École Centrale de Nantes

ÉCOLE DOCTORALE

Sciences pour l'ingénieur, Géosciences, Architecture

Année 2012 *N° B.U. :*

Thèse de Doctorat

Spécialité : Génie Mécanique

Présentée et soutenue publiquement par :

Loïc GIRALDI

le 27 novembre 2012

à l'École Centrale de Nantes

Titre

Contributions aux Méthodes de Calcul Basées sur l'Approximation de Tenseurs et Applications en Mécanique Numérique

Jury

Président :	**H.G. MATTHIES**	**Professeur, Technische Universität Braunschweig**
Rapporteurs :	**C. REY**	**Professeur, École Normale Supérieure de Cachan**
	P. VILLON	**Professeur, Université de Technologie de Compiègne**
Examinateurs :	**P. CARTRAUD**	**Professeur, École Centrale de Nantes**
	A. FALCÓ	**Professeur, Universidad CEU Cardenal Herrera**
	G. LEGRAIN	**Maître de Conférences, École Centrale de Nantes**
	A. NOUY	**Professeur, École Centrale de Nantes**
Membre invité :	**F. CHINESTA**	**Professeur, École Centrale de Nantes**

Directeur de thèse : Anthony NOUY
Co-encadrants : Patrice CARTRAUD - Grégory LEGRAIN
Laboratoire : Institut de Recherche en Génie Civil et Mécanique, École Centrale de Nantes - BP 92101 - 44321 Nantes Cedex 03

N° ED 498-247

Résumé Cette thèse apporte différentes contributions à la résolution de problèmes de grande dimension dans le domaine du calcul scientifique, en particulier pour la quantification d'incertitudes. On considère ici des problèmes variationnels formulés dans des espaces produit tensoriel.

On propose tout d'abord une stratégie de préconditionnement efficace pour la résolution de systèmes linéaires par des méthodes itératives utilisant des approximations de tenseurs de faible rang. Le préconditionneur est recherché comme une approximation de faible rang de l'inverse. Un algorithme glouton permet le calcul de cette approximation en imposant éventuellement des propriétés de symétrie ou un caractère creux. Ce préconditionneur est validé sur des problèmes linéaires symétriques ou non symétriques.

Des contributions sont également apportées dans le cadre des méthodes d'approximation directes de tenseurs qui consistent à rechercher la meilleure approximation de la solution d'une équation dans un ensemble de tenseurs de faibles rangs. Ces méthodes, parfois appelées « Proper Generalized Decomposition » (PGD), définissent l'optimalité au sens de normes adaptées permettant le calcul a priori de cette approximation. On propose en particulier une extension des algorithmes gloutons classiquement utilisés pour la construction d'approximations dans les ensembles de tenseurs de Tucker ou hiérarchiques de Tucker. Ceci passe par la construction de corrections successives de rang un et de stratégies de mise à jour dans ces ensembles de tenseurs. L'algorithme proposé peut être interprété comme une méthode de construction d'une suite croissante d'espaces réduits dans lesquels on recherche une projection, éventuellement approchée, de la solution. L'application à des problèmes symétriques et non symétriques montre l'efficacité de cet algorithme. Le préconditionneur proposé est appliqué également dans ce contexte et permet de définir une meilleure norme pour l'approximation de la solution.

On propose finalement une application de ces méthodes dans le cadre de l'homogénéisation numérique de matériaux hétérogènes dont la géométrie est extraite d'images. On présente tout d'abord des traitements particuliers de la géométrie ainsi que des conditions aux limites pour mettre le problème sous une forme adaptée à l'utilisation des méthodes d'approximation de tenseurs. Une démarche d'approximation adaptative basée sur un estimateur d'erreur a posteriori est utilisée afin de garantir une précision donnée sur les quantités d'intérêt que sont les propriétés effectives. La méthodologie est en premier lieu développée pour l'estimation de propriétés thermiques du matériau, puis est étendue à l'élasticité linéaire.

Mots-Clefs : Approximation de tenseurs – Solveurs itératifs – Proper Generalized Decomposition (PGD) – Réduction de modèle – Préconditionnement – Algorithmes gloutons – Homogénéisation

Abstract This thesis makes several contributions to the resolution of high dimensional problems in scientific computing, particularly for uncertainty quantification. We consider here variational problems formulated on tensor product spaces.

We first propose an efficient preconditioning strategy for linear systems solved by iterative solvers using low-rank tensor approximations. The preconditioner is found as a low rank approximation of the inverse of the operator. A greedy algorithm allows to compute this approximation, possibly imposing symmetry or sparsity properties. This preconditioner is validated on symmetric and non-symmetric linear problems.

We also make contributions to direct tensor approximation methods which consist in computing the best approximation of the solution of an equation in a set of low-rank tensors. These techniques, sometimes coined « Proper Generalized Decomposition » (PGD), define optimality with respect to suitable norms allowing an a priori approximation of the solution. In particular, we extend the classically used greedy algorithms to the sets of Tucker tensors and hierarchical Tucker tensors. To do so, we construct successive rank one corrections and update the approximation in the previous sets. The proposed algorithm can be interpreted as a construction of an increasing sequence of reduced spaces in which we compute a possibly approximate projection of the solution. The application of these methods to symmetric and non-symmetric problems shows the efficiency of this algorithm. The proposed preconditioner is also applied and allows to define a better norm for the approximation of the solution.

We finally apply these methods to the numerical homogenization of heterogeneous materials whose geometry is extracted from images. We first present particular treatment of the geometry and the boundary conditions to use them in the tensor approximation framework. An adaptive approximation procedure based on an a posteriori error estimator is proposed to ensure a given accuracy on the quantities of interest which are the effective properties. The method is first developed to estimate effective thermal properties, and is extended to linear elasticity.

Keywords : Tensor approximation – Iterative solvers – Proper Generalized Decomposition (PGD) – Model reduction – Preconditioning – Greedy algorithms – Homogenization

Discipline : Science de l'ingénieur

A Maria, à Léonie

Remerciements

Je tiens tout d'abord à remercier Anthony Nouy qui m'a permis d'arriver au bout de ce travail, je retiendrais particulièrement sa grande sympathie et sa disponibilité. Merci également à Patrice Cartraud qui fut à l'origine de cette thèse, et à Grégory Legrain pour son soutien durant ces 3 dernières années et jusqu'au bout du monde.

Un grand merci aux membres du jury, son président Hermann Matthies, les rapporteurs Christian Rey et Pierre Villon, ainsi qu'Antonio Falco et Francisco Chinesta.

Je remercie Gottfried Laschet et Naoki Takano pour m'avoir accueilli chaleureusement dans leurs laboratoires respectifs.

Merci à ma famille pour son soutien et sa confiance, ainsi qu'à Léa d'avoir traversé cette aventure à mes côtés.

Un dernier remerciement à tous les amis du laboratoires, qui ont fait régner la bonne humeur autour de ce travail de recherche.

Table des matières

Introduction

Contexte

Dans cette thèse, on s'intéresse aux problèmes variationnels formulés sur des espaces de grande dimension. La résolution de ce type de problème est un enjeu industriel fort, notamment dans le cadre de la quantification d'incertitudes pour des modèles régis par des équations aux dérivées partielles stochastiques. La solution de telles équations permet à l'ingénieur de prédire la réponse globale d'un système aux paramètres d'entrée aléatoires pour mener des analyses de sensibilité, de fiabilité ou même encore des optimisations robustes sous incertitudes.

Dans le cadre des méthodes de représentation fonctionnelle des incertitudes [33, 55, 62], l'aléa est tout d'abord modélisé par un nombre fini de d variables aléatoires indépendantes $\xi = (\xi^1, \ldots, \xi^d)$ à valeurs dans un espace paramétrique Ξ. Une approximation de la solution u d'une équation aux dérivées partielles stochastique est alors recherchée sous la forme $u(\xi) \approx \sum_{i=1}^{N} u_i \psi_i(\xi)$ où $(\psi_i)_{1 \leq i \leq N}$ est un ensemble de fonctions judicieusement choisi, et où $u(\xi)$ et u_i sont des éléments d'un espace fonctionnel $\mathcal{W}$. On peut typiquement utiliser une méthode de Galerkin pour calculer cette approximation [5, 31, 33, 58]. On peut aussi utiliser une méthode d'interpolation (ou collocation) [4, 83]. L'approximation est alors cherchée dans l'espace $\mathcal{W}_N = \operatorname{span}\{u(\xi_i);\ 1 \leq i \leq N\}$ où $(\xi_i)_{1 \leq i \leq N} \in \Xi^N$ sont des points de l'ensemble des paramètres. Ces approches nécessitent généralement des résolutions pour un grand nombre de points. Cette approximation peut être également estimée par une méthode de régression linéaire à partir de solutions $(u(\xi_i))_{1 \leq i \leq S}$ calculées pour un ensemble de S réalisations des paramètres aléatoires [10, 17, 73], S devant être bien supérieur à la dimension N de l'espace d'approximation.

Pour des problèmes de grande dimension d, les choix a priori des fonctions $(\psi_i)_{1 \leq i \leq N}$ conduisent généralement à un nombre d'éléments N très important pour avoir une bonne approximation, et le calcul des $(u_i)_{1 \leq i \leq N}$ devient alors prohibitif. De nombreuses approches ont été proposées pour diminuer la complexité du calcul. Elles s'appuient toutes sur le choix ou la construction d'un ensemble de fonctions de faible dimension n dans lequel on recherche une approximation de la solution.

Une première classe de méthodes de réduction de dimension s'appuie sur la résolution du problème en certains points $(\xi_i)_{1 \leq i \leq m} \in \Xi^m$ pour construire un espace d'approximation réduit $\mathcal{W}_n$ dans $\mathcal{W}$. La décomposition orthogonale aux valeurs propres (Proper Orthogonal Decomposition, POD), aussi appelée décomposition en valeurs singulières ou décomposition de Karhunen-Loève empirique suivant le contexte [25, 41, 44, 56, 57, 66] est largement utilisée. Elle consiste à définir $\mathcal{W}_n$ comme l'espace propre dominant de dimension n l'opérateur de corrélation des solutions $(u(\xi_i))_{1 \leq i \leq m}$.

On peut aussi citer la méthode dite des bases réduites (Reduced Basis) [1, 35, 59, 67, 79], où cette fois les points sont sélectionnés automatiquement et de manière optimale. L'approche se décompose en deux étapes. Dans la première, on choisit les points $(\xi_i)_{1 \leq i \leq n}$ de manière incrémentale (gloutonne)

en se basant sur un estimateur d'erreur. L'approximation sur le domaine Ξ est le résultat d'une projection de Galerkin sur l'espace $\mathcal{W}_n = \operatorname{span}\{u(\xi_i);\ 1 \leq i \leq n\}$. La méthode suppose que l'ensemble $(u(\xi))_{\xi\in\Xi}$ peut être approché précisément par des éléments d'un espace vectoriel de faible dimension.

Une autre façon d'aborder le problème consiste à trouver une représentation parcimonieuse (ou creuse) de la solution. On parle d'approximation creuse lorsque seuls quelques éléments d'une base, ou d'un « dictionnaire » de fonctions, suffisent à la bonne approximation de la solution. Pour une précision donnée, le problème de meilleure approximation creuse s'écrit comme une minimisation de la pseudo-norme ℓ^0 correspondant au nombre de termes non nuls de la représentation de l'approximation. Ce problème d'optimisation est de complexité combinatoire et devient rapidement prohibitif en termes de coût de calcul. En pratique, le problème est régularisé en utilisant des normes induisant la parcimonie telle que la norme ℓ^1 [15, 75]. On aboutit alors à un problème d'optimisation convexe pour lequel des algorithmes efficaces peuvent être employés [6]. Si le calcul est trop coûteux, des algorithmes gloutons peuvent être employés. Ils consistent à sélectionner un par un des éléments du dictionnaire [24, 65]. Ces approches supposent que la solution admet une représentation suffisamment creuse dans un dictionnaire préalablement choisi. Une analyse de convergence de ces méthodes pour une classe d'équations paramétriques stochastiques et des dictionnaires de polynômes a été effectuée dans [18, 19].

On va s'intéresser dans ce mémoire à la résolution de problèmes définis sur un espace produit tensoriel $u \in \mathcal{V} = \mathcal{V}^1 \otimes \ldots \otimes \mathcal{V}^d$ par des méthodes exploitant cette structure afin de réduire la complexité. De nombreux problèmes de grande dimension, tels que les problèmes stochastiques, appartiennent à cette classe de problèmes.

Une approche de résolution pour les problèmes définis sur un espace de tenseurs consiste à utiliser des grilles creuses (Sparse Grids) [13]. Il s'agit d'introduire une séquence d'espaces imbriqués $\mathcal{V}_1^\mu \subset \ldots \subset \mathcal{V}_n^\mu \subset \mathcal{V}^\mu$, $1 \leq \mu \leq d$ et de définir un espace d'approximation tensorisé de la forme $\mathcal{W}_n = \sum_{i\in\mathcal{I}\subset\mathbb{N}^d} \mathcal{V}_{i_1}^1 \otimes \ldots \otimes \mathcal{V}_{i_d}^d$ où l'ensemble d'indices $\mathcal{I}$ est judicieusement choisi ou construit de manière adaptative. Ces techniques de tensorisation sont également employées pour la construction de quadratures creuses en grande dimension, qui peuvent être utilisées dans le cadre des méthodes de projection pour les équations paramétriques stochastiques [20, 58]. Ces méthodes présentent classiquement une complexité non linéaire en d, ce qui les limite à des dimensions raisonnables.

Pour atteindre une complexité linéaire en d et ainsi aborder des problèmes de très grande dimension, des méthodes d'approximation de tenseurs de faible rang ont été proposées. Il s'agit de chercher une approximation u_r de la solution u sous la forme

$$u_r = \sum_{i=1}^{r} v_i^1 \otimes \ldots \otimes v_i^d, \tag{1}$$

avec r suffisamment petit pour que le calcul de u_r reste abordable. L'ensemble d'approximation n'étant pas un espace vectoriel, le problème à résoudre est maintenant un problème d'approximation non linéaire. Plusieurs méthodes ont été proposées pour la construction a priori de telles approxima-

tions. Certaines s'appuient sur les solveurs itératifs classiques de l'algèbre linéaire [7, 49, 50] couplés à des méthodes d'approximation de tenseurs. D'autres construisent directement une approximation de la solution par la formulation d'un problème d'optimisation sur un sous-ensemble de tenseurs [2, 3, 11, 16, 27, 52, 60, 61], ce problème d'optimisation étant défini pour permettre le calcul a priori de cette approximation. Ces approches sont parfois appelées « Proper Generalized Decomposition » (PGD), cette dénomination étant prinpalement utilisée dans le cas d'approximations de la forme (1) appelée approximations canoniques de rang r. Ce format canonique est bien connu pour avoir de mauvaises propriétés mathématiques en dimension $d > 2$ et pour $r > 1$. Les algorithmes utilisés s'appuient donc généralement sur des constructions gloutonnes permettant d'obtenir une succession de problèmes bien posés et de complexité réduite. Cependant ces constructions gloutonnes s'avèrent parfois très sous-optimales.

Un autre volet de la résolution d'équations concerne le préconditionnement. Ce dernier sert deux objectifs : diminuer la sensibilité de la solution aux approximations et accélérer la convergence de certains algorithmes. Le préconditionnement d'un opérateur sous format de tenseur ne peut pas être réalisé par les méthodes usuelles. En effet, on ne peut pas aisément généraliser les approches se basant sur de la décomposition ou de la factorisation de matrices (Jacobi, les factorisations LU ou de Cholesky incomplètes, ...) dans le cadre tensoriel.

Des préconditionneurs de rang un ont été proposés qui reposent sur l'approximation de l'inverse d'un opérateur sous forme de tenseur de rang un. Un premier type de préconditionneur a été proposé dans le cadre des équations stochastiques paramétriques, où l'espace de tenseurs est de la forme $\mathcal{W} \otimes \mathcal{S}$, avec $\mathcal{W}$ (resp. $\mathcal{S}$) l'espace d'approximation associé aux variables déterministes (resp. aléatoires). Le préconditionneur est alors construit à partir de l'inverse de l'espérance de l'opérateur aléatoire [32].

Un autre préconditionneur de rang un bien plus général a été proposé dans le cadre des réseaux d'automates stochastiques pour les grandes dimensions [54]. L'opérateur est tout d'abord approché par un tenseur de rang un, puis le préconditionneur est défini par l'inverse de cette approximation. On note qu'il a aussi été exploité dans le cadre des équations stochastiques paramétriques dans [78].

On peut citer aussi le préconditionneur de rang un proposé dans [84]. Le préconditionneur est basé sur l'approximation de l'inverse de rang un par une minimisation en norme canonique.

Pour construire un préconditionneur de rang plus élevé, il existe deux façons d'aborder le problème dans la littérature. La première repose sur un développement en série tronqué de l'inverse [46]. La seconde approche consiste à supposer une structure particulière de l'approximation de l'inverse. Dans [76], le préconditionneur est recherché sous la forme $P = P^1 \otimes I \otimes \ldots \otimes I + I \otimes P^2 \otimes I \otimes \ldots \otimes I + \ldots + I \otimes \ldots \otimes I \otimes P^d$.

Contributions

Dans le mémoire, un effort a été réalisé afin de présenter dans un cadre unifié les méthodes d'approximations de tenseurs. Celles ci deviennent alors directement applicables pour approcher un opérateur, une inverse ou encore la solution d'un système linéaire.

Un nouveau préconditionneur pouvant être de rang quelconque est présenté. Il approche de manière arbitraire l'inverse de l'opérateur, sans faire aucune hypothèse sur la forme de cette dernière. Il est entièrement adaptatif et est appliqué dans ce travail à des opérateurs aussi bien symétriques que non symétriques. En outre, des propriétés peuvent être imposées à ce préconditionneur telles que la symétrie ou un caractère creux. Sa construction est basée sur une approximation gloutonne de l'inverse au sens d'une norme appropriée. Ce préconditionneur est appliqué avec succès aux solveurs itératifs, et une stratégie de préconditionnement des approches de type PGD est aussi proposée. Ceci passe par le choix d'une norme appropriée qui correspond à une minimisation du résidu préconditionné.

De nouveaux solveurs sont proposés, à la croisée des chemins entre les solveurs itératifs et l'approximation directes dans des ensembles de tenseurs. Les approximations directes pour trouver une approximation de haut rang sont coûteuses et difficiles à calculer par des algorithmes génériques. Pour contourner cette difficulté, on utilise généralement une approximation gloutonne. Celle-ci est est peu coûteuse par itération, mais la construction est sous-optimale. On introduit ici plusieurs variantes permettant d'améliorer ces constructions gloutonnes. L'une d'entre elles consiste à utiliser la construction gloutonne comme initialisation pour un problème de minimisation dans un certain ensemble de tenseurs de haut rang. La construction gloutonne peut également être utilisée comme une méthode de construction progressive d'espaces d'approximation réduits dans lesquels une projection éventuellement approchée de la solution est calculée. On présente en particulier des stratégies basées sur la construction d'une suite croissante d'espaces réduits tensorisés obtenus à partir de corrections de rang un. Le calcul de la projection dans ces espaces souffre encore de la malédiction de la dimensionnalité. On propose alors le calcul d'une approximation de cette projection utilisant des formats de tenseurs hiérarchiques de Tucker dont la complexité est linéaire avec la dimension. Sur les exemples testés, les solveurs proposés sont plus précis que les solveurs itératifs mais aussi plus coûteux par itération que les solveurs basés sur une construction gloutonne de la solution. Ce surcoût est contrebalancé par la très bonne convergence des algorithmes. On observe en particulier que ces stratégies de construction d'espaces réduits fournissent des approximations de qualité équivalente à celles obtenues par approximation directe dans ces mêmes ensembles de tenseurs.

Ces algorithmes sont enfin appliqués à des problèmes d'homogénéisation numérique de matériaux hétérogènes dont les géométries proviennent d'images. Un traitement particulier de la géométrie et des conditions aux limites permet de mettre ces problèmes sous une forme adaptée à l'utilisation des méthodes d'approximation de tenseurs. Une méthode d'approximation adaptative basée sur un estimateur d'erreur a posteriori en quantité d'intérêt est proposée dans le but d'estimer le tenseur de conductivité homogénéisé avec une précision donnée. L'estimation d'erreur nécessite la résolution approchée d'un problème dual. Une relation remarquable entre les problèmes primal et dual permet d'optimiser l'efficacité de la procédure adaptative.

Organisation du mémoire

Dans le chapitre I, la méthodologie pour exprimer un problème aux limites paramétriques dans un cadre tensoriel est présentée. Le produit tensoriel ainsi que les espaces produit tensoriel sont

définis, et un exemple de formulation d'un problème discret à partir d'un problème stochastique paramétrique est présenté.

Au chapitre II, la notion de format de tenseurs est approfondie. Les tenseurs canoniques, de Tucker et hiérarchiques de Tucker sont présentés, ainsi que différentes méthodes pour approcher un tenseur dans un format donné. La présentation est très générale afin de pouvoir appliquer ces méthodes à différents objets mathématiques, tels qu'un vecteur ou un opérateur.

Au chapitre III, on aborde les solveurs itératifs couplés aux méthodes d'approximation de tenseurs. Différents solveurs issus de la littérature sont présentés, destinés aux opérateurs symétriques ou non. Le nouveau préconditionneur est alors introduit et appliqué à ce type de solveurs.

Au chapitre IV, les méthodes d'approximation directes dans un sous-ensemble de tenseurs sont développées. Ces approches sont directement issues des algorithmes d'approximation de faible rang de tenseurs pour des métriques définies à partir de l'opérateur. On y présente comment préconditionner ce type d'approche, ainsi que de nouveaux solveurs possédant de bonnes propriétés de convergences.

Dans le chapitre V, on présente l'application de ces méthodes à l'homogénéisation numérique de matériaux hétérogènes.

Formulation d'un problème variationnel sur un espace produit tensoriel

Sommaire

Ce chapitre va permettre de fixer les bases et les notations utiles pour les chapitres suivants. On abordera dans un premier temps les bases minimales d'algèbre tensoriel. Ceci nous permettra par la suite de formuler un problème aux limites sur un espace produit tensoriel. On illustrera le tout sur un problème aux limites stochastiques paramétriques pour conclure sur les apports des méthodes d'approximation de tenseurs sur la réduction de dimensionnalité des problèmes discrétisés.

I.1 Espace produit tensoriel

I.1.1 En dimension quelconque

On note $D = \{1, \dots, d\}$, $d \in \mathbb{N}^*$. Soient d espaces de Banach $(\mathcal{V}^\mu)_{\mu \in D}$. La norme sur $\mathcal{V}^\mu$ est notée $\|\cdot\|_\mu$. D'après Hackbusch [38], une construction possible du produit tensoriel et de l'espace produit tensoriel algébrique est donnée par

Définition I.1.1 (Produit tensoriel et espace produit tensoriel algébrique)**.** *Soit $\mathcal{T}$ un espace vectoriel. Soit une application $\otimes : \mathcal{V}^1 \times \ldots \times \mathcal{V}^d \to \mathcal{T}$ telle que*

- $\mathcal{T} = \operatorname{span}\left\{\otimes(v^1, \dots, v^d);\ v^\mu \in \mathcal{V}^\mu,\ \forall \mu \in D\right\}$
- *$\otimes$ est multilinéaire*
- *Si $\left(v^\mu_{i_\mu}\right)_{1 \le i_\mu \le r_\mu}$, $r_\mu \in \mathbb{N}$, $r_\mu \le n_\mu$, $\mu \in D$, sont des familles libres alors $\left(\otimes(v^1_{i_1}, \dots, v^d_{i_d})\right)_{1 \le i_\mu \le r_\mu,\ \mu \in D}$ est une famille libre.*

Alors, $\otimes$ est un produit tensoriel et $\mathcal{T}$ est un espace produit tensoriel algébrique.

On note $\mathcal{T} = {}_a\bigotimes_{\mu=1}^{d}\mathcal{V}^\mu = {}_a\bigotimes_\mu \mathcal{V}^\mu$ et $\otimes(v^1,\dots,v^d) = v^1\otimes\ldots\otimes v^d = \bigotimes_{\mu=1}^{d} v^\mu = \bigotimes_\mu v^\mu$. Par convention on notera $\bigotimes_{\substack{\mu=1\\ \mu\neq\lambda}}^{d} = \bigotimes_{\mu\neq\lambda}$.

On munit $\mathcal{T}$ d'une norme $\|\cdot\|$. On définit alors l'*espace de Banach de tenseurs* $\mathcal{V}$ comme la complétion de $\mathcal{T}$ par rapport à $\|\cdot\|$, c'est-à-dire $\mathcal{V} = \overline{\mathcal{T}}^{\|\cdot\|}$. Ainsi tout élément $v\in\mathcal{V}$ peut s'écrire

$$v = \sum_{i=1}^{\infty}\bigotimes_\mu v_i^\mu, \quad v_i^\mu\in\mathcal{V}^\mu, \tag{I.1}$$

où la convergence de la série est au sens de la norme $\|\cdot\|$. Un élément $v\in\mathcal{V}$ est appelé *tenseur*. On définit un *tenseur élémentaire* comme un tenseur $v\in\mathcal{V}$ tel que $v = \bigotimes_\mu v^\mu$. On note $\mathcal{V} = \bigotimes_\mu \mathcal{V}^\mu$ où la norme est omise s'il n'y a pas d'ambiguïté.

L'espace $\mathcal{L}(\mathcal{V}^\mu)$ des opérateurs linéaires sur $\mathcal{V}^\mu$, muni de la norme d'opérateur $\|\cdot\|_{\mathcal{L}(\mathcal{V}^\mu)}$ définie par

$$\|A^\mu\|_{\mathcal{L}(\mathcal{V}^\mu)} = \sup_{v\in\mathcal{V}^\mu\setminus\{0\}}\frac{\|A^\mu v\|_\mu}{\|v\|_\mu}, \quad \forall A^\mu\in\mathcal{L}(\mathcal{V}^\mu), \tag{I.2}$$

est un espace de Banach. On munit alors le produit tensoriel algébrique d'espaces ${}_a\bigotimes_\mu\mathcal{L}(\mathcal{V}^\mu)$ de la norme d'opérateur

$$\|A\|_{\mathcal{L}(\mathcal{V})} = \sup_{v\in\mathcal{V}\setminus 0}\frac{\|Av\|}{\|v\|}. \tag{I.3}$$

On définit alors l'espace de Banach de tenseurs $\mathfrak{L}(\mathcal{V}) = \bigotimes_\mu\mathcal{L}(\mathcal{V}^\mu) = \overline{{}_a\bigotimes_\mu\mathcal{L}(\mathcal{V}^\mu)}^{\|\cdot\|_{\mathcal{L}(\mathcal{V})}}$. Un élément $A\in\mathfrak{L}(\mathcal{V})$ s'écrit donc

$$A = \sum_{i=1}^{\infty}\bigotimes_\mu A_i^\mu, \quad A_i^\mu\in\mathcal{L}(\mathcal{V}^\mu). \tag{I.4}$$

Si la norme sur $\mathcal{V}$ est une cross norm, c'est-à-dire $\left\|\bigotimes_{\mu=1}^{d} v^\mu\right\| = \prod_{\mu=1}^{d}\|v^\mu\|_\mu$ pour les tenseurs élémentaires, alors on a aussi

$$\left\|(\bigotimes_\mu A^\mu)(\bigotimes_\mu v^\mu)\right\| \leq \left(\prod_{\mu=1}^{d}\|A^\mu\|_\mu\right)\|v\|. \tag{I.5}$$

Si de plus on suppose que la norme sur $\mathcal{V}$ est une cross norm uniforme, alors cette propriété s'étend à tous les tenseurs de ${}_a\bigotimes_\mu\mathcal{V}^\mu$

$$\left\|\left(\bigotimes_\mu A^\mu\right)v\right\| \leq \left(\prod_{\mu=1}^{d}\|A^\mu\|_{\mathcal{L}(\mathcal{V}^\mu)}\right)\|v\| \tag{I.6}$$

et donc $\left\|\bigotimes_\mu A^\mu\right\|_{\mathcal{L}(\mathcal{V})} = \prod_{\mu=1}^{d}\|A^\mu\|_{\mathcal{L}(\mathcal{V}^\mu)}$. On en déduit que si $\|\cdot\|$ est une cross norm uniforme, $\|A\|_{\mathcal{L}(\mathcal{V})}$ est finie pour tout $A\in\mathfrak{L}(\mathcal{V})$ et donc l'inclusion $\mathfrak{L}(\mathcal{V})\subset\mathcal{L}(\mathcal{V})$. En dimension finie, en raisonnant sur les dimensions on a l'égalité $\mathfrak{L}(\mathcal{V}) = \mathcal{L}(\mathcal{V})$ (voir [38, Section 4.3.7]).

Si maintenant les $(\mathcal{V}^\mu)_{\mu\in D}$ sont des espaces de Hilbert pour le produit scalaire $\langle\cdot,\cdot\rangle_\mu$ de norme

induite $\|\cdot\|_\mu$, on remarque que $\mathcal{V}$ est un espace de Hilbert pour le produit scalaire $\langle\cdot,\cdot\rangle$ défini sur les tenseurs élémentaires par

$$\left\langle \bigotimes_{\mu=1}^{d} v^\mu, \bigotimes_{\mu=1}^{d} w^\mu \right\rangle = \prod_{\mu=1}^{d} \langle v^\mu, w^\mu \rangle_\mu \tag{I.7}$$

et étendu par linéarité. La norme induite $\|\cdot\|$ possède la particularité d'être une cross norm uniforme.

I.1.2 En dimension finie

Dans le cas où les $\mathcal{V}^\mu$ sont de dimension $\dim(\mathcal{V}^\mu) = n_\mu \in \mathbb{N}^*$, l'espace produit tensoriel algébrique est alors complet et on a donc l'égalité $\mathcal{V} = \mathcal{T}$. On déduit directement de la définition que $\dim(\mathcal{V}) = \prod_{\mu\in D} n_\mu$.

Avec $(e_i^\mu)_{1\leq i\leq n_\mu}$ une base de $\mathcal{V}^\mu$, une base de $\mathcal{V}$ est construite par tensorisation des bases. Ainsi, un tenseur $v \in \mathcal{V}$ peut s'écrire

$$v = \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} v_{i_1\ldots i_d} \bigotimes_\mu e_{i_\mu}^\mu, \quad v_{i_1\ldots i_d} \in \mathbb{R}, \tag{I.8}$$

ou encore d'après (I.1)

$$v = \sum_{i=1}^{r_v} \alpha_i \bigotimes_\mu v_i^\mu, \quad \alpha_i \in \mathbb{R},\ v_i^\mu \in \mathcal{V}^\mu, \tag{I.9}$$

avec $r_v \leq \dim(\mathcal{V})$. Un élément $A \in \mathfrak{L}(\mathcal{V})$ s'écrit

$$A = \sum_{i=1}^{r_A} \beta_i \bigotimes_\mu A_i^\mu, \quad \beta_i \in \mathbb{R},\ A_i^\mu \in \mathcal{L}(\mathcal{V}^\mu). \tag{I.10}$$

On peut appliquer l'opérateur A à v de la manière suivante

$$Av = \sum_{i=1}^{r_A} \sum_{j=1}^{r_v} \beta_i\alpha_j \bigotimes_\mu \left(A_i^\mu v_j^\mu\right). \tag{I.11}$$

Exemple I.1.2. *Une matrice de $\mathbb{R}^{m\times n}$ est un tenseur de $\mathbb{R}^m \otimes \mathbb{R}^n$ avec le produit tensoriel défini par $u \otimes v = uv^T$ pour $u \in \mathbb{R}^m$ et $v \in \mathbb{R}^n$.*

I.2 Formulation d'un problème variationnel

On considère un problème variationnel générique :

$$\begin{aligned} &\text{Trouver } u \in \mathcal{V} \text{ tel que} \\ &a(\delta u, u) = \ell(\delta u), \quad \forall \delta u \in \mathcal{V}, \end{aligned} \tag{I.12}$$

où a est une forme bilinéaire, ℓ est une forme linéaire et $\mathcal{V}$ est un espace de Hilbert de tenseurs. L'espace $\mathcal{V}$ étant de dimension infinie, une méthode de Galerkin est utilisée. On introduit les espaces

de dimensions finis $\left(\mathcal{V}_{n_\mu}^\mu\right)_{\mu\in D}$ tels que $\dim(\mathcal{V}_{n_\mu}^\mu) = n_\mu$. Le problème de dimension finie est alors :

$$\begin{aligned} &\text{Trouver } u \in \mathcal{V}_n \text{ tel que} \\ &a(\delta u, u) = \ell(\delta u), \quad \forall \delta u \in \mathcal{V}_n, \end{aligned} \tag{I.13}$$

avec $\mathcal{V}_n = \bigotimes_\mu \mathcal{V}_{n_\mu}^\mu$ et $n = \prod_{\mu\in D} n_\mu$. Avec $(\varphi_i^\mu)_{1\le i\le n_\mu}$ une base de $\mathcal{V}_{n_\mu}^\mu$, d'après la section I.1, une base $(\varphi_i)_{i\in\mathcal{I}}$ de $\mathcal{V}_n$ est donnée par

$$\varphi_i = \bigotimes_\mu \varphi_{i_\mu}^\mu, \quad i \in \mathcal{I} = \{(j_1, \dots, j_d);\ 1 \le j_\mu \le n_\mu,\ \mu \in D\}. \tag{I.14}$$

La solution du problème discrétisé peut s'exprimer sous la forme

$$u = \sum_{i\in\mathcal{I}} u_i \varphi_i, \tag{I.15}$$

où $(u_i)_{i\in\mathcal{I}}$ est solution du problème linéaire défini par

$$\sum_{j\in\mathcal{I}} a_{ij} u_j = \ell_i,\ \forall i \in \mathcal{I}, \tag{I.16}$$

avec $a_{ij} = a(\varphi_i, \varphi_j),\ \ell_i = \ell(\varphi_i)$, ayant pour forme développée

$$\sum_{j_1=1}^{n_1} \dots \sum_{j_d=1}^{n_d} a_{i_1\dots i_d j_1\dots j_d} u_{j_1\dots j_d} = \ell_{i_1\dots i_d}, \quad \forall (i_1, \dots, i_d) \in \mathcal{I}. \tag{I.17}$$

On peut aussi exprimer la solution comme un minimiseur d'une certaine fonctionnelle. Pour cela, on introduit un produit scalaire $\langle\cdot,\cdot\rangle_\mathcal{V}$ sur $\mathcal{V}$ et la norme associée $\|\cdot\|_\mathcal{V}$ tels que la norme soit différente de $\|\cdot\|$ sur $\mathcal{V}$. D'après le théorème de représentation de Riesz, il existe $A \in \mathcal{L}(\mathcal{V})$ et $b \in \mathcal{V}$ tels que

$$a(u, v) = \langle Au, v\rangle_\mathcal{V} \quad \text{et} \quad \ell(v) = \langle b, v\rangle_\mathcal{V}. \tag{I.18}$$

Le problème de départ peut alors s'écrire :

$$\begin{aligned} &\text{Trouver } u \in \mathcal{V} \text{ tel que} \\ &\langle \delta u, Au - b\rangle_\mathcal{V} = 0, \quad \forall \delta u \in \mathcal{V}. \end{aligned} \tag{I.19}$$

Si A est défini, (I.19) est équivalent à

$$\begin{aligned} &\text{Trouver } u \in \mathcal{V} \text{ tel que} \\ &\langle \delta v, Au - b\rangle_\mathcal{V} = 0, \quad \forall \delta v \in A\mathcal{V}. \end{aligned} \tag{I.20}$$

Cette formulation revient à dire que u vérifie

$$\|Au - b\|_\mathcal{V}^2 = \min_{v\in\mathcal{V}} \|Av - b\|_\mathcal{V}^2 \tag{I.21}$$

Remarque I.2.1. *Si A est un opérateur symétrique et coercif, u est l'unique minimiseur de la fonctionnelle*

$$v \mapsto \frac{1}{2} \langle Av, v \rangle_{\mathcal{V}} - \langle b, v \rangle_{\mathcal{V}} . \tag{I.22}$$

I.3 Problème stochastique paramétrique

I.3.1 Cadre continu

Soit $(\Theta, \mathcal{B}, P)$ un espace probabilisé où Θ est l'ensemble des événements élémentaires, $\mathcal{B}$ une $\sigma-$algèbre sur Θ et P une mesure de probabilité sur $\mathcal{B}$. On considère le problème suivant :

$$\begin{aligned} &\text{Trouver } u(\theta) \in \mathcal{W} \text{ tel que} \\ &b_\Theta(\delta u, u(\theta); \theta) = g_\Theta(\delta u; \theta), \quad \forall \delta u \in \mathcal{W},\ \forall \theta \in \Theta, \end{aligned} \tag{I.23}$$

où $\mathcal{W}$ est un espace de Hilbert. En interprétant u comme une fonction de Θ dans $\mathcal{W}$, la formulation (I.23) est forte par rapport à la variable θ. On suppose que u est une variable aléatoire du second ordre. Soit $L^2(\Theta, dP; \mathcal{W})$ l'ensemble des variables aléatoires de second ordre à valeurs dans $\mathcal{W}$ défini par

$$L^2(\Theta, dP; \mathcal{W}) = \left\{ v : \Theta \to \mathcal{W};\ E(\|v\|^2_{\mathcal{W}}) = \int_\Theta \|v(\theta)\|^2_{\mathcal{W}}\ dP(\theta) < +\infty \right\}, \tag{I.24}$$

où E désigne l'espérance mathématique. On notera $\mathcal{S}_\Theta = L^2(\Theta, dP) = L^2(\Theta, dP; \mathbb{R})$ l'ensemble des variables aléatoires à valeurs réelles de second ordre. $\mathcal{S}_\Theta$ est un espace de Hilbert, équipé du produit scalaire $\langle \cdot, \cdot \rangle_{\mathcal{S}_\Theta} = E(\cdot\ \cdot)$ et de la norme induite $\|\cdot\|_{\mathcal{S}_\theta}$. On suppose ici que $\mathcal{W}$ ne dépend pas de l'événement élémentaire θ.

Proposition I.3.1. *D'après Defant et Floret [23, Chapitre 1, section 7.2], si P est une mesure finie (e.g. une mesure de probabilité), on a la relation*

$$L^2(\Theta, dP; \mathcal{W}) = \mathcal{W} \otimes \mathcal{S}_\Theta. \tag{I.25}$$

Le problème (I.23) admet la formulation faible suivante :

$$\begin{aligned} &\text{Trouver } u \in \mathcal{W} \otimes \mathcal{S}_\Theta \text{ tel que} \\ &a_\Theta(\delta u, u) = \ell_\Theta(\delta u), \quad \forall \delta u \in \mathcal{W} \otimes \mathcal{S}_\Theta, \end{aligned} \tag{I.26}$$

avec

$$a_\Theta(\delta u, u) = \int_\Theta b_\Theta(\delta u(\theta), u(\theta); \theta)\ dP(\theta) \quad \text{et} \quad \ell_\Theta(\delta u) = \int_\Theta g_\Theta(\delta u(\theta); \theta)\ dP(\theta). \tag{I.27}$$

Remarque I.3.2. *Dans le cas où $\mathcal{W}$ n'est pas déterministe, $L^2(\Theta, dP; \mathcal{W})$ perd la propriété d'être un espace produit tensoriel.*

I.3.2 Modélisation des incertitudes

En pratique, on discrétise les incertitudes avec un nombre fini de variables aléatoires $\xi : \theta \in \Theta \to \xi(\theta) \in \mathbb{R}^d$. Il convient alors de définir un espace probabilisé $(\Xi, \mathcal{B}_\Xi, dP_\xi)$ tel que

$$\Xi = \xi(\Theta) \subset \mathbb{R}^d, \quad P_\xi = P \circ \xi^{-1}, \tag{I.28}$$

$\mathcal{B}_\Xi$ est la tribu des boréliens sur Ξ et ξ^{-1} une application donnant l'image réciproque d'un ensemble de Ξ par ξ. La solution u étant mesurable par rapport à ξ, d'après le lemme de Doob-Dynkin [85] elle peut s'écrire comme une fonction de ξ et le problème devient :

$$\begin{aligned} &\text{Trouver } u \in \mathcal{W} \otimes \mathcal{S} \text{ tel que} \\ &a(\delta u, u) = \ell(\delta u), \quad \forall \delta u \in \mathcal{W} \otimes \mathcal{S}, \end{aligned} \tag{I.29}$$

où $\mathcal{S}$ désigne l'espace $L^2(\Xi, dP_\xi)$ et

$$a(\delta u, u) = \int_\Xi b(\delta u(y), u(y); y)\ dP_\xi(y) \quad \text{et} \quad \ell(\delta u) = \int_\Xi g(\delta u(y); y)\ dP_\xi(y). \tag{I.30}$$

Remarque I.3.3. *Pour une analyse paramétrique sur un domaine Ξ borné, on choisira typiquement une mesure P_ξ uniforme.*

I.3.3 Discrétisation

Comme dans la section I.2, on introduit les espaces de dimension finie $\mathcal{W}_m$ et $\mathcal{S}_p$ tels que $\mathcal{W}_m \subset \mathcal{W}$ soit un espace de dimension $m \in \mathbb{N}^*$ avec comme base $(\varphi_i)_{1\leq i\leq m}$, et $\mathcal{S}_p \subset \mathcal{S}$ soit un espace de dimension finie $p \in \mathbb{N}^*$ avec comme base $(\psi_i)_{1\leq i\leq p}$. On arrive au problème approximé suivant :

$$\begin{aligned} &\text{Trouver } u \in \mathcal{W}_m \otimes \mathcal{S}_p \text{ tel que} \\ &a(\delta u, u) = \ell(\delta u), \quad \forall \delta u \in \mathcal{W}_m \otimes \mathcal{S}_p. \end{aligned} \tag{I.31}$$

La solution $u \in \mathcal{W}_m \otimes \mathcal{S}_p$ s'écrit

$$u = \sum_{i=1}^{m} \sum_{j=1}^{p} u_{ij} \varphi_i \otimes \psi_j, \tag{I.32}$$

où les coefficients $(u_{ij})_{1\leq i\leq m, 1\leq j\leq p}$ sont solutions du système linéaire

$$\sum_{k=1}^{m} \sum_{l=1}^{p} a_{ijkl} u_{kl} = \ell_{ij},\ \forall (i,j) \in \{1,\dots,m\} \times \{1,\dots,p\}, \tag{I.33}$$

avec

$$a_{ijkl} = a(\varphi_i \otimes \psi_j, \varphi_k \otimes \psi_l) = \int_\Xi b(\varphi_i \psi_j(y), \varphi_k \psi_l(y); y) dP_\xi(y), \tag{I.34}$$

$$\ell_{ij} = \ell(\varphi_i \otimes \psi_j) = \int_\Xi g(\varphi_i \psi_j(y); y) dP_\xi(y). \tag{I.35}$$

I.3.4 Choix des espaces d'approximation des variables aléatoires

Comme $\mathcal{S}$ est un Hilbert, on peut définir une base hilbertienne $(\psi_\alpha)_{\alpha\in\mathbb{N}^*}$ de $\mathcal{S} = L^2(\Xi, dP_\xi)$. Une variable aléatoire $v \in \mathcal{S}$ peut alors s'écrire

$$v = \sum_{\alpha\in\mathbb{N}^*} v_\alpha \psi_\alpha, \quad \text{avec} \quad v_\alpha = E(v\psi_\alpha),\ \forall \alpha \in \mathbb{N}^*. \tag{I.36}$$

Si par exemple les $(\xi_i)_{1\leq i\leq d}$ sont des variables aléatoires gaussiennes indépendantes, alors les polynômes multidimensionnels d'Hermite normalisés forment une base hilbertienne de $L^2(\Xi, dP_\xi)$. Cette représentation est alors connue sous le nom de *décomposition sur le chaos polynomial* [14, 82]. Dans le cas d'autres types de variables aléatoires, d'autres choix de bases conduisent à une *décomposition sur le chaos généralisé* [26, 71]. $\mathcal{S}_p$ peut alors être construit en sélectionnant judicieusement certains éléments de la base. L'espace d'approximation s'écrit alors

$$\mathcal{S}_p = \text{span}\left\{(\psi_\alpha)_{\alpha\in\mathcal{I}};\ \mathcal{I} \subset \mathbb{N}^*,\ \#\mathcal{I} = p\right\}. \tag{I.37}$$

I.3.5 Hypothèse d'indépendance des variables aléatoires

Avec $\xi = (\xi_i)_{1\leq i\leq d}$, dans le cas où les $(\xi_i)_{1\leq i\leq d}$ sont des variables aléatoires indépendantes, on peut écrire

$$L^2(\Xi, dP_\xi) = \bigotimes_\mu L^2(\Xi^\mu, dP_{\xi_\mu}). \tag{I.38}$$

Remarque I.3.4. *Dans le cas où les variables aléatoires ne sont plus indépendantes, $L^2(\Xi, dP_\xi)$ perd la propriété d'être un espace produit tensoriel. Cependant d'après [71], on peut définir une base hilbertienne $(\psi_\alpha)_{\alpha\in\mathcal{I}}$ de $L^2(\Xi, dP_\xi)$ de la manière suivante*

$$\psi_\alpha(y) = \prod_{\mu=1}^{d} \psi^\mu_{\alpha_\mu}(y_\mu)\sqrt{\frac{p_{\xi_\mu}(y_\mu)}{p_\xi(y)}}, \tag{I.39}$$

où p_ξ est la fonction de densité de probabilité de ξ, p_{ξ_μ} la fonction de densité de probabilité de ξ_μ et $\left(\psi^\mu_{\alpha_\mu}\right)_{\alpha_\mu\in\mathcal{I}^\mu}$ une base hilbertienne de $L^2(\Xi^\mu, dP_{\xi_\mu})$. En introduisant un sous-ensemble $\mathcal{I}_p = \mathcal{I}^1_{p_1} \times \ldots \times \mathcal{I}^d_{p_d} \subset \mathcal{I}$, on définit un espace d'approximation $\mathcal{S}_p = \text{span}\{\psi_\alpha; \alpha \in \mathcal{I}_p\}$ qui n'a pas de structure produit mais qui est cependant isomorphe à $\mathbb{R}^{p_1} \otimes \ldots \otimes \mathbb{R}^{p_d}$.

Avec

$$\mathcal{S}^\mu_{p_\mu} \subset L^2(\Xi_\mu, dP_{\xi_\mu}),\ \dim(\mathcal{S}^\mu_{p_\mu}) = p_\mu \quad \forall \mu \in \{1, \ldots, d\}, \tag{I.40}$$

et $\left(\psi^\mu_{\alpha_\mu}\right)_{1\leq\alpha_\mu\leq p_\mu}$ une base de $\mathcal{S}^\mu_{p_\mu}$, $\forall \mu \in \{1, \ldots, d\}$, on peut alors utiliser comme espace d'approximation

$$\mathcal{S}_p = \bigotimes_\mu \mathcal{S}^\mu_{p_\mu}. \tag{I.41}$$

Une base de $\mathcal{S}_p$ est alors construite par tensorisation de bases. Un élément $u \in \mathcal{W}_m \otimes \mathcal{S}_p$ se développe

donc sous la forme

$$u = \sum_{i=1}^{m} \sum_{\alpha_1=1}^{p_1} \dots \sum_{\alpha_d=1}^{p_d} u_{i\alpha_1 \dots \alpha_d} \varphi_i \otimes \left(\bigotimes_{\mu} \psi_{\alpha_\mu}^{\mu} \right). \tag{I.42}$$

I.3.6 Équation de la chaleur

On s'intéresse au problème de l'équation de la chaleur en régime stationnaire sur un domaine Ω ayant pour formulation forte :

$$\begin{aligned} \nabla(-\kappa \nabla u) &= f \quad \text{sur } \Omega \times \Xi, \\ u &= 0 \quad \text{sur } \partial\Omega \times \Xi, \end{aligned} \tag{I.43}$$

où κ est un champ aléatoire défini sur $\Omega \times \Xi$.

Ce problème admet comme formulation faible :

$$\begin{aligned} &\text{Trouver } u \in \mathcal{W} \otimes \mathcal{S} \text{ tel que} \\ &a(\delta u, u) = \ell(\delta u), \quad \forall \delta u \in \mathcal{W} \otimes \mathcal{S}, \end{aligned} \tag{I.44}$$

où $\mathcal{W} = H_0^1(\Omega)$ et

$$\begin{aligned} a(\delta u, u) &= \int_{\Xi} \int_{\Omega} \nabla \delta u(x,y) \cdot \kappa(x,y) \cdot \nabla u(x,y) \; dx \; dP_\xi(y), \\ \ell(\delta u) &= \int_{\Xi} \int_{\Omega} \delta u(x,y) f(x,y) \; dx \; dP_\xi(y). \end{aligned} \tag{I.45}$$

On suppose qu'il existe $(\kappa_{inf}, \kappa_{sup}) \in \mathbb{R}^2$ tel que

$$0 < \kappa_{inf} < \kappa(x,\xi) < \kappa_{sup} < +\infty, \quad \forall (x,\xi) \in \Omega \times \Xi. \tag{I.46}$$

Cette hypothèse garantit la continuité et la coercitivité de la forme bilinéaire a. On supposera aussi que $f \in H^{-1}(\Omega) \otimes \mathcal{S}$ afin d'assurer la continuité de la forme linéaire. Il en découle que le problème (I.44) satisfait les hypothèses du théorème de Lax-Milgram et que le problème possède une unique solution.

I.3.6.1 Représentation des champs aléatoires

On s'intéresse au champ aléatoire $\kappa : \Omega \times \Xi \to \mathbb{R}$ et à son écriture sous format séparé. Il existe plusieurs techniques afin d'exprimer κ sous la forme

$$\kappa = \sum_{i=1}^{\infty} \kappa_i^x \otimes \kappa_i^\xi. \tag{I.47}$$

Une première méthode consiste à décomposer κ sur la base du chaos polynomial généralisé comme à la section I.3.4. Il vient alors les relations

$$\kappa_i^\xi = \psi_i, \tag{I.48}$$

$$\kappa_i^x = E(\kappa\psi_i), \tag{I.49}$$

où les $(\psi_i)_{i\in\mathbb{N}^*}$ forment une base orthonormée de $L^2(\Xi)$ que l'on a ordonné. Une autre manière d'arriver à une telle écriture est d'utiliser la *décomposition de Karhunen-Loève* [44]. Si $\kappa \in L^2(\Omega) \otimes L^2(\Xi, dP_\xi)$, alors le champ stochastique peut s'écrire

$$\kappa(x,\xi) = \mu_\kappa(x) + \sum_{i=1}^{\infty} \sqrt{\sigma_i}\eta_i(x)\zeta_i(\xi), \tag{I.50}$$

avec μ_κ l'espérance de κ, $(\sigma_i)_{i\in\mathbb{N}^*}$ une famille de réels positifs, $(\eta_i)_{i\in\mathbb{N}^*}$ une base hilbertienne de $L^2(\Omega)$, fonctions propres de l'opérateur de corrélation de κ, et $(\zeta_i)_{i\in\mathbb{N}^*}$ une famille de $L^2(\Xi, dP_\xi)$ telle que les $(\zeta_i)_{i\in\mathbb{N}^*}$ soient toutes de moyenne nulle, orthogonales et décorrélées deux à deux.

I.3.6.2 Discrétisation du problème

En cherchant une approximation de Galerkin u sous la forme

$$u = \sum_{i=1}^{m}\sum_{j=1}^{p} u_{ij}(\varphi_i \otimes \psi_j) \in \mathcal{W}_m \otimes \mathcal{S}_p, \tag{I.51}$$

et en suivant la méthode décrite en section I.3.3, on arrive directement au système linéaire

$$\sum_{j=1}^{m}\sum_{\beta=1}^{p} E(\gamma_{ij}\psi_\alpha\psi_\beta)u_{j\beta} = E(\lambda_i\psi_\alpha), \quad \forall(i,\alpha) \in \{1,\dots,m\}\times\{1,\dots p\}, \tag{I.52}$$

où

$$\gamma_{ij} = \int_\Omega \nabla\varphi_i \cdot \kappa \cdot \nabla\varphi_j \; dx \quad \text{et} \quad \lambda_i = \int_\Omega \varphi_i f \; dx. \tag{I.53}$$

En notant κ et f sous la forme

$$\kappa = \sum_{k=1}^{\infty} \kappa_k^x \otimes \kappa_k^\xi \quad \text{et} \quad f = \sum_{k=1}^{\infty} f_k^x \otimes f_k^\xi, \tag{I.54}$$

et en supposant que l'on peut permuter les signes $\sum_{k=1}^{\infty}$ et $\int_\Xi \int_\Omega$, on a alors

$$E(\gamma_{ij}\psi_\alpha\psi_\beta) = \sum_{k=1}^{\infty} \left(\int_\Omega \nabla\varphi_i \cdot \kappa_k^x \cdot \nabla\varphi_j \; dx\right)\left(\int_\Xi \psi_\alpha \cdot \kappa_k^\xi \cdot \psi_\beta \; dy\right), \tag{I.55}$$

$$E(\lambda_i\psi_\alpha) = \sum_{k=1}^{\infty} \left(\int_\Omega \varphi_i f_k^x \; dx\right)\left(\int_\Xi \psi_\alpha f_k^\xi \; dy\right). \tag{I.56}$$

On peut définir l'opérateur $A \in \mathbb{R}^{m\times m} \otimes \mathbb{R}^{p\times p}$ et le tenseur $b \in \mathbb{R}^m \otimes \mathbb{R}^p$ par

$$A = \sum_{k=1}^{\infty} A_k^x \otimes A_k^\xi, \tag{I.57}$$

$$b = \sum_{k=1}^{\infty} b_k^x \otimes b_k^\xi, \tag{I.58}$$

avec

$$(A_k^x)_{ij} = \int_\Omega \nabla\varphi_i \cdot \kappa_k^x \cdot \nabla\varphi_j \; dx, \tag{I.59}$$

$$(A_k^\xi)_{\alpha\beta} = \int_\Xi \psi_\alpha \cdot \kappa_k^\xi \cdot \psi_\beta \; dy, \tag{I.60}$$

$$(b_k^x)_i = \int_\Omega \varphi_i f_k^x \; dx, \tag{I.61}$$

$$(b_k^\xi)_\alpha = \int_\Xi \psi_\alpha f_k^\xi \; dy. \tag{I.62}$$

$\mathbf{u} = (u_{ij})_{\substack{1\le i\le m\\ 1\le j\le p}}$ est finalement solution du système linéaire

$$A\mathbf{u} = b. \tag{I.63}$$

I.4 Vers l'utilisation de méthodes d'approximation de tenseurs

Avec l'équation (I.17), on observe que le nombre d'inconnues croît exponentiellement avec le nombre de dimensions (ordre des tenseurs). Typiquement, avec $n_\mu = n,\ \mu \in D$, l'équation (I.17) est un système d'équations à n^d inconnues. Pour briser cette *malédiction de la dimensionnalité*, on propose d'explorer dans cette thèse des méthodes d'approximation de tenseurs.

Au lieu de chercher u sous la forme

$$u = \sum_{i_1=1}^{n} \cdots \sum_{i_d=1}^{n} u_{i_1\ldots i_d} \bigotimes_\mu \varphi_{i_\mu}^\mu, \tag{I.64}$$

on cherche une approximation de faible rang de la solution par exemple sous la forme

$$u_r = \sum_{i=1}^{r} \bigotimes_\mu v_i^\mu, \quad v_i^\mu \in \mathcal{V}_n^\mu, \tag{I.65}$$

où

$$v_i^\mu = \sum_{j=1}^{n} v_{ij}^\mu \varphi_j^\mu. \tag{I.66}$$

u a n^d degrés de liberté alors que u_r en a au plus rdn. On espère alors qu'avec r négligeable devant $n^d/(nd)$ il soit possible d'obtenir une bonne approximation de la solution au sens d'une certaine norme, c'est-à-dire $\|u - u_r\| \le \varepsilon$ pour un $\varepsilon > 0$ donné.

Plus généralement, le principe des méthodes d'approximation de tenseurs consiste à introduire

un ensemble d'approximation $\mathcal{M} \subset \mathcal{V}$, typiquement

$$\mathcal{M} = \left\{ v = \sum_{i=1}^{r} \bigotimes_{\mu} v_i^{\mu};\ v_i^{\mu} \in \mathcal{V}^{\mu} \right\} \tag{I.67}$$

comme c'est le cas ci-dessus. Une approximation de la solution pourra alors être recherchée en résolvant un problème d'optimisation : trouver w tel que

$$\|u - w\| \leq (1+\varepsilon) \inf_{v \in \mathcal{M}} \|u - v\|, \quad \varepsilon > 0. \tag{I.68}$$

On verra dans les chapitres suivants de quelle manière définir de bons sous-ensembles $\mathcal{M}$ pour garantir une bonne qualité d'approximation et de bonnes propriétés du problème d'approximation. On verra aussi comment procéder pour approcher le champ solution u d'un problème sans connaître u a priori : soit en introduisant des méthodes itératives, soit en modifiant la définition de l'approximation pour la rendre calculable a priori. Typiquement, on pourra introduire un problème de la forme : trouver w tel que

$$\|Aw - b\|_{\mathcal{V}} \leq (1+\varepsilon) \inf_{v \in \mathcal{M}} \|Av - b\|_{\mathcal{V}}, \quad \varepsilon > 0, \tag{I.69}$$

comme en (I.21) où on cherche à minimiser une certaine norme du résidu.

Formats de tenseurs et approximations

Sommaire

Cette partie a pour objectif approfondir les notions liées aux tenseurs en omettant leurs relations avec les problèmes variationnels. Elle va permettre de détailler les différents formats de tenseurs et les méthodes d'approximation existants. Les ouvrages de référence ayant servis à la rédaction de ce chapitre sont principalement le livre de W. Hackbusch [38] ainsi que la revue de T.G. Kolda et B.W. Bader [47], la plupart des sujets y sont traités.

Toutes ces méthodes seront par la suite introduites pour la résolution des systèmes d'équations linéaires aux chapitres III et IV.

Tous les espaces considérés dans ce chapitre seront de dimension finie.

II.1 Rang, t-rang et rang de Tucker

Soit $\mathcal{V} = \bigotimes_{\mu\in D} \mathcal{V}^\mu$ un espace de tenseur avec $D \in \{1,\dots,d\}$. Pour tout $\mu \in D$, $\mathcal{V}^\mu$ est un espace de Hilbert de dimension finie n_μ muni du produit scalaire $\langle\cdot,\cdot\rangle_\mu$ et de la norme associée $\|\cdot\|_\mu$. $\mathcal{V}$ est alors un espace de Hilbert muni du produit scalaire canonique $\langle\cdot,\cdot\rangle$ défini sur les tenseurs élémentaires par

$$\left\langle \bigotimes_\mu v^\mu, \bigotimes_\mu w^\mu \right\rangle = \prod_{\mu\in D} \langle v^\mu, w^\mu\rangle_\mu \tag{II.1}$$

et étendu par linéarité. On note $\|\cdot\|$ la norme associée.

D'après la définition donnée en section I.1, il existe une écriture de $v \in \mathcal{V}$ sous la forme

$$v = \sum_{i=1}^{r_v} \alpha_i \bigotimes_\mu v_i^\mu, \quad \alpha_i \in \mathbb{R},\ v_i^\mu \in \mathcal{V}^\mu. \tag{II.2}$$

Le plus petit $r_v \in \mathbb{N}^*$ permettant d'écrire v sous cette forme définit le *rang* de v.

Soit $t \subset D$. On note t^c le complémentaire de t dans D, c'est-à-dire $t^c = D \backslash t$. Soient $\mathcal{V}^t = \bigotimes_{\mu \in t} \mathcal{V}^\mu$ et $\mathcal{V}^{t^c} = \bigotimes_{\mu \in t^c} \mathcal{V}^\mu$. On peut alors définir une application $\mathcal{M}_{(t)} : \mathcal{V} \to \mathcal{V}^t \otimes \mathcal{V}^{t^c}$ par

$$\mathcal{M}_{(t)}(v) = v^{(t)} \quad \text{tel que} \quad v^{(t)} = \sum_{i=1}^{r_v} \alpha_i \left(\bigotimes_{\mu \in t} v_i^\mu \right) \otimes \left(\bigotimes_{\mu \in t^c} v_i^\mu \right), \tag{II.3}$$

$v^{(t)}$ est appelé la *t-matricisation* de v. Le *t-rang* de v est finalement défini par le rang de $v^{(t)}$ [34, 37]. On définit le *rang de Tucker* de v comme le d-uplet $r = (r_1, \ldots, r_d)$ où r_μ est le rang de $v^{(\{\mu\})}$, $\mu \in D$ [37].

II.2 Sous-ensembles particuliers

II.2.1 Tenseurs canoniques

L'ensemble des tenseurs canoniques de rang r, introduit par Hitchcock [40], est défini par

$$\mathcal{C}_r(\mathcal{V}) = \left\{ v = \sum_{i=1}^{r} \alpha_i \bigotimes_{\mu} v_i^\mu ;\ \alpha_i \in \mathbb{R},\ v_i^\mu \in \mathcal{V}^\mu \right\}. \tag{II.4}$$

$\mathcal{C}_1(\mathcal{V})$ est l'ensemble des tenseurs élémentaires et on a la relation $\mathcal{V} = \operatorname{span}\{\mathcal{C}_1(\mathcal{V})\}$. Pour $r = 1$ ou $d = 2$, $\mathcal{C}_r(\mathcal{V})$ est fermé, caractère important pour la formulation des problèmes d'optimisation sur cet ensemble. Sinon, si $r > 1$ et $d > 2$,$\mathcal{C}_r(\mathcal{V})$ n'est pas fermé, les problèmes d'optimisation posés sur $\mathcal{C}_r(\mathcal{V})$ ne sont pas garantis d'avoir une solution. Un autre désavantage du format concerne le phénomène d'« annulation ». Pour l'illustrer on remarque que la fonction

$$J_v : \begin{array}{l} \mathbb{R} \to \mathcal{V} \\ \alpha \mapsto (\alpha v^1) \otimes v^2 \otimes \ldots \otimes v^d - v^1 \otimes (\alpha v^2) \otimes v^3 \otimes \ldots \otimes v^d \end{array} < \tag{II.5}$$

avec $v = \bigotimes_\mu v^\mu$, est nulle. Le phénomène d'annulation intervient quand $\alpha \to \infty$. On se retrouve avec une somme finie de 2 tenseurs élémentaires de norme tendant vers l'infinie. Les algorithmes d'optimisation posés sur $\mathcal{C}_r(\mathcal{V})$ peuvent alors présenter un mauvais comportement lors du passage au numérique. Les problèmes liés à ce format sont traités plus en détails par De Silva et Lim [70].

II.2.2 Tenseurs de Tucker

Une extension naturelle aux tenseurs canoniques a été introduite par Tucker [77]. L'ensemble des tenseurs de Tucker est défini par

$$\mathcal{T}_r(\mathcal{V}) = \{ v \in \mathcal{V};\ \{\mu\} - rang(v) \leq r_\mu,\ \mu \in \{1, \ldots, d\} \}. \tag{II.6}$$

Un élément $v \in \mathcal{T}_r(\mathcal{V})$ peut s'écrire sous la forme

$$v = \sum_{i_1=1}^{r_1} \dots \sum_{i_d=1}^{r_d} \alpha_{i_1 \dots i_d} \bigotimes_{\mu} w_{i_\mu}^{\mu}, \quad \alpha_{i_1 \dots i_d} \in \mathbb{R}, \tag{II.7}$$

où les $(w_i^\mu)_{1 \leq i \leq r_\mu}$ sont une famille de r_μ vecteurs de $\mathcal{V}^\mu$. On remarque qu'on peut trouver une représentation avec des familles libres. Contrairement aux ensembles de tenseurs canoniques, $\mathcal{T}_r(\mathcal{V})$ est fermé quelque soit d et $r = (r_1, \dots, r_d)$ (Falcó et Hackbusch [28]) ce qui le rend approprié pour des problèmes d'optimisation. Malheureusement, ce type de format souffre aussi de la malédiction de la dimensionnalité puisqu'il requiert la manipulation du tenseur coeur $(\alpha_{i_1 \dots i_d})_{1 \leq i_\mu \leq r_\mu} \in \mathbb{R}^{r_1 \times \dots \times r_d}$.

II.2.3 Tenseurs hiérarchiques de Tucker

Hackbusch et Kühn [37] ont introduit le format hiérarchique de Tucker afin de pallier aux problèmes des deux précédents, à savoir les mauvaises propriétés des tenseurs canoniques, et le problème de dimensionnalité des tenseurs de Tucker.

II.2.3.1 L'arbre des dimensions

Soit $D = \{1, \dots, d\}$ et T un arbre binaire entier sur D. On note $L(T)$ les feuilles de T et $I(T) = T \setminus L(T)$. Les enfants $S(t) = \{t_1, t_2\}$ d'un noeud $t \in I(T)$ sont définis tels que $t = t_1 \cup t_2$ et $t_1 \cap t_2 = \emptyset$. Un élément $t \in L(T)$ est défini tel que $\#t = 1$. Un exemple d'arbre des dimensions est illustré en figure II.1.

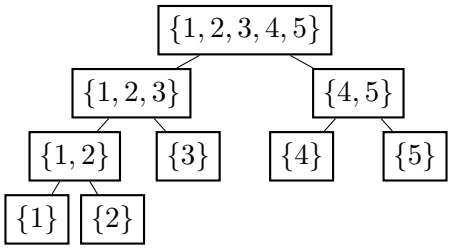


FIGURE II.1 – Un arbre des dimensions pour $d = 5$

Remarque II.2.1. *On considère uniquement des arbres binaires imbriqués. Il est possible de généraliser à d'autres types d'arbres, comme les « tensor chains » [45], mais cela sort du cadre de cette thèse.*

II.2.3.2 Les tenseurs hiérarchiques de Tucker

Pour un arbre donné T, et un ensemble de rangs $r = (r_t)_{t \in T}$ (un rang donné à chaque noeud), l'ensemble des tenseurs hiérarchiques de Tucker de rang r est donné par

$$\mathcal{H}_r^T(\mathcal{V}) = \{v \in \mathcal{V};\ t - rang(v) \leq r_t,\ \forall t \in T\}. \tag{II.8}$$

Derrière cette représentation abstraite, on peut écrire plus concrètement un élément $v \in \mathcal{H}_r^T(\mathcal{V})$ sous la forme

$$v = \sum_{i=1}^{r_D} \sum_{j=1}^{r_D} \alpha_{ij}^D v_i^{(D_1)} \otimes v_j^{(D_2)}, \quad D = D_1 \cup D_2,\ D_1 \cap D_2 = \emptyset,\ \alpha_{ij}^D \in \mathbb{R}, \tag{II.9}$$

$$v_k^t = \sum_{i=1}^{r_t} \sum_{j=1}^{r_t} \alpha_{ijk}^t v_i^{t_1} \otimes v_j^{t_2}, \quad t = t_1 \cup t_2,\ t_1 \cap t_2 = \emptyset,\ \alpha_{ijk}^t \in \mathbb{R}, \quad \forall t \in I(T) \setminus D. \tag{II.10}$$

Un tenseur $v \in \mathcal{H}_r^T(\mathcal{V})$ est donc entièrement déterminé par la donnée des tenseurs de transferts $\left(\alpha^t\right)_{t \in I(T)}$ et des vecteurs $\left(v_i^t\right)_{t \in L(T),\ i \in \{1,\dots,r_t\}}$ avec $L(T) = \{\{\mu\};\ \mu \in D\}$. Sans perte de généralité, on peut supposer que pour $t \in L(T)$, la famille $\left(v_i^t\right)_{1 \leq i \leq r_t}$ est orthonormale. De plus $\#I(T) = d - 1$ et $\#L(T) = d$ et on retrouve alors une dépendance linéaire avec la dimension de la représentation du tenseur comme pour les tenseurs canoniques. Aussi, d'après Hackbusch [38], l'ensemble $\mathcal{H}_r^T(\mathcal{V})$ est fermé.

Remarque II.2.2. *Dans le cas $d = 2$, on a l'égalité $\mathcal{C}_r(\mathcal{V}) = \mathcal{T}_{(r,r)}(\mathcal{V}) = \mathcal{H}_{(r,r,r)}^T(\mathcal{V}),\ \forall r \in \mathbb{N}^*$. De plus, on peut généraliser l'approche à des arbres non binaires pourvu qu'on ait $L(T) = \{\{\mu\};\ \mu \in D\}$. Dans ce cas, si $I(T) = \{D\}$, $v \in \mathcal{H}_r^T(\mathcal{V})$ est un tenseur de Tucker.*

II.2.4 Paramétrage des formats de tenseurs

On peut interpréter les formats de tenseurs classiques comme des applications multilinéaires $F_{\mathcal{M}} : \mathcal{F}_{\mathcal{M}}^1 \times \ldots \times \mathcal{F}_{\mathcal{M}}^m \to \mathcal{V}$ telles que $\mathcal{M} = \mathrm{Im}(F_{\mathcal{M}})$ où les $\left(\mathcal{F}_{\mathcal{M}}^i\right)_{1 \leq i \leq m}$ sont des espaces vectoriels normés. Ainsi pour $v \in \mathcal{M}$, il existe $(f^1, \ldots, f^m) \in \mathcal{F}_{\mathcal{M}}^1 \times \ldots \times \mathcal{F}_{\mathcal{M}}^m$ tel que

$$v = F_{\mathcal{M}}(f^1, \ldots, f^m). \tag{II.11}$$

Exemple II.2.3 (Paramétrage de $\mathcal{C}_r(\mathcal{V})$)**.** *Pour $v = \sum_{i=1}^r \bigotimes_\mu v_i^\mu, v_i^\mu \in \mathcal{V}^\mu = \mathbb{R}^{n_\mu}$, on peut associer à f^μ une matrice de $\mathcal{F}_{\mathcal{M}}^\mu = (\mathcal{V}^\mu)^r = \mathbb{R}^{n_\mu \times r}$ où la i-ème colonne de f^μ correspond à v_i^μ. On précisera les paramétrages pour les tenseurs de Tucker et hiérarchiques de Tucker en section II.3.2.*

On peut définir l'espace tangent à $\mathcal{M}$ en v à partir de $F_{\mathcal{M}}$:

Définition II.2.4 (Espace tangent)**.** *Soit $\mathcal{M} \subset \mathcal{V}$ une variété différentielle. On suppose qu'il existe un paramétrage de $\mathcal{M}$ par une application multilinéaire et continue $F_{\mathcal{M}} : \mathcal{F}_{\mathcal{M}}^1 \times \ldots \times \mathcal{F}_{\mathcal{M}}^m \to \mathcal{V}$. L'espace tangent $T_v(\mathcal{M})$ à $\mathcal{M}$ en v est alors défini par*

$$T_v(\mathcal{M}) = \left\{ D_f F_{\mathcal{M}}(h);\ h \in \mathcal{F}_{\mathcal{M}}^1 \times \ldots \times \mathcal{F}_{\mathcal{M}}^m \right\}, \tag{II.12}$$

où $D_f F_{\mathcal{M}}$ est la différentielle de $F_{\mathcal{M}}$ en $f = (f^1, \ldots, f^m)$.

Remarque II.2.5. *$F_{\mathcal{M}}$ multilinéaire et continue implique que $F_{\mathcal{M}}$ est Fréchet différentiable.*

A l'aide de la définition II.2.4, on peut montrer le résultat suivant qui sera utile plus tard en section III.4.2.

Lemme II.2.6. *Soit $\mathcal{U}$ un sous espace vectoriel de $\mathcal{V}$ de dimension finie* $\dim(\mathcal{U}) = p$ *et $\mathcal{M}$ est une variété différentielle telle que $\mathcal{M} \subset \mathcal{U} \subset \mathcal{V}$. On suppose que $\mathcal{M}$ est paramétré par une application multilinéaire continue. Alors on a l'inclusion*

$$T_v(\mathcal{M}) \subset \mathcal{U}, \quad \forall v \in \mathcal{M}. \tag{II.13}$$

Démonstration. Soit $(e_i)_{1\leq i\leq p}$ une base de $\mathcal{U}$. Comme $\mathrm{Im}(F_{\mathcal{M}}) \subset \mathcal{U}$ on peut décomposer $F_{\mathcal{M}}$ sur cette base comme

$$F_{\mathcal{M}}(f^1, \ldots, f^m) = \sum_{i=1}^{p} F^i_{\mathcal{M}}(f^1, \ldots, f^m) e_i, \tag{II.14}$$

où les $\left(F^i_{\mathcal{M}}\right)_{1\leq i\leq p}$ sont des fonctions à valeurs réelles de $\mathbb{R}^{(\mathcal{F}^1_{\mathcal{M}} \times \ldots \times \mathcal{F}^m_{\mathcal{M}})}$. Comme $F_{\mathcal{M}}$ est différentiable, les $\left(F^i_{\mathcal{M}}\right)_{1\leq i\leq p}$ le sont aussi et on a l'égalité

$$D_f F_{\mathcal{M}}(h) = \sum_{i=1}^{p} D_f F^i_{\mathcal{M}}(h) e_i \in \mathcal{U}, \quad \forall h \in \mathcal{F}^1_{\mathcal{M}} \times \ldots \times \mathcal{F}^m_{\mathcal{M}} \tag{II.15}$$

ce qui termine la démonstration. □

II.3 Approximation de tenseurs

Les méthodes d'approximation de tenseurs sont utiles pour deux objectifs, à savoir l'interprétation de données (« analyse en composantes principales » [66]) et leur compression. L'idée est de trouver l'approximation d'un tenseur $u \in \mathcal{V}$ dans un sous-ensemble $\mathcal{M}$ (qui peut-être $\mathcal{C}_r(\mathcal{V})$, $\mathcal{T}_r(\mathcal{V})$, $\mathcal{H}^T_r(\mathcal{V})$). Pour cela on formule le problème de minimisation que l'on doit résoudre :

$$\begin{aligned} &\text{Trouver } w \in \mathcal{M} \text{ tel que} \\ &\|u - w\| = \min_{v\in\mathcal{M}} \|u - v\| . \end{aligned} \tag{II.16}$$

Si l'ensemble $\mathcal{M}$ n'est pas fermé on écrira le problème comme

$$\begin{aligned} &\text{Trouver } w \in \mathcal{M} \text{ tel que} \\ &\|u - w\| \leq \inf_{v\in\mathcal{M}} \|u - v\| + \varepsilon, \end{aligned} \tag{II.17}$$

avec $\varepsilon > 0$. On notera $\Pi_{\mathcal{M}}(u)$ l'ensemble des solutions de (II.16) et $\Pi^{\varepsilon}_{\mathcal{M}}(u)$ celui de (II.17). ε sera omis s'il n'y a pas d'ambiguïté. $\Pi_{\mathcal{M}}(u)$ désignera la plupart du temps un ensemble d'approximations acceptables de u sans nécessairement être des minima globaux. En effet les algorithmes utilisés en sections II.3.2 et II.3.3 convergent vers des points stationnaires et non vers des minima globaux.

II.3.1 Décompositions en valeurs singulières

Les différentes décompositions en valeurs singulières (Singular Value Decomposition, SVD, [25]) permettent d'éviter de passer par des algorithmes d'optimisation pour fournir une approximation d'un tenseur. On verra tout d'abord le cas où $d = 2$, puis $d > 2$.

II.3.1.1 Décomposition en valeurs singulières ($d = 2$)

On suppose que $d = 2$, ainsi $\mathcal{V} = \mathcal{V}^1 \otimes \mathcal{V}^2$ et $\mathcal{C}_r(\mathcal{V}) = \mathcal{T}_{(r,r)}(\mathcal{V}) = \mathcal{H}^T_{(r,r,r)}(\mathcal{V}),\ \forall \in \mathbb{N}^*$. On note $n = \min(n_1, n_2)$ avec $\dim(\mathcal{V}^\mu) = n_\mu$.

Théorème II.3.1. *Soit $u \in \mathcal{V}$, alors il existe $(\sigma_i)_{1\leq i\leq n} \in (\mathbb{R}^+)^n$, $\left(v_i^1\right)_{1\leq i\leq n} \in (\mathcal{V}^1)^n$ et $\left(v_i^2\right)_{1\leq i\leq n} \in (\mathcal{V}^2)^n$ tels que*

$$u = \sum_{i=1}^{n} \sigma_i v_i^1 \otimes v_i^2 \tag{II.18}$$

et que $\left(v_i^1\right)_{1\leq i\leq n} \in (\mathcal{V}^1)^n$ et $\left(v_i^2\right)_{1\leq i\leq n} \in (\mathcal{V}^2)^n$ soient des familles orthonormales.

Les $(\sigma_i)_{1\leq i\leq n} \in (\mathbb{R}^+)^n$ sont appelées les valeurs singulières de u, $\left(v_i^1\right)_{1\leq i\leq n} \in (\mathcal{V}^1)^n$ les vecteurs singuliers à gauche de u et $\left(v_i^2\right)_{1\leq i\leq n} \in (\mathcal{V}^2)^n$ les vecteurs singuliers à droite. On suppose que les valeurs singulières sont rangées par ordre décroissant, c'est-à-dire

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0. \tag{II.19}$$

Soit $u = \sum_{i=1}^{r_u} u_i^1 \otimes u_i^2$. Les valeurs et vecteurs singuliers sont liés par les relations

$$\begin{aligned} \left\langle u, v_j^1 \right\rangle_1 &= \sum_{i=1}^{r_u} \left\langle u_i^1, v_j^1 \right\rangle_1 u_i^2 = \sigma_j v_j^2, \\ \left\langle u, v_j^2 \right\rangle_2 &= \sum_{i=1}^{r_u} \left\langle u_i^2, v_j^2 \right\rangle_2 u_i^1 = \sigma_j v_j^1. \end{aligned} \tag{II.20}$$

Avec ces notations, le théorème d'Eckart-Young s'écrit de la manière suivante :

Théorème II.3.2. *$\mathcal{V}$ étant muni de la norme canonique, un tenseur $v \in \mathcal{V}$ satisfaisant $v \in \Pi_{\mathcal{C}_r(\mathcal{V})}(u)$ est défini par*

$$v = \sum_{i=1}^{r} \sigma_i v_i^1 \otimes v_i^2. \tag{II.21}$$

Le problème (II.16) est alors immédiatement résolu dans le cas $d = 2$.

Remarque II.3.3. *Il ne faut pas confondre le produit scalaire $\langle \cdot, \cdot \rangle_\mu$ sur $\mathcal{V}^\mu$ et l'application décrite en (II.20) $\langle \cdot, \cdot \rangle_\lambda : \mathcal{V} \times \mathcal{V}^\lambda \to \bigotimes_{\mu\neq\lambda} \mathcal{V}^\mu$. C'est une particularisation du produit λ-mode $\cdot \times_\lambda \cdot$ utilisé dans [21, 47] dans le cas où $\mathcal{V}^\mu = \mathbb{R}^{n_\mu},\ \forall \mu \in D$.*

En introduisant les opérateurs linéaires $M^\mu : \mathcal{V}^\mu \to \mathcal{V}^\mu$ définis par

$$\begin{aligned} M^1(v_j^1) &= \left\langle u, \left\langle u, v_j^1 \right\rangle_1 \right\rangle_2 = \sigma_j^2 v_j^1, \\ M^2(v_j^2) &= \left\langle u, \left\langle u, v_j^2 \right\rangle_2 \right\rangle_1 = \sigma_j^2 v_j^2, \end{aligned} \tag{II.22}$$

on se rend compte que les vecteurs singuliers et le carré des valeurs singulières de u sont solutions des problèmes aux valeurs propres sur les opérateurs M^1 et M^2. Un algorithme classique de recherche de valeurs propres et vecteurs propres est la méthode de la puissance itérée avec une méthode de déflation. Elle est décrite dans l'algorithme 1.

Algorithme 1: Puissance itérée avec déflation appliquée à l'opérateur M^μ

Données : $u \in \mathcal{V}^1 \otimes \mathcal{V}^2$, $r \in \mathbb{N}^*$
Résultat : $(\sigma_i)_{1\leq i\leq r}$, $(v_i^\mu)_{1\leq i\leq r}$
Définir $\sigma_0 = 0$, $v_0^\mu = 0$;
pour $i = 1, \dots, r$ **faire**
  $M_i^\mu(\cdot) = M^\mu(\cdot) - \sum_{j=0}^{i-1} \sigma_j^2 \left\langle v_j^\mu, \cdot \right\rangle_\mu$;
  Initialiser aléatoirement v_i^μ;
  tant que σ_i *et* v_i^μ *n'ont pas convergé* **faire**
    $v_i^\mu = M_i^\mu(v_i^\mu)$;
    $\sigma_i = \sqrt{\|v_i^\mu\|_\mu}$;
    $v_i^\mu = v_i^\mu / \|v_i^\mu\|_\mu$;
  fin
fin

Exemple II.3.4. *Avec* $\mathcal{V}^1 \otimes \mathcal{V}^2 = \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \simeq \mathbb{R}^{n_1 \times n_2}$, *$u$ est identifié à une matrice et v_j^μ est un vecteur de $\mathbb{R}^{n_\mu}$. Les équations* (II.20) *se traduisent alors par*

$$\begin{aligned} \left\langle u, v_j^1 \right\rangle_1 &= u^T v_j^1 = \sigma_j v_j^2, \\ \left\langle u, v_j^2 \right\rangle_2 &= u v_j^2 = \sigma_j v_j^1, \end{aligned} \tag{II.23}$$

et les opérateurs sont égaux à $M_1 = uu^T$ et $M_2 = u^T u$.

II.3.1.2 Décompositions pour $d > 2$

Dans ce cas le théorème d'Eckart-Young n'est plus directement généralisable et d'autres approches doivent être employées. Dans la section II.3.2 on verra d'autres façons de construire une approximations dans $\mathcal{C}_r(\mathcal{V})$. En revanche, la SVD est généralisable pour les tenseurs de Tucker et hiérarchiques de Tucker d'ordre supérieur à 2.

La SVD de plus haut ordre (Higher Order SVD, HOSVD) Cette méthode trouve une approximation d'un tenseur $v \in \mathcal{V}$ dans le sous-ensemble $\mathcal{T}_r(\mathcal{V})$. Elle a été proposée originellement par Tucker [77] puis revisitée par De Lathauwer et al. [22] qui lui donnèrent ce nom. La HOSVD consiste à prendre pour $\left(v_i^\lambda\right)_{1\leq i\leq r_\lambda}$ les r_λ vecteurs singuliers à gauche de la matricisation $u^{(\{\lambda\})}$ définie en section II.1. Ces vecteurs sont orthonormés par définition. De plus, en cherchant une approximation

$$v = \sum_{i_1=1}^{r_1} \dots \sum_{i_d=1}^{r_d} \alpha_{i_1 \dots i_d} \bigotimes_\mu v_{i_\mu}^\mu, \tag{II.24}$$

on peut montrer que le problème de minimisation (II.16) avec

$$v \in \mathcal{M} = \operatorname{span}\left\{ \left(\bigotimes_\mu v_{i_\mu}^\mu \right) ;\ 1 \leq i_\mu \leq r_\mu,\ \forall \mu \in D \right\} \tag{II.25}$$

définit le tenseur coeur par

$$\alpha_{i_1 \dots i_d} = \left\langle u, \bigotimes_{\mu} v_{i_\mu}^{\mu} \right\rangle . \tag{II.26}$$

Ceci amène directement à l'algorithme 2.

Algorithme 2: HOSVD

Données : $u \in \mathcal{V}$, un multi-rang $r = (r_\mu)_{\mu \in D}$
Résultat : $v \in \mathcal{T}_r(\mathcal{V})$
pour $\mu = 1, \dots, d$ **faire**
  Calculer $(v_i^\mu)_{1 \leq i \leq r_\mu}$ comme les r_μ vecteurs singuliers à gauche de $u^{\{\mu\}}$;
fin
Calculer α tel que $\alpha_{i_1 \dots i_d} = \left\langle u, \bigotimes_\mu v_{i_\mu}^\mu \right\rangle$;

Contrairement à la SVD, la HOSVD ne résout pas le problème (II.16). Par contre, d'après Grasedyck [34] si v est une approximation de u donnée via la HOSVD, alors

$$\|u - v\| \leq \sqrt{d} \, \|u - w\| , \tag{II.27}$$

où $w \in \Pi_{\mathcal{T}_r(\mathcal{V})}(u)$. En ce sens, la HOSVD fournit une approximation $v \in \Pi^{\varepsilon}_{\mathcal{T}_r(\mathcal{V})}(u)$ quasi-optimale, avec

$$\varepsilon = (\sqrt{d} - 1) \, \|u - w\| . \tag{II.28}$$

La SVD hiérarchique (Hierarchical SVD, HSVD) La HSVD a été introduite par Grasedyck [34] pour les tenseurs hiérarchiques de Tucker. La méthode peut être vue comme une généralisation de la HOSVD à l'ensemble $\mathcal{H}_r^T(\mathcal{V})$. En considérant la t-matricisation $u^{(t)}$ d'un tenseur $u \in \mathcal{V}$, avec les notations proposées en (II.9), une approximation v de u est construite en prenant pour $\left(v_i^t\right)_{1 \leq i \leq r_t}$ les r_t vecteurs singuliers à gauche de $u^{(t)}$.

Il existe plusieurs algorithmes pour réaliser la HSVD, plus ou moins efficaces selon le sens de parcours de l'arbre T. Le plus simple, et le plus long, est présenté dans l'algorithme 3. Si le tenseur de départ u est déjà dans un format hiérarchique, la HSVD peut être appliquée avec un faible nombre d'opérations [34] .

Encore une fois la HSVD n'est pas optimale. Toujours d'après Grasedyck [34], si v est donnée via la HSVD, alors on a

$$\|u - v\| \leq \sqrt{2d - 3} \, \|u - w\| , \tag{II.29}$$

où $w \in \Pi_{\mathcal{H}_r^T(\mathcal{V})}(u)$. $v \in \Pi^{\varepsilon}_{\mathcal{H}_r^T(\mathcal{V})}(u)$ est une approximation quasi-optimale, avec

$$\varepsilon = (\sqrt{2d - 3} - 1) \, \|u - w\| . \tag{II.30}$$

Algorithme 3: HSVD de la racine aux feuilles

Données : $u \in \mathcal{V}$, un arbre des dimensions T, des t-rangs $r = (r_t)_{t \in T}$
Résultat : $v \in \mathcal{H}_r^T(\mathcal{V})$
pour $t \in L(T)$ **faire**
  Calculer $(v_i^t)_{1 \leq i \leq r_t}$ comme les r_t vecteurs singuliers à gauche de $u^{(t)}$;
fin
pour $t \in I(T) \setminus \{D\}$ *tel que* $\#t$ *croît* **faire**
  Calculer $(v_i^t)_{1 \leq i \leq r_t}$ comme les r_t vecteurs singuliers à gauche de $u^{(t)}$;
  Avec $t = t_1 \cup t_2,\ t_1 \cap t_2 = \emptyset$, calculer α^t tel que $\alpha^t_{ijk} = \left\langle v_k^t, v_i^{t_1} \otimes v_j^{t_2} \right\rangle$;
fin
Avec $D = t_1 \cup t_2,\ t_1 \cap t_2 = \emptyset$, calculer α^D tel que $\alpha^D_{ij} = \left\langle u, v_i^{t_1} \otimes v_j^{t_2} \right\rangle$;

II.3.2 Algorithmes de minimisations alternées pour résoudre le problème d'approximation de tenseurs

II.3.2.1 Cadre abstrait

Avec les paramétrages introduit en section II.2.4, le problème de minimisation initiale se ramène à

$$\begin{aligned} &\text{Trouver } (f^1, \ldots, f^m) \in \mathcal{F}^1_{\mathcal{M}} \times \ldots \times \mathcal{F}^m_{\mathcal{M}} \text{ tel que} \\ &\left\| u - F_{\mathcal{M}}(f^1, \ldots, f^m) \right\| = \min_{(g^1, \ldots, g^m) \in (\mathcal{F}^1_{\mathcal{M}} \times \ldots \times \mathcal{F}^m_{\mathcal{M}})} \left\| u - F_{\mathcal{M}}(g^1, \ldots, g^m) \right\|. \end{aligned} \tag{II.31}$$

L'Algorithme de Minimisations Alternées (AMA) consiste à minimiser sur chaque variable $f^i \in \mathcal{F}^i_{\mathcal{M}}$ successivement, ce qui donne l'algorithme 4. L'algorithme peut être initialisé par une bonne approximation de départ fournie par la HOSVD ou la HSVD pour $d > 2$. Si $d = 2$ la SVD résout directement le problème de minimisation.

Algorithme 4: Minimisations alternées dans $\mathcal{F}^1_{\mathcal{M}} \times \ldots \times \mathcal{F}^m_{\mathcal{M}}$

Données : $u \in \mathcal{V}$
Résultat : $v \in \mathcal{M}$
Initialiser $(f^1, \ldots, f^m)$ aléatoirement;
$v = F_{\mathcal{M}}(f^1, \ldots, f^m)$;
tant que v *n'a pas convergé* **faire**
  pour $\lambda = 1, \ldots, m$ **faire**
    Calculer $f^\lambda = \arg\min_{h \in \mathcal{F}^\lambda_{\mathcal{M}}} \left\| u - F_{\mathcal{M}}(f^1, \ldots, f^{\lambda-1}, h, f^{\lambda+1}, \ldots, f^m) \right\|$;
    $v = F_{\mathcal{M}}(f^1, \ldots, f^m)$;
  fin
fin
retourner v;

II.3.2.2 Minimisations alternées dans $\mathcal{C}_r(\mathcal{V})$

Pour l'ensemble des tenseurs canoniques, la première question à se poser est de savoir s'il existe un minimiseur. Malheureusement ce n'est pas garanti si $d \geq 3$ et $r \geq 2$, car alors $\mathcal{C}_r(\mathcal{V})$ n'est pas fermé

[70]. Malgré ce problème théorique, cet algorithme a fait ses preuves dans de nombreux exemples, en régularisant ou non le problème de minimisation [47].

On a l'égalité $m = d$ et $\mathcal{F}^\mu_\mathcal{M} = (\mathcal{V}^\mu)^r$. Ainsi $f^\mu \in \mathcal{F}^\mu_\mathcal{M}$ est défini par

$$f^\mu = (v^\mu_1, v^\mu_2, \dots, v^\mu_r)\,, \tag{II.32}$$

et $v = F_\mathcal{M}(f^1, \dots, f^d) = \sum_{i=1}^r \bigotimes_\mu v^\mu_i$. C'est le paramétrage utilisé dans l'exemple II.2.3.

II.3.2.3 Minimisations alternées dans $\mathcal{T}_r(\mathcal{V})$

Pour l'ensemble des tenseurs de Tucker $\mathcal{T}_r(\mathcal{V})$, $r = (r_1, \dots, r_d)$, on va utiliser un paramétrage avec $m = d + 1$ tel que

$$\begin{aligned} &\mathcal{F}^\mu_\mathcal{M} = (\mathcal{V}^\mu)^{r_\mu} \quad \text{pour} \quad \mu \in \{1, \dots, d\}, \\ &\mathcal{F}^{d+1}_\mathcal{M} = \mathbb{R}^{r_1 \times \dots \times r_d}. \end{aligned} \tag{II.33}$$

$f^\mu \in \mathcal{F}^\mu_\mathcal{M}$ est défini par

$$\begin{aligned} &f^\mu = \left(v^\mu_1, v^\mu_2, \dots, v^\mu_{r_\mu}\right) \quad \text{pour} \quad \mu \in \{1, \dots, d\}, \\ &f^{d+1} = \alpha, \end{aligned} \tag{II.34}$$

et

$$v = F_\mathcal{M}(f^1, \dots, f^{d+1}) = \sum_{i_1=1}^{r_1} \dots \sum_{i_d=1}^{r_d} \alpha_{i_1 \dots i_d} \bigotimes_\mu v^\mu_{i_\mu}. \tag{II.35}$$

Exemple II.3.5. *Avec $\mathcal{V}^\mu = \mathbb{R}^{n_\mu}$, f^μ est associé à une matrice de $\mathcal{F}^\mu_\mathcal{M} = \mathbb{R}^{n_\mu \times r_\mu}$ pour $\mu \in D$. $f^{d+1} = \alpha$ est un tenseur d'ordre d (dans le sens d'un tableau de réels à d dimensions, $(\alpha \in \bigotimes_\mu \mathbb{R}^{r_\mu})$).*

Pour les méthodes de minimisations alternées sur $\mathcal{F}^1_\mathcal{M} \times \dots \times \mathcal{F}^{d+1}_\mathcal{M}$, il faut prendre soin d'orthogonaliser les $(v^\mu_1, \dots, v^\mu_{r_\mu})$, $\mu \in D$. Ainsi le tenseur coeur $\alpha = f^{d+1}$ est toujours unique.

On peut également minimiser simultanément sur le couple $\mathcal{F}^\mu_\mathcal{M} \times \mathcal{F}^{d+1}_\mathcal{M}$, ce qui aboutit à l'algorithme AMA modifié, aussi appelé « Itérations Orthogonales de Haut Ordre » (Higher Order Orthogonal Iterations, HOOI). En effet, en remarquant que

$$\|u - v\| = \left\| u - \sum_{i \in \mathcal{I}} \alpha_i \bigotimes_\mu v^\mu_{i_\mu} \right\| \tag{II.36}$$

$$= \left\| u^{\{\lambda\}} - \sum_{i=1}^{r_\lambda} \sum_{j \in \mathcal{I}_\lambda} \alpha^{\{\lambda\}}_{ij} v^\lambda_i \otimes \left(\bigotimes_{\mu \neq \lambda} v^\mu_{j_\mu} \right) \right\| \tag{II.37}$$

$$= \left\| u^{\{\lambda\}} - \sum_{j \in \mathcal{I}_\lambda} \widetilde{v}^\lambda_{j_\lambda} \otimes \left(\bigotimes_{\mu \neq \lambda} v^\mu_{j_\mu} \right) \right\|, \tag{II.38}$$

où $\mathcal{I}_\lambda = \{(i_1, \dots, i_{\lambda-1}, i_{\lambda+1}, \dots, i_d);\ 1 \leq i_\mu \leq r_\mu\}$, la minimisation sur $\mathcal{F}^\mu_\mathcal{M} \times \mathcal{F}^{d+1}_\mathcal{M}$ revient à minimiser sur les $\left(\widetilde{v}^\lambda_{j_\lambda}\right)_{j_\lambda \in \mathcal{I}_\lambda}$. En assimilant les $\left(\widetilde{v}^\lambda_{j_\lambda}\right)_{j_\lambda \in \mathcal{I}_\lambda}$ à un élément de $\mathcal{V}^\lambda \otimes \mathbb{R}^{\#\mathcal{I}_\lambda}$, la minimisation sur les $\left(\widetilde{v}^\lambda_{j_\lambda}\right)_{j_\lambda \in \mathcal{I}_\lambda}$ se ramène à une décomposition optimale de rang r_λ, c'est-à-dire à une SVD

tronquée de rang r_λ. En pratique il est nécessaire de définir les opérateurs $[\cdot,\cdot]_\lambda : \mathcal{V} \times \mathcal{C}_1(\mathcal{V}) \to \mathcal{V}^\lambda,\ \forall \lambda \in D$, tels que

$$[u,v]_\lambda = \sum_{i=1}^{r_u} \prod_{\substack{\mu=1 \\ \mu\neq\lambda}}^{d} \langle u_i^\mu, v^\mu \rangle_\mu \, u_i^\lambda, \quad \text{avec } u = \sum_{i=1}^{r_u} \bigotimes_\mu u_i^\mu,\ v = \bigotimes_\mu v^\mu. \tag{II.39}$$

Ceci permet de construire une collection de $(\prod_{\mu=1}^d r_\mu)/r_\lambda$ vecteurs de $\mathcal{V}^\lambda$ notée w^λ définie comme

$$w^\lambda_{i_1\dots i_{\lambda-1} i_{\lambda+1}\dots i_d} = \left[u, \bigotimes_\mu v^\mu_{i_\mu} \right]_\lambda, \quad i_\mu \in \{1,\dots,r_\mu\},\ \forall \mu \in D, \tag{II.40}$$

et $w^\lambda = \left(w_i^\lambda\right)_{i\in\mathcal{I}_\lambda} \in \left(\mathcal{V}^\lambda\right)^{\#\mathcal{I}_\lambda} \simeq \mathcal{V}^\lambda \otimes \mathbb{R}^{\#\mathcal{I}_\lambda}$. La méthode proposée par De Lathauwer et al. [22] consiste à utiliser une SVD (voir section II.3.1) de w^λ dans $\mathcal{V}^\lambda \otimes \mathbb{R}^{\#\mathcal{I}_\lambda}$ pour résoudre le problème de minimisation à partir de l'équation (II.38) tout en conservant l'orthogonalité des vecteurs $\left(v_i^\lambda\right)_{1\le i\le r_\lambda}$.

Algorithme 5: Itérations orthogonales de haut ordre $\mathcal{T}_r(\mathcal{V})$

Données : $u \in \mathcal{V}$
Résultat : $v \in \mathcal{T}_r(\mathcal{V})$
Initialiser $(v_i^\mu)_{1\le i\le r_\mu}$, $\forall \mu \in D$ avec la HOSVD (algorithme 2);
tant que v *n'a pas convergé* **faire**
  pour $\lambda = 1,\dots,d$ **faire**
    Construire w^λ tel que défini en (II.40);
    Construire $\left(v_i^\lambda\right)_{1\le i\le r_\lambda}$ comme les r_λ vecteurs singuliers à gauche de w^λ;
    Construire α tel que défini en (II.26);
  fin
fin

II.3.2.4 Minimisations alternées dans $\mathcal{H}_r^T(\mathcal{V})$

On pose $m = \#I(T) + \#L(T)$. On définit $\mathcal{F}_\mathcal{M}^t$ par

$$\mathcal{F}_\mathcal{M}^t = \mathbb{R}^{r_{t_1}\times r_{t_2}\times r_t} \quad \text{si} \quad t \in I(T) \quad \text{avec} \quad t = t_1 \cup t_2,\ t_1 \cap t_2 = \emptyset, \tag{II.41}$$

$$\mathcal{F}_\mathcal{M}^t = \mathcal{V}^\mu \quad \text{si} \quad t = \{\mu\} \in L(T) \tag{II.42}$$

avec $r_D = 1$. Dans ce cas on va noter $\times_{t\in T}\mathcal{F}_\mathcal{M}^t = \mathcal{F}_\mathcal{M}^1 \times \dots \times \mathcal{F}_\mathcal{M}^m$ pour des raisons de clarté.

L'approche pour approximer un tenseur sous le format hiérarchique de Tucker consiste à minimiser alternativement sur les paramètres associés aux différents noeuds du tenseur. On arrive pour chaque noeud pour la résolution d'un système linéaire de la forme

$$A^t\alpha^t = b^t,\ \forall t \in I(T) \quad \text{ou} \quad A^t v^t = b^t, \quad \forall t \in L(T). \tag{II.43}$$

La construction des opérateurs $\left(A^t\right)_{t\in T}$ et $\left(b^t\right)_{t\in T}$ est très technique, le lecteur pourra se référer à l'article de Kressner et Tobler [51] pour des éclaircissements. De plus il peut arriver que l'opérateur

A^t soit non défini, on cherche alors α^t comme une solution de

$$\min_{\alpha \in \mathcal{F}^t_{\mathcal{M}}} \left\| A^t \alpha - b^t \right\|. \tag{II.44}$$

Pour éviter ce problème de minimisation, Kressner et Tobler [51] ont proposé une méthode de troncature adaptative afin de ne jamais avoir de problèmes singuliers.

Il est aussi possible de minimiser plusieurs noeuds à la fois en contractant des noeuds adjacents de l'arbre. Cette méthode est alors appelé algorithme des moindres carrés modifié par Kressner et Tobler [51], ou Groupe de Renormalisation de la Matrice Densité (Density Matrix Renormalization Group, DMRG) pour un certain type d'arbre T [63].

II.3.3 Approche gloutonne

Le problème des algorithmes présentés en section II.3.2 est qu'ils présupposent le rang de l'approximation à trouver. Cette hypothèse sur le rang peut résulter en une mauvaise approximation si r est trop faible, ou à des problèmes de complexité ou de convergence si r est trop grand. L'idée est alors de partir d'un rang r suffisamment petit, et de calculer des corrections successives. On construit alors une suite $(u_m)_{m \in \mathbb{N}}$ définie comme

$$\begin{cases} u_0 = 0 \\ w_m \in \Pi_{\mathcal{M}}(u - u_{m-1}), \quad \forall m \in \mathbb{N}^* \\ u_m = u_{m-1} + w_m, \quad \forall m \in \mathbb{N}^* \end{cases} \tag{II.45}$$

où $\mathcal{M}$ est un sous-ensemble de tenseurs. L'intérêt est de pouvoir considérer des ensembles $\mathcal{M}$ de petites dimensions pour diminuer la complexité de la résolution tout en espérant que la série

$$u_m = \sum_{i=1}^{m} w_i \tag{II.46}$$

converge rapidement. On verra des applications d'algorithmes gloutons aux chapitres III et IV. Le lecteur pourra se référer à Temlyakov [74] pour une analyse détaillée de ces algorithmes.

Chapitre III

Solveurs itératifs et approximation de tenseurs

Sommaire

On considère le problème générique présenté au chapitre I sous la forme (I.19) que l'on note encore

$$Au = b, \tag{III.1}$$

avec $A \in \mathcal{L}(\mathcal{V})$ et $b \in \mathcal{V}$. On retrouve un système linéaire que l'on peut résoudre avec des solveurs itératifs classiques [8, 69, 81]. Kressner et Tobler [50] ainsi que Ballani et Grasedyck [7] ont proposé de les coupler aux méthodes d'approximations de tenseurs présentées dans le chapitre II. Dans ce chapitre, on commencera par présenter ces approches. Ensuite on proposera une méthode pour réduire le rang de l'opérateur A. Ceci pourra être utile pour accélérer les algorithmes de résolutions. On s'appuiera notamment sur des résultats théoriques présentés en section III.3 afin de réduire le coût des méthodes d'approximation. Finalement on proposera un nouveau type de préconditionneur qui s'appuie sur une approximation gloutonne de A^{-1}. On prendra soin d'imposer des conditions sur la symétrie et le caractère creux du préconditionneur pour pouvoir les utiliser avec les nou-

veaux solveurs. Finalement on illustrera l'efficacité de notre préconditionneur sur des exemples où l'opérateur est symétrique ou non.

III.1 Méthodes de descente

III.1.1 Principe

Le principe de base des méthodes itératives est de construire une suite $(u_k)_{k\in\mathbb{N}} \in \mathcal{V}^{\mathbb{N}}$ telle que $u_k \to u$. Les « méthodes de descente » sont définies par la relation de récurrence suivante :

$$u_k = u_{k-1} + \arg\min_{v\in\mathcal{D}_k} \|A(u_{k-1}+v) - b\|_{\mathcal{N}} \tag{III.2}$$

où $\mathcal{D}_k$ est une direction de recherche et $\|\cdot\|_{\mathcal{N}}$ une norme dépendante de l'algorithme utilisé.

Lorsque l'on s'intéresse aux méthodes itératives couplées aux méthodes d'approximation de tenseurs, la formule de récurrence (III.2) implique que le rang de u_k va croître avec k. Afin d'éviter cet inconvénient, (III.2) est modifié en introduisant une approximation du tenseur u_k. La formule de récurrence devient alors

$$u_k \in \Pi_{\mathcal{M}}\left(u_{k-1} + \arg\min_{v\in\mathcal{D}_k} \|A(u_{k-1}+v) - b\|_{\mathcal{N}}\right) \tag{III.3}$$

où $\Pi_{\mathcal{M}}$ désigne l'ensemble des meilleures approximations sur un sous-ensemble $\mathcal{M} \subset \mathcal{V}$. Ainsi la suite $(u_k)_{k\in\mathbb{N}}$ est dans un ensemble de tenseurs à rang maximum donné et on évite les problèmes liés à l'augmentation du rang. En réalité, comme illustré en section III.1.2, l'opérateur de troncature $\Pi_{\mathcal{M}}$ est utilisé plusieurs fois dans l'algorithme de résolution.

III.1.2 Cas symétrique défini positif

Si A est symétrique défini positif, l'algorithme de référence est le gradient conjugué (Conjugate Gradient, CG) proposé par Hestenes et Stiefel [39]. Il a été adapté par Kressner et Tobler [50] aux tenseurs hiérarchiques de Tucker. L'utilisation de ce format est due à la possibilité d'atteindre des problèmes de hautes dimensions et à l'efficacité de la HSVD pour l'approximation quasi-optimale sur l'ensemble des tenseurs hiérarchiques de Tucker.

Comme A est symétrique défini positif, $\langle\cdot,\cdot\rangle_A = \langle A\cdot,\cdot\rangle$ est un produit scalaire de norme associée $\|\cdot\|_A$. On considère la fonctionnelle quadratique $J : \mathcal{V} \to \mathbb{R}$ définie par

$$J(v) = \frac{1}{2}\|v-u\|_A^2 = \frac{1}{2}\|Av-b\|_{A^{-1}}^2 . \tag{III.4}$$

J est strictement convexe et possède donc un unique minimum sur $\mathcal{V}$. On part d'un point initial u_0 qui peut être nul si aucune approximation de u n'est connue. Une méthode de descente sur J est définie par la relation de récurrence

$$u_{k+1} = u_k + \alpha_k p_k \quad \text{avec} \quad \alpha_k \in \mathbb{R},\ p_k \in \mathcal{V}. \tag{III.5}$$

Si u_k et p_k sont fixés, une minimisation de $J(u_k + \alpha_k p_k)$ par rapport à α_k donne

$$\alpha_k = \frac{\langle r_k, p_k \rangle}{\langle p_k, Ap_k \rangle} \tag{III.6}$$

où $r_k = -\nabla J(u_k) = b - Au_k$ est le résidu à la k-ième itération. Si l'on prend $p_k = -r_k$ on tombe classiquement sur une méthode de descente de gradient. Toute l'astuce du CG consiste à prendre les $(p_i)_{0 \leq i \leq k}$ A-conjugués, c'est-à-dire $\langle p_i, p_j \rangle_A = \delta_{ij} \|p_i\|_A^2$, et tels que les résidus $(r_i)_{0 \leq i \leq k}$ soient orthogonaux. Aussi la formule (III.5) implique la formule de récurrence suivante sur les résidus :

$$r_{k+1} = r_k - \alpha_k A p_k, \tag{III.7}$$

et l'orthogonalité des résidus donne

$$\alpha_k = \frac{\langle r_k, r_k \rangle}{\langle r_k, Ap_k \rangle}. \tag{III.8}$$

En construisant les directions de recherche avec la formule

$$p_{k+1} = r_{k+1} + \beta_k p_k, \tag{III.9}$$

la condition de A-orthogonalité donne

$$\beta_k = -\frac{\langle r_{k+1}, Ap_k \rangle}{\langle p_k, Ap_k \rangle} = -\frac{\langle r_{k+1}, r_{k+1} \rangle}{\langle r_k, r_k \rangle}, \tag{III.10}$$

la deuxième égalité venant de l'A-orthogonalité des $(p_i)_{0 \leq i \leq k}$, de l'orthogonalité des $(r_i)_{0 \leq i \leq k}$ et des équations (III.7) et (III.9).

Dans le cas de tenseurs, étant données les relations (III.5) et (III.9), le rang de u_k va croître fortement avec k. Afin de régler cet inconvénient, Kressner et Tobler [50] ont analysé l'introduction de l'opérateur de troncature dans l'algorithme classique du CG. Ils ont abouti alors à l'algorithme 6 dans le cas du gradient conjugué préconditionné par un préconditionneur P^{-1}. L'algorithme est en pratique initialisé avec la valeur $u_0 = 0$ si aucune approximation de la solution n'est disponible.

On remarquera que le résidu est calculé explicitement et non pas à partir de la formule de récurrence (III.7). Cette formulation permet d'éviter une stagnation trop rapide du résidu (voir Kressner et Tobler [50]).

Algorithme 6: Gradient Conjugué Préconditionné (Preconditioned Conjugate Gradient, PCG) avec troncature

Données : $u_0 \in \mathcal{M}$, $A \in \mathcal{L}(\mathcal{V})$ symétrique défini positif, $b \in \mathcal{V}$, $P \in \mathcal{L}(\mathcal{V})$ symétrique défini positif
Résultat : $u_k \in \mathcal{M}$
$r_0 = b - Au_0$;
$z_0 = Pr_0$;
$p_0 = z_0$;
$k = 0$;
tant que u_k *n'a pas convergé* **faire**
  $\alpha_k = \langle r_k, p_k \rangle \,/\, \langle p_k, Ap_k \rangle$;
  $u_{k+1} \in \Pi_{\mathcal{M}}(u_k + \alpha_k p_k)$;
  $r_{k+1} \in \Pi_{\mathcal{M}}(b - Au_{k+1})$;
  $z_{k+1} = Pr_{k+1}$;
  $\beta_k = -\langle z_{k+1}, Ap_k \rangle \,/\, \langle p_k, Ap_k \rangle$;
  $p_{k+1} \in \Pi_{\mathcal{M}}(z_{k+1} + \beta_k p_k)$;
  $k = k + 1$;
fin
retourner u_k;

III.1.3 Cas non symétrique

De nombreuses méthodes s'inspirent du CG pour résoudre les systèmes linéaires où A n'est plus symétrique. Une approche consiste à utiliser la méthode du gradient biconjugué (BiConjugate Gradient, BiCG) de Lanczos [53]. Le BiCG consiste en fait à résoudre le système suivant

$$\begin{cases} Au &= b, \\ A^*\hat{u} &= \hat{b}. \end{cases} \tag{III.11}$$

En introduisant l'opérateur $\tilde{A} : \mathcal{V}\times\mathcal{V} \to \mathcal{V}\times\mathcal{V}$ tel que $\tilde{A}(u,\hat{u}) = (A^*\hat{u}, Au)$, avec $\tilde{u} = (u,\hat{u})$, $\tilde{b} = (\hat{b}, b)$, résoudre (III.11) revient en fait à résoudre le système linéaire $\tilde{A}\tilde{u} = \tilde{b}$ avec $\tilde{A}$ symétrique[1]. Le problème est alors que $\tilde{A}$ n'est pas forcément défini positif ce qui peut causer des arrêts du BiCG. On peut prouver que le BiCG revient à chercher des résidus sous la forme

$$r_k = \varphi_k(A)r_0 \quad \text{et} \quad \widehat{r}_k = \psi_k(A)\widehat{r}_0, \tag{III.12}$$

où φ_k et ψ_k sont des polynômes. Pour accélérer la méthode, Sonneveld [72] a proposé le gradient conjugué au carré (Conjugate Gradient Squared, CGS) où on va chercher directement le résidu comme

$$r_k = \varphi_k^2(A)r_0. \tag{III.13}$$

À cause de l'utilisation du carré du polynôme φ_k, le CGS peut souffrir d'erreurs d'arrondis. Ceci a conduit au gradient biconjugué stabilisé (BiConjugate Gradient Stabilized, BiCGSTAB) de Van Der

1. $\tilde{A}$ est symétrique au sens du produit scalaire défini sur $\mathcal{V}\times\mathcal{V}$ par $\langle (u,\hat{u}),(v,\hat{v})\rangle_{\mathcal{V}\times\mathcal{V}} = \langle u, v\rangle + \langle \hat{u}, \hat{v}\rangle$

Vorst [80] où cette fois le résidu est cherché sous la forme

$$r_k = \chi_k(A)\varphi_k(A)r_0 \quad \text{où} \quad \chi_k(x) = (1-\omega_k x)\chi_{k-1}(x) \tag{III.14}$$

où ω_k est choisi de façon à minimiser le résidu r_k. Cet algorithme a alors été adapté avec succès par Kressner et Tobler [50] aux tenseurs en introduisant encore l'opérateur de troncature $\Pi_{\mathcal{M}}$. Le BiCGSTAB préconditionné avec un préconditionneur P^{-1} est présenté dans l'algorithme 7 où la formule explicite du résidu a été utilisée encore une fois.

Algorithme 7: Gradient BiConjugué Stabilisé Préconditionné avec Troncature

Données : $u_0 \in \mathcal{M}$, $A \in \mathcal{L}(\mathcal{V})$, $P \in \mathcal{L}(\mathcal{V})$, $b \in \mathcal{V}$, $\hat{r} \in \mathcal{V}$ (e.g. $\hat{r} = b$)
Résultat : $u_k \in \mathcal{M}$
$r_0 = b - Au_0$;
$\rho_0 = \langle \hat{r}, r_0 \rangle$;
$p_0 = r_0$;
$\hat{p}_0 = Pp_0$;
$v_0 = A\hat{p}_0$;
$k = 0$;
tant que u_k *n'a pas convergé* **faire**
  $\omega_k = \langle \hat{r}, r_k \rangle \,/\, \langle \hat{r}, v_k \rangle$;
  $s_k \in \Pi_{\mathcal{M}}(r_k - \omega_k v_k)$;
  $\hat{s}_k \in \Pi_{\mathcal{M}}(Ps_k)$;
  $t_k \in \Pi_{\mathcal{M}}(A\hat{s}_k)$;
  si $\|s_k\| \leq tol$ **alors**
    $u_{k+1} = u_k + \omega_k \hat{p}_k$;
    retourner u_{k+1};
  fin
  $\epsilon_k = \langle t_k, s_k \rangle \,/\, \langle t_k, t_k \rangle$;
  $u_{k+1} \in \Pi_{\mathcal{M}}(u_k + \omega_k \hat{p}_k + \epsilon_k \hat{s}_k)$;
  $r_{k+1} \in \Pi_{\mathcal{M}}(b - Au_{k+1})$;
  si $\|r_{k+1}\| \leq tol$ **alors**
    retourner u_{k+1};
  fin
  $\rho_{k+1} = \langle \hat{r}, r_{k+1} \rangle$;
  $\beta_k = \rho_{k+1}/\rho_k \omega_k / \epsilon_k$;
  $p_{k+1} \in \Pi_{\mathcal{M}}(r_{k+1} + \beta_k(p_k - \epsilon_k v_k))$;
  $\hat{p}_{k+1} \in \Pi_{\mathcal{M}}(Pp_{k+1})$;
  $v_{k+1} \in \Pi_{\mathcal{M}}(A\hat{p}_{k+1})$;
  $k = k + 1$;
fin

III.2 Méthodes de projection

III.2.1 Principe

Les méthodes de projection sont une alternative aux méthodes de descente présentées dans la section précédente. Elles consistent à définir un espace d'approximation $\mathcal{U} \subset \mathcal{V}$ tel que $\dim(\mathcal{U})$ est très inférieur à $\dim(\mathcal{V}) = n$ et un espace de projection $\mathcal{W} \subset \mathcal{V}$ avec $\dim(\mathcal{U}) = \dim(\mathcal{W})$. On vient chercher alors la projection de u dans $\mathcal{U}$ de manière à ce que le résidu soit orthogonal à $\mathcal{W}$.

En principe on construit deux suites d'espaces $(\mathcal{U}_k)_{k\geq 1}$ et $(\mathcal{W}_k)_{k\geq 1}$ tels que $\mathcal{U}_k \subset \mathcal{U}_{k+1}$ et $\mathcal{W}_k \subset \mathcal{W}_{k+1}$ et $\dim(\mathcal{U}_k) = \dim(\mathcal{W}_k)$. Il vient la suite $(u_k)_{k\geq 1}$ d'approximations de u définie par

$$u_k \in \mathcal{U}_k \quad \text{tel que} \quad \langle \delta w, b - Au_k \rangle = 0,\ \forall \delta w \in \mathcal{W}_k. \tag{III.15}$$

On reconnaît là une approximation de l'équation (I.19). Toute la difficulté réside alors dans le choix des espaces $(\mathcal{U}_k)_{k\geq 1}$ et $(\mathcal{W}_k)_{k\geq 1}$ afin que u_k soit une bonne approximation de u tout en ayant $\dim(\mathcal{U}_k)$ négligeable devant n.

III.2.2 Généralisation de la méthode de minimisation du résidu (GMRES)

Un choix populaire pour $\mathcal{U}_k$ est le k-ième sous espace de Krylov $\mathcal{K}_k(A, b) = \text{span}\left\{b, Ab, \ldots, A^{k-1}\right\}$. On peut montrer que si $\mathcal{K}_k(A, b) = \mathcal{K}_{k+1}(A, b)$ alors $\mathcal{K}_k(A, b) = \mathcal{K}_{k+q}(A, b),\ \forall q \in \mathbb{N}$, et $u \in \mathcal{K}_k(A, b)$ d'après [69, Proposition 6.6]. Aussi, la dimension maximale de $\mathcal{K}_k(A, b)$ est n.

En ce qui concerne $\mathcal{W}_k$, de nombreux choix ont été étudiés dans la littérature. On peut citer la *Méthode d'Orthogonalisation Totale* (Full Orthogonalization Method, FOM) qui consiste à prendre $\mathcal{W}_k = \mathcal{U}_k = \mathcal{K}_k(A, b)$ en utilisant une procédure d'orthogonalisation d'Arnoldi pour construire une base de $\mathcal{U}_k$. On peut aussi citer le choix $\mathcal{U}_k = \mathcal{K}_k(A, b)$ et $\mathcal{W}_k = \mathcal{K}_k(A^*, b)$ qui conduit à une méthode similaire à celle du gradient biconjugué. On va s'intéresser ici à l'algorithme proposé par Saad et Schultz [68], la *Généralisation de la Méthode de Minimisation du Résidu* (Generalized Minimal Residual, GMRES). En partant d'une initialisation $u_0 \in \mathcal{V}$, l'idée est de prendre pour $\mathcal{U}_k$ le k-ième sous espace de Krylov $\mathcal{K}_k(A, r_0)$ et $\mathcal{W}_k = A\mathcal{U}_k = A\mathcal{K}_k(A, r_0)$. On montre qu'avec cette méthode le résidu est minimisé sur l'espace affine $u_0 + \mathcal{U}_k$:

$$u_k = \arg \min_{v \in u_0 + \mathcal{U}_k} \|b - Av\|^2 . \tag{III.16}$$

En effet, on a $u_k = u_0 + U_k y_k$ où $y_k \in \mathbb{R}^k$ et $U_k : \mathbb{R}^k \to \mathcal{U}_k$ est un isomorphisme quelconque. La minimisation de $\|b - A(u_0 + U_k y)\|^2$ par rapport à y définit y_k comme

$$\langle b - A(u_0 + U_k y_k), AU_k \delta y \rangle = 0, \quad \forall \delta y \in \mathbb{R}^k. \tag{III.17}$$

Comme $\text{Im}(AU_k) = \mathcal{W}_k$, on retrouve l'équation (III.15) avec les sous-espaces $\mathcal{U}_k$ et $\mathcal{W}_k$ en remplaçant b par $b - Au_0$.

En pratique, pour éviter d'atteindre des valeurs de $\dim(\mathcal{U}_k)$ trop élevées, on réinitialise l'algorithme toutes les m itérations. On appelle habituellement cet algorithme GMRES(m). En introduisant l'opérateur de troncature, Ballani et Grasedyck [7] l'ont adapté aux espaces produit tensoriel

et sont arrivés à l'algorithme 8 où on note $(e_i)_{1\leq i\leq m}$ la base canonique de $\mathbb{R}^m$. On a ici changé la dernière étape de l'algorithme. Dans [7], Ballani et Grasedyck propose une troncature adaptative de z_{k+1} empêchant une stagnation du résidu. Dans ce cas $\mathcal{M}$ n'est plus fixé mais évolue avec les itérations de l'algorithme. On parlera plutôt de $\mathcal{M}_k$.

Algorithme 8: GMRES(m) avec troncature

Données : $u_0 \in \mathcal{M}$, $A \in \mathcal{L}(\mathcal{V})$, $P \in \mathcal{L}(\mathcal{V})$, $b \in \mathcal{V}$
Résultat : $u_k \in \mathcal{M}$
pour $k = 0, \ldots, k_{max}$ **faire**
  $r_k = b - Au_k$;
  si $\|u_k\| / \|b\| < tol$ **alors**
    retourner u_k;
  fin
  $v_1 = \Pi_{\mathcal{M}}(r_k)$;
  $v_1 = v_1 / \|v_1\|$;
  pour $j = 1, \ldots, m$ **faire**
    $w_j = Av_j$;
    Résoudre $\sum_{i=1}^{j} \langle v_l, v_i \rangle \alpha_i = \langle v_l, w_j \rangle$, $\forall l \in \{1, \ldots, j\}$;
    $v_{j+1} \in \Pi_{\mathcal{M}}(w_j - \sum_{i=1}^{j} \alpha_i v_i)$;
    $v_{j+1} = v_{j+1} / \|v_{j+1}\|$;
  fin
  Définir $W_m : \mathbb{R}^m \to \mathcal{W}_m$ tel que $W_m(e_i) = w_i,\ 1 \leq i \leq m$;
  Définir $U_m : \mathbb{R}^m \to \mathcal{U}_m$ tel que $U_m(e_i) = v_i,\ 1 \leq i \leq m$;
  Résoudre $\langle r_k + AU_m y, W_m \delta y \rangle = 0,\ \forall \delta y \in \mathbb{R}^m$;
  $z_{k+1} = u_k + U_m y$;
  $u_{k+1} \in \Pi_{\mathcal{M}}(z_{k+1})$;
fin

III.3 Approximation d'opérateurs

Dans cette partie on va proposer une méthode pour approximer un opérateur A avec une complexité réduite. Pour cela on va s'intéresser au cas $A \in \bigotimes_\mu \mathcal{U}^\mu$ où $\mathcal{U}^\mu \subset \mathcal{L}(\mathcal{V}^\mu)$ est un espace vectoriel, par exemple les sous-espaces des opérateurs symétriques ou creux, et montrer qu'on peut chercher la meilleur approximation de A directement dans $\bigotimes_\mu \mathcal{U}^\mu$. C'est en fait une généralisation du résultat d'Espig et Hackbusch [27] pour l'ensemble $\mathcal{C}_r(\mathcal{V})$ à une classe plus générale d'ensemble de tenseurs. Le problème de meilleure approximation (II.16) sera dérivé une nouvelle fois pour les opérateurs. On suppose que les $(\mathcal{V}^\mu)_{\mu\in D}$ sont de dimension finie $(n_\mu)_{\mu\in D}$. Les $\mathcal{L}(\mathcal{V}^\mu) = \mathbb{R}^{n_\mu \times n_\mu}$ sont alors des espaces de Hilbert pour la norme de Frobenius $\|\cdot\|_\mu$ et le produit scalaire associé $\langle A, B\rangle_\mu = \mathrm{trace}(A^* B)$.

Un argument sur les dimensions implique l'égalité $\mathfrak{L}(\mathcal{V}) = \bigotimes_\mu \mathcal{L}(\mathcal{V}^\mu) = \mathcal{L}(\mathcal{V})$ (voir section I.1.1). Cet espace est aussi un espace de Hilbert pour le produit scalaire $\langle\cdot,\cdot\rangle$ induit par les produits scalaires $\left(\langle\cdot,\cdot\rangle_\mu\right)_{\mu\in D}$ précédents et la norme associée $\|\cdot\|$. Cette dernière est une cross norm définie sur les tenseurs élémentaires par

$$\left\|\bigotimes_\mu A^\mu\right\| = \prod_\mu \|A^\mu\|_\mu \quad A^\mu \in \mathcal{L}(\mathcal{V}^\mu),\ \forall \mu \in D. \tag{III.18}$$

Dans cette section les notations $\mathcal{W} = \mathcal{L}(\mathcal{V})$ et $\mathcal{W}^\mu = \mathcal{L}(\mathcal{V}^\mu),\ \mu \in D$, seront utilisées.

III.3.1 Meilleure approximation sur un sous-ensemble de tenseurs

On considère un sous-ensemble de tenseurs $\mathcal{M}$ vérifiant les propriétés suivantes :

- $\mathcal{M}$ est fermé
- $0 \in \mathcal{M}$
- $\lambda\mathcal{M} = \mathcal{M}$ pour tout $\lambda \in \mathbb{R}^*$

On suppose que $\mathcal{M}$ est de la forme

$$\mathcal{M} = \left\{ \sum_{i\in\mathcal{I}} \alpha_i \otimes_\mu A^\mu_{i_\mu};\ (\alpha_i)_{i\in\mathcal{I}} \in \mathcal{B}_\mathcal{M} \subset \mathbb{R}^{\#\mathcal{I}},\ A^\mu_{i_\mu} \in \mathcal{W}^\mu \right\} \tag{III.19}$$

où $\mathcal{I} \in \mathbb{N}^d$ est un ensemble d'indices et $\mathcal{B}_\mathcal{M}$ un ensemble de $\#\mathcal{I}$ coefficients. On pourra notamment considérer pour $\mathcal{M}$ l'ensemble $\mathcal{C}_1(\mathcal{W})$, $\mathcal{T}_r(\mathcal{W})$ ou encore $\mathcal{H}^T_r(\mathcal{W})$. On définit une meilleure approximation dans $\mathcal{M}$ de $A \in \mathcal{W}$ par le problème

$$\Pi_\mathcal{M}(A) = \arg\min_{B\in\mathcal{M}} \|A - B\|^2 . \tag{III.20}$$

L'existence d'une solution est garantie car $\mathcal{M}$ est fermé. D'après la structure de $\mathcal{M}$, on peut écrire

$$\min_{B\in\mathcal{M}} \|A - B\|^2 = \min_{\substack{B\in\mathcal{M}\\ \|B\|=1}} \min_{\alpha\in\mathbb{R}} \|A\|^2 - 2\alpha\langle A, B\rangle + \alpha^2 = \|A\|^2 - \sigma(A)^2 \tag{III.21}$$

avec

$$\sigma(A) = \max_{\substack{B\in\mathcal{M}\\ \|B\|=1}} \langle A, B\rangle = \max_{B\in\mathcal{M}} \frac{\langle A, B\rangle}{\|B\|}. \tag{III.22}$$

$\sigma(A)$ peut être interprété comme la valeur singulière dominante de A (en un sens généralisé). On introduit l'application $\beta_A : \mathcal{W} \to \mathbb{R}^+$ définie par

$$\beta_A(B) = \frac{\langle A, B\rangle^2}{\|B\|^2}. \tag{III.23}$$

On a

$$\Pi_{\mathcal{M}}(A) = \arg\min_{B\in\mathcal{M}} \|A - B\|^2 = \arg\max_{B\in\mathcal{M}} \beta_A(B) \tag{III.24}$$

l'ensemble des éléments associés à la valeur singulière dominante $\sigma(A)$, tels que

$$\beta_A(B) = \sigma(A)^2 \quad \forall B \in \Pi_{\mathcal{M}}(A). \tag{III.25}$$

III.3.2 Approximation des éléments d'un sous-espace vectoriel

On suppose à présent que

$$A \in \mathcal{U} = \bigotimes_{\mu} \mathcal{U}^\mu \subset \mathcal{W} = \bigotimes_{\mu} \mathcal{W}^\mu \tag{III.26}$$

où les $\mathcal{U}^\mu \subset \mathcal{W}^\mu$ sont des sous-espaces vectoriels des $\mathcal{W}^\mu$, avec éventuellement $\mathcal{U}^\mu = \mathcal{W}^\mu$. On peut donc écrire

$$A = \sum_{i=1}^{m} \bigotimes_{\mu} A_i^\mu \quad \text{avec} \quad A_i^\mu \in \mathcal{U}^\mu \subset \mathcal{W}^\mu. \tag{III.27}$$

Pour tout $\mu \in \{1, \ldots, d\}$, on introduit le projecteur orthogonal de $\mathcal{W}^\mu$ sur $\mathcal{U}^\mu$ défini par

$$\mathcal{P}_\mu : \mathcal{W}^\mu \to \mathcal{U}^\mu \subset \mathcal{W}^\mu, \quad \mathcal{P}_\mu^2 = \mathcal{P}_\mu, \quad \mathcal{P}_\mu^* = \mathcal{P}_\mu. \tag{III.28}$$

On introduit ainsi le projecteur orthogonal de $\mathcal{W}$ sur $\mathcal{U}$ défini par

$$\mathcal{P} : \mathcal{W} \to \mathcal{U} \subset \mathcal{W}, \quad \mathcal{P}^2 = \mathcal{P}, \quad \mathcal{P}^* = \mathcal{P}, \tag{III.29}$$

et caractérisé par

$$\mathcal{P} = \bigotimes_{\mu} \mathcal{P}_\mu. \tag{III.30}$$

On montre aisément la propriété suivante.

Lemme III.3.1. *Soit $\mathcal{P} : \mathcal{W} \to \mathcal{U}$ le projecteur orthogonal de $\mathcal{W}$ sur $\mathcal{U}$, avec $\mathcal{P} = \bigotimes_\mu \mathcal{P}_\mu$. Pour tout ensemble $\mathcal{M}$ de la forme (III.19), on a*

$$\mathcal{P}(\mathcal{M}) \subset \mathcal{M}. \tag{III.31}$$

Lemme III.3.2. *Soit $A \in \mathcal{U} \subset \mathcal{W}$ et soit $\mathcal{P} : \mathcal{W} \to \mathcal{U}$ le projecteur orthogonal de $\mathcal{W}$ sur $\mathcal{U}$. Pour tout $B \in \mathcal{W}$, on a*

$$\beta_A(B) \leq \beta_A(\mathcal{P}(B)) \tag{III.32}$$

et

$$\beta_A(B) = \beta_A(\mathcal{P}(B)) \quad \Longleftrightarrow \quad B = \mathcal{P}(B). \tag{III.33}$$

Démonstration. Pour tout élément $B \in \mathcal{W}$, on a

$$\langle A, B\rangle = \langle \mathcal{P}(A), B\rangle = \langle A, \mathcal{P}^*(B)\rangle = \langle A, \mathcal{P}(B)\rangle \tag{III.34}$$

et

$$\|B\|^2 = \|\mathcal{P}(B)\|^2 + \|B - \mathcal{P}(B)\|^2 \geq \|\mathcal{P}(B)\|^2, \tag{III.35}$$

et par conséquent

$$\beta_A(B) = \frac{\langle A, B\rangle^2}{\|B\|^2} = \frac{\langle A, \mathcal{P}(B)\rangle^2}{\|B\|^2} \leq \frac{\langle A, \mathcal{P}(B)\rangle^2}{\|\mathcal{P}(B)\|^2} = \beta_A(\mathcal{P}(B)). \tag{III.36}$$

De plus, on a

$$\beta_A(B) = \beta_A(\mathcal{P}(B)) \Leftrightarrow \|B\| = \|\mathcal{P}(B)\| \Leftrightarrow \|B - \mathcal{P}(B)\| = 0. \tag{III.37}$$

□

On en déduit le théorème suivant.

Théorème III.3.3. *Soit $\mathcal{P} : \mathcal{W} \to \mathcal{U}$ le projecteur orthogonal de $\mathcal{W}$ sur un sous-espace vectoriel $\mathcal{U} \subset \mathcal{W}$. Si $A \in \mathcal{U}$ et si $\mathcal{P}(\mathcal{M}) \subset \mathcal{M}$, alors*

$$\Pi_{\mathcal{M}}(A) \subset \mathcal{U}. \tag{III.38}$$

Démonstration. Soit $C \in \Pi_{\mathcal{M}}(A) = \arg\max_{B\in\mathcal{M}} \beta_A(B)$. Par hypothèse sur $\mathcal{M}$, on a $\mathcal{P}(C) \in \mathcal{M}$. Supposons que $\mathcal{P}(C) \neq C$. Alors, par le lemme III.3.2, on a

$$\beta_A(C) < \beta_A(\mathcal{P}(C)) \tag{III.39}$$

ce qui contredit l'optimalité de C. On a donc $C = \mathcal{P}(C) \in \mathcal{U}$ et donc $\Pi_{\mathcal{M}}(A) \subset \mathcal{U}$. □

Du théorème III.3.3 et du lemme III.3.1, on déduit la propriété suivante.

Théorème III.3.4. *Si $A \in \mathcal{U} = \bigotimes_\mu \mathcal{U}^\mu$,*

$$\Pi_{\mathcal{C}_1(\mathcal{W})}(A) \subset \mathcal{C}_1(\mathcal{U}). \tag{III.40}$$

Un élément $B \in \Pi_{\mathcal{C}_1(\mathcal{W})}(A)$ s'écrit donc $B = \bigotimes_\mu B^\mu$ avec $B^\mu \in \mathcal{U}^\mu$.

Remarque III.3.5. *Les théorèmes III.3.3 et III.3.4 ne sont pas propres aux opérateurs, mais à tous les tenseurs.*

III.3.3 Analyse de l'algorithme alterné pour l'approximation de rang un

On considère le cas de l'approximation dans $\mathcal{M} = \mathcal{C}_1(\mathcal{W})$ d'un tenseur $A = \sum_{i=1}^m \bigotimes_\mu A_i^\mu \in \mathcal{U} = \bigotimes_\mu \mathcal{U}^\mu$. On considère l'application d'un algorithme alterné pour la recherche d'un élément

$B \in \Pi_{\mathcal{C}_1(\mathcal{W})}(A)$. On commence par une initialisation aléatoire $B_{(0)} = \bigotimes_\mu B^\mu_{(0)}$ puis on construit une séquence $\{B_{(n)}\}_{n\geq 1} \subset \mathcal{C}_1(\mathcal{W})$ définie pour $n \geq 1$ par

$$B_{(n)} = G_d \circ \ldots \circ G_1(B_{(n-1)}), \tag{III.41}$$

où pour $B = \bigotimes_\mu B^\mu$, $G_\mu : \mathcal{C}_1(\mathcal{W}) \to \mathcal{C}_1(\mathcal{W})$ est définie par

$$\|A - G_\mu(B)\|^2 = \min_{C \in \mathcal{C}_1^\mu(B)} \|A - C\|^2, \tag{III.42}$$

avec $\mathcal{C}_1^\mu(B)$ l'espace vectoriel défini par

$$\mathcal{C}_1^\mu(B) = \left\{ C^\mu \otimes \left(\bigotimes_{\lambda \neq \mu} B^\lambda \right) ; C^\mu \in \mathcal{W}^\mu \right\}, \tag{III.43}$$

où on a utilisé pour simplifier une permutation des dimensions.

Proposition III.3.6. *Quelle que soit l'initialisation $B_{(0)} \in \mathcal{W}$, on a*

$$B_{(n)} \in \mathcal{U} = \bigotimes_\mu \mathcal{U}^\mu \tag{III.44}$$

pour tout $n \geq 1$.

Démonstration. On a $G_\lambda(B) = C^\lambda \otimes \left(\bigotimes_{\mu \neq \lambda} B^\mu \right)$ avec C^λ solution de

$$\left\langle C^\lambda \otimes \left(\bigotimes_{\mu \neq \lambda} B^\mu \right), \delta C^\lambda \otimes \left(\bigotimes_{\mu \neq \lambda} B^\mu \right) \right\rangle = \left\langle A, \delta C^\lambda \otimes \left(\bigotimes_{\mu \neq \lambda} B^\mu \right) \right\rangle \quad \forall \delta C^\lambda \in \mathcal{W}^\lambda. \tag{III.45}$$

On en déduit

$$C^\lambda = \sum_{i=1}^m \alpha_i^\lambda A_i^\lambda \tag{III.46}$$

avec $\alpha_i^\lambda = \prod_{\mu \neq \lambda} \|B^\mu\|^{-2} \prod_{\mu \neq \lambda} \langle A_i^\mu, B^\mu \rangle_\mu$. Puisque pour tout $i \in \{1, \ldots, m\}$ on a $A_i^\lambda \in \mathcal{U}^\lambda$, on a

$$C^\lambda \in \mathcal{U}^\lambda. \tag{III.47}$$

Pour $B \in \mathcal{W}_1 \otimes \ldots \otimes \mathcal{W}_d$, on en déduit donc $G_1(B) \in \mathcal{U}_1 \otimes \mathcal{W}_2 \otimes \ldots \otimes \mathcal{W}_d$, $G_2 \circ G_1(B) \in \mathcal{U}_1 \otimes \mathcal{U}_2 \otimes \mathcal{W}_3 \otimes \ldots \otimes \mathcal{W}_d$, et par récurrence, $G_d \circ \ldots \circ G_1(B) \in \mathcal{U}_1 \otimes \ldots \otimes \mathcal{U}_d$. □

III.3.4 Préservation des patterns

On s'intéresse ici à la préservation des patterns des vecteurs ou matrices. On considère $\mathcal{W}^\mu = \mathbb{R}^{n_\mu}$ ou $\mathcal{W}^\mu = \mathbb{R}^{n_\mu \times n'_\mu}$. On considère un tenseur

$$A = \sum_{i=1}^m \bigotimes_\mu A_i^\mu. \tag{III.48}$$

On note

$$I_i^\mu = \{\gamma; (A_i^\mu)_\gamma \neq 0\} \quad \text{et} \quad I^\mu = \bigcup_{i=1}^{m} I_i^\mu. \tag{III.49}$$

I_i^μ définit le pattern de A_i^μ (vecteur ou matrice) et I^μ représente l'union de tous les patterns des $(A_i^\mu)_{\mu=1}^d$. On a

$$A \in \mathcal{U} = \bigotimes_\mu \mathcal{U}^\mu \tag{III.50}$$

avec $\mathcal{U}^\mu$ l'espace vectoriel défini par

$$\mathcal{U}^\mu = \{A^\mu \in \mathcal{W}^\mu; (A^\mu)_\gamma = 0,\ \forall \gamma \notin I^\mu\}. \tag{III.51}$$

Le projecteur orthogonal de $\mathcal{W}$ sur $\mathcal{U}$ est $\mathcal{P} = \bigotimes_\mu \mathcal{P}_\mu$ où $\mathcal{P}_\mu$ est le projecteur orthogonal de $\mathcal{W}^\mu$ sur $\mathcal{U}^\mu$ défini pour $B^\mu \in \mathcal{W}^\mu$ par

$$(\mathcal{P}_\mu(B^\mu))_\gamma = \begin{cases} (B^\mu)_\gamma & \text{si } \gamma \in I^\mu \\ 0 & \text{si } \gamma \notin I^\mu \end{cases}. \tag{III.52}$$

On déduit du théorème III.3.4 la propriété suivante.

Proposition III.3.7. *Un élément $B \in \Pi_{\mathcal{C}_1(\mathcal{W})}(A)$ s'écrit $B = \bigotimes_\mu B^\mu$ où B^μ a un pattern inclus dans I^μ.*

Lors de l'approximation d'un tenseur creux, on pourra donc se ramener à l'approximation d'un tenseur de dimension inférieure en limitant la recherche de la meilleure approximation dans l'espace $\mathcal{U}$. On pourra pour cela associer l'élément A à un élément

$$\widetilde{A} \in \bigotimes_\mu \mathbb{R}^{m_\mu} \quad \text{avec} \quad m_\mu = \#(I_\mu). \tag{III.53}$$

Notons également que d'après la proposition III.3.6, on sait que lorsqu'on utilise un algorithme alterné pour la recherche d'une meilleure approximation $B \in \mathcal{C}_1(\mathcal{W})$ de A, B possède les patterns de A après la première itération, quelle que soit l'initialisation de l'algorithme.

III.3.5 Préservation des symétries

On s'intéresse ici à la préservation des propriétés de symétrie des opérateurs. On considère $\mathcal{W}^\mu = \mathcal{M}_{n_\mu}(\mathbb{R}) = \mathbb{R}^{n_\mu \times n_\mu}$. On note $\mathcal{M}_n^{sym}(\mathbb{R}) = \{B \in \mathcal{M}_n(\mathbb{R}); B = B^T\}$ l'ensemble des matrices symétriques de $\mathcal{M}_n(\mathbb{R})$, et $\mathcal{M}_n^{skew}(\mathbb{R}) = \{B \in \mathcal{M}_n(\mathbb{R}); B = -B^T\}$ l'ensemble des matrices anti-symétriques de $\mathcal{M}_n(\mathbb{R})$. On considère un tenseur

$$A = \sum_{i=1}^{m} \bigotimes_\mu A_i^\mu. \tag{III.54}$$

On suppose que pour tout $i \in \{1, \ldots, m\}$,

$$A_i^\mu \in \mathcal{M}_{n_\mu}^{sym}(\mathbb{R}) \quad \text{pour} \quad \mu \in K_{sym} \subset D \tag{III.55}$$

et

$$A_i^\mu \in \mathcal{M}_{n_\mu}^{skew}(\mathbb{R}) \quad \text{pour} \quad \mu \in K_{skew} \subset D. \tag{III.56}$$

On a donc

$$A \in \mathcal{U} = \bigotimes_\mu \mathcal{U}^\mu, \tag{III.57}$$

avec $\mathcal{U}^\mu$ un espace vectoriel défini par

$$\mathcal{U}^\mu = \begin{cases} \mathcal{M}_{n_\mu}^{sym}(\mathbb{R}) & \text{si } \mu \in K_{sym} \\ \mathcal{M}_{n_\mu}^{skew}(\mathbb{R}) & \text{si } \mu \in K_{skew} \\ \mathcal{M}_{n_\mu}(\mathbb{R}) & \text{sinon} \end{cases}. \tag{III.58}$$

Le projecteur orthogonal de $\mathcal{W}$ sur $\mathcal{U}$ s'écrit $\mathcal{P} = \bigotimes_\mu \mathcal{P}_\mu$, avec $\mathcal{P}_\mu$ est le projecteur orthogonal de $\mathcal{W}^\mu$ sur $\mathcal{U}^\mu$ défini pour $B \in \mathcal{M}_{n_\mu}(\mathbb{R})$ par

$$\mathcal{P}_\mu(B) = \begin{cases} \frac{1}{2}(B + B^T) & \text{si } \mu \in K_{sym} \\ \frac{1}{2}(B - B^T) & \text{si } \mu \in K_{skew} \\ B & \text{sinon} \end{cases}. \tag{III.59}$$

On déduit du théorème III.3.4 la propriété suivante.

Proposition III.3.8. *Un élément $B \in \Pi_{\mathcal{C}_1(\mathcal{W})}(A)$ s'écrit $B = \bigotimes_\mu B^\mu$ où pout tout μ, B^μ possède les mêmes propriétés de symétrie que les $\{A_i^\mu\}_{i=1}^m$.*

Lors de l'approximation d'un tenseur $A \in \mathcal{W}$, on pourra donc imposer a priori les propriétés de symétrie (ou anti-symétrie) sur l'approximation recherchée. On pourra pour cela identifier A à un élément

$$\widetilde{A} \in \bigotimes_\mu \mathbb{R}^{m_\mu} \quad \text{avec} \quad m_\mu = \begin{cases} \frac{1}{2}n_\mu(n_\mu + 1) & \text{si } \mu \in K_{sym} \\ \frac{1}{2}n_\mu(n_\mu - 1) & \text{si } \mu \in K_{skew} \\ n_\mu^2 & \text{sinon} \end{cases}.$$

Notons également que d'après la proposition III.3.6, on sait que lorsqu'on utilise un algorithme alterné pour la recherche d'une meilleure approximation $B \in \mathcal{C}_1(\mathcal{W})$ de A, B possède les propriétés de symétrie de A après la première itération, quelle que soit l'initialisation de l'algorithme.

III.4 Approximation de l'inverse de l'opérateur

Après l'approximation de l'opérateur, l'approximation de l'inverse est présentée dans cette section. On va proposer une méthode qui permet de trouver une approximation dans $\mathcal{C}_r(\mathcal{W})$ et non plus $\mathcal{C}_1(\mathcal{W})$ comme c'est le cas dans la littérature [32, 54, 78]. Des approximations de rang-r ont déjà été proposées pour l'inverse si celle-ci admet un développement sous forme de série [46] ou en supposant une forme particulière [76]. La nouveauté ici réside dans le fait que la méthode est applicable à n'importe quel opérateur. On proposera aussi une manière d'imposer des propriétés telles que la

symétrie ou un caractère creux à l'approximation de l'inverse. De cette manière celle-ci pourra être utilisable avec le PCG par exemple.

Dans le cadre de la construction d'une approximation symétrique, on verra en section III.4.2.1 que l'on retrouve un cas particulier de l'équation de Sylvester. On proposera alors une manière de la résoudre avec l'utilisation des tenseurs.

III.4.1 Meilleure approximation par rapport à une norme appropriée

On introduit ici une méthode d'approximation gloutonne de A^{-1}. Soit $P_0 = 0$. Connaissant une approximation P_{r-1} de A^{-1}, on cherche une approximation $P_r = P_{r-1} + W$ avec

$$W \in \Pi_{\mathcal{C}_1(\mathcal{V})}(A^{-1} - P_{r-1}) = \arg \min_{X \in \mathcal{C}_1(\mathcal{V})} \left\| A^{-1} - P_{r-1} - X \right\|_{\mathcal{W}}^2 \tag{III.60}$$

où $\|\cdot\|_{\mathcal{W}}$ est une certaine norme sur $\mathcal{W}$. Pour être capable de calculer W, l'astuce consiste à choisir pour $\|\cdot\|_{\mathcal{W}}$ la norme $\|\cdot\|_{AA^*}$ associée au produit scalaire $\langle \cdot\, AA^*, \cdot \rangle$, où A^* est l'opérateur adjoint de A. On a alors l'égalité

$$\left\| A^{-1} - P_{r-1} - X \right\|_{AA^*}^2 = \left\| I - (P_{r-1} + X)A \right\|^2, \tag{III.61}$$

où $\|\cdot\|$ est la norme canonique induite par les normes de Frobenius. $P_r = P_{r-1} + W$ définit alors une approximation de l'inverse à gauche de A.

Remarque III.4.1. *On peut aussi construire une approximation à droite en considérant la norme associée au produit scalaire* $\langle A^*A\, \cdot, \cdot \rangle$.

En utilisant l'opérateur $[\cdot, \cdot]_\lambda$ défini en (II.39), une minimisation alternée sur les dimensions donne pour la dimension λ le problème suivant sur $W^\lambda \in \mathbb{R}^{n_\lambda \times n_\lambda}$:

$$[A^* - P_{r-1}AA^*, W]_\lambda = [WAA^*, W]_\lambda, \quad \forall \lambda \in D, \tag{III.62}$$

soit

$$Z_{r-1}^\lambda = W^\lambda Q^\lambda, \tag{III.63}$$

avec

$$Z_{r-1}^\lambda = [A^* - P_{r-1}AA^*, W]_\lambda \in \mathcal{L}(\mathcal{V}^\lambda), \tag{III.64}$$

$$Q^\lambda = \sum_{i=1}^{r_C} \prod_{\substack{\mu=1 \\ \mu \neq \lambda}}^{d} \langle W^\mu C_i^\mu, W^\mu \rangle_\mu \, C_i^\mu \in \mathcal{L}(\mathcal{V}^\lambda), \tag{III.65}$$

$$C = AA^* = \sum_{i=1}^{r_C} \bigotimes_\mu C_i^\mu. \tag{III.66}$$

Remarque III.4.2. *La dimension n_λ doit être suffisamment petite pour que cette méthode ne soit pas trop coûteuse.*

III.4.2 Approximation dans un sous-espace

Dans la section III.3, on pouvait tirer profit des propriétés de l'opérateur A quand il était dans un sous-espace de dimension inférieure $\mathcal{U} = \bigotimes_\mu \mathcal{U}^\mu$. Cette fois on ne connaît pas a priori les propriétés de A^{-1}. Toutefois, on va encore chercher son approximation dans des sous-espaces particuliers.

On note $\mathcal{P} : \mathcal{W} \to \mathcal{U}$ le projecteur orthogonal de $\mathcal{W}$ sur $\mathcal{U}$. Si $W = \bigotimes_\mu W^\mu \in \mathcal{U}$ est un minimiseur de (III.61), la condition de stationnarité donne

$$\langle I - (P_{r-1} + W)A, \delta W A\rangle = 0, \quad \forall \delta W \in T_W(\mathcal{C}_1(\mathcal{U})), \tag{III.67}$$

avec $T_W(\mathcal{C}_1(\mathcal{U}))$ l'espace tangent à $\mathcal{C}_1(\mathcal{U})$ en W. Cette équation est équivalente à

$$\langle A^* - (P_{r-1} + W)AA^*, \delta W\rangle = 0, \quad \forall \delta W \in T_W(\mathcal{C}_1(\mathcal{U})). \tag{III.68}$$

Sachant que $\mathcal{C}_1(\mathcal{U})$ est une variété différentielle incluse dans $\mathcal{U}$, il en découle que $T_W(\mathcal{C}_1(\mathcal{U})) \subset \mathcal{U}$ d'après le lemme II.2.6. Ainsi on a les égalités $T_W(\mathcal{C}_1(\mathcal{U})) = \mathcal{P}(T_W(\mathcal{C}_1(\mathcal{U})))$ et $\delta W = \mathcal{P}(\delta W)$, pour tout $\delta W \in T_W(\mathcal{C}_1(\mathcal{U}))$,et (III.68) est équivalente à

$$\langle A^* - (P_{r-1} + W)AA^*, \mathcal{P}(\delta W)\rangle = 0, \qquad \forall \delta W \in T_W(\mathcal{C}_1(\mathcal{U})) \tag{III.69}$$

$$\Leftrightarrow \quad \langle \mathcal{P}(A^* - (P_{r-1} + W)AA^*), \delta W\rangle = 0, \qquad \forall \delta W \in T_W(\mathcal{C}_1(\mathcal{U})), \tag{III.70}$$

ou encore

$$\langle \mathcal{P}(WAA^*), \delta W\rangle = \langle \mathcal{P}(A^* - P_{r-1}AA^*), \delta W\rangle\,, \quad \forall \delta W \in T_W(\mathcal{C}_1(\mathcal{U})). \tag{III.71}$$

En introduisant une minimisation alternée sur les dimensions, on peut écrire à partir de (III.71) que la minimisation sur $W^\lambda,\ \lambda \in D$, conduit à

$$[\mathcal{P}(WAA^*), W]_\lambda = [\mathcal{P}(A^* - P_{r-1}C), W]_\lambda\,, \tag{III.72}$$

où $[\cdot,\cdot]_\lambda$ est l'application définie en (II.39).

Remarque III.4.3. *$P_{r-1}AA^*$ est un tenseur de rang maximal $(r-1)r_A^2$. Si r_A est trop grand, la résolution de (III.71) peut être coûteuse. Pour éviter ce problème, on peut travailler sur une approximation de A de plus petit rang en utilisant les propriétés proposées en section III.3 comme les symétries et le caractère creux des opérateurs.*

III.4.2.1 Imposition de symétries

Formulation du système à résoudre. On sait que si A est symétrique, A^{-1} l'est aussi. Néanmoins, avec

$$A^{-1} = \sum_{i=1}^{r_P} \bigotimes_\mu P_i^\mu, \tag{III.73}$$

rien n'impose que les $\left(\bigotimes_\mu P_i^\mu\right)_{1\le i\le r_P}$ soient symétriques. Il est donc nécessaire d'imposer la symétrie dans la définition de l'approximation. Pour cela on introduit le sous-espace $\mathcal{U} \subset \mathcal{W}$ des opérateurs

symétriques et le projecteur $\mathcal{P} : \mathcal{W} \to \mathcal{U}$ défini par

$$\mathcal{P}(X) = \frac{1}{2}(X + X^*). \tag{III.74}$$

L'équation (III.71) devient alors

$$\langle WC, \delta W\rangle + \langle CW, \delta W\rangle = \langle 2\mathcal{P}(A^* - P_{r-1}C), \delta W\rangle\,, \quad \forall \delta W \in T_W(\mathcal{C}_1(\mathcal{U})), \tag{III.75}$$

avec $C = AA^* = \sum_{i=1}^{r_C} \bigotimes_\mu C_i^\mu = C^*$, $W = W^*$ et $P_{r-1} = P_{r-1}^*$. Pour appliquer un algorithme de minimisations alternées, on va supposer que l'on connaît $(W^\mu)_{\mu\in D\setminus\lambda}$ pour $W \in \bigotimes_\mu W^\mu$. La minimisation sur W^λ conduit à l'équation de stationnarité

$$\langle WC, \delta W\rangle + \langle CW, \delta W\rangle = \langle 2\mathcal{P}(A^* - P_{r-1}C), \delta W\rangle\,, \quad \forall \delta W = \delta W^\lambda \otimes \left(\bigotimes_{\mu\neq\lambda} W^\mu\right), \tag{III.76}$$

où

$$\begin{aligned}
\langle WC, \delta W\rangle &= \left\langle \sum_{i=1}^{r_C} \left(\prod_{\mu\in D\setminus\lambda} \langle W^\mu C_i^\mu, W^\mu\rangle_\mu\right) W^\lambda C_i^\lambda, \delta W^\lambda \right\rangle_\lambda,\\
\langle CW, \delta W\rangle &= \left\langle \sum_{i=1}^{r_C} \left(\prod_{\mu\in D\setminus\lambda} \langle C_i^\mu W^\mu, W^\mu\rangle_\mu\right) C_i^\lambda W^\lambda, \delta W^\lambda \right\rangle_\lambda,\\
\langle 2\mathcal{P}(A^* - P_{r-1}C), \delta W\rangle &= \left\langle \sum_{i=1}^{r_E} \left(\prod_{\mu\in D\setminus\lambda} \langle E_i^\mu, W^\mu\rangle_\mu\right) E_i^\lambda, \delta W^\lambda \right\rangle_\lambda,
\end{aligned}$$

avec $E = 2\mathcal{P}(A^* - P_{r-1}C) = \sum_{i=1}^{r_E} \bigotimes_\mu E_i^\mu$. On en déduit l'équation suivante qui définit W^λ :

$$W^\lambda Q^\lambda + G^\lambda W^\lambda = H_{r-1}^\lambda \tag{III.77}$$

qui est l'équation de Sylvester avec

$$\begin{aligned}
Q^\lambda &= \sum_{i=1}^{r_C} \left(\prod_{\mu\in D\setminus\lambda} \langle W^\mu C_i^\mu, W^\mu\rangle_\mu\right) C_i^\lambda,\\
G^\lambda &= \sum_{i=1}^{r_C} \left(\prod_{\mu\in D\setminus\lambda} \langle C_i^\mu W^\mu, W^\mu\rangle_\mu\right) C_i^\lambda,\\
H^\lambda &= \sum_{i=1}^{r_E} \left(\prod_{\mu\in D\setminus\lambda} \langle E_i^\mu, W^\mu\rangle_\mu\right) E_i^\lambda.
\end{aligned}$$

On remarque alors que si $\mathcal{U} = \bigotimes_\mu \mathcal{U}^\mu$ où les $\mathcal{U}^\mu$ est l'espace des opérateurs symétriques de $\mathcal{W}^\mu$, on a

$$\langle W^\mu C_i^\mu, W^\mu\rangle_\mu = \langle C_i^\mu, (W^\mu)^* W^\mu\rangle_\mu = \langle C_i^\mu, W^\mu (W^\mu)^*\rangle_\mu = \langle C_i^\mu W^\mu, W^\mu\rangle_\mu\,, \tag{III.78}$$

et donc $Q^\lambda = G^\lambda$. L'équation à résoudre est donc finalement

$$W^\lambda Q^\lambda + Q^\lambda W^\lambda = H^\lambda_{r-1}. \tag{III.79}$$

Cette dernière équation est appelée l'équation de Lyapunov continue qui est un cas particulier de l'équation de Sylvester. Elle peut être résolue grâce à l'algorithme de Bartels and Stewart [9]. Lorsque n_μ est grand, on préférera la méthode proposée dans le paragraphe suivant.

Résolution de l'équation de Sylvester par une méthode d'approximations de tenseurs. L'équation de Sylvester s'écrit :

$$\begin{aligned} &\text{Trouver } X \in \mathbb{R}^{n\times n} \text{ tel que} \\ &AX + XB = C, \quad (A,B,C) \in \left(\mathbb{R}^{n\times n}\right)^3. \end{aligned} \tag{III.80}$$

En se replaçant dans un cadre tensoriel où on fait identification $\mathbb{R}^{n\times n} = \mathbb{R}^n \otimes \mathbb{R}^n$, l'équation précédente est équivalente à

$$(A \otimes I_n + I_n \otimes B^T)X = C, \tag{III.81}$$

où I_n est l'opérateur identité de $\mathbb{R}^{n\times n}$ et X et C sont considérés comme des éléments de $\mathbb{R}^n \otimes \mathbb{R}^n$. Il reste à exprimer C sous format tenseur grâce à une SVD et à utiliser un des solveurs des chapitres III et IV pour trouver une approximation de X sous la forme $\sum_{i=1}^{r_X} X_i^1 \otimes X_i^2$ où $X_i^\mu \in \mathbb{R}^n$.

III.4.2.2 Imposition d'un caractère creux

On considère ici que $\mathcal{W}^\mu = \mathbb{R}^{n_\mu \times n_\mu}$. Avec les projecteurs introduits en section III.3.4, en développant le membre de gauche de l'équation (III.72) on obtient

$$\begin{aligned} [\mathcal{P}(WC), W]_\lambda &= \sum_{i=1}^{r_C} \left(\prod_{\mu \in D\setminus\lambda} \langle \mathcal{P}^\mu(W^\mu C_i^\mu), W^\mu \rangle_\mu \right) \mathcal{P}^\lambda(W^\lambda C_i^\lambda) \\ &= \mathcal{P}^\lambda(W^\lambda K^\lambda), \end{aligned} \tag{III.82}$$

avec

$$K^\lambda = \sum_{i=1}^{r_C} \left(\prod_{\mu \in D\setminus\lambda} \langle \mathcal{P}^\mu(W^\mu C_i^\mu), W^\mu \rangle_\mu \right) C_i^\lambda. \tag{III.83}$$

Cette fois l'équation est

$$\mathcal{P}^\lambda(W^\lambda K^\lambda) = [\mathcal{P}(A^* - P_{r-1}C), W]_\lambda\,, \tag{III.84}$$

et ne peut être résolue directement. On peut toutefois s'inspirer de la méthode de l'Approximation Creuse de l'Inverse (SParse Approximate Inverse, SPAI) de Grote et Huckle [36] en minimisant la fonctionnelle J^λ définie par :

$$J^\lambda(W^\lambda) = \left\| \mathcal{P}^\lambda(W^\lambda K^\lambda) - [\mathcal{P}(A^* - P_{r-1}C), W]_\lambda \right\|_\lambda^2, \quad \forall \lambda \in D. \tag{III.85}$$

III.5 Illustrations

III.5.1 Estimation de l'erreur

On rappelle que l'on est en dimension finie. On munit alors $\mathcal{L}(\mathcal{V})$ de la norme subordonnée à $\|\cdot\|$ que l'on notera aussi $\|\cdot\|$. On a ainsi les inégalités :

$$\|b\| \leq \|A\| \, \|u\| \quad \text{et} \quad \|u\| \leq \left\|A^{-1}\right\| \|b\| . \tag{III.86}$$

De même, on a les inégalités :

$$\|Av - b\| \leq \|A\| \, \|u - v\| \quad \text{et} \quad \|u - v\| \leq \left\|A^{-1}\right\| \|b - Au\| . \tag{III.87}$$

De (III.86) et (III.87), il découle que

$$\frac{1}{\mathrm{cond}(A)} \frac{\|b - Av\|}{\|b\|} \leq \frac{\|u - v\|}{\|u\|} \leq \mathrm{cond}(A) \frac{\|b - Av\|}{\|b\|}, \tag{III.88}$$

avec $\mathrm{cond}(A) = \|A\| \left\|A^{-1}\right\|$ le conditionnement de A associé à la norme $\|\cdot\|$. Ainsi si le conditionnement de l'opérateur A est très bon, le résidu relatif est un bon estimateur d'erreur du résidu relatif.

Remarque III.5.1. *Pour deux approximations u_r et $\widetilde{u}_r$ de u, les deux inégalités suivantes peuvent être vérifiées simultanément :*

$$\|b - Au_r\| \leq \|b - A\widetilde{u}_r\| \quad \textit{et} \quad \|u - u_r\| \geq \|u - \widetilde{u}_r\| . \tag{III.89}$$

Le contrôle de l'erreur en norme de résidu peut donc poser des problèmes. Ceci sera illustré sur des exemples en sections III.5.3.5 et IV.3.2.

Remarque III.5.2. *En pratique, il est plus utile de passer par une estimation de l'erreur sur une quantité d'intérêt comme illustré en section V.1.*

III.5.2 Problème de Poisson

III.5.2.1 Formulation du problème

On va s'intéresser au problème formulé sur le domaine $\Omega = \omega^d$ avec $\omega = (0, 1)$ ayant pour formulation forte :

$$\begin{aligned} -\Delta u &= f \quad \text{sur } \Omega, \\ u &= 0 \quad \text{sur } \partial\Omega, \\ f &= 1 \quad \text{sur } \Omega. \end{aligned} \tag{III.90}$$

La formulation faible du problème précédent est :

$$\begin{aligned} &\text{Trouver } u \in H_0^1(\Omega) \text{ tel que} \\ &a(u, v) = \ell(v), \quad \forall v \in H_0^1(\Omega), \end{aligned} \tag{III.91}$$

où

$$\begin{aligned} a(u,v) &= \int_\Omega \nabla u \cdot \nabla v \; dx, \\ \ell(v) &= \int_\Omega v f \; dx, \\ H_0^1(\Omega) &= \bigotimes_\mu H_0^1(\omega). \end{aligned} \tag{III.92}$$

On cherche alors une approximation de u dans l'espace $\mathcal{V} = \bigotimes_\mu \mathcal{V}^\mu$ où les $\mathcal{V}^\mu$ sont les espaces d'approximation éléments finis $P1$ inclus dans $H_0^1(\omega)$. On introduit un maillage régulier de 60 éléments de ω. On suppose ici que $d = 4$.

Avec ces notations, la discrétisation du problème amène au système linéaire

$$Au = b, \quad A \in \bigotimes_{\mu=1}^{d} \mathbb{R}^{n\times n}, \; b \in \bigotimes_{\mu=1}^{d} \mathbb{R}^n. \tag{III.93}$$

Comme l'opérateur est symétrique défini positif, on va étudier le comportement de l'algorithme du gradient conjugué (PCG). On va se placer dans le cas où les approximations $u_r \in \mathcal{H}_{10}^T(\mathcal{V})$, où T est un arbre équilibré.

III.5.2.2 Comportement du solveur et des préconditionneurs

Le but de cet section est d'analyser l'efficacité du préconditionneur proposé sur le simple modèle de Poisson. En figure III.1, le résidu relatif est tracé en fonction du nombre d'itérations du PCG pour différents préconditionneurs. Dans le cas I, l'algorithme est simplement le gradient conjugué. Le préconditionneur $\widehat{P}_1$ est tel que $\widehat{P}_1^{-1} \in \Pi_{\mathcal{C}_1(\mathcal{L}(\mathcal{V}))}(A)$, c'est-à-dire $\widehat{P}_1^{-1}$ est la meilleure approximation de rang 1 de l'opérateur A au sens de la norme canonique. Enfin, $(P_r)_{1\le r\le 10}$ désignent les préconditionneurs de rang r construits de manière gloutonne dans l'espace des opérateurs symétriques (voir la section III.4.2.1).

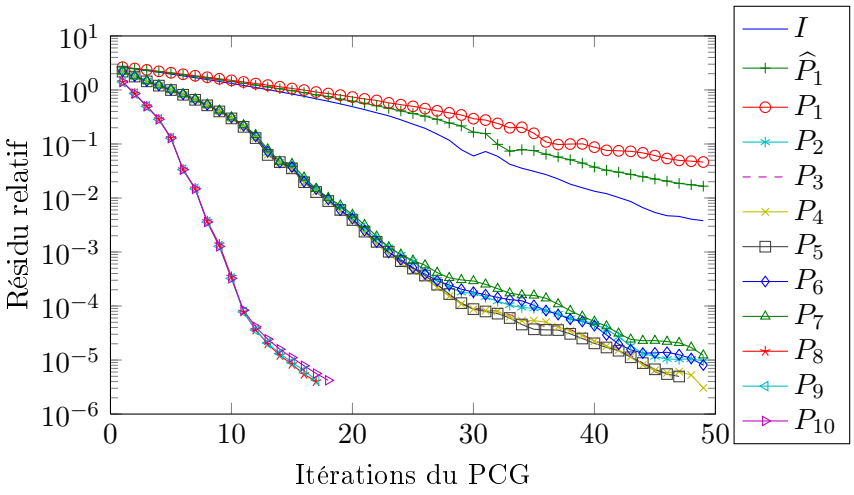


FIGURE III.1 – Résidu relatif en fonction du nombre d'itérations du PCG pour différents types de préconditionneurs pour le problème de Poisson.

On observe que $\widehat{P}_1$ et P_1 n'améliorent pas du tout la convergence du solveur, et la dégradent même. Nonobstant, à partir de $r = 2$, P_r augmente très fortement le taux de convergence du résidu, l'augmentation étant encore plus flagrante à partir de $r = 8$.

Afin de voir s'il y a un lien entre la convergence de $(P_r)_{r\in\mathbb{N}^*}$ vers A^{-1} et l'efficacité du préconditionneur, on a tracé en figure III.2 l'erreur relative

$$\frac{\|I - P_r A\|}{\|I\|} \tag{III.94}$$

en fonction de r. Malheureusement il n'y a pas de liens visibles entre l'erreur relative et le taux de

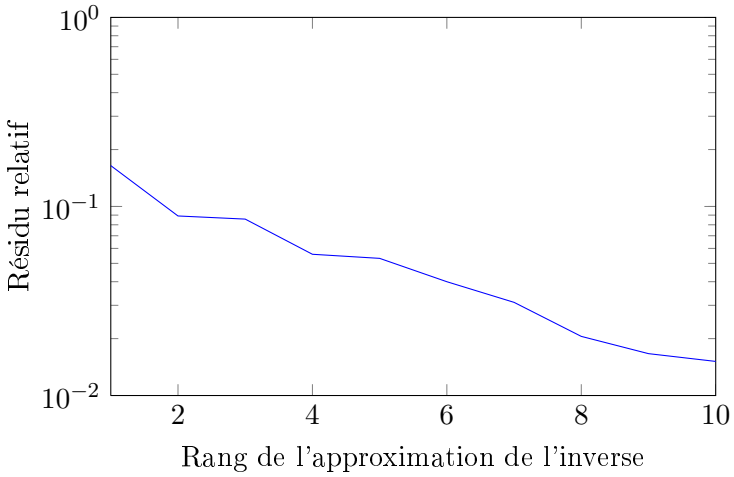


FIGURE III.2 – Erreur relative $\frac{\|I-P_r A\|}{\|I\|}$ en fonction du rang r de l'approximation de l'inverse pour le problème de Poisson.

convergence du solveur. En effet on observe une décroissance stricte de l'erreur alors qu'il semble que le conditionnement s'améliore par paliers d'après la figure III.1. Il serait nécessaire d'avoir un accès au conditionnement de l'opérateur préconditionné $P_r A$. Pour cela il faudrait estimer les normes $\|P_r A\|$ et $\left\|(P_r A)^{-1}\right\| = \left\|A^{-1} P_r^{-1}\right\|$, la difficulté résidant dans le calcul de cette dernière.

III.5.2.3 Temps de calcul

Problème de départ. Les temps de calcul pour la construction des différents préconditionneurs, la résolution et le temps total sont donnés dans le tableau III.1. La solution est cherché dans l'ensemble $\mathcal{H}_{10}^T(\mathcal{V})$. La limite de précision en résidu relatif est fixée à $5\ 10^{-6}$ car le résidu relatif n'arrive pas à atteindre 10^{-6}, peu importe le préconditionneur utilisé.

On conclut alors que le choix du préconditionneur est une affaire de compromis. Ici le préconditionneur le plus efficace en temps est P_2. Il allie la rapidité de la construction et la rapidité de la convergence du résidu. Néanmoins si le préconditionneur est déjà disponible, il est alors plus intéressant de choisir le préconditionneur P_8.

Augmentation du rang de la solution. La solution est cette fois autorisée à avoir un rang maximal de 20. Il en découle que tous les préconditionneurs aboutissent à un résidu relatif final

inférieur à 10^{-8}, comme indiqué dans le tableau III.2.

Cette fois P_2 est le meilleur préconditionneur, aussi bien pour le temps de construction du préconditionneur que la résolution du système. On observe une augmentation du temps de calcul avec l'augmentation du rang. Le temps de calcul est au moins triplé, même pour un résidu relatif grossier de 10^{-1}.

Problème avec 120 éléments par dimension. On revient ici à l'ensemble $\mathcal{H}_{10}^{T}(\mathcal{V})$, mais on divisé par deux ici la taille du maillage pour appréhender l'efficacité de la méthode. Les résultats sont indiqués au tableau III.3. On a encore fixé ici la limite en précision à 5.10^{-5} pour des raisons de stagnation du résidu relatif, comme dans le problème de départ. Lorsque le résidu relatif n'est pas atteint, le temps est noté ∞.

P_2 est une nouvelle fois le meilleur compromis pour allier rapidité et précision. Si P_8 est disponible, ce dernier offre la résolution la plus rapide du problème. On remarque que le temps de calcul souffre du raffinement du maillage du domaine ω. Néanmoins pour un nombre de degrés de liberté doublé, le temps de calcul est à peine multiplié par 4. L'augmentation du temps de calcul provient donc presque uniquement de la résolution des systèmes linéaires, car on retrouve une dépendance concordante avec la complexité des solveurs directs de l'algèbre linéaire. Typiquement, pour un système linéaire de n inconnues, ils ont une complexité entre n^2 et n^3 suivant le caractère creux de l'opérateur.

	I	$\widehat{P}_1$	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
Constr. du préc.	0	0.1	0.3	1.7	2.2	3.8	4.6	7.2	10.2	12.7	15.0	23.9
Résidu relatif	Temps de résolution											
10^{-1}	2.8	3.1	3.3	1.3	1.4	1.7	2.1	2.5	3.0	1.8	2.1	2.7
10^{-2}	7.5	6.2	7.5	1.8	2.0	2.4	3.0	3.4	4.2	2.2	2.7	3.3
10^{-3}	7.6	11.2	13.0	2.3	2.5	3.0	3.9	4.4	5.3	2.7	3.3	4.0
10^{-4}	11.5	15.0	18.5	3.3	3.3	4.0	4.9	6.8	8.2	3.0	3.6	4.2
10^{-5}	13.5	19.6	23.8	4.8	4.9	5.9	7.2	9.2	10.8	3.9	4.7	5.8
$5\ 10^{-6}$	13.9	20.7	24.5	5.1	5.1	6.5	7.7	9.6	11.2	4.4	5.2	6.5
Temps total	13.9	20.8	24.8	6.8	7.3	10.3	12.3	16.8	21.4	17.1	20.2	30.4

TABLE III.1 – Temps de calcul (s) pour la résolution du problème de Poisson en fonction du résidu relatif pour différents préconditionneurs, une approximation de rang 10 et $n = 59$.

	I	$\widehat{P}_1$	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
Constr. du préc.	0	0.1	0.3	1.7	2.2	3.8	4.6	7.2	10.2	12.7	15.0	23.9
Résidu relatif	Temps de résolution											
10^{-1}	7.2	7.7	8.1	3.7	4.5	5.5	6.8	8.5	11.0	6.2	7.7	9.6
10^{-2}	9.8	11.5	12.1	5.4	6.0	7.2	9.0	11.9	15.4	8.3	10.2	12.8
10^{-3}	11.9	14.4	18.7	6.6	7.9	9.5	11.7	14.6	18.9	10.4	12.8	15.9
10^{-4}	14.6	22.4	30.4	7.6	9.0	10.8	13.4	16.7	21.5	11.5	14.0	17.4
10^{-5}	22.0	33.4	44.3	8.5	10.1	12.1	15.0	18.7	24.2	12.5	15.4	19.0
10^{-6}	28.0	47.9	58.7	9.8	11.6	13.9	16.6	21.4	27.7	15.7	19.3	23.8
10^{-7}	35.9	59.1	72.8	11.7	13.9	16.6	20.5	25.6	32.0	16.8	20.6	25.3
10^{-8}	46.1	71.5	87.9	16.5	19.1	21.9	25.4	30.4	38.2	18.9	23.2	28.5
Temps total	60	71.6	88.2	18.2	21.3	25.7	30	37.6	48.4	31.6	38.2	52.4

TABLE III.2 – Temps de calcul (s) pour la résolution du problème de Poisson en fonction du résidu relatif pour différents préconditionneurs, une approximation de rang 20 et $n = 59$.

	I	$\widehat{P}_1$	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
Constr. du préc.	0	0.1	1.7	4.9	6.2	8.3	11.7	16.2	19.7	32.6	40.0	46.4
Résidu relatif	Temps de résolution											
10^{-1}	6.6	9.9	11.1	3.0	3.3	4.1	4.6	6.0	8.2	4.6	5.8	7.7
10^{-2}	15.4	20.3	25.9	5.6	6.6	8.1	8.6	11.5	16.6	6.3	8.0	10.5
10^{-3}	24.3	28.8	38.0	9.0	10.8	12.4	15.1	18.5	25.3	8.4	10.6	13.9
10^{-4}	30.3	∞	∞	13.5	14.8	17.6	21.0	26.9	35.9	11.1	14.1	19.0
$5\ 10^{-5}$	33.8	∞	∞	14.5	16.4	19.3	23.4	29.7	37.4	12.8	16.2	22.3
Temps total	33.8	∞	∞	19.4	22.6	27.6	35.1	45.9	57.1	45.4	56.2	68.7

TABLE III.3 – Temps de calcul (s) pour la résolution du problème de Poisson en fonction du résidu relatif pour différents préconditionneurs, une approximation de rang 10 et $n = 119$.

III.5.3 Problème symétrique

III.5.3.1 Description du problème

On considère le problème aux limites stochastique posé sur le domaine représenté en figure III.3. Les sous-domaines possèdent des conductivités aléatoires indépendantes. En ce qui concerne les

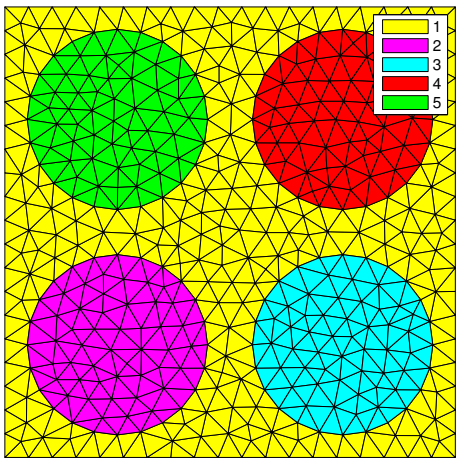


FIGURE III.3 – Domaine, sous-domaines et maillage du problème stochastique.

inclusions, leurs conductivités suivent une loi aléatoire log-uniforme comprise entre 10 et 100. Quant à la conductivité de la matrice, elle suit une loi uniforme comprise entre 1 et 2. La conductivité κ est donc fonction de 5 variables aléatoires $\xi = (\xi_1, \ldots, \xi_5)$ supposées indépendantes, chacune d'entre elles représentant la conductivité sur un sous-domaine. Le problème aux limites s'écrit :

$$\begin{aligned} \nabla(-\kappa \nabla u) &= 1 \quad \text{sur } \Omega \times \Xi, \\ u &= 0 \quad \text{sur } \partial\Omega \times \Xi. \end{aligned} \tag{III.95}$$

La conductivité κ étant uniformément bornée (inférieurement et supérieurement), le problème (III.95) est bien posé (voir la section I.3.6). On utilise des éléments finis linéaires pour la discrétisation spatiale. Le problème semi-discret s'écrit

$$\mathcal{A}(\xi)U(\xi) = \mathcal{B}(\xi), \quad \text{avec} \quad \mathcal{A}(\xi) \in \mathbb{R}^{n\times n},\ \mathcal{B}(\xi) \in \mathbb{R}^n, \tag{III.96}$$

et on utilise le chaos généralisé de degré 10 pour les dimensions stochastiques [2]. On arrive alors à un problème défini sur un espace de tenseur d'ordre 6 noté :

$$Au = b \tag{III.97}$$

où A est un opérateur symétrique défini positif de rang 5 et b est de rang 1. On va utiliser le PCG décrit en section III.1.2 avec des approximations dans l'ensemble hiérarchique de Tucker, avec un arbre équilibré et différents préconditionneurs.

2. Les variables log-uniformes sont exprimées en fonction de variables aléatoires uniformes. On utilise alors des bases de Legendre orthonormées pour la mesure uniforme.

III.5.3.2 Influence du préconditionneur

On applique les préconditionneurs suivants :

- P_E, l'inverse de l'opérateur moyen. Ce dernier est l'opérateur $\overline{\mathcal{A}}$ correspondant au problème (III.95) discrétisé avec une conductivité déterministe $E(\kappa)$. Avec I^μ l'opérateur identité de $\mathbb{R}^{p_\mu \times p_\mu}$, on définit donc $P_E = \overline{\mathcal{A}}^{-1} \otimes I^1 \otimes \ldots \otimes I^5$.
- $\widehat{P}_1$, l'inverse de l'approximation de rang 1 de l'opérateur.
- P_r, l'approximation de rang r de l'inverse dans sa version symétrique comme présenté en section III.4.

Les résultats sont illustrés en figures III.4, III.5 et III.6 pour une approximation de la solution avec un rang maximal de 10. Tous les préconditionneurs améliorent la convergence du PCG. Ils semblent tous être globalement aussi efficaces. Si on souhaite une plus grande précision finale, P_E semble être le meilleur. Si on souhaite un meilleur taux de convergence au départ de l'algorithme, on devra se tourner vers P_5. Au vu de la complexité de la résolution de l'équation (III.79), le meilleur choix semble être P_E. Cette tendance se vérifie si l'on regarde le résidu relatif préconditionné par P_5, en figure III.5, qui doit être proche de l'erreur vraie si $P_5 A \approx I$. Ceci se vérifie sur l'erreur relative par rapport à un élément de $\Pi_{\mathcal{H}_{15}^T(\mathcal{V})}(u)$ en figure III.6. Ce tenseur de $\Pi_{\mathcal{H}_{15}^T(\mathcal{V})}(u)$ est calculé par un algorithme de minimisations alternées sur toutes les variables du paramétrage $F_{\mathcal{H}_{15}^T(\mathcal{V})}$ de $\mathcal{H}_{15}^T(\mathcal{V})$ de la section II.3.2.4.

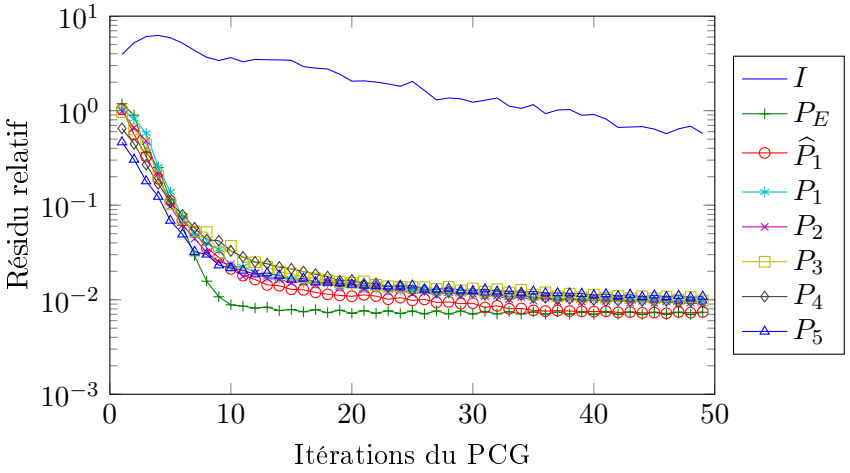


FIGURE III.4 – Résidu relatif en fonction du nombre d'itérations du PCG pour différents types de préconditionneurs pour le problème de l'équation de la chaleur.

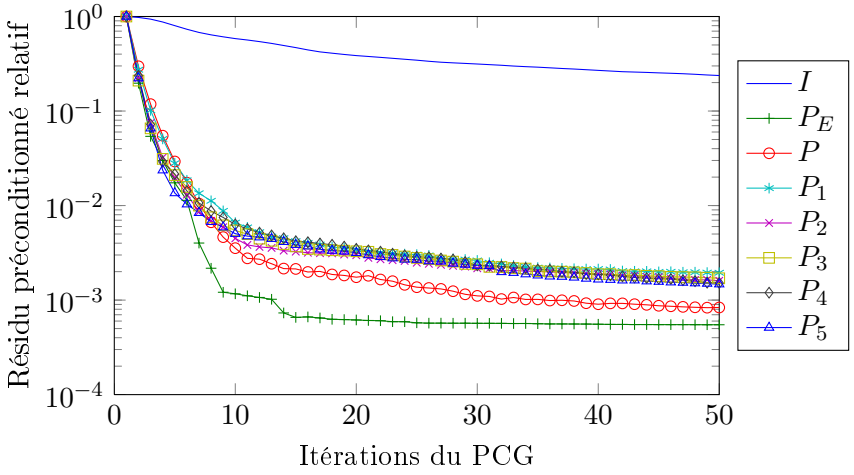


FIGURE III.5 – Résidu préconditionné relatif en fonction du nombre d'itérations du PCG pour différents types de préconditionneurs pour le problème de l'équation de la chaleur.

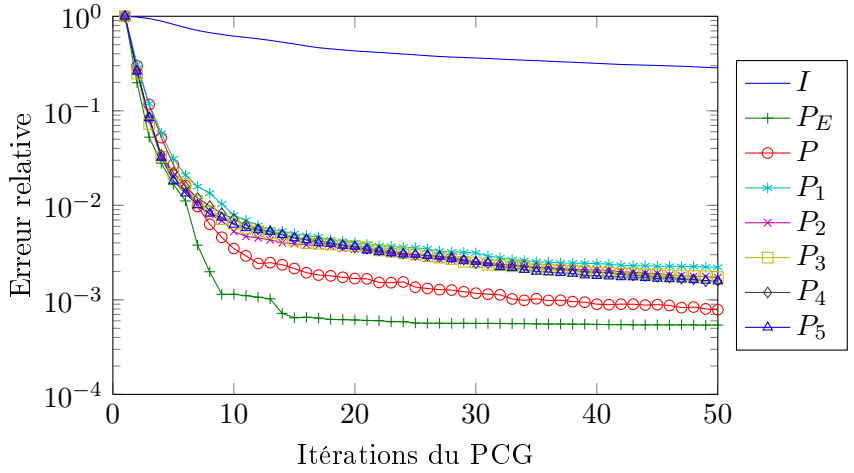


FIGURE III.6 – Erreur relative par rapport à $\Pi_{\mathcal{H}^T_{15}(\mathcal{V})}(u)$ en fonction du nombre d'itérations du PCG pour différents types de préconditionneurs pour le problème de l'équation de la chaleur.

III.5.3.3 Temps de calcul

Ici le temps de calcul est principalement dû à la construction du préconditionneur, les convergences étant quasiment identiques d'après les figures III.4, III.5, et III.6. Le temps de calcul des préconditionneurs est donné dans la table III.4.

Préconditionneur	Temps (s)
P_E	0.1
$\widehat{P}_1$	0.3
P_1	46.4
P_2	143.2
P_3	240.2
P_4	362.0
P_5	535.4

TABLE III.4 – Temps de calcul (s) pour la construction des préconditionneurs pour le problème d'équation de la chaleur.

On voit que la construction de $P_r = \sum_{i=1}^{r} \bigotimes_{\mu} P_i^{\mu}$ devient très rapidement prohibitive. Ceci provient de l'équation de Lyapunov (III.79) que l'on doit résoudre pour imposer la symétrie de P_r. L'approche utilisée pour la résoudre est la méthode de quasi-minimisation du résidu [30] (Quasi-Minimal Residual, QMR) car l'opérateur du système linéaire (III.81) n'est pas positif. Ce résultat sur les temps de calculs nous fera préférer les solveurs ne nécessitant aucune symétrie, tel que GMRES, ou encore l'utilisation d'une approximation creuse (voir diagonale) pour $\left(P_i^1\right)_{1\leq i\leq r}$ (non implémenté ici).

III.5.3.4 Influence du rang de l'approximation

On montre ici un résultat déjà illustré par Kressner et Tobler [50]. Sur la figure III.7, on a tracé le résidu relatif en fonction du nombre d'itérations du PCG pour le préconditionneur P_E pour différents rangs r_t des tenseurs hiérarchiques de Tucker : $r_t = 5$, $r_t = 10$ et $r_t = 15$.

Dans tous les cas, on observe une stagnation du résidu. La valeur limite du résidu dépend du rang de l'approximation. Plus il est élevé, plus le résidu est faible.

III.5.3.5 Influence de la variabilité

On va supposer ici que chaque conductivité des inclusions suit une loi aléatoire log-uniforme comprise entre 10 et 1000, tandis que celle de la matrice suit une loi uniforme comprise entre 1 et 100. Le but est de voir si P_E, $\widehat{P}_1$ et P_5 sont sensibles à la variabilité des paramètres aléatoires. Les solutions sont à nouveau cherchées dans l'ensemble $\mathcal{H}_{10}^{T}(\mathcal{V})$.

On observe en figure III.8 que tous les préconditionneurs sont moins efficaces que dans la section III.5.3.1. Ceci s'explique par l'extension du domaine des paramètres, la solution possède donc un contenu spectral plus étendu. De plus P_E semble toujours être le meilleur préconditionneur pour le problème proposé. On observe tout de même que P_5 améliore la convergence du résidu au départ de l'algorithme. La norme étant mesuré en résidu, ceci ne traduit pas nécessairement l'erreur vraie.

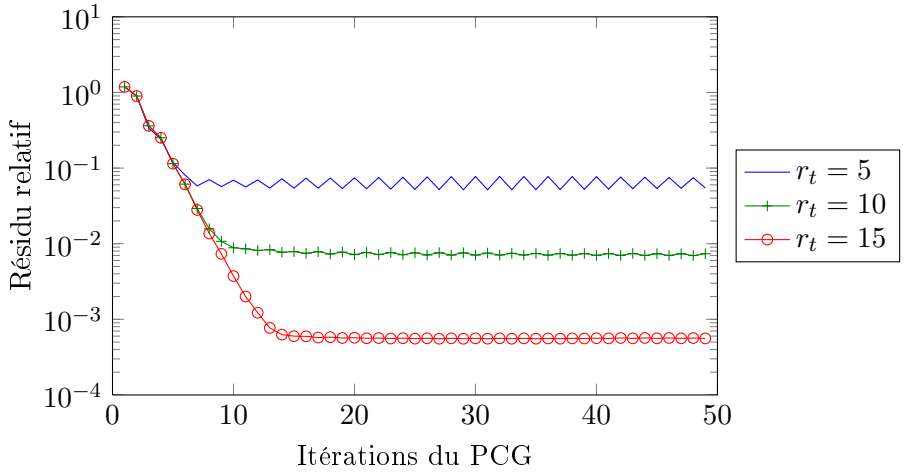


FIGURE III.7 – Résidu relatif en fonction du nombre d'itérations du PCG pour différents rangs $(r_t)_{t\in T}$ de la solution avec le préconditionneur P_E pour le problème de l'équation de la chaleur

Comme P_E est un bon point de départ pour un préconditionneur, on l'a amélioré avec l'algorithme d'approximation de rang-r de l'inverse. Pour cela on a enrichi le préconditionneur P_E avec l'algorithme glouton proposé en section III.4, avec un enrichissement construit dans le sous-espace des opérateurs symétriques. On note $\widetilde{P}_r$ l'enrichissement de rang-r du préconditionneur P_E.

Les courbes de convergence en figure III.9 nous indiquent à première vue que $\widetilde{P}_r$ détériore la qualité du préconditionneur P_E. Cependant on a mesuré l'erreur relative de la solution par rapport à un élément de $\Pi_{\mathcal{H}^T_{15}(\mathcal{V})}$. Ces résultats sont tracés en figure III.10.

On observe en fait une convergence très rapide du PCG pour le préconditionneur proposé $P_E+\widetilde{P}_5$, bien meilleure que pour les autres préconditionneurs. La stagnation de l'erreur observée pour les préconditionneurs P_E et $P_E + \widetilde{P}_5$ est due à la différence de rang entre la référence et la solution calculée.

On remarque alors que le résidu relatif préconditionné défini par

$$\frac{\left\|(P_E + \widetilde{P}_5)(Au_r - b)\right\|}{\left\|(P_E + \widetilde{P}_5)b\right\|}, \tag{III.98}$$

est un meilleur indicateur de l'erreur relative. En effet, les convergences de la figure III.11 sont très similaires à celles de l'erreur relative en figure III.10, aussi bien en convergence qu'en valeurs absolues. Cela peut s'expliquer par le fait que I et $(P_E + \widetilde{P}_5)A$ semblent avoir des propriétés spectrales très proches dans le haut du spectre, partie active du spectre lors de l'algorithme. On rappelle en effet que les approximations dans $\Pi_{\mathcal{M}_r}(u)$ sont liées a une décomposition spectrale, ce qui peut expliquer que seule la partie haute du spectre soit active, bien que numériquement on vérifie que

$$\frac{\left\|I - (P_E + \widetilde{P}_5)A\right\|}{\|I\|} \approx 47\%. \tag{III.99}$$

Ceci implique qu'il est nécessaire de mesurer de manière robuste le conditionnement de notre pré-

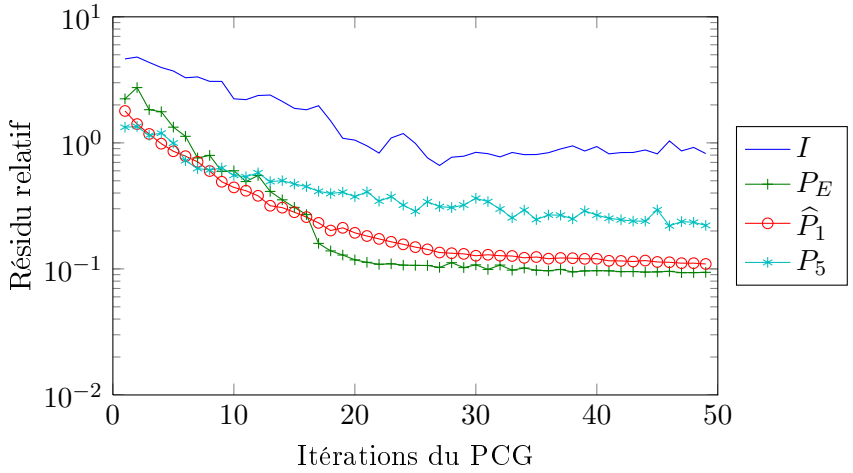


FIGURE III.8 – Résidu relatif en fonction du nombre d'itérations du PCG pour différents types de préconditionneurs pour un problème avec une grande variabilité pour le problème de l'équation de la chaleur.

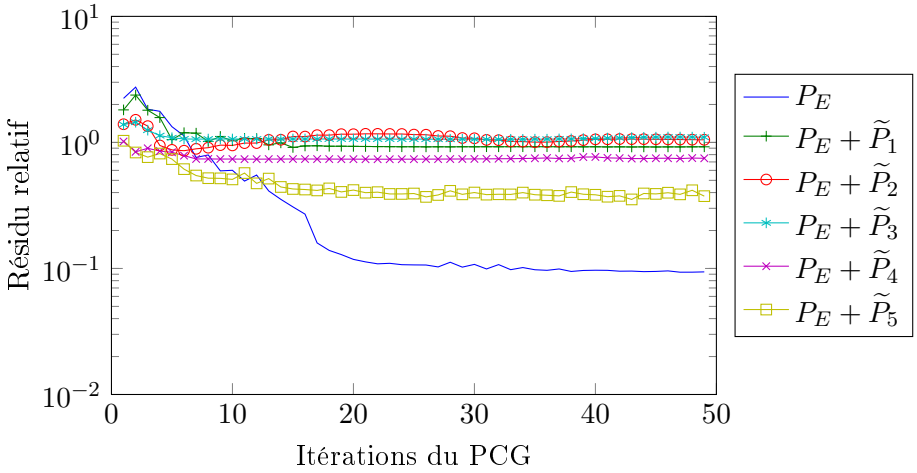


FIGURE III.9 – Résidu relatif en fonction du nombre d'itérations du PCG avec différents enrichissements du préconditionneur P_E pour un problème de la chaleur avec une grande variabilité.

conditionneur.

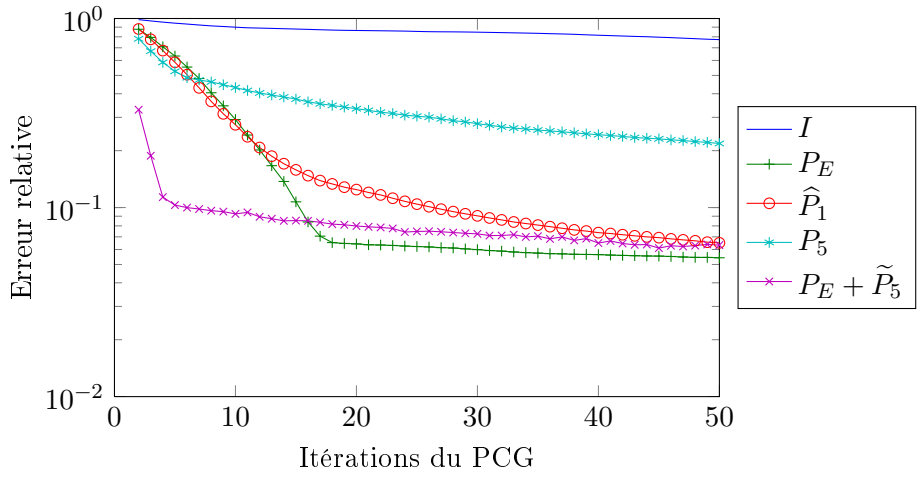


FIGURE III.10 – Erreur relative en fonction du nombre d'itérations du PCG avec différents préconditionneurs pour un problème de la chaleur avec une grande variabilité.

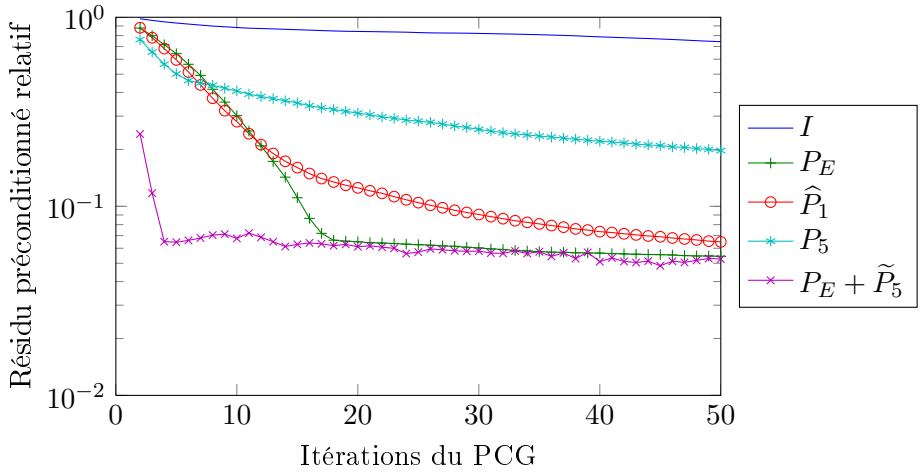


FIGURE III.11 – Résidu préconditionné relatif en fonction du nombre d'itérations du PCG avec différents préconditionneurs pour un problème de la chaleur avec une grande variabilité.

III.5.4 Problème non symétrique

III.5.4.1 Description du problème

Cet exemple est un cas non symétrique tiré de [64]. Il s'agit d'une équation de diffusion convection réaction en régime transitoire sur le domaine représenté en figure III.12. Le problème aux limites s'écrit :

$$\begin{aligned}
\frac{\partial u}{\partial t} - \nabla(\kappa \nabla u) + c(D \cdot \nabla u) + \sigma u = 0 \quad & \text{sur } \Omega \times \Omega_t \text{ avec } = \Omega_1 \cup \Omega_2, \\
u = 1 \quad & \text{sur } \Gamma_1 \times \Omega_t, \\
u = 0 \quad & \text{sur } \Gamma_2 \times \Omega_t, \\
u(t=0) = 0 \quad & \text{sur } \Omega \setminus \Gamma_1, \\
u(t=0) = 1 \quad & \text{sur } \Gamma_1, \\
-\kappa \nabla u = 0 \quad & \text{sur } (\Omega \setminus (\Gamma_1 \cup \Gamma_2)) \times \Omega_t,
\end{aligned} \tag{III.100}$$

avec $t \in \Omega_t$.

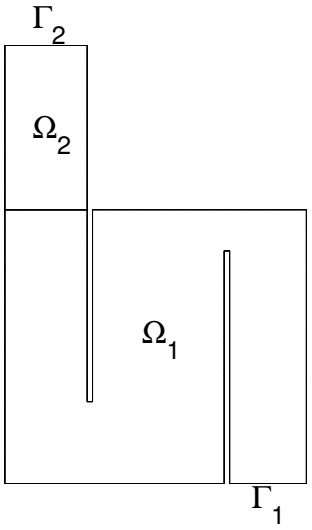


FIGURE III.12 – Géométrie du filtre à particules.

On va mener une étude paramétrique sur les variables

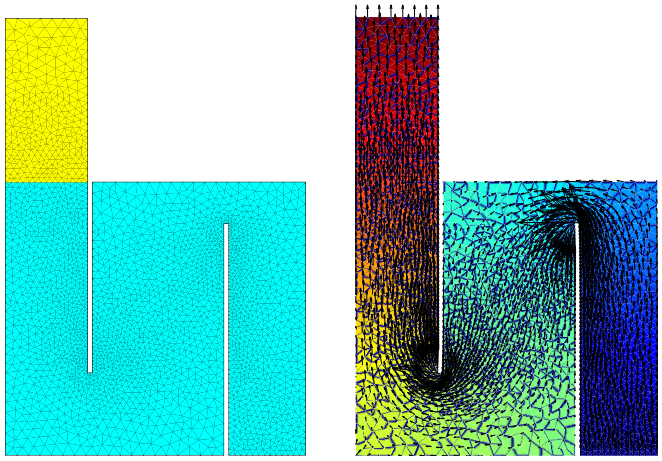
$$\begin{aligned}
\kappa = \xi_1 \in \Xi^1 = (0.01, 1), \\
c = \xi_2 \in \Xi^2 = (-3, 3), \\
\sigma = \xi_3 \in \Xi^3 = (0.5, 1.5) \quad \text{sur } \Omega_1, \\
\sigma = \xi_4 \in \Xi^4 = (9.5, 10.5) \quad \text{sur } \Omega_2.
\end{aligned} \tag{III.101}$$

D est un champ de vecteurs représentant le gradient de la solution v du problème de Poisson sur Ω

suivant :

$$\begin{aligned} \Delta v &= 0 \quad \text{sur } \Omega = \Omega_1 \cup \Omega_2, \\ \nabla v \cdot n &= 1 \quad \text{sur } \Gamma_1, \\ \nabla v \cdot n &= -1 \quad \text{sur } \Gamma_2, \\ \nabla v \cdot n &= 0 \quad \text{sur } \partial\Omega \setminus (\Gamma_1 \cup \Gamma_2). \end{aligned} \tag{III.102}$$

Pour simplifier les notations, on note $(x, t, \kappa, c, \sigma_1, \sigma_2) = (x^1, \ldots, x^6)$ et Ω^μ le domaine de la vari-



(a) Maillage (b) v et D

FIGURE III.13 – Maillage et v et champ de vecteurs D de convection.

able x^μ. Une formulation de type Galerkin Discontinue (Discontinuous Galerkin, DG) en temps est utilisée. Une partition $\left(\Omega_i^2\right)_{1\leq i\leq K}$ de Ω^2 est introduite de la forme $\Omega_i^2 = (t_i, t_{i+1})$ avec $0 = t_1 < \ldots < t_{K+1} = T$.

On introduit enfin l'espace des polynômes de degré k continus par morceaux

$$\mathcal{V}^2 = \left\{ v \in L^2(\Omega^2);\ v_{|\Omega_i^2} \in \mathbb{P}_k\left[\mathbb{R}\right] \right\} \tag{III.103}$$

servant d'espace d'approximation pour les fonctions de la variable temporelle. Pour les fonctions de la variable $x^\mu, \mu \in D \setminus \{2\}$, on va utiliser des espaces d'approximation éléments finis notés $\mathcal{V}^\mu$.

On utilise une formulation variationnelle du problème discrétisé adaptée au cadre DG sous la forme :

$$\begin{aligned} &\text{Trouver } u \in \mathcal{V} = \bigotimes_\mu \mathcal{V}^\mu \text{ tel que} \\ &a(u, v) = \ell(v), \quad \forall v \in \mathcal{V}. \end{aligned} \tag{III.104}$$

Remarque III.5.3. *Dans un cadre continu, il faut chercher la solution originellement dans*

$$H(\Lambda) = \left\{ v \in L^2(\Lambda);\ \frac{\partial v}{\partial x^\lambda} \in L^2(\Lambda), \lambda \in \{1, 2\} \right\} \tag{III.105}$$

qui est, d'après Hackbusch [38, Section 4.3.2], un espace produit tensoriel tel que

$$H(\Lambda) = H^1(\Omega^1) \otimes H^1(\Omega^2) \otimes L^2(\Omega^3) \otimes \ldots \otimes L^2(\Omega^6). \tag{III.106}$$

Pour DG, on doit travailler dans l'espace plus grand défini par

$$\widetilde{H}(\Lambda) = \left\{ v \in L^2(\Lambda);\ v_{|_{\Lambda_i}} \in H(\Lambda_i) \right\} \tag{III.107}$$

avec $\Lambda_i = \Omega^1 \otimes \Omega_i^2 \otimes \Omega^3 \otimes \ldots \otimes \Omega^6$. *On montre en annexe A que c'est aussi un espace produit tensoriel tel que*

$$\widetilde{H}(\Lambda) = H^1(\Omega^1) \otimes \widetilde{H}^1(\Omega^2) \otimes L^2(\Omega^3) \otimes \ldots \otimes L^2(\Omega^6) \tag{III.108}$$

où $\widetilde{H}^1(\Omega^2)$ *est un espace de Sobolev cassé. Tout ceci justifie rigoureusement l'utilisation de méthodes de type DG avec une approche utilisant les méthodes d'approximation de tenseurs.*

Les domaines $(\Omega^\mu)_{\mu\in\{1,3,4,5,6\}}$ sont discrétisés avec des maillages contenant respectivement 5297, 33, 60, 50 et 50 éléments linéaires. Pour Ω^2, on utilise 99 éléments de degré 0 et les maillages de $(\Omega^\mu)_{2\leq\mu\leq6}$ sont uniformes.

On arrive alors à un problème posé sur un espace de tenseurs d'ordre 6 de la forme

$$Au = b, \tag{III.109}$$

où A est un tenseur non symétrique de rang 5, et b est un tenseur de rang 5. Étant donné les propriétés de A, on va cette fois utiliser les solveurs BiCGSTAB (algorithme 7) et GMRES(5) (algorithme 8) avec troncatures dans des sous-ensembles de tenseurs.

III.5.4.2 Convergence des algorithmes

Avec les mêmes notations que pour la section III.5.3, les courbes de convergences pour le BiCGSTAB et GMRES(5) avec troncatures dans $\mathcal{H}_{10}^T(\mathcal{V})$ avec différents préconditionneurs sont illustrées plus loin, où on a utilisé les mêmes notations que pour l'application III.5.3. Cette fois P_r n'est bien sûr pas cherché dans le sous espace des opérateurs symétriques.

Comportement du solveur BiCGSTAB. Le résidu relatif en fonction du nombre d'itérations du BiCGSTAB est tracé en figure III.14 pour différents préconditionneurs. On observe tout d'abord que la convergence du BiCGSTAB avec troncature est très « chaotique » quand il est préconditionné. Néanmoins, à l'exception de P_1, les préconditionneurs fournissent un résidu relatif faible dès le départ de l'algorithme. La divergence du solveur au bout de quelques itérations implique qu'il est nécessaire de contrôler le résidu lors de l'algorithme, comme l'avait soulevé Kressner et Tobler [50].

Quant au choix du préconditionneur, on remarque qu'à partir de $r = 2$, P_r fonctionne mieux que $\widehat{P}_1$. De plus, on observe que le préconditionneur P_5 est le plus efficace de tous. Si l'on regarde maintenant l'erreur relative par rapport un élément de $\Pi_{\mathcal{H}_{12}^T}(\mathcal{V})$ sur la figure III.15, on observe que P_5 donne rapidement une bonne approximation de la solution. De plus, sur la figure III.16, on a mesuré le résidu préconditionné relatif donné par

$$\frac{\|P_5(Au_r - b)\|}{\|P_5 b\|}. \tag{III.110}$$

On observe alors que cette valeur fournit encore une fois une assez bonne estimation de l'erreur

relative (comparer avec la figure III.15), en tout cas bien meilleur que le résidu relatif.

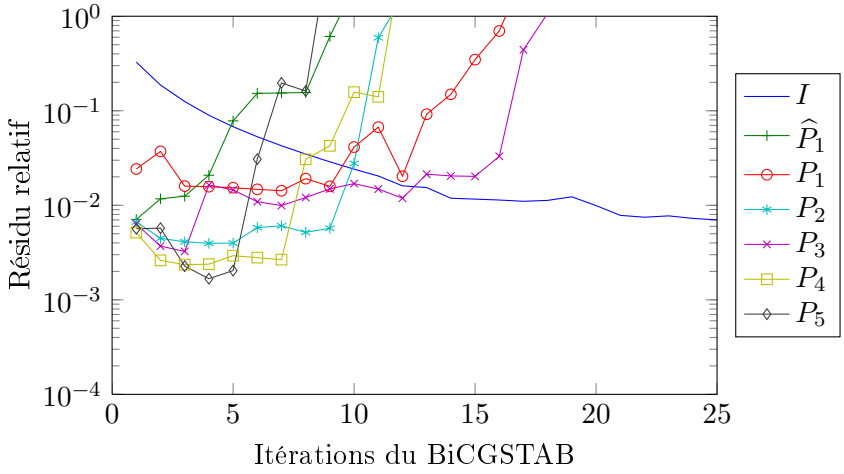


FIGURE III.14 – Résidu relatif en fonction du nombre d'itérations pour le solveur BiCGSTAB pour différents préconditionneurs.

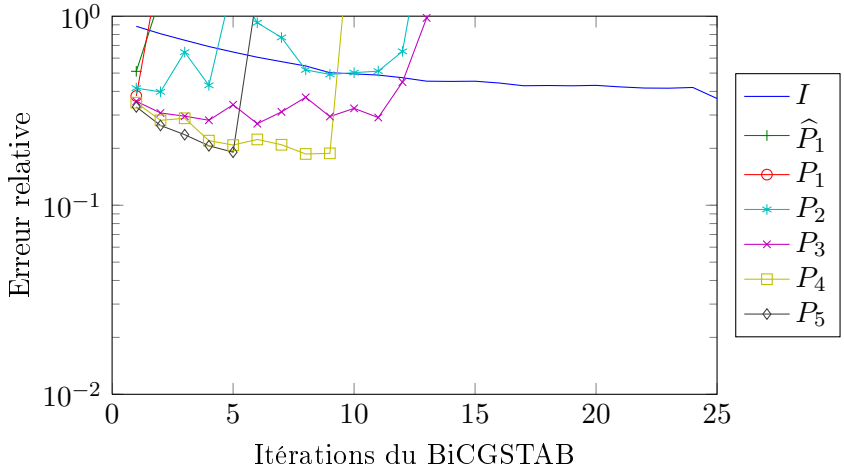


FIGURE III.15 – Erreur relative par rapport à un élément de $\mathcal{H}_{12}^T(\mathcal{V})$ en fonction du nombre d'itérations pour le solveur BiCGSTAB pour différents préconditionneurs.

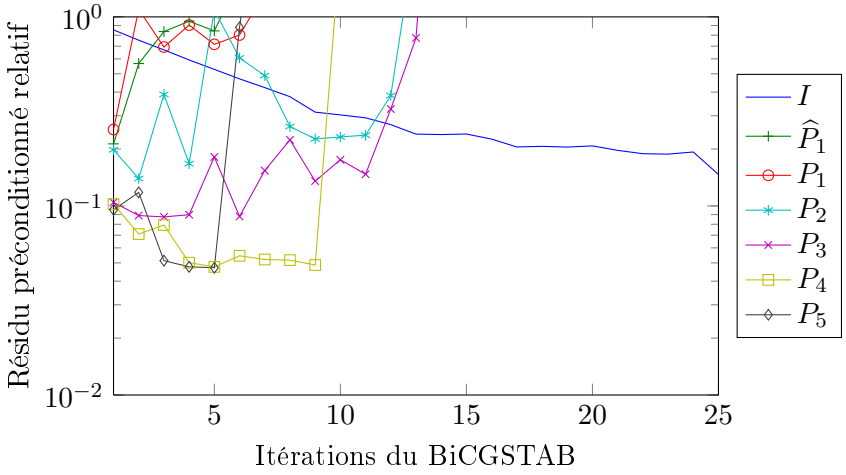


FIGURE III.16 – Résidu préconditionné relatif (avec P_5) en fonction du nombre d'itérations pour le solveur BiCGSTAB pour différents préconditionneurs.

Comportement du solveur GMRES(5). Dans la figure III.17, on a tracé le résidu relatif en fonction du nombres d'itérations de GMRES(5) pour différents préconditionneurs. Cette fois on observe que P_2 met GMRES(5) en défaut. L'algorithme GMRES classique ayant un résidu forcément décroissant, ceci est nécessairement dû aux opérations de troncatures de l'algorithme 8. On a notamment imposé ici que $\mathcal{M} = \mathcal{H}_{10}^T(\mathcal{V})$. Il en découle que la troncature $u_{k+1} \in \Pi_{\mathcal{M}}(z_{k+1})$ ne garantit plus la décroissance du résidu. Ballani et Grasedyck [7] ont proposé une démarche adaptative pour garantir la décroissance du résidu. Hormis P_2, tous les préconditionneurs P_r, $r \neq 2$ sont plus performants que $\widehat{P}_1$. On observe en particulier que l'algorithme atteint la meilleure approximation possible au bout de 2 itérations avec P_4.

On a encore mesuré l'erreur relative par rapport à un élément de $\Pi_{12}^T(\mathcal{V})$ dans la figure III.18 ainsi que le résidu préconditionné relatif $\frac{\|P_5(Au_r - b)\|}{\|P_5 b\|}$ dans la figure III.19. On observe que P_5 est le meilleur préconditionneur au départ de l'algorithme alors que P_1 donne une meilleure précision à convergence, même si la différence avec P_5 est très faible. De plus, le résidu relatif préconditionné fournit une bonne estimation de l'erreur relative sur la solution.

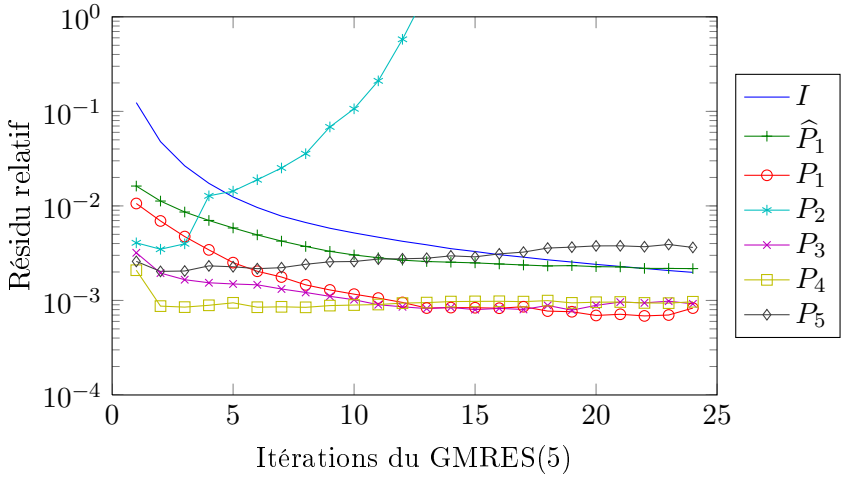


FIGURE III.17 – Résidu relatif en fonction du nombre d'itérations pour le solveur GMRES(5) pour différents préconditionneurs.

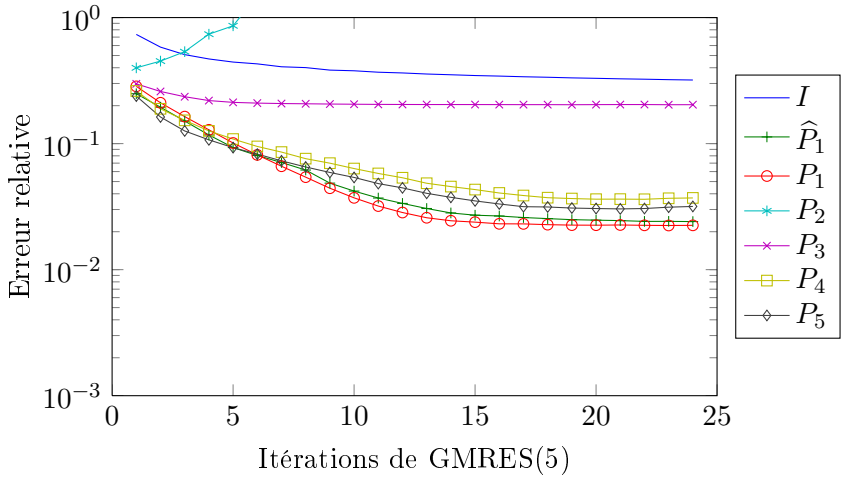


FIGURE III.18 – Résidu relatif en fonction du nombre d'itérations pour le solveur GMRES(5) pour différents préconditionneurs.

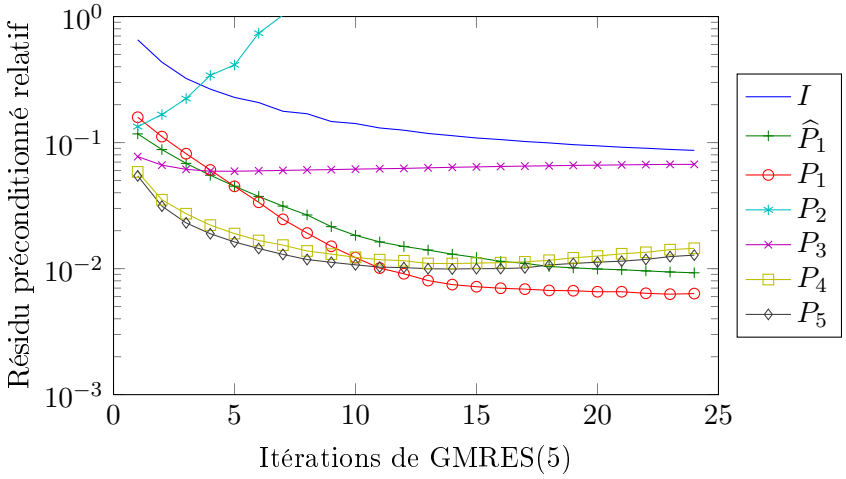


FIGURE III.19 – Résidu préconditionné relatif (avec P_5) en fonction du nombre d'itérations pour le solveur GMRES(5) pour différents préconditionneurs.

Influence des différents paramètres de GMRES A la vue des courbes de convergences de la figure III.17, on a mené l'étude, avec le préconditionneur P_3, de l'influence du rang et de la dimension maximale de l'espace de Krylov. Les nouvelles courbes de convergence sont données en figure III.20.

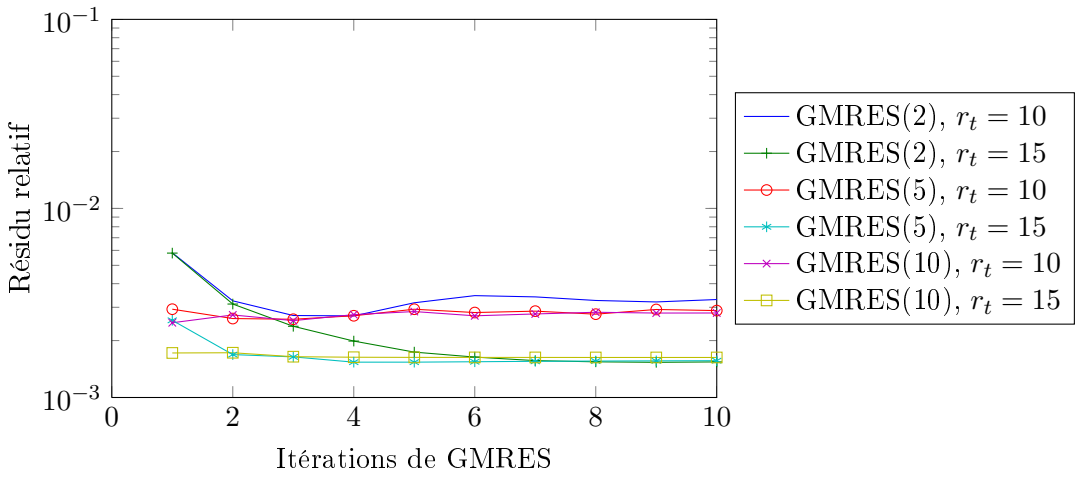


FIGURE III.20 – Résidu relatif en fonction du nombre d'itérations pour le solveur GMRES préconditionné par P_3, pour différents m et différents rangs de l'approximation.

On observe deux paquets de courbes correspondant aux rangs maximaux de la solution. Il y a peu de différences (de l'ordre de 10^{-3}) entre une solution pour un rang 10 et une solution pour un rang 15. De plus, la dimension maximale des espaces de Krylov (2, 5 ou 10) n'influe pas sur le résidu final de l'algorithme. Il en découle qu'un rang maximal de 10 pour l'approximation et l'algorithme GMRES(2) suffisent à avoir une bonne approximation de la solution en norme de résidu.

Il est nécessaire d'augmenter le rang de la solution pour avoir une meilleure approximation de u. Dans la figure III.21, on a tracé le résidu relatif en fonction du nombre d'itérations, pour un rang de 20 et le solveur GMRES(2). Cette fois le préconditionneur influe sur la convergence finale de l'algorithme et on commence à voir l'influence d'un bon préconditionneur sur GMRES. On ne captait pas ce phénomène auparavant car les différentes troncatures étaient trop restrictives et éliminaient les parties actives des préconditionneurs. On note que P_2 fait encore diverger l'algorithme, c'est pour cela qu'il n'est pas inclus dans l'étude pour un rang supérieur.

Influence de la dimension. On a cette fois utilisé un maillage plus fin du domaine spatial. La dimension de l'espace d'approximation pour la variable spatiale est passée de 2475 à 4618. Comme on a vu précédemment qu'utiliser GMRES(2) ou GMRES(5) n'influait pas sur le résultat final, on s'est limité ici à GMRES(2) pour des questions de temps de calcul. Le rang maximal de la solution est imposé à 15. Les courbes de convergences sont disponibles en figure III.22.

P_2 fait toujours diverger l'algorithme. Contrairement au problème de Poisson, ici raffiner le maillage n'augmente pas le résidu relatif que l'on obtient à convergence. La norme du résidu reste même quasiment inchangée par rapport au maillage grossier.

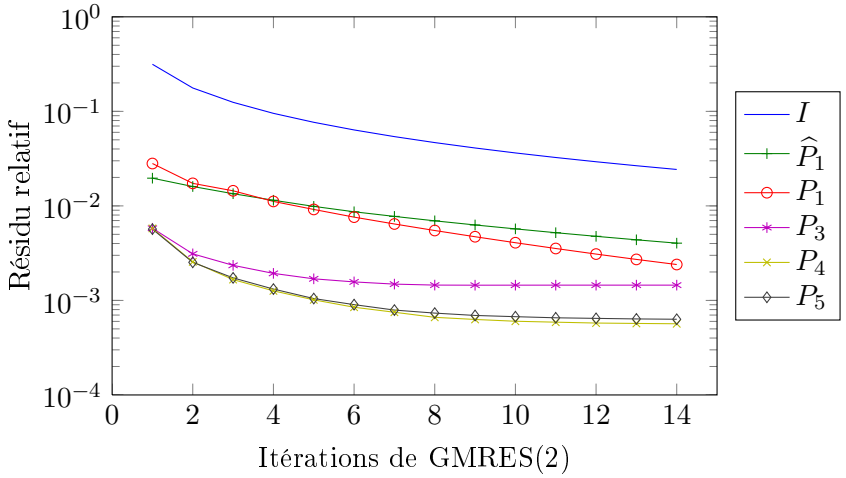


FIGURE III.21 – Résidu relatif en fonction du nombre d'itérations pour le solveur GMRES(2) pour différents préconditionneurs et un rang maximal de 20.

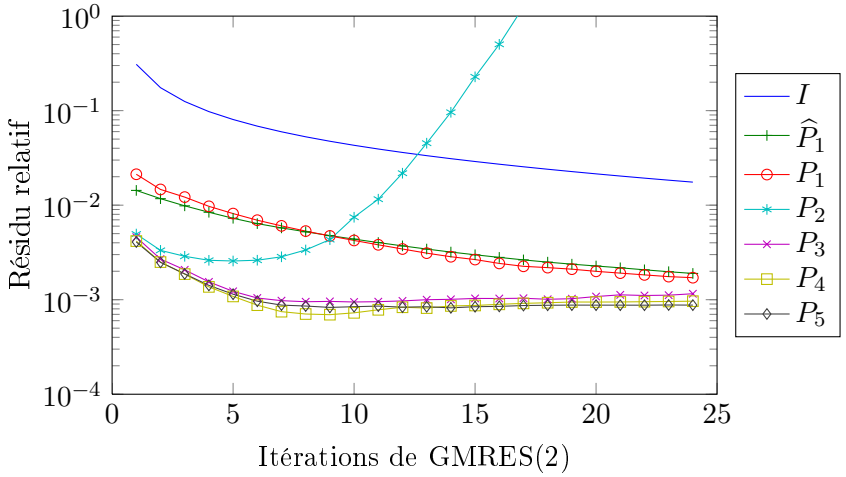


FIGURE III.22 – Résidu relatif en fonction du nombre d'itérations pour le solveur GMRES(2) pour un maillage raffiné et différents préconditionneurs.

III.5.4.3 Temps de calcul

Temps par itération en fonction du préconditionneur. On va tout d'abord s'intéresser aux temps de construction du préconditionneur. Les résultats sont donnés dans le tableau III.5.

On observe que le temps de construction du préconditionneur peut vite devenir prohibitif si l'on veut atteindre un rang élevé. Ceci provient de la structure de l'approximation de l'inverse. Chaque matrice P_i^μ du tenseur $P_r = \sum_{i=1}^r \bigotimes_\mu P_i^\mu$ possédant en effet une structure pleine. Ceci augmente fortement la complexité du calcul de l'opérateur $[WAA^*, W]_\lambda$ dans (III.62), où $W = P_r - P_{r-1} = \bigotimes_\mu P_r^\mu$. Le calcul de $\widehat{P}_1$ est quant à lui très rapide, puisqu'il nécessite des opérations moins coûteuses et une simple inversion de matrices. Cependant, il peut donner de mauvais résultats, comme pour le problème de Poisson en section III.5.2.

Préconditionneur	Temps (s)
$\widehat{P}_1$	3
P_1	68
P_2	161
P_3	279
P_4	467
P_5	632

TABLE III.5 – Temps de construction des différents préconditionneurs.

Influence des différents paramètres. Dans la table III.6, on trouve le nombre d'itérations et le temps de calcul nécessaire jusqu'à convergence de GMRES(m) en fonction de m et du rang r_t de la solution. GMRES(m) a été ici préconditionné par P_3.

m	2		5		10	
r_t	10	15	10	15	10	15
Nombre d'itér. jusqu'à stagnation	6	7	2	2	1	1
Temps (s) jusqu'à stagnation	30.6	116.6	29.2	101.2	44.9	162.0

TABLE III.6 – Temps de calcul en fonction du rang r_t et de m pour GMRES(m) préconditionné par P_3.

Comme toutes ces méthodes terminent avec une précision équivalente, le meilleur compromis est d'utiliser GMRES(2) ou GMRES(5) avec $r_t = 10$. On voit que plus m est petit, plus il est nécessaire d'ajouter des itérations jusqu'à convergence de l'algorithme.

Le temps de calcul explose avec le rang de la solution alors que la dépendance est assez faible avec la valeur de m. En effet, si maintenant on regarde le rang $r_t = 20$, avec GMRES(2), on obtient un temps de calcul de 363.5s pour 7 itérations jusqu'à stagnation. Le temps est en fait multiplié par 10 pour un rang multiplié par 2. Ceci limite à de faibles rangs l'utilisation de ces solveurs.

Enfin avec le maillage raffiné de la géométrie, pour GMRES(2) et un rang 15, l'algorithme arrive à stagnation en 7 itérations en 127s, soit à peine plus que pour le maillage de départ d'après III.6.

III.6 Résumé

Dans cette partie on a tout d'abord présenté les différentes méthodes de la littérature pour construire des approximations de solutions via les solveurs itératifs classiques. Ces derniers ont été appliqués avec succès pour des problèmes aussi bien symétriques que non symétriques.

Ensuite, une approche pour approximer un opérateur a été proposée. On a notamment montré que si un opérateur appartient à un sous-espace vectoriel $\mathcal{U} \subset \mathcal{L}(\mathcal{V})$, alors on peut chercher sa meilleure approximation sous format tensoriel directement dans ce sous-espace $\mathcal{U}$. Ceci correspond à une extension du résultat d'Espig et Hackbusch [27] aux formats de tenseurs non canoniques. Ce résultat a été appliqué au cas des sous-espaces d'opérateurs possédant des propriétés de symétries ou des structures creuses.

Finalement on a construit un nouveau préconditionneur basé sur l'approximation de l'inverse au sens d'une norme appropriée. Le préconditionneur peut être recherché sous format symétrique ou creux. Il a été utilisé avec succès avec les différents solveurs, comme préconditionneur en tant que tel, ou pour améliorer un préconditionneur déjà disponible.

Le contrôle de la convergence des algorithmes itératifs est un point délicat. La mesure du résidu en norme canonique s'avère être une très mauvaise estimation de l'erreur vraie pour des problèmes mal conditionnés. On a alors montré que l'estimation de la norme du résidu préconditionné par les préconditionneurs proposés peut-être un bon estimateur de l'erreur, à condition de choisir un préconditionneur de rang suffisamment élevé. Ceci permet un contrôle plus fin des solveurs itératifs. Idéalement, on estimera l'erreur en quantité d'intérêt, comme introduit en section V.1.

La première limite des solveurs itératifs couplés aux méthodes d'approximations de tenseurs est le manque de connaissance a priori du rang de la solution. Celui-ci est fixé dès le départ et peut ne pas être adapté à la représentation de la solution. Ballani et Grasedyck [7] ont proposé une troncature adaptative pour s'assurer de la minimisation du résidu, en faisant une simple boucle sur les rangs. Cette solution est coûteuse en temps de calcul, et n'est pour le moment appliquée que pour GMRES. L'autre problème provient des résultats de convergence théoriques de ces solveurs qui sont perdus à cause des différentes troncatures. Une bonne illustration concerne l'utilisation du préconditionneur P_2 avec GMRES(m) pour le problème III.5.4, où on observe une divergence de la solution. Ceci ne peut pas arriver sans utilisation de troncatures.

En ce qui concerne le préconditionneur proposé, il ne peut pas être calculé si une dimension $\dim(\mathcal{V}^\mu) = n_\mu$ est très grande pour un certain μ. En effet, on ne peut alors plus inverser de matrices de $\mathbb{R}^{n_\mu \times n_\mu}$. De plus, l'inverse d'une matrice creuse est généralement pleine. Il conviendrait alors d'appliquer la solution proposée en section III.4.2.2 pour rechercher des approximations creuses de l'inverse. Une piste d'amélioration serait de proposer un algorithme d'approximation adaptatif de l'inverse tel que le SPAI [36] pour le simple cadre matriciel. Il est aussi nécessaire d'étudier plus en détail l'effet du préconditionnement à droite qui a été proposé mais pas implémenté dans ce travail.

Approximations directes de solutions d'équations linéaires

Sommaire

On va présenter ici comment résoudre le système d'équations linéaires $Au = b$ avec une application directe des méthodes introduites au chapitre II. Pour cela on va chercher une approximation w de u, solution (éventuellement approchée) du problème de minimisation

$$w \in \Pi_{\mathcal{M}}(u) = \arg\min_{v \in \mathcal{M}} \|u - v\|_{\mathcal{V}}, \tag{IV.1}$$

où $\mathcal{M}$ est un sous-ensemble de tenseurs. On doit tout d'abord définir une norme appropriée pour calculer l'approximation sans connaître la solution, comme on l'a fait en section III.4.1 pour l'approximation de l'inverse de l'opérateur. On verra ensuite différentes stratégies pour minimiser $\|u - v\|_{\mathcal{V}}$, $v \in \mathcal{M}$.

L'originalité de ce travail réside tout d'abord dans la méthode de préconditionnement proposée pour ce type d'approche. De plus, on proposera une manière efficace d'approcher la solution de (IV.1) dans le cas où $\mathcal{M} = \mathcal{T}_r(\mathcal{V})$ (ensemble des tenseurs de Tucker de rang r) ou $\mathcal{M} = \mathcal{H}_r^T(\mathcal{V})$ (ensemble des tenseurs hiérarchiques de Tucker de rang r) via l'introduction de mises à jour des constructions gloutonnes.

IV.1 Choix d'une norme appropriée

Soit $F_{\mathcal{M}} : \mathcal{F}^1_{\mathcal{M}} \times \ldots \times \mathcal{F}^m_{\mathcal{M}} \to \mathcal{V}$ un paramétrage de $\mathcal{M}$, c'est-à-dire $\mathcal{M} = \mathrm{Im}(F_{\mathcal{M}})$ (voir section II.2.4). Le problème de minimisation IV.1 est alors équivalent au problème

$$\min_{(f^1,\ldots,f^m)\in\mathcal{F}^1_{\mathcal{M}}\times\ldots\times\mathcal{F}^m_{\mathcal{M}}} \left\|u - F_{\mathcal{M}}(f^1,\ldots,f^m)\right\|^2_{\mathcal{V}}, \tag{IV.2}$$

avec $v = F_{\mathcal{M}}(f^1,\ldots,f^m)$. Pour trouver une solution à ce problème de minimisation, on utilise un algorithme des minimisations alternées, présenté en section II.3.2.1. On s'intéresse ici à une étape de l'algorithme, celle de la minimisation sur un espace de paramètres $\mathcal{F}^\lambda_{\mathcal{M}}$, où l'on doit trouver $f^\lambda \in \mathcal{F}^\lambda_{\mathcal{M}}$ tel que

$$f^\lambda = \arg\min_{h\in\mathcal{F}^\lambda_{\mathcal{M}}} \left\|u - F_{\mathcal{M}}(f^1,\ldots,f^{\lambda-1},h,f^{\lambda+1},\ldots,f^m)\right\|^2_{\mathcal{V}}. \tag{IV.3}$$

Pour trouver f^λ, la première étape consiste à proposer une norme $\|\cdot\|_{\mathcal{V}}$ appropriée qui va dépendre des propriétés de l'opérateur A du système d'équations.

IV.1.1 Cas symétrique défini positif

Si $A \in \mathcal{L}(\mathcal{V})$ est un opérateur symétrique défini positif, alors $\langle A\cdot,\cdot\rangle$ définit un produit scalaire. On note $\|\cdot\|_A$ sa norme associée. L'idée est alors de prendre $\|\cdot\|_{\mathcal{V}} = \|\cdot\|_A$. On note $\delta f = (f^1,\ldots,f^{\lambda-1},\delta f^\lambda,f^{\lambda+1},\ldots,f^m$ et $f = (f^1,\ldots,f^m)$. Alors, la solution f^λ de (IV.3) satisfait

$$\langle F_{\mathcal{M}}(\delta f), b - AF_{\mathcal{M}}(f)\rangle = 0, \quad \forall \delta f^\lambda \in \mathcal{F}^\lambda_{\mathcal{M}}. \tag{IV.4}$$

IV.1.2 Cas non symétrique

Si $\langle A\cdot,\cdot\rangle$ ne définit plus un produit scalaire, on va considérer $\langle A^*A\cdot,\cdot\rangle$ avec la norme associée $\|\cdot\|_{A^*A}$. Avec les notations définies en IV.1.1, la solution f^λ de (IV.3) vérifie

$$\langle F_{\mathcal{M}}(\delta f), A^*b - A^*AF_{\mathcal{M}}(f)\rangle = 0, \quad \forall \delta f^\lambda \in \mathcal{F}^\lambda_{\mathcal{M}}, \tag{IV.5}$$

ou de façon équivalente

$$\langle AF_{\mathcal{M}}(\delta f), b - AF_{\mathcal{M}}(f)\rangle = 0, \quad \forall \delta f^\lambda \in \mathcal{F}^\lambda_{\mathcal{M}}. \tag{IV.6}$$

Remarque IV.1.1. *La « Proper Generalized Decomposition » (PGD) progressive classique est le cas particulier où $\mathcal{M} = \mathcal{C}_1(\mathcal{V})$ et où on adopte une construction gloutonne de l'approximation (voir section II.3.3). Le cas $\|\cdot\| = \|\cdot\|_A$ est appelée PGD de Galerkin, $\|\cdot\| = \|\cdot\|_{A^*A}$ est appelée PGD en minimum de résidu [2, 3, 16, 60, 61].*

IV.1.3 Préconditionnement pour l'approximation directe

Les approches précédentes ne tiennent pas compte du conditionnement de l'opérateur A définit par $\text{cond}(A) = \|A\| \left\|A^{-1}\right\|$ où $\|\cdot\|$ est la norme d'opérateur associée à la norme vectorielle $\|\cdot\|$

$$\|A\| = \sup_{v \in \mathcal{V}\setminus\{0\}} \frac{\|Av\|}{\|v\|}. \tag{IV.7}$$

L'introduction d'une perturbation δb de b conduit à la résolution du système $A(u+\delta u) = b + \delta b$ où δu est une perturbation de u. On a alors la borne

$$\frac{\|\delta u\|}{\|u\|} \leq \text{cond}(A) \frac{\|\delta b\|}{\|b\|}. \tag{IV.8}$$

De même si on introduit une perturbation δA de A, la solution de l'équation $(A+\delta A)(u+\delta u) = b$ vérifie

$$\frac{\|\delta u\|}{\|u+\delta u\|} \leq \text{cond}(A) \frac{\|\delta A\|}{\|A\|}. \tag{IV.9}$$

Ainsi plus $\text{cond}(A)$ est faible, plus la borne est restrictive et les algorithmes sont moins sensibles aux erreurs d'approximation numérique. On a $\min_{A \in \mathcal{L}(\mathcal{V})} \text{cond}(A) = 1$, le minimum étant atteint avec l'opérateur identité I sur $\mathcal{V}$. Les préconditionneurs P et Q, relativement à gauche et à droite, sont idéaux s'ils vérifient $PA = I$ ou $AQ = I$, donc si $P = Q = A^{-1}$. Calculer A^{-1} étant tout aussi difficile que le problème initial, à savoir de résoudre $Au = b$, en pratique on va construire P et/ou Q tels que $PAQ \approx I$. La méthode utilisée alors pour construire des préconditionneurs de A est présentée dans la section III.4. En pratique PAQ devient un opérateur non symétrique même si A était symétrique. On utilise alors l'approche de la section IV.1.2 pour approcher $Q^{-1}u = \widetilde{u}$ au sens de la norme $\|\cdot\|_{Q^*A^*P^*PAQ}$ associée au produit scalaire $\langle\cdot,\cdot\rangle_{Q^*A^*P^*PAQ} = \langle Q^*A^*P^*PAQ\cdot,\cdot\rangle$:

$$\min_{v\in\mathcal{M}} \|\widetilde{u} - v\|_{Q^*A^*P^*PAQ} \Leftrightarrow \min_{v\in Q\mathcal{M}} \|u - v\|_{A^*P^*PA}. \tag{IV.10}$$

Une minimisation sur la dimension λ définit la solution f^λ par

$$\begin{aligned} & \langle F_{\mathcal{M}}(\delta f), (Q^*A^*P^*P)(AQ\widetilde{u} - AQF_{\mathcal{M}}(f))\rangle = 0, \quad \forall \delta f^\lambda \in \mathcal{F}^\lambda_{\mathcal{M}}, \\ \Leftrightarrow \quad & \langle F_{\mathcal{M}}(\delta f), (Q^*A^*P^*P)(b - AQF_{\mathcal{M}}(f))\rangle = 0, \quad \forall \delta f^\lambda \in \mathcal{F}^\lambda_{\mathcal{M}}. \end{aligned} \tag{IV.11}$$

IV.2 Construction gloutonne et stratégies de mise à jour

IV.2.1 Cadre abstrait

On considère une construction gloutonne d'une séquence d'approximations $(u_r)_{r\in\mathbb{N}}$ de u définie par :

$$\begin{cases} u_0 = 0, \\ v_r \in \Pi_{\mathcal{M}}(u - u_{r-1}), \quad \forall r \in \mathbb{N}^*, \\ u_r = u_{r-1} + v_r, \quad \forall r \in \mathbb{N}^*. \end{cases} \tag{IV.12}$$

On introduit un paramétrage $F_{\mathcal{M}} : \mathcal{F}^1_{\mathcal{M}} \times \ldots \times \mathcal{F}^m_{\mathcal{M}} \to \mathcal{V}$ tel que $\mathcal{M} = \mathrm{Im}(F_{\mathcal{M}})$. A l'étape $r \in \mathbb{N}^*$, on introduit un nouvel ensemble $\mathcal{M}_r$ et un paramétrage associé $G_{\mathcal{M}_r}$ tel que $\mathcal{M}_r = \mathrm{Im}(G_{\mathcal{M}_r})$. Les paramétrages sont définis tels que

$$\begin{aligned} v_r &= F_{\mathcal{M}}(f^1_r, \ldots, f^m_r), \quad \forall r \in \mathbb{N}^*, \\ u_r &= G_{\mathcal{M}_r}(g^1_r, \ldots, g^{m_r}_r), \quad \forall r \in \mathbb{N}^*. \end{aligned} \tag{IV.13}$$

On peut typiquement prendre pour $F_{\mathcal{M}}$ et $G_{\mathcal{M}_r}$ les applications présentées en sections II.2.4 et II.3.2. La construction gloutonne avec mises à jour consiste à ajouter une étape dans (IV.12). La séquence $(u_r)_{r\in\mathbb{N}}$ est définie par

$$\begin{cases} u_0 = 0, \\ v_r \in \Pi_{\mathcal{M}}(u - u_{r-1}), \quad \forall r \in \mathbb{N}^*, \\ \widetilde{u}_r = u_{r-1} + v_r = G_{\mathcal{M}_r}(g^1_r, \ldots, g^{m_r}_r), \quad \forall r \in \mathbb{N}^*, \\ u_r \in \pi_{\mathcal{M}_r}(u - u_{r-1}), \quad \forall r \in \mathbb{N}^*, \end{cases} \tag{IV.14}$$

avec $\pi_{\mathcal{M}_r}(v)$ pouvant être

- la projection $\pi_{\mathcal{M}_r}(v) = \mathcal{P}_{\mathcal{M}_r}(v)$ de v sur $\mathcal{M}_r$ (au sens du produit scalaire $\langle\cdot,\cdot\rangle_{\mathcal{V}}$) si $\mathcal{M}_r$ est un espace vectoriel (cas $m_r = 1$).
- une solution d'un problème de meilleur approximation dans une variété $\mathcal{M}_r$: $\pi_{\mathcal{M}_r}(v) \in \Pi_{\mathcal{M}_r}(v)$ (si $\mathcal{M}_r$ est fermé)
- un élément de $\mathcal{M}_r$ obtenu après une succession de minimisations.

Dans tous les cas $\mathcal{M}_r$ est paramétré par $G_{\mathcal{M}_r}$ et u_r est construit à partir d'une minimisation alternée sur

$$(g^1, \ldots, g^{m_r}) \to \left\| u - G_{\mathcal{M}_r}(g^1, \ldots, g^{m_r}) \right\|_{\mathcal{V}}. \tag{IV.15}$$

Ainsi on est certain d'avoir l'inégalité

$$\|u - u_r\|_{\mathcal{V}} \leq \|u - \widetilde{u}_r\|_{\mathcal{V}}. \tag{IV.16}$$

$\mathcal{M}_r$ (et donc u_r) est en fait construit à partir des informations accumulées lors de la construction des familles $(v_i)_{1\leq i\leq r}$, $(\widetilde{u}_i)_{1\leq i\leq r-1}$ et $(u_i)_{1\leq i\leq r-1}$. La mise à jour sera d'autant plus profitable que l'approximation sera améliorée pour un faible coût de mise à jour. Finalement, l'algorithme de construction est proposé dans l'algorithme 9 et sa convergence est garantie selon Falcó et Nouy [29] seulement si l'algorithme alterné fournit un minimum dans $\mathcal{M}$.

Exemple IV.2.1 ($\mathcal{M} = \mathcal{C}_1(\mathcal{V})$ et $\mathcal{M}_r = \mathrm{span}\{v_i;\ 1 \leq i \leq r\}$). *Le paramétrage associé à* $v_r = \bigotimes_\mu v^\mu_r$ *est donné par*

$$v_r = F_{\mathcal{M}}(v^1_r, \ldots, v^d_r) = \bigotimes_\mu v^\mu_r, \tag{IV.17}$$

avec donc $m = d$ *et* $\mathcal{F}^\mu_{\mathcal{M}} = \mathcal{V}^\mu$, $1 \leq \mu \leq d$. *On suppose que* $\dim(\mathcal{M}_r) = r$ *(i.e.* $(v_i)_{1\leq i\leq r}$ *est une famille libre). Cette hypothèse est justifiée par la proposition IV.2.4 énoncée plus loin. On pose*

Algorithme 9: Construction gloutonne avec mises à jour

Données : $A \in \mathcal{L}(\mathcal{V}),\ b \in \mathcal{V}$
Résultat : $u_r \in \mathcal{V}$
$u_0 = 0$;
pour $r = 1, \dots, r_{max}$ **faire**
  Initialiser $(f^1, \dots, f^m)$ aléatoirement;
  $v_r = F_{\mathcal{M}}(f^1, \dots, f^m)$;
  tant que v_r *n'a pas convergé* **faire**
    pour $\lambda = 1, \dots, m$ **faire**
      Calculer $f^\lambda = \arg\min_{h \in \mathcal{F}^\lambda_{\mathcal{M}}} \left\| u - F_{\mathcal{M}}(f^1, \dots, f^{\lambda-1}, h, f^{\lambda+1}, \dots, f^m) \right\|_{\mathcal{V}}$;
      $v_r = F_{\mathcal{M}}(f^1, \dots, f^m)$;
    fin
  fin
  $u_r = u_{r-1} + v_r$;
  Identifier $(g^1, \dots, g^{m_r})$ tel que $u_r = G_{\mathcal{M}_r}(g^1, \dots, g^{m_r})$;
  pour $k = 1, \dots, k_{max}$ **faire**
    pour $\lambda = 1, \dots, m_r$ **faire**
      Calculer $g^\lambda = \arg\min_{h \in \mathcal{G}^\lambda_{\mathcal{M}_r}} \left\| u - G_{\mathcal{M}_r}(g^1, \dots, g^{\lambda-1}, h, g^{\lambda+1}, \dots, g^{m_r}) \right\|_{\mathcal{V}}$;
      $u_r = G_{\mathcal{M}_r}(g^1, \dots, g^{m_r})$;
    fin
  fin
fin
retourner u_r;

$m_r = 1$ *et* $\mathcal{G}^1_r = \mathbb{R}^r$. *Le second paramétrage est alors*

$$\widetilde{u}_r = \sum_{i=1}^{r} \widetilde{\alpha}_i v_i = G_{\mathcal{M}_r}((\widetilde{\alpha}_i)_{1 \le i \le r}). \tag{IV.18}$$

$\mathcal{M}_r$ *étant un espace vectoriel,* $\pi_{\mathcal{M}_r}(u) = \mathcal{P}_{\mathcal{M}_r}(u)$ *est une projection. La famille* $(\alpha_i)_{1 \le i \le r}$ *telle que* $u_r = G_{\mathcal{M}_r}((\alpha_i)_{1 \le i \le r}) = \pi_{\mathcal{M}_r}(u)$ *est finalement solution du système linéaire*

$$\sum_{i=1}^{r} \langle v_i, v_j \rangle_{\mathcal{V}} \alpha_i = \langle u, v_j \rangle_{\mathcal{V}}, \quad \forall j \in \{1, \dots, r\}. \tag{IV.19}$$

Exemple IV.2.2 ($\mathcal{M} = \mathcal{C}_1(\mathcal{V})$ et $\mathcal{M}_r = \mathcal{C}_r(\mathcal{V})$). *Comme dans l'exemple IV.2.1 on a* $\mathcal{F}^\mu_{\mathcal{M}} = \mathcal{V}^\mu$, $\forall \mu \in D$, *et*

$$v_r = F_{\mathcal{M}}(v_r^1, \dots, v_r^d) = \bigotimes_{\mu} v_r^\mu. \tag{IV.20}$$

On note

$$\widetilde{u}_r = \sum_{i=1}^{r} \widetilde{v}_i = \sum_{i=1}^{r} \bigotimes_{\mu} \widetilde{v}_i^\mu, \tag{IV.21}$$

avec $\widetilde{v}_r = v_r$ *et* $u_{r-1} = \sum_{i=1}^{r-1} \widetilde{v}_i$. *Les ensembles de paramètres choisis sont alors* $\mathcal{G}^\mu_r = (\mathcal{V}^\mu)^r$, *avec*

$m_r = d$, *et*

$$\widetilde{u}_r = G_{\mathcal{M}_r}(\widetilde{g}^1, \ldots, \widetilde{g}^d) \quad \textit{avec} \quad \widetilde{g}^\mu = (\widetilde{v}_1^\mu, \ldots, \widetilde{v}_r^\mu) \in (\mathcal{V}^\mu)^r\,,\ \forall \mu \in D. \tag{IV.22}$$

Cette fois $\mathcal{M}_r$ n'est pas un ensemble fermé. Le problème de meilleure approximation peut ne pas avoir de solution. On définit alors $\pi_{\mathcal{M}_r}(u)$ par une succession de minimisations sur les espaces de paramètres $(\mathcal{G}_r^\mu)_{\mu\in D}$. Cela revient à effectuer des itérations d'un algorithme des minimisations alternées sur $\mathcal{M}_r = \mathcal{C}_r(\mathcal{V})$ en partant d'une initialisation $\widetilde{u}_r$.

Exemple IV.2.3 (Mise à jour dans un autre format)**.** *La construction gloutonne peut servir de bonne initialisation à une approximation par minimisations alternées dans un autre format. L'idée est d'utiliser la mise à jour avec $\pi_{\mathcal{M}_r}(u) \in \Pi_{\mathcal{M}_r}(u)$ où $\mathcal{M}_r$ est un sous ensemble de tenseurs contenant $\mathcal{M} + \mathcal{M} + \ldots + \mathcal{M}$, $\mathcal{M}$ sommé r fois, tel que $w_r \in \mathcal{M}_r$. On peut utiliser par exemple $\mathcal{M} = \mathcal{C}_1(\mathcal{V})$ et $\mathcal{M}_r = \mathcal{H}_r^T(\mathcal{V})$ ou $\mathcal{M}_r = \mathcal{T}_r(\mathcal{V})$. On utilise alors le paramétrage de $\mathcal{M}_r$ donné par $G_{\mathcal{M}_r} = F_{\mathcal{M}_r}$, explicité dans les sections II.2.4 et II.3.2.*

IV.2.2 Mise à jour sur un sous-espace vectoriel

On s'intéresse au cas particulier où $\mathcal{M}_r$ est un espace vectoriel. Ainsi $m_r = 1$ et $\pi_{\mathcal{M}_r} = \mathcal{P}_{\mathcal{M}_r}$ est une projection. L'algorithme glouton avec mises à jour correspond alors à la construction progressive d'un espace réduit $\mathcal{M}_r$ pour la projection de la solution u. C'est alors une alternative aux constructions par espaces de Krylov. Ici les $\mathcal{M}_r$ seront contenus dans des ensembles de tenseurs de faible rang.

Pour être consistant avec les méthodes de projection présentées en section III.2, on va noter $\mathcal{U}_r = \mathcal{M}_r$ le sous espace de projection et $\mathcal{W}_r$ l'espace orthogonal au résidu. On remarque alors qu'en choisissant $\|\cdot\|_\mathcal{V} = \|\cdot\|_A$ on a $\mathcal{W}_r = \mathcal{U}_r$ comme pour l'algorithme FOM. Si la norme vérifie $\|\cdot\|_\mathcal{V} = \|\cdot\|_{A^*A}$, on a $\mathcal{W}_r = A\mathcal{U}_r$, comme pour GMRES.

On va tout d'abord établir un résultat sur la stagnation des espaces $(\mathcal{U}_r)_{r\in\mathbb{N}^*}$.

Proposition IV.2.4. *Avec les notations de la section IV.2.1, on suppose que $\mathcal{M}$ vérifie*

- $\operatorname{span}\{\mathcal{M}\} = \mathcal{V}$,
- $\lambda\mathcal{M} = \mathcal{M},\ \forall \lambda \in \mathbb{R}^*$,
- $0 \in \mathcal{M}$.

On suppose de plus que $v_r \in \mathcal{U}_r,\ \forall r \in \mathbb{N}^*$. S'il existe $s \in \mathbb{N}^*$ tel que $\mathcal{U}_s = \mathcal{U}_{s+1}$ alors $u = u_s$.

Démonstration. Si $\mathcal{U}_s = \mathcal{U}_{s+1}$ alors on a par construction

$$u_s = \arg\min_{v\in\mathcal{U}_s} \|u - v\|_\mathcal{V} = \arg\min_{v\in\mathcal{U}_{s+1}} \|u - v\|_\mathcal{V} = u_{s+1}. \tag{IV.23}$$

Par définition de v_{s+1} et comme $0 \in \mathcal{M}$, on a

$$\|u - u_s - v_{s+1}\|_\mathcal{V} = \min_{v\in\mathcal{M}} \|u - u_s - v\|_\mathcal{V} \leq \|u - u_s\|_\mathcal{V}\,. \tag{IV.24}$$

Par hypothèse, on a $v_{s+1} \in \mathcal{U}_{s+1} = \mathcal{U}_s$ et donc

$$\|u - u_s - v_{s+1}\|_\mathcal{V} \geq \min_{v\in\mathcal{U}_s} \|u - v\|_\mathcal{V} = \|u - u_s\|_\mathcal{V}\,. \tag{IV.25}$$

Les deux inégalités (IV.24) et (IV.25) impliquent l'égalité

$$\|u - u_s\|_{\mathcal{V}} = \|u - u_s - v_{s+1}\|_{\mathcal{V}}\,. \tag{IV.26}$$

De plus, $\mathcal{U}_s$ étant un espace vectoriel, il existe un unique élément $u_s \in \mathcal{U}_s$ tel que

$$\|u - u_s\|_{\mathcal{V}} = \|u - u_s - v_{s+1}\|_{\mathcal{V}} = \min_{v \in \mathcal{U}_s} \|u - v\|_{\mathcal{V}}\,. \tag{IV.27}$$

On en déduit que $u_s = u_s + v_{s+1}$ et donc $v_{s+1} = 0$. On a donc

$$\|u - u_s\|_{\mathcal{V}} = \min_{v \in \mathcal{M}} \|u - u_s - v\|_{\mathcal{V}}\,. \tag{IV.28}$$

Par conséquent, on a l'inégalité

$$\|u - u_s - \lambda v\|_{\mathcal{V}}^2 \geq \|u - u_s\|_{\mathcal{V}}^2\,, \quad \forall \lambda \in \mathbb{R},\ \forall v \in \mathcal{M}, \tag{IV.29}$$

qui s'écrit sous forme développée

$$\lambda^2 \|v\|_{\mathcal{V}}^2 - 2\lambda \langle u - u_s, v\rangle_{\mathcal{V}} = 0, \quad \forall \lambda \in \mathbb{R},\ \forall v \in \mathcal{M}. \tag{IV.30}$$

On en déduit que

$$\langle u - u_s, v\rangle_{\mathcal{V}} = 0,\ \forall v \in \mathcal{M}. \tag{IV.31}$$

Par linéarité du produit scalaire par rapport à la seconde variable on a

$$\langle u - u_s, v\rangle_{\mathcal{V}} = 0,\ \forall v \in \mathcal{V} = \operatorname{span}\{\mathcal{M}\}\,, \tag{IV.32}$$

et donc $u = u_s$. □

Remarque IV.2.5. *D'après Saad [69], on a un résultat similaire à la proposition IV.2.4 sur la stagnation des espaces de Krylov. En effet si $\mathcal{K}_s(A,b) = \mathcal{K}_{s+1}(A,b)$, sans autre hypothèse, alors $u_s = u \in \mathcal{K}_s(A,b)$.*

Remarque IV.2.6. *Le résultat IV.2.4 est valable dans un cadre général d'approximation de tenseurs, pas seulement pour la résolution d'équations linéaires.*

IV.2.3 Projection sur un sous-espace définissant un espace produit tensoriel

Avec $\mathcal{M} = \mathcal{C}_1(\mathcal{V})$, à partir de $(v_i)_{1\leq i\leq r} \in (\mathcal{M})^r$, il est possible de construire un espace bien plus riche que $\operatorname{span}\left\{(v_i)_{1\leq i\leq r}\right\}$ utilisé dans l'exemple IV.2.1. Avec $v_i = \bigotimes_\mu v_i^\mu$, $v_i^\mu \in \mathcal{V}^\mu$, on peut définir l'espace $\mathcal{U}_r \subset \mathcal{V}$ par

$$\mathcal{U}_r = \bigotimes_\mu \mathcal{U}_r^\mu \quad \text{avec} \quad \mathcal{U}_r^\mu = \operatorname{span}\left\{(v_i^\mu)_{1\leq i\leq r}\right\} \subset \mathcal{V}_r^\mu. \tag{IV.33}$$

On note $r_\mu = \dim(\mathcal{U}_r^\mu) \leq r$ et $(w_i^\mu)_{1\leq i\leq r_\mu}$ une base de $\mathcal{U}_r^\mu$. On note l'ensemble de multi-indices $\mathcal{I} = \{(i_1,\dots,i_d);\ 1\leq i_\mu \leq r_\mu,\ \forall \mu \in D\}$. Une base $(w_i)_{i\in\mathcal{I}}$ telle que $w_i = \bigotimes_\mu w_{i_\mu}^\mu$ de $\mathcal{U}_r$ est obtenue par tensorisation de bases. On choisit alors $\mathcal{M}_r = \mathcal{U}_r$ et $\pi_{\mathcal{M}_r}$ est donc une projection $\mathcal{P}_{\mathcal{M}_r} : \mathcal{V} \to \mathcal{M}_r$ définie par :

$$\begin{aligned}\mathcal{P}_{\mathcal{U}_r}(u) &= \arg\min_{\beta\in\mathbb{R}^{r_1\times\dots\times r_d}} \left\| u - \sum_{i\in\mathcal{I}} \beta_i \bigotimes_\mu w_{i_\mu}^\mu \right\| \\ &= \sum_{i\in\mathcal{I}} \alpha_i \bigotimes_\mu w_{i_\mu}^\mu.\end{aligned} \tag{IV.34}$$

où α est solution du système linéaire

$$\sum_{i\in\mathcal{I}} \langle w_j, w_i\rangle\, \alpha_i = \langle u, w_j\rangle\,, \quad \forall j \in \mathcal{I}, \tag{IV.35}$$

Remarque IV.2.7. *On pourrait considérer un ensemble d'indices $\widetilde{\mathcal{I}} \subsetneq \mathcal{I}$.*

Cette stratégie revient en fait à la construction gloutonne d'espaces réduits $\mathcal{U}_r^\mu$ pour la projection de la solution. Ceci revient à la construction progressive d'un modèle réduit. En pratique la résolution du système (IV.35) est lourde à effectuer car α est un tenseur d'ordre d. Une manière d'alléger cette projection est de ne plus chercher la projection exacte mais une approximation de celle-ci dans un format hiérarchique de Tucker.

IV.2.4 Adaptation au format hiérarchique de Tucker

Avec $\mathcal{M} = \mathcal{C}_1(\mathcal{V})$, on va conserver ici la construction progressive des espaces réduits $(\mathcal{U}_r^\mu)_{r\in\mathbb{N}^*,\mu\in D}$ tels que

$$\mathcal{U}_r^\mu = \operatorname{span}\{v_i^\mu;\ 1\leq i\leq r\}\,, \quad \forall r\in\mathbb{N}^*,\ \forall \mu\in D \tag{IV.36}$$

et $\mathcal{U}_r = \bigotimes_\mu \mathcal{U}_r^\mu$. Au lieu de projeter « brutalement » la solution sur $\mathcal{U}_r$ ($\mathcal{M}_r = \mathcal{U}_r$), on va chercher une approximation u_r dans $\mathcal{H}_r^T(\mathcal{U}_r)$. Sachant que cet ensemble est fermé, le problème de minimisation $u_r \in \Pi_{\mathcal{H}_r^T(\mathcal{U}_r)}(u) = \arg\min_{v\in\mathcal{H}_r^T(\mathcal{U}_r)} \|u-v\|_{\mathcal{V}}$ possède au moins une solution. Avec $(w_i^\mu)_{1\leq i\leq r_\mu}$ une base de $\mathcal{U}_r^\mu$, on utilise cette base pour le paramétrage des éléments de $u_r \in \mathcal{H}_r^T(\mathcal{U}_r)$ associés à la feuille $t = \{\mu\} \in L(T)$. Ainsi tout tenseur de $\mathcal{H}_r^T(\mathcal{U}_r)$ est uniquement défini par un paramétrage des tenseurs de transferts $\left(\alpha^t\right)_{t\in I(T)}$. Pour ainsi dire, on va utiliser $G_{\mathcal{H}_r^T(\mathcal{U}_r)} : \times_{t\in I(T)} \mathcal{G}^t_{\mathcal{H}_r^T(\mathcal{U}_r)} \to \mathcal{U}_r$ tel que $g_r^t = \alpha^t,\ t\in I(T)$, et

$$G_{\mathcal{H}_r^T(\mathcal{U}_r)}\left(\left(\alpha^t\right)_{t\in I(T)}\right) = F_{\mathcal{H}_r^T(\mathcal{V})}\left(\left(\alpha^t\right)_{t\in I(T)}, \left(w_i^t\right)_{1\leq i\leq r_t,\ t=\{\mu\}\in L(T)}\right) \tag{IV.37}$$

où $F_{\mathcal{H}_r^T(\mathcal{V})}$, définie en section II.3.2.4, est le paramétrage de $\mathcal{H}_r^T(\mathcal{V})$. u_r est finalement trouvé par la minimisation alternée de la fonctionnelle

$$\left(\beta^t\right)_{t\in I(T)} \to \left\| u - G_{\mathcal{H}_r^T(\mathcal{U}_r)}\left(\left(\beta^t\right)_{t\in I(T)}\right)\right\|_{\mathcal{V}}. \tag{IV.38}$$

Remarque IV.2.8. *Si $\dim(\mathcal{V}^\lambda) = n_\lambda$ est suffisamment petit, on peut rajouter la feuille $t = \{\lambda\}$ dans le paramétrage $G_{\mathcal{M}_r}$. On ne travaille alors plus dans $\mathcal{M}_r = \mathcal{H}_r^T(\bigotimes_\mu \mathcal{U}_r^\mu)$ mais $\mathcal{M}_r = \mathcal{H}_r^T(\mathcal{V}^\lambda \otimes$*

$\left(\bigotimes_{\mu\neq\lambda}\mathcal{U}_r^\mu\right)$).

IV.3 Illustrations

IV.3.1 Liste des algorithmes

Pour les constructions gloutonnes avec mises à jour, on va s'intéresser uniquement au cas où $\mathcal{M} = \mathcal{C}_1(\mathcal{V})$. Ceci permet de diminuer les coûts de calculs pour la résolution des problèmes physiques, où la dimension spatiale a souvent un grand nombre de degrés de liberté.

On introduit la liste des algorithmes étudiés ici. On note :

- $\mathfrak{C}_1$: la construction gloutonne pure,
- $\mathfrak{C}_2 : \mathcal{M}_r = \text{span}\{v_i;\ 1 \leq i \leq r\}$, la construction gloutonne avec projection sur $\mathcal{M}_r$,
- $\mathfrak{C}_3 : \mathcal{M}_r = \mathcal{C}_r(\mathcal{V})$, la construction gloutonne avec mise à jour dans $\mathcal{C}_r(\mathcal{V})$, une seule itération de minimisations alternées sur $\mathcal{M}_r$ ($k_{max} = 1$ dans l'algorithme 9),
- $\mathfrak{C}_4$: $\mathcal{M}_r = \mathcal{C}_r(\mathcal{V})$, la construction gloutonne avec mise à jour dans $\mathcal{C}_r(\mathcal{V})$, 10 itérations de minimisations alternées ($k_{max} = 10$ dans l'algorithme 9),
- $\mathfrak{C}_5$: l'algorithme de minimisations alternées dans $\mathcal{C}_r(\mathcal{V})$ avec initialisation aléatoire avec l'algorithme présenté en section II.3.2.2,
- $\mathfrak{T}$: $\mathcal{M}_r = \mathcal{U}_r$, l'algorithme de projection sur l'espace produit tensoriel $\mathcal{U}_r = \bigotimes_\mu \mathcal{U}_r^\mu$ avec $\mathcal{U}_r^\mu = \text{span}\{v_i^\mu;\ 1 \leq i \leq r\}$ de la section IV.2.3,
- $\mathfrak{H}_1 : \mathcal{M}_r = \mathcal{H}_r^T(\mathcal{U}_r)$: la construction gloutonne des espaces associés aux feuilles avec la mise à jour des tenseurs de transferts (algorithme présenté en section IV.2.4, avec $k_{max} = 3$).
- $\mathfrak{H}_2$: un algorithme des minimisations alternées dans $\mathcal{H}_r^T(\mathcal{V})$.

IV.3.2 Préconditionnement des méthodes d'approximation directes

Ici on va étudier l'influence du préconditionnement sur les méthodes d'approximations directes. Pour cela, on va introduire un exemple simple en 2D artificiellement mal conditionné. On peut ainsi comparer les approximations sous format tenseurs et une référence obtenue avec une méthode éléments finis.

Le problème considéré est une équation de la chaleur sur un domaine carré $\Omega = \omega \times \omega$ avec $\omega = [0, 1]$ tel que

$$\begin{aligned} -\nabla \cdot (\kappa \nabla u) &= 1 \quad \text{sur} \quad \Omega, \\ u &= 0 \quad \text{sur} \quad \partial\Omega. \end{aligned} \tag{IV.39}$$

La conductivité κ est alors choisie telle que

$$\begin{aligned} \kappa(x_1, x_2) &= 1 \quad 0 \leq x_1 \leq 0.5,\ 0 \leq x_2 \leq 1, \\ \kappa(x_1, x_2) &= 10^4 \quad 0.5 \leq x_1 \leq 1,\ 0 \leq x_2 \leq 1. \end{aligned} \tag{IV.40}$$

On discrétise ω avec 100 éléments linéaires. Ainsi l'opérateur issu de la méthode des éléments finis du problème complet a un conditionnement de $1.52\ 10^7$.

La solution est approximée en utilisant une séparation des variables d'espaces. Ainsi on cherche

une approximation de la forme

$$u_r = \sum_{i=1}^{r} v_i^1 \otimes v_i^2. \tag{IV.41}$$

La norme utilisée ici pour le solveur est celle de la minimisation de résidu préconditionné, et donc $\|\cdot\|_{\mathcal{V}} = \|\cdot\|_{A^*P^*PA}$. Pour des raisons d'uniformité, on considérera une norme $\|\cdot\|_{A^*A}$ lors de l'application de la méthode sans préconditionnement.

IV.3.2.1 Algorithme $\mathfrak{C}_1$

On considère ici le solveur purement glouton avec $\mathcal{M} = \mathcal{C}_1(\mathcal{V})$.

Dans la figure IV.1, on a tracé le résidu relatif en fonction du rang de l'approximation. On observe que la meilleure convergence du résidu est obtenu sans aucun préconditionneur. Ceci est en fait normal puisque le choix de la norme $\|\cdot\|_{A^*A}$ conduit à la minimisation du résidu. Ce choix est donc optimal pour l'indicateur d'erreur en résidu.

Si maintenant on regarde l'erreur relative par rapport à la solution éléments finis du problème complet en figure IV.2, on observe que l'utilisation de P_{10} donne une excellente approximation de la solution pour un rang très faible. En particulier pour un rang $r = 10$, on a une excellente erreur relative de $3.04\ 10^{-6}$ pour un très mauvais résidu relatif de 54.5.

Le résidu préconditionné relatif (avec P_{10}) est encore une fois un bon estimateur de l'erreur relative d'après la figure IV.3. On observe une stagnation une fois que l'indicateur atteint 10^{-8}. Ce phénomène provient du format canonique qui ne peut pas atteindre une précision inférieure à 10^{-8} (voir Beylkin et Mohlenkamp [11]). On arrive à avoir une erreur de référence inférieure à 10^{-8} car l'approximation u_r est reconstruite sur le même maillage que la celui utilisé pour calculer la solution sur le domaine Ω, c'est-à-dire u_r est considéré comme un élément de $\mathbb{R}^{99\times 99}$.

En revanche, la convergence des algorithmes n'est pas affectée par le préconditionnement, contrairement aux solveurs itératifs présentés dans le chapitre III.3.

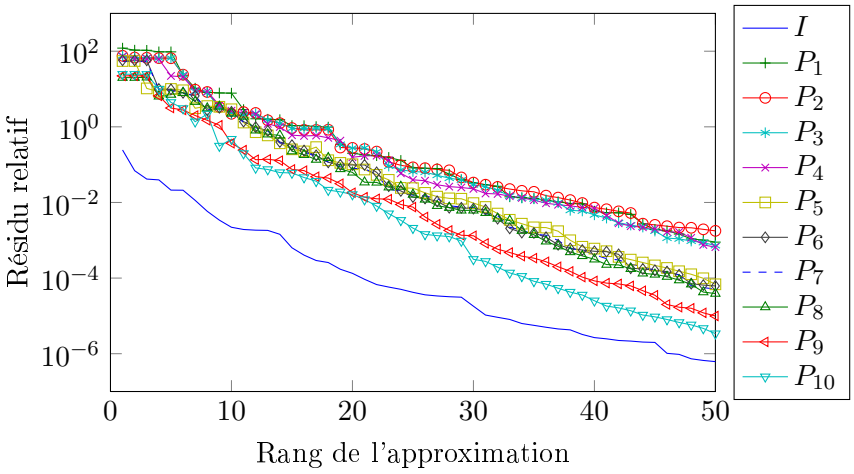


FIGURE IV.1 – Résidu relatif en fonction du rang pour un opérateur mal conditionné et différents préconditionneurs.

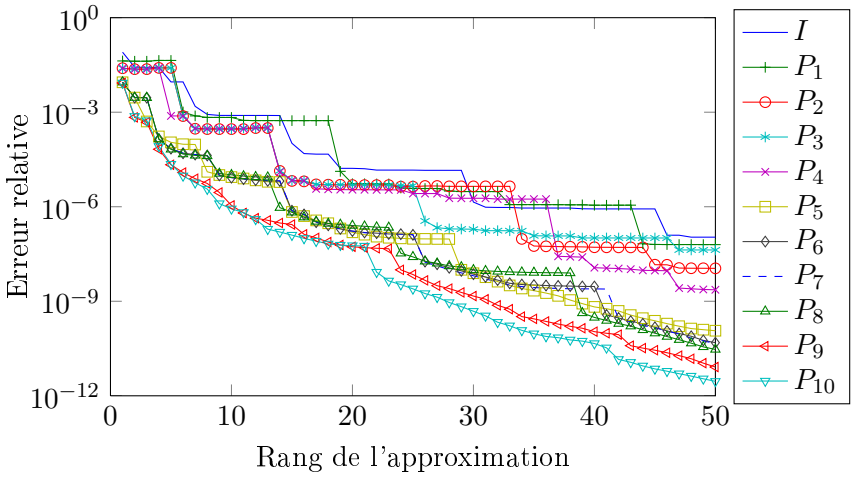


FIGURE IV.2 – Erreur relative en fonction du rang pour un opérateur mal conditionné et différents préconditionneurs.

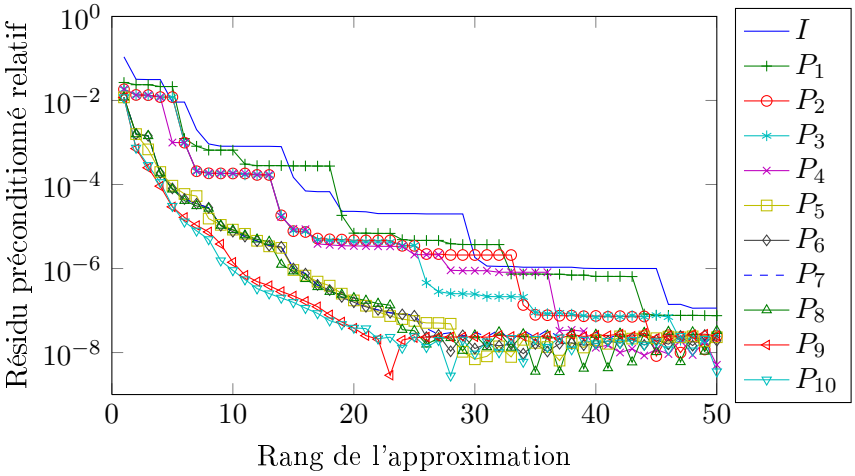


FIGURE IV.3 – Résidu préconditionné avec P_{10} relatif en fonction du rang pour un opérateur mal conditionné et différents préconditionneurs.

IV.3.2.2 Algorithme $\mathfrak{T}$

On peut faire ici les mêmes conclusions que précédemment, à savoir que le préconditionneur améliore l'erreur mais ne change pas la convergence des algorithmes, comme on peut le voir sur les figures IV.4, IV.5 et IV.6.

La stagnation observée sur la courbe IV.4, pour un rang de l'approximation supérieur à 20 et la méthode sans préconditionneur, est due à un mauvais conditionnement de l'opérateur $\mathcal{P}^*_{\mathcal{U}_r} A^* A \mathcal{P}_{\mathcal{U}_r}$, où $\mathcal{P}_{\mathcal{U}_r}$ représente la projection sur l'espace réduit $\mathcal{U}_r$. Avec

$$u_r = \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} \alpha_{ij} w_i^1 \otimes w_j^2, \tag{IV.42}$$

et $(w_i^\mu)_{1 \leq i \leq r_\mu}$ une famille orthonormale, lors de la phase de projection, α est solution d'un système linéaire

$$M\alpha = S. \tag{IV.43}$$

On vérifie alors que pour un rang $(22, 22)$, on a $\text{cond}(M) = 5.89\ 10^{13}$. La stagnation provient ainsi du mauvais conditionnement de l'opérateur de départ A.

On voit en figure IV.3 un avantage à l'utilisation des tenseurs de Tucker. La valeur de 10^{-8} n'est plus une limite en précision du format et d'ailleurs on voit que le résidu relatif préconditionné peut atteindre la précision machine. Ceci vient de l'orthogonalisation des vecteurs des sous-espaces associés. En effet, si on a

$$u_r = \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} \alpha_{ij} w_i^1 \otimes w_j^2, \tag{IV.44}$$

avec $(w_i^\mu)_{1 \leq i \leq r_\mu}$ une famille orthonormale, alors on a aussi l'égalité

$$\|u_r\| = \|\alpha\|_{\text{Frobenius}}. \tag{IV.45}$$

Ce résultat s'étend aussi au format hiérarchique de Tucker. Si $u_r \in \mathcal{H}_r^T(\mathcal{V})$ tel que

$$u_r = \sum_{i=1}^{r_{D_1}} \sum_{j=1}^{r_{D_2}} \alpha_{ij}^D v_i^{D_1} \otimes v_i^{D_2}, \quad D = D_1 \cup D_2,\ D_1 \cap D_2 = \emptyset \tag{IV.46}$$

où $\left(v_i^{D_\mu}\right)_{1 \leq i \leq r_{D_\mu}}$ est une famille orthonormale, alors on a

$$\|u_r\| = \left\|\alpha^D\right\|_{\text{Frobenius}}. \tag{IV.47}$$

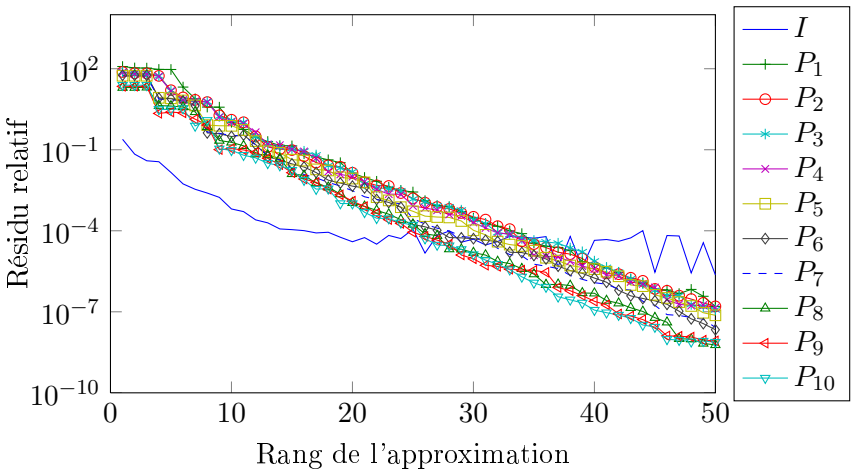


FIGURE IV.4 – Résidu relatif en fonction du rang pour un opérateur mal conditionné et différents préconditionneurs.

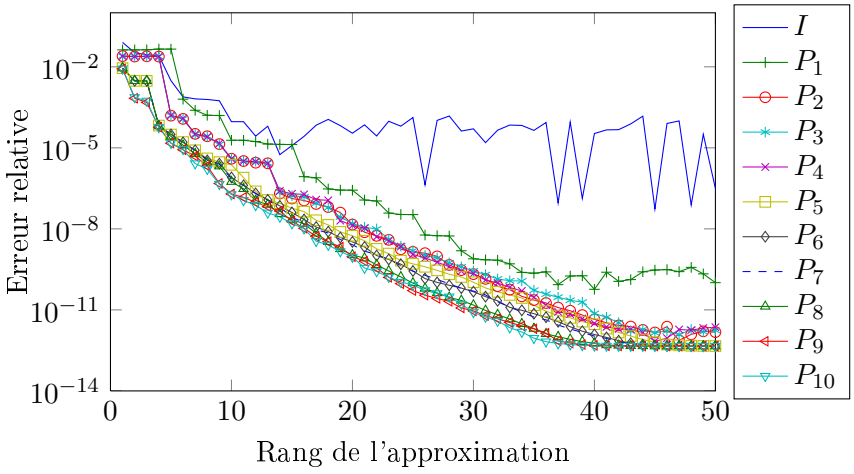


FIGURE IV.5 – Erreur relative en fonction du rang pour un opérateur mal conditionné et différents préconditionneurs.

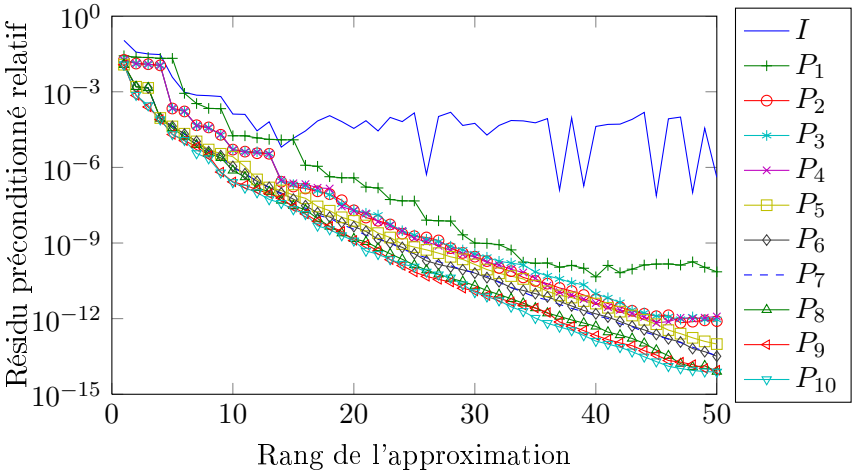


FIGURE IV.6 – Résidu préconditionné relatif par P_{10} en fonction du rang pour un opérateur mal conditionné et différents préconditionneurs.

IV.3.3 Problème de Poisson

On traite ici le problème de la section III.5.2.

IV.3.3.1 Tenseurs canoniques, $\mathcal{M}_r \subset \mathcal{C}_r(\mathcal{V})$

On suppose ici que $d = 8$. On étudie les algorithmes $(\mathfrak{C}_i)_{1\leq i\leq 5}$. Les résidus relatifs en fonction du rang de l'approximation pour les algorithmes $(\mathfrak{C}_i)_{1\leq i\leq 5}$ sont illustrés en figure IV.7.

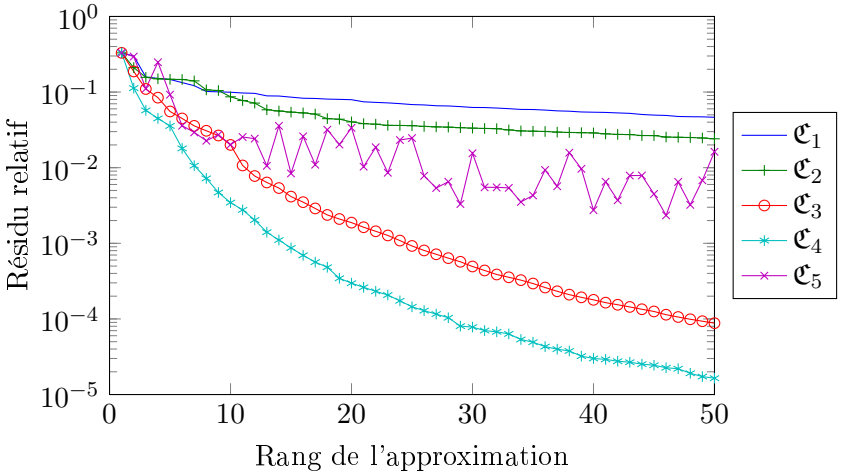


FIGURE IV.7 – Résidu relatif en fonction du rang de l'approximation pour les algorithmes $(\mathfrak{C}_i)_{1\leq i\leq 5}$ pour la résolution du problème de Poisson.

On remarque que l'on peut catégoriser les algorithmes en 3 classes :

- $\mathfrak{C}_1$ et $\mathfrak{C}_2$,
- $\mathfrak{C}_3$ et $\mathfrak{C}_4$,
- $\mathfrak{C}_5$.

Les algorithmes $\mathfrak{C}_1$ et $\mathfrak{C}_2$ ont un coût faible par itérations pour construire une approximation de rang faible de la solution. Elles possèdent un taux de convergence similaire ce qui donne un avantage à la construction gloutonne pure $\mathfrak{C}_1$. La contrepartie du faible coût par itération est le très faible taux de convergence des algorithmes. En effet on remarque qu'un tenseur de rang de 5 construit avec $\mathfrak{C}_4$ est meilleur qu'un tenseur de rang 50 calculé avec l'algorithme $\mathfrak{C}_2$.

$\mathfrak{C}_3$ et $\mathfrak{C}_4$ possèdent aussi un taux de convergence similaire. Il est bien meilleur que celui de $\mathfrak{C}_1$ et $\mathfrak{C}_2$ pour un coût bien plus élevé. On observe qu'une seule itération de minimisations alternées sur les dimensions améliore fortement la construction gloutonne pure $\mathfrak{C}_1$. Dans notre implémentation, la construction de u_{50} avec $\mathfrak{C}_3$ est réalisée environ 4 fois plus rapidement qu'avec $\mathfrak{C}_4$ (et 17 fois plus lentement qu'avec $\mathfrak{C}_1$), pour un gain d'un ordre de grandeur sur le résidu relatif.

L'initialisation aléatoire de $\mathfrak{C}_5$ le rend très instable. En revanche, l'initialisation gloutonne semble « stabiliser » la convergence de l'algorithme de minimisations alternées dans $\mathcal{C}_r(\mathcal{V})$ (algorithme $\mathfrak{C}_4$).

De tous ces résultats, il semble que les algorithmes $\mathfrak{C}_3$ et $\mathfrak{C}_4$ sont à favoriser pour avoir une solution précise du système linéaire pour un rang faible et un temps de calcul raisonnable.

IV.3.3.2 Tenseurs de Tucker, $\mathcal{M}_r \subset \mathcal{T}_r(\mathcal{V})$

On étudie ici le cas de l'algorithme $\mathfrak{T}$. Comme cet algorithme souffre de la malédiction de la dimensionnalité liée au format de Tucker, on se limite au cas $d = 3$.

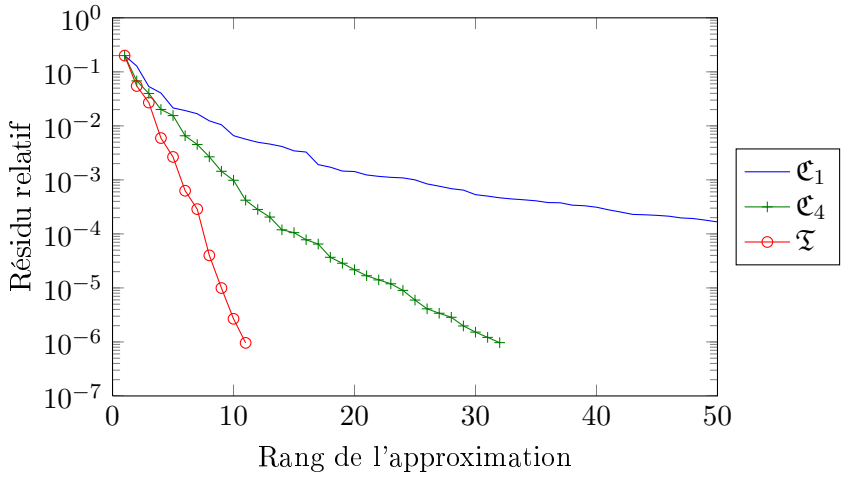


FIGURE IV.8 – Résidu relatif en fonction du rang de l'approximation pour les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$ et $\mathfrak{T}$ pour la résolution du problème de Poisson.

Dans la figure IV.8, on voit l'apport de la méthode de projection et du format de Tucker par rapport au tenseur canonique. L'algorithme $\mathfrak{T}$ surpasse l'algorithme $\mathfrak{C}_4$ en précision du résidu, ainsi qu'en temps de calcul.

IV.3.3.3 Tenseurs hiérarchiques de Tucker, $\mathcal{M}_r \subset \mathcal{H}_r^T(\mathcal{V})$

Les algorithmes testés ici sont $\mathfrak{H}_1$ et $\mathfrak{H}_2$. Le résidu relatif en fonction du rang de l'approximation est tracé en figure IV.9. Tout d'abord, les algorithmes $\mathfrak{H}_1$ et $\mathfrak{H}_2$ ont des convergences très similaires.

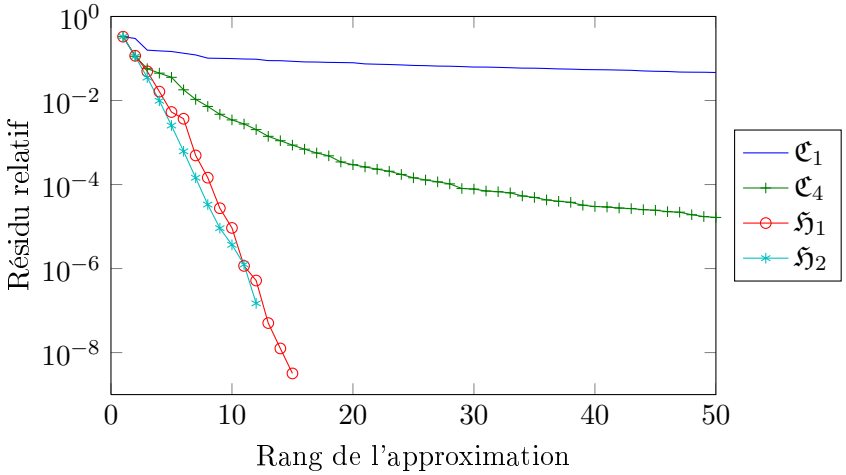


FIGURE IV.9 – Résidu relatif en fonction du rang de l'approximation pour les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$ et $(\mathfrak{H}_i)_{1\leq i\leq 2}$ pour la résolution du problème de Poisson.

En ce sens, $\mathfrak{H}_1$ fournit une décomposition quasi-optimale de la solution, pour une complexité très

différente. En effet pour $\mathfrak{H}_1$ il n'y a pas d'optimisations sur les éléments associés aux feuilles de l'arbre T, mais une construction gloutonne des sous-espaces associés aux feuilles.

De plus on retrouve le même type de convergence que pour l'algorithme $\mathfrak{T}$ en dimension $d = 3$, ce qui montre que $\mathfrak{H}_1$ est une généralisation convaincante de $\mathfrak{T}$ aux grandes dimensions.

IV.3.3.4 Temps de calcul

Problème de départ. Les résidus relatifs en fonction du temps de calcul sont donnés en figure IV.10. On est allé jusqu'à un rang maximal de 150 pour $\mathfrak{C}_1$, 50 pour $\mathfrak{C}_4$ et 15 pour $\mathfrak{H}_1$.

L'implémentation a été entièrement réalisée avec MATLAB®, à partir de la la bibliothèque Hierarchical Tucker Toolbox [48]. Cette implémentation souffre de plusieurs problèmes, dont le principal est la mauvaise gestion de la contraction de tenseurs. En particulier, pour deux tenseurs $x \in \mathbb{R}^{p\times q\times r}$ et $y \in \mathbb{R}^{q\times n}$, la contraction du deuxième indice de x avec le premier indice de y passe par une matricisation de x et une multiplication matricielle. Ceci a pour but d'utiliser de manière efficace la multiplication native de MATLAB®. Ainsi on considère $x^{\{2\}} \in \mathbb{R}^{pr\times q}$, pour avoir $z^{\{2\}} = x^{\{2\}}y \in \mathbb{R}^{pr\times n}$ et revenir à $z \in \mathbb{R}^{p\times n\times r}$. Un profilage de code nous indique que ces opérations de matricisation/dématricisation représente 30% du temps de calcul total pour la construction du tenseur hiérarchique de Tucker de rang 15 à cause de multiples copies inutiles. Comme ces opérations ne sont pas nécessaires dans une implémentation en C ou en FORTRAN, on a aussi tracé le résidu relatif en fonction du temps sans ces copies, sous le sigle $\widetilde{\mathfrak{H}}_1$. On note que ceci est essentiellement dû à la mise à jour des tenseurs de transferts.

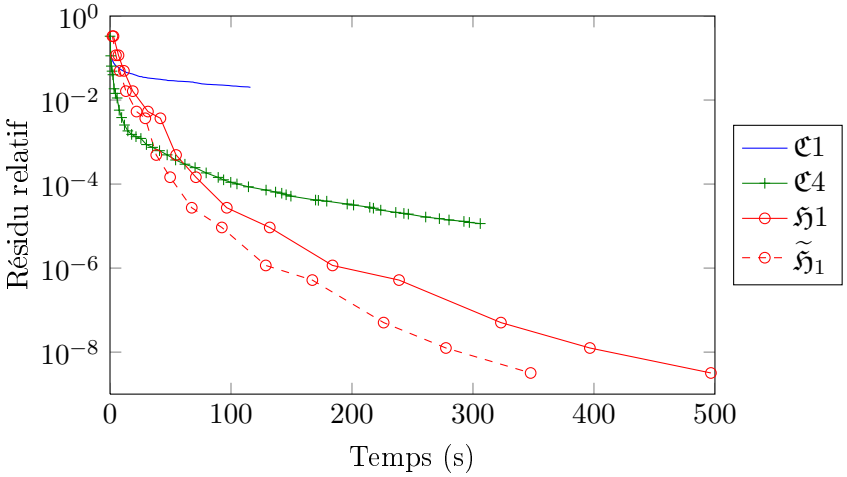


FIGURE IV.10 – Résidu relatif en fonction du temps de calcul pour le problème de Poisson et les différents algorithmes.

D'après la figure IV.10, $\mathfrak{C}_1$ fournit une approximation grossière de la solution en 100s. Pour le même temps de calcul, on gagne 2 ordres de grandeur sur l'erreur en résidu en utilisant $\mathfrak{C}_4$ ou $\mathfrak{H}_1$. En 300s, $\mathfrak{H}_1$ fournit une meilleure approximation en résidu que $\mathfrak{C}_4$ de 2 ordres de grandeur, alors qu'on peut en espérer 4 en regardant $\widetilde{\mathfrak{H}}_1$.

Maillage raffiné. Cette fois on considère un maillage de 120 éléments de ω. Le résidu relatif en fonction du temps est disponible en figure IV.11.

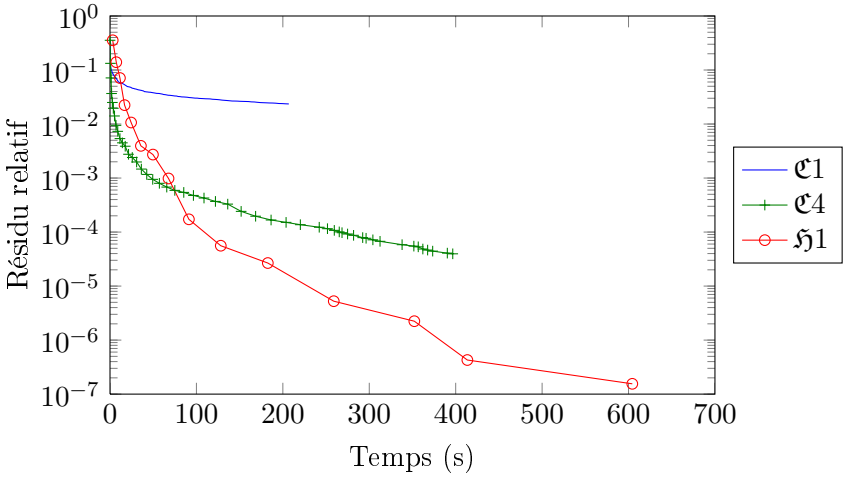


FIGURE IV.11 – Résidu relatif en fonction du temps pour un problème de Poisson avec un maillage raffiné.

La première remarque concerne le résidu. La diminution de la taille du maillage augmente le résidu relatif.

On remarque que cette fois, le temps est augmenté de 78% pour calculer un tenseur de rang 150 avec $\mathfrak{C}_1$. Le temps pour calculer un tenseur de rang 50 avec $\mathfrak{C}_4$ est augmenté de 29%, alors que le calcul d'un tenseur de rang 15 avec $\mathfrak{H}_1$ est quant à lui augmenté de 21%. Comme on pouvait s'y attendre $\mathfrak{H}_1$ est l'algorithme le moins sensible à la taille du maillage. Il devrait être encore plus efficace pour les maillages les plus fins utilisés dans des problèmes non académiques.

Si maintenant on compare avec le PCG de la section III.5.3, le temps de calcul de ce dernier était quasiment triplé pour une augmentation de la taille du maillage. Ceci nous porte à croire que l'algorithme $\mathfrak{H}_1$ proposé dans cette thèse sera d'autant plus efficace par rapport aux autres algorithmes que le maillage sera fin.

Dimension $d = 4$. Enfin, pour comparer avec le PCG utilisé dans la section III.5.3, on s'est placé en dimension 4. Le résidu relatif en fonction du temps est tracé en figure IV.12.

Cette fois 150 itérations de $\mathfrak{C}_1$ offrent une convergence bien plus rapide en temps que les autres algorithmes, le faible nombre de dimensions étant favorable aux tenseurs canoniques. En revanche on arrive toujours à une meilleure précision avec l'utilisation de $\mathfrak{C}_4$ et $\mathfrak{H}_1$ pour des rangs respectifs de 50 et de 15. Par contre, si l'on compare maintenant avec la section III.5.3, l'utilisation des algorithmes directs est beaucoup plus lente. On préférera alors le PCG. Cependant, les méthodes d'approximations directes utilisées ici ne nécessitent aucun préconditionneur, ni connaissance a priori du rang de la solution.

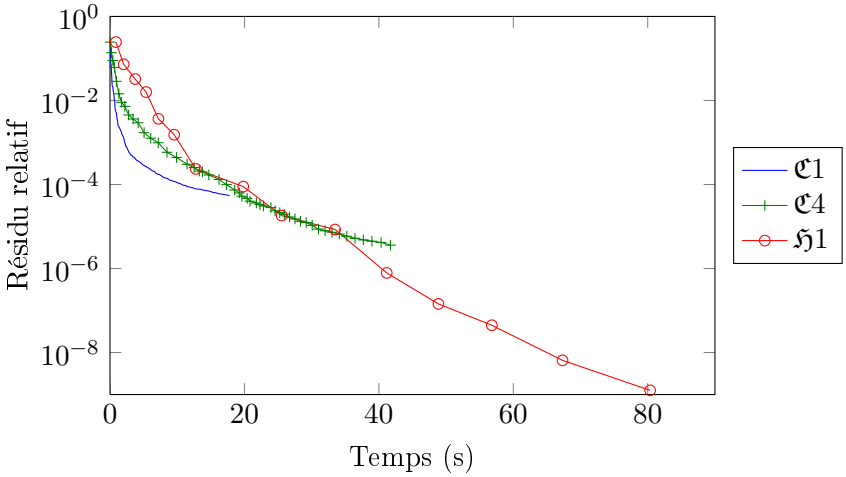


FIGURE IV.12 – Résidu relatif en fonction du temps pour un problème de Poisson en dimension 4.

IV.3.4 Problème symétrique

On considère le problème d'équation de la chaleur présenté en section III.5.3. L'opérateur A associé étant symétrique, on considère ici la norme $\|\cdot\|_{\mathcal{V}} = \|\cdot\|_A$.

IV.3.4.1 Convergence des algorithmes

Le problème étant de dimension $d = 6$, on se limite aux algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$, $\mathfrak{H}_1$ et $\mathfrak{H}_2$. Le résidu relatif en fonction du rang de l'approximation est tracé en figure IV.13 pour les différents algorithmes.

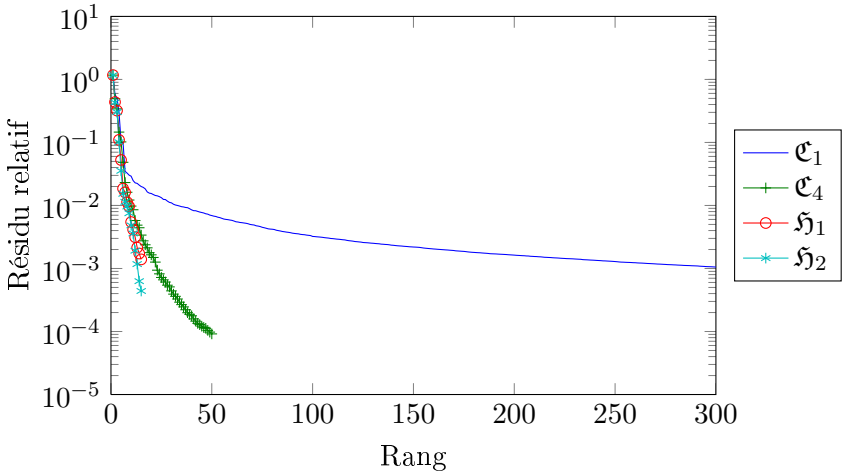


FIGURE IV.13 – Résidu relatif en fonction du rang de l'approximation pour le problème d'équation de la chaleur et les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$, $\mathfrak{H}_1$ et $\mathfrak{H}_2$.

On observe cette fois encore que l'algorithme $\mathfrak{C}_1$ a un taux de convergence très faible avec le rang, ce qui n'est pas le cas pour les autres algorithmes.

Les algorithmes $\mathfrak{C}_4$, $\mathfrak{H}_1$ et $\mathfrak{H}_2$ convergent très rapidement pour les faibles rangs. $\mathfrak{H}_1$ fournit encore une décomposition quasi-optimale de la solution. On atteint néanmoins la meilleure minimisation du résidu avec l'algorithme $\mathfrak{C}_4$, ainsi que $\mathfrak{H}_2$,

Si maintenant on compare à l'algorithme PCG et aux résultats de la section III.5.3, l'algorithme $\mathfrak{H}_1$, pour un rang $r_t = 10$ fournit un résidu relatif de 5.5 10^{-3}, $\mathfrak{H}_2$ de 4.8 10^{-3} et quant au PCG préconditionné avec P_E, il donne un résidu relatif de 7.5 10^{-3}. Les 3 algorithmes donnent donc un résidu relatif très proche, néanmoins $\mathfrak{H}_1$ et $\mathfrak{H}_2$ ne nécessitent pas de préconditionneurs.

IV.3.4.2 Influence de la variabilité

On a appliqué les mêmes algorithmes au problème de grande variabilité présenté en section III.5.3.5. Le résidu relatif en fonction du rang est affiché en figure IV.14.

Les conclusions sont identiques à celles de la section IV.3.4.1. Cependant on remarque que le résidu relatif est ici supérieur d'au moins un ordre de grandeur, tout comme c'était le cas pour le PCG en section III.5.3.5.

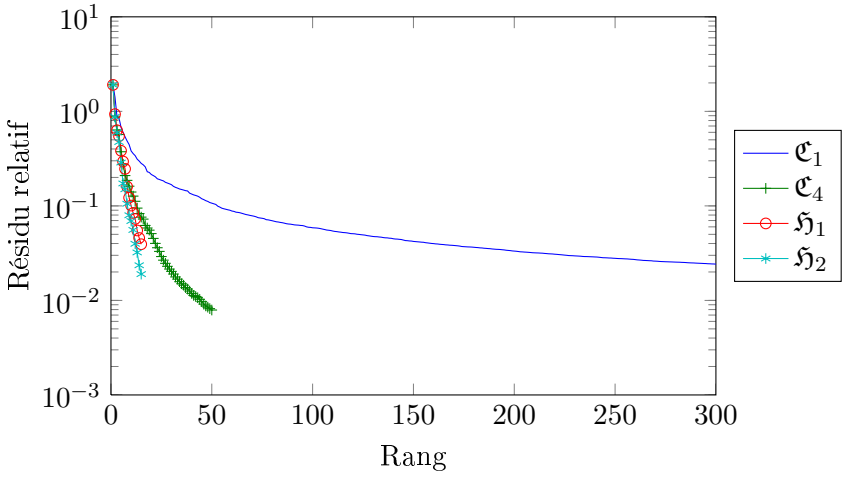


FIGURE IV.14 – Résidu relatif en fonction du rang de l'approximation pour le problème d'équation de la chaleur avec une grande variabilité et les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$, $\mathfrak{H}_1$ et $\mathfrak{H}_2$.

IV.3.4.3 Influence du maillage

Cette fois le maillage a de la première dimension a été raffiné. Elle passe de 386 degrés de liberté à 1641. Le résidu relatif en fonction du rang est donné en figure IV.15.

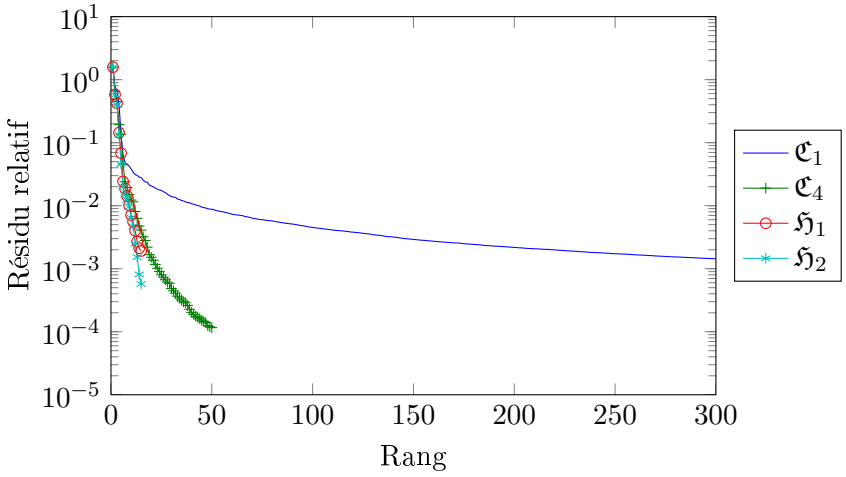


FIGURE IV.15 – Résidu relatif en fonction du rang de l'approximation pour le problème d'équation de la chaleur avec un maillage raffiné et les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$, $\mathfrak{H}_1$ et $\mathfrak{H}_2$.

La figure est quasiment identique à celle de la figure IV.13 pour le maillage initial, les conclusions sont donc aussi les mêmes : (1) $\mathfrak{C}_1$ converge très lentement au contraire de $\mathfrak{C}_4$, $\mathfrak{H}_1$ et $\mathfrak{H}_2$, (2) $\mathfrak{H}_1$ fournit une décomposition quasi-optimale si on compare à $\mathfrak{H}_2$ et (3) $\mathfrak{C}_4$ donne le résidu relatif le plus petit de tous.

IV.3.4.4 Temps de calcul

Problème initial. En figure IV.16, on a tracé le résidu relatif en fonction du temps pour le problème d'équation de la chaleur.

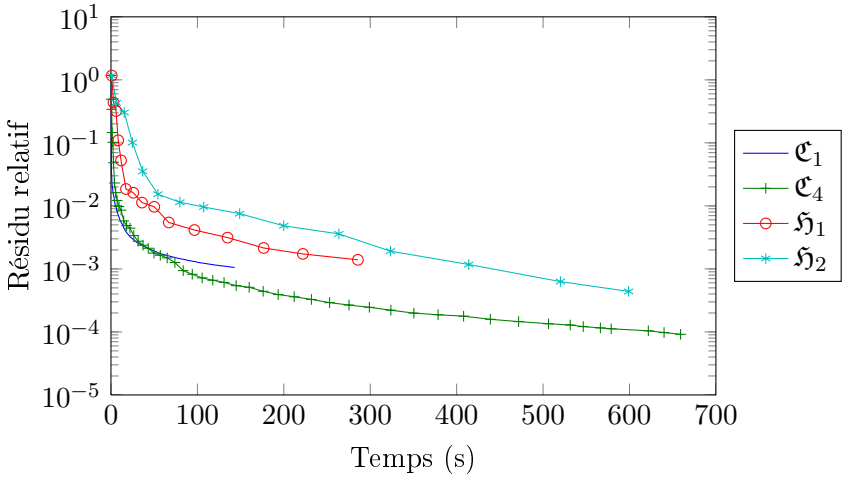


FIGURE IV.16 – Résidu relatif en fonction du temps pour le problème de l'équation de la chaleur et les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$, $\mathfrak{H}_1$ et $\mathfrak{H}_2$.

Cette fois les algorithmes $\mathfrak{C}_1$ et $\mathfrak{C}_4$ sont les plus performants. Cela provient des dimensions des espaces d'approximation éléments finis. On a vu dans la section IV.3.3 que les algorithmes $\mathfrak{C}_1$ et $\mathfrak{C}_4$ sont assez sensibles à la taille des maillages. Ces maillages possédant peu de noeuds ici, les algorithmes $\mathfrak{C}_1$ et $\mathfrak{C}_4$ sont les plus performants. Si on compare maintenant avec l'utilisation du PCG, les temps de résolution ici sont équivalents au seul temps de construction du préconditionneur.

Problème avec une grande variabilité. Le résidu relatif en fonction du temps a été tracé cette fois pour le problème avec une plus grande variabilité en figure IV.17.

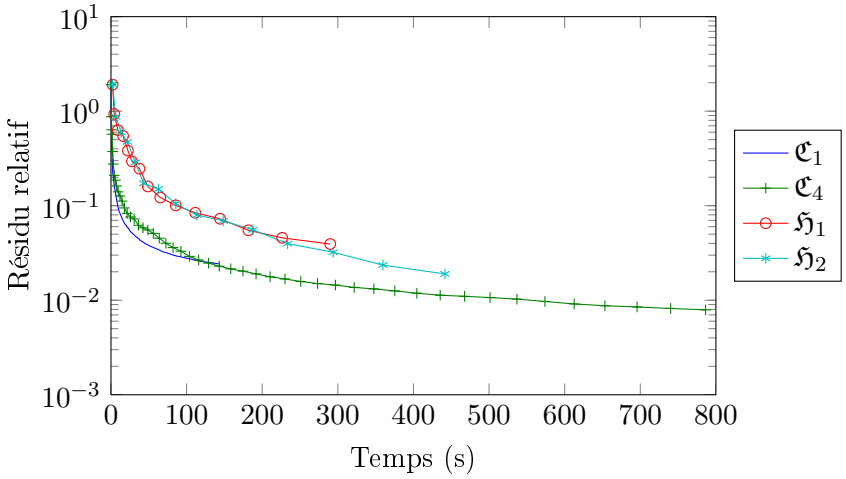


FIGURE IV.17 – Résidu relatif en fonction du temps pour le problème de l'équation de la chaleur avec une grande variabilité et les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$, $\mathfrak{H}_1$ et $\mathfrak{H}_2$.

Les conclusions sont identiques à celles du paragraphe précédent, à savoir que $\mathfrak{C}_1$ et $\mathfrak{C}_4$ sont plus efficaces que $\mathfrak{H}_1$ et $\mathfrak{H}_2$ à cause des maillages avec peu de degrés de liberté utilisés ici. Cependant si on compare les temps de calculs, les temps de résolution sont un peu plus long ici que dans le cas où

la variabilité est plus faible. Les algorithmes de minimisations alternées nécessitent plus d'itérations pour converger.

Problème avec un maillage raffiné. Le résidu relatif en fonction du temps pour le maillage raffiné est indiqué en figure IV.18.

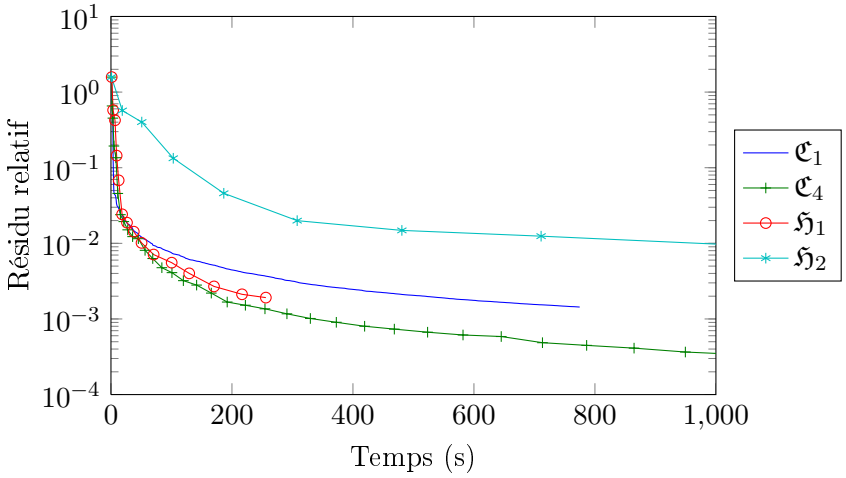


FIGURE IV.18 – Résidu relatif en fonction du temps pour le problème de l'équation de la chaleur avec un maillage raffiné et les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$, $\mathfrak{H}_1$ et $\mathfrak{H}_2$.

Comme on pouvait s'y attendre, $\mathfrak{H}_1$ devient cette fois plus rapide que $\mathfrak{C}_1$, en étant équivalent à $\mathfrak{C}_4$. Ceci est dû au faible rang dans la représentation de $\mathfrak{H}_1$. Si on compare avec la figure IV.16, on observe pour les mêmes raisons que $\mathfrak{H}_1$ est peu sensible au raffinement du maillage, au contraire de $\mathfrak{C}_1$.

IV.3.5 Problème non symétrique

Le problème traité est celui de l'équation de diffusion-convection-réaction en régime transitoire présenté en section III.5.4. L'opérateur n'étant pas symétrique, on considère la norme $\|\cdot\|_{\mathcal{V}} = \|\cdot\|_{A^*A}$.

IV.3.5.1 Convergence des algorithmes

Problème initial. Le résidu relatif en fonction du rang de l'approximation est tracé en figure IV.19 pour les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$ et $\mathfrak{H}_1$.

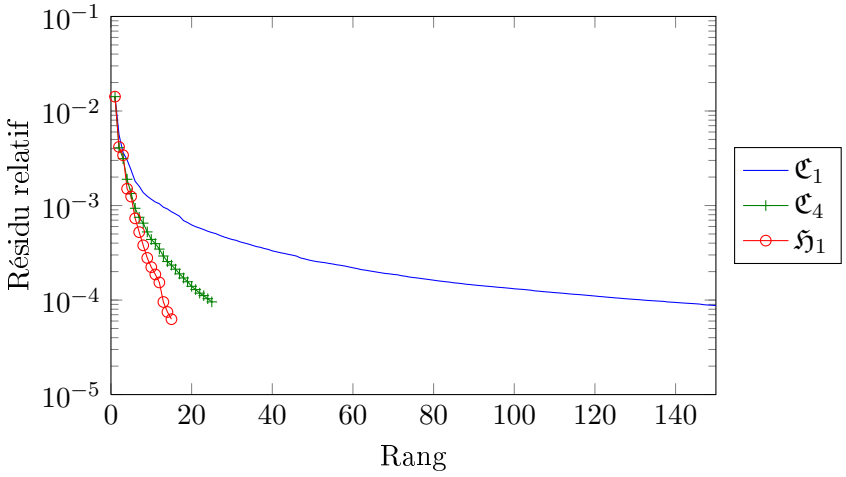


FIGURE IV.19 – Résidu relatif en fonction du rang pour le problème non symétrique et les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$, et $\mathfrak{H}_1$.

Les méthodes d'approximations directs dans les sous-ensembles de tenseur sont encore bien plus précises que les solveurs itératifs de la section III. En effet, pour un même rang, le résidu relatif est plus petit d'un ordre de grandeur par rapport à celui de GMRES (voir section III.5.4), encore une fois sans utiliser de préconditionneur.

Influence du maillage. On considère ici le même problème avec un maillage de la dimension associée à la variable spatiale passant à 4618 degrés de liberté. On étudie les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$ et $\mathfrak{H}_1$. Il est nécessaire de distinguer ici deux versions de l'algorithme $\mathfrak{H}_1$. La première est l'algorithme classique qui garde la notation $\mathfrak{H}_1$, la seconde est celle où l'opérateur A^*A est orthogonalisé avant l'application de l'algorithme $\mathfrak{H}_1$, que l'on note $\mathfrak{H}_1^{\perp}$. Le résidu relatif en fonction du rang de l'approximation est disponible en figure IV.20.

Encore une fois la convergence de $\mathfrak{H}_1$ est meilleure que celle des autres algorithmes. Par ailleurs le comportement de $\mathfrak{H}_1^{\perp}$ est intéressant. Il indique que l'opérateur A^*A est très mal conditionné, puisque l'erreur numérique induite par l'orthogonalisation empêche l'algorithme de converger. La conséquence est que l'on est obligé de manipuler un opérateur avec un rang élevé pour résoudre le problème. Il est en fait nécessaire ici de bien préconditionner l'opérateur pour que l'orthogonalisation n'entraîne plus cette divergence. On note que pour le problème avec un maillage grossier de la section IV.3.5.1, l'opérateur A^*A était orthogonalisé. On utilisait l'algorithme $\mathfrak{H}_1^{\perp}$ qui était alors noté $\mathfrak{H}_1$.

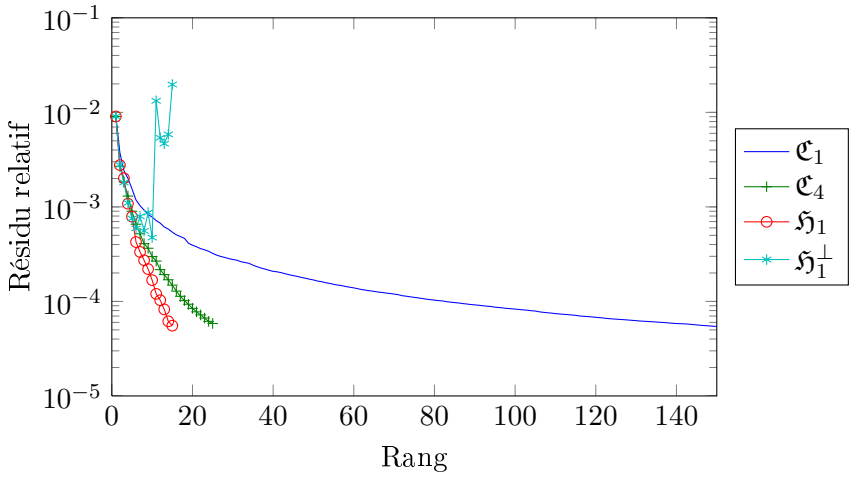


FIGURE IV.20 – Résidu relatif en fonction du rang pour le problème non symétrique, un maillage plus fin et les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$, $\mathfrak{H}_1$ et $\mathfrak{H}_1^\perp$.

IV.3.5.2 Temps de calcul

Problème initial Le résidu relatif en fonction du temps est donné en figure IV.21 pour le problème initial et les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$ et $\mathfrak{H}_1$.

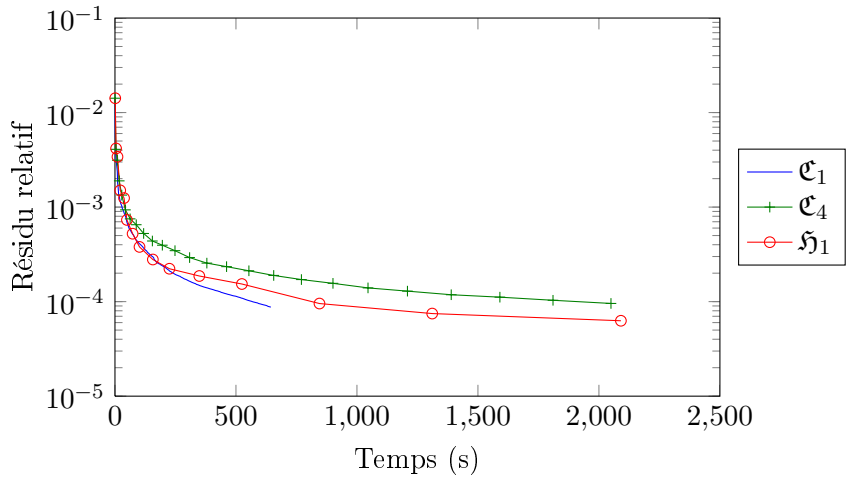


FIGURE IV.21 – Résidu relatif en fonction du temps pour le problème non symétrique et les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$ et $\mathfrak{H}_1$.

Dans notre implémentation $\mathfrak{H}_1$ est aussi efficace que $\mathfrak{C}_1$ dans un premier temps, puis $\mathfrak{H}_1$ devient bien plus lent. Ceci s'explique par l'augmentation du rang de la solution qui rend la manipulation des tenseurs de transferts plus coûteuses que la minimisation dans $\mathcal{C}_1(\mathcal{V})$. Dans ce cas-ci $\mathfrak{C}_4$ n'est pas compétitif avec les autres algorithmes. Si on compare maintenant avec GMRES en section III.5.4, le temps de résolution correspond au temps nécessaire pour construire le préconditionneur proposé. On a de plus un résidu relatif de deux ordres de grandeur plus petit ici que pour GMRES ($6.27\ 10^{-5}$ pour un rang de 15 pour $\mathfrak{H}_1$ contre $1.63\ 10^{-3}$ pour GMRES(10) et un rang de 15).

Influence du maillage Pour le cas où le maillage est raffiné, on a tracé le résidu relatif en fonction du temps pour les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$, $\mathfrak{H}_1$ et $\mathfrak{H}_1^\perp$ en figure IV.22.

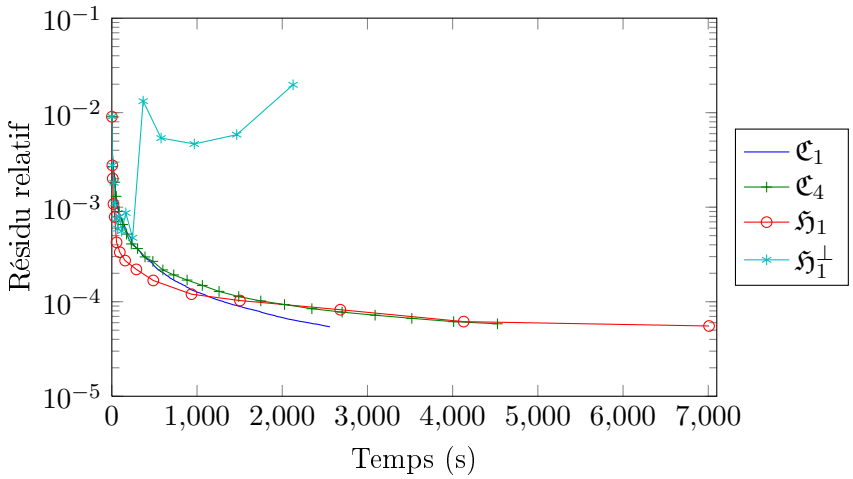


FIGURE IV.22 – Résidu relatif en fonction du temps pour le problème non symétrique et les algorithmes $\mathfrak{C}_1$, $\mathfrak{C}_4$, $\mathfrak{H}_1$ et $\mathfrak{H}_1^\perp$.

Au départ des algorithmes, $\mathfrak{H}_1$ est le plus rapide de tous. Au bout d'un certain temps (ici environ 1000s), $\mathfrak{C}_1$ devient le plus rapide des algorithmes. On peut raisonnablement espérer un croisement entre ces deux courbes plus tard avec une bonne implémentation des tenseurs hiérarchiques de Tucker. En effet, si l'opérateur A^*A n'est pas orthogonalisé, ses différents tenseurs de transfert sont super diagonaux, et on pourrait les considérer comme creux pour les manipuler plus efficacement. Une autre façon d'améliorer le temps de calcul est de préconditionner efficacement l'opérateur A^*A pour pouvoir employer l'algorithme $\mathfrak{H}_1^\perp$ sans perturber la convergence. Comme on l'a montré en section IV.3.2, le résidu serait moins bon mais l'erreur bien meilleure. Dans ce cas, le temps de construction d'un tenseur de rang 15 devrait s'approcher du temps de celui de $\mathfrak{H}_1^\perp$ ici.

IV.4 Résumé

On a appliqué ici les méthodes d'approximation de tenseurs du chapitre II pour la résolution de systèmes linéaires.

La première partie de ce chapitre a été dédiée au choix de la norme pour l'application des méthodes d'approximation de tenseurs du chapitre II. Une norme a été proposée pour préconditionner ces approches. Cette dernière revient à une formulation du problème comme une minimisation du résidu préconditionné. La principale conclusion concernant le préconditionnement est qu'il diminue l'erreur relative en norme canonique mais n'améliore pas la vitesse de convergence en fonction du rang de l'approximation.

Ensuite, différentes constructions pour l'approximation de la solution ont été proposées. Des méthodes ont été développées pour l'utilisation de formats de tenseurs non canoniques pour approximer une solution. On a tout d'abord défini un nouvel algorithme de projection, dans lequel on effectue une construction gloutonne de sous-espaces à partir de problèmes de meilleure approximation de rang 1. Cette méthode fournit de bons résultats pour des problèmes de faible dimension. Cependant l'utilisation de tenseurs de Tucker implique que cet algorithme souffre de la malédiction de la dimensionnalité. Elle a donc été étendue naturellement aux formats hiérarchiques de Tucker en remplaçant la phase de projection par un algorithme de minimisations alternées sur les tenseurs de transfert. Tous ces algorithmes ont été appliqués avec succès à des problèmes symétriques et non symétriques.

En ce qui concerne le coût de calcul, tous les temps indiqués sont à relativiser. Tout d'abord l'implémentation des tenseurs hiérarchiques de Tucker utilisée n'est pas du tout adaptée aux problèmes traités. Une fois qu'une implémentation efficace sera réalisée, il restera à analyser les critères d'arrêt des différents algorithmes. Ceux utilisés ici sont trop restrictifs ce qui augmente les différents temps de calculs. On espère diminuer ces temps en les assouplissant.

Chapitre V

Application à l'homogénéisation numérique

Sommaire

La première partie de ce chapitre concerne un article soumis à « Computer Methods Applied to Mechanics and Engineering ». Il s'agit d'appliquer les méthodes d'approximations de tenseurs à la résolution des problèmes d'homogénéisation thermique. L'article est introduit tel quel en section V.1 avec des notations adaptées au reste de cette thèse. Le lecteur retrouvera dans les sections 2 et 3 de l'article des sujets déjà traités aux chapitres II et IV. On proposera ensuite en section V.2 des extensions à l'article.

V.1 Tensors-based methods for numerical homogenization from high-resolution images

V.1.1 Résumé

On présente une stratégie numérique basée sur les méthodes d'approximation de tenseurs pour les résolutions des problèmes d'homogénéisation numérique, où les géométries sont issues d'images de haute résolution. On introduit tout d'abord des processus numériques spécifiques pour exprimer les problèmes d'homogénéisation basés sur les images dans un cadre de tenseurs. Ceci inclut l'approximation de tenseurs dans un format adéquat des champs de propriétés matériaux, ainsi que des indicatrice de phases reconstruites à partir d'images segmentées. On introduit ensuite des variantes de la PGD pour la construction d'une décomposition dans différents formats de tenseurs de la solution d'un problème aux limites. Une nouvelle définition de la PGD est présentée, celle-ci permet une construction progressive d'une décomposition de la solution sous un format de tenseur de Tucker. Ce format est adapté à cette application et améliore les propriétés de convergence des décompositions de tenseur. Finalement, on introduit une stratégie d'estimation de l'erreur basée sur un problème dual pour estimer les quantités d'intérêt, c'est-à-dire les propriétés homogénéisées du matériau. On présente aussi une stratégie adaptative complète basée sur l'estimation de l'erreur pour la construction progressive de décompositions de tenseur (des solutions des problèmes primaux et

duaux) fournissant la prédiction des quantités homogénéisées avec une précision donnée.

Tensor-based methods for numerical homogenization from high-resolution images[☆]

L. Giraldi, A. Nouy*, G. Legrain, P. Cartraud

LUNAM Université, GeM, UMR CNRS 6183, École Centrale de Nantes, Université de Nantes, France

Abstract

We present a complete numerical strategy based on tensor approximation techniques for the solution of numerical homogenization problems with geometrical data coming from high resolution images. We first introduce specific numerical treatments for the translation of image-based homogenization problems into a tensor framework. It includes the tensor approximations in suitable tensor formats of fields of material properties or indicator functions of multiple material phases recovered from segmented images. We then introduce some variants of Proper Generalized Decomposition (PGD) methods for the construction of tensor decompositions in different tensor formats of the solution of boundary value problems. A new definition of PGD is introduced which allows the progressive construction of a Tucker decomposition of the solution. This tensor format is well adapted to the present application and improves convergence properties of tensor decompositions. Finally, we introduce a dual-based error estimation strategy for assessing the error on quantities of interest, namely the homogenized properties of the material. We also provide a complete goal-oriented adaptive strategy for the progressive construction of tensor decompositions (of primal and dual solutions) yielding to predictions of homogenized quantities with a prescribed accuracy.

Keywords: Image-based computing, Numerical Homogenization, Tensor methods, Proper Generalized Decomposition (PGD), Model Reduction, Goal-oriented error estimation

Introduction

With the development of affordable high resolution imaging techniques, such as X-ray microtomography, high resolution geometrical characterization of material microstructures is increasingly used in industry. However, the amount of informations that are available is still difficult to handle in numerical models. This is why dedicated approaches have been proposed in order to incorporate these informations for simulation purposes

[☆]This work is supported by the French National Research Agency (grant ANR-2010-COSI-006-01).

*Corresponding Author

Email addresses: Loic.Giraldi@ec-nantes.fr (L. Giraldi), Anthony.Nouy@ec-nantes.fr (A. Nouy), Gregory.Legrain@ec-nantes.fr (G. Legrain), Patrice.Cartraud@ec-nantes.fr (P. Cartraud)

[56]. The most used approach in this context is the voxel-based finite element method introduced in [21, 25], where each voxel of the model is transformed into a finite element. The approach is straightforward and automatic for the generation of the computational model (see [42] for a review). However, it leads to huge numerical models, as the number of elements corresponds to the number of voxels in the image (in the order of 8 billion of elements for a full resolution $2000 \times 2000 \times 2000$ voxels CT scan). In addition, the representation of the interfaces is not smooth, which induces local oscillations in the mechanical fields [9, 39, 52]. The size of the model can be decreased with the use of an octree coarsening away from the interfaces [39] or by decreasing the resolution of the image [3, 37, 41]. However, this can severely decrease the geometrical accuracy (more jagged interfaces) and increase the oscillations. In order to get rid of these oscillations, mesh smoothing techniques can be considered, e.g. [6]. Ultimately, full resolution images can still be considered, using Fast Fourier Transforms (FFT) algorithms [43] in the case of periodic problems.

A second class of approaches consists in extracting the material interfaces from the image and then in constructing an unstructured conforming mesh from these informations, e.g. [40, 55, 56]. This allows to generate smooth interfaces and adapt the mesh in order to master the size of the model. However, meshing complex geometries is still difficult and usually requires human guidance.

Finally, non-conforming approaches can be considered (see [13, 53] among others): these approaches allows to avoid meshing issues. In particular, the eXtended Finite Element Method (X-FEM) has been used by the authors for the treatment of 2D and 3D image-based analysis [34, 35, 39]. An integrated approach was proposed in order to incorporate the geometrical informations into the numerical model. It is based on the use of Level-set functions [50], for both segmentation and mechanical analysis. Thanks to the use of tailored enrichment functions, it is possible to represent the interfaces on a non-conforming mesh. The size of the numerical model is decreased thanks to the use of an octree database that enables to keep maximum geometrical accuracy near the interfaces. This allows to obtain a good compromise between easy mesh generation and accuracy (both geometrical and mechanical). More recently, an improvement was proposed by the use of a high-order two mesh strategy that enables high geometrical and mechanical accuracy on coarse meshes [36].

Despite of the improvements in the numerical efficiency of the methods discussed above, image-based computations are still computationaly demanding, leading to time consuming studies especially for large resolution images. There is still a need for new approaches that would allow the efficient resolution of such large scale problems.

This is why an alternative path is proposed in this paper. It relies on the use of tensor approximation methods for the solution of image-based homogenization problems. The basic idea is to interpret 2 or 3-dimensional fields as 2 or 3-order tensors, and to use tensor approximation methods for the approximate solution of boundary value problems. The use of suitable tensor formats allows to drastically reduce the computational costs (time and memory storage) and therefore allows the computation on very high resolution images. This paper provides a complete tensor-based numerical methodology, going from the translation of homogenization problems into a tensor framework, to the development of a goal-oriented adaptive construction of tensor decompositions based on error estimation methods, and dedicated to the present application.

We first translate image-based homogenization problems to a tensor framework by introducing suitable tensor approximations of geometrical data. Tensor approximation methods are applied to indicator functions of material phases, which are previously smoothed in order to improve the convergence properties of their decompositions. Suitable weak formulations of boundary value problems preserving tensor format are introduced in order to handle the different types of boundary conditions that are used in classical numerical homogenization methods. Regarding the construction of tensor approximations of the solution of PDEs, we use Proper Generalized Decomposition methods (PGD), which is a family of methods for the construction of tensor decompositions without a priori information on the solution of the PDE [10, 28, 44, 45] (see [12] for a short review on PGD methods). Theoretical convergence properties have been recently obtained for a class of PGD algorithms [7, 15, 16]. Note that a basic PGD algorithm has been used in [11] for the numerical solution of PDEs with heterogeneous materials whose geometry is easily represented in a tensor format. The method has also been used for deriving efficient non-concurrent non-linear homogenization strategies [31].

Here, we introduce variants of these methods for the construction of tensor decompositions in different tensor formats. In particular, we introduce a new definition of PGD which allows the progressive construction of a Tucker decomposition of the solution. This tensor format is well adapted to the present application and yields to improved convergence properties of tensor decompositions. We finally devise a goal-oriented error estimation strategy in order the assess the error on quantities of interest which are the homogenized properties. Error estimation methods have been first introduced in the context of PGD in [2, 29]. Here, we use a classical dual-based error estimator (see [1]), which has been used in [2] in the context of PGD methods. The originality of the present contribution consists in providing a complete adaptive strategy for the progressive construction of tensor decompositions yielding to predictions of homogenized quantities with a prescribed accuracy. Note that the proposed adaptive strategy could also be used in other context for goal-oriented approximation of PDEs in tensor formats.

The outline is as follows: Section 1 presents the homogenization problems and their variational formulations. Section 2 introduces the tensor framework and notations used for separated representations. Then section 3 presents how the solution of PDEs can be approximated under separated representations with the PGD. In particular, we detail a new algorithm for the progressive construction of a Tucker decomposition. Next, image geometry and boundary conditions are expressed in a tensor format in section 4. First numerical examples are introduced in section 5. Then, in section 6, we introduce a goal-oriented adaptive algorithm using error estimators on homogenized properties. Finally, the article presents an application on a cast iron image extracted from a tomography, where we use the complete goal-oriented adaptive solution method.

1. Homogenization problems and variational formulations

In this section, we introduce classical homogenization methods for a linear heat diffusion problem. Homogenization problems are boundary value problems formulated on a domain Ω which constitutes a representative volume of an heterogeneous material. The solution of these problems allows to extract effective or apparent macroscopic properties of the material depending on whether Ω is larger than the representative volume element

(RVE). Note however that the prediction of the size of the representative volume is out of the scope of this paper. The reader can refer to [23, 24, 38, 47] for methodologies to estimate the size of the representative volume. In the following, we will identify both apparent and homogenized properties.

1.1. Scale transition and localization problems

We denote by u and $\underline{q}$ the temperature and flux fields respectively. The macroscopic gradient of the field $\underline{\nabla} u^M$ and the macroscopic flux $\underline{q}^M$ are defined through a spatial averaging of the corresponding microscopic quantities $\underline{\nabla} u$ and $\underline{q}$ over the representative volume Ω:

$$\underline{\nabla} u^M =< \underline{\nabla} u >= \frac{1}{|\Omega|} \int_\Omega \underline{\nabla} u \; d\Omega \tag{1}$$

$$\underline{q}^M =< \underline{q} >= \frac{1}{|\Omega|} \int_\Omega \underline{q} \; d\Omega \tag{2}$$

The inverse process yielding the microscopic fields from the macroscopic ones is called localization. Given $\underline{\nabla} u^M$ or $\underline{q}^M$, microscopic fields $\underline{\nabla} u$ and $\underline{q}$ are obtained by solving localization problems which are boundary value problems defined on Ω:

$$\begin{cases} \underline{\nabla} \cdot \underline{q}(\underline{x}) = 0 \;\; \text{on} \;\; \Omega \\ \underline{q}(\underline{x}) = -\underline{\underline{K}}(\underline{x}) \cdot \underline{\nabla} u(\underline{x}) \;\; \text{on} \;\; \Omega \\ + \text{ boundary conditions depending on } \underline{\nabla} u^M \text{ or } \underline{q}^M \end{cases} \tag{3}$$

where $\underline{\underline{K}}$ is the conductivity field. If the data is the macroscopic gradient $\underline{\nabla} u^M$ or the macroscopic flux $\underline{q}^M$, we obtain the microscopic fields respectively by

$$\underline{\nabla} u(\underline{x}) = \underline{\underline{A}}(\underline{x}) \cdot \underline{\nabla} u^M \tag{4}$$

or

$$\underline{q}(\underline{x}) = \underline{\underline{B}}(\underline{x}) \cdot \underline{q}^M \tag{5}$$

where $\underline{\underline{A}}$ and $\underline{\underline{B}}$ are called localization tensors. Once we have obtained the microscopic fields through equations (4) (resp. (5)) for a given $\underline{\nabla} u^M$ (resp. $\underline{q}^M$), we can deduce the other macroscopic field $\underline{q}^M$ (resp. $\underline{\nabla} u^M$) by using the constitutive relation and a spatial averaging. We finally define the effective (homogenized) conductivity tensor $\underline{\underline{K}}_h$ such that

$$\underline{q}^M = -\underline{\underline{K}}_h \cdot \underline{\nabla} u^M \tag{6}$$

1.2. Boundary conditions

The homogenized conductivity tensor depends on the localization process which is governed by the choice of the boundary conditions. These boundary conditions are expressed as a function of a macroscopic field which is the input data of the microscopic problem. Classically, three sets of boundary conditions are considered [22–24, 48]. They are described below.

1.2.1. Natural Boundary Conditions (NBC)

Boundary value problem (3) is considered with the following Neumann boundary conditions:

$$\underline{q}(\underline{x}) \cdot \underline{n} = \underline{q}^M \cdot \underline{n} \quad \text{on } \partial\Omega \tag{7}$$

with $\underline{q}^M$ given and $\underline{n}$ the outer pointing normal of $\partial\Omega$. The solution u is a priori defined up to a constant. A well-posed problem can be obtained by introducing the following solution space

$$H^1_m(\Omega) = \left\{ v \in H^1(\Omega); \int_\Omega v \; d\Omega = 0 \right\} \tag{8}$$

The weak formulation of the problem is then:

$$\begin{aligned} &\text{Find } u \in H^1_m(\Omega) \text{ such that} \\ &\forall \delta u \in H^1_m(\Omega), \quad \int_\Omega \underline{\nabla} \delta u \cdot \underline{\underline{K}} \cdot \underline{\nabla} u \; d\Omega = - \int_{\partial\Omega} \delta u \; \underline{q}^M \cdot \underline{n} \; d\Gamma \end{aligned} \tag{9}$$

Using equation (5), we obtain

$$\underline{\nabla} u^M = < \underline{\nabla} u > = < -\underline{\underline{K}}^{-1} \cdot \underline{q} > = - < \underline{\underline{K}}^{-1} \cdot \underline{\underline{B}} > \cdot \underline{q}^M := -(\underline{\underline{K}}^{nbc}_h)^{-1} \cdot \underline{q}^M \tag{10}$$

Then, the solutions obtained from 3 different values of $\underline{q}^M$ (e.g. $(1,0,0)^T$, $(0,1,0)^T$ and $(0,0,1)^T$) yield the complete characterization of $\underline{\underline{K}}^{nbc}_h$.

1.2.2. Periodic Boundary Conditions (PBC)

In this case, problem (3) is considered with the following periodic boundary conditions:

$$\begin{aligned} u(\underline{x}) - \underline{\nabla} u^M \cdot \underline{x} \text{ is } \Omega\text{-periodic} \\ \text{and } \; \underline{q}(\underline{x}) \cdot \underline{n} \text{ is } \Omega\text{-antiperiodic} \end{aligned} \tag{11}$$

with $\underline{\nabla} u^M$ given. We denote by $\tilde{u}(\underline{x}) = u(\underline{x}) - \underline{\nabla} u^M \cdot \underline{x}$. The solution $\tilde{u}$ is a priori defined up to a constant. A well-posed problem can be obtained by introducing the following solution space

$$H^1_{per,m}(\Omega) = \left\{ v \in H^1_{per}(\Omega); \int_\Omega v \; d\Omega = 0 \right\} \tag{12}$$

where $H^1_{per}(\Omega)$ is the subspace of periodic functions in $H^1(\Omega)$. The weak formulation of the problem is then:

$$\begin{aligned} &\text{Find } \tilde{u} \in H^1_{per,m}(\Omega) \text{ such that} \\ &\forall \delta u \in H^1_{per,m}(\Omega), \quad \int_\Omega \underline{\nabla} \delta u \cdot \underline{\underline{K}} \cdot \underline{\nabla} \tilde{u} \; d\Omega = - \int_\Omega \underline{\nabla} \delta u \cdot \underline{\underline{K}} \cdot \underline{\nabla} u^M \; d\Omega \end{aligned} \tag{13}$$

Thanks to the linearity of the problem, $\tilde{u}$ can be written as $\tilde{u} = \underline{\chi} \cdot \underline{\nabla} u^M$ which yields $\underline{\nabla} u = \underline{\nabla} u^M + (\underline{\underline{\nabla}} \; \underline{\chi})^T \cdot \underline{\nabla} u^M = (\underline{\underline{I}} + (\underline{\underline{\nabla}} \; \underline{\chi})^T) \cdot \underline{\nabla} u^M$. Thus, the localization tensor defined in equation (4) is found to be $\underline{\underline{A}} = \underline{\underline{I}} + (\underline{\underline{\nabla}} \; \underline{\chi})^T$. We thus obtain

$$\underline{q}^M = < \underline{q} > = < -\underline{\underline{K}} \cdot \underline{\nabla} u > = - < \underline{\underline{K}} \cdot (\underline{\underline{I}} + \underline{\underline{\nabla}} \; (\underline{\chi})^T) > \cdot \underline{\nabla} u^M := -\underline{\underline{K}}^{pbc}_h \cdot \underline{\nabla} u^M \tag{14}$$

3 different problems have to be solved for obtaining $\underline{\underline{K}}_h^{pbc}$ (e.g. with $\underline{\nabla} u^M = (1,0,0)^T$, $(0,1,0)^T$ and $(0,0,1)^T$). It can be noticed that for a material with a periodic microstructure, the localization problem defined by (3) and boundary conditions (11) can be rigorously justified by the asymptotic expansion method [5, 33, 49]. However, note that periodic boundary conditions can be used even if the microstructure is not periodic (see e.g. [23, 24]).

1.2.3. Essential Boundary Conditions (EBC)

In this case, problem (3) is considered with the following Dirichlet boundary conditions:

$$u(\underline{x}) = \underline{\nabla} u^M \cdot \underline{x} \quad \text{on } \partial\Omega \tag{15}$$

with $\underline{\nabla} u^M$ given. Note that the solution u can be expressed under the form

$$u(\underline{x}) = \underline{\nabla} u^M \cdot \underline{x} + \tilde{u}(\underline{x}) \tag{16}$$

where $\tilde{u}$ is the solution of the following weak formulation:

$$\begin{aligned} &\text{Find } \tilde{u} \in H_0^1(\Omega) \text{ such that} \\ &\forall \delta u \in H_0^1(\Omega),\ \int_\Omega \underline{\nabla} \delta u \cdot \underline{\underline{K}} \cdot \underline{\nabla} \tilde{u}\ d\Omega = -\int_\Omega \underline{\nabla} \delta u \cdot \underline{\underline{K}} \cdot \underline{\nabla} u^M\ d\Omega \end{aligned} \tag{17}$$

By linearity of the problem, 3 different choices for $\underline{\nabla} u^M$ (e.g. $(1,0,0)^T$, $(0,1,0)^T$ and $(0,0,1)^T$) are required to completely characterize the localization tensor $\underline{\underline{A}}$ defined in equation (4). The homogenized tensor $\underline{\underline{K}}_h^{ebc}$ is then obtained by

$$\underline{q}^M = <\underline{q}> = <-\underline{\underline{K}} \cdot \underline{\nabla} u> = -<\underline{\underline{K}} \cdot \underline{\underline{A}}> \cdot \underline{\nabla} u^M := -\underline{\underline{K}}_h^{ebc} \cdot \underline{\nabla} u^M \tag{18}$$

1.3. Unified formulation

Weak formulations (9), (13) and (17) can be unified with an abuse of notation for EBC and PBC, for which $\tilde{u}$ is replaced by u:

$$\begin{aligned} &\text{Find } u \in \mathcal{V} \text{ such that} \\ &\forall \delta u \in \mathcal{V},\ a(\delta u, u) = l(\delta u) \end{aligned} \tag{19}$$

with

$$a(\delta u, u) = \int_\Omega \underline{\nabla} \delta u \cdot \underline{\underline{K}} \cdot \underline{\nabla} u\ d\Omega$$

and

$$\begin{cases} l(\delta u) = -\displaystyle\int_{\partial\Omega} \delta u\ \underline{q}^M \cdot \underline{n}\ d\Gamma & \text{for } \mathbf{NBC} \\ l(\delta u) = -\displaystyle\int_\Omega \underline{\nabla} \delta u \cdot \underline{\underline{K}} \cdot \underline{\nabla} u^M\ d\Omega & \text{for } \mathbf{PBC} \text{ and } \mathbf{EBC} \end{cases} \tag{20}$$

Function space $\mathcal{V}$ is defined by:

$$\begin{cases} \mathcal{V} = H_m^1(\Omega) & \text{for } \mathbf{NBC} \\ \mathcal{V} = H_{per,m}^1(\Omega) & \text{for } \mathbf{PBC} \\ \mathcal{V} = H_0^1(\Omega) & \text{for } \mathbf{EBC} \end{cases} \tag{21}$$

The generic weak formulation (19) will be used in order to simplify the presentation of the proposed solution strategy. We will then come back to the underlying problems for detailing some technical issues and for discussing some specificities of the localization problems.

2. Tensor spaces and separated representations

In this section, we consider a scalar field of interest $w : \Omega \to \mathbb{R}$ which can be the gray intensity level of the image representing the heterogeneous material, a component of the conductivity field $\underline{\underline{K}}$ or the solution of localization problem (19). The domain $\Omega \subset \mathbb{R}^d$ being an image, it has a product structure. We have $\Omega = \Omega^x \times \Omega^y \times \Omega^z$ for $d = 3$ (resp. $\Omega = \Omega^x \times \Omega^y$ for $d = 2$), with $\Omega^\mu = (0, l^\mu)$. In this section, and without loss of generality, we only consider the 3-dimensional case but the different notions extend naturally to arbitrary dimension d. Under some regularity assumptions, a function w defined on such a cartesian domain can be identified with an element of a suitable tensor space. In this section, we introduce some general notions about tensors (as elements of tensor product spaces) and their approximation using separated representations of the form:

$$w(x, y, z) \approx w_m(x, y, z) = \sum_{i=1}^{m} v_i^x(x) v_i^y(y) v_i^z(z) \tag{22}$$

2.1. Tensor spaces

For a general introduction to tensor spaces and tensor approximations, the reader can refer to Hackbusch [18]. We suppose that $w \in \mathcal{V}$ where $\mathcal{V}$ is a Hilbert tensor space defined in the set of functions $\mathbb{R}^\Omega$. The inner product of $\mathcal{V}$ is denoted $\langle \cdot, \cdot \rangle$ and its associated norm $\| \cdot \|$. With $\mathcal{V}^x$ (resp. $\mathcal{V}^y$, $\mathcal{V}^z$) a Hilbert space of functions of $\mathbb{R}^{\Omega^x}$ (resp. $\mathbb{R}^{\Omega^y}$, $\mathbb{R}^{\Omega^z}$), the tensor product $\otimes$ between functions is defined in a usual way, such that

$$(v^x \otimes v^y \otimes v^z)(x, y, z) = v^x(x) v^y(y) v^z(z) \tag{23}$$

for all $v^\mu \in \mathcal{V}^\mu$, $\mu \in \{x, y, z\}$. We define the set of rank-1 (or elementary) tensors

$$\mathcal{C}_1 = \{ v = v^x \otimes v^y \otimes v^z ; v^\mu \in \mathcal{V}^\mu, \mu \in \{x, y, z\} \}, \tag{24}$$

The algebraic tensor product $\otimes_a$ of spaces $\mathcal{V}^x$, $\mathcal{V}^y$ and $\mathcal{V}^z$ is then defined by

$$\mathcal{V}^x \otimes_a \mathcal{V}^y \otimes_a \mathcal{V}^z = \mathrm{span}(\mathcal{C}_1) \tag{25}$$

Finally, the Hilbert tensor space is defined by

$$\mathcal{V} = \mathcal{V}^x \otimes \mathcal{V}^y \otimes \mathcal{V}^z = \overline{\mathcal{V}^x \otimes_a \mathcal{V}^y \otimes_a \mathcal{V}^z}^{\|\cdot\|} \tag{26}$$

where $\overline{(\cdot)}^{\|\cdot\|}$ denotes the completion with respect to the norm $\| \cdot \|$.

In the case of a cartesian domain $\Omega = \Omega^x \times \Omega^y \times \Omega^z$, tensor spaces $\mathcal{V}$ that are involved in the formulation of localization problems (19), and given in (21), have the following tensor product structure:

$$\mathcal{V} = \overline{\mathcal{V}^x \otimes_a \mathcal{V}^y \otimes_a \mathcal{V}^z}^{\|\cdot\|_{H^1(\Omega)}} \tag{27}$$

with

$$\begin{cases} \mathcal{V} = H^1_m(\Omega), & \mathcal{V}^\mu = H^1_m(\Omega^\mu) & \text{for } \mathbf{NBC} \\ \mathcal{V} = H^1_{per,m}(\Omega), & \mathcal{V}^\mu = H^1_{per,m}(\Omega^\mu) & \text{for } \mathbf{PBC} \\ \mathcal{V} = H^1_0(\Omega), & \mathcal{V}^\mu = H^1_0(\Omega^\mu) & \text{for } \mathbf{EBC} \end{cases} \tag{28}$$

This justifies the existence of a separated representation of type (22) for the solution of homogenization problems (19).

2.2. Tensor decompositions

We introduce here two classical tensor decomposition formats.

Rank-m (canonical) decomposition. For a given $m \in \mathbb{N}$, the set of rank-m tensors $\mathcal{C}_m$ is defined by

$$\mathcal{C}_m = \left\{ w_m = \sum_{i=1}^{m} v_i; v_i \in \mathcal{C}_1 \right\} \tag{29}$$

Definition (26) implies that for all $w \in \mathcal{V}$, there exists a sequence of tensors, $(w_m)_{m\in\mathbb{N}}$, such that $w_m \in \mathcal{C}_m$ and w_m converges to w. Therefore, for any $w \in \mathcal{V}$, this condition justifies the existence of the approximation of type (22) with an arbitrary accuracy.

Tucker tensors. For a given multi-index $\mathbf{r} = (r^x, r^y, r^z) \in \mathbb{N}^3$, we define the Tucker tensors set $\mathcal{T}_\mathbf{r}$ as follows:

$$\mathcal{T}_\mathbf{r} = \left\{ w_\mathbf{r} = \sum_{i=1}^{r^x} \sum_{j=1}^{r^y} \sum_{k=1}^{r^z} \lambda_{ijk}\ v^x_i \otimes v^y_j \otimes v^z_k; \lambda_{ijk} \in \mathbb{R}, v^\mu_p \in \mathcal{V}^\mu, \langle v^\mu_p, v^\mu_q \rangle_\mu = \delta_{pq} \right\} \tag{30}$$

with $\langle \cdot, \cdot \rangle_\mu$ the inner product on Hilbert space $\mathcal{V}^\mu$. An approximation $w_\mathbf{r} \in \mathcal{T}_\mathbf{r}$ of an element $w \in \mathcal{V}$ is called a rank-$\mathbf{r}$ Tucker approximation of w. We have the property that $\mathcal{C}_r \subset \mathcal{T}_{(r,r,r)}$. Therefore, for all $w \in \mathcal{V}$, there also exists a sequence $(w_m)_{m\in\mathbb{N}}$, with $w_m \in \mathcal{T}_{\mathbf{r}_m}$ and such that w_m converges to w. The reader can refer to [27] for a review of definitions and constructions of tensor representations in finite dimensional algebraic tensor spaces $\mathcal{V} = \mathbb{R}^{N_x} \otimes \mathbb{R}^{N_y} \otimes \mathbb{R}^{N_z}$.

3. Proper Generalized Decomposition

The homogenization problem we want to solve has been recasted as follows (see section 1.3):

$$\begin{aligned} &\text{Find } u \in \mathcal{V} \text{ such that} \\ &\forall \delta u \in \mathcal{V},\ a(\delta u, u) = l(\delta u) \end{aligned} \tag{31}$$

where $\mathcal{V}$ is a tensor Hilbert space with tensor structure given in (28). Proper Generalized Decomposition (PGD) methods constitute a family of algorithms for the construction of a tensor approximation of the solution u of (31), without a priori information on the solution. It can be achieved by formulating best approximation problems on tensors subsets using operator-based norms instead of natural norms in tensor space $\mathcal{V}$. In this section, we first recall the principle of PGD methods. We then introduce a now classical algorithm for the construction of rank-m approximations and we recall some properties

of this approximation. Then, we will introduce a new algorithm for the progressive construction of a Tucker representation of the solution. This algorithm provides better convergence properties than classical PGD definitions based on canonical decompositions.

Remark 1. *In this section, we consider that the problems are formulated on a d-dimensional domain $\Omega \subset \mathbb{R}^d$, with $\Omega = \Omega^x \times \Omega^y \times \Omega^z$ for $d = 3$ or $\Omega = \Omega^x \times \Omega^y$ for $d = 2$. The notation $\otimes_\mu$ will stand for $\otimes_{\mu\in\{x,y,z\}}$ for $d = 3$ and $\otimes_{\mu\in\{x,y\}}$ for $d = 2$.*

3.1. A priori definition of an approximation on a tensor subset

Let us consider a tensor subset $\mathcal{M}$ (e.g. rank-1 tensors set, Tucker tensors set...). Given a norm $\|\cdot\|$ in $\mathcal{V}$, a best approximation $u^* \in \mathcal{M}$ of $u \in \mathcal{V}$ can be naturally defined as

$$\|u - u^*\| = \min_{v\in\mathcal{M}} \|u - v\| \tag{32}$$

The idea is to find an approximation u^* of u without a priori information on u. Therefore, the norm must be chosen in such a way that problem (32) can be solved without knowing u. In the present application, where u is solution of (31) with a a symmetric coercive and continuous bilinear form, the norm $\|\cdot\|$ and associated inner product $\langle\cdot,\cdot\rangle$ can be chosen as follows:

$$\|v\|^2 = a(v, v), \quad \langle u, v\rangle = a(u, v)$$

where the norm $\|\cdot\|$ is equivalent to the initial norm on $\mathcal{V}$. We then have

$$\|u - v\|^2 = a(u, u) + a(v, v) - 2a(u, v) = a(u, u) + a(v, v) - 2l(v),$$

and minimization problem (32) is then equivalent to

$$J(u^*) = \min_{v\in\mathcal{M}} J(v), \quad J(v) = \frac{1}{2}a(v, v) - l(v) \tag{33}$$

We note that problem (33) that defines the tensor approximation in $\mathcal{M}$ does not involve the solution u. It makes the approximation u^* computable without a priori information on u. Of course, tensor subsets $\mathcal{M}$ must be such that minimization problems on $\mathcal{M}$ are well-posed. Moreover, tensor subsets $\mathcal{M}$ (such as rank-1 tensors, rank-$\mathbf{r}$ Tucker tensors) are manifolds which are not necessarily linear spaces, so that even if problem (31) is a linear approximation problem, minimization problem (33) is no more a linear approximation problem and specific algorithms have to be devised.

Remark 2. *Note that a necessary condition for optimality of u^* writes*

$$a(u^*, \delta v) = l(\delta v) \quad \forall \delta v \in T_{u^*}(\mathcal{M}) \tag{34}$$

where $T_{u^*}(\mathcal{M})$ is the tangent linear space to $\mathcal{M}$ at u^*.

3.2. Construction of a rank-m (canonical) decomposition

The aim is to find an optimal rank-m approximation $u_m \in \mathcal{C}_m$ of u of the form

$$u_m = \sum_{i=1}^{m} v_i \tag{35}$$

with $v_i \in \mathcal{C}_1$. A direct definition of an optimal approximation in $\mathcal{C}_m$ would be defined by the optimization problem (32). However, it is well known that for $m \geq 2$ and $d \geq 3$, $\mathcal{C}_m$ is not a weakly closed set so that the minimization problem (32) is ill-posed for $\mathcal{M} = \mathcal{C}_m$ [15]. A modified direct definition of an approximation with arbitrary precision $\varepsilon > 0$ could be formulated as follows:

$$\begin{aligned} &\text{Find } u_m \text{ such that} \\ &\|u - u_m\| \leq \inf_{v \in \mathcal{C}_m} \|u - v\| + \varepsilon \end{aligned} \tag{36}$$

The solution of this problem yields the whole set of rank-1 elements $\{v_i\}_{i=1}^m$ at once. In order to avoid the introduction of the tolerance ε (i.e. $\varepsilon = 0$), it could be possible to consider a weakly closed subset of $\mathcal{C}_m$ (e.g. by adding some orthogonality constraints between rank-one elements). However, these direct constructions are computationnally expensive since when increasing m, optimization problems are defined on sets with increasing dimensionality and computational complexity drastically increases.

In order to circumvent the above difficulties, a progressive definition of PGD is classically introduced [10, 15], which consists in building the sequence $(v_i)_{1 \leq i \leq m}$ term by term (in a greedy fashion). Even if the progressive definition is sub-optimal compared to the direct PGD approach, $\mathcal{C}_1$ is weakly closed [15] so that successive best approximation problems on $\mathcal{C}_1$ are well-posed. Moreover, optimization problems on $\mathcal{C}_1$ have almost the same complexity, so that the complexity of computing a rank-m approximation scales (approximately) linearly with the rank m. The algorithm used is defined in Algorithm 1. Step 5 of Algorithm 1 corresponds to a classical projection onto the subspace spanned by

Algorithm 1 Progressive construction of a rank-m approximation

1: Set $u_0 = 0$
2: **for** $i = 1$ to m **do**
3: Compute $w_i = \otimes_\mu w_i^\mu \in \arg\min_{w \in \mathcal{C}_1} \|u - u_{i-1} - w\|$
4: Set $\mathcal{U}_i = \text{span}\{w_k\}_{k=1}^i$
5: Compute $u_i = \arg\min_{v \in \mathcal{U}_i} \|u - v\|$
6: **end for**

all rank-one elements previously generated. At step m, the solution $u_m = \arg\min_{v \in \mathcal{U}_m} \|u - v\|$ can be written under the form

$$u_m = \sum_{i=1}^m \lambda_i w_i \quad \text{with } w_i = \otimes_\mu w_i^\mu$$

where the $(\lambda_i)_{i=1}^m \in \mathbb{R}^m$ are coefficients that are solutions of the following system of m equations:

$$\sum_{j=1}^m A_{ij} \lambda_j = b_i, \quad \forall i \in \{1, \ldots, m\} \tag{37}$$

with

$$A_{ij} = a(w_i, w_j) = a(\otimes_\mu v_i^\mu, \otimes_\mu v_j^\mu), \quad b_i = l(w_i) = l(\otimes_\mu v_i^\mu)$$

Different algorithms have been proposed for computing $w_m \in \arg\min_{v\in\mathcal{C}_1} \|u - u_{m-1} - v\|$ (step 3). One possibility is an alternating minimization algorithm presented in Algorithm 2. The integer k_{max} represents the maximum number of iterations and the tolerance ε_s is associated to a stagnation criterium. In practice, we will take $k_{max} = 10$ and $\varepsilon_s = 5.10^{-2}$.

Algorithm 2 Alternating minimization algorithm for computing $w_m \in \arg\min_{w\in\mathcal{C}_1} \|u - u_{m-1} - w\|$

1: Initialize randomly $v^\mu \in \mathcal{V}^\mu$ for all μ
2: Set $w_0 = 0$
3: **for** $k = 1$ to k_{max} **do**
4: **for** $\mu = x, \ldots$ **do**
5: Compute $v^\mu \in \arg\min_{v^\mu\in\mathcal{V}^\mu} \|u - u_{m-1} - \otimes_\beta v^\beta\|$
6: **end for**
7: Set $w_k = \otimes_\mu v^\mu$
8: **if** $\|w_k - w_{k-1}\|/\|w_{k-1}\| < \varepsilon_s$ **then**
9: **break**
10: **end if**
11: **end for**
12: Set $w_m = w_k$

Remark 3. *In the case where $\|\cdot\|$ is the classical canonical norm on a finite dimensional space, Algorithm 2 corresponds to the Alternating Least Squares (ALS) algorithm on $\mathcal{C}_1$ [8, 20, 27].*

Remark 4. *If the domain Ω is 2-dimensional, the method can be considered as a generalization of the SVD with respect to general norms on Hilbert spaces [15].*

Minimization problem of step 5 of Algorithm 2, corresponding to the computation of v^μ, is equivalent to

$$\min_{v^\mu\in\mathcal{V}^\mu} \frac{1}{2} a(\otimes_\beta v^\beta, \otimes_\beta v^\beta) + a(u_{m-1}, \otimes_\beta v^\beta) - l(\otimes_\beta v^\beta) \tag{38}$$

It reduces to the solution of the following linear problem:

$$\begin{aligned} &\text{Find } v^\mu \in \mathcal{V}^\mu \text{ such that} \\ &\forall \delta\varphi \in \mathcal{V}^\mu,\ a^\mu(\delta\varphi, v^\mu) = l^\mu_m(\delta\varphi) \end{aligned} \tag{39}$$

with

$$\begin{aligned} a^\mu(\delta\varphi, \varphi) &= a(\delta\varphi \otimes (\otimes_{\beta\neq\mu} v^\beta), \varphi \otimes (\otimes_{\beta\neq\mu} v^\beta)), \\ l^\mu_m(\delta\varphi) &= l(\delta\varphi \otimes (\otimes_{\beta\neq\mu} v^\beta)) - a(\delta\varphi \otimes (\otimes_{\beta\neq\mu} v^\beta), u_{m-1}) \end{aligned}$$

Problem (39) is a one-dimensional problem defined on $\mathcal{V}^\mu \subset H^1(\Omega^\mu)$. The proposed algorithm then only involves the solution of uncoupled one-dimensional problems, which is the reason for the efficiency of the PGD method.

Remark 5. *Assuming that $\mathcal{V}^\mu = \mathbb{R}^n$, $\forall \mu$, and that the complexity of a linear solver on a problem of size n is n^3, then the construction of all the vectors of a rank-r decomposition with algorithm 2 requires at most $r\, k_{max}\, d\, n^3$ operations. The update of the λ's requires $1 + 2^3 + \ldots + r^3 = \frac{1}{4} r^2 (r+1)^2$ operations. Finally the complexity of the algorithm 1 is $\frac{1}{4} r^2 (r+1)^2 + r\, k_{max}\, d\, n^3$.*

3.3. Construction of a rank-$\mathbf{r}$ Tucker decomposition

An approximation of the solution using a Tucker format could be searched by solving an optimization problem:

$$\|u - u_{\mathbf{r}}\| = \min_{v \in \mathcal{T}_{\mathbf{r}}} \|u - v\| \tag{40}$$

This problem is well-posed [14]. However, this direct construction of a Tucker approximation $u_{\mathbf{r}} \in \mathcal{T}_{\mathbf{r}}$ is computationally expensive and when the objective is to find an approximation with a desired accuracy, we will rather prefer an adaptive construction.

We now propose a sub-optimal but progressive construction of a sequence of Tucker approximations $\{u_m\}_{m \in \mathbb{N}}$ of the solution, where u_m is the best approximation of the solution in a linear subspace $\mathcal{U}_m$ of $\mathcal{T}_{(m,\ldots,m)}$, with $\mathcal{U}_m$ of the form:

$$\mathcal{U}_m = \otimes_\mu \mathcal{U}_m^\mu \tag{41}$$

that means $\mathcal{U}_m = \mathcal{U}_m^x \otimes \mathcal{U}_m^y \otimes \mathcal{U}_m^z$ for dimension $d = 3$ or $\mathcal{U}_m^x \otimes \mathcal{U}_m^y$ for dimension $d = 2$. Linear subspaces $\mathcal{U}_m^\mu \subset \mathcal{V}^\mu$ are defined progressively, by completing the previous spaces $\mathcal{U}_{m-1}^\mu$ by fonctions w_m^μ extracted from a rank-one corrections $w_m = \otimes_\mu w_m^\mu$ of u_{m-1}. The sequences of linear spaces $\{\mathcal{U}_m^\mu\}_{m \in \mathbb{N}}$ (and $\{\mathcal{U}_m\}_{m \in \mathbb{N}}$) are increasing with m (with respect to inclusion). Let us detail this procedure.

Let $\mathcal{U}_0^\mu = 0$ and $\mathcal{U}_0 = 0$. At iteration m, knowing $u_{m-1} \in \mathcal{U}_{m-1} = \otimes_\mu \mathcal{U}_{m-1}^\mu$, we start by computing an optimal correction $w_m \in \mathcal{C}_1$ of u_{m-1}, defined by

$$w_m = \otimes_\mu w_m^\mu \in \underset{w \in \mathcal{C}_1}{\operatorname{argmin}} \|u - u_{m-1} - w\| \tag{42}$$

The new linear subspace $\mathcal{U}_m^\mu$ is then defined by

$$\mathcal{U}_m^\mu = \mathcal{U}_{m-1}^\mu + span\{w_m^\mu\}$$

which has a dimension $dim(\mathcal{U}_m^\mu) := r_m^\mu \leq m$. The linear space $\mathcal{U}_m$ is then defined by (41) and has a dimension $dim(\mathcal{U}_m) := r_m = \prod_\mu r_m^\mu$. The next approximation $u_m \in \mathcal{U}_m$ is then defined by the following linear approximation problem:

$$\|u - u_m\| = \min_{v \in \mathcal{U}_m} \|u - v\| \tag{43}$$

This procedure requires the solution of a succession of best approximation problems (42) on the manifold $\mathcal{C}_1$, which are nonlinear approximation problems with moderate complexity, and a succession of best approximation problems on linear subspaces $\mathcal{U}_m$, which are approximation problems with increasing complexity r_m. However, these latter problems remain classical linear approximation problems. The algorithm is summed up in Algorithm 3. Step 6 (best approximation in $\mathcal{C}_1$) can be accomplished thanks to the

Algorithm 3 Progressive construction of the Tucker decomposition

Set $u_0 = 0$
for $\mu = x, \dots$ **do**
  Set $U_0^\mu = 0$
end for
for $i = 1$ to m **do**
  Compute $w_i = \otimes_\mu w_i^\mu \in \arg\min_{w \in \mathcal{C}_1} \|u - u_{i-1} - w\|$
  for $\mu = x, \dots$ **do**
    Set $\mathcal{U}_i^\mu = \mathcal{U}_{i-1}^\mu + \text{span}(w_i^\mu)$
  end for
  Compute $u_i = \arg\min_{v \in \mathcal{U}_i} \|u - v\|$
end for

alternating minimization algorithm (Algorithm 2). In practice, an orthonormal basis $\{v_j^\mu\}_{1 \le j \le r_m^\mu}$ of $\mathcal{U}_m^\mu$ is built using the Gram-Schmidt procedure applied to the generating set $\{w_j^\mu\}_{1 \le j \le m}$. The approximation u_m, solution of (43), can be written under the form

$$u_m = \sum_{\boldsymbol{i} \in \mathcal{I}_m} \lambda_{\boldsymbol{i}} v_{\boldsymbol{i}}, \quad v_{\boldsymbol{i}} = \otimes_\mu v_{i_\mu}^\mu$$

with $\mathcal{I}_m = \{\boldsymbol{i} \in \mathbb{N}^d; 1 \le i_\mu \le r_m^\mu\}$. The set of coefficients $(\lambda_{\boldsymbol{i}})_{\boldsymbol{i} \in \mathcal{I}_m}$ is solution of a linear system of equations

$$\sum_{\boldsymbol{j} \in \mathcal{I}_m} A_{\boldsymbol{ij}} \lambda_{\boldsymbol{j}} = b_{\boldsymbol{i}}, \quad \boldsymbol{i} \in \mathcal{I}_m \tag{44}$$

with

$$A_{\boldsymbol{ij}} = a(v_{\boldsymbol{i}}, v_{\boldsymbol{j}}) = a(\otimes_\mu v_{i_\mu}^\mu, \otimes_\mu v_{j_\mu}^\mu), \quad b_{\boldsymbol{i}} = l(v_{\boldsymbol{i}}) = l(\otimes_\mu v_{i_\mu}^\mu)$$

Remark 6. *The orthonormalization of basis functions allows to detect if at step m, a new function w_m^μ is contained in the previous linear space U_{m-1}^μ, in which case $U_m^\mu = U_{m-1}^\mu$ and $r_m^\mu = r_{m-1}^\mu$. This allows to ensure the uniqueness of the best approximation u_m in U_m (regularity of system (44)). Also, it allows to automatically detect and take part of the independence (or quasi-independence) of the solution with respect to a certain variable.*

Remark 7. *Again, assuming that $\mathcal{V}^\mu = \mathbb{R}^n$, $\forall \mu$, and that the complexity of a linear solver on a problem of size n is n^3, then the construction of all the vectors of a rank-$(r_1, \dots, r_d)$ decomposition requires at most $r\, k_{max}\, d\, n^3$ operations, for $r_1 = \dots = r_d = r$. The update of the core tensor μ requires at most $1 + 2^{3d} + \dots + r^{3d} \le r^{3d+1}$ operations. Finally, the complexity of the algorithm 3 is bounded by $r^{3d+1} + r\, k_{max}\, d\, n^3$. This indicates that we should use this method in 2D or when the geometry presents particular symmetries such that one or two r_μ are small compared to the others, like in section 5.2 for instance.*

4. Tensor format for image-based homogenization problems

In order to apply the PGD method to the solution of numerical homogenization problems, specific reformulations and approximations have to be introduced in order to recast the problem in a suitable format adapted to tensor-based methods. These specific

treatments concern the separated representation of the conductivity field $\underline{\underline{K}}$, yielding an approximation of operator under a tensor format, and the introduction of suitable reformulations for imposing the different types of boundary conditions.

In [30, 31], the authors already introduced a similar computational method for homogenization in the EBC case and for simple geometries. In the present contribution, these works are extended to the different boundary conditions that are classically used in computational homogenization, which necessitates the introduction of suitable reformulations of the variational problem. Moreover, we introduce specific numerical treatments for dealing with real geometries extracted from images.

4.1. Separated representation of the conductivity field

Let us consider a composite material with two phases numbered 1 and 2 in which the conductivity is constant and is denoted $\underline{\underline{K}}_1$ and $\underline{\underline{K}}_2$ respectively. If $I : \Omega \to \{0, 1\}$ is the characteristic function of phase 1, the local conductivity $\underline{\underline{K}}$ is rewritten

$$\underline{\underline{K}}(\underline{x}) = I(\underline{x})\underline{\underline{K}}_1 + (1 - I(\underline{x}))\,\underline{\underline{K}}_2 = \underline{\underline{K}}_2 + I(\underline{x})\left(\underline{\underline{K}}_1 - \underline{\underline{K}}_2\right)$$

Therefore, a separated representation of the scalar field $I \in L^2(\Omega) = \otimes_\mu L^2(\Omega^\mu)$ yields a separated representation of $\underline{\underline{K}}$. Suppose that the image contains $P^x \times P^y \times P^z$ voxels for $d = 3$, or $P^x \times P^y$ pixels for $d = 2$. For $d = 3$, the discrete characteristic function can be written

$$I(x, y, z) = \sum_{i=1}^{P^x} \sum_{j=1}^{P^y} \sum_{k=1}^{P^z} I_{ijk} \phi_i^x(x) \phi_j^y(y) \phi_k^z(z) \tag{45}$$

where I_{ijk} is the value of I in the voxel (i, j, k) and where the $\{\phi_i^\mu\}_{1 \le i \le P^\mu}$ are piecewise constant interpolation functions. For $d = 2$, we can write

$$I(x, y) = \sum_{i=1}^{P^x} \sum_{j=1}^{P^y} I_{ij} \phi_i^x(x) \phi_j^y(y) \tag{46}$$

where I_{ij} is the value of I in the pixel (i, j). We then identify function I with $\mathbf{I} = (I_{ijk}) \in \mathbb{R}^{P^x} \otimes \mathbb{R}^{P^y} \otimes \mathbb{R}^{P^z}$ for $d = 3$ (resp. $\mathbf{I} = (I_{ij}) \in \mathbb{R}^{P^x} \otimes \mathbb{R}^{P^y}$ for $d = 2$).

In this section, we propose to slightly smooth the geometry in order to obtain a lower rank tensor decomposition. This can be seen as a particular approximation of the geometry. Note that voxel-based, remeshing or level-set techniques also introduce approximations which result in different representations of the actual geometry.

4.1.1. Singular value decompositions

We now equip $\otimes_\mu \mathbb{R}^{P^\mu}$ with the canonical inner product, which is defined for $\mathbf{X}, \mathbf{Y} \in \otimes_\mu \mathbb{R}^{P^\mu}$ by

$$\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i=1}^{P^x} \sum_{j=1}^{P^y} X_{ij} \cdot Y_{ij} \qquad \text{for } d = 2$$

$$\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i=1}^{P^x} \sum_{j=1}^{P^y} \sum_{k=1}^{P^z} X_{ijk} \cdot Y_{ijk} \qquad \text{for } d = 3$$

The associate norm is denoted $\|\cdot\|$. For $d = 2$, it is well known that the best rank-m approximation of $\mathbf{I}$ with respect to the norm $\|\cdot\|$ is the classical Singular Value Decomposition (SVD) truncated at rank m. For $d = 3$, several alternatives have been proposed for extending the concept of SVD. One could apply the progressive PGD approach presented in section 3, with the canonical norm $\|\cdot\|$, to obtain a separated representation of $\mathbf{I}$. However, much more efficient constructions have been proposed for tensor decomposition in $\mathbb{R}^{P^x} \otimes \mathbb{R}^{P^y} \otimes \mathbb{R}^{P^z}$, the tensor to be decomposed being known a priori [27]. For example, the ALS algorithm computes an approximation of $\mathbf{I}$ on $\mathcal{C}_m$ by approximating the best approximation problem on $\mathcal{C}_m$. We can also mention the Higher-Order Singular Value Decomposition (HOSVD) that defines an approximation of $\mathbf{I}$ in the Tucker set $\mathcal{T}_{\mathbf{r}}$. An implementation of these methods can be found in the MATLAB$^{\text{TM}}$ Tensor Toolbox [4]. Note that the above definitions can be considered as particular cases of PGD methods for particular choices of norm. In fact, for the above particular norm (which is a crossnorm), these methods can be considered as multidimensional versions of the SVD.

Finally, we obtain a separated approximation $\mathbf{I}_m$ of $\mathbf{I}$ in $\mathcal{C}_m$ (or eventually in $\mathcal{T}_{\mathbf{r}}$ for the Tucker format). The error $\varepsilon = \|\mathbf{I} - \mathbf{I}_m\|/\|\mathbf{I}\|$ can be controlled in order to provide a desired accuracy on the description of the geometry. Note that the approximation $\mathbf{I}_m$ can be identified with an approximation I_m of I in the tensor space $L^2(\Omega)$.

4.1.2. Regularized indicator functions

When computing a separated decomposition I_m of I using classical tensor decompositions, we observe that the number of terms needed to have a good representation of I is highly dependent on the regularity properties of function I. The indicator function, which presents strong discontinuities at the material interfaces, is a typical example of an irregular function which yields a bad convergence of separated representations. In order to circumvent this convergence issue, the image is slightly smoothed before using tensor approximations. In practice, the iso zero of a level-set ϕ defined from the image is used to represent the interfaces. A smoothed indicator function I_s is obtained by the following operation

$$I_s = \frac{1}{2}\left(1 + \tanh\left(\frac{2\phi}{\delta}\right)\right) \tag{47}$$

where δ represents a characteristic length. This formula is applied on the whole level-set, no matter the distance to the interfaces. The effect of smoothing is illustrated on figure 1.

4.1.3. Illustration

We here illustrate the impact of the regularization of the indicator function on a 2D example. We consider the 2D image I with 512×512 pixels represented in figure 2. A very small smoothing is applied ($\delta = 2$ pixels). In this 2-dimensional case, the rank-m tensor approximation corresponds to a rank-m truncated SVD. On figure 3, we illustrate the convergence of SVD applied to the original image I or to the smoothed image I_s. This figure shows that even for a small smoothing, we obtain a faster convergence of the decomposition. Indeed, for a fixed error ϵ of 10^{-2}, about 100 additional modes are necessary for the decomposition of the original image. The smoothing then yields to lower rank representations and therefore to more efficient tensor-based algorithms. However, the smoothing introduces an approximation of the geometry that has to be controlled.

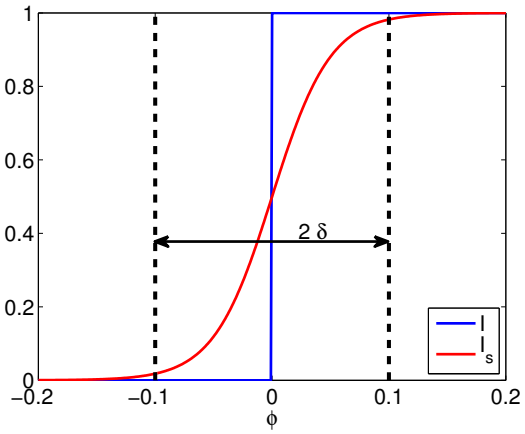


Figure 1: Smoothing effect on the characteristic function

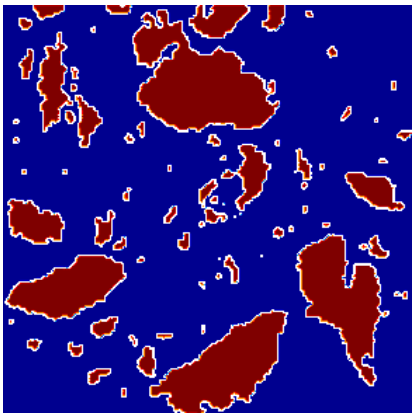


Figure 2: 2D image taken from [17]

Note that another advantage of the smoothing of the characteristic function is that it removes the oscillations at the boundaries and prevents the approximation I_m to be negative, which would lead to a negative conductivity in some regions of the domain and therefore to an ill-posed boundary value problem.

4.2. Tensor format of boundary conditions

4.2.1. Natural Boundary conditions (NBC)

Problem (9) is reformulated as follows:

$$\text{Find } u \in H^1(\Omega) \text{ such that } \forall \delta u \in H^1(\Omega)$$

$$\int_\Omega \underline{\nabla} \delta u \cdot \underline{\underline{K}} \cdot \underline{\nabla} u \; d\Omega + \gamma \int_\Omega \delta u \; d\Omega \int_\Omega u \; d\Omega = - \int_{\partial\Omega} \delta u \; \underline{q}^M \cdot \underline{n} \; d\Gamma \tag{48}$$

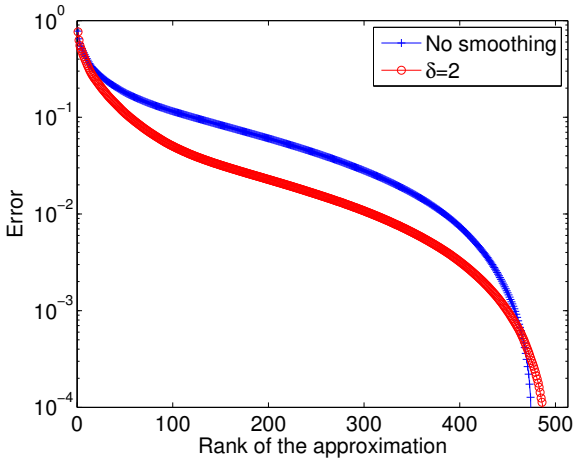


Figure 3: Convergence in L^2 norm of the separated representations of the initial and smoothed images with respect to their rank.

with $\gamma > 0$. For any γ, the solution of (48) verifies $\int_\Omega u \, d\Omega = 0$ and therefore, $u \in H^1_m(\Omega)$ is the solution of the initial problem. We solve d problems associated with $\underline{\nabla} u^M = \underline{e}_\mu$, where $\{e_\mu\}$ is the canonical basis of $\mathbb{R}^d$. The right-hand side can be expressed as a sum over the boundary of the cartesian domain. The normal flux $\underline{q}^M \cdot \underline{n}$ being constant on each face, each term in the right-hand side is a rank-one tensor, so that the resulting right-hand side can be represented by rank-2^d tensor, where 2^d is the number of boundary faces in dimension d.

4.2.2. Periodic Boundary Conditions (PBC)

Problem (13) is first reformulated as follows:

$$\begin{aligned} &\text{Find } \tilde{u} \in H^1_{per}(\Omega) \text{ such that } \forall \delta u \in H^1_{per}(\Omega) \\ &\int_\Omega \underline{\nabla} \delta u \cdot \underline{\underline{K}} \cdot \underline{\nabla} \tilde{u} \; d\Omega + \gamma \int_\Omega \delta u \; d\Omega \int_\Omega u \; d\Omega = - \int_\Omega \underline{\nabla} \delta u \cdot \underline{\underline{K}} \cdot \underline{\nabla} u^M \; d\Omega \end{aligned} \tag{49}$$

with $\gamma > 0$. For any γ, the solution of (49) verifies $\int_\Omega u \; d\Omega = 0$ and therefore, $u \in H^1_{per,m}(\Omega)$ is the solution of the initial problem. The term $\underline{\nabla} u^M$ is uniform over Ω. The tensor format of the right-hand side then follows directly from the tensor format of the conductivity field. We solve d problems associated with $\underline{\nabla} u^M = \underline{e}_\mu$, where $\{e_\mu\}$ is the canonical basis of $\mathbb{R}^d$.

The Ω-periodicity of tensor approximations of type $u_m = \sum_{i=1}^m \otimes_\mu v_i^\mu$ is naturally obtained by imposing Ω^μ-periodicity for all 1-dimensional functions v_i^μ.

4.2.3. Essential Boundary conditions (EBC)

The weak formulation of the boundary value problem associated with EBC is (17). $\underline{\nabla} u^M$ being uniform on Ω, the tensor format of the right-hand side follows directly from the tensor format of the conductivity field. We solve d problems associated with $\underline{\nabla} u^M = \underline{e}_\mu$, where $\{e_\mu\}$ is the canonical basis of $\mathbb{R}^d$.

Finally, we impose homogeneous boundary conditions for all 1-dimensional functions $v_i^\mu \in H_0^1(\Omega^\mu)$ in tensor approximations of type $u_m = \sum_{i=1}^m \otimes_\mu v_i^\mu$, thus imposing homogeneous boundary conditions for $u_m \in H_0^1(\Omega)$.

5. First applications

5.1. PGD using canonical tensor format

The aim is to illustrate the behavior of the progressive PGD method and estimate the impact of PGD approximations on the quality of the quantities of interest which are the homogenized tensors.

We consider the 2D image represented in figure 4. The picture contains 128×128 pixels and is associated with a domain $\Omega = (0,1)^2$. The domain contains random inclusions in a matrix. The materials phases are isotropic with conductivities $k_i = 10W.m^{-1}.K^{-1}$ for the inclusions and $k_m = 1W.m^{-1}.K^{-1}$ for the matrix. We use a characteristic length of $\delta = 1$ pixel for the smoothing of the indicator function.

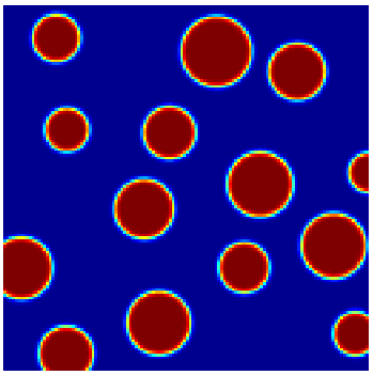
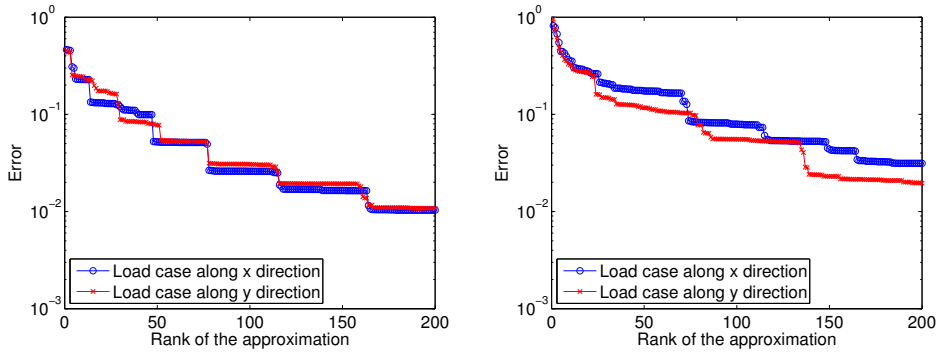


Figure 4: Random inclusions into a matrix

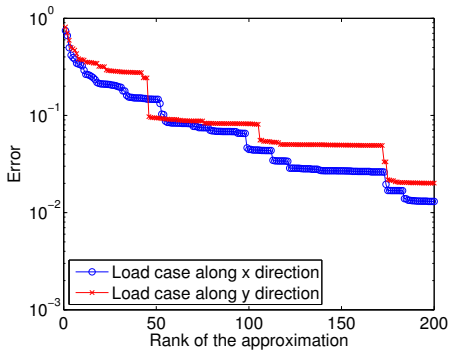
Problem (19) has been solved for NBC, PBC and EBC using a classical Finite Element Method (FEM). These FEM solutions and the corresponding homogenized tensors are taken as reference solutions.

In order to quantify only the error coming from the PGD approximation, we use an exact representation of the original image. This avoids any degradation of the geometry due to truncation of its decomposition. We here apply a progressive PGD algorithm to construct rank-m separated representations of the solutions. On figure 5 is plotted the convergence of the PGD approximations with respect to the rank of the approximation. We observe similar convergence properties for different problems associated with the 3 types of boundary conditions. In fact, tensor decompositions being related to singular value decompositions (spectral decompositions), the observed convergence reflects the spectral content of the solutions. The observed plateaux can be explained by clusters of singular values. Figure 6 shows the convergence of the corresponding estimations of the homogenized tensors.

We can see that in the 3 cases, we have a good convergence of the homogenized tensor with the rank of the approximation. Besides, a slower convergence is observed for the



(a) NBC (b) PBC



(c) EBC

Figure 5: Convergence of the progressive rank-m PGD approximation with respect to rank m. Relative error in L^2 norm.

NBC case. This can be explained by the fact that for the NBC case, high frequency modes in the separated approximations have non negligible spatial means, as opposed to the cases of PBC and EBC.

If we increase the contrast, by taking $k_i = 1000W.m^{-1}.K^{-1}$, we observe on figure 7 a slower convergence. In figure 8, we also observe a slower convergence for the homogenized tensors. The large increase of the contrast deteriorates the conditioning of the operator. In order to improve the present results, preconditioning techniques should be introduced. Note that some preconditioning techniques have already been introduced for operators in tensor format, see e.g. [26, 32, 54]. Preconditioning techniques that are adapted to the present framework are under investigation and will be introduced in a subsequent paper.

5.2. PGD using Tucker format

In this section we compare the convergence between the progressive PGD with canonical tensor format (algorithm 1) with the progressive PGD with Tucker format (algorithm 3).

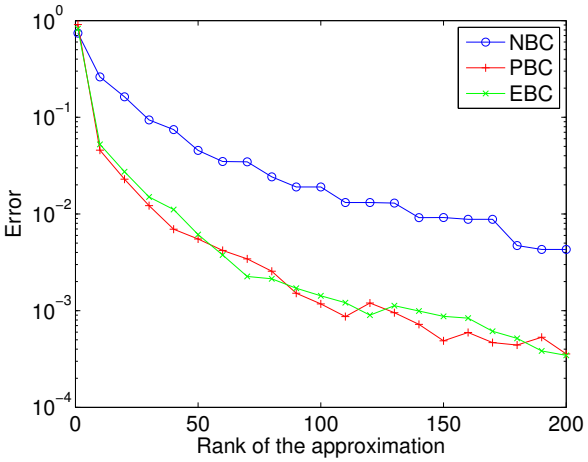


Figure 6: Relative error in canonical norm on the homogenized tensor as a function of the rank of the approximation

We consider the 3D image with 128^3 voxels represented in figure 9. This image is associated with a domain $\Omega = (0,1)^3$ which contains fibers in a matrix. Phases are isotropic with thermal conductivities $k_f = 10W.m^{-1}.K^{-1}$ for the fibers and $k_m = 1W.m^{-1}.K^{-1}$ for the matrix. The image has been separated using the Tucker ALS method from the MATLAB™ Tensor Toolbox [4] with a rank-(119,119,18) Tucker decomposition, automatically taking into account the anisotropy of the original microstructure. The relative error in L^2 norm between the separated representation and the real characteristic function is lower than 1%.

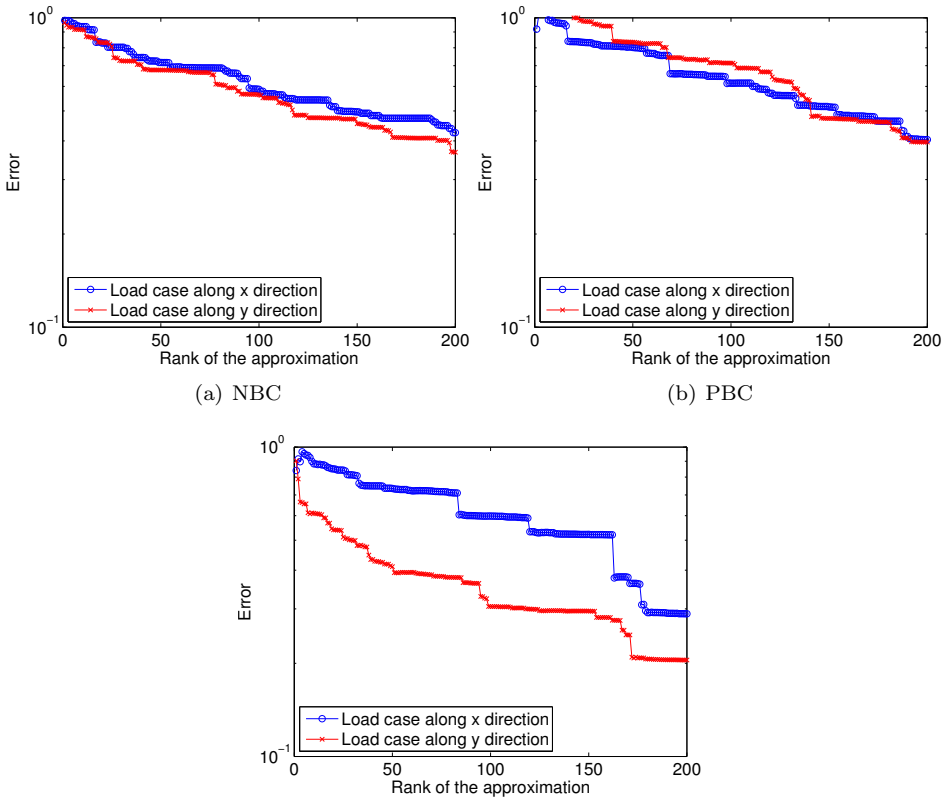
The two alternatives proposed in algorithms 1 and 3 have been tested with PBC, for the load case $\underline{\nabla} u^M = (1,0,0)^T$. Here, the residual serves as an error indicator. It is plotted figure 10. This figure shows that the algorithm 3 converges faster than 1. Indeed, while 40 iterations are needed to reach a residual of 0.0532 for algorithm 3, algorithm 1 needs 90 iterations to reach the same value. However, algorithm 3 has a much higher computational cost than algorithm 1 due to the projection on the subspaces $(U_m)_{m\in\mathbb{N}^*}$. This limitation restricts the use of algorithm 3 to low dimensional cases (2 or 3).

Even if the residual has a poor convergence rate in both cases, we can see on figure 11 that the homogenized value $\underline{\underline{K}}^h_{xx}$ converges rapidly. The fast convergence of homogenized value in contrast with residual justifies the definition of new error indicators. They will be defined in the next section via goal oriented error estimation and the construction of adaptive algorithms for the solution of problem (19).

6. Goal-oriented error estimation

We consider the variational problem:

$$\begin{aligned} &\text{Find } u \in \mathcal{V} \text{ such that} \\ &\forall \delta u \in \mathcal{V},\ a(\delta u, u) = l(\delta u) \end{aligned} \tag{50}$$



(a) NBC (b) PBC

(c) EBC

Figure 7: Convergence of the progressive rank-m PGD approximation with respect to rank m. Relative error in L^2 norm, $k_i = 1000W.m^{-1}.K^{-1}$.

We denote by $u_m \in \mathcal{V}$ an approximation of u obtained with a PGD algorithm. The aim is here to provide a methodology for an adaptive approximation of the solution relying on the classical dual-primal error estimator [1, 2]. After presenting this adaptive method in a general context of linear quantities of interest, we provide a specific methodology for the present context of computational homogenization, where different linear quantities of interest (the coefficients of the homogenized tensor) have to be computed.

6.1. Quantity of interest and adjoint problem

We define a quantity of interest $Q(u)$ with $Q : \mathcal{V} \to \mathbb{R}$ a linear functional on $\mathcal{V}$. We introduce the adjoint problem

$$\begin{aligned} &\text{Find } \phi \in \mathcal{V} \text{ such that} \\ &\forall \delta u \in \mathcal{V},\ a(\delta u, \phi) = Q(\delta u) \end{aligned} \tag{51}$$

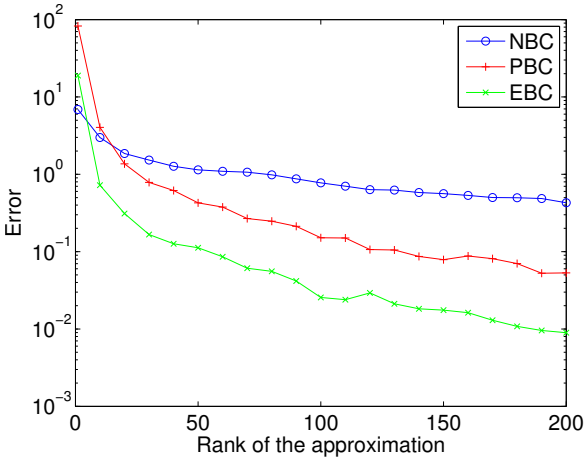


Figure 8: Relative error in canonical norm on the homogenized tensor as a function of the rank of the approximation. $k_i = 1000W.m^{-1}.K^{-1}$.

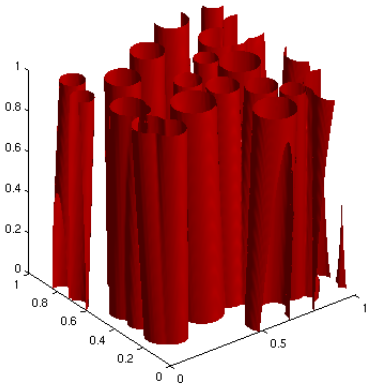


Figure 9: Random fibers into a matrix

The quantity of interest can then be expressed $Q(u) = a(u, \phi)$. The error in the quantity of interest can be expressed:

$$Q(u) - Q(u_m) = a(u - u_m, \phi) = l(\phi) - a(u_m, \phi) = \langle \phi, R(u_m) \rangle$$

where $R(u_m)$ is the residual associated with the approximation u_m.

6.2. Approximation of the adjoint problem

Suppose that we have an approximation ϕ_n of ϕ. We then have

$$Q(u) - Q(u_m) = a(u - u_m, \phi - \phi_n) + a(u - u_m, \phi_n)$$

If the primal and adjoint problems are solved with a sufficient accuracy, the first term in the right hand side can be neglected and we have the following estimation of the error

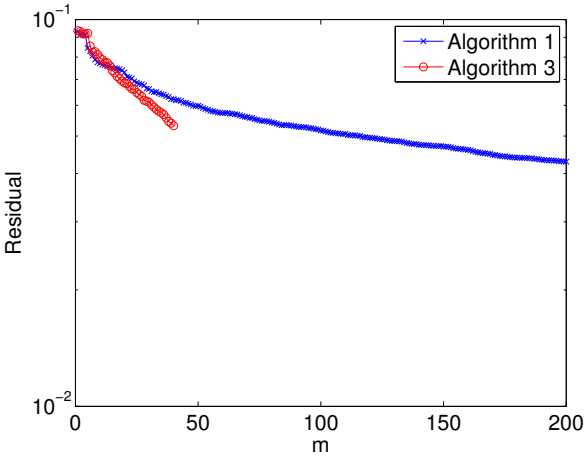


Figure 10: Residual norm with respect to m. Comparison between canonical decomposition (algorithm 1) or Tucker representation (algorithm 3).

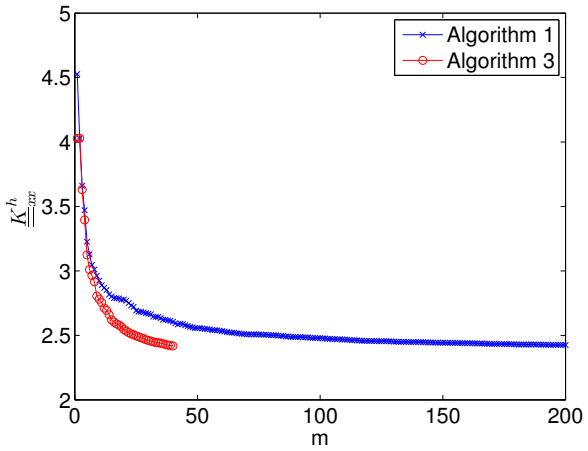


Figure 11: $\underline{\underline{K}}^h_{xx}$ with respect to m. Comparison between canonical decomposition (algorithm 1) or Tucker representation (algorithm 3).

in the quantity of interest:

$$Q(u) - Q(u_m) \approx a(u - u_m, \phi_n)$$

In practice, in order to guaranty a good estimation of the error, we can construct a sequence of approximations ϕ_n of ϕ until the estimation $a(u - u_m, \phi_n)$ has converged with a desired tolerance.

6.3. Improvement of the estimation

Note that the estimation $Q(u_m)$ converges with a convergence rate at least of the same order as u_m. Indeed,

$$|Q(u) - Q(u_m)| = |a(u - u_m, \phi)| \leq C\|u - u_m\|\|\phi\|$$

where C is the continuity constant of a. After solving the adjoint problem and with no additional cost, we can introduce an improved estimation of the quantity of interest:

$$\widehat{Q}(u_m, \phi_n) = Q(u_m) + a(u - u_m, \phi_n) \tag{52}$$

We have

$$|Q(u) - \widehat{Q}(u_m, \phi_n)| = |a(u - u_m, \phi - \phi_n)| \leq C\|u - u_m\|\|\phi - \phi_n\|$$

Note that if the adjoint problem were solved exactly, $\widehat{Q}(u_m, \phi)$ would be equal to the exact quantity of interest $Q(u)$, even if u_m is only an approximation of u. So in practice, the error analysis is also a way of improving the estimation of quantities of interest.

6.4. Adaptive approximation using error estimation

Suppose that we have a rank-m_i approximation u_{m_i} of u. In order to estimate the error on the quantity of interest, one should look at the convergence of the sequence $(E_n^{m_i})_{n\in\mathbb{N}^*}$ defined by $E_n^{m_i} = a(u - u_{m_i}, \phi_n) = \widehat{Q}(u_{m_i}, \phi_n) - Q(u_{m_i})$, where ϕ_n is a sequence of approximations of ϕ.

In practice, one has to find an estimation $\widetilde{E}_i$ of the true error E_i defined by

$$E_i = a(u - u_{m_i}, \phi) = \lim_{n\to\infty} E_n^{m_i} \tag{53}$$

This estimator $\widetilde{E}_i$ is defined by $\widetilde{E}_i = E_{N_i}^{m_i}$, with N_i such that

$$\max_{j\in\{0,\ldots,j_{max}\}} \frac{|E_{N_i+j+1}^{m_i} - E_{N_i+j}^{m_i}|}{|E_{N_i+j}^{m_i}|} < \varepsilon_{stag} \tag{54}$$

with $\varepsilon_{stag} = 10^{-2}$ a stagnation criterion. Note that this condition is chosen in order take into account possible oscillations of the sequence $\{E_n^{m_i}\}_n$. In practice, we choose $j_{max} = 9$.

If $|\widetilde{E}_i|$ is lower than a prescribed tolerance ε_{tol}, $Q(u_{m_i})$ is considered as a good estimation of $Q(u)$. Otherwise, we compute a rank-m_{i+1} approximation $u_{m_{i+1}}$ of u with $m_{i+1} = m_i + m_{step}$, we compute $\widetilde{E}_{i+1}$ and we compare it again to the prescribed tolerance. In this case, $u_{m_{i+1}}$ is computed starting from u_{m_i} in a greedy fashion. The adaptive algorithm is summed up in algorithm 4.

6.5. Expression of the quantities of interest

In the context of homogenization, the quantities of interest are the components of the homogenized tensor. They can be extracted using the following linear functionals:

$$\begin{cases} \textbf{NBC: } Q_\mu(u) = \dfrac{1}{|\Omega|}\displaystyle\int_\Omega \underline{\nabla} u \cdot \underline{e_\mu}\, d\Omega \\ \textbf{PBC, EBC: } Q_\mu(u) = \dfrac{1}{|\Omega|}\displaystyle\int_\Omega \underline{\nabla} u \cdot \underline{\underline{K}} \cdot \underline{e_\mu}\, d\Omega \end{cases}$$

Algorithm 4 Adaptive approximation algorithm using error estimation

Set $u_{m_0} = 0$
Set $i = 0$
while $|\widetilde{E}_i| > \varepsilon_{tol}$ **and** $m_i \leq m_{max}$ **do**
  Set $i = i + 1$
  Compute $u_{m_i} = u_{m_{i-1}} + \sum_{j=1}^{m_{step}} \otimes_\mu v^\mu_{m_{i-1}+j}$ using algorithms 1 or 3
  Compute $\widetilde{E}_i = a(u - u_{m_i}, \phi_{N_i}) = \langle R(u_{m_i}), \phi_{N_i} \rangle$
end while
Compute $\widehat{Q}(u_{m_i}, \phi_{N_i})$
return $\widehat{Q}(u_{m_i}, \phi_{N_i})$ and $\widetilde{E}_i$

where $\{\underline{e}_\mu\}_\mu$ is the canonical basis of $\mathbb{R}^d$, if the following values of $\underline{q}^M$ or $\underline{\nabla} u^M$ are used

$$\begin{cases} \textbf{NBC}\text{: } \underline{q}^M_\mu = \underline{e}_\mu \\ \textbf{PBC, EBC}\text{: } (\underline{\nabla} u^M)_\mu = \underline{e}_\mu \end{cases}$$

Indeed, for a given type of boundary conditions (NBC, PBC or EBC), if we denote by $\{u_\mu\}_\mu$ the solutions of the d boundary value problems associated with each load case μ, we have

$$\begin{cases} \textbf{NBC}\text{: } (\underline{\underline{K}}^h)^{-1}_{\beta\mu} = Q_\mu(u_\beta) \\ \textbf{PBC, EBC}\text{: } \underline{\underline{K}}^h_{\beta\mu} = Q_\mu(u_\beta) \end{cases}$$

We introduce the corresponding weak formulations

$$a(\delta u, u_\mu) = l_\mu(\delta u) \quad \forall \delta u \in \mathcal{V} \tag{55}$$

At the end, we have 6 problems to solve for a 3D problem, the 3 primal problems and the 3 adjoint problems.

6.6. Adaptive approximation in 2D

We consider homogenization with EBC on the microstructure represented in figure 2. The phases have homogeneous conductivities with conductivities $k_m = 1W.m^{-1}.K^{-1}$ for the matrix and $k_i = 10W.m^{-1}.K^{-1}$ for the inclusions. The image I contains 512×512 pixels and is associated with a domain $\Omega = (0,1)^2$. The image I has been smoothed with a characteristic length of $\delta = 2$ pixels. A SVD has been applied to obtain the separated representation I_s of I such that the relative error in L^2 norm is lower than 1%. As a consequence, I_s is a rank-309 approximation of I.

6.6.1. Estimation of K^h_{xx}

First, we are interested in $\underline{\underline{K}}^h_{xx} = Q_x(u_x)$. We apply the algorithm 4 to find a good approximation of this homogenized value. The parameters are $m_{step} = 20$ and $m_{max} = 400$. The tolerance is fixed to $\varepsilon_{tol} = 10^{-2}$.

In the end, the estimated error is $\widetilde{E}_3 = E^{60}_{45} = 0.87\ 10^{-2}$ and the estimated quantity of interest are $Q_x(u_{x,60}) = 1.79W.m^{-1}.K^{-1}$ and $\widehat{Q}_x(u_{x,60}, u_{\phi,45}) = 1.80W.m^{-1}.K^{-1}$. The convergence of $(|E^{m_i}_n|)_{n\in\mathbb{N}^*}$ is shown in figure 12.

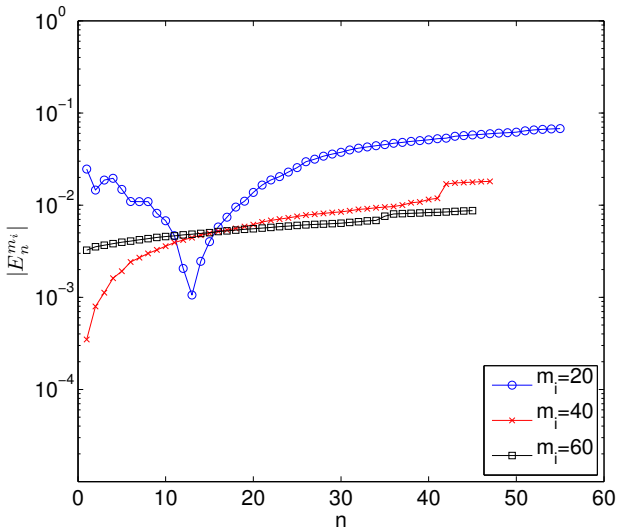


Figure 12: Error estimation as a function of the rank n of the approximation of the adjoint solution ϕ_x when computing $\underline{\underline{K}}^h_{xx} = Q_x(u_x)$ with $\varepsilon_{stag} = 10^{-2}$

Note that we are interested in the right part of the curves, when the error estimator converges. The behavior of the curve $m_i = 20$ around $n = 13$ can be explained by a change of sign in the estimated error.

6.6.2. Estimation of K^h_{xy}

We now look at $\underline{\underline{K}}^h_{xy} = Q_y(u_x)$. We apply again the algorithm 4 with $m_{step} = 20$ and $m_{max} = 400$. However, given that the homogenized material is expected to be isotropic, we take $\varepsilon_{tol} = 10^{-3}$. Finally we get $\widetilde{E}_2 = 6.56\ 10^{-4}$ which results in $Q_y(u_{x,40}) = 0.99\ 10^{-2} W.m^{-1}.K^{-1}$ and $\widehat{Q}_y(u_{x,40}, \phi_{y,45}) = 1.06\ 10^{-2} W.m^{-1}.K^{-1}$. Different error estimations for different iterations of the adaptive algorithm are plotted in figure 13. Figure 13 shows that the stagnation criterium proposed in (54) is too restrictive. This is due to the isotropy of the material and the particularly small value of $\underline{\underline{K}}^h_{xy}$. Indeed, if we relax the stagnation criterium by setting $\varepsilon_{stag} = 10^{-1}$, we can see that for $m_i = 40$ the rank N_i of the adjoint approximation goes from 226 to 40 with a good approximation of the absolute error $|E^{m_i}_{N_i}|$. The latter goes from $7.70\ 10^{-4}$ to $3.99\ 10^{-4}$ so we still have a good estimation of the absolute error. The loss of precision is counterbalanced by much faster computations of the different estimates.

6.7. Specificities in the context of homogenization

For NBC, noting that $\underline{q}^M_\mu = \underline{e}_\mu$ and using the Green formula we obtain

$$Q_\mu(u) = -\frac{1}{|\Omega|} l_\mu(u) \tag{56}$$

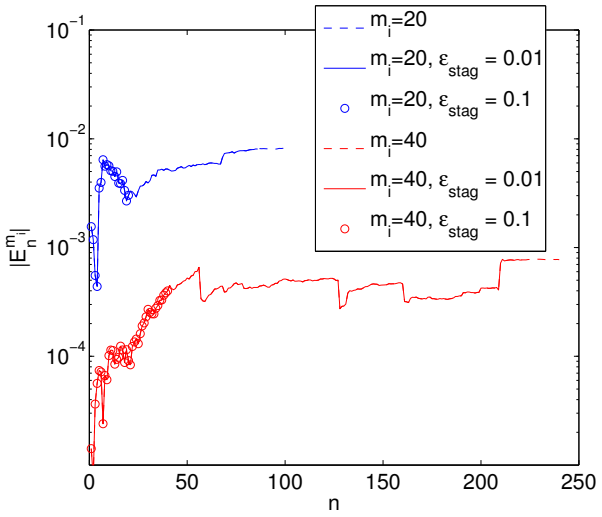


Figure 13: Error estimation as a function of the rank n of the approximation of the adjoint solution ϕ_y when computing $\underline{\underline{K}}^h_{xy} = Q_y(u_x)$ with different ε_{stag}

This equality also holds for PBC and EBC if $\underline{\nabla} u^M_\mu = \underline{e}_\mu$. Therefore, the following important property holds for every type of boundary conditions:

$$\phi_\mu = -\frac{1}{|\Omega|} u_\mu \tag{57}$$

for all μ. That means that for d-dimensional problems, we only need to solve d problems instead of $2 \times d$ problems.

With these properties, a special adaptive algorithm can be derived for homogenization. First, the diagonal terms of the homogenized tensor $\underline{\underline{K}}^h_{\mu\mu}$ are determined using the relation (57). On one hand, supposing that we already have an approximation $u_m = \sum_{i=1}^m v_i$ of u, this relation implies that $\mathcal{V}_m = \text{span}\{v_i\}_{i=1}^m$ is a good space for the approximation of ϕ through Galerkin projection. On the other hand, with $\phi_m = \sum_{i=1}^m v_i$ an approximation of ϕ, $\mathcal{V}_m = \text{span}\{v_i\}_{i=1}^m$ is also a good approximation space for the Galerkin projection of ϕ. The algorithm for the estimation of the diagonal components of the homogenized tensor is written in algorithm 5.

One can choose to solve only primal or adjoint problems for these steps. Then, given that we have already approximations $(u_{\mu,m})$, $\forall \mu$ and consequently $(\phi_{\mu,m})$, $\forall \mu$, we can have a first error estimation of non diagonal terms of the homogenized tensor $\underline{\underline{K}}^h_{\mu\beta}$, $\forall (\mu, \beta)$, $\mu \neq \beta$. If the error is too high, we can apply algorithm 4 to find an estimation of $\underline{\underline{K}}^h_{\mu\beta}$. We denote $\widetilde{E}_{\mu\beta}$ the error estimation on $\underline{\underline{K}}^h_{\mu\beta}$. The adaptive algorithm is summed up in algorithm 6.

6.8. Remark on the approximation of the adjoint problem

Note that the adjoint problem must be solved with a precision higher than the primal problem, in the following sense. Suppose that u_m is the Galerkin projection of u in a

Algorithm 5 Adaptive approximation algorithm for diagonal terms

Set $u_{m_0} = 0$
Set $i = 0$
Set $E_0 > \varepsilon_{tol}$
Compute $u_{m_1} = \sum_{j=1}^{m_1} \otimes_\mu v_j^\mu$
while $|E_i| > \varepsilon_{tol}$ **and** $m_i \leq m_{max}$ **do**
  Set $i = i + 1$
  Set $\phi_{m_i} = -\frac{1}{|\Omega|} u_{m_i}$
  Compute $\widetilde{E}_i = a(u - u_{m_i}, \phi_{N_i})$, with $N_i \geq m_i$.
  Set $u_{m_{i+1}} = -|\Omega| \times \phi_{N_i}$
end while
Compute $\widehat{Q}(u_{m_{i+1}}, \phi_{N_i})$
return $\widehat{Q}(u_{m_{i+1}}, \phi_{N_i})$ and $\widetilde{E}_i$

Algorithm 6 Adaptive approximation algorithm using error estimation for homogenization

for $\mu \in \{x, \ldots\}$ **do**
  Compute $\widetilde{E}_{\mu\mu}$ with algorithm 5
end for
for $\mu \in \{x, \ldots\}$ **do**
  for $\beta \in \{x, \ldots\} \backslash \{\mu\}$ **do**
    Compute $\widetilde{E}_{\beta\mu}$ using previously computed u_{μ,m_μ} and u_{β,m_β}
    if $|\widetilde{E}_{\beta\mu}| > \varepsilon_{tol}$ **then**
      Compute $\widetilde{E}_{\beta\mu}$ with algorithm 4, enriching u_{μ,m_μ} and u_{β,m_β}
    end if
  end for
end for

linear subspace $\mathcal{V}_m \subset \mathcal{V}$, that means

$$\forall \delta u \in \mathcal{V}_m, \quad a(\delta u, u_m) = l(\delta u)$$

If ϕ_n is defined as the Galerkin approximation of ϕ in $\mathcal{V}_m$, i.e. $\phi_n = \phi_m^*$ with

$$\forall \delta u \in \mathcal{V}_m, \quad a(\delta u, \phi_m^*) = Q(\delta u)$$

then,

$$a(u - u_m, \phi_n) = 0$$

yielding $Q(u) - Q(u_m) \approx 0$, which is of course a bad estimation of the error. For example, when constructing a rank-m approximation $u_m = \sum_{i=1}^m \lambda_i v_i$ with an update of coefficients λ_i, we have that u_m is the Galerkin approximation of u in $\mathcal{V}_m = span\{v_i\}_{i=1}^m$. We then obtain a bad estimation of the error if we compute an approximation ϕ_n in $\mathcal{V}_m$.

A possible remedy consists in defining $\phi_n = \phi_m^* + \delta\phi_{m,n}$ with $\delta\phi_{m,n}$ an approximation of $\delta\phi_m = \phi - \phi_m^*$ which is the solution of

$$a(\delta u, \delta\phi_m) = Q(\delta u) - a(\delta u, \phi_m^*) \quad \forall \delta u \in \mathcal{V}$$

Note that if $\phi_m^* = \sum_{i=1}^{m} \mu_i v_i$ and $\delta\phi_{m,n} = \sum_{i=m+1}^{n} \mu_i v_i$, we can improve the approximation ϕ_n by defining it as the Galerkin projection of ϕ in $\mathcal{V}_n = span\{v_i\}_{i=1}^{n} \supset \mathcal{V}_m$.

As mentioned above, whatever the approximation strategy for the adjoint problem, if we are able to construct a sequence of approximations ϕ_n, the convergence of the quantity $a(u - u_m, \phi_n)$ should be checked and compared to $Q(u_m)$.

7. Thermal homogenization of ductile cast iron

The last application concerns a segmented image that represents a sample of a ductile cast iron. The image has been obtained by Computed Tomography (CT) and it contains 128^3 voxels of size 2.2833 μm. The material is composed of iron and graphite. The indicator function of the graphite's phase is plotted in figure 14. The conductivities are $k_i = 76.2\ W.m^{-1}.K^{-1}$ for the iron phase and $k_g = 24.0\ W.m^{-1}.K^{-1}$ for the carbon phase. It turns out that the characteristic function I of the graphite phase is hardly

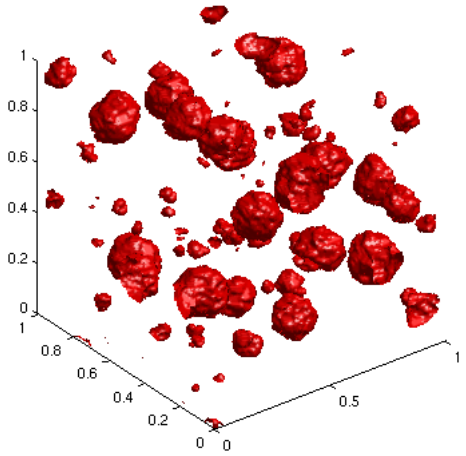


Figure 14: Ductile cast iron, characteristic function of the graphite phase

separable. As a consequence, we describe the exact geometry with the specific Tucker representation introduced in (45) with a minimal linear smoothing on one element. The algorithm 6 is employed. The homogenization with the PBC is applied. The tolerance on the components of the conductivity tensor is set to 0.1 $W.m^{-1}.K^{-1}$. We choose a stagnation criterium $\varepsilon_{stag}^{\mu\beta}$ for the computation of the error indicator defined by

$$\varepsilon_{stag}^{\mu\beta} = \begin{cases} 10^{-2} \text{ if } \mu = \beta \\ 10^{-1} \text{ otherwise} \end{cases} \tag{58}$$

as we expect that the material is isotropic. u_μ is increased by rank-50 tensor at each step of the adaptive algorithm.

From a numerical point of view, only rank-100 approximations of the solutions of the different loadings were necessary to get the diagonal components $\underline{\underline{K}}_{\mu\mu}^h, \mu \in \{x, \ldots, z\}$

with the tolerance of 0.1 $W.m^{-1}.K^{-1}$. However, rank-350 (resp. rank-300, rank-250) approximations of the adjoint solutions was needed to obtain the error estimate of $\underline{\underline{K}}^h_{xy}$ (resp. $\underline{\underline{K}}^h_{xz}$, $\underline{\underline{K}}^h_{yz}$) with the desired accuracy.

In the end, the estimated homogenized conductivity is

$$\underline{\underline{K}}^h = \begin{pmatrix} 73.35 & 1.8\ 10^{-3} & -4.5\ 10^{-3} \\ & 73.36 & -5.6\ 10^{-4} \\ sym & & 73.36 \end{pmatrix} \tag{59}$$

with estimated componentwise absolute error

$$|\widetilde{E}| = \begin{pmatrix} 7.5\ 10^{-2} & 4.7\ 10^{-4} & 3.9\ 10^{-4} \\ & 8.7\ 10^{-2} & 5.2\ 10^{-4} \\ sym & & 9.3\ 10^{-2} \end{pmatrix} \tag{60}$$

As expected, the material appears to be isotropic. Note that the obtained precision is better than the one prescribed in the adaptive algorithm. This is due to the fact that the error associated with an approximation u_m is not estimated at each iteration m but only every 50 iterations. A reference finite element computation provides the following homogenized tensor

$$\underline{\underline{K}}^h_{FEM} = \begin{pmatrix} 73.16 & 3.6\ 10^{-4} & -4.5\ 10^{-3} \\ & 73.19 & 5.7\ 10^{-4} \\ sym & & 73.20 \end{pmatrix} \tag{61}$$

The true componentwise absolute error is

$$|\widetilde{E}_{true}| = \begin{pmatrix} 1.9\ 10^{-1} & 1.4\ 10^{-3} & 7.7\ 10^{-5} \\ & 1.7\ 10^{-1} & 1.1\ 10^{-3} \\ sym & & 1.6\ 10^{-1} \end{pmatrix} \tag{62}$$

As we can see, the estimated error underestimates the true error (except for the xz component) but provides a very good estimation of this true error.

Remark 8. *Negative values for the homogenized tensors are meaningless. They are in fact almost null values so we can set them to 0.*

Remark 9. *For very large images, the methodology can be directly applied but practical issues have to be addressed in order to overcome memory limitations due to the representation of the geometry. In practice, the image should be decomposed into sub-images and tensor approximation methods applied on these different sub-images. This way, the operator in Tucker tensor format is decomposed in smaller operators associated to each sub-image.*

8. Conclusion

A complete numerical strategy based on tensor approximations has been proposed for image-based numerical homogenization. We have first introduced some specific treatments of the geometry, including a smoothing of indicator functions in order to obtain

accurate low-rank representations of the geometrical data. This yields a formulation of boundary value problems in a suitable tensor format. Suitable weak formulations of boundary value problems that preserve tensor format have been introduced in order to handle the different types of boundary conditions that are used in classical numerical homogenization methods. Then, some variants of Proper Generalized Decomposition (PGD) methods have been introduced for the a priori construction of tensor decompositions of the solution of boundary value problems. In particular, a new definition of PGD has been proposed for the progressive construction of a decomposition of the solution in Tucker format. This tensor format appears to be well adapted to the present application and the proposed algorithms yields a rapid convergence of tensor decompositions. Finally, a dual-based error estimation method has been introduced in order to assess the quality of the prediction of homogenized properties. A goal oriented adaptive strategy has been proposed for the construction of tensor approximations that satisfy a prescribed accuracy on predicted homogenized properties. The complete methodology has been successfully applied to images of real 2 or 3-dimensional microstructures.

Considering the computational time, figure 15 is of interest. A 3D problem has been solved for several resolutions using FEM and PGD approaches. The FEM solution is used as a reference for the computation of the L^2 relative error ε. The time reference is the computational time of the FEM solution on the image with the largest number of voxels. The operator used for the PGD is treated as in section 7, meaning that the whole picture has been taken into account, without truncation. We can see on figure 15 that for

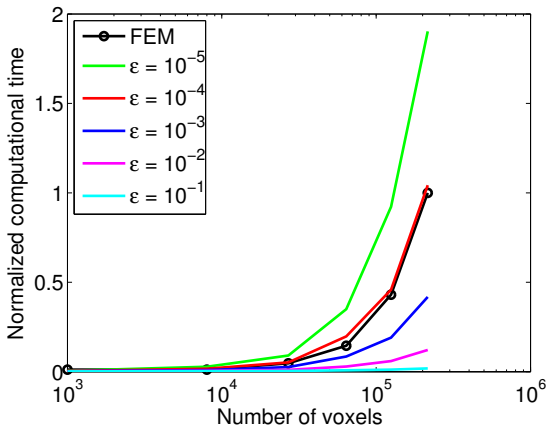


Figure 15: Normalized computational time as a function of the number of voxels and the relative error ε in L^2 norm

a precision better than 10^{-4} FEM is more interesting than PGD approaches. However looking for such a precision is meaningless since errors done in the model should be larger than 10^{-2} because of the image processing technique. An other advantage of the separate representation in term of computation is the low memory requirement. Indeed since only small 1D domains are considered, we do not have to manage very large stiffness matrices.

In order to go further in this work, additional parameters will be added to the solution such as the conductivity of each phase of the heterogeneous material. However to

solve these high dimensional PDEs other tensor formats can be employed, in particular hierarchical Tucker tensor format and its derivations [19, 46]. Indeed the canonical decomposition has several drawbacks as discussed by de Silva and Lim in [51] especially in terms of existence, quality of the approximation and the so called "cancellation" effect. The orthogonality of the vectors in the Tucker tensors avoids these problems but this decomposition is still suffering from the curse of dimensionality since the size of the core tensor is growing with r^d with $r = r^x = r^y = \ldots$. The hierarchical Tucker tensor format writes the core tensor as a hierarchy of smaller orders such that the linearity dependance with the dimension is recovered while keeping the orthogonality features of the Tucker decomposition.

References

[1] Ainsworth, M., Oden, J.T., 2000. A Posteriori Error Estimation in Finite Element Analysis. Wiley-Interscience. 1st edition.

[2] Ammar, A., Chinesta, F., Diez, P., Huerta, A., 2010. An error estimator for separated representations of highly multidimensional models. Computer Methods in Applied Mechanics and Engineering 199, 1872–1880.

[3] Babin, P., Valle, G., Dendievel, R., Lassoued, N., Salvo, L., 2005. Mechanical properties of bread crumbs from tomography based Finite Element simulations. Journal of Materials Science 40, 5867–5873.

[4] Bader, B., Kolda, T., 2011. Matlab tensor toolbox version 2.4. `http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/`.

[5] Bensoussan, A., Lions, J., Papanicolaou, G., 1978. Asymptotic Analysis for Periodic Structures. North-Holland.

[6] Camacho, D.L.A., Hopper, R.H., Lin, G.M., Myers, B.S., 1997. An improved method for finite element mesh generation of geometrically complex structures with application to the skullbase. Journal of Biomechanics 30, 1067 – 1070.

[7] Cances, E., Ehrlacher, V., Lelievre, T., 2011. Convergence of a greedy algorithm for high-dimensional convex nonlinear problems. Mathematical Models & Methods In Applied Sciences 21, 2433–2467.

[8] Carroll, J.D., Chang, J., 1970. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. Psychometrika 35, 283–319.

[9] Charras, G.T., Guldberg, R.E., 2000. Improving the local solution accuracy of large-scale digital image-based finite element analyses. Journal of Biomechanics 33, 255 – 259.

[10] Chinesta, F., Ammar, A., Cueto, E., 2010. Recent advances in the use of the Proper Generalized Decomposition for solving multidimensional models. Archives of Computational Methods in Engineering 17, 327–350.

[11] Chinesta, F., Ammar, A., Lemarchand, F., Beauchene, P., Boust, F., 2008. Alleviating mesh constraints: Model reduction, parallel time integration and high resolution homogenization. Computer Methods in Applied Mechanics and Engineering 197, 400–413.

[12] Chinesta, F., Ladeveze, P., Cueto, E., 2011. A short review on model order reduction based on proper generalized decomposition. Archives of Computational Methods in Engineering 18, 395–404.

[13] Duster, A., Parvizian, J., Yang, Z., Rank, E., 2008. The finite cell method for three-dimensional problems of solid mechanics. Computer Methods in Applied Mechanics and Engineering 197, 3768–3782.

[14] Falco, A., Hackbusch, W., submitted. On minimal subspaces in tensor representations. Foundations of Computational Mathematics .

[15] Falco, A., Nouy, A., 2011. A proper generalized decomposition for the solution of elliptic problems in abstract form by using a functional Eckart-Young approach. Journal of Mathematical Analysis and Applications 376, 469–480.

[16] Falcó, A., Nouy, A., 2012. Proper generalized decomposition for nonlinear convex problems in tensor Banach spaces. Numerische Mathematik doi:10.1007/s00211-011-0437-5.

[17] Golanski, D., Terada, K., Kikuchi, N., 1997. Macro and micro scale modeling of thermal residual stresses in metal matrix composite surface layers by the homogenization method. Computational Mechanics 19, 188–202.

[18] Hackbusch, W., 2012. Tensor Spaces and Numerical Tensor Calculus. Springer. 2012 edition.
[19] Hackbusch, W., Kühn, S., 2009. A new scheme for the tensor representation. Journal of Fourier Analysis and Applications 15, 706–722.
[20] Harshman, R.A., 1970. Foundations of the PARAFAC procedure: Model and conditions for an"explanatory"multi-mode factor analysis. UCLA Working Papers in Phonetics 16, 1–84.
[21] Hollister, S., Kikuchi, N., 1994. Homogenization theory and digital imaging: a basis for studying the mechanics and design principles of bone tissue. Biotechnology and Bioengineering 43, 586–596.
[22] Jiang, M., Jasiuk, I., Ostoja-Starzewski, M., 2002. Apparent thermal conductivity of periodic two-dimensional composites. Computational Materials Science 25, 329–338.
[23] Kanit, T., Forest, S., Galliet, I., Mounoury, V., Jeulin, D., 2003. Determination of the size of the representative volume element for random composites: statistical and numerical approach. International Journal of Solids and Structures 40, 3647–3679.
[24] Kanit, T., N'Guyen, F., Forest, S., Jeulin, D., Reed, M., Singleton, S., 2006. Apparent and effective physical properties of heterogeneous materials: Representativity of samples of two materials from food industry. Computer Methods in Applied Mechanics and Engineering 195, 3960–3982.
[25] Keyak, J., Meagher, J., Skinner, H., Mote, C., Jr, 1990. Automated three-dimensional finite element modelling of bone: a new method. Journal of Biomedical Engineering 12, 389 – 397.
[26] Khoromskij, B., 2009. Tensor-Structured preconditioners and approximate inverse of elliptic operators in r^d. Constructive Approximation 30, 599–620.
[27] Kolda, T.G., Bader, B.W., 2009. Tensor decompositions and applications. SIAM Review 51, 455.
[28] Ladevèze, P., Passieux, J., Néron, D., 2010. The LATIN multiscale computational method and the Proper Generalized Decomposition. Computer Methods in Applied Mechanics and Engineering 199, 1287–1296.
[29] Ladevèze, P., Chamoin, L., 2011. On the verification of model reduction methods based on the proper generalized decomposition. Computer Methods in Applied Mechanics and Engineering 200, 2032–2047.
[30] Lamari, H., Ammar, A., Cartraud, P., Chinesta, F., Jacquemin, F., Legrain, G., 2010a. Recent advances in material homogenization. International Journal of Material Forming 3, 899–902.
[31] Lamari, H., Ammar, A., Cartraud, P., Legrain, G., Chinesta, F., Jacquemin, F., 2010b. Routes for efficient computational homogenization of nonlinear materials using the proper generalized decompositions. Archives of Computational Methods in Engineering 17, 373–391.
[32] Langville, A.N., Stewart, W.J., 2004. A kronecker product approximate preconditioner for SANs. Numerical Linear Algebra with Applications 11, 723–752.
[33] Laschet, G., 2002. Homogenization of the thermal properties of transpiration cooled multi-layer plates. Computer Methods in Applied Mechanics and Engineering 191, 4535–4554.
[34] Legrain, G., Allais, R., Cartraud, P., 2011a. On the use of the extended finite element method with quatree/octree meshes. International Journal for Numerical Methods in Engineering 86, 717–743.
[35] Legrain, G., Cartraud, P., Perreard, I., Moës, N., 2011b. An x-fem and level set computational approach for image-based modelling: Application to homogenization. International Journal for Numerical Methods in Engineering 86, 915–934.
[36] Legrain, G., Chevaugeon, N., Dréau, K., 2012. Computational homogenization using high order x-fem and levelsets: Uncoupling geometry and approximation. Computer Methods in Applied Mechanics and Engineering n/a, Submitted.
[37] Lewis, A., Geltmacher, A., 2006. Image-based modeling of the response of experimental 3d microstructures to mechanical loading. Scripta Materialia 55, 81 – 85.
[38] Lian, W., 2011. Contribution à l'Homogénéisation Numérique du Comportement Elastique de Matériaux à Microstructure Complexe Caractérisés par Imagerie. Ph.D. thesis. Ecole Centrale Nantes. Nantes.
[39] Lian, W.D., Legrain, G., Cartraud, P., 2012. Image-based computational homogenization and localization: comparison between X-FEM/levelset and voxel-based approaches. Computational Mechanics , Submitted.
[40] Madi, K., Forest, S., Boussuge, M., Gailliégue, S., Lataste, E., Buffiére, J.Y., Bernard, D., Jeulin, D., 2007. Finite element simulations of the deformation of fused-cast refractories based on x-ray computed tomography. Computational Materials Science 39, 224 – 229.
[41] Maire, E., Fazekas, A., Salvo, L., Dendievel, R., Youssef, S., Cloetens, P., Letang, J.M., 2003. X-ray tomography applied to the characterization of cellular materials. related finite element modeling problems. Composites Science and Technology 63, 2431 – 2443.
[42] Mishnaevsky Jr., L.L., 2005. Automatic voxel-based generation of 3d microstructural fe models and its application to the damage analysis of composites. Materials Science and Engineering: A 407, 11

– 23.
[43] Moulinec, H., Suquet, P., 1998. A numerical method for computing the overall response of nonlinear composites with complex microstructure. Computer Methods in Applied Mechanics and Engineering 157, 69 – 94.
[44] Nouy, A., 2010a. A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. Computer Methods in Applied Mechanics and Engineering 199, 1603–1626.
[45] Nouy, A., 2010b. Proper Generalized Decompositions and separated representations for the numerical solution of high dimensional stochastic problems. Archives of Computational Methods in Engineering 17, 403–434.
[46] Oseledets, I.V., 2011. Tensor-Train decomposition. SIAM Journal on Scientific Computing 33, 2295.
[47] Ostoja-Starzewski, M., 2006. Material spatial randomness: From statistical to representative volume element. Probabilistic Engineering Mechanics 21, 112–132.
[48] Ozdemir, I., Brekelmans, W., Geers, M., 2008. Computational homogenization for heat conduction in heterogeneous solids. International Journal for Numerical Methods in Engineering 73, 185–204.
[49] Sanchez-Palencia, E., 1980. Non homogeneous media and vibration theory. volume 127 of *Lecture Notes in Physics*. Springer Verlag, Berlin.
[50] Sethian, J., 1999. Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences. Cambridge Monographs on Applied and Computational Mathematics (No. 3), Cambridge University Press. 2nd edition.
[51] de Silva, V., Lim, L., 2008. Tensor rank and the Ill-Posedness of the best Low-Rank approximation problem. SIAM Journal on Matrix Analysis and Applications 30, 1084.
[52] Takano, N., Zako, M., Kubo, F., Kimura, K., 2003. Microstructure-based stress analysis and evaluation for porous ceramics by homogenization method with digital image-based modeling. International Journal of Solids and Structures 40, 1225 – 1242.
[53] Terada, K., Asai, M., Yamagishi, M., 2003. Finite cover method for linear and non-linear analyses of heterogeneous solids. International journal for numerical methods in engineering 58, 1321–1346.
[54] Touzene, A., 2008. A tensor sum preconditioner for stochastic automata networks. INFORMS Journal on Computing 20, 234–242.
[55] Ulrich, D., van Rietbergen, B., Weinans, H., Rüegsegger, P., 1998. Finite element analysis of trabecular bone structure: a comparison of image-based meshing techniques. Journal of Biomechanics 31, 1187 – 1192.
[56] Young, P., Beresford-West, T., Coward, S., Notarberardino, B., Walker, B., Abdul-Aziz, A., 2008. An efficient approach to converting three-dimensional image data into highly accurate computational models. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 366, 3155–3173.

V.2 Extension à la mécanique

On va étendre brièvement la méthode d'homogénéisation thermique présentée en section V.1 à l'élasticité linéaire.

V.2.1 Homogénéisation en mécanique

On note le champ de déplacement u, ε celui de déformation et σ le champ de contraintes. On définit la contrainte macroscopique σ^M et la déformation macroscopique ε^M sur un domaine Ω comme la moyenne de ces champs sur le domaine, c'est-à-dire par

$$\varepsilon^M =< \varepsilon >= \frac{1}{|\Omega|}\int_\Omega \varepsilon \; d\Omega, \tag{V.1}$$

$$\sigma^M =< \sigma >= \frac{1}{|\Omega|}\int_\Omega \sigma \; d\Omega. \tag{V.2}$$

Si à l'inverse on cherche les champs microscopiques à partir des valeurs macroscopiques, on doit résoudre les problèmes de localisations

$$\begin{aligned} &\nabla \cdot \sigma = 0 \quad \text{sur} \quad \Omega, \\ &\sigma = C : \varepsilon \quad \text{sur} \quad \Omega, \\ &+ \text{ des conditions aux limites dépendantes de } \sigma^M \text{ ou } \varepsilon^M, \end{aligned} \tag{V.3}$$

où C est le tenseur d'élasticité. On obtient alors les champs microscopiques via les tenseurs de localisation A et B par

$$\varepsilon = A\varepsilon^M, \tag{V.4}$$

$$\text{ou} \quad \sigma = B\sigma^M. \tag{V.5}$$

Une fois que toutes les valeurs macroscopiques sont à disposition, on définit le tenseur d'élasticité effectif C_h par

$$\sigma^M = C_h\varepsilon^M. \tag{V.6}$$

V.2.2 Les conditions aux limites

Le tenseur C_h va dépendre des conditions aux limites utilisées dans le problème V.3. On considère ici celles utilisées classiquement dans la littérature [42, 43].

V.2.2.1 Conditions aux limites en contraintes uniformes (Stress Uniform Boundary Conditions, SUBC).

Les conditions aux limites SUBC sont définies par

$$\sigma \; n = \sigma^M \; n \quad \text{sur} \quad \partial\Omega. \tag{V.7}$$

u est alors défini à un déplacement de solide rigide près. Il convient de bloquer les translations et les rotations en cherchant u dans l'espace

$$\mathcal{V}_m = \left\{ v \in \left(H^1(\Omega)\right)^d ;\ \int_\Omega v\ d\Omega = 0;\ \int_\Omega v \wedge x\ dx = 0 \right\}, \tag{V.8}$$

Le problème à résoudre est alors

$$\begin{aligned} &\text{Trouver } u \in \mathcal{V}_m \text{ tel que} \\ &\int_\Omega \varepsilon(\delta u) : C : \varepsilon(u)\ d\Omega = \int_\Omega \delta u \cdot \sigma^M n\ d\Gamma, \quad \forall \delta u \in \mathcal{V}_m. \end{aligned} \tag{V.9}$$

On sait aussi que

$$\varepsilon^M = <\varepsilon> = <C^{-1}\sigma> = <C^{-1}B>\sigma^M = (C_h^{SUBC})^{-1}\sigma^M. \tag{V.10}$$

Ainsi d^2 valeurs de σ^M (typiquement $(e_i \otimes e_j)_{1\leq i,j\leq d}$ où $(e_i)_{1\leq i\leq d}$ est une base de $\mathbb{R}^d$) donnent le tenseur homogénéisé C_h^{SUBC}. Pour des raisons de symétries, $\frac{d(d+1)}{2}$ valeurs de σ^M sont suffisantes (e.g. $(e_i \otimes e_j)_{1\leq i\leq j\leq d}$).

V.2.2.2 Conditions aux limites périodiques (PBC).

Cette fois on considère les conditions aux limites suivantes :

$$\begin{aligned} &u - \varepsilon^M x \text{ est } \Omega\text{-périodique,} \\ &\sigma n \text{ est } \Omega\text{-antipériodique.} \end{aligned} \tag{V.11}$$

On pose $\widetilde{u} = u - \varepsilon^M x$. Cette variable est définie à un mouvement de translation près. On va donc définir l'ensemble

$$\mathcal{V}_{m,per} = \left\{ v \in \left(H^1_{per}(\Omega)\right)^d ;\ \int_\Omega v\ d\Omega = 0 \right\}. \tag{V.12}$$

Ainsi le problème

$$\begin{aligned} &\text{Trouver } \widetilde{u} \in \mathcal{V}_{m,per} \text{ tel que} \\ &\int_\Omega \varepsilon(\delta u) : C : \varepsilon(\widetilde{u})\ d\Omega = -\int_\Omega \varepsilon(\delta u) : C : \varepsilon^M\ d\Omega, \quad \forall \delta u \in \mathcal{V}_{m,per} \end{aligned} \tag{V.13}$$

est bien posé. Par linéarité du problème, il existe un opérateur χ tel que $\varepsilon(\widetilde{u}) = \chi\varepsilon^M$. Il en vient $\varepsilon(u) = (I+\chi)\varepsilon^M$, $A = I + \chi$ et

$$\sigma^M = <\sigma> = <C\varepsilon(u)> = <C(I+\chi)>\varepsilon^M = C_h^{PBC}\varepsilon^M. \tag{V.14}$$

Encore une fois, $\frac{d(d+1)}{2}$ valeurs de ε^M donnent le tenseur homogénéisé C_h^{PBC}.

V.2.2.3 Conditions aux limites essentielles (Kinematic Uniform Boundary Conditions, KUBC)

Les conditions aux limites sont définies par

$$u = \varepsilon^M x \quad \text{sur} \quad \partial\Omega. \tag{V.15}$$

On procède à un relèvement et on va chercher u sous la forme

$$u = \varepsilon^M x + \widetilde{u}, \tag{V.16}$$

avec

$$\widetilde{u} = 0 \quad \text{sur} \quad \partial\Omega. \tag{V.17}$$

Le problème devient alors

$$\begin{aligned} &\text{Trouver } \widetilde{u} \in \left(H_0^1(\Omega)\right)^d \text{ tel que} \\ &\int_\Omega \varepsilon(\delta u) : C : \varepsilon(\widetilde{u})\ d\Omega = -\int_\Omega \varepsilon(\delta u) : C : \varepsilon^M\ d\Omega, \quad \forall \delta u \in \left(H_0^1(\Omega)\right)^d. \end{aligned} \tag{V.18}$$

Enfin on a la relation

$$\sigma^M = <\sigma> = <C\varepsilon(u)> = <CA\varepsilon^M> = <CA>\varepsilon^M = C_h^{KUBC}\varepsilon^M, \tag{V.19}$$

et $\frac{d(d+1)}{2}$ valeurs de ε^M donnent le tenseur homogénéisé C_h^{KUBC}.

V.2.2.4 Formulation variationnelle générique

On pose

$$a(\delta u, u) = \int_\Omega \varepsilon(\delta u) : C : \varepsilon(u)\ d\Omega. \tag{V.20}$$

Les déplacements de solides rigides sont bloqués en modifiant la forme bilinéaire par

$$\widetilde{a}(\delta u, u) = a(\delta u, u) + \alpha \int_\Omega \delta u\ d\Omega \cdot \int_\Omega u\ d\Omega + \beta \left(\int_\Omega \delta u \wedge x\ d\Omega\right) \cdot \left(\int_\Omega u \wedge x\ d\Omega\right), \tag{V.21}$$

où les coefficients $\alpha > 0$ et $\beta > 0$ sont choisis de manière à ne pas modifier le conditionnement du problème.

V.2.3 Expression spécifique du champ de déplacement

Dans cette section on illustre la méthode dans un cas 2D, la généralisation à la 3D étant directe. Une méthode va être proposée pour exprimer le champ de déplacement de façon à ce que les expressions des champs de déformations et de contraintes soient relativement simples à écrire.

On pourrait approximer le déplacement comme Bognet et al. [12] par exemple, qui ont utilisé

$$u_r(x,y) = \sum_{i=1}^{r} \begin{pmatrix} v_i^x(x) w_i^x(y) \\ v_i^y(x) w_i^y(y) \end{pmatrix} \Leftrightarrow u_r = \sum_{i=1}^{r} \begin{pmatrix} v_i^x \otimes w_i^x \\ v_i^y \otimes w_i^y \end{pmatrix}. \tag{V.22}$$

On pourrait aussi imaginer découpler les rangs des composantes et ainsi avoir une approximation de la forme

$$u_{r_x r_y} = \begin{pmatrix} \sum_{i=1}^{r_x} v_i^x \otimes w_i^x \\ \sum_{j=1}^{r_y} v_j^y \otimes w_j^y \end{pmatrix}. \tag{V.23}$$

L'utilisation de ces approximations conduisent à des expressions complexes des champs de contraintes et de déformations dans un format de tenseurs. Pour simplifier ceci, l'approximation proposée ici est de la forme

$$u_r(x,y) = \sum_{i=1}^{r} v_i^x(x) v_i^y(y) \begin{pmatrix} w_i^1 \\ w_i^2 \end{pmatrix} \Leftrightarrow u_r = \sum_{i=1}^{r} v_i^x \otimes v_i^y \otimes w_i, \tag{V.24}$$

où les v_i^α sont des champs scalaires et les w_i sont des éléments de $\mathbb{R}^2$. Pour exprimer simplement les tenseurs des déformations et des contraintes dans la notation de Voigt, on introduit les opérateurs L_x et L_y tels que

$$L_x = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{et} \quad L_y = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{V.25}$$

Le tenseur des déformations s'écrit alors avec les notations de Voigt

$$\varepsilon(u_r) = \begin{pmatrix} \frac{\partial u_{r,x}}{\partial x} \\ \frac{\partial u_{r,y}}{\partial y} \\ \frac{\partial u_{r,x}}{\partial y} + \frac{\partial u_{r,y}}{\partial x} \end{pmatrix} \tag{V.26}$$

$$= \sum_{i=1}^{r} (v_i^x)' \otimes v_i^y \otimes \begin{pmatrix} w^1 \\ 0 \\ w^2 \end{pmatrix} + v_i^x \otimes (v_i^y)' \otimes \begin{pmatrix} 0 \\ w^2 \\ w^1 \end{pmatrix} \tag{V.27}$$

$$= \sum_{i=1}^{r} (v_i^x)' \otimes v_i^y \otimes (L_x w_i) + v_i^x \otimes (v_i^y)' \otimes (L_y w_i). \tag{V.28}$$

On sait que le champ de tenseur des contraintes en élasticité linéaire est donné par $\sigma = C\varepsilon$. On va exprimer C pour un matériau hétérogène à 2 phases numérotées de tenseurs d'élasticité respectifs C_1 et C_2. On note I_1 l'indicatrice de la phase 1. Le tenseur d'élasticité sur le domaine est alors défini par

$$C = I_1 C_1 + (1 - I_1) C_2. \tag{V.29}$$

Après une SVD de I_1, on peut écrire

$$C = \sum_{i=1}^{r_C} I_i^x \otimes I_i^y \otimes D_i, \tag{V.30}$$

où $D_i = C_1$ ou C_2. Le champ de tenseur des contraintes est finalement donné par

$$\sigma = C\varepsilon = \sum_{i=1}^{r_C}\sum_{j=1}^{r}(I_i^x(v_j^x)') \otimes (I_i^y v_j^y) \otimes (D_i L_x w_j) + (I_i^x v_j^x) \otimes (I_i^y (v_j^y)') \otimes (D_i L_y w_j). \tag{V.31}$$

Remarque V.2.1. *On peut avoir une écriture encore plus réduite en écrivant*

$$C = C_2 + I_1(C_1 - C_2). \tag{V.32}$$

V.2.4 Exemple

La méthode précédente est appliquée à l'image de 50×50 pixels représentée sur la figure V.1. L'inclusion a un module d'Young de 10 alors que la matrice à un module d'Young de 1. Les coefficients de Poisson sont imposés à 0.3. La méthode d'homogénéisation va être appliquée avec des conditions aux limites $KUBC$. L'indicatrice non lissée est approximée par en un tenseur de rang 8 avec une SVD pour une erreur relative en norme de Frobénius de $1.12\ 10^{-16}$. Étant donné la dimensionnalité

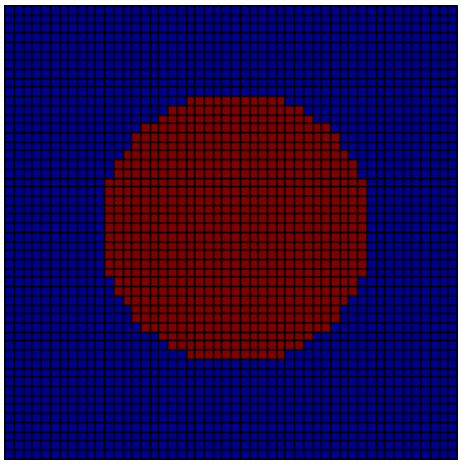


FIGURE V.1 – Image de départ

du problème, on va comparer l'algorithme glouton dans $\mathcal{C}_1(\mathcal{V})$, c'est-à-dire l'algorithme $\mathfrak{C}_1$ comme noté en section IV.3.1, et la méthode de projection de la section IV.2.3, notée $\mathfrak{T}$. On note X (resp. Y, XY), le problème ayant pour conditions aux limites $\varepsilon^M = e_1 \otimes e_1$ (resp. $e_2 \otimes e_2$, $e_1 \otimes e_2$). Les courbes de convergence sont données en figure V.2.

Encore une fois la méthode de projection converge plus rapidement que la construction gloutonne.

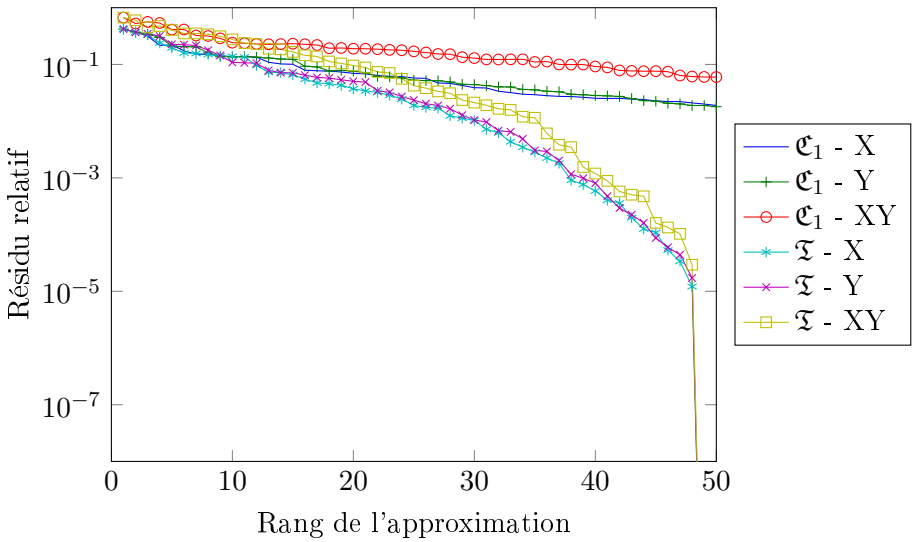


FIGURE V.2 – Résidu relatif en fonction du rang de l'approximation

Conclusion et perspectives

Lors de l'analyse d'un problème physique, l'augmentation du nombre de paramètres d'étude entraîne la malédiction de la dimensionnalité rendant difficile la résolution de problèmes aux limites de grandes dimensions. La formulation de tels problèmes est présentée au début de ce mémoire, où elle est illustrée en détail sur un problème de diffusion de la chaleur avec des paramètres stochastiques.

Afin de réduire la complexité de la résolution des problèmes en grande dimension, on s'est intéressé ici à l'utilisation des méthodes d'approximation de tenseurs pour approcher la solution. Celles-ci ont déjà fait leurs preuves par des constructions directes ou gloutonnes dans le format canonique, notamment avec les algorithmes PGD. Malheureusement ce type d'approximation souffre de problèmes mathématiques qui empêchent le développement d'algorithmes efficaces pour l'approximation de tenseurs en grande dimension. Pour contourner ces problèmes théoriques et pratiques, il est nécessaire d'introduire d'autres formats, comme les tenseurs de Tucker et les tenseurs hiérarchiques de Tucker. Ces formats alternatifs ainsi que les méthodes d'approximation afférentes ont été présentés dans le second chapitre de ce mémoire. Un effort d'abstraction a été effectué pour que la présentation des méthodes soit indépendante de la nature des espaces et des métriques utilisées, le but étant de pouvoir les utiliser pour approcher un tenseur quelconque, un opérateur, son inverse mais aussi la solution d'un système linéaire.

Ces méthodes ont été utilisées ensuite pour préconditionner des solveurs itératifs. Ces solveurs font appel aux algorithmes classiques, tel que le gradient conjugué, couplés aux méthodes d'approximation de tenseurs, notamment les généralisations de la décomposition en valeurs singulières. Malheureusement, les propriétés théoriques des solveurs originaux sont perdues et la convergence n'est plus forcément garantie. De plus ils nécessitent généralement un bon préconditionneur. C'est pourquoi un nouveau préconditionneur a été proposé ici. Son calcul repose sur une construction gloutonne d'une approximation de l'inverse de l'opérateur à partir d'une norme appropriée. Il a été appliqué avec succès à différents problèmes symétriques et non symétriques. Il permet aussi une bonne estimation de l'erreur relative pourvu que l'approximation de l'inverse soit suffisamment précise.

Le préconditionneur proposé est coûteux à calculer car il est basé sur une inversion de matrices. Une piste d'amélioration consisterait à développer une méthode d'approximation creuse de l'inverse adapté aux tenseurs, tel que l'algorithme SPAI. On pourrait aussi améliorer l'efficacité du préconditionneur en approchant l'inverse dans un format autre que le canonique. Dans tous les cas, il serait nécessaire de se reposer sur une estimation robuste du conditionnement de l'opérateur, ce qui est un enjeu pour savoir quand arrêter l'algorithme de construction.

Une autre approche a été présentée pour résoudre directement un système linéaire à partir des

méthodes d'approximation de tenseurs. Ceci passe par le choix de normes appropriées permettant la construction a priori d'une approximation de la solution (PGD). On a proposé ici une manière de préconditionner ce type de méthode par l'utilisation de normes adaptées. D'après les tests effectués, le préconditionnement de ces méthodes améliore l'erreur par rapport à la solution mais ne change pas la vitesse de convergence des algorithmes.

On a aussi proposé d'autres algorithmes pour trouver des approximations de la solution dans des formats non canoniques. On passe notamment par la définition d'espaces produit tensoriel réduits construits de manière gloutonne. Deux manières d'approximer la solution consistent alors à : (1) calculer la projection sur ces espaces réduits en utilisant les tenseurs de Tucker, (2) approximer la projection avec les tenseurs hiérarchiques de Tucker. Ceci passe par un algorithme de minimisations alternées sur les tenseurs de transfert. La première méthode souffre encore de la malédiction de la dimensionnalité, alors que la seconde a été appliquée avec succès sur des problèmes allant jusqu'à 8 dimensions, sur des exemples symétriques ou non. Le premier avantage de ces algorithmes par rapport aux solveurs itératifs classiques est que ces méthodes n'ont pas nécessairement besoin d'être préconditionnées dans les cas simples. De plus leur convergence est rigoureusement garantie et elles sont naturellement adaptatives, on n'a pas besoin de fixer le rang de l'approximation à l'avance.

Les problèmes restant à traiter concernent tout d'abord le choix de l'arbre pour les tenseurs hiérarchiques de Tucker. En effet, l'arbre peut avoir une incidence sur le rang nécessaire pour avoir une approximation d'un tenseur avec une précision donnée, le but étant d'avoir un rang minimal. De plus un rang élevé peut difficilement être atteint pour des raisons de complexité des algorithmes. Néanmoins cette complexité pourrait être réduite en se reposant sur les propriétés de variété différentielle de l'ensemble des tenseurs (hiérarchiques) de Tucker.

Ces méthodes ont finalement été appliquées à des problèmes d'homogénéisation de matériaux hétérogènes basées sur des images. On a proposé une méthode d'approximation adaptative basée sur les propriétés homgénéisées pour les déterminer avec une précision donnée. La méthode présentée pour le calcul des propriétés thermiques a été étendue au cadre de la mécanique où une certaine formulation du champ de déplacement permet d'arriver à une représentation réduite des champs de tenseurs de déformation et de contrainte.

A court terme, il serait nécessaire de revoir l'implémentation des tenseurs hiérarchiques de Tucker pour rendre les algorithmes plus efficaces. En effet, le code actuel est utile uniquement pour du prototypage. Une bonne implémentation permettra de mieux mesurer l'efficacité des méthodes présentées. Une perspective à long terme serait d'étendre les méthodes d'approximation de tenseurs au traitement des non-linéarités. Les méthodes de préconditionnement automatique proposées dans cette thèse pourraient être appliquées à une succession de problèmes linéarisés.

Propriété d'espace produit tensoriel de l'espace de Sobolev anisotrope cassé

Les notations utilisées sont celles de la section III.5.4. L'espace $H(\Lambda)$ défini tel que

$$H(\Lambda) = \left\{ v \in L^2(\Lambda);\ \frac{\partial v}{\partial x^\lambda} \in L^2(\Lambda), \lambda \in \{1,2\} \right\} \tag{A.1}$$

est muni de la norme

$$\|v\|_{H(\Lambda)} = \left(\|v\|^2_{L^2(\Lambda)} + \sum_{\lambda=1}^{2} \left\| \frac{\partial v}{\partial x^\lambda} \right\|^2_{L^2(\Lambda)} \right)^{1/2}. \tag{A.2}$$

D'après Hackbusch [38], on sait que

$$H(\Lambda) = H^1(\Omega^1) \otimes H^1(\Omega^2) \otimes L^2(\Omega^3) \otimes \ldots \otimes L^2(\Omega^6). \tag{A.3}$$

Comme en section III.5.4 on introduit la partition $\left(\Omega^2_i\right)_{1\leq i\leq K}$ de Ω^2. On pose $\Lambda_i = \Omega^1 \times \Omega^2_i \times \Omega^3 \times \ldots \times \Omega^6$. On introduit l'espace

$$\widetilde{H}(\Lambda) = \left\{ v \in L^2(\Lambda);\ v_{|_{\Lambda_i}} \in H(\Lambda_i) \right\} \supset H(\Lambda). \tag{A.4}$$

On munit cet espace de la norme $\|\cdot\|_{\widetilde{H}(\Lambda)}$ telle que

$$\|v\|_{\widetilde{H}(\Lambda)} = \sum_{i=1}^{K} \left\| v_{|_{\Lambda_i}} \right\|_{H(\Lambda_i)}. \tag{A.5}$$

Soit l'espace de Sobolev cassé $\widetilde{H}^1(\Omega^2)$ défini tel que

$$\widetilde{H}^1(\Omega^2) = \{ v \in L^2(\Omega^2);\ v_{|_{\Omega^2_i}} \in H^1(\Omega^2_i) \}. \tag{A.6}$$

On a bien sûr l'inclusion

$$H^1(\Omega^1) \otimes \widetilde{H}^1(\Omega^2) \otimes L^2(\Omega^3) \otimes \ldots \otimes L^2(\Omega^6) \subset \widetilde{H}(\Lambda). \tag{A.7}$$

Il y a en fait l'égalité entre ces deux espaces. En effet d'après Hackbusch [38], $H^1(\Omega^1)_a \otimes H^1(\Omega^2_i)_a \otimes L^2(\Omega^3)_a \otimes \ldots_a \otimes L^2(\Omega^6)$ est dense dans $H(\Lambda_i)$. Il en découle immédiatement que $H^1(\Omega^1)_a \otimes \widetilde{H}^1(\Omega^2)_a \otimes$

$L^2(\Omega^3)_a \otimes \ldots_a \otimes L^2(\Omega^6)$ est dense dans $\widetilde{H}(\Lambda)$. Ainsi on a la relation

$$\widetilde{H}(\Lambda) = H^1(\Omega^1) \otimes \widetilde{H}^1(\Omega^2) \otimes L^2(\Omega^3) \otimes \ldots \otimes L^2(\Omega^6). \tag{A.8}$$

Bibliographie

[1] B. Almroth, P. Stern et F. Brogan. "Automatic choice of global shape functions in structural analysis". *AIAA* 16 (1978), p. 525.

[2] A. Ammar et al. "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* 139.3 (2006), p. 153–176.

[3] A. Ammar et al. "A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids - Part II : Transient simulation using space-time separated representations". *Journal of Non-Newtonian Fluid Mechanics* 144.2-3 (2007), p. 98–121.

[4] I. Babuška, F. Nobile et R. Tempone. "A Stochastic Collocation Method for Elliptic Partial Differential Equations with Random Input Data". *SIAM Journal on Numerical Analysis* 45.3 (2007), p. 1005–1034.

[5] I. Babuška, R. Tempone et G. E. Zouraris. "Solving elliptic boundary value problems with uncertain coefficients by the finite element method : the stochastic formulation". *Computer Methods in Applied Mechanics and Engineering* 194.12–16 (2005), p. 1251–1294.

[6] F. Bach. "Optimization with Sparsity-Inducing Penalties". *Foundations and Trends® in Machine Learning* 4.1 (2011), p. 1–106.

[7] J. Ballani et L. Grasedyck. "A projection method to solve linear systems in tensor format". *Numerical Linear Algebra with Applications* (2012), n/a–n/a.

[8] R. Barrett et al. *Templates for the Solution of Linear Systems : Building Blocks for Iterative Methods*. 2e éd. Society for Industrial et Applied Mathematics, 1994.

[9] R. H. Bartels et G. W. Stewart. "Algorithm 432 : Solution of the matrix equation AX + XB = C". *Commun. ACM* 15.9 (1972), 820–826.

[10] M. Berveiller, B. Sudret et M. Lemaire. "Stochastic finite element : a non intrusive approach by regression". *Revue européenne de mécanique numérique* 15.1-2-3 (2006), p. 81–92.

[11] G. Beylkin et M. J. Mohlenkamp. "Algorithms for Numerical Analysis in High Dimensions". *SIAM Journal on Scientific Computing* 26.6 (2005), p. 2133–2159.

[12] B. Bognet et al. "Advanced simulation of models defined in plate geometries : 3D solutions with 2D computational complexity". *Computer Methods in Applied Mechanics and Engineering* 201–204 (2012), p. 1–12.

[13] H.-J. Bungartz et M. Griebel. "Sparse grids". *Acta Numerica* 13 (2004), p. 147–269.

[14] R. H. Cameron et W. T. Martin. "The Orthogonal Development of Non-Linear Functionals in Series of Fourier-Hermite Functionals". *The Annals of Mathematics* 48.2 (1947), p. 385–392.

[15] S. S. Chen, D. L. Donoho et M. A. Saunders. "Atomic Decomposition by Basis Pursuit". *SIAM Journal on Scientific Computing* 20.1 (1998), p. 33–61.

[16] F. Chinesta et al. "Alleviating mesh constraints : Model reduction, parallel time integration and high resolution homogenization". *Computer Methods in Applied Mechanics and Engineering* 197.5 (2008), p. 400–413.

[17] S.-K. Choi, R. V. Grandhi et R. A. Canfield. "Structural reliability under non-Gaussian stochastic behavior". *Computers & Structures* 82.13–14 (2004), p. 1113–1121.

[18] A. Cohen, R. Devore et C. Schwab. "Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDEs". *Analysis and Applications* 09.01 (2011), p. 11–47.

[19] A. Cohen, R. DeVore et C. Schwab. "Convergence Rates of Best N-term Galerkin Approximations for a Class of Elliptic sPDEs". *Foundations of Computational Mathematics* 10.6 (2010), p. 615–646.

[20] T. Crestaux. "Méthode adaptative d'intégration multi-dimensionnelle et sélection d'une base de polynômes de chaos". Thèse de doct. Université de Paris XIII, 2011.

[21] L. De Lathauwer, B. De Moor et J. Vandewalle. "A Multilinear Singular Value Decomposition". *SIAM Journal on Matrix Analysis and Applications* 21.4 (2000), p. 1253–1278.

[22] L. De Lathauwer, B. De Moor et J. Vandewalle. "On the Best Rank-1 and Rank-(R1, R2, ..., RN) Approximation of Higher-Order Tensors". *SIAM Journal on Matrix Analysis and Applications* 21.4 (2000), p. 1324–1342.

[23] A. Defant et K. Floret. *Tensor Norms and Operator Ideals*. 1re éd. North-Holland, 1992.

[24] A. Doostan et H. Owhadi. "A non-adapted sparse approximation of PDEs with stochastic inputs". *Journal of Computational Physics* 230.8 (2011), p. 3015–3034.

[25] C. Eckart et G. Young. "The approximation of one matrix by another of lower rank". *Psychometrika* 1.3 (1936), p. 211–218.

[26] O. G. Ernst et al. "On the convergence of generalized polynomial chaos expansions". *ESAIM : Mathematical Modelling and Numerical Analysis* 46.2 (2011), p. 317–339.

[27] M. Espig et W. Hackbusch. "A regularized Newton method for the efficient approximation of tensors represented in the canonical tensor format". *Numerische Mathematik* (), p. 1–37.

[28] A. Falcó et W. Hackbusch. "On Minimal Subspaces in Tensor Representations". *Soumis à Foundations of Computational Mathematics* (2010).

[29] A. Falcó et A. Nouy. "Proper generalized decomposition for nonlinear convex problems in tensor Banach spaces". *Numerische Mathematik* 121.3 (2012), p. 503–530.

[30] R. W. Freund et N. M. Nachtigal. "QMR : a quasi-minimal residual method for non-Hermitian linear systems". *Numerische Mathematik* 60.1 (1991), p. 315–339.

[31] R. Ghanem. "Ingredients for a general purpose stochastic finite elements implementation". *Computer Methods in Applied Mechanics and Engineering* 168.1–4 (1999), p. 19–34.

[32] R. G. Ghanem et R. M. Kruger. "Numerical solution of spectral stochastic finite element systems". *Computer Methods in Applied Mechanics and Engineering* 129.3 (1996), p. 289–303.

[33] R. G. Ghanem et P. D. Spanos. *Stochastic Finite Elements : A Spectral Approach*. Springer-Verlag. 1991.

[34] L. Grasedyck. "Hierarchical Singular Value Decomposition of Tensors". *Siam Journal on Matrix Analysis and Applications* 31.4 (2010), p. 2029–2054.

[35] M. A. Grepl et al. "Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations". *Esaim-Mathematical Modelling and Numerical Analysis-Modelisation Mathematique Et Analyse Numerique* 41.3 (2007), p. 575–605.

[36] M. J. Grote et T. Huckle. "Parallel Preconditioning with Sparse Approximate Inverses". *SIAM Journal on Scientific Computing* 18.3 (1997), p. 838.

[37] W. Hackbusch et S. Kuehn. "A New Scheme for the Tensor Representation". *Journal of Fourier Analysis and Applications* 15.5 (2009), p. 706–722.

[38] W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Springer Series in Computational Mathematics 42. Springer, 2012.

[39] M. R. Hestenes et E. Stiefel. "Methods of Conjugate Gradients for Solving Linear Systems". *Journal of Research of the National Bureau of Standards* 49.9 (1952), p. 409–436.

[40] F. L. Hitchcock. "The expression of a tensor or a polyadic as a sum of products". *Journal of Mathematics and Physics* 6 (1927), p. 164–189.

[41] H. Hotelling. "Analysis of a complex of statistical variables into principal components". *Journal of Educational Psychology* 24.6 (1933), p. 417–441.

[42] T. Kanit et al. "Determination of the size of the representative volume element for random composites : statistical and numerical approach". *International Journal of Solids and Structures* 40.13–14 (2003), p. 3647–3679.

[43] T. Kanit et al. "Apparent and effective physical properties of heterogeneous materials : Representativity of samples of two materials from food industry". *Computer Methods in Applied Mechanics and Engineering* 195.33–36 (2006), p. 3960–3982.

[44] K. Karhunen. "Zur Spektraltheorie stochastischer Prozesse". *Annales Academiae Scientiarum Fennicae* 37 (1946).

[45] B. Khoromskij. "O(d log N)-Quantics Approximation of N-d Tensors in High-Dimensional Numerical Modeling". *Constructive Approximation* 34.2 (2011), p. 257–280.

[46] B. Khoromskij. "Tensor-Structured Preconditioners and Approximate Inverse of Elliptic Operators in R^d". *Constructive Approximation* 30.3 (2009), p. 599–620.

[47] T. G. Kolda et B. W. Bader. "Tensor Decompositions and Applications". *Siam Review* 51.3 (2009), p. 455–500.

[48] D. Kressner et C. Tobler. *htucker - A MATLAB toolbox for tensors in hierarchical Tucker format*. 2012.

[49] D. Kressner et C. Tobler. "Krylov Subspace Methods for Linear Systems with Tensor Product Structure". *Siam Journal on Matrix Analysis and Applications* 31.4 (2010), p. 1688–1714.

[50] D. Kressner et C. Tobler. "Low-Rank Tensor Krylov Subspace Methods for Parametrized Linear Systems". *Siam Journal on Matrix Analysis and Applications* 32.4 (2011), p. 1288–1316.

[51] D. Kressner et C. Tobler. "Preconditioned Low-Rank Methods for High-Dimensional Elliptic PDE Eigenvalue Problems". *Computational Methods in Applied Mathematics* 11.3 (2011), p. 363–381.

[52] P. Ladevèze. *Nonlinear Computational Structural Mechanics*. Springer-Verlag, 1999.

[53] C. Lanczos. "Solution of Systems of Linear Equations by Minimized Iterations". *Journal of Research of the National Institute of Standards and Technology* 49.1 (1952), p. 33–53.

[54] A. N. Langville et W. J. Stewart. "A Kronecker product approximate preconditioner for SANs". *Numerical Linear Algebra with Applications* 11.8-9 (2004), 723–752.

[55] O. P. Le Maître et O. M. Knio. *Spectral Methods for Uncertainty Quantification : With Applications to Computational Fluid Dynamics*. 1st Edition. Springer, 2010.

[56] M. Loève. *Probability Theory II*. 4th. Springer, 1978.

[57] J. Lumley. "The Structure of Inhomogeneous Turbulent Flows". *Atmospheric turbulence and radio propagation*. Sous la dir. d'A. Yaglom et V. Tatarski. Nauka, 1967, p. 166–178.

[58] H. G. Matthies et A. Keese. "Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations". *Computer Methods in Applied Mechanics and Engineering* 194.12–16 (2005), p. 1295–1331.

[59] A. K. Noor et J. M. Peters. "Reduced basis technique for nonlinear analysis of structures". *AIAA* 18.4 (1980), p. 455–462.

[60] A. Nouy. "A priori model reduction through Proper Generalized Decomposition for solving time-dependent partial differential equations". *Computer Methods in Applied Mechanics and Engineering* 199.23–24 (2010), p. 1603–1626.

[61] A. Nouy. "Proper Generalized Decompositions and Separated Representations for the Numerical Solution of High Dimensional Stochastic Problems". *Archives of Computational Methods in Engineering* 17.4 (2010), p. 403–434.

[62] A. Nouy. "Recent Developments in Spectral Stochastic Methods for the Numerical Solution of Stochastic Partial Differential Equations". *Archives of Computational Methods in Engineering* 16.3 (2009), p. 251–285.

[63] I. V Oseledets. "DMRG Approach to Fast Linear Algebra in the TT-Format". *Computational Methods in Applied Mathematics* 11.3 (2011), p. 382 –393.

[64] N. Parés, P. Díez et A. Huerta. "Bounds of functional outputs for parabolic problems. Part I : Exact bounds of the discontinuous Galerkin time discretization". *Computer Methods in Applied Mechanics and Engineering* 197.19–20 (2008), p. 1641–1660.

[65] Y. Pati, R. Rezaiifar et P. Krishnaprasad. "Orthogonal matching pursuit : recursive function approximation with applications to wavelet decomposition". *1993 Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers, 1993.* 1993, 40 –44 vol.1.

[66] K. Pearson. "On lines and planes of closest fit to systems of points in space". *Philosophical Magazine* 2.6 (1901), p. 559–572.

[67] G. Rozza, D. B. P. Huynh et A. T. Patera. "Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations". *Archives of Computational Methods in Engineering* 15.3 (2008), p. 229–275.

[68] Y. Saad et M. H. Schultz. "GMRES : A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems". *SIAM Journal on Scientific and Statistical Computing* 7.3 (1986), p. 856.

[69] Y. Saad. *Iterative Methods for Sparse Linear Systems, Second Edition.* 2^{e} éd. Society for Industrial et Applied Mathematics, 2003.

[70] V. de Silva et L.-H. Lim. "Tensor Rank and the Ill-Posedness of the Best Low-Rank Approximation Problem". *Siam Journal on Matrix Analysis and Applications* 30.3 (2008), p. 1084–1127.

[71] C. Soize et R. G. Ghanem. "Physical Systems with Random Uncertainties : Chaos Representations with Arbitrary Probability Measure". *SIAM Journal on Scientific Computing* 26.2 (2004), p. 395–410.

[72] P. Sonneveld. "CGS, a fast Lanczos-type solver for nonsymmetric linear systems". *SIAM Journal on Scientific and Statistical Computing* 10.1 (1989), 36–52.

[73] B. Sudret. "Global sensitivity analysis using polynomial chaos expansions". *Reliability Engineering & System Safety* 93.7 (2008), p. 964–979.

[74] V. N. Temlyakov. "Greedy Approximation". *Acta Numerica* 17 (2008), p. 235–409.

[75] R. Tibshirani. "Regression Shrinkage and Selection Via the Lasso". *Journal of the Royal Statistical Society, Series B* 58 (1994), 267–288.

[76] A. Touzene. "A Tensor Sum Preconditioner for Stochastic Automata Networks". *INFORMS Journal on Computing* 20.2 (2008), p. 234–242.

[77] L. R. Tucker. "Some mathematical notes on three-mode factor analysis". *Psychometrika* 31.3 (1966), p. 279–311.

[78] E. Ullmann. "A Kronecker Product Preconditioner for Stochastic Galerkin Finite Element Discretizations". *SIAM Journal on Scientific Computing* 32.2 (2010), p. 923–946.

[79] K. Veroy et A. T. Patera. "Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations : rigorous reduced-basis a posteriori error bounds". *International Journal for Numerical Methods in Fluids* 47.8-9 (2005), p. 773–788.

[80] H. A. van der VORST. "Bi-CGSTAB : A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems". *SIAM Journal on Scientific and Statistical Computing* 13.2 (1992), p. 631–644.

[81] H. A. van der VORST. *Iterative Krylov Methods for Large Linear Systems*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2003.

[82] N. WIENER. "The Homogeneous Chaos". *American Journal of Mathematics* 60.4 (1938), p. 897–936.

[83] D. B. XIU et J. S. HESTHAVEN. "High-order collocation methods for differential equations with random inputs". *Siam Journal on Scientific Computing* 27.3 (2005), p. 1118–1139.

[84] E. ZANDER. "Tensor Approximation Methods for Stochastic Problems". Thèse de doct. TU Braunschweig, 2012.

[85] B. ØKSENDAL. *Stochastic Differential Equations : An Introduction with Applications*. 6th. Springer-Verlag, 2003.