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

O. Friderikos, Marc Olive, Emmanuel Baranger, Dimitris Sagris, Constantine David

A NON-INTRUSIVE SPACE-TIME INTERPOLATION FROM COMPACT STIEFEL MANIFOLDS OF PARAMETRIZED RIGID-VISCOPLASTIC FEM PROBLEMS

O. FRIDERIKOS, M. OLIVE, E. BARANGER, D. SAGRIS, AND C. DAVID

Abstract. This work aims to interpolate Reduced Order Model (ROM) basis constructed via the Proper Orthogonal Decomposition (POD) which correspond to a limited number of training points to derive a robust ROM of the system dynamics for an unseen target parameter. A novel Space-Time (ST) POD basis interpolation, for which we define ROM spatial and time basis *curves on compact Stiefel manifolds*, is proposed. A classical interpolation is finally defined on a *mixed part* encoded in a square matrix to obtain a ST interpolation on POD. The method is illustrated with the adaptation of rigid-thermoviscoplastic finite element ROMs considering an open forging metal forming process to new values of the shear friction factor. 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.

1. Introduction

Computational metal forming has been widely used in academia laboratories and manufacturing industry over the last decades, becoming nowadays a mature, well established technology. Nevertheless, various numerical issues are still under investigation and new challenging fields are emerging, among others, massive parallel computing, adaptive meshing and remeshing, coupling approaches, multiscale modeling, optimization of processes or parameter identification, computation at the micro-scale level, machine learning and stochastic approaches [1, 22]. One of the key challenging future topics mentioned in [1] is the introduction of model order reduction techniques to combat the high computational cost. Due to multiple sources of strong non-linearities inherent in metal forming problems, the design analysis of large scale models turns often to be prohibitively expensive. Indeed, simulation of complex configurations can be intractable since the computational times can highly increase for a parametric analysis. To this end, ROMs via the POD have been chosen to reduce the problem dimensionality while maintaining solution accuracy.

ROMs are used to decrease the complexity of large-scale systems and solve parametrized problems. One popular method is the Proper Orthogonal Decomposition (POD) [24, 25, 5], also known as Kharhunen-Loève Decomposition (KLD) [31, 39], Singular Value Decomposition (SVD) [21] or Principal Component Analysis (PCA) [28, 2, 26, 27].

For a parametric analysis, the method starts by a training stage during which the problem is solved for a limited number of parameters. 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. Now, for a new parameter value, interpolation methods has to be defined from the underlying spanned *subspaces* of the POD basis [4]. For instance, a POD basis interpolation can be performed using *normal coordinates* on Grassmann manifolds by evaluating the geodesic paths between the subspaces on this manifold, all this being done in the framework of *Riemannian geometry*, which is a specific matter of differential geometry.

An introduction to the geometry of the Grassmann manifolds (strongly connected to the geometry of Stiefel manifold) can be found in [15] in which Newton and conjugate gradient algorithms on these manifolds are developed. Algorithms for numerical optimization on Riemannian manifolds are provided in [3].

Note that for the method developed by Amsallem [4], the solution is dependent on the reference point selection, reflecting the choice of a *local chart* in the Grassmann manifold. Naturally, one can choose as a reference point the closest training point to the target one, however, this approach is not straightforward for a multivariate parametric analysis. Another approach using inverse distance weighting was initiated by Mosquera et. al [42, 43], but this approach also relies on several choices (as the one of the weights). Furthermore, an extension of the 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 [44].

In *standard* POD interpolation, the spatial ROM basis corresponding to the target parameter is used to generate a ROM FEM model which is expected to have a lower computational cost compared to the high-fidelity problem. In *Space-Time* (ST) POD basis interpolation, the reduced spatial and time basis are considered separately, so to define specific *curves* on associated compact Stiefel manifold. We then consider a *mixed part* defined from square matrix directly deduced using the space part, the singular values and the time part, so to obtain an *interpolated snapshot matrix*, keeping track of accurate space and time eigenvectors. Note that to obtain a well defined interpolation on compact Stiefel manifold, we had to introduce an new SVD procedure, called the *oriented compact SVD*. Such an oriented SVD produces unique right and left eigenvectors for *generic matrices*, for which singular values are all distinct.

In ST approach, the off-line stage characterizes the procedure of solving FEM problems corresponding to the training points. The on-line stage concerns the interpolation on Grassmann manifolds to determine the spatial and time ROM basis for the target point in order to construct the related ROM snapshot matrix. In fact, ST interpolation offers the advantage of reconstructing a snapshot matrix without relaunching ROM-FEM computations. To this end, it results into near-real time solutions due to direct matrix multiplications in the online stage.

We refer to the following non-exhaustive literature review. Shinde et al. [49] proposed to approximate the spatial and temporal basis functions by linear interpolation of their modes to study the flow past a cylinder at low Reynolds number. A Space-Time interpolation with an application for parametric studies of thermo-mechanical welding problems with a moving heat source in presented in [40]. Furthermore, POD/Galerkin and POD/interpolation techniques are investigated in the context of eddy current damage detection in [30]. Using a RBF interpolation, a non-intrusive reduced-order modelling approach for nonlinear parametrized time-dependent PDEs based on a two-level POD method is proposed in [6, 7].

The method illustrated here is applied to a coupled thermomechanical analysis using a rigid visco-plastic (RVP) FEM based on an incremental implicit approach [33], [38], [32], [17]. The RVP formulation is especially 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, simulations are performed in an in-house Matlab program which consists of two independent FEM solvers. A solver for viscoplastic deformation analysis [19] and a solver for heat transfer analysis. A staggered procedure is used to solve the system of coupled equations.

The present 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 Grassmann manifolds. 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 in section 3, where the computational framework for the reduced order model adaptation based on a novel non-intrusive Space-Time interpolation on compact Stiefel manifolds is developed. Next, 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 using a bulk metal forming forging process is shown, and further computational aspects are discussed. Finally, section 6 highlights the main results and some important outcomes.

2. Grassmannian manifold and POD

Let recall here the important link between proper orthogonal decomposition and Grassmann manifold [4, 15, 3, 42, 43].

Take $\mathbf{S}$ to be any real matrix of size $n \times m$, taken here to be a snapshot matrix with $n = 3N_S$ obtained from spatial discretization N_s and $m = N_t$ obtained from 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

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

is minimal, where matrix $\mathbf{\Pi}_p$ correspond to the orthogonal projection on $\mathcal{V}_p$ (see [43] for more details). Now, the set $\mathcal{G}(p,n)$ of all p dimensional subspaces in $\mathbb{R}^n$ defines a *Grassmannian manifold*, and spatial POD of mode p on $\mathbf{S}$ defines a *point* $\mathbf{m}$ in $\mathcal{G}(p,n)$.

Take now a set $\{\lambda_1, \dots, \lambda_N\}$ of parameters (which is for instance the shear friction factor in section 5). Parameters $\lambda_1, \dots, \lambda_N$ leads to snapshot matrices $\mathbf{S}^{(1)}, \dots, \mathbf{S}^{(N)}$ with spatial POD of mode p corresponding to points $\mathbf{m}_i$ ($i = 1, \dots, N$) in $\mathcal{G}(p,n)$. To any new target parameter $\widetilde{\lambda}$, it is possible to make an interpolation on $\mathcal{G}(p,n)$ using a local chart given by normal coordinates [4, 42, 43], so to obtain a point $\widetilde{\mathbf{m}} \in \mathcal{G}(p,n)$. From such a point, some POD-Galerkin approach [43] can lead to a new snapshot matrix $\widetilde{\mathbf{S}}$.

In fact, we propose here another approach as we consider *space–time interpolation* using *compact Stiefel manifold* $\mathcal{S}t_c(p,n)$, that is the manifold defined by sets of p orthonormal vectors in $\mathbb{R}^n$ [15]. The main observation is that interpolation on Grassmann manifolds can be considered both for the space and for the time parts. But this defines in fact some *curves* on the associated compact Stiefel manifold.

Nevertheless, all computations will be done using *normal coordinates* defined by an exponential and logarithmic map [20, section 2.C.2] on the Grassmann manifold $\mathcal{G}(p,n)$, directly obtained from its Riemannian structure summarized now.

2.1. **Riemannian geometry on Grassmann manifolds.** Let $p \leq n$ be two non-zero integers and $\mathcal{G}(p,n)$ the Grassmann manifold of p dimensional subspace in $\mathbb{R}^n$. Such manifold is in fact a *complete Riemannian manifold* [20], so that any two points can be joined by a geodesic, and the shortest geodesic between two points defines a distance on $\mathcal{G}(p,n)$.

We summarize now some essential results about Grassmannian manifold. Note also that a strong mathematical background is given in [34].

First of all, Grassmann manifolds are special cases of *quotient manifolds*. To do so, let define the compact Stiefel manifold $\mathcal{S}t_c(p,n)$ to be the set of p orthonormal ordered vectors of $\mathbb{R}^n$. Taking the canonical basis of $\mathbb{R}^n$, this translates into:

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

where $\mathbf{I}_p$ is the identity matrix in $\mathrm{Mat}_{p,p}(\mathbb{R})$. Any given matrix $\mathbf{Y} \in \mathcal{S}t_c(p,n)$ defines (in a non–unique way) a p dimensional subspace in $\mathbb{R}^n$, which is a point in $\mathcal{G}(p,n)$, so that we have a submersion

$$\pi \;:\; \mathcal{S}t_c(p,n) \longrightarrow \mathcal{G}(p,n).$$

Remark 2.1. Another point of view is to consider a submersion coming from the non–compact Stiefel manifold (defined as sets of odered linearly independant vectors), and thus using the set of rank p matrices in $\mathrm{Mat}_{n,p}(\mathbb{R})$. According to our computations, it was more relevant to consider the compact case.

The *fiber* at any p dimensional subspace $\mathbf{m} \in \mathcal{G}(p,n)$, generated by orthonormal vectors $\mathbf{y}_1, \dots, \mathbf{y}_p$, is the set

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

where $\mathbf{Y} := [\mathbf{y}_1, \dots, \mathbf{y}_p]$ and $\mathrm{O}(p)$ is the group of orthogonal transformation on $\mathbb{R}^p$.

Remark 2.2. 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.

For any point $\mathbf{Y} \in \mathcal{S}t_c(p,n)$, its *tangent space* [15] is defined as the vector space

$$\mathcal{T}_{\mathbf{Y}} := \left\{\mathbf{Z} \in \mathrm{Mat}_{n,p}(\mathbb{R}), \quad \mathbf{Z}^T\mathbf{Y} \text{ is skew–symmetric}\right\}$$

and an $\mathrm{O}(p)$ invariant Riemannian metric on $\mathcal{S}t_c(p,n)$ is given point–wise by the scalar product [52, 3]:

$$\langle \mathbf{Z}_1, \mathbf{Z}_2 \rangle_{\mathbf{Y}} := \mathrm{tr}(\mathbf{Z}_1^T \mathbf{Z}_2), \quad \forall \mathbf{Z}_1, \mathbf{Z}_2 \in \mathcal{T}_{\mathbf{Y}}.$$

According to the general framework of such Riemannian quotient manifold [37, Chapter 2], the Grassmannian $\mathcal{G}(p,n)$ inherits of a Riemannian metric for which π becomes a *Riemannian submersion*. More precisely, for any point $\mathbf{m} \in \mathcal{G}(p,n)$, the tangent space $\mathcal{T}_{\mathbf{m}}$ is isomorphic to any *horizontal space*

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

where $\mathbf{Y}$ is in the fiber of $\mathbf{m}$. For a given choice of $\mathbf{Y} \in \pi^{-1}(\mathbf{m})$, we can associate to any velocity vector $v \in \mathcal{T}_{\mathbf{m}}$ a unique *horizontal lift* $\mathbf{Z}$ in $\mathcal{T}_{\mathbf{Y}}$. Finally, a Riemannian metric on $\mathcal{G}(p,n)$ is given point–wise by the scalar product:

$$\langle v_1, v_2 \rangle_{\mathbf{m}} := \mathrm{tr}\left(\mathbf{Z}_1^T \mathbf{Z}_2\right) \tag{2}$$

where $\mathbf{Z}_i$ is a horizontal lift of v_i in $\mathcal{T}_{\mathbf{Y}}$.

Such a Riemannian metric leads to explicit geodesics given by [3, 15]:

Theorem 2.3. *Let $\mathbf{m} \in \mathcal{G}(p,n)$ and $\mathbf{Y} \in \mathcal{S}t_c(p,n)$ in its fiber. For any $v \in \mathcal{T}_{\mathbf{m}}$ with horizontal lift given by $\mathbf{Z}$ in $\mathcal{T}_{\mathbf{Y}}$, let $\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be a thin singular value decomposition of $\mathbf{Z}$. Then*

$$\alpha_v : t \in \mathbb{R} \mapsto \mathbf{m}(t) = \pi\left[\left(\mathbf{Y}\mathbf{V}\cos(t\mathbf{\Sigma}) + \mathbf{U}\sin(t\mathbf{\Sigma})\right)\mathbf{V}^T\right] \in \mathcal{G}(p,n) \tag{3}$$

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

Remark 2.4. Up to our knowledge, there is no proof that

$$\mathbf{Y}(t) := (\mathbf{YV}\cos(t\mathbf{\Sigma}) + \mathbf{U}\sin(t\mathbf{\Sigma}))\,\mathbf{V}^T \in \mathcal{S}t_c(p,n) \tag{4}$$

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

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

so that

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

Finally we have:

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

which concludes the proof.

Remark 2.5. In many cases, formulas of the geodesic don't use the right multiplication by $\mathbf{V}^T$, as for instance in [3, 43]. Of course, $\mathbf{V}$ being in $\mathrm{O}(p)$ both matrices

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

belong to the same fiber in $\mathcal{S}t_c(p,n)$. Now, the choice of such right multiplication in (4) is related to the choice of the horizontal lift $\mathbf{Z} = \mathbf{U\Sigma V}^T$. Indeed, we know that a given geodesic $\gamma(t)$ in $\mathcal{G}(p,n)$ has many *horizontal lifts* $\Gamma(t)$ in $\mathcal{S}t_c(p,n)$ (meaning that velocity vectors are always horizontal), each being uniquely defined by the initial velocity vector $\dot{\Gamma}(0)$. Taking back the path given by (4), we have

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

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

As a first consequence of Theorem 2.3, the geodesic distance between two points $\mathbf{m}_0, \mathbf{m}_1 \in \mathcal{G}(p,n)$ is given by the principal Jordan's angles [29, 9]. To do so, let consider $\mathbf{Y}_i$ in the fiber of $\mathbf{m}_i$ $(i = 0,1)$, both being in $\mathcal{S}t_c(p,n)$. Writing a thin singular value decomposition

$$\mathbf{Y}_0^T\mathbf{Y}_1 = \mathbf{U\Sigma V}, \quad \mathbf{\Sigma} = \mathrm{diag}(\sigma_1,\dots,\sigma_p), \quad 0 \le \sigma_p \le \dots \le \sigma_1 \le 1$$

then the Jordan principal angles between $\mathbf{m}_0$ and $\mathbf{m}_1$ are given by

$$0 \le \theta_i := \arccos\sigma_i \le \frac{\pi}{2}.$$

Finally, from [34, Theorem 12.2], the Riemannian distance is given by

$$d_r(\mathbf{m}_0, \mathbf{m}_1) = \left(\sum_{i=1}^{p} \theta_i^2 \right)^{1/2}. \tag{5}$$

Another consequence of Theorem 2.3 is an explicit formula for the exponential map [3, 43]:

Definition 2.6. Let $\mathbf{m} \in \mathcal{G}(p, n)$, then the exponential map at $\mathbf{m}$ is defined by

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

where α_v is given by (3) and $\mathbf{Y}$ is in the fiber of $\mathbf{m}$.

For any point $\mathbf{m} \in \mathcal{G}(p, n)$, the exponential map $\mathrm{Exp}_{\mathbf{m}}$ does not produce a local chart defined on all the tangent space $\mathcal{T}_{\mathbf{m}} \simeq \mathbb{R}^{p(n-p)}$. Nevertheless, a local chart can be defined on any open set $\mathrm{V} \subset \mathcal{T}_{\mathbf{m}}$ such that $\mathrm{Exp}_{\mathbf{m}}|_{\mathrm{V}}$ is an embedding, which then defines the so called normal coordinates on V.

A classical way to define such an open space is to use an open ball with radius $\pi/2$, where $\pi/2$ is the *injectivity radius* [20] of any Grassmann manifold [34] such that $\min(p, n-p) \geq 2$.

Theorem 2.7. *Let p, n be two integers such that $\min(p, n-p) \geq 2$. For $\mathbf{m} \in \mathcal{G}(p,n)$ define the open ball*

$$\mathrm{D}_{\mathbf{m}} := \left\{ v \in \mathcal{T}_{\mathbf{m}}, \quad \sqrt{\langle v, v \rangle_{\mathbf{m}}} < \frac{\pi}{2} \right\} \subset \mathcal{T}_{\mathbf{m}}$$

with $\langle \cdot, \cdot \rangle_{\mathbf{m}}$ given by (2). Then the restriction of $\mathrm{Exp}_{\mathbf{m}}$ on $\mathrm{D}_{\mathbf{m}}$ defines an embedding of $\mathrm{D}_{\mathbf{m}}$ into $\mathbb{R}^{p(n-p)}$.

In fact, it is also possible to define directly some inverse map of the exponential map, called the logarithm map [3]. For any $\mathbf{m}$ and $\mathbf{Y}$ in its fibers, let us first define the open space

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

Now we have:

Definition 2.8 (Logarithm map in Grassmannian manifold)**.** Let $\mathbf{m} \in \mathcal{G}(p,n)$ and $\mathbf{Y} \in \mathcal{S}t_c(p,n)$ be in its fiber. The logarithm map at $\mathbf{m}$ is defined on the open space $\mathrm{U}_{\mathbf{m}}$ by

$$\mathbf{m}_1 \in \mathrm{U}_{\mathbf{m}} \mapsto \mathrm{Log}_{\mathbf{m}}(\mathbf{m}_1) \in \mathcal{T}_{\mathbf{m}} \text{ with horizontal lift } \mathbf{U}\arctan(\mathbf{\Sigma})\mathbf{V}^T \tag{7}$$

where, taking $\mathbf{Y}_1$ in the fiber of $\mathbf{m}_1$, diagonal matrix $\mathbf{\Sigma}$ is given by the thin singular value decomposition

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

Remark 2.9. Such a logarithm map is such that or any $\mathbf{m} \in \mathcal{G}(p,n)$ and $\mathbf{m}_1 \in \mathrm{U}_{\mathbf{m}}$, we have

$$\mathrm{Exp}_{\mathbf{m}} \circ \mathrm{Log}_{\mathbf{m}}(\mathbf{m}_1) = \mathbf{m}_1$$

while we don't have $\mathrm{Log}_{\mathbf{m}} \circ \mathrm{Exp}_{\mathbf{m}}(v) = v$ for all $v \in \mathcal{T}_{\mathbf{m}}$.

2.2. Target algorithm on compact Stiefel manifolds.

All the mathematical background summarized in subsection 2.1 can be used to obtain interpolation on Grassmann manifold [4, 43]. Nevertheless, it can be noticed that all computations and formulas are given in the associated compact Stiefel manifold. From now on, we need to make a specific definition:

Definition 2.10 (Genericity)**.** A matrix is said to be *generic* if all its singular values are distinct. The set of generic matrix in $\mathcal{S}t_c(p,n)$ (resp. $\mathrm{Mat}_{n,p}(\mathbb{R})$) is denoted $\mathcal{S}t_c^0(p,n)$ (resp. $\mathrm{Mat}^0_{n,p}(\mathbb{R})$).

Let now be given N generic matrices $\mathbf{Y}_1, \ldots, \mathbf{Y}_N$ in $\mathcal{S}t_c^0(p,n)$. The idea here is that each matrix $\mathbf{Y}_k$ corresponds to some parameter $\lambda_k \in \mathbb{R}$ (with $\lambda_1 < \ldots < \lambda_N$) and the *target algorithm* given below, under assumption of genericity for some matrices, defines a map

$$\lambda \in [\lambda_1, \lambda_N] \mapsto \mathbf{Y}(\lambda) \in \mathcal{S}t_c(p,n)$$

where $\mathbf{Y}(\lambda)$ is obtained by (8).

Algorithm 2.11 (Target algorithm)**.** *Let $\mathbf{Y}_1, \ldots, \mathbf{Y}_N$ be matrices in $\mathcal{S}t_c^0(p,n)$*

(1) *Choose a reference matrix $\mathbf{Y}_{i_0} \in \{\mathbf{Y}_1, \ldots, \mathbf{Y}_N\}$.*

(2) *For $k \in \{1, \ldots, N\}$ with $k \neq i_0$ compute*

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

where $\mathbf{U}_k \in \mathrm{Mat}_{n,p}(\mathbb{R})$, $\mathbf{V}_k \in \mathrm{Mat}_{p,p}(\mathbb{R})$ and $\mathbf{\Sigma}_k \in \mathrm{Mat}_{p,p}(\mathbb{R})$ (diagonal) are given by any thin SVD

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

(3) *Define a $n \times p$ matrix and compute a thin SVD*

$$\mathbf{Z}(\lambda) := \sum_{i=1}^{N} \prod_{i \neq j} \frac{\lambda - \lambda_j}{\lambda_i - \lambda_j} \mathbf{Z}_i = \mathbf{U}(\lambda)\mathbf{\Sigma}(\lambda)\mathbf{V}(\lambda)^T, \quad \mathbf{Z}_{i_0} := 0,$$

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

(4) *Define the $n \times p$ matrix in $\mathcal{S}t_c(p,n)$ (see Remark 2.5):*

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

Note: cos and sin act only on diagonal entries.

In this algorithm, 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 2.12. Using this target algorithm to parameter $\lambda := \lambda_k$ 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 $\mathbf{Y}_1, \ldots, \mathbf{Y}_N$.

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 2.9). As a consequence, a projection matrix is given by $\mathbf{Y}(\lambda_k)^T\mathbf{Y}(\lambda_k)$ or $\mathbf{Y}_k^T\mathbf{Y}_k$.

3. Space-Time Interpolation on compact Stiefel manifolds

As already noticed, POD is extracting the optimal space structures and the associated time modes. An important property is that the spatial and temporal orthogonal modes are *coupled*: each space component is associated with a time 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 2.11.

The starting point here is 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 $\lambda_k \in \mathbb{R}$, with $\lambda_1 < \ldots < \lambda_N$ and $n = 3N_s$ corresponding to spatial part, while $m = N_t$ correspond to the time part.

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 singular value decomposition. Suppose now that $\mathbf{S}$ is of rank k. Any thin singular value decomposition of $\mathbf{S}$ with singular values $\sigma_1 > \ldots > \sigma_k$ leads to *spatial orthonormal vectors* $\phi_1, \ldots, \phi_k$ in $\mathbb{R}^n$ and *time orthonormal vectors* $\psi_1, \ldots, \psi_k$ in $\mathbb{R}^m$. Because of sign indeterminacy, such vectors are not well defined, so up to now, *we do not have well defined maps*

$$(9) \quad \mathbf{S} \mapsto \mathbf{\Phi}(\mathbf{S}) := [\phi_1, \ldots, \phi_k] \in \mathcal{S}t_c(k, n), \qquad \mathbf{S} \mapsto \mathbf{\Psi}(\mathbf{S}) := [\psi_1, \ldots, \psi_k] \in \mathcal{S}t_c(k, m).$$

To overcome this difficulty, we need to introduce a *new SVD* so that, under assumption of genericity (see Definition 2.10), the maps given by (9) can be well defined.

3.1. Oriented SVD on generic matrices.

The main idea of the new SVD introduced here is to make a choice on orientation for each space time vectors. A first Lemma, obtained by direct computation, allows us to use initial matrix $\mathbf{S}$ to choose orientation:

Lemma 3.1. *Let consider* $\mathbf{s}_1, \ldots, \mathbf{s}_m \in \mathbb{R}^n$ *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 to a non–zero singular value* σ*. Then, there exist* $i \in \{1, \ldots, m\}$ *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 consider

$$i(\phi) := \min\{j, \quad \langle \mathbf{s}_j, \phi \rangle \neq 0\}, \quad \mathbf{s}(\phi) := \mathbf{s}_{i(\phi)}. \tag{10}$$

Any spatial eigenvector can therefore have a specific orientation:

Definition 3.2 (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 deduce the new SVD:

Lemma 3.3 (Oriented compact SVD)**.** *Let* $\mathbf{S} \in \mathrm{Mat}^0_{n,m}(\mathbb{R})$ *of rank* k *such that all its non-zero singular values are distinct. Then, there exists one and only one couple of matrices*

$$\mathbf{\Phi} = [\phi_1, \dots, \phi_k] \in \mathrm{Mat}_{n,k}(\mathbb{R}), \quad \mathbf{\Psi} = [\psi_1, \dots, \psi_k] \in \mathrm{Mat}_{m,k}(\mathbb{R}) \tag{11}$$

such that

$$\langle \phi_i, \phi_j \rangle = \langle \psi_i, \psi_j \rangle = \delta_{ij}, \quad \mathbf{S} = \mathbf{\Phi\Sigma\Psi}^T, \quad \mathbf{\Sigma} := \mathit{Diag}(\sigma_1, \dots, \sigma_k) \in \mathrm{Mat}_{k,k}(\mathbb{R}) \tag{12}$$

and ϕ_i *are oriented spatial unit eigenvectors:*

$$\langle \mathbf{s}(\phi_i), \phi_i \rangle > 0 \tag{13}$$

with $\mathbf{s}(\phi_i)$ *defined by* (10)*. Such a decomposition is called an* oriented compact singular value decomposition.

Proof. First of all, any couple (ϕ, ψ) of spatial–time 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 unicity, so that there exists some unit spatial vectors $\phi = -\phi'$ associated to σ such that

$$\langle \mathbf{s}(\phi), \phi \rangle > 0 \text{ and } \langle \mathbf{s}(\phi'), \phi' \rangle > 0. \tag{14}$$

It is clear that $\mathbf{s}(\phi) = \mathbf{s}(\phi')$ so we have

$$\langle \mathbf{s}(\phi'), \phi' \rangle >= -\langle \mathbf{s}(\phi), \phi \rangle > 0 \tag{15}$$

which is a contradiction, so we can conclude. □

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

Algorithm 3.4 (Compact oriented SVD)**.** *Let* $\mathbf{S} \in \mathrm{Mat}_{n,m}(\mathbb{R})$ *of rank* k *such that all its non-zero singular values are distinct.*

(1) *Compute any compact SVD of* $\mathbf{S}$ *so that to obtain spatial unit vectors* $\phi_1, \dots, \phi_k$ *and time unit vectors* $\psi_1, \dots, \psi_k$*.*
(2) *Consider the column vectors* $\mathbf{s}_1, \dots, \mathbf{s}_m$ *of* $\mathbf{S}$*.*

(3) *Define*

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

with

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

(4) *Make*

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

3.2. **Space–time interpolation algorithm.** Take back parameters $\lambda_1 < \ldots < \lambda_N$, 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 oriented compact 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 than minimum rank of all matrices $\mathbf{S}^{(k)}$). Using compact oriented SVD given by Algorithm 3.4, we can consider POD of mode p on each matrix $\mathbf{S}^{(k)}$ and define

$$\mathbf{S}_p^{(k)} := \mathbf{\Phi}_k \mathbf{\Sigma}_k \mathbf{\Psi}_k^T \in \mathrm{Mat}_{n,m}(\mathbb{R}) \tag{16}$$

with $\mathbf{\Sigma}_k$ corresponding to singular values,

$$\mathbf{\Phi}_k := [\phi_1^{(k)}, \ldots, \phi_p^{(k)}] \in \mathcal{S}t_c^0(p, n), \quad \mathbf{\Psi}_k := [\psi_1^{(k)}, \ldots, \psi_p^{(k)}] \in \mathcal{S}t_c^0(p, m) \tag{17}$$

and $\phi_1^{(k)}, \ldots, \phi_p^{(k)}$ (resp. $\psi_1^{(k)}, \ldots, \psi_p^{(k)}$) correspond to spatial oriented eigenvectors (resp. time ones) of $\mathbf{S}^{(k)}$.

The goal is to define an *interpolated curve* on the reduced matrices $\mathbf{S}_p^{(k)}$, meaning some curve

$$\varphi \ : \ \lambda \in [\lambda_1, \lambda_N] \mapsto \varphi(\lambda) \in \mathrm{Mat}_{n,m}(\mathbb{R}) \text{ such that } \varphi(\lambda_k) = \mathbf{S}_p^{(k)} \tag{18}$$

and the space-time interpolation aim to make use of space and time eigenvectors.

First of all, using the target algorithm 2.11 for both space part $\mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_N$ and time part $\mathbf{\Psi}_1, \ldots, \mathbf{\Psi}_N$ leads to two curves

$$\lambda \mapsto \mathbf{\Phi}(\lambda), \quad \lambda \mapsto \mathbf{\Psi}(\lambda) \tag{19}$$

which are *not interpolated curves*, as we can not have for the space part $\mathbf{\Phi}(\lambda_k) = \mathbf{\Phi}_k$ (see Remark 2.12), and the same being observed for the time part.

Now, the use of a *mixed part* can leads us to have a well defined interpolated curve (18). To do this, let us first define

$$\widetilde{\mathbf{\Phi}}_k := \mathbf{Y}(\lambda_k), \quad \widetilde{\mathbf{\Psi}}_k := \mathbf{Y}(\lambda_k) \tag{20}$$

using (8) from the target algorithm 2.11 for the spatial part $\mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_N$ and for the time part $\mathbf{\Psi}_1, \ldots, \mathbf{\Psi}_N$. Finally, define for each parameter λ_k a *mixed matrix*

$$\mathbf{S}_k^* := \widetilde{\mathbf{\Phi}}_k^T \mathbf{S}_p^{(k)} \widetilde{\mathbf{\Psi}}_k \in \mathrm{Mat}_{p,p}(\mathbb{R}). \tag{21}$$

Then we have:

Lemma 3.5. *Let $\lambda \mapsto \Gamma(\lambda) \in \mathrm{Mat}_{p,p}(\mathbb{R})$ be any interpolated curve for the matrices $\mathbf{S}_k^*$, so that $s(\lambda_k) = \mathbf{S}_k^*$ for $k = 1, \dots, N$. Then, using the curves $\lambda \mapsto \boldsymbol{\Phi}(\lambda)$ and $\lambda \mapsto \boldsymbol{\Psi}(\lambda)$ from (19), the curve*

$$\kappa \, : \, \lambda \mapsto \boldsymbol{\Phi}(\lambda)\Gamma(\lambda)\boldsymbol{\Psi}(\lambda)^T \tag{22}$$

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

Proof. For any $k = 1, \dots, N$ we have

$$\kappa(\lambda_k) = \widetilde{\boldsymbol{\Phi}}_k \Gamma(\lambda_k) \widetilde{\boldsymbol{\Psi}}_k^T = \widetilde{\boldsymbol{\Phi}}_k \mathbf{S}_k^* \widetilde{\boldsymbol{\Psi}}_k^T \tag{23}$$

$$= \widetilde{\boldsymbol{\Phi}}_k \widetilde{\boldsymbol{\Phi}}_k^T \boldsymbol{\Phi}_k \boldsymbol{\Sigma}_k \boldsymbol{\Psi}_k^T \widetilde{\boldsymbol{\Psi}}_k \widetilde{\boldsymbol{\Psi}}_k^T \tag{24}$$

where $\widetilde{\boldsymbol{\Phi}}_k \widetilde{\boldsymbol{\Phi}}_k^T$ correspond to the projection matrix on the subspace $\mathbf{m}_k := \pi\left(\widetilde{\boldsymbol{\Phi}}_k\right) = \pi\left(\boldsymbol{\Phi}_k\right)$ (see Remark 2.12) so that

$$\widetilde{\boldsymbol{\Phi}}_k \widetilde{\boldsymbol{\Phi}}_k^T \boldsymbol{\Phi}_k = \boldsymbol{\Phi}_k \tag{25}$$

and the same being true for the time part, we can conclude. □

The space–time interpolation algorithm is now given by:

Algorithm 3.6 (Space–time interpolation)**.** *Under the genericity assumption, let $\mathbf{S}^{(1)}, \dots, \mathbf{S}^{(N)}$ be matrices in $\mathrm{Mat}_{n,m}(\mathbb{R})$ corresponding to parameters $\lambda_1 < \dots < \lambda_N$. Take a parameter $\widetilde{\lambda} \in [\lambda_1, \lambda_N]$.*

(1) *First consider the target algorithm 2.11 applied to the spatial parts $\boldsymbol{\Phi}_1, \dots, \boldsymbol{\Phi}_N$ and a reference point corresponding to $i_0 \in \{1, \dots, N\}$. For $\lambda_1, \dots, \lambda_N$ and $\widetilde{\lambda}$, use (8) to define*

$$\widetilde{\boldsymbol{\Phi}}_k := \mathbf{Y}(\lambda_k), \quad \widetilde{\boldsymbol{\Phi}} := \mathbf{Y}(\widetilde{\lambda}) \tag{26}$$

(2) *Apply the target algorithm 2.11 to the times parts $\boldsymbol{\Psi}_1, \dots, \boldsymbol{\Psi}_N$ and a reference point corresponding to the same i_0. For $\lambda_1, \dots, \lambda_N$ and $\widetilde{\lambda}$, use (8) to define*

$$\widetilde{\boldsymbol{\Psi}}_k := \mathbf{Y}(\lambda_k), \quad \widetilde{\boldsymbol{\Psi}} := \mathbf{Y}(\widetilde{\lambda}) \tag{27}$$

(3) *For each $k = 1, \dots, N$, define the square matrix*

$$\mathbf{S}_k^* := \widetilde{\boldsymbol{\Phi}}_k^T \mathbf{S}^{(k)} \widetilde{\boldsymbol{\Psi}}_k \in \mathrm{Mat}_{p,p}(\mathbb{R}) \tag{28}$$

(4) *Use standard interpolation on square matrices $\mathbf{S}_1^*, \dots \mathbf{S}_N^*$ to obtain square matrix $\mathbf{S}^*$, for instance:*

$$\mathbf{S}^* := \sum_{i=1}^{N} \prod_{i \neq j} \frac{\lambda - \lambda_j}{\lambda_i - \lambda_j} \mathbf{S}_i^* \tag{29}$$

(5) *Using spatial part* $\widetilde{\mathbf{\Phi}}$ *from* (26)*, time part* $\widetilde{\mathbf{\Psi}}$ *from* (27)*, and coupled part* $\mathbf{S}^*$ *from* (29)*, the interpolated snapshot matrix corresponding to* $\widetilde{\lambda}$ *is finally given by*

$$\widetilde{\mathbf{S}} := \widetilde{\mathbf{\Phi}}\mathbf{S}^*\widetilde{\mathbf{\Psi}}^T$$

4. Rigid-Viscoplastic FEM Formulation

The main defining characteristic of 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 of fluid flow problems and it is also called as *flow formulation* [23]. Although it is not possible to calculate the residual stresses and the spring-back effect, the flow formulation presents outstanding advantages. Unlike the elasto-plastic 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 [33, 13].

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:

$$\sigma'_{ij} = \frac{2}{3}\frac{\bar{\sigma}}{\dot{\bar{\varepsilon}}}\dot{\varepsilon}_{ij}, \qquad \dot{\bar{\varepsilon}} = \left(\frac{2}{3}\dot{\varepsilon}_{ij}\dot{\varepsilon}_{ij}\right)^{\frac{1}{2}} \tag{30}$$

(e) Incompressibility condition:

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

(f) Boundary conditions:

$$\begin{aligned} \mathbf{v} = \hat{\mathbf{v}} &\quad \text{on} \quad \partial\Omega_v \\ \mathbf{F} = \hat{\mathbf{F}} &\quad \text{on} \quad \partial\Omega_F \\ \text{friction and contact} &\quad \text{on} \quad \partial\Omega_c \end{aligned}$$

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 ˆ 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:

$$\partial\Omega = \partial\Omega_v \cup \partial\Omega_F \cup \partial\Omega_c \tag{31}$$

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

$$\Pi(v) := \int_\Omega \bar{\sigma}\dot{\bar{\varepsilon}} dV - \int_{\partial\Omega_F} F_i v_i dS \tag{32}$$

where the first term in (32) 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 Marcov-Hill [41] 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

$$\delta\Pi := \int_\Omega \bar{\sigma}\delta\dot{\bar{\varepsilon}} dV + \frac{1}{2}\int_\Omega K\dot{\varepsilon}_v\delta\dot{\varepsilon}_v dV - \int_{\partial\Omega_F} F_i\delta v_i dS = 0 \tag{33}$$

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 4.1. 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 new 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 structure remeshings for future investigation.

4.3. **Discretization and iteration.** The discretization of the functional follows the standard procedure of the finite element method. Eq. (33) is expressed in terms of nodal point velocities v_i and their variations δv_i. Using the variational principle

$$\delta \Pi = \sum_{m=1}^{M} \frac{\partial \Pi^{(m)}}{\partial v_i} \delta v_i = 0, \qquad i = 1, 2, ..., 2N_s, \tag{34}$$

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 arbitrariness of δv_i, a set of algebraic equations (stiffness equations) are obtained

$$\frac{\partial \Pi}{\partial v_i} = \sum_{m=1}^{M} \frac{\partial \Pi^{(m)}}{\partial v_i} = 0. \tag{35}$$

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

$$\left. \frac{\partial \Pi}{\partial v_i} \right|_{\mathbf{v}=\mathbf{v}_0} + \left. \frac{\partial^2 \Pi}{\partial v_i \partial v_j} \right|_{\mathbf{v}=\mathbf{v}_0} \Delta v_j = 0 \tag{36}$$

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 (36) with respect to Δv_j, the assumed velocity field is updated by the form (written in vector notation)

$$\mathbf{v}^{(i)} = \mathbf{v}^{(i-1)} + \alpha (\Delta \mathbf{v})^{(i)} \tag{37}$$

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

$$\frac{\| \Delta \mathbf{v} \|_{L_2}}{\| \mathbf{v} \|_{L_2}} \leq e_1, \qquad \left\| \frac{\partial \Pi}{\partial \mathbf{v}} \right\|_{L_2} \leq e_2 \tag{38}$$

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

$$-\rho c \frac{\partial T}{\partial t} + k\nabla^2 T + \xi \bar{\sigma} \dot{\bar{\varepsilon}} = 0 \tag{39}$$

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 know as the Taylor-Quinney coefficient.

In a weak form, and using the divergence theorem

$$-\int_\Omega \xi \bar{\sigma} \dot{\bar{\varepsilon}} \delta T dV + \int_\Omega k \nabla T \delta(\nabla T) dV + \int_\Omega \rho c \frac{\vartheta T}{\vartheta t} \delta T dV - \int_{\partial\Omega} q_n \delta T dS = 0 \tag{40}$$

where

$$q_n := k \frac{\partial T}{\partial n} \tag{41}$$

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. [51], it can be seen that heat balance equations such as (40), upon finite element discretization are reduced to the form:

$$\boldsymbol{C}\dot{\boldsymbol{T}} + \boldsymbol{K}\boldsymbol{T} = \boldsymbol{Q} \tag{42}$$

where $\boldsymbol{C}$ is the heat capacity matrix, $\boldsymbol{K}$ is 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 (42) can be found in numerical analysis books [45],[14]. It suffices to say that one-step time integration is used. Convergence of a scheme requires consistency and stability. Consistency is satisfied by a general time integration scheme

$$^{t+\Delta t}\boldsymbol{T} =^t \boldsymbol{T} + \Delta t[(1-\theta)^t\dot{\boldsymbol{T}} + \theta^{t+\Delta t}\dot{\boldsymbol{T}}] \tag{43}$$

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 4.2. *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 for 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, the state of the system is advanced by sequentially executing and exchange information between these two solvers [16]. The equations for the mechanical analysis and the temperature calculation are *strongly coupled*, thereby making necessary the simultaneous solution of the finite element counterparts [33, 46, 47].

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 die is set to $v = 1$ mm/s while the lower die is stationary. 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 1). In our calculations we employ a conventional rate-dependent power law to describe the material flow stress equation

$$\bar{\sigma}(\dot{\bar{\varepsilon}}) = 1000\dot{\bar{\varepsilon}}^{0.1} \qquad \text{(MPa)} \tag{44}$$

The solution convergence is assumed when the velocity error norm and the force error norm (38) becomes less than 10^{-6}. The type of element used is the linear isoparametric rectangular element with four point integration. However, one point integration is used for the dilatation term, the second integral of the functional in (33). This is known as the reduced integration scheme which imposes the volume constancy averaged over the linear rectangular element. The computational grid composed of 100 elements interconnected at $N_s = 121$ nodes with 2 degrees of freedom, resulting in a global stiffness matrix of size 242×242. For the rigid-viscoplastic analysis the limiting strain rate $\dot{\bar{\varepsilon}}_0$ is chosen to be 0.01 and the penalty constant or bulk modulus K is set to 10^5.

Among the various models of friction, the one of Chen and Kobayashi [12] is adapted to model the sliding contact at the tool-workpiece interface. This model allows the variation of the tangential traction with the relative velocity at the tool-workpiece interface

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

where $\mathbf{v}_s$ is the relative velocity in the tangential direction between the tool and the workpiece, and v_0 is a positive constant several orders of magnitude smaller than $\mathbf{v}_s$; m is the friction factor $(0 < m < 1)$ and k is the material shear yield stress $k = \bar{\sigma}/\sqrt{3}$. For the compression tests considered here, the relative tangential velocity at the tool-workpiece interface at the beginning of deformation is zero. The present analysis assumes that the friction factor remains constant throughout compression. Investigations reported on frictional shear stress measurements over the interface between a cylindrical workpiece and a die during plastic compression suggest that a constant frictional shear stress is reasonably justified for certain conditions of lubrication [8]. The basic characteristics of algorithms used in the RVP FEM analysis are summarized in Table 1.

Basic characteristics of algorithms	
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 5.1. Note that during the course of the simulation we avoid remeshing of the workpiece. As discussed in [48], 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, the following training points are selected, denoted by the variable $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$. 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 1.

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 $\mathbf{S}^{(i)} \in \mathbb{R}^{2N_s \times N_t}$ of size 242 × 7, corresponding to parameter λ_i, are associated to the nodal velocity field in x and y directions. Using the POD/SVD, the ROM spatial part $\{\mathbf{m}_i\}_{i=1}^N$ in $\mathcal{G}(p, 2N_s)$ and time part $\{\mathbf{m}'_i\}_{i=1}^N$ in $\mathcal{G}(p, N_t)$ is constructed for the training points λ_i.

For validating the parametric Space-Time interpolation, the target point snapshot matrix $\widetilde{\mathbf{S}}$ is computed via Algorithm 3.6. ST POD interpolation is finally 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 $\mathbf{S}^{\text{FEM}}$, respectively, the L_2-error measure is defined as

$$e_{L_2}(\widetilde{\mathbf{s}}_i) := \frac{\|\widetilde{\mathbf{s}}_i - \mathbf{s}_i^{\text{FEM}}\|_{L_2}}{\|\mathbf{s}_i^{\text{FEM}}\|_{L_2}}, \quad i = 1, \ldots, N_t. \tag{45}$$

Additionally, the relative Frobenius error norm is used

$$e_F(\widetilde{\mathbf{S}}) := \|\widetilde{\mathbf{S}} - \mathbf{S}^{\text{FEM}}\|_F / \|\mathbf{S}^{\text{FEM}}\|_F \tag{46}$$

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 2. 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 3. 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 (45)) between the interpolated and the HF-FEM solution for various POD modes is displayed in Figure 4. As the reference point for attaching the tangent space, the subspaces $\mathbf{m}_0(\lambda = 0.5)$ and $\mathbf{m}'_0(\lambda = 0.5)$ are chosen for the spatial and time parts. In general, the relative error for all POD modes lies 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 5.2. In the case of using $p = 7$ POD modes for the time basis interpolation, the Grassmannian manifold $\mathcal{G}(p, p)$ *reduces to one point*, so it is not relevant to use the target Algorithm 2.11: any new parameter will give rise to the same matrix $\mathbf{\Psi}_{i_0}$ 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 high-fidelity FEM solutions, respectively, at the time increments during the deformation. Figure 5 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 with respect to the number of POD modes. 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 6). 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 asses 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 $\mathbf{m}_0(\lambda = 0.5)$ (resp. $\mathbf{m}'_0(\lambda = 0.5)$) as the reference point on the Grassmann manifold. The relative L_2-error norm $e_{L_2}(\widetilde{\mathbf{s}}_i)$ for various POD modes corresponding to target points $\widetilde{\lambda} = 0.8$ is shown in Figure 7. Again, one can observe that the error lies between the values 0.014 and 0.026, thus the interpolation is stable within the parametric range.

Remark 5.3. We need to know what is the optimal choice in the sense of interpolation quality related to the Riemannian distances (5) between the training points and the applied interpolation method (Lagrange in our case). Or in other words, how the choice of the local chart, i.e., of the reference point $\mathbf{m}_0$ on the Grassmannian, and the distances between it and the other training points affects interpolation accuracy. And furthermore, what is the size of the diameter of a 'small' neighborhood around $\mathbf{m}_0$? How do we arrange (sample) the training points on the Grassmann manifolds without any a priori knowledge of the underlying system dynamics? It would suffice to simply select uniform grids in case of multi-parametric problems? To the best of the authors knowledge, we cannot be quite explicit about how all the above can be a priori chosen.

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 $\mathbf{S}^{(i)} \in \mathbb{R}^{N_s \times N_t}$ of size 121 × 7, corresponding to λ_i, are associated to nodal temperatures. Using the POD/SVD, the ROM spatial basis $\{\mathbf{m}_i\}_{i=1}^{N}$ and time basis $\{\mathbf{m}'_i\}_{i=1}^{N}$ are constructed for the training points $\lambda_i \in \Lambda_t$. The Space-Time interpolation (see Algorithm 3.6) is finally compared against the high-fidelity FEM solution.

Figure 8 presents the temperature profiles at the final compression state obtained using different values of the shear friction factor m (represented by parameter λ). 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.

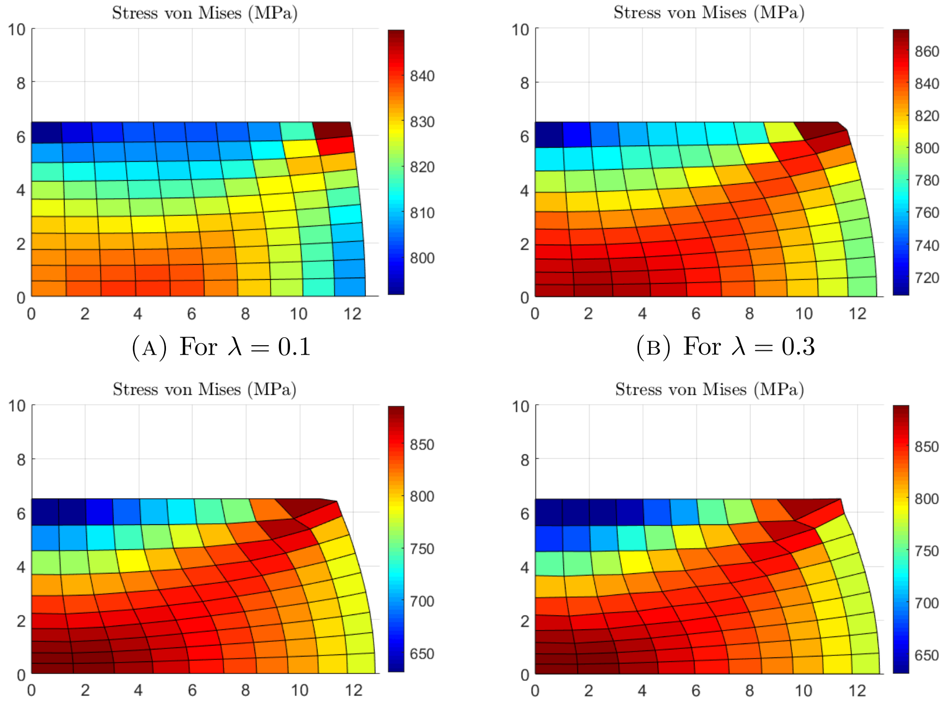
The eigenvalue spectrum of snapshot matrices $\mathbf{S}^{(i)}$ corresponding to training points $\lambda_i \in \Lambda_t$ is shown in a semi-log scale in Figure 9. We can observe that the distance between the first and the last eigenvalue of the curves is of the order of 5 up to 6 orders of magnitude. 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 10. Most of the 99.9% of the total energy is contained by the first two POD modes.

The relative L_2-error norm $e_{L_2}(\widetilde{\mathbf{s}}_i)$ (45) between the interpolated and the HF-FEM solution for various POD modes is shown in Figure 11. Additionally, the Frobenius relative error norm (46) for the POD modes is presented in Figure 12. 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 13 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.

6. Conclusions

A novel non-intrusive Space-Time POD basis interpolation from compact Stiefel manifolds is developed and applied to parametric high nonlinear metal forming problems. Apart from the separate interpolation of POD spatial and time basis on associated Grassmannian manifolds, an interpolation function is defined on a set of parametric snapshot matrices. This function results from curves defined on compact Stiefel manifold for the space and the time 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 time basis for the construction of the target ROM snapshot matrix. The method is tested using a coupled thermomechanical rigid-viscoplastic FEM formulation which has been successfully utilized in the manufacturing industry in a variety of applications. Numerical investigations considered the reconstruction of the ROM snapshot matrices of the velocity and temperature fields. The error norm associated to the high-fidelity FEM solutions is within an acceptable low range, thereby highlighting the potential of the proposed Space-Time interpolation for robust near real-time predictions.



(A) For $\lambda = 0.1$

(B) For $\lambda = 0.3$

(C) For $\lambda = 0.5$

(D) For $\lambda = 0.9$

FIGURE 1. Deformation patterns of the benchmark metal forming example using different values for the shear friction factor m represented by the parameter λ.

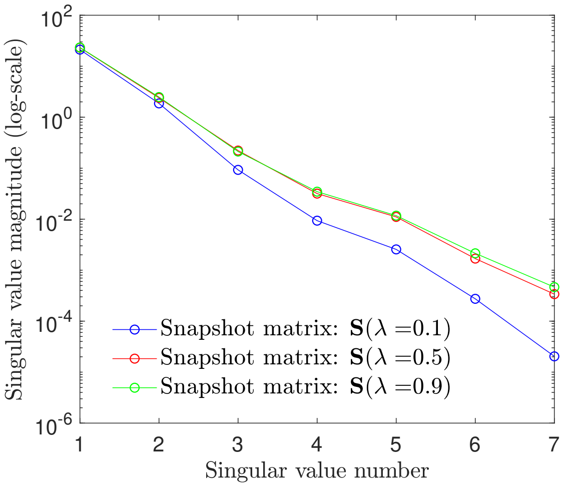


FIGURE 2. The eigenvalue spectrum of snapshot matrices $\mathbf{S}_i$ corresponding to training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$.

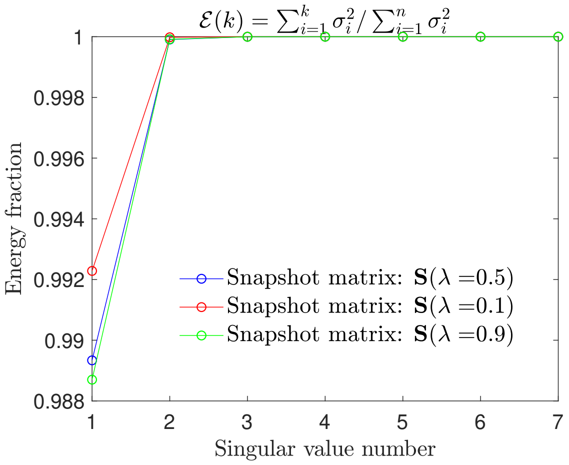


FIGURE 3. Energy captured by the singular values of snapshot matrices $\mathbf{S}_i$ corresponding to training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$.

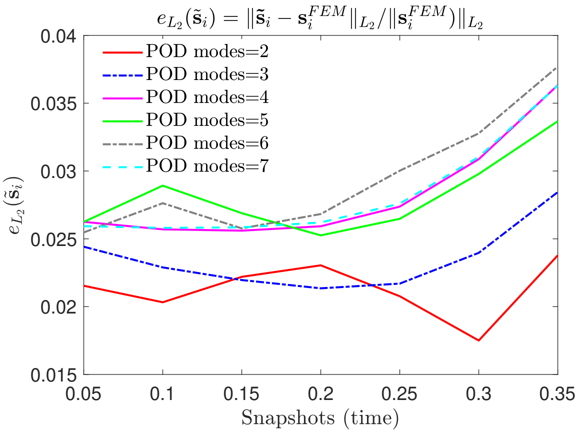


Figure 4. Performance of the ST ROM using the relative L_2-error norm $e_{L_2}(\widetilde{\mathbf{s}}_i)$ for various POD modes; training points: $\mathbf{m}_0(\lambda = 0.5)$ (reference point); $\mathbf{m}_1(\lambda = 0.1)$; $\mathbf{m}_2(\lambda = 0.9)$; target point: $\widetilde{\mathbf{m}}(\lambda = 0.3)$ (resp. $\mathbf{m}_i$ and $\widetilde{\mathbf{m}}'$ for the time ones).

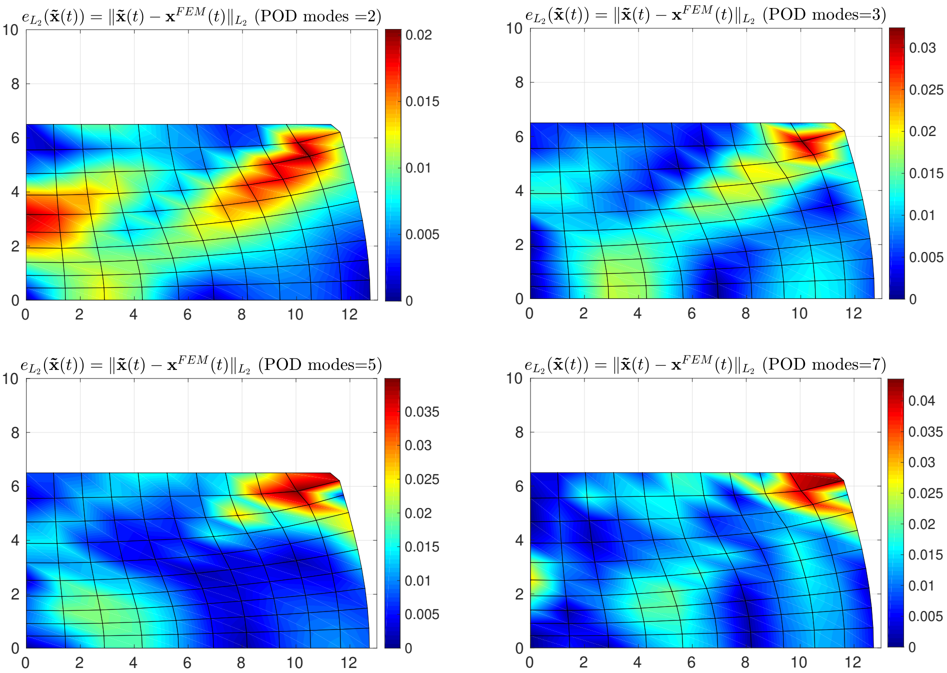


FIGURE 5. The position vector error $e_{L_2}(\widetilde{\mathbf{x}}(t)) = \|\widetilde{\mathbf{x}}(t) - \mathbf{x}^{\mathrm{FEM}}(t)\|_{L_2}$ of the nodal points at the final deformation state $t = 0.35$ s superimposed on the high-fidelity FEM solution; POD modes $p = \{2, 3, 5, 7\}$; target point: $\widetilde{\mathbf{m}}(\lambda = 0.3)$ (resp. $\widetilde{\mathbf{m}}'$ for the time one).

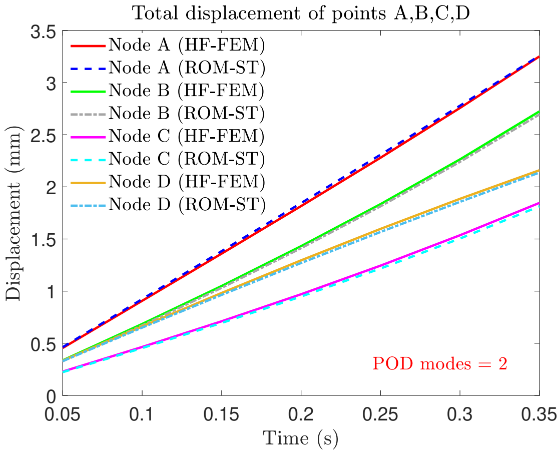


Figure 6. ST POD interpolation; comparison of the total displacement of selected nodes against the high-fidelity FEM solution; training points: $\mathbf{m}_0(\lambda = 0.5)$ (reference point); $\mathbf{m}_1(\lambda = 0.1)$; $\mathbf{m}_2(\lambda = 0.9)$; target point: $\widetilde{\mathbf{m}}(\lambda = 0.3)$ (resp. $\mathbf{m}'_i$ and $\widetilde{\mathbf{m}}'$ for the time ones); POD modes $p = 2$.

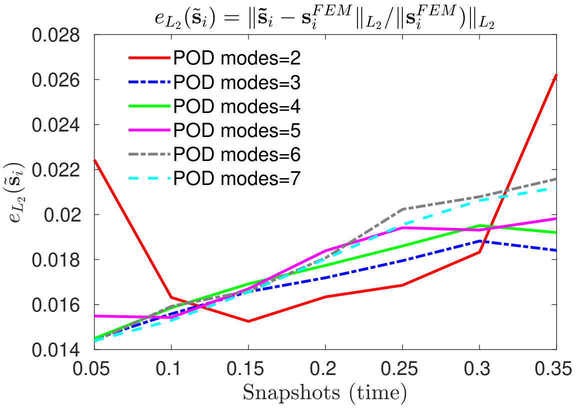
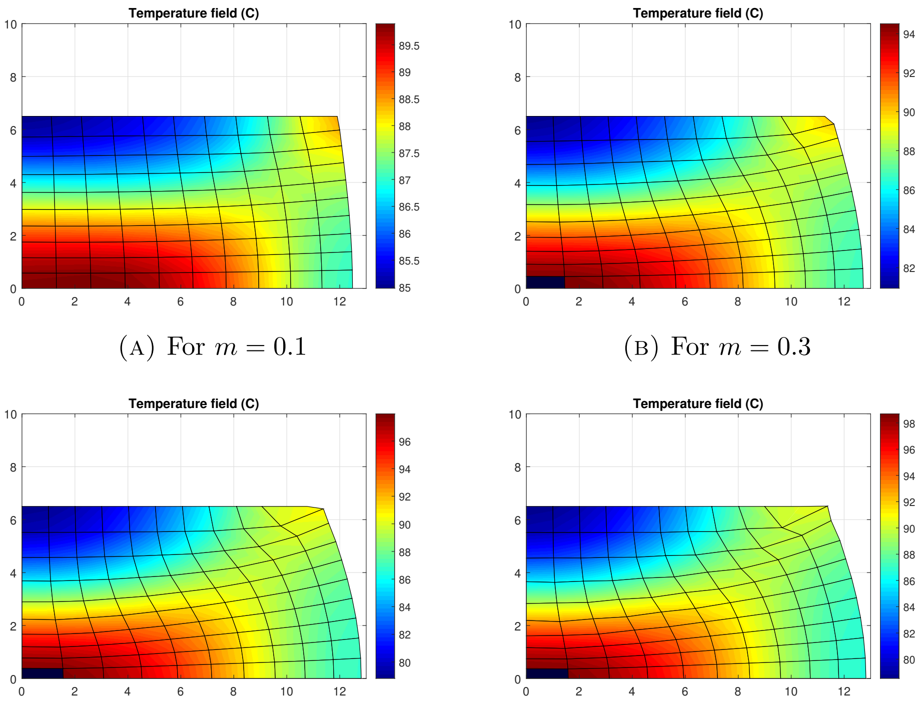


Figure 7. Performance of the ST POD using the relative L_2-error norm $e_{L_2}(\widetilde{\mathbf{s}}_i)$ for various POD modes; training points: $\mathbf{m}_0(\lambda = 0.5)$ (reference point); $\mathbf{m}_1(\lambda = 0.1)$; $\mathbf{m}_2(\lambda = 0.9)$; target point: $\widetilde{\mathbf{m}}(\lambda = 0.8)$ (resp. $\mathbf{m}'_i$ and $\widetilde{\mathbf{m}}'$ for the time ones).



(A) For $m = 0.1$

(B) For $m = 0.3$

(C) For $m = 0.5$

(D) For $m = 0.9$

FIGURE 8. Temperature profiles at the final compression state obtained using different values of the shear friction factor m represented by parameter λ.

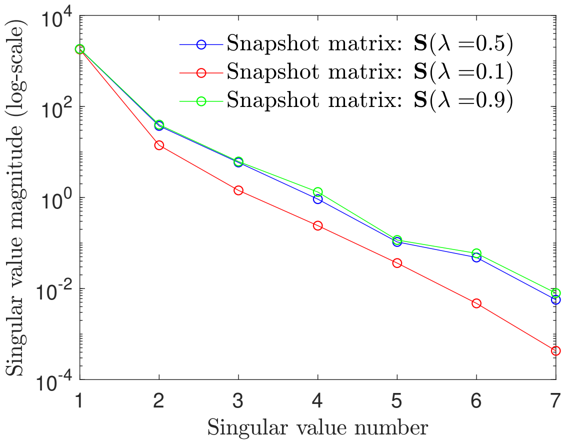


Figure 9. The eigenvalue spectrum of snapshot matrices $\mathbf{S}_i$ corresponding to training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$.

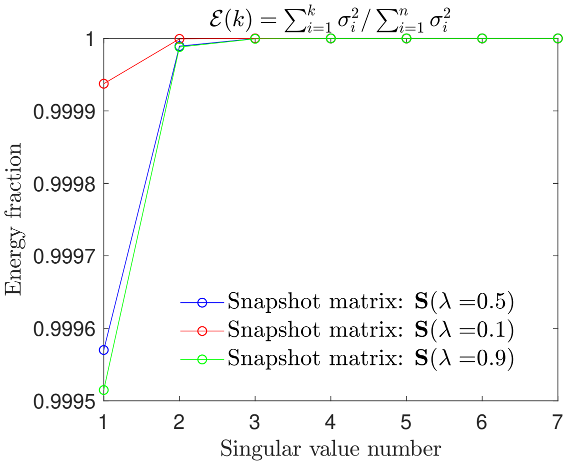


Figure 10. Energy captured by the singular values of snapshot matrices $\mathbf{S}_i$ corresponding to training points $\lambda \in \Lambda_t = \{0.1, 0.5, 0.9\}$.

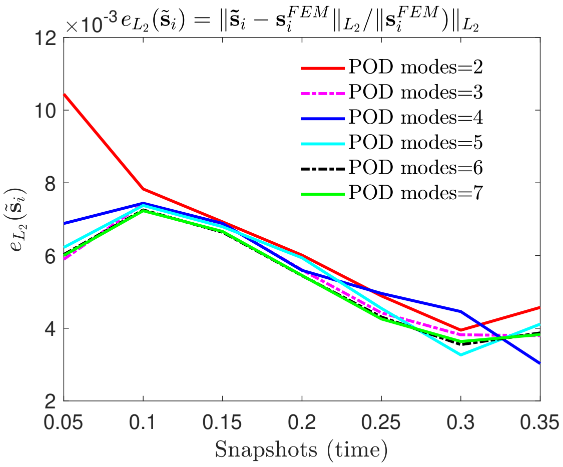


FIGURE 11. Performance of the ST ROM using the relative L_2-error norm $e_{L_2}(\widetilde{\mathbf{s}}_i)$ for various POD modes; training points: $\mathbf{m}_0(\lambda = 0.5)$ (reference point); $\mathbf{m}_1(\lambda = 0.1)$; $\mathbf{m}_2(\lambda = 0.9)$; target point: $\widetilde{\mathbf{m}}(\lambda = 0.3)$ (resp. $\mathbf{m}'_i$ and $\widetilde{\mathbf{m}}'$ for the time ones).

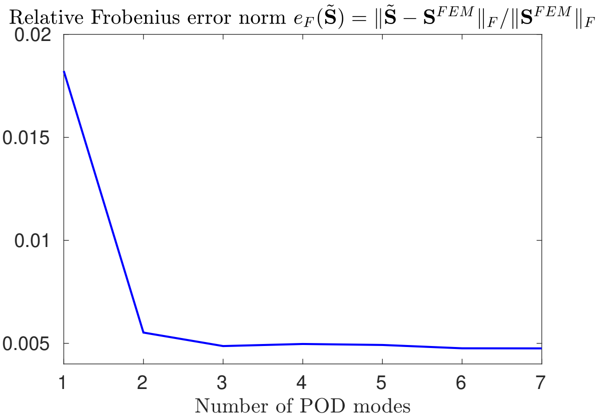


FIGURE 12. Performance of the ST ROM using the relative Frobenius error norm $e_F(\widetilde{\mathbf{S}})$ against the number of POD vectors; training points: $\mathbf{m}_0(\lambda = 0.5)$ (reference point); $\mathbf{m}_1(\lambda = 0.1)$; $\mathbf{m}_2(\lambda = 0.9)$; target point: $\widetilde{\mathbf{m}}(\lambda = 0.3)$ (resp. $\mathbf{m}'_i$ and $\widetilde{\mathbf{m}}'$ for the time ones).

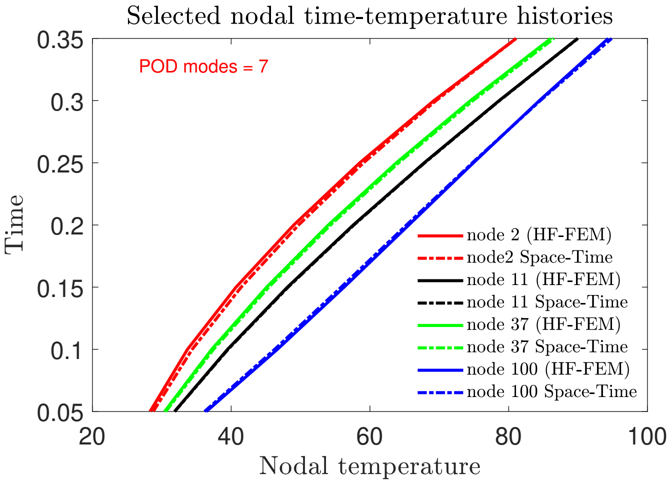


FIGURE 13. ST POD interpolation (Lagrange); Temperature evolution of selected nodal points validated against the high-fidelity FEM solution; ST POD and HF-FEM solutions virtually coincide; training points: $\mathbf{m}_0(\lambda = 0.5)$ (reference point); $\mathbf{m}_1(\lambda = 0.1)$; $\mathbf{m}_2(\lambda = 0.9)$; target point: $\widetilde{\mathbf{m}}(\lambda = 0.3)$; (resp. $\mathbf{m}'_i$ and $\widetilde{\mathbf{m}}'$ for the time ones); POD modes $p = 7$.

References

[1] S. Andrietti, J.L. Chenot, M. Bernacki, P.O. Bouchard, L. Fourment, E. Hachem, E. Perchat, Recent and future developments in finite element metal forming simulation, Computer Methods in Materials Science, 15, 2, (2015), 265-293.

[2] H. Abdi, L.J. Williams, Principal component analysis, Wiley Interdisciplinary Reviews: Computational Statistics, 2.4, (2010), 433-459.

[3] P-A Absil, Robert Mahony, Rodolphe Sepulchre, Riemannian geometry of Grassmann manifolds with a view on algorithmic computation, Acta Applicandae Mathematica, 80.2, (2004), 199-220.

[4] D. Amsallem, J. Cortial, K. Carlberg, C. Farhat, A method for interpolating on manifolds structural dynamics reduced-order models, International Journal of Numerical Methods in Engineering, 80,9, (2009), 1241-125.

[5] N. Aubry, On the hidden beauty of the proper orthogonal decomposition, Theoretical and Computational Fluid Dynamics, 2, (1991), 339-352.

[6] Audouze C, De Vuyst F, Nair PB, Reduced-order modeling of parameterized PDEs using time-space-parameter principal component analysis, International journal for numerical methods in engineering, (2009), 80(8):1025-1057.

[7] Audouze C, De Vuyst F, Nair PB, Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations, Numerical Methods for Partial Differential Equations, (2013), 29(5):1587-1628.

[8] G.T. van Rooyen, W.A. Backofen, International Journal of Mechanical Sciences, 8, (1966), 731.

[9] Björck Åke, Gene H. Golub, Numerical methods for computing angles between linear subspaces, Mathematics of computation 27.123, (1973), 579-594.

[10] Boothby William M, An introduction to differentiable manifolds and Riemannian geometry, (Academic press, 1986).

[11] R. Bro, E. Acar, T.Kolda, Resolving the Sign Ambiguity in the Singular Value Decomposition, Sandia Report, SAND2007-6422, (2007).

[12] C.C. Chen, S. Kobayashi, Rigid plastic finite element analysis of ring compression, Applications of Numerical Methods to Forming Processes, ASME, 28, (1978), 163-174.

[13] J.L. Chenot, Recent contributions to the finite element modelling of metal forming processes, Journal of Materials Processing Technology, 34(1-4), (1992), 9-18.

[14] G. Dahlquist, A. Bjorck, Numerical Methods, Prentice-Hall, (New Jersey, 1974).

[15] Alan Edelman, Tomas A Arias, Steven T Smith, The geometry of algorithms with orthogonality constraints, SIAM journal on Matrix Analysis and Applications, 20,2, (1998), 303-353.

[16] C. Felippa, K. Park, Staggered transient analysis procedures for coupled mechanical systems: Formulation, Computer Methods in Applied Mechanics and Engineering, 24, (1), (1980), 61-111.

[17] Z.O. Feng, G. De Saxcé, Rigid-plastic implicit integration scheme for analysis of metal forming, Eur. J. Mech., A/Solids, 15, 1, (1996), 51-66.

[18] O.C. Zienkiewicz, Flow formulation for numerical solution of metal forming processes. In Numerical analyses of forming processes. Ed. J.F.T. Pittman, O.C.Zienkiewicz, R.D.Wood and J.M.Alexander, (1984), 1-44.

[19] O. Friderikos, Two-Dimensional Rigid-Plastic FEM Simulation of Metal Forming Processes in MATLAB, Proceedings of the 4th International Conference on Manufacturing and Materials Engineering (ICMMEN), 3-5 October, Thessaloniki, Greece, (2011).

[20] Gallot, S., Hulin, D., & Lafontaine J. Riemannian geometry (Vol. 2). (Springer-Verlag, Berlin, 1990).

[21] Gene H Golub, CFV Loan, Matrix Computations, (3rd edn., vol. 1, 1996).

[22] Z. Gronostajski, Z. Pater, L. Madej et al., Recent development trends in metal forming, Archives of Civil and Mechanical Engineering, 19(3), (2019), 898-941.

[23] R. Hill, The Mathematical Theory of Plasticity, (Oxford University Press, London, 1971).

[24] Philip Holmes et al., Turbulence, coherent structures, dynamical systems and symmetry, (Cambridge university press, 2012).

[25] Thibault Henri, Jean-Pierre Yvon, Convergence estimates of POD-Galerkin methods for parabolic problems, IFIP Conference on System Modeling and Optimization, Springer, (2003), 295-306.

[26] Jackson JE, Principal components and factor analysis: part I - principal components, Journal of Quality Technology, 12, (1980), 201-213.

[27] Jackson JE, Principal components and factor analysis: part II - additional topics related to principal components, Journal of Quality Technology, 13, (1981), 46-58.

[28] I.T. Jolliffe, Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed., NY: Springer, (2002), p. 487.

[29] Jordan Camille, Essai sur la géométrie à n dimensions, Bulletin de la Société mathématique de France 3, (1875), 103-174.

[30] M. Joyner, Comparison of two techniques for implementing the Proper orthogonal decomposition method in damage detection problems, Mathematical and Computer Modelling, 40, (2004), 553-571.

[31] Kari Karhunen, Zur spektraltheorie stochastischer prozesse, Ann.Acad. Sci. Fennicae, AI34, (1946).

[32] S. Kobayashi, Rigid-plastic finite element analysis of axisymmetric metal forming processes, R.F. Jones, Jr. et al., eds., Numerical Modeling of Manuf. Process (ASME, New York, 1977), (1973), 49-65.

[33] S. Kobayashi, SI Oh, T. Altan, Metal Forming and the Finite Element Method. (Oxford University Press, 1989).

[34] Kozlov Sergey Emel'yanovich, Geometry of the real Grassmannian manifolds, Parts I, II, Zapiski Nauchnykh Seminarov POMI 246, (1997), 84-107.

[35] Lafontaine J., An introduction to differential manifolds (Springer, 2015).

[36] Lee John M., Smooth manifolds, Introduction to Smooth Manifolds (Springer, New York, 2013).

[37] Lee J. M., Introduction to Riemannian manifolds (Vol. 2, Springer, 2018).

[38] C.H. Lee, S. Kobayashi, New solutions to rigid-plastic deformation problems using matrix method, Transactions of ASME 7, Journal of Engineering for Industry, 95, (1973), 865-873.

[39] M. Loève, Probability theory, Vol. II, Graduate Texts in Mathematics, Vol. 46, Springer, (1978).

[40] Y. Lu, N. Blal, A. Gravouil, Space–time POD based computational vademecums for parametric studies: application to thermo-mechanical problems, Advanced Modeling and Simulations in Engineering Sciences, 5:3, (2018).

[41] A.A. Markov, On variational principles on theory of plasticity, Prik. Mat. Mekh., 11, (1947), 339-350.

[42] Rolando Mosquera Meza, Interpolation sur les variétés grassmanniennes et applications à la réduction de modèles en mécanique, PhD thesis, La Rochelle, (2018).

[43] Rolando Mosquera, Aziz Hamdouni, Abdallah El Hamidi, Cyrille Alleryet, POD basis interpolation via Inverse Distance Weighting on Grassmann manifolds, Discrete & Continuous Dynamical Systems, Series S, 12, 6, (2018), 1743-1759.

[44] G Muhlbach, The General Neville-Aitken-Algorithm and Some Applications, Numerische Mathematik, 31, (1978-79), 97-110.

[45] A. Ralston, A first course in numerical analysis (McGraw-Hill, New York, 1965).

[46] N. Rebelo, S. Kobayashi, A coupled analysis of viscoplastic deformation and heat transfer—I. Theoretical considerations, Int. J. Mech. Sci., 22, (1980), 699-705.

[47] N. Rebelo, S. Kobayashi, A Coupled Analysis of Viscoplastic Deformation and Heat Transfer-II. Applications, Int. J. Mech. Sci. 22, (1980), 707-718.

[48] David Ryckelynck, Djamel Missoum-Benziane, Multi-level A Priori Hyper-Reduction of mechanical models involving internal variables, Computer Methods in Applied Mechanics and Engineering, Elsevier, 199, (2010), 1134-1142.

[49] V. Shinde, E. Longatte, F. Baj, Y. Hoarau, and M. Braza, A galerkin-free model reduction approach for the Navier-Stokes equations, Journal of Computational Physics, 309, (2016), 148-163.

[50] O.C. Zienkiewicz, P.O. Godbole, Flow of plastic and visco-plastic solids with special reference to extrusion and forming processes, International Journal for Numerical Methods in Engineering, 8, (1974), 3-16.

[51] O.C. Zienkiewicz, The Finite Element Method (3rd Edn. McGraw-Hill, New York, 1977).

[52] Yung-Chow Wong, Differential geometry of Grassmann manifolds, Proceedings of the National Academy of Sciences of the United States of America, 57.3, (1967), 589.

(Orestis Friderikos) Université Paris-Saclay, ENS Paris-Saclay, CNRS, LMT - Laboratoire de Mécanique et Technologie, 91190, Gif-sur-Yvette, France
Email address: ofriderikos@ihu.gr

(Marc Olive) Université Paris-Saclay, ENS Paris-Saclay, CNRS, LMT - Laboratoire de Mécanique et Technologie, 91190, Gif-sur-Yvette, France
Email address: marc.olive@math.cnrs.fr

(Emmanuel Baranger) Université Paris-Saclay, ENS Paris-Saclay, CNRS, LMT - Laboratoire de Mécanique et Technologie, 91190, Gif-sur-Yvette, France
Email address: emmanuel.baranger@ens-paris-saclay.fr

(Dimitris Sagris) Mechanical Engineering Department, Laboratory of Manufacturing Technology & Machine Tools, International Hellenic University, GR-62124 Serres Campus, Greece.
Email address: dsagris@ihu.gr

(Constantine David) Mechanical Engineering Department, Laboratory of Manufacturing Technology & Machine Tools, International Hellenic University, GR-62124 Serres Campus, Greece.
Email address: david@ihu.gr