

# Méthode de décomposition tensorielle pour la calibration de lois de comportement en sciences des matériaux

C. OLIVIER<sup>1,2</sup>, D. RYCKELYNCK<sup>1</sup>, J. CORTIAL<sup>2</sup>, C. REY<sup>2</sup>

<sup>1</sup> Mines ParisTech, PSL - Research University, Centre des matériaux, Evry, {clement.olivier;david.ryckelynck}@mines-paristech.fr

<sup>2</sup> Safran Tech, Safran Group, Magny-Les-Hameaux, {clement.olivier;julien.cortial,christian.rey}@safrangroup.com

**Résumé** — On souhaite construire un métamodèle à représentation tensorielle sur un vaste espace paramétrique de lois de comportement élastoviscoplastiques pour en faciliter la calibration. Un train de tenseurs est construit pour chaque variable mécanique à l'aide de la méthode TT-cross. Pour avoir un échantillonnage parcimonieux du domaine paramétrique, la méthode est complétée par une approximation des fibres multidimensionnelles par la Gappy POD sur des bases réduites issues de la snapshot POD. Les performances (en ligne et hors ligne) de la méthode sont illustrées sur un modèle à 6 paramètres.

**Mots clés** — mécanique de matériaux, modèles de substitution, décomposition tensorielle

## 1 Introduction

Les moteurs d'hélicoptères sont conçus pour être capables de fonctionner en cas d'urgence au delà de leur domaine d'utilisation nominal sur de courtes durées. Lors de ce type d'épisode, les aubes de turbines hautes pression sont soumises à des contraintes thermomécaniques extrêmes. La modélisation du comportement anisotherme nécessaire à la bonne compréhension de leurs déformations et de leur endommagement nécessite de disposer de lois matériaux complexes. Ces modèles numériques sont décrits par des systèmes d'équations algébriques non-linéaires couplées et paramétrées par un grand nombre de coefficients matériaux (de l'ordre de la vingtaine) qui doivent être calibrés au préalable. Cette étape est capitale pour garantir la fidélité des prédictions des résultats numériques attendues vis-à-vis des comportements physiques.

La procédure d'identification consiste à trouver des valeurs de paramètres pour lesquels les essais expérimentaux coïncident avec les simulations numériques du modèle. Toutefois, une exploration exhaustive du domaine paramétrique n'est pas envisageable, étant donné la croissance exponentielle du nombre de simulations à réaliser en fonction du nombre de paramètres. Ce constat est d'autant plus vrai que l'effort nécessaire pour effectuer un seul calcul est important.

Une technique communément mise en œuvre pour pallier cette difficulté est d'introduire un modèle de substitution significativement plus rapide à évaluer que le modèle de référence et qui exprime de manière suffisamment précise la relation entre les paramètres à identifier et sorties d'intérêt. Cette approche rend possible l'exploration interactive de l'espace paramétrique et la visualisation en temps réel de la réponse du modèle numérique sous forme de courbes, surfaces, ou autres indicateurs exploitables par des experts en mécanique des matériaux. La Proper Generalized Decomposition (PGD) [1] permet de construire des modèles de substitution fondés sur une représentation tensorielle fournissant une bonne approximation de l'ensemble des solutions pour toutes combinaisons admissibles de paramètres. Ce type de méthode a montré son efficacité en permettant la mise en place d'«abaques numériques». Cependant, les non linéarités qui apparaissent dans nos applications ne permettent pas de suivre une telle approche.

Le travail présenté vise à développer une alternative non-intrusive à la PGD pour la construction d'une représentation tensorielle des solutions. L'approche proposée consiste à construire, pour chacune des variables mécaniques d'intérêt qui apparaissent dans les équations, un *train de tenseurs* en utilisant une version modifiée de l'algorithme TT-cross [2].

Une fois les décompositions en train de tenseurs construites, la représentation compacte ainsi obtenue permet, lors de la phase en ligne (*Online*), un accès en temps réel à une approximation de l'ensemble des solutions du modèle pour toutes les variables mécaniques et quel que soit le choix du jeu de paramètres.

La principale contribution du présent travail concerne essentiellement une adaptation de la phase

hors ligne (*Offline*) de construction de la décomposition en train de tenseurs. Pour répondre à l'enjeu capital de limiter le nombre d'appels au modèle physique de référence, notre approche s'appuie sur une exploration parcimonieuse du domaine paramétrique. L'approximation des solutions utilise une approche de type *Gappy POD* [3] permettant de reconstruire les solutions globalement à partir de leur connaissance sur un domaine partiel.

La méthodologie proposée est illustrée sur un exemple concret d'un modèle de loi matériau élasto-viscoplastique hautement non linéaire. On observe que la construction du modèle de substitution, extrêmement rapide par rapport à un plan factoriel, permet d'aboutir à une approximation suffisamment précise par rapport au modèle haute-fidélité et dans le cadre d'une utilisation pour un travail de calibration.

## 2 Formulation du problème

### 2.1 Modèle élasto-viscoplastique

On se place dans le cadre des petites déformations. Le modèle considéré est une loi de comportement élasto-viscoplastique dans laquelle interviennent les variables mécaniques, dépendantes du temps ( $t \in \mathcal{T}$ ), suivantes :

- Le tenseur des déformations :  $\underline{\varepsilon} = \underline{\varepsilon}_e + \underline{\varepsilon}_{vp}$  (somme d'une partie élastique et viscoplastique) ;
- Le tenseur des contraintes :  $\underline{\sigma}$  ;
- Une variable interne d'écrouissage :  $\underline{X}$  ;
- La déformation plastique cumulée :  $p$ .

Huits coefficients matériaux apparaissent dans la loi.

Le module d'Young ainsi que le coefficient de Poisson sont fixés à  $E = 200\,000$  MPa et  $\nu = 0.3$ . Le tableau suivant présente les intervalles de variation des coefficients matériaux que l'on considère comme paramètres d'entrées du modèle :

	$n$	$K$	$R_0$	$Q$	$b$	$C$
Minimum	2	100 MPa	1 MPa	1	0.02	150 MPa
Maximum	12	10000 MPa	200 MPa	2000	2000	150000 MPa

D'un point de vue construction d'un modèle de substitution, ce qui importe, c'est la relation entre les paramètres (et le temps) (entrées du modèle) et les variables mécaniques (sorties du modèle) :

$$(n, K, R_0, Q, b, C, t) \mapsto (\underline{\varepsilon}, \underline{\sigma}, \underline{X}, p)$$

Le comportement élastique est donné par (1)

$$\underline{\sigma} = \frac{E}{1+\nu} \left( \underline{\varepsilon}_e + \frac{\nu}{1-2\nu} \text{Tr}(\underline{\varepsilon}_e) \mathbb{I} \right) \quad (1)$$

La loi de Norton (2) donne le comportement viscoplastique, avec la fonction de charge donnée par (3) et la normale à la fonction de charge (4). On utilise un critère de Von Mises (5) en rappelant la définition de la contrainte déviatorique (6).

$$\frac{d}{dt} \underline{\varepsilon}_{vp} = N \left\langle \frac{f}{K} \right\rangle^n \quad (2)$$

$$f = J(\underline{\sigma}^D - \underline{X}) - R \quad (3)$$

$$N = \frac{3}{2} \frac{\underline{\sigma}^D - \underline{X}}{J(\underline{\sigma}^D - \underline{X})} \quad (4)$$

$$J(\underline{\sigma}^D - \underline{X}) = \sqrt{\frac{3}{2} (\underline{\sigma}^D - \underline{X}) : (\underline{\sigma}^D - \underline{X})} \quad (5)$$

$$\underline{\sigma}^D = \underline{\sigma} - \frac{1}{3} \text{Tr}(\underline{\sigma}) \mathbb{I} \quad (6)$$

L'écrouissage isotrope non linéaire est modélisé par (7), où (8) définit le taux de déformation viscoplastique.

$$R = R_0 + Q \left(1 - e^{-bp}\right) \quad (7)$$

$$\frac{dp}{dt} = \sqrt{\frac{2}{3}} \frac{d}{dt} \varepsilon_{vp} : \frac{d}{dt} \varepsilon_{vp} \quad (8)$$

Enfin, on ajoute un écrouissage cinématique linéaire donné par (9).

$$\underline{X} = \frac{2}{3} C \varepsilon_{vp} \quad (9)$$

On modélise un essai de traction cyclique (motif triangulaire) uniaxial piloté en déplacement avec une déformation longitudinale maximale de 1%. Il n'y a pas de contrainte dans les directions radiales. Les variables internes sont initialement nulles :

$$p(t=0) = 0 \quad \text{and} \quad \underline{X}(t=0) = \underline{0}$$

Le modèle est résolu sur un maillage constitué d'une maille élémentaire carrée avec 4 noeuds, de 1mm de côté.

### Remarques sur le modèle

La loi de comportement modélisée présente d'importantes non-linéarités : d'une part à cause de la loi d'écrouissage isotrope, mais également et principalement en raison de la présence de la loi de Norton définie par une fonction de seuil (partie positive  $\langle \cdot \rangle$ ). Cette non linéarité plus importante qu'une simple non linéarité polynomiale est particulièrement difficile à reproduire lors de la construction d'un modèle de substitution.

## 2.2 Représentation tensorielle

On note  $\mathbf{Y}^\alpha$  avec  $\alpha \in \llbracket 1, N \rrbracket$  l'ensemble des sorties (i.e : les variables mécaniques d'intérêt) exprimées par les équations. Par définition,  $\mathbf{Y}^\alpha$  dépend du temps et peut correspondre à un scalaire (ex : la déformation plastique cumulée  $p$ ), un vecteur (ex : les déplacements  $u$ ) ou une tenseur (ex : le tenseur des contraintes  $\underline{\sigma}$ ). On peut donc exprimer  $\mathbf{Y}^\alpha$  comme une fonction multivariée dépendante du temps et des paramètres matériaux  $(\mu_1, \dots, \mu_{\tilde{d}}) \in \mathcal{D}_1 \times \dots \times \mathcal{D}_{\tilde{d}}$  telle que

$$\mathbf{Y}^\alpha(t; \boldsymbol{\mu}) = \begin{pmatrix} Y_1^\alpha(t; \boldsymbol{\mu}) \\ \vdots \\ Y_{N^\alpha}^\alpha(t; \boldsymbol{\mu}) \end{pmatrix}$$

avec les  $\mathcal{D}_i$  correspondant aux intervalles de définition pour chaque paramètre.

La représentation tensorielle de l'ensemble des solutions se fait par une approche discrète. L'échantillonnage temporel  $\{t^{(1)}, \dots, t^{(n_t)}\}$  est fourni par le schéma d'intégration du système d'équations. On définit l'espace paramétrique finement discrétisé  $D = D_1 \times \dots \times D_{\tilde{d}}$  tel que

$$D_i = \{\mu_i^{(k)} \in \mathcal{D}_i \mid \forall k \in I_i\} \quad \text{pour } i = 1, \dots, \tilde{d}$$

Les discrétisations permettent de définir pour chaque  $\alpha \in \llbracket 1, N \rrbracket$ , un tenseur  $\mathcal{A}^\alpha$  de dimension  $\tilde{d} + 2$  correspondant à la variable mécanique  $\mathbf{Y}^\alpha$  de telle sorte que

$$\mathcal{A}^\alpha(i_1, \dots, i_{\tilde{d}}, i_t, i_{comp}) = Y_{i_{comp}}^\alpha(\mu_1^{(i_1)}, \dots, \mu_{\tilde{d}}^{(i_{\tilde{d}})}, t^{(i_t)}) \quad \forall (i_1, \dots, i_{\tilde{d}}, i_t, i_{comp}) \in I_1 \times \dots \times I_{\tilde{d}} \times \llbracket 1, n_t \rrbracket \times \llbracket 1, N^\alpha \rrbracket$$

Ces tenseurs sont définis de manière implicite dans le sens où l'accès à leurs éléments nécessite de résoudre le système d'équations pour chaque jeu de valeurs de paramètres.

### 3 Décomposition tensorielle

L'approche par force brute, consistant à calculer l'ensemble des composantes des tenseurs  $\mathcal{A}^\alpha$  est clairement inenvisageable en terme de temps de calcul et d'espace de stockage lorsque le nombre de paramètres augmente.

L'idée proposée est de construire pour chaque  $\alpha$ , une approximation sous la forme d'un train de tenseurs  $\mathcal{T}^\alpha$ . La compression liée à la décomposition en train de tenseurs permet de résoudre le problème de stockage mémoire. Le problème du temps de calcul est résolu par l'utilisation d'une variation de l'algorithme TT-cross permettant une construction qui se contente d'une exploration parcimonieuse des tenseurs  $\mathcal{A}^\alpha$ .

#### 3.1 Format en train de tenseurs (format TT)

On peut montrer que tout tenseur admet une décomposition exacte en train de tenseurs [2]. Dans ce format, les éléments du tenseur ne sont pas directement stockés en mémoire mais sont définis de manière implicite sous forme d'une série de produits matrices/matrices et matrices/vecteurs entre les facteurs (*tensor carriages*)  $G_k(i_k)$  :

$$\mathcal{T}(i_1, \dots, i_d) = G_1(i_1)G_2(i_2) \dots G_d(i_d)$$

Ces opérations algébriques élémentaires (illustrées par la figure 1) permettent un accès temps réel aux éléments du tenseur, ce qui est attendu de notre modèle de substitution.

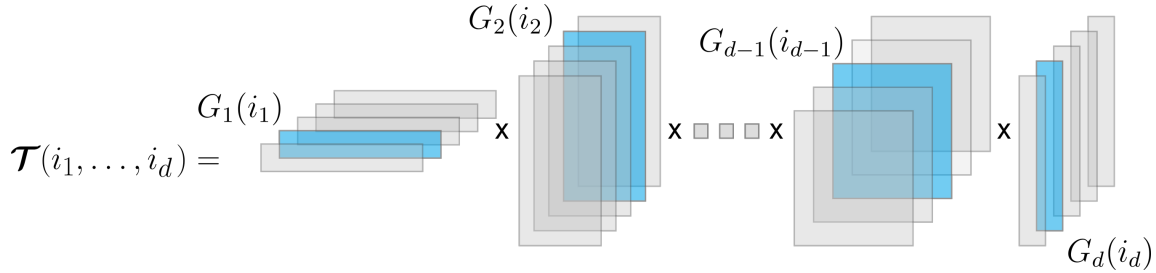


FIGURE 1 – Illustration de l'évaluation d'un élément du train de tenseurs

Pour un tenseur d'ordre  $d$ , le stockage du train de tenseurs consiste en  $d$  ensembles (pour  $k \in \llbracket 1, d \rrbracket$ ) constitués de  $n_k$  matrices (ou vecteurs pour  $k = 1$  et  $k = d$ ) :

$$\{G_k(1), G_k(2), \dots, G_k(n_k)\}$$

En supposant que chaque dimension soit de taille  $n$ , le format TT donne accès à  $n^d$  éléments mais ne nécessite de stocker que  $dn^2r$  éléments, où  $r$  est lié au rang du tenseur dans les différentes directions. Dans la pratique, ce rang est faible. Le fait de passer d'une dépendance en puissance de  $d$  à une dépendance linéaire en  $d$  permet de vaincre la malédiction de la dimension.

#### 3.2 Algorithme TT-cross

Pour un tenseur donné  $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ , l'algorithme TT-cross [2], permet de construire une représentation (exacte ou approchée) en train de tenseurs notée  $\mathcal{T}$ .

##### Notations

Le déploiement matriciel (*matricization*) d'un tenseur est une opération qui consiste à réindexer ses éléments afin de l'interpréter comme une matrice. Le  $q$ -ème déploiement de  $\mathcal{A}$ , noté  $[\mathcal{A}]_q$  consiste à regrouper les indices de  $\mathcal{A}$  en deux groupes (les  $q$  premiers indices et les  $d - q$  derniers indices) de telle sorte que ces nouveaux multi-indices numérotent respectivement les lignes et colonnes de la matrice  $[\mathcal{A}]_q$ .

$$[\mathcal{A}]_q = \left[ \mathcal{A}((i_1, \dots, i_q), (i_{q+1}, \dots, i_d)) \right] \in \mathbb{R}^{(n_1 \dots n_q) \times (n_{q+1} \dots n_d)}$$

Les indices regroupés sont mis entre parenthèses. Les crochets du terme de droite indique que l'on fait référence au tenseur complet (la matrice dans ce cas) et non au scalaire correspondant à l'élément  $(i_1, \dots, i_d)$  du tenseur.

L'algorithme TT-cross pour un tenseur d'ordre  $d$  est :

### Initialisation

On définit le tenseur  $\mathcal{A}^{(1)} \in \mathbb{R}^{s_0 \times n_1 \times \dots \times n_d}$  avec  $s_0 = 1$  tel que :

$$\mathcal{A} = [\mathcal{A}^{(1)}((\alpha_0, i_1), i_2, \dots, i_d)]$$

Uniquement les deux premiers indices sont regroupés.

### Étapes $k = 1, \dots, (d-1)$

On considère la matrice  $A_k$  comme le second déploiement du tenseur  $\mathcal{A}^{(k)} \in \mathbb{R}^{s_{k-1} \times n_k \times \dots \times n_d}$  préalablement défini :

$$A_k = [\mathcal{A}^{(k)}]_2 \in \mathbb{R}^{(s_{k-1}n_k) \times (n_{k+1} \dots n_d)} \quad (10)$$

On construit une approximation  $T_k$  de rang  $r_k \ll (s_{k-1}n_k)$  de la matrice  $A_k$  décomposée de la manière suivante :

$$A_k \simeq T_k = G_k \widehat{A}_k \quad (11)$$

avec

$$G_k \in \mathbb{R}^{(s_{k-1}n_k) \times s_k} \quad \text{et} \quad \widehat{A}_k \in \mathbb{R}^{s_k \times (n_{k+1} \dots n_d)}$$

notons que  $s_k \geq r_k$

L'étape  $k$  s'achève par la définition du tenseur  $\mathcal{A}^{(k+1)} \in \mathbb{R}^{s_k \times n_{k+1} \times \dots \times n_d}$  :

$$[\mathcal{A}^{(k+1)}]_1 = \widehat{A}_k$$

### Étape finale $k = d$

L'étape finale consiste à définir :

$$G_d = \mathcal{A}^{(d)} \in \mathbb{R}^{s_{d-1} \times n_d}$$

### Passage au format TT

L'algorithme fait apparaître la formule de récurrence suivante :

$$[\mathcal{A}^{(k)}]_2 \simeq G_k [\mathcal{A}^{(k+1)}]_1$$

où le passage d'un déploiement à un autre joue un rôle essentiel.

On peut montrer que les matrices  $G_k$  correspondent, à réindexation près, aux facteurs du train de tenseurs.

## 3.3 Approximation de faible rang

L'algorithme TT-cross consiste à réaliser de manière itérative des approximations de faible rang (équation (11)) sur les matrices  $A_k$  (définition (10)). Les matrices  $A_k$  sont définies à partir du déploiement des tenseurs  $\mathcal{A}^{(k)}$  issus eux-mêmes des approximations précédentes. La définition de ces tenseurs dépend donc directement du choix de l'approximation de faible rang. Quel que soit ce choix, la construction du tenseur  $\mathcal{A}^{(k)}$  requiert le calcul d'éléments du tenseur initial  $\mathcal{A}$ , et implique donc de résoudre le système d'équations pour un certain nombre de valeurs de paramètres.

Grâce à l'utilisation de la méthode Gappy POD [3] pour l'approximation de faible rang, il est possible d'écrire les matrices  $A_k$  directement comme des sélections de fibres du tenseur  $\mathcal{A}$ . Cela permet de limiter le nombre d'appels au modèle haute-fidélité pour l'assemblage des matrices.

Pour mettre en place la méthode Gappy POD, il est nécessaire de construire préalablement à chaque étape  $k$ , une base réduite de l'espace considéré. La construction de cette base est réalisée sur le principe de la Snapshot POD [4].

Une séquence de Halton [5] est utilisée pour générer le plan d'expérience correspondant aux snapshots. Le choix des points d'estimation permettant de reconstruire l'approximation sur la base POD est effectué via l'algorithme Q-DEIM [6].

### Prise en compte des variables mécaniques multiples $\mathcal{A}^\alpha$

Par définition des tenseurs  $\mathcal{A}^\alpha$ , la résolution du système d'équations sous-jacent pour un jeu de paramètres (correspondant aux indices  $(i_1, \dots, i_d)$ ) donne accès pour tous les  $\alpha \in \llbracket 1, N \rrbracket$  aux tenseurs complets :

$$\mathcal{A}^\alpha(i_1, \dots, i_d, :, :) \quad (\text{Notation Matlab})$$

Les résolutions coûteuses sont rentabilisées par la construction simultanée des décompositions en train de tenseurs correspondant à chaque variable mécanique. Le masque utilisé pour la reconstruction Gappy POD est alors défini comme l'agrégation des points d'interpolation obtenus séparément pour chaque variable mécanique.

## 4 Résultats

La méthode a été appliquée pour construire une représentation en train de tenseurs des variables mécaniques  $\underline{\varepsilon}$ ,  $\underline{\varepsilon}_{vp}$ ,  $\underline{\sigma}$ ,  $p$ . Pour une variable  $\square$ , on note  $\square^{HF}$  et  $\square^{TT}$  lorsqu'elle correspond respectivement aux solutions du système d'équations et au format TT.

Les décompositions ont été réalisées en discrétisant les intervalles de chaque paramètre en des grilles régulières de 30 points. La discrétisation temporelle est constituée d'une grille régulière de 537 points.

### 4.1 Indicateurs de performance

Le temps de calcul pour obtenir les solutions de la loi de comportement est d'environ 1s, en utilisant le solver Zset (<http://zset-software.com>). La phase hors ligne a nécessité de résoudre 360000 fois le système d'équations et a duré 9 heures sur un cluster de 16 cœurs.

Une fois construit, le temps de réponse du modèle de substitution est de 10 ms sur un ordinateur de bureau. Une exploitation temps réel est donc envisageable.

Le stockage complet du train de tenseurs, permettant d'obtenir une approximation des solutions pour tout jeu de paramètres, nécessite de stocker 8 500 000 valeurs (virgule flottante, double précision). En comparaison, pour un jeu de paramètres donné, la solution (contenant toutes les variables mécaniques en fonction du temps) représente 10 000 valeurs.

### 4.2 Mesures d'erreurs

On estime a posteriori l'erreur d'approximation du modèle de substitution en comparant les résultats fournis par le modèle de substitution et haute-fidélité sur un échantillon de solutions calculées pour 20000 jeux de paramètres choisis aléatoirement (loi uniforme sur chaque intervalle de paramètre).

On définit pour chaque simulation et chaque variable mécanique, une mesure d'erreur à partir de la norme  $L^2$ .

Pour les variables scalaire et tensorielles, on définit respectivement la norme  $L^2$  :

$$\|x\|_{L^2}^2 = \int_{\mathcal{T}} x^2 dt \quad \text{et} \quad \|\underline{x}\|_{L^2}^2 = \int_{\mathcal{T}} \underline{x} : \underline{x} dt$$

Les histogrammes sur la figure 2 donnent, pour le tenseur des contraintes et la déformation plastique cumulée, la répartition des simulations en fonction des mesures d'erreur suivantes :

$$e_\sigma = \frac{\|\underline{\sigma}^{HF} - \underline{\sigma}^{TT}\|_{L^2}}{\|\underline{\sigma}^{HF}\|_{L^2}}, \quad e_p = \frac{\|p^{HF} - p^{TT}\|_{L^2}}{\|\underline{\varepsilon}^{HF}\|_{L^2}}$$

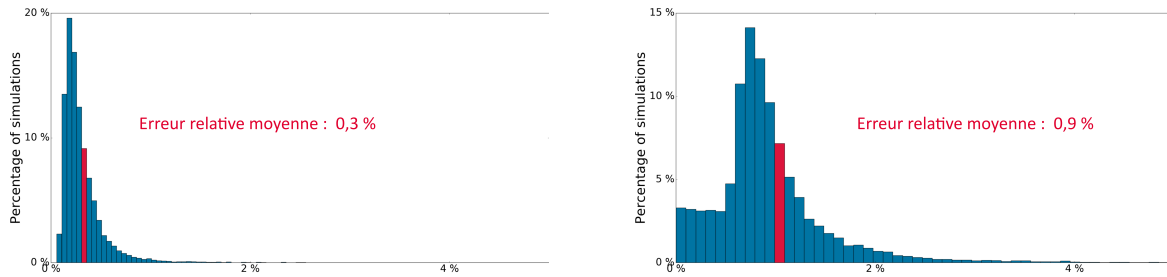


FIGURE 2 – Histogrammes de distribution des erreurs pour  $e_\sigma$  (à gauche) et  $e_p$  (à droite).

Les résultats mettent en valeur la très bonne précision du metamodelle.

Nous avons choisi au dénominateur de l'erreur relative  $e_p$  la norme des déformations totales car pour certaines valeurs des paramètres, la solution produite est élastique et donc à déformation plastique nulle.

## 5 Perspectives

La méthode proposée est non intrusive et la TT-cross peut être appliquée en dimension arbitraire. Cependant, il est probable que le nombre de simulations à réaliser dépende fortement de la complexité de la loi de comportement et du nombre de paramètres. La pertinence de l'approche développée pour des lois matériaux plus complexes, nécessaires à la simulation des aubes de turbines d'hélicoptères, est en cours d'évaluation.

## Références

- [1] Francisco Chinesta, Adrien Leygue, Felipe Bordeu, Jose V. Aguado, Elias Cueto, David Gonzalez, Iciar Alfaro, Amine Ammar, and Antonio Huerta. PGD-based computational vademecum for efficient design, optimization and control. *Archives of Computational Methods in Engineering*, 20(1) :31–59, 2013.
- [2] Ivan Oseledets and Eugene Tyrtshnikov. TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432(1) :70–88, 2010.
- [3] Richard Everson and Lawrence Sirovich. Karhunen-Loève procedure for gappy data. *Journal of the Optical Society of America A*, 12 :1657–1664, August 1995.
- [4] Lawrence Sirovich. Turbulence and the dynamics of coherent structures. part 1 : Coherent structures. *Quarterly of Applied Mathematics*, 45(3) :561–571, October 1987.
- [5] John H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2(1) :84–90, 1960.
- [6] Zlatko Drmac and Serkan 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.