Computational Mechanics (2021) 68:1407–1419
https://doi.org/10.1007/s00466-021-02075-5

ORIGINAL PAPER

A pruning algorithm preserving modeling capabilities for polycrystalline data

Harris Farooq[1,2] · David Ryckelynck[1] · Samuel Forest[1] · Georges Cailletaud[1] · Aldo Marano[1]

Received: 21 January 2021 / Accepted: 3 August 2021 / Published online: 25 September 2021

Abstract
We are exploring the idea of data pruning via hyperreduction modeling. The main novelty of this paper is a lossy data compression/decompression approach for ploycrystalline data, which is based on a hyperreduction scheme that preserves data driven modeling capabilities after compression. We assume to know a mechanical model whose equations are satisfied by the data. It is shown that the proposed reconstruction of the data performs an oblique projection of selected original data. This is achieved by the solution of reduced mechanical equations. High resolution crystal plasticity finite element simulations demand computational and storage resources that are unusual, especially in cases where hundreds of grains are interacting under cyclic loading. The development of image-based modeling via computed tomography highlights the problem of long-term storage of simulation data by using data pruning. The present paper focuses on modeling cyclic strain-ratcheting as an example of numerical modeling that the proposed algorithm preserves. The size of the remaining sampled data can be user-defined, depending on the needs concerning storage space. The relevance of the pruned data is tested afterwards for statistics on the predicted strain, as if full finite element data were available. The proposed method is compared to the Gappy POD method, when no additional modeling step is expected after data pruning.

Keywords Model order reduction · CPFEM · Ratcheting · Data compression · Pruning algorithm · Material database

1 Introduction

With the development and the generalization of Computed Tomography (CT), the volume of data acquired has drastically increased. This raises new challenges, such as data storage, data mining or the development of relevant experiments-simulations dialogue methods such as model calibration and model validation. When material microstructures are under consideration, continuum models can be an attractive approach to assess mechanical properties. Coupling micro-mechanical laws with complex geometries has in present times gained massive popularity [22,23]. This is followed by a numerical procedure to solve the boundary value problem. One of the most popular methods to discretize the weak form of balance equations is the Finite Element method, which is also employed in this article. Using this method, the constitutive equations can be integrated at each material point and insight can be gained as to what is happening locally at certain geometric locations for various parameter variations. In this context, of particular interest are strain-ratcheting simulations for polycrystalline materials, where plasticity accumulates at certain locations during cycling loading. Analyzing such a problem is crucial for lifetime assessment of engineering components. This requires considerable computational and storage resources for direct numerical simulations. The current state of computer science is considerably advanced, with numerous parallel computing strategies available [47,53]. But, full resolution simulations are now related to massive microstructural data. As a result,

✉ David Ryckelynck
david.ryckelynck@mines-paristech.fr

Harris Farooq
hf354@cam.ac.uk

Samuel Forest
samuel.forest@mines-paristech.fr

Georges Cailletaud
georges.cailletaud@mines-paristech.fr

Aldo Marano
aldo.marano@mines-paristech.fr

[1] MAT-Centre des matériaux, CNRS UMR 7633, MINES ParisTech, PSL University, BP 87, 91003 Evry, France

[2] Present Address: Deparment of Materials Science and Metallurgy, University of Cambridge, Cambridge, UK

finite element simulations using X-ray CT data or synthetic microstructures [45] lead to an explosion of the volume of data to be stored.

Recently, databases of simulation data have demonstrated their viability and computational advantages for facilitating crystal plasticity predictions [1]. Obviously, much more can be expected from a database for crystal plasticity. It should foster: reproducibility as in other scientific domains [2], the training of artificial neural networks [18] or Bayesian networks [40], the exploration of new physical parameters or new assumptions, the detailed comparison between various materials, and additional model updating [25,46] when new observational data are available. Therefore, the long-term storage of a database in crystal plasticity is nowadays an issue. Data compression/decompression schemes are already available in the framework of high performance computing [33]. Lossy compression allows the precision of the data to be reduced in a way that has an insignificant impact on the data. In [33], when applying their proposed compression scheme to data produced by the CHIME experiment, the data are compressed to 28% of their original size.

The present paper aims to explore the idea of data pruning via numerical modeling in order to release more storage space when needed, as a complementary approach to usual data compression techniques. The main novelty of this paper is a lossy data compression/decompression approach based on a hyperreduction scheme that preserves data driven modeling capabilities after compression of data in crystal plasticity. Models are designed to incorporate the relevant information contained in data by using less storage space. In a sense they compress these data, and model predictions enable their reconstruction. But models may have a very complex reconstruction algorithm. In this paper we consider hyperreduced order modeling, with low computational complexity, for the decompression task. An additional novelty of this paper, is the proof that the displacements computed via hyperreduction fulfill the equation of an oblique projection of the original data. The method is based on feature extraction from data, which is an other way for lossy compression. It can be performed via clustering [30], linear dimensionality reduction or via autoencoders [18]. In [25], data pruning via linear dimensionality reduction and hyperreduction is shown to preserve simulation capabilities for sand materials, while reducing the storage resources. In the present paper, this data pruning strategy is evaluated for polycrystalline materials (Face Centered Cubic materials in the present case for instance). The pruning algorithm can be supplemented by usual data compression. The decompression step fosters extrapolation of pruned data and preserves modeling capabilities after data reduction. In this paper the input data are: (i) a user-defined region of interest, (ii) simulation data related to 10 loading cycles applied to a finite element model of a polycrystalline aggregate, (iii) hyperreduction parameters, (iv) modeling parameters. The output pruned data are empirical reduced bases and a reduced integration domain in order to setup the equations for the reconstruction of the original data in the decompression step. We show accurate modeling capabilities by using the pruned data when considering the extrapolation of the cyclic simulation over 190 additional cycles. Because of the cyclic strain ratcheting, approximation errors may be accumulated over cycles.

Hyperreduction methods belong to projection-based model order reduction methods (PBMORs) via linear dimensionality reduction. PBMOR methods pertain to problems where the simulation data that we wish to produce belong to a small vector space (i.e. having a small dimension) when the quantity of interest is not well established prior to any numerical prediction, and may require more data extrapolation or exploration. Several authors [34] have proposed model reduction techniques for the approximation of plasticity problems. For instance [6,7,10,38,50,52] proposed to use the LATIN method to iteratively approximate the solution, without using simulation data forecast by any finite element model. Fritzen et al. [19,20] use a space–time technique where a low number of nonlinear equations is solved in the reduced setting but full spatial information can be reconstructed at any given time. Access to reconstructed data enables to define quantities of interest as if finite element simulations had been performed. Michel and Suquet [36] and Franciosi and Berbenni [17] have proposed to use the nonuniform transformation field analysis (NTFA) approach where they consider nonuniform plastic strain fields with the aim of reducing the number of macroscopic internal variables. Another model order reduction technique used frequently is the proper orthogonal decomposition (POD), first proposed by [27,31] developed initially for statistical analyses. This POD basis comprises of the state subspace which are related to different time steps of a simulation or even different mechanical problems altogether. Using POD bases to predict mechanical models was first done by [32] for weather forecasts. In the above PBMOR methods, except for NTFA, the reduced equations are setup on the full finite element mesh. In addition, these model order reduction methods produce their own compressed representation of the simulated data. The issue here, is to compress simulation data that have not been produced by a model reduction method.

In PBMOR, a reduced basis is substituted for the usual finite element shape functions. Therefore dedicated numerical cubature schemes have been developed and named hyperreduction methods [13,41]. In hyperreduction PBMOR methods, the original mesh is sampled either at the level of integration points or at the level of elements. The idea of mesh sampling for reduced basis approximations has its origin in the Gappy POD method [12]. In [24], integration points are sampled for hyperreduced multiscale homogenization problems. In [41,42], the cubature is restricted to a reduced

integration domain (RID). The RID is a subdomain of the original domain where the finite element equations are set up. This original domain can also be the one observed via CT. In [25], the reduced integration domain method has been used to prune tomographic data and related simulation data in the framework of model calibration for homogeneous materials undergoing strain localizations. The pruning procedure simply consists in deleting the data outside the RID, where the RID contains the finite elements connected to interpolation points computed by the discrete empirical interpolation method (DEIM) [9] by considering the primal (displacement) reduced basis as well as the reduced basis generated using the strain field.

We restrict our attention to representative volume elements where a virtual microstructural realization is generated and appropriate boundary conditions are prescribed [28,39,49,55]. These boundary conditions usually represent an average stress or strain state of the material, which conforms to the macroscopic averaged response of the material.

The paper is organized in the following manner. Section 2 presents the crystal plasticity material model, the finite element mesh description as well as the material parameters used. In Sect. 3 the hyperreduction framework is explained, as well as the Gappy POD method. Section 4 shows the setup of a cyclic simulation. It contains the results of the analysis and a comparison of the proposed method with the Gappy POD. This is followed by the conclusions in Sect. 5.

2 Crystal plasticity model and finite element mesh description

Data pruning is relevant for simulation data of complex physics-based computational models. In the present work, a small strain crystal plasticity formulation is used for the computation as most local strains remain below 5%. Hence, nonlinear contributions to the mechanical balance equation are local in space. But they are highly nonlinear, with possible contribution of the local strain history. Hence the time instant t has to be introduced in the following equations, and the related partial derivative is denoted by an upper dot (i.e. $\dot{\boldsymbol{\varepsilon}}$ for strain rate). In a polycrystalline aggregate, the domain occupied by the material is split into subdomains termed grains, having the same crystal orientation. The displacement fields are supposed to be continuous at the grain boundaries. To ease the interpretation of the numerical results, a rate independent single crystal plasticity model recently proposed by [16] is used. Also, the model exclusively uses kinematic hardening because it governs ratcheting effect. In Nickel-baed superalloys for instance, kinematic hardening dominates isotropic hardening under cyclic loading conditions [11]. Face Centered Cubic (FCC) single crystal metallic materials are considered that are characterized by N plastic slip systems, each having a slip system direction $\boldsymbol{l}^s$ and the normal to the slip plane $\boldsymbol{n}^s$. The partition of the strain tensor introduces an elastic and a plastic part, denoted by $\boldsymbol{\varepsilon}^e$ and $\boldsymbol{\varepsilon}^p$ respectively:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p \tag{1}$$

The Hooke law relates the stress tensor, denoted by $\boldsymbol{\sigma}$, to the elastic strain tensor. For cubic elasticity, a fourth rank tensor of elasticity moduli $\boldsymbol{C}$, involving three independent parameters, governs the elastic behavior.

$$\boldsymbol{\sigma} = \boldsymbol{C}\boldsymbol{\varepsilon}^e \tag{2}$$

The plastic strain rate results from the slip processes with respect to all active slip systems,

$$\dot{\boldsymbol{\varepsilon}}^p = \sum_{s=1}^{N} \dot{\gamma}^s \boldsymbol{m}^s \tag{3}$$

with $\boldsymbol{m}^s$ being the orientation tensor

$$\boldsymbol{m}^s = \frac{\boldsymbol{l} \otimes \boldsymbol{n}^s + \boldsymbol{n} \otimes \boldsymbol{l}^s}{2} \tag{4}$$

The amount of slip rate on each slip system is denoted by the variable $\dot{\gamma}^s$. The driving force for plastic slip on slip system s is the resolved shear stress, computed using Cauchy stress according to:

$$\tau^s = \boldsymbol{\sigma} : \boldsymbol{m}^s = \sigma_{ij} m^s_{ij} \tag{5}$$

The yield criterion is a generalization of Schmid's law involving scalar hardening stresses r^s and x^s according to [35].

$$f^s(\boldsymbol{\sigma}, x^s, r^s) = |\tau^s - x^s| - r^s \tag{6}$$

Here, r^s denotes the radius of the elastic domain in terms of the resolved shear stress and x^s is a scalar back-stress characterizing the center of the elastic range in the one-dimensional space of resolved shear stresses. In slip based crystal plasticity, there are N such elastic ranges. Plastic slip can occur only if the function f^s becomes positive. In [16], the rate of slip on each slip system follows a rate-independent formulation of the form:

$$\dot{\gamma}^s = \dot{\varepsilon} \left\langle \frac{f^s}{P} \right\rangle \operatorname{sign}(\tau^s - x^s) \tag{7}$$

where P is a positive constant having the units of stress and $\dot{\varepsilon}$ is a non-negative homogeneous function of order one in the total strain rate. The Macauley brackets $\langle\bullet\rangle = \text{Max}(0, \bullet)$ are used to distinguish the elastic range from the plastic one, depending on the sign of the yield function f^s. In this model, $\dot{\varepsilon}$ is taken to be the total equivalent distortional strain rate:

$$\dot{\varepsilon} = \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}' : \dot{\boldsymbol{\varepsilon}}'}, \quad \dot{\boldsymbol{\varepsilon}}' = \dot{\boldsymbol{\varepsilon}} - \frac{1}{3}(\text{tr}(\dot{\boldsymbol{\varepsilon}}))\boldsymbol{I} \tag{8}$$

where $\dot{\boldsymbol{\varepsilon}}'$ is the deviatoric part of the total strain rate tensor $\dot{\boldsymbol{\varepsilon}}$ and : is the inner product for second order tensors. Since the rate of inelasticity is linear in the total equivalent strain rate $\dot{\boldsymbol{\varepsilon}}$, all the evolution equations in the proposed theory are homogeneous of order one in time, characterizing a rate-independent response. Also the rate of inelasticity is used for all states entailing no need for special treatment of loading and unloading conditions. Also the functional form of f and the evolution equations for isotropic (r^s) as well as kinematic (x^s) hardening remain unchanged. The cumulative slip variable v^s is defined for each slip system by the following evolution equation:

$$\dot{v}^s = |\dot{\gamma}^s| \tag{9}$$

Evolution equations for the kinematic hardening variables are taken from [8,35]. The nonlinear kinematic hardening evolution law

$$\dot{x}^s = C\,\dot{\gamma}^s - D\,\dot{v}^s\,x^s \tag{10}$$

depends on two material parameters, C and D. In the present paper, there is no isotropic hardening so that the stress r^s has a constant value r_0, which is the value of the initial resolved shear stress.

Here, the stress is estimated via the finite element approximation of the strain tensor:

$$\boldsymbol{\varepsilon} \approx \sum_{i=1}^{N_d} \boldsymbol{\varepsilon}(\boldsymbol{\varphi}_i)\,a_i, \quad \boldsymbol{\varepsilon}(\boldsymbol{\varphi}_i) = \frac{1}{2}\left(\nabla^T\boldsymbol{\varphi}_i + \nabla\boldsymbol{\varphi}_i\right) \tag{11}$$

according to the finite element approximation of the displacement field:

$$\boldsymbol{u}(\boldsymbol{x}, t) = \sum_{i=1}^{N_d} \boldsymbol{\varphi}_i(\boldsymbol{x})\,a_i(t) \quad \forall\,\boldsymbol{x} \in \Omega \tag{12}$$

Table 1 The crystal plasticity parameter set being used

Cubic elasticity	C_{1111} = 259,600 MPa
	C_{1122} = 179,000 MPa
	C_{1212} = 109,600 MPa
Critical resolved shear stress	R_0 = 320 MPa
Kinematic hardening	C = 100,000 MPa
	D = 1000
Overstress	P = 9 MPa

We denote by $\boldsymbol{\sigma}(\boldsymbol{a})$ the stress predicted by the finite element method. The finite element balance equation reads (in absence of body forces) :

$$\boldsymbol{c}(\boldsymbol{a}, t) = 0 \quad \text{with} \tag{13}$$

$$c_i(\boldsymbol{a}, t) = \int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{\varphi}_i) : \boldsymbol{\sigma}(\boldsymbol{a})\,dV - \int_{\partial_F\Omega} \boldsymbol{\varphi}_i\,\boldsymbol{F}(t)\,d\Gamma \quad i \in \{1, \ldots, N_d \tag{14}$$

where $\boldsymbol{F}(t)$ is a Neumann boundary condition imposed at the boundary $\partial_F\Omega$, $(\boldsymbol{\varphi}_i(x))_{i=1,\ldots N_d}$ are the shape functions of the FE model, N_d is the number of degrees of freedom in the finite element model, and $\boldsymbol{a}$ the vector of degrees of freedom.

Literature findings show that the macroscopic representation of a micro-heterogeneous metallic material can be achieved with as few as one hundred grains [26], but matching macroscopic properties is not the goal here. Rather, the aim here is to have a large enough statistical pool of information so that local material response can be analyzed. Of course computational limitations have to be acknowledged and extremely large polycrystals cannot be used. The crystal plasticity parameter set employed in this paper is given in Table 1. This is the same parameter set as the one used in the statistical analysis performed in [14] for Nickel-based superalloy IN718.

An implicit finite element code [54] is used to solve the problem where the global equilibrium is solved using a Newton–Raphson algorithm. Integration of constitutive equations at the Gauss points is performed using the second order Runge-Kutta method with automatic time stepping [5].

$\boldsymbol{\Sigma}$ is the spatial average of tensor $\boldsymbol{\sigma}$:

$$\boldsymbol{\Sigma} = \frac{1}{V}\int_\Omega \boldsymbol{\sigma}\,d\Omega \tag{15}$$

We restrict our attention to representative volumes of cubic shape. The six faces of Ω are denoted $Face_i^+$, $Face_i^-$, for i=1, 2, 3. $Face_i^+$ and $Face_i^-$ are opposite faces. Load controlled mixed boundary conditions were imposed on the geometry

such that the traction vector: $\underset{\sim}{\sigma}.\underline{n}$ is prescribed as:

$$\Sigma_{11} = \text{constant on} Face_1^+$$
$$u_1 = 0 \quad \forall \underline{x} \in Face_1^-$$

Homogeneous Neumann boundary conditions (free boundary conditions) are imposed on all opposing faces i.e.

$$\sigma_{21}n_1 = \sigma_{31}n_1 = 0 \quad \forall \underline{x} \in Face_1^+ \cup Face_1^-$$
$$\sigma_{12}n_2 = \sigma_{22}n_2 = \sigma_{32}n_2 = 0 \quad \forall \underline{x} \in Face_2^+ \cup Face_2^-$$
$$\sigma_{13}n_3 = \sigma_{23}n_3 = \sigma_{33}n_3 = 0 \quad \forall \underline{x} \in Face_3^+ \cup Face_3^-$$

These boundary conditions are complemented by suitable additional Dirichlet conditions to fix the overall rigid body motion.

3 Hyperreduction method for data pruning in computational mechanics of materials

In the above mechanical model, the original data that require large storage ressources are:

- the internal variables of the constitutive equations $((\gamma^s, x^s)_{s=1,\dots N}, \boldsymbol{\varepsilon}^p)$,
- the primal variable $\boldsymbol{u}$,
- the dual variable $\boldsymbol{\sigma}$ and the strain field $\boldsymbol{\varepsilon}$,

The following data are less memory demending:

- the finite element mesh of Ω,
- the crystal orientations at all Gauss points of the mesh,
- all constitutive parameters ($\boldsymbol{C}$, C, D, r^s, P),
- and the boundary conditions.

The internal variables, the primal variable, the dual variable and the crystal orientations are stored as series of large vectors. The sizes of these vectors scale linearly with the number of elements or with the number of nodes, in the mesh of Ω.

The proposed data pruning aims at restricting some modeling data to a reduced domain. This reduced domain is denoted by $\Omega_R \subset \Omega$. In practice, it is the union of the supports of the finite element shape functions $(\boldsymbol{\varphi}_i)_{i\in\mathcal{F}}$, where $\mathcal{F}$ is a set of indices introduced for the matrix formulation of the pruning method:

$$\Omega_R = \cup_{i\in\mathcal{F}} \text{sup}(\boldsymbol{\varphi}_i) \tag{16}$$

The detailed construction of the set $\mathcal{F}$ is given below. Since modeling capabilities must be preserved by the method, the original data that are saved after data pruning are the constitutive parameters ($\boldsymbol{C}$, C, D, r^s, P), the finite element mesh restricted to Ω_R, the crystal orientations in Ω_R and the boundary conditions restrained to Ω_R. Additional data have also to be saved in order to enable the lossy reconstruction of all the original data in the decompression step. This additional data are related to the hyperreduction scheme introduced in this section. A two steps decompression scheme is performed: it starts with a modeling step, where the original data are estimated in Ω_R; then the Gappy POD method [12] interpolates the estimated data from Ω_R to the full domain Ω. In this paper, this interpolation step solely concerns the total strain $\boldsymbol{\varepsilon}$.

In the Gappy POD method [12], any vector $\boldsymbol{a}$ belongings to the column space of $\boldsymbol{V}$, denoted by $colspan(\boldsymbol{V})$, can be estimated by using few entries of $\boldsymbol{a}$ denoted by $\boldsymbol{a}[\mathcal{F}]$, if $\boldsymbol{V}[\mathcal{F},:]$ is a full column rank matrix. Here, $colspan(\boldsymbol{V})$ is a linear latent space. It is usually obtained via the Snapshot POD [48] or a truncated singular value decomposition of a snapshot matrix containing training data. Let us denote by $\boldsymbol{Q} \in \mathbb{R}^{N_d\times m}$ the snapshot matrix of finite element predictions for displacement $\boldsymbol{a}$: $Q_{ij} = a_i(t_j)$ where snapshots are selected at computational time instants $\{t_j\}_{j=1}^m$. Its truncated singular value decomposition reads:

$$\boldsymbol{Q} = \boldsymbol{V}\,\boldsymbol{S}\,\boldsymbol{W}^T + \boldsymbol{R}, \quad \boldsymbol{V}^T\,\boldsymbol{V} = \boldsymbol{W}^T\,\boldsymbol{W} = \boldsymbol{I}_N,$$
$$\boldsymbol{V}^T\boldsymbol{R} = 0 \tag{17}$$

where $\boldsymbol{S}$ is a diagonal matrix than contains the N_R largest singular values in decreasing order, $\boldsymbol{V} \in \mathbb{R}^{N_d\times N_R}$ and $\boldsymbol{W} \in \mathbb{R}^{m\times N_R}$ are orthogonal matrices, and $\boldsymbol{I}_{N_R}$ is the identity matrix of size N_R. $\boldsymbol{R} \in \mathbb{R}^{N_d\times m}$ is the residual of this truncated decomposition.

In this paper, $\boldsymbol{a}[\mathcal{F}]$ is considered as pruned data when considering the Gappy POD method as a compression/decompression scheme for the original data $\boldsymbol{a}$. Knowing $\boldsymbol{V}$, $\mathcal{F}$ and $\boldsymbol{a}[\mathcal{F}]$, the decompression step related to the Gappy POD reads:

$$\boldsymbol{a}^G = \boldsymbol{V}\,(\boldsymbol{V}[\mathcal{F},:]^T\,\boldsymbol{V}[\mathcal{F},:])^{-1}\,\boldsymbol{V}[\mathcal{F},:]^T\,\boldsymbol{a}[\mathcal{F}] \tag{18}$$

When $\mathcal{F} = \mathcal{P}$, where $\mathcal{P}$ is the set of interpolation points for the columns of $\boldsymbol{V}$ by following the discrete empirical interpolation method [3,9], then $\boldsymbol{V}[\mathcal{P},:]$ is a square invertible matrix and the decompression procedure can be simplified as:

$$\boldsymbol{a}^{DEIM} = \boldsymbol{V}\,\boldsymbol{V}[\mathcal{P},:]^{-1}\,\boldsymbol{a}[\mathcal{P}] \tag{19}$$

By using this discrete empirical interpolation method, the Gappy POD is a rigorous interpolation scheme: $\boldsymbol{a}^{DEIM}[\mathcal{P}] =$

$\boldsymbol{a}[\mathcal{P}]$. When $\mathcal{F}$ contains interpolation points, $\mathcal{P} \subset \mathcal{F}$, the sub matrix $\boldsymbol{V}[\mathcal{F},:]$ is full column rank [15].

Property: The Gappy POD decompression of the pruned data is the following oblique projection:

$$\boldsymbol{a} \in \mathbb{R}^{N_d}, \quad \boldsymbol{a}^G = \boldsymbol{V}\,(\boldsymbol{Z}^T\,\boldsymbol{V})^{-1}\,\boldsymbol{Z}^T\,\boldsymbol{a}, \qquad \boldsymbol{Z}^T = \boldsymbol{V}[\mathcal{F},:]^T\,\boldsymbol{I}_{N_d}[\mathcal{F},:] \tag{20}$$

Therefore, if $\boldsymbol{a} \in colspan(\boldsymbol{V})$, and if $\boldsymbol{V}[\mathcal{F},:]$ is full rank, the reconstructed vector $\boldsymbol{a}^G$ is exact: $\boldsymbol{a}^G = \boldsymbol{a}$.

Some remarks are due at this stage on dimensionality reduction in the litterature. Dimensionality reduction is an interesting approach to save memory space when considering data storage. The singular value decomposition as well as the principal component analysis enable dimensionality reduction of data. This is a first step of the proposed data pruning procedure. Recently, autoencoders have been proposed for nonlinear dimensionality reduction schemes of time-dependent finite element models [21,29]. Autoencoders can capture nonlinear latent spaces when the singular value decomposition fails to sufficiently reduce the dimension of data. We can also mention the extension of the Gappy POD to simulation data having tensor format of arbritary order in [37].

Unfortunately, the pruning algorithm via the Gappy POD does not enable data extrapolation via mechanical modeling. Hence, we propose to compute $\boldsymbol{a}[\mathcal{F}]$ by using the hyperreduction method [44], as a first decompression step before using the Gappy POD. It aims at predicting $\boldsymbol{a}[\mathcal{F}]$ by using physical governing equations set up on a reduced integration domain (RID) and a projection on a vector subspace related to displacement field.

In essence, to set up the hyperreduced equations for a given FE model, this approach accounts for the low rank approximation of the reduced balance equations. For the sake of simplicity this can be elaborated using the linear step of a Newton Raphson algorithm, where the FE balance equation reads:

$$\boldsymbol{a}^{FE(n)}(t) = \sum_{j=1}^{n-1} \delta\boldsymbol{a}^{FE(j)}(t) \tag{21}$$

$$\boldsymbol{K}^{(n)}\delta\boldsymbol{a}^{FE(n)}(t) = -\boldsymbol{c}(\boldsymbol{a}^{FE(n)}, t) \tag{22}$$

where n is the iteration index of the Newton Raphson algorithm, $\delta\boldsymbol{a}^{FE(n)}$ is a linear correction for the approximate displacement vector $\boldsymbol{a}^{FE(n)} \in \mathbb{R}^{N_d}$, and $\boldsymbol{K}^{(n)} \in \mathbb{R}^{N_d \times N_d}$ is the tangent stiffness matrix of the FE model, whereas $\boldsymbol{c} \in \mathbb{R}^{N_d}$ is the residual of the balance equation. Newton Raphson iterations stop when $\|\boldsymbol{c}(\boldsymbol{a}^{FE(n)}, t)\| < \epsilon_{tol}$ for a given tolerance ϵ_{tol}. Let us introduce reduced order model variables denoted by $\boldsymbol{b}^{R(n)} = \sum_{j=1}^{n-1} \delta\boldsymbol{b}^{R(j)} \in \mathbb{R}^{N_R}$. The product $\boldsymbol{V}\boldsymbol{b}^{R(n)}$ has to approximate $\boldsymbol{a}^{FE(n)}$. In order to find unique solutions $\delta\boldsymbol{b}^{R(n)}$ the rank of $\boldsymbol{K}^{(n)}\boldsymbol{V}$ needs to only be N_R. Since N_d is usually larger than N_R, a few rows of $\boldsymbol{K}^{(n)}\boldsymbol{V}$ can be selected to preserve the rank of the submatrix. As proposed in [43], the hyperreduced balance equations are restricted to the RID using convenient test functions such that:

$$\boldsymbol{b}^{R(n)}(t) = \sum_{j=1}^{n-1} \delta\boldsymbol{b}^{R(j)}(t) \tag{23}$$

$$(\boldsymbol{V}[\mathcal{F},:])^T\boldsymbol{K}^{(n)}[\mathcal{F},:]\boldsymbol{V}\delta\boldsymbol{b}^{R(n)}(t) = -(\boldsymbol{V}[\mathcal{F},:])^T\boldsymbol{c}[\mathcal{F}](\boldsymbol{V}\boldsymbol{b}^{R(n)}, t) \tag{24}$$

Here, the Newton Raphson iterations stop when $\|(\boldsymbol{V}[\mathcal{F},:])^T\boldsymbol{c}[\mathcal{F}](\boldsymbol{V}\boldsymbol{b}^{R(n)}, t)\| < \epsilon_{tol}$. The hyperreduced solution is denoted by $\boldsymbol{a}^{HR(n)}$, $\boldsymbol{a}^{HR(n)} = \boldsymbol{V}\boldsymbol{b}^{R(n)}$.

Also, the RID must be large enough to have $\text{rank}(\boldsymbol{K}^{(n)}[\mathcal{F},:]\boldsymbol{V}) = N_R$; hence cardinality of the number of degrees of freedom in the RID should be greater than or equal to N_R.

According to the two steps decompression procedure of the proposed method, the extent of the RID must enable the reconstruction of full fields, via the Gappy POD. This concerns the finite element displacement fields and the finite element strains. These reconstructions have to be exact if displacement and strain fields both belong to $colspan(\boldsymbol{V})$ and $colspan(\boldsymbol{V}^\epsilon)$ respectively, where $\boldsymbol{V}^\epsilon$ is a reduced order basis for strain approximation. (For the sake of simplicity the reduced basis dedicated to the displacement does not have a superscript). So, by construction, the RID contains the DEIM interpolation points for both $\boldsymbol{V}$ and $\boldsymbol{V}^\epsilon$. The set of interpolation points related to $\boldsymbol{V}^\epsilon$ is denoted by $\mathcal{P}^\epsilon$. The number of interpolation points is equal to the number of empirical modes in the reduced basis. By following the k-SWIM algorithm proposed in [25], the sets $\mathcal{P}$ and $\mathcal{P}^\epsilon$, are not restrained to interpolation points of reduced bases. The number of points in this set is extended by a factor k. The k-SWIM points are more spread on Ω than the DEIM points. This algorithm is described in the "Appendix 1". When $k = 1$ the k-SWIM points are the DEIM points themselves. In addition, for each set of points $\mathcal{P}$ and $\mathcal{P}^\epsilon$ built by k-SWIM, we extract from the finite element mesh the degrees of freedom of the elements that contain the points in $\mathcal{P}$ and in $\mathcal{P}^\epsilon$. These sets of degrees of freedom are denoted $\mathcal{P}^+$ and $\mathcal{P}^{\epsilon+}$ respectively. In many practical situations, $\mathcal{F}$ also includes the degrees of freedom of a zone of interest. These degrees of freedom are denoted by $\mathcal{F}_o$. Therefore, we obtain the following set of degrees of freedom:

$$\mathcal{F} = \mathcal{F}_o \cup \mathcal{P}^+ \cup \mathcal{P}^{\epsilon+} \tag{25}$$

Hence, the parameters of the proposed pruning algorithm are those of the hyperreduction method: the number of empirical modes in the reduced bases $\boldsymbol{V}$ and $\boldsymbol{V}^\epsilon$, the parameter k and

the set $\mathcal{F}_o$. The smaller the number of these parameters, the better the compression.

Property: If the hyperreduced matrix $(\boldsymbol{V}[\mathcal{F},:])^T \boldsymbol{K}^{(n)}[\mathcal{F},:]\boldsymbol{V}$ is full rank, and if $\delta\boldsymbol{a}^{FE(n)} \in colspan(\boldsymbol{V})$ for all n, then the hyperreduced prediction $\delta\boldsymbol{a}^{HR(n)}$ is the following oblique projection of $\delta\boldsymbol{a}^{FE(n)}$:

$$\delta\boldsymbol{a}^{HR(n)} = \boldsymbol{V}\,(\boldsymbol{A}^T\boldsymbol{V})^{-1}\,\boldsymbol{A}^T\,\delta\boldsymbol{a}^{FE(n)} \tag{26}$$

where $\boldsymbol{A}^T = (\boldsymbol{V}[\mathcal{F},:])^T\boldsymbol{K}^{(n)}[\mathcal{F},:]$.

Thus, the proposed decompression procedure is an implicit projection, performed via hyperreduced balance equations, without knowing $\delta\boldsymbol{a}^{FE(n)}$. Such an approach enables data extrapolation via mechanical modeling.

We need to prove that $\delta\boldsymbol{b}^{R(n)} = (\boldsymbol{A}^T\boldsymbol{V})^{-1}\,\boldsymbol{A}^T\,\delta\boldsymbol{a}^{FE(n)}$ for all n. We propose a recursive proof. Let's assume that $\boldsymbol{a}^{HR(n)} = \boldsymbol{a}^{FE(n)}$. This is true for $n = 1$. We also assume it exists $\delta\boldsymbol{b}^{FE(n)}$ such that $\delta\boldsymbol{a}^{FE(n)} = \boldsymbol{V}\delta\boldsymbol{b}^{FE(n)}$. According to the balance equation (22) of the finite element model, the following equation holds: $\boldsymbol{A}^T\,\delta\boldsymbol{a}^{FE(n)} = -(\boldsymbol{V}[\mathcal{F},:])^T\boldsymbol{c}[\mathcal{F}](\boldsymbol{a}^{FE(n)},t)$. So $\boldsymbol{A}^T\,\delta\boldsymbol{a}^{FE(n)} = -(\boldsymbol{V}[\mathcal{F},:])^T\boldsymbol{c}[\mathcal{F}](\boldsymbol{a}^{HR(n)},t)$, which is the right hand side term of Eq. (24). $\boldsymbol{A}^T\boldsymbol{V}$ is the hyperreduced tangent matrix in balance equation (24). Hence we get $\delta\boldsymbol{b}^{R(n)} = (\boldsymbol{A}^T\boldsymbol{V})^{-1}\boldsymbol{A}^T\delta\boldsymbol{a}^{FE(n)}$. Since $\delta\boldsymbol{a}^{FE(n)} = \boldsymbol{V}\delta\boldsymbol{b}^{FE(n)}$, thus we also have $\delta\boldsymbol{b}^{R(n)} = \delta\boldsymbol{b}^{FE(n)}$ and therefore $\delta\boldsymbol{a}^{FE(n+1)} = \delta\boldsymbol{a}^{HR(n+1)}$. This ends the proof.

Once the hyperreduced prediction is known, we have access to local strain tensors for points in Ω_R:

$$\begin{aligned}\boldsymbol{\varepsilon}(\boldsymbol{x},t) &= \frac{1}{2}(\nabla^T\boldsymbol{u}^{HR} + \nabla\boldsymbol{u}^{HR}), \quad \boldsymbol{u}^{HR}(\boldsymbol{x},t)\\ &= \sum_{i=\mathcal{F}\cup\mathcal{I}} a_i^{HR}(t)\boldsymbol{\varphi}_i(\boldsymbol{x}), \quad \forall\,\boldsymbol{x}\in\Omega_R\end{aligned} \tag{27}$$

These strain tensors are usually evaluated at the Gauss points of the reduced mesh that covers Ω_R. Let $\boldsymbol{\alpha}$ denote the vector of all Gauss point values of strain tensor for the original finite element mesh. The hyperreduction predicts only few rows of this vector: $\boldsymbol{\alpha}[\mathcal{F}^\epsilon]$ where $\mathcal{F}^\epsilon$ contains the indices of strain components at Gauss points of the reduced mesh. The reduced basis $\boldsymbol{V}^\epsilon$ is obtained by the truncated singular value decomposition of a snapshot matrix that contains a collection of full vectors $\boldsymbol{\alpha}$ for the same selected computational time instants $\{t_j\}_{j=1}^m$ introduced for displacements. This snapshot matrix is denoted by $\boldsymbol{Q}^\epsilon$. It reads: $Q^\epsilon_{ij} = \alpha_i(t_j)$. The Gappy POD for the decompression of the full strain field, at Gauss points of the original finite element mesh, reads:[1] 10.5

$$\boldsymbol{\alpha}^G = \boldsymbol{V}^\epsilon\,(\boldsymbol{V}^\epsilon[\mathcal{F}^\epsilon,:]^T\,\boldsymbol{V}^\epsilon[\mathcal{F}^\epsilon,:])^{-1}\,\boldsymbol{V}^\epsilon[\mathcal{F}^\epsilon,:]^T\,\boldsymbol{\alpha}[\mathcal{F}^\epsilon] \tag{28}$$

[1] 8.5

Such kind of rule can be obtained for any variable equipped with a reduced basis.

We have to pay attention to the interface between the RID and its counterpart. Let $\Omega\backslash\Omega_R$ denote the counterpart of Ω_R. We call Γ^I the interface between Ω_R and $\Omega\backslash\Omega_R$. Next, we introduce $\mathcal{I}$, the set of degrees of freedom indices related to the interface Γ^I:

$$\mathcal{I} = \left\{ i\in\{1,...,N_d\} \mid \int_{\Gamma^I} \boldsymbol{\varphi}_i^T\,.\,\boldsymbol{\varphi}_i\,d\Gamma \neq 0 \right\} \tag{29}$$

The matrix $\boldsymbol{K}^{(n)}$ is sparse. In the sequel, we assume that non zero entries in $\boldsymbol{K}^{(n)}[\mathcal{F},:]$ are only in the submatrix $\boldsymbol{K}^{(n)}[\mathcal{F},\mathcal{F}\cup\mathcal{I}]$ and these entries can be computed by using solely the reduced mesh that covers Ω_R. This assumption is too strong in case of contact problems as shown in [15]. So we restrict our attention to contactless problems for the sake of simplicity. More details on the hyperreduction of contact problems are available in [15]. The solution of the hyperreduced balance equation requires to save $\boldsymbol{V}[\mathcal{F}\cup\mathcal{I},:]$ which is the restriction of the reduced basis to the RID. The reconstruction of the full strain field over Ω by using the Gappy POD requires to save the full strain modes $\boldsymbol{V}^\epsilon$ and $\mathcal{F}^\epsilon$. Once the full strain field is reconstructed, the solution of the constitutive equations gives access to all mechanical variables. In practice, the memory size required to save the strain modes $\boldsymbol{V}^\epsilon$ is one order of magnitude larger than the mechanical data required for the setting of the hyperreduced equations. The smaller the number of strain modes, the higher the data pruning, but the larger the approximation error in the decompression step.

4 Example of data extrapolation from pruned data in cyclic crystal plasticity

4.1 Feature selection for data pruning

A thousand grains polycrystal was chosen as an example where Fig. 1 shows a meshed finite element geometry with mixed traction boundary conditions. This finite element mesh was generated using the Voronoi tessellation technique with the help of the software VORO++ [45]. The mesh consists of 1,174,719 nodes and 859,848 reduced C8D10 quadratic tetrahedral elements. Each element consists of four Gauss points. The crystallographic orientations chosen were randomly distributed throughout the polycrystal. The stress loading Σ_{11} follows a triangular signal where $\max_t \Sigma_{11} = 1000$ MPa and $R_\Sigma = \frac{\min_t \Sigma_{11}(t)}{\max_t \Sigma_{11}(t)} = -0.85$. These boundary values have been chosen carefully in order to simulate strain-ratcheting, which is a strain accumulation during loading cycles.

In order to show the extrapolation capabilities of the pruning algorithm via hyperreduction, the reduced bases

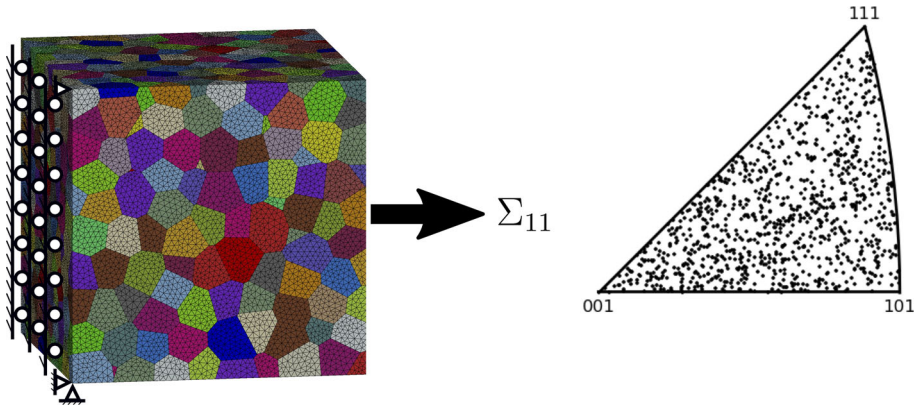

Fig. 1 A 1000 grains meshed microstructure with the inverse pole figure. Colors are related to grain orientations

are trained with simulation data related to the first ten loading cycles only. In the sequel, we show the ability of these reduced bases to predict up to 200 cycles. Figure 2 shows the local Euclidian norm for each displacement mode $\sum_{i=1}^{N_d} \boldsymbol{\varphi}_i(\boldsymbol{x})\ V_{ik}$ for $k = 1, \ldots 6$. Figure 3 shows for each strain mode, the local von Mises strain $\epsilon_{vM} = \sqrt{\frac{2}{3}\boldsymbol{\epsilon}'_{ij}\boldsymbol{\epsilon}'_{ij}}$, where $\boldsymbol{\epsilon}'$ is the deviatoric part of the local strain tensor. As anticipated, the magnitude of singular values exponentially decreases with the number of modes. For displacement 9 and for strain 20 modes are considered to get projection errors of simulation data on reduced bases around 2%. This cut-off is arbitrary and dependent on the user. By looking at the contour plot of the modes in both cases it can be seen that the first few modes are homogeneously distributed while the later ones are high gradient modes.

Figure 4 shows the reduced meshes constructed using the selected POD modes. A small RID (RID1) has been obtained by the k-SWIM algorithm with $k = 1$ in Algorithm 1. With this setting $\mathcal{P}$ and $\mathcal{P}^{\epsilon}$ are the interpolation point of the DEIM algorithm. A larger RID (RID2), with $k = 3$ in Algorithm 1, has been also generated. It contains twice as many elements that are more spread out in Ω. The part of the finite element mesh provided in the zone of interest depends on which part of the mesh is the most critical according to the user. In the present case, a small central part of the boundary where pressure is being applied is included in the zone of interest. Before pruning the data it has beeen observed that this region experiences the largest strain values under a tensile load. It is therefore a zone of interest. One can notice that the smaller the zone of interest, the number of modes and k, the higher the pruning of the original data.

The data to save in a storage system are summarized in Table 2, for three approaches: the original data set, the Gappy POD, the hyperreduction. The data pruning using hyperreduction consists in saving a physical model. No FE simulation data is stored, only reduced bases. These reduced bases can obviously be supplemented by saving the reduced coordinates $\boldsymbol{b}^R$ related to displacement and also the one related to the strain field. These coordinates occupy a negligible memory size.

Table 2 Data pruning summary

	Data to save for $\overline{m}$ time steps
FE	$\{\boldsymbol{\alpha}^{FE}(t_j)\}_{j=1}^{\overline{m}}$
Gappy POD	$\{\boldsymbol{\alpha}^{FE}(t_j)[\mathcal{F}^{\epsilon}]\}_{j=1}^{\overline{m}}, \boldsymbol{V}^{\epsilon}$
Hyperreduction	Constitutive parameters, grain orientations in Ω_R, $\boldsymbol{V}^{\epsilon}, \boldsymbol{V}[\mathcal{F} \cup \mathcal{I}, :]$

4.2 Data extrapolation over 200 cycles

After constructing the reduced basis and the RIDs using simulation data over 10 cycles, extrapolation of the original data was performed via hyperreduction for two hundred loading cycles. This aims to evaluate the modeling capabilities of the data pruning method. Let $\overline{m}$ denote the number of time steps involved in the extrapolation of the strain field for two hundred loading cycles. Here, $\overline{m} = 20\,m$ (m is the number of columns in $\mathbf{Q}$), which represents an extrapolation of the original FE data over 190 loading cycles.

To compare the results, the original full field simulation was also run for two hundred cycles. Figure 5 shows reconstructed macroscopic data. It shows the usual plot of macroscopic stress (Σ_{11}, enforced value as Neumann boundary condition) versus the computed average strain (E_{11}) response of the three cases, hyperreduction using RID1 (small RID), hyperreduction using RID2 (large RID) and the original data for comparison. To avoid clutter, responses are plotted only at cycles 1 and 200. It can be seen that the larger RID gives more accurate predictions of decompressed data than the small RID at cycle 200. The pruning looses almost no information for the first 10 cycles for both RIDs, with decompression errors below 1% for all simulation data.

Statistics have been performed on local results that pertain to the time step taken at the tensile peak of cycle 200, after the decompression step of the full strain field by using the Gappy

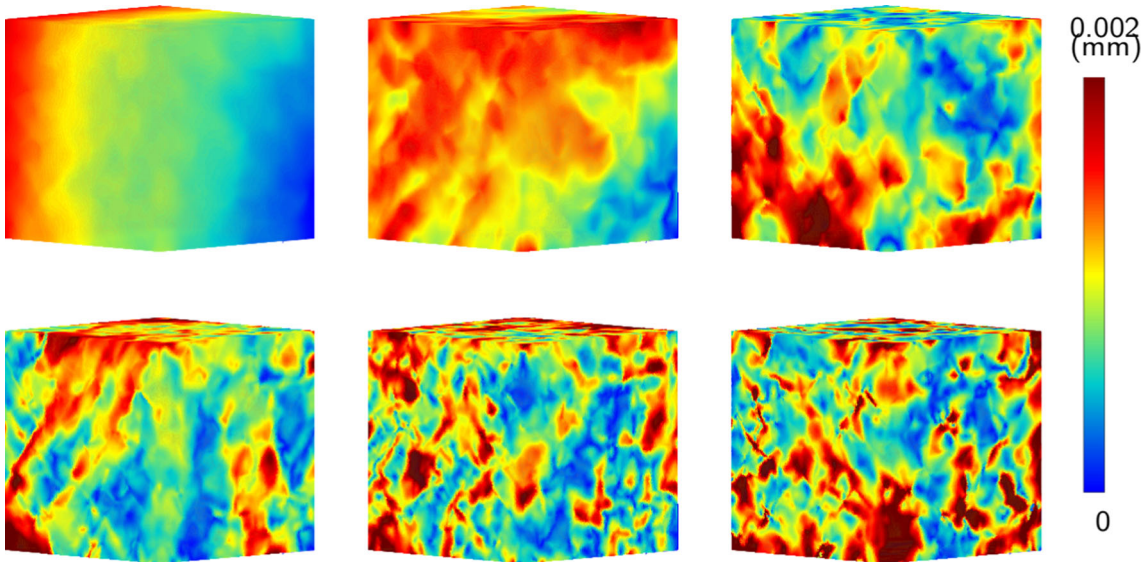

Fig. 2 Six first POD modes for displacement among 9 selected ($N = 9$) for the reduced basis $\boldsymbol{V}$. Here, the norms of local displacement vectors have been plotted for each mode

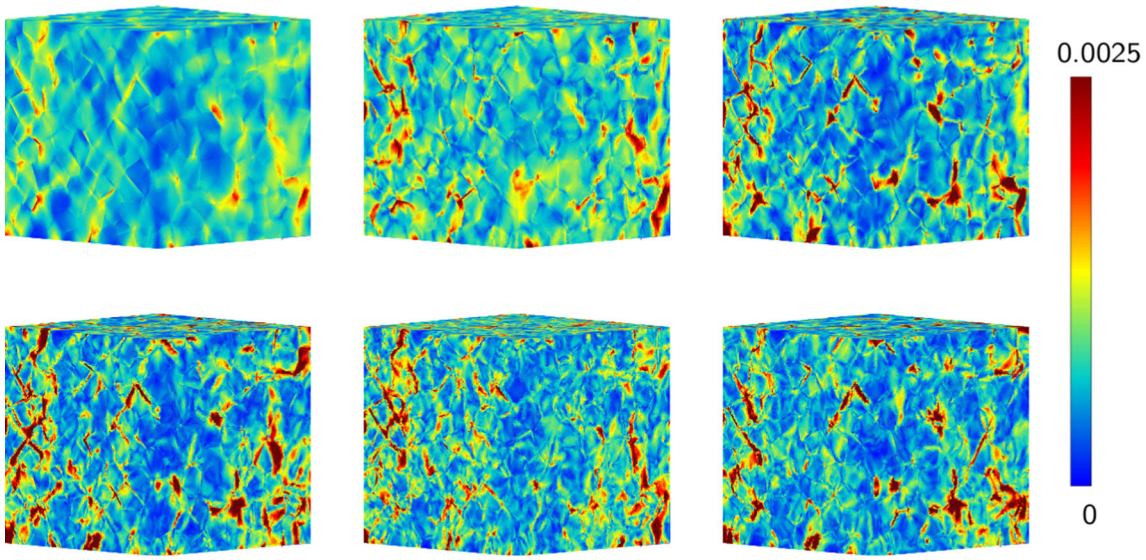

Fig. 3 Six first POD modes for strain field among 20 selected for the reduced basis $\boldsymbol{V}^{\epsilon}$. Here the local von Mises strains are plotted for each mode

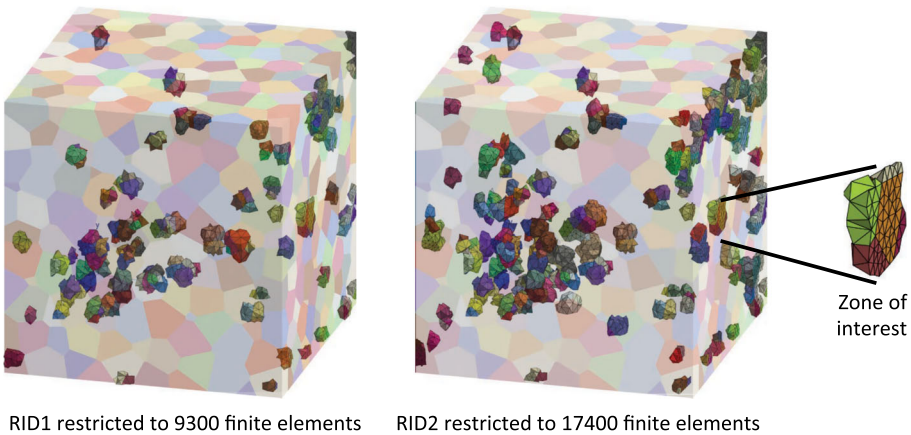

Fig. 4 On the left a small reduced integration domain (RID1) restricted to 9300 finite elements, on the right a larger one (RID2) involving 17,400 finite elements. The original FE mesh is shown in transparency behind the RIDs. It contains 859,848 elements. Colors are related to grain orientations. The contribution of $\mathcal{F}_o$, as a zone of interest, is shown on the right of RID2

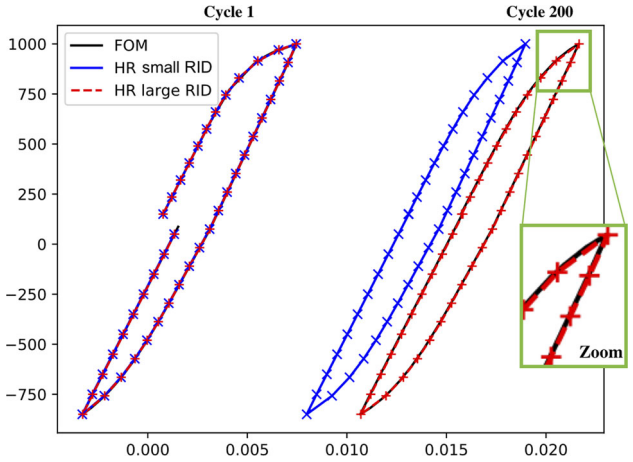

Fig. 5 Reconstructed macroscopic data. Macroscopic stress vs macroscopic strain response for cycle 1 and cycle 200 for the original FE prediction as well as the HROM for both the small RID (RID1) and the large RID (RID2)

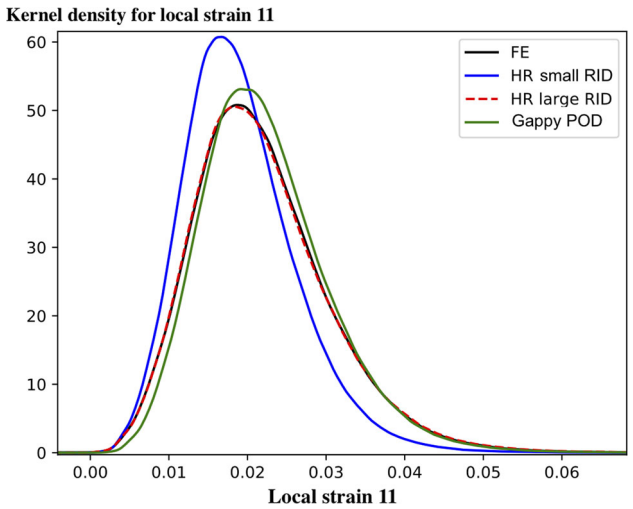

Fig. 6 Kernel of the probability density function for Gauss point values of ε_{11} at the tensile peak of the 200th cycle

POD applied to FE strains, RID1, RID2 and the original FE data. The Gappy POD applied to FE strains requires to save the strain field in Ω_R over the 200 cycles as pruned data. It is more memory demanding than saving only the data related to the last cycle, but mandatory to monitor the strain-ratcheting in time. Anyway, it represents a small selection of the original strain field. Figure 6 shows the kernel of the probability density function for ε_{11}, estimated from Gauss point values by using the method proposed in [4]. This method is available in Scipy [51]. The strain distribution plots show that locally the decompression of the full strain field via hyperreduction over RID2 is more precise than using RID1 or the Gappy POD. The oblique projection involved in hyperreduction appears to be more accurate than the oblique projection performed by the Gappy POD. The former includes physical equations whereas the latter does not. Average values of local strain ε_{11} and the standard deviation are reported in Table 3.

The computational requirements to decompress the strain field over 200 cycles can be viewed in Table 3 where the numbers related to the original full field data have been also reported. The original FE simulation takes 1034 h. It should be noted that the number of parallel processors utilized in this study have been arbitrarily chosen. For the full field simulation 24 processors were selected because that is the maximum number available on each cluster node. For the HROM and Hybrid simulations, four processors were used each. The computation time for the prediction of the full strain field via hyperreduction was 27 h for RID1 while for RID2 it took 38 h. Although the time to run the hyperreduction on the large RID was 40% longer, much accurate prediction in strain was achieved. The storage requirement for the pruned data is almost the same for both RIDs. It is less than 7% of the original storage space. This storage space is occupied by the reduced bases $\boldsymbol{V}$ and $\boldsymbol{V}^{\epsilon}$. Therefore, we save 93% of storage space, the reconstruction of the full strain field takes 38 h and its accuracy is higher than 99%. The decompression based on FE strain over the RID and the Gappy POD is the fastest decompression procedure. But it does not enable data extrapolation.

Table 3 Finite element mesh details and the resources needed to run the full field, hyperreduced and Gappy POD, predictions

	Original FE data	Gappy POD	Small RID	Large RID
Nodes	1,174,719	19,942	19,942	35,893
$card(\mathcal{F})$	–	–	X	X
Elements	859,848	9286	9286	17,381
Computation time × processors	1034 h × 24	0.08 h × 1	27 h × 4	38 h × 4
Storage memory Gbs	101.	5.	3.5	3.5
E_{11} (error)	0.019	0.02 (5%)	0.017 (10%)	0.019 (< 1%)
Std deviation for ε_{11} (error)	0.84	0.88 (5%)	0.86 (2%)	0.84 (< 1%)

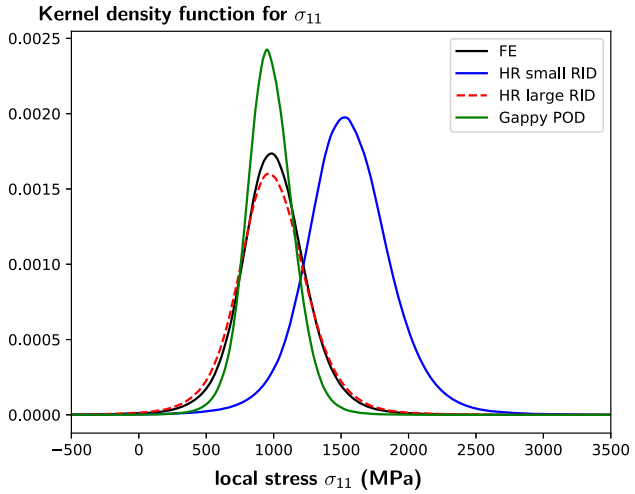

Fig. 7 Kernel of the probability density function for Gauss point values of σ_{11} at the tensile peak of the 200th cycle

The hyperreduction scheme being an approximate solution of mechanical equations, we can compare the computation time of this method with the finite element method. Here, the following simulation speedups are achieved by hyperreduction for the full cyclic simulation: 900 for the larger RID and 1200 for the smallest RID, when using a single processor. Hence, the modeling step of the decompression procedure is very fast compared to the finite element method.

In Fig. 7, similarly to strain decompression, the accuracy of stress decompression via hyperreduction over RID2 is more precise than using RID1 or the Gappy POD. A dedicated reduced basis for stress has been computed for that purpose, although the construction of RID1 and RID2 does not account for this reduced basis. Note that the average value of stress σ_{11} is imposed in the finite element model by the Neumann boundary condition. This boundary condition is used to plot the macroscopic response in Fig. 5. The reconstruction procedure accurately replicate this average stress by using RDI2 of the hyperreduction but not with RID1.

5 Conclusion

Crystalline plasticity modeling is fed by experimental data and produces large volumes of simulated data. Erasing simulation data to free up memory space becomes necessary as more and more mechanical tests are performed. But there is a risk in erasing data: the risk of not being able to investigate modeling issues further once the data has been erased. In the present paper, we show that the data pruning we have developed allows to reduce the volume of data saved in storage system while keeping the possibility to evaluate the strain-ratcheting phenomenon. It is a very complex phenomenon, depending on the loading conditions applied to a polycrystal and the mechanical behavior of each grain. The decompression procedure of strain fields involves physical governing equations. It is nevertheless 1000 times faster than the original FE simulation. Simulation data in crystal plasticity have been compressed to 7% of their original size with approximation error lower than 1%, for total strain. Results show the capability of choosing the size of the pruned data via the following hyperreduction parameters: the number of displacement and strain modes, the parameter k for the RID construction, and the zone of interest. This pruning method has extrapolation capabilities via mechanical modeling. In this paper, the input data of the pruning algorithm are related to 10 cycles of a cyclic simulation. After the data pruning, the recovery procedure has accurately extrapolated the pruned data up to 200 cycles. Although model reduction in crystal plasticity is not new, the fact that such methods can perform data pruning while preserving modeling capabilities is rather ignored in the literature. The proposed method has to be considered only if we want to preserve such modeling capabilities. If not, it is useless.

The hyper-reduction scheme involved in the data pruning procedure has shown its accuracy when considering various parameter variations such as loading parameter or constitutive coefficients. The method can be applied to even more complex loading conditions, including multiaxial mechanical testing, and to other microstructures of composite materials. In the related works in progress, we investigate the insertion of defect in the reduced integration domain in order to evaluate stress variations due to defects.

Acknowledgements This work was supported by Safran via the sponsorship chair Cristal.

Appendix

References

1. Alharbi HF, Kalidindi SR (2015) Crystal plasticity finite element simulations using a database of discrete Fourier transforms. Int J Plast 66:71–84. https://doi.org/10.1016/j.ijplas.2014.04.006
2. Bacry E, Gaiffas S, Leroy F, Morel M, Nguyen DP, Sebiat Y, Sun D (2020) Scalpel3: a scalable open-source library for healthcare claims databases. Int J Med Inform 141:104203. https://doi.org/10.1016/j.ijmedinf.2020.104203
3. Barrault M, Maday Y, Nguyen N, Patera A (2004) An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. C R Math Acad Sci Paris Ser I 339:667–672
4. Bashtannyk D, Hyndman R (2001) Bandwidth selection for kernel conditional density estimation. Comput Stat Data Anal 36:279–298
5. Besson J, Cailletaud G, Chaboche J, Forest S (2009) Non-linear mechanics of materials, vol 167. Springer, Berlin
6. Bhattacharyya M, Fau A, Nackenhorst U, Néron D, Ladevèze P (2018) A multi-temporal scale model reduction approach for the computation of fatigue damage. Comput Methods Appl Mech Eng 340:630–656. https://doi.org/10.1016/j.cma.2018.06.004

Algorithm 1: k-SWIM Selection of Variables with EmpIrical Modes

Input : integer k, linearly independent empirical modes $\boldsymbol{v}_l \in \mathbb{R}^d, l = 1, \ldots, M$
Output: variables index set $\mathcal{P}^{(k)}$

set $\mathcal{P}_0 := \emptyset$, $j = 0$, $\boldsymbol{U}_1 = [\,]$; // initialization
for $l = 1, \ldots, M$ **do**
  $\boldsymbol{r}_l \leftarrow \boldsymbol{v}_l - \boldsymbol{U}_l \left((\boldsymbol{U}_l[\mathcal{P}_j, :])^T \boldsymbol{U}_l[\mathcal{P}_j, :] \right)^{-1} (\boldsymbol{U}_l[\mathcal{P}_j, :])^T \boldsymbol{v}_l[\mathcal{P}_j]$; // residual vector
  for $n = 1, \ldots, k$ **do**
    $j \leftarrow j + 1$; // add the k largest value of the residual
    $i_j \leftarrow \arg\max_{i \in \{1,\ldots,d\} \setminus \mathcal{P}_{j-1}} |\boldsymbol{r}_l[i]|$; // index selection
    $\boldsymbol{r}_l[i_j] \leftarrow 0$; // variable already selected
    $\mathcal{P}_j \leftarrow \mathcal{P}_{j-1} \cup \{i_j\}$; // extend index set
  $\boldsymbol{U}_{l+1} \leftarrow [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_l]$; // truncated reduced matrix
set $\mathcal{P}^{(k)} := \mathcal{P}_j$.

7. Boucard PA, Ladevèze P, Poss M, Rougée P (1997) A nonincremental approach for large displacement problems. Comput Struct 64(1):499–508. https://doi.org/10.1016/S0045-7949(96)00165-4
8. Busso EP, Cailletaud G (2005) On the selection of active slip systems in crystal plasticity. Int J Plast 21(11):2212–2231
9. Chaturantabut S, Sorensen D (2010) Nonlinear model reduction via discrete empirical interpolation. SIAM J Sci Comput 32(5):2737–2764
10. Chinesta F, Ladeveze P, Ibanez R, Aguado JV, Abisset-Chavanne E, Cueto E (2017) Data-driven computational plasticity. Procedia Eng 207:209–214. https://doi.org/10.1016/j.proeng.2017.10.763
11. Cruzado A, Lorca J, Segurado J (2017) Modeling cyclic deformation of inconel 718 superalloy by means of crystal plasticity and computational homogenization. Int J Solids Struct 122–123:148–161. https://doi.org/10.1016/j.ijsolstr.2017.06.014
12. Everson R, Sirovich L (1995) Karhunen–Loève procedure for gappy data. J Opt Soc Am A 12:1657–1664
13. Farhat C, Avery P, Chapman T, Cortial J (2014) Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. Int J Numer Methods Eng 98(9):625–662
14. Farooq H, Cailletaud G, Forest S, Ryckelynck D (2019) Crystal plasticity modeling of the cyclic behavior of polycrystalline aggregates under non-symmetric uniaxial loading: Global and local analyses. Int J Plast 126:102619. https://doi.org/10.1016/j.ijplas.2019.10.007
15. Fauque J, Ramière I, Ryckelynck D (2018) Hybrid hyper-reduced modeling for contact mechanics problems. Int J Numer Methods Eng 115(1):309–317. https://doi.org/10.1002/nme.5798
16. Forest S, Rubin M (2016) A rate-independent crystal plasticity model with a smooth elastic–plastic transition and no slip indeterminacy. Eur J Mech A Solids 55:278–288
17. Franciosi P, Berbenni S (2007) Heterogeneous crystal and polycrystal plasticity modeling from a transformation field analysis within a regularized Schmid law. J Mech Phys Solids 55(11):2265–2299. https://doi.org/10.1016/j.jmps.2007.04.012
18. Frankel A, Jones R, Alleman C, Templeton J (2019) Predicting the mechanical response of oligocrystals with deep learning. Comput Mater Sci 169:109099. https://doi.org/10.1016/j.commatsci.2019.109099
19. Fritzen F, Hassani M (2018) Space-time model order reduction for nonlinear viscoelastic systems subjected to long-term loading. Meccanica 53(6):1333–1355
20. Fritzen F, Leuschner M (2013) Reduced basis hybrid computational homogenization based on a mixed incremental formulation. Comput Methods Appl Mech Eng 260:143–154. https://doi.org/10.1016/j.cma.2013.03.007
21. Gao H, Wang JX, Zahr MJ (2020) Non-intrusive model reduction of large-scale, nonlinear dynamical systems using deep learning. Physica D 412:132614. https://doi.org/10.1016/j.physd.2020.132614
22. Gérard C, Cailletaud G, Bacroix B (2013) Modeling of latent hardening produced by complex loading paths in FCC alloys. Int J Plast 42:194–212
23. Gu T, Medy JR, Klosek V, Castelnau O, Forest S, Hervé-Luanco E, Lecouturier-Dupouy F, Proudhon H, Renault PO, Thilly L, Villechaise P (2019) Multiscale modeling of the elasto-plastic behavior of architectured and nanostructured Cu-Nb composite wires and comparison with neutron diffraction experiments. Int J Plast
24. Hernández J, Oliver J, Huespe A, Caicedo M, Cante J (2014) High-performance model reduction techniques in computational multiscale homogenization. Comput Methods Appl Mech Eng 276:149–189
25. Hilth W, Ryckelynck D, Menet C (2019) Data pruning of tomographic data for the calibration of strain localization models. Math Comput Appl 24(1)
26. Kanit T, Forest S, Galliet I, Mounoury V, Jeulin D (2003) Determination of the size of the representative volume element for random composites: statistical and numerical approach. Int J Solids Struct 40(13):3647–3679
27. Karhunen K (1946) Zur spektraltheorie stochastischer prozesse. Ann Acad Sci Finnicae Ser A 1:34
28. Kotha S, Ozturk D, Ghosh S (2019) Parametrically homogenized constitutive models (phcms) from micromechanical crystal plasticity fe simulations, part i: sensitivity analysis and parameter identification for titanium alloys. Int J Plast 120:296–319
29. Lee K, Carlberg KT (2020) Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. J Comput Phys 404:108973. https://doi.org/10.1016/j.jcp.2019.108973
30. Liu Z, Bessa M, Liu WK (2016) Self-consistent clustering analysis: an efficient multi-scale scheme for inelastic heterogeneous materials. Comput Methods Appl Mech Eng 306:319–341. https://doi.org/10.1016/j.cma.2016.04.004
31. Loève M (1963) Probability theory. The university series in higher mathematics, NJ, 3rd edn. Van Nosterand, Princeton
32. Lorenz EN (1956) Empirical orthogonal functions and statistical weather prediction. Stat Forecast 1
33. Masui K, Amiri M, Connor L, Deng M, Fandino M, Höfer C, Halpern M, Hanna D, Hincks A, Hinshaw G, Parra J, Newburgh L, Shaw J, Vanderlinde K (2015) A compression scheme for radio data in high performance computing. Astron Comput 12:181–190. https://doi.org/10.1016/j.ascom.2015.07.002
34. Matouš K, Geers MG, Kouznetsova VG, Gillman A (2017) A review of predictive nonlinear theories for multiscale modeling of heterogeneous materials. J Comput Phys 330:192–220
35. Méric L, Poubanne P, Cailletaud G (1991) Single crystal modeling for structural calculations: part 1—model presentation. J Eng Mater Technol 113
36. Michel J, Suquet P (2003) Nonuniform transformation field analysis. Int J Solids Struct 40(25):6937–6955
37. Olivier C, Ryckelynck D, Cortial J (2019) Multiple tensor train approximation of parametric constitutive equations in elasto-viscoplasticity. Math Comput Appl. https://doi.org/10.3390/mca24010017

38. Pelle JP, Ryckelynck D (2000) An efficient adaptive strategy to master the global quality of viscoplastic analysis. Comput Struct 78(1):169–183
39. Prithivirajan V, Sangid MD (2018) The role of defects and critical pore size analysis in the fatigue response of additively manufactured in718 via crystal plasticity. Mater Des 150:139–153
40. Rovinelli A, Sangid MD, Proudhon H, Guilhem Y, Lebensohn RA, Ludwig W (2018) Predicting the 3d fatigue crack growth rate of small cracks using multimodal data via Bayesian networks: in-situ experiments and crystal plasticity simulations. J Mech Phys Solids 115:208–229. https://doi.org/10.1016/j.jmps.2018.03.007
41. Ryckelynck D (2005) A priori hyperreduction method: an adaptive approach. J Comput Phys 202(1):346–366
42. Ryckelynck D (2009) Hyper-reduction of mechanical models involving internal variables. Int J Numer Methods Eng 77(1):75–89
43. Ryckelynck D, Lampoh K, Quilici S (2016) Hyper-reduced predictions for lifetime assessment of elasto-plastic structures. Meccanica 51(2):309–317. https://doi.org/10.1007/s11012-015-0244-7
44. Ryckelynck D, Missoum-Benziane D, Musienko A, Cailletaud G (2010) Toward "green" mechanical simulations in materials science: hyper-reduction of a polycrystal plasticity model. Revue Européenne de Mécanique Numérique/European Journal of Computational Mechanics 19(4):365–388
45. Rycroft C (2009) Voro++: a three-dimensional voronoi cell library in C++. Chaos 19. https://doi.org/10.1063/1.3215722
46. Sedighiani K, Diehl M, Traka K, Roters F, Sietsma J, Raabe D (2020) An efficient and robust approach to determine material parameters of crystal plasticity constitutive laws from macro-scale stress–strain curves. Int J Plast 134:102779. https://doi.org/10.1016/j.ijplas.2020.102779
47. Shantsev DV, Jaysaval P, de la Kethulle de Ryhove S, Amestoy PR, Buttari A, L'Excellent JY, Mary T (2017) Large-scale 3-D EM modelling with a block low-rank multifrontal direct solver. Geophys J Int 209(3):1558–1571
48. Sirovich L (1987) Turbulence and the dynamics of coherent structures, parts I, II and III. Q Appl Math XLV(3):561–590
49. Sun F, Meade ED, ODowd NP (2018) Microscale modelling of the deformation of a martensitic steel using the voronoi tessellation method. J Mech Phys Solids 113:35–55
50. Verwaerde R, Guidault PA, Boucard PA (2021) A non-linear finite element connector model with friction and plasticity for the simulation of bolted assemblies. Finite Elem Anal Des 195:103586. https://doi.org/10.1016/j.finel.2021.103586
51. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Jarrod Millman K, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey C, Polat İ, Feng Y, Moore EW, Vand erPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, Contributors S (2020) SciPy 1.0: fundamental algorithms for scientific computing in python. Nat Methods 17:261–272 . https://doi.org/10.1038/s41592-019-0686-2
52. Wang P, Dong XH, Fu LJ (2010) Simulation of bulk metal forming processes using one-step finite element approach based on deformation theory of plasticity. Trans Nonferrous Met Soc China 20(2):276–282 . https://doi.org/10.1016/S1003-6326(09)60134-5
53. Yagawa G, Shioya R (1993) Parallel finite elements on a massively parallel computer with domain decomposition. Comput Syst Eng 4(4):495–503
54. Z-set package: non-linear material & structure analysis suite (2013). www.zset-software.com
55. Zhang H, Diehl M, Roters F, Raabe D (2016) A virtual laboratory using high resolution crystal plasticity simulations to determine the initial yield surface for sheet metal forming operations. Int J Plast 80:111–138