde{\lambda}$, so from (8) we can define matrices in $\mathcal{S}t_c(p,m)$:

$$\boldsymbol{\Psi}(\lambda_k) := \mathbf{Y}(\lambda_k), \quad \boldsymbol{\Psi}(\widetilde{\lambda}) := \mathbf{Y}(\widetilde{\lambda}). \tag{23}$$

4. For each $k = 1, \dots, N$, define the square matrix of the mixed part

$$\mathbf{M}_k := \boldsymbol{\Phi}(\lambda_k)^T \mathbf{S}_p^{(k)} \boldsymbol{\Psi}(\lambda_k) \in \mathrm{Mat}_{p,p}(\mathbb{R}). \tag{24}$$

5. Use a standard interpolation on square matrices $\mathbf{M}_1, \dots \mathbf{M}_N$, for instance:

$$\mathbf{M}(\widetilde{\lambda}) := \sum_{i=1}^{N} \prod_{i \neq j} \frac{\widetilde{\lambda} - \lambda_j}{\lambda_i - \lambda_j} \mathbf{M}_i \tag{25}$$

6. Using the spatial part $\boldsymbol{\Phi}(\widetilde{\lambda}) \in \mathcal{S}t_c(p, n)$ from (22), the temporal part $\boldsymbol{\Psi}(\widetilde{\lambda}) \in \mathcal{S}t_c(p, m)$ from (23), and the mixed part $\mathbf{M}(\widetilde{\lambda}) \in \mathrm{Mat}_{p,p}(\mathbb{R})$ from (25), the interpolated snapshot matrix corresponding to $\widetilde{\lambda}$ is finally given by

$$\widetilde{\mathbf{S}} := \boldsymbol{\Phi}(\widetilde{\lambda}) \mathbf{M}(\widetilde{\lambda}) \boldsymbol{\Psi}(\widetilde{\lambda})^T \in \mathrm{Mat}_{n,m}(\mathbb{R}).$$

4 Rigid-Viscoplastic FEM Formulation

The main defining characteristic of the RVP formulation is that it neglects the elasticity effects. This *idealization* is based on the fact that elastic components of strain remain small as compared with irreversible strains. This means that the additive decomposition of the total strain-rate tensor $\dot{\varepsilon}_{ij} = \dot{\varepsilon}^e_{ij} + \dot{\varepsilon}^p_{ij}$ simplifies to $\dot{\varepsilon}_{ij} = \dot{\varepsilon}^p_{ij}$, where $\dot{\varepsilon}^e_{ij}$ is the elastic component of the strain-rate tensor, $\dot{\varepsilon}^p_{ij}$ is the plastic component and $\dot{\varepsilon}_{ij}$ is the total strain-rate tensor. Therefore, the RVP formulation turns out to be very similar to fluid flow problems, and it is also called *flow formulation* [44]. Although it is not possible to calculate the residual stresses and the spring-back effect, the flow formulation presents several advantages. Unlike the elastoplastic FEM, the RVP formulation, even though more approximate, is more stable, simpler to be implemented in computer codes, and can use relatively larger time increments, thus improving the computational efficiency. A thorough overview of the foundation of the theory can be found in [34, 1].

4.1 Governing Field Equations

Classical rigid viscoplastic problems consider the plastic deformation of an isotropic body occupying a domain $\Omega \subset \mathbb{R}^3$. The domain Ω and its boundary $\partial\Omega$ represent the current configuration of a body according to the *Updated Lagrangian* formulation. The governing equations that have to be satisfied are:

(a) Equilibrium condition:

$$\sigma_{ij,j} = 0$$

(b) Compatibility conditions:

$$\dot{\varepsilon}_{ij} = \frac{1}{2}(v_{i,j} + v_{j,i})$$

(c) Yield criterion:

$$\bar{\sigma} := \left(\frac{2}{3}\sigma'_{ij}\sigma'_{ij}\right)^{\frac{1}{2}} = \bar{\sigma}(\bar{\varepsilon}, \dot{\bar{\varepsilon}}, T)$$

(d) Constitutive equations:

$$\sigma'_{ij} = \frac{2}{3}\frac{\bar{\sigma}}{\dot{\bar{\varepsilon}}}\dot{\varepsilon}_{ij}, \qquad \dot{\bar{\varepsilon}} = \left(\frac{2}{3}\dot{\varepsilon}_{ij}\dot{\varepsilon}_{ij}\right)^{\frac{1}{2}} \tag{26}$$

(e) Incompressibility condition:

$$\dot{\varepsilon}_v := \dot{\varepsilon}_{kk} = 0$$

(f) Boundary conditions:

$$\mathbf{v} = \hat{\mathbf{v}} \quad \text{on} \quad \partial\Omega_v$$

$$\mathbf{F} = \hat{\mathbf{F}} \quad \text{on} \quad \partial\Omega_F$$

$$\text{friction and contact} \quad \text{on} \quad \partial\Omega_c$$

In the above equations $\boldsymbol{\sigma} = (\sigma_{ij})$ is the stress tensor, $\dot{\boldsymbol{\varepsilon}} = (\dot{\varepsilon}_{ij})$ is the strain rate tensor, v_i are velocity components, $\bar{\sigma}$ is the effective stress, $\dot{\bar{\varepsilon}}$ is the second invariant of $\dot{\boldsymbol{\varepsilon}}$ called effective strain rate, and $\boldsymbol{\sigma'} = (\sigma'_{ij})$ is the deviatoric stress tensor defined by $\sigma'_{ij} = \sigma_{ij} - \delta_{ij}\sigma_{kk}/3$.

The hat symbol $\hat{}$ denotes prescribed values. Generally, the boundary $\partial\Omega$ consists of three distinct parts: over $\partial\Omega_v$ velocity conditions are prescribed (essential boundary conditions), $\partial\Omega_F$ is the part where the traction conditions are imposed in the form of nodal point forces (natural boundary conditions), while the boundary conditions along $\partial\Omega_c$ are mixed, and neither the velocity nor the force can be described. Therefore, we have the disjoint union:

$$\partial\Omega = \partial\Omega_v \cup \partial\Omega_F \cup \partial\Omega_c \tag{27}$$

4.2 Variational form

In a variational formulation, the functional Π (energy rate) is defined by an integral form in accordance with the virtual work-rate principle

$$\Pi(v) := \int_\Omega \bar{\sigma}\dot{\bar{\varepsilon}} dV - \int_{\partial\Omega_F} F_i v_i dS \tag{28}$$

where the first term in (28) represents the internal deformation work-rate, whereas the second term represents the work-rate done by the external forces. F_i denotes prescribed surface tractions on the boundary surface $\partial\Omega_F$. Recalling the MarKov-Hill [45,46] variational principle, among all virtual (admissible) continuous and continuously differentiable velocity fields v_i satisfying the conditions of compatibility and incompressibility, as well

as the velocity boundary conditions, the real velocity field gives to the functional Π a stationary value, i.e., the first-order variation vanishes. Moreover, in order to relax the incompressibility constraint condition $\dot{\varepsilon}_v = \dot{\varepsilon}_{kk} = 0$ on an admissible velocity field, a classical penalized form is used

$$\delta\Pi := \int_\Omega \bar{\sigma}\delta\dot{\bar{\varepsilon}}dV + \frac{1}{2}\int_\Omega K\dot{\varepsilon}_v\delta\dot{\varepsilon}_v dV - \int_{\partial\Omega_F} F_i\delta v_i dS = 0 \tag{29}$$

where K is a large positive constant which penalizes the dilatational strain-rate component. It can be shown that the mean stress is $\sigma_m = K\dot{\varepsilon}_{kk}$.

Remark 6 A limitation of the Updated Lagrangian method for large deformation problems is the excessive element distortion. To this end, remeshing processes are necessary to simulate unconstrained plastic flows. A mesh generation process is activated in case of zero or negative determinant of the Jacobian matrix, or due to various element quality criteria. Then, a new mesh is calculated conforming to the current state of the geometry followed by an interpolation of the state variables between the old and the newly generated mesh. Thus, the information of the remapping process has to adequately be transferred to the ROM basis obtained using the POD snapshot method. We remark that at this first attempt, we avoid remeshings of the workpiece during the course of the simulation. This topic will be addressed in a future investigation.

4.3 Discretization and iteration

The discretization of the functional follows the standard procedure of the finite element method. Eq. (29) is expressed in terms of nodal point velocities v_i and their variations

δv_i. Using the variational principle

$$\delta \Pi = \sum_{m=1}^{M} \frac{\partial \Pi^{(m)}}{\partial v_i} \delta v_i = 0, \qquad i = 1, 2, ..., 2N_s, \tag{30}$$

where δv_i are arbitrary except that they must be zero to satisfy the corresponding essential boundary conditions, and M denotes the number of elements. From the arbitrariness of δv_i, a set of algebraic equations (stiffness equations) are obtained

$$\frac{\partial \Pi}{\partial v_i} = \sum_{m=1}^{M} \frac{\partial \Pi^{(m)}}{\partial v_i} = 0. \tag{31}$$

As the resulting algebraic equations are highly nonlinear, they linearized by the Taylor expansion near an assumed velocity field $\mathbf{v} = \mathbf{v}_0$ as

$$\left. \frac{\partial \Pi}{\partial v_i} \right|_{\mathbf{v}=\mathbf{v}_0} + \left. \frac{\partial^2 \Pi}{\partial v_i \partial v_j} \right|_{\mathbf{v}=\mathbf{v}_0} \Delta v_j = 0 \tag{32}$$

where the first factor of the second term is also known as the Jacobian of the system (Hessian matrix), and Δv_j is a first-order correction of the velocity component v_j. Solving (32) with respect to Δv_j, the assumed velocity field is updated by the form (written in vector notation)

$$\mathbf{v}^{(i)} = \mathbf{v}^{(i-1)} + \alpha (\Delta \mathbf{v})^{(i)} \tag{33}$$

where $0 \leq \alpha \leq 1$ and i is the iteration step. The solution is obtained by the Direct iteration method [34,47] and/or by Newton-Raphson type methods. The iteration process is repeated until the following described convergence criteria are satisfied simultaneously

$$\frac{\| \Delta \mathbf{v} \|_{L_2}}{\| \mathbf{v} \|_{L_2}} \leq e_1, \qquad \left\| \frac{\partial \Pi}{\partial \mathbf{v}} \right\|_{L_2} \leq e_2 \tag{34}$$

namely, the velocity error norm and the norm of the residual equations, where e_1 and e_2 are sufficiently small specified tolerance numbers.

4.4 Heat Transfer Analysis

In the present model, a thermodynamically sound derivation is adopted using the conservation of energy

$$-\rho c \frac{\partial T}{\partial t} + k\nabla^2 T + \xi \bar{\sigma}\dot{\bar{\varepsilon}} = 0 \tag{35}$$

where ρc is the volume-specific heat of the material, $\xi \bar{\sigma}\dot{\bar{\varepsilon}}$ represents the work heat rate per unit volume due to plastic deformation, k is the thermal conductivity, T is the temperature and ξ is a coefficient that presents the fraction of the deformation energy dissipated into heat also known as the Taylor-Quinney coefficient.

In a weak form, and using the divergence theorem

$$-\int_{\Omega} \xi \bar{\sigma}\dot{\bar{\varepsilon}}\delta T dV + \int_{\Omega} k\nabla T \delta(\nabla T) dV + \int_{\Omega} \rho c \frac{\vartheta T}{\vartheta t}\delta T dV - \int_{\partial\Omega} q_n \delta T dS = 0 \tag{36}$$

where

$$q_n := k\frac{\partial T}{\partial n} \tag{37}$$

is the heat flux across the boundary $\partial\Omega$ and n denotes the unit normal vector to the boundary surface $\partial\Omega$.

In standard finite element books, e.g. [48], it can be seen that the heat balance equations such as (36), upon finite element discretization are reduced to the form:

$$\boldsymbol{C}\dot{\boldsymbol{T}} + \boldsymbol{K}\boldsymbol{T} = \boldsymbol{Q} \tag{38}$$

where $\boldsymbol{C}$ is the heat capacity matrix, $\boldsymbol{K}$ denotes the heat conduction matrix, $\boldsymbol{Q}$ is the heat flux vector, $\boldsymbol{T}$ is the vector of nodal point temperatures, and $\dot{\boldsymbol{T}}$ is the rate of temperature increase vector of nodal points.

The theory necessary to integrate (38) can be found in numerical analysis books [49, 50]. It suffices to say that one-step time integration is used. The convergence of a scheme requires consistency and stability. Consistency is satisfied by a general time integration scheme

$$^{t+\Delta t}\boldsymbol{T} = {}^{t}\boldsymbol{T} + \Delta t[(1-\theta)^{t}\dot{\boldsymbol{T}} + \theta^{t+\Delta t}\dot{\boldsymbol{T}}] \tag{39}$$

where θ is a parameter varying between 0 and 1 ($\theta = 0$: Forward difference, $\theta = 1/2$: Crank-Nicholson, $\theta = 2/3$: Galerkin, $\theta = 1$: Backward difference).

Remark 7 Unconditional stability is obtained for $\theta \geq 0.5$. This is important, because it is desirable to take time steps as large as the deformation formulation allows, since this is the most expensive part of the process.

4.5 Computational Procedure for Thermo-Mechanical Analysis

For solving *coupled* thermomechanical problems, two different approaches can be used. In the traditional *monolithic* approach, a single solver is in charge of the solution of

the entire system of equations. In an alternative approach, the mechanical and thermal solvers deal respectively with the viscoplastic flow and the thermal field equations. Thus, in the so-called *staggered solution* procedure used here, the state of the system is advanced by sequentially executing and exchange information between these two solvers [51]. The equations for the mechanical analysis and the temperature calculation are *strongly coupled*, thereby making necessary the simultaneous solution of the finite element counterparts [34, 52, 53].

5 Numerical Investigations

The purpose of this section is to evaluate the performance of the ST POD interpolation using the velocity and temperature fields during the course of the simulation of the forming process. As a benchmark test case, a rectangular cross-section bar is compressed between two parallel flat dies under the condition of a constant shear friction factor m at the die-workpiece interface. The initial workpiece has dimensions $h = 20$ mm (height) and $w = 20$ mm (width). Plane strain conditions are considered. Due to the symmetry of the problem, only one quarter of the cross-section is analyzed. The velocity of the upper and the lower die is set to $v = 1$ mm/s. The initial temperature of the die and the workpiece is set to $T = 25$ °C. The bar is compressed until a 35% reduction in height is achieved. The final simulation state is accomplished in 7-time steps with a constant time increment $\Delta t = 0.5$ s. One can observe the complexity of the nonuniform deformation presented by the barreling of the free surface (Figure 6). In our calculations, we employ a conventional rate-dependent power law to describe the material flow stress equation

$$\bar{\sigma}(\dot{\bar{\varepsilon}}) = 1000\dot{\bar{\varepsilon}}^{0.1} \qquad \text{(MPa)} \tag{40}$$

743 The solution convergence is assumed when the velocity error norm and the force error
744 norm (34) becomes less than 10^{-6}. The type of element used is the linear isoparametric
745 rectangular element with four-point integration. However, one point integration is used
746 for the dilatation term, the second integral of the functional in (29). This is known as the
747 reduced integration scheme which imposes the volume constancy averaged over the linear
748 rectangular element. The computational grid composed of 100 elements interconnected at
749 $N_s = 121$ nodes with 2 degrees of freedom, resulting in a global stiffness matrix of size
750 242×242. For the rigid-viscoplastic analysis, the limiting strain rate $\dot{\bar{\varepsilon}}_0$ is chosen to be
751 0.01 and the penalty constant (or bulk modulus) K is set to 10^5.

752 Among the various models of friction, the one proposed in [54] is adapted to model
753 the sliding contact at the tool-workpiece interface. This model allows the variation of the
754 tangential traction with the relative velocity at the tool-workpiece interface

755 $$\mathbf{t}_f = -mk\frac{\mathbf{v}_s}{|\mathbf{v}_s|} \simeq -mk\left\{\frac{2}{\pi}\arctan\left(\frac{|\mathbf{v}_s|}{v_0}\right)\right\}\frac{\mathbf{v}_s}{|\mathbf{v}_s|}$$

756 where $\mathbf{v}_s$ is the relative velocity in the tangential direction between the tool and the
757 workpiece, and v_0 is a positive constant several orders of magnitude smaller than $\mathbf{v}_s$; m is
758 the friction factor ($0 < m < 1$) and k is the material shear yield stress $k = \bar{\sigma}/\sqrt{3}$. For the
759 compression tests considered here, the relative tangential velocity at the tool-workpiece
760 interface at the beginning of deformation is zero. The present analysis assumes that the
761 friction factor remains constant throughout compression. Investigations on frictional shear
762 stress measurements over the interface between a cylindrical workpiece and a die during
763 plastic compression are reported in [55]. The basic characteristics of algorithms used in
764 the RVP FEM analysis are summarized in Table 1.

Basic characteristics of algorithms in RVP FEM	
Type of problem	Two dimensional, plane strain, rigid viscoplastic material flow, isotropic, homogeneous
Thermomechanical problem solution	Loose coupling (staggered) - Backward Euler difference ($\theta = 1$)
Type of elements	4-node quadrilateral isoparametric elements, bilinear shape functions
Flow stress equation	Power law: $\bar{\sigma}(\dot{\bar{\varepsilon}}) = c\dot{\bar{\varepsilon}}^p$, c, p constants
Iteration method	Direct, BFGS with line search
Remeshing	N/A
Boundary conditions	Sliding friction on S_c

Table 1: Numerical algorithms.

Remark 8 Note that during the course of the simulation we avoid remeshing of the workpiece. As discussed in [56], remeshing techniques can be taken into account provided that mesh transfer operations are applied to the reduced-basis.

5.1 Mechanical field

The first case for numerical illustration of the method considers the velocity field during the simulation of the forming process using the shear friction factor m as the investigated parameter. From now on, let the shear friction factor m denoted as λ for convenience with the previous sections notation. For the numerical study, the following training points are selected $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$. The choice made here, is to use a minimum number of sampling points equi-

distributed over the parametric range. The target point is set to $\widetilde{\lambda} = 0.3$. See the FEM solutions for the training and target points at the final state of the computation in Figure 6.

For each parametric simulation, a sequence of snapshots uniformly distributed over time using an increment of $\Delta t = 0.5$ s is extracted for all nodes of the workpiece. The space-time snapshot matrices $\mathbf{S}^{(i)} \in \mathrm{Mat}_{2N_s,N_t}(\mathbb{R})$ with $2N_s = 242$ and $N_t = 7$, corresponding to parameter values λ_i, are associated with the nodal velocity field in x and y directions.

For the parametric Space-Time interpolation, the snapshot matrix $\widetilde{\mathbf{S}}$ of mode p corresponding to the target point $\widetilde{\lambda}$ is computed via the target Algorithm 3. The target Algorithm 1 is applied to the spatial $\boldsymbol{\Phi}_p^{(1)}, \ldots, \boldsymbol{\Phi}_p^{(N)}$ and temporal parts $\boldsymbol{\Psi}_p^{(1)}, \ldots, \boldsymbol{\Psi}_p^{(N)}$, with reference parameter value $\lambda_{i_0} = 0.5$. In order to assess the interpolation acuracy, the snapshot matrix $\widetilde{\mathbf{S}}$ is compared against the high-fidelity FEM solution by introducing the following a posteriori errors. Using the interpolated and the HF-FEM snapshot matrices $\widetilde{\mathbf{S}}$ and $\mathbf{S}^{\mathrm{FEM}}$, respectively, the relative L_2-error measure is defined as

$$e_{L_2}(\widetilde{\mathbf{s}}_i) := \frac{\|\widetilde{\mathbf{s}}_i - \mathbf{s}_i^{\mathrm{FEM}}\|_{L_2}}{\|\mathbf{s}_i^{\mathrm{FEM}}\|_{L_2}}, \quad i = 1, \ldots, p \leq N_t. \tag{41}$$

Additionally, the relative Frobenius error norm of $\widetilde{\mathbf{S}}$ and $\mathbf{S}^{\mathrm{FEM}}$ is defined as

$$e_F(\widetilde{\mathbf{S}}) := \|\widetilde{\mathbf{S}} - \mathbf{S}^{\mathrm{FEM}}\|_F / \|\mathbf{S}^{\mathrm{FEM}}\|_F. \tag{42}$$

The eigenvalue spectrum of snapshot matrices $\mathbf{S}^{(i)}$ corresponding to training points $\lambda_i \in \Lambda_t$ is exhibited in a semi-log scale in Figure 7. We can observe that the distance between the first and the last eigenvalue is from 5 up to 6 orders of magnitude. Moreover, the percentage of energy $\mathcal{E}(k) = \sum_{i=1}^{k} \sigma_i^2 / \sum_{i=1}^{N_t} \sigma_i^2$ captured from the POD modes is shown in Figure 8. It is evident that most of the 99.9% of the total energy is contained by the first two POD modes.

The relative L_2-error norm $e_{L_2}(\widetilde{\mathbf{s}}_i)$ (see (41)) between the interpolated and the HF-FEM solution for various POD modes is displayed in Figure 9. In general, the relative error for all POD modes lie within a range of 0.0175 up to 0.038. It can be observed that the interpolated ST

POD solution delivers good accuracy and is reliable enough to predict the velocity field for the investigated target point.

Remark 9 In the case of using $p = 7$ POD modes for the temporal basis interpolation, the Grassmannian manifold $\mathcal{G}(p, p)$ *reduces to one point*, so it is not relevant to use the target Algorithm 1: any new parameter value will give rise to the same matrix $\boldsymbol{\Psi}_{i_0}$ in the associated compact Stiefel manifold, corresponding to the reference point.

Additionally, the position vector error $e_{L_2}(\widetilde{\mathbf{x}}(t)) = \|\widetilde{\mathbf{x}}(t) - \mathbf{x}^{\text{FEM}}(t)\|_{L_2}$ at the nodal points is computed for p =2,3,5 and 7 POD modes, where $\widetilde{\mathbf{x}}(t)$ and $\mathbf{x}^{\text{FEM}}(t)$ denotes the position vector of the ST POD and the high-fidelity FEM solutions, respectively, at the time increments during the deformation. Figure 10 presents the local error $e_{L_2}(\widetilde{\mathbf{x}}(t))$ superimposed at the final loading state $t = 0.35$ s obtained from the high-fidelity FEM solution. Different patterns of the spatial error distribution can be observed concerning the number of POD modes p. It is interesting to observe that in both cases, the maximum error is located near the upper-right location of the deforming workpiece.

The evolution of the deformation process can be also represented using the time-displacement histories of some selected nodes of the workpiece (Figure 11). The ST POD predictions are compared against the high-fidelity FEM counterpart solution using $p = 2$ POD modes. Again, it can be observed that the interpolated ST POD solution is accurate and reliable to predict the evolution of the displacement field for the investigated target point during the forming process.

For the preceding numerical investigations, the ST POD efficiency is demonstrated using a single target point, i.e., $\widetilde{\lambda} = 0.3$. To further assess the interpolation performance, a new target point is now considered, $\widetilde{\lambda} = 0.8$. Interpolation is performed using the same set of training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$, with reference parameter value $\lambda_{i_0} = 0.5$. The relative L_2-error norm $e_{L_2}(\widetilde{\mathbf{s}}_i)$ for various POD modes p corresponding to target point $\widetilde{\lambda} = 0.8$ is shown in Figure 12.

Again, one can observe that the relative error lies within a narrow range of the values, i.e., 0.014 up to 0.026.

5.2 Temperature field

To further investigate the performance of the proposed ST POD interpolation, the temperature field obtained from the coupled thermomechanical simulation of the forming process is considered. Again, for the temperature field, we consider the shear friction factor m as the investigated system parameter. The training points selected for the mechanical field analysis are also used in this study, i.e., $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$. The target point is set to $\widetilde{\lambda} = 0.3$. For each parametric problem, snapshots are uniformly distributed over time using an increment step size $\Delta t = 0.5$ s. The final deformation state is reached at $t = 0.35$ s. The space-time snapshot matrices $\mathbf{S}^{(i)} \in \mathrm{Mat}_{N_s,N_t}(\mathbb{R})$ of size 121×7, corresponding to λ_i, are associated with nodal temperatures. We will now compare the Space-Time interpolation (see Algorithm 3) against the high-fidelity FEM solution. Again, for the target Algorithm 1 applied to the spatial $\boldsymbol{\Phi}_p^{(1)}, \ldots, \boldsymbol{\Phi}_p^{(N)}$ and temporal parts $\boldsymbol{\Psi}_p^{(1)}, \ldots, \boldsymbol{\Psi}_p^{(N)}$, the reference parameter value $\lambda_{i_0} = 0.5$ is used.

Figure 13 presents the temperature profiles at the final compression state obtained using different values of the shear friction factor m (represented by parameter λ). The temperature rises due to plastic work conversion to heat assuming a constant value for the Taylor-Quinney coefficient $\xi = 0.9$. In all cases, the maximum temperature is located at the center of the workpiece with values ranging from $T = 89.5$ °C up to $T = 98$ °C.

The eigenvalue spectrum of snapshot matrices $\mathbf{S}^{(i)}$ corresponding to training points $\lambda_i \in \Lambda_t$ is shown in a semi-log scale in Figure 14. We can observe that the distance between the first and the last eigenvalue of the curves is of the order of 5 up to 6 orders of magnitude. Moreover, the system energy $\mathcal{E}(k) = \sum_{i=1}^{k} \sigma_i^2 / \sum_{i=1}^{N_t} \sigma_i^2$ captured from the POD modes is shown in Figure 15. Most of the 99.9% of the total energy is contained by the first two POD modes.

The relative L_2-error norm $e_{L_2}(\widetilde{\mathbf{s}}_i)$ (41) between the interpolated and the HF-FEM snapshot matrices $\widetilde{\mathbf{S}}$ and $\mathbf{S}^{\text{FEM}}$, respectively, for various modes p is shown in Figure 16. Additionally, the Frobenius relative error norm (42) for the POD modes is presented in Figure 17. In general, the obtained results are found to have less than 1% relative error for POD modes $p > 1$ and therefore are acceptable as fast near real-time numerical predictions.

Finally, Figure 18 shows the ST POD time-temperature histories for some selected nodes of the workpiece using $p = 7$ modes. The predictions are compared against the high-fidelity counterpart solution, and it is difficult to distinguish differences among these plots. It is revealed that the interpolated ST POD solution delivers good accuracy for all selected nodes.

5.3 Computational complexity

The computational cost of the ST POD interpolation scales with the computational complexity of SVD and the matrix operations in the target ST Algorithm 3. It is evident, that the cost of ST POD interpolation will be lower compared to the standard POD Galerkin nonlinear approaches and even lower than the full order FEM solution. The coupled thermomechanical FEM simulation for the target point takes 35.123 seconds in wall-clock time. On the other hand, the ST interpolation for the mechanical problem using a ROM POD basis of mode $p = 4$ results in 0.147 seconds in wall-clock time. The ST interpolation for the thermal problem using a ROM POD basis of mode $p = 4$ results in 0.153 seconds in wall-clock time. Therefore, the total ST interpolation takes 0.3 seconds in wall-clock time corresponding to a time speed-up of 116.96. All experiments in this section were implemented in Matlab and run on a 4th Generation Intel(R) Core(TM) i7-4600U CPU @ 2.10GHz, 8GB RAM, 250 GB SSD, Debian 9 x64.

6 Conclusions

A novel non-intrusive Space-Time POD basis interpolation scheme on compact Stiefel manifolds is developed and applied to parametric high nonlinear metal forming problems. Apart from the separate interpolation of POD spatial and temporal basis on associated Grassmannian manifolds, an interpolation function is defined on a set of parametric snapshot matrices. This function results from curves, which are defined on compact Stiefel manifolds both for space and the temporal part, and also the use of some mixed part encoded by a square matrix. This latter matrix provides a link between the interpolated space and temporal basis for the construction of the target ROM snapshot matrix. To prove the efficiency of the method it has been used a coupled thermomechanical rigid-viscoplastic FEM formulation which is integrated into the manufacturing industry in a variety of applications. The performed numerical investigations have considered the reconstruction of the ROM snapshot matrices both of the velocity and the temperature fields. Moreover, the error norms of the Space-Time POD interpolated ROM models concerning the associated high-fidelity FEM counterpart solutions are validating the accuracy of the proposed interpolation scheme. In conclusion, the overall results demonstrate the potential use of the proposed ST POD interpolation scheme for near real-time parametric simulations using off-line computed ROM POD databases, supporting thus manufacturing industries to accelerate design-to-production timespans, and thereby reducing costs while ensuring the design of superior processes.

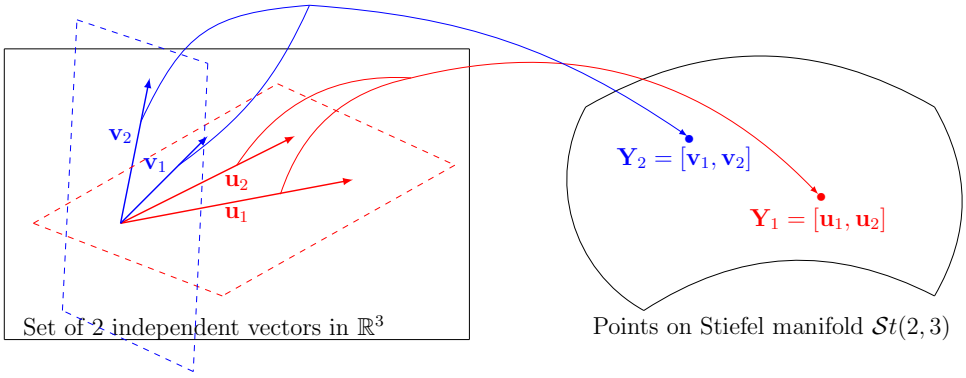


Fig. 1: Points on Stiefel manifold. The linearly independent vectors in $\mathbb{R}^3$ spanning the red and blue planes correspond to points in $\mathcal{S}t_c(2,3)$.

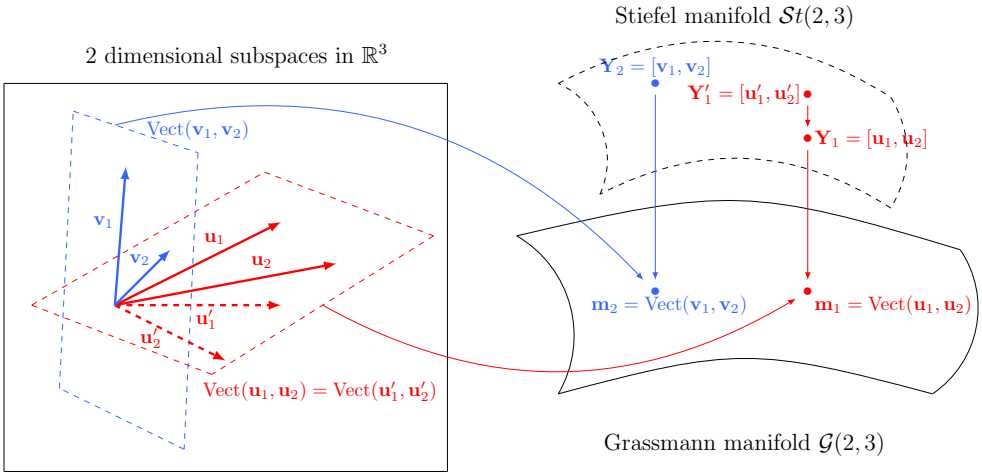


Fig. 2: Points on Stiefel $\mathcal{S}t(2,3)$ and Grassmann manifold $\mathcal{G}(2,3)$.

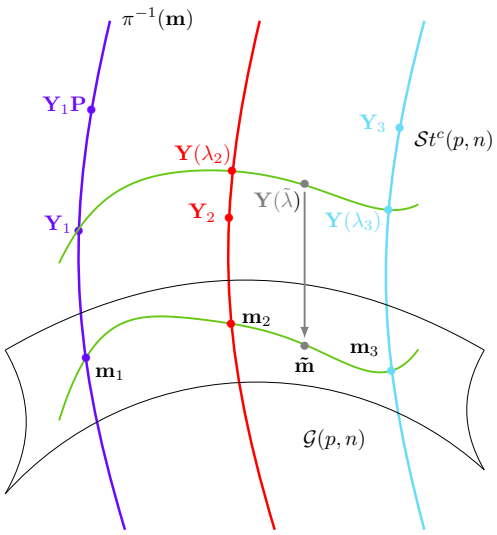


Fig. 3: There is a natural projection $\pi : \mathcal{S}t_c(p,n) \longrightarrow \mathcal{G}(p,n)$ from the compact Stiefel manifold $\mathcal{S}t_c(p,n)$ to the Grassmannian $\mathcal{G}(p,n)$ of p-dimensional subspaces in $\mathbb{R}^n$ which sends a p-frame to the subspace spanned by that frame. The fiber over a given point $\mathbf{m}$ on $\mathcal{G}(p,n)$ is the set of all orthonormal p-frames spanning the subspace $\mathbf{m}$. Computations on $\mathcal{S}t_c(p,n)$ using the target Algorithm 1 for $\lambda := \lambda_k$, lead to some matrix $\mathbf{Y}(\lambda_k)$ generally different from $\mathbf{Y}_k$ (except for the reference point), and thus do not produce an interpolation on the points $\mathbf{Y}_1, \ldots, \mathbf{Y}_N$.

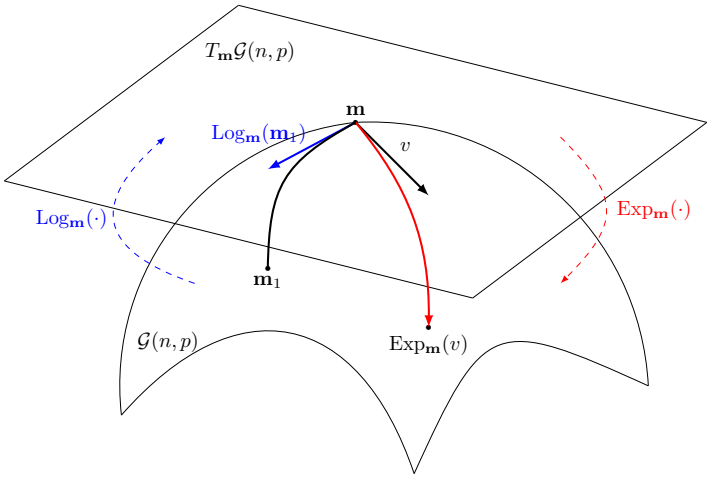


Fig. 4: The exponential $\mathrm{Exp}_{\mathbf{m}}$ and the logarithm $\mathrm{Log}_{\mathbf{m}}$ map on the Grassmann manifold $\mathcal{G}(p,n)$.

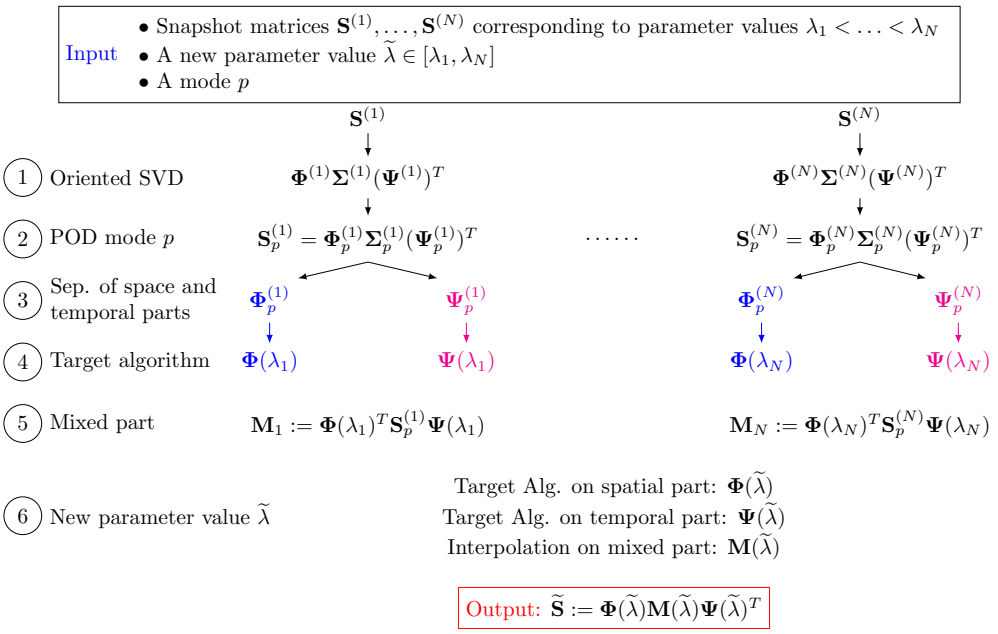
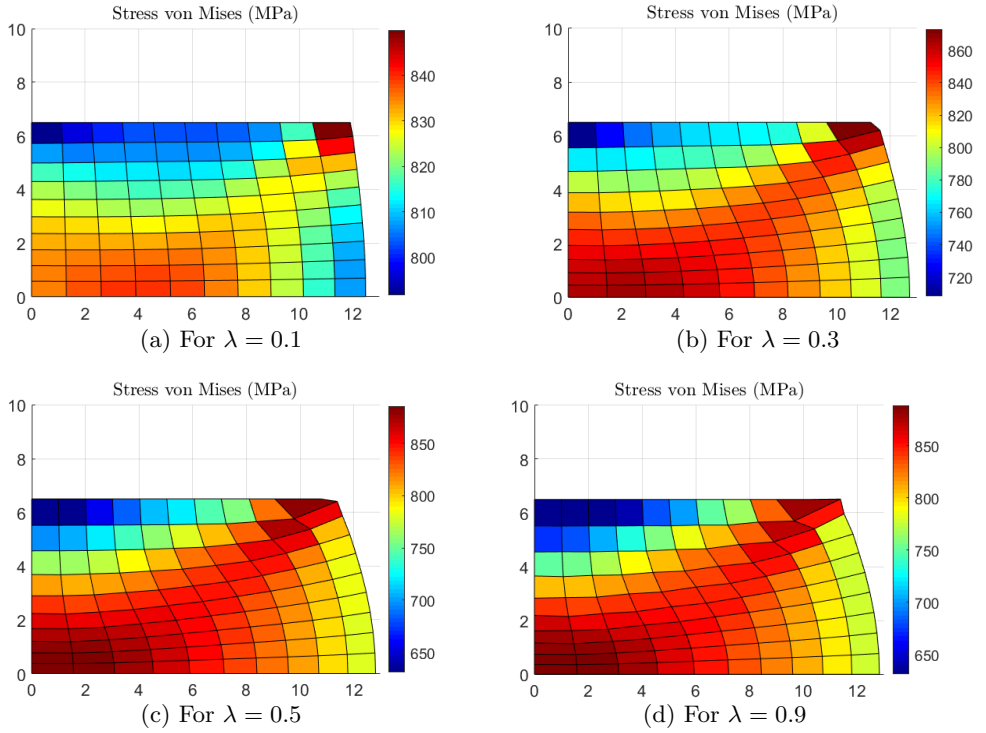


Fig. 5: The Space-Time Algorithm.



(a) For $\lambda = 0.1$ (b) For $\lambda = 0.3$

(c) For $\lambda = 0.5$ (d) For $\lambda = 0.9$

Fig. 6: Deformation patterns of the benchmark metal forming example using different values for the shear friction factor m represented by the parameter λ.

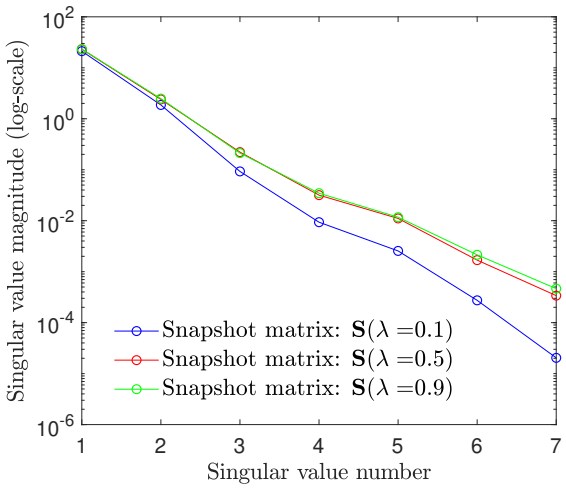


Fig. 7: The eigenvalue spectrum of snapshot matrices $\mathbf{S}^{(i)}$ corresponding to training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$.

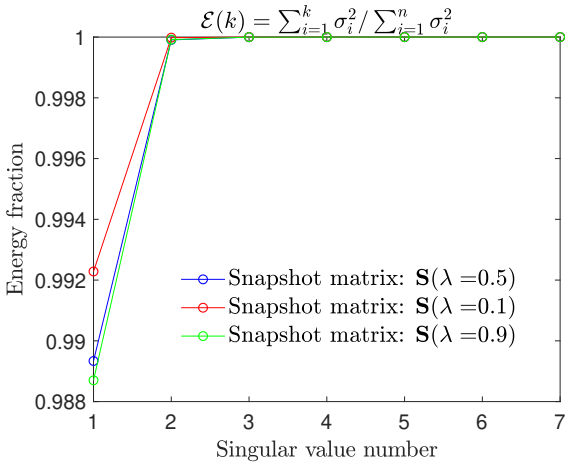


Fig. 8: Energy captured by the singular values of snapshot matrices $\mathbf{S}^{(i)}$ corresponding to training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$.

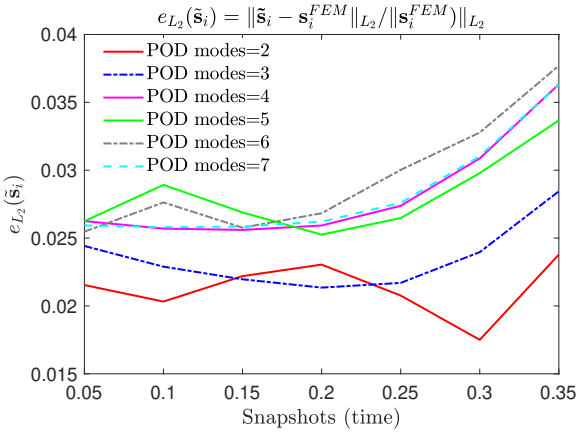


Fig. 9: Performance of ST POD interpolation using the relative L_2-error norm $e_{L_2}(\widetilde{\mathbf{s}}_i)$ for various modes p; training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$; reference parameter value $\lambda_{i_0} = 0.5$; target point $\widetilde{\lambda} = 0.3$.

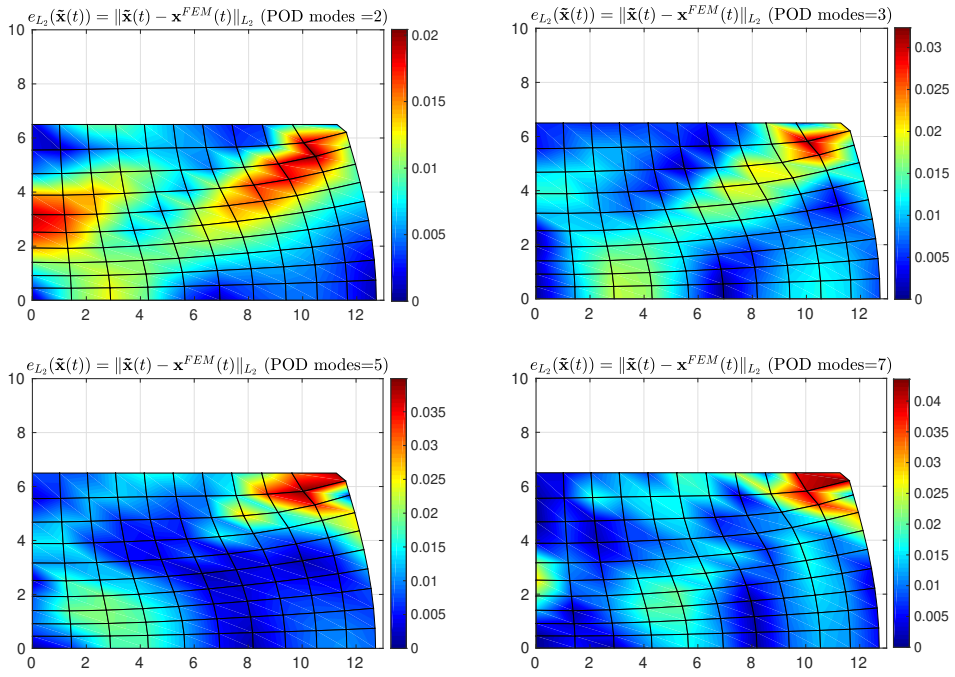


Fig. 10: The position vector error $e_{L_2}(\widetilde{\mathbf{x}}(t)) = \|\widetilde{\mathbf{x}}(t) - \mathbf{x}^{\mathrm{FEM}}(t)\|_{L_2}$ of the nodal points at the final deformation state $t = 0.35$ s superimposed on the high-fidelity FEM solution; POD modes $p = \{2, 3, 5, 7\}$; training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$; reference parameter value $\lambda_{i_0} = 0.5$; target point $\widetilde{\lambda} = 0.3$.

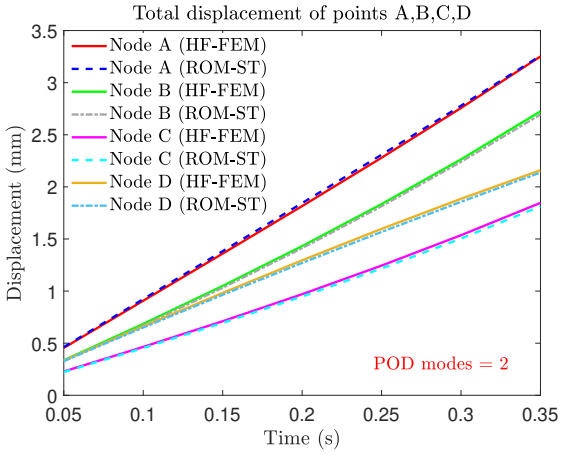


Fig. 11: Comparison of the total displacement of selected nodes against the high-fidelity FEM solution; training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$; reference parameter value $\lambda_{i_0} = 0.5$; target point $\widetilde{\lambda} = 0.3$; POD modes $p = 2$.

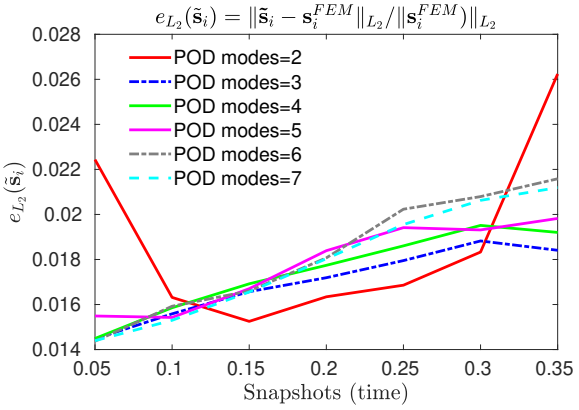
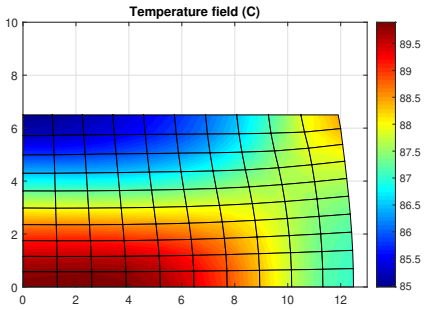
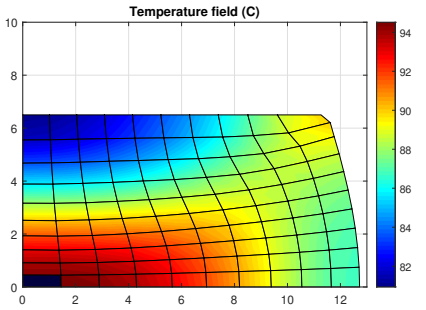


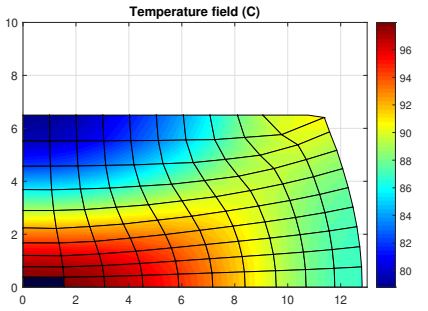
Fig. 12: Performance of ST POD interpolation using the relative L_2-error norm $e_{L_2}(\widetilde{\mathbf{s}}_i)$ for various POD modes p; training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$; reference parameter value $\lambda_{i_0} = 0.5$; target point $\widetilde{\lambda} = 0.8$.



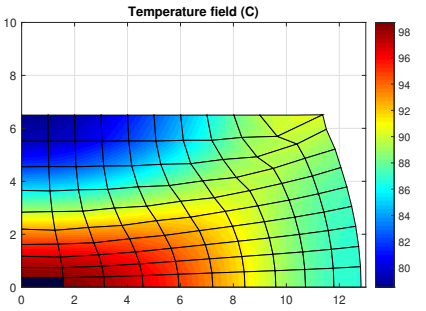
(a) For $m = 0.1$



(b) For $m = 0.3$



(c) For $m = 0.5$



(d) For $m = 0.9$

Fig. 13: Temperature profiles at the final compression state $t = 0.35$ s obtained using different values of the shear friction factor m represented by parameter λ.

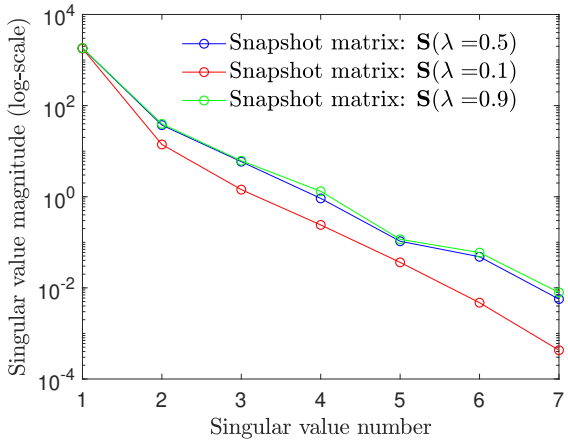


Fig. 14: The eigenvalue spectrum of snapshot matrices $\mathbf{S}^{(i)}$ corresponding to training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$.

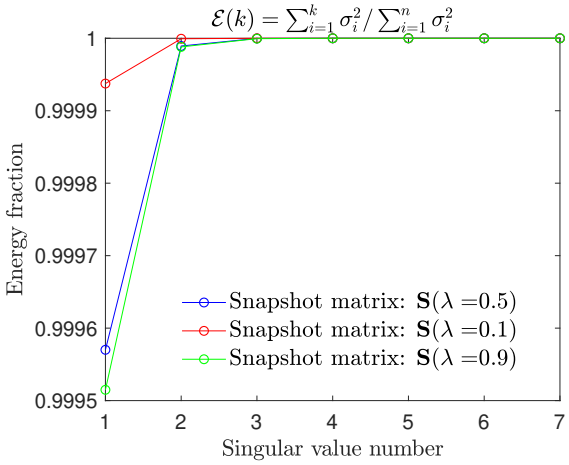


Fig. 15: Energy captured by the singular values of snapshot matrices $\mathbf{S}^{(i)}$ corresponding to training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$.

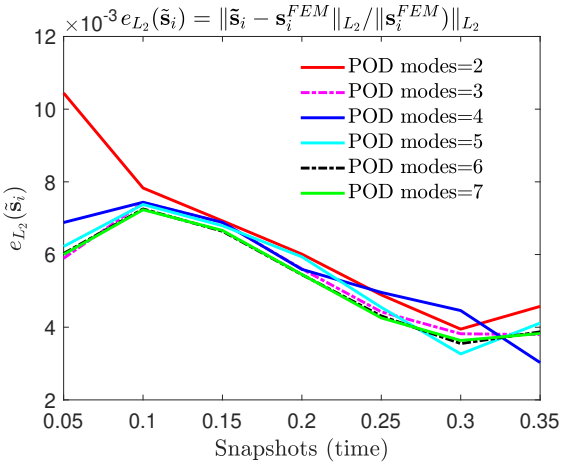


Fig. 16: Performance of ST POD interpolation using the relative L_2-error norm $e_{L_2}(\widetilde{\mathbf{s}}_i)$ for various POD modes p; training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$; reference parameter value $\lambda_{i_0} = 0.5$; target point $\widetilde{\lambda} = 0.3$.

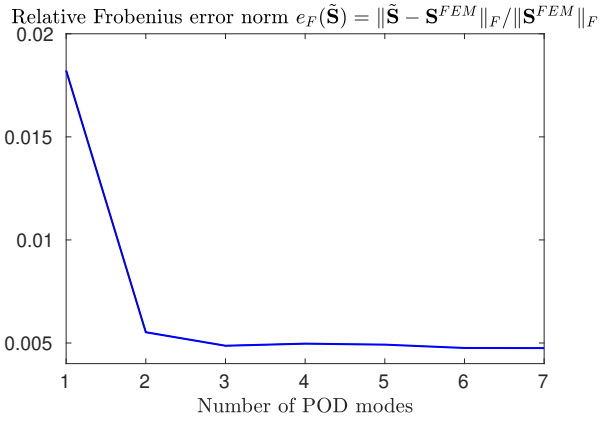


Fig. 17: Performance of the POD interpolation using the relative Frobenius error norm $e_F(\widetilde{\mathbf{S}})$ against the number of POD modes p; training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$; reference parameter value $\lambda_{i_0} = 0.5$; target point $\widetilde{\lambda} = 0.3$.

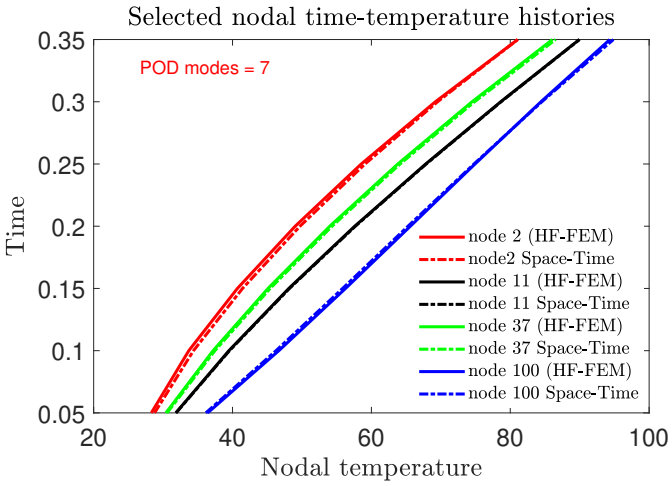


Fig. 18: Temperature evolution of selected nodal points validated against the high-fidelity FEM solution; ST POD and HF-FEM solutions virtually coincide; training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$; reference parameter value $\lambda_{i_0} = 0.5$; target point $\widetilde{\lambda} = 0.3$; POD modes $p = 7$.

References

1. J.-L. Chenot. Recent contributions to the finite element modelling of metal forming processes. *Journal of Materials Processing Technology*, 34(1-4):9–18, sep 1992.
2. Z. Gronostajski, Z. Pater, L. Madej, A. Gontarz, L. Lisiecki, A. Łukaszek-Sołek, J. Łuksza, S. Mróz, Z. Muskalski, W. Muzykiewicz, M. Pietrzyk, R.E. Śliwa, J. Tomczak, S. Wiewiórowska, G. Winiarski, J. Zasadziński, and S. Ziółkiewicz. Recent development trends in metal forming. *Archives of Civil and Mechanical Engineering*, 19(3):898–941, may 2019.
3. Francisco Chinesta, Pierre Ladeveze, and Elías Cueto. A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18(4):395–404, oct 2011.
4. C. Allery, A. Hamdouni, D. Ryckelynck, and N. Verdon. A priori reduction method for solving the two-dimensional burgers' equations. *Applied Mathematics and Computation*, 217(15):6671–6679, apr 2011.
5. Philip Holmes, John L Lumley, Gahl Berkooz, and Clarence W Rowley. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 2012.
6. Thibault Henri and Jean-Pierre Yvon. Convergence estimates of POD-galerkin methods for parabolic problems. In *IFIP International Federation for Information Processing*, pages 295–306. Kluwer Academic Publishers, 2005.
7. Nadine Aubry. On the hidden beauty of the proper orthogonal decomposition. *Theoretical and Computational Fluid Dynamics*, 2(5-6):339–352, aug 1991.
8. Kari Karhunen. Zur spektraltheorie stochastischer prozesse. *Ann. Acad. Sci. Fennicae, AI*, 34, 1946.
9. M. Loève. Elementary probability theory. In *Probability Theory I*, pages 1–52. Springer New York, 1977.

911 10. Gene H Golub and CFV Loan. *Matrix Computations, 3rd edn., vol. 1.* JHU Press, 1996.

912 11. Ian T Jolliffe. Springer series in statistics. *Principal component analysis*, 29, 2002.

913 12. Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary*
914 *reviews: computational statistics*, 2(4):433–459, 2010.

915 13. J. Edward Jackson. Principal components and factor analysis: Part i principal components.
916 *Journal of Quality Technology*, 12(4):201–213, oct 1980.

917 14. J. Edward Jackson. Principal components and factor analysis: Part II—additional topics
918 related to principal components. *Journal of Quality Technology*, 13(1):46–58, jan 1981.

919 15. Patricia Astrid, Siep Weiland, Karen Willcox, and Ton Backx. Missing point estimation in
920 models described by proper orthogonal decomposition. *IEEE Transactions on Automatic*
921 *Control*, 53(10):2237–2251, nov 2008.

922 16. Annika Radermacher and Stefanie Reese. POD-based model reduction with empirical in-
923 terpolation applied to nonlinear elasticity. *International Journal for Numerical Methods in*
924 *Engineering*, 107(6):477–495, dec 2015.

925 17. Saifon Chaturantabut and Danny C. Sorensen. Nonlinear model reduction via discrete em-
926 pirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, jan 2010.

927 18. R. Everson and L. Sirovich. Karhunen–loève procedure for gappy data. *Journal of the Optical*
928 *Society of America A*, 12(8):1657, aug 1995.

929 19. Kevin Carlberg, Charbel Farhat, Julien Cortial, and David Amsallem. The GNAT method
930 for nonlinear model reduction: Effective implementation and application to computational
931 fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242:623–647, jun
932 2013.

933 20. David Amsallem, Julien Cortial, Kevin Carlberg, and Charbel Farhat. A method for inter-
934 polating on manifolds structural dynamics reduced-order models. *International journal for*
935 *numerical methods in engineering*, 80(9):1241–1258, 2009.

21. Rolando Mosquera Meza. *Interpolation sur les variétés grassmanniennes et applications à la réduction de modèles en mécanique*. PhD thesis, La Rochelle, 2018.
22. Silvère Bonnabel and Rodolphe Sepulchre. Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1055–1070, jan 2010.
23. John M Lee. Smooth manifolds. In *Introduction to Smooth Manifolds*, pages 1–31. Springer, 2013.
24. Sylvestre Gallot, Dominique Hulin, and Jacques Lafontaine. *Riemannian geometry*, volume 2. Springer, 1990.
25. Rolando Mosquera, , Aziz Hamdouni, Abdallah El Hamidi, and Cyrille Allery. POD basis interpolation via inverse distance weighting on grassmann manifolds. *Discrete & Continuous Dynamical Systems - S*, 12(6):1743–1759, 2019.
26. G. Muhlbach. The general neville-aitken-algorithm and some applications. *Numerische Mathematik*, 31(1):97–110, mar 1978.
27. Y. Lu, N. Blal, and A. Gravouil. Space–time POD based computational vademecums for parametric studies: application to thermo-mechanical problems. *Advanced Modeling and Simulation in Engineering Sciences*, 5(1), feb 2018.
28. M. Oulghelou and C. Allery. Non intrusive method for parametric model order reduction using a bi-calibrated interpolation on the grassmann manifold. *Journal of Computational Physics*, 426:109924, feb 2021.
29. Vilas Shinde, Elisabeth Longatte, Franck Baj, Yannick Hoarau, and Marianna Braza. A galerkin-free model reduction approach for the navier–stokes equations. *Journal of Computational Physics*, 309:148–163, mar 2016.
30. C. Audouze, F. De Vuyst, and P. B. Nair. Reduced-order modeling of parameterized PDEs using time-space-parameter principal component analysis. *International Journal for Numer-*

961 *ical Methods in Engineering*, 80(8):1025–1057, nov 2009.

962 31. Youngsoo Choi and Kevin Carlberg. Space–time least-squares petrov–galerkin projection
963 for nonlinear model reduction. *SIAM Journal on Scientific Computing*, 41(1):A26–A58, jan
964 2019.

965 32. Youngsoo Choi, Peter Brown, William Arrighi, Robert Anderson, and Kevin Huynh.
966 Space–time reduced order model for large-scale linear dynamical systems with application to
967 boltzmann transport problems. *Journal of Computational Physics*, 424:109845, jan 2021.

968 33. Christophe Audouze, Florian De Vuyst, and Prasanth B. Nair. Nonintrusive reduced-order
969 modeling of parametrized time-dependent partial differential equations. *Numerical Methods*
970 *for Partial Differential Equations*, 29(5):1587–1628, feb 2013.

971 34. Shiro Kobayashi, Shirō Kobayashi, Soo-Ik Oh, and Taylan Altan. *Metal forming and the*
972 *finite-element method*, volume 4. Oxford University Press on Demand, 1989.

973 35. C. H. Lee and S. Kobayashi. New solutions to rigid-plastic deformation problems using a
974 matrix method. *Journal of Engineering for Industry*, 95(3):865–873, aug 1973.

975 36. Shiro Kobayashi. Rigid-plastic finite element analysis of axisymmetric metal forming pro-
976 cesses. *Numerical Modeling of Manuf. Process (ASME, New York, 1977)*, pages 49–65, 1977.

977 37. ZQ Feng and G De Saxcé. Rigid-plastic implicit integration scheme for analysis of metal
978 forming. *European Journal of Mechanics, A/Solids*, 15(1):51–66, 1996.

979 38. O. Friderikos. Two-dimensional rigid-plastic fem simulation of metal forming processes in
980 matlab. Proceedings of the 4th International Conference on Manufacturing and Materials
981 Engineering (ICMMEN), 3-5 October, Thessaloniki, Greece, 2011.

982 39. Alan Edelman, Tomás A Arias, and Steven T Smith. The geometry of algorithms with
983 orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, 20(2):303–
984 353, 1998.

985 40. P.-A. Absil, R. Mahony, and R. Sepulchre. Riemannian geometry of grassmann manifolds
986 with a view on algorithmic computation. *Acta Applicandae Mathematicae*, 80(2):199–220,
987 jan 2004.

988 41. S. E. Kozlov. Geometry of the real grassmannian manifolds. parts i, ii. *Zapiski Nauchnykh*
989 *Seminarov POMI*, 246:84–107, 1997.

990 42. Yung-Chow Wong. Differential geometry of grassmann manifolds. *Proceedings of the National*
991 *Academy of Sciences of the United States of America*, 57(3):589, 1967.

992 43. S. Berceanu. On the geometry of complex grassmann manifold, its noncompact dual and
993 coherent states. *Bulletin of the Belgian Mathematical Society - Simon Stevin*, 4(2):205–243,
994 1997.

995 44. Rodney Hill. *The mathematical theory of plasticity*, volume 11. Oxford university press, 1998.

996 45. A.A. Markov. *On variational principles in the theory of plasticity.* Division of Applied
997 Mathematics, Brown University, 1948.

998 46. R Hill. A variational principle of maximum plastic work in classical plasticity. *The Quarterly*
999 *Journal of Mechanics and Applied Mathematics*, 1(1):18–28, 1948.

1000 47. S.I. Oh. Finite element analysis of metal forming processes with arbitrarily shaped dies.
1001 *International Journal of Mechanical Sciences*, 24(8):479–493, jan 1982.

1002 48. O. C. Zienkiewicz and P. N. Godbole. Flow of plastic and visco-plastic solids with special
1003 reference to extrusion and forming processes. *International Journal for Numerical Methods*
1004 *in Engineering*, 8(1):1–16, 1974.

1005 49. Anthony Ralston and Philip Rabinowitz. *A first course in numerical analysis.* Courier
1006 Corporation, 2001.

1007 50. Germund Dahlquist and Åke Björck. *Numerical Methods in Scientific Computing, Volume I.*
1008 Society for Industrial and Applied Mathematics, jan 2008.

1009 51. C.A. Felippa and K.C. Park. Staggered transient analysis procedures for coupled mechanical
1010 systems: Formulation. *Computer Methods in Applied Mechanics and Engineering*, 24(1):61–
1011 111, oct 1980.
1012 52. N. Rebelo and S. Kobayashi. A coupled analysis of viscoplastic deformation and heat trans-
1013 fer—i. *International Journal of Mechanical Sciences*, 22(11):699–705, jan 1980.
1014 53. N. Rebelo and S. Kobayashi. A coupled analysis of viscoplastic deformation and heat trans-
1015 fer—II. *International Journal of Mechanical Sciences*, 22(11):707–718, jan 1980.
1016 54. CC Chen. Rigid-plastic finite-element analysis of ring compression. *Applications of numerical*
1017 *methods of forming processes*, 1978.
1018 55. G.T. van Rooyen and W.A. Backofen. A study of interface friction in plastic compression.
1019 *International Journal of Mechanical Sciences*, 1(1):1–27, jan 1960.
1020 56. D. Ryckelynck. Hyper-reduction of mechanical models involving internal variables. *Interna-*
1021 *tional Journal for Numerical Methods in Engineering*, 77(1):75–89, jan 2009.