



HAL
open science

La méthode des points de référence, une technique d'hyper-réduction: Application à la réduction de modèle non linéaire par PGD

Pierre-Alain Guidault, Matteo Capaldo, David Néron, Pierre Ladevèze

► To cite this version:

Pierre-Alain Guidault, Matteo Capaldo, David Néron, Pierre Ladevèze. La méthode des points de référence, une technique d'hyper-réduction: Application à la réduction de modèle non linéaire par PGD. 13e colloque national en calcul des structures, Université Paris-Saclay, May 2017, Giens, Var, France. hal-01899339

HAL Id: hal-01899339

<https://hal.science/hal-01899339>

Submitted on 19 Oct 2018

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

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

La méthode des points de référence, une technique d'hyper-réduction: Application à la réduction de modèle non linéaire par PGD

P.-A. Guidault¹, M. Capaldo², D. Néron¹, P. Ladevèze¹

¹ LMT, ENS Cachan, CNRS, Université Paris-Saclay, {pierre-alain.guidault, david.neron, pierre.ladeveze}@ens-paris-saclay.fr

² EDF – R&D, THEMIS/R22, matteo.capaldo@edf.fr

Résumé — Une technique d'approximation, appelée méthode des points de référence (RPM), est proposée afin de réduire la complexité des opérations algébriques nécessaires à la construction de modèles réduits dans le cadre de la *Proper Generalized Decomposition* pour des problèmes non-linéaires en temps ou paramétrés. Basée sur le concept de points de référence et de *patches* d'influence, la RPM introduit une version compressée des données à partir de laquelle une représentation par patch en variables séparées de rang faible des opérateurs est construite par des formules explicites à faible coût de calcul.

Mots clés — réduction de modèle non linéaire, PGD, base réduite, RPM, hyper-réduction.

1 Introduction

La complexité croissante des modèles dans le domaine du calcul en sciences de l'ingénieur conduit à des modèles numériques dont le coût de calcul peut être prohibitif pour les stratégies de calcul classiques. En alternative, les techniques de réduction de modèle cherchent la solution du problème considéré dans une base réduite (ROB), dont la dimension est bien plus faible que la taille du problème initial. Dans une phase d'apprentissage, les techniques basées sur la *Proper Orthogonal Decomposition* (POD) construisent une ROB en considérant les modes POD les plus pertinents à partir de solutions obtenues pour des instants/paramètres particuliers (*snapshots*). La technique de base réduite (*Reduced Basis*) quant à elle sélectionne ces *snapshots* par un algorithme glouton. Ainsi, elle assure la qualité de la ROB pour la construction du modèle réduit et permet de pallier la dépendance au cas de la POD.

Une autre famille de techniques de réduction de modèle est celle basée sur la *Proper Generalized Decomposition* (PGD). Cette technique ne nécessite pas de *snapshots* générés pendant une phase d'apprentissage pour construire la ROB. Celle-ci est générée automatiquement, à la volée, par un algorithme glouton [3]. En calcul des structures, la PGD a été utilisée sous le nom d'approximation radiale dans la méthode LATIN (Large Time Increment) [6], une stratégie de résolution non-incrémentale de problèmes d'évolution non-linéaires. Une des améliorations significatives a été l'introduction d'une *étape préliminaire* consistant, à chaque itération LATIN, à mettre à jour les fonctions du temps de la représentation séparée espace-temps de l'itéré à partir de la ROB générée jusqu'à l'itération précédente [7]. Appelée *étape de mise-à-jour* dans [8], cette étape a montré son efficacité pour réduire le nombre de fonctions PGD générées et conserver les plus pertinentes. Cependant, cette étape peut être coûteuse en temps CPU.

Les techniques de réduction de modèle sont particulièrement efficaces lorsque le modèle réduit n'a besoin d'être construit qu'une seule fois ou au préalable, *offline*, avant la résolution *online* du problème qui peut alors être très rapide. En revanche, pour les problèmes non-linéaires, le modèle réduit est coûteux à assembler rendant ainsi l'intérêt de la réduction de modèle moindre. En effet, la complexité des opérations algébriques nécessaires à l'évaluation des termes non-linéaires (Jacobien et résidu) et l'assemblage du modèle réduit est de l'ordre de la dimension du problème initial. Ce point peut être critique en particulier lorsque la base réduite évolue au cours des itérations de la méthode de résolution non-linéaire.

Plusieurs techniques ont été introduites pour répondre à ce problème, en particulier dans le domaine des techniques de réduction de modèle s'appuyant sur une étape d'apprentissage *offline*. Parmi elles, l'*Empirical Interpolation Method* (EIM) est probablement l'une des plus populaires [1, 4]. Plusieurs variantes existent telles que la *Best Interpolation Point Method* (BPIM) ou la *Discrete EIM* (DEIM). Une autre famille de méthodes est celle issue de l'application de la *Gappy-POD*, comme l'*A priori Hyper-Reduction* (APHR) ou la *Missing Point Estimation* (MPE) entre autres.

Les techniques basées sur la PGD pour pallier cette difficulté sont de loin moins nombreuses. Le fait que la base réduite évolue fréquemment rend les techniques précédentes a priori moins adaptées. Basée sur le concept de points, d’instantants, de paramètres de référence, la RPM [2] construit une version *compressée* des données à partir de laquelle une représentation par patch en variables séparées de rang faible des opérateurs est obtenue par des formules explicites pour un coût de calcul très faible. Il ne s’agit pas d’une technique d’interpolation mais d’une méthode d’approximation des intégrales impliquées dans la construction du modèle réduit. La RPM est ici utilisée, à chaque itération de la méthode LATIN, uniquement pour approcher la résolution de l’étape préliminaire qui n’est rien d’autre qu’une mise-à-jour du modèle réduit à partir de la base réduite construite au cours des itérations précédentes. Aussi, si l’approximation n’est pas de bonne qualité, la génération de nouveaux couples PGD permet de corriger l’approximation, la convergence étant toujours assurée.

2 Réduction de modèle PGD d’une EDP elliptique non-linéaire

2.1 Formulation du problème de référence

On s’intéresse à une EDP elliptique non-linéaire, paramétrée, définie sur un domaine spatial $\Omega \subset \mathbb{R}^2$ et un espace des paramètres $\mathcal{D} \subset \mathbb{R}^2$ définie comme suit [4] :

Problème 1 (Formulation forte) Trouver $u(\mathbf{x}, \boldsymbol{\mu})$ avec $\mathbf{x} = (x_1, x_2) \in \Omega =]0, 1[^2$ et $\boldsymbol{\mu} = (\mu_1, \mu_2) \in \mathcal{D} = [0.01, 10]^2$ et des conditions de Dirichlet homogènes sur $\partial\Omega \times \mathcal{D}$, tel que :

$$\begin{cases} -\nabla^2 u + g(u; \boldsymbol{\mu}) &= f(\mathbf{x}) & \text{on } \Omega \times \mathcal{D} \\ u &= 0 & \text{on } \partial\Omega \times \mathcal{D} \end{cases}$$

avec $g(u; \boldsymbol{\mu}) = \frac{\mu_1}{\mu_2} (e^{\mu_2 u} - 1)$ et $f(\mathbf{x}) = 100 \sin(2\pi x_1) \sin(2\pi x_2)$.

Une formulation faible en espace du **Problème 1** peut être proposée. Dans ce cas, la solution u est identifiée à une fonction définie sur \mathcal{D} à valeurs dans l’espace de Hilbert $\mathcal{V} = H_0^1(\Omega) = \{u \in H^1(\Omega); u = 0 \text{ sur } \partial\Omega\}$, i.e. $u(\boldsymbol{\mu}) : \mathbf{x} \mapsto u(\boldsymbol{\mu})(\mathbf{x}) \approx u(\mathbf{x}, \boldsymbol{\mu})$, et le **Problème 1** peut s’écrire sous la forme :

Problème 2 (Formulation faible en espace) Pour $\boldsymbol{\mu} \in \mathcal{D}$, trouver $u : \mathcal{D} \rightarrow \mathcal{V}$ tel que :

$$\forall v \in \mathcal{V}, \quad a(u, v) + \int_{\Omega} g(u; \boldsymbol{\mu}) v \, dx = l(v)$$

où $a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v \, dx$ (resp. $l(v) = \int_{\Omega} f(\mathbf{x}) v \, dx$) est une forme bilinéaire (resp. linéaire) sur \mathcal{V} .

Le terme $g(u; \boldsymbol{\mu})$ est une fonction non-linéaire par rapport au paramètre $\boldsymbol{\mu}$ et au champ solution $u(\mathbf{x}, \boldsymbol{\mu})$. Les formes bilinéaire $a(u, v)$ et linéaire $l(v)$ sont bornées et continues sur \mathcal{V} et indépendante du paramètre. En introduisant l’espace fonctionnel $\mathcal{S} = L^2(\mathcal{D}, \mathcal{V}) = \{u : \mathcal{D} \rightarrow \mathcal{V}, \int_{\mathcal{D}} \|u\|_{\mathcal{V}}^2 \, d\boldsymbol{\mu} < \infty\}$ où $\|\cdot\|_{\mathcal{V}}$ est une norme sur \mathcal{V} , une formulation faible sur l’espace des paramètres et le domaine spatial du **Problème 2** peut également être définie :

Problème 3 (Formulation faible espace-paramètre) Trouver $u \in \mathcal{S}$ tel que :

$$\forall v \in \mathcal{S}, \quad A(u, v) + G(u, v) = L(v)$$

où $A(u, v) = \int_{\mathcal{D}} a(u(\boldsymbol{\mu}), v(\boldsymbol{\mu})) \, d\boldsymbol{\mu}$, $L(v) = \int_{\mathcal{D}} l(v(\boldsymbol{\mu})) \, d\boldsymbol{\mu}$ et $G(u, v) = \int_{\mathcal{D} \times \Omega} g(u; \boldsymbol{\mu}) v \, dx \, d\boldsymbol{\mu}$.

Le **Problème 2** peut être classiquement résolu en introduisant un espace d’approximation éléments finis $\mathcal{V}_h \in \mathcal{V}$ et en utilisant une méthode de résolution non-linéaire de type Newton (**Figure 1**).

2.2 Résolution non-linéaire par un schéma à directions alternées : la méthode LATIN

Par un schéma itératif à deux étapes et deux directions de recherche, la méthode LATIN génère des approximations de la solution qui appartiennent alternativement à deux variétés, \mathbf{A}_d and Γ . La variété \mathbf{A}_d est définie par les équations linéaires, éventuellement globales, du problème tandis que la variété Γ contient les équations locales, éventuellement non-linéaires. De façon similaire aux techniques de lagrangien augmenté, le point clé de la stratégie consiste à introduire l’espace $\Gamma = \{(u, w) \in \mathcal{V} \times \mathcal{W}; w = g(u; \boldsymbol{\mu}), \forall \boldsymbol{\mu} \in \mathcal{D}\}$ avec $\mathcal{W} = L^2(\Omega)$. La solution est cherchée en résolvant alternativement une *étape locale* et une *étape globale* comme décrit ci-après.

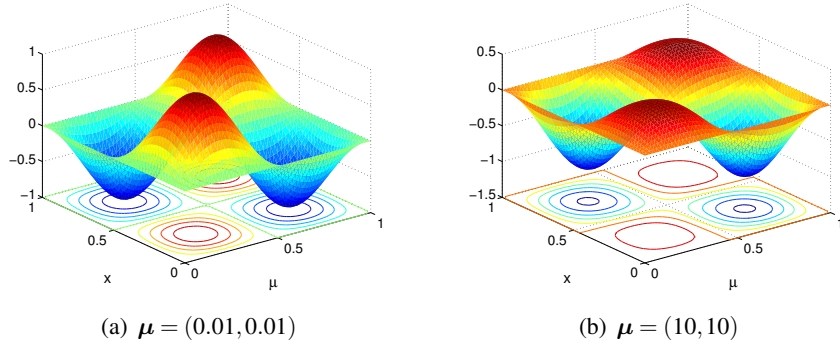


FIGURE 1 – Solution du **Problème 2** pour différentes valeurs du paramètre et un maillage éléments finis de 50×50 éléments quadrangulaires bilinéaires (espace d'approximation EF de dimension $N = 2601$)

2.2.1 Etape locale à l'itération $n + 1$

Elle consiste à trouver l'itéré $s^{(n+1/2)} = (\hat{u}^{(n+1/2)}, \hat{w}^{(n+1/2)}) \in \Gamma$, i.e. qui vérifie l'équation non-linéaire $\hat{w}^{(n+1/2)} = g(\hat{u}^{(n+1/2)}; \mu)$ pour tout μ , connaissant $s^{(n)} = (u^{(n)}, w^{(n)}) \in \mathbf{A}_d$ obtenu à l'étape globale précédente, et satisfaisant la direction de recherche \mathbf{E}^+ :

$$\mathbf{E}^+ : \forall \mu \in \mathcal{D}, \forall \mathbf{x} \in \Omega, \quad (\hat{w}^{(n+1/2)} - w^{(n)}) + \mathbf{H}^+(\hat{u}^{(n+1/2)} - u^{(n)}) = 0 \quad (1)$$

où \mathbf{H}^+ est un opérateur symétrique défini positif, paramètre de la méthode. Ce problème est non-linéaire mais local en variable d'espace. Un choix simple est de prendre $\mathbf{H}^+ \rightarrow \infty$, ainsi :

$$\forall \mu \in \mathcal{D}, \forall \mathbf{x} \in \Omega, \quad \begin{cases} \hat{u}^{(n+1/2)} &= u^{(n)} \\ \hat{w}^{(n+1/2)} &= g(\hat{u}^{(n+1/2)}; \mu) = g(u^{(n)}; \mu) \end{cases} \quad (2)$$

2.2.2 Etape globale à l'itération $n + 1$

Un itéré $s^{(n+1)} = (u^{(n+1)}, w^{(n+1)}) \in \mathbf{A}_d$ est déterminé, connaissant $s^{(n+1/2)} = (\hat{u}^{(n+1/2)}, \hat{w}^{(n+1/2)}) \in \Gamma$ connu à partir de l'étape locale précédente, et vérifiant la direction de recherche \mathbf{E}^- :

$$\mathbf{E}^- : \forall \mu \in \mathcal{D}, \forall \mathbf{x} \in \Omega, \quad (w^{(n+1)} - \hat{w}^{(n+1/2)}) - \mathbf{H}^-(u^{(n+1)} - \hat{u}^{(n+1/2)}) = 0 \quad (3)$$

Comme la direction de recherche \mathbf{H}^+ , \mathbf{H}^- est un paramètre de la méthode. Il est montré dans [6] qu'il est judicieux de choisir l'opérateur tangent :

$$\mathbf{H}^- = \left. \frac{\partial g(u; \mu)}{\partial u} \right|_{u=\hat{u}^{(n+1/2)}} = \left. \frac{\partial g(u; \mu)}{\partial u} \right|_{u=u^{(n)}} \quad (4)$$

La seconde égalité est due au choix de direction de recherche \mathbf{E}^- (2). Un opérateur constant pourrait être utilisé. Cependant, le choix d'un opérateur tangent améliore drastiquement le taux de convergence. Le problème de l'étape global s'énonce comme suit :

Problème 4 (Étape globale) Connaissant $s^{(n+1/2)} = (\hat{u}^{(n+1/2)}, \hat{w}^{(n+1/2)})$, trouver $s^{(n+1)} = (u^{(n+1)}, w^{(n+1)}) \in \mathbf{A}_d$ vérifiant la direction de recherche (3) et tel que :

$$\forall v \in \mathcal{S}, \quad \int_{\mathcal{D}} a(u^{(n+1)}, v) d\mu + \int_{\mathcal{D} \times \Omega} w^{(n+1)} v dx d\mu = \int_{\mathcal{D} \times \Omega} f(\mathbf{x}) v dx d\mu \quad (5)$$

En introduisant la direction de recherche (3) dans (5) et dû à (2) et (4), le problème en correction de l'itéré $\delta u^{(n+1)} = u^{(n+1)} - u^{(n)}$ suivant est obtenu :

Problème 5 (Étape globale sur la correction) Trouver $\delta u^{(n+1)} = u^{(n+1)} - u^{(n)} \in \mathcal{S}$ et $w^{(n+1)} \in \mathcal{S}$ tel que :

$$\forall v \in \mathcal{S}, \quad \int_{\mathcal{D}} a(\delta u^{(n+1)}, v) d\mu + \int_{\mathcal{D} \times \Omega} \mathbf{H}^-(u^{(n)}; \mu) \delta u^{(n+1)} v dx d\mu = - \int_{\mathcal{D}} \mathcal{R}(u^{(n)}, v; \mu) d\mu$$

$$\forall \mu \in \mathcal{D}, \forall \mathbf{x} \in \Omega, \quad w^{(n+1)} = \hat{w}^{(n+1/2)} + \mathbf{H}^- \delta u^{(n+1)} \quad (6)$$

$$\text{où } \mathcal{R}(u^{(n)}, v; \boldsymbol{\mu}) = a(u^{(n)}, v) - \int_{\Omega} f(\boldsymbol{x}) v \, dx + \int_{\Omega} g(u^{(n)}; \boldsymbol{\mu}) v \, dx$$

A chaque itération, étape locale et étape globale sont résolues jusqu'à convergence. L'indicateur de convergence suivant est utilisé :

$$\delta_L^{(n+1)} = \frac{\|u^{(n+1)} - \hat{u}^{(n+1/2)}\|}{\frac{1}{2} \|u^{(n+1)} + \hat{u}^{(n+1/2)}\|} \quad \text{avec} \quad \|\cdot\| = \int_{\mathcal{D} \times \Omega} \cdot^2 \, dx \, d\boldsymbol{\mu}$$

Sous l'hypothèse d'une fonction monotone $g(u; \boldsymbol{\mu})$, le schéma itératif converge vers la solution pour tout choix d'opérateur symétrique défini positif de \mathbf{H}^- [6].

Contrairement aux méthodes de résolution incrémentales, la correction de l'itéré $\delta u^{(n+1)} \in \mathcal{S} = L^2(\mathcal{D}, \mathcal{V})$ doit être stockée sur tout le domaine spatiale et tout l'espace des paramètres. Afin de pallier ce problème de stockage de données, une PGD de l'itéré est classiquement utilisée dans la méthode LATIN [6].

2.3 Réduction de modèle par PGD

En notant $\mathcal{P} = L^2(\mathcal{D}; \mathbb{R}^2) := L^2(\mathcal{D})$ et en identifiant l'espace $\mathcal{S} = L^2(\mathcal{D}, \mathcal{V})$ avec l'espace tensoriel $\mathcal{V} \otimes \mathcal{P}$, la correction $\delta u^{(n+1)}$ de l'étape globale (**Problème 5**) est cherchée dans $\mathcal{V} \otimes \mathcal{P}$. L'approximation d'ordre k suivante de l'itéré $u^{(n+1)}$ est proposée :

$$u^{(n+1)}(\boldsymbol{x}, \boldsymbol{\mu}) \approx u_k^{(n+1)}(\boldsymbol{x}, \boldsymbol{\mu}) = u^{(0)}(\boldsymbol{x}, \boldsymbol{\mu}) + \sum_{i=1}^k \Phi_i(\boldsymbol{x}) \lambda_i(\boldsymbol{\mu}) \quad (7)$$

où $u^{(0)} \in \mathbf{A}_d$ est une estimation initiale admissible de la solution. Chaque couple PGD $(\lambda_i, \Phi_i) \in \mathcal{P} \times \mathcal{V}$ est déterminé par un algorithme glouton. La construction d'une fonction spatiale Φ_i est l'étape la plus coûteuse du processus. Ainsi, à l'itération $n+1$, il est judicieux d'utiliser d'abord la base réduite $\mathcal{W}_k = \{\Phi_i\}_{1 \leq i \leq k}$ générée à l'itération n afin de mettre à jour les fonctions du paramètre $\{\lambda_i\}_{1 \leq i \leq k}$ [7]. On procède avec l'étape globale à l'itération $n+1$ comme suit :

1. *Etape préliminaire* : une approximation de la solution, notée $\check{u}^{(n+1)} = (\check{u}^{(n+1)}, \check{w}^{(n+1)})$, est construite à partir de la ROB obtenue jusqu'à l'itération n . En tirant parti de la linéarité du **Problème 4**, une approximation d'ordre k de la correction $\delta u^{(n+1)}$ est cherchée telle que :

$$\check{u}^{(n+1)} = u^{(n)} + \delta u^{(n+1)} \quad \text{avec} \quad \delta u^{(n+1)}(\boldsymbol{x}, \boldsymbol{\mu}) \approx \delta u_k^{(n+1)}(\boldsymbol{x}, \boldsymbol{\mu}) \approx \sum_{i=1}^k \Phi_i(\boldsymbol{x}) \delta \lambda_i(\boldsymbol{\mu}) \quad (8)$$

En supposant qu'une approximation d'ordre k de $u^{(n)}$ est obtenue à l'itération n précédente, une approximation de $\check{u}^{(n+1)}$ peut être déterminée ainsi :

$$\check{u}^{(n+1)} \approx u^{(n)} + \delta u_k^{(n+1)} = u^{(0)}(\boldsymbol{x}, \boldsymbol{\mu}) + \sum_{i=1}^k \Phi_i(\boldsymbol{x}) (\lambda_i(\boldsymbol{\mu}) + \delta \lambda_i(\boldsymbol{\mu})) \quad (9)$$

Le problème à résoudre à l'étape préliminaire s'écrit :

Problème 6 (Etape préliminaire) Trouver les corrections $\{\delta \lambda_i\}_{1 \leq i \leq k}$ telles que :

$$1 \leq j \leq k, \forall \lambda^* \in \mathcal{P},$$

$$\sum_{i=1}^k \int_{\mathcal{D}} \lambda^* \left(a(\Phi_i, \Phi_j) + \int_{\Omega} \Phi_i \mathbf{H}^-(u^{(n)}; \boldsymbol{\mu}) \Phi_j \, dx \right) \delta \lambda_i \, d\boldsymbol{\mu} = - \int_{\mathcal{D}} \lambda^* \mathcal{R}(u^{(n)}, \Phi_j; \boldsymbol{\mu}) \, d\boldsymbol{\mu}$$

Les termes $a(\Phi_i, \Phi_j)$ sont indépendants du paramètre et peuvent être précalculés si la base réduite reste inchangée. En revanche, les termes non-linéaires $\Phi_i \mathbf{H}^-(u^{(n)}; \boldsymbol{\mu}) \Phi_j$ et le résidu $\mathcal{R}(u^{(n)}, \Phi_j; \boldsymbol{\mu})$ dépendent de $g(u^{(n)}; \boldsymbol{\mu})$ et doivent être évalués pour chaque paramètre et chaque itéré $u^{(n)}$ pour un coût de calcul de l'ordre de grandeur de la taille du problème initial. L'indicateur de performance suivant est ensuite utilisé :

$$\eta_0 = \frac{e_1 - e_2}{e_1} \geq 0 \quad \text{avec} \quad e_1 = \frac{\|u^{(n)} - \hat{u}^{(n-1/2)}\|}{\frac{1}{2} \|u^{(n)} + \hat{u}^{(n-1/2)}\|} \quad \text{et} \quad e_2 = \frac{\|\check{u}^{(n+1)} - \hat{u}^{(n+1/2)}\|}{\frac{1}{2} \|\check{u}^{(n+1)} + \hat{u}^{(n+1/2)}\|} \quad (10)$$

Si la valeur de η_0 est suffisamment élevée, la correction $\delta u_k^{(n+1)}$ est significative et l'étape globale de l'itération $n + 1$ est considérée comme résolue. Sinon, on génère un nouveau couple PGD comme décrit ci-après.

2. *Generation d'un nouveau couple PGD* : Une decomposition $\delta u_k^{(n+1)}$ d'ordre k est connue, une nouvelle approximation de $u^{(n+1)}$ est définie par : $u^{(n+1)} \approx u_{k+1}^{(n+1)} = u^{(n)} + \delta u_k^{(n+1)} + \Phi \lambda = u^{(n)} + \delta u_{k+1}^{(n+1)}$. La construction PGD progressive d'un nouveau couple $(\Phi, \lambda) \in \mathcal{V} \otimes \mathcal{P}$ est obtenue classiquement par un algorithme de puissance itérée [8, 6].

3 La méthode des points de référence

La méthode des points de référence [5] a pour objectif de réduire le coût de calcul des intégrales issues de la projection Galerkin sur la ROB. Deux ingrédients sont introduits : (i) un cadre d'approximation introduisant des points de référence afin de simplifier l'évaluation des intégrales, (ii) une technique de reconstruction à coût de calcul faible permettant de reconstruire par des formules explicites une représentation par patch en variables séparées de rang faible des opérateurs sans avoir recours à des techniques de SVD, ce qui est un grand avantage pour le calcul d'intégrale par séparation de variables. Contrairement à l'EIM, la RPM n'est pas basée sur l'approximation des fonctions non-linéaires (i.e. l'opérateur tangent \mathbf{H}^-). Il s'agit d'une technique d'approximation des intégrales mises en jeu dans la projection Galerkin, de la même manière qu'une technique d'intégration numérique. Plus précisément, elle permet d'approcher, pour un coût faible, chaque contribution α_{ij} au modèle réduit du **Problème 6** :

$$\alpha_{ij} = \int_{\mathcal{D} \times \Omega} \lambda^*(\mu) \Phi_i(\mathbf{x}) \mathbf{H}^-(u^{(n)}; \mathbf{x}, \mu) \Phi_j(\mathbf{x}) \delta \lambda_i(\mu) dx d\mu, \quad 1 \leq i, j \leq k \quad (11)$$

sans sommation sur les indices répétés. La RPM est utilisée à chaque itération de la méthode LATIN afin de résoudre l'étape préliminaire. Même si l'approximation RPM des intégrales n'est pas de bonne qualité, la génération de nouveaux couples PGD permet de corriger et d'améliorer la solution au cours des itérations LATIN.

3.1 Forme compressée et composantes généralisées

L'espace des paramètres est partitionné en m_μ sous-intervalles \mathcal{D}_i de taille $\Delta \mu_i$. Le centre μ_i du sous-intervalle \mathcal{D}_i est appelé *paramètre de référence*. Pour le domaine spatial, m_x points x_j sont introduits et le domaine Ω est divisé en m_x sous-domaines Ω_j de taille $\Delta \omega_j$. Les points x_j sont appelés *points de référence*. Ces points et paramètres de référence sont définis arbitrairement et peuvent être choisis indépendamment des espaces de discrétisation.

Une zone d'influence est définie autour de chaque point du domaine espace-paramètre défini par les coordonnées du point de référence x_j et du paramètre de référence μ_i . La partie du domaine, $\mathcal{D}_i \times \Omega_j$, est appelé *patch de référence* (i, j) . Ainsi, le domaine $\mathcal{D} \times \Omega$ est découpé en $m_\mu \times m_x$ patches.

Une fonction f définie sur le domaine $\mathcal{D} \times \Omega$ est représentée par ses *composantes généralisées*, $\bar{f} := \{(\bar{a}_{ij}, \bar{b}_{ij})\}$, définies comme suit. Pour $1 \leq i \leq m_\mu$ et $1 \leq j \leq m_x$:

$$\bar{f} := \left\{ \begin{array}{l|l} \bar{a}_{ij}(\mu) = f(\mu, x_j) & \text{si } \mu \in \mathcal{D}_i \\ \bar{a}_{ij}(\mu) = 0 & \text{sinon} \end{array} \quad \left| \quad \begin{array}{l|l} \bar{b}_{ij}(x) = f(\mu_i, x) & \text{si } x \in \Omega_j \\ \bar{b}_{ij}(x) = 0 & \text{sinon} \end{array} \right. \right\} \quad (12)$$

Notons que par construction $\bar{a}_{ij}(\mu_i) = \bar{b}_{ij}(x_j)$. Les composantes généralisées fournissent une sorte de *forme compressée* des quantités (**Figure 2**). Pour une fonction de deux variables, une composante généralisée associée à un point ou paramètre de référence correspond à une ligne. En dimension plus élevée, elle correspond à un hyperplan, une hypersurface ou une sous-variété suivant les cas.

3.1.1 Reconstruction d'une représentation par patch en variables séparées de rang faible

Une représentation par patch en variables séparées de rang faible de F , notée $\bar{\bar{F}}$, est obtenue à partir de la forme compressée \bar{F} en générant un produit de fonctions espace-paramètre par patch $\mathcal{D}_i \times \Omega_j$: pour $1 \leq i \leq m_\mu$ et $1 \leq j \leq m_x$:

$$\forall (\mu, x) \in \mathcal{D}_i \times \Omega_j, \quad F(\mu, x) \approx \bar{\bar{F}}(\mu, x) = a_{ij}(\mu) b_{ij}(x) \quad (13)$$

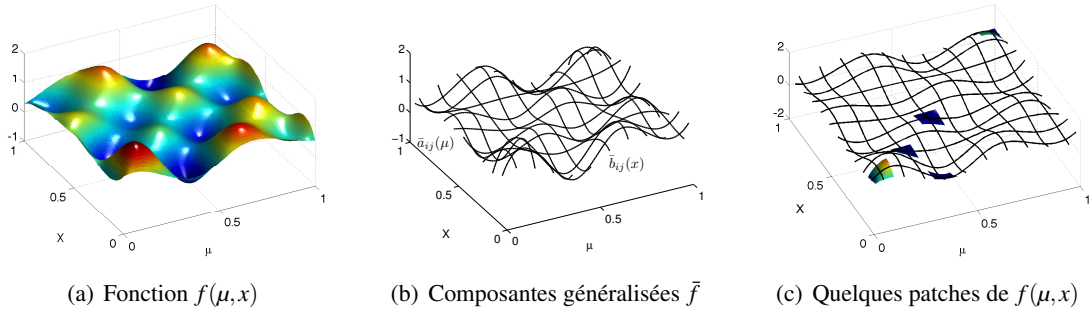


FIGURE 2 – Composantes généralisés et quelques patches d’une fonction f pour $m_\mu = 10$ and $m_x = 10$

où les produits de fonctions $a_{ij}(\mu)b_{ij}(x)$ définis sur chaque patch $\mathcal{D}_i \times \Omega_j$ sont déterminés à partir des composantes généralisées de \bar{F} , $\{\bar{A}_{ij}(\mu)\bar{B}_{ij}(x)\}$, par minimisation de la fonctionnelle suivante : pour $1 \leq i \leq m_\mu$,

$$J(\{(a_{ij}, b_{ij})\}_{1 \leq j \leq m_x}) = \sum_{k=1}^{m_x} [\Delta\omega_k \|(\bar{A}_{ik}(\mu) - a_{ik}(\mu)b_{ik}(x_k)) \lambda_{ik}\|_{\mathcal{D}_i}^2 + \Delta\mu_i \|\bar{B}_{ik}(x) - a_{ik}(\mu_i)b_{ik}(x)\|_{\Omega_k}^2] \quad (14)$$

où $\|\cdot\|_{\mathcal{D}_i}$ et $\|\cdot\|_{\Omega_j}$ sont les normes $L^2(\mathcal{D}_i)$ et $L^2(\Omega_j)$ classiques. Le terme λ_{ik} est une coefficient d’influence donnant plus de poids aux patches voisins du patch $\mathcal{D}_i \times \Omega_k$ suivant la coordonnée spatiale. La minimisation de (14) donnent les formules explicites suivantes : pour $1 \leq i \leq m_\mu$ et $1 \leq j \leq m_x$:

$$a_{ij}(\mu) = \frac{\sum_{k=1}^{m_x} \Delta\omega_k \bar{A}_{ik}(\mu) \bar{A}_{ik}(\mu_i) \lambda_{ik}^2}{\sum_{k=1}^{m_x} \Delta\omega_k \bar{A}_{ik}(\mu_i)^2 \lambda_{ik}^2} \quad \text{et} \quad b_{ij}(x) = \bar{B}_{ij}(x) \quad (15)$$

On notera que le domaine spatial et l’espace des paramètres ne sont pas traités de la même manière. En effet, on constate que pour $1 \leq i \leq m_\mu$, $1 \leq j \leq m_x$ et $\forall x \in \Omega$, $\bar{F}(x, \mu_i) = \bar{B}_{ij}(x)$ (**Figure 3**). Ce choix est raisonnable en mécanique des structures où les gradients en espace des quantités sont généralement plus grands que les gradients en paramètre (ou en temps en quasi-statique). L’erreur e est définie par :

$$e = \frac{\|F - \bar{F}\|_{\mathcal{D} \times \Omega}}{\|\bar{F}\|_{\mathcal{D} \times \Omega}} \quad \text{avec} \quad \|F\|_{\mathcal{D} \times \Omega}^2 = \int_{\mathcal{D} \times \Omega} F^2 dx d\mu = \sum_{i=1}^{m_\mu} \sum_{j=1}^{m_x} \int_{\mathcal{D}_i \times \Omega_j} F^2 dx d\mu \quad (16)$$

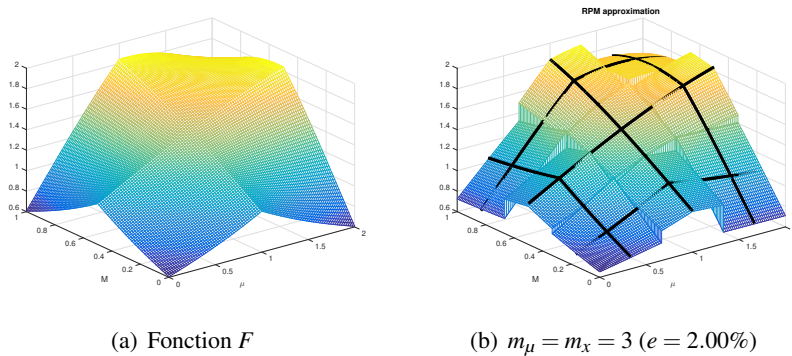


FIGURE 3 – Fonction $F(x, \mu) = e^{-(x-0.5)(\mu-1)} + \sin(x\mu)$ et son approximation \bar{F} for $m_\mu = 3$ et $m_x = 3$

3.2 Calcul d’intégrale par la RPM

La représentation par patch en variables séparées de rang faible, \bar{F} (13), approximation de F , ayant été déterminée par les formules explicites (15), l’intégrale de F sur le domaine espace-paramètre peut

être approchée comme suit :

$$\int_{\mathcal{D} \times \Omega} F(\mu, x) dx d\mu = \sum_{i=1}^{m_\mu} \sum_{j=1}^{m_x} \int_{\mathcal{D}_i \times \Omega_j} F(\mu, x) dx d\mu \approx \sum_{i=1}^{m_\mu} \sum_{j=1}^{m_x} \int_{\mathcal{D}_i \times \Omega_j} a_{ij}(\mu) b_{ij}(x) dx d\mu \quad (17)$$

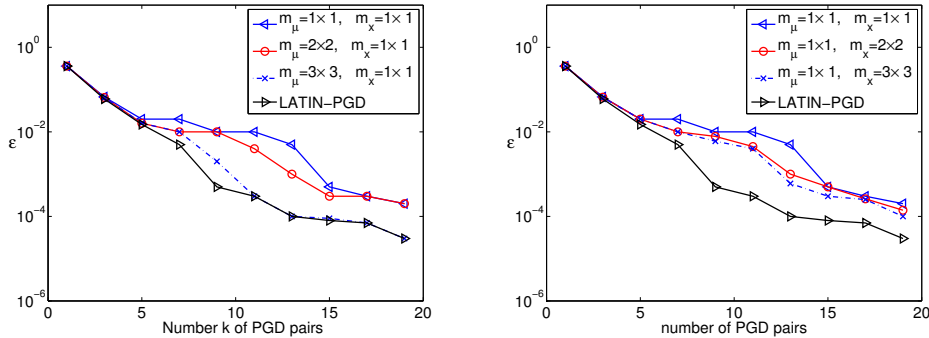
De la même manière, la RPM peut être utilisée pour approcher chaque contribution α_{ij} au modèle réduit définie par (11) et impliquée dans la projection Galerkin de l'étape préliminaire (**Problème 6**). Sans technique d'interpolation ou d'approximation comme l'EIM ou la RPM, la dépendance au paramètre de la direction de recherche \mathbf{H}^- ne permet pas la séparation des intégrales ce qui conduit à une complexité numérique en $O(pNk^2)$ (resp. $O(pN^2k)$) si l'opérateur discrétisé est creux (resp. plein) pour chaque itéré $u^{(n)}(\mathbf{x}, \boldsymbol{\mu})$. La RPM permet de fournir une approximation des intégrales par assemblage de produits tensoriels de matrices par patch. On montre que le gain total en coût de calcul dans le cas d'opérateur creux est de l'ordre de $\left(\frac{m_\mu}{p} + \frac{m_x}{N}\right)^{-1}$. Notons qu'en pratique, le nombre de points de référence m_x est négligeable devant N et on a $\frac{N}{m_x} \gg \frac{p}{m_\mu}$. Le gain est ainsi de l'ordre de $\frac{p}{m_\mu}$.

4 Exemple numérique

La RPM est ici utilisée pour approcher l'étape préliminaire, **Problème 6**, de la stratégie de calcul LATIN-PGD afin de résoudre le **Problème 3**. Dans la suite, cette stratégie est dénommée LATIN-PGD-RPM. L'espace d'approximation éléments finis \mathcal{V}_h est fixe et obtenu par un maillage de 50×50 éléments quadrangulaires bilinéaires conduisant à un espace d'approximation de dimension $N = 2601$. La solution de référence u_{ref} est la solution obtenue en résolvant le **Problème 2** par une méthode de Newton (méthode classique sans réduction de modèle) pour chaque valeur de paramètre (**Figure 1**). Un nombre $p = 225$ de valeurs de paramètre $\boldsymbol{\mu}$ sont choisies à partir d'une grille régulière. L'erreur relative suivante ε est définie :

$$\varepsilon = 1/p \sum_{j=1}^p \frac{\|u_{ref}(\mathbf{x}, \boldsymbol{\mu}_j) - u_m(\mathbf{x}, \boldsymbol{\mu}_j)\|_{L^2}}{\|u_{ref}(\mathbf{x}, \boldsymbol{\mu}_j)\|_{L^2}} \quad (18)$$

où u_m est la solution obtenue par réduction de modèle. L'ajout de points de référence fait tendre la courbe



(a) Erreur ε for $m_\mu = \{1 \times 1, 2 \times 2, 3 \times 3\}$ and $m_x = 1 \times 1$ (b) Erreur ε for $m_\mu = 1 \times 1$ and $m_x = \{1 \times 1, 2 \times 2, 3 \times 3\}$

FIGURE 4 – Evolution de ε (18) en fonction du nombre de paires PGD différents nombres de points et paramètres de référence

de la LATIN-PGD-RPM vers celle obtenue avec la LATIN-PGD (**Figure 4**). La RPM n'est utilisée que pour résoudre le **Problème 6**. Aussi, si l'approximation n'est pas de bonne qualité, la génération de nouveaux couples PGD permet de corriger l'approximation et de faire décroître l'erreur de façon monotone, la convergence étant toujours assurée. Bien entendu, cela conduit à générer un peu plus de couples PGD comparé à la LATIN-PGD pour un niveau d'erreur donné. Pour une erreur de $\varepsilon = 10^{-2}$, le gain en temps CPU par rapport à la méthode directe passe de 6 pour la LATIN-PGD à 18 pour la LATIN-PGD-RPM (**Table 1**). Ce gain de 3 entre les deux méthodes devrait être bien plus important pour des problèmes plus complexes, à grand nombre de paramètres.

	LATIN-PGD	LATIN-PGD-RPM		
		$m_\mu = 1 \times 1$	$m_\mu = 2 \times 2$	$m_\mu = 3 \times 3$
		$m_x = 1 \times 1$	$m_x = 1 \times 1$	$m_x = 1 \times 1$
Nombre de couples PGD	7	9	7	7
Gain par rapport à la méthode directe	6	18	14.2	11.5

TABLE 1 – Gain en temps CPU par rapport à la méthode directe pour une erreur de $\varepsilon = 10^{-2}$

5 Conclusion

Une technique d’approximation, appelée méthode des points de référence (RPM), a été proposée afin de réduire la complexité des opérations algébriques nécessaires à la construction de modèles réduits dans le cadre de la Proper Generalized Decomposition pour des problèmes non-linéaires en temps ou paramétrés.

La RPM a été utilisée pour approcher, à chaque itération de la LATIN, l’étape dite préliminaire, qui correspond à une mise à jour du modèle réduit à partir de la base réduite générée jusqu’à l’itération précédente. La RPM permet d’approcher les intégrales issues de la projection Galerkin sur une base réduite évolutive. Pour cela deux ingrédients ont été introduits : (i) un cadre d’approximation introduisant des points de référence afin de simplifier l’évaluation des intégrales, (ii) une technique de reconstruction à coût de calcul faible permettant de reconstruire par des formules explicites une représentation par patch en variables séparées de rang faible des opérateurs sans avoir recours à des techniques de SVD, ce qui est un grand avantage pour le calcul d’intégrale par séparation de variables.

Un application de la RPM et de la LATIN-PGD pour résoudre une EDP elliptique non-linéaire, paramétrée, a été proposée. Même avec peu de points et paramètres de référence, la courbe d’erreur obtenue avec RPM est très proche de celle obtenue sans RPM, la convergence étant toujours assurée. Par ailleurs, les gains en complexité et en temps CPU ont pu être divisés par un ordre de grandeur avec la RPM. Dans l’exposé, une comparaison de l’EIM et de la RPM sera proposée.

Remerciements

L’Agence Nationale pour la Recherche est remerciée pour son soutien financier du projet SIM-DREAM du programme COSINUS 2010 (ANR-10-COSI-000).

Références

- [1] M. Barrault, Y. Maday, N.C. Nguyen et A.T. Patera. An ‘empirical interpolation’ method : application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Académie des Sciences. Paris.*, Ser. I(339) :667–672, 2004.
- [2] M. Capaldo, P.-A. Guidault, D. Néron, et P. Ladevèze. The Reference Point Method, a “hyperreduction” technique : Application to PGD-based nonlinear model reduction. *Soumis*, 2016.
- [3] F. Chinesta, P. Ladevèze et E. Cueto. A short review on model order reduction based on proper generalized decomposition. *Arch Comput Methods Eng*, (18) :395–404, 10 2011.
- [4] M.A. Grepl, Y. Maday, N.C. Nguyen et A.T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *Modélisation mathématique et analyse numérique*, 41(3) :575–605, 2007.
- [5] P. Ladevèze. A computational technique for the integrals over the time-space domain in connection with the latin method (in french). Technical Report 193, LMT Cachan, 1997.
- [6] P. Ladevèze. *Nonlinear Computational Structural Mechanics - new approaches and non-incremental methods of calculation*. Mechanical Engineering Series. Springer New York, 1999.
- [7] P. Ladevèze, J.-C. Passieux et D. Néron. The latin multiscale computational method and the proper generalized decomposition. *Computational Methods Applied Mechanical Engineering*, (199) :1287–1296, 6 2010.
- [8] A. Nouy. A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. *Computational Methods Applied Mechanical Engineering*, 199(23-24) :1603–1626, January 2010.