

Extensions de la méthode PGD pour les problèmes à grand nombre de paramètres

C. Paillet¹, P. Ladevèze¹ et D. Néron¹

¹ LMT, ENS Cachan, {charles.paillet,pierre.ladeveze,david.neron}@lmt.ens-cachan.fr

Résumé — Les méthodes de réduction de modèle sont des outils clés en ingénierie pour réaliser des simulations complexes en temps quasi réel. La méthode PGD (Proper Generalized Decomposition) en particulier a permis de calculer avec succès des modèles réduits pour divers problèmes de mécanique des solides, y compris non linéaires. Cependant, une limitation majeure de ces méthodes concerne le nombre de paramètres qui peuvent être pris en compte (de l'ordre de la dizaine voire vingtaine). Ce travail a pour objectif de proposer des algorithmes alternatifs pour dépasser cette limite.

Mots clés — PGD, multiéchelle, multiparamétrique.

1 Introduction

La simulation numérique est devenue un outil indispensable dans de multiples domaines de l'ingénierie. On dispose aujourd'hui de modèles de plus en plus complexes, efficaces pour décrire une large variété de phénomènes. Mais malgré les évolutions permanentes des moyens de calculs dont on dispose, la mise en œuvre numérique peut rester extrêmement coûteuse et forme un verrou majeur pour les applications industrielles. Les techniques de réduction de modèles [1] offrent un énorme potentiel de développement d'outils novateurs pour le calcul hautes performances. Elles permettent de représenter de manière simplifiée les évolutions de systèmes physiques qui nécessitent *a priori* plusieurs milliers voire quelques millions de degrés de liberté.

Les méthodes de séparation de variables consistent à choisir des décompositions de champs sous la forme d'un produit de fonctions dépendantes des différentes variables du problème. Il s'agit d'un mode de représentation parmi d'autres [2, 3] qui, en simplifiant la représentation et le stockage des variables, permet de s'affranchir de la « malédiction de la dimensionnalité ». Par exemple, un champ u dépendant de l'espace \underline{x} , et d'un nombre n de paramètres λ_i (temps, paramètres matériaux ...) peut être approximé par N modes sous la forme :

$$\underline{u}(\underline{x}, \lambda_1, \dots, \lambda_n) \approx \underline{u}_N(\underline{x}, \lambda_1, \dots, \lambda_n) = \sum_{j=1}^N \tilde{X}_j(\underline{x}) \prod_{i=1}^n x_{ij}(\lambda_i) \quad (1)$$

Les méthodes de réduction *a posteriori*, telles que la POD [4, 5] ou les reduced basis [6], nécessitent le calcul de plusieurs solutions complètes au cours d'une phase d'apprentissage afin de construire les modes utilisés pour représenter la solution (1). Au contraire, la PGD consiste à construire les modes en même temps que la résolution du problème. La méthode PGD classique introduite dans [7] est un processus itératif qui est résumé dans l'Algorithme 1. En particulier, associée à la méthode LaTIn [8], elle permet de prendre en compte les aspects non linéaires : des bibliothèques de modes réutilisables peuvent ainsi être créées [9]. Quelle que soit la technique utilisée, actuellement, le nombre de paramètres que l'on peut traiter via de telles méthodes est limité à une vingtaine environ (voir par exemple [10]), ce qui constitue un verrou scientifique majeur pour l'essor de ces techniques.

2 Problème traité

On s'intéresse à un problème de statique dépendant d'un nombre n de paramètres $\underline{\lambda} = \{\lambda_1, \dots, \lambda_n\}$. Une fois discrétisé, il s'exprime sous la forme :

$$\mathbb{A}(\boldsymbol{\lambda})\underline{X}(\boldsymbol{\lambda}) = \underline{F}(\boldsymbol{\lambda}), \quad \underline{X} \in \mathbb{R}^{n_x} \times \mathbf{I} \quad (2)$$

où n_x est le nombre de degrés de liberté spatiaux du problème et $\mathbf{I} = I_1 \times \dots \times I_n$ l'espace de discrétisation des n paramètres.

On va s'intéresser en particulier à la dépendance paramétrique de l'opérateur \mathbb{A} . On suppose sans perte de généralité que les paramètres λ_i sont définis sur des intervalles centrés en 0. On définit alors l'opérateur $\mathbb{A}^0 = \mathbb{A}(\boldsymbol{\lambda} = \mathbf{0})$. On peut alors réécrire (2) sous la forme :

$$\mathbb{A}^0(\mathbb{I} - \mathbb{D})\underline{X} = \underline{F}$$

avec $\mathbb{D} = \mathbb{I} - \mathbb{A}^{0^{-1}}\mathbb{A}$. Soit $\underline{X}_0 = \mathbb{A}^{0^{-1}}\underline{F}$. En développant $(\mathbb{I} - \mathbb{D})^{-1}$ on obtient une approximation de la solution sous la forme :

$$\underline{X} = (\mathbb{I} + \mathbb{D} + \mathbb{D}^2 + \dots)\underline{X}_0 \quad (3)$$

En pratique, plus faibles seront les variations des paramètres et plus petite sera la norme de \mathbb{D} , donc plus rapide sera la vitesse de convergence de l'approximation (3). Cette formulation composée d'un nombre fini et limité de termes illustre comment on peut théoriquement calculer la solution en fonction des paramètres, sans avoir à recourir au calcul complet de l'ensemble des solutions qui se heurte à la « malédiction de la dimensionnalité ». Cette idée de développement itératif conduit au choix du conditionneur retenu en section 4 et la nature de ses différents termes sera discutée en section 5.

3 PGD Classique

Algorithm 1 PGD

1:	for $m = 1$ to M do	▷ Number of modes
2:	Initialize \tilde{X}_m	
3:	for $k = 1$ to k_{max} do	▷ Fixed point iterations
4:	for $i = 1$ to n do	▷ Number of parameters
5:	Compute $x_{i,m} = P_m(\tilde{X}_m, \{x_{j,m}\}_{j \neq i})$	▷ Parametric problems
6:	Normalize $x_{i,m}$	
7:	end for	
8:	Compute $\tilde{X}_m = S_m(\{x_{i,m}\}_i)$	▷ Spatial problem
9:	Check convergence of $\tilde{X}_m \prod_{i=1}^n x_{i,m}$	
10:	end for	
11:	Set $\underline{X}_m = \underline{X}_{m-1} + \tilde{X}_m \prod_{i=1}^n x_{i,m}$	
12:	Check convergence	
13:	end for	

De nombreuses variantes de la méthode PGD sont possibles [11]. Pour un problème multi-paramétrique, on retiendra la méthode classique décrite par exemple dans [12]. On cherche à calculer une solution approchée du problème (2). Le premier mode PGD de la solution est recherché sous la forme :

$$\underline{X}(\boldsymbol{\lambda}) = \tilde{X} \prod_{i=1}^n x_i(\lambda_i)$$

où $\tilde{X} \in \mathcal{U}^h \equiv \mathbb{R}^{n_x}$ et $x_i \in I_i$

Sous forme variationnelle, le problème étudié devient :

$$\int_{\mathbf{I}} \underline{X}^* \left(\mathbb{A} \tilde{X} \prod_{i=1}^n x_i(\lambda_i) - \underline{b} \right) d\mathbf{I} = 0 \quad \forall \underline{X}^* \in \mathcal{U}^h \times \mathbf{I} \quad (4)$$

On peut le décomposer en $n + 1$ équations couplées, chacune dépendant d'un unique paramètre et que l'on peut résoudre via un algorithme de point fixe. En particulier, on obtient un problème uniquement spatial qu'on notera :

$$\tilde{\underline{X}} = S_1(\{x_i\}_i)$$

On obtient par ailleurs n équations paramétriques notées :

$$x_i = P_1(\tilde{\underline{X}}, \{x_j\}_{j \neq i})$$

A l'issue du point fixe, on ajoute de nouveaux modes PGD de la même manière (algorithme glouton). On peut réutiliser la formulation 4 en remplaçant $\underline{F}(\underline{\lambda})$ par l'expression du résidu à l'itération i :

$$R_i(\underline{\lambda}) = \underline{F}(\underline{\lambda}) - \mathbb{A}(\underline{\lambda})\underline{X}_m$$

où \underline{X}_m est la solution complète à l'itération m :

$$\underline{X}_m(\underline{\lambda}) = \sum_{j=1}^m \tilde{\underline{X}}_j \prod_{i=1}^n x_i^j(\lambda_i)$$

4 Extended-PGD (PGD étendue)

Cette méthode a été développée à partir de [13] et s'inspire d'une méthode du gradient conjugué préconditionnée pour réduire le nombre de résolutions spatiales complètes. En effet, c'est de très loin l'opération la plus coûteuse dans l'algorithme PGD classique lorsqu'on l'applique à des problèmes industriels à plusieurs centaines de milliers, voire quelques millions de degrés de liberté.

Les évolutions proposées dans l'algorithme Extended-PGD concernent deux points clefs :

- une séparation entre le calcul des modes spatiaux et le calcul des modes paramétriques
- un préconditionnement systématique des calculs modaux

Pour cela, on va isoler le calcul de la base des modes spatiaux de l'algorithme d'optimisation par point fixe, qui ne sera plus utilisé que pour le calcul des fonctions paramétriques. De plus, on va introduire dès l'initialisation un préconditionneur "naturel" qui permet d'obtenir des modes spatiaux plus pertinents.

4.1 Principe

Initialisation On calcule :

$$\underline{X}_0 = \mathbb{A}^{0-1} \underline{F}$$

\underline{X}_0 est le premier mode de notre solution que l'on peut décomposer sous la forme : $\underline{X}_0 = \tilde{\underline{X}}_0 x_0(\underline{\lambda})$. Le résidu associé à cette première approximation d'ordre 1 est :

$$\underline{R}_0(\underline{\lambda}) = \underline{F} - \mathbb{A}(\underline{\lambda})\underline{X}_0 \quad (5)$$

Construction de la base de Krylov On cherche une approximation d'ordre 1 du résidu $\underline{R}_0 \simeq \tilde{\underline{R}}_0 \gamma(\underline{\lambda})$. On notera cette opération :

$$1|PGD(\underline{R}_0 | \mathbb{A}_0^{-1})$$

On définit le second vecteur de notre base en normalisant au sens du produit scalaire $(A, B)_{\mathbb{A}_0^{-1}} = A^T \mathbb{A}_0^{-1} B$ le vecteur :

$$\tilde{\underline{X}}_1 = \mathbb{A}_0^{-1} \tilde{\underline{R}}_0$$

En s'inspirant de la méthode du gradient conjugué, on peut construire une base de Krylov par itérations successives en calculant les vecteurs sous la forme :

$$\tilde{\underline{X}}_i = \mathbb{A}_0^{-1} \tilde{\underline{Z}}_i \quad (6)$$

où $1|PGD(\mathbb{A} \tilde{\underline{X}}_{i-1} | \mathbb{A}_0^{-1}) = \tilde{\underline{Z}}_i z_i(\underline{\lambda})$. Un algorithme de Gram Schmidt permet d'orthogonaliser ces modes et d'obtenir après K itérations une base :

$$\mathcal{K} = \{\tilde{\underline{X}}_0, \dots, \tilde{\underline{X}}_K\}$$

On définit la norme $\|A\|_{\mathbb{M}} = \int_{\mathbf{I}} A^T \mathbb{M} A d\mathbf{I}$. Pour calculer la $1|PGD(\mathbb{A}\tilde{\mathbf{X}}_i|\mathbb{A}_0^{-1})$, on cherche à minimiser $\|\mathbb{R} - \tilde{\mathbf{R}}\gamma(\underline{\lambda})\|_{\mathbb{A}_0^{-1}}$. On obtient un système d'équations couplées que l'on choisit de résoudre par substitution. Ainsi, on ne résoud qu'une partie du système et on se limite à la résolution du problème spatial. En posant $\mathbb{R} = \int_{\mathbf{I}} \mathbb{R} \mathbb{R}^T d\mathbf{I}$, celui-ci s'exprime sous la forme d'un quotient de Rayleigh généralisé qu'on cherche à maximiser :

$$\frac{\tilde{\mathbf{R}}^T \mathbb{R} \tilde{\mathbf{R}}}{\tilde{\mathbf{R}}^T \mathbb{A}_0^{-1} \tilde{\mathbf{R}}} \quad (7)$$

Calcul des fonctions paramétriques A partir de la base de Krylov \mathcal{K} obtenue, on va chercher à exprimer la solution dans cet espace en minimisant l'erreur effectuée au sens de la norme énergétique :

$$\varepsilon = \|\mathbb{A}(\underline{\lambda}) (\tilde{\mathbf{X}}_i x_i(\underline{\lambda})) - \mathbb{R}_{i-1}(\underline{\lambda})\|_{\mathbb{A}_0^{-1}}$$

Sous forme faible, ce problème devient :

$$\int_{\mathbf{I}} \tilde{\mathbf{X}}_i x_i^*(\underline{\lambda}) \left(\mathbb{A}(\underline{\lambda}) (\tilde{\mathbf{X}}_i x_i(\underline{\lambda})) - \mathbb{R}_{i-1}(\underline{\lambda}) \right) d\mathbf{I} = 0 \quad \forall x_i^* \in \mathcal{F}(\mathbf{I}) \quad (8)$$

et se résoud classiquement par un point fixe par exemple.

On montre dans [14] que pour décrire la partie linéaire d'une fonction dépendant de multiples paramètres, il faut au moins 3 à 4 tenseurs d'ordre 1 pour l'approximer avec une précision suffisante. Ainsi, quand on se place dans une optique de minimisation du nombre de fonctions spatiales à calculer, on peut chercher à profiter pleinement de chaque mode.

On suppose qu'on a une solution sous la forme :

$$\underline{\mathbf{X}} = \sum_{j=1}^m \tilde{\mathbf{X}}_j \prod_{i=1}^n x_i^j$$

On peut ajouter un produit de fonctions paramétriques x_j^2 associé au mode $\tilde{\mathbf{X}}_j$ via la méthode de Galerkin ou la minimisation du résidu. On minimise alors la fonction :

$$\varepsilon = \|\mathbb{A}(\underline{\lambda}) (\tilde{\mathbf{X}}_j x_j^2(\underline{\lambda})) - \mathbb{R}_{i-1}^{(1)}(\underline{\lambda})\|_{\mathbb{A}_0^{-1}} \quad (9)$$

avec $\mathbb{R}_{i-1}^{(1)} = \mathbb{R}_{i-1} - \mathbb{A} \tilde{\mathbf{X}}_i x_i$

En généralisant cette étape à tous les modes, pour chaque mode spatial, on calcule S produits de fonctions paramétriques et on note cette opération d'enrichissement :

$$\underline{\mathbf{g}}_i^{(S)} = S|PGD(\tilde{\mathbf{X}}_i \mathbb{R}_{i-1}, \tilde{\mathbf{X}}_i^T \mathbb{A} \tilde{\mathbf{X}}_i)$$

Une fois ce premier mode calculé, on enrichit la solution en mettant à jours le résidu et en adoptant la même démarche. L'algorithme est présenté dans 2 dans le cas particulier où les bases de Krylov retenues sont de taille 1.

4.2 Exemple de problème et résultats

On choisit comme problème modèle tridimensionnel le cas d'un cube encastré sur une face et sollicité en traction. Un premier découpage en 27 sous-espaces est visible Figure 1. Chacun des sous-espaces tel que celui mis en relief sur cette Figure possède une rigidité indépendante et la solution particulière présentée correspond à un choix arbitraire de paramètres.

La Figure 2(a) compare la convergence de la méthode Extended PGD avec la méthode classique, qui est plus lente. C'est aussi le cas pour un exemple similaire à 64 paramètres visible Figure 2(b).

Il faut cependant remarquer que les taux d'erreur restent élevés. C'est ici la norme globale du résidu qui est tracée, ce qui signifie qu'en moyenne, l'erreur qui est effectuée pour chaque jeu de paramètres diminue. On peut pourtant rencontrer pour des jeux de paramètres particulièrement éloignés de la moyenne (donc de la solution $\underline{\mathbf{X}}_0$) des cas pour lesquels la solution fournie par le modèle réduit conduit à des erreurs qui restent élevées.

Algorithm 2 Extended PGD $K = 1$

- 1: Initialize $\tilde{\underline{X}}_0 = \mathbb{A}_0^{-1} \underline{F}$
 - 2: Normalize $\tilde{\underline{X}}_0$
 - 3: Set $\omega_0 = \underline{X}_0^T \underline{F}$ and $\kappa_0 = \tilde{\underline{X}}_0^T \mathbb{A} \tilde{\underline{X}}_0$
 - 4: Compute $\underline{g}_0^{(S)} = S |PGD(\omega_0, \kappa_0) = \sum_{s=1}^S \prod_{i=1}^n x_{i,0,s}$ ▷ S : number of fonctions per spatial mode
 - 5: Set $\underline{X}_0 = \tilde{\underline{X}}_0 \sum_{s=1}^S \prod_{i=1}^n x_{i,0,s}$
 - 6: Set $\underline{X} = \underline{X}_0$
 - 7: Compute $\underline{R} = \underline{b} - \mathbb{A} \underline{X}$
 - 8: **for** $m = 1$ to m_{max} **do** ▷ Number of modes
 - 9: Compute $\tilde{\underline{X}}_m = 1 |PGD(\underline{R}, \mathbb{H}^{-1})$
 - 10: Orthogonalize $\tilde{\underline{X}}_m, \tilde{\underline{X}}_{m-1}$ and $\tilde{\underline{X}}_{m-2}$
 - 11: Compute $\underline{g}_m^{(S)} = \sum_{s=1}^S \prod_{i=1}^n x_{i,m,s}$
 - 12: Set $\underline{X}_m = \tilde{\underline{X}}_m \sum_{s=1}^S \prod_{i=1}^n x_{i,m,s}$
 - 13: Set $\underline{X} = \underline{X} + \underline{X}_m$
 - 14: Compute $\underline{R} = \underline{F} - \mathbb{A} \underline{X}$
 - 15: Check convergence
 - 16: **end for**
-

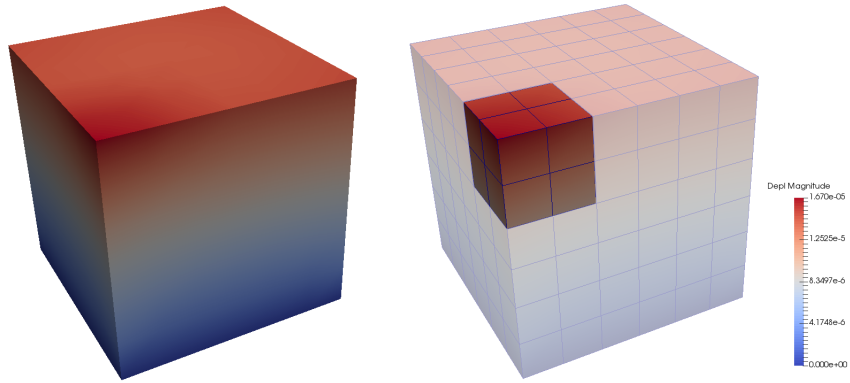


FIGURE 1 – Solution particulière pour un problème à 27 paramètres : déplacement dans un cube en traction. A droite, maillage utilisé et mise en relief d'un des 27 sous-domaines

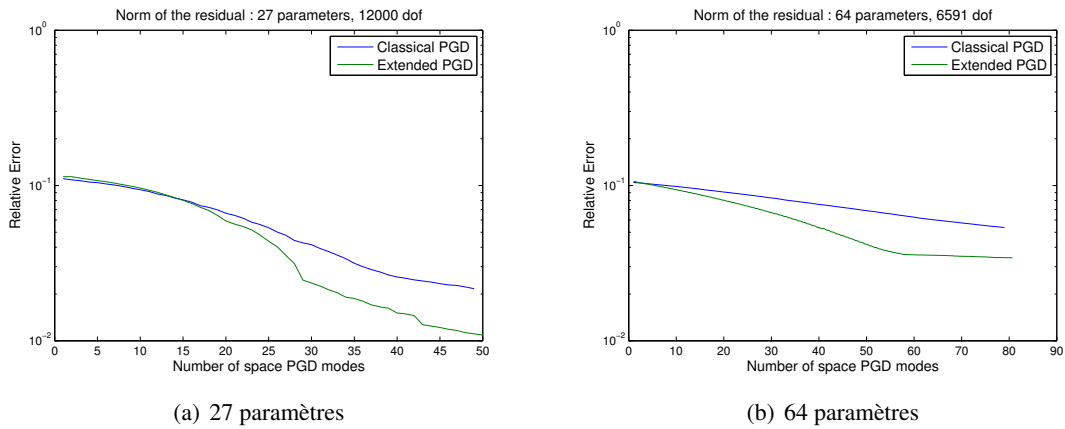


FIGURE 2 – Comparaison entre la méthode PGD classique et une méthode Extended PGD

5 PGD multiéchelle en paramètres

5.1 Principe

La PGD étendue permet bien d'accélérer la convergence pour des problèmes à grand nombre de paramètres. Pour la prolonger, on peut analyser plus précisément la relation (3). On se place de le cas étudié dans les exemples de la forme (1), des problèmes à nombreux paramètres matériaux, constants par sous-domaines. Ces paramètres peuvent être vus comme des champs de l'espace étudié. Pour mettre en place cette vision, une discréetisation spatiale complètement discontinues a été retenue. La Weak-Trefftz Discontinuous Method introduite dans [15] et appliquée dans [16] pour le calcul des phénomènes « moyennes fréquences ». La construction de l'opérateur \mathbb{A} se fait par assemblage et on sépare dans la formulation discontinue les influences (ici linéaires) des différents paramètres. Ainsi, le problème (2) peut se réécrire localement :

$$\forall E \in \mathbf{E} \quad \sum_{E' \in \mathbf{E}} [\mathbf{A}_0^{EE'} + \lambda_E \overline{\mathbf{A}}^{EE'} + \lambda_{E'} \overline{\mathbf{A}}'^{EE'}] \underline{X}_{E'} = F_E \quad \underline{X}_{E'} \in \mathbb{R}^{n_E} \times \mathbf{I}$$

Reprenons alors la formulation (3) : à l'ordre 0, cette approximation conduit à un résidu de la forme $\underline{R}_0 = \underline{F} - \mathbb{A}\underline{X}_0$. La solution approchée à l'ordre 1 est $\underline{X} \approx \underline{X}_0 + \underline{X}_1$ avec

$$\underline{X}_1 = \mathbb{D}\underline{X}_0 = \mathbb{A}_0^{-1}\underline{R}_0$$

En développant cette expression, on obtient :

$$\underline{X}_1 = \mathbb{A}_0^{-1} \left[\sum_{E \in \mathbf{E}} \sum_{E' \in \mathbf{E}} -\lambda_E \overline{\mathbf{A}}^{EE'} - \lambda_{E'} \overline{\mathbf{A}}'^{EE'} \right] \underline{X}_0$$

Or, en pratique chaque paramètre λ_E est local et n'a d'influence que sur le domaine C_E qui comprend l'élément E et son environnement immédiat. Donc :

$$\underline{X}_1 = \mathbb{A}_0^{-1} \left[\sum_{E \in \mathbf{E}} \left[\sum_{E' \in C_E} -\lambda_E \overline{\mathbf{A}}^{EE'} - \lambda_{E'} \overline{\mathbf{A}}'^{EE'} \right] \right] \underline{X}_0 = \mathbb{A}_0^{-1} \left[\sum_{E \in \mathbf{E}} \lambda_E \left[\sum_{E' \in C_E} \tilde{\underline{Z}}_E^{E'} \right] \right]$$

On peut interpréter $\tilde{\underline{Z}}_E = \sum_{E' \in C_E} \tilde{\underline{Z}}_E^{E'}$ comme une contrainte locale. D'après le Principe de Saint-Venant,

$\mathbb{A}_0^{-1}\tilde{\underline{Z}}_E$ sera aussi localisée sur un voisinage de E , certes plus étendu que C_E .

De même, l'approximation d'ordre 2 donne le terme :

$$\underline{X}_2 = \mathbb{A}_0^{-1} \left[\sum_{E \in \mathbf{E}} \left[\sum_{E' \in C_E} -\lambda_E \overline{\mathbf{A}}^{EE'} - \lambda_{E'} \overline{\mathbf{A}}'^{EE'} \right] \right] \sum_{E'' \in \mathbf{E}} \tilde{\underline{Z}}_{E''}$$

Le résidu n'est non-linéaire en λ_E que sur un voisinage de E et donc, d'après le principe de Saint-Venant \underline{X}_2 aussi. Ailleurs, \underline{X}_2 est linéaire en λ_E . En conservant le second ordre de l'approximation (3), on a donc remarqué que :

- la solution est linéaire par rapport aux λ_E , excepté dans l'environnement immédiat C_E de E
- au premier ordre, la zone d'influence de λ_E est limitée à C_E

Ces remarques ont été développées dans [14] et aboutissent à la PGD multiéchelle en paramètres. L'idée de base consiste à introduire une description à deux échelles de chaque paramètre : l'échelle « macro », sur tout le domaine sur lequel les paramètres ont une influence linéaire ; l'échelle « micro », sur le voisinage des éléments associés aux paramètres E sur lequel les fonctions paramétriques seront discrétisées plus finement.

5.2 Méthode

Une proposition de mise en application se trouve résumée dans l'Algorithme 3. En particulier, des méthodes spécifiques sont mises en place pour ne jamais avoir à assembler l'opérateur \mathbb{A} , qui dépend d'un grand nombre de paramètres très localisés.

Initialisation Comme pour la méthode précédente, on initialise la solution par \underline{X}_0 .

Base spatiale On choisit de construire l'approximation PGD du résidu localement. Ainsi, pour chaque élément on calcule $\tilde{\underline{Z}}_E = 1|PGD(\underline{R}_E, \mathbb{A}_0^E)$. On retrouve ensuite un unique vecteur spatial en combinant linéairement les vecteurs locaux obtenus. La combinaison optimale s'obtient en minimisant :

$$\sum_{E \in \mathbf{E}} \|\|(\underline{R}_E - \gamma(\underline{\lambda}) a_E \mathbb{A}_0^E \tilde{\underline{Z}}_E)\|\|_{\mathbb{A}_0^{E-1}}$$

On résoudra ce problème d'approximation PGD par substitution, ce qui permet d'obtenir un problème aux valeurs propres dont les a_E sont solutions sans avoir à calculer les fonctions paramétriques γ .

Fonctions « Macro » Une fois un vecteur spatial calculé, on calcule les fonctions « Macro » en suivant la technique PGD classique vue en (8), qui nous permet de déterminer les fonctions globales $f_E(\underline{\lambda}_E)$. Comme on l'a montré précédemment, on peut se contenter de calculer des fonctions linéaires, donc très légères à manipuler.

Ensuite, on calcule localement la norme du nouveau résidu : $\varepsilon_E = \|\| \underline{R}_E - \tilde{\underline{X}}_m^E f \|\|_{\mathbb{A}_0^{E-1}}$. On sélectionne les éléments pour lesquels cette norme est supérieure à un certain seuil, par exemple supérieure à $\alpha \varepsilon_E^{max}$ avec $\alpha \sim 0.5 - 0.8$.

Fonctions « micro » Pour ces éléments, on va affiner notre solution par une fonction « micro ». On réécrit notre problème sous la forme faible (8), en modifiant cependant l'expression de la fonction paramétrique \underline{x} . Cette expression va maintenant varier localement : pour les éléments E' du voisinage C_E de E , on n'utilisera plus les expressions globales $f_{E'}(\underline{\lambda}_{E'})$ mais des fonctions spécifiques et discrétisées plus finement : $g_{EE'}(\underline{\lambda}_{E'})$. Ces solutions se calculent via un point fixe sur les paramètres des éléments de C_E .

Ainsi, les solutions se présentent sous la forme :

$$\underline{X}_m = \underline{X}_0 + \sum_{n=1}^m \sum_{E \in \mathbf{E}} \tilde{\underline{X}}_E^n \prod_{E' \notin C_E} f_{E'}^{(n)}(\underline{\lambda}_{E'}) \prod_{E' \in C_E} g_{EE'}^{(n)}(\underline{\lambda}_{E'})$$

Algorithm 3 Parametric multiscale PGD

```

1: Initialize  $\underline{X} = \mathbb{A}_0^{-1} \underline{F}$ 
2: Set  $\underline{X} = \underline{X}_0$ 
3: for  $m = 1$  to  $m_{max}$  do ▷ Number of modes
4:   for  $E \in \mathbf{E}$  do ▷ Number of elements
5:     Set  $\underline{R}_E = \underline{F} - [\mathbb{A}\underline{X}]_E$ 
6:     Compute  $\tilde{\underline{Z}}_E = 1|PGD(\underline{R}_E, \mathbb{A}_0^E)$  ▷ local PGD spatial approximation
7:   end for
8:   Compute  $\tilde{\underline{X}}_m = \sum_{E \in \mathbf{E}} a_E \tilde{\underline{Z}}_E$  ▷ global PGD spatial approximation
9:   Orthogonalize  $\tilde{\underline{X}}_m, \tilde{\underline{X}}_{m-1}$  and  $\tilde{\underline{X}}_{m-2}$ 
10:  Compute  $\underline{f}(\underline{\lambda})$  ▷ Macro functions
11:   $\forall E \in \mathbf{E}$ , compute  $\varepsilon_E = \|\| \underline{R}_E - \tilde{\underline{X}}_m^E \underline{f}(\underline{\lambda}) \|\|$ 
12:  for  $E \in \mathbf{E}$  do
13:     $m_E = \{\}$ 
14:    if  $\varepsilon_E > \alpha \varepsilon_E^{max}$  then
15:       $m_E = \{E' \in C_E\}$ 
16:       $\forall E' \in C_E$  compute  $g_m^{EE'}$  ▷ Micro functions
17:    end if
18:  end for
19:  Compute  $\underline{X} = \underline{X} + \sum_{E \in \mathbf{E}} \tilde{\underline{X}}_m^E \prod_{E' \notin m_E} f_{E'}(\underline{\lambda}_{E'}) \prod_{E' \in m_E} g^{EE'}(\underline{\lambda}_{E'})$ 
20:  Check convergence
21: end for

```

6 Conclusion

Les deux méthodes introduites ici, la PGD étendue et son développement, la PGD multiéchelle en paramètres, sont de nouvelles démarches adaptées aux problèmes à grand nombre de paramètres. La première vision permet de minimiser le nombre d'opérations spatiales et donc de réduire les coûts de calcul pour les problèmes de grande taille. La deuxième approche permet de réaliser l'intégralité des opérations à un niveau élémentaire et est particulièrement adaptée aux traitements de problèmes pour lesquels les paramètres sont définis localement, à l'échelle des éléments de la discrétisation.

Références

- [1] Francisco Chinesta, Antonio Huerta, Gianluigi Rozza, and Karen Willcox. Model order reduction : a survey. Wiley, 2016.
- [2] Wolfgang Hackbusch. *Tensor spaces and numerical tensor calculus*, volume 42. Springer Science & Business Media, 2012.
- [3] Mike Espig, Wolfgang Hackbusch, Alexander Litvinenko, Hermann G Matthies, and Philipp Wähnert. Efficient low-rank approximation of the stochastic galerkin matrix in tensor formats. *Computers & Mathematics with Applications*, 67(4) :818–829, 2014.
- [4] Thuan Lieu, Charbel Farhat, and Michel Lesoinne. Reduced-order fluid/structure modeling of a complete aircraft configuration. *Computer methods in applied mechanics and engineering*, 195(41) :5730–5742, 2006.
- [5] Max D Gunzburger, Janet S Peterson, and John N Shadid. Reduced-order modeling of time-dependent pdes with multiple parameters in the boundary data. *Computer Methods in Applied Mechanics and Engineering*, 196(4) :1030–1047, 2007.
- [6] Anthony T Patera and Gianluigi Rozza. Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations, 2007.
- [7] Pierre Ladevèze. The large time increment method for the analysis of structures with non-linear behavior described by internal variables. *Comptes rendus de l'académie des sciences, Serie II*, 309(11) :1095–1099, 1989.
- [8] Philippe Boisse, P Bussy, and Pierre Ladeveze. A new approach in non-linear mechanics : The large time increment method. *International journal for numerical methods in engineering*, 29(3) :647–663, 1990.
- [9] David Néron, Pierre-Alain Boucard, and Nicolas Relun. Time-space pgd for the rapid solution of 3d non-linear parametrized problems in the many-query context. *International Journal for Numerical Methods in Engineering*, 103(4) :275–292, 2015.
- [10] Tan Bui-Thanh, Karen Willcox, and Omar Ghattas. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing*, 30(6) :3270–3288, 2008.
- [11] Anthony Nouy. A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 199(23) :1603–1626, 2010.
- [12] Francisco Chinesta, Roland Keunings, and Adrien Leygue. *The Proper Generalized Decomposition for Advanced Numerical Simulations : A Primer*, chapter Parametric Models, pages 71–88. Springer International Publishing, Cham, 2014.
- [13] Pierre Ladevèze. De nouvelles méthodes avec conditionneur pour le calcul de pgd-modèles. Technical report, LMT, 2014.
- [14] Pierre Ladevèze. Une nouvelle méthode pour le calcul de modèles réduits : la pgd multiechelle en paramètre. Technical report, LMT, en cours de publication [2016].
- [15] Pierre Ladevèze. De nouvelles formulations variationnelles pour approximations discontinues. Technical report, LMT, 2011.
- [16] Pierre Ladevèze and Hervé Riou. On trefftz and weak trefftz discontinuous galerkin approaches for medium-frequency acoustics. *Computer Methods in Applied Mechanics and Engineering*, 278 :729 – 743, 2014.