



HAL
open science

A nonintrusive distributed reduced-order modeling framework for nonlinear structural mechanics-Application to elastoviscoplastic computations

Fabien Casenave, Nissrine Akkari, Felipe Bordeu, Christian Rey, David Ryckelynck

► To cite this version:

Fabien Casenave, Nissrine Akkari, Felipe Bordeu, Christian Rey, David Ryckelynck. A nonintrusive distributed reduced-order modeling framework for nonlinear structural mechanics-Application to elastoviscoplastic computations. *International Journal for Numerical Methods in Engineering*, 2019, 121 (1), pp.32-53. 10.1002/nme.6187. hal-02530085

HAL Id: hal-02530085

<https://hal.science/hal-02530085>

Submitted on 5 Jun 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

A Nonintrusive Distributed Reduced Order Modeling Framework for nonlinear structural mechanics – application to elastoviscoplastic computations

Fabien Casenave*, Nissrine Akkari*, Felipe Bordeu*, Christian Rey*, David Ryckelynck†

August 27, 2019

Abstract

In this work, we propose a framework that constructs reduced order models for nonlinear structural mechanics in a nonintrusive fashion, and can handle large scale simulations. We identify three steps that are carried out separately in time, and possibly on different devices: (i) the production of high-fidelity solutions by a commercial software, (ii) the *offline* stage of the model reduction and (iii) the *online* stage where the reduced order model is exploited. The nonintrusivity assumes that only the displacement field solution is known, and relies on operations on simulation data during the *offline* phase by using an in-house code. The compatibility with a new commercial code only needs the implementation of a routine converting the mesh and result format into our in-house data format. The nonintrusive capabilities of the framework are demonstrated on numerical experiments using commercial versions of the finite element softwares Zset and Ansys Mechanical. The nonlinear constitutive equations are evaluated by using the same external plugins as for Zset or Ansys Mechanical. The large scale simulations are handled using domain decomposition and parallel computing with distributed memory. The features and performances of the framework are evaluated on two numerical applications involving elastoviscoplastic materials: the second one involves a model of high-pressure blade, where the framework is used to extrapolate cyclic loadings in 6.5 hours, whereas the reference high-fidelity computation would take 9.5 days.

1 Introduction

A performance race is currently underway in the aircraft engine industry. The materials of some critical parts of the engine are pushed to the limit of their strength to increase as much as possible the efficiency of the propulsion. In particular, the high-pressure turbine blades, which are located directly downstream from the chamber of combustion, undergo extreme thermal loading. The possible causes of failure for these turbines include temperature creep rupture and high-cycle fatigue [36, 20]. Many efforts are spent to increase the strength of these turbines, with the use of thermal barriers [43], advanced superalloys [14] and complex internal cooling channels [5, 48], see Figure 1 for a representation of a high-pressure turbine blade. The lifetime prediction of such complex parts is a very demanding computational task: the meshes are very large to account for small structures such as the cooling channels, the constitutive laws are strongly nonlinear and involve a large number of internal variables, and more importantly, a large number of cycles has to be simulated. Indeed, failures come from local structural effects whose evolution cannot be predicted without computing potentially hundreds of cycles.

Like the lifetime prediction of turbine blades, many industrial needs involve very intensive numerical procedures, where the approximations of partial differential equations are solved. Efforts are spent to mitigate the runtime issues. In the context of nonlinear structural mechanics, we consider the conjugation of parallel computing with distributed memory to accelerate the resolution of large problems with Reduced Order Modeling, which learns about the physical operators from a solution set computed beforehand to accelerate further computations.

*SafranTech, Rue des Jeunes Bois - Châteaufort, 78114 Magny-les-Hameaux, France

†MINES ParisTech, PSL - Research University, Centre des matériaux, CNRS UMR-7633, Evry, France



Figure 1: Schematic of a high-pressure turbine blade [1]. Internal cooling channels allow the cool air to exit and protect the outer surface of the blade with a cooling film.

In what follows, we consider *a posteriori* reduced order modeling methods, where the equations of the physics are solved in the *online* stage. These methods require the resolution of high-fidelity solutions, in general using a finite element code. In this context, the reduced model consists in solving the same Galerkin problem as the high-fidelity code, but on a reduced order basis instead of a finite element basis. These methods are intrinsically intrusive: when applied using a given finite element code, modifications have to be implemented up to the assembling routines. As a consequence, the majority of the contributions to these methods are illustrated using in-house codes. We refer to the introduction of [29] for a detailed reflection on the notion of intrusivity of approximation methods for parametric equations, and the references within for methods falling into intrusive and nonintrusive categories. Following [29], we call a reduction method nonintrusive if it only relies on commercial software to compute the high-fidelity solution, and *a posteriori* reduced order modeling methods belong to the so-called *Galerkin-like* methods, for which only few nonintrusive contributions are acknowledged.

In the industry, nonintrusive methods to accelerate numerical simulations are widely used. Besides the practicality, more important reasons explain the use of nonintrusive methods: all the simulation methodologies are certified using a given version of a code and modifying this code means a new certification procedure; moreover, commercial codes are very often used in design offices to take advantage of assistance contracts provided by the editors of the software. Hence, deriving nonintrusive procedures appears a prerequisite to spread these methods in the industry. The literature about nonintrusive methods reducing the computational complexity of numerical simulation is vast, but in the large majority, not being *Galerkin-like*, they tend to “forget” the original equations. Concerning statistical and regression learning techniques, we can cite the following reviews [37, 7, 24, 33, 47]. In the *a posteriori* reduced order modeling, efforts have been spent to mitigate the intrusivity constraints. In [15], a reduced basis method is proposed for boundary elements methods only requiring the high-fidelity code to multiply the operator by user-provided vectors (approximately using fast multipole method, the matrix operator never being constructed). In [29], parametric problems are approximated only assuming we can use the first iteration of the high-fidelity iterative solver. In [9, 31], a reduced order basis is computed, and the coefficients of the reduced solution on this basis are computed using interpolation or regression methods instead of solving the Galerkin method in the *online* phase. In [17], these coefficients are approximated using a two-grid method and only resorting to the finite-element code in a nonintrusive fashion.

The contribution of this work consists in the construction of a framework able to reduce large scale nonlinear structural mechanics problems in a nonintrusive fashion, namely using commercial codes. The framework uses different algorithms taken from the literature and employs them to treat large

scale problems. All the computation procedure is parallel with distributed memory: the computation of the high-fidelity model, the reduction routines using in-house distributed python routines, up to the visualization and post-treatment. The nonintrusivity feature is obtained by having coded all the finite-element operation needed in *a posteriori* reduced order modeling procedure in a collection of in-house routines. A detailed numerical application is provided in Section 5.2, where we explain how this reduced order framework has been used to tackle the lifetime prediction of large models of high-pressure turbine blades. To the limit of our knowledge, this is the first nonintrusive *a posteriori* reduced order modeling framework making use of parallel computing with distributed memory.

In what follows, we first present in Section 2 the problem of interest: a quasistatic evolution of an elastoviscoplastic body under a time-dependent loading. Then, the *a posteriori* reduced order modeling of this problem is introduced in Section 3. Section 4 presents the framework, with the algorithmic choices made to obtain a nonintrusive parallel procedure. Finally, the features and performance of the framework are illustrated in two numerical experiments involving elastoviscoplastic materials in Section 5.

2 High-fidelity model for nonlinear structural mechanics

Consider a structure, denoted Ω , whose boundary $\partial\Omega$ is partitioned as $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ such that $\partial\Omega_D \cap \partial\Omega_N = \emptyset$, see Figure 2.

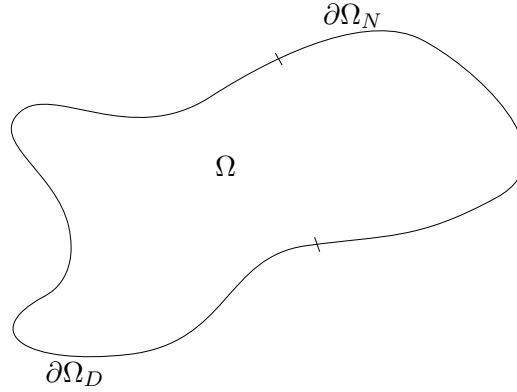


Figure 2: Illustration of the structure of interest Ω .

The structure undergoes a quasi-static time-dependent loading, composed of Dirichlet boundary conditions on $\partial\Omega_D$ in the form of prescribed zero displacement and Neumann boundary conditions on $\partial\Omega_N$ in the form of a prescribed traction T_N , as well as a volumic force f . From [12, Equation (1.1)], under the small deformation assumption, the evolution of the structure over the time interval $t \in [0, T]$ is governed by equations

$$\epsilon(u) = \frac{1}{2} (\nabla u + \nabla^T u) \quad \text{in } \Omega \times [0, T] \quad (\text{compatibility}), \quad (1a)$$

$$\text{div}(\sigma) + f = 0 \quad \text{in } \Omega \times [0, T] \quad (\text{equilibrium}), \quad (1b)$$

$$\sigma = \sigma(\epsilon(u), y) \quad \text{in } \Omega \times [0, T] \quad (\text{constitutive law}), \quad (1c)$$

$$u = 0 \quad \text{in } \partial\Omega_D \times [0, T] \quad (\text{prescribed zero displacement}), \quad (1d)$$

$$\sigma \cdot n = T_N \quad \text{in } \partial\Omega_N \times [0, T] \quad (\text{prescribed traction}), \quad (1e)$$

$$u = 0, y = 0 \quad \text{in } \Omega \text{ at } t = 0 \quad (\text{initial condition}), \quad (1f)$$

where u is the unknown displacement, ϵ is the linear strain tensor, σ is the Cauchy stress tensor, y denotes the internal variables of the constitutive law, and n is the exterior normal on $\partial\Omega$.

Due to the nonintrusive nature of our framework, we do not restrict our high-fidelity model to any particular numerical method: we simply need some discretized approximation of the solution to (1a)-(1f) on some mesh of Ω . However, we present a classical numerical method, using a Galerkin method on a finite element space and a Newton algorithm. Actually, our reduced model will use these tools.

Define $H_0^1(\Omega) = \{v \in L^2(\Omega) \mid \frac{\partial v}{\partial x_i} \in L^2(\Omega), 1 \leq i \leq 3 \text{ and } v|_{\partial\Omega_D} = 0\}$. Denote $\{\varphi_i\}_{1 \leq i \leq N}$, a finite element basis whose span, denoted \mathcal{V} , constitutes an approximation of $H_0^1(\Omega)^3$. A discretized weak formulation obtained by reaction elimination from (1a)-(1f) writes: find $u \in \mathcal{V}$ such that for all $v \in \mathcal{V}$,

$$\int_{\Omega} \sigma(\epsilon(u), y) : \epsilon(v) = \int_{\Omega} f \cdot v + \int_{\partial\Omega_N} T_N \cdot v, \quad (2)$$

that we denote for concision $\mathcal{F}(u) = 0$. A Newton algorithm can be used to solve the nonlinear global equilibrium problem $\mathcal{F}(u) = 0$ at each time step

$$\frac{D\mathcal{F}}{Du}(u^k)(u^{k+1} - u^k) = -\mathcal{F}(u^k), \quad (3)$$

where u^k is the k -th iteration of the discretized displacement field for the considered time-step (the element $u^k \in \mathbb{R}^N$ is identified with $\sum_{i=1}^N u_i^k \varphi_i \in \mathcal{V}$),

$$\frac{D\mathcal{F}}{Du}(u^k)_{ij} = \int_{\Omega} \epsilon(\varphi_j) : \mathcal{K}(\epsilon(u^k), y) : \epsilon(\varphi_i), \quad (4)$$

where $\mathcal{K}(\epsilon(u^k), y)$ is the local tangent operator, and

$$\mathcal{F}(u^k, y)_i = \int_{\Omega} \sigma(\epsilon(u^k), y) : \epsilon(\varphi_i) - \int_{\Omega} f \cdot \varphi_i - \int_{\partial\Omega_N} T_N \cdot \varphi_i. \quad (5)$$

The Newton algorithm stops when the norm of the residual divided by the norm of the external forces vector is smaller than a user-provided tolerance, denoted $\epsilon_{\text{Newton}}^{\text{HFM}}$.

In Equation (3), f , T_N , u^k and y from (5) are known and enforce the time-dependence of the solution. Notice that depending on the constitutive law, the computation of the functions $(u^k, y) \mapsto \sigma(\epsilon(u^k), y)$ and $(u^k, y) \mapsto \mathcal{K}(\epsilon(u^k), y)$ can involve complex ordinary differential equations (ODEs), and therefore advanced numerical procedures. The constitutive laws considered in our applications will be made explicit in Section 5. In this work, these variables are assumed to be computed by an external software.

3 Reduced Order Modeling

In a reduced order model context, the considered high-fidelity model may also depend on some parameter μ , and the collection of all the computed solutions (temporal and/or parametric) is referred to as snapshots. We denote u_s , $1 \leq s \leq N_c$ the snapshots at our disposal, where N_c denotes their number. We consider an *a posteriori* projection-based reduced order model: a Galerkin method is no longer written on a finite element basis $(\varphi_i)_{1 \leq i \leq N}$, but rather on a reduced order basis $(\psi_i)_{1 \leq i \leq n}$, with $n \ll N$, constructed via some operation of the snapshots. We assume the problem at hand to be reducible, in the sense that the obtained approximation is accurate with n small enough for practical computational gains to be obtained. We define the matrix $\Psi \in \mathbb{R}^{N \times n}$, whose columns contain the vectors of the reduced order basis. These reduced order methods are usually composed of an *offline* stage: the learning phase where expensive operators are allowed on the high-fidelity model, and an *online* stage: the exploitation phase where an approximation of the high-fidelity model has to be computed in a complexity which is independent of the dimension N of the finite element basis of the high fidelity problem.

3.1 *Online* stage: the reduced problem

Once the reduced order basis is determined, the reduced Newton algorithm can be constructed

$$\frac{D\mathcal{F}}{Du}(\hat{u}^k)(\hat{u}^{k+1} - \hat{u}^k) = -\mathcal{F}(\hat{u}^k), \quad (6)$$

where \hat{u}^k is the k -th iteration of the reduced displacement field for the considered time-step – the element $\hat{u}^k \in \mathbb{R}^n$ is identified with $\sum_{i=1}^n \hat{u}_i^k \psi_i \in \hat{\mathcal{V}} := \text{Span}(\psi_i)_{1 \leq i \leq n}$,

$$\frac{D\mathcal{F}}{Du}(\hat{u}^k)_{ij} = \int_{\Omega} \epsilon(\psi_j) : \mathcal{K}(\epsilon(\hat{u}^k), y) : \epsilon(\psi_i) \quad (7)$$

and

$$\mathcal{F}(\hat{u}^k, y)_i = \int_{\Omega} \sigma(\epsilon(\hat{u}^k), y) : \epsilon(\psi_i) - \int_{\Omega} f \cdot \psi_i - \int_{\partial\Omega_N} T_N \cdot \psi_i. \quad (8)$$

The reduced Newton algorithm stops when the norm of the reduced residual divided by the norm of the reduced external forces vector is smaller than a user-provided tolerance, denoted $\epsilon_{\text{Newton}}^{\text{ROM}}$. We say that the *online* stage is efficient if (6) can be constructed in computational complexity independent of N .

3.2 *Offline* stage: the three-step procedure

In the light of the context presented so far, we can define the following three-step procedure common to *a posteriori* projection-based reduced order models:

1. (Data generation) This step corresponds to the generation of the snapshots by solving the high-fidelity model. These snapshots represent the data of our reduced order modeling: we search for a small dimension functional subspace that describe correctly this data. Besides, the global behavior of the structure Ω is only accessible through the knowledge of this data.
2. (Data compression) This step corresponds to the generation of the reduced order basis $(\psi_i)_{1 \leq i \leq n}$. Usually, it consists of some post-treatment of the snapshots, by looking for a hidden low-rank structure. The data is compressed into the reduced order basis.
3. (Operator compression) This step corresponds to the additional treatment needed to guarantee the efficiency of the *online* stage, by pre-treatment of the computationally demanding integration over Ω and $\partial\Omega_N$. Notice that in Equation (6), without additional approximation, the numerical integration step will in practice strongly limit the efficiency of the ROM (no interesting speedup with respect to the computation of the high-fidelity model will be obtained in practice). The need to construct such approximation does not depend on the complexity of the high-fidelity model (like linear vs. nonlinear), but on the type of parameter-dependence of the problem. This step is actually needed for all classes of problems reduced by projection-based methods.

In the favorable case of a linear problem with an affine dependence in the parameter μ , for instance $A_{\mu}u = b$, where $A_{\mu} = A_0 + \mu A_1$, the reduced problem $\Psi^T A_{\mu} \Psi \hat{u} = \Psi^T b$ is not assembled in the *online* phase, but the matrices $\Psi^T A_0 \Psi$ and $\Psi^T A_1 \Psi$, and the vector $\Psi^T b$, can be precomputed in the *offline* stage so that the reduced problem can be constructed efficiently, without approximation, by summing two small matrices. The construction of $\Psi^T A_0 \Psi$ and $\Psi^T A_1 \Psi$ in the *offline* stage does constitute the operator compression step.

On the contrary, there exist linear problems that require an additional approximation, and non-linear problem for which the operation compression step can be carried-out exactly. In the former case, take $A_{\mu}u = b$ with $A_{ij} = \int_{\Omega} \nabla(g(x, \mu)\varphi_j(x)) \cdot \nabla\varphi_i(x)$ and $b_i = \int_{\Omega} f(x)\varphi_i(x)$, where u is the unknown, f some known loading and $g(x, \mu)$ a known function whose dependencies in x and μ cannot be separated: the previous trick consisting of precomputing some reduced matrices

cannot be applied, and a treatment is needed to, for instance, approximately separate the dependencies in x and μ of g as $g(x, \mu) \approx \sum_{k=1}^d g_k^a(x) g_k^b(\mu)$. Then, $\Psi^T A_\mu \Psi \approx \sum_{k=1}^d g_k^b(\mu) A_k$ where $(A_k)_{ij} = \int_{\Omega} \nabla (g_k^a(x) \varphi_j(x)) \cdot \nabla \varphi_i(x)$, so that the efficiency of the *online* stage is recovered; the Empirical Interpolation Method has been introduced in [11, 34] for this matter. An example of linear problem in harmonic aeroacoustics with nonaffine dependence in the frequency is available in [15]. Some nonlinearities can be treated without approximation, for example, the advection term in fluid dynamics (with a ROM based on a Galerkin method in our context) only requires the precomputation of an order-3 tensor in the form $\int_{\Omega} \psi_i \cdot (\psi_j \cdot \nabla) \psi_k$, $1 \leq i, j, k \leq n$, see [3] for a reduction of the nonlinear Navier-Stokes equations, with an exact operator compression step. Other examples can be found in structural dynamics with geometric nonlinearities, where order-2 and -3 tensors can be precomputed, see [32, Section 3.2] and [38].

We now present the algorithms chosen to tackle each of the presented step in a nonintrusive and distributed fashion.

4 A distributed nonintrusive framework

The framework consists in a set of python classes. The parallel processing is performed by MPI for Python (mpi4py, see [22, 23, 21]) and the framework can be used indifferently in sequential or in parallel. In the latter case, we suppose that the high-fidelity commercial code provides the mesh, loading and solutions independently for a collection of subdomains Ω_l , $1 \leq l \leq n_d$, which constitute a partition of the structure Ω : $\Omega_l \cap \Omega_{l'} = \emptyset$ for all $l \neq l'$, and $\bigcup_{l=1}^{n_d} \Omega_l = \Omega$, see Figure 3. In our application, Ω_l are the subdomains used in a domain decomposition method, but they can come from any subdivision of the high-fidelity solutions.

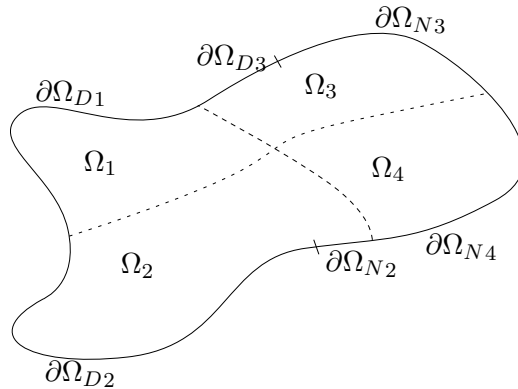


Figure 3: Illustration of the structure of interest in a domain decomposition context.

In this section, we explain how the steps of reduced order modeling described in the previous section are carried-out in a nonintrusive and distributed fashion in our framework.

4.1 Data generation

In our nonintrusive context, we recall that we restrict ourselves to the use of commercial software. Depending on the software, some data may need to be reconstructed. For instance, with Zset [39], we dispose of σ , ϵ and all the internal variables on all the integration points of the mesh. With Ansys Mechanical, these dual quantities are interpolated at the vertices of the mesh. To be as general as possible, we suppose that we only dispose of the displacement field, solution to (1a)-(1f). We can reconstruct the complete history of all dual quantities at all the integration points using a constitutive law solver, based on the knowledge of displacement field. To do so, we interfaced our python library

to the constitutive law API Umat [46], making possible the use of an external reusable plugin. The mesh is converted from the format of the considered code into an in-house unstructured mesh format. The loading is converted in the form of assembled external forces vectors. To do so, an in-house finite element assembler has been implemented in python as well.

4.2 Data compression

Once we dispose of the snapshots u_s , $1 \leq s \leq N_c$, the reduced order basis can be computed. We use the snapshot POD (see Algorithm 1 and [18, 44]), for its good performance in a distributed framework.

Definition 4.1 (ϵ -truncation of the eigendecomposition of a matrix) *It consists in the n largest eigenvalues λ_i , $1 \leq i \leq n$, and associated eigenvectors ξ_i , $1 \leq i \leq n$ of a square matrix, such that $n = \max(n_1, n_2)$, where n_1 and n_2 are respectively the smallest integers such that $\sum_{i=1}^{n_1} \lambda_i \geq (1 - \epsilon^2) \sum_{i=1}^{N_c} \lambda_i$ and $\lambda_{n_2} \leq \epsilon^2 \lambda_0$.*

Algorithm 1 Data compression by snapshot POD.

- 1: Choose a tolerance ϵ_{POD}
 - 2: Compute the correlation matrix $C_{i,j} = \int_{\Omega} u_i \cdot u_j$, $1 \leq i, j \leq N_c$
 - 3: Compute the ϵ_{POD} -truncated eigendecomposition of C : ξ_i and λ_i , $1 \leq i \leq n$
 - 4: Compute the reduced order basis $\psi_i(x) = \frac{1}{\sqrt{\lambda_i N_c}} \sum_{j=1}^{N_c} u_j(x) \xi_{ij}$, $1 \leq i \leq n$
-

Each subdomain can construct its local correlation matrix without any interprocess communication: $C_{i,j}^l = \int_{\Omega_l} u_i|_{\Omega_l} \cdot u_j|_{\Omega_l}$, $1 \leq l \leq n_d$. Then, a distribution and reduction operation is carried-out for each process to dispose of the complete correlation matrix $C = \sum_{l=1}^{n_d} C^l$. Each process can compute the same ϵ -truncated eigendecomposition of C and construct the restriction of the global POD modes to their subdomain as $\psi_i|_{\Omega_l}(x) = \frac{1}{\sqrt{\lambda_i N_c}} \sum_{j=1}^{N_c} u_j|_{\Omega_l}(x) \xi_{ij}$, $1 \leq i \leq n$, $1 \leq l \leq n_d$. In distributed mode, this step requires an interprocess communication of type all-to-all, for an object of size $\frac{1}{2} N_c (N_c + 1)$. In our type of problems, the number of computed snapshots N_c is usually much smaller than the number of dofs N , which means that the interprocess communication time is very limited with respect to the rest of the execution time. For instance, for $N_c = 1000$ snapshots, the packages to be communicated have a volume of approximately 4 MB. The construction of the $L^2(\Omega)$ -scalar product matrices is carried-out by our in-house python library.

4.3 Operator compression

Among the available methods to treat the operator compression step, we chose to construct a reduced quadrature with positive weights, as described below.

In the finite element method, the integrals in the high-fidelity model (3) are computed using (in general exact) quadrature formulas. Applying the chosen quadrature to the reduced internal forces vector yields

$$\hat{F}_i^{\text{int}}(t) = \sum_{e \in E} \sum_{k=1}^{n_e} \omega_k \sigma(\epsilon(\hat{u}), y)(x_k, t) : \epsilon(\psi_i)(x_k), \quad (9)$$

where E denotes the set of elements of the mesh, n_e denotes the number of integration points for the element e , ω_k and x_k are the integration weights and points of the considered element, where the stress tensor $\sigma(\epsilon(\hat{u}), y)$ for the considered reduced solution \hat{u} and internal variables y is seen as a function of space and time and where we recall that Ψ_i , $1 \leq i \leq n$, denote the vectors of the reduced order

basis. We denote also N_G the total number of integration points. The Energy Conserving Sampling and Weighting (ECSW, see [27, 28, 40]) and Empirical Cubature Method (ECM, see [30]) look for approximations of (9) in the form of reduced quadrature. In ECSW, a reduced mesh composed of a small number of elements is considered, while in ECM, a reduced set of integration points is selected globally. In both cases, the associated integration weights are taken positive, as was initially proposed in the computer graphics community [6], so that the reduced quadrature can be written

$$\hat{F}_i^{\text{int}}(t) := \int_{\Omega} \sigma(\epsilon(\hat{u}), y)(x, t) : \epsilon(\psi_i)(x) \approx \sum_{k'=1}^d \hat{\omega}_{k'} \sigma(\epsilon(\hat{u}), y)(\hat{x}_{k'}, t) : \epsilon(\psi_i)(\hat{x}_{k'}), 1 \leq i \leq n, \quad (10)$$

where $\hat{\omega}_{k'} > 0$ and $\hat{x}_{k'}$ are respectively the reduced integration weights and points, where d is the number of terms in the summation which is now called length of the reduced quadrature.

Remark 4.2 (Positivity of the weights) *As noted in [27, Remark 2] for ECSW and [30, Section 1.2.2-2] for ECM (originally motivated in [6]), the positivity of the weights of the reduced quadrature preserves the spectral properties of the operator associated with the physical problem: suppose*

that for all $\mathcal{O} \subset \Omega$, all $v \in H_0^1(\Omega)^3$ such that $v \neq 0$ and all y , $\int_{\mathcal{O}} \epsilon(v) : \mathcal{K}(\epsilon(v), y) : \epsilon(v) > 0$.

Then for all $x \in \Omega$, $\epsilon(v) : \mathcal{K}(\epsilon(v), y) : \epsilon(v)(x) > 0$. Consider a reduced quadrature with positive weights defined by $(\hat{\omega}_{k'}, \hat{x}_{k'})_{1 \leq k' \leq d}$ and a vector of internal variables y , and denote $\mathbb{R}^{n \times n} \ni$

$\frac{D\hat{\mathcal{F}}}{Du}(\hat{v})_{ij} := \sum_{k'=1}^d \hat{\omega}_{k'} (\epsilon(\psi_j) : \mathcal{K}(\epsilon(\hat{v}), y) : \epsilon(\psi_i))(\hat{x}_{k'})$, the global reduced tangent matrix (7) computed

with a reduced quadrature. For all $\hat{v} \in \hat{\mathcal{V}}$ (associated with an element of \mathbb{R}^n), $\hat{v}^t \left(\frac{D\hat{\mathcal{F}}}{Du}(\hat{v}) \right) \hat{v} =$

$\sum_{k'=1}^d \underbrace{\hat{\omega}_{k'}}_{>0} \underbrace{\epsilon(\hat{v}) : \mathcal{K}(\epsilon(\hat{v}), y) : \epsilon(\hat{v})}_{>0}(\hat{x}_{k'}) > 0$, which means that the global reduced tangent matrix computed with a reduced quadrature is positive definite.

We now explain how such reduced quadrature can be computed. Consider the computation of the components of the reduced internal forces when constructing the reduced problem (6): $\int_{\Omega} \sigma(\epsilon(\hat{u}), y) : \epsilon(\psi_i)$, $1 \leq i \leq n$, where we recall that at convergence of the Newton algorithm, \hat{u} is the reduced solution. The goal is to construct a reduced quadrature formula that can accurately approximate this quantity on the snapshots at our disposal, namely $\int_{\Omega} \sigma(\epsilon(u_s), y) : \epsilon(\psi_i)$, $1 \leq i \leq n$, $1 \leq s \leq N_c$. First denote $f_q := \sigma(\epsilon(u_{(q//n)+1}), y) : \epsilon(\psi_{(q\%n)+1})$, where $//$ and $\%$ denote respectively the quotient and the remainder of the Euclidean division. Then denote \mathcal{Z} a subset of $[1; N_G]$ of size d and $J_{\mathcal{Z}} \in \mathbb{R}^{nN_c \times d}$ and $b \in \mathbb{N}^{nN_c}$ such that for all $1 \leq q \leq nN_c$ and all $1 \leq q' \leq N_G$,

$$J_{\mathcal{Z}} = \left(f_q(x_{q'}) \right)_{\substack{1 \leq q \leq nN_c, \\ q' \in \mathcal{Z}}}, \quad b = \left(\int_{\Omega} f_q \right)_{1 \leq q \leq nN_c}. \quad (11)$$

The notion of a convenient reduced quadrature formula implies a trade-off between speed of evaluation and accuracy of the approximation: the fewer reduced integration points we keep, the faster the reduced quadrature is computed but potentially the approximation error is large. On the contrary, if we keep all the integration points, the quadrature is exact but no reduction in the execution time is obtained. The problem of finding the best reduced quadrature formula of length d for the reduced internal forces vector can be written (c.f. [30, Equation (23)])

$$(\hat{\omega}, \mathcal{Z}) = \arg \min_{\hat{\omega}' > 0, \mathcal{Z}' \subset [1; N_G]} \|J_{\mathcal{Z}'} \hat{\omega}' - b\|, \quad (12)$$

where $\|\cdot\|$ stands for the Euclidean norm. As indicated in [27, Section 5.3], citing [4], taking the length of the reduced quadrature formula in the objective function of the optimization leads to a NP-hard

problem. Approximating the solution to this optimization problem has been intensively addressed by the signal processing community, where different variants of greedy algorithms have been proposed. In [27] a Sparse NonNegative Least-Squares (NNLS) algorithm, proposed in [41], is used to reach a solution in reasonable time. Here, we consider a modification of the Nonnegative Orthogonal Matching Pursuit algorithm, see [50, Algorithm 1] and Algorithm 2 below, a variant of the Matching Pursuit algorithm [35] adapted to the nonnegative requirement.

Algorithm 2 Nonnegative Orthogonal Matching Pursuit.

Input: J, b , tolerance $\epsilon_{\text{Op.comp}}$.

Initialization: $\mathcal{Z} = \emptyset, k' = 0, \hat{\omega} = 0$ and $r_0 = b$

- 1: **while** $\|r_{k'}\|_2 > \epsilon \|b\|_2$ **do**
- 2: $\mathcal{Z} \leftarrow \mathcal{Z} \cup \max \text{index} \left(J_{[1:N_G]}^T r_{k'} \right)$
- 3: $\hat{\omega} \leftarrow \arg \min_{\hat{\omega}' > 0} \|b - J_{\mathcal{Z}} \hat{\omega}'\|_2$
- 4: $r_{k'+1} \leftarrow b - J_{\mathcal{Z}} \hat{\omega}$
- 5: $k' \leftarrow k' + 1$
- 6: **end while**
- 7: $d \leftarrow k'$
- 8: $\hat{x}_k := x_{\mathcal{Z}_k}, 1 \leq k \leq d$

Output: $\hat{\omega}_k, \hat{x}_k, 1 \leq k \leq d$.

In this work, we consider the simple heuristics for the operator compression, introduced in [28, Section 3.3], and well adapted to a distributed context: consider a subdomain $\Omega_l, 1 \leq l \leq n_d$, and apply the Nonnegative Orthogonal Matching Pursuit (Algorithm 2) to derive a reduced quadrature formula to approximate the integration of the reduced internal forces vector locally on Ω_l :

$$\sum_{k'=1}^{d^l} \hat{\omega}_{k'}^l \sigma(\epsilon(\hat{u}), y)(\hat{x}_{k'}^l, t) : \epsilon(\psi_i)(\hat{x}_{k'}^l) \approx \int_{\Omega_l} \sigma(\epsilon(\hat{u}), y) : \epsilon(\psi_i), \quad (13)$$

with $1 \leq i \leq n$, $\hat{\omega}_{k'}^l > 0$, $\hat{x}_{k'}^l \in \Omega_l, 1 \leq k' \leq d^l, d^l \in \mathbb{N}$. Then, an approximation of the integration over the complete domain Ω can be obtained by simply summing the independently constructed local reduced quadrature formulas, where we recall that $\Omega_l \cap \Omega_{l'} = \emptyset$ for all $l \neq l'$, and $\bigcup_{l=1}^{n_d} \Omega_l = \Omega$. This procedure is suboptimal in the sense that for a given accuracy level, it will probably choose more integration points for the reduced quadrature than considering the complete domain. However, it is executed independently in each subdomain without any communication between the processes.

Remark 4.3 ((Discrete) Empirical Interpolation Method - (D)EIM) *(D)EIM is an offline-online procedure that computes an approximation for a separated variable representation of two-variable functions. The offline stage contains a greedy procedure that selects, at each step, the values for each variable that maximizes the error of the currently constructed approximation, see [11, 34] for details on the procedure. Consider the computation of the reduced internal forces vector $\hat{F}_i^{\text{int}}(t) :=$*

$$\int_{\Omega} \sigma(\epsilon(\hat{u}), y)(x, t) : \epsilon(\psi_i)(x), 1 \leq i \leq n. \text{ Consider } \sigma(\epsilon(\hat{u}), y) \text{ as a function of time } t \text{ and space } x,$$

the EIM enables to recover an approximation in the form $\sigma(\epsilon(\hat{u}), y)(x, t) \approx \sum_{k=1}^d \alpha_k(t) \sigma(\epsilon(u), y)(x, t_k)$,

where $\alpha_k(t)$ is the solution to $\sum_{l=1}^d B_{k,l} \alpha_l(t) = \sigma(\epsilon(\hat{u}), y)(x, t_k)$, with $B \in \mathbb{R}^{d \times d}$ a matrix constructed

during the offline stage of the EIM, and where we recall that u is the solution to the high-fidelity model, and \hat{u} is the reduced solution under construction. The sets $(x_l)_{1 \leq l \leq d}$ and $(t_l)_{1 \leq l \leq d}$, are referred to as magic points. Making use of the EIM approximation, the efficiency of the online stage is recovered using

$$\hat{F}_i^{\text{int}}(t) \approx \sum_{k=1}^d \alpha_k(t) \int_{\Omega} \sigma(\epsilon(u), y)(x, t_k) : \epsilon(\psi_i)(x), \text{ where the integrals involve the high-fidelity solution}$$

u at fixed time steps t_l and can therefore be precomputed. Notice that for clarity of the presentation, we considered σ as a scalar quantity, but the generalization to a 3×3 order-2 tensor can be made by adding all the components in the spatial sampling. While the EIM constructs an approximation of the integrand $\sigma(\epsilon(\hat{u}), y) : \epsilon(\psi_i)$, the Discretized Empirical Interpolation Method (DEIM) constructs an approximation of the integrated vectors \hat{F}_i^{int} directly: the method has been proposed in [19] and applied for nonlinear elasticity in [42]. (D)EIM leads to a reduced problem that is not of Galerkin type, and no stabilization is guaranteed. In practice in our simulations, we did not manage to obtain stable POD-EIM schemes for elastoviscoplasticity, except when we stopped the greedy selection of the offline stage of the EIM very early and when we kept only the first iteration of the reduced Newton algorithm (6). A stabilization procedure has been proposed in [8].

Remark 4.4 (Selection of the reduced integration points: robustness and sparsity) *As stated before, the derivation of the reduced quadrature is a well-known but difficult problem. From [6, Section 5.1], the positivity constraint on the weight has an additional advantage: it protects the learning procedure against over-fitting: allowing the weights to take negative values, we can still continue to add integration points to increase the accuracy of the reduced quadrature on the learning data. However, doing so would increase the complexity of the quadrature formula, and eventually deteriorate the numerical stability of the quadrature (i.e. its ability to correctly integrate new data, outside the learning data). With the positivity constraint of the weight, we cannot reach any accuracy for the quadrature formula, but a low rank reconstruction is implicitly imposed, leading to sparse quadrature formula. Consider line 4 of Algorithm 2: this is a linear regression model in absence of noise where the number of unknowns is (potentially much) larger than the number of observation. From [45], in this precise context, the nonnegative constraints alone are sufficient to lead to a sparse solution, and the performance of the nonnegative least square with regard to prediction and estimation is comparable to that of the lasso. From this observation, we tried a simple heuristics for computing a reduced quadrature formula: taking random sets of candidate reduced integration points, and associated reduced integration weights solving line 4 of Algorithm 2, we can choose the one that minimizing the integration error. The sparsity is automatically obtained, and the performance of the procedure was globally comparable with the Nonnegative Orthogonal Matching Pursuit – sometimes much more accurate solution were obtained. However, since it was not clear how to choose the size of the random sets and the performance changed from one execution to the other, we kept the Nonnegative Orthogonal Matching Pursuit for our numerical applications.*

4.4 *Online* stage and posttreatment

The *online* stage consists in applying the Galerkin method on the POD basis and treating the nonlinearity using a reduced Newton algorithm (see Section 3.1), where the reduced quadrature formulas are used to compute the integrals. In our implementation, the *online* stage can be executed in sequential or in parallel, with or without the same number of processors as the *offline* stage processing.

Depending on the application, the dual quantities (stress and constitutive law internal variables) may need to be reconstructed on the high-fidelity mesh. We propose a procedure using DEIM and Gappy-POD tools locally on each subdomains, described in Algorithm 3, see [26] for the original presentation of the Gappy-POD.

We now give some comments on Algorithm 3. Recall that during the *online* resolution of the reduced order model, the constitutive law is solved for a set of reduced integration points selected during the operator compression step. In practice in our case, applying the Gappy-POD using the information at the reduced integration points did not provide satisfying reconstructions at the extrapolated cycles for the cumulated plasticity field, which is of prime interest for lifetime computations. It has been noted in [49] that the mask used for the Gappy-POD (in our case, the set of reduced integration points) plays an important role in the quality of the reconstruction, namely one has to ensure that the matrix M is invertible. We now explain how our choices in Algorithm 3 ensure the invertibility of the matrix M .

Proposition 4.5 (Well-posedness of the reconstruction) *The linear system considered in the online stage of Algorithm 3 is invertible.*

Algorithm 3 Dual quantity reconstruction (DEIM and Gappy-POD) of the cumulated plasticity p .

Offline stage: consider subdomain Ω_l , $1 \leq l \leq n_d$

- 1: Choose a tolerance $\epsilon_{\text{Gappy-POD}}$
- 2: Apply the snapshot POD (Algorithm 1) on the high-fidelity snapshots p_s^l , $1 \leq s \leq N_c$ to obtain the orthonormal vectors $\psi_i^{p^l}$, $1 \leq i \leq n^l$
- 3: Apply the EIM to the collection of vectors $\psi_i^{p^l}$, $1 \leq i \leq n^l$ to select n^l distinct indices $\{j_k^l\}_{1 \leq k \leq n^l}$ ($1 \leq j_k^l \leq N_G$), and complete (without repeat) this set of indices by the indices of the integration points of the local reduced quadrature formula to obtain the set $\{j_k^l\}_{1 \leq k \leq m^l}$ ($1 \leq j_k^l \leq N_G$), where $m^l \leq n^l + d^l$
- 4: Construct the matrix $M^l \in \mathbb{N}^{n^l \times n^l}$ such that $M_{i,j}^l = \sum_{k=1}^{m^l} \psi_i^{p^l}(x_{j_k^l}) \psi_j^{p^l}(x_{j_k^l})$ (Gappy scalar product of the POD modes)

Online stage: consider a time t and a subdomain l , $1 \leq l \leq n_d$

- 1: Construct $b \in \mathbb{R}^{m^l}$, where b_k is the *online* prediction of p^l at time t and the indices j_k^l (from the evaluation of the constitutive law solver at the corresponding integration point)
 - 2: Solve the (small) linear system: $Mw = b$
 - 3: Compute the reconstructed value for p^l on the complete subdomain Ω^l with $\sum_{i=1}^{n^l} w_i \psi_i^{p^l}$
-

Proof 1 *The EIM applied to the collection of vectors $\psi_i^{p^l}$, $1 \leq i \leq n^l$, produced the set of integration points j_k^l , $1 \leq k \leq n^l$, and a permutation ζ^l of $\{1, 2, \dots, n^l\}$, since we allow the greedy algorithm in the EIM to compute n^l iterations. Notice that this is possible since the vectors $\psi_i^{p^l}$, $1 \leq i \leq n^l$ is a free family. Define now the matrix $F \in \mathbb{R}^{n^l \times n^l}$ such that $F_{i,k} = \psi_{\zeta^l(i)}^{p^l}(x_{j_k^l})$. From [16, Theorem 1.2], F is invertible. In particular, the matrix $\left(\psi_i^{p^l}(x_{j_k^l})\right)_{1 \leq i \leq n^l, 1 \leq k \leq m^l}$ (where we consider the indices of the integration points as well as the indices selected by EIM) is of rank n^l , and M is then invertible.*

In Algorithm 3, the construction of the POD basis and the EIM algorithm selecting n indices represents the DEIM part. In practice, we used a recently proposed implementation: the QDEIM [25], which consists in a QR decomposition with column pivoting applied to the collection of the POD modes $\psi_i^{p^l}$, $1 \leq i \leq n^l$. The part of Algorithm 3 consisting in solving the (small) linear system and recombining the solution with the POD modes $\psi_i^{p^l}$, $1 \leq i \leq n^l$ corresponds to the Gappy-POD treatment, for which we have ensured well-posedness. The DEIM part enables to add, in the reduced integration points set, new points associated with the approximation of the cumulated plasticity field, which improved the quality of the reconstruction for this field.

5 Numerical applications

In Section 5.1 is presented an application using the commercial code Ansys, to illustrate the nonintrusivity capabilities of our framework. In Section 5.2, we consider our main application of interest: the cyclic extrapolation of an elastoviscoplastic structure using a reduced order model. In what follows, ROM means Reduced Order Model, and HFM means High-Fidelity Model.

Two constitutive laws are considered in the numerical applications:

- (elas) temperature-dependent cubic elasticity and isotropic thermal expansion: for this law, there no internal variable to compute, and the constitutive law is $\sigma = \mathcal{A} : (\epsilon - \epsilon^{\text{th}})$, where $\epsilon^{\text{th}} = \alpha(T - T_0)I$, with I the second-order identity tensor and α the thermal expansion coefficient in MPa.K⁻¹ depending on the temperature, and where \mathcal{A} , the elastic stiffness tensor, does not

depend on the solution u and is defined in Voigt notations by

$$\mathcal{A} = \begin{pmatrix} y_{1111} & y_{1122} & y_{1122} & 0 & 0 & 0 \\ y_{1122} & y_{1111} & y_{1122} & 0 & 0 & 0 \\ y_{1122} & y_{1122} & y_{1111} & 0 & 0 & 0 \\ 0 & 0 & 0 & y_{1212} & 0 & 0 \\ 0 & 0 & 0 & 0 & y_{1212} & 0 \\ 0 & 0 & 0 & 0 & 0 & y_{1212} \end{pmatrix}, \quad (14)$$

where the temperature T is given by the thermal loading, $T_0 = 20^\circ\text{C}$ is a reference temperature and the coefficients y_{1111} , y_{1122} and y_{1212} (Young moduli in MPa) depend on the temperature.

- (evp) Norton flow with nonlinear kinematic hardening and constant isotropic hardening: for this law, the elastic part is given by $\sigma = \mathcal{A} : (\epsilon - \epsilon^{\text{th}} - \epsilon^P)$, where \mathcal{A} and ϵ^{th} are the same as the (elas) law, ϵ^P is the plastic strain tensor, and the viscoplastic part requires solving the following ODE: $\dot{\epsilon}^P = \dot{p} (n_r - D\epsilon^P)$, where p is the cumulated plasticity such that $\dot{p} = \left\langle \frac{f_r}{K} \right\rangle^m$, with yield criterion $f_r := s_{\text{eq}} - R_0 \geq 0$, where $s := (\sigma - \frac{1}{3}\text{Tr}(\sigma)I) - C\epsilon^P$. In what precedes, $n_r := \frac{3}{2} \frac{s}{s_{\text{eq}}}$ is the normal to the yield surface (with $\|\cdot\|$ the Euclidean norm and Tr the trace operator), $\langle \cdot \rangle$ denotes the positive part operator, $\cdot_{\text{eq}} = \sqrt{\frac{3}{2} \|\cdot\|^2}$. The coefficients C and D are hardening material coefficients (in MPa), K is the Norton material coefficient (in MPa), m is the Norton exponential material coefficient (dimensionless) and R_0 is the isotropic hardening material coefficient (in MPa): they all depend on the temperature. The internal variables considered here are ϵ^P and p , and the ODE is completed with the initial conditions $\epsilon^P = 0$ and $p = 0$ at $t = 0$.

5.1 Rotating bike crank using Ansys Mechanical

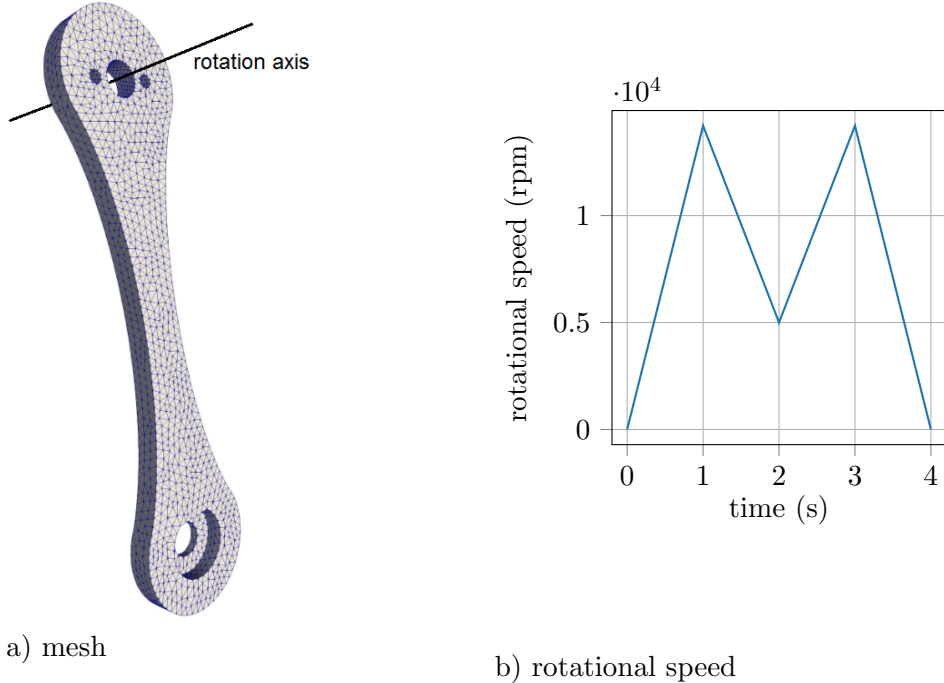


Figure 4: a) Mesh for the bike crank, b) Rotational speed with respect to time.

To illustrate the nonintrusivity capabilities of our framework, we consider a simple test case, where we only assess the ability of the reduced order model to recompute the same loading as the high-fidelity computation used in the data generation step. The structure is maintained at a temperature of 20°C and is subject to very strong rotational effects, see Figure 4 for a representation of the mesh and a description of the loading. The loading is not physical: it has been chosen to trigger the

| | |
|----------------------------------|---------------------------------------------------------------------------------------|
| number of dofs | 101'952 |
| number of (quadratic) tetrahedra | 21'409 |
| number of integration points | 107'045 |
| number of time steps | 30 per cycle |
| constitutive law | evp (Norton flow with nonlinear kinematic hardening and constant isotropic hardening) |

Table 1: Characteristics for the bike crank test case.

strongest possible strain-induced elastoviscoplastic effects: a slightly stronger rotation makes Ansys fail to compute the result, as we reach the limit of the strength of the material. Doing so, we challenge the ability of the framework to deal with a strongly nonlinear problem.

The characteristics of the test-case are provided in Table 1. Ansys only provides the dual-quantities on the nodes of the mesh, by interpolating them from their values at the integration points. To access the high-fidelity quadrature scheme (in order to compute an accurate reduced quadrature in the operator compression step), we recompute the evolution of the dual-quantities at the integration points using the constitutive law solver and the global equilibrium at each time step given by Ansys and the finite element interpolation by our framework, before the data compression step. This test case is computed in sequential for the data generation, data compression and operator compression steps. The *online* stage is computed in sequential as well.

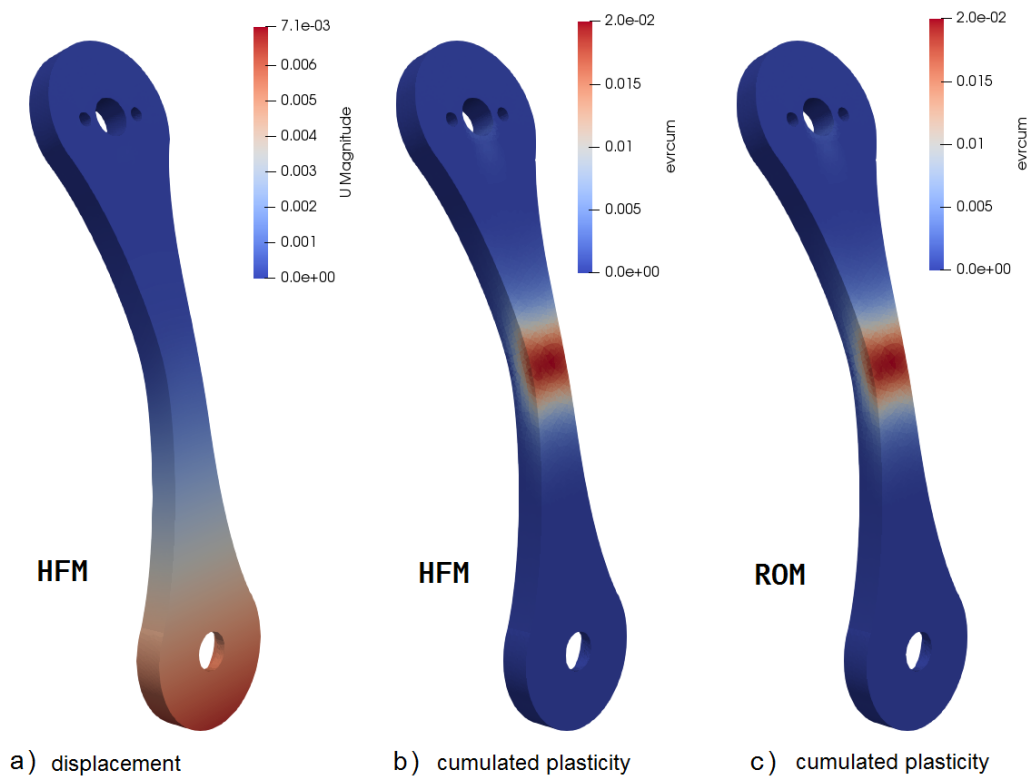


Figure 5: Solution fields on the deformed mesh (amplified 5 times) at $t = 2.2s$: a) displacement for the HFM, b) cumulated plasticity for the HFM, c) cumulated plasticity for the ROM.

In Figure 5 are represented some solution fields at time $t = 2.2s$. We notice that the reduced order model has correctly reconstructed the plasticity field, then illustrating the ability of the framework to reduced strongly nonlinear problems computed by commercial software, i.e. in a nonintrusive fashion.

We now present a meaningful industrial test case with a physical loading and strain and temperature-induced elastoviscoplastic effects, and such that the reduced model is not limited to reproduce only the high-fidelity computation.

5.2 Elastoviscoplastic cycle extrapolation of a high-pressure turbine blade using Zset

In this section, we use the presented framework for the application of industrial interest motivated in the introduction, namely the efficient computation of a large number of cycles of a high-pressure turbine blade undergoing a loading representing aircraft missions (either take-off-cruise-landing or test-bench cycles). The lifetime prediction being a rapidly computed post-treatment of the cyclic simulation, we focus our task to the efficient computation of many cycles.

As numerical procedure, we propose to compute the first cycle using a high-fidelity simulation, then construct a reduce-order model to compute a large number of cycles. Notice that this is not the classical context of reduced order modeling, since the ROM is used for the acceleration of a single computation. We need efficient computations for the complete numerical sequence, since, unlike many-queries contexts, long run-times for the *offline* stage cannot be compensated by an intensive use of the ROM. As a consequence, we give the following definition for the speedup:

Definition 5.1 *The speedup is defined as the ratio between the run-time for the high-fidelity code and the run time of the complete numerical procedure, including the three steps of the offline stage: data generation (first cycle high-fidelity cycle), data compression and operator compression, as well as the online stage.*

With this definition of the speedup, we can evaluate the opportunity of using reduced order modeling tools to accelerate a single computation.

We consider a model of high-pressure turbine blade, with 4 internal cooling channels. The lower part, called the foot, is modeled by an elastic material (even if plasticity occurs here, it is not of interest for computing the lifetime of the blade) whereas the upper part is modeled by an elastoviscoplastic law. The high-fidelity computation is carried-out in parallel on 48 processors using Zset [39] with an AMPFETI solver (Adaptive MultiPreconditioned FETI [13]), see Figure 6. Notice that in order to obtain a fast complete procedure, we also spend efforts to accelerate the computation of the high-fidelity model, which is not quantified by our definition of the speedup. The loading is represented in Figure 7: the axis of rotation is directed along the x-axis (and located far below the blade) and the rotational speed consists in a linear increase followed by a linear decrease; the temperature field starts by a uniform 20°C at $t = 0$ s, is linearly interpolated node by node up to the temperature field represented in Figure 7 b) at $t = 20$ s and decreases again linearly up to a uniform 20°C at $t = 40$ s. The characteristics of the test-case are provided in Table 2.

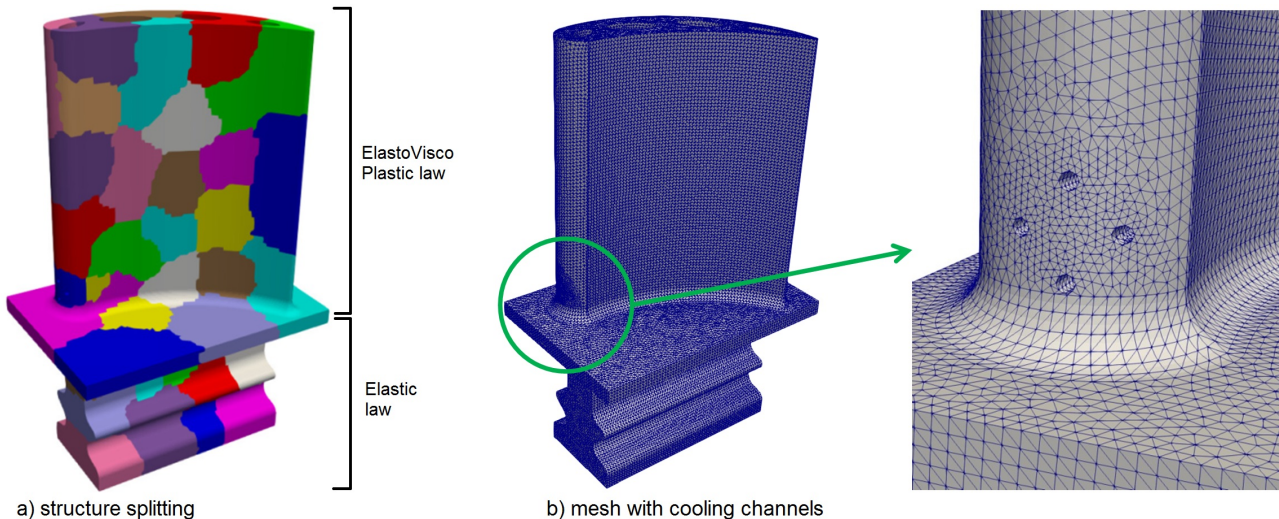
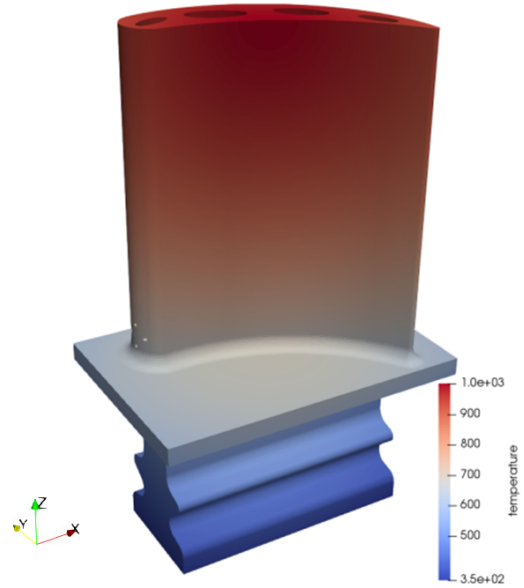
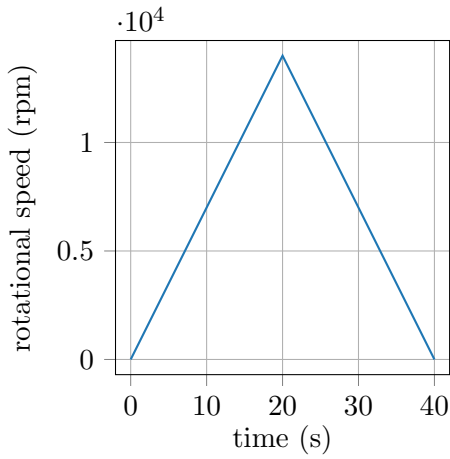


Figure 6: a) Structure split in 48 subdomains - the top part of the blade's material is modeled by an elastoviscoplastic law and the foot's one by an elastic law, b) Mesh for the high-pressure turbine blade with a zoom on the cooling channels.

Details are provided in Table 3 on the computational procedure. In particular, the computations for all the steps are done in parallel with distributed memory and using MPI for the communications



a) rotational speed

b) temperature field at max loading

Figure 7: a) Rotational speed with respect to time, b) Temperature field at maximum loading ($t = 20$ s).

| | |
|----------------------------------|---------------------------------------------------------------------------------------|
| number of dofs | 4,892'463 |
| number of (quadratic) tetrahedra | 1'136'732 |
| number of integration points | 5'683'660 |
| number of time steps | 40 per cycle |
| constitutive law for the foot | elas (temperature-dependent cubic elasticity and isotropic thermal expansion) |
| constitutive law for the blade | evp (Norton flow with nonlinear kinematic hardening and constant isotropic hardening) |

Table 2: Characteristics for the high-pressure turbine blade test case.

between processors (48 processors within 2 nodes). Notice that in our framework, even the visualization is done in parallel with distributed memory using a parallel version of Paraview [2, 10]. The choices for the stopping criteria are made rather empirical $\epsilon_{\text{Newton}}^{\text{HFM}}$ and $\epsilon_{\text{Newton}}^{\text{ROM}}$ are chosen equal to 10^{-6} , $\epsilon_{\text{POD}} = 10^{-7}$ and $\epsilon_{\text{Gappy-POD}} = 10^{-7}$ provide the more accurate possible compression (below this value for the data compression, the representation of the snapshots on the basis is not more accurate), $\epsilon_{\text{Op.comp.}} = 10^{-5}$: below this value, the convergence of the Distributed NonNegative Orthogonal Matching Pursuit is very slow, and even never reached in some of our numerical tests, see Remark 4.4. The complete procedure takes 6h31min to compute an approximation for the 100 cycles, whereas the high-fidelity model would have taken 9 days and 14 hours, which corresponds to a speedup of approximately 35. While this speedup may appear low, we recall that all the *offline* stage is taken into account, and that we used a state-of-the-art AMPFETI solver for the high-fidelity reference. Using a standard commercial software for the high-fidelity reference would have lead to a more impressive speedup. With this test-case, we illustrate that a computation that takes many days even with state-of-the-art high-fidelity solvers can be done overnight, which is of great practical interest in design phases of a mechanical part in an industrial context.

The set of integration points selected for the operator compression step is represented in Figure 8.

To assess the accuracy of the ROM and its ability to extrapolate the cyclic loading, the HFM has been run on 10 cycles, for a duration of 23h20min. We recall that the ROM is computed from the information of the first cycle of the HFM. To comply with a real-life situation, the HFM has not been computed up to the 100th cycle, the assessment of the extrapolation quality being made by comparing

| step | algorithm | wall-time (CPUs) |
|-------------------------------------------------|---------------------------------------------------------------------------------------------|------------------|
| Data generation | AMPFETI solver in Zset, $\epsilon_{\text{Newton}}^{\text{HFM}} = 10^{-6}$ | 2h18min (48) |
| Data compression | Distributed Snapshot POD, $\epsilon_{\text{POD}} = 10^{-7}$ | 9min (48) |
| Operator compression | Distributed NonNegative Orthogonal Matching Pursuit, $\epsilon_{\text{Op.comp.}} = 10^{-5}$ | 44min (48) |
| Online computation | ROM for 100 cycles, $\epsilon_{\text{Newton}}^{\text{ROM}} = 10^{-6}$ | 3h13min (48) |
| Reconstruction of σ , ϵ and p | Distributed Gappy-POD, $\epsilon_{\text{Gappy-POD}} = 10^{-7}$ | 7min (48) |

Table 3: Description of the computational procedure (p denotes the cumulated plasticity).

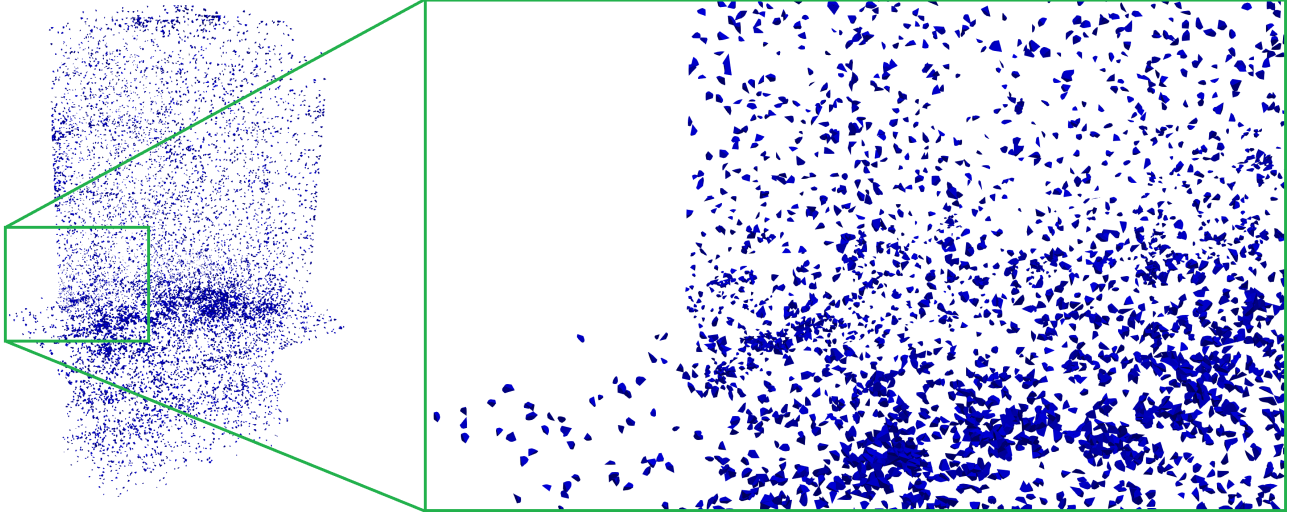


Figure 8: Set of integration points selected for the operator compression step: each integration point is materialized by a polyhedron obtained by Voronoi tessellation of the mesh, where the seeds are the integration points. Selecting all the integration points would lead to representing the complete structure. We notice smaller polyhedra around the internal cooling channels due to a refined mesh.

the solutions at the 10th cycle. In Figure 9, we compare the displacement and the stress at the middle of the 1st and 10th cycles. For these quantities, the extrapolation works well, but the solution is not changing much between the 1st and 10th cycles. In Figure 10, are compared the displacement on the deformed mesh amplified 1000 times at the end of the cycles, when the loading is released. The residual displacement is well predicted. In Figures 11 and 12 are represented the stress and cumulated plasticity fields at the end of the cycles. These quantity undergo a visible evolution during the cycles and are of prime interest for the lifetime predictions. By comparing the zones circled in green, we notice that the ROM correctly extrapolates the evolution of the fields between the 1st and 10th cycles. In the zones circled in red, we notice some discontinuities of the Gappy-POD reconstruction between the subdomains. This is due to the independent local treatments of the Gappy-POD procedure. The representation with saturated colors helps the visualization but amplifies the discontinuities for the small values. In Figure 13 are represented difference maps between the ROM and the HFM for the displacement and the cumulated plasticity. Finally, in Figure 14 are represented cyclic evolution of σ_{33} with respect to ϵ_{33} and cumulated plasticity with respect to time, for two integration points. Although the first cycle appears very well approximated by the ROM, errors are accumulated in the following cycles.

Conclusion and outlook

In this work, we introduce a framework to reduce nonlinear structural mechanics problems, in a non-intrusive fashion and in parallel with distributed memory. Algorithmic choices have been justified to obtain an efficient complete procedure, with a particular attention to optimize the *offline* stage as well. The nonintrusive features of the framework are illustrated by reducing an Ansys Mechanical computation, and its large scale applicability is illustrated on a problem of industrial interest by constructing a reduced order model to extrapolate a strong cyclic loading on a 5 million dofs elastoviscoplastic high-pressure turbine blade on 48 processors.

In this second application, the accuracy can be further increased during the extrapolated cycles

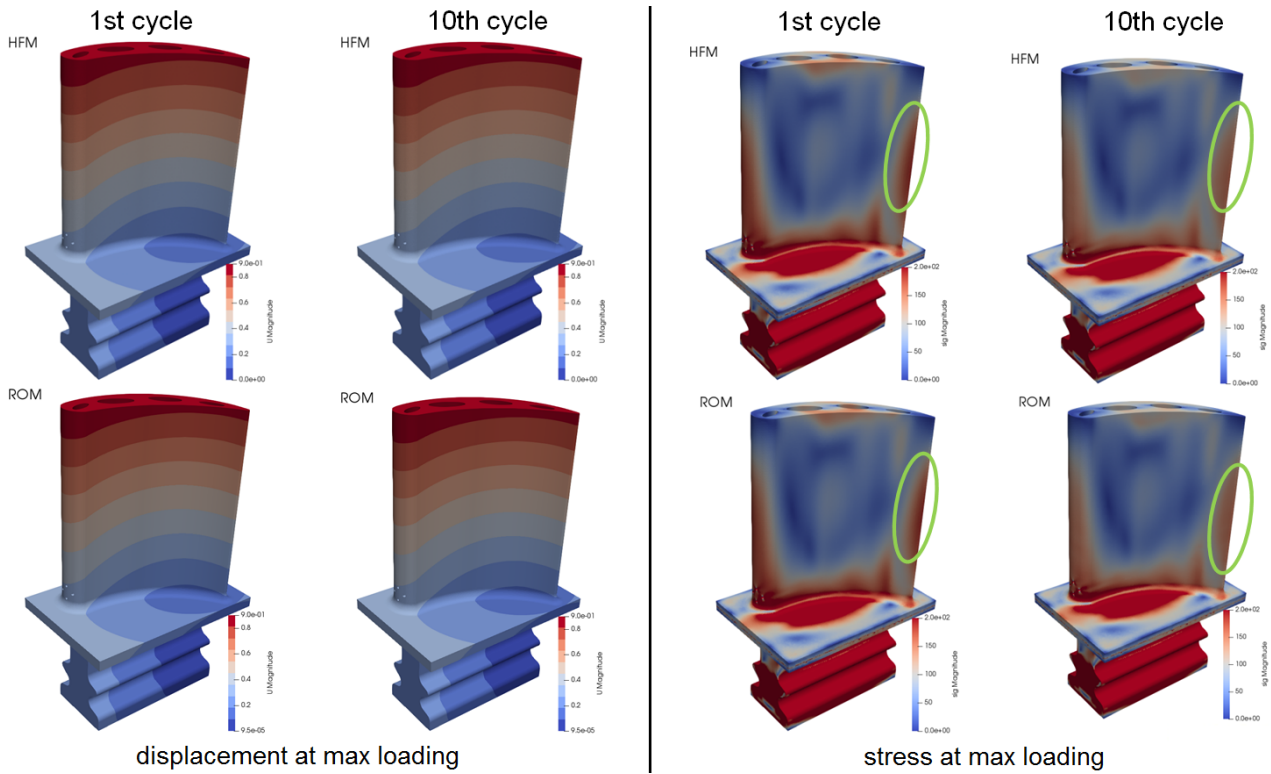


Figure 9: Comparison between HFM and ROM at the middle of the 1st and 10th cycles (maximum loading). Left-hand side: magnitude of the displacement, right-hand side: magnitude of the stress.

by, for instance, running the high-fidelity model each time some error indicator becomes too large, and updating the POD basis using the incremental POD. The heuristic used in the operator compression step was chosen to favor scalability; it is desirable to also guarantee some accuracy properties. Besides, the current implementation requires to load in memory, locally in each subdomain, the high-fidelity data for all time steps. Even if we can always increase the number of computer nodes to increase the available memory, it is desirable to devise memory-free strategies, where the data is not simultaneously loaded in memory for all the time steps. This would enable to reduce computations with a very large number of snapshots.

References

- [1] File:gaturbineblade.svg. Wikipedia, the free encyclopedia, image under the Creative Commons Attribution-Share Alike 3.0 Unported license, 2009.
- [2] J. Ahrens, B. Geveci, and C. Law. Paraview: An end-user tool for large data visualization, visualization handbook. Elsevier, 2005.
- [3] N. Akkari, R. Mercier, G. Lartigue, and V Moureau. Stable pod-galerkin reduced order models for unsteady turbulent incompressible flows. *55th AIAA Aerospace Sciences Meeting. Grapevine, Texas.*, 2017.
- [4] Edoardo Amaldi and Viggo Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1-2):237–260, 1998.
- [5] S. Amaral, T. Verstraete, R. Van den Braembussche, and T. Arts. Design and optimization of the internal cooling channels of a high pressure turbine blade—part i: Methodology. *Journal of Turbomachinery*, 132, 2010.
- [6] S. S. An, T. Kim, and D. L. James. Optimizing cubature for efficient integration of subspace deformations. *ACM transactions on graphics (TOG)*, 27(5):165, 2008.

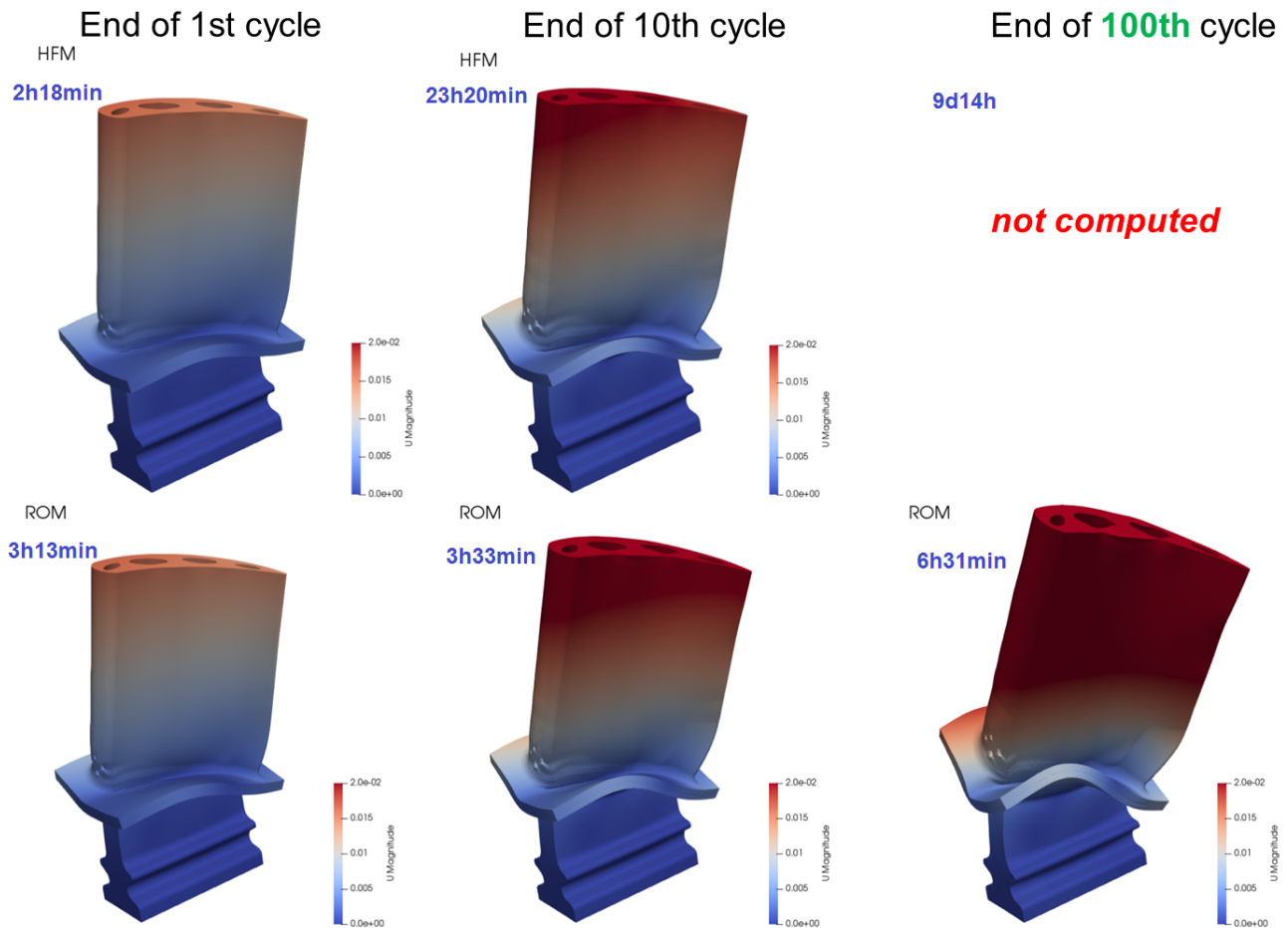


Figure 10: Magnitude of the displacement fields on a deformed mesh (amplified 1000 times) for the HFM and the ROM at the end of the 1st and 10th cycles, and for the ROM at the end of the 100th cycle (minimum loading).

- [7] C. V. Ananth and D. G. Kleinbaum. Regression models for ordinal responses: a review of methods and applications. *International Journal of Epidemiology*, 26(6):1323–1333, 1997.
- [8] J. P. Argaud, B. Bouriquet, H. Gong, Y. Maday, and O. Mula. Stabilization of (g)eim in presence of measurement noise: Application to nuclear reactor physics. In Marco L. Bittencourt, Ney A. Dumont, and Jan S. Hesthaven, editors, *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016*, pages 133–145, Cham, 2017. Springer International Publishing.
- [9] C. Audouze, F. De Vuyst, and P. B. Nair. Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations. *Numerical Methods for Partial Differential Equations*, 29(5):1587–1628, 2013.
- [10] U. Ayachit. The paraview guide: A parallel visualization application. Kitware, 2015.
- [11] M. Barrault, Y. Maday, N.-C. Nguyen, and A. T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique*, 339(9):667 – 672, 2004.
- [12] M. Bonnet, A. Frangi, and C. Rey. *The Finite Element Method in Solid Mechanics*. Collana di istruzione scientifica: Serie di ingegneria meccanica. McGraw-Hill Education, 2014.
- [13] C. Bovet, A. Parret-Fréaud, N. Spillane, and P. Gosselet. Adaptive multipreconditioned feti: Scalability results and robustness assessment. *Computers & Structures*, 193:1 – 20, 2017.
- [14] P. Caron and O. Lavigne. Recent studies at onera on superalloys for single crystal turbine blades. *AerospaceLab*, pages 1–14, 2011.

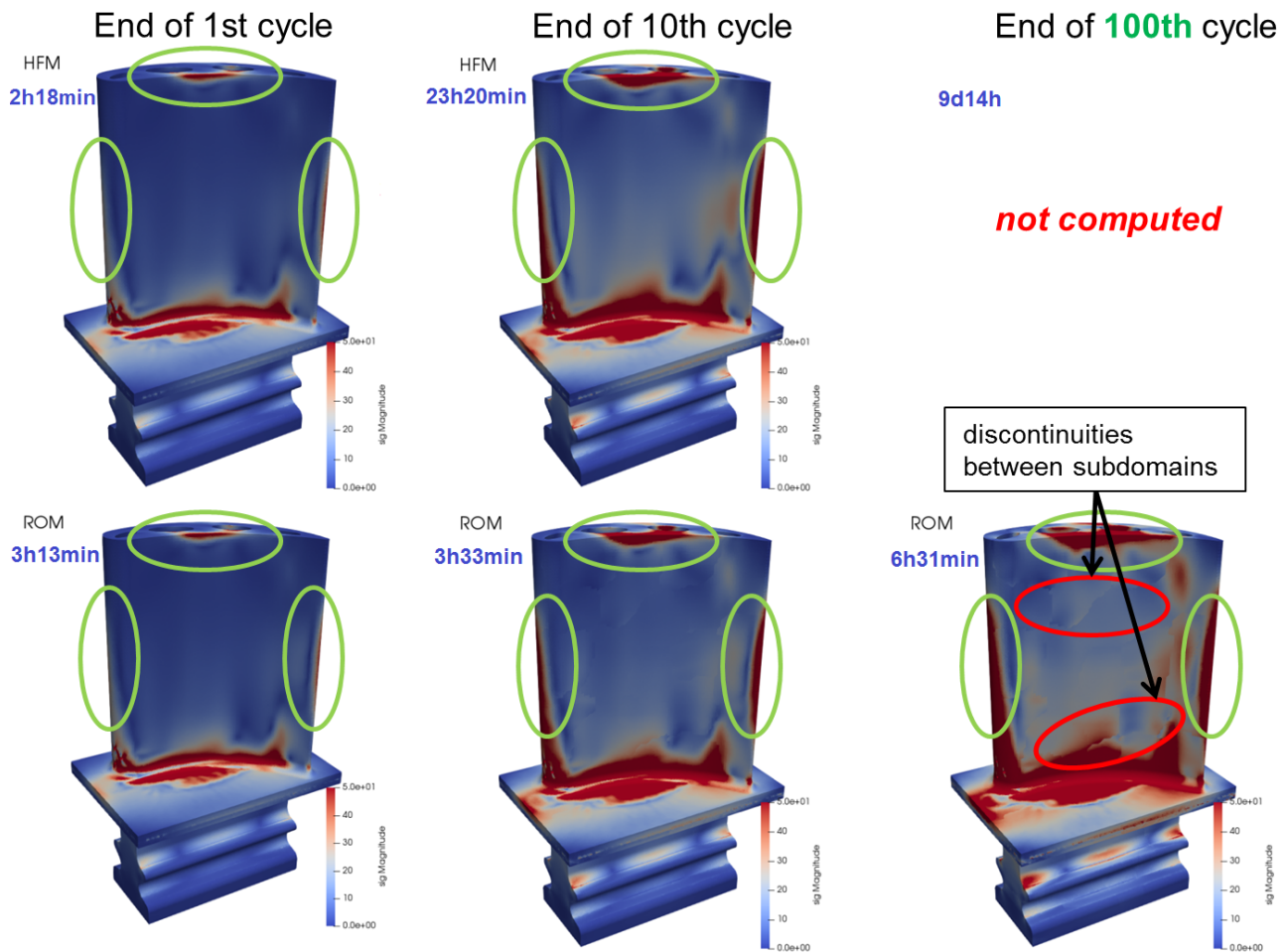


Figure 11: Magnitude of the stress fields for the HFM and the ROM at the end of the 1st and 10th cycles, and for the ROM at the end of the 100th cycle (minimum loading) - with saturated colors.

- [15] F. Casenave, A. Ern, and T. Lelièvre. A nonintrusive reduced basis method applied to aeroacoustic simulations. *Advances in Computational Mathematics*, 41(5):961–986, Oct 2015.
- [16] F. Casenave, A. Ern, and T. Lelièvre. Variants of the empirical interpolation method: Symmetric formulation, choice of norms and rectangular extension. *Applied Mathematics Letters*, 56:23–28, 2016.
- [17] R. Chakir and Y. Maday. Une méthode combinée d’éléments finis à deux grilles/bases réduites pour l’approximation des solutions d’une e.d.p. paramétrique. *Comptes Rendus Mathématique*, 347(7):435 – 440, 2009.
- [18] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7):808–817, 2000.
- [19] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.
- [20] B. A. Cowles. High cycle fatigue in aircraft gas turbines—an industry perspective. *International Journal of Fracture*, 80(2-3):147–163, 1996.
- [21] L. Dalcín, R. Paz, P. Kler, and A. Cosimo. Parallel distributed computing using python. *Advances in Water Resources*, 34(9):1124–1139, 2011.
- [22] L. Dalcín, R. Paz, and M. Storti. Mpi for python. *Journal of Parallel and Distributed Computing*, 65(9):1108–1115, 2005.

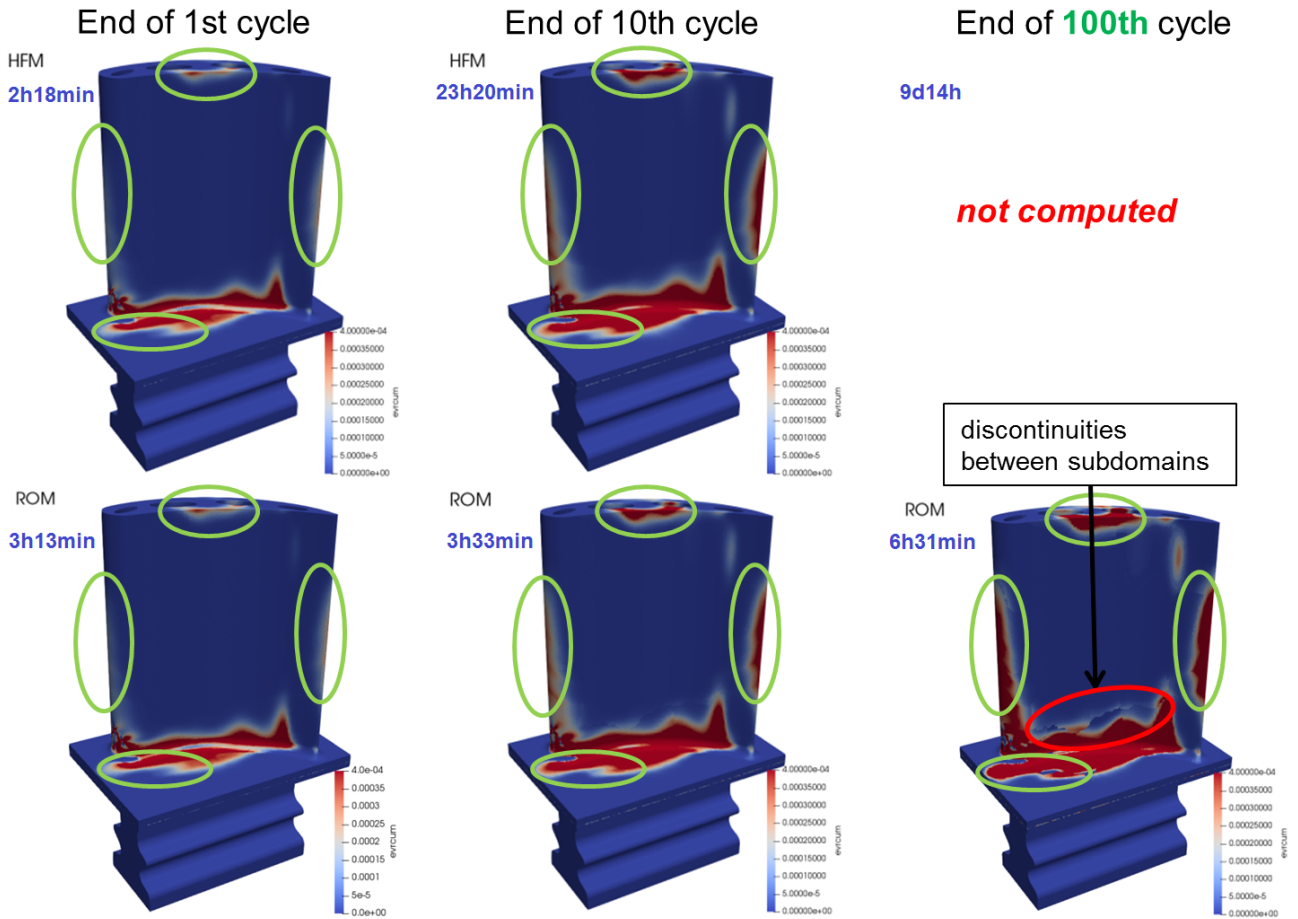


Figure 12: Magnitude of the cumulated plasticity fields for the HFM and the ROM at the end of the 1st and 10th cycles, and for the ROM at the end of the 100th cycle (minimum loading) - with saturated colors.

- [23] L. Dalcín, R. Paz, M. Storti, and Jo. D'Elía. Mpi for python: Performance improvements and mpi-2 extensions. *Journal of Parallel and Distributed Computing*, 68(5):655–662, 2008.
- [24] A. Dasgupta, Y. V. Sun, I. R. König, J. E. Bailey-Wilson, and J. D. Malley. Brief review of regression-based and machine learning methods in genetic epidemiology: the genetic analysis workshop 17 experience. *Genetic Epidemiology*, 35(S1):S5–S11, 2011.
- [25] Z. Drmač and S. Gugercin. A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions. *SIAM Journal on Scientific Computing*, 38(2):A631–A648, 2016.
- [26] R. Everson and L. Sirovich. Karhunen–loève procedure for gappy data. *J. Opt. Soc. Am. A*, 12(8):1657–1664, Aug 1995.
- [27] C. Farhat, P. Avery, T. Chapman, and J. Cortial. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *International Journal for Numerical Methods in Engineering*, 98(9):625–662, 2014.
- [28] C. Farhat, T. Chapman, and P. Avery. Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models. *International Journal for Numerical Methods in Engineering*, 102(5):1077–1110, 2015.

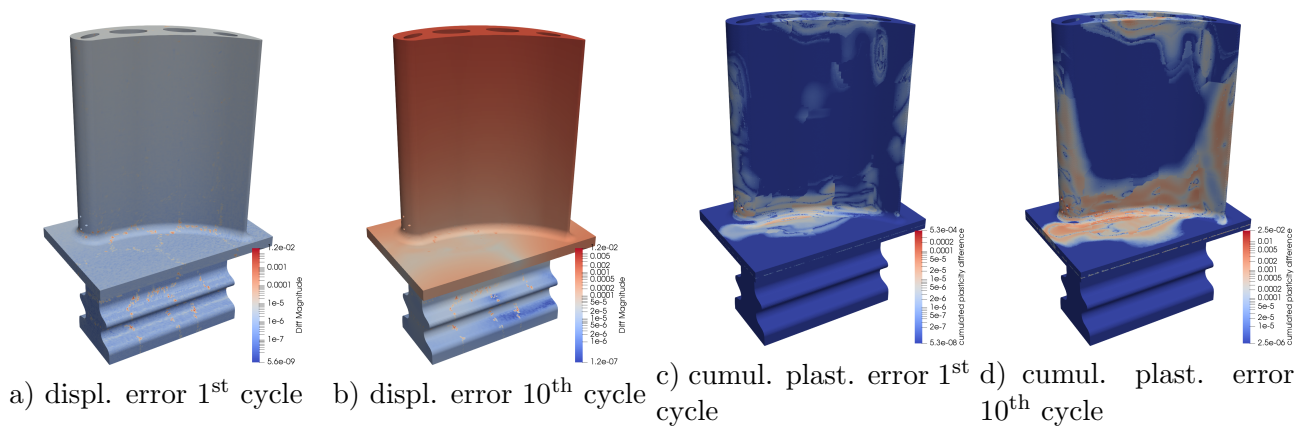


Figure 13: Difference maps between the ROM and the HFM (divided by the largest reference value on the map): displacement at the middle of a) the 1st and b) 10th cycles, cumulated at the end of c) the 1st and d) 10th cycles.

- [29] L. Giraldi, A. Litvinenko, D. Liu, H. Matthies, and A. Nouy. To be or not to be intrusive? the solution of parametric and stochastic equations—the “plain vanilla” galerkin case. *SIAM Journal on Scientific Computing*, 36(6):A2720–A2744, 2014.
- [30] J. A. Hernandez, M. A. Caicedo, and A. Ferrer. Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. *Computer methods in applied mechanics and engineering*, 313:687–722, 2017.
- [31] J. S. Hesthaven and S. Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55 – 78, 2018.
- [32] R. J. Kuether. *Nonlinear modal substructuring of geometrically nonlinear finite element models*. PhD thesis, The University of Wisconsin-Madison, 2014.
- [33] G. Luo. A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 5(1):18, 2016.
- [34] Y. Maday, N.-C. Nguyen, A. T. Patera, and S. H. Pau. A general multipurpose interpolation procedure: the magic points. *Communications on Pure and Applied Analysis*, 8(1):383–404, 2009.
- [35] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, Dec 1993.
- [36] Z. Mazur, A. Luna-Ramírez, J.A. Juárez-Islas, and A. Campos-Amezcuca. Failure analysis of a gas turbine blade made of inconel 738lc alloy. *Engineering Failure Analysis*, 12(3):474 – 486, 2005.
- [37] P. Meer, D. Mintz, A. Rosenfeld, and D. Y. Kim. Robust regression methods for computer vision: A review. *International Journal of Computer Vision*, 6(1):59–70, 1991.
- [38] M. P. Mignolet, A. Przekop, S. A. Rizzi, and S. M. Spottswood. A review of indirect/non-intrusive reduced order modeling of nonlinear geometric structures. *Journal of Sound and Vibration*, 332(10):2437–2460, 2013.
- [39] Mines ParisTech and ONERA the French aerospace lab. Zset: nonlinear material & structure analysis suite. <http://www.zset-software.com>, 1981-present.
- [40] A. Paul-Dubois-Taine and D. Amsallem. An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models. *International Journal for Numerical Methods in Engineering*, 102(5):1262–1292, 2015.

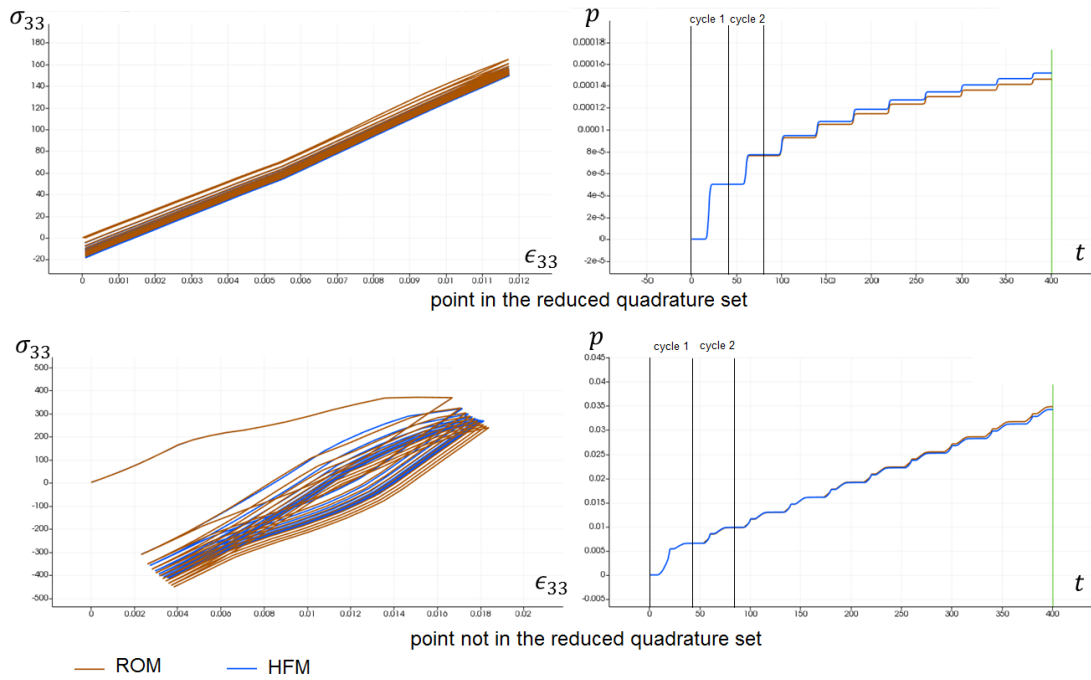


Figure 14: Comparison of reconstructed dual quantities using Algorithm 3 between the HFM (blue) and the ROM (brown) on the first 10 cycles, for a point in the reduced integration points set (top line) and not in this set (bottom line). Left-hand side: cyclic plots of σ_{33} with respect to ϵ_{33} , right-hand side: cumulated plasticity with respect to time.

- [41] R. Peharz and F. Pernkopf. Sparse nonnegative matrix factorization with 10-constraints. *Neurocomputing*, 80:38 – 46, 2012. Special Issue on Machine Learning for Signal Processing 2010.
- [42] A. Radermacher and S. Reese. Pod-based model reduction with empirical interpolation applied to nonlinear elasticity. *International Journal for Numerical Methods in Engineering*, 107(6):477–495, 2016.
- [43] U. Schulz, C. Leyens, K. Fritscher, M. Peters, B. Saruhan, O. Lavigne, J.-M. Dorvaux, M. Poulain, R. Mévrel, and M. Caliez. Some recent trends in research and technology of advanced thermal barrier coatings. 7:73–80, 2003.
- [44] L. Sirovich. Turbulence and the dynamics of coherent structures, parts I, II and III. *Quarterly of Applied Mathematics*, XLV:561–590, 1987.
- [45] M. Slawski and M. Hein. Non-negative least squares for high-dimensional linear models: Consistency and sparse recovery without regularization. *Electron. J. Statist.*, 7:3004–3056, 2013.
- [46] M. Smith. *ABAQUS/Standard User’s Manual, Version 6.9*. Simulia, 2009.
- [47] I. Uysal and H. A. Güvenir. An overview of regression techniques for knowledge discovery. *Knowl. Eng. Rev.*, 14(4):319–340, December 1999.
- [48] T. Verstraete, S. Amaral, R. Van den Braembussche, and T. Arts. Design and optimization of the internal cooling channels of a high pressure turbine blade—part ii: Optimization. *Journal of Turbomachinery*, 132, 2010.
- [49] K. Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & fluids*, 35(2):208–226, 2006.
- [50] M. Yaghoobi, D. Wu, and M. E. Davies. Fast non-negative orthogonal matching pursuit. *IEEE Signal Processing Letters*, 22(9):1229–1233, 2015.