



PGD-Abaques virtuels pour l'optimisation géométrique des structures

Amaury Courard

► **To cite this version:**

Amaury Courard. PGD-Abaques virtuels pour l'optimisation géométrique des structures. Génie mécanique [physics.class-ph]. Université Paris-Saclay, 2016. Français. <NNT : 2016SACLN010>. <tel-01266281>

HAL Id: tel-01266281

<https://tel.archives-ouvertes.fr/tel-01266281>

Submitted on 2 Feb 2016

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

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

NNT : 2016SACLN010

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PARIS-SACLAY,
préparée à l'École Normale Supérieure de Cachan

ÉCOLE DOCTORALE N° 579
Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMaG)

Spécialité de doctorat
MÉCANIQUE - GÉNIE MÉCANIQUE - GÉNIE CIVIL

Par

M. Amaury Courard

PGD-Abaques virtuels pour l'optimisation géométrique des structures

Thèse présentée et soutenue à Cachan le 18 janvier 2016 :

Composition du Jury :

M. Rodolphe Le Riche	Directeur de recherche CNRS, Ecole des Mines de Saint-Etienne	Président
M. Anthony Gravouil	Professeur, INSA Lyon	Rapporteur
M. Pierre Villon	Professeur, UTC	Rapporteur
M. Philippe Andolfatto	Ingénieur, AIRBUS Defence & Space	Invité
M. Alain Bergerot	Ingénieur, AIRBUS Defence & Space	Invité
M. Ludovic Ballere	Ingénieur, AIRBUS Defence & Space	Examinateur
M. Pierre Ladev�ze	Professeur, ENS Cachan	Directeur de th�se
M. David N�ron	Professeur, ENS Cachan	Co-encadrant



À Mahaut,

Ces remerciements viennent conclure trois années de thèse passées au LMT Ca-
chan. Je tiens tout d'abord à remercier les membres du jury, notamment Rodolphe Le
Riche pour avoir accepté d'en être le président ainsi que Pierre Villon et Anthony Gra-
vouil pour le travail de relecture minutieux fourni, en cette période si chargée qu'est la
fin de l'année. Un grand merci à Ludovic Ballere, Philippe Andolfatto et Alain Bergerot
avec qui j'ai travaillé avec plaisir. Ils m'ont, non seulement, laissé une grande liberté
dans mes recherches mais ont aussi contribué grandement à leur réussite en y appor-
tant un regard extérieur. Mon seul regret aura été de ne pas avoir pu travailler, à cause
de la nature de la thèse, plus en interaction avec eux.

Vient ensuite l'équipe encadrante du LMT avec, en premier lieu, Pierre Ladevèze
dont l'esprit tortueux a accouché du sujet de la thèse et qui m'a donné une grande li-
berté et autonomie dans mon travail. Ses explications claires et son regard bienveillant
ont été d'un grand soutien. Une mention spéciale à David Néron pour l'ensemble de
son œuvre. Toujours présent et souriant, il n'aura jamais douté, ni de l'issue de la thèse,
ni de mes capacités, même dans les moments de grand doute. C'est grâce à lui que j'ai
pu rédiger ce manuscrit dans les temps impartis et ses corrections précieuses rendues
à une vitesse dépassant l'entendement.

Concernant mes rencontres au LMT, je ne peux pas oublier de citer la diaspora ita-
lienne composée de Matteo, Enrico, Andrea, Omar et Claudio qui m'a accueilli et re-
cueilli lors de mon arrivée. De même, je remercie Dominik pour avoir animé mes trois
années de thèse en tant que voisin de bureau. On s'est beaucoup marrés, souvent en-
gueulés mais jamais ennuyés.

Je remercie ensuite Elena et ses mignons, Maxime J., Thibault, Pierre et Hadrien
pour ces derniers mois de thèse entre coups de pied dans les tibias à la savate et le
développement d'une sociologie exhaustive du monde de la recherche et notamment
de la place du doctorant. Je remercie une deuxième fois Hadrien qui aura joué le rôle
de manuel de SAMCEF lors de cette thèse, le véritable étant particulièrement indigeste.
Je tiens aussi à signaler ma reconnaissance à Maxime B., Clément, Benoît et Ionuț pour
avoir daigné m'accepter dans leurs parties de squash quand l'un venait à manquer.

Je termine par ma famille qui a répondu présent le jour de ma soutenance malgré
ce sujet obscur. Je remercie, en particulier, ma mère et ma sœur pour avoir corrigé les
fautes d'orthographe de ce manuscrit sans en comprendre le moindre mot. La dernière
personne qui sera mentionnée dans ces remerciements, est la plus importante, Angé-
lique. Merci de m'avoir soutenu jusqu'au bout, jusqu'aux affres de la préparation de la
soutenance.

Table des matières

Table des matières	i
Introduction	1
1 Réduction de modèle et variations de géométrie	7
1.1 Proper Orthogonal Decomposition	13
1.1.1 Méthode de séparation de variables	13
1.1.2 Réduction de modèle par la méthode Galerkin-POD	17
1.2 Reduced Basis	19
1.3 PGD-Abaque virtuel	20
1.3.1 Proper Generalized Decomposition	20
1.3.2 Exemple numérique	24
1.3.3 Résultats	26
1.3.4 Phase d'update	27
1.4 Cas des paramètres géométriques	29
2 Paramétrisation de la rigidité et maillages universels	35
2.1 Paramétrisation de la rigidité	36
2.1.1 Une nouvelle relation de comportement	36
2.1.2 Calcul de la matrice de rigidité	38
2.1.3 Calcul de la contrainte	40
2.1.4 Application numérique	41
2.1.5 Conclusions partielles	43
2.2 Maillages universels	44
2.2.1 Introduction	44
2.2.2 Sélection des nœuds à projeter	46
2.2.3 Projection des nœuds sélectionnés sur Γ	46
2.2.4 Relaxation des nœuds dans le voisinage du contour Γ	47
2.2.5 Conditions suffisantes	49
2.2.6 Algorithme des maillages universels	51
2.3 Adaptation des maillages universels à la PGD géométrique	51
2.3.1 Double relaxation	52
2.3.2 Nouvelle vérification	53
2.3.3 Nouvel algorithme	53

2.3.4	Méthode PGD-maillages universels	55
2.3.5	Application numérique	55
2.4	Conclusions	57
3	Méthode des transformations géométriques	61
3.1	Transformations géométriques	62
3.1.1	Proper Generalized Decomposition pour les variations de géométrie	63
3.1.2	Calcul de la matrice de rigidité	65
3.1.3	Exemples de transformations géométriques affines	68
3.2	Application numérique	70
3.3	Application à un démonstrateur industriel	72
3.3.1	Présentation du démonstrateur	72
3.3.2	Extension de la méthode au cas axisymétrique	74
3.3.3	Partition de la géométrie	76
3.3.4	Intégration de la méthode dans un process industriel de conception	79
3.4	Résultats	80
4	Discrete Proper Generalized Decomposition	85
4.1	Nouveau démonstrateur	86
4.2	Paramètres discrets	88
4.3	Discrete Proper Generalized Decomposition ou dPGD	89
4.4	Approche hybride	91
4.4.1	Exemples académiques	93
4.4.2	Application au démonstrateur industriel	98
4.5	Conclusions	102
	Conclusion	105
	Bibliographie	109

Introduction

De par la rapide et constante augmentation de la puissance de calcul, en particulier avec le développement du Calcul Haute Performance, il est devenu possible de simuler le comportement de structures extrêmement complexes, ce qui était inenvisageable il y a encore une dizaine d'années. Toutefois, dans le cadre de l'optimisation de forme, il peut être nécessaire d'évaluer un grand nombre de fois la solution d'un problème paramétrique, le but final étant de trouver le jeu de paramètres géométriques optimal de conception.

Malgré la puissance de calcul actuelle, ces temps de résolution cumulés peuvent rendre l'optimisation hors de portée d'un point de vue économique et industriel. En conséquence, les structures industrielles ont tendance à être sur-dimensionnées afin de réduire le temps d'optimisation (la phase d'optimisation est interrompue avant d'atteindre le jeu de paramètres optimal). Cela peut être problématique dans un domaine tel que l'aéronautique où la réduction de la masse est un enjeu industriel. En effet, sur-dimensionner une structure, c'est, avant tout, augmenter son poids et par conséquent, sa consommation de carburant.

Les algorithmes d'optimisation proprement dits sont nombreux et ne sont pas au cœur de ce travail. Retenons simplement qu'ils conduisent tous à l'évaluation de la solution du problème pour différents jeux de paramètres. Dans ce contexte *multi-query* (voir [Boyaval *et al.*, 2010, Manzoni *et al.*, 2012b]), il peut être judicieux de résoudre le problème paramétrique complet une bonne fois pour toutes et de stocker la solution obtenue dans une base de données. On pourra ensuite s'y référer, à la volée, à chaque fois que l'évaluation de la solution est nécessaire. C'est pourquoi le concept d'abaque est toujours d'actualité.

Les abaques furent, pendant des siècles, de puissants outils d'aide à la décision dans des domaines variés tels que l'architecture, l'ingénierie (*Vademecum des Mechaniker* [Bernoulli, 1836]) et les sciences en général. Ainsi, en 1202, grâce à son *Liber Abaci*, Leonardo Fibonacci fut le premier à introduire le système de numération indo-arabe en Europe occidentale où prévalait encore le système romain. Il y décrit non seulement les nombres indo-arabes mais aussi comment les utiliser. Fibonacci, étant issu d'une famille de marchands, explique, en particulier, en quoi ils facilitent les conversions de monnaies et le calcul de profit et d'intérêt. Cet abaque fut d'une grande aide pour tous les novices au prise avec le système de numération indo-arabe et a contribué grandement à sa diffusion en Europe occidentale en partant des banquiers et marchands florentins. Les abaques sont, par la suite, devenus l'outil de référence d'aide à

la conception de l'ingénieur. En effet, le temps de conception se retrouve fortement réduit grâce à la consultation immédiate de résultats déjà établis une bonne fois pour toutes. Un exemple d'abaque utilisé de nos jours pour le dimensionnement de poutres en flambage est présenté sur la Figure 1. Cet abaque, issu de l'Eurocode 3, est utilisé quotidiennement pour la conception de structure métallique en Génie Civil.

Table B.3: Values of factors C_1 , C_2 and C_3 corresponding to values of factor k : Transverse loading cases



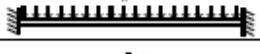

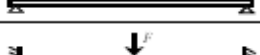

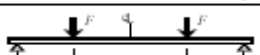



Loading and support conditions	Bending moment diagram	Values of k	Values of factors		
			C_1	C_2	C_3
		1,0	1,132	0,459	0,525
		0,5	0,972	0,304	0,980
		1,0	1,285	1,562	0,753
		0,5	0,712	0,652	1,070
		1,0	1,365	0,553	1,730
		0,5	1,070	0,432	3,050
		1,0	1,565	1,267	2,640
		0,5	0,938	0,715	4,800
		1,0	1,046	0,430	1,120
		0,5	1,010	0,410	1,890

FIGURE 1 – Abaque aidant au dimensionnement de poutres en flambage pour des chargements transversaux [Eurocode 3, 2010].

Ce travail se base sur cette idée d'abaques, construits cette fois numériquement par le biais d'outils de simulations dédiés. On parle alors d'*abaques virtuels* (en anglais *virtual charts* [Ladevèze, 2013] ou *computational vademecum* [Chinesta *et al.*, 2013]). En variant la valeur donnée à un jeu de paramètres géométriques, on peut construire toute une famille de structures à partir d'une structure de référence. Les résultats des simulations numériques sont stockés dans des abaques virtuels, pour chaque famille de structures. L'utilisateur final n'aura alors qu'à rechercher les jeux de paramètres licites parmi tous les jeux de paramètres considérés dans un certain intervalle de valeurs. La construction d'un abaque virtuel peut être coûteux en temps mais, puisqu'il est établi en amont, il est avantageux de perdre du temps à cette étape au regard de celui gagné lors de son utilisation, surtout dans le cas de tâches répétitives.

Cependant, comme il a été précédemment mentionné, la construction d'abaques virtuels demeure hors de portée d'approches éléments finis standards utilisant la « force brute », c'est-à-dire basées sur la résolution du problème paramétrique complet pour chaque jeu de paramètres, à cause du temps de calcul prohibitif.

Le choix, qui a été fait au cours de la thèse, est de construire les abaques à l'aide d'une technique de réduction de modèle. En effet, la réduction de modèle permet la construction rapide d'approximation de champs et la représentation à variables séparées qu'elle donne de ces champs, en facilite l'utilisation effective.

La communauté de la réduction de modèle est très active et de nombreuses straté-

gies différentes sont, de nos jours, disponibles. Une des méthodes les plus populaires repose sur la Proper Orthogonal Decomposition (POD), qui est basée sur un ensemble de résolutions préalables appelées *snapshots* ou échantillons [Chatterjee, 2000] du problème complet pour des valeurs données des paramètres. De ces échantillons est extraite une base réduite à l'aide de diverses méthodes toutes équivalentes : Décomposition en Valeurs Singulières (SVD) [Beltrami, 1873, Jordan, 1874a, Sylvester, 1889a], transformation de Karhunen-Loève (KLT) [Karhunen, 1947, Loève, 1955] connue aussi sous le nom d'Analyse en Composantes Principales (PCA) [Pearson, 1901, Hotelling, 1933]. Dans un second temps, le problème paramétrique complet est projeté sur cette base réduite grâce à une méthode type Galerkin. La dimension du problème à résoudre est alors fortement réduite. Elle est utilisée dans de nombreux domaines : en mécanique des fluides [Holmes, 2012], pour la résolution de problèmes élastoplastiques dynamiques [Corigliano *et al.*, 2015], aéroélastiques [Epureanu *et al.*, 2004], stochastiques [Schenk et Schuëller, 2003], multiparamétriques [Gunzburger *et al.*, 2007], d'équations paraboliques [Atwell et King, 2001], dans le domaine des MEMS (microelectromechanical systems) [Hung et Senturia, 1999], pour l'analyse modale [Han et Feeny, 2003], de systèmes multicorps [Quaranta *et al.*, 2004], en météorologie et océanographie [Cao *et al.*, 2007] (voir [Kerschen *et al.*, 2005] pour une revue des applications), etc. La méthode des Reduced Basis (RB, voir par exemple [Maday et Ronquist, 2004, Veroy et Patera, 2005, Rozza *et al.*, 2008, Nguyen, 2008, Rozza, 2011, Galvis et Kang, 2014]) ajoute une sélection automatique de ces *snapshots* grâce à un algorithme *greedy* basé sur des indicateurs d'erreur.

Dans le cadre de cette thèse, il a été fait le choix d'utiliser la Proper Generalized Decomposition (PGD). Elle ne nécessite aucune connaissance *a priori* de la solution réelle. La base réduite PGD est construite de manière itérative en même temps que l'approximation. Il est alors possible de calculer une approximation de façon automatique. Elle fut introduite pour la première fois dans [Ladevèze, 1985] sous le nom d'*Approximation Radiale* dans le cadre du solveur non incrémental LaTIn, dédié à la résolution de problèmes fortement non linéaires. Dans cet environnement, elle fut utilisée pour la construction d'approximations à variables séparées de champs dépendant du temps et de l'espace et fut appliquée à des problèmes visco-plastiques [Boisse *et al.*, 1990, Cognard et Ladevèze, 1993, Relun *et al.*, 2013], au délaminage des composites [Alix et Ladevèze, 1992], à des problèmes paramétrés [Vitse *et al.*, 2014, Néron *et al.*, 2015], multi-physiques [Cognard *et al.*, 1999, Néron et Dureisseix, 2008a, Néron et Dureisseix, 2008b, Néron et Ladevèze, 2010], multi-échelles [Cremonesi *et al.*, 2013], etc.

La PGD a fait l'objet d'un regain d'intérêt suite aux travaux des équipes de recherche de Chinesta [Mokdad *et al.*, 2007, Chinesta *et al.*, 2010a, González *et al.*, 2010, Bognet *et al.*, 2012a]. Elle a depuis été appliquée à différents types de domaines : problèmes stochastiques [Nouy, 2007, Nouy, 2009], multidimensionnels [Chinesta *et al.*, 2010b], hyperélastiques [Niroomandi *et al.*, 2013], paraboliques [Bonithon *et al.*, 2011], visco-élastiques [Ammar *et al.*, 2015], en acoustique [Barbarulo *et al.*, 2014], en mécanique des fluides [Ammar *et al.*, 2007a, Ammar *et al.*, 2007b], en *monitoring* de procédés thermiques [Aguado *et al.*, 2015], en mécanique de la rupture [Giner *et al.*, 2013], pour des

simulations temps réel en chirurgie [González *et al.*, 2015, Alfaro *et al.*, 2014] ou encore pour la vérification de modèle et estimation d'erreur [Ladevèze et Chamoin, 2011, Ladevèze et Chamoin, 2013].

Concernant le cas des variations de géométrie pour l'optimisation de forme, la littérature dans le domaine de la réduction de modèle est plus pauvre car elles impliquent des difficultés techniques qui seront détaillées au cours de la thèse. Toutefois, au niveau du principe, les choses sont simples : il s'agit d'un problème avec paramètres où ces paramètres sont associés à la géométrie. Des méthodes basées sur la POD ont traité de ce problème. Dans [Rozza et Veroy, 2007, Manzoni *et al.*, 2012a], les paramètres géométriques sont pris en compte dans l'environnement de la RB et [Raghavan *et al.*, 2013] mélange une approche PCA et Diffusion approximation. L'article [Ammar *et al.*, 2014] prolonge la PGD aux variations de géométrie et applique la méthode à des exemples académiques en profitant du fait que les variations géométriques peuvent être vues comme un problème paramétrique bien adapté à la PGD. La même approche a été également adoptée dans [Zlotnik *et al.*, 2015]. Dans les deux cas, l'étude s'est restreinte à des problèmes thermiques.

Ce qui caractérise nos travaux est d'aller vers des outils aptes à traiter des situations significatives de la complexité des problèmes mécaniques rencontrés au niveau industriel. En particulier, l'exploitation de codes éléments finis industriels est une contrainte majeure.

Le long de la thèse, nous présenterons deux méthodes permettant de prendre en compte les variations de géométrie dans le cadre de la PGD. La première est basée sur la paramétrisation de la rigidité d'une structure de référence englobant toutes les configurations possibles. Cette structure est composée de deux parties distinctes : une partie « dure » correspondant à la structure associée à un paramètre donné et une partie « molle » pour le reste de la géométrie. Pour mettre en place cette méthode de manière efficace, nous proposons également une extension de cette méthode par les maillages universels [Rangarajan et Lew, 2012] permettant la construction rapide des maillages conformes pour des géométries immergées dans un même maillage par le déplacement local de certains des nœuds. Nous montrerons comment modifier l'algorithme initial des maillages universels afin de l'adapter au cadre PGD.

La deuxième méthode repose, quant à elle, sur le recours à des transformations géométriques autorisant la redéfinition des problèmes associés à chaque configuration géométrique sur une structure de référence suivant la voie empruntée par [Ammar *et al.*, 2014] et [Zlotnik *et al.*, 2015], étendue, ici, à des géométries complexes type spline. Le travail est mené dans le cadre de l'élaboration d'un démonstrateur illustrant une famille de problèmes industriels proposé par AIRBUS Defence & Space. Sur ce dernier point, nous nous sommes attachés à intégrer la méthodologie développée dans un process industriel de conception basé sur des logiciels commerciaux. À cause de la nature hautement intrusive de la PGD, nous avons dû coupler un code MATLAB avec le logiciel éléments finis SAMCEF dans cette optique.

Dans le dernier chapitre, nous nous intéressons au fait que certains types de paramètres sont intrinsèquement discrets. Le chargement, par exemple peut présenter

plusieurs configurations différentes d'application. Ces cas de chargement peuvent être regroupés dans une même variable discrète prenant comme valeur le numéro de la configuration. Il est montré que la PGD peut intégrer ce type de paramètres sans grandes modifications et qu'il est possible d'adopter une démarche hybride, à savoir d'utiliser, à la fois, des paramètres discrets et des paramètres continus. Cette démarche a été adoptée pour l'étude du démonstrateur industriel qui a été étoffé pour l'occasion. En effet, le problème considéré dans ce chapitre présente des non-linéarités fortes telles que des grands déplacements, grandes déformations et non-linéarités matériau.

La thèse est structurée comme suit :

- Le premier chapitre présente le problème d'optimisation associé à la pièce industrielle étudiée. Après une rapide présentation des notions liées à l'optimisation de forme et des outils à disposition, une série de rappels sur les principales techniques de réduction de modèle existantes sera fait. On s'attardera plus particulièrement sur la méthode PGD, technique retenue pour la suite de l'étude. Ce chapitre se termine sur quelques observations portant sur les variations géométriques et sur les difficultés supplémentaires qu'elles apportent en comparaison des autres types de paramètres.
- Le second chapitre est consacré à la première méthode développée pour prendre en compte les paramètres géométriques au sein de la PGD. Cette méthode, basée sur la paramétrisation d'une géométrie englobant toutes les configurations possibles, est appliquée sur un exemple académique. Dans un second temps, on incorpore la technique des maillages universels, adaptée pour l'occasion à la PGD, dans cette méthode. Elle est de nouveau testée sur le même exemple.
- Dans le troisième chapitre, une deuxième méthode est introduite reposant sur l'utilisation des transformations géométriques. Elle est appliquée, dans le cas 2D, à l'exemple académique précédent et au démonstrateur industriel dans le cas axisymétrique. Il est proposé une méthodologie pour intégrer cette méthode dans un process industriel de conception utilisant des codes éléments commerciaux.
- Le dernier chapitre traite plus généralement de la méthode PGD en proposant de l'étendre aux paramètres discrets. Il est montré que la PGD n'est pas modifiée dans ce contexte et qu'il est très facile d'adopter des approches hybrides prenant en compte des paramètres discrets et des paramètres continus. Cette méthode est appliquée à des exemples académiques et au démonstrateur industriel en prenant en compte les non-linéarités fortes qu'il comporte.

Chapitre 1

Réduction de modèle et variations de géométrie

L'optimisation de forme est présentée ainsi que les différents concepts qu'elle regroupe. On expose plus précisément le problème d'optimisation qui a été envisagé pour la thèse, à savoir trouver le jeu de paramètres, définissant la forme d'une pièce industrielle, qui permettra de réduire au maximum la masse de celle-ci, tout en respectant les contraintes liées à sa tenue mécanique. Après avoir justifié le choix de la réduction de modèle pour la construction d'une approximation de la solution, les méthodes Galerkin-POD, Reduced Basis et PGD sont rappelées et la notion d'abaque virtuel définie. Ce chapitre se termine sur les particularités des paramètres géométriques et en quoi leur prise en compte dans les différentes méthodes de réduction de modèle n'est pas immédiate.

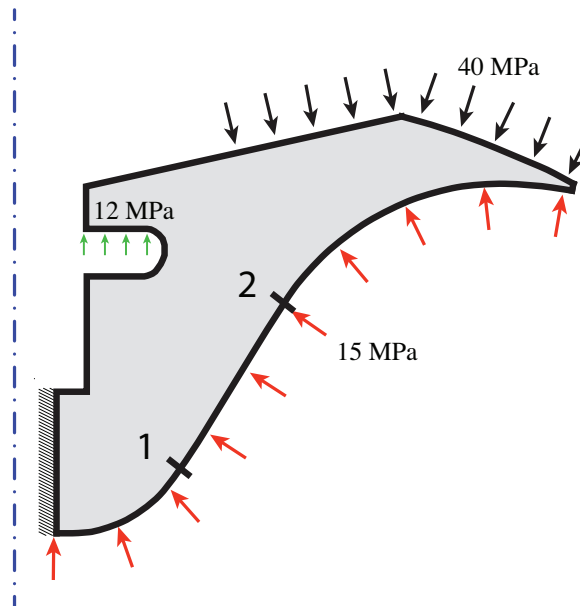


FIGURE 1.1 – Structure considérée.

Au cours de cette thèse, nous nous intéresserons à un démonstrateur, illustrant une famille de problèmes industriels, proposé par AIRBUS Defence & Space et portant sur une composante de lanceur spatial présentée sur la Figure 1.1. La géométrie de la structure est axisymétrique et le matériau est métallique. La problématique sera d'optimiser la forme de cette pièce en jouant sur certains paramètres géométriques afin de réduire sa masse. On devra, de plus, veiller à obtenir une structure finale conforme au cahier des charges, c'est-à-dire de limiter la contrainte maximale en deçà d'un seuil donné.

Dans un cadre plus général, lors de la conception d'une pièce industrielle, la structure dimensionnée doit supporter les sollicitations extérieures propres à son utilisation, et, dans le même temps, rester la plus légère possible ou que la quantité de matière utilisée soit la plus faible possible. Une conception débouchant sur une pièce sur-dimensionnée peut se révéler, en effet, très coûteuse lors de la phase de fabrication (quantité de matériaux employée excessive) ou pendant la phase d'utilisation, tout particulièrement dans le domaine des lanceurs où la réduction de la masse structurale permet de maximiser la performance en augmentant la masse de la charge utile que l'on peut mettre à poste. C'est pourquoi l'optimisation de forme est un enjeu économique et industriel de premier ordre car c'est en amont que la problématique du juste dimensionnement doit être prise en compte. Toutefois, l'optimisation de forme peut regrouper des catégories de problèmes sensiblement différents. En effet, le cas le plus général possible serait d'obtenir la géométrie optimale telle que la structure réponde aux sollicitations mécaniques. Mais il peut aussi s'agir de la simple recherche du jeu de paramètres optimal pour une géométrie déjà définie en amont. L'optimisation de forme peut avoir un sens très large selon le point de vue adopté. Il est donc opportun

de définir les différentes catégories de problèmes que peut regrouper l'optimisation de forme. On décompte trois grandes classes de problèmes [Allaire, 2007] :

Optimisation paramétrique : Il s'agit du cas le plus simple d'optimisation de forme. La géométrie est fixée et la forme est représentée par un nombre réduit de paramètres (par exemple un rectangle sera paramétré par deux longueurs, un cercle par un rayon etc.). Le problème se réduit à la recherche du jeu optimal des paramètres. Ce type d'optimisation correspond au cas classique d'un dimensionnement de structure dont la géométrie a été choisie au préalable, c'est pourquoi [Duysinx, 1996] lui attribue le nom de *dimensionnement automatique des structures*. Sur la Figure 1.2(a) est représenté un exemple d'optimisation paramétrique où l'épaisseur de deux barres est paramétrée et le problème consiste à rechercher l'épaisseur optimale. La géométrie est fixée (les deux barres sont représentées par deux rectangles) et peu de paramètres composent le problème.

Optimisation géométrique : La frontière optimale de la structure est recherchée dans un cadre large mais toujours à topologie fixée. Ainsi, la frontière peut être paramétrée par un ensemble de points de contrôle dont les différentes positions déterminent les configurations géométriques de la structure. Il s'agit d'un cas plus général que l'optimisation paramétrique mais, puisque la topologie est fixée, le nombre de membres structuraux ainsi que le nombre de trous ne peuvent varier. Ce type d'optimisation comporte généralement plus de paramètres de conception que la simple optimisation paramétrique. La Figure 1.2(b) traite le cas d'une optimisation géométrique où la géométrie d'un trou est paramétrée. Ici, c'est véritablement sur la forme du trou que porte l'optimisation. Elle peut être en forme d'ellipse, de cercle ou autre. Pour une illustration de l'optimisation géométrique, on pourra se reporter à [Le Riche et Cailletaud, 1998].

Optimisation topologique : Il s'agit du cadre le plus général de l'optimisation de forme [Bendsøe, 1995, Bendsøe et Sigmund, 2003]. Ici, aucun *a priori* n'est donné sur la topologie (connectivité des domaines, présence et nombre de trous, etc.). Cette généralité est aussi synonyme d'un très grand nombre de paramètres d'optimisation (Figure 1.2(c)).

Pour notre cas d'étude, à savoir l'optimisation d'une structure industrielle liée à l'aérospatiale, nous nous intéresserons aux deux premiers types d'optimisation cités. Plus précisément, le problème d'optimisation sera de minimiser la masse M de la structure par rapport à un ensemble de paramètres de conception $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_n \subset \mathbb{R}^n$ tout en maintenant, par exemple, la contrainte de Von Mises σ_{VM} sous un certain seuil σ_{VM}^0 afin de garantir la viabilité de la structure. Cela se ramène donc au cas d'un problème d'optimisation sous contraintes que l'on pourrait formuler

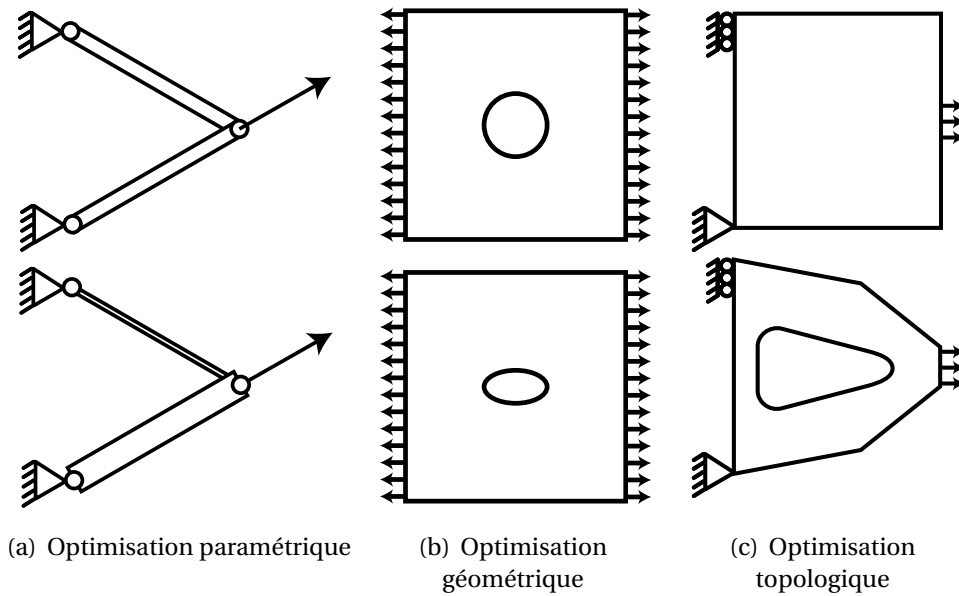


FIGURE 1.2 – Les trois classes de problèmes d'optimisation de structures [Duysinx, 1996].

de la manière suivante :

Problème 1.1. Trouver $\alpha^* = (\alpha_1^*, \dots, \alpha_n^*) \in \mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_n$ tel que

$$\alpha^* = \arg \min_{\substack{\alpha \in \mathcal{A} \\ \sigma_{VM}(\alpha) \leq \sigma_{VM}^0}} M(\alpha)$$

Dans un premier temps, on calcule l'expression de la contrainte afin de restreindre l'espace \mathcal{A} , de tous les paramètres géométriques, à \mathcal{A}_{σ_0} , celui des paramètres engendrant les structures licites, i.e. les structures dont la contrainte de Von Mises n'excède pas le seuil σ_{VM}^0 . La recherche du jeu de paramètres optimal pourra, ensuite, se limiter à l'espace \mathcal{A}_{σ_0} . Dès lors, il suffira de résoudre le problème d'optimisation sans contrainte suivant :

Problème 1.2. Trouver $\alpha^* = (\alpha_1^*, \dots, \alpha_n^*) \in \mathcal{A}_{\sigma_0}$ tel que

$$\alpha^* = \arg \min_{\alpha \in \mathcal{A}_{\sigma_0}} M(\alpha)$$

Pour cela, on pourra se référer à l'un des très nombreux algorithmes existants. Sont à disposition des algorithmes de recherche locale :

- algorithmes de recherche directe [Wright, 1996, Lewis *et al.*, 2000, Kolda *et al.*, 2003] (par exemple l'algorithme de Hooke-Jeeves [Hooke et Jeeves, 1961])
- méthodes de descente (algorithme de gradient à pas constant, algorithme de gradient à pas optimal [Cauchy, 1847], algorithme des gradients conjugués [Hestenes et Stiefel, 1952, Fletcher et Reeves, 1964])
- méthodes de Newton ou quasi-Newton [Fletcher, 1970]

ou de recherche globale :

- méthodes énumératives [Jones *et al.*, 1993]
- méthodes de recherche aléatoire [Boender *et al.*, 1982]
- méthodes de regroupement [Kan et Timmer, 1987]
- méthodes de descentes généralisées
- méthodes évolutionnaires [Holland, 1962, Le Riche et Gaudin, 1998]
- méthodes basées sur l'intelligence en essais [Dorigo et Stützle, 2003, Poli *et al.*, 2007]
- recuit simulé [Kirkpatrick *et al.*, 1983]

Néanmoins, la difficulté réelle se situe au niveau de la définition de l'espace de recherche \mathcal{A}_{σ_0} car cela revient à évaluer σ pour tout $\alpha \in \mathcal{A}$, ce qui peut se révéler être hors de portée des méthodes classiques en termes de temps de calcul. Pour répondre à cela, des méthodes de type construction de méta-modèles à partir de données partiellement convergées [Laurent *et al.*, 2013a, Laurent *et al.*, 2013b] ou surfaces de réponse [Myers *et al.*, 2009] sont à disposition dans le cadre de l'optimisation.

Nous allons restreindre notre étude aux méthodes de réduction de modèle qui permettent la construction rapide d'approximations sous une forme à variables séparées. Ce dernier point est important car, non seulement le temps de construction de l'approximation est réduit mais son utilisation est aussi grandement facilitée. En effet, évaluer un champ écrit sous forme séparée se ramène à de simples sommes et produits de fonctions. Pour revenir au cadre de la thèse, l'approximation, calculée à partir d'une méthode de réduction de modèle, peut être stockée dans des abaques virtuels une fois pour toutes (il peut s'agir du déplacement, de la contrainte, de quantités d'intérêt, etc.). La réduction de modèle est donc une approche intéressante dans un contexte *multi-query* (qui demande un nombre important d'évaluations [Boyaval *et al.*, 2010, Manzoni *et al.*, 2012b]) caractérisant les problèmes d'optimisation, chaque évaluation correspondant à lire la valeur dans un abaque virtuel.

L'idée générale de la réduction de modèle permet la résolution de problèmes paramétriques dont un exemple typique peut s'écrire sous la forme variationnelle suivante :

Problème 1.3. Pour $\alpha \in \mathcal{A}$, trouver $u(\alpha) \in \mathcal{V}$ tel que

$$\forall v \in \mathcal{V}, \quad a(u(\alpha), v; \alpha) = \ell(v; \alpha)$$

où $\mathcal{V} = \mathbf{H}_0^1(\Omega) = \{v \in \mathbf{H}^1(\Omega) \mid v|_{\partial_u \Omega} = \mathbf{0}\}$, $a(\square, \square; \alpha)$ est une forme bilinéaire et $\ell(\square; \alpha)$ est une forme linéaire.

Dans le cadre de l'élasticité linéaire et stationnaire, le champ recherché sera le vecteur déplacement u et le problème sera régi par les équations et conditions de la forme :

■ Équations d'équilibre

$$\begin{cases} \nabla \cdot \sigma = 0 & \text{dans } \Omega \times \mathcal{A} \\ \sigma \cdot n = f & \text{sur } \partial_d \Omega \times \mathcal{A} \end{cases} \quad (1.1)$$

■ Relation de comportement

$$\sigma = \mathbb{C}(\alpha) \varepsilon \quad \text{dans } \Omega \times \mathcal{A} \quad (1.2)$$

$$= \mathbb{C}(\alpha) \nabla_s(u) \quad \text{dans } \Omega \times \mathcal{A} \quad (1.3)$$

où ∇_s est la partie symétrique de ∇ et \mathbb{C} est l'opérateur de Hooke.

■ Conditions aux limites homogènes

$$u = \mathbf{0} \quad \text{sur } \partial_u \Omega \times \mathcal{A} \quad (1.4)$$

Le Problème 1.3 devient dans ce contexte :

Problème 1.4. Pour $\alpha \in \mathcal{A}$, trouver $u(\alpha) \in \mathcal{V}$ tel que

$$\forall v \in \mathcal{V}, \quad \int_{\Omega} \mathbb{C} \varepsilon(u) : \varepsilon(v) \, d\Omega = \int_{\partial_d \Omega} f \cdot v \, dS$$

où $\mathcal{V} = \mathbf{H}_0^1(\Omega) = \{v \in \mathbf{H}^1(\Omega) \mid v|_{\partial_u \Omega} = \mathbf{0}\}$, $a(\square, \square; \alpha) = \int_{\Omega} \sigma(\square) : \varepsilon(\square) \, d\Omega$
et $\ell(\square; \alpha) = \int_{\partial_d \Omega} f \cdot \square \, dS$.

La contrainte σ est obtenue dans un second temps par un post-traitement du déplacement u en utilisant $\sigma = \mathbb{C} \varepsilon(u)$. Dans une approche type « force brute », le déplacement u serait déterminé pour chaque valeur de jeu de paramètres via la résolution du problème associé à cette valeur. En utilisant les éléments finis comme technique de résolution, cela se traduit par la résolution $n_{\mathcal{A}}$ (nombre de jeux de paramètres) problèmes de taille $n \times n$ (n est le nombre de degrés de liberté), ce qui est évidemment prohibitif si $n_{\mathcal{A}}$ est grand.

1.1 Proper Orthogonal Decomposition

La Proper Orthogonal Decomposition (POD) peut être envisagée sous deux angles différents, d'un côté méthode de séparation de variables et de l'autre outil au service de la réduction de modèle. Intéressons-nous tout d'abord à la méthode de séparation de variables. Dans ce contexte, nous le verrons, la POD regroupe des techniques, toutes équivalentes, mais portant des noms différents selon la communauté scientifique l'utilisant.

1.1.1 Méthode de séparation de variables

1.1.1.1 Décomposition en Valeurs Singulières

La *Décomposition en Valeurs Singulières* (SVD) fut proposée, dans le cadre de l'algèbre linéaire, simultanément par Eugenio Beltrami [Beltrami, 1873] et Camille Jordan [Jordan, 1874a, Jordan, 1874b] puis par James Joseph Sylvester [Sylvester, 1889a, Sylvester, 1889b, Sylvester, 1889c]. Elle fut ensuite étendue aux équations intégrales grâce aux contributions d'Erhard Schmidt (plus connu pour l'algorithme de Gram-Schmidt) [Schmidt, 1907] puis de Hermann Weyl [Weyl, 1912]. C'est à partir de cet instant que la SVD est sortie du monde purement mathématique pour devenir un puissant outil numérique. Pour une revue exhaustive de la genèse de la SVD, on ne saurait que trop conseiller la lecture de [Stewart, 1993].

Théorie Soit \mathbb{M} une matrice $m \times n$ à valeurs sur un corps K (pouvant être le corps des réels ou des complexes). La SVD affirme qu'il existe une décomposition de \mathbb{M} telle que :

$$\mathbb{M} = \mathbb{U}\mathbb{S}\mathbb{V}^* \quad (1.5)$$

où \mathbb{S} est une matrice $m \times n$ dont les coefficients diagonaux sont des réels positifs (appelés *valeurs singulières*), les autres coefficients étant nuls. \mathbb{U} est une matrice unitaire de taille $m \times m$, \mathbb{V} une matrice unitaire de taille $n \times n$ et $*$ est l'opérateur adjoint.

Démonstration. [Ottaviani et Paoletti, 2015] On construit la matrice $\mathbb{M}^*\mathbb{M}$ de taille $n \times n$ qui est semi-définie positive¹. D'après le théorème spectral, il existe une matrice unitaire² $\mathbb{V} \in n \times n$ tel que :

$$\mathbb{V}^*\mathbb{M}^*\mathbb{M}\mathbb{V} = \begin{bmatrix} \mathbb{D}_{rr} & \mathbb{O}_{r \times (n-r)} \\ \mathbb{O}_{(n-r) \times r} & \mathbb{O}_{n-r} \end{bmatrix} \quad (1.6)$$

avec \mathbb{D}_{rr} , une matrice diagonale $r \times r$ ($r = \text{rang}(\mathbb{M}^*\mathbb{M})$) dont les coefficients sont réels, strictement positifs et rangés par ordre décroissant.

1. On suppose $m \geq n$. Dans le cas contraire, on raisonnerait sur la matrice $\mathbb{M}\mathbb{M}^*$ de taille $m \times m$, aussi semi-définie positive.

2. Si K est le corps de réels alors \mathbb{V} est plus particulièrement orthogonale.

Il est possible de décomposer $\mathbb{V} = [\mathbb{V}_1 \quad \mathbb{V}_2]$ de telle sorte que :

$$\begin{bmatrix} \mathbb{V}_1^* \\ \mathbb{V}_2^* \end{bmatrix} \mathbb{M}^* \mathbb{M} [\mathbb{V}_1 \quad \mathbb{V}_2] = \begin{bmatrix} \mathbb{V}_1^* \mathbb{M}^* \mathbb{M} \mathbb{V}_1 & \mathbb{V}_1^* \mathbb{M}^* \mathbb{M} \mathbb{V}_2 \\ \mathbb{V}_2^* \mathbb{M}^* \mathbb{M} \mathbb{V}_1 & \mathbb{V}_2^* \mathbb{M}^* \mathbb{M} \mathbb{V}_2 \end{bmatrix} = \begin{bmatrix} \mathbb{D}_{rr} & \mathbb{O} \\ \mathbb{O} & \mathbb{O} \end{bmatrix} \quad (1.7)$$

On pose $\mathbb{U}_1 = \mathbb{M} \mathbb{V}_1 \mathbb{D}_{rr}^{-\frac{1}{2}}$ et on vérifie de manière immédiate que $\mathbb{U}_1^* \mathbb{U}_1 = \mathbb{I}_r$. Mais la matrice \mathbb{U}_1 n'est pas unitaire car elle peut être non carrée. On la complète alors par \mathbb{U}_2 afin d'obtenir une matrice $\mathbb{U} = [\mathbb{U}_1 \quad \mathbb{U}_2]$ unitaire de taille $m \times m$ et on définit $\mathbb{S} = \begin{bmatrix} \mathbb{D}_{rr}^{\frac{1}{2}} & \mathbb{O}_{r \times (n-r)} \\ \mathbb{O}_{(m-r) \times r} & \mathbb{O}_{(m-r) \times (n-r)} \end{bmatrix}$. On obtient alors le résultat souhaité. ■

Ainsi, pour une fonction $u(x, y)$ avec $(x, y) \in \Omega \subset \mathbb{R}^2$, il est possible, après discrétisation de Ω (N_x nœuds en x et N_y nœuds en y), de définir la matrice :

$$\mathbb{F} = [u(x_i, y_j)]_{(i,j) \in [1, N_x] \times [1, N_y]} \quad (1.8)$$

et d'y appliquer une SVD. Soit $(\mathbb{U}, \mathbb{S}, \mathbb{V}) \in \mathbb{O}_{N_x}(\mathbb{R}) \times \mathcal{M}_{N_x N_y}(\mathbb{R}) \times \mathbb{O}_{N_y}(\mathbb{R})$ ¹ tels que :

$$\mathbb{F} = \mathbb{U} \mathbb{S} \mathbb{V}^T \quad (1.9)$$

où \mathbb{S} est une matrice dont les éléments diagonaux sont positifs et classés par ordre décroissant, les autres coefficients étant nuls.

On peut à partir de cette décomposition écrire u sous forme séparée :

$$u(x, y) = \sum_{k=1}^N a_k \gamma_k(x) \delta_k(y) \quad (1.10)$$

où $a_k = \mathbb{S}(k, k)$, γ_k et δ_k sont, respectivement, les k -ièmes colonnes de \mathbb{U} et \mathbb{V} . N représente les nombres d'éléments non nuls de \mathbb{S} .

Exemple numérique Un exemple d'utilisation de la SVD, tiré de [Chatterjee, 2000], est donné Figure 1.3. La fonction de référence $u(x, y) = e^{-|(x-0.5)(y-1)|} + \sin(xy)$ pour $0 \leq x \leq 1$ et $0 \leq y \leq 2$ est tracée Figure 1.3(a). La Figure 1.4(a) montre la décroissance rapide des valeurs singulières et la prépondérance des toutes premières valeurs sur les suivantes. Cela est confirmé par les Figures 1.3(b), 1.3(c) et 1.3(d) où une approximation de u est construite progressivement en ajoutant un mode à la fois ainsi que par la courbe de convergence de la méthode (voir Figure 1.4(b)). Avec seulement trois modes, on construit une approximation d'une très grande fidélité (erreur de l'ordre de 0,2%), ce qui laisse supposer que l'on puisse tronquer la décomposition SVD au troisième ordre réduisant ainsi fortement la dimensionnalité du problème.

1. $\mathbb{O}_n(\mathbb{R})$ est le *groupe orthogonal* des matrices de dimension $n \times n$.

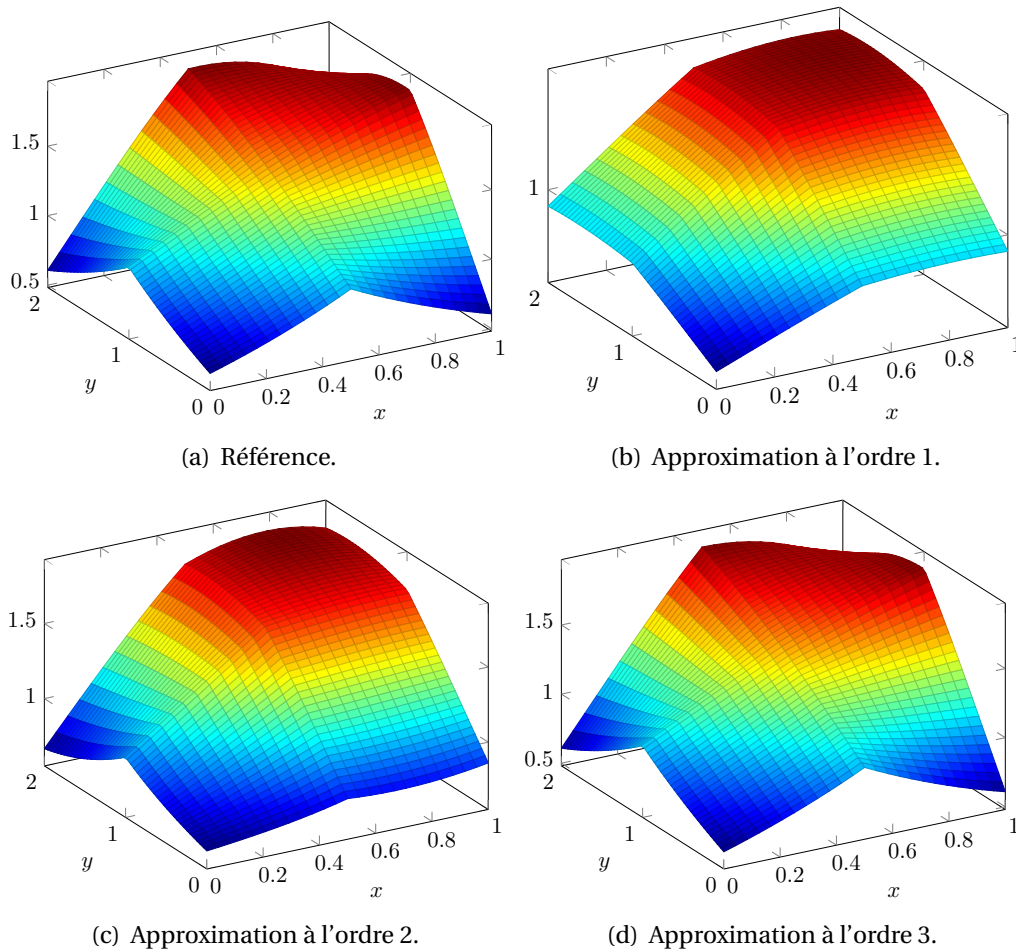


FIGURE 1.3 – Fonction $u(x, y) = e^{-(x-0.5)(y-1)} + \sin(xy)$ et ses approximations aux trois premiers ordres.

Il est à noter que nous avons traité la SVD dans le cadre de deux variables (espace/paramètre). Cette méthode peut être étendue au cas multiparamétrique comme il a été démontré dans les travaux de De Lathauwer (voir [De Lathauwer *et al.*, 2000]). Elle prend alors le nom de HOSVD (*High Order Singular Value Decomposition*). Une *toolbox* MATLAB, traitant de la décomposition à variables séparées pour des champs multiparamétriques, est disponible sous le nom PARAFAC (voir [Bro, 1997]).

1.1.1.2 Transformation de Karhunen-Loève

Connue aussi sous le nom d'Analyse en Composantes Principales (PCA en anglais) [Pearson, 1901, Hotelling, 1933], la transformation Karhunen-Loève (KLT en anglais) [Karhunen, 1947, Loève, 1955] prend en considération les valeurs propres de la matrice de covariance $\mathbb{M}^T \mathbb{M}$ (où $\mathbb{M} \in \mathcal{M}_{m,n}(\mathbb{R})$). La différence qu'elle présente avec la SVD n'est que formelle puisque les valeurs singulières sont les racines carrées des valeurs propres

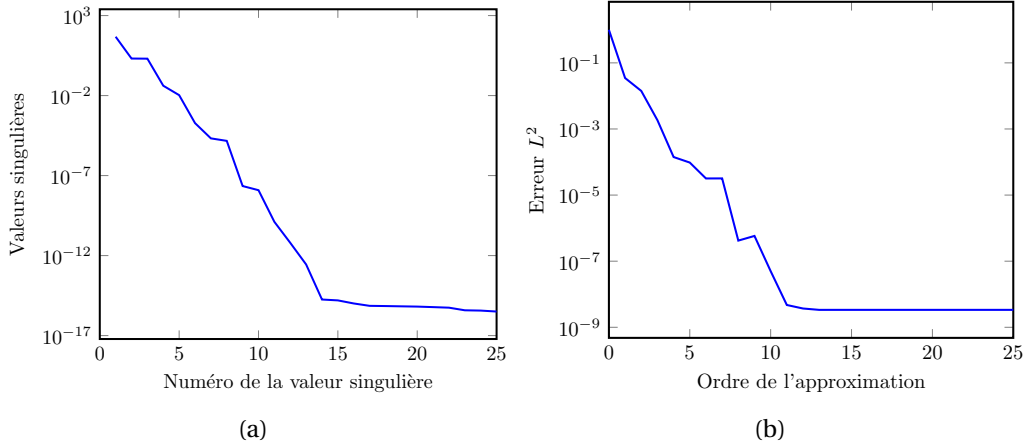


FIGURE 1.4 – (a) Décroissance des valeurs singulières et (b) courbe de convergence de la SVD appliquée à la fonction $u(x, y) = e^{-|(x-0.5)(y-1)|} + \sin(xy)$.

de $\mathbb{M}^T \mathbb{M}$ et nous allons voir que SVD et KLT sont, fondamentalement, de même nature. On peut se reporter à [Wu *et al.*, 2003] pour une démonstration de l'équivalence SVD, KLT et PCA en dimension finie. Nous allons présenter brièvement la relation liant ces méthodes. Comme vu dans la démonstration de la SVD, la matrice $\mathbb{M}^T \mathbb{M}$ est orthogonalement diagonalisable. Soit $\mathbb{V} \in \mathcal{O}_n(\mathbb{R})$ telle que :

$$\mathbb{V}^T \mathbb{M}^T \mathbb{M} \mathbb{V} = \mathbb{D} \quad (1.11)$$

où \mathbb{D} est diagonale.

On note $\mathbb{T} = \mathbb{M} \mathbb{V}$, la nouvelle matrice de données de n variables à présent décorréliées. Or toutes les composantes de cette nouvelle matrice ne sont pas de même importance. C'est pourquoi, on vient rechercher ses N composantes principales puis la tronquer. On garde alors la matrice de taille $m \times N$:

$$\mathbb{T}_N = \mathbb{M} \mathbb{V}_N \quad (1.12)$$

ce qui permet, maintenant, de travailler sur un espace réduit à N variables.

Lien avec la SVD Reprenons la matrice \mathbb{T} que nous venons de définir.

$$\mathbb{T} = \mathbb{M} \mathbb{V} \quad (1.13)$$

Appliquons maintenant la SVD à \mathbb{M} :

$$\mathbb{T} = \mathbb{U} \mathbb{S} \mathbb{V}^T \quad (1.14)$$

$$= \mathbb{U} \mathbb{S} \quad (1.15)$$

Ainsi, SVD et KLT sont l'avvers et le revers de la même médaille, leur nature intrinsèque étant identique et leur distinction purement formelle. Remarqué ce fait, on peut appliquer une SVD pour avoir une décomposition effective de type KLT, ce qui est très souvent fait en pratique.

Nous allons maintenant nous pencher sur l'utilisation de la POD dans le contexte de la réduction de modèle, notamment dans la construction de bases réduites.

1.1.2 Réduction de modèle par la méthode Galerkin-POD

Dans le cadre de la réduction de modèle, la POD est une des plus anciennes techniques utilisées et est, de nos jours, assez répandue dans le monde scientifique, notamment en mécanique des fluides [Holmes, 2012], pour la résolution de problèmes élastoplastiques [Corigliano *et al.*, 2015, Zhang *et al.*, 2015], aéroélastiques [Eppureanu *et al.*, 2004], stochastiques [Schenk et Schuëller, 2003], multiparamétriques [Gunzburger *et al.*, 2007], d'équations paraboliques [Atwell et King, 2001], dans le domaine des MEMS (microelectromechanical systems) [Hung et Senturia, 1999], pour l'analyse modale [Han et Feeny, 2003], de systèmes multicorps [Quaranta *et al.*, 2004], en météorologie et océanographie [Cao *et al.*, 2007] (voir [Kerschen *et al.*, 2005] pour une revue des applications), etc.

Considérons un problème défini sur l'espace Ω et paramétré par $\alpha \in \mathcal{A}$ ¹. Une méthode classique basée sur les éléments finis consisterait à résoudre ce problème pour chaque valeur de paramètre après discrétisation de \mathcal{A} . Si nous notons n , le nombre de degrés de liberté de Ω et $n_{\mathcal{A}}$, le nombre de nœuds du maillage de \mathcal{A} , cela revient à résoudre $n_{\mathcal{A}}$ fois un problème éléments finis de taille $n \times n$. Or les solutions obtenues sont issues de problèmes similaires et possèdent, en conséquence, des caractéristiques communes. On peut donc légitimement penser qu'une partie de l'information contenue dans chaque solution est redondante quand l'on prend le problème paramétré dans son ensemble. À partir de cette constatation, la Galerkin-POD suppose que l'espace de recherche de la solution est d'une dimension bien inférieure à celle de l'espace de recherche de la méthode classique, soit $n \times n_{\mathcal{A}}$. Le problème se résume alors en la recherche d'une base de cet espace.

Soit $u(\alpha, \mathbf{X})$ la solution recherchée du problème paramétré, avec $\alpha \in \mathcal{A}$ et $\mathbf{X} \in \Omega$. Dans la stratégie POD, on va construire une approximation sous la forme :

$$\mathbf{u}(\alpha, \mathbf{X}) = \sum_{i=1}^N \lambda_i(\alpha) \mathbf{\Lambda}_i(\mathbf{X}) \quad (1.16)$$

où N est la dimension de l'espace d'approximation.

1. On note en gras une variable α et son espace \mathcal{A} associé lorsque cette variable est multidimensionnelle. Lorsque cette variable α est scalaire, son espace associé \mathcal{A} et elle-même sont écrits avec une épaisseur de typographie normale.

La base $\mathbf{\Lambda}_{i \in [1, N]}$ construite (lors de la phase *offline*, voir ci-après), est, dans le cadre de la POD, une base orthogonale. Dans le contexte de la réduction de modèle, les fonctions $\lambda_i(\alpha)$ seront calculées dans un second temps lors de la phase dite *online*. Or c'est bien la première étape, la construction de la base qui est la plus problématique. En effet, la construction de la base nécessite une connaissance *a priori* de la solution. La résolution du problème complet n'étant pas envisageable, il est néanmoins possible de résoudre le problème pour des valeurs données de paramètre afin d'obtenir une connaissance partielle de la solution. Pour cela, on discrétise grossièrement l'espace des paramètres \mathcal{A} et on calcule des solutions $\mathbf{u}(\alpha_i, \mathbf{X})_{i \in [1, n_S]}$ partielles, appelées échantillons ou *snapshots* [Chatterjee, 2000]. On peut ensuite construire notre base en appliquant une des stratégies à disposition (SVD, KLT, PCA) à la matrice des *snapshots* $\mathbb{M}_S = [\mathbf{u}(\alpha_i, \mathbf{X})]_{i \in [1, n_S]}$. On notera, par la suite, cette base $(\mathbf{\Lambda}_1, \dots, \mathbf{\Lambda}_N)$ avec $N \leq n_S$. Cette première phase est celle de la construction du modèle réduit. Il reste maintenant à l'utiliser.

L'utilisation du modèle réduit consiste en la détermination à l'aide de la base construite des coefficients $(\lambda_i)_{i \in [1, N]}$ pour tout nouveau problème associé à un paramètre $\alpha \in \mathcal{A}$. On rappelle l'expression de l'approximation de $\mathbf{u}(\alpha)$ recherchée :

$$\mathbf{u}(\alpha) = \sum_{i=1}^N \lambda_i(\alpha) \mathbf{\Lambda}_i \quad (1.17)$$

Pour déterminer $(\lambda_i)_{i \in [1, N]}$, on projette $\mathbf{u}(\alpha)$ sur l'espace $\mathcal{W}_N = \text{Vect}\{\mathbf{\Lambda}_i\}_{i \in [1, N]}$. Supposons que l'on ait pris une série de snapshots $\mathbf{u}(\alpha_i)_{i \in [1, n_S]}$ et qu'à partir de cela, l'on ait extrait une base réduite $(\mathbf{\Lambda}_1, \dots, \mathbf{\Lambda}_N)$ (avec $N \leq n_S \ll n$). Il nous faut maintenant résoudre le problème suivant :

Problème 1.5. Pour $\alpha \in \mathcal{A}$, trouver $\mathbf{u}(\alpha) \in \mathcal{W}_N$ tel que

$$\forall v \in \mathcal{W}_N, \quad \alpha(\mathbf{u}(\alpha), v; \alpha) = \ell(v; \alpha) \quad (1.18)$$

On choisit alors comme fonction test $v = \mathbf{\Lambda}_k$ pour tout $k \in [1, N]$ ce qui revient à calculer les coefficients $(\lambda_1, \dots, \lambda_N)$ en résolvant le système linéaire suivant :

$$\forall j \in [1, N], \quad \sum_{k=1}^N \lambda_k(\alpha) \alpha(\mathbf{\Lambda}_j, \mathbf{\Lambda}_k) = \ell(\mathbf{\Lambda}_j; \alpha) \quad (1.19)$$

On est donc passé de la résolution d'un système $n \times n$ à celle d'un système $N \times N$ avec $N \ll n$. Les coûts de calcul ont donc été fortement réduits et cela autorise une utilisation quasi-temps réel du modèle réduit.

La stratégie Galerkin-POD peut être résumée comme suit :

Phase offline Durant cette étape, qualifiée aussi de *phase d'apprentissage*, on va extraire une série de *snapshots* $\mathbf{u}(\alpha_i)_{i \in [1, n_s]}$ du problème total. À partir de ces *snapshots*, on construit une base réduite $(\mathbf{\Lambda}_1, \dots, \mathbf{\Lambda}_N)$ à l'aide, par exemple d'une SVD. On peut, ensuite, calculer les termes $\alpha(\mathbf{\Lambda}_j, \mathbf{\Lambda}_k)$, indépendants des paramètres, une fois pour toutes. La phase *offline* est, de très loin, la plus coûteuse en temps de calcul. Cependant, le temps perdu à cette étape sera très largement compensé par celui gagné lors de la phase *online*, i.e. lors de l'utilisation du modèle réduit.

Phase online Alors que la phase *offline* est celle de la construction du modèle réduit à proprement dit, la phase *online* est celle de son utilisation. On vient résoudre, pour tout nouveau problème associé à un paramètre $\alpha \in \mathcal{A}$, les coefficients $(\lambda_i)_{i \in [1, N]}$ à la volée en se basant sur une projection de type Galerkin.

Il faut néanmoins préciser que le choix de la base réduite et donc des *snapshots* est déterminant pour la suite de la méthode. En effet, rien ne garantit *a priori* qu'une quantité suffisante d'information soit présente dans les échantillons sélectionnés. Une idée, pour assurer la validité et la légitimité de la méthode, pourrait être d'augmenter le nombre de *snapshots*. Or, pour certains problèmes hyperboliques de type problème transitoire, le nombre de *snapshots* nécessaires peut rendre la méthode inutilisable en pratique [Glüsmann et Kreuzer, 2009, Boucinha *et al.*, 2014]. En effet, augmenter le nombre de *snapshots* revient dans le même temps à augmenter la complexité de la méthode. Par conséquent, un nombre trop important de *snapshots* entraîne une explosion du temps de calcul. Une autre idée serait de pouvoir sélectionner de manière judicieuse les *snapshots* à l'aide d'estimateurs d'erreur. C'est précisément sur ce dernier point que se démarque la méthode Reduced Basis.

1.2 Reduced Basis

La démarche Reduced Basis (RB) est similaire à celle Galerkin-POD, à ceci près que l'accent est mis, à présent, sur la recherche des *snapshots* appropriés (voir [Maday et Ronquist, 2004, Veroy et Patera, 2005, Rozza *et al.*, 2008, Nguyen, 2008, Galvis et Kang, 2014]). Cette distinction est d'importance car la RB permet la construction de bases réduites de taille exploitable car, comme il a été mentionné, dans certains cas, la POD peut nécessiter un grand nombre de *snapshots* si leur choix n'a pas été fait de manière étudiée. Les *snapshots* sont recherchés par l'intermédiaire d'un algorithme *greedy* (glouton) et non donnés *a priori*. Ainsi supposons que l'on ait déjà choisi un ensemble de N paramètres $(\alpha_1, \dots, \alpha_N) \in \mathcal{A}^N$. L'approximation \mathbf{u}_N est construite en projetant orthogonalement $\mathbf{u}(\alpha)$ sur l'espace $\mathcal{W}_N = \text{Vect}\{\mathbf{u}(\alpha_i)\}_{i \in [1, N]}$:

$$\mathbf{u}_N(\alpha) = P_N \mathbf{u}(\alpha) \quad (1.20)$$

où P_N est la projection orthogonale sur \mathcal{W}_N .

Si l'on souhaite enrichir la base réduite, c'est-à-dire ajouter un nouveau *snapshot*, on le choisira de telle sorte que α_{N+1} , le paramètre associé à ce *snapshot*, minimise l'erreur $\|\mathbf{u}(\alpha) - \mathbf{u}_N(\alpha)\|$ [Rozza, 2011] :

$$\alpha_{N+1} = \arg \max_{\alpha \in \mathcal{I}} \|\mathbf{u}(\alpha) - \mathbf{u}_N(\alpha)\| \quad (1.21)$$

$(\mathbf{u}(\alpha_1), \dots, \mathbf{u}(\alpha_{N+1}))$, à savoir les *snapshots* eux-mêmes, sont choisis comme base réduite, après une orthonormalisation de type Gram-Schmidt. L'algorithme de sélection des *snapshots* est retranscrit dans l'ALG.1. Nous avons donc, ici aussi, une légère différence avec la méthode POD dans le sens où c'est bien $(\mathbf{u}(\alpha_1), \dots, \mathbf{u}(\alpha_{N+1}))$ qui est la base réduite bien. Autrement dit, les *snapshots* forment, eux-mêmes, la base, bien que l'on puisse toujours effectuer une SVD *a posteriori*.

Cette étape de sélection des *snapshots* se fait au prix d'une augmentation sensible du temps de calcul de la phase *offline*. Le calcul de l'erreur vraie $\|\mathbf{u}(\alpha) - \mathbf{u}_N(\alpha)\|$ est, certes, non accessible car impliquant la connaissance de la solution vraie. Cependant il est possible de majorer cette erreur *a posteriori* par une borne $\Delta_N(\alpha)$. C'est sur cette problématique qu'une grande partie de la littérature porte, à savoir la construction d'estimateurs d'erreur *a posteriori* afin de réduire le temps de calcul lié à la sélection des *snapshots* [Prud'homme *et al.*, 2002, Barrault *et al.*, 2004, Rozza *et al.*, 2008, Rozza, 2011].

Algorithme 1 : Algorithme greedy d'échantillonnage

Entrées : $S_1 = \{\alpha_1\}$
1 **pour** $k = 2$ à k_{max} **faire**
2 $\alpha_k = \arg \max_{\alpha \in \mathcal{I}} \|\mathbf{u}(\alpha) - \mathbf{u}_N(\alpha)\|;$
3 $S_k = S_{k-1} \cup \{\alpha_k\};$

1.3 PGD-Abaque virtuel

1.3.1 Proper Generalized Decomposition

La Proper Generalized Decomposition (PGD) diffère des techniques de réduction de modèle citées précédemment car elle ne nécessite pas de connaissance *a priori* de la solution réelle. La base réduite PGD est construite de manière itérative en même temps que l'approximation. Néanmoins, dans le cadre de la PGD, cette base n'est pas orthogonale contrairement à la POD, bien que l'on puisse l'orthonormaliser après coup (ce qui est très souvent fait en pratique). La sélection de snapshots n'étant pas nécessaire pour déterminer une base réduite, la méthode PGD permet le calcul d'une approximation de façon automatique.

Elle fut introduite pour la première fois dans [Ladevèze, 1985] sous le nom d'*Approximation Radiale* dans le cadre du solveur non incrémental LaTIn pour la résolution

de problèmes fortement non linéaires [Ladevèze, 1999]. Dans cet environnement, elle fut appliquée aux problèmes viscoplastiques [Boisse *et al.*, 1990, Cognard et Ladevèze, 1993, Relun *et al.*, 2013], au délaminage des composites [Allix et Ladevèze, 1992], aux problèmes paramétrés [Vitse *et al.*, 2014, Néron *et al.*, 2015], multiphysiques [Cognard *et al.*, 1999, Néron et Dureisseix, 2008a, Néron et Dureisseix, 2008b, Néron et Ladevèze, 2010], multiéchelles [Cremonesi *et al.*, 2013], pour les problèmes de contact [Giacoma *et al.*, 2014, Giacoma *et al.*, 2015], etc.

La PGD a aussi connu un développement important hors de la méthode LaTIn. Alors que la PGD traitait principalement de problèmes temps/espace (solutions de type $\mathbf{u}(t, \mathbf{X})$), elle va s'étendre à la résolution de problèmes paramétriques (solutions de type $\mathbf{u}(\alpha, \mathbf{X})$) grâce aux travaux des équipes de recherche de Chinesta [Mokdad *et al.*, 2007, Chinesta *et al.*, 2010a, González *et al.*, 2010, Bognet *et al.*, 2012a] pour n'en citer que quelques-uns. Elle a depuis été appliquée à différents types de domaines : problèmes stochastiques [Nouy, 2007, Nouy, 2009], multidimensionnels [Chinesta *et al.*, 2010b], hyperélastiques [Niroomandi *et al.*, 2013], paraboliques [Bonithon *et al.*, 2011], viscoélastiques [Ammar *et al.*, 2015], en acoustique [Barbarulo *et al.*, 2014], en mécanique des fluides [Ammar *et al.*, 2007a, Ammar *et al.*, 2007b], en *monitoring* de procédés thermiques [Aguado *et al.*, 2015], en mécanique de la rupture [Giner *et al.*, 2013], pour des simulations temps réel en chirurgie [González *et al.*, 2015, Alfaro *et al.*, 2014], en dynamique transitoire [Boucinha *et al.*, 2013, Boucinha *et al.*, 2014] ou encore pour la vérification de modèle et estimation d'erreur [Ladevèze et Chamoin, 2011, Ladevèze et Chamoin, 2013]. Pour une synthèse de la méthode, on peut se reporter à [Ladevèze, 2014].

1.3.1.1 PGD-Abaque virtuel

La Proper Generalized Decomposition, en permettant le calcul rapide et automatique d'approximations à variables séparées, est le cadre privilégié de la construction d'abaques virtuels. En effet, la résolution du problème paramétrique complet étant accessible, les modes PGD calculés peuvent être stockés, une fois pour toutes, dans des abaques pour une utilisation ultérieure. Ainsi, un abaque virtuel du déplacement \mathbf{u} dépendant d'un jeu de paramètres $\alpha \in \mathcal{A}$ peut être vu comme une approximation à variables séparées de \mathbf{u} .

$$\forall \mathbf{X} \in \Omega, \forall \alpha = (\alpha_1, \dots, \alpha_m) \in \mathcal{A}, \quad \mathbf{u}(\alpha, \mathbf{X}) = \sum_{k=1}^n \lambda_k(\alpha) \mathbf{\Lambda}_k(\mathbf{X}) \quad (1.22)$$

où $\lambda_k(\alpha) = \prod_{j=1}^m \lambda_k^j(\alpha_j)$ et n est le nombre de « modes » utilisés dans l'approximation.

En pratique, ce seront les fonctions λ_k et les éléments de la base réduite $\mathbf{\Lambda}_k$ que l'on stockera dans un abaque virtuel. L'utilisateur final n'a alors qu'à particulariser cette solution pour un jeu de paramètres donné en procédant à une simple somme de produits de fonctions à la volée.

1.3.1.2 Présentation du problème

La PGD est illustrée à travers le Problème 1.4 linéaire élastique où la loi constitutive est paramétrée par un ensemble de modules d'Young $\alpha = (\alpha_1, \dots, \alpha_m) \in \mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_m$.

On peut écrire la formulation variationnelle totale du problème, i.e. le problème est non seulement intégré sur l'espace Ω mais aussi sur les paramètres. Pour cela, on introduit les espaces fonctionnels suivants $\mathcal{V} = \mathbf{H}_0^1(\Omega) = \{v \in \mathbf{H}^1(\Omega) \mid v|_{\partial u \Omega} = \mathbf{0}\}$, $\mathcal{F} = L^2(\mathcal{A})$ et $\mathcal{F}_j = L^2(\mathcal{A}_j)$ pour $j = \llbracket 1, m \rrbracket$. Par conséquent, le problème s'écrit maintenant :

Problème 1.6. Trouver $u \in \mathcal{F} \otimes \mathcal{V}$ tel que

$$\forall v \in \mathcal{F} \otimes \mathcal{V}, \quad \int_{\mathcal{A}} \int_{\Omega} \mathbb{C} \varepsilon(u) : \varepsilon(v) \, d\Omega \, d\mathcal{A} = \int_{\mathcal{A}} \int_{\partial_d \Omega} \mathbf{f} \cdot v \, dS \, d\mathcal{A}$$

où \otimes est le produit tensoriel.

On recherche une approximation PGD du déplacement u . Deux possibilités s'offrent à nous. Soit une décomposition type paramètres/espace :

$$u(\alpha, \mathbf{X}) \approx u_n(\alpha, \mathbf{X}) = \sum_{i=1}^n \lambda_i(\alpha) \Lambda_i(\mathbf{X}) \quad (1.23)$$

avec pour tout $i \in \llbracket 1, n \rrbracket$, $(\lambda_i, \Lambda_i) \in \mathcal{F} \times \mathcal{V}$ où $\lambda_i(\alpha) = \lambda_i(\alpha_1, \dots, \alpha_m)$.

soit une décomposition selon chaque composante du paramètre α :

$$u_n(\alpha, \mathbf{X}) = \sum_{i=1}^n \prod_{j=1}^m \lambda_i^j(\alpha_j) \Lambda_i(\mathbf{X}) \quad (1.24)$$

avec pour tout $(i, j) \in \llbracket 1, n \rrbracket \times \llbracket 1, m \rrbracket$, $\lambda_i^j \in \mathcal{F}_j$.

Les différents modes sont calculés via un algorithme itératif de type Galerkin progressif [Chinesta *et al.*, 2011]. À l'étape d'enrichissement n , on suppose connue la représentation à variables séparées de u_n . Le nouveau $(m+1)$ -uplet $(\lambda^1, \dots, \lambda^m, \Lambda) \in \mathcal{F}_1 \times \dots \times \mathcal{F}_m \times \mathcal{V}$ est alors calculé à l'étape $n+1$ de la manière suivante :

$$u_{n+1} = u_n + \prod_{r=1}^m \lambda^r \Lambda \quad (1.25)$$

On choisit la fonction test $v \in T_{u_{n+1}}(\mathcal{F} \otimes \mathcal{V})$ (espace tangent à l'espace $\mathcal{F} \otimes \mathcal{V}$ en u_{n+1}) :

$$v = \prod_{r=1}^m \lambda^r \Lambda^* + \sum_{j=1}^m \lambda^{j*} \prod_{\substack{r=1 \\ r \neq j}}^m \lambda^r \Lambda \quad (1.26)$$

avec $\Lambda^* \in \mathcal{V}$ et $\lambda^{j*} \in \mathcal{F}_j$ pour $j \in \llbracket 1, m \rrbracket$.

Le Problème 1.6 devient :

Problème 1.7. Trouver $(\lambda^1, \dots, \lambda^m, \Lambda) \in \mathcal{F}_1 \times \dots \times \mathcal{F}_m \times \mathcal{V}$ tels que

$$\forall (\lambda^{1*}, \dots, \lambda^{m*}, \Lambda^*) \in \mathcal{F}_1 \times \dots \times \mathcal{F}_m \times \mathcal{V},$$

$$\int_{\mathcal{A}} \int_{\Omega} \mathbb{C}(\alpha) \nabla_s \left(\mathbf{u}_n + \prod_{r=1}^m \lambda^r \Lambda \right) : \nabla_s \left(\prod_{r=1}^m \lambda^r \Lambda^* + \sum_{j=1}^m \lambda^{j*} \prod_{\substack{r=1 \\ r \neq j}}^m \lambda^r \Lambda \right) d\Omega d\mathcal{A} =$$

$$\int_{\mathcal{A}} \int_{\partial_d \Omega} \mathbf{f} \cdot \left(\prod_{r=1}^m \lambda^r \Lambda^* + \sum_{j=1}^m \lambda^{j*} \prod_{\substack{r=1 \\ r \neq j}}^m \lambda^r \Lambda \right) dS d\mathcal{A}$$

Les calculs des différentes intégrales seraient grandement facilités par la représentation à variables séparées de l'intégrande puisqu'ils pourraient être effectués indépendamment l'un de l'autre, à la condition, naturellement, que l'opérateur de Hooke $\mathbb{C}(\alpha)$ puisse aussi être écrit sous forme séparée. Cela sera explicité à travers l'exemple numérique présenté dans la sous-section suivante. On remarque que le problème n'est plus linéaire à la vue de la fonction test v choisie (voir (1.26)). Une stratégie type algorithme de point fixe est, alors, adoptée afin de calculer le $(m+1)$ -uplet $(\lambda^1, \dots, \lambda^m, \Lambda)$. Les problèmes paramétriques, ainsi que le problème en espace, sont résolus alternativement. L'algorithme Greedy-PGD est détaillé dans l'ALG.2. On peut se référer à [Nouy, 2010, Allier *et al.*, 2015] pour des stratégies de résolution PGD alternatives.

Algorithme 2 : Algorithme Greedy-PGD

Entrées : Une séquence de paramètres $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m) \in \mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_m$

1 **Level 1 : Algorithme Greedy;**

2 **tant que** $\|\mathbf{R}(\mathbf{u}_n)\| > \text{critère}$ (avec $\mathbf{R}(\mathbf{u}_n)$ étant le résidu) **faire**

3 **Level 2 : Algorithme de point-fixe;**

4 Initialisation du $(m+1)$ -uplet $(\lambda^1, \dots, \lambda^m, \Lambda)$;

5 **pour** $k = 1$ à k_{max} **faire**

6 **pour** $j = 1$ à m **faire**

7 Résoudre le problème paramétrique associé à λ^j

8 Résoudre le problème spatial associé à Λ

9 $\mathbf{u}_n \leftarrow \mathbf{u}_{n-1} + \prod_{r=1}^m \lambda^r \Lambda$;

10 $n \leftarrow n + 1$;

1.3.2 Exemple numérique

1.3.2.1 Formulation du problème à résoudre

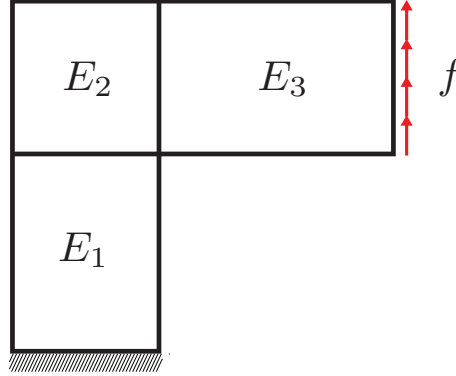


FIGURE 1.5 – La structure test est partitionnée en trois sous-domaines, le module d’Young de chaque sous-domaine étant considéré comme un paramètre.

Illustrons la démarche décrite précédemment par un exemple numérique simple. La géométrie considérée est une structure en forme de Γ (voir Figure 1.5), encastrée en son pied et soumise à un chargement de 1000 N dirigé vers le haut en son autre extrémité. La structure Ω est partitionnée en trois sous-domaines $\Omega_1, \Omega_2, \Omega_3$ et le module d’Young de chaque sous-domaine, considéré comme un paramètre, peut varier de 50 GPa à 1000 GPa ($\alpha = (E_1, E_2, E_3) \in \mathcal{A} = [50 \text{ GPa}; 1000 \text{ GPa}]^3$). Le coefficient de Poisson est, quant à lui, homogène et égal à 0,3.

Nous allons, maintenant, nous attacher à résoudre le Problème 1.7. Pour cela, il est judicieux d’écrire la matrice de Hooke sous forme séparée afin d’obtenir une intégrande entièrement à variables séparées, comme mentionné précédemment. Une possibilité est de décomposer la matrice de Hooke de la sorte :

$$\mathbb{C}(E_1, E_2, E_3) = (E_1 \Phi_1 + E_2 \Phi_2 + E_3 \Phi_3) \mathbb{H} \quad (1.27)$$

avec

$$\blacksquare \Phi_i(\mathbf{X}) = 1 \text{ si } \mathbf{X} \in \Omega_i \text{ sinon } \Phi_i(\mathbf{X}) = 0 \text{ pour } i \in \llbracket 1, 3 \rrbracket$$

$$\blacksquare \mathbb{H} = \begin{bmatrix} \frac{1-\nu}{(1+\nu)(1-2\nu)} & \frac{\nu}{(1+\nu)(1-2\nu)} & 0 \\ \frac{\nu}{(1+\nu)(1-2\nu)} & \frac{1-\nu}{(1+\nu)(1-2\nu)} & 0 \\ 0 & 0 & \frac{1}{2(1+\nu)} \end{bmatrix}$$

Cette formulation nous donne une intégrande sous forme entièrement séparée, ce qui nous sera utile lors du calcul des intégrales. On peut, désormais, réécrire le Problème 1.7 particularisé à notre exemple. Le champ de déplacement \mathbf{u} est recherché

sous la forme :

$$\mathbf{u}(\boldsymbol{\alpha}, \mathbf{X}) = \mathbf{u}(E_1, E_2, E_3, \mathbf{X}) = \sum_{i=1}^n \lambda_i^1(E_1) \lambda_i^2(E_2) \lambda_i^3(E_3) \boldsymbol{\Lambda}_i(\mathbf{X}) \quad (1.28)$$

La fonction test \mathbf{v} s'écrit quant à elle :

$$\begin{aligned} \mathbf{v} = & \lambda^{1*}(E_1) \lambda^2(E_2) \lambda^3(E_3) \boldsymbol{\Lambda} + \lambda^1(E_1) \lambda^{2*}(E_2) \lambda^3(E_3) \boldsymbol{\Lambda} + \\ & \lambda^1(E_1) \lambda^2(E_2) \lambda^{3*}(E_3) \boldsymbol{\Lambda} + \lambda^1(E_1) \lambda^2(E_2) \lambda^3(E_3) \boldsymbol{\Lambda}^* \end{aligned} \quad (1.29)$$

On introduit (1.28) et (1.29) dans le Problème 1.7. On définit, par souci de clarté, les coefficients suivants pour $i \in \llbracket 1, n+1 \rrbracket$ et $r \in \llbracket 1, 3 \rrbracket$:

$$\begin{aligned} a_i^r &= \int_{\mathcal{A}_i} \lambda_i^r \lambda^r d\mathcal{A}_r, \quad b_i^r = \int_{\mathcal{A}_i} \lambda_i^r \lambda^r E_r d\mathcal{A}_r, \quad c_r = \int_{\mathcal{A}_r} \lambda^r d\mathcal{A}_r \\ G_i^r &= \int_{\Omega_r} \mathbb{H} \nabla_s(\boldsymbol{\Lambda}_i) : \nabla_s(\boldsymbol{\Lambda}) d\Omega_r, \quad G_i^{r*} = \int_{\Omega_r} \mathbb{H} \nabla_s(\boldsymbol{\Lambda}_i) : \nabla_s(\boldsymbol{\Lambda}^*) d\Omega_r \end{aligned} \quad (1.30)$$

En appliquant le principe de stationnarité, on arrive aux quatre équations suivantes :

$$\begin{aligned} (a_{n+1}^2 a_{n+1}^3 G_{n+1}^1 E_1 + b_{n+1}^2 a_{n+1}^3 G_{n+1}^2 + a_{n+1}^2 b_{n+1}^3 G_{n+1}^3) \lambda^1 = \\ c_2 c_3 \int_{\partial_d \Omega} \mathbf{f} \cdot \boldsymbol{\Lambda} dS - \sum_{i=1}^n (a_i^2 a_i^3 G_i^1 E_1 + b_i^2 a_i^3 G_i^2 + a_i^2 b_i^3 G_i^3) \lambda_i^1 \end{aligned} \quad (1.31)$$

$$\begin{aligned} (b_{n+1}^1 a_{n+1}^3 G_{n+1}^1 + a_{n+1}^1 a_{n+1}^3 G_{n+1}^2 E_2 + a_{n+1}^1 b_{n+1}^3 G_{n+1}^3) \lambda^2 = \\ c_1 c_3 \int_{\partial_d \Omega} \mathbf{f} \cdot \boldsymbol{\Lambda} dS - \sum_{i=1}^n (b_i^1 a_i^3 G_i^1 + a_i^1 a_i^3 G_i^2 E_2 + a_i^1 b_i^3 G_i^3) \lambda_i^2 \end{aligned} \quad (1.32)$$

$$\begin{aligned} (b_{n+1}^1 a_{n+1}^2 G_{n+1}^1 + a_{n+1}^1 b_{n+1}^2 G_{n+1}^2 + a_{n+1}^1 a_{n+1}^2 G_{n+1}^3 E_3) \lambda^3 = \\ c_1 c_2 \int_{\partial_d \Omega} \mathbf{f} \cdot \boldsymbol{\Lambda} dS - \sum_{i=1}^n (b_i^1 a_i^2 G_i^1 + a_i^1 b_i^2 G_i^2 + a_i^1 a_i^2 G_i^3 E_3) \lambda_i^3 \end{aligned} \quad (1.33)$$

$$\begin{aligned} (b_{n+1}^1 a_{n+1}^2 a_{n+1}^3 G_{n+1}^{1*} + a_{n+1}^1 b_{n+1}^2 a_{n+1}^3 G_{n+1}^{2*} + a_{n+1}^1 a_{n+1}^2 b_{n+1}^3 G_{n+1}^{3*}) = \\ c_1 c_2 c_3 \int_{\partial_d \Omega} \mathbf{f} \cdot \boldsymbol{\Lambda}^* dS - \sum_{i=1}^n (b_i^1 a_i^2 a_i^3 G_i^{1*} + a_i^1 b_i^2 a_i^3 G_i^{2*} + a_i^1 a_i^2 b_i^3 G_i^{3*}) \end{aligned} \quad (1.34)$$

Les trois premières équations relatives aux fonctions paramétriques sont scalaires. Quant à la dernière, relative au mode spatial, elle peut être résolue en utilisant la méthode numérique de son choix (éléments finis, différences finies...). Dans notre cas, nous avons choisi d'utiliser les éléments finis. Dans ce cadre, il suffit, alors, de définir les matrices \mathbb{K}_1 , \mathbb{K}_2 et \mathbb{K}_3 , les matrices de rigidité associées respectivement aux matrices de Hooke $\Phi_1\mathbb{H}$, $\Phi_2\mathbb{H}$ et $\Phi_3\mathbb{H}$. Ainsi, l'équation (1.34) associée au problème spatial devient :

$$(b_{n+1}^1 a_{n+1}^2 a_{n+1}^3 \mathbb{K}_1 + a_{n+1}^1 b_{n+1}^2 a_{n+1}^3 \mathbb{K}_2 + a_{n+1}^1 a_{n+1}^2 b_{n+1}^3 \mathbb{K}_3) \mathbf{\Lambda} = c_1 c_2 c_3 \mathbf{F} - \sum_{i=1}^n (b_i^1 a_i^2 a_i^3 \mathbb{K}_1 + a_i^1 b_i^2 a_i^3 \mathbb{K}_2 + a_i^1 a_i^2 b_i^3 \mathbb{K}_3) \mathbf{\Lambda}_i \quad (1.35)$$

où \mathbf{F} est le vecteur de chargement.

Le problème étant formulé correctement, on applique ensuite l'ALG.2 présenté précédemment.

1.3.3 Résultats

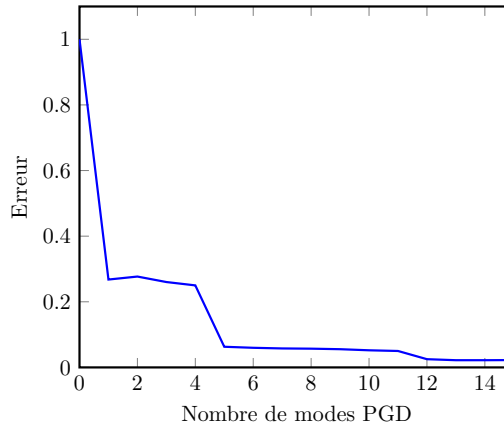


FIGURE 1.6 – Courbe de convergence de l'erreur par rapport à une solution de référence éléments finis.

Les modes, i.e. les fonctions λ_i^1 , λ_i^2 , λ_i^3 et $\mathbf{\Lambda}_i$, sont calculés grâce à l'ALG.2. Les quatre premiers modes associés aux modules d'Young E_1 , E_2 et E_3 sont représentés Figure 1.7 et les quatre premiers modes spatiaux sont représentés Figure 1.8. La Figure 1.7(a) montre la prépondérance des premiers modes sur les autres et tout particulièrement du premier. En effet, on s'aperçoit que l'amplitude de la fonction λ^1 devient d'autant plus faible que l'ordre du mode est élevé. Ainsi, il apparaît clairement que la plus grande partie de l'information est contenue dans les premiers modes et qu'augmenter le degré de l'approximation revient à raffiner cette dernière localement.

On remarque, sur la Figure 1.6, la décroissance rapide de l'erreur par rapport à une solution de référence calculée à partir d'une résolution éléments finis pour tous les jeux de paramètres. L'erreur est de l'ordre de 5% pour seulement 5 modes et de l'ordre de 1% pour 20 modes.

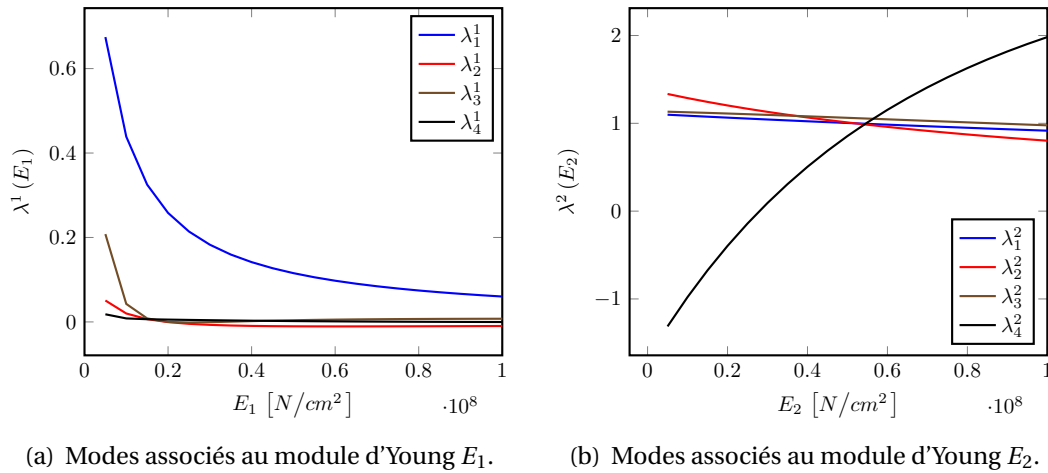
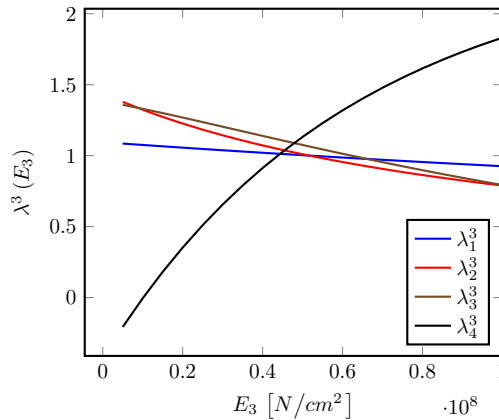
(a) Modes associés au module d'Young E_1 .(b) Modes associés au module d'Young E_2 .(c) Modes associés au module d'Young E_3 .

FIGURE 1.7 – Premiers modes associés à chaque module d'Young E_1 , E_2 et E_3 considérés comme des paramètres.

1.3.4 Phase d'update

Durant la présentation de l'algorithme PGD, il n'a pas été fait de distinction entre la résolution des problèmes paramétriques et de celui en espace. Or il y a une asymétrie flagrante entre leur résolution, le problème spatial étant nettement plus lourd. En effet, le nombre de degrés de liberté, lié à la discrétisation en espace, est, en pratique, beaucoup plus élevé que le nombre de paramètres, lié à la discrétisation de \mathcal{A}_j pour

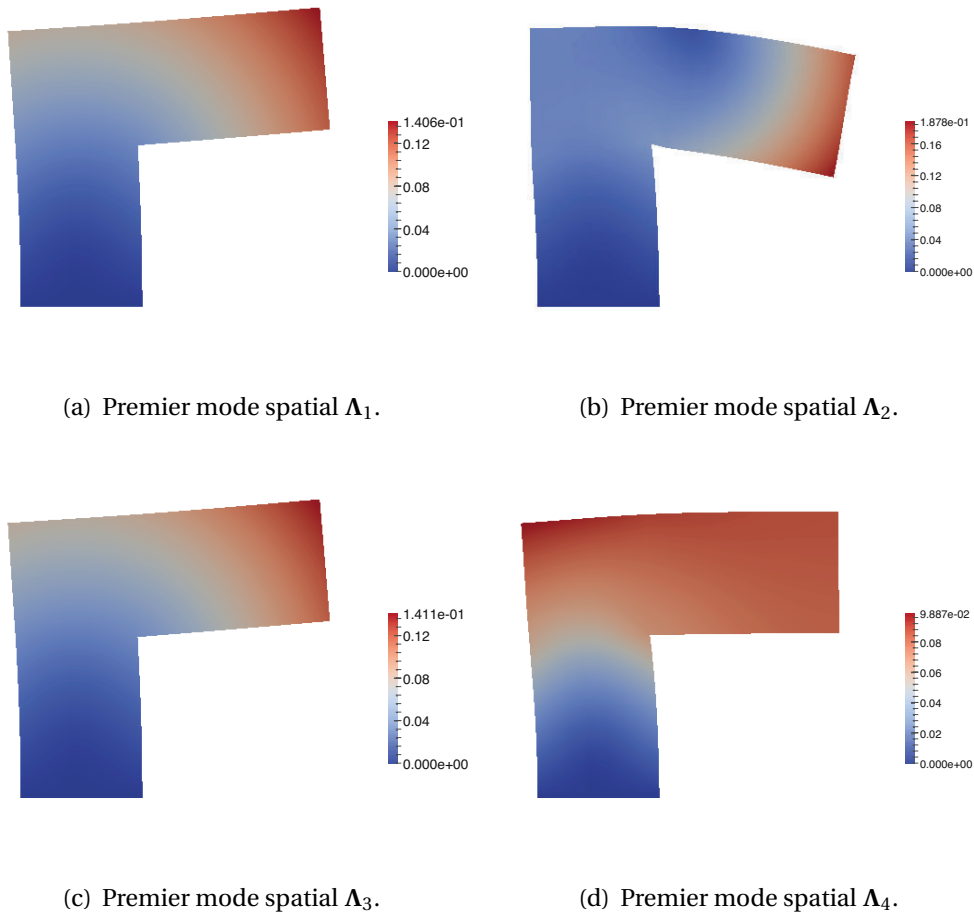


FIGURE 1.8 – Premiers modes spatiaux.

$j \in \llbracket 1, m \rrbracket$. Il serait donc opportun de calculer le moins possible d'éléments de la base spatiale quitte à devoir calculer un plus grand nombre de fonctions paramétriques.

La démarche adoptée [Cognard et Ladevèze, 1993, Allier *et al.*, 2015] est donc de stopper l'algorithme de point fixe après quelques itérations (typiquement 3 – 4), de conserver le mode spatial calculé, d'orthonormaliser la base spatiale en construction afin, dirons-nous, de la nettoyer et enfin de venir recalculer en bloc toutes les fonctions paramétriques lors de la phase appelée *Update*. L'algorithme PGD modifié est décrit ALG.3.

Dans le cas où seul un paramètre α est en jeu, l'update est trivial. Il suffit, après avoir orthonormalisé la base spatiale à l'itération n à l'aide de l'algorithme de Gram-Schmidt par exemple, de résoudre le système suivant :

$$([\Lambda_1, \dots, \Lambda_n]^T \mathbb{K}(\alpha) [\Lambda_1, \dots, \Lambda_n]) \begin{bmatrix} \lambda_1(\alpha) \\ \vdots \\ \lambda_n(\alpha) \end{bmatrix} = \begin{bmatrix} \Lambda_1^T \\ \vdots \\ \Lambda_n^T \end{bmatrix} \mathbf{F} \quad (1.36)$$

Algorithme 3 : Algorithme Greedy-PGD avec update

Entrées : Une séquence de paramètres $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m) \in \mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_m$

- 1 **Level 1 : Algorithme Greedy;**
- 2 **tant que** $\|\mathbf{R}(\mathbf{u}_n)\| > \text{critère}$ (avec $\mathbf{R}(\mathbf{u}_n)$ étant le résidu) **faire**
- 3 **Level 2 : Algorithme de point fixe;**
- 4 Initialisation due $(m+1)$ -uplet $(\lambda^1, \dots, \lambda^m, \Lambda)$;
- 5 **pour** $k = 1$ à k_{max} **faire**
- 6 **pour** $j = 1$ à m **faire**
- 7 Résoudre le problème paramétrique associé à λ^j
- 8 Résoudre le problème spatial associé à Λ
- 9 **Update;**
- 10 Actualisation des fonctions $(\lambda_i^1, \dots, \lambda_i^m)_{i \in \{1..n\}}$;
- 11 $\mathbf{u}_n \leftarrow \mathbf{u}_{n-1} + \prod_{r=1}^m \lambda^r \Lambda$;
- 12 $n \leftarrow n + 1$;

Néanmoins lorsque le nombre de paramètres augmente, cela devient plus complexe et la phase d'update n'est plus si immédiate. Il faut alors avoir recours à un algorithme de type point fixe pour recalculer les fonctions paramétriques.

1.4 Cas des paramètres géométriques

Les paramètres géométriques sont de nature différente des autres types de paramètres tels, par exemple, les paramètres matériau (modules d'Young, coefficients de Poisson...) car leurs variations affectent directement le domaine de définition Ω_α ¹ de la structure de telle sorte que le problème linéaire élastique présenté SEC.1.3.1.2 devient :

■ Équations d'équilibre

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma} = \mathbf{0} & \text{dans } \Omega_\alpha \times \mathcal{A} \\ \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{f} & \text{sur } \partial_d \Omega_\alpha \times \mathcal{A} \end{cases} \quad (1.37)$$

■ Relation de comportement

$$\boldsymbol{\sigma} = \mathbb{C} \boldsymbol{\varepsilon} \text{ dans } \Omega_\alpha \times \mathcal{A} \quad (1.38)$$

$$= \mathbb{C} \nabla_s(\mathbf{u}) \text{ dans } \Omega_\alpha \times \mathcal{A} \quad (1.39)$$

■ Conditions aux limites homogènes

$$\mathbf{u} = \mathbf{0} \text{ sur } \partial_u \Omega_\alpha \times \mathcal{A} \quad (1.40)$$

1. Ω_α correspond à la géométrie associée au jeu de paramètres α .

où $\alpha = (\alpha_1, \dots, \alpha_m) \in \mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_m$ est, ici, un ensemble de paramètres définissant la géométrie de la structure.

Dans ce contexte, la formulation variationnelle du problème s'écrit :

Problème 1.8. *Pour tout $\alpha \in \mathcal{A}$, trouver $u \in \mathcal{V}(\alpha)$ tel que*

$$\forall v \in \mathcal{V}(\alpha), \quad \int_{\Omega_\alpha} \mathbb{C} \varepsilon(u) : \varepsilon(v) \, d\Omega = \int_{\partial_d \Omega_\alpha} \mathbf{f} \cdot v \, dS$$

où $\mathcal{V}(\alpha) = \mathbf{H}_0^1(\Omega_\alpha) = \{v \in \mathbf{H}^1(\Omega_\alpha) \mid v|_{\partial_u \Omega_\alpha} = \mathbf{0}\}$.

On s'aperçoit que pour chaque jeu de paramètres, un nouveau problème est à résoudre pour un domaine de définition différent. Une possibilité repose sur le choix d'une structure de référence Ω_0 et d'une transformation géométrique \mathcal{T}_α liant cette structure à celle courante Ω_α afin de ramener toutes les configurations sur le même domaine de définition. Il s'agit, en d'autres termes, d'effectuer un changement de variables :

Problème 1.9. *Trouver $u \in \mathcal{V}_0$ tel que*

$$\forall v \in \mathcal{V}_0, \quad \int_{\Omega_0} \sigma(u(\alpha, \mathcal{T}_\alpha(\mathbf{X}_0))) : \varepsilon(v(\alpha, \mathcal{T}_\alpha(\mathbf{X}_0))) J_{\mathcal{T}_\alpha}(\mathbf{X}_0) \, d\Omega_0 = \int_{\partial_d \Omega_0} \mathbf{f}(\alpha, \mathcal{T}_\alpha(\mathbf{X}_0)) \cdot v(\alpha, \mathcal{T}_\alpha(\mathbf{X}_0)) J_{\mathcal{T}_\alpha}(\mathbf{X}_0) \, dS_0$$

avec $\mathcal{V}_0 = \mathbf{H}_0^1(\Omega_0) = \{v \in \mathbf{H}^1(\Omega_0) \mid v|_{\partial_u \Omega_0} = \mathbf{0}\}$ et $J_{\mathcal{T}_\alpha}$ est le module du déterminant de la matrice Jacobienne de la transformation géométrique \mathcal{T}_α .

De cette sorte, la dépendance en les paramètres est transférée du domaine d'intégration au Jacobien. C'est l'option choisie par [Lassila et Rozza, 2010, Rozza, 2011, Manzoni *et al.*, 2012a] dans le cadre des Reduced Basis et par [Ammar *et al.*, 2014, Zlotnik *et al.*, 2015] dans celui de la PGD. Pour ces deux méthodes, la même problématique revient sur le choix de la paramétrisation \mathcal{T}_α de la géométrie car, une mauvaise paramétrisation peut entraîner une augmentation importante du temps de résolution du problème. En effet, en prenant une formulation Galerkin du problème, c'est-à-dire :

Problème 1.10. Pour $\alpha \in \mathcal{A}$, trouver $\mathbf{u}(\alpha) \in \mathcal{W}_n$ tel que

$$\forall \mathbf{v} \in \mathcal{W}_n, \quad a(\mathbf{u}(\alpha), \mathbf{v}; \alpha) = \ell(\mathbf{v}; \alpha) \quad (1.41)$$

On montre [Lassila et Rozza, 2010] que, pour avoir des temps de calcul raisonnables, $a(\square, \square; \alpha)$ et $\ell(\square; \alpha)$ doivent être paramétrés affinement. Autrement dit, que l'on puisse les formuler de la sorte :

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}; \alpha) &= \sum_{k=1}^{M_a} \Theta_a^k(\alpha) a^k(\mathbf{u}, \mathbf{v}) \\ \ell(\mathbf{v}; \alpha) &= \sum_{k=1}^{M_l} \Theta_l^k(\alpha) \ell^k(\mathbf{v}) \end{aligned} \quad (1.42)$$

Cette séparation de variables permet de calculer efficacement les différentes intégrales présentes dans la formulation faible du problème et donc de réduire notablement le temps de calcul. L'obtention de transformations géométriques affines est abordée de manières diverses. On peut directement choisir des transformations affines en faisant en sorte de partitionner la géométrie de référence en sous-domaines, chaque sous-domaine subissant une transformation affine. En choisissant des triangles comme sous-domaines, on garantit que la transformation est affine [Ammar *et al.*, 2014, Zlotnik *et al.*, 2015]. Néanmoins cette solution n'est pas la panacée car nous verrons que, pour certaines géométries complexes type splines, un grand nombre de triangles et donc de sous-domaines sont nécessaires pour représenter fidèlement le contour, entraînant une augmentation du temps de calcul. D'autres choix de transformations sont possibles et seront exposés SEC.3.1.3.

On peut aussi construire un *mapping* grâce à des méthodes telles que la *Free-Form Deformations* (FFD) [Sederberg et Parry, 1986, Lassila et Rozza, 2010] ou la *Radial Basis Functions* (RBF) [Buhmann, 2000, Manzoni *et al.*, 2012a] et de recourir ensuite à la *Empirical Interpolation Method* (EIM, très présente dans la communauté RB) [Barrault *et al.*, 2004] afin de décomposer le mapping construit en somme de mappings affines. À la suite de cela, les différentes méthodes suivent leur stratégie habituelle exposée plus haut.

Dans [Raghavan *et al.*, 2013], une voie différente est empruntée puisque, dans cet article qui traite d'optimisation de forme, la PCA est utilisée pour décomposer les snapshots générés, à partir de géométries admissibles dans un voisinage du point d'évaluation, afin d'obtenir une base réduite $(\Lambda_1, \dots, \Lambda_m)$. Dans un second temps, les coefficients $(\lambda_1, \dots, \lambda_m)$, calculés en projetant les snapshots sur la base réduite, sont analysés afin de déterminer la dimensionnalité réelle du problème. Car, en effet, suivant le choix de paramétrisation fait, le nombre de paramètres représentant la géométrie peut ne pas être optimal.

L'exemple donné est celui d'une ellipse paramétrée par huit points de contrôle (soit 16 paramètres) alors que l'on sait *a priori* que seuls deux paramètres sont nécessaires

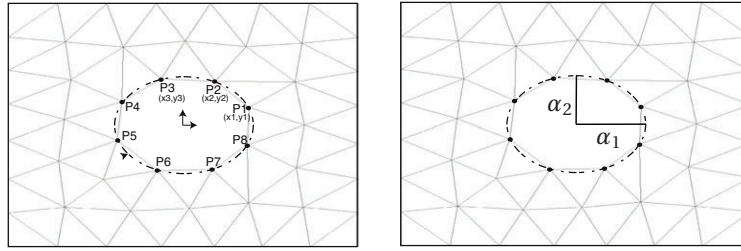


FIGURE 1.9 – L'ellipse est paramétrée par 8 points de contrôle (gauche) équivalents à 16 paramètres alors que seuls 2 paramètres (demi-petit axe et demi-grand axe) sont nécessaires (droite). Cet exemple est tiré de [Raghavan *et al.*, 2013].

pour représenter une ellipse (ses demi-grand axe et demi-petit axe, voir Figure 1.9). Une étude des coefficients $(\lambda_1, \dots, \lambda_m)$ permet alors de réduire la dimensionnalité du problème de 16 à 2. Cette méthode permet, dans un sens, de découvrir la véritable dimensionnalité du problème qui peut, dans certains cas de géométries complexes, être cachée par les paramètres de conception.

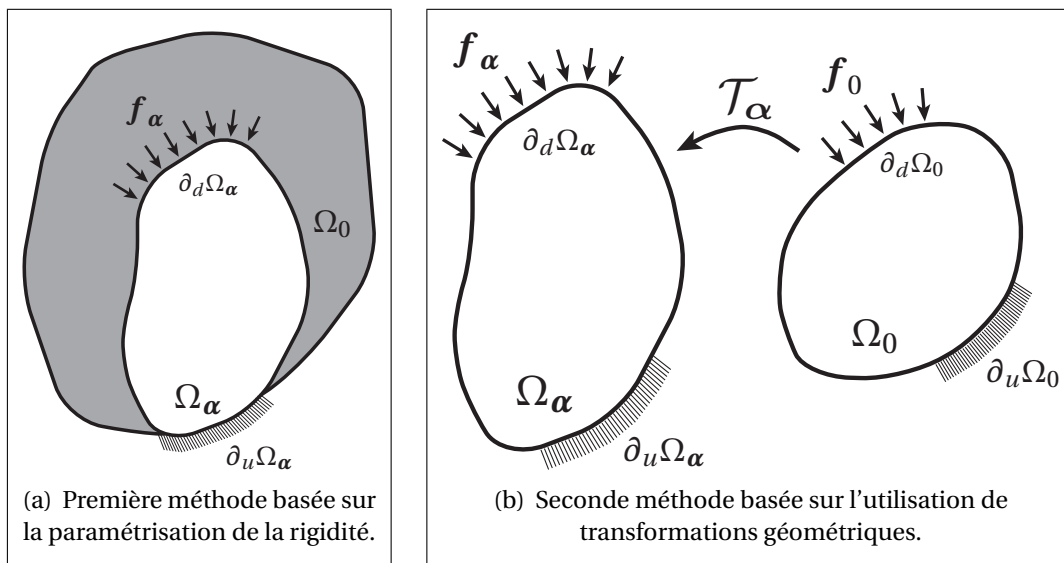


FIGURE 1.10 – Méthodes développées lors de la thèse afin de prendre en compte les variations de géométrie dans la méthode PGD.

Dans les chapitres suivants, on s'intéressera tout particulièrement à la méthode PGD et deux approches différentes seront abordées afin de traiter des variations de géométrie. La première sera basée sur la paramétrisation de la rigidité d'un domaine Ω_0 englobant toutes les configurations géométriques possibles. Ainsi quand on se situera dans la structure courante étudiée, la rigidité sera celle classique alors que lorsque l'on

se situera à l'extérieur, elle sera très faible. Ω_0 est composé, pour ainsi dire, d'une partie *dure* et d'une partie *molle* (voir Figure 1.10(a)). Dans la seconde méthode, une structure, parmi les configurations géométriques possibles, sera choisie comme configuration de référence et la définition du problème sera ramenée à cette structure à l'aide de transformations géométriques (voir Figure 1.10(b)).

Chapitre 2

Paramétrisation de la rigidité et maillages universels

Une première approche est proposée afin de prendre en compte les variations de géométrie dans la méthode PGD. Cette technique est basée sur la paramétrisation de la rigidité d'un domaine Ω_0 englobant toutes les géométries possibles. Ce domaine est alors composé d'une partie molle et d'une partie dure, dans le sens où, lorsque l'on se situe dans la structure courante, la rigidité est celle classique et quand on se trouve en dehors, elle devient très faible. Cette méthode est testée sur un exemple numérique. Dans un second temps, les maillages universels sont présentés et une extension de la technique est proposée afin de l'adapter à la PGD. Il est montré comment la paramétrisation de la rigidité combinée aux maillages universels définit un cadre pertinent à la PGD géométrique.

2.1 Paramétrisation de la rigidité

2.1.1 Une nouvelle relation de comportement

Nous avons présenté, dans le chapitre précédent, la PGD au travers du Problème 1.6 dans le cadre de paramètres matériau. Il s'agit maintenant d'étendre l'approche PGD au cas des variations de géométrie. Pour bien avoir à l'esprit ce que changent les paramètres géométriques, réécrivons le Problème 1.6 :

Problème 2.1. Trouver $u \in \mathcal{F} \otimes \mathcal{V}(\alpha)$ tel que

$$\forall v \in \mathcal{F} \otimes \mathcal{V}(\alpha), \quad \int_{\mathcal{A}} \int_{\Omega_\alpha} \mathbb{C} \varepsilon(u) : \varepsilon(v) \, d\Omega \, d\mathcal{A} = \int_{\mathcal{A}} \int_{\partial_d \Omega_\alpha} f \cdot v \, dS \, d\mathcal{A}$$

Comme nous avons pu le dire, une approche consiste à choisir une structure de référence et une transformation géométrique afin de définir le problème sur un domaine unique. Cette méthodologie sera détaillée dans le chapitre suivant. Ce chapitre, quant à lui, présente une voie différente. Dans cette première approche, on introduit une structure de référence Ω_0 mais ce choix est fait de telle sorte que sa géométrie englobe toutes les configurations possibles (voir Figure 2.1).

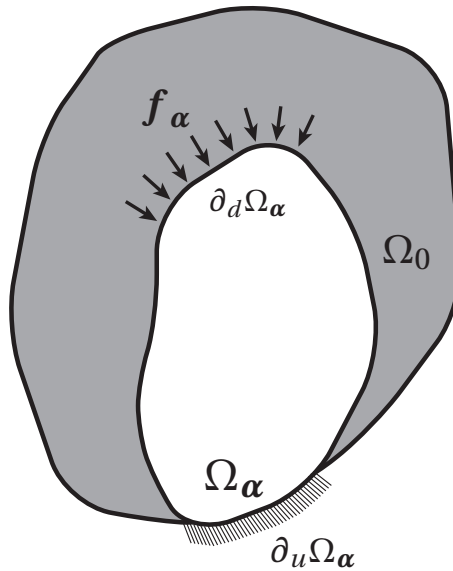


FIGURE 2.1 – La structure courante $\Omega(\alpha)$ est immergée dans une structure de référence Ω_0 englobant toutes les configurations géométriques possibles.

Le problème sera alors défini sur Ω_0 et il restera à déterminer si l'on se trouve dans la structure courante ou en dehors lors des différents calculs, ce qui n'est pas immédiat *a priori*.

Dans cette optique, on décide de paramétrer la rigidité de la structure Ω_0 . Quand on se trouve dans la structure courante associée au paramètre α , la rigidité est la rigidité normale associée aux paramètres du matériau mais quand l'on se situe hors de la structure courante, la rigidité est alors choisie très faible. Pour fixer les idées, on peut imaginer la structure de référence Ω_0 comme étant composée d'une partie *dure* Ω_α et d'une partie *molle* $\Omega_0 \setminus \Omega_\alpha$. Cela revient à définir sur Ω_0 une nouvelle matrice de Hooke \mathbb{C}_α :

$$\mathbb{C}_\alpha = \mathbb{C}\psi_\alpha + a\mathbb{C}(1 - \psi_\alpha) \text{ avec } a \ll 1 \quad (2.1)$$

où \mathbb{C} est la matrice de Hooke classique.

Le coefficient a paramètre la rigidité de la partie molle $\Omega_0 \setminus \Omega_\alpha$. En variant sa valeur, cette rigidité devient plus ou moins faible. Quant à la fonction ψ_α , il s'agit de la fonction indicatrice de Ω_α sur Ω_0 qui détermine si l'on se trouve dans la structure courante ou non :

$$\begin{cases} \psi_\alpha(\mathbf{X}) &= 1 \text{ si } \mathbf{X} \in \Omega_\alpha \\ &= 0 \text{ sinon} \end{cases} \quad (2.2)$$

La dépendance en les paramètres géométriques a été, dans ce cas, transférée du domaine d'intégration à la fonction ψ_α . La nouvelle relation de comportement étant définie, il convient à présent de formuler le nouveau problème variationnel associé :

Problème 2.2. Trouver $u \in \mathcal{F} \otimes \mathcal{V}_0$ tel que

$$\forall v \in \mathcal{F} \otimes \mathcal{V}_0, \quad \int \int_{\mathcal{A} \Omega_0} \mathbb{C}_\alpha \nabla_s(u) : \nabla_s(v) d\Omega_0 d\mathcal{A} = \int \int_{\mathcal{A} \partial_d \Omega_\alpha} \mathbf{f} \cdot \mathbf{v} dS_0 d\mathcal{A}$$

où $\mathcal{V}_0 = \mathbf{H}_0^1(\Omega_0) = \{v \in \mathbf{H}^1(\Omega_0) \mid v|_{\partial_u \Omega_0} = \mathbf{0}\}$.

Ce type d'approche est très classique dans le domaine des méthodes numériques et a, notamment, été utilisé dans la *Finite Cell Method* (voir [Schillinger et Rues, 2014] pour une revue détaillée) ou encore dans la méthode des *level sets* [Osher et Sethian, 1988] utilisée, en particulier, dans la technique XFEM [Belytschko et Black, 1999, Moës *et al.*, 1999, Sukumar *et al.*, 2001] et en optimisation de forme [Allaire *et al.*, 2002, Allaire *et al.*, 2004]. Néanmoins, son application dans le cadre de la PGD est, quant à elle, nouvelle.

2.1.2 Calcul de la matrice de rigidité

Afin de résoudre le Problème 2.2, on discrétise Ω_0 et on utilise une méthodologie éléments finis. La matrice de rigidité d'un élément du maillage de Ω_0 s'écrit alors :

$$\tilde{\mathbb{K}}^e = \int_{\Omega_0^e} \mathbb{B}^T \mathbb{C}_\alpha \mathbb{B} d\Omega_0 \quad (2.3)$$

$$= \int_{\Omega_0^e} \mathbb{B}^T \mathbb{C} \mathbb{B} \psi_\alpha d\Omega + a \int_{\Omega_0^e} \mathbb{B}^T \mathbb{C} \mathbb{B} (1 - \psi_\alpha) d\Omega_0 \quad (2.4)$$

$$= (1 - a) \int_{\Omega_0^e} \mathbb{B}^T \mathbb{C} \mathbb{B} \psi_\alpha d\Omega_0 + a \int_{\Omega_0^e} \mathbb{B}^T \mathbb{C} \mathbb{B} d\Omega_0 \quad (2.5)$$

où \mathbb{B} est la matrice des dérivées partielles des fonctions de forme.

Une difficulté apparaît car la frontière $\partial\Omega_\alpha$ de la structure courante peut couper l'élément et il peut être lourd et coûteux en termes de calcul de quantifier les contributions des parties *dure* et *molle*. On propose donc de diviser chaque volume d'élément Ω_0^e en deux parties disjointes Ω_+^e et Ω_-^e , correspondant respectivement aux volumes de la partie comprise dans la structure courante et de celle en dehors (voir Figure 2.2).

$$\Omega_0^e = \Omega_+^e \cup \Omega_-^e \quad (2.6)$$

où \cup représente l'union disjointe.

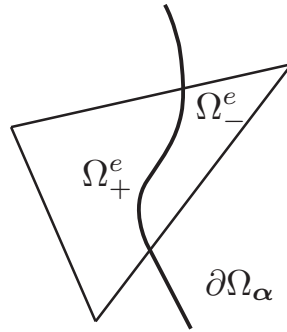


FIGURE 2.2 – La frontière $\partial\Omega_\alpha$ coupe l'élément de volume Ω_0^e en deux parties disjointes Ω_+^e et Ω_-^e .

Ainsi, plus particulièrement, lorsque l'élément est entièrement immergé dans la structure courante :

$$\Omega_0^e = \Omega_+^e \quad (2.7)$$

et si, au contraire, il est entièrement immergé dans la partie *molle* :

$$\Omega_0^e = \Omega_-^e \quad (2.8)$$

La matrice de rigidité élémentaire s'écrit, donc, en optant pour ces notations :

$$\tilde{\mathbb{K}}^e = \int_{\Omega_+^e} \mathbb{B}^T \mathbb{C} \mathbb{B} d\Omega_0 + a \int_{\Omega_-^e} \mathbb{B}^T \mathbb{C} \mathbb{B} d\Omega_0 \quad (2.9)$$

Toujours dans l'optique de déterminer efficacement, en termes de temps de calcul, la matrice de rigidité, nous allons introduire l'approximation suivante :

$$\int_{\Omega_+^e} \mathbb{B}^T \mathbb{C} \mathbb{B} d\Omega_0 \approx \frac{\Omega_+^e}{\Omega_0^e} \int_{\Omega_0^e} \mathbb{B}^T \mathbb{C} \mathbb{B} d\Omega_0 \quad (2.10)$$

où $\int_{\Omega_0^e} \mathbb{B}^T \mathbb{C} \mathbb{B} d\Omega_0$ est la matrice de rigidité qu'aurait l'élément de volume Ω_0^e s'il était complètement immergé dans la structure courante. On appellera cette matrice par la suite \mathbb{K}_0^e .

En remarquant que,

$$\Omega_-^e = \Omega_0^e \setminus \Omega_+^e \quad (2.11)$$

on peut déduire :

$$\int_{\Omega_-^e} \mathbb{B}^T \mathbb{C} \mathbb{B} d\Omega_0 = \int_{\Omega_0^e} \mathbb{B}^T \mathbb{C} \mathbb{B} d\Omega_0 - \int_{\Omega_+^e} \mathbb{B}^T \mathbb{C} \mathbb{B} d\Omega_0 \quad (2.12)$$

$$= \left(1 - \frac{\Omega_+^e}{\Omega_0^e}\right) \mathbb{K}_0^e \quad (2.13)$$

On obtient donc l'expression de la matrice de rigidité élémentaire suivante :

$$\tilde{\mathbb{K}}^e = \left[\frac{\Omega_+^e}{\Omega_0^e} + \left(1 - \frac{\Omega_+^e}{\Omega_0^e}\right) a \right] \mathbb{K}_0^e \quad (2.14)$$

Remarque. La légitimité de l'approximation (2.10) provient du fait que, dans le cas de l'élément CST (Constant Strain Triangle ou triangle à trois nœuds), elle correspondrait à l'expression exacte de la matrice de rigidité. En effet, dans ce cas précis, puisque $\mathbb{B}^T \mathbb{C} \mathbb{B}$ est constant sur Ω_0^e :

$$\int_{\Omega_+^e} \mathbb{B}^T \mathbb{C} \mathbb{B} d\Omega_0 = \Omega_+^e \mathbb{B}^T \mathbb{C} \mathbb{B} \quad (2.15)$$

$$= \frac{\Omega_+^e}{\Omega_0^e} \underbrace{\Omega_0^e \mathbb{B}^T \mathbb{C} \mathbb{B}}_{\mathbb{K}_0^e} \quad (2.16)$$

ce qui donne le résultat souhaité.

Maintenant que l'on possède une expression de la matrice de rigidité, ou tout du moins une approximation, permettant son calcul, il ne reste qu'à appliquer la méthodologie PGD présentée SEC.1.3.

2.1.3 Calcul de la contrainte

Dans le cadre des éléments finis, le calcul de la contrainte σ pour un élément se fait de la manière suivante :

$$\sigma(\mathbf{X}) = \mathbb{C}_\alpha(\mathbf{X}) \mathbb{B}(\mathbf{X}) \mathbf{u}(\mathbf{X}) \quad (2.17)$$

Or, en pratique, on détermine, très souvent, σ uniquement aux points de Gauss de l'élément, définis lors du calcul de la matrice de rigidité. σ est interpolée, dans un second temps, sur le reste de l'élément.

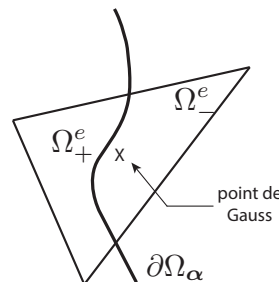


FIGURE 2.3 – Exemple d'un point de Gauss d'un élément CST se trouvant hors du domaine Ω_α . La contrainte calculée sera alors extrêmement faible puisque la rigidité de la partie Ω_+^e ne sera pas prise en compte.

Apparaît alors le souci suivant. Lorsqu'un élément est coupé par la frontière $\partial\Omega_\alpha$, il peut arriver, quand le nombre de points de Gauss est faible (pour l'élément CST un seul point de Gauss est nécessaire pour l'intégration), que tous ces points, ou au moins une grande partie, se situent à l'extérieur du domaine Ω_α (voir Figure 2.3).

Par conséquent, la contrainte calculée ne sera pas représentative de celle réelle car la contribution de la partie rigide de l'élément ne sera pas, ou peu, prise en compte. Pour éviter de sous-estimer la contrainte au niveau des éléments proches de la frontière $\partial\Omega_\alpha$, il est conseillé de choisir un nombre conséquent de points de Gauss. Durant nos simulations, nous avons choisi d'en prendre six.

2.1.4 Application numérique

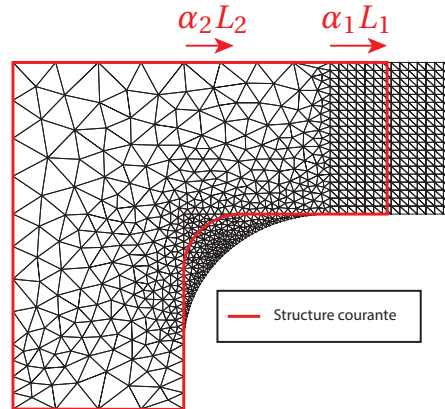


FIGURE 2.4 – Seule la structure de référence Ω_0 est maillée. La longueur du porte-à-faux et le congé sont paramétrés respectivement par α_1 et α_2

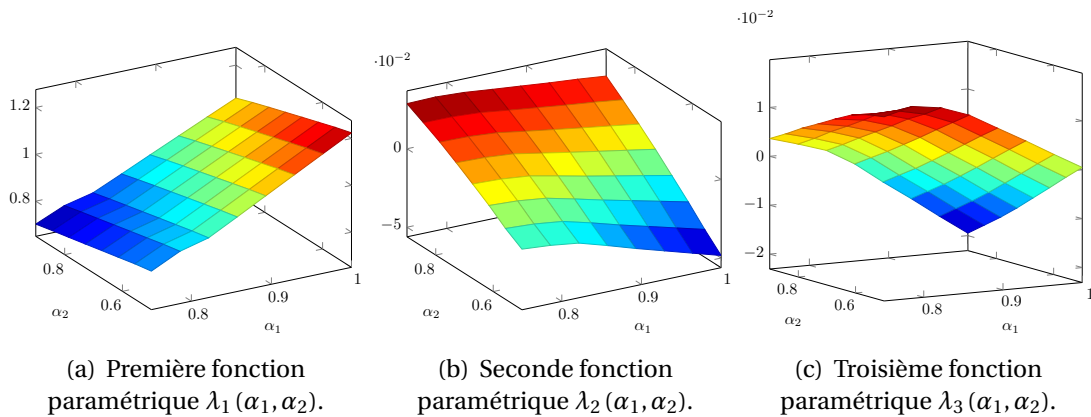


FIGURE 2.5 – Trois premières fonctions paramétriques.

Nous allons reprendre l'exemple exposé dans la SEC.1.3.2 et l'adapter légèrement afin de traiter un cas de variations de géométrie (voir Figure 2.4). Nous rajoutons ainsi un congé à la géométrie et nous paramétrons à la fois ce congé par $\alpha_2 \in \mathcal{A}_2 = [0,5; 0,9]$ et le porte-à-faux par $\alpha_1 \in \mathcal{A}_1 = [0,75; 1]$ ($\alpha = (\alpha_1, \alpha_2) \in \mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$). La structure de référence Ω_0 englobant toutes les configurations possibles est la configuration où $\alpha_1 = 1$ et $\alpha_2 = 1$ et seule cette structure sera maillée (voir Figure 2.4). Le problème est, quant à lui, toujours linéaire élastique. Le matériau est isotrope de module d'Young 200 GPa, de coefficient de Poisson 0,3 et le paramètre a choisi est de 10^{-6} (paramètre définissant la rigidité de la partie *molle*). La structure est encastée en son pied et un chargement de 1000 N est appliqué en son autre extrémité. Nous avons fait le choix de

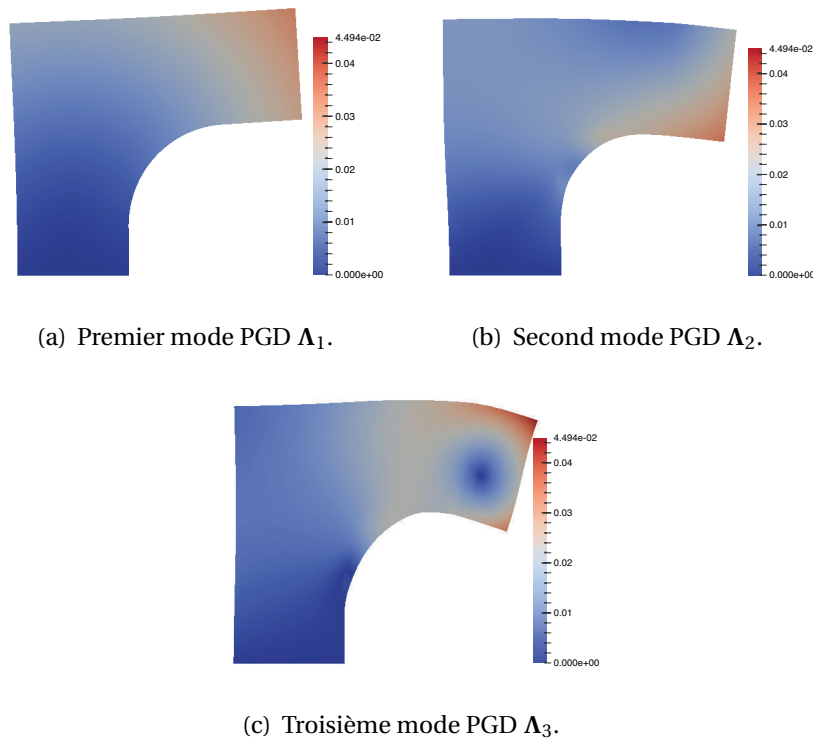


FIGURE 2.6 – Trois premiers modes PGD.

regrouper les paramètres géométriques au sein des mêmes fonctions λ_k , autrement dit le déplacement \mathbf{u} a été recherché sous la forme :

$$\mathbf{u}(\boldsymbol{\alpha}, \mathbf{X}) = \mathbf{u}(\alpha_1, \alpha_2, \mathbf{X}) = \sum_{k=1}^n \lambda_k(\alpha_1, \alpha_2) \Lambda_k(\mathbf{X}) \quad (2.18)$$

Les trois premières fonctions paramétriques λ_1 , λ_2 et λ_3 sont reportées Figure 2.5 et les trois premiers modes PGD Λ_1 , Λ_2 et Λ_3 Figure 2.6. La contrainte de Von Mises est tracée Figure 2.7 pour trois configurations géométriques (les deux configurations extrêmes 2.7(a) 2.7(c) et une configuration intermédiaire 2.7(b)). Dans chaque cas, la structure Ω_0 est représentée et la configuration géométrique considérée est indiquée par le contour rouge.

Toutefois, si l'on regarde de plus près le congé, on s'aperçoit que le contour n'est pas net. Ceci est dû au fait que la contrainte représentée dans PARAVIEW (utilisé pour la visualisation) est une contrainte moyennée sur l'élément, il n'est pas possible de définir la contrainte au niveau des points de Gauss. Les éléments coupés par la frontière Γ sont alors tracés complètement car leur suppression partielle n'est pas autorisée.

La convergence de la méthode a aussi été prise en compte et la décroissance du résidu est représentée sur la Figure 2.8. Pour une comparaison avec une solution de référence, on peut se reporter à ce qui sera fait pour la méthode suivante Figure 2.25.

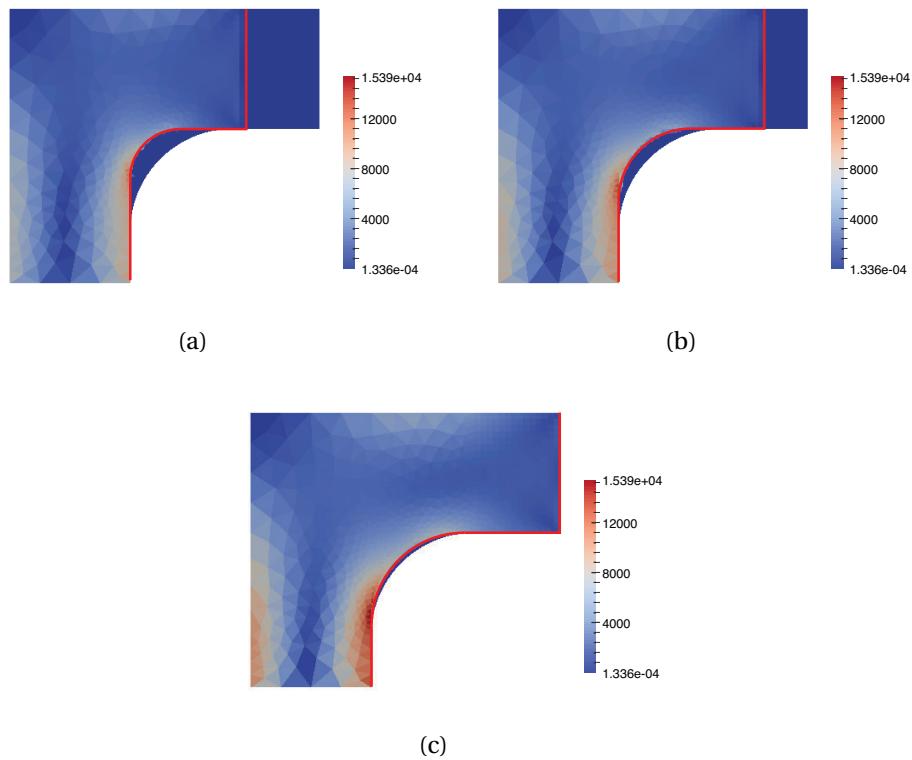


FIGURE 2.7 – Contrainte de Von Mises pour trois configurations géométriques différentes (25 modes PGD).

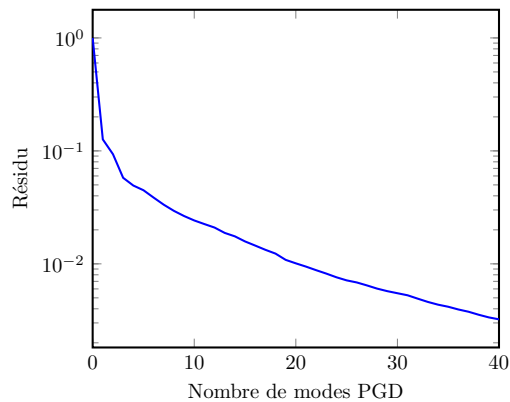


FIGURE 2.8 – Convergence du résidu.

2.1.5 Conclusions partielles

La paramétrisation donne un cadre intéressant pour l'introduction de paramètres géométriques dans la PGD et les premiers résultats obtenus sont encourageants. Néanmoins quelques bémols doivent être signalés. Si la surface d'application du chargement dépend du chargement (ce qui était le cas de l'exemple numérique), des difficul-

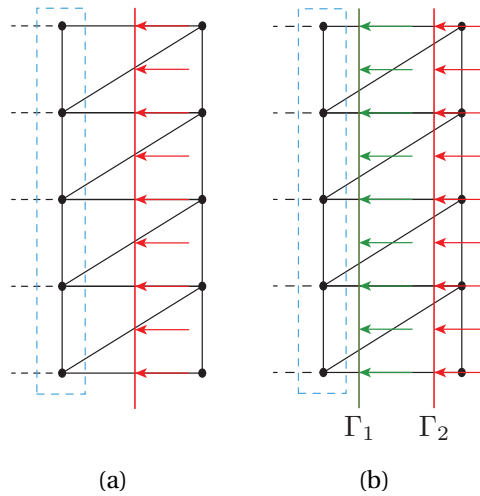


FIGURE 2.9 – Lorsque la surface d’application du chargement coupe un élément, on décide de choisir les points de Ω_α les plus proches comme points d’application (a). Les points d’application du chargement sont les mêmes pour les configurations 1 et 2 (b).

tés pour définir les points d’application apparaissent. Si l’on regarde la Figure 2.9(a) qui correspond à un zoom au niveau du chargement pour l’exemple numérique précédent, on remarque que le chargement ne peut être appliqué tel quel puisqu’aucun nœud ne s’y trouve.

On peut envisager d’ajouter des nœuds sur la frontière mais une augmentation du nombre des nœuds implique qu’un post-traitement devra être effectué pour que l’on puisse combiner des données calculées à partir de maillages différents (type projection). Nous avons, pour notre part, choisi une option plus simple qui est celle d’appliquer le chargement sur les nœuds de Ω_α les plus proches de la frontière Γ_α (nœuds encadrés de la Figure 2.9(a)). Cependant, cela signifie que plusieurs configurations géométriques peuvent avoir le même chargement alors que leur frontière est différente comme le montre la Figure 2.9(b).

Il a alors été tenté de rechercher une autre voie permettant la construction rapide de maillages conformes qui, dans le même temps, comporteraient le même nombre d’éléments et de nœuds, autrement dit une méthode de génération de maillages à partir d’un maillage de référence.

2.2 Maillages universels

2.2.1 Introduction

Les maillages universels [Rangarajan et Lew, 2012] reposent sur l’idée simple, qu’à partir d’un même maillage, on peut définir des maillages pour un grand nombre de

géométries différentes, toutes immergées dans ce *maillage universel*, en venant déplacer localement des nœuds pour les faire épouser le contour de ces géométries. Plus précisément, après avoir immergé un domaine Ω dans un maillage composé de triangles à trois nœuds, cela revient à sélectionner les éléments dont un, et un seul, nœud est contenu dans ce domaine (voir Figure 2.10) et à projeter les deux nœuds restant sur le contour Γ du domaine¹.

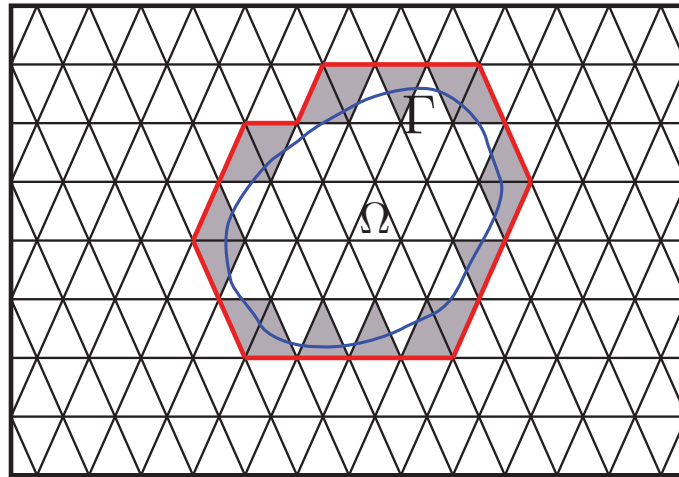


FIGURE 2.10 – Seuls les éléments ayant exactement un nœud à l’intérieur du domaine Ω (éléments grisés) sont sélectionnés.

La seconde étape de cette méthode consiste à venir *relaxer* les nœuds à la fois contenus dans le domaine Ω et dans un voisinage à définir de Γ , dans le souci de garantir la qualité du maillage final. Cette méthodologie peut être aussi employée pour mailler l’extérieur du domaine immergé si l’on veut, par exemple, modéliser l’écoulement d’un fluide autour d’un profil en mouvement telles des pales d’éolienne en rotation ou une aile d’avion plus ou moins inclinée.

Il est à noter qu’il n’est en rien requis que le maillage initial soit conforme, aucun nœud de ce maillage ne devant être *a priori* confondu dans le contour de la géométrie immergée. Les maillages universels ont donc été pensés à l’origine pour la résolution de problèmes dont le domaine évolue avec le temps tels que les problèmes avec des surfaces libres, transformations de phase, propagation de fissures [Rangarajan *et al.*, 2014], interaction fluide/structure [Rangarajan et Lew, 2012], etc. De premiers travaux ont été initiés sur l’extension des maillages universels du 2D au 3D sur des tétraèdres [Kabaria et Lew, 2014]. Nous resterons néanmoins dans notre étude au cas 2D. Commençons par décrire la technique des maillages universels telle qu’elle a été développée dans [Rangarajan, 2012] et nous verrons dans un second temps comment, nous avons du l’adapter afin de l’utiliser dans le cadre de la méthode PGD.

1. Pour la suite de la présentation des maillages universels, l’indice α sera omis pour alléger les notations. Ainsi, plutôt que d’envisager un domaine Ω_α de frontière $\partial\Omega_\alpha$, on parlera de domaine Ω et de frontière Γ .

2.2.2 Sélection des nœuds à projeter

Soit Ω un domaine régulier C^2 borné de \mathbb{R}^2 immergé dans un maillage composé uniquement de triangles à trois nœuds et Γ son contour. On définit une distance signée $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$ par rapport à Γ telle que :

$$\phi(\mathbf{X}) = \begin{cases} -\min_{\mathbf{Y} \in \Gamma} d(\mathbf{X}, \mathbf{Y}) & \text{si } \mathbf{X} \in \Omega \\ \min_{\mathbf{Y} \in \Gamma} d(\mathbf{X}, \mathbf{Y}) & \text{sinon} \end{cases} \quad (2.19)$$

où d est la distance euclidienne dans \mathbb{R}^2 .

Ainsi, si $\phi(\mathbf{X})$ est négatif, cela signifie que \mathbf{X} se situe à l'intérieur du domaine Ω et inversement lorsque $\phi(\mathbf{X})$ est positif, \mathbf{X} se situe à l'extérieur du domaine Ω . On remarque, du même coup, que $\Gamma = \phi^{-1}(\{0\})$. À l'aide de cette distance, nous pouvons définir les notions de *nœud positif* et d'*élément coupé positivement*. Ainsi un nœud \mathbf{n} d'un élément est dit *positif* lorsque $\phi(\mathbf{n})$ est positif. Un élément sera, quant à lui, coupé positivement si, exactement deux de ses nœuds sont positifs (voir Figure 2.11).

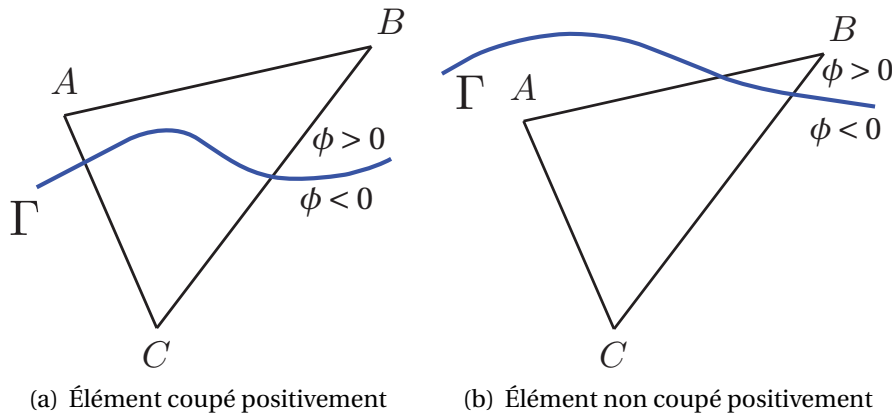


FIGURE 2.11 – Exemple d'un élément coupé positivement et d'un élément non coupé positivement.

La sélection des nœuds à projeter se déroule donc de la manière suivante. On vient calculer la distance $\phi(\mathbf{n})$ pour tout nœud \mathbf{n} du maillage et ensuite conserver que les éléments coupés positivement, c'est-à-dire n'ayant qu'un seul nœud négatif. Les deux autres nœuds seront alors projetés sur Γ .

2.2.3 Projection des nœuds sélectionnés sur Γ

Les nœuds positifs trouvés précédemment sont à projeter en le point le plus proche de Γ au sens de la distance euclidienne. Il reste à déterminer comment calculer explicitement la projection $\pi(\mathbf{X})$ sur Γ d'un point \mathbf{X} de coordonnées $(x_{\mathbf{X}}, y_{\mathbf{X}})$. Pour cela,

supposons que Γ ait une représentation paramétrique régulière, i.e. :

$$\Gamma = \{(x(\theta), y(\theta)) : x, y \in C^2(I, \mathbb{R}), \theta \in I \subset \mathbb{R}\} \quad (2.20)$$

Pour calculer $\pi(\mathbf{X})$, on minimise la distance euclidienne entre le point \mathbf{X} et un point de Γ afin de trouver $\theta_{\mathbf{X}}$ ([Rangarajan et Lew, 2011]). Ainsi, on aboutit à la résolution de l'équation suivante :

$$(x_{\mathbf{X}} - x(\theta_{\mathbf{X}})) x'(\theta_{\mathbf{X}}) + (y_{\mathbf{X}} - y(\theta_{\mathbf{X}})) y'(\theta_{\mathbf{X}}) = 0 \quad (2.21)$$

et ensuite, on en déduit l'expression de $\pi(\mathbf{X}) = (x(\theta_{\mathbf{X}}), y(\theta_{\mathbf{X}}))$.

Il est à noter que, si le point \mathbf{X} n'est pas suffisamment proche de Γ , la projection peut ne pas être unique. Dans la Figure 2.12 est présenté un exemple où le point \mathbf{X} a une infinité de projections possibles. Si nous avions considéré un point plus proche de Γ , la projection aurait, alors, été unique. Nous verrons au paragraphe 2.2.5 ce que cela implique comme condition sur le maillage.

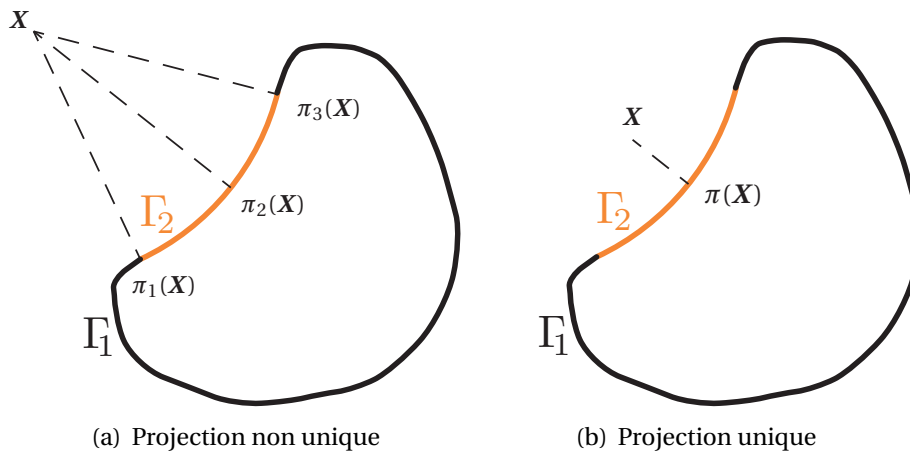


FIGURE 2.12 – Le contour Γ est la réunion d'une partie Γ_1 et d'un arc de cercle Γ_2 . Dans le cas (a), le point \mathbf{X} à projeter se situe au centre du cercle dont Γ_2 est un arc. Dans cette situation, le point \mathbf{X} a plusieurs projections possibles (et même une infinité pour être plus exact) alors que si le point \mathbf{X} avait été suffisamment proche du contour (cas (b)), sa projection aurait été unique.

2.2.4 Relaxation des nœuds dans le voisinage du contour Γ

Une fois que les nœuds positifs des éléments ont été projetés sur le contour Γ , un problème subsiste. En effet, du fait du déplacement de certains nœuds du maillage initial, le nouveau maillage peut contenir des éléments de piètre qualité. C'est pourquoi

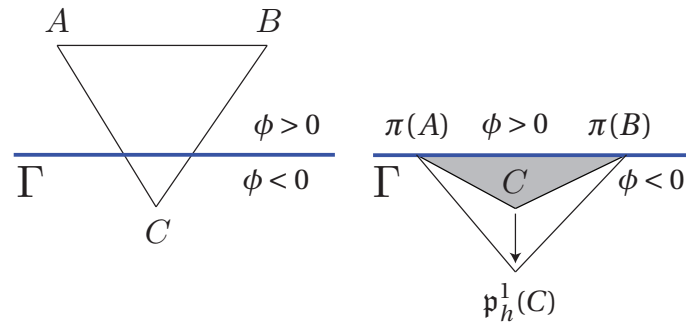


FIGURE 2.13 – La relaxation des nœuds proches du contour Γ assure la bonne qualité des nouveaux éléments.

l'algorithme des maillages universels comprend une étape de relaxation des nœuds dans le voisinage du contour.

Ainsi, comme montré dans la Figure 2.13, la relaxation du nœud C , i.e. son éloignement du contour Γ , améliore la qualité de l'élément obtenu à la fin du processus (élément transparent) par rapport à celui obtenu sans relaxation (élément grisé).

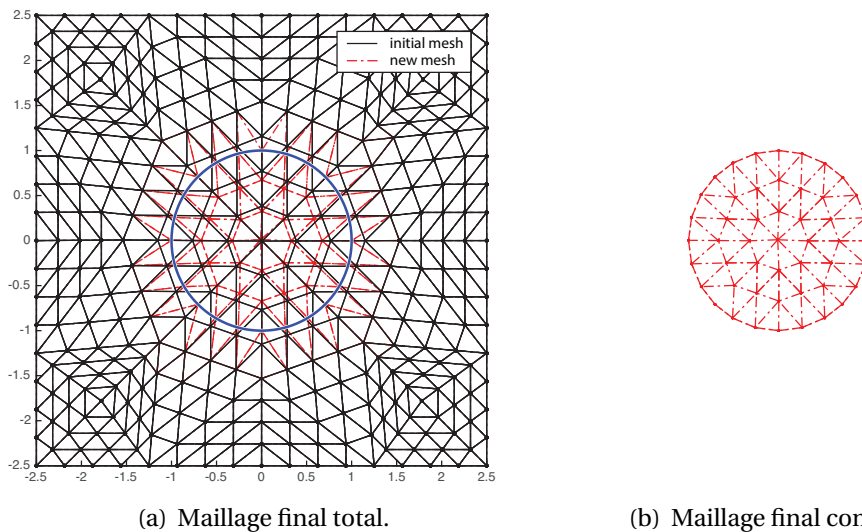


FIGURE 2.14 – Exemple de construction d'un maillage pour le cercle unité à partir d'un maillage universel (paramètres utilisés pour la relaxation : $r = 3h$ et $\eta = 0.3$). Seule la partie du nouveau maillage comprise à l'intérieur du cercle est conservée.

Nous allons, maintenant, nous attacher à décrire comment cette étape de relaxation se déroule en pratique. Commençons par définir la fonction de relaxation $p_h^1 : \mathbb{R}^2 \rightarrow \mathbb{R}$:

$$p_h^1(\mathbf{X}) = \begin{cases} \mathbf{X} - \eta_1 h \left(1 + \frac{\phi(\mathbf{X})}{r_1}\right) \nabla \phi(\mathbf{X}) & \text{si } -r_1 < \phi(\mathbf{X}) < 0 \\ \mathbf{X} & \text{sinon} \end{cases} \quad (2.22)$$

où η_1 est un paramètre compris entre 0 et 1 (il s'agit du paramètre de relaxation à proprement dit) et r_1 est de l'ordre de quelques multiples de la taille du maillage h (ce paramètre définit la taille du voisinage où s'effectue la relaxation).

Autrement dit, les nœuds, situés à l'intérieur du domaine Ω et dans le voisinage du contour Γ défini par la distance r_1 , sont éloignés de Γ avec une atténuation de cet effet plus le nœud est distant du contour. La Figure 2.14 montre un exemple d'application de l'algorithme des maillages universels au cercle unité, immergé dans un maillage du carré $[-2,5;2,5]^2$. On ne conserve, à la fin du processus, que la partie du maillage universel se situant à l'intérieur du cercle unité.

2.2.5 Conditions suffisantes

Nous avons vu Section 2.2.3 un exemple où la projection d'un point X sur le contour Γ par π n'était pas unique car X n'était pas suffisamment proche de Γ . Nous allons énoncer deux conditions *suffisantes* garantissant l'injectivité de π .

Angle de conditionnement aigu : Nous allons ici introduire une nouvelle notion, celle d'angle de conditionnement (*conditioning angle*). Dans un élément coupé positivement, l'angle de conditionnement est l'angle associé au nœud extérieur à Ω le plus proche de Γ (voir Figure 2.15). Dans le cas où deux nœuds sont à la même distance de Γ , l'angle de conditionnement sera celui le plus faible. Si jamais les deux angles sont égaux alors l'angle de conditionnement pourra être choisi arbitrairement.

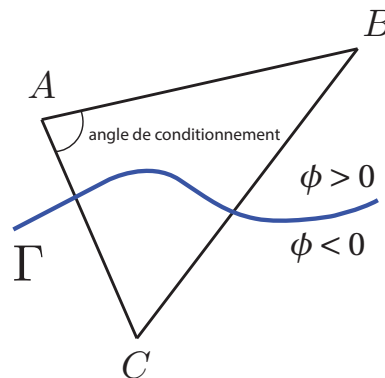


FIGURE 2.15 – Exemple d'angle de conditionnement d'un élément coupé positivement.

La première condition suffisante porte précisément sur l'angle de conditionnement. En effet, si l'angle de conditionnement n'est pas aigu ($\geq 90^\circ$) alors l'injectivité de la projection π peut ne pas être garantie. La Figure 2.16 expose un exemple de deux éléments coupés positivement, les triangles ABC et ACD dont l'angle de conditionnement au nœud C est obtus. Cela a, pour conséquence, la superposition des images de A et de C par la projection π et la réduction de l'image du segment $[AC]$ à un point.

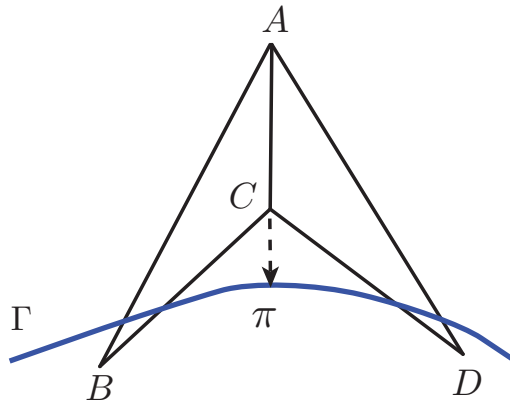


FIGURE 2.16 – Les éléments ABC et ADC sont coupés positivement et ont chacun un angle de conditionnement obtus. On s'aperçoit que, dans ce cas, la projection π n'est pas injective car A et C ont la même image.

Maillage fin : Nous allons voir que définir un maillage dont tous les éléments coupés positivement possèdent un angle de conditionnement aigu ne suffit pas à garantir l'injectivité de π . Nous pouvons remarquer sur la Figure 2.17 que π n'est pas injective sur le côté $[AB]$ car plusieurs points de ce segment possèdent la même image. On s'aperçoit qu'en remaillant plus finement ce problème disparaît.

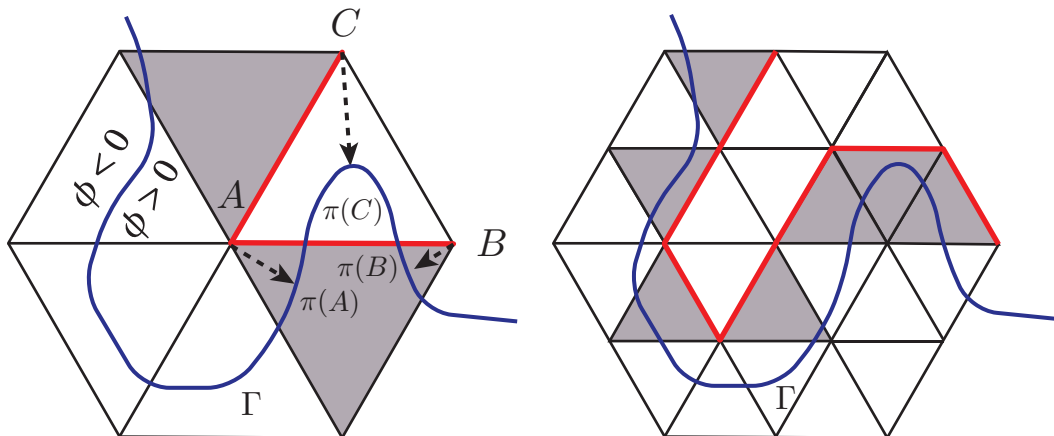


FIGURE 2.17 – π n'est pas injective sur le côté $[AB]$ bien que tous les angles de conditionnement soient aigus (gauche). Un simple raffinement du maillage rectifie cela (droite). Les éléments grisés sont coupés positivement.

D'autres exemples montrant l'importance de construire un maillage assez fin dans le but de garantir l'injectivité de π peuvent être trouvés dans [Rangarajan, 2012]. On y trouve aussi comment calculer la taille maximale du maillage. Pour des exemples type ingénieur, les maillages, utilisés en pratique pour garantir la qualité des résultats, suffisent dans la très grande majorité des cas. C'est pourquoi nous ne nous étendons pas plus sur le sujet.

2.2.6 Algorithme des maillages universels

La démarche des maillages universels est retranscrite dans l'ALG.4. C'est notamment grâce à cet algorithme que le maillage du cercle unité, présenté Figure 2.14, a été construit.

Algorithme 4 : Algorithme Maillages Universels

Entrées : Un maillage $m = \{el_i = (\mathbf{n}_1^i, \mathbf{n}_2^i, \mathbf{n}_3^i)_{i \in \llbracket 1, nb_{elem} \rrbracket}\}$ de triangles respectant les 2 conditions suffisantes, un contour Γ

Output : Nouveau maillage m^* conforme

- 1 **Initialisation** $m^* = \{\}$;
- 2 **Détermination des éléments coupés positivement et relaxation de nœuds intérieurs;**
- 3 **pour** $i = 1$ à $nb_{nœuds}$ **faire**
- 4 **si** $\phi(\mathbf{n}(i)) < 0$ **alors**
- 5 $\mathbf{n}(i)$ est dans le domaine Ω
- 6 **sinon**
- 7 $\mathbf{n}(i)$ est ajouté à la liste l_e des nœuds positifs de tous les éléments auxquels il appartient.
- 8 **si** $-r_1 < \phi(\mathbf{n}(i))$ **alors**
- 9 Relaxation de $\mathbf{n}(i)$ en calculant $p_h^1(\mathbf{n}(i))$
- 10 **Projection des nœuds sur le contour** Γ ;
- 11 **pour** $e = 1$ à nb_{elem} **faire**
- 12 **si** $taille(l_e)=2$ (*l'élément est coupé positivement*) **alors**
- 13 Projection des nœuds \mathbf{n}_i^e et \mathbf{n}_j^e positifs sur Γ ;
- 14 l'élément est ajouté à m^*
- 15 **sinon si** $taille(l_e)=0$ (*l'élément est entièrement compris dans Ω*) **alors**
- 16 l'élément est ajouté à m^*

2.3 Adaptation des maillages universels à la PGD géométrique

Les maillages universels peuvent se révéler utiles dans le cas de la PGD géométrique puisqu'ils permettent la construction peu coûteuse de maillages conformes. Nous pouvons nous en inspirer pour créer des maillages du domaine de référence Ω_0 conformes aux différentes configurations géométriques et possédant chacun le même nombre d'éléments et de nœuds. Ce dernier point est très important car lors du calcul des modes spatiaux PGD, la conservation du nombre des nœuds évitent des projections fastidieuses entre chaque maillage. Toutefois, l'algorithme des maillages universels ne peut être utilisé tel quel puisque seul l'intérieur ou l'extérieur du domaine Ω_α

est maillé *in fine* dans sa formulation classique (voir [Rangarajan et Lew, 2012]) et nous souhaiterions, dans notre cas, mailler le domaine complet Ω_0 . Nous avons dû porter, en conséquence, quelques modifications à la méthode des maillages universels.

2.3.1 Double relaxation

Le but final étant de conserver la totalité du maillage, il est nécessaire de garantir la qualité des éléments non seulement à l'intérieur du domaine défini par la géométrie immergée (comme dans l'approche classique) mais aussi à l'extérieur. Ainsi un deuxième type de relaxation est introduit, dans ce travail, pour les nœuds se situant dans un voisinage (à définir) extérieur du domaine. On définit alors une nouvelle fonction de relaxation $p_h^2 : \mathbb{R}^2 \rightarrow \mathbb{R}$ en s'inspirant de la première.

$$p_h^2(\mathbf{X}) = \begin{cases} \mathbf{X} - \eta_2 h \left(1 - \frac{\phi(\mathbf{X})}{r_2}\right) \nabla \phi(\mathbf{X}) & \text{si } 0 < \phi(\mathbf{X}) < r_2 \\ \mathbf{X} & \text{sinon} \end{cases} \quad (2.23)$$

où η_2 est un paramètre compris entre 0 et 1 (paramètre de relaxation) et r_2 est de l'ordre de quelques multiples de la taille du maillage h (paramètre définissant la taille du voisinage de relaxation).

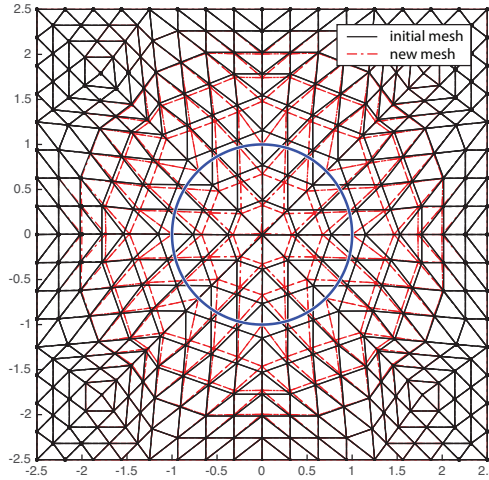


FIGURE 2.18 – Exemple d'application au cercle unité de l'algorithme modifié des maillages universels (paramètres utilisés pour la double relaxation : $r_1 = r_2 = 3h$ et $\eta_1 = \eta_2 = 0.3$). Cette fois-ci, le maillage entier est conservé.

La Figure 2.18 reprend l'exemple du cercle unité immergé dans un maillage de triangles mais cette fois-ci, non seulement l'intérieur du cercle est maillé mais aussi l'extérieur par l'intermédiaire d'une double relaxation. Après application de l'algorithme ALG.5, la totalité du nouveau maillage est conservée. Nous avons donc construit un maillage conforme au contour Γ (ici le cercle unité) tout en maintenant le nombre de nœuds et d'éléments du maillage total.

2.3.2 Nouvelle vérification

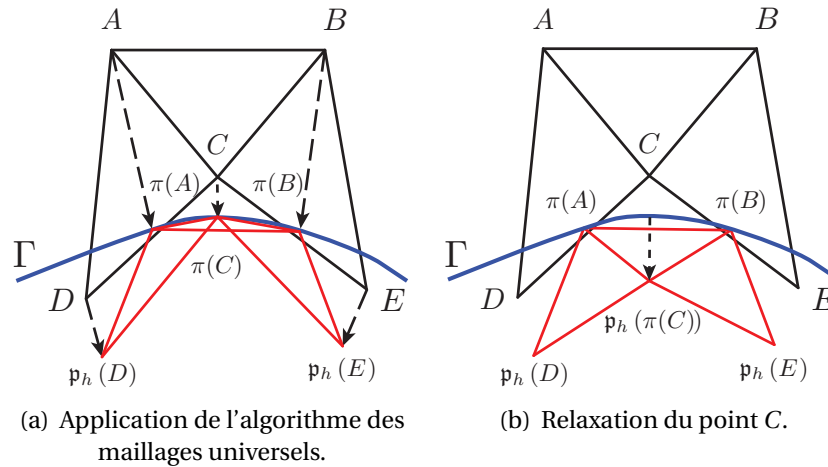


FIGURE 2.19 – Le triangle ABC se retrouve distordu après application de l'algorithme des maillages universels. La relaxation du point C , dans un second temps, permet à l'élément ABC de recouvrer une qualité convenable.

Le fait d'utiliser l'algorithme des maillages universels pour mailler à la fois l'intérieur et l'extérieur d'un domaine peut entraîner une complication non prise en compte précédemment, vu qu'une seule partie du nouveau maillage (l'intérieur ou l'extérieur) était conservée. Dans le cas présent, il peut arriver que des éléments situés dans le voisinage extérieur du contour Γ se retrouvent confondus avec ledit contour une fois projetés ou tout du moins qu'ils se retrouvent fortement aplatis. Ce cas est illustré sur la Figure 2.19. L'élément ABC n'aurait pas été conservé lors de l'utilisation classique de l'algorithme des maillages universels vu qu'il est initialement situé entièrement à l'extérieur du domaine à mailler. Or, nous voulons à présent garder le maillage dans son entier (intérieur comme extérieur) et la distorsion de forme de l'élément ABC n'est pas acceptable. C'est pourquoi, on procède à la relaxation du nœud C vers l'intérieur du domaine après sa projection sur le contour Γ en utilisant (2.22). Cette configuration correspond au cas d'un élément extérieur au contour dont tous les nœuds ont été projetés et donc en pratique, il suffit de comptabiliser le nombre de nœuds projetés de chaque élément extérieur pour traiter ce problème.

Néanmoins, le maillage final obtenu après relaxation de ces nœuds se révèle moins fidèle au contour Γ que celui construit à partir de l'algorithme des maillages universels classique (Figure 2.19). Mais il s'agit du prix à payer pour conserver intact le nombre d'éléments et de nœuds, cela empêchant toute suppression d'éléments.

2.3.3 Nouvel algorithme

Le nouvel algorithme des maillages universels permettant la construction de maillages conformes pour le domaine entier comprenant le contour Γ (et plus uniquement

l'extérieur ou l'intérieur) est présenté dans l'ALG.5. Il s'agit dans sa très grande majorité de l'ALG.4 auquel a été ajoutée une étape de vérification de la présence d'éléments aplatis et une deuxième relaxation.

Algorithme 5 : Algorithme Maillages Universels avec double relaxation

Entrées : Un maillage $m = \{e_{l_i} = (\mathbf{n}_1^i, \mathbf{n}_2^i, \mathbf{n}_3^i)_{i \in [1, nb_{elem}]}\}$ de triangles respectant les 2 conditions suffisantes, un contour Γ

Output : Nouveau maillage m^* conforme

- 1 **Initialisation** $m^* = \{\}$;
 - 2 **Détermination des éléments coupés positivement et relaxation de nœuds intérieurs**;
 - 3 **pour** $i = 1$ à $nb_{nœuds}$ **faire**
 - 4 **si** $\phi(\mathbf{n}(i)) < 0$ **alors**
 - 5 $\mathbf{n}(i)$ est dans le domaine Ω
 - 6 **sinon**
 - 7 $\mathbf{n}(i)$ est ajouté à la liste l_e des nœuds positifs de tous les éléments auxquels il appartient.
 - 8 **si** $-r_1 < \phi(\mathbf{n}(i))$ **alors**
 - 9 Relaxation de $\mathbf{n}(i)$ en calculant $p_h^1(\mathbf{n}(i))$
 - 10 **Projection des nœuds sur le contour Γ** ;
 - 11 **pour** $e = 1$ à nb_{elem} **faire**
 - 12 **si** $taille(l_e)=2$ (l'élément est coupé positivement) **alors**
 - 13 Projection des nœuds \mathbf{n}_i^e et \mathbf{n}_j^e positifs sur Γ ;
 - 14 l'élément est ajouté à m^* ;
 - 15 \mathbf{n}_i^e et \mathbf{n}_j^e sont ajoutés à la liste l_{proj}^e des nœuds projetés de chaque élément auquel ils appartiennent.
 - 16 **sinon si** $taille(l_e)=0$ (l'élément est entièrement compris dans Ω) **alors**
 - 17 l'élément est ajouté à m^*
 - 18 **Vérification de la présence d'éléments aplatis dans m^*** ;
 - 19 **pour** $i = 1$ à $nb_{nœuds}^*$ **faire**
 - 20 **si** $taille(l_{proj}^e)=3$ (l'élément est aplati) **alors**
 - 21 Relaxation du nœud central
 - 22 **Relaxation des nœuds extérieurs**;
 - 23 **pour** $i = 1$ à $nb_{nœuds}^*$ (on relaxe les nœuds du nouveau maillage m^*) **faire**
 - 24 **si** $\phi(\mathbf{n}^*(i)) < r_2$ **alors**
 - 25 Relaxation de $\mathbf{n}^*(i)$ en calculant $p_h^2(\mathbf{n}^*(i))$
-

2.3.4 Méthode PGD-maillages universels

Maintenant que nous avons à disposition un outil permettant la construction de maillages conformes à partir d'un même maillage de référence, nous pouvons l'intégrer dans notre méthodologie PGD.

Un domaine Ω_0 englobant toutes les géométries est tout d'abord défini. Ce domaine doit être assez grand pour permettre la relaxation des nœuds environnant la frontière Γ_α de chaque configuration. Sa dimension est directement liée à la valeur donnée au paramètre r_2 caractérisant le voisinage de relaxation à l'extérieur de Ω_α (voir Figure 2.20 pour un exemple de domaine de référence et d'application du nouvel algorithme des maillages universels).

L'ALG.5 est ensuite utilisé pour générer le maillage de chaque configuration. On se retrouve alors avec un ensemble de maillages possédant le même nombre de nœuds et d'éléments. Toutefois, puisque la position des nœuds diffère pour chaque maillage, l'application de la méthodologie PGD n'est pas encore possible. En effet, le calcul des modes PGD Λ_j doit se faire sur des nœuds ayant la même position.

Pour cela, une configuration μ est choisie arbitrairement parmi toutes les configurations et nous allons baser notre approche sur celle qui sera exposée dans le chapitre suivant. En effet, à chaque élément du maillage de la configuration α est associé un élément du maillage de la configuration μ et une transformation géométrique $\mathcal{T}_{\alpha\mu}$ les relie (voir la SEC.3.1.3 pour plus de détails). On peut donc facilement exprimer les matrices de rigidité élémentaires de chaque configuration géométrique comme suit. Pour $\alpha \in \mathcal{A}$:

$$\tilde{\mathbb{K}}_\alpha^e = \int_{\Omega_\alpha^e} \mathbb{B}^T(\alpha, \mathbf{X}) \mathbb{C}_\alpha(\mathbf{X}) \mathbb{B}(\alpha, \mathbf{X}) d\Omega_\alpha \quad (2.24)$$

$$= \int_{\Omega_\mu^e} \hat{\mathbb{B}}^T(\alpha, \mathbf{X}_\mu) \mathbb{C}_\alpha(\mathcal{T}_{\alpha\mu}(\mathbf{X}_\mu)) \hat{\mathbb{B}}(\alpha, \mathbf{X}_\mu) J_{\mathcal{T}_{\alpha\mu}}(\mathbf{X}_\mu) d\Omega_\mu \quad (2.25)$$

où $\hat{\mathbb{B}}(\alpha, \mathbf{X}_\mu) = \mathbb{B}(\mathcal{T}_{\alpha\mu}(\mathbf{X}_\mu))$ et $J_{\mathcal{T}_{\alpha\mu}}$ est le module du déterminant de la matrice Jacobienne de la transformation géométrique $\mathcal{T}_{\alpha\mu}$.

La relation de comportement est celle définie en (2.1), mais ici, nous n'avons plus besoin de nous préoccuper de savoir si un élément est coupé ou non puisque les maillages sont conformes. L'algorithme PGD est ensuite appliqué pour construire les différents modes.

2.3.5 Application numérique

Reprenons exactement le même exemple que SEC.2.1.4. Le maillage du domaine de référence Ω_0 est donné Figure 2.20 ainsi que le maillage généré à l'aide de l'algorithme de maillages universels pour une configuration géométrique donnée.

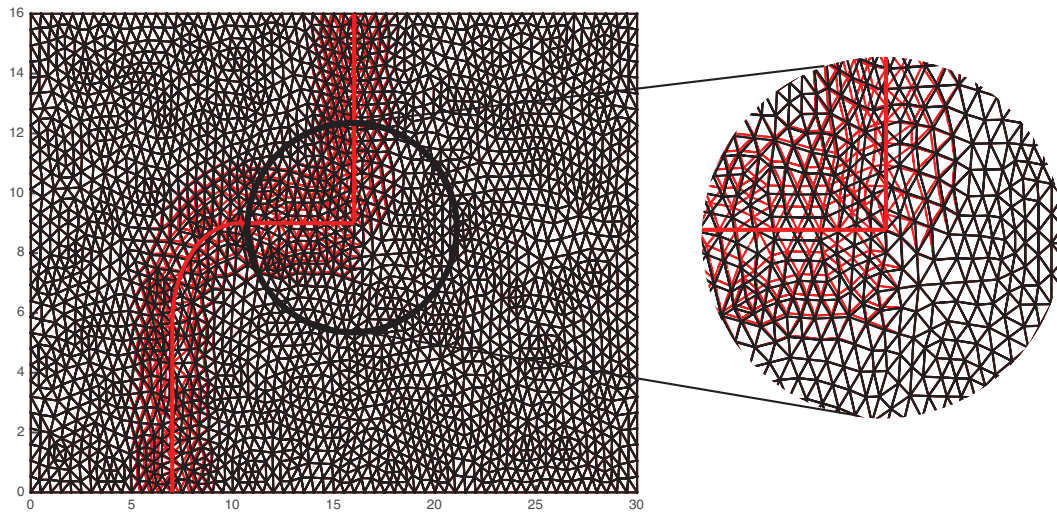


FIGURE 2.20 – Construction d'un maillage pour une configuration donnée à partir du maillage de référence.

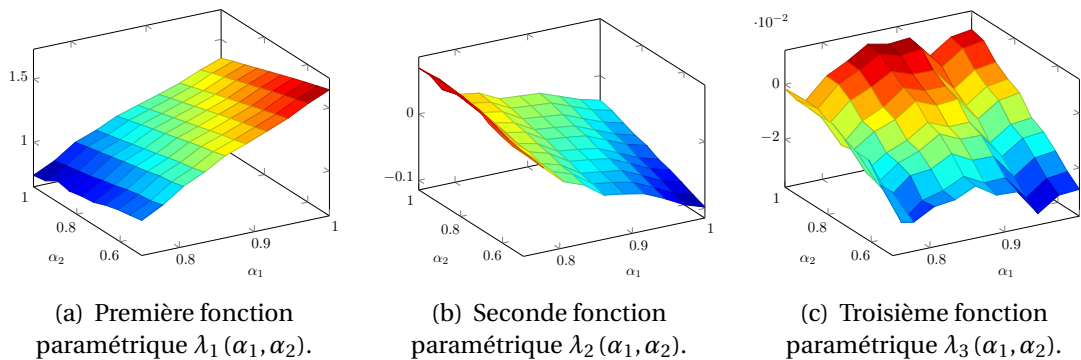


FIGURE 2.21 – Trois premières fonctions paramétriques.

Les trois premières fonctions paramétriques λ_i sont tracées Figure 2.21. Les modes PGD Λ_i sont tracés Figure 2.22 pour le maillage de la configuration $\alpha_1 = 0,75$ et $\alpha_2 = 0,5$.

La convergence de la méthode a été étudiée et les courbes associées à l'erreur en résidu et à l'erreur par rapport à une solution de référence éléments finis sont données Figure 2.23. Comme pour l'exemple précédent, la contrainte de Von Mises a été calculée pour trois configurations géométriques différentes (les deux extrêmes et une intermédiaire) et les résultats sont exposés Figure 2.24.

Ces résultats sont à comparer avec ceux obtenus par une méthode directe (éléments finis). On peut voir, sur la Figure 2.25, les contraintes de Von Mises pour la solution de référence, définie à partir d'une résolution éléments finis pour tous les jeux de

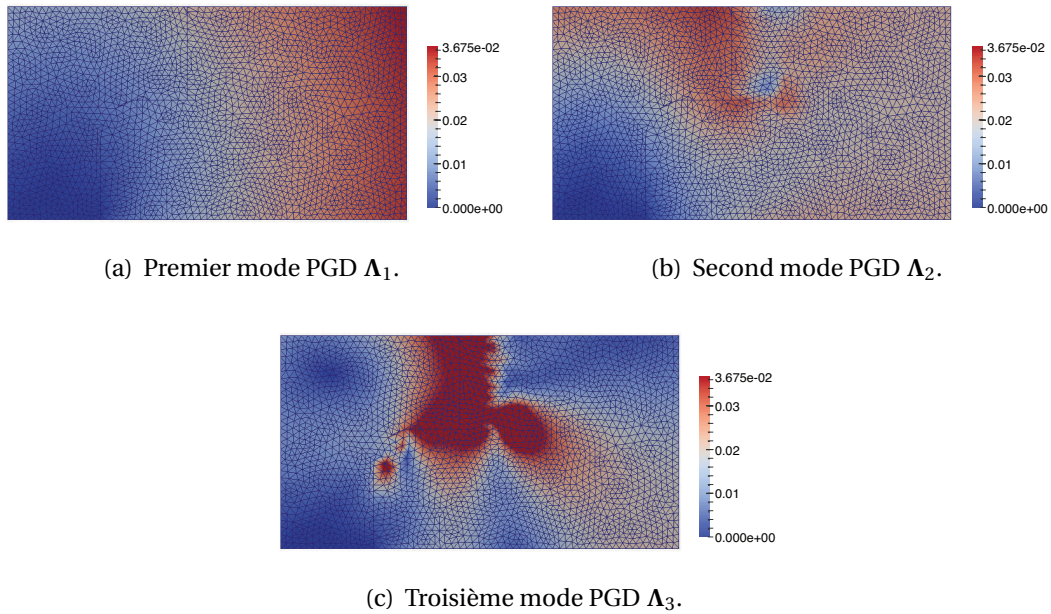


FIGURE 2.22 – Trois premiers modes PGD.

paramètres. Il s'agit des mêmes configurations géométriques que celles exposées Figure 2.24 et les mêmes maillages ont été utilisés pour la comparaison. On remarque la grande fidélité de la solution PGD (construite avec 39 modes) par rapport à la solution de référence.

Toutefois, un plus grand nombre de modes est, cette fois-ci, nécessaire pour représenter la solution en comparaison de la méthode présentée dans la première partie du chapitre (39 modes contre 25 modes précédemment). De plus, les fonctions paramétriques λ_k paraissent plus chahutées que celles obtenues par la paramétrisation de la rigidité seule.

2.4 Conclusions

Nous venons de mettre en place un cadre pour la prise en compte des variations de géométrie dans la méthode PGD. En combinant maillages universels et paramétrisation de la rigidité, nous avons réussi à traiter un exemple de variations de géométrie par la PGD. Néanmoins, nous n'avons pas résolu la difficulté évoquée dans le CHAP.1, à savoir que $\mathcal{a}(\square, \square; \alpha)$ et $\ell(\square; \alpha)$ doivent être paramétrés affinement si l'on veut obtenir des temps de calcul raisonnables. En d'autres termes, il s'agit dans notre cas d'écrire la matrice de rigidité $\tilde{\mathbb{K}}$ et le chargement à variables séparées. Cela pourrait être traité en ayant recours à la méthode EIM [Barrault *et al.*, 2004] (ce qui est très souvent le cas dans la communauté des Reduced Basis) ou par la méthode des points de référence (RPM) [Néron et Ladevèze, 2010, Capaldo *et al.*, 2014] afin de décomposer ces opéra-

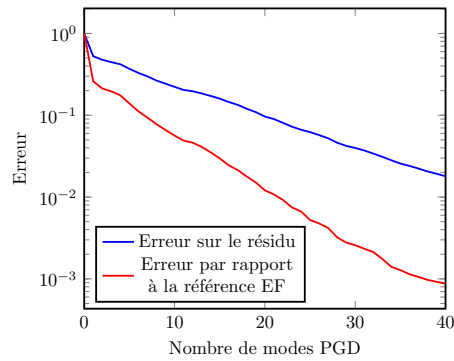


FIGURE 2.23 – Courbe de convergence de la méthode PGD-maillages universels. En bleu est tracée la courbe correspondant à l’erreur sur le résidu et en rouge celle par rapport à la référence éléments finis.

teurs. Nous allons voir dans le chapitre suivant, un autre cadre permettant la prise en compte des variations de géométrie par la PGD et cette fois, nous nous appliquerons à traiter la séparation des variables des différents opérateurs.

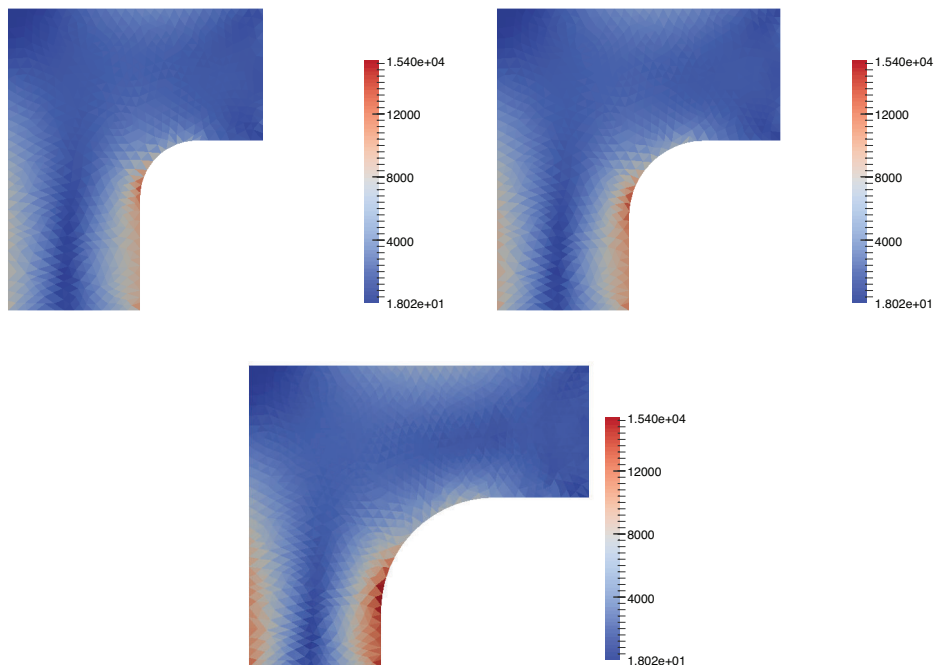


FIGURE 2.24 – Contrainte de Von Mises pour trois configurations géométriques différentes (solution PGD avec 39 modes).

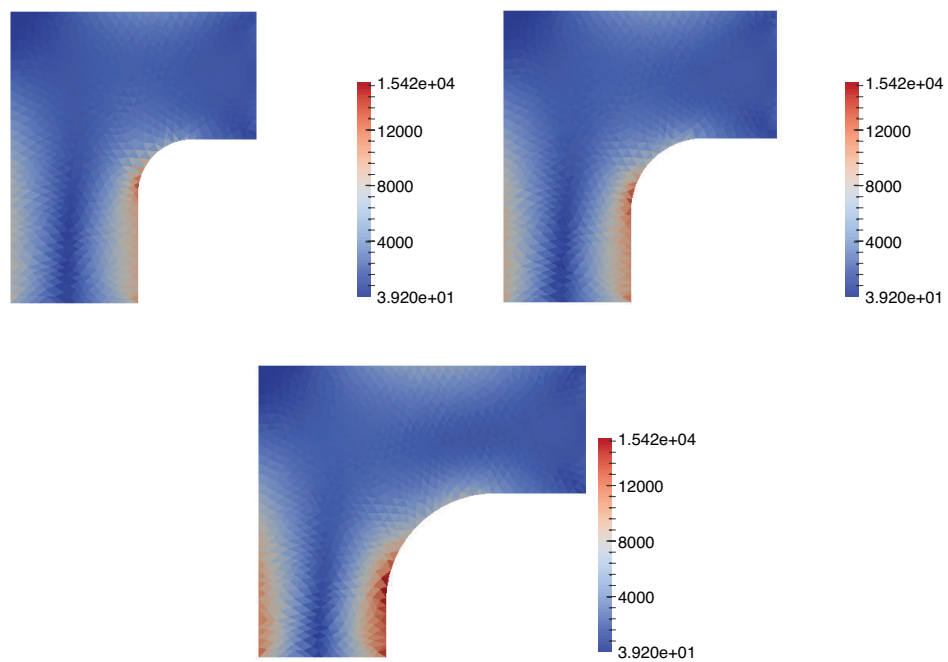


FIGURE 2.25 – Contrainte de Von Mises pour trois configurations géométriques différentes (solution de référence éléments finis).

Chapitre 3

Méthode des transformations géométriques

Un nouveau cadre est introduit pour la prise en compte des variations de géométrie au sein de la PGD. Cette méthode utilise des transformations géométriques afin de ramener la définition du problème variationnel à un domaine de référence fixe. Nous prenons en considération, cette fois-ci, l'importance de la séparabilité de la matrice de rigidité, chose qui avait été ignorée dans le chapitre précédent et nous verrons les conséquences que cela a dans le choix des transformations géométriques. Il sera traité non seulement du cas 2D mais aussi, dans un second temps, du cas axisymétrique. De plus, nous nous attacherons à valider cette nouvelle méthode sur notre exemple académique puis à l'appliquer au démonstrateur industriel et de proposer une approche pour intégrer cette méthode dans un process industriel de conception. Un exemple d'abaque virtuel sera donné et on s'intéressera à son utilisation pratique par l'ingénieur.

3.1 Transformations géométriques

Nous avons conclu le CHAP.1 sur quelques considérations sur les paramètres géométriques dans le cadre de la réduction de modèle. Réécrivons le Problème 1.6 mais cette fois-ci, $\alpha = (\alpha_1, \dots, \alpha_m) \in \mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_m$ est un ensemble de paramètres géométriques :

Problème 3.1. Trouver $u \in \mathcal{F} \otimes \mathcal{V}(\alpha)$ tel que

$$\forall v \in \mathcal{F} \otimes \mathcal{V}(\alpha), \int_{\mathcal{A}} \int_{\Omega_\alpha} \mathbb{C} \varepsilon(u) : \varepsilon(v) \, d\Omega \, d\mathcal{A} = \int_{\mathcal{A}} \int_{\partial_d \Omega_\alpha} f \cdot v \, dS \, d\mathcal{A}$$

avec $\mathcal{V}(\alpha) = \mathbf{H}_0^1(\Omega_\alpha) = \{v \in \mathbf{H}^1(\Omega_\alpha) \mid v|_{\partial_u \Omega_\alpha} = \mathbf{0}\}$.

On s'aperçoit que la dépendance du domaine d'intégration Ω_α en les paramètres complique les calculs des différentes intégrales puisqu'ils doivent se faire l'un à la suite de l'autre. Si la dépendance du domaine d'intégration pouvait être, de quelque manière que ce soit, contournée, cela faciliterait grandement le problème. La solution, introduite dans [Ammar *et al.*, 2014] et suivie dans cette section, est de recourir à un changement de variables. En d'autres termes, il s'agit de définir une structure de référence Ω_0 , qui peut être choisie arbitrairement parmi toutes les configurations par exemple, et se reporter, à partir de la structure courante Ω_α , où le problème était initialement défini, à cette structure de référence grâce à une transformation géométrique \mathcal{T}_α . Ainsi,

$$\mathbf{X} = \mathcal{T}_\alpha(\mathbf{X}_0) \text{ avec } \mathbf{X} \in \Omega_\alpha \text{ et } \mathbf{X}_0 \in \Omega_0 \quad (3.1)$$

$$\mathbf{u}(\alpha, \mathbf{X}) = \mathbf{u}(\alpha, \mathcal{T}_\alpha(\mathbf{X}_0)) \quad (3.2)$$

On réécrit le Problème 3.1 en appliquant le changement de variables :

Problème 3.2. Trouver $u \in \mathcal{F} \otimes \mathcal{V}_0$ tel que

$$\forall v \in \mathcal{F} \otimes \mathcal{V}_0, \int_{\mathcal{A}} \int_{\Omega_0} \mathbb{C} \varepsilon(u(\alpha, \mathcal{T}_\alpha(\mathbf{X}_0))) : \varepsilon(v(\alpha, \mathcal{T}_\alpha(\mathbf{X}_0))) J_{\mathcal{T}_\alpha}(\mathbf{X}_0) \, d\Omega_0 \, d\mathcal{A} = \int_{\mathcal{A}} \int_{\partial_d \Omega_0} f(\alpha, \mathcal{T}_\alpha(\mathbf{X}_0)) \cdot v(\alpha, \mathcal{T}_\alpha(\mathbf{X}_0)) J_{\mathcal{T}_\alpha}(\mathbf{X}_0) \, dS_0 \, d\mathcal{A}$$

avec $\mathcal{V}_0 = \mathbf{H}_0^1(\Omega_0) = \{v \in \mathbf{H}^1(\Omega_0) \mid v|_{\partial_u \Omega_0} = \mathbf{0}\}$ et $J_{\mathcal{T}_\alpha}$ est le module du déterminant de la matrice Jacobienne de la transformation géométrique \mathcal{T}_α .

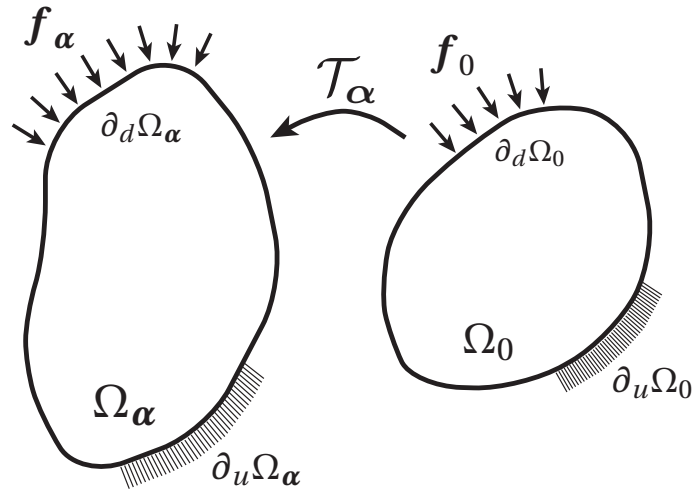


FIGURE 3.1 – Définition du problème sur une structure de référence Ω_0

Grâce à cette formulation, la dépendance selon les paramètres géométriques a été transférée du domaine d'intégration à l'intégrande via le Jacobien. On remarque que la qualité du maillage dans la structure courante est directement liée à la transformation géométrique (son maillage est un *mapping* de celui de la structure de référence [Galland *et al.*, 2011]). Il faudra s'attacher à contrôler la qualité des éléments transformés, i.e. éviter l'obtention d'éléments distordus. Nous avons éludé ce problème en prenant des maillages fins, de l'ordre de ceux utilisés en pratique pour la résolution de problèmes type ingénieur. Toutefois, on pourra s'intéresser, dans un avenir proche, à évaluer de manière efficace la qualité des maillages obtenus. Pour la suite de l'étude, on supposera que les éléments transformés ont une qualité acceptable.

La démarche consistant à recourir à des transformations géométriques est standard en optimisation et est notamment exposée dans [Allaire, 2007]. Désormais, il est plus aisé de construire une approximation PGD du champ de déplacement u en recourant à la démarche décrite SEC.1.3.

3.1.1 Proper Generalized Decomposition pour les variations de géométrie

Le Problème 3.2 est, tout d'abord, discrétisé en espace afin d'utiliser une approche de résolution type éléments finis. Le problème devient :

Problème 3.3. Trouver $u \in \mathcal{F} \otimes \mathbb{R}^N$ tel que

$$\forall \mathbf{U}^*(\alpha) \in \mathcal{F} \otimes \mathbb{R}^N, \quad \int_{\mathcal{A}} \mathbf{U}^T(\alpha) \mathbb{K}(\alpha) \mathbf{U}^*(\alpha) \, d\mathcal{A} = \int_{\mathcal{A}} \mathbf{F}^T(\alpha) \mathbf{U}^*(\alpha) \, d\mathcal{A} \quad (3.3)$$

où N est le nombre de degrés de liberté de la discrétisation en espace.

À partir de cette formulation, l'approximation PGD du vecteur déplacement \mathbf{U} est calculée :

$$\mathbf{U}(\boldsymbol{\alpha}) = \mathbf{U}(\alpha_1, \dots, \alpha_m) = \sum_{k=1}^n \prod_{j=1}^m \lambda_k^j(\alpha_j) \boldsymbol{\Lambda}_k \quad (3.4)$$

On suit la démarche classique de la résolution PGD dans le cas d'un problème avec paramètres en recourant, en particulier, à une approche Galerkin progressive. À l'itération n , la fonction test suivante est choisie pour le calcul du nouveau $(m+1)$ -uplet :

$$\mathbf{U}^*(\boldsymbol{\alpha}) = \mathbf{U}^*(\alpha_1, \dots, \alpha_m) = \prod_{j=1}^m \lambda^j(\alpha_j) \boldsymbol{\Lambda}^* + \sum_{i=1}^m \lambda^{i*}(\alpha_i) \prod_{\substack{j=1 \\ j \neq i}}^m \lambda^j(\alpha_j) \boldsymbol{\Lambda} \quad (3.5)$$

avec $\boldsymbol{\Lambda}^* \in \mathbb{R}^N$ et $\lambda^{i*} \in \mathcal{F}_i$ pour $i \in \llbracket 1, m \rrbracket$.

En réutilisant l'ALG.2 défini précédemment, on résout alternativement les systèmes linéaires suivants :

$\forall r \in \llbracket 1, m \rrbracket$,

$$\left\{ \boldsymbol{\Lambda}^T \left[\int_{\bar{\mathcal{A}}_r} \prod_{\substack{i=1 \\ i \neq r}}^m (\lambda^i(\alpha_i))^2 \mathbb{K}(\boldsymbol{\alpha}) d\bar{\mathcal{A}}_r \right] \boldsymbol{\Lambda} \right\} \lambda^r(\alpha_r) = \left[\int_{\bar{\mathcal{A}}_r} \prod_{\substack{i=1 \\ i \neq r}}^m \lambda^i(\alpha_i) \mathbf{F}^T(\boldsymbol{\alpha}) d\bar{\mathcal{A}}_r \right] \boldsymbol{\Lambda} \\ - \sum_{k=1}^{n-1} \left\{ \boldsymbol{\Lambda}_k^T \left[\int_{\bar{\mathcal{A}}_r} \prod_{\substack{i=1 \\ i \neq r}}^m \lambda^i(\alpha_i) \lambda_k^i(\alpha_i) \mathbb{K}(\boldsymbol{\alpha}) d\bar{\mathcal{A}}_r \right] \boldsymbol{\Lambda} \right\} \lambda_k^r(\alpha_r) \quad (3.6)$$

avec $\bar{\mathcal{A}}_r = \mathcal{A}_1 \times \dots \times \mathcal{A}_{r-1} \times \mathcal{A}_{r+1} \times \dots \times \mathcal{A}_m$

$$\left[\int_{\mathcal{A}} \prod_{i=1}^m (\lambda^i(\alpha_i))^2 \mathbb{K}(\boldsymbol{\alpha}) d\mathcal{A} \right] \boldsymbol{\Lambda} = \int_{\mathcal{A}} \prod_{i=1}^m \lambda^i(\alpha_i) \mathbf{F}(\boldsymbol{\alpha}) d\mathcal{A} \\ - \sum_{k=1}^{n-1} \left[\int_{\mathcal{A}} \prod_{i=1}^m \lambda^i(\alpha_i) \lambda_k^i(\alpha_i) \mathbb{K}(\boldsymbol{\alpha}) d\mathcal{A} \right] \boldsymbol{\Lambda}_k \quad (3.7)$$

On peut noter que le calcul des diverses intégrales serait facilité de manière significative si la matrice de rigidité \mathbb{K} était à variables séparées. Cela sera le sujet de la prochaine sous-section.

3.1.2 Calcul de la matrice de rigidité

Nous nous placerons dans un premier temps dans le cas 2D, le cas 3D pouvant être facilement déduit de cette approche. Nous étendrons ensuite notre méthodologie au cas axisymétrique lors du traitement du démonstrateur industriel. Des difficultés supplémentaires apparaîtront alors.

Commençons par écrire la matrice de rigidité $\mathbb{K}^e(\alpha)$ d'un élément de la structure courante. En appliquant un changement de variables, elle sera, alors, définie sur un élément de la structure de référence.

$$\mathbb{K}^e(\alpha) = \int_{\Omega_\alpha^e} \mathbb{B}^T(\alpha, \mathbf{X}) \mathbb{C} \mathbb{B}(\alpha, \mathbf{X}) \, d\Omega \quad (3.8)$$

$$= \int_{\Omega_0^e} \hat{\mathbb{B}}^T(\alpha, \mathbf{X}_0) \mathbb{C} \hat{\mathbb{B}}(\alpha, \mathbf{X}_0) J_{\mathcal{T}_\alpha}(\mathbf{X}_0) \, d\Omega_0 \quad (3.9)$$

où

- \mathbb{B} : matrice de dérivées partielles des fonctions de forme
- \mathbb{C} : matrice de Hooke (ici indépendante de α parce que nous ne traitons pas de paramètres matériaux)
- $\hat{\mathbb{B}}(\alpha, \mathbf{X}_0) = \mathbb{B}(\mathcal{T}_\alpha(\mathbf{X}_0))$

Pour obtenir une formulation à variables séparées de $\mathbb{K}^e(\alpha)$, on doit, tout d'abord, écrire $\hat{\mathbb{B}}$ sous forme séparée.

Écrivons la matrice $\hat{\mathbb{B}}$:

$$\hat{\mathbb{B}} = [\hat{\mathbb{B}}_1, \hat{\mathbb{B}}_2, \dots, \hat{\mathbb{B}}_p] \quad (3.10)$$

où p correspond au nombre de fonctions de forme utilisées et donc au type d'élément utilisé.

$$\forall I \in \llbracket 1, p \rrbracket, \hat{\mathbb{B}}_I = \begin{bmatrix} N_{I,x} & 0 \\ 0 & N_{I,y} \\ N_{I,y} & N_{I,x} \end{bmatrix} \quad (3.11)$$

Il est nécessaire de rappeler que le problème est défini sur la structure de référence Ω_0 . Ainsi $N_{I,x}$ et $N_{I,y}$ sont inconnus au contraire de N_{I,x_0} et N_{I,y_0} . Évidemment, il est possible d'exprimer $N_{I,x}$ et $N_{I,y}$ en fonction de N_{I,x_0} et N_{I,y_0} :

$$\begin{bmatrix} N_{I,x_0} \\ N_{I,y_0} \end{bmatrix} = \underbrace{\begin{bmatrix} x_{,x_0} & y_{,x_0} \\ y_{,y_0} & y_{,y_0} \end{bmatrix}}_{J_{\mathcal{T}_\alpha}^T} \begin{bmatrix} N_{I,x} \\ N_{I,y} \end{bmatrix} \quad (3.12)$$

Donc,

$$\begin{bmatrix} N_{,x} \\ N_{,y} \end{bmatrix} = \mathbb{J}_{\mathcal{T}_\alpha}^{-T} \begin{bmatrix} N_{,x_0} \\ N_{,y_0} \end{bmatrix} \quad (3.13)$$

En introduisant la comatrice de $\mathbb{J}_{\mathcal{T}_\alpha}$:

$$\begin{bmatrix} N_{,x} \\ N_{,y} \end{bmatrix} = \frac{1}{J_{\mathcal{T}_\alpha}} \text{com}(\mathbb{J}_{\mathcal{T}_\alpha}) \begin{bmatrix} N_{,x_0} \\ N_{,y_0} \end{bmatrix} \quad (3.14)$$

(3.11) devient, en définissant $\text{Com} = \text{com}(\mathbb{J}_{\mathcal{T}_\alpha})$:

$\forall I \in \llbracket 1, p \rrbracket$,

$$\hat{\mathbb{B}}_I = \frac{1}{J_{\mathcal{T}_\alpha}} \begin{bmatrix} \text{Com}_{1,1} N_{I,x_0} + \text{Com}_{1,2} N_{I,y_0} & 0 \\ 0 & \text{Com}_{2,1} N_{I,x_0} + \text{Com}_{2,2} N_{I,y_0} \\ \text{Com}_{2,1} N_{I,x_0} + \text{Com}_{2,2} N_{I,y_0} & \text{Com}_{1,1} N_{I,x_0} + \text{Com}_{1,2} N_{I,y_0} \end{bmatrix} \quad (3.15)$$

La matrice Jacobienne $\mathbb{J}_{\mathcal{T}_\alpha}$ de la transformation géométrique dépend à la fois des variables d'espace et des paramètres géométriques. C'est pourquoi, il n'est pas trivial d'obtenir une représentation à variables séparées. Il peut être envisagé d'obtenir une forme séparée du Jacobien grâce à une HOSVD (*Higher-Order Singular Value Decomposition* [De Lathauwer *et al.*, 2000]) mais, en raison du grand nombre de HOSVD nécessaires (quatre pour chaque élément), cela augmenterait le temps de calcul de manière importante. On propose, par conséquent, une autre solution, à savoir de supprimer la dépendance de la matrice Jacobienne selon les variables d'espace. Pour cela, la manière la plus simple est d'avoir recours uniquement à des transformations géométriques affines selon les variables d'espace. Lors du calcul de la matrice Jacobienne, les variables d'espace disparaîtront naturellement lors de la dérivation. Dans cette optique, la structure est divisée en sous-domaines dans lesquels est définie une transformation géométrique affine. Cette solution a aussi été proposée dans [Ammar *et al.*, 2014] où les géométries considérées étaient décomposées en sous-domaines triangulaires. Cette approche est particulièrement adaptée dans le cas de géométries linéaires, autrement dit composées des segments. Comme il sera exposé SEC.3.3, le démonstrateur industriel, que nous étudierons, présentera une géométrie complexe composée de splines. Or, l'utilisation de triangles ne permet pas de représenter fidèlement ce genre de frontières.

SEC.3.1.3, des exemples des transformations géométriques affines seront présentés suivis d'une application numérique sur un cas académique.

Nous écrivons une transformation géométrique par la suite comme :

$$\mathcal{T}_\alpha(\mathbf{X}_0) = \begin{pmatrix} \mathcal{T}_\alpha^x(\mathbf{X}_0) \\ \mathcal{T}_\alpha^y(\mathbf{X}_0) \end{pmatrix} \quad (3.16)$$

$$= \begin{pmatrix} a_x(\alpha) x_0 + b_x(\alpha) y_0 + c_x(\alpha) \\ a_y(\alpha) x_0 + b_y(\alpha) y_0 + c_y(\alpha) \end{pmatrix} \quad (3.17)$$

avec $\mathbf{X}_0 = (x_0, y_0)$.

Donc les matrices Jacobienne et Com deviennent :

$$\mathbb{J}_{\mathcal{T}_\alpha} = \begin{bmatrix} \frac{\partial \mathcal{T}_\alpha^x}{\partial x_0} & \frac{\partial \mathcal{T}_\alpha^y}{\partial x_0} \\ \frac{\partial \mathcal{T}_\alpha^x}{\partial y_0} & \frac{\partial \mathcal{T}_\alpha^y}{\partial y_0} \end{bmatrix} \quad (3.18)$$

$$= \begin{bmatrix} a_x(\alpha) & a_y(\alpha) \\ b_x(\alpha) & b_y(\alpha) \end{bmatrix} \quad (3.19)$$

$$\text{Com} = \begin{bmatrix} b_y(\alpha) & -b_x(\alpha) \\ -a_y(\alpha) & a_x(\alpha) \end{bmatrix} \quad (3.20)$$

Le Jacobien ne dépend plus des variables d'espace mais il n'est toujours pas évident de donner une représentation à variables séparées à la matrice $\hat{\mathbb{B}}$. Pour cela, définissons les matrices suivantes :

$$\begin{aligned} \mathbb{B}_1^I(\mathbf{X}_0) &= \begin{bmatrix} N_{I,x_0}(\mathbf{X}_0) & 0 \\ 0 & 0 \\ 0 & N_{I,x_0}(\mathbf{X}_0) \end{bmatrix} \\ \mathbb{B}_2^I(\mathbf{X}_0) &= \begin{bmatrix} N_{I,y_0}(\mathbf{X}_0) & 0 \\ 0 & 0 \\ 0 & N_{I,y_0}(\mathbf{X}_0) \end{bmatrix} \\ \mathbb{B}_3^I(\mathbf{X}_0) &= \begin{bmatrix} 0 & 0 \\ 0 & N_{I,x_0}(\mathbf{X}_0) \\ N_{I,x_0}(\mathbf{X}_0) & 0 \end{bmatrix} \\ \mathbb{B}_4^I(\mathbf{X}_0) &= \begin{bmatrix} 0 & 0 \\ 0 & N_{I,y_0}(\mathbf{X}_0) \\ 0 & 0 \\ N_{I,y_0}(\mathbf{X}_0) & 0 \end{bmatrix} \end{aligned} \quad (3.21)$$

et les coefficients :

$$\beta_1 = b_y(\alpha), \beta_2 = -a_y(\alpha), \beta_3 = -b_x(\alpha), \beta_4 = a_x(\alpha) \quad (3.22)$$

Ainsi, la matrice $\hat{\mathbb{B}}$ peut être écrite :

$$\hat{\mathbb{B}}(\alpha, \mathbf{X}_0) = \frac{1}{J_{\mathcal{T}_\alpha}} \sum_{i=1}^4 \beta_i(\alpha) \mathbb{B}_i(\mathbf{X}_0) \quad (3.23)$$

ce qui conduit à l'expression de \mathbb{K}^e sous forme séparée suivante :

$$\mathbb{K}^e(\alpha) = \frac{1}{J_{\mathcal{T}_\alpha}} \sum_{i=1}^4 \sum_{j=1}^4 \beta_i(\alpha) \beta_j(\alpha) \int_{\Omega_0^e} \mathbb{B}_i^T(\mathbf{X}_0) \mathbb{C} \mathbb{B}_j(\mathbf{X}_0) d\Omega_0 \quad (3.24)$$

Les différentes matrices qui apparaissent dans la décomposition de $\mathbb{K}^e(\alpha)$ doivent être définies pour chaque élément et les coefficients dépendant des paramètres géométriques pour chaque sous-domaine. Le nombre de termes présents dans la décomposition de la matrice de rigidité est fixe et ne dépend pas de la discrétisation de l'espace des paramètres \mathcal{A} .

3.1.3 Exemples de transformations géométriques affines

Nous avons vu que les transformations géométriques affines permettaient de donner une représentation à variables séparées de $\mathbb{K}^e(\alpha)$. Une question se pose, à présent : *les transformations géométriques affines sont-elles capables de représenter une grande diversité de variations de géométrie ?* En effet, la validité de notre méthode repose sur la réponse à cette question. Si un faible nombre de cas de variations de géométrie peut être pris en compte, la méthode devient alors caduque. Essayons de donner une description des transformations affines élémentaires que l'on peut rencontrer. On peut noter, tout de suite, que la composition de transformations affines est, elle-même, une transformation affine.

Les transformations affines contiennent les similitudes (homothéties, réflexions et mouvements rigides), les dilatations dans une direction donnée et les transvections. On écrit les parties linéaires des transformations citées précédemment ainsi que leur matrice Jacobienne (concernant les similitudes, on s'intéressera uniquement aux homothéties) dans le plan (x, y) en utilisant les coordonnées $\mathbf{X}_0 = (\eta, \xi)$:

- Homothétie ou dilatation homogène de coefficient α (Figure 3.2(a))

$$\mathcal{T}_\alpha(\eta, \xi) = \alpha \begin{pmatrix} \eta \\ \xi \end{pmatrix}, \mathbb{J}_{\mathcal{T}_\alpha} = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix} \quad (3.25)$$

avec $\alpha = (\alpha)$.

- Dilatation dans les directions x et y de coefficients, respectivement, α et β (Figure 3.2(b))

$$\mathcal{T}_\alpha(\eta, \xi) = \begin{pmatrix} \alpha\eta \\ \beta\xi \end{pmatrix}, \mathbb{J}_{\mathcal{T}_\alpha} = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \quad (3.26)$$

avec $\alpha = (\alpha, \beta)$.

- Transvection dans la direction x (Figure 3.2(c))

$$\mathcal{T}_\alpha(\eta, \xi) = \begin{pmatrix} \eta + \alpha\xi \\ \xi \end{pmatrix}, \mathbb{J}_{\mathcal{T}_\alpha} = \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} \quad (3.27)$$

avec $\alpha = (\alpha)$.

- De plus, tous les triangles sont liés entre eux par des transformations affines définies par la position de leurs sommets. Ainsi, en considérant la transformation du triangle $((0, 0), (0, 1), (1, 0))$ vers le triangle $((0, 0), (\alpha, \beta), (\gamma, \delta))$ (Figure 3.2(d)), on obtient :

$$\mathcal{T}_\alpha(\eta, \xi) = \begin{pmatrix} \alpha\eta + \beta\xi \\ \gamma\eta + \delta\xi \end{pmatrix}, \mathbb{J}_{\mathcal{T}_\alpha} = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} \quad (3.28)$$

avec $\alpha = (\alpha, \beta, \gamma, \delta)$.

Dans [Ammar *et al.*, 2014], cette dernière propriété a été mise à profit pour partitionner des géométries. En effet, il serait quasiment toujours possible de décrire une variation géométrique avec uniquement des triangles en raffinant suffisamment la partition de la géométrie. Cependant, dans la SEC.3.3, il sera montré que, dans notre cas, même si l'on peut utiliser des triangles, le temps de calcul peut devenir problématique si le nombre de triangles nécessaires à la bonne représentation du contour devient important car certains éléments de la décomposition de la matrice de rigidité sont définis pour chaque sous-domaine. Une augmentation du nombre de sous-domaines entraîne, par conséquent, une augmentation du temps de calcul.

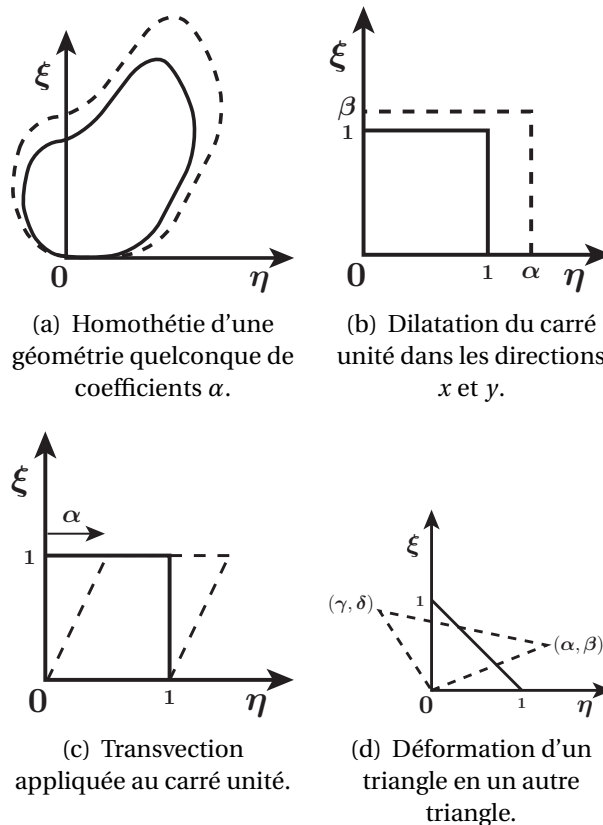


FIGURE 3.2 – Exemples de transformations géométriques affines.

3.2 Application numérique

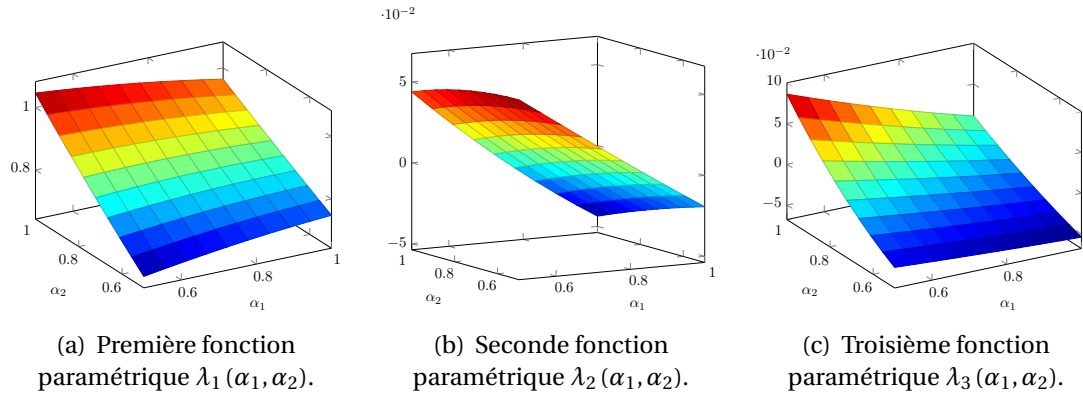


FIGURE 3.3 – Trois premières fonctions paramétriques.

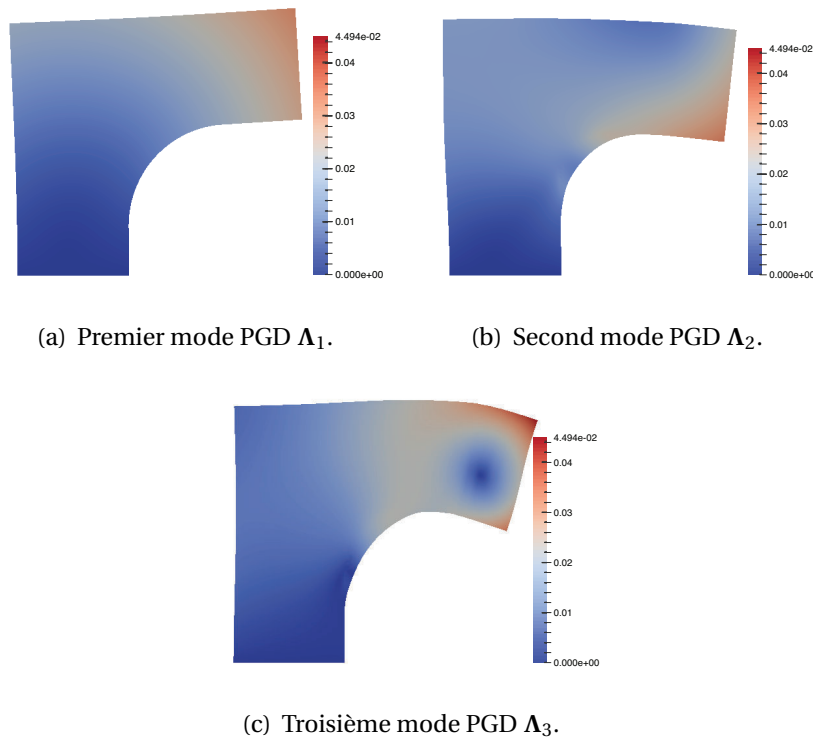


FIGURE 3.4 – Trois premiers modes PGD.

L'exemple est toujours le même mais cette fois-ci, l'intervalle où appartient α_1 est plus grand, $\alpha_1 \in [0, 5; 1]$. Comme pour les applications précédentes, les fonctions paramétriques λ_1 , λ_2 et λ_3 sont données Figure 3.3 et les modes Λ_1 , Λ_2 et Λ_3 sont représentés Figure 3.4. On remarque ici que les fonctions paramétriques sont beaucoup

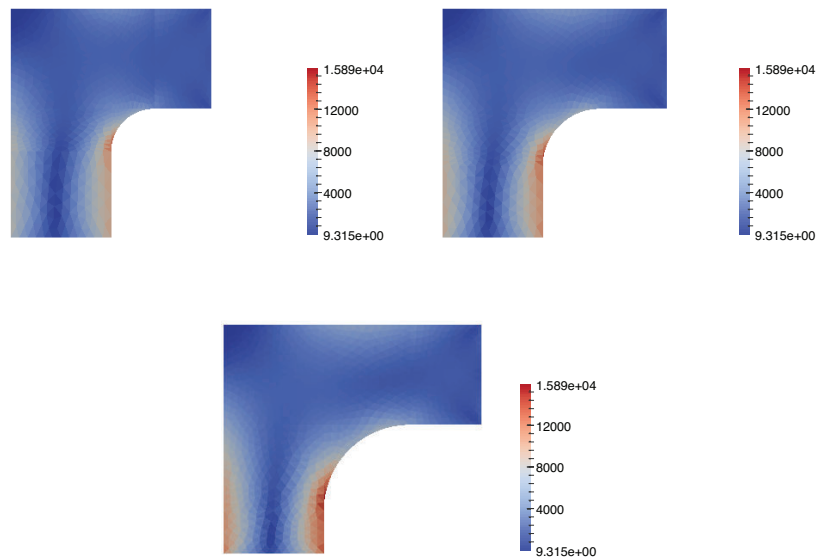


FIGURE 3.5 – Contrainte de Von Mises pour trois configurations géométriques différentes (solution PGD avec 6 modes).

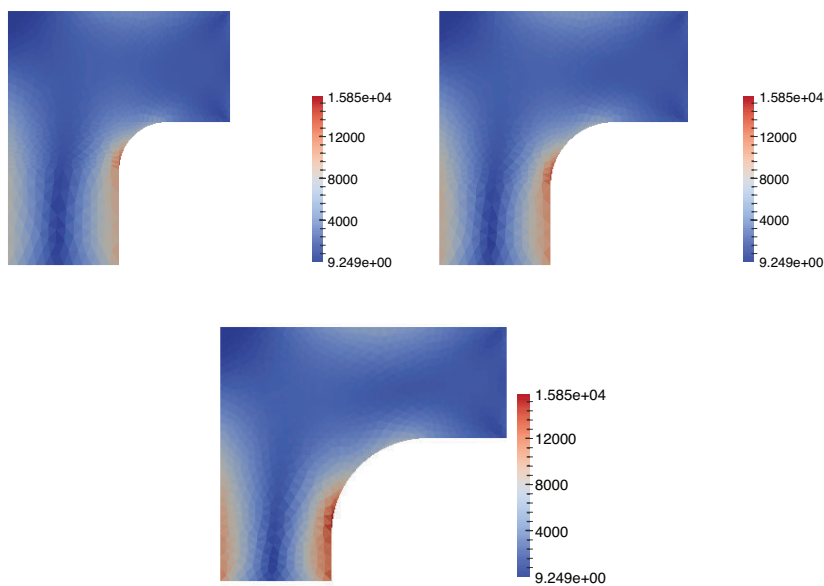


FIGURE 3.6 – Contrainte de Von Mises pour trois configurations géométriques différentes (solution de référence).

plus lissées que dans le cadre de la paramétrisation de la rigidité, cela étant très probablement dû au fait que l'approche est ici continue alors que précédemment elle était plutôt discrète. Un seul maillage est ici construit. Il subit ensuite des contractions et di-

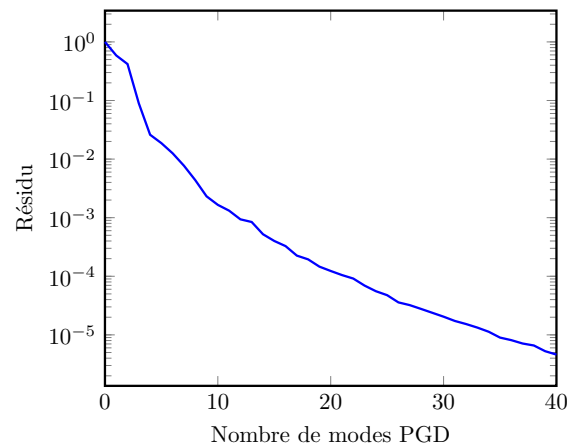


FIGURE 3.7 – Courbe de convergence de la méthode des transformations géométriques pour un exemple 2D.

latations suivant les transformations géométriques en jeu. Dans la méthode développée au CHAP.2, une série de maillages était définie lui conférant un caractère discret.

On note aussi la très nette amélioration de la convergence (voir Figure 3.7). Cela est confirmé par la comparaison des contraintes de Von Mises obtenues pour une solution PGD ne contenant que six modes (voir Figure 3.5) avec celle de la solution éléments finis de référence (voir Figure 3.6).

3.3 Application à un démonstrateur industriel

À la vue des résultats encourageants obtenus par cette méthode sur notre exemple académique, il a été décidé en collaboration avec AIRBUS Defence & Space de l'appliquer à un démonstrateur industriel plus complexe qui avait déjà été évoqué dans le CHAP.1. Une difficulté supplémentaire a été ajoutée dans cette étude, à savoir la nécessité de son intégration dans un process de conception industriel. Attachons-nous tout d'abord à présenter le cas industriel en question.

3.3.1 Présentation du démonstrateur

Pendant la phase de conception, on peut jouer sur un certain nombre de paramètres tels que des chargements, des paramètres matériau (modules de Young, coefficients de Poisson...) et des paramètres géométriques. Ici, on s'intéressera aux paramètres géométriques.

Comme montré dans la Figure 3.8, la structure considérée est une géométrie axisymétrique dont le matériau est linéaire élastique et isotrope, paramétrée par la position des points 1 et 2 appartenant au contour, soit un total de quatre paramètres géométriques (positions en x et en y pour chacun des points). Le jeu de paramètres est noté

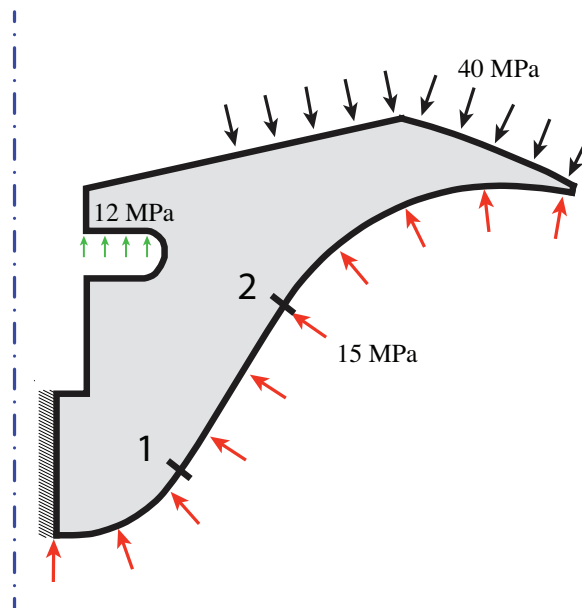


FIGURE 3.8 – Structure considérée.

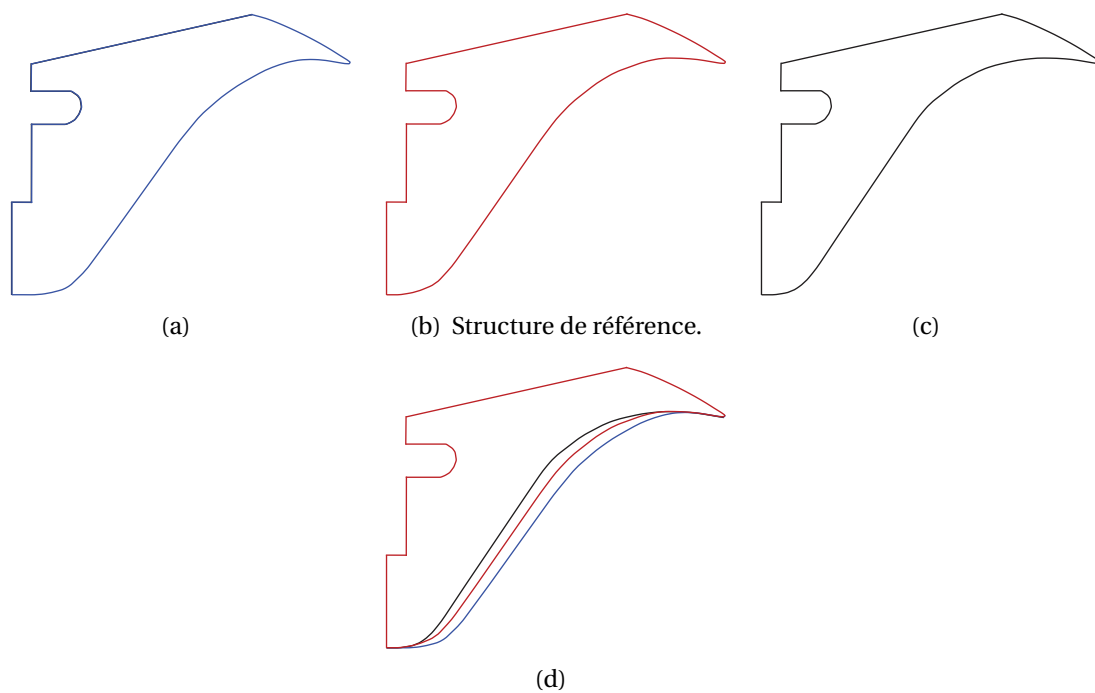


FIGURE 3.9 – Les variations de géométrie sont délimitées par les structures (a) et (c).

$\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (r_1, z_1, r_2, z_2)$ appartenant à l'espace de conception \mathcal{A} . Les variations de géométries que subit la structure sont visibles sur la Figure 3.9.

3.3.2 Extension de la méthode au cas axisymétrique

La formulation du cas axisymétrique est proche de celle 2D. Toutefois, nous allons voir que des difficultés non négligeables apparaissent, notamment dues à la présence du rayon $r(\alpha)$ dans le calcul des intégrales. Nous présenterons alors des solutions permettant de contourner ces points durs.

Il nous faut, tout d'abord, écrire de nouveau la matrice de rigidité élémentaire $\mathbb{K}^e(\alpha)$ afin d'avoir à l'esprit les différences fondamentales avec le cas 2D :

$$\mathbb{K}^e(\alpha) = 2\pi \int_{\Omega^e(\alpha)} \mathbb{B}^T(\alpha, \mathbf{X}) \mathbb{C} \mathbb{B}(\alpha, \mathbf{X}) r(\alpha, \mathbf{X}) d\Omega \quad (3.29)$$

$$= 2\pi \int_{\Omega_0^e} \hat{\mathbb{B}}^T(\alpha, \mathbf{X}_0) \mathbb{C} \hat{\mathbb{B}}(\alpha, \mathbf{X}_0) \hat{r}(\alpha, \mathbf{X}_0) J_{\mathcal{T}_\alpha}(\mathbf{X}_0) d\Omega_0 \quad (3.30)$$

où

- \mathbb{B} : matrice de dérivées partielles des fonctions de forme
- r : rayon
- $\hat{\mathbb{B}}(\alpha, \mathbf{X}_0) = \mathbb{B}(\mathcal{T}_\alpha(\mathbf{X}_0))$ et $\hat{r}(\alpha, \mathbf{X}_0) = r(\mathcal{T}_\alpha(\mathbf{X}_0))$

On remarque immédiatement que la modélisation axisymétrique introduit le rayon \hat{r} , qui dépend des paramètres géométriques, dans l'intégrande. Cela va sensiblement compliquer le calcul d'une expression à variables séparées de la matrice de rigidité. Comme pour le cas 2D, nous allons nous pencher sur la matrice des dérivées partielles $\hat{\mathbb{B}}$ et nous attacher à l'écrire sous forme séparée. Dans le cas axisymétrique, $\hat{\mathbb{B}}$ devient :

$$\forall I \in \llbracket 1, p \rrbracket, \hat{\mathbb{B}}_I = \begin{bmatrix} N_{I,r} & 0 \\ 0 & N_{I,z} \\ \frac{N_I}{r} & 0 \\ N_{I,z} & N_{I,r} \end{bmatrix} \quad (3.31)$$

On réintroduit les matrices Jacobienne et Com définies dans le cas 2D, qui demeurent identiques (il suffit de remplacer x par r et y par z dans leur expression) et on écrit la matrice des dérivées partielles :

$$\forall I \in \llbracket 1, p \rrbracket,$$

$$\hat{\mathbb{B}}_I = \frac{1}{J_{\mathcal{T}_\alpha}} \begin{bmatrix} \text{Com}_{1,1} N_{I,r_0} + \text{Com}_{1,2} N_{I,z_0} & 0 \\ 0 & \text{Com}_{2,1} N_{I,r_0} + \text{Com}_{2,2} N_{I,z_0} \\ \frac{N_I}{r} & 0 \\ \text{Com}_{2,1} N_{I,r_0} + \text{Com}_{2,2} N_{I,z_0} & \text{Com}_{1,1} N_{I,r_0} + \text{Com}_{1,2} N_{I,z_0} \end{bmatrix} \quad (3.32)$$

La démarche est similaire au cas 2D, néanmoins une nouvelle matrice apparaît dans la décomposition de $\hat{\mathbb{B}}$ due à la présence du terme $\frac{N_I}{r}$. En effet les nouvelles matrices \mathbb{B}_i sont à présent :

$$\begin{aligned}
\mathbb{B}_1^I(\mathbf{X}_0) &= \begin{bmatrix} N_{I,r_0}(\mathbf{X}_0) & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & N_{I,r_0}(\mathbf{X}_0) \end{bmatrix} & \mathbb{B}_2^I(\mathbf{X}_0) &= \begin{bmatrix} N_{I,z_0}(\mathbf{X}_0) & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & N_{I,z_0}(\mathbf{X}_0) \end{bmatrix} \\
\mathbb{B}_3^I(\mathbf{X}_0) &= \begin{bmatrix} 0 & 0 \\ 0 & N_{I,r_0}(\mathbf{X}_0) \\ 0 & 0 \\ N_{I,r_0}(\mathbf{X}_0) & 0 \end{bmatrix} & \mathbb{B}_4^I(\mathbf{X}_0) &= \begin{bmatrix} 0 & 0 \\ 0 & N_{I,z_0}(\mathbf{X}_0) \\ 0 & 0 \\ N_{I,z_0}(\mathbf{X}_0) & 0 \end{bmatrix} \\
\mathbb{B}_5^I(\mathbf{X}_0) &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ N_I(\mathbf{X}_0) & 0 \\ 0 & 0 \end{bmatrix}
\end{aligned} \tag{3.33}$$

Et la matrice $\hat{\mathbb{B}}$ devient :

$$\hat{\mathbb{B}}(\boldsymbol{\alpha}, \mathbf{X}_0) = \frac{1}{J_{\mathcal{F}_\alpha}} \sum_{i=1}^4 \beta_i(\boldsymbol{\alpha}) \mathbb{B}_i(\mathbf{X}_0) + \frac{1}{\hat{r}(\boldsymbol{\alpha}, \mathbf{X}_0)} \mathbb{B}_5(\mathbf{X}_0) \tag{3.34}$$

les coefficients β_i étant ceux du cas 2D.

Ce qui conduit à l'expression de \mathbb{K}^e suivante :

$$\begin{aligned}
\mathbb{K}^e(\boldsymbol{\alpha}) &= \frac{1}{J_{\mathcal{F}_\alpha}} \sum_{i=1}^4 \sum_{j=1}^4 \beta_i(\boldsymbol{\alpha}) \beta_j(\boldsymbol{\alpha}) \int_{\Omega_0^e} \mathbb{B}_i^T(\mathbf{X}_0) \mathbb{C} \mathbb{B}_j(\mathbf{X}_0) \hat{r}(\boldsymbol{\alpha}, \mathbf{X}_0) d\Omega_0 \\
&\quad + J_{\mathcal{F}_\alpha} \int_{\Omega_0^e} \frac{1}{\hat{r}(\boldsymbol{\alpha}, \mathbf{X}_0)} \mathbb{B}_5^T(\mathbf{X}_0) \mathbb{C} \mathbb{B}_5(\mathbf{X}_0) d\Omega_0 \tag{3.35}
\end{aligned}$$

Cette expression de \mathbb{K}^e n'est que partiellement satisfaisante. Partiellement car \hat{r} et $1/\hat{r}$ contiennent à la fois les variables d'espace et les paramètres géométriques. Comme il a été fait le choix de transformations géométriques affines, \hat{r} est déjà sous forme séparée et s'écrit $\hat{r} = a_r r_0 + b_r z_0 + c_r$.

Le cas de $1/\hat{r}$ est légèrement plus compliqué puisque $1/\hat{r}$ n'a pas de forme séparée immédiate. Pour cette raison, un développement à l'ordre 1 de $1/\hat{r}$ est calculé dans le voisinage du barycentre (r_{bar}, z_{bar}) du sous-domaine où l'élément appartient :

$$\begin{aligned}
\frac{1}{\hat{r}} &= \frac{1}{a_r r_0 + b_r z_0 + c_r} \\
&\approx \frac{1}{a_r r_{bar} + b_r z_{bar} + c_r} - \frac{a_r}{(a_r r_{bar} + b_r z_{bar} + c_r)^2} (r_0 - r_{bar}) \\
&\quad - \frac{b_r}{(a_r r_{bar} + b_r z_{bar} + c_r)^2} (z_0 - z_{bar}) \tag{3.36}
\end{aligned}$$

La matrice de rigidité élémentaire \mathbb{K}^e peut, finalement, s'écrire dans une formulation entièrement séparée :

$$\mathbb{K}^e(\boldsymbol{\alpha}) = \sum_{i=1}^4 \sum_{j=1}^4 \left(m^{ij}(\boldsymbol{\alpha}) \mathbb{M}^{ij} + m_{r_0}^{ij}(\boldsymbol{\alpha}) \mathbb{M}_{r_0}^{ij} + m_{z_0}^{ij}(\boldsymbol{\alpha}) \mathbb{M}_{z_0}^{ij} \right) + m^{55}(\boldsymbol{\alpha}) \mathbb{M}^{55} + m_{r_0}^{55}(\boldsymbol{\alpha}) \mathbb{M}_{r_0}^{55} + m_{z_0}^{55}(\boldsymbol{\alpha}) \mathbb{M}_{z_0}^{55} \quad (3.37)$$

avec

$$\blacksquare m^{ij} = \frac{c_r}{J_{\bar{\mathcal{J}}\boldsymbol{\alpha}}} \beta_i \beta_j \text{ et } \mathbb{M}^{ij} = \int_{\Omega_0^e} \mathbb{B}_i^T \mathbb{C} \mathbb{B}_j \, d\Omega_0$$

$$\blacksquare m_{r_0}^{ij} = \frac{a_r}{J_{\bar{\mathcal{J}}\boldsymbol{\alpha}}} \beta_i \beta_j \text{ et } \mathbb{M}_{r_0}^{ij} = \int_{\Omega_0^e} r_0 \mathbb{B}_i^T \mathbb{C} \mathbb{B}_j \, d\Omega_0$$

$$\blacksquare m_{z_0}^{ij} = \frac{b_r}{J_{\bar{\mathcal{J}}\boldsymbol{\alpha}}} \beta_i \beta_j \text{ et } \mathbb{M}_{z_0}^{ij} = \int_{\Omega_0^e} z_0 \mathbb{B}_i^T \mathbb{C} \mathbb{B}_j \, d\Omega_0$$

$$\blacksquare m^{55} = \frac{J_{\bar{\mathcal{J}}\boldsymbol{\alpha}}}{a_r r_{bar} + b_r z_{bar} + c_r} \text{ et } \mathbb{M}^{55} = \int_{\Omega_0^e} \mathbb{B}_5^T \mathbb{C} \mathbb{B}_5 \, d\Omega_0$$

$$\blacksquare m_{r_0}^{55} = -\frac{a_r J_{\bar{\mathcal{J}}\boldsymbol{\alpha}}}{(a_r r_{bar} + b_r z_{bar} + c_r)^2} \text{ et } \mathbb{M}_{r_0}^{55} = \int_{\Omega_0^e} (r_0 - r_{bar}) \mathbb{B}_5^T \mathbb{C} \mathbb{B}_5 \, d\Omega_0$$

$$\blacksquare m_{z_0}^{55} = -\frac{b_r J_{\bar{\mathcal{J}}\boldsymbol{\alpha}}}{(a_r r_{bar} + b_r z_{bar} + c_r)^2} \text{ et } \mathbb{M}_{z_0}^{55} = \int_{\Omega_0^e} (z_0 - z_{bar}) \mathbb{B}_5^T \mathbb{C} \mathbb{B}_5 \, d\Omega_0$$

Le calcul d'une expression à variables séparées de la matrice de rigidité est plus lourde que dans le cas 2D mais demeure accessible par le biais d'approximations. Appliquons désormais notre méthode au démonstrateur industriel présenté précédemment.

3.3.3 Partition de la géométrie

Avant d'appliquer la méthodologie que nous venons d'exposer, une division de la géométrie de la structure doit être choisie en prenant en compte le fait que les transformations géométriques associées à chaque sous-domaine doivent être affines selon les variables d'espace.

On peut voir un exemple de division de la géométrie Figure 3.10. Les sous-domaines sont principalement des triangles dont les transformations géométriques sont aisément calculables et données SEC.3.1.3. La construction d'une transformation affine dans le voisinage des congés est néanmoins plus complexe. En effet, si l'on souhaite représenter de manière précise la géométrie, l'utilisation uniquement de triangles requerrait un nombre important de sous-domaines. Cela ne peut être envisagé puisque le temps de calcul de la matrice de rigidité est directement lié au nombre de sous-domaines présents dans la partition (voir la sous-section précédente).

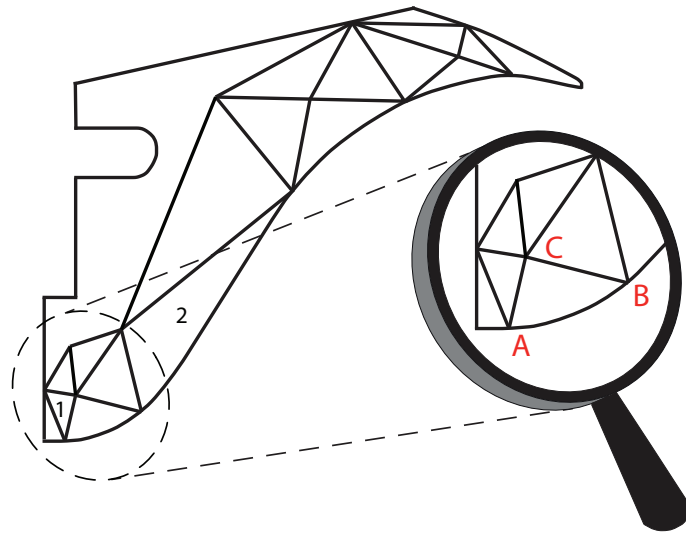


FIGURE 3.10 – Une possible possible partition de la géométrie.

De même, l'utilisation de triangles à six nœuds ne peut être considérée même s'ils représenteraient très précisément la géométrie parce que la transformation géométrique associant un triangle à six nœuds à un autre triangle à six nœuds est quadratique et non affine.

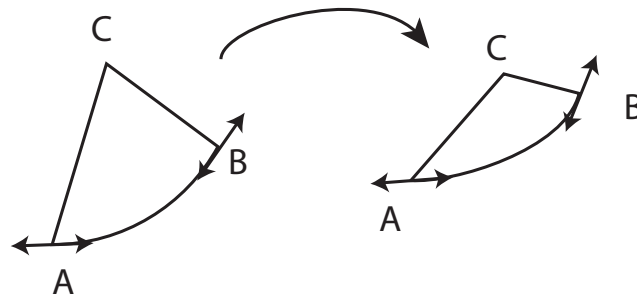


FIGURE 3.11 – La déformation du sous-domaine ABC est déterminée par le déplacement de A , B ainsi que des tangentes en ces points.

C'est pourquoi, il a été décidé d'utiliser des triangles à trois nœuds modifiés, présentant deux côtés droits et un courbe comme montré dans le zoom Figure 3.10. Pour cela, le congé, dans la structure de référence, est modélisé par un arc de cercle. Supposons alors que, pour chaque configuration géométrique, la position des points A et B ainsi que les tangentes en ces points soient connues et que le point C soit « libre » (voir Figure 3.11).

Recherchons d'abord la transformation géométrique inverse \mathcal{T}_α^{-1} (de la structure courante Ω_α vers la structure de référence Ω_0) associée aux déplacements des points A et B , qui est aussi affine.

La transformation affine inverse s'écrit comme suit :

$$\mathcal{T}_\alpha^{-1} : (r_0, z_0) = \begin{pmatrix} \lambda_r r + \gamma_r z + \delta_r \\ \lambda_z r + \gamma_z z + \delta_z \end{pmatrix} \quad (3.38)$$

Il reste à déterminer les six coefficients $\lambda_r, \gamma_r, \delta_r, \lambda_z, \gamma_z$ et δ_z . Quatre équations sont données par (3.38) car la position de A et B est connue pour les structures courante et de référence.

$$\begin{cases} r_0^A = \lambda_r r_A + \gamma_r z_A + \delta_r \\ z_0^A = \lambda_z r_A + \gamma_z z_A + \delta_z \\ r_0^B = \lambda_r r_B + \gamma_r z_B + \delta_r \\ z_0^B = \lambda_z r_B + \gamma_z z_B + \delta_z \end{cases} \quad (3.39)$$

Comme il a été mentionné précédemment, le congé est modélisé par un arc de cercle de rayon R_0 appelé par la suite (Γ_0) dans la structure de référence. Donc,

$$\forall (r_0, z_0) \in \Gamma_0, (r_0 - r_{centre})^2 + (z_0 - z_{centre})^2 = R_0^2 \quad (3.40)$$

En substituant (3.38) dans (3.40), cela conduit à :

$$(\lambda_r r + \gamma_r z + \delta_r - r_{centre})^2 + (\lambda_z r + \gamma_z z + \delta_z - z_{centre})^2 = R_0^2 \quad (3.41)$$

Dérivons cela par rapport à r :

$$\begin{aligned} (\lambda_r r + \gamma_r z + \delta_r - r_{centre}) \left(\lambda_r + \gamma_r \frac{dz}{dr} \right) \\ + (\lambda_z r + \gamma_z z + \delta_z - z_{centre}) \left(\lambda_z + \gamma_z \frac{dz}{dr} \right) = 0 \end{aligned} \quad (3.42)$$

ou encore

$$(r_0 - r_{centre}) \left(\lambda_r + \gamma_r \frac{dz}{dr} \right) + (z_0 - z_{centre}) \left(\lambda_z + \gamma_z \frac{dz}{dr} \right) = 0 \quad (3.43)$$

ce qui donne les deux dernières équations manquantes :

$$\begin{cases} (r_0^A - r_{centre}) \left(\lambda_r + \gamma_r \frac{dz}{dr} \Big|_{r=r_A} \right) \\ \quad + (z_0^A - z_{centre}) \left(\lambda_z + \gamma_z \frac{dz}{dr} \Big|_{r=r_A} \right) = 0 \\ (r_0^B - r_{centre}) \left(\lambda_r + \gamma_r \frac{dz}{dr} \Big|_{r=r_B} \right) \\ \quad + (z_0^B - z_{centre}) \left(\lambda_z + \gamma_z \frac{dz}{dr} \Big|_{r=r_B} \right) = 0 \end{cases} \quad (3.44)$$

Les six coefficients $\lambda_r, \gamma_r, \delta_r, \lambda_z, \gamma_z$ et δ_z de la transformation géométrique inverse étant déterminés, la transformation géométrique directe peut être facilement obtenue par une simple inversion.

On peut signaler l'ordre de grandeur de l'erreur commise dans l'approximation de $1/\hat{r}$ pour notre partition. Elle s'étend de 1,3% pour le sous-domaine 1, jusqu'à 6,3% pour le sous-domaine 2 (voir Figure 3.10).

3.3.4 Intégration de la méthode dans un process industriel de conception

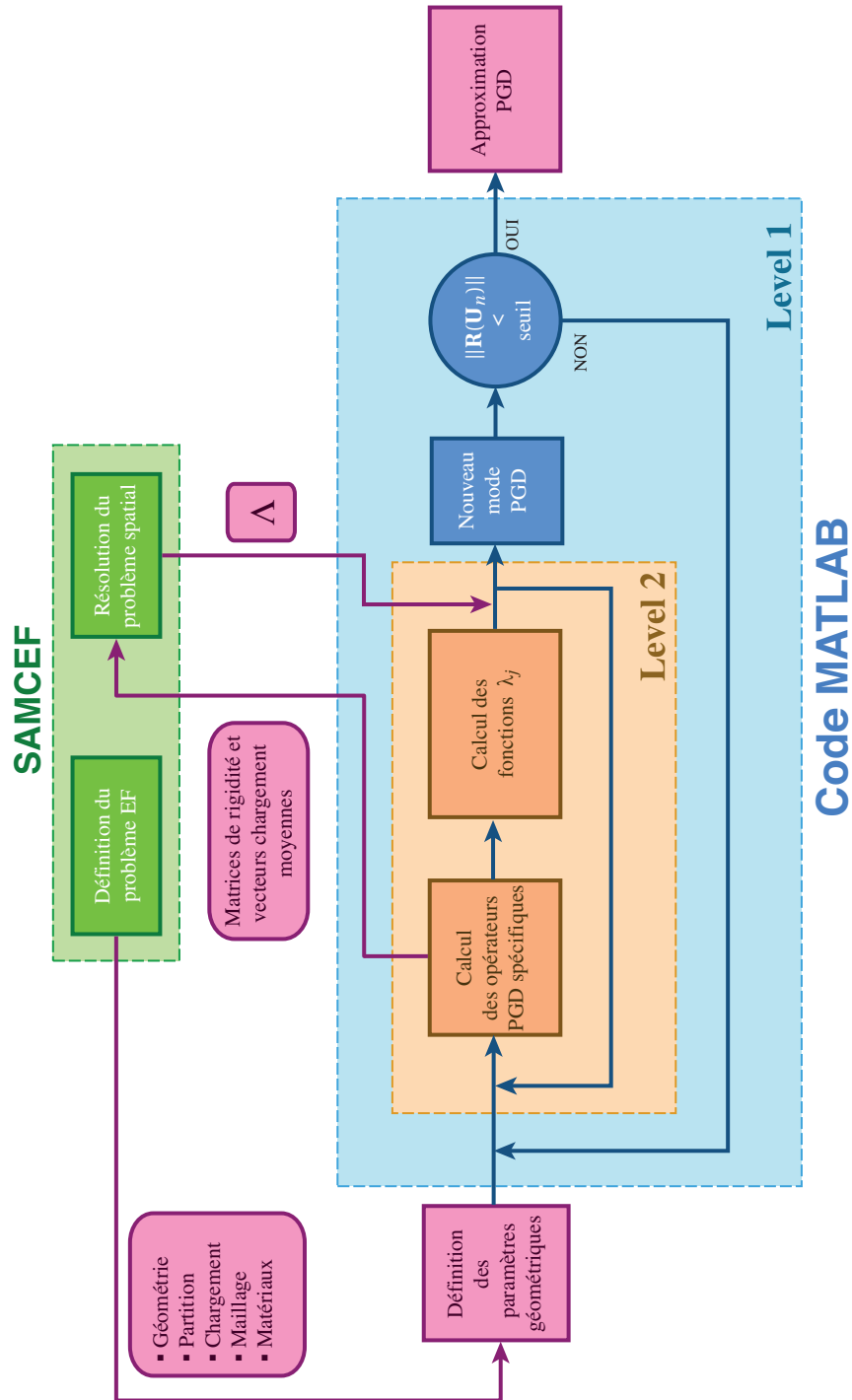


FIGURE 3.12 – Communication entre SAMCEF et le code MATLAB.

Comme il a été précédemment dit, une attention particulière a été portée sur l'intégration de la méthodologie développée dans un vrai process industriel de conception. Le principal souci rencontré avec l'approche détaillée SEC.3.3.2 est le caractère hautement intrusif de la PGD. En effet, le système linéaire défini dans (3.7) n'est pas un problème éléments finis classique pouvant être résolu par l'intermédiaire d'un logiciel commercial car la matrice de rigidité et le vecteur chargement sont moyennés sur l'espace des paramètres \mathcal{A} .

Toutefois, la méthode proposée nécessite d'être interfacée avec des logiciels commerciaux tel que SAMCEF à cause de problématiques industrielles. L'idée est d'intégrer SAMCEF dans une fonction MATLAB ad hoc qui s'occupera des opérations spécifiques liées à la PGD. Nous avons tiré avantage de la possibilité qu'offre SAMCEF de stopper le processus à une étape prédéfinie et de le relancer à cette étape après coup. Des fonctions ont été fournies par SAMTECH (groupe SIEMENS) permettant le transfert de données entre les codes commercial et PGD.

Pour résumer, d'un côté, les possibilités de SAMCEF ont été utilisées au maximum, i.e. la géométrie, la partition, les chargements, le maillage et les matériaux sont définis dans SAMCEF et la résolution du problème spatial est aussi effectuée dans SAMCEF. De l'autre, les opérateurs PGD spécifiques, à savoir les matrices de rigidité et les vecteurs chargement moyennés ainsi que les problèmes paramétriques sont calculés dans MATLAB, les fonctions fournies par SAMTECH effectuant la communication entre les deux codes. La procédure complète est détaillée dans l'ALG.6 et un schéma de la communication entre SAMCEF et MATLAB est donné Figure 3.12.

3.4 Résultats

La méthode développée a été appliquée au démonstrateur industriel présenté dans la SEC.3.3.1. Une approximation PGD du déplacement a été choisie pour un niveau d'erreur sur le résidu de 5.6% soit 17 modes (voir Figure 3.13). À partir de là, il a pu être déduit une approximation PGD de la déformation ϵ . Nous nous apercevons grâce à la Figure 3.16, qui compare la solution PGD et la solution directe obtenue par des simulations éléments finis pour deux jeux de paramètres, que les résultats obtenus montrent une bonne précision. Il pourrait être envisagé d'améliorer le taux de convergence de la méthode en adoptant une des stratégies présentées dans [Nouy, 2010]. Les trois premiers modes PGD sont donnés Figure 3.15 et les quatre premières fonctions paramétriques associées à chaque paramètre sont présentées Figure 3.14. Il est à noter que l'intervalle de chaque paramètre a été adimensionné, raison pour laquelle chaque intervalle est ramené à $[-1; 1]$. On remarque que les paramètres α_1 et α_2 ont une influence sensiblement plus faible que les paramètres α_3 et α_4 sur le résultat en comparant les pentes des courbes tracées sur la Figure 3.14.

Une question demeure : comment explorer efficacement les abaques virtuels construits ? En effet, on doit garder à l'esprit que les abaques virtuels sont, avant tout de

Algorithme 6 : SAMCEF-PGD algorithm

Entrées : Une suite de paramètres $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m) \in \mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_m$

- 1 **Définition du problème avec SAMCEF;**
- 2 Définition de la géométrie, des matériaux, partitions, chargements et maillage;
- 3 **Lancement PGD-algorithme dans MATLAB;**
- 4 Chargement des données à partir de SAMCEF
- 5 **Level 1 : Algorithme greedy;**
- 6 **tant que** $\|\mathbf{R}(\mathbf{U}_n)\| > \text{critère}$ (avec $\mathbf{R}(\mathbf{u}_n)$ étant le résidu) **faire**
- 7 **Level 2 : Algorithme de point-fixe;**
- 8 Initialisation du $(m + 1)$ -uplet $(\lambda^1, \dots, \lambda^m, \Lambda)$;
- 9 **pour** $k = 1$ à k_{max} **faire**
- 10 Calcul des matrices de rigidité élémentaires moyennées et des vecteurs chargement;
- 11 **pour** $j = 1$ à m **faire**
- 12 Résolution du problème paramétrique associé à λ^j ;
- 13 Transfert des matrices de rigidité et vecteurs de chargement à SAMCEF;
- 14 Résolution du problème spatial en utilisant SAMCEF;
- 15 Actualisation des fonctions $(\lambda_i^1, \dots, \lambda_i^m)_{i \in \{1..n\}}$;
- 16 $\mathbf{U}_n \leftarrow \mathbf{U}_{n-1} + \prod_{r=1}^m \lambda^r \Lambda$;
- 17 $n \leftarrow n + 1$;

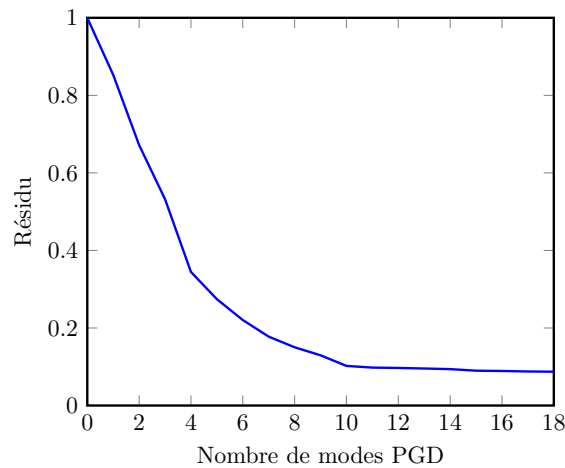


FIGURE 3.13 – Courbe de convergence.

chose, un outil d'aide à la décision et que l'utilisateur final doit pouvoir les parcourir rapidement et aisément. Or, pour le moment, ce n'est pas le cas car, dans l'état actuel des choses, il ne peut que particulariser sa solution avec un jeu de paramètres donnés. L'idée serait de construire des abaques graphiques qui permettraient l'exploration ra-

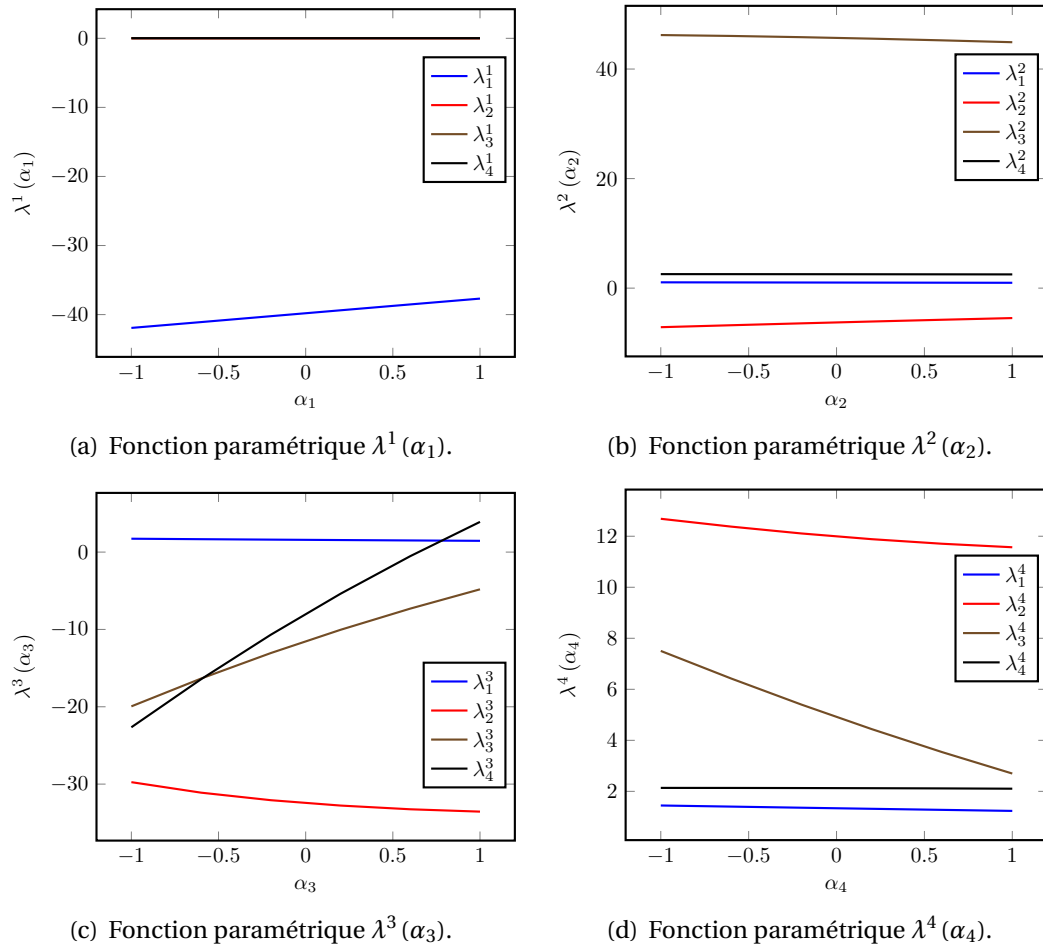


FIGURE 3.14 – Fonctions paramétriques associées aux quatre premiers modes PGD.

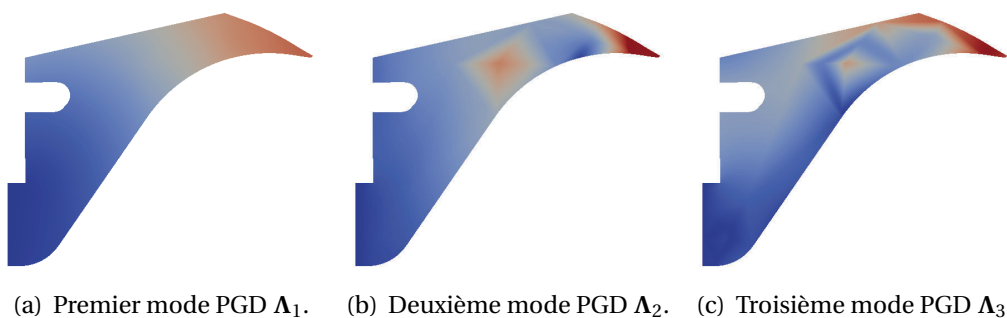


FIGURE 3.15 – Trois premiers modes PGD.

pide de la solution paramétrique. C'est pourquoi, il a été décidé d'exploiter le plug-in PXDMF pour PARAVIEW développé à l'École Centrale de Nantes dédié à la visualisation de champs à variables séparées [Bognet *et al.*, 2012b]. Étant donnés les modes

associés à chaque champ considéré (déplacement, déformation, contrainte...), PARAVIEW reconstruit les champs complets en temps réel en effectuant des sommes et des multiplications. Il est alors possible de visualiser l'évolution de ces champs en faisant varier chaque paramètre géométrique à l'aide de curseurs.

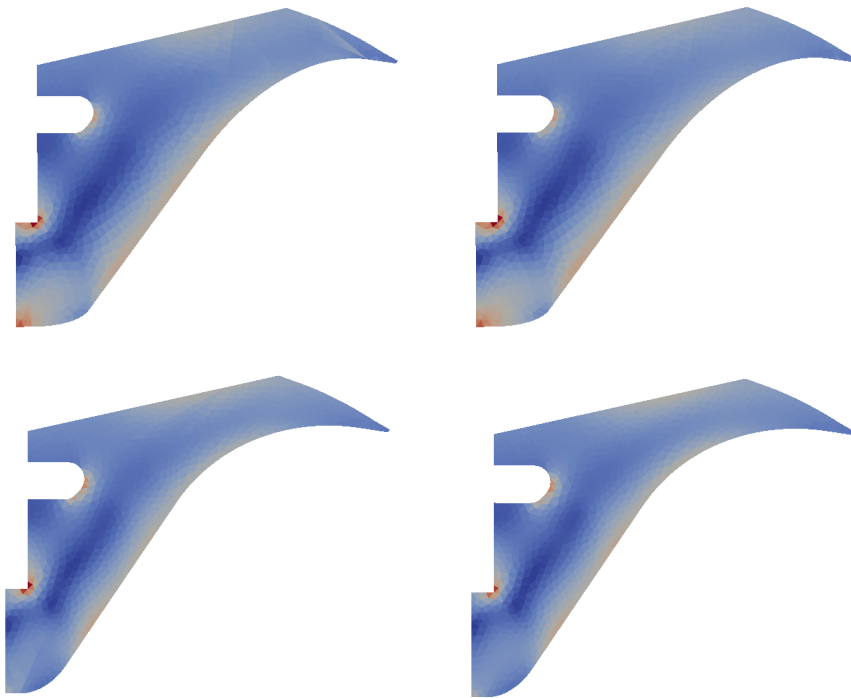


FIGURE 3.16 – Comparaison entre les solutions (déformation) obtenues avec 17 modes PGD (à gauche) et celles obtenues par les éléments finis (à droite) pour deux jeux de paramètres

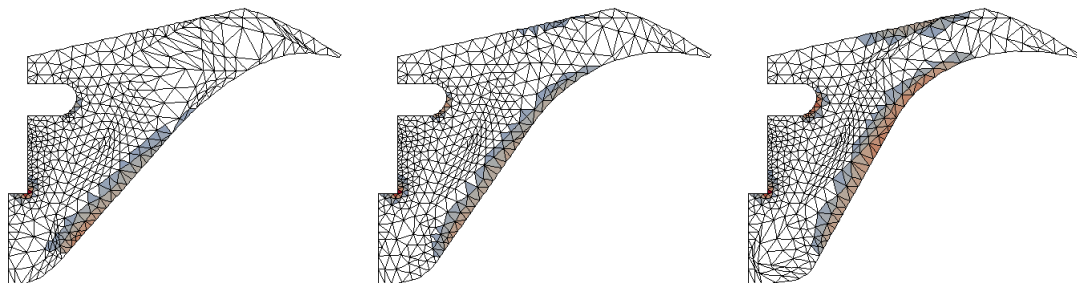


FIGURE 3.17 – Exemple d'abaque virtuel associé aux zones plastiques pour trois jeux de paramètres différents.

Un critère de conception pouvant intervenir est la taille des zones plastiques. Pour donner une estimation de ces zones, des calculs linéaires élastiques ont été effectués. Les zones plastiques ont, alors, été évaluées en définissant un seuil et en ne considérant uniquement les éléments dont la déformation était supérieure à ce seuil. La Figure 3.17 montre un exemple d'exploration à l'aide du *plug-in* PXDMF de l'abaque virtuel créé. Il est important de noter que, l'abaque virtuel ayant été construit, son exploration se fait en temps réel par l'utilisateur. Cette stratégie pourrait être étendue à un véritable calcul non-linéaire en utilisant le cadre proposé dans [Néron *et al.*, 2015].

Chapitre 4

Discrete Proper Generalized Decomposition

Une nouvelle extension de la PGD est présentée ici. En restant le plus général possible, c'est-à-dire sans préjuger de la nature des paramètres en jeu (matériaux, géométriques, etc.), on propose de prendre en considération non seulement des variables continues au sein de la méthode PGD mais aussi des variables discrètes. Cela a l'avantage de pouvoir modéliser des situations intrinsèquement discrètes telles que des cas de chargement ou des configurations de matériaux. On montrera que, sur le principe, la PGD demeure inchangée, seules quelques modifications mineures étant apportées. Cette méthode est testée à la fois sur deux exemples académiques et sur un nouveau démonstrateur industriel comportant de fortes non-linéarités. Un exemple d'abaque virtuel est également donné en fin de chapitre.

4.1 Nouveau démonstrateur

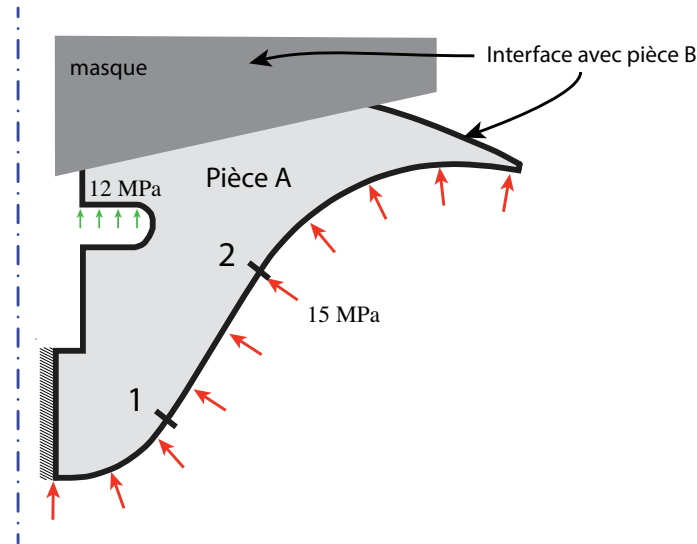


FIGURE 4.1 – Nouveau démonstrateur industriel. Un masque est appliqué sur une partie de la nouvelle géométrie.

À la vue des résultats prometteurs obtenus dans le chapitre précédent, il a été décidé, conjointement avec AIRBUS Defence & Space, de s'intéresser, dans un second temps, à un démonstrateur industriel plus complet. La géométrie de celui-ci est très similaire à celle de l'ancien démonstrateur. Néanmoins, pour des raisons de confidentialité, un masque a été appliqué sur la partie de la géométrie modifiée (voir Figure 4.1). Il s'agit toujours d'une pièce métallique (appelée pièce A), étudiée en élasticité linéaire, et les paramètres géométriques demeurent inchangés. De plus, la structure s'interface, à présent, au niveau de sa frontière supérieure, avec une pièce B. L'interface des pièces A et B est fixée et ne dépend pas de paramètres géométriques.

Plusieurs précisions sont à apporter concernant la pièce B. Premièrement, elle présente un certain nombre de non-linéarités fortes telles que :

- grands déplacements
- grandes déformations
- plasticité
- hyperélasticité

Deuxièmement, la pièce B est composée, entre autres, d'un matériau composite pouvant présenter un certain nombre de configurations différentes (nombre d'empile-

ments, variations d'épaisseur de pli, etc.). Six configurations représentatives de l'étendue des situations possibles ont été sélectionnées pour notre étude. Elles sont définies et construites à l'aide d'un outil métier développé par AIRBUS Defence & Space. On précise qu'aucun paramètre géométrique n'est considéré pour la pièce B.

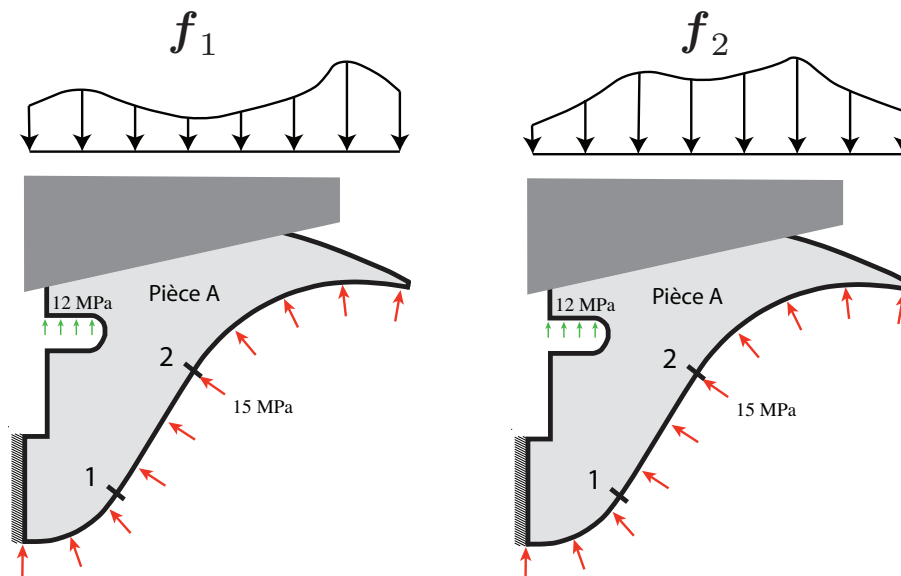


FIGURE 4.2 – Deux cas de chargement différents issus d'un calcul non-linéaire sur la structure complète (pièce A + pièce B) pour deux configurations de composite données. Ces chargements seront appliqués à la pièce A lors de son dimensionnement.

L'idée est, alors, de procéder à une série de simulations non-linéaires complètes à l'aide de SAMCEF pour plusieurs valeurs de paramètres géométriques et configurations de composite. Ensuite, un chargement, déduit de l'état de contraintes à l'interface des pièces A et B, est calculé et appliqué à la pièce A (voir Figure 4.2). On se retrouve, par conséquent, avec un chargement dépendant à la fois des paramètres géométriques et de la configuration composite. Cela ne présente pas de difficulté particulière puisqu'il peut toujours être décomposé grâce à une technique de séparation de variables (SVD, POD, PGD, etc.).

On construit, ensuite, un modèle PGD de la pièce A pour le chargement considéré comme pour le chapitre précédent. Or, le chargement, dépendant directement des configurations du composite, ne peut être pris tel quel comme paramètre au sein de la méthode PGD puisqu'il est, en partie, de nature discret (sa dépendance selon les paramètres géométriques est, elle, continue). En effet, nous n'avons pas encore d'outil à disposition permettant la prise en considération de ce type de paramètres, la PGD portant sur des paramètres continus. C'est pourquoi, on propose, dans ce chapitre, une extension de la méthode PGD aux paramètres discrets.

4.2 Paramètres discrets

Lors de la conception d'une structure industrielle, il arrive couramment de tester cette structure pour un ensemble de cas de chargement $\mathbf{f} = (f_1, \dots, f_{n_c})$. Or, si l'on souhaite choisir comme paramètre ces configurations de chargement, il peut être difficile, voire impossible, de représenter ce paramètre par une variable continue. Sur la Figure 4.3, deux cas de chargement sont représentés pour une structure donnée. On remarque alors le paramètre « chargement » est de nature intrinsèquement discrète.

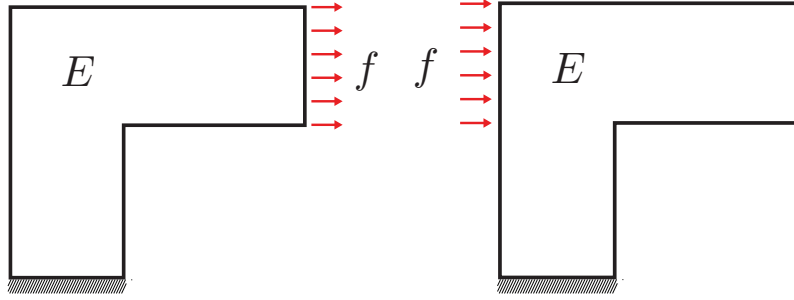


FIGURE 4.3 – Deux cas de chargement d'une même structure. Le chargement est un bon exemple de paramètre discret.

Or, jusqu'à maintenant, la PGD a été uniquement considérée dans le cadre de paramètres continus évoluant dans des intervalles de valeurs données. L'idée développée dans ce chapitre est d'étendre la méthode PGD aux paramètres discrets.

Dans le contexte PGD, il peut être envisagé, dans un premier temps, de construire une approximation, et donc un abaque virtuel, pour chaque configuration, i.e. de résoudre le problème :

Problème 4.1. Pour tout $p \in \mathcal{P} = \llbracket 1, n_c \rrbracket$, trouver $\mathbf{u}_p \in \mathcal{V}$ tel que

$$\forall \mathbf{v}_p \in \mathcal{V}, \quad \int_{\Omega} \mathbb{C} \varepsilon(\mathbf{u}_p) : \varepsilon(\mathbf{v}_p) \, d\Omega = \int_{\partial_d \Omega} \mathbf{f}_p \cdot \mathbf{v}_p \, dS$$

Si l'on note n_{PGD}^p le nombre de modes PGD nécessaires pour représenter la solution associée au cas de chargement \mathbf{f}_p , on se retrouve avec un total de $\sum_{p=1}^{n_c} n_{PGD}^p$ modes pour le problème complet.

Néanmoins, puisque tous ces problèmes ne diffèrent que par leur chargement, il est légitime de penser que leurs solutions ont en commun un certain nombre de propriétés et, par conséquent, aussi des modes PGD. Ainsi, le nombre de modes nécessaires pour représenter la solution du problème complet peut être très probablement

réduit. On pourra alors étendre cette approche à d'autres problèmes rencontrant des paramètres discrets comme pour le cas d'une structure où plusieurs matériaux différents doivent être testés, le module d'Young et le coefficient de Poisson étant alors discrets.

4.3 Discrete Proper Generalized Decomposition ou dPGD

L'idée est d'appliquer la méthodologie PGD mais, comme il n'est plus possible d'intégrer sur l'espace des configurations, on vient sommer ces configurations. Ainsi pour un problème mécanique d'élasticité linéaire, cela revient à écrire :

Problème 4.2. *Trouver $\mathbf{u} \in \mathcal{C} \otimes \mathcal{V}$ tel que*

$$\forall \mathbf{v} \in \mathcal{C} \otimes \mathcal{V}, \quad \sum_{p=1}^{n_c} \int_{\Omega} \mathbb{C} \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) \, d\Omega = \sum_{p=1}^{n_c} \int_{\partial_d \Omega} \mathbf{f}_p \cdot \mathbf{v} \, dS$$

où $\mathcal{C} = \mathbb{R}^{\text{card}(\mathcal{P})}$ et le déplacement \mathbf{u} est recherché sous forme PGD :

$$\mathbf{u}(p, \mathbf{X}) \approx \mathbf{u}_n(p, \mathbf{X}) = \sum_{i=1}^n \gamma(p) \boldsymbol{\Lambda}(\mathbf{X})$$

En posant, par souci de généralité de la forme des équations :

$$\int_{\mathcal{P}} \bullet \, d\mathcal{P} = \sum_{p=1}^{n_c} \bullet \Delta_p \text{ avec } \Delta_p = 1^1 \quad (4.1)$$

on peut réécrire le Problème 4.2 sous une formulation PGD classique :

Problème 4.3. *Trouver $\mathbf{u} \in \mathcal{C} \otimes \mathcal{V}$ tel que*

$$\forall \mathbf{v} \in \mathcal{C} \otimes \mathcal{V}, \quad \int_{\mathcal{P}} \int_{\Omega} \mathbb{C} \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) \, d\Omega \, d\mathcal{P} = \int_{\mathcal{P}} \int_{\partial_d \Omega} \mathbf{f}_p \cdot \mathbf{v} \, dS \, d\mathcal{P}$$

En déroulant la méthodologie PGD, on suppose connu \mathbf{u}_n , le déplacement à l'itération n et l'on cherche à calculer le nouveau couple $(\gamma, \boldsymbol{\Lambda})$. On écrit :

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \gamma \boldsymbol{\Lambda} \quad (4.2)$$

1. On discutera SEC.4.4.1.2 sur ce qu'implique un tel choix de pondération.

On choisit la fonction test $v \in T_{u_{n+1}}(\mathcal{C} \otimes \mathcal{V})$:

$$v = \gamma \Lambda^* + \gamma^* \Lambda \quad (4.3)$$

avec $\Lambda^* \in \mathcal{V}$ et $\gamma^* \in \mathcal{C}$.

Et on insère le tout dans le Problème 4.2. On obtient alors :

Problème 4.4. Trouver $(\gamma, \Lambda) \in \mathcal{C} \times \mathcal{V}$ tels que

$$\forall (\gamma^*, \Lambda^*) \in \mathcal{C} \times \mathcal{V},$$

$$\int_{\mathcal{P}} \int_{\Omega} \mathbb{C} \nabla_s \left(\sum_{i=1}^n \gamma_i \Lambda + \gamma \Lambda \right) : \nabla_s (\gamma \Lambda^* + \gamma^* \Lambda) d\Omega d\mathcal{P} = \int_{\mathcal{P}} \int_{\partial_d \Omega} \mathbf{f}_p \cdot (\gamma \Lambda^* + \gamma^* \Lambda) dS d\mathcal{P}$$

Après discrétisation du problème en espace dans une approche éléments finis, cela revient à résoudre :

Problème 4.5. Trouver $(\gamma, \Lambda) \in \mathcal{C} \times \mathcal{V}$ tels que

$$\forall (\gamma^*, \Lambda^*) \in \mathcal{C} \times \mathcal{V},$$

$$\int_{\mathcal{P}} \left(\sum_{i=1}^n \gamma_i \Lambda_i + \gamma \Lambda \right)^T \mathbb{K} (\gamma \Lambda^* + \gamma^* \Lambda) d\mathcal{P} = \int_{\mathcal{P}} \mathbf{F}_p^T (\gamma \Lambda^* + \gamma^* \Lambda) d\mathcal{P}$$

En appliquant le principe de stationnarité, cela revient à résoudre simultanément les deux équations suivantes :

$$\forall p \in \mathcal{P}, \quad (\Lambda^T \mathbb{K} \Lambda) \gamma(p) = \mathbf{F}_p^T \Lambda - \sum_{i=1}^n (\Lambda_i^T \mathbb{K} \Lambda) \gamma_i(p) \quad (4.4)$$

et

$$\left(\int_{\mathcal{P}} \gamma^2 d\mathcal{P} \right) \mathbb{K} \Lambda = \int_{\mathcal{P}} \gamma \mathbf{F}_p d\mathcal{P} - \sum_{i=1}^n \left(\int_{\mathcal{P}} \gamma \gamma_i d\mathcal{P} \right) \mathbb{K} \Lambda_i \quad (4.5)$$

On se rend alors compte qu'aucune difficulté supplémentaire n'est apparue dans la méthode due aux paramètres discrets. Elle reste identique à ceci près que l'intégration est remplacée par une sommation.

4.4 Approche hybride

Dans une optique d'aide au dimensionnement de structures, il serait intéressant de développer une approche hybride, à savoir prendre en considération à la fois des paramètres continus (paramètres matériau, géométriques, etc.) et discrets (cas de chargement). À la vue de ce qui a été écrit précédemment sur les paramètres discrets, cela ne présente pas de difficultés particulières. Dans ce contexte, le Problème 4.2 devient :

Problème 4.6. Trouver $\mathbf{u} \in \mathcal{F} \otimes \mathcal{C} \otimes \mathcal{V}$ tel que

$$\forall \mathbf{v} \in \mathcal{F} \otimes \mathcal{C} \otimes \mathcal{V}, \quad \int_{\mathcal{P}} \int_{\mathcal{A}} \int_{\Omega} \mathbb{C} \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) \, d\Omega \, d\mathcal{A} \, d\mathcal{P} = \int_{\mathcal{P}} \int_{\mathcal{A}} \int_{\partial_d \Omega} \mathbf{f}_p(\boldsymbol{\alpha}) \cdot \mathbf{v} \, dS \, d\mathcal{A} \, d\mathcal{P}$$

Il est à noter que l'on s'est placé dans le cas le plus général possible. On ne préjuge pas de l'indépendance du chargement \mathbf{f}_p selon les paramètres continus $\boldsymbol{\alpha} \in \mathcal{A}$. De même, aucun *a priori* n'est donné sur la nature de ces paramètres. Selon le type de paramètres choisis, on suivra des stratégies PGD différentes. Mais pour l'instant, en toute généralité, nous recherchons le déplacement $\mathbf{u}(\boldsymbol{\alpha}, p, \mathbf{X})$ sous forme séparée :

$$\mathbf{u}(\boldsymbol{\alpha}, p, \mathbf{X}) = \sum_{i=1}^n \prod_{j=1}^m \lambda_i^j(\alpha_j) \gamma_i(p) \Lambda_i(\mathbf{X})$$

À l'étape d'enrichissement $n+1$, on a :

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \prod_{j=1}^m \lambda^j(\alpha_j) \gamma \Lambda \quad (4.6)$$

où $(\lambda^1, \dots, \lambda^m, \gamma, \Lambda) \in \mathcal{F}_1 \times \dots \times \mathcal{F}_m \times \mathcal{C} \times \mathcal{V}$ est le nouveau $(m+2)$ -uplet recherché.

On choisit la fonction test $\mathbf{v} \in T_{\mathbf{u}_{n+1}}(\mathcal{F} \otimes \mathcal{C} \otimes \mathcal{V})$ suivante :

$$\mathbf{v} = \prod_{j=1}^m \lambda^j \gamma \Lambda^* + \sum_{i=1}^m \lambda^{i*} \prod_{\substack{j=1 \\ j \neq i}}^m \lambda^j \gamma \Lambda + \prod_{j=1}^m \lambda^j \gamma^* \Lambda \quad (4.7)$$

avec $\Lambda^* \in \mathcal{V}$, $\gamma^* \in \mathcal{C}$ et $\lambda^i \in \mathcal{F}_i$ pour $i \in \llbracket 1, m \rrbracket$.

Cela aboutit *in fine* à la résolution du système d'équations suivant après discrétisation en espace :

$\forall p \in \mathcal{P}$,

$$\begin{aligned} \left\{ \Lambda^T \left[\int_{\mathcal{A}} \prod_{i=1}^m (\lambda^i(\alpha_i))^2 \mathbb{K}(\alpha) \, d\mathcal{A} \right] \Lambda \right\} \gamma(p) &= \left[\int_{\mathcal{A}} \prod_{i=1}^m \lambda^i(\alpha_i) \mathbf{F}_p(\alpha)^T \, d\mathcal{A} \right] \Lambda - \\ &\quad \sum_{i=1}^n \left\{ \Lambda_i^T \left[\int_{\mathcal{A}} \prod_{i=1}^m \lambda^i(\alpha_i) \lambda_k^i(\alpha_i) \mathbb{K}(\alpha) \, d\mathcal{A} \right] \Lambda \right\} \gamma_i(p) \end{aligned} \quad (4.8)$$

$\forall r \in \llbracket 1, m \rrbracket$,

$$\begin{aligned} \left(\int_{\mathcal{P}} \gamma^2 \, d\mathcal{P} \right) \left\{ \Lambda^T \left[\int_{\bar{\mathcal{A}}_r} \prod_{\substack{i=1 \\ i \neq r}}^m (\lambda^i(\alpha_i))^2 \mathbb{K}(\alpha) \, d\bar{\mathcal{A}}_r \right] \Lambda \right\} \lambda^r(\alpha_r) &= \\ \left[\int_{\bar{\mathcal{A}}_r} \prod_{\substack{i=1 \\ i \neq r}}^m \lambda^i(\alpha_i) \left(\int_{\mathcal{P}} \gamma \mathbf{F}_p^T(\alpha) \, d\mathcal{P} \right) \, d\bar{\mathcal{A}}_r \right] \Lambda & \\ - \sum_{k=1}^{n-1} \left\{ \Lambda_k^T \left[\int_{\bar{\mathcal{A}}_r} \prod_{\substack{i=1 \\ i \neq r}}^m \lambda^i(\alpha_i) \lambda_k^i(\alpha_i) \mathbb{K}(\alpha) \, d\bar{\mathcal{A}}_r \right] \Lambda \right\} \left(\int_{\mathcal{P}} \gamma \gamma_k \, d\mathcal{P} \right) \lambda_k^r(\alpha_r) \end{aligned} \quad (4.9)$$

avec $\bar{\mathcal{A}}_r = \mathcal{A}_1 \times \dots \times \mathcal{A}_{r-1} \times \mathcal{A}_{r+1} \times \dots \times \mathcal{A}_m$

$$\begin{aligned} \left(\int_{\mathcal{P}} \gamma^2 \, d\mathcal{P} \right) \left[\int_{\mathcal{A}} \prod_{i=1}^m (\lambda^i(\alpha_i))^2 \mathbb{K}(\alpha) \, d\mathcal{A} \right] \Lambda &= \int_{\mathcal{A}} \prod_{i=1}^m \lambda^i(\alpha_i) \left(\int_{\mathcal{P}} \gamma^2 \mathbf{F}_p(\alpha) \, d\mathcal{P} \right) \, d\mathcal{A} \\ - \sum_{k=1}^{n-1} \left[\int_{\mathcal{A}} \prod_{i=1}^m \lambda^i(\alpha_i) \lambda_k^i(\alpha_i) \mathbb{K}(\alpha) \, d\mathcal{A} \right] \left(\int_{\mathcal{P}} \gamma \gamma_k \, d\mathcal{P} \right) \Lambda_k \end{aligned} \quad (4.10)$$

On pourra, pour faciliter l'intégration du chargement selon les paramètres continus et discrets, procéder à une SVD, POD ou PGD sur ledit chargement. Il s'agit d'une technique très couramment utilisée en réduction de modèle, à savoir décomposer le chargement sous forme séparée dans une étape préliminaire.

On note que l'approche hybride est légèrement plus complexe que l'approche classique mais elle n'est pas plus compliquée. En effet, seules quelques modifications des équations de la PGD définies dans le premier chapitre ont été apportées de telle sorte que l'ALG.3 demeure d'actualité.

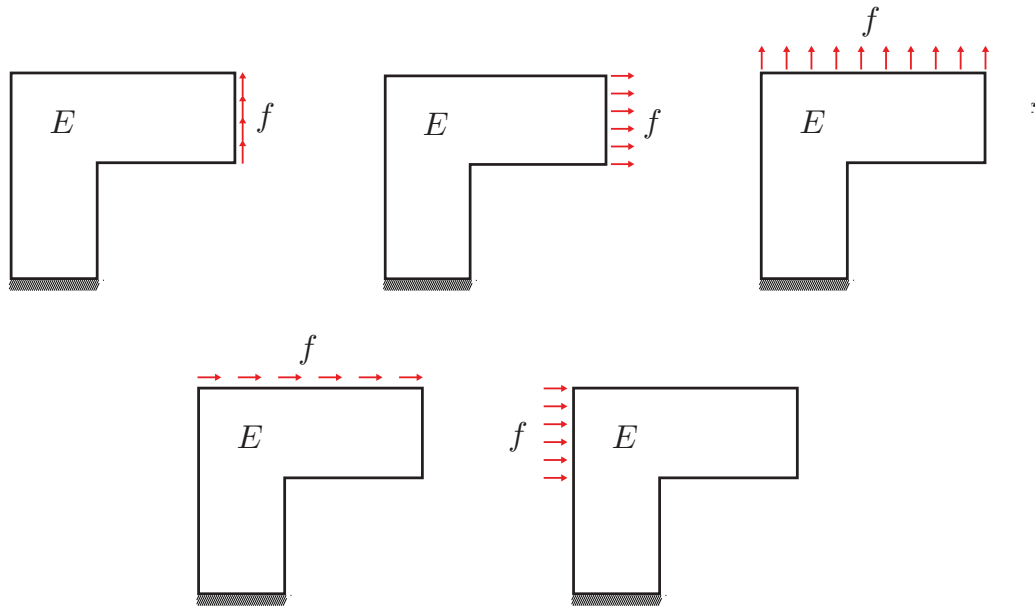


FIGURE 4.4 – Cinq configurations de chargement.

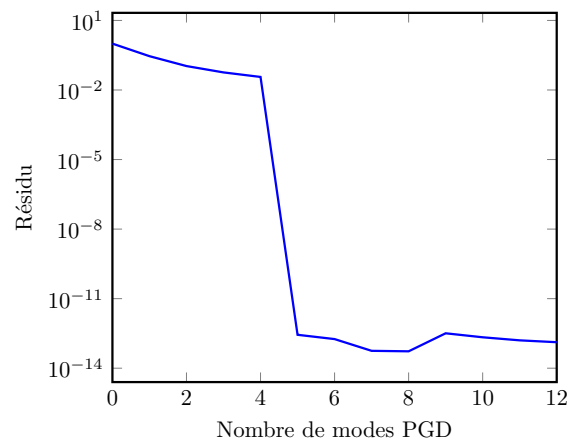


FIGURE 4.5 – Courbe de convergence.

4.4.1 Exemples académiques

4.4.1.1 Exemple 1

Illustrons cela sur notre fil rouge, à savoir la structure en forme de Γ . Nous avons deux paramètres, le module de Young $E \in [50 \text{ GPa}, 1000 \text{ GPa}]$ considéré continu et le chargement qui, lui, est discret (cinq cas de chargement, illustrés sur la Figure 4.4, sont envisagés) et de module 1000 N. Le déplacement $u(E, p, \mathbf{X})$ est recherché sous forme

PGD. Il s'écrit alors de la manière suivante :

$$\mathbf{u}(E, p, \mathbf{X}) = \sum_{i=1}^m \lambda_i(E) \gamma_i(p) \Lambda_i(\mathbf{X}) \quad (4.11)$$

avec $p \in \mathcal{P} = [1, 5]$.

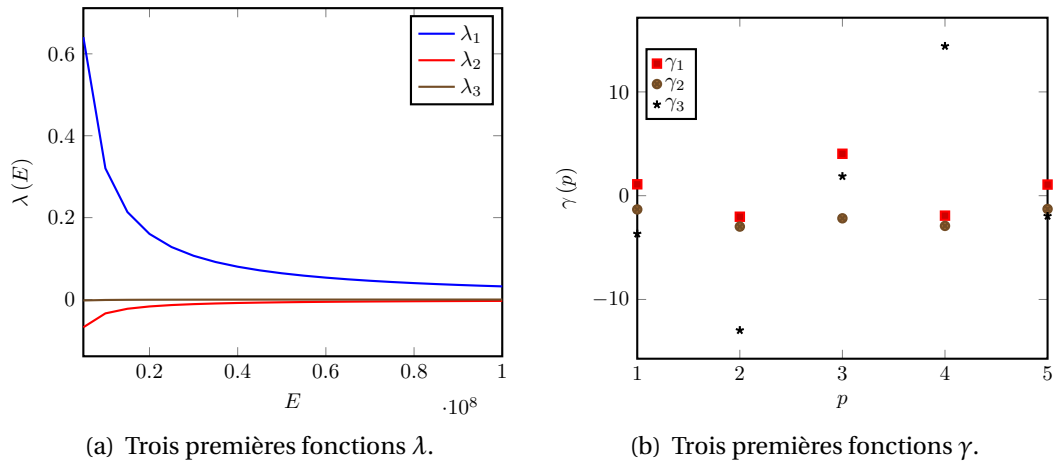


FIGURE 4.6 – Trois premières fonctions λ et γ .

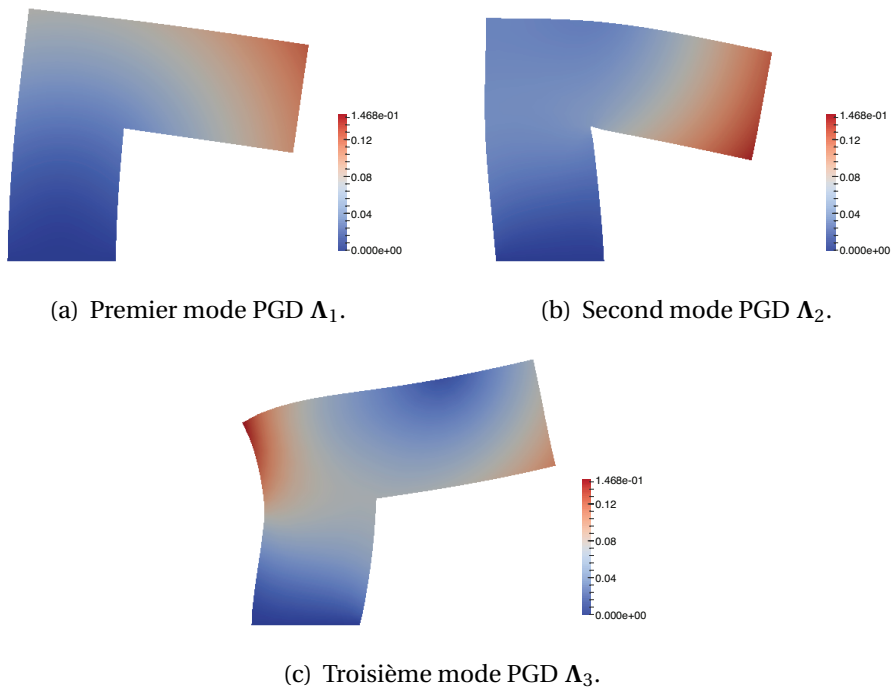


FIGURE 4.7 – Trois premiers modes PGD.

On peut voir la convergence rapide de la méthode sur la Figure 4.5. Avec 5 modes, l'erreur sur le résidu est de l'ordre de 10^{-13} , ce qui signifie que le nombre de modes PGD nécessaires est, au plus, égal à celui de la méthode consistant à construire un modèle PGD pour chaque configuration de chargement. En effet, un seul mode PGD est nécessaire pour représenter chaque cas. On se retrouvera, alors, au final avec 5 modes PGD. Sur les Figures 4.6 et 4.7 sont tracés les trois premières fonctions λ et γ ainsi que les trois premiers modes PGD.

4.4.1.2 Exemple 2

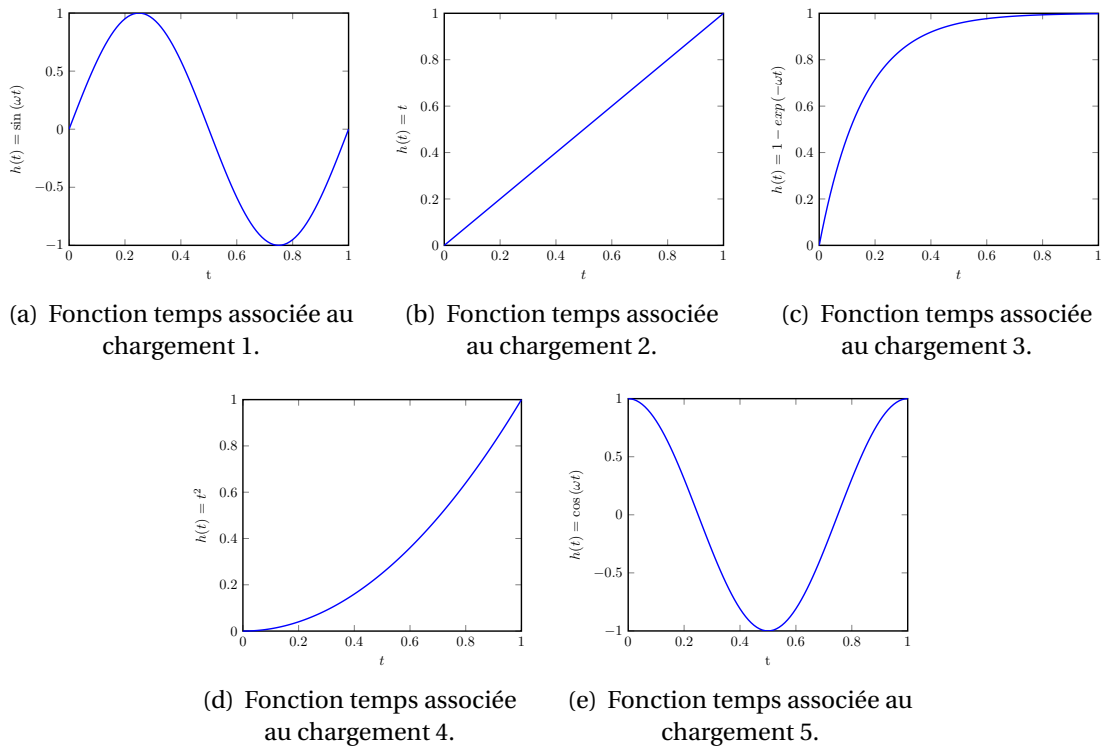


FIGURE 4.8 – Fonctions dépendant du temps associées à chaque configuration de chargement.

Compliquons, à présent, l'Exemple 1 afin de le rendre un peu plus consistant car la convergence trop rapide pour ce dernier empêche toute réelle interprétation des résultats. Le nombre de modes, issus de la dPGD, et le nombre de modes, obtenus par la construction d'un modèle PGD pour chaque cas de chargement, sont égaux. Pour cela, nous allons reprendre la structure définie dans le CHAP.1 (voir Figure 1.5) et conserver les cinq cas de chargement définis sur la Figure 4.4. De plus, à chaque cas de chargement, nous ajoutons une fonction temps propre (voir Figure 4.8). L'étude reste, néanmoins, stationnaire. La solution recherchée (le déplacement) dépend, alors,

de quatre paramètres continus (temps, E_1 , E_2 , E_3) et d'un paramètre discret (cas de chargement). Le déplacement s'écrit, dans ce cas, sous la forme PGD suivante :

$$\mathbf{u}(E_1, E_2, E_3, p, t, \mathbf{X}) = \sum_{i=1}^m \lambda_i^1(E_1) \lambda_i^2(E_2) \lambda_i^3(E_3) \gamma_i(p) \delta_i(t) \mathbf{\Lambda}_i(\mathbf{X}) \quad (4.12)$$

Une solution de référence éléments finis a été calculée pour tous les jeux de paramètres et nous avons, dans un premier temps, construit un modèle PGD pour chaque cas de chargement (on appellera cette approche « PGD classique » par la suite) et déterminé le nombre de modes nécessaires pour obtenir une erreur de l'ordre de 5 %.

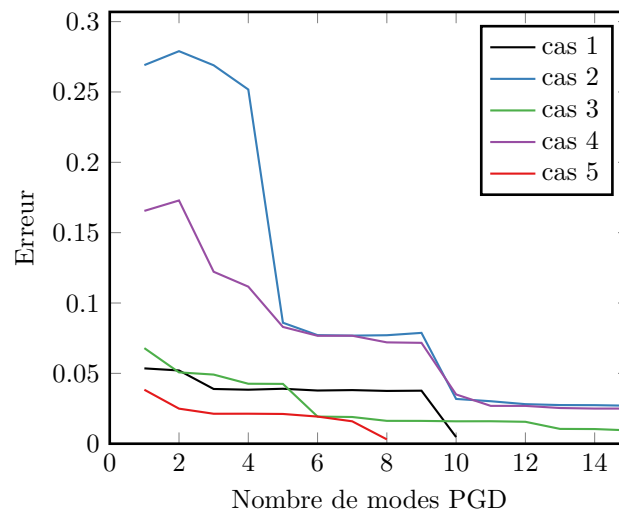


FIGURE 4.9 – Courbes de convergence associées à chaque configuration de chargement (PGD classique).

En étudiant les courbes de convergence tracées sur le Figure 4.9, on s'aperçoit que, pour le niveau de précision souhaitée, 3 modes PGD sont nécessaires dans les cas de chargements 1 et 3, 10 modes pour les cas 2 et 4, un seul pour le cas 5. Ainsi, il faut un total de 27 modes pour représenter, avec une fidélité de l'ordre de 95 %, la solution pour les cinq configurations de chargement en utilisant la méthode PGD classique.

La même démarche a été adoptée pour la dPGD. Les courbes de convergence associées à chaque configuration de chargement ont été tracées sur la Figure 4.10. Plusieurs points sont à noter.

Tout d'abord, concernant les cas de chargement 1 et 5, on remarque un niveau d'erreur très élevé (proche de 1) pour les premiers modes contrairement aux autres cas de chargement (erreur entre 0,2 et 0,41). Pour comprendre cela, regardons les premières fonctions δ_i représentées sur la Figure 4.11. En effet, la fonction δ_1 est très similaire à la fonction de temps associée au second cas de chargement et le niveau d'erreur correspondant à ce cas est, alors, le plus faible. En outre, cette fonction est très éloignée des fonctions sinus et cosinus des cas 1 et 5 (ces fonctions alternent, dans le temps, le

signe du chargement). Ce n'est qu'à partir des modes 5 et 6 qu'apparaissent des fonctions sinusoïdales et que les erreurs correspondant au cas 1 et 5 chutent fortement.

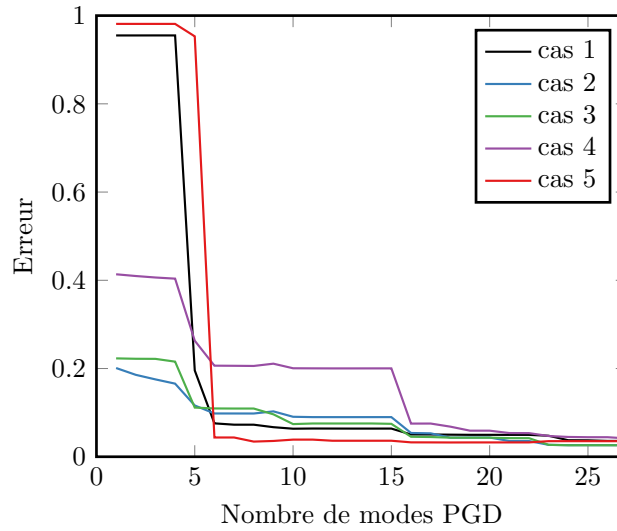


FIGURE 4.10 – Courbes de convergence associées à chaque configuration de chargement (dPGD).

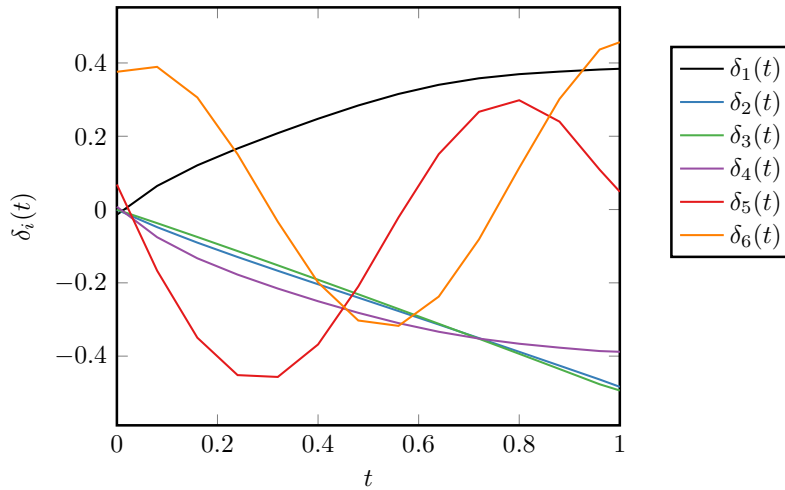


FIGURE 4.11 – Six premières fonctions δ_i .

Le second point porte sur le nombre de modes PGD nécessaires. Pour cela, on doit déterminer le cas de chargement limitant, c'est-à-dire celui dont le nombre de modes PGD correspondant à une erreur de l'ordre de 5% est le plus grand. Il s'agit, ici, du cas de chargement 4 (21 modes). Nous avons donc gagné, en recourant à la dPGD, 6 modes par rapport à la PGD classique, ce qui correspond à un gain de 22%. Ce gain, sans être

négligeable, est inférieur à celui escompté. En effet, on connaissait *a priori* un mino- rant du nombre de modes nécessaires en étudiant les résultats obtenus à partir de la PGD classique, à savoir 10 (nombre de modes des situations 2 et 4). Cela laissait une marge confortable en comparaison du nombre total de modes qui était de 27 (le gain maximal théorique est, donc, de 63%). Si l'on décortique les résultats donnés par la Figure 4.10, on note que 16 modes sont nécessaires pour les cas 1, 2 et 3 et seulement 6 pour le cas 5. De plus, on peut légitimement penser que toutes les situations n'ont pas la valeur, et que certaines prévalent sur d'autres. La PGD classique nous dit que les cas 2 et 4 nécessitent un nombre de modes supérieur par rapport aux autres confi- gurations. Leur poids dans la décomposition PGD est, par conséquent, plus lourd que celui des autres configurations. On peut alors, dans un avenir proche, investiguer dans cette direction en réfléchissant sur une manière judicieuse de pondérer les différentes configurations lors du calcul des modes PGD, c'est-à-dire sur le choix du coefficient Δ_p de l'équation (4.1). Nous avons affecté, dans notre application, la valeur 1 à ce coef- ficient, ce qui signifie que nous n'avons privilégié aucune configuration parmi toutes celles considérées. Nous avons simplement sommé les énergies de chaque situation alors que l'on peut imaginer, aisément, que ces situations ont des énergies différentes. On pourrait, donc, en jouant sur le coefficient Δ_p , affecter un poids plus ou moins im- portant à chaque configuration.

4.4.2 Application au démonstrateur industriel

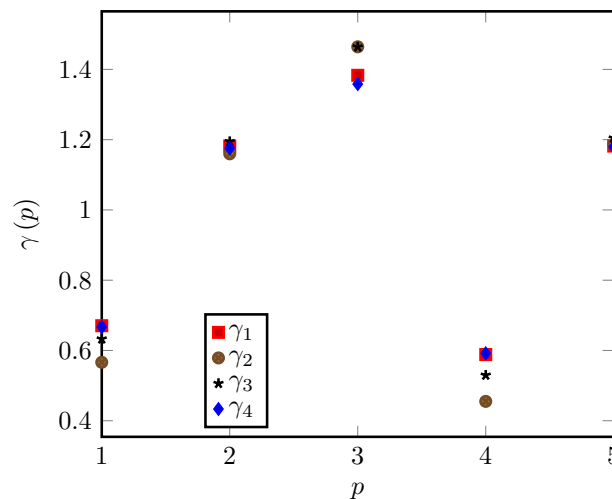
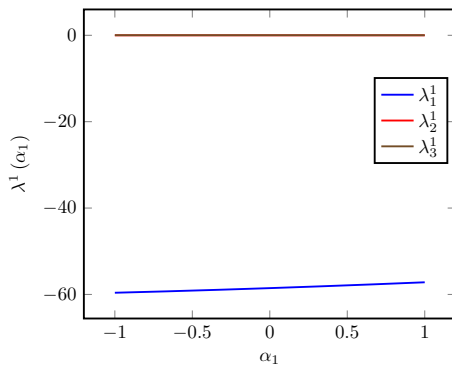
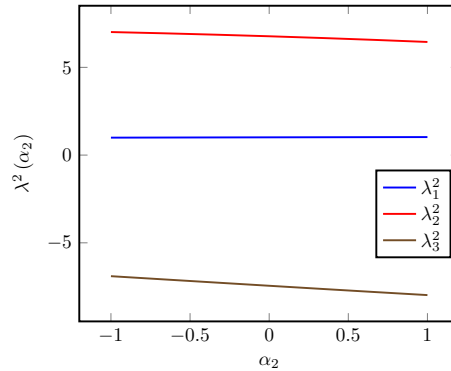


FIGURE 4.12 – Quatre premières fonctions γ .

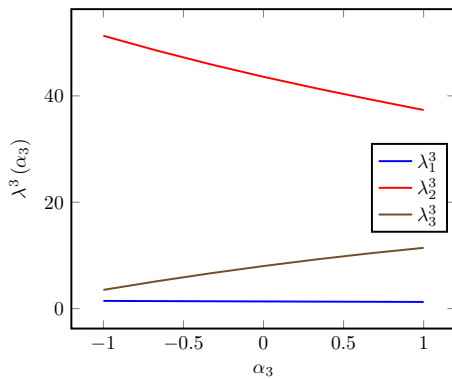
L'approche hybride a ensuite été appliquée sur le démonstrateur présenté en début de chapitre. En amont, nous avons décidé d'étudier l'influence des paramètres géomé- triques et des différentes configurations de composite sur l'état de contraintes à l'in- terface des deux pièces. Pour cela, une série de simulations non-linéaires complètes a



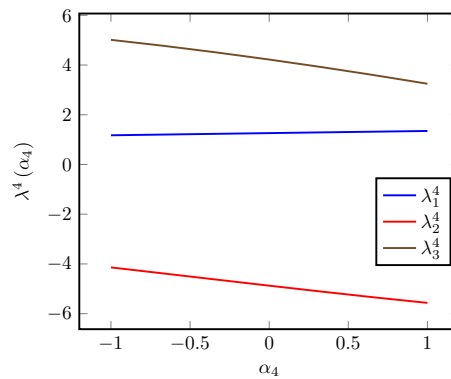
(a) Trois premières fonctions λ^1 .



(b) Trois premières fonctions λ^2 .



(c) Trois premières fonctions λ^3 .



(d) Trois premières fonctions λ^4 .

FIGURE 4.13 – Premières fonctions λ^1 , λ^2 , λ^3 et λ^4 .

été menée à l'aide de SAMCEF pour plusieurs jeux de paramètres géométriques de la pièce A (à configuration de composite fixée pour la pièce B). Il s'avère, en définitive, que les paramètres géométriques ont une faible influence sur l'état de contraintes à l'interface et que l'on peut, par conséquent, en négliger la dépendance.

La même démarche a été appliquée concernant les configurations de composite (à paramètres géométriques fixés). On remarque, ici, en revanche, la forte influence de ces configurations sur l'état de contraintes. On arrive alors à la situation suivante. Pour chaque configuration de composite, on calcule un chargement équivalent à l'interface des pièces A et B (on se place au pas de chargement final des simulations non-linéaires). Puisque l'on souhaite dimensionner la pièce A, on lui applique le chargement calculé (voir Figure 4.2) et on utilise la méthode dPGD hybride, où les paramètres géométriques définissant la frontière inférieure de la pièce sont considérés comme les paramètres de la PGD. En d'autres termes, l'ensemble {pièce A + pièce B} est considéré dans la première partie de l'étude afin de calculer les différents cas de chargement associés à chaque configuration de composite puis, dans une deuxième partie, seule la pièce A est étudiée en faisant varier les paramètres géométriques dans un cadre dPGD.

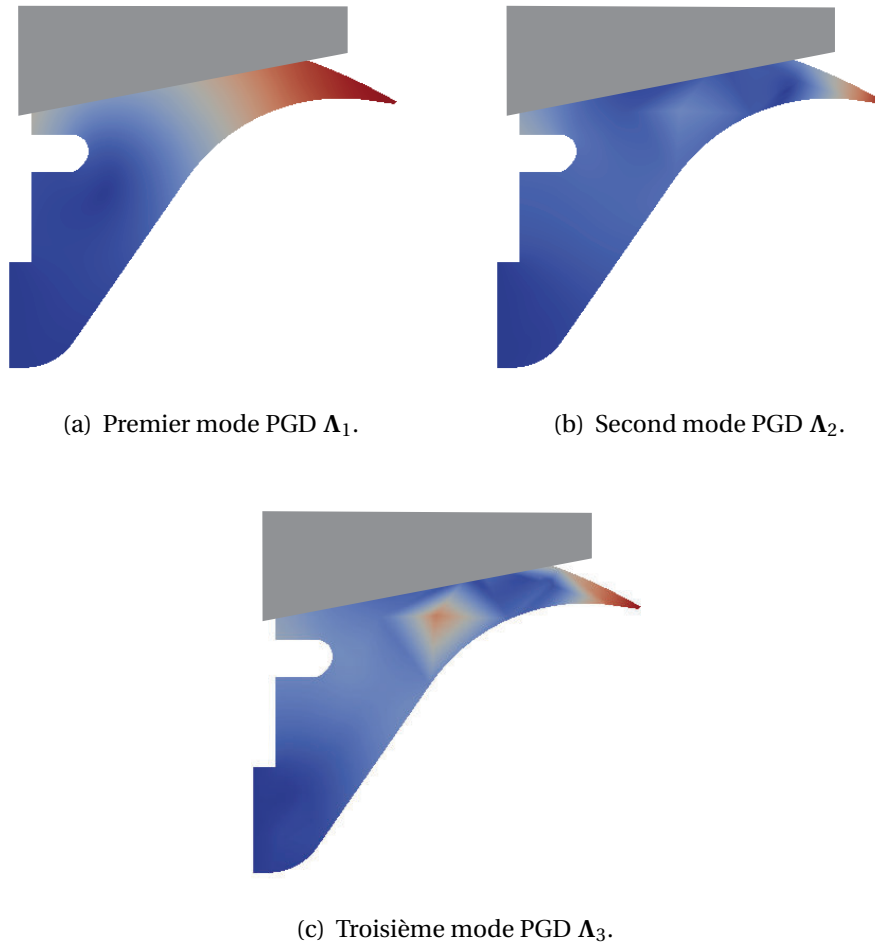


FIGURE 4.14 – Trois premiers modes PGD.

Si la dépendance de l'état de contraintes à l'interface pièce A/pièce B selon les paramètres géométriques n'avait pas été négligeable, on aurait pu, en toute généralité, construire pour chaque configuration composite p , une suite de chargements $\mathbf{f}_p(\boldsymbol{\alpha}) = (\mathbf{f}_p^1(\boldsymbol{\alpha}), \dots, \mathbf{f}_p^n(\boldsymbol{\alpha}))$. Chaque composante $\mathbf{f}_p^i(\boldsymbol{\alpha})$ correspond au chargement équivalent, calculé à l'interface pièce A/pièce B, pour la configuration de chargement p , et pour un jeu de paramètres géométriques donné. On aurait alors, dans la méthodologie PGD, un chargement $\mathbf{f}_p(\boldsymbol{\alpha})$ dépendant à la fois de la configuration du composite et des paramètres géométriques, ce qui ne présente aucune difficulté particulière. On aurait alors recouru à une technique de séparation de variables type SVD ou PGD afin d'écrire le chargement sous forme séparée.

Les premières fonctions γ et λ sont tracées, respectivement, sur les Figures 4.12 et 4.13. On remarque que les paramètres α_1 et α_2 ont toujours une faible influence sur le comportement de la structure (les pentes des fonctions associées à ces paramètres sont

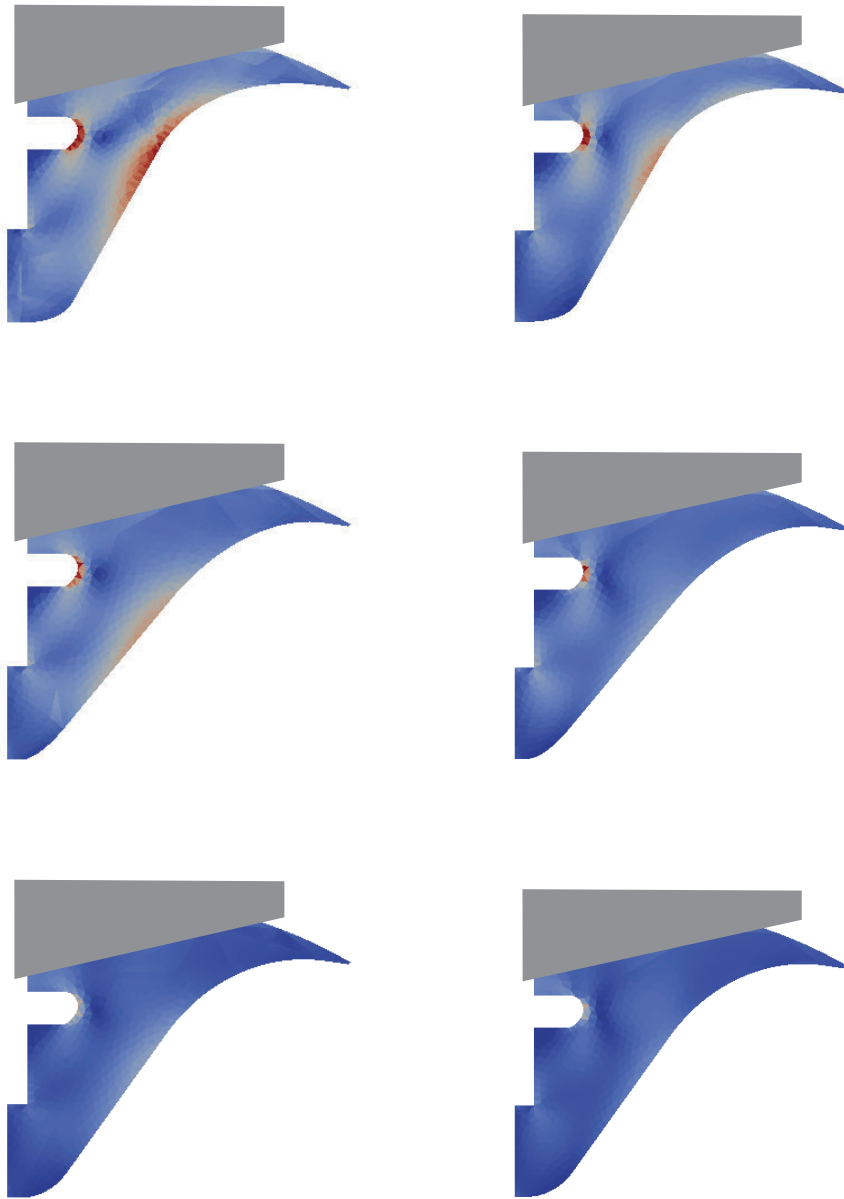


FIGURE 4.15 – Comparaison entre la contrainte de Von Mises obtenue par la méthode dPGD (41 modes, à gauche) et celle obtenue par les éléments finis de référence (à droite) pour trois configurations (les jeux de paramètres géométriques ainsi que le cas de chargement sont différents pour les trois configurations considérées).

très faibles) et que le chargement a , quant à lui, a une grande incidence sur les résultats.

Les trois premiers modes PGD sont représentés sur la Figure 4.14 et une comparaison entre la solution dPGD (41 modes) et une solution de référence éléments finis a été

effectuée. Les résultats sont visibles pour trois configurations différentes sur la Figure 4.15 où a été tracée la contrainte de Von Mises. On note une légère surestimation de la contrainte dans le cas de la dPGD. Néanmoins, la répartition de la contrainte, son ordre de grandeur ainsi que les localisations des maxima sont respectés. On calculera, dans un futur proche, un nombre plus important de modes dans le but d'affiner la solution, ce qui n'a pu être fait dans le temps imparti de la thèse. On peut, tout de même, noter qu'avec seulement 41 modes, on obtient une bonne évaluation de la contrainte de Von Mises du problème complet. Ceci est à comparer avec le nombre de modes qu'il aurait fallu calculer en construisant six modèles PGD (c'est-à-dire pour chaque cas de chargement). Cela correspond, si l'on se réfère aux résultats du chapitre précédent, à $6 \times 17 = 102$ modes. On a donc gagné un nombre conséquent de modes grâce à la dPGD et cela laisse encore de la marge au calcul de modes supplémentaires pour l'obtention d'une solution plus fine. Comme dans l'exemple industriel précédent, un abaque virtuel associé aux zones plastiques a été construit. Ici, nous avons pris en considération la contrainte de Von Mises et nous avons eu recours une fois de plus à la fonction *Threshold* de PARAVIEW pour ne conserver que les éléments dont la contrainte de Von Mises dépassait un seuil donné. Un extrait de cet abaque est donné Figure 4.16 pour quatre configurations de composite et jeux de paramètres différents.

On aurait pu, en toute généralité, interpoler les résultats obtenus pour des cas de chargement intermédiaires. Cependant, il faut garder à l'esprit que, dans notre démonstrateur, à chaque cas de chargement est associée une configuration de composite et qu'interpoler des configurations de composite n'a pas de sens physique réel.

4.5 Conclusions

La prise en compte de paramètres discrets ouvre de nouvelles perspectives pour la méthode PGD. Il a été montré que la PGD était une méthode adaptée à ce type de paramètres et que seules de légères modifications étaient nécessaires pour son utilisation. On peut alors penser recourir à des paramètres discrets dans la PGD pour modéliser, non seulement, des cas de chargements mais aussi le matériau lui-même. En effet, on peut très bien imaginer des cas où seuls quelques matériaux doivent être testés : composite, acier, alliage de titane, aluminium, etc. Il est alors parfaitement inutile de parcourir toute une plage de valeurs de paramètres matériau, comme on le ferait avec une variable continue. Il serait plus judicieux d'utiliser des paramètres discrets. Le profit, que l'on espère à terme retirer de ce type d'approche, est de pouvoir mélanger des configurations de structures très différentes entre elles de par leur matériau ou cas de chargement et, dans le même temps, de les considérer dans une même étude.

Il est envisagé dans un futur proche d'explorer cette direction et d'étudier la dPGD pour différents types de problèmes (temps/espace/paramètres, équation de la chaleur, grand nombre de paramètres, configurations différentes de matériaux, etc.).

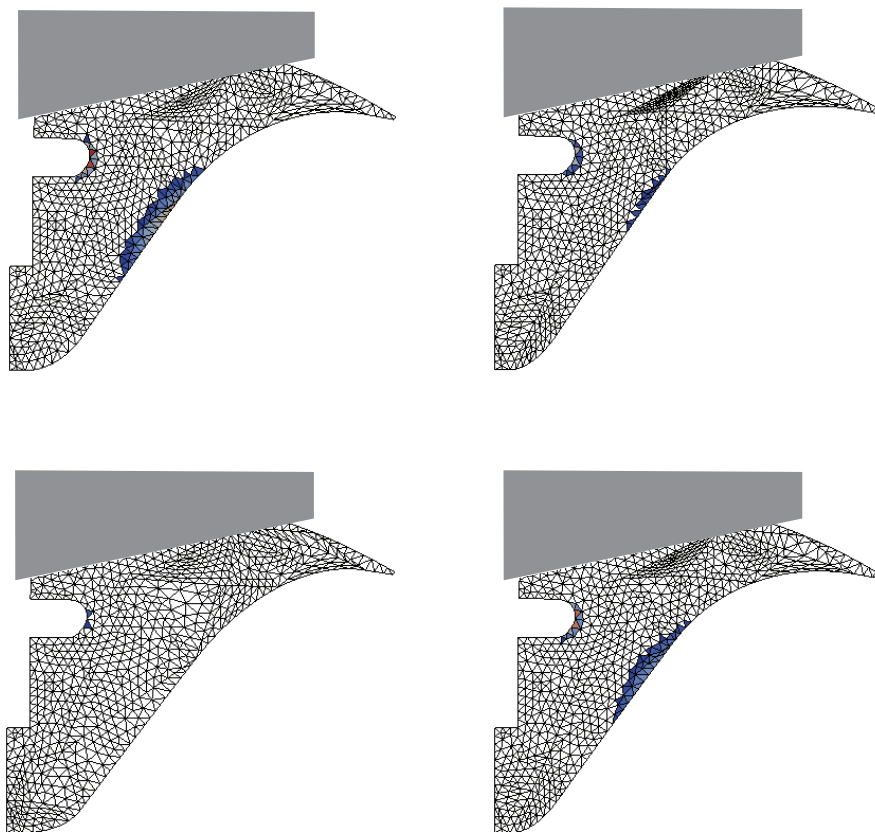


FIGURE 4.16 – Abaque virtuel représentant l'évolution des zones plastiques selon les paramètres géométriques et le cas de chargement considéré. Les zones plastiques sont estimées en ne conservant que les éléments dont la contrainte de Von Mises dépasse un certain seuil. Les quatre configurations illustrées sont associées à des jeux de paramètres géométriques et des cas de chargement tous différents.

Conclusion

Les travaux effectués au cours de cette thèse ont porté sur la construction d'abaques virtuels pour l'optimisation géométrique des structures. Nous ne nous sommes pas intéressés à l'optimisation en tant que telle puisque la littérature regorge de nombreuses méthodes traitant du sujet. Toutefois, lors de l'optimisation, de multiples évaluations de la solution du problème paramétrique complet sont nécessaires, entraînant une explosion du temps de calcul. L'idée, qui a été exposée et développée dans la thèse, est de résoudre, une fois pour toutes, le problème complet et de stocker la solution et les quantités d'intérêt utiles à l'optimisation dans des abaques virtuels. Les évaluations seront alors réduites à une simple lecture d'entrées d'un abaque. Le temps perdu pendant la construction des abaques sera très largement compensé par celui gagné lors de leur utilisation.

La recherche de la solution du problème paramétrique complet étant très souvent hors de portée des méthodes type « force brute », nous nous sommes attachés à baser notre méthodologie sur la *Proper Generalized Decomposition*, une technique de réduction de modèle. Elle permet la construction d'une approximation à variables séparées de la solution du problème complet (typiquement le déplacement) de manière automatique. Aucune connaissance *a priori* de la solution vraie n'est requise pour nourrir le modèle. Il a fallu, néanmoins, créer un cadre dédié à la PGD permettant la prise en compte des paramètres géométriques. En effet, ce type de paramètres influe directement sur le domaine d'intégration rendant difficile une résolution du problème complet sans passer par des évaluations du problème particularisé pour des valeurs données de paramètres. Deux méthodes ont été développées dans cette optique.

Dans un premier temps, le choix a été fait de paramétrer la rigidité d'une géométrie de référence englobant l'ensemble des configurations possibles dans l'esprit des *level sets* ou de la *Finite Cell Method*. Ainsi, lorsque l'on se trouve dans la structure courante étudiée, la rigidité associée est celle classique et lorsque l'on se situe en dehors, la rigidité est choisie très faible. La géométrie de référence est, dans un sens, composée d'une partie *dure* et d'une partie *molle*. Une technique pour calculer de manière effective une approximation de la matrice de rigidité est proposée et la méthode a été testée sur un exemple académique. Cette méthode présente, cependant, quelques difficultés, notamment pour l'application des chargements. Le contour de la structure pouvant couper des éléments du maillage, il n'est pas évident de choisir sur quels nœuds appliquer le chargement. C'est pourquoi, il a été proposé une extension de cette méthode grâce aux maillages universels. Ces derniers permettent la construction rapide de maillages

conformes pour des géométries immergées dans un même maillage, dit universel, par le simple déplacement local de nœuds. Cette technique développée dans [Rangarajan et Lew, 2012] a dû être modifiée afin de l'adapter au cadre PGD. Une application numérique a ensuite été effectuée pour vérifier la validité de la nouvelle méthode.

Dans un second temps, une nouvelle approche a été développée reposant sur l'utilisation de transformations géométriques. Cela permet la redéfinition du problème paramétrique complet sur une structure de référence. Il est montré que, pour obtenir des temps de calculs raisonnables, il est souhaitable de recourir à des transformations affines comme il avait été proposé dans [Ammar *et al.*, 2014, Zlotnik *et al.*, 2015]. Nous sommes, cependant, allés plus loin en ne nous restreignant pas uniquement au choix d'une partition triangulaire mais en proposant plusieurs types de transformations affines possibles afin de traiter des géométries complexes de type splines de l'ordre de celles rencontrées dans l'industrie. Outre une application de la technique sur un exemple académique 2D, nous nous sommes attachés à traiter le cas d'un démonstrateur issu de l'industrie aérospatiale proposée par AIRBUS Defence & Space. La méthode a dû non seulement être étendue au cas axisymétrique mais aussi être intégrée dans un process industriel de conception basé sur l'utilisation de logiciels éléments finis commerciaux. À cause du caractère hautement intrusif de la PGD, nous avons couplé un code MATLAB, effectuant les diverses opérations propres à la PGD, au logiciel SAMCEF pour le calcul des problèmes en espace. L'abaque virtuel de la déformation est donné, son exploration étant rendue possible par l'utilisation du *plug-in* PXDMF de PARAVIEW développé à l'École Centrale de Nantes permettant la visualisation de champs à variables séparées.

La thèse se termine sur de nouvelles considérations concernant la PGD. En effet, il est montré que, dans certains cas, paramétrer un problème via des variables continues est difficile voire impossible à cause de la nature même du paramètre en jeu. En prenant l'exemple des cas de chargement, il est expliqué pourquoi il serait plus judicieux de recourir à des paramètres discrets prenant pour valeurs le numéro des configurations considérées. Il est démontré que la formulation PGD n'est en rien altérée par la prise en considération de tels paramètres et qu'il est même possible d'adopter des approches hybrides mélangeant paramètres continus et discrets. Cette extension de la PGD a été testée sur deux exemples académiques. Elle a ensuite été appliquée au démonstrateur industriel précédent où six cas de chargement sont considérés ainsi que des variations de géométrie. Nous avons, de plus, pris en compte, dans les simulations, les fortes non-linéarités que comportait le modèle (grands déplacements, plasticité, grandes déformations et non-linéarités matériau). Les applications industrielles de ce type d'approches sont importantes puisqu'il est très courant, lors de la conception d'une structure, de la tester pour tout un ensemble de cas chargement. Ces cas pourront alors être considérés dans un même paramètre PGD. En outre, on peut envisager, pour aller plus loin, de paramétrer le choix même du matériau de la structure. À chaque matériau sera associé un numéro de configuration et il ne restera qu'à associer un paramètre regroupant toutes les configurations.

Dans le prolongement de ce travail, plusieurs perspectives sont envisageables :

- Utilisation de la méthode basée sur les maillages universels afin de calculer les matrices de rigidité pour certaines configurations géométriques en se rapprochant de l'esprit *snapshot* de la POD. À partir de ces échantillons, une approximation de la matrice de rigidité sous forme séparée pourra alors être calculée via la méthode des points de référence (RPM) [Capaldo *et al.*, 2014].
- Application de la PGD géométrique à des problématiques de fabrication type tolérancement.
- Introduire une pondération dans la dPGD et en étudier l'influence sur la convergence de la méthode.
- Application des deux méthodes (maillages universels et transformations géométriques) à des problèmes non-linéaires via, par exemple, le solveur LaTIn.
- Extension des deux méthodes à des problèmes 3D.
- Intégration des méthodes développées dans un process d'optimisation complet. Nous nous sommes arrêtés ici à la construction des champs nécessaires à l'optimisation.
- Utilisation d'estimateurs d'erreur afin d'obtenir des quantités d'intérêt locales. En effet, la solution complète est très souvent inutile pour la conception ou l'optimisation, seule la valeur de certains champs dans des zones localisées étant nécessaire.
- Construction d'un outil d'aide à la décision complet dans le cadre de la conception prenant en compte des paramètres de tout type : discrets/continus, matériau, géométriques, stochastiques, etc.

Bibliographie

- [Aguado *et al.*, 2015] AGUADO, J. V., HUERTA, A., CHINESTA, F. et CUETO, E. (2015). Real-time monitoring of thermal processes by reduced-order modeling. *International Journal For Numerical Methods In Engineering*, 102(5):991–1017.
- [Alfaro *et al.*, 2014] ALFARO, I., GONZÁLEZ, D., BORDEU, F., LEYGUE, A., AMMAR, A., CUETO, E. et CHINESTA, F. (2014). Real-time in silico experiments on gene regulatory networks and surgery simulation on handheld devices. *Journal of Computational Surgery*, 1(1).
- [Allaire, 2007] ALLAIRE, G. (2007). *Conception optimale de structures*, volume 58. Springer.
- [Allaire *et al.*, 2002] ALLAIRE, G., JOUVE, F. et TOADER, A.-M. (2002). A level-set method for shape optimization. *Comptes Rendus Mathématique*, 334(12):1125–1130.
- [Allaire *et al.*, 2004] ALLAIRE, G., JOUVE, F. et TOADER, A.-M. (2004). Structural optimization using sensitivity analysis and a level-set method. *Journal of computational physics*, 194(1):363–393.
- [Allier *et al.*, 2015] ALLIER, P.-E., CHAMOIN, L. et LADEVÈZE, P. (2015). Proper generalized decomposition computational methods on a benchmark problem : introducing a new strategy based on constitutive relation error minimization. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):1–25.
- [Allix et Ladevèze, 1992] ALLIX, O. et LADEVÈZE, P. (1992). Interlaminar interface modelling for the prediction of delamination. *Composite structures*, 22(4):235–242.
- [Ammar *et al.*, 2014] AMMAR, A., HUERTA, A., CHINESTA, F., CUETO, E. et LEYGUE, A. (2014). Parametric solutions involving geometry : a step towards efficient shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 268:178–193.
- [Ammar *et al.*, 2007a] AMMAR, a., MOKDAD, B., CHINESTA, F. et KEUNINGS, R. (2007a). A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids. *Journal of Non-Newtonian Fluid Mechanics*, 144(2-3):98–121.
- [Ammar *et al.*, 2007b] AMMAR, A., MOKDAD, B., CHINESTA, F. et KEUNINGS, R. (2007b). A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids : Part II : Transient simulation using space-time separated representations. *Journal of Non-Newtonian Fluid Mechanics*, 144(2-3):98–121.

- [Ammar *et al.*, 2015] AMMAR, A., ZGHAL, A., MOREL, F. et CHINESTA, F. (2015). On the space-time separated representation of integral linear viscoelastic models. *Comptes Rendus de Mécanique*, 343(4):247–263.
- [Atwell et King, 2001] ATWELL, J. A. et KING, B. B. (2001). Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations. *Mathematical and computer modelling*, 33(1):1–19.
- [Barbarulo *et al.*, 2014] BARBARULO, A., RIOU, H., KOVALEVSKY, L. et LADEVEZE, P. (2014). PGD-VTCR : A Reduced Order Model Technique to Solve Medium Frequency Broad Band Problems on Complex Acoustical Systems. *Journal of Mechanical Engineering*, 60(5):307–313.
- [Barrault *et al.*, 2004] BARRAULT, M., Y., M., NGUYEN, N. et PATERA, A. (2004). An “empirical interpolation” method : Application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus de l’Académie des Sciences Paris*, 339:667–672.
- [Beltrami, 1873] BELTRAMI, E. (1873). Sulle funzioni bilineari. *Giornale di Matematiche ad Uso degli Studenti Delle Università*, 11:98–106.
- [Belytschko et Black, 1999] BELYTSCHKO, T. et BLACK, T. (1999). Elastic crack growth in finite elements with minimal remeshing. *International journal for numerical methods in engineering*, 45(5):601–620.
- [Bendsøe, 1995] BENDSØE, M. P. (1995). Optimization of structural topology, shape, and material.
- [Bendsøe et Sigmund, 2003] BENDSØE, M. P. et SIGMUND, O. (2003). Topology optimization : theory, methods and applications.
- [Bernoulli, 1836] BERNOULLI, C. (1836). *Vademecum des Mechanikers*. J. G. Cotta, Stuttgart und Tübingen.
- [Boender *et al.*, 1982] BOENDER, C. G. E., KAN, A. R., TIMMER, G. et STOUGIE, L. (1982). A stochastic method for global optimization. *Mathematical programming*, 22(1): 125–140.
- [Bognet *et al.*, 2012a] BOGNET, B., BORDEU, F., CHINESTA, F., LEYGUE, a. et POITOU, a. (2012a). Advanced simulation of models defined in plate geometries : 3D solutions with 2D computational complexity. *Computer Methods in Applied Mechanics and Engineering*, 201-204:1–12.
- [Bognet *et al.*, 2012b] BOGNET, B., LEYGUE, A., CHINESTA, F. et BORDEU, F. (2012b). Parametric shape and material simulations for optimized parts design. In *ASME 2012 11th Biennial Conference on Engineering Systems Design and Analysis*, pages 217–218. American Society of Mechanical Engineers.
- [Boisse *et al.*, 1990] BOISSE, P., BUSSY, P. et LADÈVEZE, P. (1990). A new approach in non-linear mechanics : The large time increment method. *International journal for numerical methods in engineering*, 29(3):647–663.

- [Bonithon *et al.*, 2011] BONITHON, G., JOYOT, P., CHINESTA, F. et VILLON, P. (2011). Non-incremental boundary element discretization of parabolic models based on the use of the proper generalized decompositions. *Engineering Analysis with Boundary Elements*, 35(1):2–17.
- [Boucinha *et al.*, 2014] BOUCINHA, L., AMMAR, A., GRAVOUIL, A. et NOUY, A. (2014). Ideal minimal residual-based proper generalized decomposition for non-symmetric multi-field models—application to transient elastodynamics in space-time domain. *Computer Methods in Applied Mechanics and Engineering*, 273:56–76.
- [Boucinha *et al.*, 2013] BOUCINHA, L., GRAVOUIL, A. et AMMAR, A. (2013). Space-time proper generalized decompositions for the resolution of transient elastodynamic models. *Computer Methods in Applied Mechanics and Engineering*, 255:67–88.
- [Boyaval *et al.*, 2010] BOYAVAL, S., LE BRIS, C., LELIÈVRE, T., MADAY, Y., NGUYEN, N. C. et PATERA, A. T. (2010). Reduced basis techniques for stochastic problems. *Archives of Computational Methods in Engineering*, 17(4):435–454.
- [Bro, 1997] BRO, R. (1997). Parafac. tutorial and applications. *Chemometrics and intelligent laboratory systems*, 38(2):149–171.
- [Buhmann, 2000] BUHMANN, M. D. (2000). Radial basis functions. *Acta Numerica*, 9:1–38.
- [Cao *et al.*, 2007] CAO, Y., ZHU, J., NAVON, I. M. et LUO, Z. (2007). A reduced-order approach to four-dimensional variational data assimilation using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids*, 53(10):1571–1584.
- [Capaldo *et al.*, 2014] CAPALDO, M., NÉRON, D., GUIDAULT, P.-A. et LADEVÈZE, P. (2014). An approximation framework dedicated to PGD-based nonlinear solver. *In WCCM XI 11th. World Congress on Computational Mechanics*.
- [Cauchy, 1847] CAUCHY, A. (1847). Méthode générale pour la résolution des systèmes d'équations simultanées. *Comptes Rendus de l'Académie des Sciences Paris*, 25:536–538.
- [Chatterjee, 2000] CHATTERJEE, A. (2000). An introduction to the proper orthogonal decomposition. *Current science*, 78(7):808–817.
- [Chinesta *et al.*, 2010a] CHINESTA, F., AMMAR, A. et CUETO, E. (2010a). Proper generalized decomposition of multiscale models. *International Journal for Numerical Methods in Engineering*, 83(8-9):1114–1132.
- [Chinesta *et al.*, 2010b] CHINESTA, F., AMMAR, A. et CUETO, E. (2010b). Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. *Archives of Computational methods in Engineering*, 17(4):327–350.
- [Chinesta *et al.*, 2011] CHINESTA, F., AMMAR, A., LEYGUE, A. et KEUNINGS, R. (2011). An overview of the proper generalized decomposition with applications in computational rheology. *Journal of Non-Newtonian Fluid Mechanics*, 166(11):578–592.

- [Chinesta *et al.*, 2013] CHINESTA, F., LEYGUE, A., BORDEU, F., AGUADO, J., CUETO, E., GONZALEZ, D., ALFARO, I., AMMAR, A. et HUERTA, A. (2013). PGD-based computational vademecum for efficient design, optimization and control. *Archives of Computational Methods in Engineering*, 20(1):31–59.
- [Cognard *et al.*, 1999] COGNARD, J., LADEVÈZE, P. et TALBOT, P. (1999). A large time increment approach for thermo-mechanical problems. *Advances in Engineering Software*, 30(9):583–593.
- [Cognard et Ladevèze, 1993] COGNARD, J.-Y. et LADEVÈZE, P. (1993). A large time increment approach for cyclic viscoplasticity. *International Journal of Plasticity*, 9(2):141–157.
- [Corigliano *et al.*, 2015] CORIGLIANO, A., DOSSI, M. et MARIANI, S. (2015). Model order reduction and domain decomposition strategies for the solution of the dynamic elastic–plastic structural problem. *Computer Methods in Applied Mechanics and Engineering*, 290:127–155.
- [Cremonesi *et al.*, 2013] CREMONESI, M., NÉRON, D., GUIDAULT, P.-A. et LADEVÈZE, P. (2013). A PGD-based homogenization technique for the resolution of nonlinear multiscale problems. *Computer Methods in Applied Mechanics and Engineering*, 267:275–292.
- [De Lathauwer *et al.*, 2000] DE LATHAUWER, L., DE MOOR, B. et VANDEWALLE, J. (2000). A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278.
- [Dorigo et Stützle, 2003] DORIGO, M. et STÜTZLE, T. (2003). The ant colony optimization metaheuristic : Algorithms, applications, and advances. *In Handbook of metaheuristics*, pages 250–285. Springer.
- [Duysinx, 1996] DUYSINX, P. (1996). Introduction à l’optimisation topologique.
- [Epureanu *et al.*, 2004] EPUREANU, B. I., TANG, L. S. et PAIÏLDOUSSIS, M. P. (2004). Coherent structures and their influence on the dynamics of aeroelastic panels. *International Journal of Non-Linear Mechanics*, 39(6):977–991.
- [Eurocode 3, 2010] COMITÉ EUROPÉEN DE NORMALISATION (2010). *Eurocode 3 : Design of steel structures : Part 3 : Buildings*.
- [Fletcher, 1970] FLETCHER, R. (1970). A new approach to variable metric algorithms. *The computer journal*, 13(3):317–322.
- [Fletcher et Reeves, 1964] FLETCHER, R. et REEVES, C. M. (1964). Function minimization by conjugate gradients. *The computer journal*, 7(2):149–154.
- [Galland *et al.*, 2011] GALLAND, F., GRAVOUIL, A., MALVESIN, E. et ROCHETTE, M. (2011). A global model reduction approach for 3d fatigue crack growth with confined plasticity. *Computer Methods in Applied Mechanics and Engineering*, 200(5):699–716.
- [Galvis et Kang, 2014] GALVIS, J. et KANG, S. K. (2014). Spectral multiscale finite element for nonlinear flows in highly heterogeneous media : A reduced basis approach. *Journal of Computational and Applied Mathematics*, 260:494–508.

- [Giacoma *et al.*, 2014] GIACOMA, A., DUREISSEIX, D., GRAVOUIL, A. et ROCHETTE, M. (2014). A multiscale large time increment/fas algorithm with time-space model reduction for frictional contact problems. *International Journal for Numerical Methods in Engineering*, 97(3):207–230.
- [Giacoma *et al.*, 2015] GIACOMA, A., DUREISSEIX, D., GRAVOUIL, A. et ROCHETTE, M. (2015). Toward an optimal a priori reduced basis strategy for frictional contact problems with latin solver. *Computer Methods in Applied Mechanics and Engineering*, 283:1357–1381.
- [Giner *et al.*, 2013] GINER, E., BOGNET, B., RODENAS, J. J., LEYGUE, A., FUENMAYOR, F. J. et CHINESTA, F. (2013). The proper generalized decomposition (PGD) as a numerical procedure to solve 3D cracked plates in linear elastic fracture mechanics. *International Journal of Solids and Structures*, 50(10):1710–1720.
- [Glüsmann et Kreuzer, 2009] GLÜSMANN, P. et KREUZER, E. (2009). On the application of karhunen–loeve transform to transient dynamic systems. *Journal of Sound and Vibration*, 328(4):507–519.
- [González *et al.*, 2015] GONZÁLEZ, D., ALFARO, I., QUESADA, C., CUETO, E. et CHINESTA, F. (2015). Computational vademecums for the real-time simulation of haptic collision between nonlinear solids. *Computer Methods in Applied Mechanics and Engineering*, 283(1):210–223.
- [González *et al.*, 2010] GONZÁLEZ, D., AMMAR, A., CHINESTA, F. et CUETO, E. (2010). Recent advances on the use of separated representations. *International Journal for Numerical Methods in Engineering*, 81(5):637–659.
- [Gunzburger *et al.*, 2007] GUNZBURGER, M., PETERSON, J. et SHADID, J. (2007). Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data. *Computational Methods in Applied Mechanical Engineering*, 196: 1030–1047.
- [Han et Feeny, 2003] HAN, S. et FEENY, B. (2003). Application of proper orthogonal decomposition to structural vibration analysis. *Mechanical Systems and Signal Processing*, 17(5):989–1001.
- [Hestenes et Stiefel, 1952] HESTENES, M. R. et STIEFEL, E. (1952). Methods of conjugate gradients for solving linear systems.
- [Holland, 1962] HOLLAND, J. H. (1962). Outline for a logical theory of adaptive systems. *Journal of the ACM (JACM)*, 9(3):297–314.
- [Holmes, 2012] HOLMES, P. (2012). *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press.
- [Hooke et Jeeves, 1961] HOOKE, R. et JEEVES, T. A. (1961). Direct Search Solution of Numerical and Statistical Problems. *Journal of the ACM (JACM)*, 8(2):212–229.
- [Hotelling, 1933] HOTELLING, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417–441,498–520.

- [Hung et Senturia, 1999] HUNG, E. S. et SENTURIA, S. D. (1999). Generating efficient dynamical models for microelectromechanical systems from a few finite-element simulation runs. *Microelectromechanical Systems, Journal of*, 8(3):280–289.
- [Jones *et al.*, 1993] JONES, D. R., PERTTUNEN, C. D. et STUCKMAN, B. E. (1993). Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181.
- [Jordan, 1874a] JORDAN, C. (1874a). Mémoire sur les formes bilinéaires. *Journal de mathématiques pures et appliquées*, pages 35–54.
- [Jordan, 1874b] JORDAN, C. (1874b). Sur la réduction des formes bilinéaires. *Comptes Rendus de des Sciences*, 78:614–617.
- [Kabaria et Lew, 2014] KABARIA, H. et LEW, A. J. (2014). Universal meshes for smooth surfaces with no boundary in three dimensions. *International Journal for Numerical Methods in Engineering*. Preprint.
- [Kan et Timmer, 1987] KAN, A. R. et TIMMER, G. (1987). Stochastic global optimization methods part I : Clustering methods. *Mathematical programming*, 39(1):27–56.
- [Karhunen, 1947] KARHUNEN, K. (1947). *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*, volume 37. Universitat Helsinki.
- [Kerschen *et al.*, 2005] KERSCHEN, G., GOLINVAL, J.-c., VAKAKIS, A. F. et BERGMAN, L. A. (2005). The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems : an overview. *Nonlinear dynamics*, 41(1-3):147–169.
- [Kirkpatrick *et al.*, 1983] KIRKPATRICK, S., GELATT JR, C. et VECCHI, M. (1983). Optimization by simulated annealing. *SCIENCE*, 220(4598).
- [Kolda *et al.*, 2003] KOLDA, T. G., LEWIS, R. M. et TORCZON, V. (2003). Optimization by direct search : New perspectives on some classical and modern methods. *SIAM review*, 45(3):385–482.
- [Ladevèze, 1985] LADEVÈZE, P. (1985). Sur une famille d’algorithmes en mécanique des structures. *Comptes-rendus des séances de l’Académie des sciences. Série 2, Mécanique-physique, chimie, sciences de l’univers, sciences de la terre*, 300(2):41–44.
- [Ladevèze, 1999] LADEVÈZE, P. (1999). *Nonlinear Computational Structural Mechanics—New Approaches and Non-Incremental Methods of Calculation*. Springer Verlag.
- [Ladevèze, 2013] LADEVÈZE, P. (2013). The Virtual Chart concept in computational structural mechanics (plenary lecture). In *COMPLAS XII, 3-5 September 2013, Barcelona (Spain)*.
- [Ladevèze, 2014] LADEVÈZE, P. (2014). PGD in linear and nonlinear Computational Solid Mechanics. In *Separated Representations and PGD-Based Model Reduction*, pages 91–152. Springer.

- [Ladevèze et Chamoin, 2011] LADEVÈZE, P. et CHAMOIN, L. (2011). On the verification of model reduction methods based on the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering*, 200(23):2032–2047.
- [Ladevèze et Chamoin, 2013] LADEVÈZE, P. et CHAMOIN, L. (2013). Toward guaranteed pgd-reduced models. *Bytes and Science. CIMNE : Barcelona*.
- [Lassila et Rozza, 2010] LASSILA, T. et ROZZA, G. (2010). Parametric free-form shape design with pde models and reduced basis method. *Computer Methods in Applied Mechanics and Engineering*, 199(23):1583–1592.
- [Laurent *et al.*, 2013a] LAURENT, L., BOUCARD, P.-A. et SOULIER, B. (2013a). A dedicated multiparametric strategy for the fast construction of a cokriging metamodel. *Computers & Structures*, 124:61–73.
- [Laurent *et al.*, 2013b] LAURENT, L., BOUCARD, P.-A. et SOULIER, B. (2013b). Generation of a cokriging metamodel using a multiparametric strategy. *Computational Mechanics*, 51(2):151–169.
- [Le Riche et Cailletaud, 1998] LE RICHE, R. et CAILLETAUD, G. (1998). A mixed evolutionary/heuristic approach to shape optimization. *International Journal For Numerical Methods In Engineering*, 41:1463–1484.
- [Le Riche et Gaudin, 1998] LE RICHE, R. et GAUDIN, J. (1998). Design of dimensionally stable composites by evolutionary optimization. *Composite Structures*, 41(2):97–111.
- [Lewis *et al.*, 2000] LEWIS, R. M., TORCZON, V. et TROSSET, M. W. (2000). Direct search methods : then and now. *Journal of computational and Applied Mathematics*, 124(1): 191–207.
- [Loève, 1955] LOÈVE, M. (1955). *Probability Theory : Foundations, Random Sequences*. New York : D. Van Nostrand Company.
- [Maday et Ronquist, 2004] MADAY, Y. et RONQUIST, E. (2004). The reduced-basis element method : application to a thermal fin problem. *Journal on Scientific Computing*, 26 (1):240–258.
- [Manzoni *et al.*, 2012a] MANZONI, A., QUARTERONI, A. et ROZZA, G. (2012a). Model reduction techniques for fast blood flow simulation in parametrized geometries. *International journal for numerical methods in biomedical engineering*, 28(6-7):604–625.
- [Manzoni *et al.*, 2012b] MANZONI, A., QUARTERONI, A. et ROZZA, G. (2012b). Shape optimization for viscous flows by reduced basis methods and free-form deformation. *International Journal for Numerical Methods in Fluids*, 70(5):646–670.
- [Moës *et al.*, 1999] MOËS, N., DOLBOW, J. et BELYTSCHKO, T. (1999). A finite element method for crack growth without remeshing. *International journal for numerical methods in engineering*, 46(1):131–150.
- [Mokdad *et al.*, 2007] MOKDAD, B., PRULIÈRE, E., AMMAR, A. et CHINESTA, F. (2007). On the simulation of kinetic theory models of complex fluids using the Fokker-Planck approach. *Applied Rheology*, 17(2):26494.

- [Myers *et al.*, 2009] MYERS, R. H., MONTGOMERY, D. C. et ANDERSON-COOK, C. M. (2009). *Response surface methodology : process and product optimization using designed experiments*, volume 705. John Wiley & Sons.
- [Néron *et al.*, 2015] NÉRON, D., BOUCARD, P.-A. et RELUN, N. (2015). Time-space PGD for the rapid solution of 3D nonlinear parametrized problems in the many-query context. *International Journal for Numerical Methods in Engineering*, 103(4):275–292.
- [Néron et Dureisseix, 2008a] NÉRON, D. et DUREISSEIX, D. (2008a). A computational strategy for poroelastic problems with a time interface between coupled physics. *International Journal for Numerical Methods in Engineering*, 73(6):783–804.
- [Néron et Dureisseix, 2008b] NÉRON, D. et DUREISSEIX, D. (2008b). A computational strategy for thermo-poroelastic structures with a time-space interface coupling. *International Journal for Numerical Methods in Engineering*, 75(9):1053–1084.
- [Néron et Ladevèze, 2010] NÉRON, D. et LADEVÈZE, P. (2010). Proper generalized decomposition for multiscale and multiphysics problems. *Archives of Computational Methods in Engineering*, 17(4):351–372.
- [Nguyen, 2008] NGUYEN, N. C. (2008). A multiscale reduced-basis method for parametrized elliptic partial differential equations with multiple scales. *Journal of Computational Physics*, 227(23):9807–9822.
- [Niroomandi *et al.*, 2013] NIROOMANDI, S., ALFARO, I., GONZÁLEZ, D., CUETO, E. et CHINESTA, F. (2013). Model order reduction in hyperelasticity : a proper generalized decomposition approach. *International Journal for Numerical Methods in Engineering*, 96(3):129–149.
- [Nouy, 2007] NOUY, A. (2007). A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 196(45-48):4521–4537.
- [Nouy, 2009] NOUY, A. (2009). Recent developments in spectral stochastic methods for the numerical solution of stochastic partial differential equations. *Archives of Computational Methods in Engineering*, 16(3):251–285.
- [Nouy, 2010] NOUY, A. (2010). A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 199(23):1603–1626.
- [Osher et Sethian, 1988] OSHER, S. et SETHIAN, J. A. (1988). Fronts propagating with curvature-dependent speed : algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49.
- [Ottaviani et Paoletti, 2015] OTTAVIANI, G. et PAOLETTI, R. (2015). A geometric perspective on the singular value decomposition. *arXiv preprint arXiv :1503.07054*.
- [Pearson, 1901] PEARSON, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

- [Poli *et al.*, 2007] POLI, R., KENNEDY, J. et BLACKWELL, T. (2007). Particle swarm optimization. *Swarm intelligence*, 1(1):33–57.
- [Prud'homme *et al.*, 2002] PRUD'HOMME, C., ROVAS, D. V., VEROY, K., MACHIELS, L., MADAY, Y., PATERA, A. T. et TURINICI, G. (2002). Reliable real-time solution of parametrized partial differential equations : Reduced-basis output bound methods. *Journal of Fluids Engineering*, 124(1):70–80.
- [Quaranta *et al.*, 2004] QUARANTA, G., MANTEGAZZA, P. et MASARATI, P. (2004). Assessing the local stability of periodic motions for large multibody non-linear systems using proper orthogonal decomposition. *Journal of Sound and Vibration*, 271(3):1015–1038.
- [Raghavan *et al.*, 2013] RAGHAVAN, B., BREITKOPF, P., TOURBIER, Y. et VILLON, P. (2013). Towards a space reduction approach for efficient structural shape optimization. *Structural and Multidisciplinary Optimization*, 48(5):987–1000.
- [Rangarajan, 2012] RANGARAJAN, R. (2012). *Universal meshes : A new paradigm for computing with nonconforming triangulations*. Thèse de doctorat, Stanford.
- [Rangarajan *et al.*, 2014] RANGARAJAN, R., CHIARAMONTE, M. M., HUNSWECK, M. J., SHEN, Y. et LEW, A. J. (2014). Simulating curvilinear crack propagation in two dimensions with universal meshes. *International Journal for Numerical Methods in Engineering*.
- [Rangarajan et Lew, 2011] RANGARAJAN, R. et LEW, A. J. (2011). Parameterization of planar curves immersed in triangulations with application to finite elements. *International Journal for Numerical Methods in Engineering*, 88(6):556–585.
- [Rangarajan et Lew, 2012] RANGARAJAN, R. et LEW, A. J. (2012). Universal meshes: A new paradigm for computing with nonconforming triangulations. *arXiv preprint arXiv :1201.4903*.
- [Relun *et al.*, 2013] RELUN, N., NÉRON, D. et BOUCARD, P.-A. (2013). A model reduction technique based on the PGD for elastic-viscoplastic computational analysis. *Computational Mechanics*, 51(1):83–92.
- [Rozza, 2011] ROZZA, G. (2011). Reduced basis approximation and error bounds for potential flows in parametrized geometries. *Communication in Computational Physics*, 9:1–48.
- [Rozza *et al.*, 2008] ROZZA, G., HUYNH, D. B. P. et PATERA, A. T. (2008). Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):229–275.
- [Rozza et Veroy, 2007] ROZZA, G. et VEROY, K. (2007). On the stability of the reduced basis method for Stokes equations in parametrized domains. *Computer methods in applied mechanics and engineering*, 196(7):1244–1260.
- [Schenk et Schuëller, 2003] SCHENK, C. et SCHUËLLER, G. (2003). Buckling analysis of cylindrical shells with random geometric imperfections. *International Journal of Non-Linear Mechanics*, 38(7):1119–1132.

- [Schillinger et Ruess, 2014] SCHILLINGER, D. et RUESS, M. (2014). The finite cell method : A review in the context of higher-order structural analysis of cad and image-based geometric models. *Archives of Computational Methods in Engineering*, pages 1–65.
- [Schmidt, 1907] SCHMIDT, E. (1907). Zur Theorie der linearen und nichtlinearen Integralgleichungen. I. Teil : Entwicklung willkürlicher Funktionen nach Systemen vorgeschriebener. *Mathematische Annalen*, 63:433–476.
- [Sederberg et Parry, 1986] SEDERBERG, T. W. et PARRY, S. R. (1986). Free-form deformation of solid geometric models. In *Computer Graphics*, volume 20, pages 151–160. ACM.
- [Stewart, 1993] STEWART, G. W. (1993). On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566.
- [Sukumar *et al.*, 2001] SUKUMAR, N., CHOPP, D. L., MOËS, N. et BELYTSCHKO, T. (2001). Modeling holes and inclusions by level sets in the extended finite-element method. *Computer methods in applied mechanics and engineering*, 190(46):6183–6200.
- [Sylvester, 1889a] SYLVESTER, J. J. (1889a). A new proof that a general quadric may be reduced to its canonical form (that is, a linear function of squares) by means of a real orthogonal substitution. *Messenger of Mathematics*, 19:1–5.
- [Sylvester, 1889b] SYLVESTER, J. J. (1889b). On the reduction of a bilinear quantic of the n th order to the form of a sum of n products by a double orthogonal substitution. *Messenger of Mathematics*, 19:42–46.
- [Sylvester, 1889c] SYLVESTER, J. J. (1889c). Sur la réduction biorthogonale d’une forme linéo-linéaire à sa forme canonique. *Comptes Rendus de l’Académie des Sciences*, 108:651–653.
- [Veroy et Patera, 2005] VEROY, K. et PATERA, A. T. (2005). 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):773–788.
- [Vitse *et al.*, 2014] VITSE, M., NÉRON, D. et BOUCARD, P.-A. (2014). Virtual charts of solutions for parametrized nonlinear equations. *Computational Mechanics*, 54(6): 1529–1539.
- [Weyl, 1912] WEYL, H. (1912). Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung). *Mathematische Annalen*, 71(4):441–479.
- [Wright, 1996] WRIGHT, M. H. (1996). Direct search methods : Once scorned, now respectable. *Pitman Research Notes in Mathematics Series*, pages 191–208.
- [Wu *et al.*, 2003] WU, C., LIANG, Y., LIN, W., LEE, H. et LIM, S. (2003). A note on equivalence of proper orthogonal decomposition methods. *Journal of Sound and Vibration*, 265(5):1103–1110.

-
- [Zhang *et al.*, 2015] ZHANG, Y., COMBESCURE, A. et GRAVOUIL, A. (2015). Efficient hyper reduced-order model (hrom) for parametric studies of the 3d thermo-elasto-plastic calculation. *Finite Elements in Analysis and Design*, 102:37–51.
- [Zlotnik *et al.*, 2015] ZLOTNIK, S., DÍEZ, P., MODESTO, D. et HUERTA, A. (2015). Proper generalized decomposition of a geometrically parametrized heat problem with geophysical applications. *International Journal for Numerical Methods in Engineering*, 103(10):737–758.

Titre : PGD-Abaques virtuels pour l'optimisation géométrique des structures

Mots clés : abaque virtuel, réduction de modèle, PGD, variations de géométrie, paramètres discrets

Résumé : Lors de l'optimisation géométrique de structures, un grand nombre d'évaluations de champs est nécessaire. L'idée, développée dans cette thèse, est la construction efficace et rapide d'approximations de ces champs à travers la *Proper Generalized Decomposition* (PGD), une méthode de réduction de modèle. Les résultats, calculés une fois pour toutes, sont stockés dans des abaques virtuels pour une utilisation ultérieure dans un processus d'optimisation. Le problème considéré est paramétrique et les paramètres définissent la géométrie. Ce type de problème est particulièrement adapté à la PGD. En effet, de nombreux travaux ont traité de la résolution de problèmes paramétriques et des premières études ont porté, en particulier, sur la prise en compte de paramètres géométriques. Toutefois, ce qui caractérise nos travaux est d'aller vers des outils aptes à traiter des

situations significatives de la complexité des problèmes rencontrés au niveau industriel. En particulier, l'exploitation de codes éléments finis commerciaux est une contrainte majeure. Il a été décidé de développer des méthodes permettant de traiter des problèmes à paramètres géométriques par la PGD, et, en partenariat avec AIRBUS Defence & Space, d'appliquer ces techniques à un démonstrateur industriel présentant une géométrie complexe (type splines) et de fortes non-linéarités (géométriques, matériaux). Notre démarche a été implémentée dans un process industriel utilisant des codes éléments finis commerciaux. On propose aussi une nouvelle extension de la PGD aux paramètres discrets autorisant la prise en considération, dans une même résolution, de configurations de structures complètement différentes (cas de chargement, matériaux, etc.).

Title : PGD-Virtual Charts for shape optimisation of structures

Keywords: virtual chart, model reduction, PGD, geometry variations, discrete parameters

Abstract : During shape optimisation of structures, numerous evaluations of fields are necessary. The idea, developed in this PhD thesis, is the efficient construction of approximations of these fields through the *Proper Generalized Decomposition* (PGD), a model reduction technique. The results, computed once and for all, are stored in virtual charts for a subsequent use into an optimisation process. Geometry variations correspond to a parametric problem, where the parameters is the geometry. This kind of problem is well suited for PGD. Many studies dealt with the resolution of parametric problems and recent works treated, particularly, the introduction of geometric parameters. However, our approach is to deal with configurations of the complexity

of industrial problems. The use of commercial finite elements software is a crucial issue. It was decided, in partnership with AIRBUS Defence & Space, to develop techniques allowing the resolution of geometrically parametrised problems thanks to PGD and to apply them to an industrial demonstrator. The geometry considered is defined by splines and the behaviour of the structure is highly non-linear (geometric and material non-linearities). The approach was implemented into a genuine industrial design process using commercial finite elements software. The thesis proposed, also, a new extension of PGD to discrete parameters. It allows to take into account completely different configurations (loadings, materials, etc.) in the same resolution.