Peer-reviewed version available at *Math. Comput. Appl.* **2019**, *24*, 17; doi:10.3390/mca24010017

Article

Multiple Tensor Train approximation of parametric constitutive equations in elasto-viscoplasticity

Clément Olivier[1,2,‡], David Ryckelynck[1,*,‡], Julien Cortial[2,‡]

[1] Safran Tech, Rue des Jeunes Bois, Chateaufort, CS 80112, 78772 Magny-Les-Hameaux, France; julien.cortial@safrangroup.com

[2] MINES ParisTech, PSL - Research University, Centre des matériaux, CNRS UMR 7633, 10 rue Desbruères, 91003 Evry, France; david.ryckelynck@mines-paristech.fr, clement.olivier@mines-paristech.fr

* Correspondence: david.ryckelynck@mines-paristech.fr

Abstract: This work presents a novel approach to construct surrogate models of parametric Differential Algebraic Equations based on a tensor representation of the solutions. The procedure consists in building simultaneously, for every output of the reference model, an approximation given in tensor-train format. A parsimonious exploration of the parameter space coupled with a compact data representation allows to alleviate the curse of dimensionality. The approach is thus appropriate when many parameters with large domains of variation are involved. The numerical results obtained for a nonlinear elasto-viscoplastic constitutive law show that the constructed surrogate model is sufficiently accurate to enable parametric studies such as the calibration of material coefficients.

Keywords: parameter-dependent model; surrogate modeling; tensor-train decomposition; gappy POD; heterogeneous data; elasto-viscoplasticity

1. Introduction

Predictive numerical simulations in solid mechanics require complex material laws that involve systems of highly nonlinear Differential Algebraic Equations (DAEs). These models are essential in challenging industrial applications, for instance to study the effects of the extreme thermo-mechanical loadings that turbine blades may sustain in helicopter engines [1] and [2].

These DAE systems are referred to as constitutive laws in the material science community. They express, for a specific material, the relationship between the mechanical quantities such as the strain, the stress and miscellaneous internal variables, and stand as the closure relations of the physical equations of mechanics. Complex constitutive equations are often tuned through a set of parameters called material coefficients.

An appropriate calibration of these coefficients is necessary to ensure that the numerical model mimics the actual physical behavior. Numerical parametric studies, consisting in analyzing the influence of the parameter values on the solutions, are typically used to perform the identification. However, when the number of parameters increases and unless the computational effort required for a single numerical simulation is negligible, the exploration of the parameter domain turns into a tedious task and exhaustive analyses become unfeasible. Moreover, defining an unambiguous criterion measuring the fidelity of the model to experimental data is a challenge for models with complex behaviors.

A common technique to mitigate the aforementioned challenges is to build surrogate models (or metamodels) mapping points of a given parameter space (considered as the inputs of the model) to the outputs of interest of the model. The real-time prediction of DAE solutions for arbitrary parameter

values, enabled by the surrogate model, helps the comprehension of constitutive laws and facilitate the conduct of parametric studies. In particular, the robustness of the calibration process can be dramatically improved using surrogate model approaches.

The idea of representing the set of all possible parameter-dependent solutions of ODEs and PDEs as a multiway tensor was pioneered with the introduction of the Proper Generalized Decomposition (PGD) [3–5]. In this representation, each dimension corresponds to a spatial/temporal coordinate or a parameter coefficient. The resulting tensor is never assembled explicitly but instead remains an abstract object for which a low-rank approximation based on a Canonical Polyadic decomposition [6] is computed. The PGD method further alleviates the curse of dimensionality by introducing a multidimensional weak formulation over the entire parameter space, and the solutions are sought in a particular form where all variables are separated. When differential operators admit a tensor decomposition, the PGD method is very efficient because the multiple integrals involved in the multidimensional weak form of the equations can be rewritten as a sum of products of simple integrals.

Unfortunately realistic constitutive equations or even less sophisticated elasto-viscoplastic models admit no tensor decomposition with respect to the material coefficients and the time variables. An extension of the PGD to highly nonlinear laws is therefore non-trivial. However, many other tensor decomposition approaches have been successfully proposed to approximate functions or solutions of differential equations defined over high dimensional spaces. We refer the reader to [7–9] for detailed reviews on tensor decomposition techniques and their applications.

Among the existing formats – CP decomposition [6,10,11], Tucker decomposition [8,12], Hierarchical Tucker decomposition [8,13] – this work investigates the tensor-train (TT) decomposition [14,15]. The TT-cross algorithm, introduced in [14] and further developed in [16,17], is a sampling procedure to build an approximation of a given tensor under the tensor-train format. Sampling procedures in parameter space have proven their ability to reduce nonlinear and non-separable DAEs by using the Proper Orthogonal Decomposition (POD) [18], the Gappy POD [19], or the Empirical Interpolation Method (EIM) [20,21]. These last methods are very convenient when the solutions have only two variables, hence they are considered as second order tensors.

This paper aims to extend the sampling procedure of the TT-cross method to DAEs having heterogenous and time-dependent outputs. A common sampling of the parameter space is proposed, though several TT-cross approximations are computed to cope with heterogeneous outputs. These outputs can be scalars, vectors or tensors, with various physical units. In the proposed algorithm, sampling points are not specific to any output although parameters do not affect equally each DAE output. The proposed method is named multiple TT-cross approximation. Similarly to the construction of a reduced integration domain for the hyperreduction of partial differential equations [22] or for the GNAT method [23], the set of sampling points is the union of contributions from the various outputs of the DEA. In this paper, the multiple TT-cross incorporates the Gappy POD method and the developments are focused on the numerical outputs obtained through a numerical integration scheme applied to the DAE.

2. Materials and Methods

The parametized material model generates several time-dependent quantities of interest (QoI). These quantities can be scalar-, vector- or even tensor-valued (e.g. stress) and are generally of distinct natures, namely expressed with different physical units and/or have different magnitudes. Therefore, the generated data will be segregated according to the QoI it relates to. It will also be structured in a tensor-like fashion to make it amenable to the numerical methods presented in this paper.

For a given $\chi = 1, \dots, N$ denoting an arbitrary QoI, the tensor of order d, $\boldsymbol{\mathcal{A}}^\chi \in \mathbb{R}^{n_1^\chi \times \cdots \times n_d^\chi}$ (denoted with bold calligraphic letter) refers to a multidimensional array (also called multiway array). Each element of $\boldsymbol{\mathcal{A}}^\chi$ identified by the indices $(i_1, \dots i_d) \in D_1 \times \cdots \times D_{d-1} \times D_d^\chi$ is denoted by:

$$\boldsymbol{\mathcal{A}}^\chi(i_1, \dots, i_d) \in \mathbb{R}$$

where $D_k = [1 : n_k]$ for $k < d$, is the set of natural numbers from 1 to n_k (inclusive) and $D_d^\chi = [1 : n_d^\chi]$. The last index is specific to each $\mathcal{A}^\chi$, while the other are common to all tensors for $\chi = 1, ..., N$. Hence, a common sampling of the parameter space $D_1 \times \ldots \times D_{d-1}$ can be achieved. The vector $\mathcal{A}^\chi(i_1, \ldots, i_{d-1}, :) \in \mathbb{R}^{n_d^\chi}$ contains all the components of output χ at all time instants used for the numerical solution of the DAE and for a given point in the parameter space.

The q^{th} matricization of $\mathcal{A}^\chi$ denoted by $\langle \mathcal{A}^\chi \rangle_q$ consists in dividing the dimensions of $\mathcal{A}^\chi$ into two groups, the q leading dimensions and the $(d - q)$ trailing dimensions, such that the newly defined multi-indices enumerate respectively the rows and columns of the matrix $\langle \mathcal{A}^\chi \rangle_q$. For instance, the elements of $\langle \mathcal{A}^\chi \rangle_1$ and $\langle \mathcal{A}^\chi \rangle_2$ are given by:

$$\langle \mathcal{A}^\chi \rangle_1 (i_1, j^\star) = \mathcal{A}^\chi(i_1, \ldots, i_d)$$

$$\langle \mathcal{A}^\chi \rangle_2 (i_1 + (i_2 - 1)n_1, j^{\star\star}) = \mathcal{A}^\chi(i_1, \ldots, i_d)$$

where $j^\star$ enumerates the multiple indices $(i_2, \ldots, i_d)$ and $j^{\star\star}$ enumerates $(i_3, \ldots, i_d)$. Here again, these matricizations are purely formal because of the curse of dimensionality. In particular, the number of columns in $\langle \mathcal{A}^\chi \rangle_1$ is equal to $n_2 \ldots n_{d-1}\, n_d^\chi$. The Frobenius norm is denoted by $\|.\|$ without the usual subscript $\mathcal{F}$. For $\mathcal{A}^\chi \in \mathbb{R}^{n_1 \times \ldots n_d^\chi}$, it reads:

$$\|\mathcal{A}^\chi\| = \sqrt{\sum_{i_1,\ldots,i_d \in D_1 \times \cdots \times D_{d-1} \times D_d^\chi} \mathcal{A}^\chi(i_1, \ldots, i_d)^2}$$

The Frobenius norm of a tensor is invariant under all matricizations of a given tensor.

In [14], the Singular Value Decomposition (SVD) is considered in the algorithm called TT-SVD. Because of the curse of dimensionality, the TT-SVD has no practical use, even if tensors have a low rank. More convenient approaches, aim to sample the entries of tensors.

For instance, in the snapshot proper orthogonal decomposition (POD) [18], the sampling procedure aims to estimate the rank and an orthogonal reduced basis for the approximation of a matrix A. The method consists in applying the truncated SVD on the submatrix $\tilde{A} = A(:, \mathcal{J}_{pod})$ constituted by a selection of columns $\mathcal{J}_{pod}$ of A. Hence the accuracy of the resulting POD reduced basis relies on the quality of the sampling procedure that generally introduces a sampling error. This sampling procedure seams to be convenient when considering the first matricizations $\langle \mathcal{A}^\chi \rangle_q$ if the product $n_1\, n_2 \ldots n_q$ and $\mathrm{Card}(\mathcal{J}_{pod})$ are reasonably small regarding the available computing ressources. But, for large values of q, the curse of dimensionality makes the snapshot POD alone, intractable.

A more practical approach to effectively construct an approximate TT decomposition, called the TT-cross method, is proposed in [14]. The TT-cross consists in dropping the concept of a POD basis and using the Pseudo-Skeleton Decomposition (PSD) introduced in [24] as low-rank approximation. Unlike the TT-SVD, the TT-cross enables to build an approximation based on a sparse exploration of a reference tensor. The Pseudo-Skeleton Decomposition can be used to approximate any matrix $A \in \mathbb{R}^{n \times m}$ and is written as:

$$A = \underbrace{A(:, \mathcal{J}_{psd}) \left[A(\mathcal{I}_{psd}, \mathcal{J}_{psd}) \right]^{-1} A(\mathcal{I}_{psd}, :)}_{= T_{psd}} + E_{psd} \tag{1}$$

where the sets $\mathcal{I}_{psd}$ and $\mathcal{J}_{psd}$ are respectively a selection of row and column indices. The definition is valid only when the matrix $A(\mathcal{I}_{psd}, \mathcal{J}_{psd})$ is non-singular. In particular, the number s of rows and columns has to be identical.

This approximation (1) features an interpolation property at the selected rows and columns:

$$T_{psd}(\mathcal{I}_{psd},:) = A(\mathcal{I}_{psd},:) \quad \text{and} \quad T_{psd}(:,\mathcal{J}_{psd}) = A(:,\mathcal{J}_{psd}) \tag{2}$$

The Pseudo-Skeleton Decomposition is a matrix factorization similar to the decomposition used in the Adaptive Cross Approximation (ACA) [25] and the CUR decomposition [26,27]. Additionally, these references provide algorithms to effectively build the factorization. That decomposition has also been used in the context of model order reduction, for instance in the Empirical Interpolation Method (EIM) proposed in [20,21].

The condition that $A(\mathcal{I}_{psd},\mathcal{J}_{psd})$ must be non-singular make difficult to share sampling points for various matrices $\langle \boldsymbol{\mathcal{A}}^{\chi} \rangle_q$ with $\chi = 1,\ldots,N$ having their own rank.

The Gappy POD introduced in [19] aims at relaxing the aforementioned constraint by combining beneficial features of the Snapshot POD and the Pseudo-Skeleton Decomposition. Indeed, the Gappy POD a) relies on a POD basis that remains computationally affordable, b) requires only a limited number of rows of the matrix to be approximated and c) enables to reuse the set of selected rows for different matrices. These properties are key ingredients for an efficient, parsimonious exploration of the reference tensors. The Gappy POD approximation T_{gap} of a matrix $A \in \mathbb{R}^{n\times m}$ is given by:

$$A = \underbrace{V[V(\mathcal{I}_{gap},:)]^{\dagger} A(\mathcal{I}_{gap},:)}_{=T_{gap}} + E_{gap} \tag{3}$$

where † denotes the Moore-Penrose pseudo-inverse [28], $\mathcal{I}_{gap}$ is a row selection of s rows and where $V \in \mathbb{R}^{n\times r}$ is a POD basis matrix of rank r such that :

$$A(:,\mathcal{J}_{pod}) = V\,S\,W^T + E_{pod} \tag{4}$$

In the sequel, because the simulation data in $\boldsymbol{\mathcal{A}}^{\chi}$ are outputs of a DAE system, it does not make sense to sample the last index i_d during column sampling of $\langle \boldsymbol{\mathcal{A}}^{\chi} \rangle_q$. Each numerical solution of the DAE system generates all the last components of each tensor $\boldsymbol{\mathcal{A}}^{\chi}$. Hence, the column sampling is restricted to indices $i_{q+1},\ldots i_{d-1}$ and is replicated for all values of i_d in D_d^{χ}. This special column sampling is denoted by $\mathcal{J}_{pod}^{\chi}$. It is performed randomly by using a low-discrepancy Halton sequence [29].

The matrix $V(\mathcal{I}_{gap},:)$ must have linearly independent columns to ensure that the approximation is meaningful. Since V is a rank-r POD basis, there exists a set of s rows such that this property holds as long as $s \geq r$. Here, $\mathcal{I}_{gap}$ contains at least the interpolation indices related to V. This latter set is denoted by $\mathcal{I}^{\chi}$, such that $V(\mathcal{I}^{\chi},:)$ is invertible. In the numerical results presented hereafter $\mathcal{I}^{\chi}$ is the obtained using the Q-DEIM algorithm [30] that was shown to be a superior alternative to the better-known DEIM procedure [31, Algorithm 1].

Unlike the PSD, the Gappy POD enables to select a number of rows that exceeds the rank of the low-rank approximation:

$$\mathcal{I}_{gap} = \mathcal{I}^1 \cup \ldots \cup \mathcal{I}^N \tag{5}$$

This make possible to share sampling points between matrices having their own rank. In this case, the interpolation property does not hold as in the PSD case (2).

T_{gap} is the approximation of A by the product of three matrices: V, $\left[V(\mathcal{I}_{gap},:)\right]^{\dagger}$ and $A(\mathcal{I}_{gap},:)$. The TT-cross approximation can be understood as a generalization of such product of matrices. A

tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is said to be in tensor-train format (TT format) if its elements are given by the following matrix products:

$$\mathcal{T}(i_1, \ldots, i_d) = G_1(i_1) \ldots G_d(i_d) \in \mathbb{R} \tag{6}$$

where the so-called *tensor carriages* (or *core tensors*) are such that for $k = 1, \ldots, d$,:

$$G_k(i_k) \in \mathbb{R}^{r_{k-1} \times r_k} \quad \forall i_k \in D_k$$

In the original definition of the tensor-train format [14], the leading and trailing factors (corresponding to $G_1(i_1)$ and $G_d(i_d)$ for any choice of i_1 and i_d) are respectively row and column vectors. Here, the convention $r_0 = r_d = 1$ is adopted so that row matrices $G_1(i_1)$ and column matrices $G_d(i_d)$ can be interpreted as vectors or matrices depending on the context.

The TT format allows significant gains in terms of memory storage and therefore is well-suited to high order tensors. The storage complexity is $\mathcal{O}(n\bar{r}^2 d)$ where $\bar{r} = \max(r_1, \ldots, r_{d-1})$ and depends linearly on the order d of the tensor. In many applications of practical interest the small TT-ranks r_k enable to alleviate the curse of dimensionality [14].

The sequential computational complexity of the evaluation of a single element of a tensor in TT format is $\mathcal{O}\left(d\bar{r}^2\right)$. Assuming that $\bar{r}$ is small enough, the low computational cost allows a real-time evaluation of the underlying tensor. Therefore, in terms of online exploitation, this representation conforms with the expected requirements of the surrogate model. Figure 1 illustrates the sequence of matrix multiplications required to compute one element of the tensor train.

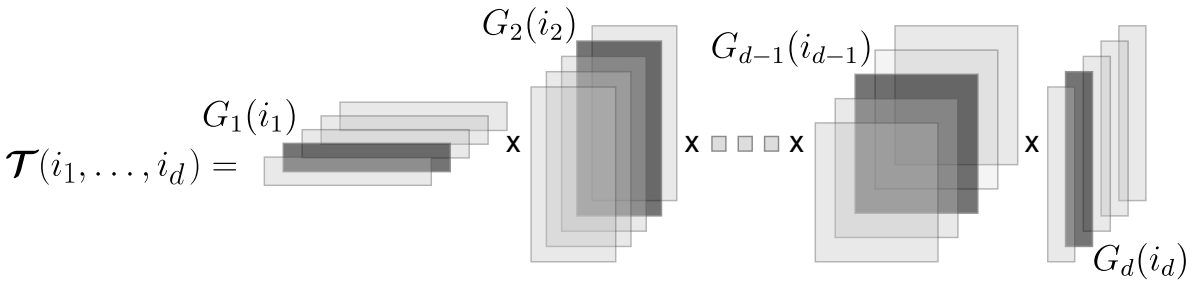


Figure 1. Illustration of the evaluation of one element of the tensor train. The entry $\mathcal{T}(i_1, \ldots, i_d) \in \mathbb{R}$ is obtained by multiplying the set of matrices $G_1(i_1), G_2(i_2) \ldots, G_d(i_d)$ identified by a darker shade.

The objective of the proposed approach is to build for each physics-based tensor $\boldsymbol{\mathcal{A}}^\chi$ an approximate tensor $\boldsymbol{\mathcal{T}}^\chi$ given in TT format by using a nested row sampling of the simulation data. Algorithm 1 provides the set of matrices $\{G_1^\chi, \ldots, G_d^\chi\}$ that enable to define the tensor-train decompositions and aggregate sets for row sampling. It is a sequential algorithm that navigates from dimension 1 to dimension $d-1$ of tensors $\boldsymbol{\mathcal{A}}^\chi$.

The method provided by Algorithm 1 is non-intrusive and relies on the numerical solutions of the DAEs in a black-box fashion.

At each iteration $k = 1, \ldots, d-1$, the Snapshot POD method, used to build the POD reduced basis (9), requires to sample a set $\mathcal{J}_k^\chi$. The column sampling amounts to a parsimonious selection of $\tilde{n}_k$ points in the partial discretized parameter domain $\mathcal{D}_{k+1} \times \cdots \times \mathcal{D}_{d-1}$ and an exhaustive sampling of the last dimension for each tensor $\boldsymbol{\mathcal{A}}^\chi$. The considered submatrices $\tilde{A}_k^\chi = A_k^\chi\left(:, \mathcal{J}_k^\chi\right)$ are then constituted of $\tilde{n}_k^\chi = \tilde{n}_k n_d^\chi$ columns (See Figure 2).

In the row sampling step, specific sets of interpolant rows $\mathcal{I}_k^\chi$ are first determined independently for each output χ but a common, aggregated set $\mathcal{I}_k$ (10) is then used to sample the entries of all outputs. Indeed, computing the elements of all submatrices $A_k^\chi\left(\mathcal{I}_k, :\right)$ requires m_k calls to the DAE system solver with: $m_k = \mathrm{Card}(\mathcal{I}_{k-1})\, \tilde{n}_k$ with $\mathcal{I}_0 = D_1$. Furthermore, the Gappy POD naturally accommodates a

Algorithm 1: Multiple TT decomposition

Input: Tensors $\boldsymbol{\mathcal{A}}^{\chi} \in \mathbb{R}^{n_1 \times \cdots \times n_{d-1} \times n_d^{\chi}}$ for $\chi = 1, \ldots, N$ associated with a DAE system
Output: Sets of matrices $\left\{G_1^{\chi}, \ldots, G_d^{\chi}\right\}$ for $\chi = 1, \ldots, N$.
Initialization:
For each χ, define the matrix $A_1^{\chi} \in \mathbb{R}^{(s_0 n_1) \times \left(n_2 \ldots n_{d-1} n_d^{\chi}\right)}$ with $s_0 = 1$, as the first matricization of the tensor $\boldsymbol{\mathcal{A}}^{\chi}$:

$$A_1^{\chi} = \langle \boldsymbol{\mathcal{A}}^{\chi} \rangle_1 \tag{7}$$

For $k = 1, \ldots, d-1$ **do**

Snapshot POD:
Define consistent sets of sampling columns $\mathcal{J}_k^{\chi}$ and evaluate the DAE to fill the matrices $\tilde{A}_k^{\chi}$ defined as:

$$\tilde{A}_k^{\chi} = A_k^{\chi}\left(:, \mathcal{J}_k^{\chi}\right) \quad \text{for} \quad \chi = 1, \ldots, N$$

Apply the truncated SVD (4) on each $\tilde{A}_k^{\chi}$ with the truncation tolerance ϵ to get the rank-r_k^{χ} matrices:

$$\tilde{A}_k^{\chi} = V_k^{\chi}\, S_k^{\chi}\, {W_k^{\chi}}^T + E_{pod\,k}^{\chi} \quad \text{with} \quad \left\| E_{pod\,k}^{\chi} \right\| \leq \epsilon \left\| \tilde{A}_k^{\chi} \right\| \tag{8}$$

$$V_k^{\chi} \in \mathbb{R}^{(s_{k-1} n_k) \times r_k^{\chi}} \quad \text{for} \quad \chi = 1, \ldots, N \tag{9}$$

Row Sampling:
From each χ, select a set of rows $\mathcal{I}_k^{\chi}$ applying the Q-DEIM algorithm [30] to the basis V_k^{χ}.
Define the union of all selected rows and the corresponding row selection matrix:

$$\mathcal{I}_k = \bigcup_{\chi=1}^{N} \mathcal{I}_k^{\chi} \tag{10}$$

and

$$s_k = \mathrm{Card}(\mathcal{I}_k) \tag{11}$$

Output definitions:
Compute the matrices $G_k^{\chi} \in \mathbb{R}^{(s_{k-1} n_k) \times s_k}$ such that:

$$G_k^{\chi} = V_k^{\chi} \left[V_k^{\chi}(\mathcal{I}_k, :)\right]^{\dagger}$$

Tensorization:
Define, formally, the tensors $\boldsymbol{\mathcal{A}}^{\chi,(k+1)} \in \mathbb{R}^{s_k \times n_{k+1} \times \cdots \times n_{d-1} \times n_d^{\chi}}$ such that:

$$\left\langle \boldsymbol{\mathcal{A}}^{\chi,(k+1)} \right\rangle_1 = A_k^{\chi}(\mathcal{I}_k, :) \in \mathbb{R}^{s_k \times \left(n_{k+1} \ldots n_{d-1} n_d^{\chi}\right)} \tag{12}$$

Matricization:
Define, formally, the matrix $A_{k+1}^{\chi} \in \mathbb{R}^{(s_k n_{k+1}) \times \left(n_{k+2} \ldots n_{d-1} n_d^{\chi}\right)}$ as the second matricization of the tensor $\boldsymbol{\mathcal{A}}^{\chi,(k+1)}$:

$$A_{k+1}^{\chi} = \left\langle \boldsymbol{\mathcal{A}}^{\chi,(k+1)} \right\rangle_2 \tag{13}$$

Finalization:
For each $\chi = 1, \ldots, N^{\chi}$, define the matrix $G_d^{\chi} \in \mathbb{R}^{\left(s_{d-1} n_d^{\chi}\right) \times s_d}$ with $s_d = 1$ such that:

$$G_d^{\chi} = A_d^{\chi} \tag{14}$$

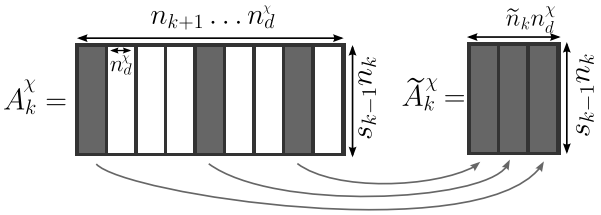


Figure 2. Definition of the submatrix $\tilde{A}_k^\chi$ used to construct the POD reduced basis. In the illustration, the Snapshot POD sample size is $\tilde{n}_k = 3$.

number of rows larger than the rank r_k^χ for each approximation of A_k^χ, and considering a larger sample size for each individual χ is expected to provide a more accurate approximation.

The tensorization and matricization steps are purely formal. No call to the DAE system solver is done here. They define the way the simulation data must be ordered in matrices to be approximated at the next iteration. The recursive definition of the matrix A_k^χ implies that the latter is equal to the k^{th} matricization of a subtensor extracted from $\mathcal{A}^\chi$. Equivalently, the matrix A_k^χ corresponds to a submatrix of the k^{th} matricization of $\mathcal{A}^\chi$, as illustrated in Figure 3.

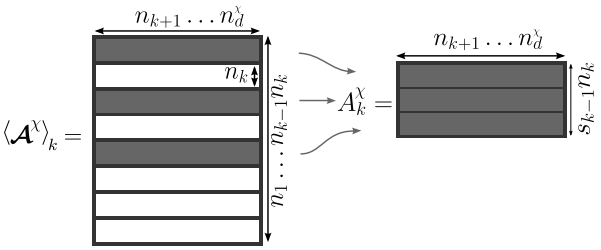


Figure 3. Definition of A_k^χ based on $\mathcal{A}^\chi$. In the illustration, the number of rows selected at the previous iteration $k-1$ is $s_{k-1} = 3$.

To quantify the theoretical accumulation of errors introduced at each iteration, Proposition 1 gives an upper bound for the approximation error associated with a tensor-train decomposition built by the snapshot POD followed by the row sampling steps, when a full column sampling is performed.

Proposition 1. *Consider* $\mathcal{A}^\chi \in \mathbb{R}^{n_1 \times \cdots \times n_{d-1} \times n_d^\chi}$ *and its tensor-train approximation* $\mathcal{T}^\chi$ *constructed by Algorithm 1. Assuming that for all* $k \in [1 : d-1]$

$$\left\| \left(\mathbb{I} - V_k^\chi V_k^{\chi\, T} \right) A_k^\chi \right\| \leq \epsilon \left\| A_k^\chi \right\| \tag{15}$$

the following inequality holds:

$$\| \mathcal{A}^\chi - \mathcal{T}^\chi \| \leq \sum_{k=1}^{d-1} \frac{\epsilon}{\sigma_{\min}\left(V_k^\chi(I_k, :) \right)} \prod_{k'=1}^{k-1} \frac{\min(\sigma_{\max}(V_{k'}^\chi(I_{k'}, :)) + \epsilon,\, 1)}{\sigma_{\min}(V_{k'}^\chi(I_{k'}, :))} \| \mathcal{A}^\chi \| \tag{16}$$

where $\sigma_{\min}$ *and* $\sigma_{\max}$ *refer to the smallest and the largest singular values of its matrix argument.*

The proof is given in [32] (Proposition 12).

Proposition 1 suggests that the approximation error

$$\| \mathcal{A}^\chi - \mathcal{T}^\chi \|$$

can be controlled by the truncation tolerances ϵ set by the user. However, the bound (16) tends to be very loose and the hypothesis (15) may be diffi cult to verify when the basis V_k^χ stems from a column sampling of the matrix A_k^χ. Hence, the convergence should be assessed empirically in practical cases.

3. Results

3.1. Outputs partitioning as formal tensors

The physical model described in A is represented as the relations between 6 ($d = 7$) parameters (inputs of the model) and the time-dependent mechanical variables (outputs of the model)

$$(n, K, R_0, Q, b, C) \mapsto \left(\underset{\sim}{\varepsilon}(t), \underset{\sim}{\varepsilon}_{vp}(t), \underset{\sim}{\sigma}(t), p(t)\right)$$

where $\underset{\sim}{\varepsilon}$, $\underset{\sim}{\varepsilon}_{vp}$,$\underset{\sim}{\sigma}$ have six components each and p is a scalar. $\underset{\sim}{\varepsilon}$, $\underset{\sim}{\varepsilon}_{vp}$ and p have the same units but have different physical meanings.

The surrogate model is defined by introducing $N = 4$ groups of outputs as tensors $\boldsymbol{\mathcal{A}}^{\chi}$. The formal tensors $\boldsymbol{\mathcal{A}}^{1}$, ... $\boldsymbol{\mathcal{A}}^{4}$ are related to $\underset{\sim}{\varepsilon}$, $\underset{\sim}{\varepsilon}_{vp}$,$\underset{\sim}{\sigma}$ and p, respectivelly.

For each parameter, the interval of definition is discretized by a regular grid with 30 points:

$$n_1 = n_2 = n_3 = n_4 = n_5 = n_6 = 30$$

The time interval discretized is the one used for the numerical solution, it corresponds to a regular grid with $n_t = 537$ points. Then:

$$n_7^1 = n_7^2 = n_7^3 = 6n_t \quad \text{and} \quad n_7^4 = n_t$$

The Snapshot POD sample sizes are:

$$\tilde{n}_1 = \tilde{n}_2 = \tilde{n}_3 = \tilde{n}_4 = \tilde{n}_5 = 100 \quad \text{and} \quad \tilde{n}_6 = 30$$

3.2. Performance indicators

The truncation tolerance is chosen here to be $\epsilon = 10^{-3}$. The construction of the tensor-train decompositions requires to solve the system of DAEs $\sum_{k=1}^{d-1} s_k n_k \tilde{n}_k$ times with as many sets of parameter values. In the proposed numerical example, it amounts to 514 050 solutions. 15 hours are necessary on a 16-core workstation to carry out the computations. 98% of the effort is devoted to the solution of the physical model and the remaining 2% to the decomposition operations.

For a single simulation on a personal laptop computer, the solution of the physical model takes 0.7 s, whereas the surrogate model is evaluated in only 1 ms, corresponding to a speed-up of 700.

Storing the Multiple TT approximations requires 2 709 405 double-precision floating-point values. For comparison purposes, storing a single solution (constituted of the multiple time-dependent outputs) of the DAE system involves 10 203 values. Therefore, the storage of the tensor-train decompositions is commensurate with the storage of 265 solutions while it can express the approximation of 30^6 solutions.

For $\chi = 1, \ldots, 4$, the rank r_k^{χ} is bounded from above by the theoretical maximum rank $r_{max,k}^{\chi}$ of the matrix A_k^{χ}. More specifically, $r_{max,k}^{\chi}$ corresponds to the case where A_k^{χ} has full rank and is the k^{th} matricizations of the tensors $\boldsymbol{\mathcal{A}}^{\chi}$. Given the choice of truncation tolerance $\epsilon = 10^{-3}$, the TT-ranks listed in Table 1 show that the resulting tensor trains involve low rank approximations. Table 2 emphasizes that in practice $r_k^{\chi} \ll r_{max,k}^{\chi}$ except for $k = 1$ where $r_{max,k}^{\chi}$ is already "small".

3.3. Approximation error

The accuracy of the surrogate model is estimated a posteriori by measuring the discrepancy between its own outputs and the outputs of the original physical model. The estimation is conducted by comparing solutions associated with 20 000 new samples of parameter set values randomly selected according to a uniform law on each discretized parameter intervals. The difference between the surrogate and the physical models is measured based on the following norms:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 November 2018 doi:10.20944/preprints201811.0308.v1

Peer-reviewed version available at Math. Comput. Appl. 2019, 24, 17; doi:10.3390/mca24010017

Table 1. TT-ranks of the outputs of interest and theoretical maximum ranks.

	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
r^1_k	7	9	10	24	27	30
r^2_k	13	23	29	123	143	134
r^3_k	11	17	20	67	90	100
r^4_k	9	12	14	24	20	21
$r^1_{max,k} = r^2_{max,k} = r^3_{max,k}$	30	30^2	30^3	30^4	$6 \times 30 n_t$	$6 \times n_t$
$r^4_{max,k}$	30	30^2	30^3	$30^2 n_t$	$30 n_t$	n_t

Table 2. Ratio between the theoretical maximum ranks and the TT-ranks of the outputs of interest.

	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
$r^1_{max,k}/r^1_k$	4.3	1.0×10^2	2.7×10^3	3.4×10^4	3.6×10^3	1.1×10^2
$r^2_{max,k}/r^2_k$	2.3	3.9×10^1	9.3×10^2	6.6×10^3	6.8×10^2	2.4×10^1
$r^3_{max,k}/r^3_k$	2.7	5.3×10^1	1.4×10^3	1.2×10^4	1.1×10^3	3.2×10^1
$r^4_{max,k}/r^4_k$	3.3	7.5×10^1	1.9×10^3	2.0×10^4	8.1×10^2	2.6×10^1

$$\|x\|^2_{[0,T]} = \int_0^T x^2 dt \quad \text{et} \quad \left\|\underset{\sim}{X}\right\|^2_{[0,T]} = \int_0^T \underset{\sim}{X} : \underset{\sim}{X}\, dt$$

where x and $\underset{\sim}{X}$ are respectively scalar and tensor time-dependent function.

For the mechanical variable $\square$ (where $\square$ can stand for any one of $\underset{\sim}{\varepsilon}$, $\underset{\sim}{\varepsilon}_{vp}$, $\underset{\sim}{\sigma}$ and p), $\square^{PM}$ and $\square^{TT}$ denote the output corresponding respectively to the solution of the DAEs and the surrogate model. A relative error is associated with each mechanical variable, namely:

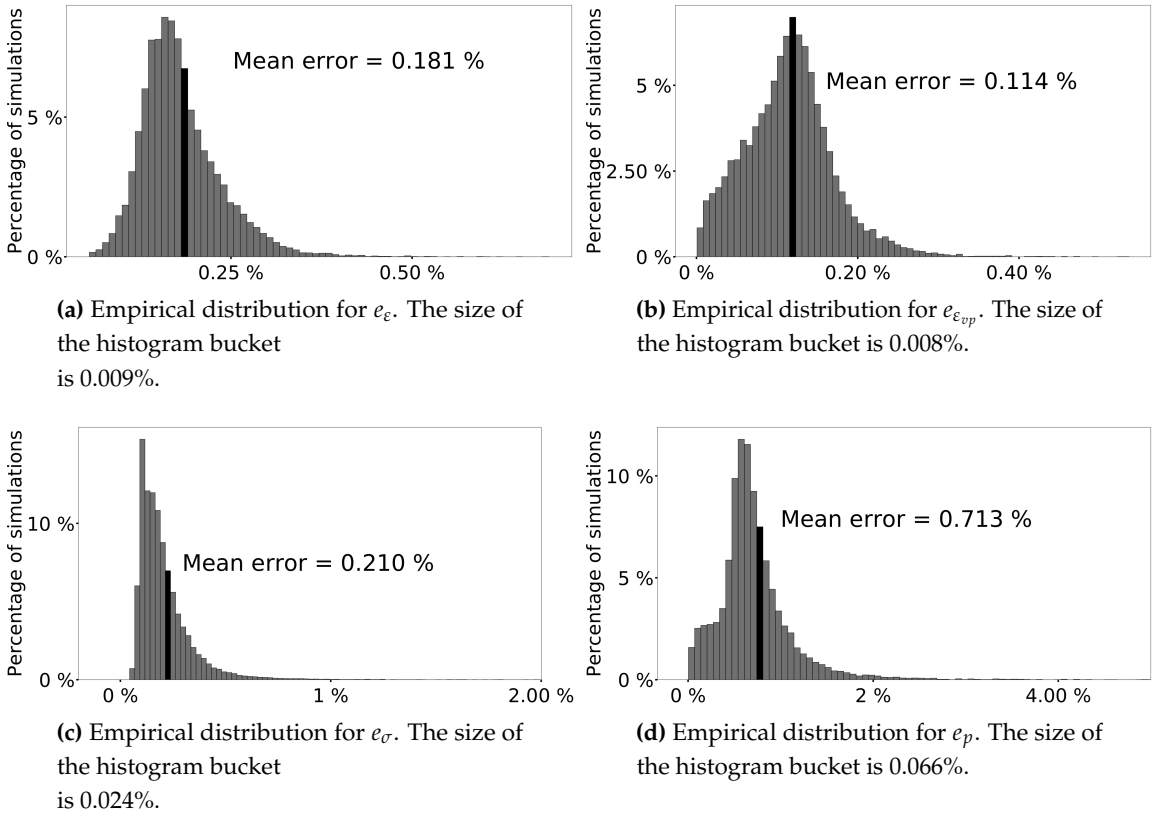
- Total strain tensor: $e_\varepsilon = \dfrac{\left\|\underset{\sim}{\varepsilon}^{PM} - \underset{\sim}{\varepsilon}^{TT}\right\|_{[0,T]}}{\left\|\underset{\sim}{\varepsilon}^{PM}\right\|_{[0,.]}}$;
- Viscoplastic strain tensor: $e_{\varepsilon_{vp}} = \dfrac{\left\|\underset{\sim}{\varepsilon}^{PM}_{vp} - \underset{\sim}{\varepsilon}^{TT}_{vp}\right\|_{[0,T]}}{\left\|\underset{\sim}{\varepsilon}^{PM}\right\|_{[0,.]}}$;
- Stress tensor: $e_\sigma = \dfrac{\left\|\underset{\sim}{\sigma}^{PM} - \underset{\sim}{\sigma}^{TT}\right\|_{[0,T]}}{\left\|\underset{\sim}{\sigma}^{PM}\right\|_{[0,.]}}$;
- Cumulative viscoplastic deformation: $e_p = \dfrac{\left\|p^{PM} - p^{TT}\right\|_{[0,T]}}{\left\|\underset{\sim}{\varepsilon}^{PM}\right\|_{[0,.]}}$.

Depending on the parameter values, the viscoplastic part of the behavior may or may not be negligible as measured by the magnitudes of $\|p\|$ and $\left\|\underset{\sim}{\varepsilon}_{vp}\right\|$ relative to $\left\|\underset{\sim}{\varepsilon}\right\|$. Hence, in the proposed application, the focus is on comparing the norm of the approximation error for $\underset{\sim}{\varepsilon}$, $\underset{\sim}{\varepsilon}_{vp}$ and p with respect to the norm of $\underset{\sim}{\varepsilon}$.

The histograms featured on Figures 4a, 4b, 4c and 4d present, for each mechanical variables, the empirical distribution of the relative error for all simulation results. The surrogate model given by the tensor-train decompositions features a level of error that is sufficiently low to carry out parametric studies such as calibration of constitutive laws where errors lower than 2% are typically tolerable.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 November 2018 doi:10.20944/preprints201811.0308.v1

Peer-reviewed version available at Math. Comput. Appl. 2019, 24, 17; doi:10.3390/mca24010017



(a) Empirical distribution for e_ε. The size of the histogram bucket is 0.009%.

(b) Empirical distribution for $e_{\varepsilon_{vp}}$. The size of the histogram bucket is 0.008%.

(c) Empirical distribution for e_σ. The size of the histogram bucket is 0.024%.

(d) Empirical distribution for e_p. The size of the histogram bucket is 0.066%.

Figure 4. Empirical distribution of the errors for every mechanical variables.

3.4. Convergence with respect to the truncation tolerance

A first surrogate model is constructed from the physical model with the prescribed truncation tolerance $\epsilon = 10^{-3}$. Then, this first surrogate model is used as an input for Algorithm 1. Running the algorithm several times with different truncation tolerances:

$$\epsilon \in \left\{1 \times 10^{-3};\ 2 \times 10^{-3};\ 4.6 \times 10^{-3}; 1 \times 10^{-2};\ 2 \times 10^{-2};\ 4.6 \times 10^{-2};\ 1 \times 10^{-1}\right\}$$

generates as many new surrogate models.

Figures 6a, 6b, 6c and 6d present the evolution of the relative error distribution (for the different mechanical variables) with respect to the truncation tolerance based on a random sample of 20 000 parameter set values chosen as in Section 3.3. Figure 5 details the graphical notations. The results empirically show for each mechanical output, the relative error decreases together with ϵ. It is consistent with the expected behavior of the algorithm.

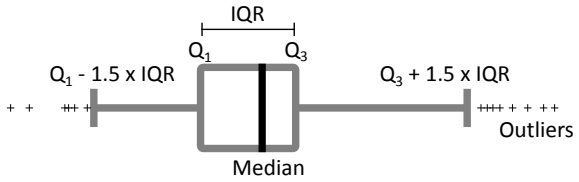
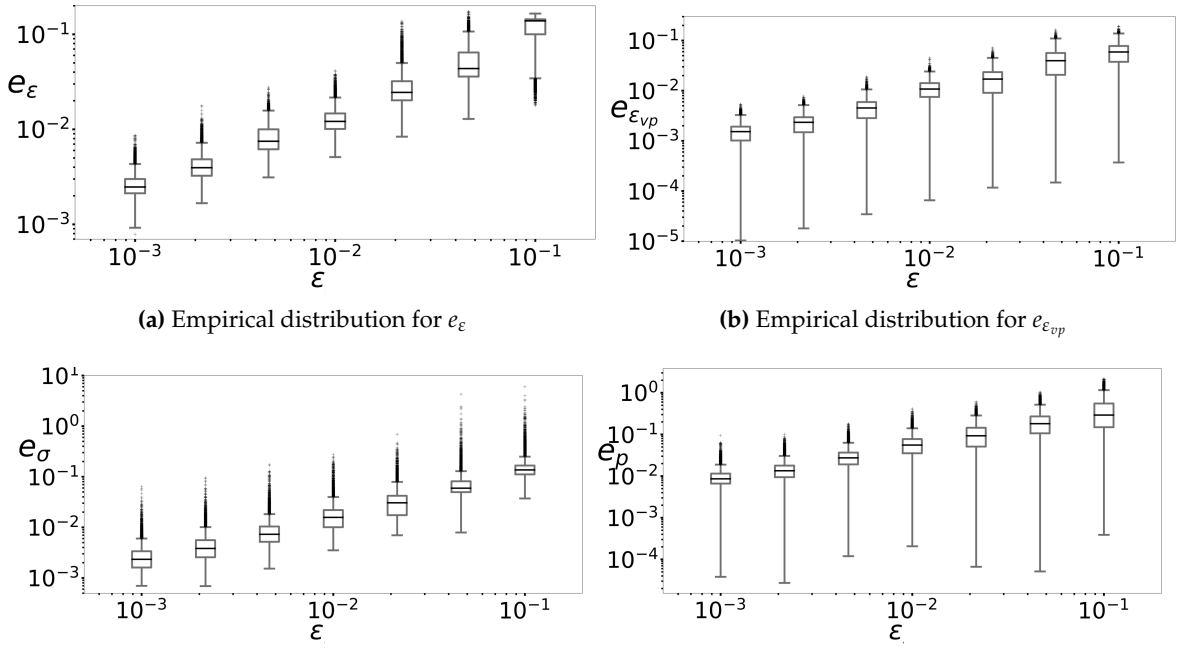


Figure 5. The left and right sides are the first and third quartiles (respectively Q_1 and Q_3). The line inside the box represents the median. The reach of the whiskers past the first and third quartiles is 1.5 times the interquartile range (IQR). The crosses represent the outliers lying beyond the whiskers.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 November 2018 doi:10.20944/preprints201811.0308.v1

Peer-reviewed version available at Math. Comput. Appl. 2019, 24, 17; doi:10.3390/mca24010017



(a) Empirical distribution for e_ε

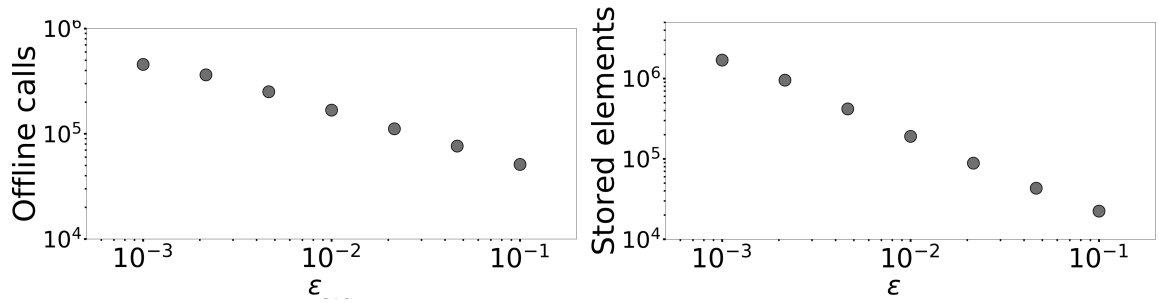
(b) Empirical distribution for $e_{\varepsilon_{vp}}$

(c) Empirical distribution for e_σ

(d) Empirical distribution for e_p

Figure 6. Empirical distribution of the relative approximation error for every mechanical variables.

Plots in Figure 7a and 7b show the dependence of the number of stored elements and the number of calls to the physical model on ϵ.



(a) Dependence of the number of calls to physical model on ϵ

(b) Dependence of the number of stored elements on ϵ

Figure 7. Dependence of computational cost and memory storage indicators on ϵ

3.5. On fly error estimation

Based on the physical model, the surrogate model gives an approximation of each output of interest. However, the approximate outputs may be inconsistent with the physics in the sense that they may lead to non-zero residuals when introduced into (the discrete version of) the DAE system describing the physical model.

A *coherence estimator* is an indicator that measures how closely the physical equations are verified by the outputs of the surrogate model. It is reasonable to expect the accuracy of the metamodel to be correlated with the coherence estimator.

Using Equation (A1) let:

$$\underset{\sim}{\sigma}^{eq,TT} = \frac{E}{1+\nu}\left(\underset{\sim}{\varepsilon}_e^{TT} + \frac{\nu}{1-2\nu} Tr\left(\underset{\sim}{\varepsilon}_e^{TT}\right)\underset{\sim}{\mathbb{I}}\right)$$

and define the associated coherence estimator as follows:

$$\eta_\sigma = \frac{\left\|\underset{\sim}{\sigma}^{TT} - \underset{\sim}{\sigma}^{eq,TT}\right\|_{[0,T]}}{\left\|\underset{\sim}{\sigma}^{TT}\right\|_{[0,T]}} \tag{17}$$

Figure 8 displays the relation between the relative error for $\underset{\sim}{\sigma}$ and the effectivity of the estimator η_σ / e_σ for 20 000 simulation results drawn randomly. The error increases with the final cumulative deformation, that is when the material exhibits a more intense viscoplastic behavior.

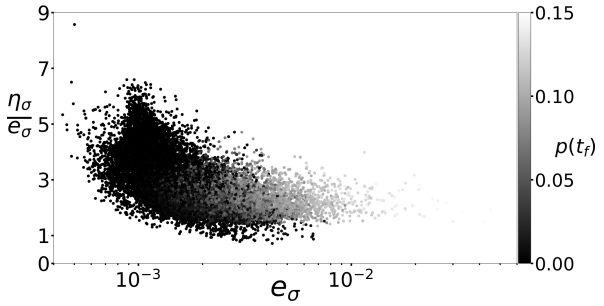


Figure 8. Effectivity of the coherence estimator η_σ (17) associated with σ. The color scale indicates the final cumulative deformation.

Furthermore, the plot shows a correlation between the coherence estimator and the relative error. In particular, the effectivity tends to be larger than 1 which indicates that the coherence estimator behaves like an upper bound of the relative error. Excluding a few outliers, the coherence estimator does not overestimate the relative error by more than a factor 7.

Finally, the effectivity of the coherence estimator empirically converges to 1 (that is, the estimator becomes sharper) as the magnitude of the relative error increases.

This coherence estimator is very cheap to compute and only relies on outputs of the surrogate model. The results suggest that the coherence estimator could be used as an online error indicator that increases the reliability of the surrogate model at the current point when exploring in real-time the parameter domain.

4. Discussion

The present work assesses the performance of tensor-train representations for the approximation of numerical solutions of nonlinear DAE systems. The proposed method enables to incorporate a large number of simulation results ($\simeq$ 500 000 scalar values) to produce a metamodel that is accurate over the entire parameter domain. More specifically, numerical results show that the Multiple TT decomposition gives promising results when used as a surrogate model for an elasto-viscoplastic constitutive law. For this particular application, the surrogate model exhibits a satisfying accuracy given the moderate computational effort spent for its construction and the data storage requirements. Moreover, the observed behavior of the proposed empirical coherence estimator indicates that the latter could be exploited to assess the approximation error in real time.

The application to more complex material constitutive laws of industrial interest and involving a larger number of parameters [32] corroborate the aforementioned results in terms of compactness and accuracy of the surogate models. Surrogate models have the potential to transform the way of carrying out parametric studies in material science. In particular, [32] demonstrates that the exploitation of

models based on the Multiple TT approach simplifies the process of calibration of constitutive laws. Future work will investigate the combination of the proposed method with "usual" model order reduction techniques such as hyper-reduction [33] in order to take into account the space dimension.

Author Contributions: Conceptualization, D.R.; Methodology, C.O., D.R. and J.C.; Software, C.O.; Validation, C.O.; Formal Analysis, C.O. and J.C.; Investigation, C.O.; Resources, J.C.; Data Curation, C.O.; Writing - Original Draft Preparation, C.O.; Writing - Review & Editing, D.R. and J.C.; Visualization, C.O.; Supervision, D.R. and J.C.; Project Administration, J.C.; Funding Acquisition, D.R. and J.C.;

Funding: This research was funded by the Association Nationale de la Recherche et de la Technologie (ANRT) [grant number CIFRE 2014/0923].

Acknowledgments: The authors gratefully acknowledge fruitful discussions with Safran Helicopter Engines.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

DAE	Differential Algebraic Equation
DEIM	Discrete Empirical Interpolation Method
DOAJ	Directory of Open Access Journals
EIM	Empirical Interpolation Method
PGD	Proper Generalized Decomposition
POD	Proper Orthogonal Decomposition
PSD	Pseudo-Skeleton Decomposition
SVD	Singular Value Decomposition
TT	Tensor Train

Appendix A Elasto-viscoplastic model

The application case consists of a nonlinear constitutive law in elasto-viscoplasticity [34,35] linking the following time-dependent mechanical variables:

- The strain tensor: $\underset{\sim}{\varepsilon} = \underset{\sim}{\varepsilon}_e + \underset{\sim}{\varepsilon}_{vp}$ [Dimensionless] (sum of an elastic part and a viscoplastic part);
- The stress tensor: $\underset{\sim}{\sigma}$ [MPa];
- An internal hardening variable: $\underset{\sim}{X}$ [MPa];
- The cumulative viscoplastic deformation: p [Dimensionless].

where $\underset{\sim}{\varepsilon}$, $\underset{\sim}{\varepsilon}_e$, $\underset{\sim}{\varepsilon}_{vp}$, $\underset{\sim}{\sigma}$ and $\underset{\sim}{X}$ are second order tensors in $\mathbb{R}^{3\times3}$.

The hypotheses of the infinitesimal strain theory are assumed to hold.

The model involves eight material coefficients: E, ν, n, K, R_0, Q, b and C. The Young and Poisson coefficients are set to $E = 200\,000$ MPa and $\nu = 0.3$. Table A1 presents the range of variation of the other material coefficients considered as inputs parameters of the model.

Table A1. Parameter range of variations considered in the model. When applicable, the unit is indicated between brackets.

	n	K [MPa.s^{-n}]	R_0 [MPa]	Q [MPa]	b	C [MPa]
Lower bound	2	100	1	1	0.02	150
Upper bound	12	10 000	200	2 000	2 000	150 000

Appendix A.0.1 System of equations

The elastic behavior is governed by:

$$\underset{\sim}{\sigma} = \frac{E}{1+\nu}\left(\underset{\sim}{\varepsilon}_e + \frac{\nu}{1-2\nu} Tr\left(\underset{\sim}{\varepsilon}_e\right)\underset{\sim}{\mathbb{I}}\right) \tag{A1}$$

Peer-reviewed version available at Math. Comput. Appl. 2019, 24, 17; doi:10.3390/mca24010017

The viscoplastic behavior is described by the Norton flow rule (A2) formulated with the von Mises criterion (A5). The yield function and the normal to the yield function are given by (A3) and (A4). (A6) gives the definition of the deviatoric stress tensor involved in (A5).

$$\frac{d}{dt}\underset{\sim}{\varepsilon}_{vp} = \underset{\sim}{N}\left(\frac{f}{K}\right)_+^n \tag{A2}$$

$$f = J\left(\underset{\sim}{\sigma}^D - \underset{\sim}{X}\right) - R \tag{A3}$$

$$\underset{\sim}{N} = \frac{3}{2}\frac{\underset{\sim}{\sigma}^D - \underset{\sim}{X}}{J\left(\underset{\sim}{\sigma}^D - \underset{\sim}{X}\right)} \tag{A4}$$

$$J\left(\underset{\sim}{\sigma}^D - \underset{\sim}{X}\right) = \sqrt{\frac{3}{2}\left(\underset{\sim}{\sigma}^D - \underset{\sim}{X}\right) : \left(\underset{\sim}{\sigma}^D - \underset{\sim}{X}\right)} \tag{A5}$$

$$\underset{\sim}{\sigma}^D = \underset{\sim}{\sigma} - \frac{1}{3}Tr\left(\underset{\sim}{\sigma}\right)\underset{\sim}{\mathbb{I}} \tag{A6}$$

where $(.)_+$ denotes the positive part function.

The operator **:** denotes the contracted product defined as:

$$\underset{\sim}{Z}_1 : \underset{\sim}{Z}_2 = \sum_{i=1}^{3}\sum_{j=1}^{3} Z_1^{ij} Z_2^{ij} \quad \text{for} \quad \underset{\sim}{Z}_1, \underset{\sim}{Z}_2 \in \mathbb{R}^{3\times 3}$$

The nonlinear isotropic hardening is modeled by (A7) where (A8) gives the viscoplastic cumulative rate.

$$R = R_0 + Q\left(1 - e^{-bp}\right) \tag{A7}$$

$$\frac{dp}{dt} = \sqrt{\frac{2}{3}\frac{d}{dt}\underset{\sim}{\varepsilon}_{vp} : \frac{d}{dt}\underset{\sim}{\varepsilon}_{vp}} \tag{A8}$$

Finally the linear kinematic hardening is given by:

$$\underset{\sim}{X} = \frac{2}{3}C\underset{\sim}{\varepsilon}_{vp} \tag{A9}$$

The case of a uniaxial cyclic tensile testing driven by deformation is considered. The loading is applied by imposing $\varepsilon^{11}(t)$ with the pattern shown in Figure A1 and $\sigma^{12}(t) = \sigma^{13}(t) = \sigma^{23}(t) = \sigma^{22}(t) = \sigma^{33}(t) = 0$.

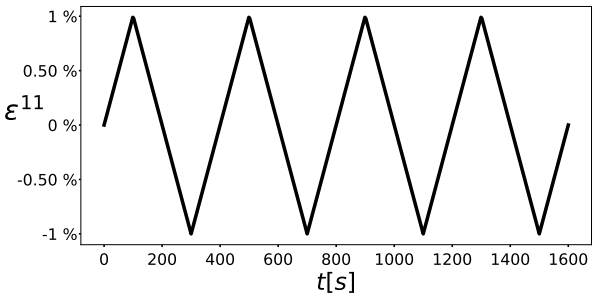


Figure A1. The applied strain component $\varepsilon^{11}(t)$ consists of a triangular pattern of period 400s with a peak-to-peak amplitude of 2% centered in 0.

The initial conditions for the internal variables are:

$$p(t=0) = 0 \quad \text{and} \quad \underset{\sim}{\boldsymbol{X}}(t=0) = \underset{\sim}{\boldsymbol{0}}$$

The model is highly nonlinear. First the isotropic hardening law introduces an exponential nonlinearity. The most significant nonlinearity arises from the Norton law (A2) featuring the positive part function. Capturing the resulting threshold effect is particularly challenging for surrogate models.

References

1. Ghighi, J.; Cormier, J.; Ostoja-Kuczynski, E.; Mendez, J.; Cailletaud, G.; Azzouz, F. A microstructure sensitive approach for the prediction of the creep behaviour and life under complex loading paths. *Technische Mechanik* **2012**, *32*, 205–220.
2. Le Graverend, J.B.; Cormier, J.; Gallerneau, F.; Villechaise, P.; Kruch, S.; Mendez, J. A microstructure-sensitive constitutive modeling of the inelastic behavior of single crystal nickel-based superalloys at very high temperature. *International Journal of Plasticity* **2014**, *59*, 55–83.
3. Ladevèze, P.; Nouy, A. On a multiscale computational strategy with time and space homogenization for structural mechanics. *Computer Methods in Applied Mechanics and Engineering* **2003**, *192*, 3061 – 3087. Multiscale Computational Mechanics for Materials and Structures.
4. Ammar, A.; Mokdad, B.; Chinesta, F.; Keunings, R. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. *Journal of Non-Newtonian Fluid Mechanics* **2006**, *139*, 153 – 176.
5. Nouy, A. A priori model reduction through Proper Generalized Decomposition for solving time-dependent partial differential equations. *Computer Methods in Applied Mechanics and Engineering* **2010**, *199*, 1603 – 1626.
6. Hitchcock, F.L. The Expression of a Tensor or a Polyadic as a Sum of Products. *Journal of Mathematics and Physics* **1927**, *6*, 164–189.
7. Khoromskij, B.N. Tensors-structured numerical methods in scientific computing: Survey on recent advances. *Chemometrics and Intelligent Laboratory Systems* **2012**, *110*, 1–19.
8. Grasedyck, L.; Kressner, D.; Tobler, C. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen* **2013**, *36*, 53–78.
9. Bigoni, D.; Engsig-Karup, A.P.; Marzouk, Y.M. Spectral Tensor-Train Decomposition. *SIAM Journal on Scientific Computing* **2016**, *38*, A2405–A2439.
10. Harshman, R.A. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics* **1970**, *16*, 1–84.
11. Kiers, H.A. Towards a standardized notation and terminology in multiway analysis. *Journal of chemometrics* **2000**, *14*, 105–122.
12. Tucker, L.R. The extension of factor analysis to three-dimensional matrices. *Contributions to mathematical psychology* **1964**, *110119*.
13. Hackbusch, W.; Kühn, S. A new scheme for the tensor representation. *Journal of Fourier analysis and applications* **2009**, *15*, 706–722.
14. Oseledets, I.; Tyrtyshnikov, E. TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications* **2010**, *432*, 70–88.
15. Oseledets, I.V. Tensor-Train Decomposition. *SIAM Journal on Scientific Computing* **2011**, *33*, 2295–2317.
16. Savostyanov, D.; Oseledets, I. Fast adaptive interpolation of multi-dimensional arrays in tensor train format. Multidimensional (nD) Systems (nDs), 2011 7th International Workshop on, 2011, pp. 1–8.
17. Savostyanov, D.V. Quasioptimality of maximum-volume cross interpolation of tensors. *Linear Algebra and its Applications* **2014**, *458*, 217–244.
18. Sirovich, L. Turbulence and the Dynamics of Coherent Structures. Part 1: Coherent Structures. *Quarterly of Applied Mathematics* **1987**, *45*, 561–571.
19. Everson, R.; Sirovich, L. Karhunen-Loève procedure for gappy data. *Journal of the Optical Society of America A* **1995**, *12*, 1657–1664.
20. Maday, Y.; Nguyen, N.C.; Patera, A.T.; Pau, S.H. A general multipurpose interpolation procedure: the magic points. *Communications on Pure and Applied Analysis* **2009**, *8*, 383–404.

21. Barrault, M.; Maday, Y.; Nguyen, N.C.; Patera, A.T. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *Comptes-Rendus Mathématiques* **2004**, *339*, 667–672.
22. Ryckelynck, D.; Lampoh, K.; Quilicy, S. Hyper-reduced predictions for lifetime assessment of elasto-plastic structures. *Meccanica* **2016**, *51*, 309–317.
23. Carlberg, K.; Farhat, C.; Cortial, J.; Amsallem, D. The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics* **2013**, *242*, 623–647.
24. Tyrtyshnikov, E.; Goreinov, S.; Zamarashkin, N. Pseudo-Skeleton Approximations. Technical report, Institute of Numerical Mathematics of the Russian Academy of Sci., Leninski Prosp. 32-A Moscow 117334, Russia, 1993.
25. Bebendorf, M. Approximation of boundary element matrices. *Numerische Mathematik* **2000**, *86*, 565–589.
26. Berry, M.W.; Pulatova, S.A.; Stewart, G. Algorithm 844: Computing sparse reduced-rank approximations to sparse matrices. *ACM Transactions on Mathematical Software (TOMS)* **2005**, *31*, 252–269.
27. Stewart, G. Four algorithms for the the efficient computation of truncated pivoted QR approximations to a sparse matrix. *Numerische Mathematik* **1999**, *83*, 313–323.
28. Golub, G.H.; Van Loan, C.F. *Matrix computations*, 4 ed.; The Johns Hopkins University Press, 2013.
29. Halton, J.H. On the Efficiency of Certain Quasi-random Sequences of Points in Evaluating Multi-dimensional Integrals. *Numer. Math.* **1960**, *2*, 84–90.
30. Drmac, Z.; Gugercin, S. A New Selection Operator for the Discrete Empirical Interpolation Method—Improved A Priori Error Bound and Extensions. *SIAM Journal on Scientific Computing* **2016**, *38*, A631–A648.
31. Chaturantabut, S.; Sorensen, D.C. Nonlinear Model Reduction via Discrete Empirical Interpolation. *SIAM Journal on Scientific Computing* **2010**, *32*, 2737–2764.
32. Olivier, C. Décompositions tensorielles et factorisations de calculs intensifs appliquées à l'identification de modèles de comportement non linéaire. PhD thesis, PSL Reasearch University, 2017.
33. Ryckelynck, D.; Vincent, F.; Cantournet, S. Multidimensional a priori hyper-reduction of mechanical models involving internal variables. *Computer Methods in Applied Mechanics and Engineering* **2012**, *225–228*, 28–43.
34. Besson, J.; Cailletaud, G.; Chaboche, J.L.; Forest, S. *Non-linear mechanics of materials*, 1 ed.; Vol. 167, Springer Netherlands, 2010.
35. Lemaitre, J.; Chaboche, J.L. *Mechanics of solid materials*; Cambridge University Press, 1994.