université PARIS-SACLAY

école normale supérieure paris-saclay

NNT : 2019SACLN051

Thèse de doctorat

Optimisation de structures viscoplastiques par couplage entre métamodèle multi-fidélité et modèles réduits

Thèse de doctorat de l'Université Paris-Saclay
préparée à École normale supérieure de Paris-Saclay

Ecole doctorale n°579 Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMaG)
Spécialité de doctorat : Solides, structures, matériaux

Thèse présentée et soutenue à l'École Normale Supérieure de Paris-Saclay, le 11 octobre 2019, par

STÉPHANE NACHAR

Composition du Jury :

M. Jean-Charles Passieux Professeur des Universités, INSA Lyon, ICA	(Président du jury)
M. Pierre Duysinx Professeur des Universités, Université de Liège, LTAS	(Rapporteur)
M. Pierre Joyot Maitre de Conférences, ESTIA, I2M	(Rapporteur)
M. Fernando Lopez-Caballero Maître de conférences, CentraleSupelec, MSSMAT	(Examinateur)
M. Felipe Bordeu Ingénieur de Recherche, Safran Tech	(Examinateur)
M. Christian Rey Ingénieur de Recherche, Safran Tech	(Examinateur)
M. Pierre-Alain Boucard Professeur des Universités, ENS Paris-Saclay, LMT	(Directeur de thèse)
M. David Néron Professeur des Universités, ENS Paris-Saclay, LMT	(Co-directeur de thèse)

Remerciements

Ces trois années auront été un réel plaisir tant sur le plan scientifique que sur le plan humain. J'ai adoré travailler sur ce sujet et y développer mes idées et outils dans un cadre idéal.

Tout d'abord, merci à messieurs Joyot et Dussinyx pour avoir accepté la relecture de ce manuscrit, et à l'ensemble des membres du jury pour l'honneur de m'avoir décerné ce grade de docteur. Je remercie mes deux directeurs de thèse, David Néron et Pierre-Alain Boucard pour avoir accepté que je construise cette thèse avec ma vision des choses et ma vision du sujet. PAB, merci d'avoir été toujours là en cas de besoin quelque soit l'heure. Ta franchise est une des raisons pour lesquelles c'est toujours un plaisir de travailler avec toi.

Le laboratoire aura toujours été un cadre de travail plaisant, grâce au travail et à l'ambiance que mettent certaines personnes comme Philippe et Pierre, Lydia, l'ambianceur Boubou et notre vieil ami Frisou. Merci également à mes vieux amis doctorants (Paul, Matthieu, Camille, Maxime, Clement, ...), la team Lebanon (Rana, Farah, Jacques, Kiko) et les moins vieux (Richard, Gab, Myriam, Bhimal, ...) pour les pauses-cafés et autres joyeuseries salvatrices. En particulier, merci à ma vieille compère Marie Tirvaudey toujours là dans les bons coups. Je sais que je peux toujours compter sur toi.

Cette thèse est l'aboutissement de sept années à Cachan, sept années passées trop vite grâce à vous les amis. Merci à la team Physiciens (un gros bisous à Marion), mes amis handballeurs et surtout la team Jap' composée de Lorène, Louis, Yassin, Clotilde, Max, Ronan et Chloé. Mention spéciale à mon successeur Ronan qui est malgré lui l'initiateur du virage "fonctionnel" de cette thèse.

Un grand merci à toute ma famille pour me prouver jour après jour que même si je ne donne pas toujours des nouvelles, vous faites toujours tout pour moi. Merci Hala pour m'avoir donné le goût de l'aéronautique, 3amo et 3amto mes seconds parents, et Jad mon second frère. Caro et Juju, n'oubliez jamais que votre frère est tout aussi fier de vous. Continuez de m'impressionner comme vous le faites. Papa et Maman, mon goût du travail et ma volonté d'aller toujours au bout de mes engagements ne viennent pas de nulle part. C'est à vous deux que je les dois.

Enfin, merci à ma douce et tendre Margaux qui a accepté depuis de nombreuses années la lourde tâche de partager ma vie au quotidien et qui le fait avec merveille. Ce manuscrit est pour toi.

Table des matières

Introduction **8**

I Outils existants pour la conception **13**

1 Résolution d'un modèle mécanique non linéaire paramétré **15**
1.1 Problème mécanique de référence 15
1.2 Loi de comportement élasto-viscoplastique 17
1.2.1 Comportement élastique et plastique 17
1.2.2 Loi de comportement de type Lemaître-Chaboche 17
1.3 Méthodes de résolution de problèmes non linéaires 19
1.3.1 Méthode Newton-Raphson 20
1.3.2 Méthode LATIN 21

2 Stratégie pour l'approximation & l'optimisation globale d'une quantité d'intérêt **24**
2.1 Présentation de la stratégie 24
2.2 Stratégies d'optimisation de structures 26
2.3 Méthodes d'échantillonnage initial dans $\mathcal{D}$ 28
2.4 Méthodes de métamodélisation 29
2.4.1 Régression linéaire 30
2.4.2 Extensions à la régression linéaire 32
2.4.3 Un exemple d'extension : la régression SVM 33
2.5 Régression par processus gaussiens (Krigeage) 33
2.5.1 Définition 33
2.5.2 Fonctions de covariance 35
2.5.3 Détermination des hyperparamètres du métamodèle 36
2.5.4 Optimisation des hyperparamètres 37
2.6 Critères de validation du métamodèle 39

3 Modèles réduits pour les problèmes mécaniques **41**
3.1 Approximation d'ordre réduit 41
3.2 Analyse de données *offline* et interpolation *online* POD 43
3.3 La méthode Reduced Basis 45
3.4 Approximation *online* PGD 45
3.5 Réduction de modèles pour des problèmes non linéaires 47

Bilan **48**

II Contributions aux méthodes employées pour le couplage entre mé-

tamodèles multi-fidélité et modèles réduits **49**

4 Analyse des méthodes de krigeage multi-fidelité **51**
4.1 Méthodes de prise en compte de données multi-fidélité . . . 51
4.2 Méthodes de cokrigeage . . . 53
4.2.1 Généralisation du krigeage aux fonctions multivariées . . . 53
4.2.2 Un exemple de cokrigeage : cokrigeage intercorrélation . . . 54
4.2.3 Cokrigeage auto-régressif . . . 54
4.3 Krigeage avec prise en compte de données bruitées . . . 56
4.4 Méthodes de krigeage récursif . . . 57
4.4.1 Krigeage hiérarchique . . . 57
4.4.2 Evofusion . . . 58

5 Réflexion sur le cadre LATIN-PGD pour les problèmes élastoplastiques **61**
5.1 Détail de l'approche variable interne de la LATIN-PGD . . . 61
5.1.1 Formulation de l'étape locale . . . 62
5.1.2 Calcul de l'opérateur tangent . . . 64
5.1.3 Formulation de l'étape globale . . . 67
5.1.4 Formulation de l'étape de mise à jour . . . 75
5.1.5 Critère d'erreur . . . 76
5.1.6 Analyse de l'approche à variables internes . . . 76
5.2 Approche fonctionnelle de la LATIN . . . 76
5.2.1 Formulation de l'étape globale . . . 77
5.2.2 Choix de la direction de recherche élastique . . . 78
5.3 Implémentation parallèle de la méthode . . . 79
5.3.1 Tensorisation des champs et des opérateurs . . . 79
5.3.2 Parallélisation des calculs . . . 79
5.4 Mise en œuvre de la méthode sur trois cas-tests . . . 81
5.4.1 Cas-test 1 : Barre en traction-compression . . . 81
5.4.2 Cas-test 2 : Pièce d'amortissement . . . 82
5.4.3 Cas-test 3 : Aube de turbine . . . 83

6 Mise en œuvre d'une stratégie multiparamétrique de réexploitation des modes **86**
6.1 Méthodes d'approximation du champ initial . . . 86
6.1.1 Utilisation du champ le plus proche . . . 86
6.1.2 Interpolation de la base spatiale dans la variété Grassmannienne . . . 87
6.1.3 Interpolation des champs et de la base spatiale . . . 88
6.2 Gain apporté par la méthode . . . 90

Bilan **92**

III Mise en œuvre d'une stratégie de couplage robuste **93**

7 Présentation de la campagne d'essais numériques **95**

7.1 Mise en place de la campagne d'essais . . . 95
7.2 Cas-test 1 : Plaque avec inclusion . . . 96
7.2.1 Présentation du problème . . . 96
7.2.2 Analyse de la réponse au problème . . . 97
7.3 Cas-test 2 : Pièce d'amortissement . . . 99
7.3.1 Présentation du problème . . . 99
7.3.2 Analyse de la réponse au problème . . . 100
7.4 Cas-test 3 : Aube de turbine . . . 102
7.4.1 Présentation du problème . . . 102
7.4.2 Analyse de la réponse au problème . . . 103

8 Génération rapide d'un abaque virtuel de la quantité d'intérêt 106
8.1 Généralisation de la stratégie au krigeage multi-fidélité . . . 106
8.2 Détermination des paramètres de la stratégie . . . 107
8.2.1 Génération complète du métamodèle — Cas-test 1 . . . 107
8.2.2 Génération complète du métamodèle — Cas-test 2 . . . 108
8.3 Validation sur le cas-test de l'aube . . . 109

9 Optimisation bayésienne de la quantité d'intérêt 110
9.1 Généralisation de la stratégie au krigeage multi-fidélité . . . 110
9.1.1 Critères d'enrichissement . . . 111
9.1.2 Amélioration de la stratégie dans le cas du krigeage multi-fidélité . . . 113
9.2 Détermination des paramètres de la stratégie . . . 114
9.2.1 Recherche du minimum global — Cas-test 1 . . . 114
9.2.2 Recherche du maximum global — Cas-test 2 . . . 115
9.3 Validation sur le cas-test de l'aube . . . 117

Bilan 118

Conclusion 119

Bibliographie 122

Introduction

Les phases de conception et de validation de pièces mécaniques nécessitent des outils de calculs rapides et fiables, permettant de faire des choix technologiques en un temps court. Le contexte industriel considéré dans cette thèse est celui de l'entreprise aéronautique Safran. Une des activités de Safran est de concevoir, dans le cadre du consortium CFM International, les turboréacteurs d'avions commerciaux. L'actuelle génération CFM LEAP met en avant le perfectionnement de leurs aubes de soufflantes, leur ayant permis un gain de masse de 450 kg par rapport à la génération CFM56. Ce gain a notamment été permis par l'utilisation d'outils de simulation numérique, limitant ainsi l'emploi de prototypes physiques longs à produire et coûteux.

Néanmoins, les différents outils numériques nécessitent un compromis entre le temps de calcul et la qualité du modèle virtuel. En effet, pour permettre une bonne estimation du comportement des pièces, le modèle numérique doit être capable de prendre en compte des lois de comportement matériaux fortement non linéaires, ainsi que des détails structuraux comme les trous de refroidissement inférieurs aux dimensions de la pièce de plusieurs ordres de grandeur. La résolution des équations aux dérivées partielles associées à un modèle élasto-viscoplastique nécessite plusieurs heures de calcul pour obtenir une cartographie de la déformation plastique au cours des cycles de chargement.

Pour la phase d'avant-projet, de nombreuses méthodes d'aide à la décision en temps réduit ont été développées, mais il est encore peu fréquent de voir un couplage de plusieurs de ces méthodes. Il serait donc essentiel pour le bureau d'étude d'avoir un chaînage pertinant des outils numériques les plus aboutis, tout en étant capable d'exploiter au maximum l'architecture parallèle actuelle, pour leurs différents objectifs présentés ci-dessous.

- Le premier objectif présenté est le plus adapté aux outils existants. L'objectif est un essai de qualification dans lequel la réponse à un certain cas d'étude est testée sur la configuration choisie. Il s'agit d'une simulation à caractère exceptionnel, réalisé avec le modèle le plus précis possible.
- Le second objectif a pour but d'améliorer les règles de conception par la génération d'abaques, dits *virtuels*, permettant d'accéder rapidement à l'influence de modifications sur le produit selon certains critères de conception. Il s'agit donc *a contrario* ici d'utiliser l'abaque pour des études récurrentes sur un même problème. L'abaque peut être employé en phase de conception, mais également en phase de validation pour la prise en compte d'incertitudes [Sudret, 2012, Lopez-Caballero et Khalil, 2018].
- Dans un troisième cas, la volonté n'est pas d'établir des règles de conception, mais de directement rechercher le meilleur choix de conception parmi l'ensemble des possibilités en réalisant une *optimisation* globale, potentiellement sous certaines contraintes. De multiples méthodes d'optimisation existent pour permettre une optimisation géométrique [Allaire, 2007], une optimisation paramétrique [Duysinx, 1996] ou encore une

optimisation topologique [Bendsøe, 1995]. Il s'agit donc ici d'une étude ponctuelle, pour laquelle on souhaite résoudre le problème une seule fois, le plus rapidement possible.

Ces requêtes nécessitent de tester une ou plusieurs conceptions possibles. Chacun de ces choix est caractérisé par d paramètres de conception scalaires. Ces paramètres définissent une variable $\underline{x}$ appartenant à l'espace comportant l'ensemble des possibilités de conception, appelé l'*espace de conception* $\mathcal{D} \in \mathbb{R}^d$. Pour chacune des configurations $\underline{x} \in \mathcal{D}$, le test du prototype associé permet d'en extraire une *quantité d'intérêt* scalaire y nommée dans la suite QoI (pour Quantity of Interest), et aussi éventuellement de valider des contraintes de conception.

Une première approche pour accélérer la résolution des équations aux dérivées partielles (EDP) pour une configuration $\underline{x}$ donnée consiste à modifier la méthode de résolution non linéaire du modèle mécanique. Dans les codes de calcul commerciaux, la méthode itérative de Newton ou les méthodes quasi-Newton sont couramment employées. Ces techniques permettent l'obtention de la solution $\mathbf{s}$ du problème mécanique par calcul de la solution exacte sur l'ensemble de l'espace Ω de manière incrémentale sur l'espace temporel I. D'autres méthodes plus adaptées à notre contexte existent également comme la méthode LATIN [Ladevèze, 1985]. Cette méthode de résolution permet la génération de la solution $\mathbf{s}$ du problème mécanique par corrections successives d'une initialisation grossière $\mathbf{s}_0$ sur l'ensemble du domaine spatio-temporel. La méthode s'appuie sur un partitionnement du problème en deux sous-problèmes : un problème non linéaire local en espace, et un problème linéaire global en temps et en espace. Avec cette méthode, une approximation de la solution du problème spatio-temporel complet est exploitable $\mathbf{s}_0(M,t)$, permettant ainsi de calculer une approximation de la QoI. La solution à convergence $\mathbf{s}$ permet le calcul exact de la QoI $y(\underline{x})$.

Pour la génération d'un abaque virtuel ou pour une optimisation globale, l'ensemble des configurations ne peuvent évidemment pas être testées dans un temps raisonnable. Pour pallier cette limitation, il est possible de substituer la résolution des EDP par l'appel à un modèle réduit. Le modèle réduit est obtenu par la recherche d'une approximation de rang m $\underline{\tilde{u}}_m$ des champs du problème spatio-temporel paramétré par $\underline{x}$. Le champ $\underline{\tilde{u}}_m$ s'écrit alors sous la forme d'une somme de m fonctions de l'espace $\underline{\Lambda}_k$, du temps λ_k et potentiellement d'extra-paramètres $\lambda_k^{(p)}$:

$$\underline{\tilde{u}}_m(M,t,\mu) = \sum_{k=1}^{m} \underline{\Lambda}_k(M) \cdot \lambda_k(t) \cdot \lambda_k^{(p)}(\mu) \tag{1}$$

Le principal avantage de ces techniques est que le temps de calcul est directement corrélé au nombre m de modes générés, et donc à la qualité de l'approximation. Parmi ces méthodes, on peut distinguer deux grandes familles :

- Une première concerne les méthodes avec apprentissage se basant sur la résolution d'un problème de substitution comme la Proper Orthogonalized Decomposition (POD) [Ryckelynck *et al.*, 2006, Lieu *et al.*, 2006], ou la résolution de problèmes similaires comme la méthode Reduced-Basis [Maday et Ronquist, 2004, Rozza *et al.*, 2008]. Les champs mécaniques spatio-temporels sont calculés dans un premier temps pour différentes configurations $\underline{x} \in \mathcal{D}$. La décomposition de ces *snapshots* va permettre de générer

une base de m modes spatiaux optimaux $(\underline{\Lambda}_k) \in \mathcal{U}^m$. Dans le cas de la POD, les équations du problème seront projetées sur cette base réduite pour générer un problème de substitution. La résolution de ce problème pour d'autres configurations $\underline{x}^* \in \mathcal{D}$ consistera alors à calculer la meilleure combinaison des vecteurs de la base. Ces méthodes sont attractives, car elles permettent de s'appuyer sur les codes de calcul existants et certifiés. Néanmoins, ils nécessitent une phase en amont potentiellement longue et coûteuse, et un choix judicieux de snapshots.

- La seconde concerne les méthodes sans apprentissage comme la Proper Generalized Decomposition (PGD) [Chinesta *et al.*, 2010, Ammar *et al.*, 2012, Néron *et al.*, 2015]. Ces méthodes sont plus adaptées à notre cas d'étude, car la solution du problème paramétré est recherchée en même temps que la meilleure base pour sa représentation. Ces méthodes permettent d'obtenir une précision du modèle réduit uniquement dans les zones intéressantes de l'espace de conception par un enrichissement progressif de la base de modes.

Par ailleurs, cet enrichissement progressif de la base de modes a été imaginée initialement pour la méthode LATIN et est parfaitement adaptée à celle-ci. En effet, les corrections successives des champs peut être recherchée sous une forme réduite PGD temps-espace pour accélérer le temps de calcul. C'est pour cela que la méthode LATIN est aussi souvent qualifiée de LATIN-PGD.

Ce cadre peut être amélioré dans le cas de multiples résolutions du problème paramétrique. En effet, la différence de comportement entre deux configurations testées $\underline{x} \in \mathcal{D}$ est souvent faible. Néanmoins, le second appel au solveur a le même coût de calcul que le premier. Il serait donc attractif de tirer profit des champs déjà obtenus pour approximer le champ initial $\mathbf{s}_0$ de la méthode LATIN pour le calcul d'une nouvelle configuration $\underline{x}^*$. Cette stratégie multi-paramétrique a été introduite par [Boucard et Ladevèze, 1999].

Ces méthodes ont un intérêt pour avoir une approximation des champs solution pour tout $\underline{x} \in \mathcal{D}$, mais elles apportent un surplus d'informations non nécessaires. En effet, pour permettre le choix de conception, la seule réponse attendue est la QoI. Pour réduire le nombre d'appels au solveur, il est possible d'approximer la QoI $\underline{x} \in \mathcal{D} \mapsto y(\underline{x}) \in \mathbb{R}$ et de la remplacer par une fonction de substitution nommée *métamodèle* $\underline{x} \in \mathcal{D} \mapsto \hat{y}(\underline{x}) \in \mathbb{R}$. Cette fonction est construite à partir d'un ensemble d'observations sur n points sélectionnés $\underline{x}_i$ et des réponses correspondantes de la QoI $y_i = y(\underline{x}_i)$. Toute réponse $\hat{y}(\underline{x})$ est *prédite* de manière immédiate à partir de ce métamodèle généré. La difficulté réside dans le moyen de qualifier la précision du métamodèle, le nombre de données nécessaires à la création d'un métamodèle suffisamment précis, et le temps nécessaire à l'obtention de ces données. De multiples méthodes de métamodélisation existent dont une présentation large est proposée dans [De Lozzo, 2015]. La méthode des surfaces de réponse, élaborée comme outil d'aide aux manipulations expérimentales et de dépouillement de résultats, a laissé place à des méthodes plus adaptées comme les réseaux de neurones [McCulloch et Pitts, 1943, Hebb, 2005], la régression par processus gaussien ou krigeage [Matheron, 1963, Rasmussen et Williams, 2006], l'interpolation par fonctions à base radiale [Broomhead et Lowe, 1988] ou encore les machines à vecteurs de support [Vapnik *et al.*, 1996]. De précédents travaux sur le krigeage ont permis de mettre en évidence

leur avantage pour l'analyse de données complexes, éventuellement bruitées.

Ces méthodes nécessitent malgré tout un certain nombre d'observations pour obtenir un métamodèle juste, ou pour s'assurer d'obtenir l'optimum global. Pour limiter le coût de calcul des observations, une première approximation globale de la fonction coût est générée en considérant un grand nombre d'observations obtenues à partir d'une approximation de la solution. Cette approximation est ensuite exploitée pour réaliser des calculs exacts dans des zones d'intérêt. Ces zones peuvent être celles où la probabilité d'obtenir l'optimum est grande. Une limitation à cette stratégie concerne la prise en compte de ces données multi-fidélités dans un unique métamodèle. L'emploi de multiples modèles mécaniques est permis par l'existence de plusieurs méthodes de krigeage multi-fidélité [Forrester *et al.*, 2006, Han *et al.*, 2012, Zimmerman et Holland, 2005].

Cette thèse s'inscrit donc dans l'élaboration d'une stratégie adaptative et d'outils rapides d'aide à la conception et à la validation de produits mécaniques, en collaboration avec Safran. Une stratégie d'apprentissage automatique adaptée à la création d'abaques virtuels et à l'optimisation a été développée, couplant l'emploi de métamodèles multi-fidélité et la génération de modèles réduits.

Le premier niveau d'amélioration se situe au niveau du solveur mécanique non linéaire. La résolution des problèmes élasto-viscoplastiques est faite avec la méthode LATIN-PGD, en s'appuyant notamment sur les travaux de [Passieux, 2008] et [Relun *et al.*, 2013]. Afin d'exploiter l'architecture HPC actuelle tout en simplifiant son élaboration, la méthode a été implémentée dans un démonstrateur orienté objet Matlab/OpenMP. Ce démonstrateur a permis le calcul approché d'un cycle de chargement extrême d'une aube à 1.7 MDOFs en 45mn, et le calcul exact en 1h27mn sur les 20 cœurs Intel® Xeon® E5-2630 d'une station de travail.

Le second niveau d'amélioration se situe dans la mise en place d'une stratégie multiparamétrique. Celle-ci permet l'initialisation du calcul à partir d'une interpolation par pondération inverse à la distance (Inverse distance weighting — IDW) des solutions précédemment calculés. L'utilisation de la stratégie permet des gains de temps de l'ordre de 4× sur 100 configurations. Ce solveur LATIN-PGD avec la stratégie multiparamétrique a été exploité par la suite comme une boîte noire pour le calcul des champs et des QoI dans le cadre de la génération adaptative d'abaques virtuels et dans le cadre d'une optimisation globale de la QoI.

Le troisième niveau d'amélioration porte sur la méthode de krigeage multi-fidélité employée. La génération de métamodèles de krigeage multi-fidélité s'appuie sur un premier comparatif des méthodes existantes réalisé durant la thèse de [Courrier *et al.*, 2016] au sein du LMT. Une analyse des méthodes de krigeage multi-fidélité et du krigeage avec données bruitées a été effectuée et a conduit à une amélioration de la méthode Evofusion avec la prise en compte d'une variance sur les données basse fidélité.

Ces méthodes ont été couplées dans une stratégie de génération de métamodèle adaptée à la génération d'abaques virtuels et à l'optimisation. L'utilisation du krigeage multi-fidélité et les ajouts successifs de points $\underline{x}$ maximisant le gain d'information estimé ont permis un gain de l'ordre de 15× par rapport au krigeage avec des données uniquement exactes à même qualité de métamodèle. L'ensemble de ces améliorations ont permis de mettre en œuvre la

génération d'abaques virtuels et l'optimisation d'une aube test de turboréacteur, permettant de calculer les paramètres optimaux de celle-ci en 44 minutes au lieu de 6 heures, soit un gain d'un facteur 8.

Le manuscrit s'articulera autour de trois parties :

- La première partie s'intéressera aux outils existants en bureau des méthodes. Le problème élasto-viscoplastique sera présenté avec les méthodes de résolution non linéaires usuellement employées. Pour l'exploitation de multiples calculs, différentes stratégies d'étude de la réponse du problème seront présentées et plusieurs méthodes de métamodélisation couramment employées par ces stratégies seront évoquées. L'accent sera plus particulièrement mis sur l'utilisation du krigeage. Enfin, des modèles réduits étant utilisés pour accélérer l'étude de l'espace paramétrique et des différentes configurations, le dernier chapitre aura pour objet les méthodes associées et particulièrement la méthode PGD.
- La seconde partie présentera dans un premier temps une analyse des méthodes pour la prise en compte des données multi-fidélité dans le cadre du krigeage. Les différentes méthodes présentées conduisent à une proposition d'amélioration de la méthode Evofusion. Le chapitre suivant présentera l'apport de la LATIN-PGD sur la résolution d'EDP non linéaire. Les deux approches de la méthode ont été implémentées. La recherche d'une implémentation HPC a permis de mettre en place une vision tensorielle des opérateurs et de nombreuses contributions techniques ont été apportées dans ce cadre. Les approches ont ensuite été comparées à partir de cas-test de taille croissante. Le dernier chapitre présente la stratégie multi-paramétrique basée sur l'utilisation de la méthode LATIN-PGD. Différentes méthodes d'interpolation multivariée sont présentées et l'interpolation IDW choisie a été appliquée à la résolution de multiples problèmes similaires.
- La troisième partie explicitera le couplage des méthodes par l'instauration d'un algorithme robuste de génération de métamodèle. Les différents cas-tests seront présentés ainsi que l'analyse de la réponse aux problèmes. Ces cas-tests vont permettre de tester et valider une stratégie d'enrichissement du métamodèle robuste et rapide adaptée à la génération d'un abaque virtuel, puis une stratégie d'enrichissement adaptée à la recherche de l'optimum global du problème.

Première partie

Outils existants pour la conception

Chapitre 1

Résolution d'un modèle mécanique non linéaire paramétré

Sommaire

1.1 Problème mécanique de référence 15
1.2 Loi de comportement élasto-viscoplastique 17
1.2.1 Comportement élastique et plastique 17
1.2.2 Loi de comportement de type Lemaître-Chaboche 17
1.3 Méthodes de résolution de problèmes non linéaires 19
1.3.1 Méthode Newton-Raphson 20
1.3.2 Méthode LATIN 21

Dans le cadre de cette thèse, on s'intéresse à un problème mécanique quasi-statique dont le matériau est associé à un comportement élasto-viscoplastique non linéaire. Deux méthodes de résolution de ce problème seront également présentées.

1.1 Problème mécanique de référence

On rappelle dans cette partie le cadre de la mécanique des milieux continus qui va nous intéresser dans ce manuscrit. Une description plus détaillée est présentée dans [Ciarlet, 1988].

Le problème de référence étudié est un problème d'évolution isotherme quasi-statique défini sur le domaine spatio-temporel $\Omega \times I$. On considère une structure soumise à des forces volumiques internes $\underline{f}_d$, à des forces surfaciques $\underline{F}_d$ sur le bord $\partial_2\Omega$, et à des déplacements imposés $\underline{U}_d$ sur le bord $\partial_1\Omega$.

Le mouvement au sein de la structure se traduit par un champ de déplacement vis-à-vis de la configuration initiale $\underline{U} : \Omega \times I \mapsto \mathbb{R}^3$. On note S^3 l'espace des matrices symétriques 3×3. L'hypothèse des petites perturbations est supposée, permettant la linéarisation du champ de déformation de la structure $\boldsymbol{\varepsilon} : \Omega \times I \mapsto S^3$ avec $\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla \underline{U} + \nabla^T \underline{U})$. La schématisation des efforts intérieurs se traduit par le champ de contrainte $\boldsymbol{\sigma} : \Omega \times I \mapsto S^3$ vérifiant le théorème de Cauchy et le problème aux conditions limites suivant :

$$\underline{\mathrm{div}}(\boldsymbol{\sigma}) + \underline{f}_d = 0 \text{ sur } \Omega, \quad \boldsymbol{\sigma}\underline{n} = \underline{F}_d \text{ sur } \partial_2\Omega, \quad \underline{U} = \underline{U}_d \text{ sur } \partial_1\Omega \tag{1.1}$$

Le comportement du matériau composant la structure est modélisé par une loi de comportement non linéaire à variables internes explicitée dans la section suivante. Cette loi peut s'écrire à partir de l'histoire de la déformation au cours du temps (1.2), ou à partir de l'état actuel du matériau et donc des variables internes $\dot{\boldsymbol{\varepsilon}}_p$ et $\underline{V}_k$ (1.3). Les deux notations seront employées en fonction des besoins dans le manuscrit :

$$\forall (M,t) \in \Omega \times I, \quad \boldsymbol{\sigma}(M,t) = \mathcal{B}(\boldsymbol{\varepsilon}(M,\tau), \tau \in [0,t]) \tag{1.2}$$

$$\forall (M,t) \in \Omega \times I, \quad \dot{\boldsymbol{\varepsilon}}_p(M,t) = \mathbb{B}(\boldsymbol{\sigma}(M,t), V_k(M,t)) \tag{1.3}$$

Revenons au problème (1.1), vérifier ce problème à un temps $t \in I$ est équivalent à minimiser la fonctionnelle énergétique $\mathcal{E} : \mathcal{U} \mapsto \mathbb{R}$ définie par :

$$\mathcal{E}(\underline{\mathrm{U}}) = \int_\Omega \mathcal{B}(\boldsymbol{\varepsilon}) : \boldsymbol{\varepsilon}\, \mathrm{d}\Omega - \int_\Omega \underline{\mathrm{f}}_d \cdot \underline{\mathrm{U}} \mathrm{d}\Omega - \int_{\partial_2\Omega} \underline{\mathrm{F}}_d \cdot \underline{\mathrm{U}} \mathrm{dS} \tag{1.4}$$

avec $\mathcal{U} = \{\underline{\mathrm{U}} \in \mathrm{H}^{1,p}(\Omega) \ / \ \mathcal{E}(\underline{\mathrm{U}}) \leqslant +\infty,\ \underline{\mathrm{U}} = \underline{\mathrm{U}}_d \text{ sur } \partial_1\Omega\}$ et $\mathrm{H}^{1,p}(\Omega)$ l'espace de Sobolev dans $L^p(\Omega, \mathbb{R}^3)$ à dérivées partielles dans $L^p(\Omega, \mathbb{R}^3)$ avec $p \geqslant 1$.

Notons $\mathcal{W} = \{\underline{\mathrm{U}} : \Omega \times I \mapsto \mathbb{R}^3 / \forall t \in I, M \mapsto \underline{\mathrm{U}}(M,t) \in \mathcal{U},\ \int_I \mathcal{E}(\underline{\mathrm{U}})\, \mathrm{dt} \leqslant +\infty\}$, la minimisation de la fonctionnelle énergétique (1.4) peut se traduire par la formulation faible suivante :

$$\text{Trouver } \underline{\mathrm{U}} \in \mathcal{W} / \forall t \in I, \forall \underline{\mathrm{U}}^* \in \mathcal{W}^*, \mathcal{R}\left(\underline{\mathrm{U}}(M,t); \underline{\mathrm{U}}^*(M,t)\right) = 0 \tag{1.5}$$

avec $\mathcal{W}^*$ l'espace obtenu à partir de $\mathcal{W}$ avec une mise à zéro de la condition limite $\underline{\mathrm{U}}_d = \underline{0}$, et avec le résidu $\mathcal{R}$ défini par :

$$\mathcal{R}(\underline{\mathrm{U}}; \underline{\mathrm{U}}^*) = \int_{I\times\Omega} \mathcal{B}(\boldsymbol{\varepsilon}[\underline{\mathrm{U}}]) : \boldsymbol{\varepsilon}(\underline{\mathrm{U}}^*)\, \mathrm{d}\Omega\mathrm{dt} - \int_{I\times\Omega} \underline{\mathrm{f}}_d \cdot \underline{\mathrm{U}}^*\, \mathrm{d}\Omega\mathrm{dt} - \int_{I\times\partial_2\Omega} \underline{\mathrm{F}}_d \cdot \underline{\mathrm{U}}^*\, \mathrm{dsdt} \tag{1.6}$$

Le résidu peut se séparer en deux termes :

— un opérateur non-linéaire continu noté $a : \mathcal{W} \times \mathcal{W} \mapsto \mathbb{R}$ tel que

$$\forall \underline{u}, \underline{v} \in \mathcal{W},\ a(\underline{u}, \underline{v}) = \int_{I\times\Omega} \mathcal{B}(\boldsymbol{\varepsilon}[\underline{u}]) : \boldsymbol{\varepsilon}(\underline{v})\, \mathrm{d}\Omega\mathrm{dt} \tag{1.7}$$

— un opérateur linéaire continu noté $\ell : \mathcal{W} \mapsto \mathbb{R}$ tel que

$$\forall \underline{v} \in \mathcal{W},\ \ell(\underline{v}) = \int_{I\times\Omega} \underline{\mathrm{f}}_d \cdot \underline{v}\, \mathrm{d}\Omega\mathrm{dt} + \int_{I\times\partial_2\Omega} \underline{\mathrm{F}}_d \cdot \underline{v}\, \mathrm{dsdt} \tag{1.8}$$

En l'absence de méthode directe donnant un accès immédiat à la solution du problème (1.5), il est nécessaire de définir un algorithme itératif permettant la résolution du problème par linéarisation de celui-ci. Deux de ces méthodes seront décrites en section 1.3.

1.2 Loi de comportement élasto-viscoplastique

1.2.1 Comportement élastique et plastique

Le matériau employé dans les exemples est un matériau métallique soumis à de fortes températures. Dans le cas de faibles sollicitations, celui-ci a un comportement réversible et la loi de Hooke (1.9) définit le comportement élastique liant la déformation $\boldsymbol{\varepsilon}$ à la contrainte $\boldsymbol{\sigma}$. Ce comportement est caractérisé par le module d'Young E et le coefficient de Poisson ν :

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon} = \frac{E}{1+\nu}\left(\boldsymbol{\varepsilon} + \frac{\nu}{1-2\nu}\mathrm{tr}\left(\boldsymbol{\varepsilon}\right)\mathbb{1}\right) \tag{1.9}$$

Cette constatation est modélisée par la notion de limite d'élasticité. Par exemple, en traction-compression simple dans la direction x, on observera un comportement élastique linéaire tant que $|\sigma_{xx}| - \sigma_0 \leq 0$, σ_0 étant dans ce cas la limite d'élasticité.

On définit alors un critère $f(\boldsymbol{\sigma})$ définissant le domaine d'élasticité comme étant l'ensemble des contraintes telles que le critère est négatif. Dans le cas d'étude, il s'agit du critère de Von Mises définissant un ellipsoïde dans l'espace des contraines principales déviatoriques $\boldsymbol{\sigma}_D = \boldsymbol{\sigma} - \frac{1}{3}\mathrm{tr}\left(\boldsymbol{\sigma}\right)\mathbb{1}$ comme présenté figure 1.1.

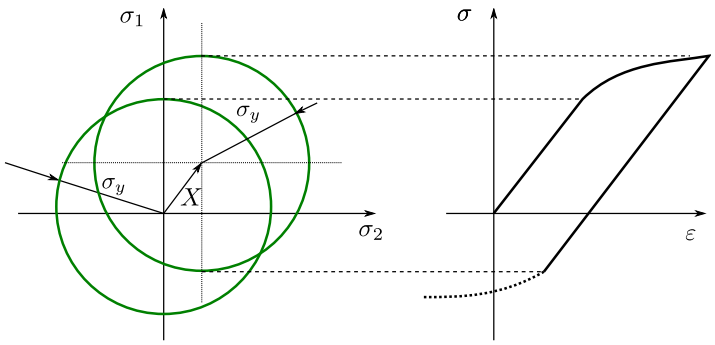


FIGURE 1.1 – Schématisation du comportement élasto-viscoplastique

Mais lorsque le chargement mécanique atteint la valeur seuil, le comportement du matériau devient non linéaire et irréversible : il se déforme plastiquement. Des déformations permanentes apparaissent après un cycle charge-décharge contrairement au cadre élastique. La variation de la surface seuil sera alors fonction de la déformation plastique créée dans le matériau.

1.2.2 Loi de comportement de type Lemaître-Chaboche

On s'intéresse à la modélisation de Lemaître-Chaboche [Lemaitre et Chaboche, 1994] du comportement élasto-viscoplastique de ce type de matériau. Une description macroscopique de celui-ci sera présentée ici.

Le comportement du matériau se décrit à partir des variables d'état observables que sont la déformation totale $\boldsymbol{\varepsilon}$, et des variables internes décrivant l'état actuel du matériau. Ces variables internes sont la déformation inélastique $\boldsymbol{\varepsilon_p}$ et les variables d'écrouissage p et $\boldsymbol{\alpha}$, associées aux forces thermodynamiques R et $\mathbf{X}$.

Ce domaine évolue en fonction de l'état actuel du matériau et se traduit par l'équation (1.10) avec $\boldsymbol{\sigma}_D = \boldsymbol{\sigma} - \frac{1}{3}\mathrm{tr}\,(\boldsymbol{\sigma})\,\mathbb{1}$ la contrainte déviatorique, $\boldsymbol{\sigma}_{eq} = \sqrt{\frac{3}{2}\boldsymbol{\sigma}_D : \boldsymbol{\sigma}_D}$ la contrainte équivalente de Von Mises, et σ_y la taille de la zone élastique initiale :

$$f = (\boldsymbol{\sigma} - \mathbf{X})_{eq} - (\sigma_y - R) \tag{1.10}$$

$$\begin{cases} f \leqslant 0 \longrightarrow \text{Comportement élastique} \\ f \geqslant 0 \longrightarrow \text{Comportement viscoplastique} \end{cases}$$

La dimension et le centre du domaine actuel d'élasticité sont pilotés par l'écrouissage statique R et un unique écrouissage cinématique linéaire $\mathbf{X}$. Dans le cas de chargement cyclique à grand nombre de cycles, deux d'écrouissages cinématiques peuvent être à considérer pour traduire le comportement à faible et à grand nombre de cycles.

La variable primale associée à l'écrouissage isotrope R est la déformation plastique cumulée p définie au cours du temps par :

$$p(t) = \int_{\tau=0}^{T} \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}_p(\tau) : \dot{\boldsymbol{\varepsilon}}_p(\tau)}\, d\tau \tag{1.11}$$

Le taux de déformation plastique $\dot{p}$ est piloté par la loi de Norton-Hoff (1.12) avec k, N deux coefficients scalaires et $\langle \bullet \rangle_+$ les brackets de Macaulay désignant la partie positive.

$$\dot{p} = \left\langle \frac{f}{k} \right\rangle_+^N \tag{1.12}$$

Les équations d'état associées sont :

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon_e} \tag{1.13}$$

$$\mathbf{X} = \frac{2}{3}C\boldsymbol{\alpha} \tag{1.14}$$

$$R = R_\infty(1 - e^{-bp}) \tag{1.15}$$

avec C, R_∞, b des coefficients scalaires.

Et à partir du pseudo-potentiel de dissipation F (1.16), les équations d'évolution sont définies par l'équation (1.17) avec $\mathbb{N}$ le vecteur normal unitaire ($(\mathbb{N})_{eq} = \sqrt{\frac{3}{2}\mathbb{N}_D : \mathbb{N}_D} = 1$).

$$F = f + \frac{3\gamma}{4C}\mathbf{X} : \mathbf{X} - \frac{2\gamma C}{3}\boldsymbol{\alpha} : \boldsymbol{\alpha} \tag{1.16}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\boldsymbol{\varepsilon_p}\\ -\boldsymbol{\alpha}\\ -p\end{bmatrix} = \dot{p}\begin{bmatrix}\frac{\partial F}{\partial \boldsymbol{\sigma}}\\ \frac{\partial F}{\partial \mathbf{X}}\\ \frac{\partial F}{\partial R}\end{bmatrix} = \left\langle \frac{f}{k}\right\rangle_+^N \begin{bmatrix}\frac{3}{2}\mathbb{N}\\ -\frac{3}{2}\mathbb{N}+\frac{3\gamma}{2C}\mathbf{X}\\ -1\end{bmatrix} \tag{1.17}$$

$$\text{avec } \mathbb{N} = \frac{\boldsymbol{\sigma}_D - \mathbf{X}}{(\boldsymbol{\sigma}_D - \mathbf{X})_{eq}}$$

La formulation de l'écrouissage isotrope R permet la stabilisation cyclique : lorsque les cycles avec plasticité s'accumulent, le taux de déformation plastique p augmente indéfiniment, mais l'écrouissage isotrope R se stabilise. Néanmoins, cette formulation est dite *non-normale* dans la mesure où l'équation d'état (1.15) permettant de lier les variables primale et duale n'est plus linéaire. Cette remarque est à mettre en regard de la méthode LATIN présentée en Section 1.3. Dans le problème de référence présenté en Section 1.1, les champs duaux $\underline{V}_k$ correspondent à l'écrouissage cinématique linéaire $\mathbf{X}$ et isotrope R.

1.3 Méthodes de résolution de problèmes non linéaires

Il existe une très grande variété de méthodes de résolution numérique de problèmes aux EDP non linéaires spatio-temporelles. L'approche de type Newton est la plus couramment employée. Néanmoins dans le cadre de ces travaux, la méthode LATIN présente le triple intérêt de permettre aisément l'exploitation des informations précédemment calculées, la génération d'une solution sous forme à variables séparées et un contrôle de la qualité de la solution spatio-temporelle complète.

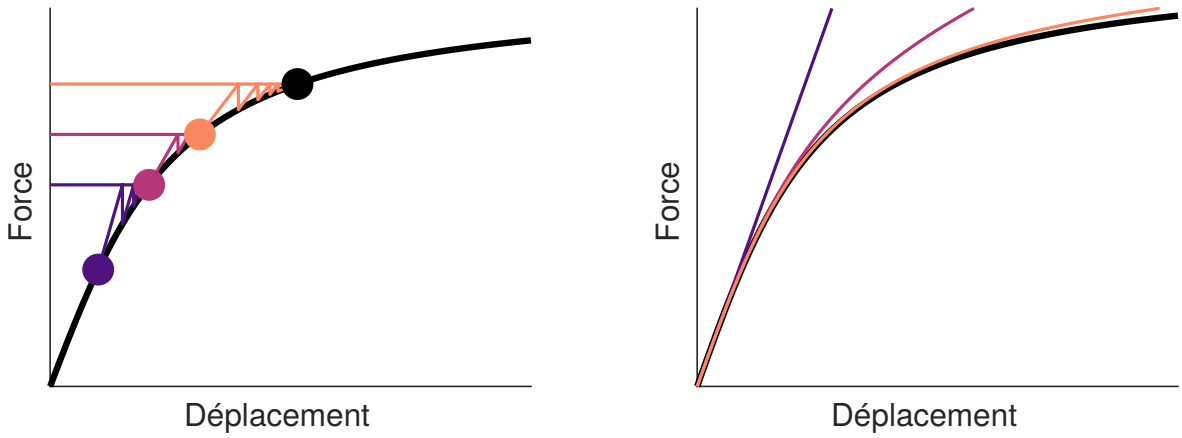


FIGURE 1.2 – Comparaison schématique entre la méthode Newton-Raphson (à gauche) et la méthode LATIN (à droite)

1.3.1 Méthode Newton-Raphson

Les méthodes de type Newton-Raphson sont des méthodes de recherche d'un point d'annulation d'une fonction réelle par l'écriture à chaque itération du développement limité à l'ordre 1 du résidu autour de l'itéré précédent. Sa convergence peut être contrôlée par le résidu et est quadratique localement, ce qui en fait une méthode robuste. Son inconvénient est qu'elle nécessite le calcul de la dérivée de la fonction à chaque pas de convergence ce qui peut être impossible ou coûteux.

Les méthodes dites *Newton modifiée* pallient ce problème au prix d'un taux de convergence moindre. Elles consistent à ne pas calculer la dérivée à chaque pas de convergence, mais une unique fois (méthode de Newton modifiée avec direction constante, de convergence locale linéaire) ou encore d'utiliser une direction sécante (convergence locale de ratio $\alpha = \frac{1+\sqrt{5}}{2}$).

La résolution de l'équation (1.5) issue du problème de référence est faite à chaque pas de temps. Cette représentation conduit à une équation vectorielle non linéaire sur les déplacements nodaux $\underline{\mathrm{U}}$.

$$\mathcal{R}(\underline{\mathrm{U}}; t) = \mathbf{a}(\underline{\mathrm{U}}; t) - \underline{\ell}(t) = \underline{0} \tag{1.18}$$

avec $\mathbf{a}$ un opérateur différentiel non linéaire. À l'itération k de la résolution, la méthode de Newton consiste à calculer la matrice de rigidité $\mathbf{K}^{(k)}$ linéarisant $\mathbf{a}$ au point $\underline{\mathrm{U}}^{(k)}$, et à l'utiliser pour calculer la correction $\delta\underline{\mathrm{U}}^{(k)}$ à apporter à $\underline{\mathrm{U}}^{(k)}$ pour s'approcher de la solution. L'algorithme 1 détaille les étapes.

Algorithm 1: Méthode de Newton

- **Initialisation** : Pas de temps initial $\underline{\mathrm{U}}_0$
- **Boucle sur les pas de temps** i_t
  - **Initialisation de la boucle** : $\underline{\mathrm{U}}_{i_t} \longleftarrow \underline{\mathrm{U}}_{i_t-1}$, $R_{i_t} = \mathcal{R}_{i_t}(\underline{\mathrm{U}}_{i_t})$
  - **Boucle de convergence - Tant que** $||R_{i_t}|| > \nu$
    - **Pour chaque element** i_e
      - **Pour chaque point de Gauss** i_g
        - Intégration de la loi de comportement
        - Linéarisation du comportement local
        - Calcul de la matrice de rigidité locale
    - Assemblage de la matrice de rigidité tangente associée $\mathbf{K}^{(k)}$
    - Calcul de la correction $\delta\underline{\mathrm{U}}_{i_t}$ par résolution du système linéaire associé
    - **Pour chaque element** i_e
      - **Pour chaque point de Gauss** i_g
        - Calcul du résidu $\mathcal{R}_{i_t}(\underline{\mathrm{U}}) = \mathbf{a}(\underline{\mathrm{U}}_{i_t}) - \underline{\ell}_{i_t}$

La matrice de rigidité tangente $\mathbf{K}^{(k)}$ est symétrique définie positive grâce à l'hypothèse de convexité stricte de la loi de comportement. La matrice de rigidité tangente $\mathbf{K}^{(k)}$ correspond à l'assemblage d'une matrice de rigidité pour un comportement linéaire fictif issu de la linéarisation de la loi de comportement $\mathcal{B}$ pour $\underline{\mathrm{U}}_{i_t}^{(k)}$. Il est donc nécessaire à chaque pas de

convergence de calculer le gradient de la loi, assembler la matrice et la factoriser (au moyen de l'algorithme LDL^T par exemple). Ce coût est équivalent à la réalisation d'un calcul élastique.

Il est avantageux dans certains cas d'employer une méthode de Newton modifiée avec une matrice de rigidité constante au cours des itérations de convergence, voire au cours des pas de chargement. Cette variante a des propriétés de convergence dégradées, mais permet de réaliser une unique étape assemblage-factorisation au cours du calcul dans le cas où la matrice employée est la matrice de rigidité élastique.

1.3.2 Méthode LATIN

La méthode LATIN [Ladevèze, 1985, Ladevèze, 1999] est une méthode de résolution reposant sur la séparation des difficultés. Les équations régissant le problème sont séparées en deux sous-problèmes, dont l'espace des solutions définissent deux variétés A_d et Γ. Dans le cas d'étude, l'espace A_d contient l'ensemble des solutions aux équations d'équilibre, d'admissibilité, et aux équations d'état du matériau. Ces équations linéaires sont globales en temps et en espace. A contrario, la variété Γ contient les équations non linéaires. On notera $A_{d,0}$ l'ensemble des solutions aux équations d'équilibre, d'admissibilité à zéro, et aux équations d'état du matériau.

La solution $\mathbf{s}$ est recherchée alternativement dans chacune des variétés grâce à l'utilisation de directions de recherche $\boldsymbol{\Theta}^+$ et $\boldsymbol{\Theta}^-$. Ces opérateurs permettent de coupler les deux sous-problèmes. L'écriture des grandeurs physiques mises en jeu se fait sous la forme de champs mécaniques primaux et des champs mécaniques duaux associés au problème. Dans cette section, à titre d'exemple, les champs pris en compte seront la dérivée de la déformation plastique $\dot{\boldsymbol{\varepsilon}}_p$ comme champ primal, et la contrainte $\boldsymbol{\sigma}$ comme champ dual. La solution est obtenue à la fin du schéma itératif et est dans $A_d \cap \Gamma$.

La méthode s'applique également à problèmes multi-échelles [Ladevèze *et al.*, 2002], pour lesquels les problèmes micro et macro sont résolus alternativement ; décomposition de domaine [Champaney *et al.*, 1997], pour laquelle la séparation opère sur les sous-domaines d'une part et les interfaces d'autre part ; ou encore en multi-physique. Cette stratégie a également été appliquée à d'autres types d'applications comme l'endommagement de composites [Allix et Ladevèze, 1992], la dynamique rapide [Lemoussu *et al.*, 2002] ou des problèmes avec assemblages [Champaney, 1996] par exemple.

L'algorithme LATIN se décompose en une étape d'initialisation et une boucle de convergence itérative avec l'étape locale et l'étape globale :

Initialisation : l'algorithme est initialisé par une solution $\mathbf{s}_0 \in A_d$. Classiquement, la solution élastique associée au problème convient. D'autres méthodes exploitant les informations issues des précédents calculs existent et seront présentées en section 6.1.

Étape locale : l'étape locale consiste à trouver $\hat{\mathbf{s}} \in \Gamma$ sachant $\mathbf{s} \in A_d$ en exploitant la direction de recherche $\boldsymbol{\Theta}^+$:

$$\boldsymbol{\Theta}^+(\hat{\boldsymbol{\sigma}} - \boldsymbol{\sigma}) + (\hat{\dot{\boldsymbol{\varepsilon}}}_p - \dot{\boldsymbol{\varepsilon}}_p) = 0 \tag{1.19}$$

La direction de montée $\boldsymbol{\Theta}^+ = +\infty$ est la plus employée. Elle revient à exploiter notamment la déformation calculée à l'étape globale pour intégrer le comportement par un algorithme de retour radial. La solution du sous-problème est donc recherchée à chaque point de Gauss de chaque élément de la structure de manière indépendante, en résolvant un problème non linéaire en chaque pas de temps.

Étape globale : l'étape globale consiste à rechercher la solution $\mathbf{s} \in A_d$ sachant $\hat{\mathbf{s}} \in \Gamma$ en exploitant la direction de recherche $\boldsymbol{\Theta}^-$:

$$\boldsymbol{\Theta}^-(\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}}) - (\dot{\boldsymbol{\varepsilon}}_p - \hat{\dot{\boldsymbol{\varepsilon}}}_p) = 0 \tag{1.20}$$

Le sous-problème est global en espace et en temps, mais linéaire. Ce problème est équivalent à rechercher une correction $\delta\mathbf{s} \in A_{d,0}$ de la solution $\tilde{\mathbf{s}} \in A_d$ calculée à l'itération précédente. L'équation (1.20) devient alors

$$\boldsymbol{\Theta}^-((\tilde{\boldsymbol{\sigma}} + \delta\boldsymbol{\sigma}) - \hat{\boldsymbol{\sigma}}) - (\tilde{\dot{\boldsymbol{\varepsilon}}}_p + \delta\dot{\boldsymbol{\varepsilon}}_p) - \hat{\dot{\boldsymbol{\varepsilon}}}_p) = 0 \tag{1.21}$$

À la manière de l'approche de type Newton, la direction de recherche globale $\boldsymbol{\Theta}^-$ peut être choisie à partir d'une linéarisation du comportement obtenu à l'étape locale.

Critère d'arrêt : le critère d'arrêt est basé sur l'écart entre les grandeurs $\mathbf{s}_{i+1}$ et $\hat{\mathbf{s}}_{i+1/2}$ tel que présenté à l'équation (1.22). Ce critère d'arrêt exploite la norme matricielle $||\bullet||_{\mathbf{C}}$ construite à partir de la déformation $\boldsymbol{\varepsilon}$ et de la contrainte $\boldsymbol{\sigma}$. L'opérateur de Hooke élémentaire $\mathbf{C}$ est un des opérateurs usuellement employés pour cette norme.

$$\nu_{i+1} = \frac{||\mathbf{s}_{i+1} - \hat{\mathbf{s}}_{i+1/2}||_{\mathbf{C}}}{\frac{1}{2}||\mathbf{s}_{i+1} + \hat{\mathbf{s}}_{i+1/2}||_{\mathbf{C}}}, \quad ||\mathbf{s}||_{\mathbf{C}} = \int_{\Omega\times I} \frac{1}{2}\left(\boldsymbol{\varepsilon} : \mathbf{C}\,\boldsymbol{\varepsilon} + \boldsymbol{\sigma} : \mathbf{C}^{-1}\boldsymbol{\sigma}\right) \mathrm{d}\Omega\mathrm{d}t \tag{1.22}$$

Algorithm 2: Méthode LATIN

- **Initialisation** : Approximation initiale $\mathbf{s}_0 \in A_d$
- **Boucle de convergence - Tant que** $||\nu|| > tol$
  - **Etape Locale • Boucle sur les pas de temps** i_t
    - **Pour chaque element** e
      - **Pour chaque point de Gauss** g
        - Intégration de la loi de comportement
        - Linéarisation du comportement local
  - **Etape Globale • Boucle sur les pas de temps** i_t
    - Assemblage de l'opérateur tangent
    - Calcul de la correction $\delta\underline{\mathrm{U}}_{i_t}$ par résolution du système linéaire associé
  - **Calcul de l'erreur • Boucle sur les pas de temps** i_t
    - **Pour chaque element** e
      - **Pour chaque point de Gauss** g
        - Calcul de l'erreur LATIN ν

Une vision simple (voire simpliste) est de considérer la méthode LATIN comme une méthode de Newton-Raphson où l'ordre entre la boucle temporelle, et la boucle de convergence a été inversé. L'apport de ce changement est triple :

Avantage 1 : Il est possible de simplifier la résolution à l'étape globale en recherchant la correction $\delta\mathbf{s}$ sous la forme de champs à variables séparés. Le coût de la résolution du problème global en est réduit et l'ensemble de champs $\mathbf{s}$ obtenus à chaque itération peut s'écrire sous forme PGD. Ceci se fait au prix d'un relâchement sur la direction de recherche $\boldsymbol{\Theta}^-$ qui n'est plus vérifiée exactement. L'écart vis-à-vis de celle-ci est alors minimisé.

Avantage 2 : Dans le cadre de la méthode Newton-Raphson, l'interruption volontaire ou involontaire de la résolution en cours de calcul permet d'obtenir la solution convergée jusqu'au pas de temps i_t. Dans le cas de la méthode LATIN, une approximation de la solution spatio-temporelle complète est disponible à chaque itération.

Avantage 3 : L'initialisation de la méthode de Newton-Raphson par une solution approchée spatio-temporelle est complexe. Par sa structure, la méthode LATIN permet naturellement l'initialisation du calcul par une approximation de la solution spatio-temporelle précalculée.

Ces trois avantages font de la méthode la candidate idéale pour la résolution d'un problème multi-paramétrique. L'avantage 1 permet de générer naturellement une base de modes associée aux champs solutions pour réduire le temps de calcul par itération. Ceci aboutit au cadre LATIN-PGD présenté au chapitre 5. L'avantage 2 permet de générer des champs approximés ou exacts, permettant de calculer une QoI approximée ou exacte très simplement et d'accélérer la génération d'un métamodèle de la QoI par l'emploi de données multi-fidélités. L'avantage 3 permet d'exploiter les configurations déjà calculées et de mettre en place une interpolation issue des champs mécaniques pour initialiser le calcul suivant, et donc d'accélérer le calcul comme le présente le chapitre 6.

Chapitre 2

Stratégie pour l'approximation & l'optimisation globale d'une quantité d'intérêt

Sommaire

2.1 Présentation de la stratégie 24
2.2 Stratégies d'optimisation de structures 26
2.3 Méthodes d'échantillonnage initial dans $\mathcal{D}$ 28
2.4 Méthodes de métamodélisation 29
2.4.1 Régression linéaire 30
2.4.2 Extensions à la régression linéaire 32
2.4.3 Un exemple d'extension : la régression SVM 33
2.5 Régression par processus gaussiens (Krigeage) 33
2.5.1 Définition 33
2.5.2 Fonctions de covariance 35
2.5.3 Détermination des hyperparamètres du métamodèle 36
2.5.4 Optimisation des hyperparamètres 37
2.6 Critères de validation du métamodèle 39

2.1 Présentation de la stratégie

Dans le cadre d'une conception mécanique optimisée et robuste, il est nécessaire de pouvoir tester différentes configurations du produit vis-à-vis d'une ou plusieurs Quantités d'Intérêt (Quantity of Interest — QoI) y. Le but est de vérifier rapidement le comportement de la structure, ou de l'optimiser.

Dans le cadre de ces travaux, les configurations sont régies par d variables quantitatives sur une plage continue. L'espace de variation définissant l'ensemble des configurations $\underline{x}$ choisies est nommé espace de conception $\mathcal{D}$. Il s'agit de l'ensemble des éléments admissibles du problème, pouvant nécessiter la vérification de contraintes de conception.

La QoI sera une fonction de $\mathcal{D}$ à valeurs dans $\mathbb{R}$. Dans le cas d'étude, elle proviendra du post-traitement de la solution d'un problème mécanique paramétrée par $\underline{x}$. Cette solution peut s'obtenir de plusieurs manières : de manière exacte par l'appel à un solveur mécanique, par approximation modale, par emploi de fonctions analytiques approximantes...

Deux problèmes d'estimation de la QoI seront abordés dans ce manuscrit :

— Un problème de prédiction (ou régression) où une approximation de la QoI y sur l'espace de conception complet $\mathcal{D}$ sera générée. Cette approximation peut ensuite être utilisée à la manière d'un abaque (virtuel) par les bureaux d'études.

— Un problème d'optimisation paramétrique global avec la recherche de l'optimum global $\boldsymbol{X}^* = \text{argmin}_{\boldsymbol{X}\in\mathcal{D}}\, y(\boldsymbol{X})$ sur l'espace de conception $\mathcal{D}$.

Il existe de nombreuses méthodes d'échantillonnage de $\mathcal{D}$ pour générer une bonne approximation de la QoI. Ces méthodes sont présentées dans [Fang *et al.*, 2005]. De même, il existe de nombreuses familles de méthodes d'optimisation globales, notamment présentées dans [Nocedal et Wright, 2006]. Citons les méthodes exploratoires ou encore les méthodes heuristiques ou métaheuristiques biomimétiques comportant une composante aléatoire et une composante d'exploitation des données précédentes pour minimiser le nombre d'observations nécessaires.

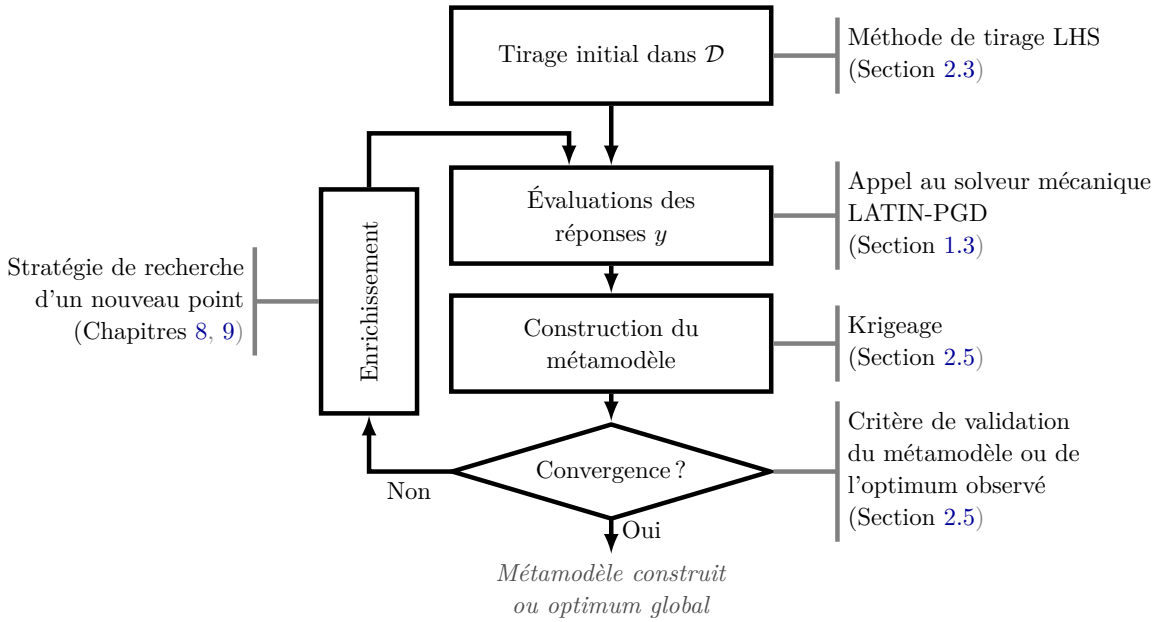


FIGURE 2.1 – Représentation schématique de la stratégie d'apprentissage

La résolution de ces problèmes se faisant souvent avec un budget calcul limité, il est nécessaire de réduire le nombre d'appels au solveur mécanique et donc le nombre d'observations. Ainsi, il est important d'exploiter au mieux l'information déjà obtenue. Pour se faire, la stratégie d'apprentissage se base sur un premier tirage initial à discrépance faible dans l'espace de conception $\mathcal{D}$, puis ajoute séquentiellement de nouveaux points qui complètent l'ensemble initial. La sélection des nouveaux points est basée sur la maximisation de critères d'enrichissement quantifiant le gain d'information estimé sur le métamodèle, ou sur l'optimum global.

2.2 Stratégies d'optimisation de structures

L'une des applications voulue de la stratégie d'enrichissement du métamodèle concerne l'optimisation globale de la fonction objectif. L'optimisation peut porter sur le choix d'un ensemble de paramètres $\underline{x}^*$ dans un espace paramétrique complet $\mathcal{D}$ dans le cas de l'optimisation paramétrique. Elle peut également avoir pour but le perfectionnement de la forme d'un objet étudié parmi un ensemble de morphologies possibles dans le cadre de l'optimisation de forme. Celle-ci peut être topologique, géométrique ou paramétrique [Allaire, 2007].

L'optimisation globale recouvre deux problématiques que sont la recherche d'un minimum global, et la recherche d'un maximum global. La formulation d'un problème de minimisation globale consiste à répondre au problème (2.1) :

$$\text{Trouver } (\underline{x}^*, \mathrm{y}^*) \in \mathcal{D} \times \mathbb{R}, \left| \begin{array}{l} \text{solution de } \min_{\underline{x} \in \mathcal{D}} y(\underline{x}) \\ \text{respectant les contraintes } C(\underline{x}) \end{array} \right. \tag{2.1}$$

Les contraintes C sont représentables par une fonction booléenne d'acceptation. Il s'agit d'un ensemble de cas d'égalités $g(\underline{x}) = 0$ ou d'inégalités $h(\underline{x}) \leq 0$ à vérifier. Ces cas peuvent être linéaires ou non-linéaires ce qui complexifie la résolution du problème d'optimisation.

Il existe également de nombreuses méthodes d'optimisation locales dans la mesure où la recherche de minima locaux exploite au maximum les informations relatives à l'évolution de la fonction objectif. À la différence de l'optimisation globale, l'optimisation locale consiste à trouver un voisinage de $\underline{x}^*$ dans $\mathcal{D}$ noté $\mathrm{V}(\underline{x}^*)$ dans lequel $\underline{x}^*$ est un optimum :

$$\text{Trouver } (\underline{x}^*, \mathrm{y}^*) \in \mathcal{D} \times \mathbb{R}, \left| \begin{array}{l} \text{solution de } \min_{\underline{x} \in \mathrm{V}(\underline{x}^*)} y(\underline{x}) \\ \text{respectant les contraintes } C(\underline{x}) \end{array} \right. \tag{2.2}$$

Parmi les méthodes d'optimisation locale sans contraintes, on peut distinguer :

- les méthodes de recherche directe (*direct search* [Wright, 1996, Lewis *et al.*, 2000]) qui utilisent uniquement les réponses de la fonction pour localiser le minimum et non le gradient exact ou estimé de la fonction objectif.
- les méthodes à directions de descente exploitant les gradients de la fonction aux points étudiés. L'algorithme tente de converger itérativement vers le minimum local selon une direction de descente et une amplitude calculées par l'algorithme à partir de ces informations. Citons parmi ces méthodes les méthodes du gradient conjugué (stochastique [Robbins et Monro, 1951] ou non).
- les méthodes de Newton recherchant les points critiques de la fonction objectif à partir de la hessienne de la fonction ou d'une approximation de celle-ci (méthodes quasi-Newton DFP [Davidon, 1959, Fletcher et Powell, 1963] ou encore BFGS [Broyden, 1970, Fletcher, 1970, Goldfarb, 1970, Shanno, 1970])

Il existe de multiples méthodes adaptées à des problèmes d'optimisation sous des contraintes bien spécifiques. La programmation linéaire, quadratique ou encore les méthodes de pénalisation et les méthodes lagrangiennes notamment employées en calcul de structures font partie de ces méthodes.

Ces méthodes d'optimisation locales sont à la base d'un certain nombre de méthodes d'optimisation globale. Il existe des méthodes stochastiques exploitant une part de recherche exploratoire, et une part de recherche dans la zone de la valeur observée optimale comme les méthodes évolutionnaires avec l'algorithme génétique (*genetic algorithm* — GA) [Holland, 1975, Goldberg, 1989]) ou l'optimisation par essaims (*particle swarm optimisation* — *PSO* [Kennedy et Eberhart, 1995]). L'ensemble de ces méthodes d'optimisation globale sont classifiées dans la figure 2.2.

La méthode d'optimisation exploitée dans le cadre de ces travaux est une méthode stochastique. Il s'agit de l'optimisation bayésienne initiée par Kushner [Kushner, 1964] et exploité par Jones [Jones *et al.*, 1998] avec l'algorithme EGO. Le principe consiste à quantifier la probabilité espérée d'amélioration de l'optimum global $y_{\min}$ observé dans $\mathcal{D}$ appellé *Expected Improvement* (EI) dont il existe des variantes. Cette méthode sera présentée au chapitre 9.

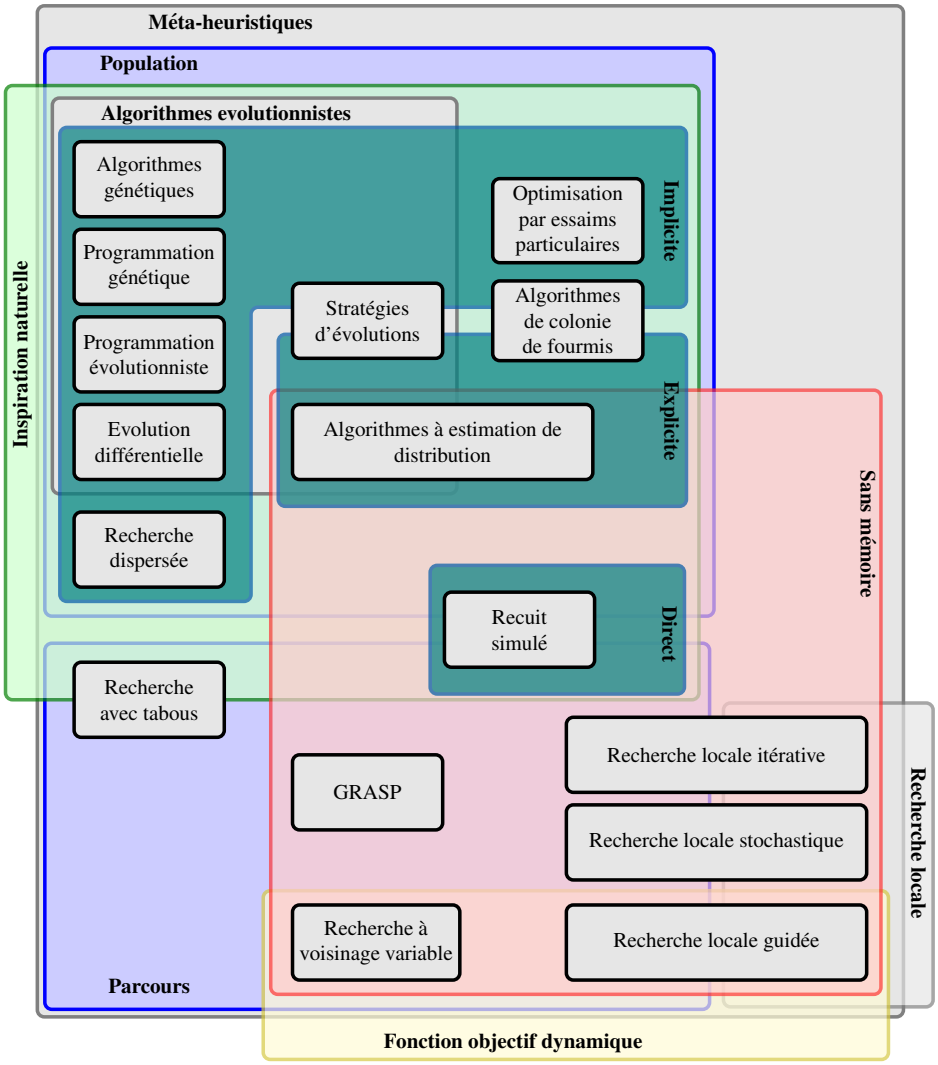


FIGURE 2.2 – Classification des algorithmes méta-heuristiques en fonction des caractéristiques de fonctionnement [Dréo, 2004]

2.3 Méthodes d'échantillonnage initial dans $\mathcal{D}$

Le but de l'échantillonnage initial est de choisir dans l'espace de conception $\mathcal{D}$, un minimum de points de manière judicieuse pour que le premier métamodèle généré possède l'information la plus dense possible. Pour cela, sans information *a priori* sur la fonction objectif, le but est de remplir au mieux l'espace $\mathcal{D}$. Pour se faire, trois classes de méthode de tirages existent : les tirages déterministes, les tirages aléatoires et les tirages quasi-aléatoires.

Tirages déterministes

Dans cette classe d'algorithmes, le choix des jeux de paramètres est réalisé à partir d'un schéma prédéfini. On peut citer comme algorithmes issus de cette famille les plans factoriels complets et fractionnaires [Collombier, 1995], les plans bayésiens exploitant le critère maximum d'entropie [Lindley, 1956], et les critères géométriques *minimax* et *maximin*.

Les plans factoriels correspondent à l'algorithme le plus simple à implémenter. L'idée est de choisir dans $\mathcal{D}$ une discrétisation de chaque paramètre et de tester dans le cas complet toutes les possibilités ou certaines en considérant uniquement les facteurs de base dans le cas des plans fractionnels.

Les plans bayésiens ont eux pour but de maximiser le critère d'entropie de Lindley [Lindley, 1956] sur un tirage aléatoire initial. Ils sont coûteux en haute dimension et ont tendance à choisir les jeux de paramètres au bord de l'espace $\mathcal{D}$ [Ancelet *et al.*, 2012].

Enfin, les méthodes basées sur des critères géométriques déterminent les points initiaux sur $\mathcal{D}$ par le recouvrement de l'espace $\mathcal{D}$ par les plus petites boucles centrées sur les points de l'espace (critère *minimax*) ou par la maximisation du rayon de boules non intersécantes centrées sur les points de l'espace (critère *maximin*) [Pronzato et Müller, 2012].

L'ensemble de ces méthodes présentent l'inconvénient de ne pas maximiser la distance entre les points du tirage. Ce critère appelé discrépance δ permet de quantifier la qualité d'un tirage et est basé sur la distance maximale entre les points voisins, usuellement au sens de la norme canonique.

$$\delta = \max_{i \in [\![1,n]\!]} \min_{j \neq i} ||\underline{x}_i - \underline{x}_j|| \tag{2.3}$$

Une idée étant d'exploiter des tirages aléatoires ou quasi-aléatoires maximisant ce critère.

Tirages aléatoires

Cette classe d'algorithmes réalise ses tirages de manière aléatoire. On peut citer le type de tirage le plus connu et le plus simple d'accès, le tirage Monte-Carlo [Metropolis et Ulam, 1949]. Cette méthode consiste à réaliser un tirage uniforme dans l'hypercube unité de même dimension que $\mathcal{D}$ et de réaliser la fonction transformant l'hypercube en $\mathcal{D}$.

Le problème de cette méthode provient du faible taux de convergence de la distribution uniforme en $1/\sqrt{n}$. Il existe plusieurs méthodes pour améliorer le concept s'appuyant sur des méthodes de réduction de variance comme le tirage d'importance (utilisation d'une densité biaisée), la variable de contrôle ou la stratification ; ou l'optimisation de plans d'expériences.

Ici, on se contentera de l'étude de l'échantillonnage stratifié avec le tirage LHS. Le tirage LHS se base sur un partitionnement uniforme sans recouvrement de $\mathcal{D}$. Ainsi, on force le tirage à respecter les proportions dans chaque découpe. Au niveau du choix des allocations de tirages dans chaque zone, une stratification proportionnelle est possible en considérant que pour chaque zone, un nombre de tirages proportionnel à sa taille est alloué. On a ainsi diminution de la variance et donc à variance fixé, diminution du nombre de points. Néanmoins, le taux de convergence reste en $O\left(1/\sqrt{n}\right)$.

Tirages quasi-aléatoires

Ce type de tirages améliore la méthode de Monte-Carlo en modifiant la génération des nombres aléatoires : c'est le principe de l'optimisation des plans d'expériences avec les suites à discrépance faible, c'est-à-dire des suites générant une bonne répartition uniforme des points dans l'espace. Les suites de Halton [Halton, 1964] et de Sobol [Sobol, 1967] en sont des exemples.

2.4 Méthodes de métamodélisation

Les métamodèles sont des modèles d'approximation générés à partir de données issus de calculs complexes ou de résultats d'expériences. Ils notamment utilisées en bureau d'étude pour prédire une quantité d'intérêt lorsque celle-ci n'est pas mesurable en un temps réduit. Ces méthodes étaient employées historiquement pour réaliser l'ajustement des courbes expérimentales (curve fitting), ou pour prédire des zones d'intérêt (minières par exemple [Krige, 1951]). Aujourd'hui, leur emploi permet l'automatisation du traitement de grandes quantités de données numériques. Il s'agit de générer un modèle liant une entrée $\underline{x}$ à la sortie $\mathrm{y} = y(\underline{x})$, à partir d'un ensemble de N couples entrée-sortie $\mathcal{T} = \{(\underline{x}_i, \mathrm{y}_i)\}_{i=1}^{N}$, appelé échantillon d'apprentissage ou training set. Chacun de ces couples est appelé observation.

Les données d'entrée $\underline{x}$ seront dans le cadre de ces travaux un vecteur de dimension d caractérisant les paramètres de conception. Les données de sortie seront des valeurs réelles : on parle de méthodes de régression par opposition aux méthodes de classification où y est une variable catégorielle ou nominale d'un ensemble fini (comme un homme ou une femme).

Deux phases caractérisent les métamodèles :

— La phase de génération (*fit*) : le métamodèle est généré à partir d'un ensemble d'observations sur n points sélectionnés $\underline{x}_i$, et des réponses correspondantes de la QoI $\mathrm{y}_i = y(\underline{x}_i)$. Des informations auxiliaires comme les gradients de la fonction aux points sélectionnés peuvent également être fournies.

— La phase de prédiction (*predict*) : une fois le métamodèle construit, il est alors utilisable et peut fournir à coût extrêmement faible une prédiction de la QoI $\hat{y}(\underline{x})$ et de ses gradients pour tout point $\underline{x}$ de l'espace de conception $\mathcal{D}$.

La phase de génération n'est pas définitive. Une procédure d'enrichissement est couramment mise en place pour enrichir le métamodèle à partir d'observations maximisant le gain d'information estimé.

De manière générale, l'élaboration d'un modèle de substitution nécessite un ensemble d'apprentissage $\mathcal{T}$, mais également un choix des fonctions supports employées Φ et des paramètres associés $\underline{\theta}$. Ces deux derniers éléments sont spécifiques au type de métamodèle et ne sont pas toujours nécessaires. Par ailleurs, en plus de la réponse approchée, il est possible d'estimer une forme d'erreur associée au métamodèle. La figure 2.3 présente schématiquement ces éléments.

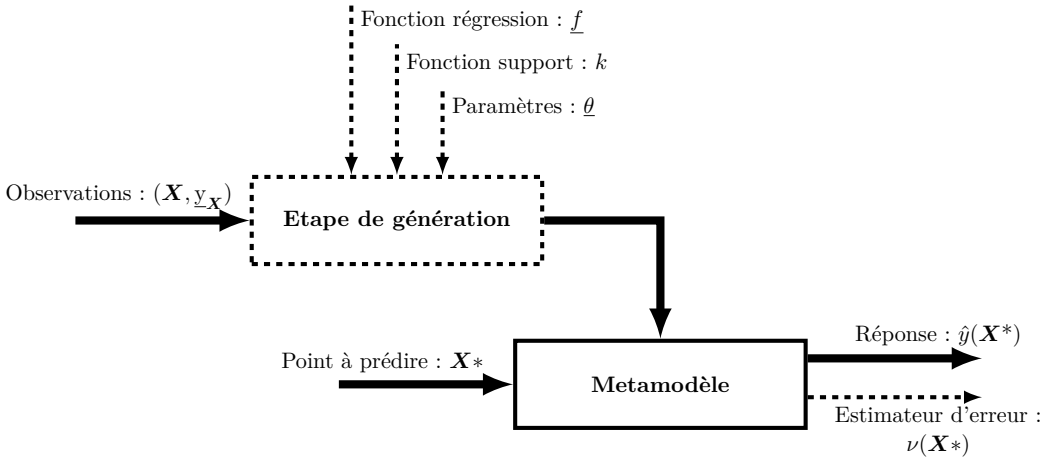


FIGURE 2.3 – Représentation schématique d'un modèle de substitution

Les choix des paramètres du métamodèle et des fonctions supports, lorsqu'il y en a, sont des procédures spécifiques à certains métamodèles et nécessitent l'emploi de techniques adaptées.

Pour fixer les notations, on définit $\boldsymbol{X}$ la matrice de conception contenant l'ensemble des points connus avec $\boldsymbol{X}_{ij} = (\underline{x}_i)_j$ et $\underline{\mathrm{y}}$ le vecteur de réponse avec $\underline{\mathrm{y}}_j = \mathrm{y}_j$.

2.4.1 Régression linéaire

2.4.1.1 Définition du modèle

Dans le cadre des méthodes de régression linéaire, la sortie $y(\underline{x})$ est modélisée par une combinaison linéaire de l'entrée notée $\hat{y}$:

$$\hat{y}(\underline{x}) = \underline{x}^T \underline{\beta} + Z(\underline{x}) \tag{2.4}$$

$$\forall i \in [\![1, \mathrm{N}]\!], \mathrm{y}_i = \underline{x}_i^T \underline{\beta} + Z_i \tag{2.5}$$

Le terme $\underline{x}^T\underline{\beta}$ représente le produit scalaire entre le vecteur d'entrée avec $\underline{\beta}$ le vecteur contenant les poids associés au modèle. Le terme Z est l'erreur résiduelle entre la prédiction linéaire et la vraie réponse.

Cette méthode est l'une des plus simples, mais également l'une des plus employées pour mettre en oeuvre une régression. Le terme Z_j représente l'erreur sur l'observation j. Dans le cas d'essais virtuels, on considère couramment que le solveur est exact avec $\forall i \in [\![1, \mathrm{N}]\!], Z_i = 0$.

Le cas simple consiste à ne pas modéliser l'erreur Z. Les poids $\underline{\beta}$ sont alors déterminés par minimisation de la somme des carrés des résidus d'observation (Sum Squared Error - SSE) (2.6). La moyenne de celle-ci est l'erreur aux moindres carrés (Mean Squared Error - MSE). Il s'agit simplement de minimiser les distances entre les observations et le modèle.

$$\underline{\hat{\beta}} = \underset{\underline{\beta}\in\mathbb{R}}{\operatorname{argmin}}\,\mathrm{SSE}(\underline{\beta}) = \underset{\underline{\beta}\in\mathbb{R}}{\operatorname{argmin}} \sum_{i=1}^{\mathrm{N}} \left(\mathrm{y}_i - \underline{x}_i^{\mathrm{T}}\underline{\beta}\right) \tag{2.6}$$

Ces méthodes de régression sont limitées dans la mesure où l'erreur n'est pas modélisée. Pour se faire, on va se placer dans le cadre probabiliste en définissant la fonction erreur Z comme une distribution normale centrée de variance fixée $\sigma^2(\underline{x}) = \sigma^2$. On peut alors réécrire le modèle sous une forme probabiliste (2.7).

$$p(y|\underline{x}, \underline{\theta}) = N\left(\underline{x}^T\beta, \sigma^2\right) \tag{2.7}$$

2.4.1.2 Définition des hyperparamètres à optimiser

Dans ce cas de figure, on a défini la forme de la distribution $p(y|\underline{x}, \underline{\theta})$ comme un processus gaussien et la distribution de $\underline{\beta}$ comme une distribution uniforme. les paramètres du modèle sont $\underline{\theta} = (\underline{\beta}, \sigma^2)$. Ces paramètres sont définis *a priori* comme appelées hyperparamètres. Ils permettent de définir le comportement du modèle généré, mais n'apparaissent pas dans la phase de prédiction lorsque le modèle est créé. La détermination des hyperparamètres est la difficulté majeure des métamodèles, et ce qui explique le coût de génération du métamodèle. En effet, leur choix est fréquemment réalisé par optimisation d'une fonctionnelle.

2.4.1.3 Choix de la fonctionnelle à optimiser

La formulation probabiliste permet d'extraire aisément les outils de réflexion sur les meilleurs hyperparamètres. Par exemple, une méthode courante pour déterminer les hyperparamètres d'un modèle statistique est d'optimiser le maximum de vraisemblance (Maximum likelihood estimation - MLE) ou le logarithme de celle-ci, défini par l'équation (2.8).

$$\underline{\hat{\theta}} = \underset{\underline{\theta}}{\operatorname{argmax}} \log p(\mathcal{T}|\underline{\theta}) = \underset{\underline{\theta}}{\operatorname{argmax}} \sum_{i=1}^{N} \log p(\mathrm{y}_i|\underline{x}_i, \underline{\theta}) \tag{2.8}$$

La fonctionnelle minimisée sera alors la fonction (2.9) :

$$\begin{aligned} l(\underline{\theta}) &= \sum_{i=1}^{N} \log \left[(2\pi\sigma^2)^{-\frac{1}{2}} \exp\left(-(2\sigma^2)^{-1} (y_i - \underline{\beta}^T \underline{x}_i)^2 \right) \right] \\ &= -\frac{N}{2} log(2\pi\sigma^2) - \frac{1}{2\sigma^2} SSE(\underline{\beta}) \end{aligned} \tag{2.9}$$

On retrouve la minimisation de l'erreur aux moindres carrées. La différentiation de la fonctionnelle permet d'en déduire les poids optimaux pour cette méthode (2.10) :

$$\underline{\hat{\beta}}_{LS} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \underline{\mathrm{y}} \tag{2.10}$$

2.4.2 Extensions à la régression linéaire

Il existe également des méthodes de régression linéaire dites « robustes » limitant le biais engendré par des valeurs fortement erronées comme la régression ridge [Hoerl et Kennard, 1970] ou lasso [Tibshirani, 1996]. Celles-ci sont des extensions de la régression linéaire et s'obtiennent simplement en modifiant la distribution associée à $\underline{\beta}$ (gaussienne pour ridge, laplace pour lasso). L'impact déterministe sur le modèle est l'ajout d'un terme de régularisation de Tikhonov (L2), d'un terme L1 ou les deux dans le cas de l'elastic net [Zou et Hastie, 2017]. Ces méthodes peuvent être couplées à l'algorithme LARS [Efron *et al.*, 2004] pour les problèmes en grande dimension.

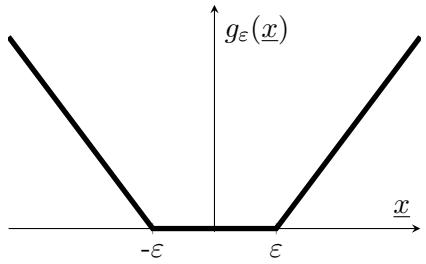
Néanmoins, la régression linéaire reste en l'état très restrictive. Une idée très simple pour générer des modèles non linéaires est de projeter d'abord l'entrée $\underline{x}$ en utilisant un ensemble de fonctions de base $\Phi(\underline{x})$. Un exemple simple dans le cas d'une entrée scalaire est l'utilisation d'une base polynomiale $\Phi(x) = [1, x, x^2, \ldots, x^n]^T$ correspondant à la méthode de régression polynomiale [Box et Wilson, 1951, Myers, 1982]. L'utilisation d'une base quadratique $\Phi(\underline{x}) = [1, x_1, x_2, \ldots, x_d, x_1^2, x_1 x_2, \ldots, x_d^2]^T$ avec une fonctionnelle moindre carrée pondérée permet de retomber sur les moindres carrées mobiles [Lancaster et Salkauskas, 1981]. On pourra consulter [Nealen, 2004] pour une introduction aux méthodes moindres carrées, moindres carrées pondérées, et moindres carrées mobiles.

D'autres généralisations s'appuient sur la définition d'une fonction d'erreur particulière comme les Machines à Vecteur Support [Vapnik *et al.*, 1996, Gunn *et al.*, 1998], les réseaux de neurones [McCulloch et Pitts, 1943, Rosenblatt, 1958], ou la régression par processus gaussiens. Ces méthodes se distinguent par la méthode de détermination des fonctions de base à partir des données. Il existe d'ailleurs un lien fort entre l'ensemble des méthodes évoquées.

Enfin, citons comme méthode de régression les méthodes basées sur les arbres de décision et sur des ensembles de ceux-ci [Freund et Schapire, 1997, Breiman, 2001].

2.4.3 Un exemple d'extension : la régression SVM

La méthode des machines à vecteur de support (Support Vector Machines - SVM) a été initialement employée en classification [Vapnik, 1963, Duda *et al.*, 2000] avant d'être utilisée en régression [Vapnik *et al.*, 1996]. Le concept de base est la fonction d'erreur à marge ε (ε-Insensitive Loss Function - ILF) définie par la fonction g_ε.



$$g_\varepsilon(\underline{x}) = \begin{cases} |\underline{x}| - \varepsilon & \text{si} |\underline{x}| \geq \varepsilon \\ 0 & \text{sinon} \end{cases} \tag{2.11}$$

Le principe de cette régression repose sur la construction de marges ε définissant une zone dans laquelle les réponses de la fonction à approcher n'affecteront pas la réponse prédite par le modèle de régression à vecteur support.

La fonction $\underline{x} \mapsto e^{-g_\varepsilon(\underline{x})}$ peut s'interpréter comme un modèle de vraisemblance pour les résidus de régression (comme la fonction MSE). L'utilisation de la SVM donc est essentiellement motivée par la volonté de d'avoir un régresseur sparse, limitant le nombre d'observations pris en compte. On se place dans le cas de la généralisation d'une régression linéaire avec l'erreur $g_\varepsilon(\underline{x})$ et une distribution *a priori* gaussienne de $\underline{w} = \boldsymbol{X}^T \underline{\beta}$.

Le maximum de vraisemblance associé à $\underline{w}$ est obtenu en minimisant l'équation (2.12) selon $\underline{w}$. Cette équation est appelée également formulation primale [Vapnik *et al.*, 1996].

$$\frac{1}{2}||\underline{w}||^2 + C \sum_{i=1} g_\varepsilon(\underline{x})(y_i - \boldsymbol{X}^T \underline{\beta}) \tag{2.12}$$

Le métamodèle s'écrit alors $f(\underline{x}) = \sum_i \underline{\beta}_i \, \underline{x}_i^T \underline{x}$. Le problème peut aussi exploiter les fonctions noyaux pour générer un métamodèle sous la forme $f(\underline{x}) = \sum_i \underline{\beta}_i k(\underline{x}_i, \underline{x})$.

2.5 Régression par processus gaussiens (Krigeage)

2.5.1 Définition

Le krigeage [Krige, 1951, Matheron, 1963], également appelé régression par processus gaussiens [Rasmussen et Williams, 2006] peut être présenté comme une extension à la régression linéaire. L'erreur du modèle présente dans (2.4) est modélisée par un processus gaussien de moyenne nulle et de variance fonction de l'entrée :

$$Z(\underline{x}) \sim N\left(\underline{0}, \sigma^2 r\right) \tag{2.13}$$

L'interpolant s'écrit alors :

$$\hat{y}(\underline{x}) = \underline{f}(\underline{x})^{\mathrm{T}}\underline{\beta} + Z(\underline{x}) \tag{2.14}$$

Il existe trois types de krigeage

- le krigeage simple pour lequel la partie déterministe $\underline{f}(\underline{x})^{\mathrm{T}}\underline{\beta}$ est connue et constante.
- le krigeage ordinaire pour lequel la partie déterministe $\underline{f}(\underline{x})^{\mathrm{T}}\underline{\beta}$ s'écrit $\underline{\beta}$ avec $\underline{\beta}$ inconnu.
- le krigeage universel pour lequel la partie déterministe s'écrit $\underline{f}(\underline{x})^{\mathrm{T}}\underline{\beta}$ avec $\underline{\beta}$ inconnu.

Pour définir la corrélation entre les observations, on considère une fonction de covariance $k(\underline{x}, \underline{x}') = \sigma^2 r(\underline{x}, \underline{x}'; \underline{\theta})$ avec σ^2 la variance du processus et r une fonction de corrélation (ou fonction noyau) paramétrée par $\underline{\theta}$. Celle-ci est également choisie par l'utilisateur tout comme $\underline{f}$.

Le processus gaussien $\hat{y}$ est conditionné aux observations, ce qui signifie qu'en faisant l'hypothèse que les réponses y_i sont des variables aléatoires, on définit une distribution jointe entre la prédiction et les observations par l'équation (2.15) :

$$\begin{pmatrix} \hat{y}(\underline{x}) \\ \underline{y} \end{pmatrix} \sim N\left(\begin{pmatrix} \underline{f}(\underline{x})^{\mathrm{T}}\underline{\beta} \\ \mathbf{F}'\underline{\beta} \end{pmatrix}, \sigma^2 \begin{pmatrix} \mathbf{1} & \underline{r}^{\mathrm{T}}(\underline{x}) \\ \underline{r}(\underline{x}) & \mathbf{R} \end{pmatrix} \right) \tag{2.15}$$

Avec $\mathbf{F}$ matrice de régression telle que $\mathbf{F}_{ij} = \underline{f}_i(\underline{x}_j)$, $\mathbf{R}$ matrice de corrélation entre les données observées telle que $\mathbf{R} = r(\underline{x}_i, \underline{x}_j)$ et $\underline{r}(\underline{x})$ matrice de corrélation entre les données observées et les données prédites telle que $\underline{r}_j(\underline{x}) = r(\underline{x}_j, \underline{x})$.

On caractérise ainsi le métamodèle $\underline{x} \longrightarrow \left[\hat{y}(\underline{x}) | \hat{y}(\boldsymbol{X}) = \underline{y}, \underline{\beta}, \underline{\theta}\right]$ comme un processus gaussien $\hat{y}(\underline{x})$ conditionné aux observations $\hat{y}(\boldsymbol{X}) = \underline{y}$. Il est donc possible d'exprimer ce processus en fonction de sa moyenne $m(\underline{x})$ et sa covariance $Cov(\underline{x}, \underline{x}')$, définies par l'équation (2.16) :

$$\begin{cases} m(\underline{x}) & = \underline{f}(\underline{x})^{\mathrm{T}}\underline{\beta} + \underline{r}(\underline{x})^{\mathrm{T}}\mathbf{R}^{-1}(\underline{\mathrm{y}} - \mathbf{F}\underline{\beta}) \\ Cov(\underline{x}, \underline{x}') & = \sigma^2\left(1 - \underline{r}(\underline{x})^{\mathrm{T}}\mathbf{R}^{-1}\underline{r}(\underline{x}')\right) \end{cases} \tag{2.16}$$

La moyenne est l'interpolant utilisé pour la prédiction. Celui-ci peut s'écrire en notant $\underline{\tilde{\mathrm{y}}} = \underline{\mathrm{y}} - \mathbf{F}\underline{\beta})$:

$$Z(\underline{x}) = \sum_{i=1}^{n} \alpha_i \tilde{\mathrm{y}}_i \tag{2.17}$$

avec $\underline{\alpha} = \underline{r}(\underline{x})^{\mathrm{T}}\mathbf{R}^{-1}$. Cette équation traduit une autre approche de la régression par processus gaussiens qu'est la génération du meilleur prédicteur linéaire non biaisé (2.17) habituellement présenté en géostatistique.

Le grand intérêt du krigeage est l'exploitation possible de la covariance associée au processus gaussien pour la définition d'un intervalle de confiance statistique du métamodèle sur l'espace de conception. En effet, la variance notée $Var(\underline{x}) = Cov(\underline{x}, \underline{x})$ peut être obtenue sur tout l'espace de conception et représente par ailleurs l'erreur aux moindres carrés MSE. On peut ainsi calculer l'intervalle de confiance à 95% correspondant à $m(\underline{x}) \pm 3Var(\underline{x})$ comme présenté sur la figure 2.4.

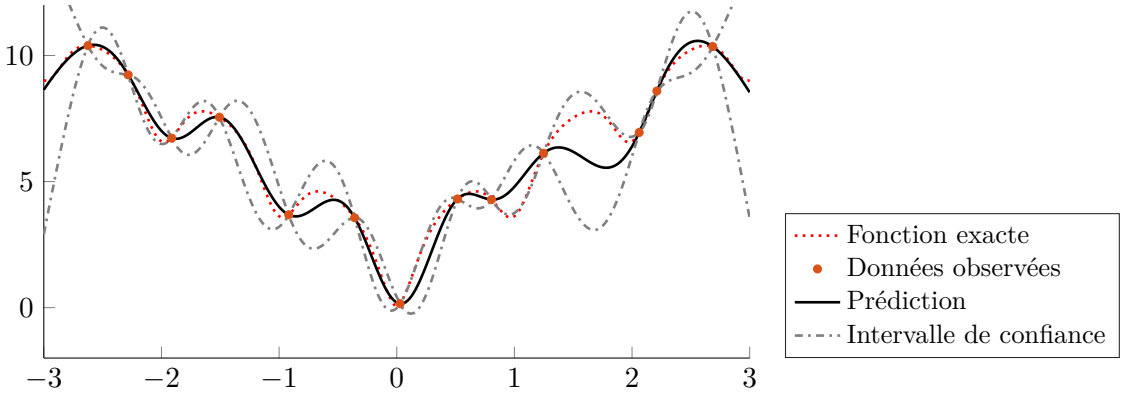


FIGURE 2.4 – Krigeage réalisé à partir d'observations de la fonction Ackley

2.5.2 Fonctions de covariance

La fonction de covariance $(\underline{x}, \underline{x}') \in \mathcal{D}^2 \mapsto k(\underline{x}, \underline{x}')$ permet le lien entre les observations et contrôle par cet aspect la régularité du processus Gaussien et donc de l'interpolant. Les fonctions pouvant être employées sont des fonctions positives semi-définies, *i.e.* $\forall N \in \mathbb{N}, \forall \underline{a} \in \mathbb{R}^N, \forall \underline{x} \in \mathcal{D}$:

$$\sum_{i,j=1}^{N} a_i a_j k(x_i, x_j) \geqslant 0, \qquad \sum_{i,j=1}^{N} a_i a_j k(x_i, x_j) = 0 \leftrightarrow \underline{a} = \underline{0} \tag{2.18}$$

On se place dans le cadre des fonctions de base radiale (Radial Basis Functions — RBF) où ces fonctions sont stationnaires et donc pilotées uniquement par l'écart ou la distance entre les deux variables d'entrée. Parmi ces fonctions, on peut citer la fonction noyau Matérn ν :

$$h \mapsto C_\nu(d) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\, \frac{h}{\theta} \right)^{\nu} K_\nu \left(\sqrt{2\nu}\, \frac{h}{\theta} \right) \tag{2.19}$$

avec Γ la fonction gamma et K_ν la fonction de Bessel modifiée d'ordre 2. Cette fonction est usuellement utilisée pour trois paramètres ν particuliers et le processus gaussien utilisant cette fonction est $\lceil \nu \rceil - 1$ fois différentiable.

— $\nu = \frac{3}{2}$: $h \mapsto \left(1 + \frac{\sqrt{3}h}{\theta}\right) \exp\left(-\frac{\sqrt{3}h}{\theta}\right)$

— $\nu = \frac{5}{2}$: $h \mapsto \left(1 + \frac{\sqrt{5}h}{\theta} + \frac{5h^2}{3\theta^2}\right) \exp\left(-\frac{\sqrt{5}h}{\theta}\right)$

— $\nu \to \infty$: $h \mapsto \exp\left(-\frac{d^2}{2\theta^2}\right)$ (Egalement appelée fonction gaussienne)

La figure 2.5 montre l'influence du choix de la fonction de covariance sur la génération du métamodèle. La non-prise en compte de la partie régressive ou la présence d'une régression constante ou linéaire a été testée tout comme les fonctions de corrélation exponentielle carrée (exp^2), ou de Matérn 3/2 ou 5/2 pour des résultats similaires. Les hyperparamètres ont un impact bien plus significatif sur l'erreur aux moindres carrées et justifient leur optimisation.

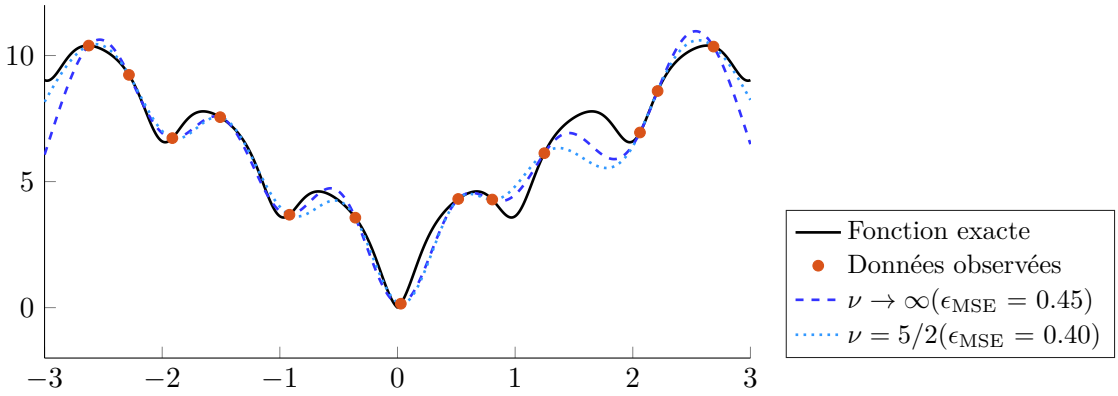


FIGURE 2.5 – Génération du métamodèle de la fonction Ackley en fonction des fonctions de régression et de corrélation employées (ϵ_{MSE} est l'erreur aux moindres carrées du métamodèle sur la solution exacte pour 1000 observations équiréparties)

2.5.3 Détermination des hyperparamètres du métamodèle

Les équations (2.16) correspondent au cas où l'ensemble des paramètres $\underline{\beta}$, σ^2 et $\underline{\theta}$ sont considérés comme étant connus. Toutefois, dans de nombreuses applications pratiques, il n'est pas toujours facile de spécifier tous les aspects de régression et de la fonction de covariance avec confiance.

Heureusement, plusieurs méthodes existent pour déterminer ces paramètres. La première méthode propose l'inférence bayésienne des paramètres à partir des observations [Robert, 2007]. Cette méthode permet de prendre en compte le jugement d'experts par un choix de la distribution *a priori*. La difficulté réside dans l'absence de forme analytique de la distribution *a posteriori*, nécessitant l'emploi de multiples intégrations numériques au moyen d'une méthode de Monte-Carlo avec chaîne de Markov par exemple. Le coût de calcul de cette méthode est très élevé. Dans le cadre de cette thèse, le temps de génération du métamodèle n'est pas un facteur limitant. Le choix optimal pour déterminer les hyperparamètres aurait donc pu être celui-ci, mais ne sera pas mis en œuvre pour des questions de complexité de mise en œuvre.

Une autre approche plus courante consiste à obtenir une forme analytique pour $\underline{\beta}$ en s'appuyant sur les propriétés de meilleur interpolant linéaire non biaisé, et optimiser σ^2 et $\underline{\theta}$ via une fonctionnelle caractérisant l'erreur du métamodèle comme la fonction de vraisemblance.

Pour déterminer le poids optimal $\hat{\underline{\beta}}$, le prédicteur est supposé :

- être une combinaison linéaire des observations : $\hat{y}(\underline{x}) = \underline{\beta}^{\mathrm{T}}(\underline{x})\underline{\mathrm{y}}$
- être non biaisé : $\mathbb{E}(\hat{y}(\underline{x}) - y(\underline{x})) = 0 \longrightarrow \underline{f}(\underline{x})^{\mathrm{T}}\underline{\beta}(\underline{x}) - y(\underline{x}) = \underline{0}$
- minimiser l'erreur aux moindres carrées MSE : $\sigma^2(1 + \underline{\beta}^{\mathrm{T}}\mathbf{R}\underline{\beta} - \underline{\beta}^{\mathrm{T}}\underline{r})$

Le modèle généré $\hat{y}$ est défini par la moyenne du processus m exprimé dans l'équation (2.20).

$$\begin{cases} m(\underline{x}) & = \underline{f}(\underline{x})^{\mathrm{T}}\hat{\underline{\beta}} + \underline{r}(\underline{x})^{\mathrm{T}}\mathbf{R}^{-1}(\underline{\mathrm{y}} - \mathbf{F}\,\hat{\underline{\beta}}) \\ Cov(\underline{x},\underline{x}') & = \sigma^2\left(1 - \begin{pmatrix}\underline{f}^{\mathrm{T}}(\underline{x})\\ \underline{r}^{\mathrm{T}}(\underline{x})\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}0 & \mathbf{F}^{\mathrm{T}}\\ \mathbf{F} & \mathbf{R}\end{pmatrix}^{-1}\begin{pmatrix}\underline{f}(\underline{x})\\ \underline{r}(\underline{x})\end{pmatrix}\right) \\ & = \sigma^2\left(1 - \underline{r}(\underline{x})^{\mathrm{T}}\mathbf{R}^{-1}\underline{r}(\underline{x}') + \underline{u}(\underline{x})^{\mathrm{T}}(\mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{F})^{-1}\underline{u}(\underline{x}')\right) \\ \hat{\underline{\beta}} & = (\mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\underline{\mathrm{y}} \\ \underline{u}(\underline{x}) & = \mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\underline{r}(\underline{x}) - \underline{f}(\underline{x}) \end{cases} \tag{2.20}$$

La variance associée au métamodèle est donc :

$$\mathrm{MSE} = \sigma^2\left(1 - \underline{r}(\underline{x})^{\mathrm{T}}\mathbf{R}^{-1}\underline{r}(\underline{x}) + \underline{u}(\underline{x})^{\mathrm{T}}(\mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{F})^{-1}\underline{u}(\underline{x})\right) \tag{2.21}$$

Les équations correspondent au cas où $\underline{\beta}$ n'est pas connu *a priori*. Elles s'obtiennent également à partir de (2.16) après intégration de la densité *a posteriori* de $\underline{\beta}$, en supposant $p(\underline{\beta}|\sigma^2) \sim 1$ et $p(\sigma^2) \sim \frac{1}{\sigma^2}$.

Il reste maintenant à déterminer les hyperparamètres σ^2 et $\underline{\theta}$ du métamodèle. Ceux-ci s'obtiennent par optimisation du modèle à générer en fonction d'une fonctionnelle caractérisant la qualité de celui-ci comme la fonction de vraisemblance ou les fonctions de validation croisée.

2.5.4 Optimisation des hyperparamètres

Fonction de vraisemblance

Le paramètre σ^2 peut s'obtenir également par intégration de sa densité *a posteriori*. Le processus sera alors une distribution *Student-t* plus complexe à manipuler, mais beaucoup moins sensible aux données aberrantes. Une autre possibilité pour optimiser σ^2 et $\underline{\theta}$ est de maximiser la fonction de vraisemblance associée au processus gaussien (2.22). En effet, celle-ci possède une forme analytique (ce que ne possède pas la distribution *Student-t*).

$$p(\underline{\mathrm{y}}|\underline{\beta},\sigma^2,\underline{\theta}) = \frac{1}{(2\pi\sigma^2)^{n/2}\sqrt{\det\mathbf{R}}}\exp\left(-\frac{1}{2}\frac{\tilde{\underline{\mathrm{y}}}^{\mathrm{T}}\mathbf{R}^{-1}\tilde{\underline{\mathrm{y}}}}{\sigma^2}\right) \tag{2.22}$$

avec n le nombre d'observations. La maximisation de celle-ci au regard de $\underline{\beta}$ conditionnellement à σ^2 et $\underline{\theta}$ permet de retrouver $\hat{\underline{\beta}}$.

$\underline{\beta}$ étant choisi, la maximisation de la fonction suivant σ^2 à $\underline{\theta}$ imposé donne :

$$\hat{\sigma} = \frac{\tilde{\underline{\mathrm{y}}}^{\mathrm{T}}\mathbf{R}^{-1}\tilde{\underline{\mathrm{y}}}}{n} \tag{2.23}$$

$\underline{\beta}$ et σ^2 étant choisis, le maximum estimé $\underline{\theta}$ pour la fonction de vraisemblance associée s'obtient en maximisant le logarithme de celle-ci :

$$\mathcal{L}(\underline{\theta};\underline{\mathrm{y}}) = -\frac{1}{2}\log\hat{\sigma}^2 - \frac{1}{2}\log\det\mathbf{R} - \frac{n}{2}\log 2\pi \tag{2.24}$$

Son gradient peut également s'écrire analytiquement :

$$\frac{\partial\mathcal{L}(\underline{\theta};\underline{\mathrm{y}})}{\partial\underline{\theta}_i} = \frac{1}{2}\underline{\tilde{\mathrm{y}}}^{\mathrm{T}}\mathbf{R}^{-1}\frac{\partial\mathbf{R}}{\partial\underline{\theta}_i}\mathbf{R}^{-1}\underline{\tilde{\mathrm{y}}} - \frac{1}{2}\mathrm{tr}\left(\mathbf{R}^{-1}\frac{\partial\mathbf{R}}{\partial\underline{\theta}_i}\right) \tag{2.25}$$

Ces deux fonctions sont peu coûteuses à calculer, la maximisation de $\mathcal{L}(\underline{\theta};\underline{\mathrm{y}})$ s'obtient numériquement par un processus d'optimisation globale, en exploitant (ou non) le gradient de la fonction. La figure 2.6 montre l'évolution de la fonction de vraisemblance dans un cas simple avec comme fonction noyau la fonction exponentielle carrée (Squared Exponential) $k(x,y) = \theta_1\exp\left(-\frac{|x-y|^2}{\theta_2}\right)$. Il s'agit d'une méthode de détermination des hyperparamètres rapide, mais moins performante que la validation croisée.

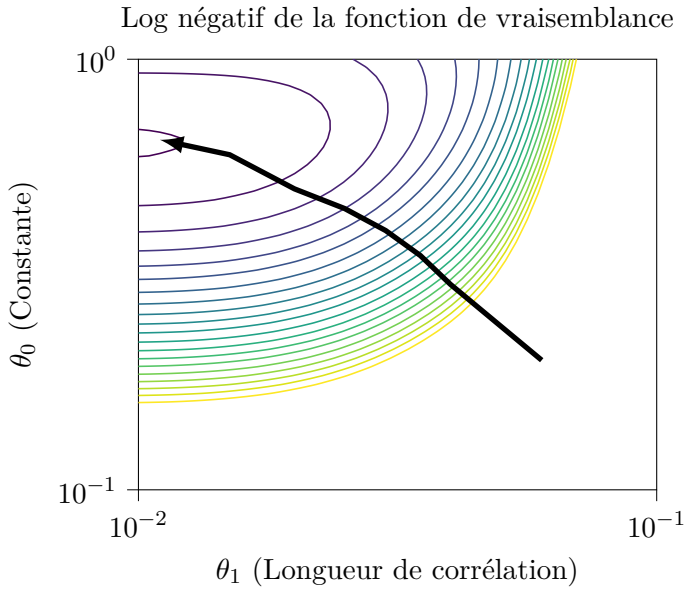


FIGURE 2.6 – Log négatif de la fonction de vraisemblance d'un krigeage (la flèche représente la recherche de la valeur optimale)

Validation croisée :

Il est également possible de tester $\underline{\theta}$ en partitionnant l'ensemble contenant les n observations $\boldsymbol{X}$ en un ensemble pour la construction du métamodèle, et un ensemble disjoint de validation $\boldsymbol{X}^*$ pour quantifier l'écart du métamodèle aux données tests. On parle de validation croisée.

Cette validation croisée peut être réalisée une fois (*holdout method*), k fois dans le cas où $\boldsymbol{X}$ est partitionné en k ensembles disjoints utilisés à tour de rôle comme ensembles de validation

(*k-fold method*), ou n fois dans le cas du *leave-one-out* (LOO). Cette dernière technique est la plus performante pour un faible nombre d'observations, on se place dans ce cadre dans la suite.

On considère donc n observations $(\underline{x}_i, \mathrm{y}_i)$ avec $i \in [\![1, n]\!]$. La probabilité associée à la prédiction de y_i connaissant les autres observations est la suivante :

$$\log p(\mathrm{y}_i | \boldsymbol{X}, \underline{\mathrm{y}}_{-i}, \underline{\theta}) = -\frac{1}{2}\log \sigma_i^2 - \frac{(\mathrm{y}_i - \mu_i)^2}{2\sigma_i^2} - \frac{1}{2}\log 2\pi \tag{2.26}$$

avec (μ_i, σ_i) la moyenne et la variance associée au métamodèle dans lequel $(\underline{x}_i, \mathrm{y}_i)$ n'est pas observé.

On définit alors la fonctionnelle comme étant la somme des opposés des probabilités prédictives (à une constante près) :

$$\mathcal{L}_{LOO}(\sigma^2, \underline{\theta}; \underline{\mathrm{y}}) = \sum_{i=1}^{n} \log \sigma_i^2 + \frac{(\mathrm{y}_i - \mu_i)^2}{\sigma_i^2} \tag{2.27}$$

Cette méthode est plus coûteuse, car elle nécessite la génération de n métamodèles. Une astuce technique permet néanmoins de rendre cette méthode équivalente en temps de calcul à la recherche de la fonction de vraisemblance [Rasmussen et Williams, 2006].

Méthode d'optimisation des hyperparamètres :

les deux fonctionnelles étant définies, il est nécessaire de les optimiser à l'aide d'une méthode d'optimisation globale. Pour réaliser celle-ci, l'ensemble des techniques présentées en section 2.2 conviennent. Dans la suite, la technique d'optimisation bayésienne [Hutter *et al.*, 2011] sera employée.

2.6 Critères de validation du métamodèle

Les fonctionnelles employées pour le choix des hyperparamètres peuvent également être utilisées pour valider le métamodèle en cas de méconnaissance d'une solution de référence. En cas d'existence d'une solution de référence, différents opérateurs de corrélation existent pour estimer la qualité de la prédiction.

Le coefficient de corrélation linéaire de Bravais-Pearson traduit une dépendance linéaire entre deux variables en un nombre de points échantillonnés n^*. Il fournit une indication sur la qualité des variations d'une réponse par rapport à l'autre. Ce critère ne reflète pas les différences d'amplitude d'une réponse par rapport à une autre ou encore d'un offset. Le coefficient de concordance r_{ccc} tient compte de ces aspects [Lin, 1989, Nickerson, 1997]. Ces deux coefficients

sont exprimés ci-dessous.

$$r^2 = \left(\frac{\sigma_{xy}}{\sigma_x \sigma_y} \right)^2 \qquad r_{ccc} = \frac{2.\sigma_{xy}}{\sigma_x^2 + \sigma_y^2 + (\bar{y} - \bar{\hat{y}})^2} \tag{2.28}$$

avec $\bar{y}$ et $\bar{\hat{y}}$ respectivement les moyennes des évaluations des réponses exactes et approchées aux n^* points échantillonnées et les termes $\sigma_{xy} = \frac{1}{n^*} \sum_{i=1}^{n^*} \left(y(\underline{x}_i) - \bar{y} \right) \left(\hat{y}(\underline{x}_i) - \bar{\hat{y}} \right)$,
$\sigma_x = \sqrt{\frac{1}{n^*} \sum_{i=1}^{n^*} \left(y(\underline{x}_i) - \bar{y} \right)^2}$, $\sigma_y = \sqrt{\frac{1}{n^*} \sum_{i=1}^{n^*} \left(\hat{y}(\underline{x}_i) - \bar{\hat{y}} \right)^2}$.

Plus ces coefficients sont proches de 1, plus le métamodèle est précis.

Chapitre 3

Modèles réduits pour les problèmes mécaniques

Sommaire

3.1 Approximation d'ordre réduit 41
3.2 Analyse de données *offline* et interpolation *online* POD 43
3.3 La méthode Reduced Basis 45
3.4 Approximation *online* PGD 45
3.5 Réduction de modèles pour des problèmes non linéaires 47

3.1 Approximation d'ordre réduit

La réduction de modèle (ROM) est une technique permettant de réduire la complexité de calcul des modèles mathématiques dans les simulations numériques. L'emploi de modèles réduits n'est pas nouveau : ils sont couramment utilisés en industrie. En Génie civil par exemple, les pré-dimensionnements sont obtenus à partir de modèles 0D ou 1D suffisamment précis pour analyser les problèmes structurels. Ils sont aussi massivement employés en dynamique des fluides ou en électromagnétisme pour simplifier les équations d'évolution.

Cette famille de méthodes peut également être vue comme des méthodes de métamodélisation multivariée, s'appliquant à des champs mécaniques complets et non plus à une quantité scalaire. Les méthodes de réduction de modèle ont pour visée une diminution de la complexité par la réduction de la dimension d'espace d'état ou des degrés de liberté associés au modèle. L'avantage de ce type d'approche est de ne pas modifier la description de la géométrie, des lois de comportement, des conditions initiales ou des conditions aux limites. Les grandes hypothèses physiques du modèle sont ainsi conservées.

Le problème étudié dans la suite est le problème d'évolution (3.1) traduisant les équations présentées en section 1.1. Ce problème est paramétré par le vecteur $\underline{\mu}$:

$$\text{Trouver } \underline{u} \in \mathcal{W} \text{ tel que } \forall \underline{v} \in \mathcal{W}_0,\ a(\underline{u}, \underline{v}; \underline{\mu}) = \ell(\underline{v}; \underline{\mu}) \tag{3.1}$$

avec $a : \mathcal{W} \times \mathcal{W} \mapsto \mathbb{R}$ est un opérateur bilinéaire elliptique continu et $\ell : \mathcal{W} \mapsto \mathbb{R}$ est un

opérateur linéaire continu. On définit également $\mathcal{I} = L^2(I, \mathbb{R})$ l'espace des fonctions à carré sommable dans I.

Une approximation de la solution $\underline{u}$ est définie en se plaçant dans l'approximation de $\mathcal{W}$, $\tilde{\mathcal{W}} = \mathcal{U} \otimes \mathcal{I}$.

Dans le cadre de ces travaux, la solution est recherchée dans le cadre d'une représentation à variables séparées, sous la forme d'une somme de produits de fonctions spatiales $\Lambda \in \mathcal{U}$, et de fonctions du temps $\lambda \in \mathcal{I}$ (3.2). Ces fonctions diffèrent pour chaque jeu de paramètres $\underline{\mu}$.

$$u(M,t) \approx \sum_{i=1}^{m} \Lambda_i(M)\lambda_i(t) \tag{3.2}$$

Cette approximation a été construite à partir de m paires de fonctions temps-espace et est ainsi considérée de rang m.

Approximation de rang m

On définit les produits scalaires associées aux espaces $\mathcal{U}$, $\mathcal{I}$ et $\mathcal{W}$ comme étant $\forall u, v \in \mathcal{W}$, $\langle u, v\rangle_{\mathcal{U}} = \int_\Omega uv \,\mathrm{d}\Omega$, $\langle u, v\rangle_{\mathcal{I}} = \int_I uv \,\mathrm{dt}$ et $\langle u, v\rangle_{\mathcal{W}} = \int_{I\times\Omega} uv \,\mathrm{d}\Omega\mathrm{dt}$. Les normes associées seront notées respectivement $||\bullet||_{\mathcal{U}}$, $||\bullet||_{\mathcal{I}}$ et $||\bullet||_{\mathcal{W}}$.

La construction de ces champs est fréquemment obtenue par approximation glouton en cherchant l'approximation de rang 1 minimisant l'erreur d'approximation de rang $m-1$ (3.3) caractérisée par $\delta u(M,t) = u(M,t) - \sum_{i=1}^{m-1} \Lambda_i(M)\lambda_i(t)$:

$$\varepsilon_m(M,t) = ||\delta u(M,t) - \Lambda(M)\lambda(t)||^2_{\mathcal{W}} \tag{3.3}$$

Il s'agit d'un problème de minimisation de l'erreur (3.3) selon $\Lambda \in \mathcal{U}$, $\lambda \in \mathcal{I}$. Dans la suite, on s'affranchira de la notation des variables associées aux fonctions dans la suite afin de clarifier les équations.

La solution du problème (3.3) notée δu vérifie l'équation (3.4) sous forme variationnelle :

$$\forall \lambda^* \in \mathcal{I}, \forall \Lambda^* \in \mathcal{U}, \int_{I\times\Omega} (\delta u - \Lambda\lambda)(\Lambda\lambda^* + \Lambda^*\lambda) \,\mathrm{d}\Omega\mathrm{dt} = 0 \tag{3.4}$$

Pouvant s'écrire également comme deux sous-problèmes couplés :

$$\begin{cases} \forall \lambda^* \in \mathcal{I}, \int_I \lambda^* \left(\int_\Omega (\delta u - \Lambda\lambda)\Lambda \,\mathrm{d}\Omega\right) \mathrm{dt} = 0 & \iff \lambda \int_{\mathcal{U}} \Lambda^2 d\Omega = \int_{\mathcal{U}} \delta u \Lambda d\Omega \\ \forall \Lambda^* \in \mathcal{U}, \int_\Omega \Lambda^* \left(\int_I (\delta u - \Lambda\lambda)\lambda \,\mathrm{dt}\right) \mathrm{d}\Omega = 0 & \iff \Lambda \int_I \lambda^2 \,\mathrm{dt} = \int_I \delta u \lambda \,\mathrm{dt} \end{cases} \tag{3.5}$$

Les fonctions sont définies et uniques à un facteur multiplicatif près. Usuellement, la fonction spatiale est normée.

Par substitution de Λ dans le sous-problème associé à λ, il est possible de montrer que le problème s'apparente à un problème aux valeurs propres associé à l'opérateur $\mathbf{T} : \lambda \longrightarrow \mathbf{T}(\lambda) = (\delta u, \langle \delta u,\, \lambda \rangle_{\mathcal{I}})_{\mathcal{U}}$ [Nouy, 2010].

La solution (Λ, λ) du problème de minimisation (3.3) vérifie donc les propriétés suivantes :

— La fonction temporelle λ minimise le quotient de Rayleigh :

$$\mathbf{R}(\lambda) = \frac{||\langle \delta u,\, \lambda \rangle_{\mathcal{I}}||^2_{\mathcal{U}}}{||\lambda||^2_{\mathcal{I}}} \tag{3.6}$$

— La fonction spatiale Λ est définie à partir de la fonction temporelle :

$$\Lambda = \frac{\langle \delta u,\, \lambda \rangle_{\mathcal{I}}}{||\lambda||^2_{\mathcal{I}}} \tag{3.7}$$

— L'erreur est aisément quantifiable à partir de la plus grande valeur propre ω associée au quotient de Rayleigh $\mathbf{R}(\lambda)$

$$\varepsilon_m = ||\delta u||_{\mathcal{W}} - \omega \tag{3.8}$$

Par cette dernière propriété (3.8) et les propriétés d'orthogonalité des modes, la méthode d'approximation glouton est équivalente à la recherche de l'approximation optimale de rang m. L'approximation s'écrit alors en fonction des m premiers modes propres de $\mathbf{T}$ notés λ_i, et des fonctions spatiales associées $\Lambda_i = \int_{\mathcal{I}} u\lambda_i dt$.

Cette analyse justifie l'emploi de méthodes de génération de bases réduites *a priori*, ou en temps réel pour le calcul. Dans la littérature, les techniques de réduction du modèle basées sur la projection diffèrent sur ce point. Une première famille de techniques implique une phase d'apprentissage, appelée étape hors ligne (*offline*). La seconde famille de méthodes cherche à construire la base d'approximation en même temps que la résolution du problème. On citera notamment ici la Proper Generalized Decomposition [Ladevèze, 1999, Chinesta *et al.*, 2014]. Ces deux familles de méthodes seront développées dans la suite et appliquées à la résolution de problèmes non-linéaires paramétriques.

3.2 Analyse de données *offline* et interpolation *online* POD

Les méthodes de génération *a posteriori* de bases réduites ont l'avantage de pouvoir exploiter les méthodes de résolution conventionnelle de problèmes mécaniques. La méthode est plus employée est la Proper Orthogonalized Décomposition (POD)

Il existe deux approches de celle-ci. La première appelée *Snapshot* consiste à considérer pendant la phase d'apprentissage une discrétisation grossière du domaine des paramètres. La seconde consiste à résoudre le modèle original sur l'ensemble de l'intervalle de paramètres, puis à utiliser le modèle réduit correspondant pour résoudre efficacement des problèmes similaires comportant de légères variations des paramètres matériels ou de conditions limites [Zimmermann, 2013].

Le point fort de ces techniques est que le nombre de modes permettant une bonne approximation est faible. Néanmoins, la pertinence des paramètres choisis pour construire la base est cruciale pour la qualité de l'approximation.

Cette méthode consiste à trouver la meilleure décomposition de rang-m $\underline{u}_m$ du champ connu $\underline{u}$ sous la forme réduite (3.2), au sens de la norme $\mathcal{L}^2$. Le problème de minimisation s'écrit sous la forme :

$$\min_{(\underline{\Lambda}_i,\lambda_i)_{i\in[|1,m|]}} ||\underline{U}(M,t;\underline{\mu}) - \sum_{i=1}^{m} \Lambda_i(M)\lambda_i(t;\underline{\mu})||^2_{\mathcal{W}} \tag{3.9}$$

sous la contrainte d'orthogonalité des fonctions spatiales : $(\underline{\Lambda}_i, \underline{\Lambda}_j) = \delta_{ij}$.

Ce problème peut également s'écrire comme la recherche d'une décomposition spectrale optimale :

$$\min_{(\Lambda_i)_{i\in[|1,m|]},(\Lambda_i,\Lambda_j)=\delta_{ij}} ||\underline{U}(M,t;\underline{\mu}) - \sum_{i=1}^{m} (\underline{U}(M,t;\underline{\mu}),\Lambda_i)\Lambda_i(M)||^2_{\mathcal{W}} \tag{3.10}$$

La décomposition est unique (à l'exception des permutations) et les modes extraits représentent les vecteurs associés aux plus grandes valeurs singulières de $\underline{U}$.

Dans le cadre d'un problème mécanique d'évolution spatio-temporel, il existe plusieurs méthodes de génération de l'approximation réduite. Une des méthodes consiste à considérer un nombre défini de snapshots de la solution. La concaténation de ces snapshots définit la matrice d'observation Ξ. Ces snapshots peuvent être issues de solutions extraites de données expérimentales, de solution obtenue pour des problèmes similaires à des instants spatiaux ou temporels donnés.

On se place dans un cadre discret où il existe trois méthodes équivalentes de construire la POD via la *Décomposition de Karhunen-Loeve* (KLD) [Karhunen, 1947, Loeve, 1948], l'*analyse en composantes principales* (PCA) [Pearson, 1901, Jolliffe, 2002] ou la *décomposition en valeurs singulières* (SVD) [Golub et Van Loan, 1983].

Pour la KLD et la PCA, la décomposition est trouvée en recherchant les valeurs propres et les vecteurs propres de la matrice de covariance $C = \Xi\Xi^T$. Les vecteurs propres de cette matrice sont les fonctions $\Lambda_i(M)$.

Dans le cadre de la SVD, la décomposition est obtenue sous la forme :

$$\Xi = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{3.11}$$

avec $\mathbf{U}$ et $\mathbf{V}$ des matrices unitaires et $\mathbf{S}$ la matrice diagonale comportant les valeurs singulières de Ξ. Par convention, les modes singuliers sont rangés matriciellement par valeur singulière décroissante. Dans le cadre d'une approximation de rang-m, les modes spatiaux $\underline{\Lambda}$ correspondent alors aux m premières colonnes de $\mathbf{U}$ et les modes temporels aux m premières colonnes de $\mathbf{V}\mathbf{S}^T$. La base ainsi construite peut être exploitée pour une projection de Galerkin du problème sur celle-ci.

3.3 La méthode Reduced Basis

Pour la méthode Galerkin-POD, la qualité de la solution finale est affectée par la pertinence des snapshots. En particulier, pour des problèmes qui présentent de fortes variations des conditions aux limites ou de fortes évolutions dynamiques, le nombre de snapshots nécessaires pour avoir une base adéquate peut augmenter au point que la technique en devient inadaptée [Boucinha *et al.*, 2013]. Pour ce type de cas, la méthode Reduced Basis est plus adéquate. Elle a été introduite au début des années 1980 pour l'analyse non linéaire des structures en dynamique dans un premier temps avant d'être étendue à différents types d'EDP paramétrées [Fink et Rheinboldt, 1983, Maday et Ronquist, 2004, Rozza *et al.*, 2007].

La technique repose sur un enrichissement progressif de la base réduite par les snapshots permettant de minimiser au fur et à mesure l'indicateur d'erreur. Il s'agit donc d'un algorithme glouton. La manière dont est définie cette erreur garantit la qualité de la solution trouvée sur le modèle réduit. Les deux méthodes présentées correspondent à une approche *offline-online* où la base d'approximation ainsi que l'ensemble des opérateurs sont construits dans une phase préalable *offline*, les temps de calcul sont alors très faibles durant la phase d'utilisation *online*.

3.4 Approximation *online* PGD

Il existe une famille de méthodes en plein essor pour la réduction de modèles : la Proper Generalized Decomposition. Initiée sous le nom d'approximation radiale [Ladevèze, 1985] dans le cadre du framework LATIN pour une décomposition espace/temps, elle est également employée pour des problèmes de rhéologie [Chinesta *et al.*, 2011], pour le contrôle de la mise en place automatisée de fibres composites [Joyot *et al.*, 2016], ou encore en corrélation d'images [Passieux et Périé, 2012]. La méthode a été généralisée avec une utilisation en multiparamétrique [Heyberger *et al.*, 2013] — voire en grande dimension [Paillet *et al.*, 2018] — avec emploi possible de méthodes d'hyper-réduction [Capaldo *et al.*, 2017].

La méthode est par ailleurs particulièrement adaptée pour des cas d'application comme l'optimisation [Schmidt *et al.*, 2011], la quantification d'incertitudes [Nouy, 2007] en *dynamic data-driven application systems* (DDDAS) [Ghnatios *et al.*, 2012] ou encore en haptique [Bordeu *et al.*, 2011].

L'objectif est de générer la meilleure approximation possible de la solution (3.1) du problème avec une représentation séparée des variables. Dans le cadre de la PGD, on n'impose plus la forme de la base comme dans la POD et la solution du problème est recherchée en même temps que la meilleure base pour sa représentation. L'approximation réalisée est enrichie jusqu'à ce qu'un critère d'erreur donné soit atteint.

La construction des modes est réalisée par un algorithme glouton. Ainsi, supposons qu'une

approximation d'ordre $m-1$ est définie :

$$\underline{\tilde{u}}_{m-1}(\underline{x},t) = \sum_{k=1}^{m-1} \lambda_k(t)\ \underline{\Lambda}_k(\underline{x}) \tag{3.12}$$

Usuellement, on améliore l'approximation de la solution de (3.4) par deux étapes. La première est une étape de correction des fonctions temporelles λ_k pour améliorer l'approximation de la solution sans régénérer de modes. Cette étape peu coûteuse permet de limiter le nombre de modes générés et le temps de calcul. Si besoin, une seconde étape correspondant à la génération d'un nouveau mode est exécutée. On cherche alors une nouvelle approximation $\underline{\tilde{u}}_m$ sous la forme de l'équation (3.13).

$$\underline{\tilde{u}}_m = \underline{\tilde{u}}_{m-1} + \delta u = \underline{\tilde{u}}_{m-1} + \underline{\Lambda}\lambda \tag{3.13}$$

De multiples méthodes existent pour l'obtention de l'approximation. La première présentée est basée sur l'approximation de Galerkin, et la seconde sur la minimisation du résidu $\mathcal{R}$. Ces deux méthodes s'appuient sur deux problèmes couplés à la manière d'un algorithme de type Gauss-Seidel. Le détail des méthodes est visible dans [Nouy, 2010].

Approximation de Galerkin

La définition classique de la Proper Generalized Decomposition consiste à définir la fonction test suivante :

$$v(M,t) = \Lambda^*(M)\lambda(t) + \Lambda(M)\lambda^*(t) \tag{3.14}$$

La recherche de la nouvelle paire $(\underline{\Lambda},\lambda)$ consiste à écrire l'orthogonalité du résidu.

$$\text{Trouver } (\underline{\Lambda},\lambda) \in \mathcal{U}\times\mathcal{I} \,/\, \forall(\underline{\Lambda}^*,\lambda^*) \in \mathcal{U}\times\mathcal{I}, \tag{3.15}$$

$$a(\underline{\tilde{u}}_{m-1} + \underline{\Lambda}\lambda, \underline{\Lambda}^*\lambda + \underline{\Lambda}\lambda^*) = \ell(\underline{\Lambda}^*\lambda + \underline{\Lambda}\lambda^*) \tag{3.16}$$

L'équation peut alors se traduire par la résolution simultanée de deux sous-problèmes :

$$\text{Trouver } (\underline{\Lambda},\lambda) \in \mathcal{U}\times\mathcal{I} \,/ \tag{3.17}$$

$$\forall(\underline{\Lambda}^*) \in \mathcal{U},\ a(\underline{\Lambda}\lambda, \underline{\Lambda}^*\lambda) = \ell(\underline{\Lambda}^*\lambda) - a(\underline{\tilde{u}}_{m-1}, \underline{\Lambda}^*\lambda) \tag{3.18}$$

$$\forall(\lambda^*) \in \mathcal{I},\ a(\underline{\Lambda}\lambda, \underline{\Lambda}\lambda^*) = \ell(\underline{\Lambda}\lambda^*) - a(\underline{\tilde{u}}_{m-1}, \underline{\Lambda}\lambda^*) \tag{3.19}$$

La résolution s'obtient avec un algorithme de point fixe. Cette méthode permet de générer à la volée des modes de la solution, en choisissant l'approximation du résultat réalisé par le nombre de modes générés m.

Minimisation du résidu

La seconde approche consiste à déterminer les meilleures fonctions spatiales et temporelles minimisant l'erreur en résidu. L'ajout d'un nouveau couple s'écrit comme la minimisation de

la norme du résidu,

$$(\lambda_m(t), \underline{\Lambda}_m) = \underset{(\lambda_m, \underline{\Lambda}_m)}{\operatorname{argmin}} \langle R(u_{m-1} + \lambda_m \underline{\Lambda}_m), R(u_{m-1} + \lambda_m \underline{\Lambda}_m) \rangle \tag{3.20}$$

De même que précédemment, la résolution s'obtient avec un algorithme de point fixe :

— connaissant la fonction du temps λ_m, la fonction d'espace est recherchée telle que :

$$\underline{\Lambda}_m = \underset{\underline{\Lambda}_m}{\operatorname{argmin}} \langle R(u_{m-1} + \lambda_m \underline{\Lambda}_m), R(u_{m-1} + \lambda_m \underline{\Lambda}_m) \rangle \tag{3.21}$$

— connaissant la fonction d'espace $\underline{\Lambda}_m$, la fonction du temps est recherchée telle que :

$$\lambda_m = \underset{\lambda_m}{\operatorname{argmin}} \langle R(u_{m-1} + \lambda_m \underline{\Lambda}_m), R(u_{m-1} + \lambda_m \underline{\Lambda}_m) \rangle \tag{3.22}$$

Il est également possible d'obtenir l'approximation par une méthode hybride en recherchant la fonction spatiale par approximation de Galerkin, et les autres fonctions par minimisation du résidu [Bhattacharyya *et al.*, 2018]. Cette méthode possède l'avantage de simplifier l'obtention de la fonction spatiale par rapport à la minimisation du résidu.

3.5 Réduction de modèles pour des problèmes non linéaires

Le problème non-linéaire (3.1) n'est pas directement résolvable. Les méthodes de résolution proposées en section 1.3 exploitent dans certains cas les méthodes de réduction de modèles pour permettre une résolution plus rapide. Les techniques les plus connues utilisées sont :

— La méthode Newton-Raphson, principalement couplée à la POD et à aux méthodes Reduced-Basis pour les stratégies basées sur une phase d'apprentissage. La méthode a également été associée à des techniques d'hyperréduction [Ryckelynck *et al.*, 2006].

— La méthode LATIN, bien adaptée à l'approximation PGD car il s'agit d'une méthode non incrémentale qui génère des approximations de la solution sur l'ensemble du domaine temps-paramètre-espace par des enrichissements successifs.

— La méthode asymptotique numérique (MAN) [Signorini, 1943, Descamps *et al.*, 1997], couplée à la PGD et aux techniques de construction d'une base réduite dans une phase d'apprentissage [Nguyen, 2012]. L'avantage de cette approche est la présence d'un seul opérateur tangent, identique pour chaque itération. l'inconvénient est la nécessité d'avoir un ordre élevé d'interpolation des termes non linéaires pour obtenir des résultats appropriés.

Dans le cadre de cette thèse, la méthode LATIN sera utilisée avec l'approximation PGD et sera présentée au chapitre 5.

Bilan

Le problème mécanique paramétré présenté en section [1.1] sera appliqué à la loi de comportement non-linéaire elasto-viscoplastique [1.2]. La résolution du problème non-linéaire est réalisée au moyen de la méthode LATIN de la section [1.3] présentant le triple avantage d'être une méthode permettant de générer aisément une approximation spatio-temporelle complète, de pouvoir l'améliorer après arrêt du solveur, et de pouvoir exploiter cette approximation pour le calcul d'un problème similaire.

Cette méthode a donc un grand intérêt pour la mise en œuvre d'une stratégie d'analyse paramétrique d'une structure [2.1] et même son optimisation [2.2]. Le tirage initial dans l'espace paramétrique est effectué au moyen d'un tirage LHS minimisant la discrépance [2.3], et les observations de la quantité d'intérêt serviront à construire un métamodèle [2.4], notamment au moyen de la régression par processus gaussien [2.5] permettant de générer un estimateur d'erreur du métamodèle [2.6] et de minimiser le nombre d'observations et donc le nombre de problèmes mécaniques à résoudre.

Néanmoins, la clé de ces travaux réside dans la possibilité de réduire également le temps de calcul de ces problèmes mécaniques au moyen de modèles réduits [3.1]. Parmi ces méthodes, la méthode LATIN couplée à la PGD permet d'exploiter les avantages de la méthode LATIN et notamment la possibilité de générer à la volée une approximation spatio-temporelle complète sur forme réduite. Cette approximation et la possibilité d'avoir la solution exacte avec la même méthode permet d'envisager le couplage entre la méthode LATIN-PGD et les méthodes de krigeage multi-fidélité présentées par la suite.

Deuxième partie

Contributions aux méthodes employées pour le couplage entre métamodèles multi-fidélité et modèles réduits

Chapitre 4

Analyse des méthodes de krigeage multi-fidelité

Sommaire

4.1 Méthodes de prise en compte de données multi-fidélité 51
4.2 Méthodes de cokrigeage 53
4.2.1 Généralisation du krigeage aux fonctions multivariées 53
4.2.2 Un exemple de cokrigeage : cokrigeage intercorrélation 54
4.2.3 Cokrigeage auto-régressif 54
4.3 Krigeage avec prise en compte de données bruitées 56
4.4 Méthodes de krigeage récursif 57
4.4.1 Krigeage hiérarchique 57
4.4.2 Evofusion 58

4.1 Méthodes de prise en compte de données multi-fidélité

L'emploi de données de sources multiples est particulièrement prisé en mécanique des fluides. Les modèles basse fidélité sont moins coûteux parce qu'ils sont généralement une simplification des modèles haute fidélité. Cette simplification peut se faire de différentes manières, par exemple en linéarisant le système, en utilisant des modèles physiques plus simples, des maillages grossiers ou des résultats partiellement convergés.

La prise en compte de ces différents types de données s'apparente à la fusion de données. Ce domaine de recherche consiste à confronter et intégrer des informations d'origine multiples dans le but de réduire l'incertitude sur l'information résultante. Elle est couramment employée dans d'autres domaines que la mécanique, notamment pour les systèmes d'information géographique, la surveillance par capteurs multiples (Sensor fusion), ou encore pour l'étude de populations [Liggins II *et al.*, 2017]. Notre cas d'étude est d'ailleurs similaire à un cas de fusion de mesures de même type, extraites de solveurs mécaniques.

Dans le cadre de ces travaux, la fusion des données sera apportée par la définition d'un lien de corrélation entre les données basse-fidélité et les données hautes-fidélités, mais il existe d'autres modèles permettant de modéliser l'incertitude sur l'information apportée comme les

probabilités inférieure et supérieure de Dempster [Dempster, 2008] ou encore les possibilités-nécessités de Dubois et Prade [Dubois et Prade, 2012].

On considère les fonctions QoI issues de s solveurs à disposition $y_1, y_2, \ldots, y_s$ avec $\forall i \in [\![1, s]\!], y_i : \mathcal{D} \mapsto \mathbb{R}$. Ces solveurs seront classés dans l'ordre de qualité croissante pour simplifier l'analyse avec y_s supposé exact. Il s'agit ici de générer un modèle liant l'entrée $\underline{x}$ à la sortie y, à partir d'un ensemble de s échantillons d'apprentissage issus de chaque solveur $\mathcal{T}_i = \left\{(\underline{x}_j^i, \mathrm{y}_j^i)\right\}_{i \in [\![1, N_i]\!]}$.

Pour simplifier la démarche, seuls deux niveaux de données seront exploités dans la suite : des données dites basse fidélité, notées $\bullet_1$ et des données haute fidélité, notées $\bullet_2$, considérées comme étant exactes. Néanmoins, ces méthodes peuvent étendre à s niveaux de fidélité.

La prise en compte de données multi-fidélité dans la génération d'un métamodèle de krigeage peut se faire de plusieurs manières.

La première solution consiste à s'appuyer sur l'extension du krigeage pour les fonctions multivariées de type $\tilde{\hat{y}} = [\hat{y}_1(\underline{x}), \hat{y}_2(\underline{x})]^{\mathrm{T}}$ pour modéliser la réponse des solveurs et donc de la quantité d'intérêt. Dans ce cadre, la distribution jointe présentée dans le cadre du krigeage (2.15), se généralise. La difficulté dans ce cas est la définition de la matrice de covariance et notamment des termes associant les données basse fidélité et les données haute fidélité. De plus, la matrice de covariance peut perdre ses propriétés définies positives dans certains cas, et le nombre d'hyperparamètres à quantifier est bien plus important ce qui rend complexe le processus de génération du métamodèle.

La seconde consiste à être capable de quantifier l'erreur sur la quantité d'intérêt et de la prendre en compte dans le métamodèle. On parle alors de krigeage avec données bruitées. L'avantage est la simplicité de génération du métamodèle, mais l'inconvénient est l'impossibilité de prendre en compte alors l'information apportée par une observation basse-fidélité et haute-fidélité en un point.

Enfin une troisième solution est de générer des métamodèles associés à chaque niveau de fidélité de façon hiérarchique. C'est le cadre exploité par les méthodes de cokrigeage hiérarchique et Evofusion.

4.2 Méthodes de cokrigeage

4.2.1 Généralisation du krigeage aux fonctions multivariées

On généralise la régression par processus gaussien (2.15) aux fonctions multivariées $\tilde{\hat{y}} = [\hat{y}_1(\underline{x}), \hat{y}_2(\underline{x})]^{\mathrm{T}}$. On obtient alors la distribution jointe suivante :

$$\begin{pmatrix} \hat{y}_2(\underline{x}) \\ \underline{\mathrm{y}}_2 \\ \underline{\mathrm{y}}_1 \end{pmatrix} \sim N\left(\begin{pmatrix} \underline{f}_2(\underline{x})^{\mathrm{T}}\underline{\beta}_2 \\ \mathbf{F}_2^{\mathrm{T}}\underline{\beta}_2 \\ \mathbf{F}_1^{\mathrm{T}}\underline{\beta}_1 \end{pmatrix}, \begin{pmatrix} k_{22}(\underline{x},\underline{x}) & \underline{\mathrm{k}}_2^{\mathrm{T}}(\underline{x}) & \underline{\mathrm{k}}_1^{\mathrm{T}}(\underline{x}) \\ \underline{\mathrm{k}}_2(\underline{x}) & \mathbf{C}_{22} & \mathbf{C}_{21} \\ \underline{\mathrm{k}}_1(\underline{x}) & \mathbf{C}_{12} & \mathbf{C}_{11} \end{pmatrix} \right) \tag{4.1}$$

Avec $\forall s \in \{1,2\}$, $\forall t \in \{1,2\}$, $\mathbf{F}_s$ matrice de régression telle que $\mathbf{F}_{sij} = (\underline{f}_s)_i(\underline{x}_j)$, $\mathbf{C}_{st}$ matrice de covariance entre les données observées telle que $(\mathbf{C}_{st})_{ij} = k_{st}\left((\underline{x}_s)_i, (\underline{x}_t)_j; \underline{\theta}_{st}\right)$ et $\underline{\mathrm{k}}(\underline{x})$ vecteur de covariance entre les données observées et les données prédites telles que $(\underline{\mathrm{k}}_s)_j(\underline{x}) = k_s\left((\underline{x}_s)_j, \underline{x}\right)$.

On note $\mathbf{F} = \begin{bmatrix} \mathbf{F}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_1 \end{bmatrix}$, $\underline{\beta} = \begin{bmatrix} \underline{\beta}_2 \\ \underline{\beta}_1 \end{bmatrix}$, $\underline{\mathrm{k}}(\underline{x}) = \begin{bmatrix} \underline{\mathrm{k}}_2(\underline{x}) \\ \underline{\mathrm{k}}_1(\underline{x}) \end{bmatrix}$ et $\mathbf{C} = \begin{bmatrix} \mathbf{C}_{22} & \mathbf{C}_{21} \\ \mathbf{C}_{12} & \mathbf{C}_{11} \end{bmatrix}$. À partir de cette notation, on retrouve la formulation univariée.

$$\begin{pmatrix} \hat{y}(\underline{x}) \\ \underline{\mathrm{y}} \end{pmatrix} \sim N\left(\begin{pmatrix} \underline{f}_2(\underline{x})^{\mathrm{T}}\underline{\beta}_2 \\ \mathbf{F}^{\mathrm{T}}\underline{\beta} \end{pmatrix}, \begin{pmatrix} k_{22}(\underline{x},\underline{x}) & \underline{\mathrm{k}}^{\mathrm{T}}(\underline{x}) \\ \underline{\mathrm{k}}(\underline{x}) & \mathbf{C} \end{pmatrix} \right) \tag{4.2}$$

De la même manière qu'en section 2.5, on caractérise ainsi le métamodèle comme un processus gaussien $\hat{y}(\underline{x})$ exprimé en fonction de sa moyenne $m(\underline{x})$ et sa covariance $Cov(\underline{x}, \underline{x}')$, définies par l'équation (4.3).

$$\begin{cases} m(\underline{x}) & = \underline{f}(\underline{x})^{\mathrm{T}}\underline{\beta} + \underline{\mathrm{k}}(\underline{x})^{\mathrm{T}}\mathbf{C}^{-1}(\underline{\mathrm{y}} - \mathbf{F}\underline{\beta}) \\ Cov(\underline{x}, \underline{x}') & = k_{22}(\underline{x}, \underline{x}) - \underline{\mathrm{k}}(\underline{x})^{\mathrm{T}}\mathbf{C}^{-1}\underline{\mathrm{k}}(\underline{x}') \end{cases} \tag{4.3}$$

La matrice $\mathbf{C}$ doit avoir les mêmes propriétés que dans le cas univarié, c'est-à-dire qu'elle doit être symétrique définie positive. Dans ce cas de figure, le choix de bonnes fonctions de covariance ne suffit plus : des contraintes sur le choix sur les hyperparamètres sont également nécessaires [Gneiting *et al.*, 2010, Bertram et Zimmermann, 2018] ce qui rend l'optimisation de ceux-ci plus complexe.

Dans le cadre du krigeage universel ($\underline{\beta}$ déterminé par maximisation de la fonction de vraisemblance), on a le processus gaussien défini à l'équation (4.4) fonction de sa moyenne $m(\underline{x})$

et sa covariance $Cov(\underline{x}, \underline{x}')$:

$$\begin{cases} m(\underline{x}) & = \underline{f}(\underline{x})^{\mathrm{T}}\hat{\underline{\beta}} + \underline{\mathrm{k}}(\underline{x})^{\mathrm{T}}\mathbf{C}^{-1}(\underline{\mathrm{y}} - \mathbf{F}\,\hat{\underline{\beta}}) \\ Cov(\underline{x}, \underline{x}') & = k_2(\underline{x}, \underline{x}) - \begin{pmatrix} \underline{f}^{\mathrm{T}}(\underline{x}) \\ \underline{\mathrm{k}}^{\mathrm{T}}(\underline{x}) \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \mathbf{0} & \mathbf{F}^{\mathrm{T}} \\ \mathbf{F} & \mathbf{C} \end{pmatrix}^{-1} \begin{pmatrix} \underline{f}(\underline{x}) \\ \underline{\mathrm{k}}(\underline{x}) \end{pmatrix} \\ \bar{\underline{\beta}} & = (\mathbf{F}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{C}^{-1}\underline{\mathrm{y}}, \\ \underline{u}(\underline{x}) & = \mathbf{F}^{\mathrm{T}}\mathbf{R}^{-1}\underline{r}(\underline{x}) - \underline{f}(\underline{x}) \end{cases} \tag{4.4}$$

avec $\hat{\underline{\beta}}$ la restriction de $\bar{\underline{\beta}}$ aux termes haute fidélité.

La variance associée au métamodèle est alors :

$$\mathrm{MSE}(\underline{x}) = k_2(\underline{x}, \underline{x}) - \underline{\mathrm{k}}(\underline{x})^{\mathrm{T}}\mathbf{C}^{-1}\underline{\mathrm{k}}(\underline{x}) + \underline{u}(\underline{x})^{\mathrm{T}}(\mathbf{F}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{F})^{-1}\underline{u}(\underline{x}) \tag{4.5}$$

Ces équations permettent de construire un métamodèle grâce au cokrigeage multi-fidélité. La difficulté principale se situe dans le choix des fonctions de corrélation. On expose dans la suite trois choix présents dans la littérature.

4.2.2 Un exemple de cokrigeage : cokrigeage intercorrélation

Cette première approche est proposée par [Zimmermann et Han, 2010]. Elle est la plus simple des trois méthodes de cokrigeage multi-fidélité qui sont présentées, que ce soit dans son écriture mathématique ou dans son implémentation numérique. La principale hypothèse de cette méthode est de supposer que les covariances des processus gaussien $\hat{y}_1$ et $\hat{y}_2$ sont identiques $k_1 = k_2 = k$, et que les fonctions d'intercorrélation sont égales : $k_{21} = k_{12} = \gamma k$ avec $\gamma \in [0, 1[$.

Cette hypothèse est forte. Elle suppose que le comportement des données de multiples qualités soit semblable pour assurer un métamodèle multi-fidélité de qualité.

On a alors $\mathbf{F} = \begin{bmatrix} \mathbf{F}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_1 \end{bmatrix}$, $\underline{\mathrm{k}}(\underline{x}) = \begin{bmatrix} k(\boldsymbol{X}_2, \underline{x}) \\ k(\boldsymbol{X}_1, \underline{x}) \end{bmatrix}$ et $\mathbf{C} = \begin{bmatrix} k(\boldsymbol{X}_2, \boldsymbol{X}_2) & \gamma k(\boldsymbol{X}_2, \boldsymbol{X}_1) \\ \gamma k(\boldsymbol{X}_1, \boldsymbol{X}_2) & k(\boldsymbol{X}_1, \boldsymbol{X}_1) \end{bmatrix}$.

Cet exemple se généralise à la structure de covariance suivante [Gneiting *et al.*, 2010] :

$$\begin{cases} k_{22}(\underline{x}, \underline{x}') = \sigma_2^2 r_2(\underline{x}, \underline{x}'; \underline{\theta}_2) \\ k_{11}(\underline{x}, \underline{x}') = \sigma_1^2 r_1(\underline{x}, \underline{x}'; \underline{\theta}_1) \\ k_{12}(\underline{x}, \underline{x}') = k_{21} = \gamma\sigma_1\sigma_2 r_{12}(\underline{x}, \underline{x}'; \underline{\theta}_{12}) \end{cases} \tag{4.6}$$

4.2.3 Cokrigeage auto-régressif

Cette dernière approche fût développée dans [Kennedy et O'Hagan, 2000] et est l'une des plus répandues dans la littérature [Forrester *et al.*, 2007, Miller *et al.*, 2012, Le Gratiet, 2013a].

Ce modèle s'appuie sur une hypothèse de structure markovienne [O'Hagan, 1998] selon laquelle $\forall \underline{x}$, si on se donne la valeur de $y_1(\underline{x})$, on ne peut rien apprendre de plus sur $\hat{y}_2(\underline{x})$ à partir de $\hat{y}_1(\underline{x}')$,$\forall \underline{x}' \neq \underline{x}$.

$$Cov[y_2(\underline{x}), y_1(\underline{x}')|\hat{y}_1(\underline{x})] = 0,\ \forall \underline{x}' \neq \underline{x} \tag{4.7}$$

Cette hypothèse amène à une construction sous la forme d'un modèle autorégressif, à savoir :

$$\hat{y}_2(\underline{x}) = \rho(\underline{x})\hat{y}_1(\underline{x}) + Y_c(\underline{x}) \tag{4.8}$$

où ρ représente un facteur d'échelle entre $\hat{y}_2$ et $\hat{y}_1$ avec $\rho(\underline{x}) = \dfrac{Cov(\hat{y}_2(\underline{x}), \hat{y}_1(\underline{x}))}{var(\hat{y}_1(\underline{x}))}$ et Y_c un processus gaussien indépendant de $\hat{y}_1$.

Déterminer les termes de la matrice $\mathbf{C}_{12}$ est alors possible :

$$\begin{aligned} Cov_{12}(\underline{x}_2, \underline{x}_1) &= Cov[\hat{y}_2(\underline{x}_2), \hat{y}_1(\underline{x}_1)] \\ &= Cov[\rho\hat{y}_1(\underline{x}_2) + Y_c(\underline{x}_2), \hat{y}_1(\underline{x}_1)] \\ &= Cov[\rho\hat{y}_1(\underline{x}_2), \hat{y}_1(\underline{x}_1)] + \underbrace{Cov[Y_c(\underline{x}_2), \hat{y}_1(\underline{x}_1)]}_{0} \\ &= \rho k_1(\underline{x}_2, \underline{x}_1) \end{aligned} \tag{4.9}$$

Dans ce cas, on a la structure suivante :

$$\begin{aligned} &\underline{\beta} = \begin{pmatrix} \underline{\beta}_2 \\ \underline{\beta}_1 \end{pmatrix}, \quad \underline{f}(\underline{x}) = \begin{pmatrix} \underline{f}_2(\underline{x}) \\ \rho\underline{f}_1(\underline{x}) \end{pmatrix}, \quad \mathbf{F} = \begin{bmatrix} \underline{f}_2(\boldsymbol{X}_2) & \rho\underline{f}_1(\boldsymbol{X}_2) \\ \mathbf{0} & \underline{f}_1(\boldsymbol{X}_1) \end{bmatrix} \\ &\underline{\mathrm{k}}(\boldsymbol{X}) = \begin{bmatrix} \rho^2 k_1(\boldsymbol{X}_2, \boldsymbol{X}) + k_2(\boldsymbol{X}_2, \boldsymbol{X}) \\ \rho k_1(\boldsymbol{X}_1, \boldsymbol{X}) \end{bmatrix} \\ &\mathbf{C} = \begin{bmatrix} \rho^2 k_1(\boldsymbol{X}_2, \boldsymbol{X}_2) + k_2(\boldsymbol{X}_2, \boldsymbol{X}_2) & \rho k_1(\boldsymbol{X}_2, \boldsymbol{X}_1) \\ \rho k_1(\boldsymbol{X}_1, \boldsymbol{X}_2) & k_1(\boldsymbol{X}_1, \boldsymbol{X}_1) \end{bmatrix} \end{aligned} \tag{4.10}$$

Et le processus gaussien généré a pour moyenne m et pour covariance $Cov(\underline{x}, \underline{x}')$ défini ci-dessous :

$$\begin{cases} m(\underline{x}) & = \underline{f}(\underline{x})^{\mathrm{T}}\hat{\underline{\beta}} + \underline{\mathrm{k}}(\underline{x})^{\mathrm{T}}\mathbf{C}^{-1}(\underline{\mathrm{y}} - \mathbf{F}\ \hat{\underline{\beta}}) \\ Cov(\underline{x}, \underline{x}') & = k_2(\underline{x}, \underline{x}) - \begin{pmatrix} \underline{f}^{\mathrm{T}}(\underline{x}) \\ \underline{\mathrm{k}}^{\mathrm{T}}(\underline{x}) \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} 0 & \mathbf{F}^{\mathrm{T}} \\ \mathbf{F} & \mathbf{C} \end{pmatrix}^{-1} \begin{pmatrix} \underline{f}(\underline{x}) \\ \underline{\mathrm{k}}(\underline{x}) \end{pmatrix} \\ \hat{\underline{\beta}} & = (\mathbf{F}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{F})^{-1}\mathbf{F}^{\mathrm{T}}\mathbf{C}^{-1}\underline{\mathrm{y}} \end{cases} \tag{4.11}$$

Le Gratiet [Gratiet, 2012] a montré qu'il était possible d'écrire de manière récursive cette méthode, et ce, avec ρ dépendant de $\underline{x}$. Dans le cas où $\boldsymbol{X}_1 \subset \boldsymbol{X}_2$ et d'un krigeage simple, le processus gaussien s'écrit :

$$\begin{cases} m(\underline{x}) & = \rho(\boldsymbol{X})m_1(\underline{x}) + \underline{f}_2(\underline{x})^{\mathrm{T}}\underline{\beta} + \underline{\mathrm{k}}_2(\underline{x})^{\mathrm{T}}\mathbf{C}_2^{-1}(\underline{\mathrm{y}}_2 - \rho(\boldsymbol{X}_2) \odot m_1(\boldsymbol{X}_2) - \mathbf{F}\ \hat{\underline{\beta}}) \\ Cov(\underline{x}, \underline{x}') & = \rho(\boldsymbol{X})Cov_1(\underline{x}, \underline{x}') + \sigma^2\left(1 - \underline{r}_2(\underline{x})^{\mathrm{T}}\mathbf{R}_2^{-1}\underline{r}_2(\underline{x}')\right) \end{cases} \tag{4.12}$$

Le détail et la preuve dans le cadre du cokrigeage universel sont visibles dans [Le Gratiet, 2013b].

On le voit au travers des deux exemples, les différentes méthodes de cokrigeage nécessitent de modifier la structure de covariance présente dans le cadre du krigeage ce qui implique de réimplémenter les méthodes de génération et de prédiction. De plus, le nombre de paramètres à optimiser est plus important, et l'optimisation de ces paramètres est contrainte par la nécessité d'une matrice de covariance définie positive. Pour s'affranchir de ces difficultés, on va étudier les deux autres méthodes d'implémentation.

4.3 Krigeage avec prise en compte de données bruitées

La prise en compte de données bruitées se fait aisément par la prise en compte d'une variance sur les observations. Nous supposons un bruit d'observation gaussien indépendant avec une moyenne zéro et une variance $\sigma_\varepsilon^2(\underline{x})$. Il s'agit de « l'effet nugget » employé notamment pour améliorer le conditionnement de la matrice de covariance. Les n observations sont alors définies sous la forme $\mathrm{y}_i = y(\underline{x}_i) + \sigma_\varepsilon(\underline{x}_i)\boldsymbol{\varepsilon}_i$ où $(\boldsymbol{\varepsilon}_i)$ sont indépendants et distribués de manière identique par rapport à une loi normale centrée réduite.

Par rapport au krigeage sans données bruitées, la différence se situe dans la définition de la variance associée aux observations : celle-ci prend en compte maintenant le terme de bruit.

$$\begin{pmatrix} \hat{y}(\underline{x}) \\ \underline{y} \end{pmatrix} \sim N\left(\begin{pmatrix} \underline{f}(\underline{x})^{\mathrm{T}}\underline{\beta} \\ \mathbf{F}^{\mathrm{T}}\underline{\beta} \end{pmatrix}, \begin{pmatrix} k(\underline{x},\underline{x}') & \underline{\mathrm{k}}'(\underline{x}) \\ \underline{\mathrm{k}}(\underline{x}) & \mathbf{C}+\boldsymbol{\Delta} \end{pmatrix} \right) \tag{4.13}$$

avec $\boldsymbol{\Delta}$ la matrice (diagonale) de bruit telle que $\boldsymbol{\Delta}_{ij} = \sigma_\varepsilon(\underline{x}_i)\delta_{ij}$ et δ_{ij} le symbole de Kronecker.

Le métamodèle ainsi généré n'est plus interpolant. La figure 4.1 montre la génération d'un krigeage avec données bruitées, dans le cas où l'incertitude sur celle-ci est quantifiable.

La difficulté de la méthode concerne la définition de la fonction bruit σ_ε et l'absence de lien établi entre les deux types de données. Dans le cas σ_ε constant, il est possible d'optimiser ce paramètre de la même manière que les hyperparamètres, avec plus ou moins de réussite comme en atteste la figure 4.2.

La quantification de l'erreur sur les données bruités peut se faire à partir d'observations sur l'écart entre des données bruitées, et des données exactes : on exploitera cet aspect dans le cadre de la méthode Evofusion présentée dans la section suivante sur les méthodes de krigeage récursif.

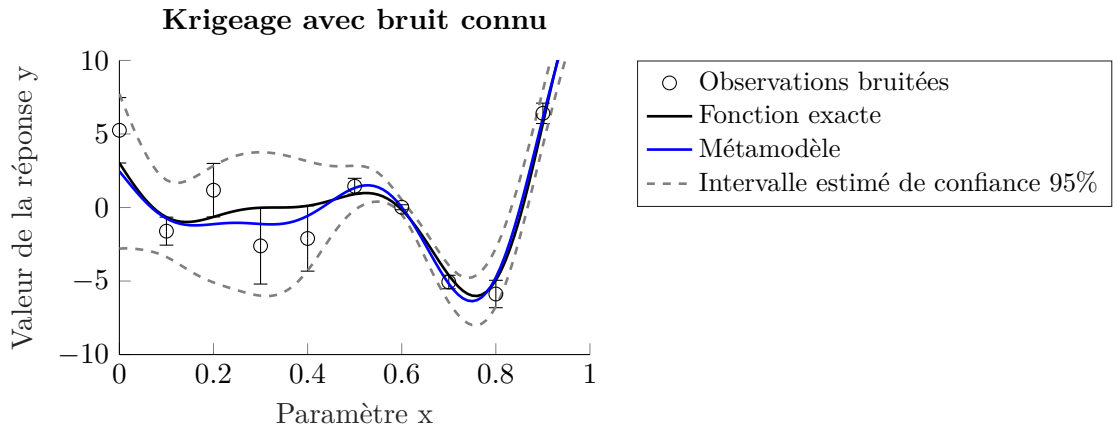


FIGURE 4.1 – Krigeage avec prise en compte des données bruitées (Bruit pour chaque observation connu)

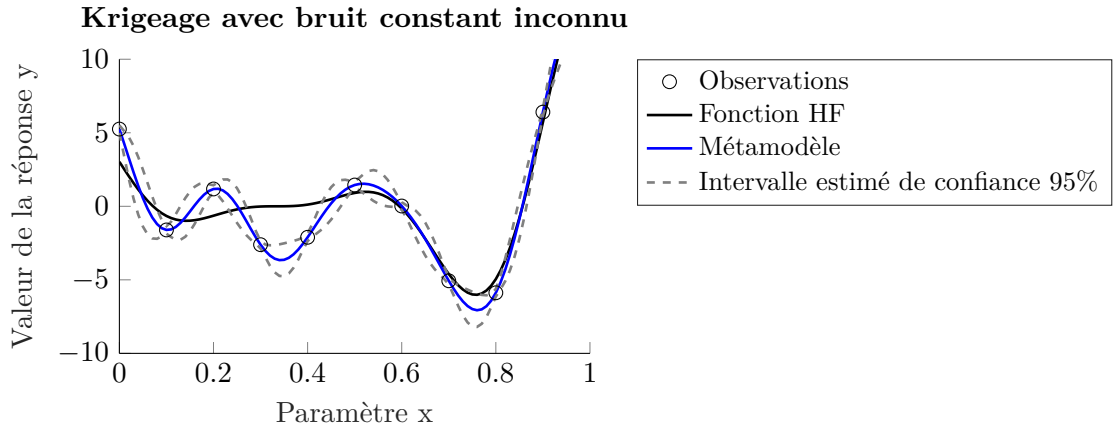


FIGURE 4.2 – Krigeage avec prise en compte des données bruitées (Bruit inconnu estimé)

4.4 Méthodes de krigeage récursif

Les deux méthodes présentées dans cette section ont l'avantage d'être simples d'implémentation et de pouvoir employer des données de multiples niveaux de fidélité.

4.4.1 Krigeage hiérarchique

Le krigeage hiérarchique a été introduit par [Han et Görtz, 2012]. L'idée du krigeage hiérarchique est de remplacer, dans les équations du krigeage, la fonction déterministe par le métamodèle des données basse fidélité, celui-ci étant pondéré par un facteur $\underline{\beta}$. Ce facteur $\underline{\beta}$ fait ainsi le lien entre les données haute et basse fidélité. On suppose que le processus gaussien utilisé pour déterminer le métamodèle est construit par :

$$\hat{y}_{HF}(\underline{x}) = \hat{y}_{BF}(\underline{x})\underline{\beta} + Z(\underline{x}) \tag{4.14}$$

avec $\hat{y}_{BF}(\underline{x})$ le métamodèle créé à partir des données $\underline{\mathrm{y}}_{BF}$ (ce métamodèle de données basse fidélité n'est pas nécessairement issu du krigeage). On constate donc que la seule modification du krigeage hiérarchique par rapport au krigeage est le remplacement du terme de régression qui devient $\hat{y}_{BF}(\underline{x})\underline{\beta}$.

4.4.2 Evofusion

Cette méthode se rapproche de la méthode de krigeage hiérarchique dans la mesure ou le métamodèle créé se base sur les données basse fidélité également en lieu et place de la fonction déterministe [Forrester *et al.*, 2006].

Algorithm 3: Evofusion Algorithm

Entrées : Observations basse fidélité $\mathcal{T}_1 = (\boldsymbol{X}_1, \underline{\mathrm{y}}_1)$ et haute fidélité $\mathcal{T}_2 = (\boldsymbol{X}_2, \underline{\mathrm{y}}_2)$
1) Génération du métamodèle issu des données basse fidélité $(\boldsymbol{X}_1, \underline{\mathrm{y}}_1) \longrightarrow \hat{y}_1$
2) Calcul de l'écart entre le métamodèle basse fidélité $\hat{y}_1$ et les données haute fidélité $\underline{\tilde{\mathrm{y}}}_2$ aux points d'observations haute fidélité $\boldsymbol{X}_2$: $y_c = y_2 - \hat{y}_1(\boldsymbol{X}_2)$
3) Génération du métamodèle comportant les données correctives $(\boldsymbol{X}_2, \underline{\mathrm{y}}_{corr}) \longrightarrow \hat{y}_{corr}$
4) Correction des observations basse-fidélité à partir du métamodèle de correction : $\underline{\mathrm{y}}_{1_c} = \underline{\tilde{\mathrm{y}}}_1 + \hat{y}_c(\boldsymbol{X}_1)$
5) Génération du métamodèle avec les données corrigées et les observations haute fidélité : $(\boldsymbol{X}_1, \underline{\mathrm{y}}_{1_c}) \bigcup (\boldsymbol{X}_2, \underline{\mathrm{y}}_2) \longrightarrow \hat{y}$
Sortie : Métamodèle multi-fidélité : $\hat{y}$

Cette technique peut être classée dans la catégorie des méthodes à correction de surface. L'idée est de créer un métamodèle représentant le modèle haute fidélité à l'aide de données haute fidélité $\mathcal{T}_2$ et des données base fidélité $\mathcal{T}_1$ corrigées. Cette réévaluation s'effectue grâce à la construction d'un métamodèle d'erreur $\hat{y}_{Err}$ visant à estimer la différence entre le modèle haute fidélité et le modèle basse fidélité.

La correction apportée par le métamodèle d'erreur $\hat{y}_{corr}$ est entachée d'une certaine imprécision due au manque d'informations sur l'erreur. Dans le cas du krigeage interpolant, le risque lors de la génération de $\hat{y}$ est d'avoir des données basse-fidélité corrigées éloignées de la valeur exacte. Sans variance associée à ces valeurs, le processus d'enrichissement peut être grandement faussée, notamment dans le cadre de l'optimisation globale EGO. L'algorithme 3 présente les détails de la méthode.

Il est possible associer à ces observations un bruit σ_ε comme proposé en section 4.3. Ce bruit peut être quantifié par la variance associée au métamodèle d'erreur. En effet, la somme des variables aléatoires $\underline{\tilde{\mathrm{y}}}_1$ et $\hat{y}_c(\boldsymbol{X}_1)$ donne une variance associée $\sigma_\varepsilon(\boldsymbol{X}_1)$ telle que

$$\sigma_\varepsilon(\boldsymbol{X}_1)^2 = Var(\underline{\tilde{\mathrm{y}}}_1)^2 + Var(\hat{y}_c(\boldsymbol{X}_1))^2 + 2Cov(\underline{\tilde{\mathrm{y}}}_1, \hat{y}_c(\boldsymbol{X}_1)) \tag{4.15}$$

L'hypothèse d'indépendance des variables est considérée : $Cov(\underline{\tilde{\mathrm{y}}}_1, \hat{y}_c(\boldsymbol{X}_1)) = 0$. De plus,

aucun bruit de mesure n'est considéré sur les observations BF initialement : $Var(\underline{\tilde{y}}_1) = 0$. On considérera donc dans la suite $\sigma_\varepsilon(\boldsymbol{X}_1) = Var(\hat{y}_c(\boldsymbol{X}_1))$. Ce raisonnement convient également pour les données HF dans la mesure où $Var(\hat{y}_c(\boldsymbol{X}_2)) = 0$ en ces points.

La figure 4.3 présente une schématisation de la méthode au travers des fonctions 4.16 issues de [Forrester *et al.*, 2006] :

$$\begin{aligned} y_1 &: x \mapsto 0.5(6x-2)^2 + \sin(12x-4) + 10(x-0.5) - 5 \\ y_2 &: x \mapsto 2y_1(x) - 20x + 20 \end{aligned} \tag{4.16}$$

On voit l'intérêt d'associer une certaine variance aux données BF. Dans le cas où les données BF corrigées sont interpolantes, le métamodèle serait fortement faussé.

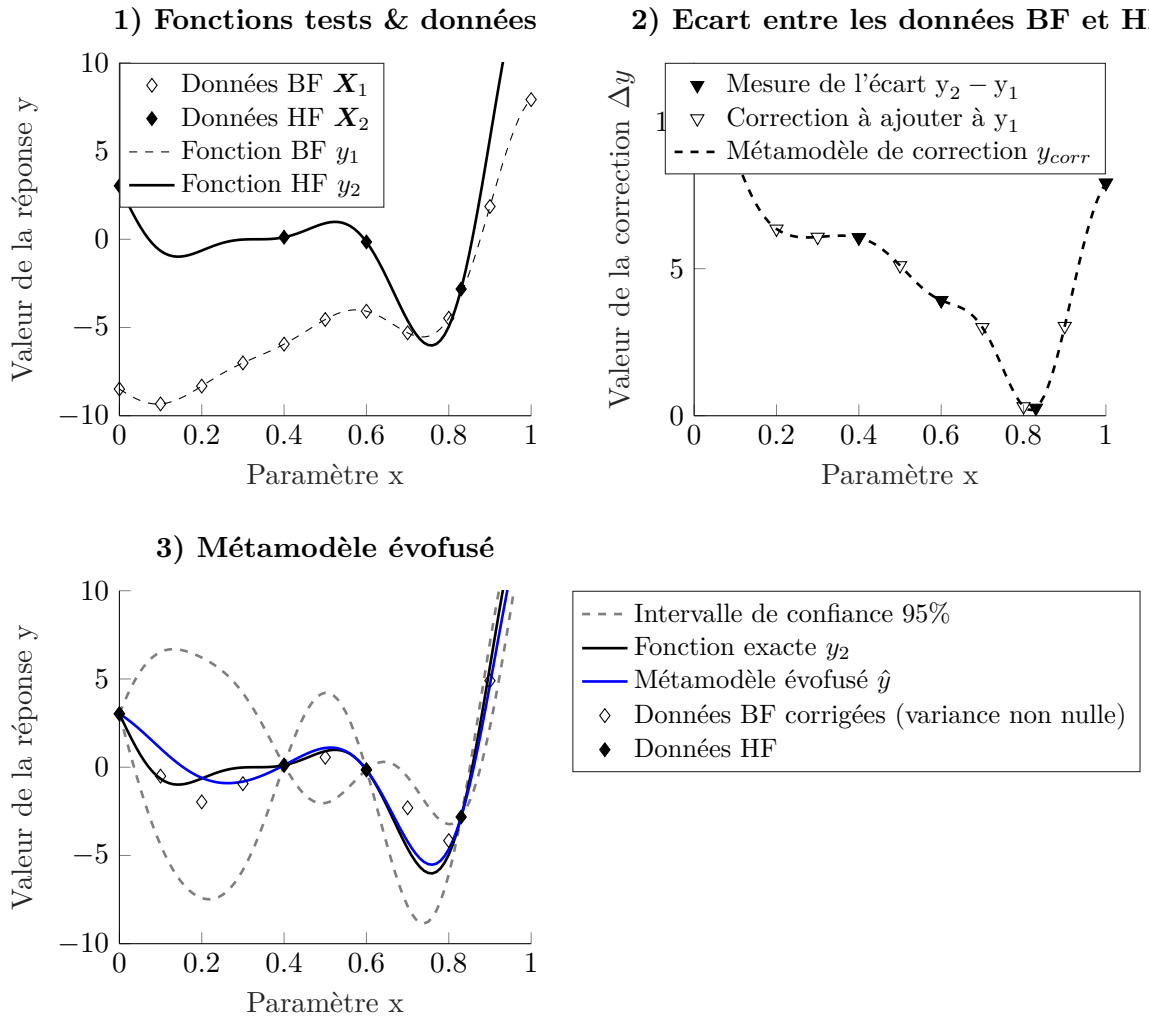


FIGURE 4.3 – Schématisation de la méthode Evofusion

Très peu de comparatifs des méthodes de krigeage multi-fidélité ont été réalisés dans la littérature [Fernández-Godino *et al.*, 2016, Park *et al.*, 2017]. [Courrier *et al.*, 2016] a montré un

cas-test mécanique que la méthode Evofusion faisait partie des plus adaptées pour la gestion des données multi-fidélité. Celle-ci présente également l'avantage d'être récursive et de ne pas nécessiter de modification majeure vis-à-vis du krigeage : cette méthode sera employée dans la suite.

Chapitre 5

Réflexion sur le cadre LATIN-PGD pour les problèmes élastoplastiques

Sommaire

5.1 Détail de l'approche variable interne de la LATIN-PGD 61
5.1.1 Formulation de l'étape locale 62
5.1.2 Calcul de l'opérateur tangent 64
5.1.3 Formulation de l'étape globale 67
5.1.4 Formulation de l'étape de mise à jour 75
5.1.5 Critère d'erreur 76
5.1.6 Analyse de l'approche à variables internes 76
5.2 Approche fonctionnelle de la LATIN 76
5.2.1 Formulation de l'étape globale 77
5.2.2 Choix de la direction de recherche élastique 78
5.3 Implémentation parallèle de la méthode 79
5.3.1 Tensorisation des champs et des opérateurs 79
5.3.2 Parallélisation des calculs 79
5.4 Mise en œuvre de la méthode sur trois cas-tests 81
5.4.1 Cas-test 1 : Barre en traction-compression 81
5.4.2 Cas-test 2 : Pièce d'amortissement 82
5.4.3 Cas-test 3 : Aube de turbine 83

5.1 Détail de l'approche variable interne de la LATIN-PGD

Cette approche de la formulation LATIN, initiée durant la thèse de [Cognard, 1989], est l'approche majoritairement employée pour les problèmes élasto-viscoplastiques [Boisse, 1987, Arzt, 1994, Passieux, 2008, Relun *et al.*, 2013]. Elle présente des propriétés de convergence intéressantes dans un certain cadre dont la preuve est visible dans [Ladevèze, 1985]. Elle s'appuie sur une représentation de l'état du matériau par l'expression de variables internes, permettant ainsi d'exploiter les équations d'évolutions de manière découplée en tout point de l'espace et du temps à partir des champs issus de l'étape globale. Les étapes présentées en section 1.3 seront détaillées ici dans le cadre LATIN-PGD associée à la loi élasto-viscoplastique de Marquis-Chaboche présentée en section 1.2.

Dans cette approche, les variables mises en jeu pour le calcul sont les couples taux de déformation plastique - contrainte $(\dot{\boldsymbol{\varepsilon}}_p, \boldsymbol{\sigma})$ et le couple associé aux variables internes $(\underline{A}_k, \underline{V}_k)$.

5.1.1 Formulation de l'étape locale

5.1.1.1 Définition de la direction de recherche horizontale

Cette étape est complexe à mettre en œuvre dans le cas d'une direction de recherche locale $\boldsymbol{\Theta}^-$ quelconque. Deux directions de recherche ont été expérimentées. La première présentée correspond à $\boldsymbol{\Theta}^- = 0$ soit la conservation des variables duales issues de l'étape globale $\hat{\boldsymbol{\sigma}} = \boldsymbol{\sigma}$ et $\hat{V}_k = \underline{V}_k$. Celle-ci a été employée dans le cadre de la thèse de Nicolas Relun [Relun, 2011] et a l'avantage d'être explicite, mais au prix d'une forte sensibilité aux perturbations et de la violation des propriétés de monotonie stricte de l'écrouissage au cours des itérations. Les propriétés de convergence de la formulation LATIN nécessitent la réécriture des variables afin d'en extraire une formulation normale, dans laquelle les équations d'état liant les variables primales et duales sont linéaires :

$$\begin{cases} \boldsymbol{\sigma} = \mathbf{K}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon_p}) \\ \underline{V}_k = \mathbf{V}\underline{A}_k \end{cases} \tag{5.1}$$

$$\begin{cases} \dfrac{d}{dt}\begin{bmatrix} \boldsymbol{\varepsilon_p} \\ -\underline{A}_k \end{bmatrix} = \mathbf{B}\left(\begin{bmatrix} \boldsymbol{\sigma} \\ \underline{V}_k \end{bmatrix}\right) \\ \boldsymbol{\varepsilon_p} = \underline{A}_k = 0 \text{ pour } t = 0 \end{cases} \tag{5.2}$$

Il est ainsi nécessaire d'effectuer un changement de variable pour les variables associées à l'écrouissage isotrope pour se ramener à ce cas de figure avec $\bar{R} = R_\infty \dot{\bar{p}}$ comme présenté dans l'algorithme 4.

Algorithm 4: Étape Locale - Direction de recherche horizontale

Entrée : $\boldsymbol{\sigma}, \mathbf{X}, \bar{R}$

Pour chaque pas de temps, élément, point de Gauss

- **Initialisation**
  - $\dot{\boldsymbol{\varepsilon}}_p = \mathbf{0}$, $\dot{\alpha} = \mathbf{0}$, $\dot{\bar{p}} = 0$
- **Calcul de la fonction seuil associée :**
  - $\delta \longleftarrow \dfrac{\sqrt{b}}{2R_\infty}\bar{R}$ | $f \longleftarrow J_2(\boldsymbol{\sigma}_D - \mathbf{X}) - \sigma_y + R_\infty\,\delta\,(2-\delta)$
- **Si** $f > 0$, **calcul des champs**
  - $\lambda \longleftarrow \left(\dfrac{f}{K}\right)^n$
  - $\dot{\boldsymbol{\varepsilon}}_p \longleftarrow \sqrt{\dfrac{3}{2}}\lambda\,\mathbb{N}$ | $\dot{\alpha} \longleftarrow \dot{\boldsymbol{\varepsilon}}_p - \dfrac{3}{2}\dfrac{\gamma}{C}\lambda\mathbf{X}$ | $\dot{\bar{p}} \longleftarrow \lambda\sqrt{R_\infty}(1-\delta)$

Sortie : $\dot{\boldsymbol{\varepsilon}}_p, \dot{\alpha}, \dot{\bar{p}}$

5.1.1.2 Définition de la direction de recherche verticale

La seconde direction de recherche correspond à la conservation de la déformation $\boldsymbol{\varepsilon}$ calculée à l'étape globale. L'intégration temporelle du comportement matériau est réalisée pour chaque point de Gauss. Cette méthode de calcul supprime la propriété d'indépendance temporelle et est plus coûteuse que la précédente. Néanmoins, elle reste parallélisable sur l'ensemble des éléments et reste négligeable vis-à-vis du coût de l'étape globale. En contrepartie, elle permet une convergence plus rapide.

Le schéma d'intégration temporelle choisi est un schéma d'Euler implicite usuellement employé pour les relations de comportement élastoplastiques. L'algorithme de détermination de l'incrément plastique est un algorithme de retour radial présenté dans [Simo et Taylor, 1986].

Algorithm 5: Etape Locale - Direction de recherche verticale

Entrée : $\boldsymbol{\varepsilon}$

Pour chaque élément, pour chaque point de Gauss

- **Initialisation**
  - $p = 0, \boldsymbol{\varepsilon_p} = \mathbf{0}, \boldsymbol{\sigma} = \mathbf{0}, \mathbf{X} = \mathbf{0}, R = 0$
- **Boucle sur les pas de temps** i_t
  - Calcul de la contrainte prédictive $\boldsymbol{\sigma} \longleftarrow \boldsymbol{\sigma} + \mathbf{K_e}\Delta\boldsymbol{\varepsilon}$
  - **Si** $J_2(\boldsymbol{\sigma} - \mathbf{X}) - \sigma_y - R > 0$ *(Cas plastique)*
    - **Recherche de** $f(\Delta p) = 0$ - *Algorithme de Newton*
      - **Initialisation**
        - $f = 1$
        - $\Delta p = \left(\dfrac{J_2(\boldsymbol{\sigma} - \mathbf{X}) - \sigma_y - R}{K}\right)^n$
      - **Tant que** $|f| < 10^{-2}$
        - Calcul de l'écart à la surface seuil corrigée
          $f \longleftarrow J_2(\Delta p) - R(\Delta p) - \sigma_y - \delta(\Delta p)$
        - Calcul de la tangente de f en Δp (ou de la sécante) et correction de Δp
    - **Calcul des champs**
      - $p \longleftarrow p + \Delta p$
      - $\boldsymbol{\varepsilon_p} \longleftarrow \boldsymbol{\varepsilon_p} + \Delta p * \mathbb{N}$
      - $\boldsymbol{\sigma} \longleftarrow \boldsymbol{\sigma} - 2\mu\Delta\boldsymbol{\varepsilon_p}$
      - $\mathbf{X} \longleftarrow \mathbf{X} + \dfrac{1}{1+\gamma\Delta p}\left(\frac{2}{3}C\dot{\boldsymbol{\varepsilon}}_p - \gamma\Delta p\mathbf{X}\right)$
      - $R \longleftarrow R_\infty(1 - e^{-bp})$

Sortie : $p, \boldsymbol{\varepsilon_p}, \boldsymbol{\sigma}, \mathbf{X}, R$

La discrétisation temporelle globale peut être sous-discrétisée indépendamment pour chaque élément à partir du taux de déformation plastique p calculé à l'étape de convergence précédente afin de réduire l'erreur d'intégration numérique. Le champ de déformation $\boldsymbol{\varepsilon}$ est alors

interpolé linéairement. La vitesse de convergence de la solution sur un point de Gauss est corrélée à son taux de déformation plastique.

5.1.2 Calcul de l'opérateur tangent

5.1.2.1 Formulation de l'opérateur

L'utilisation du couple $(\dot{\boldsymbol{\varepsilon}}_p, \boldsymbol{\sigma})$ pour la formulation présente plusieurs difficultés. La première concerne le choix de la direction de recherche globale $\boldsymbol{\Theta}^-$. Cette direction de recherche est un opérateur de $(\Omega \times I)$ dans l'espace des tenseurs d'ordre 4 faisant le lien entre $\boldsymbol{\sigma}$ et $\dot{\boldsymbol{\varepsilon}}_p$. Étant donné les propriétés de $\dot{\boldsymbol{\varepsilon}}_p$, l'opérateur devra être tel que $\boldsymbol{\Theta}^-(\boldsymbol{\sigma})$ soit déviatorique. Ainsi, on s'attend à avoir $\boldsymbol{\Theta}^-(\boldsymbol{\sigma}) = \boldsymbol{\Theta}^-(\boldsymbol{\sigma}_D)$.

Cette direction fait le lien entre la contrainte et la dérivée de la déformation plastique, donc entre des termes ayant des évolutions temporelles *a priori* décorrélées. Ainsi, il est complexe de définir un opérateur fixe à partir de la réponse du problème élastique. Une solution consiste à s'appuyer sur la définition de l'opérateur tangent $\dfrac{\partial \dot{\boldsymbol{\varepsilon}}_p}{\partial \boldsymbol{\sigma}}$. Cet opérateur est fonction du temps et de l'espace. La résolution du sous-problème global devant être effectuée sur tous les pas de temps, l'assemblage et la factorisation du calcul peuvent être coûteux si ceux-ci n'ont pas été réfléchis au préalable.

La seconde difficulté inhérente à l'utilisation de ce couple de variables concerne la présence de zones spatio-temporelles dans lesquelles le comportement matériau est purement linéaire. Dans ce cadre, l'opérateur tangent n'est pas défini. Il sera donc indispensable de régulariser l'opérateur dans certains cas.

L'opérateur tangent sera également noté $\boldsymbol{\Theta}^-$ par simplicité. Celui-ci se calcule à partir des équations de la section 1.2.

$$\boldsymbol{\Theta}^- = \frac{\partial \mathbf{B}}{\partial \boldsymbol{\sigma}} = \frac{3}{2} \frac{\partial \langle f/k \rangle_+^N}{\partial \boldsymbol{\sigma}} \otimes \mathbb{N} + \frac{3}{2} \left\langle \frac{f}{k} \right\rangle_+^N \frac{\partial \mathbb{N}}{\partial \boldsymbol{\sigma}} \tag{5.3}$$

Il est nécessaire de préciser l'obtention de $\dfrac{\partial J_2}{\partial \boldsymbol{\sigma}}$. Celui-ci s'obtient sachant que $\dfrac{\partial \sigma_D}{\partial \boldsymbol{\sigma}}$ correspond à l'opérateur déviatorique $\mathbb{1}_D$ se traduisant en notation de Voigt de cette fonction :

$$\mathbb{1}_{\mathrm{D}} = \mathbb{1} - \frac{1}{3} \underline{s} \otimes \underline{s} \tag{5.4}$$

avec $\underline{s} = [1\,1\,1\,0\,0\,0]^{\mathrm{T}}$

Les champs σ_D et $\mathbf{X}$ étant déjà déviatoriques, le terme n'a pas d'incidence ici. On obtient alors $\dfrac{\partial J_2}{\partial \boldsymbol{\sigma}} = \dfrac{3}{2}\mathbb{N}$.

Il est ainsi possible d'exprimer les termes de $\boldsymbol{\Theta}^-$:

$$\frac{\partial \langle f/k \rangle_+^N}{\partial \boldsymbol{\sigma}} = \sqrt{\frac{3}{2}} \frac{N}{f} \left(\frac{f}{k}\right)^N \mathbb{N} \quad \Big| \quad \frac{\partial \mathbb{N}}{\partial \boldsymbol{\sigma}} = \sqrt{\frac{3}{2}} \frac{1}{J_2} \mathbb{1}_D - \sqrt{\frac{3}{2}} \frac{1}{J_2} \mathbb{N} \otimes \mathbb{N} \tag{5.5}$$

Ainsi l'opérateur tangent peut s'écrire sous la forme suivante :

$$\boldsymbol{\Theta}^- = \alpha\, \mathbb{1}_D + \beta\, \mathbb{N} \otimes \mathbb{N} \quad \text{avec} : \begin{cases} \alpha &= \dfrac{3}{2} \left(\dfrac{f}{k}\right)^N \dfrac{1}{J_2} \\ \beta &= \left(\dfrac{3}{2}\right)^2 \left(\dfrac{f}{k}\right)^N \left(\dfrac{N}{f} - \dfrac{1}{J_2}\right) \end{cases} \tag{5.6}$$

Donc, à espace et temps fixés, l'opérateur tangent s'écrit se factorise en une somme de deux termes proportionnels à l'identité déviatorique et à une matrice de rang 1 de vecteur directeur unitaire $\mathbb{N}$. On retrouve une décomposition similaire à la décomposition harmonique d'un tenseur d'ordre quatre [Kachanov, 1993, Desmorat *et al.*, 2018].

L'avantage en terme d'implémentation est la possibilité d'écrire et d'effectuer les calculs à partir de la décomposition pour permettre un gain de temps de calcul non négligeable. De plus, il est ainsi possible de définir analytiquement la pseudo-matrice de Moore-Penrose [Moore, 1920, Penrose, 1955] de $\boldsymbol{\Theta}^-$.

5.1.2.2 Calcul de la pseudo-inverse

L'opérateur tangent n'est pas *a priori* inversible. En effet, l'identité déviatorique a pour noyau l'ensemble des tenseurs sphériques, et le second terme est de rang 1. Néanmoins, il est possible de définir une pseudo-inverse $\tilde{\boldsymbol{\Theta}}^-$ ayant de bonnes propriétés. En effet, la décomposition de la direction de recherche est adaptée à l'utilisation du lemme d'inversion matricielle de Shermann-Morrison permettant le calcul analytique de l'inverse d'une matrice :
Soit $M = \mathbf{A} + \underline{u}\underline{v}^{\mathrm{T}}$. Si $\mathbf{A}$ est inversible et $\underline{v}^{\mathrm{T}} \mathbf{A}^{-1} \underline{u} \neq 1$:

$$\left(\mathbf{A} + \underline{u}\underline{v}^{\mathrm{T}}\right)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1} \underline{u}\underline{v}^{\mathrm{T}} \mathbf{A}^{-1}}{1 + \underline{v}^{\mathrm{T}} \mathbf{A}^{-1} \underline{u}} \tag{5.7}$$

$\mathbb{1}_{\mathrm{D}}$ n'est pas inversible, mais a pour pseudo-inverse de Moore-Penrose $\mathbb{1}_{\mathrm{D}}$. L'expression de Sherman-Morrison se généralise dans ce cas de figure et permet le calcul analytique de la pseudo-inverse de Moore-Penrose [Riedel, 1992]. La pseudo inverse de Moore-Penrose associée à $\boldsymbol{\Theta}^-$, notée $\tilde{\boldsymbol{\Theta}}^-$ est définie dans le cas où $\alpha \neq 0$ et $(\alpha + \beta) \neq 0$:

$$\tilde{\boldsymbol{\Theta}}^- = \tilde{\alpha}\, \mathbb{1}_D + \tilde{\beta}\, \mathbb{N} \otimes \mathbb{N} \quad \text{avec} : \begin{cases} \tilde{\alpha} &= \dfrac{1}{\alpha} \\ \tilde{\beta} &= -\dfrac{\beta/\alpha^2}{1 + \beta/\alpha (\mathbb{N}^{\mathrm{T}} \mathbb{N})} = -\dfrac{\beta}{\alpha(\alpha + \beta)} \end{cases} \tag{5.8}$$

Par cette définition, on a par ailleurs $\boldsymbol{\Theta}^- \circ \tilde{\boldsymbol{\Theta}}^- = \tilde{\boldsymbol{\Theta}}^- \circ \boldsymbol{\Theta}^- = \mathbb{1}_{\mathrm{D}}$.

Vérification : On se place en notation de Voigt, à espace et temps donné. $\mathbb{N}$ se caractérise par un vecteur à 6 composantes et $\mathbb{1}_{\mathrm{D}}$ par une matrice 6x6.
Les propriétés de $\mathbb{1}_{\mathrm{D}}$ et $\mathbb{N}$ sont telles que :

$$\begin{aligned} \mathbb{1}_{\mathrm{D}}^2 &= \mathbb{1}_{\mathrm{D}} \\ \mathbb{1}_{\mathrm{D}}\mathbb{N} &= \mathbb{N} \quad \text{(Propriété déviatorique)} \\ \mathbb{N}^{\mathrm{T}}\mathbb{N} &= 1 \quad \text{(Propriété unitaire)} \end{aligned}$$

On a alors :

$$\begin{aligned} \boldsymbol{\Theta}^{-}\tilde{\boldsymbol{\Theta}}^{-} &= (\alpha\, \mathbb{1}_{\mathrm{D}} + \beta\, \mathbb{N}\mathbb{N}^{\mathrm{T}})(\tilde{\alpha}\, \mathbb{1}_{\mathrm{D}} + \tilde{\beta}\, \mathbb{N}\mathbb{N}^{\mathrm{T}}) \\ &= \alpha\tilde{\alpha}\, \mathbb{1}_{\mathrm{D}}^2 + \beta\tilde{\beta}\, \mathbb{N}(\mathbb{N}^{\mathrm{T}}\mathbb{N})\mathbb{N}^{\mathrm{T}} + \alpha\tilde{\beta}\, (\mathbb{1}_{\mathrm{D}}\mathbb{N})\mathbb{N}^{\mathrm{T}} + \tilde{\alpha}\beta\mathbb{N}(\mathbb{N}^{\mathrm{T}}\mathbb{1}_{\mathrm{D}}) \\ &= (\alpha\tilde{\alpha})\, \mathbb{1}_{\mathrm{D}} + (\alpha\tilde{\beta} + \tilde{\alpha}\beta)\, \mathbb{N}\mathbb{N}^{\mathrm{T}} \end{aligned}$$

La décomposition est vraie pour tout vecteur $\mathbb{N}$, on a $\boldsymbol{\Theta}^{-}\tilde{\boldsymbol{\Theta}}^{-} = \mathbb{1}_{\mathrm{D}}$ en imposant :

$$\begin{cases} 1 &= \alpha\tilde{\alpha} \\ 0 &= \alpha\tilde{\beta} + \tilde{\alpha}\beta \end{cases} \Longrightarrow \begin{cases} \tilde{\alpha} &= \dfrac{1}{\alpha} \\ \tilde{\beta} &= -\dfrac{\beta}{\alpha(\alpha+\beta)} \end{cases}$$

Le cheminement précédent avec $\tilde{\boldsymbol{\Theta}}^{-}\boldsymbol{\Theta}^{-}$ amène à la même égalité. À partir de ces deux équations et des propriétés de $\mathbb{1}_{\mathrm{D}}$ et $\mathbb{N}$, on vérifie aisément les propriétés d'une pseudo-inverse de Moore-Penrose.

5.1.2.3 Accélération des opérations par factorisation

L'écriture factorisée des opérateurs $\boldsymbol{\Theta}^{-}$ et $\tilde{\boldsymbol{\Theta}}^{-}$ permet de réduire à la fois le stockage mémoire ainsi que le temps de calcul. Concernant le temps de calcul, les opérations matricielles peuvent être effectuées à partir de la forme factorisée ce qui est bénéfique, notamment dans le cas d'une décomposition PGD des champs mécaniques.

Une des opérations les plus coûteuses concerne le calcul de la double contraction $(\mathbf{U}, \mathbf{V}) \mapsto \boldsymbol{\Theta}^{-}\mathbf{U} : \mathbf{V}$, présente dans le calcul des fonctions temporelles. Néanmoins, une astuce technique consiste à exploiter les propriétés de $\boldsymbol{\Theta}^{-}$ et de calculer une formulation de la racine carrée matricielle de $\boldsymbol{\Theta}^{-}$, notée $\sqrt{\boldsymbol{\Theta}^{-}}$. Celle-ci s'obtient également sous la même forme factorisée.

Celle-ci vérifie à espace et temps donné, $\sqrt{\boldsymbol{\Theta}^{-}}^{2} = \boldsymbol{\Theta}^{-}$ et permet ainsi d'écrire $\sqrt{\boldsymbol{\Theta}^{-}}\mathbf{U} : \sqrt{\boldsymbol{\Theta}^{-}}\mathbf{V}$ au lieu de $\boldsymbol{\Theta}^{-}\mathbf{U} : \mathbf{V}$.

Dans le cas où $\alpha \geq 0$ et $(\alpha + \beta) \geq 0$, $\sqrt{\boldsymbol{\Theta}^{-}}$ est définie et vérifie :

$$\sqrt{\boldsymbol{\Theta}^{-}} = \check{\alpha}\, \mathbb{1}_D + \check{\beta}\, \mathbb{N} \otimes \mathbb{N} \quad \text{avec :} \begin{cases} \check{\alpha} &= \sqrt{\alpha} \\ \check{\beta} &= \sqrt{\alpha+\beta} - \sqrt{\alpha} \end{cases} \tag{5.9}$$

Vérification : On se place dans le même cadre que la preuve précédente.

$$\begin{aligned}\sqrt{\boldsymbol{\Theta}^{-}}^{2} &= (\check{\alpha}\,\mathbb{1}_{\mathrm{D}} + \check{\beta}\,\mathbb{N}\mathbb{N}^{\mathrm{T}})^2 \\ &= (\check{\alpha}^2\,\mathbb{1}_{\mathrm{D}}^2 + \check{\beta}^2\,\mathbb{N}(\mathbb{N}^{\mathrm{T}}\mathbb{N})\mathbb{N}^{\mathrm{T}}) + \check{\alpha}\check{\beta}(\mathbb{1}_{\mathrm{D}}\mathbb{N}\mathbb{N}^{\mathrm{T}} + \mathbb{N}\mathbb{N}^{\mathrm{T}}\mathbb{1}_{\mathrm{D}}) \\ &= \check{\alpha}^2\,\mathbb{1}_{\mathrm{D}} + (\check{\beta}^2 + 2\check{\alpha}\check{\beta})\,\mathbb{N}\mathbb{N}^{\mathrm{T}}\end{aligned}$$

La décomposition est vraie pour tout vecteur $\mathbb{N}$. On a alors :

$$\left\{\begin{array}{ll}\alpha &= \check{\alpha}^2 \\ \beta &= \check{\beta}^2 + 2\check{\alpha}\check{\beta}\end{array}\right. \Longrightarrow \left\{\begin{array}{ll}\check{\alpha} &= \sqrt{\alpha} \\ \check{\beta} &= \sqrt{\alpha+\beta} - \sqrt{\alpha}\end{array}\right.$$

La classe HDOWN présente dans ROMlab exploite les techniques présentées pour l'utilisation et les opérations matricielles dans lesquelles $\boldsymbol{\Theta}^{-}$ ou $\tilde{\boldsymbol{\Theta}}^{-}$ sont présents.

5.1.2.4 Régularisation de l'opérateur tangent dans le cas élastique

La direction de recherche présente la particularité de ne pas être définie lorsque le comportement du matériau en une position ou un instant donné est un comportement élastique. [Relun, 2011] a montré qu'une régularisation de type $\boldsymbol{\Theta}^{-} + 10^{-3}\rho_{max}\mathbb{1}_{\mathrm{D}}$ avec ρ_{max} la valeur propre maximale observée de $\boldsymbol{\Theta}^{-}$ était adéquat. Cette régularisation sera utilisée ici et exploitera la factorisation de l'opérateur tangent pour déterminer rapidement ρ_{max}.

5.1.3 Formulation de l'étape globale

5.1.3.1 Relations entre les champs mécaniques

À chaque étape globale, un certain nombre de champs mécaniques sont calculés. Les champs cinématiquement admissibles à zéro de déplacement U, de déformation $\boldsymbol{\varepsilon}$, de déformation plastique $\boldsymbol{\varepsilon_p}$ et de contrainte $\boldsymbol{\sigma}$ sont corrélés spatialement via différents opérateurs.

De plus, dans le cadre d'emploi de la PGD, la décomposition des champs sous la forme de fonctions du temps et de l'espace permet de calculer plus aisément les différents champs et même de prendre en compte la même fonction temporelle pour chacun des termes.

On décompose les champs $\bullet$ en partie admissible issue du problème élastique $\bullet_{CA}$ et en une correction issue cinématiquement et statiquement admissible à zéro $\bullet_{CA_0}$. Cette correction se

décompose sous la forme de modes spatiaux $\Lambda_\bullet$ et temporels $\lambda_\bullet$.

$$\begin{aligned} \mathrm{U}(M,t) &= \mathrm{U}_{CA}(M,t) + \lambda_U(t)\Lambda_U(M) \\ \boldsymbol{\sigma}(M,t) &= \boldsymbol{\sigma}_{CA}(M,t) + \lambda_{\boldsymbol{\sigma}}(t)\,\Lambda_{\boldsymbol{\sigma}}(M) \\ \boldsymbol{\varepsilon}(M,t) &= \boldsymbol{\varepsilon}_{CA}(M,t) \;+\; \lambda_{\boldsymbol{\varepsilon}}(t)\;\Lambda_{\boldsymbol{\varepsilon}}(M) \\ \boldsymbol{\varepsilon_p}(M,t) &= \lambda(t)\Lambda(M) \end{aligned} \tag{5.10}$$

L'écriture du problème d'équilibre sous forme faible permet de mettre en évidence le lien entre les fonctions temporelles associées à $\boldsymbol{\varepsilon_p}$ et $\boldsymbol{\varepsilon}$. En insérant la relation de Hooke entre $\boldsymbol{\sigma}$ et $\boldsymbol{\varepsilon_e}$, et la décomposition de $\boldsymbol{\varepsilon_e}$ dans celle-ci, le problème revient à chercher $\Lambda_{\boldsymbol{\varepsilon}}(M) \in \mathcal{U}$ et $\lambda_{\boldsymbol{\varepsilon}}(t)$ tels que $\forall \mathrm{U}^* \in \mathcal{U}^*$,

$$\int_{I\times\Omega} \mathbf{K}_e\left(\lambda_{\boldsymbol{\varepsilon}}(t)\Lambda_{\boldsymbol{\varepsilon}}(M) - \lambda(t)\Lambda(M)\right) : \boldsymbol{\varepsilon}(\mathrm{U}^*(M,t))\,\mathrm{d}\Omega\mathrm{d}t = 0 \tag{5.11}$$

Dans la suite de la démonstration, les variables des différents champs seront omises pour simplifier les expressions.

En introduisant dans (5.11) la décomposition PGD dans le champ test $U^* = \Lambda_U^*\lambda_U + \Lambda_U\lambda_U^*$ avec $\Lambda_U^* \in \mathcal{U}^*$ et λ_U^* quelconque, on obtient :

$\forall \Lambda_U^* \in \mathcal{U}^*$, $\forall \lambda_U^*$,

$$\int_{I\times\Omega} \mathbf{K}_e\left(\lambda_{\boldsymbol{\varepsilon}}\Lambda_{\boldsymbol{\varepsilon}} - \lambda\Lambda\right) : \boldsymbol{\varepsilon}\left(\Lambda_U^*\right)\lambda_U\,\mathrm{d}\Omega\mathrm{d}t + \int_{I\times\Omega} \mathbf{K}_e\left(\lambda_{\boldsymbol{\varepsilon}}\Lambda_{\boldsymbol{\varepsilon}} - \lambda\Lambda\right) : \boldsymbol{\varepsilon}\left(\Lambda_U\right)\lambda_U^*\,\mathrm{d}\Omega\mathrm{d}t = 0 \tag{5.12}$$

L'équation (5.12) étant valable quelles que soient λ_U^* et Λ_U^*, on en extrait deux équations découplées :

$$\int_I \left(\lambda_{\boldsymbol{\varepsilon}}\left[\int_\Omega \mathbf{K}_e\Lambda_{\boldsymbol{\varepsilon}} : \boldsymbol{\varepsilon}\left(\Lambda_U\right)\mathrm{d}\Omega\right] - \lambda\left[\int_\Omega \mathbf{K}_e\Lambda : \boldsymbol{\varepsilon}\left(\Lambda_U\right)\mathrm{d}\Omega\right]\right)\lambda_U^*\,\mathrm{d}t = 0 \tag{5.13}$$

$$\int_\Omega \mathbf{K}_e\left(\Lambda_{\boldsymbol{\varepsilon}}\left[\int_I \lambda_{\boldsymbol{\varepsilon}}\lambda_U\,\mathrm{d}t\right] - \Lambda\left[\int_I \lambda\lambda_U\,\mathrm{d}t\right]\right) : \boldsymbol{\varepsilon}\left(\Lambda_U^*\right)\mathrm{d}\Omega = 0 \tag{5.14}$$

L'équation (5.13) étant valable quelle que soit λ_U^*, on obtient l'existence d'une relation de proportionnalité entre λ et $\lambda_{\boldsymbol{\varepsilon}}$. Étant donné l'unicité de la décomposition PGD à un facteur de proportionnalité près, on peut imposer $\lambda = \lambda_{\boldsymbol{\varepsilon}}$.

L'équation (5.14) nous donne la méthode d'obtention de Λ_U et $\Lambda_{\boldsymbol{\varepsilon}}$ à partir de Λ. En effet, en substituant $\lambda_{\boldsymbol{\varepsilon}}$ et en exploitant le fait qu'on recherche Λ_U tel que $\Lambda_{\boldsymbol{\varepsilon}} = \boldsymbol{\varepsilon}\left(\Lambda_U\right)$ l'équation devient la suivante : $\forall \Lambda_U^* \in \mathcal{U}^*$,

$$\int_\Omega \mathbf{K}_e\,\boldsymbol{\varepsilon}\left(\Lambda_U\right) : \boldsymbol{\varepsilon}\left(\Lambda_U^*\right)\mathrm{d}\Omega = \int_\Omega \mathbf{K}_e\,\Lambda : \boldsymbol{\varepsilon}\left(\Lambda_U^*\right)\mathrm{d}\Omega \tag{5.15}$$

On retrouve un problème élastique se discrétisant sous la forme d'un problème $\mathbf{K}\Lambda_U = \underline{\mathrm{F}}$ avec Λ_U admissible à zéro.

En exprimant la loi de Hooke, et en exploitant la décomposition proposée, on a l'équation suivante :

$$\Lambda_{\boldsymbol{\sigma}}\lambda_{\boldsymbol{\sigma}} = \mathbf{K}_e\left(\Lambda_{\boldsymbol{\varepsilon}} - \Lambda\right)\lambda \tag{5.16}$$

La correction de la contrainte peut alors se décomposer de la manière suivante :

$$\begin{cases} \Lambda_{\boldsymbol{\sigma}} = \mathbf{K}_e(\Lambda_{\boldsymbol{\varepsilon}} - \Lambda) \\ \lambda_{\boldsymbol{\sigma}} = \lambda \end{cases} \tag{5.17}$$

On définit les opérateurs linéaires $\boldsymbol{U}$, $\boldsymbol{E}$, $\boldsymbol{C}$ et $\boldsymbol{C_{\mathbf{D}}}$ tels que :

$$\Lambda_U = \boldsymbol{U}(\Lambda),\ \Lambda_{\boldsymbol{\varepsilon}} = \boldsymbol{E}(\Lambda),\ \Lambda_{\boldsymbol{\sigma}} = \boldsymbol{C}(\Lambda),\ \Lambda^{\mathrm{D}}_{\boldsymbol{\sigma}} = \Lambda_{\boldsymbol{\sigma}} - 1/3\ \mathrm{tr}\left(\Lambda_{\boldsymbol{\sigma}}\right) = \boldsymbol{C_{\mathbf{D}}}(\Lambda) \tag{5.18}$$

Dans le cas d'étude, ces opérateurs sont supposés indépendants du temps.

Au bilan, on obtient donc les relations suivantes :

$$\begin{cases} \Lambda_U = \boldsymbol{U}(\Lambda) \\ \Lambda_{\boldsymbol{\varepsilon}} = \boldsymbol{E}\left(\Lambda\right) = \boldsymbol{\varepsilon}(\boldsymbol{U}(\Lambda)) \\ \Lambda_{\boldsymbol{\sigma}} = \boldsymbol{C}\left(\Lambda\right) = \mathbf{K}_e(\boldsymbol{E}(\Lambda) - \Lambda) \end{cases} \qquad \begin{cases} \lambda_U = \lambda \\ \lambda_{\boldsymbol{\varepsilon}} = \lambda \\ \lambda_{\boldsymbol{\sigma}} = \lambda \end{cases} \tag{5.19}$$

L'ensemble des champs mécaniques s'expriment donc en fonction de Λ et λ. La décomposition PGD de la correction de $\boldsymbol{\varepsilon_p}$ étant unique à un facteur multiplicatif près, on décide de normer Λ selon la norme 2. Le poids du mode est alors porté par λ.

La classe Global_Fields présente dans ROMlab permet de récupérer rapidement et avec une empreinte mémoire faible les champs U, $\boldsymbol{\varepsilon}$, $\boldsymbol{\sigma}$ et $\boldsymbol{\varepsilon}_p$ sous forme spatio-temporelle, ou sous une forme à variable séparée.

5.1.3.2 Écriture de l'erreur en direction de recherche

On rappelle que l'étape globale consiste à trouver $\Delta\boldsymbol{\varepsilon_p}$ sous une forme à variable séparée telle que le couple $(\Delta\boldsymbol{\varepsilon_p}, \Delta\boldsymbol{\sigma})$ vérifie l'admissibilité statique et cinématique à zéro, ainsi que l'équation d'équilibre. Le lien avec la variété Γ est fait à partir de la direction de recherche :

$$\boldsymbol{\Theta}^{-}(\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}}) - (\dot{\boldsymbol{\varepsilon}}_p - \hat{\dot{\boldsymbol{\varepsilon}}}_p) = \mathbf{0} \tag{5.20}$$

La recherche de la solution sous forme PGD impose de relâcher la condition sur la direction de recherche pour éviter un problème sur-contraint. Celle-ci est minimisée vis-à-vis d'une certaine norme $\mathbf{M}$.

Connaissant les couples précédemment calculés $(\tilde{\dot{\boldsymbol{\varepsilon}}}_p, \tilde{\boldsymbol{\sigma}}) \in \mathbf{A}_{\mathrm{d}}$ et $(\hat{\dot{\boldsymbol{\varepsilon}}}_p, \hat{\boldsymbol{\sigma}}) \in \Gamma$, le problème de minimisation s'écrit :

Trouver $\Delta\boldsymbol{\varepsilon_p}(M,t) \in \mathcal{W}^0$ tel que,

$$\Delta\boldsymbol{\varepsilon_p} = \underset{\Delta\boldsymbol{\varepsilon_p}\in\mathcal{W}^0}{\operatorname{argmin}} \left\| \boldsymbol{\Theta}^-(\boldsymbol{\sigma}-\hat{\boldsymbol{\sigma}}) - (\dot{\varepsilon}_p - \hat{\dot{\varepsilon}}_p) \right\|_{\mathbf{M}} \tag{5.21}$$

avec $\mathcal{W}_0$ l'espace obtenu à partir de $\mathcal{W}$ avec une mise à zéro de la condition limite $\underline{\mathrm{U}}_d = \underline{0}$, $\boldsymbol{\sigma} = \tilde{\boldsymbol{\sigma}} + \boldsymbol{C}(\Delta\boldsymbol{\varepsilon_p})$ et $\dot{\varepsilon}_p = \tilde{\dot{\varepsilon}}_p + \Delta\boldsymbol{\varepsilon_p}$

La solution étant recherchée sous la forme d'une approximation PGD, le problème (5.21) s'écrit également :

Trouver $(\Lambda(M), \lambda(t)) \in \mathcal{U}^0 \times \mathcal{I}$ tel que,

$$(\Lambda,\lambda) = \underset{(\Lambda,\lambda)\in\mathcal{U}^0\times\mathcal{I}}{\operatorname{argmin}} \left\| \boldsymbol{\Theta}^-(\boldsymbol{C}(\Lambda)\lambda) - \Lambda\dot{\lambda} + Q \right\|_{\mathbf{M}} \tag{5.22}$$

avec $\mathcal{U}^0$ l'espace obtenu à partir de $\mathcal{U}$ avec une mise à zéro de la condition limite $\underline{\mathrm{U}}_d = \underline{0}$ et $Q = \boldsymbol{\Theta}^-(\tilde{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}) - (\tilde{\dot{\varepsilon}}_p - \hat{\dot{\varepsilon}}_p)$.

La semi-norme mise en jeu $\|\bullet\|_{\mathbf{M}}$ est définie par :

$$\begin{array}{rccl} & \mathcal{W}^0 & \mapsto & \mathbb{R} \\ \|\bullet\|_{\mathbf{M}} : & \boldsymbol{\varepsilon_p}(M,t) & \mapsto & \displaystyle\int_{I\times\Omega} \boldsymbol{\varepsilon_p}(M,t) : \mathbf{M} : \boldsymbol{\varepsilon_p}(M,t)\, \mathrm{d}\Omega\mathrm{d}t \end{array} \tag{5.23}$$

Pour se rapprocher de la semi-norme énergétique, le choix $\mathbf{M} = \tilde{\boldsymbol{\Theta}}^-$ est le choix le plus approprié.

Les fonctions tests sont issues de $\mathcal{W}^0$ et peuvent s'écrire sous la forme suivante :

$$\forall {\boldsymbol{\varepsilon_p}}^* \in \mathcal{W}^0, \exists(\Lambda^*,\lambda^*) \in \mathcal{U}^0\times\mathcal{I} \,/\, {\boldsymbol{\varepsilon_p}}^* = \Lambda^*\lambda + \Lambda\lambda^* \tag{5.24}$$

La fonctionnelle du problème (5.21) peut se décomposer en deux termes.

$\forall(\Lambda^*,\lambda^*) \in \mathcal{U}^0\times\mathcal{I}$,

$$\int_{I\times\Omega} \left[\boldsymbol{\Theta}^-(\boldsymbol{C}(\Lambda^*)\lambda) - \Lambda^*\dot{\lambda} + Q\right] : \left[\boldsymbol{C}_{\mathbf{D}}(\Lambda)\lambda - \tilde{\boldsymbol{\Theta}}^-(\Lambda\dot{\lambda}) + \tilde{\boldsymbol{\Theta}}^-(Q)\right] \mathrm{d}\Omega\mathrm{d}t = 0 \tag{5.25}$$

$$\int_{I\times\Omega} \left[\boldsymbol{\Theta}^-(\boldsymbol{C}(\Lambda)\lambda^*) - \Lambda\dot{\lambda}^* + Q\right] : \left[\boldsymbol{C}_{\mathbf{D}}(\Lambda)\lambda - \tilde{\boldsymbol{\Theta}}^-(\Lambda\dot{\lambda}) + \tilde{\boldsymbol{\Theta}}^-(Q)\right] \mathrm{d}\Omega\mathrm{d}t = 0 \tag{5.26}$$

Ainsi, le problème de minimisation peut se décomposer en deux sous-problèmes couplés. Le premier porte sur la recherche de la meilleure fonction spatiale, connaissant la fonction temporelle (5.25). Le second sous-problème correspond à la recherche de la meilleure fonction temporelle, la fonction spatiale (5.26) étant connue. La recherche du minimum peut se faire par l'algorithme des directions alternées.

Il est possible de résoudre le problème de deux manières. La première consiste à écrire la stationnarité de la fonctionnelle à minimiser, la seconde consiste à réaliser une minimisation approchée au moyen de la méthode de Galerkin. Dans le cadre des thèses de Passieux [Passieux, 2008] et de Relun [Relun, 2011], les deux sous-problèmes ont été résolus par des méthodes de minimisation du résidu. Cette méthode étant complexe à mettre en œuvre et coûteuse, une approche hybride a été envisagée dans la thèse de Bhattacharyya [Bhattacharyya *et al.*, 2018] en résolvant le problème spatial par une approche de Galerkin. Cette approche a été privilégiée dans cette thèse.

Le critère d'arrêt de l'algorithme des directions alternées est la stagnation de la fonction temporelle λ par rapport à la précédente notée $\tilde{\lambda}$, $\left\langle \lambda - \tilde{\lambda}, \lambda - \tilde{\lambda} \right\rangle \geq \frac{1}{2}\nu_U \left(\langle \lambda, \lambda \rangle + \left\langle \tilde{\lambda}, \tilde{\lambda} \right\rangle \right)$ avec $\nu_U = 10^{-6}$. Usuellement, ce critère est atteint après 3 itérations.

Une relaxation est habituellement mise en place pour améliorer la convergence avec $\dot{\boldsymbol{\varepsilon}}_p = \tilde{\dot{\boldsymbol{\varepsilon}}}_p + \mu \Delta \dot{\boldsymbol{\varepsilon}}_p$. Une étude paramétrique a montré que le coefficient $\mu = 0.8$ était optimal pour les cas tests.

Une fois le nouveau couple (Λ,λ) généré, Λ est orthonormalisée vis-à-vis de la base des modes spatiaux générés précédemment suivant la norme 2.

5.1.3.3 Calcul de la fonction spatiale par approximation de Galerkin

On suppose connaître la fonction temporelle $\lambda \in \mathcal{L}^2(I)$. On considère $\mathcal{S}_0$ l'espace des champs statiquement admissible à zéro $\boldsymbol{\sigma}$ tel que $\int_{I\times\Omega} \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\mathrm{U}^*)\, \mathrm{d}\Omega\mathrm{d}t = 0$, $\forall \mathrm{U}^* \in \mathcal{W}_0$.

La fonction spatiale doit vérifier l'équation d'équilibre :

$$\int_{I\times\Omega} \Delta\boldsymbol{\varepsilon} : \boldsymbol{\sigma}^*\, \mathrm{d}\Omega\mathrm{d}t = 0, \quad \forall \boldsymbol{\sigma}^* \in \mathcal{S}_0 \tag{5.27}$$

Celle-ci se vérifie également avec $\Delta\dot{\boldsymbol{\varepsilon}}$. On obtient avec la loi de Hooke,

$$\int_{I\times\Omega} \left[\mathbf{K}_e^{-1} \Delta\dot{\boldsymbol{\sigma}} + \Delta\dot{\boldsymbol{\varepsilon}}_p \right] : \boldsymbol{\sigma}^*\, \mathrm{d}\Omega\mathrm{d}t = 0, \quad \forall \boldsymbol{\sigma}^* \in \mathcal{S}_0 \tag{5.28}$$

On injecte de direction de recherche (5.20) pour obtenir l'équation suivante :

$$\int_{I\times\Omega} \left[\mathbf{K}_e^{-1} \Delta\dot{\boldsymbol{\sigma}} + \boldsymbol{\Theta}^- \boldsymbol{\sigma} - Q \right] : \boldsymbol{\sigma}^*\, \mathrm{d}\Omega\mathrm{d}t = 0, \quad \forall \boldsymbol{\sigma}^* \in \mathcal{S}_0 \tag{5.29}$$

La décomposition PGD permet d'obtenir l'expression suivante :

$$\int_\Omega \left[\left(\int_I \dot{\lambda}\lambda\, \mathrm{d}t \right) \mathbf{K}_e^{-1} \Lambda_{\boldsymbol{\sigma}} + \left(\int_I \boldsymbol{\Theta}^- \lambda^2\, \mathrm{d}t \right) \Lambda_{\boldsymbol{\sigma}} - \int_I Q\lambda\, \mathrm{d}t \right] : \Lambda_{\boldsymbol{\sigma}}^*\, \mathrm{d}\Omega = 0 \tag{5.30}$$

On définit le changement de variable homogène à une déformation :

$$\Lambda_{\bar{\varepsilon}} = \mathbb{Z}\Lambda_{\boldsymbol{\sigma}} - \delta \tag{5.31}$$

avec $a = \int_I \dot{\lambda}\lambda \,\mathrm{dt} = \lambda_{end}^2 - \lambda_0^2$, $\mathbb{Z} = a\mathbf{K}_e^{-1} + \left(\int_I \boldsymbol{\Theta}^- \lambda^2 \,\mathrm{dt}\right)$, $\delta = \int_I Q\lambda \,\mathrm{dt}$.

$\mathbb{Z}$ est supposé inversible en tout point de l'espace et $\Lambda_{\bar{\boldsymbol{\varepsilon}}}$ doit vérifier par construction les équations d'admissibilité et d'équilibre. L'équation d'équilibre (5.27) peut ainsi s'écrire comme la recherche d'une fonction $\Lambda_{\bar{\mathrm{U}}} \in \mathcal{U}^0(\Omega)$ telle que

$$\int_\Omega \left[\mathbb{Z}^{-1}\boldsymbol{\varepsilon}(\Lambda_{\bar{\mathrm{U}}}) + \mathbb{Z}^{-1}\delta\right] : \boldsymbol{\varepsilon}(\Lambda^*_{\bar{\mathrm{U}}}) \,\mathrm{d}\Omega = 0, \quad \forall \Lambda^*_{\bar{\mathrm{U}}} \in \mathcal{U}^0(\Omega) \tag{5.32}$$

La résolution de ce problème peut se faire de manière approximée au moyen d'une minimisation de Galerkin. La solution approchée s'obtient la résolution du système :

$$\tilde{\mathbf{K}}\Lambda_{\bar{\mathrm{U}}} = \underline{\tilde{\mathrm{F}}} \tag{5.33}$$

avec $\tilde{\mathbf{K}} = \left[\int_{\Omega_{el}} \mathbf{B}^{\mathrm{T}}\mathbb{Z}^{-1}\mathbf{B} \,\mathrm{d}\Omega_{el}\right]$ et $\underline{\tilde{\mathrm{F}}} = \left[\int_{\Omega_{el}} \mathbf{B}^{\mathrm{T}}\mathbb{Z}^{-1}\delta \,\mathrm{d}\Omega_{el}\right]$.

À partir de $\Lambda_{\bar{\mathrm{U}}}$, on obtient $\Lambda_{\bar{\boldsymbol{\varepsilon}}}$ et $\Lambda_{\boldsymbol{\sigma}} = \mathbb{Z}^{-1}(\Lambda_{\bar{\boldsymbol{\varepsilon}}} + \delta)$.

En exploitant l'équation d'équilibre (5.28) sous la forme PGD, et l'équation (5.30) après injection du changement de variable (5.31), on a :

$$\int_I \left[a\mathbf{K}_e^{-1}\Lambda_{\boldsymbol{\sigma}} + a\Lambda - \Lambda_{\bar{\boldsymbol{\varepsilon}}}\right] : \Lambda^*_{\boldsymbol{\sigma}} \,\mathrm{dt} = 0, \quad \forall \Lambda^*_{\boldsymbol{\sigma}} \tag{5.34}$$

On obtient $\Lambda_{\bar{\boldsymbol{\varepsilon}}} = a\left(\mathbf{K}_e^{-1}\Lambda_{\boldsymbol{\sigma}} + \Lambda\right)$, à comparer avec $\Lambda_{\boldsymbol{\varepsilon}} = \mathbf{K}_e^{-1}\Lambda_{\boldsymbol{\sigma}} + \Lambda$.

On constate donc que pour $a \neq 0$, $\begin{cases} \Lambda_{\boldsymbol{\varepsilon}} & = {}^1\!/_a\, \Lambda_{\bar{\boldsymbol{\varepsilon}}} \\ \Lambda_U & = {}^1\!/_a\, \Lambda_{\bar{\mathrm{U}}} \\ \Lambda & = \Lambda_{\boldsymbol{\varepsilon}} - \mathbf{K}_e^{-1}\Lambda_{\boldsymbol{\sigma}} \end{cases}$

L'algorithme 6 de recherche approchée de la fonction spatiale découle de l'analyse précédente. Celle-ci est écrite de manière discrète pour mettre en évidence les astuces de calcul. On note $\mathbf{I}_t$ la matrice d'intégration temporelle et ω le poids d'intégration spatiale.

Cette étape est la plus coûteuse dans le cas de problèmes de taille conséquente. Le fait d'avoir une factorisation de $\boldsymbol{\Theta}^-$ et de pouvoir générer une factorisation de Cholesky de $\mathbb{Z}$ accélère grandement la construction des opérateurs. Néanmoins la nécessité de construire à chaque génération de fonction spatiale un problème spatial complet et sa factorisation interroge quant à l'emploi de la méthode dans un code de calcul. Soulignons également que les fonctions spatiales Λ_U, $\Lambda_{\boldsymbol{\varepsilon}}$, $\Lambda_{\boldsymbol{\sigma}}$ peuvent être régénérées à partir de Λ. Il est ainsi possible de ne pas les garder en mémoire si besoin.

5.1.3.4 Calcul de la fonction temporelle par minimisation de l'erreur en direction de recherche

La fonction temporelle est recherchée comme étant la fonction minimisant l'erreur en direction de recherche, connaissant la fonction spatiale Λ. La fonction doit alors vérifier la stationnarité

Algorithm 6: Génération de la fonction spatiale $\underline{\Lambda}$

Entrée : $\underline{\lambda}$

Calcul des termes $a = \lambda_{end}^2 - \lambda_0^2$ et $\underline{\delta} = \mathbf{Q}\,\mathbf{I}_t\,\underline{\lambda}$

Génération de $\tilde{\mathbf{K}}$ et $\underline{\tilde{\mathrm{F}}}$

  Pour chaque élément, chaque point de Gauss i

    Calcul et factorisation (Cholesky) de $\mathbb{Z}$: $\mathbf{L}_i$

    Calcul de $\tilde{\mathbf{K}}$ élémentaire : $\tilde{\mathbf{K}}_i = \omega_i \left(\mathbf{L}_i \backslash \mathbf{B}_i\right)^{\mathrm{T}} \left(\mathbf{L}_i \backslash \mathbf{B}_i\right)$

    Calcul de $\underline{\tilde{\mathrm{F}}}$ élémentaire : $\underline{\tilde{\mathrm{F}}}_i = \omega_i \left(\mathbf{L}_i \backslash \mathbf{B}_i\right)^{\mathrm{T}} \left(\mathbf{L}_i \backslash \underline{\delta}_i\right)$

Assemblage et résolution de $\tilde{\mathbf{K}}\underline{\Lambda}_{\bar{\mathrm{U}}} = \underline{\tilde{\mathrm{F}}} \longrightarrow \underline{\Lambda}_{\bar{\mathrm{U}}}$

Construction des champs solution

  $\underline{\Lambda}_U = \underline{\Lambda}_{\bar{\mathrm{U}}}/a$

  $\underline{\Lambda}_{\boldsymbol{\varepsilon}} = \boldsymbol{\varepsilon}(\underline{\Lambda}_U)$

  Calcul de $\underline{\Lambda}_{\boldsymbol{\sigma}}$ et $\underline{\Lambda}$

    Pour chaque élément, chaque point de Gauss i

      $\underline{\Lambda}_{\boldsymbol{\sigma},i} = \mathbf{L}^{\mathrm{T}} \backslash \Big(\mathbf{L} \backslash (\underline{\Lambda}_{\bar{\varepsilon},i} + \underline{\delta}) \Big)$

      $\underline{\Lambda}_i = \underline{\Lambda}_{\boldsymbol{\varepsilon},i} - \mathbf{K}_e^{-1} \underline{\Lambda}_{\boldsymbol{\sigma},i}$

Sorties : $\underline{\Lambda}$, $\underline{\Lambda}_U$, $\underline{\Lambda}_{\boldsymbol{\varepsilon}}$, $\underline{\Lambda}_{\boldsymbol{\sigma}}$

de la fonctionnelle soit :

Trouver $\lambda \in \mathcal{L}^2(I)$ telle que :

$$\int_{I\times\Omega} \Big[\boldsymbol{\Theta}^-(\Lambda_{\boldsymbol{\sigma}}\lambda^*) - \Lambda\dot{\lambda}^* - Q\Big] : \Big[\Lambda^{\mathrm{D}}_{\boldsymbol{\sigma}}\lambda - \tilde{\boldsymbol{\Theta}}^-(\Lambda\dot{\lambda}) + \tilde{\boldsymbol{\Theta}}^-(Q)\Big]\,\mathrm{d}\Omega\mathrm{d}t = 0, \forall \lambda* \in \mathcal{L}^2(I) \quad (5.35)$$

L'expression (5.35) peut s'arranger sous la forme suivante :

$$\int_I \begin{bmatrix}\dot{\lambda}^* \\ \lambda^*\end{bmatrix}^{\mathrm{T}} \begin{bmatrix}A_{11} & A_{01} \\ A_{10} & A_{00}\end{bmatrix} \begin{bmatrix}\dot{\lambda} \\ \lambda\end{bmatrix} - \begin{bmatrix}\dot{\lambda}^* \\ \lambda^*\end{bmatrix}^{\mathrm{T}} \begin{bmatrix}b_1 \\ b_0\end{bmatrix} + C\,\mathrm{d}t = 0, \forall \lambda^* \in \mathcal{L}^2(I) \quad (5.36)$$

avec C indépendant de λ^* et :

$$\left\{\begin{array}{lcll} A_{11} & = & \int_\Omega \Lambda & : \tilde{\boldsymbol{\Theta}}^-(\Lambda)\ \mathrm{d}\Omega \\ A_{01} & = & -\int_\Omega \Lambda & : \Lambda^{\mathrm{D}}_{\boldsymbol{\sigma}}\ \mathrm{d}\Omega \\ A_{10} & = & -\int_\Omega \boldsymbol{\Theta}^-(\Lambda_{\boldsymbol{\sigma}}) & : \tilde{\boldsymbol{\Theta}}^-(\Lambda)\ \mathrm{d}\Omega = A_{01} \\ A_{00} & = & \int_\Omega \boldsymbol{\Theta}^-(\Lambda_{\boldsymbol{\sigma}}) & : \Lambda^{\mathrm{D}}_{\boldsymbol{\sigma}}\ \mathrm{d}\Omega \\ b_1 & = & -\int_\Omega \Lambda & : \tilde{\boldsymbol{\Theta}}^-(Q)\ \mathrm{d}\Omega \\ b_0 & = & \int_\Omega \boldsymbol{\Theta}^-(\Lambda_{\boldsymbol{\sigma}}) & : \tilde{\boldsymbol{\Theta}}^-(Q)\ \mathrm{d}\Omega = \int_\Omega \Lambda^{\mathrm{D}}_{\boldsymbol{\sigma}} : Q\,\mathrm{d}\Omega \end{array}\right. \quad (5.37)$$

La particularité de l'expression (5.35) est la présence de la dérivée de la fonction test λ^*. Une intégration par parties des composantes liées à $\dot{\lambda}^*$ permet d'exprimer le problème uniquement

selon λ^* avec présence de conditions initiales et finales. Le problème peut se résoudre alors par méthode de tir avec projection ou non [Passieux, 2008], ou par méthodes des différences finies dans le cas de la méthode Lagrange-Galerkin.

Dans le cadre des ces travaux, le schéma considéré est un schéma Galerkin discontinu d'ordre 0 vérifiant les propriétés de conservativité et de consistance des flux. Ce schéma est similaire à l'utilisation d'une méthode des volumes finis décentrée [Di Pietro et Ern, 2012], et s'écrit de la même manière qu'un schéma d'Euler implicite.

Une discrétisation de I sera réalisée au moyen de $n+1$ volumes de contrôle disjoints $K_i =]t_{i-1/2}, t_{i+1/2}[$ formant un maillage $\mathcal{T}$. Chaque volume de contrôle K_i est caractérisé par le point t_i.

Les fonctions λ et $\dot{\lambda}$ sont alors approximées par les fonctions $\check{\lambda} : t \mapsto \sum_{i=0}^{n} \lambda_i \mathbb{1}_{K_i}(t)$ et $\check{\dot{\lambda}} : t \mapsto \sum_{i=0}^{n} \dot{\lambda}_i \mathbb{1}_{K_i}(t)$ avec $\forall i \in [|0,n|], \lambda_i = \lambda(t_i)$, $\dot{\lambda}_i = \dfrac{\lambda(t_i) - \lambda(t_{i-1})}{t_i - t_{i-1}}$ et $\lambda_{-1} = 0$. La figure 5.1 représente cette approximation.

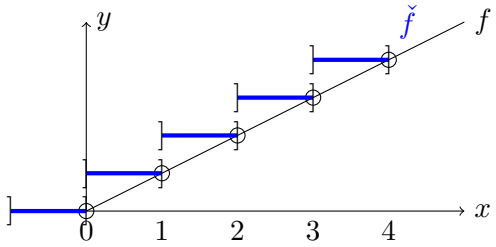


FIGURE 5.1 – Représentation de l'approximation de la fonction temporelle $\check{f}$ d'une fonction f dans le cas de la méthode Galerkin Discontinu d'ordre 0

On approxime donc la solution en simplifiant le problème (5.35) par la recherche de la solution dans l'espace discrétisé $\mathcal{X}$. On note la matrice caractérisant le schéma temporel $\mathbb{T}_i = \begin{bmatrix} 1/dt_i & -1/dt_i \\ 1 & 1 \end{bmatrix}$

Le problème discrétisé consiste à trouver $\underline{\lambda} \in \mathbb{R}^{n+1}$ tel que :

$$\sum_{i=0}^{n} \left(\begin{bmatrix} \lambda_i^* \\ \lambda_{i-1}^* \end{bmatrix}^{\mathrm{T}} \mathbb{A}_i \begin{bmatrix} \lambda_i \\ \lambda_{i-1} \end{bmatrix} - \begin{bmatrix} \lambda_i^* \\ \lambda_{i-1}^* \end{bmatrix}^{\mathrm{T}} \mathbb{B}_i \right) \int_I \mathbb{1}_{K_i}(t)\,\mathrm{dt} = 0, \forall \lambda^* \in \mathcal{X} \tag{5.38}$$

$$\text{avec } \forall i, \begin{cases} \mathbb{A}_i = \mathbb{T}_i^{\mathrm{T}} \begin{bmatrix} A_{11}(t_i) & A_{01}(t_i) \\ A_{10}(t_i) & A_{00}(t_i) \end{bmatrix} \mathbb{T}_i \\ \mathbb{B}_i = \mathbb{T}_i^{\mathrm{T}} \begin{bmatrix} b_1(t_i) \\ b_0(t_i) \end{bmatrix} \end{cases}$$

L'assemblage du système nous donne le système $\mathbf{A}\underline{\lambda} = \underline{b}$ à résoudre. La matrice $\mathbf{A}$ du schéma

est tridiagonale et symétrique définie positive. La résolution peut donc exploiter la factorisation de Cholesky de $\mathbf{A}$ ou les techniques adaptées aux matrices tridiagonales.

5.1.4 Formulation de l'étape de mise à jour

Il n'est pas indispensable de générer un nouveau mode à chaque itération de convergence. En effet, la base spatiale générée au cours des itérations précédentes peut être suffisamment riche pour corriger les champs globaux à partir des champs locaux. Ainsi, une étape intermédiaire a été mise en place afin de limiter la génération coûteuse d'un nouveau mode.

Soit la déformation plastique générée à l'étape globale précédente $\boldsymbol{\varepsilon_p} = \sum_{i=1}^{m} \Lambda_i \tilde{\lambda}_i$, il est possible de calculer de nouvelles fonctions temporelles λ_i minimisant l'erreur en direction de recherche. Ces fonctions sont alors solution du problème suivant :

Trouver $(\lambda_i)_{i=1..n}$ telles que :

$$(\lambda_i)_{i=1..n} = \underset{\Delta\boldsymbol{\varepsilon_p}\in\mathcal{W}^0}{\operatorname{argmin}} \left\| \boldsymbol{\Theta}^-(\boldsymbol{C}(\dot{\boldsymbol{\varepsilon}}_p) - \hat{\boldsymbol{\sigma}}) - (\dot{\boldsymbol{\varepsilon}}_p - \hat{\dot{\boldsymbol{\varepsilon}}}_p) \right\|_{\mathbf{M}} \tag{5.39}$$

En suivant le même raisonnement qu'en section précedente avec la fonction test $\boldsymbol{\varepsilon_p}^*(M,t) = \sum_{i=1}^{m} \Lambda_i^*(M)\lambda_i^*(t)$, on obtient le problème suivant à résoudre :

$$\sum_{i=0}^{m}\sum_{j=0}^{m} \int_{I\times\Omega} \left[\boldsymbol{\Theta}^-(\boldsymbol{C}(\Lambda_i)\lambda_i^*) - \Lambda_i\dot{\lambda}_i^* + Q\right] : \left[\boldsymbol{C}_{\mathbf{D}}(\Lambda_j)\lambda_j - \tilde{\boldsymbol{\Theta}}^-(\Lambda_j\dot{\lambda}_j) + \tilde{\boldsymbol{\Theta}}^-(Q)\right] \mathrm{d}\Omega\mathrm{d}t = 0 \tag{5.40}$$

avec $Q = -\boldsymbol{\Theta}^-(\hat{\boldsymbol{\sigma}}) + \hat{\dot{\boldsymbol{\varepsilon}}}_p$

Le problème est semblable au problème à résoudre pour la génération d'une fonction temporelle. Le problème se discrétise de la même façon sous la forme suivante.

Trouver $(\underline{\lambda}_j) \in \mathcal{X}^m$ tel que :

$$\sum_{j=1}^{m}\sum_{k=1}^{m}\sum_{i=0}^{n} \left(\begin{bmatrix} \lambda_{j,i}^* \\ \lambda_{j,i-1}^* \end{bmatrix}^{\mathrm{T}} \mathbb{A}_i^{(j,k)} \begin{bmatrix} \lambda_{k,i} \\ \lambda_{k,i-1} \end{bmatrix} + \begin{bmatrix} \lambda_{j,i}^* \\ \lambda_{j,i-1}^* \end{bmatrix}^{\mathrm{T}} \mathbb{B}_i^{(j)} \right) \int_I \mathbb{1}_{K_i}(t)\,\mathrm{d}t = 0, \forall j, \forall \lambda_j^* \in \mathcal{X} \tag{5.41}$$

avec $\forall(i,j,k)$

$$\begin{cases} \mathbb{A}_i^{(j,k)} = \mathbb{T}_i^{\mathrm{T}} \begin{bmatrix} A_{11}^{(j,k)}(t_i) & A_{01}^{(j,k)}(t_i) \\ A_{10}^{(j,k)}(t_i) & A_{00}^{(j,k)}(t_i) \end{bmatrix} \mathbb{T}_i \\ \mathbb{B}_i^{(j)} = \mathbb{T}_i^{\mathrm{T}} \begin{bmatrix} b_1^{(j)}(t_i) \\ b_0^{(j)}(t_i) \end{bmatrix} \end{cases} \quad \text{et } \forall(j,k) \begin{cases} A_{11}^{(j,k)} &= \int_\Omega \Lambda_j : \tilde{\boldsymbol{\Theta}}^-(\Lambda_k)\,\mathrm{d}\Omega \\ A_{01}^{(j,k)} &= -\int_\Omega \Lambda_j : \Lambda_{\boldsymbol{\sigma},k}^{\mathrm{D}}\,\mathrm{d}\Omega \\ A_{00}^{(j,k)} &= \int_\Omega \boldsymbol{\Theta}^-(\Lambda_{\boldsymbol{\sigma},j}) : \Lambda_{\boldsymbol{\sigma},k}^{\mathrm{D}}\,\mathrm{d}\Omega \\ b_1^{(j)} &= -\int_\Omega \Lambda_j : \tilde{\boldsymbol{\Theta}}^-(Q)\,\mathrm{d}\Omega \\ b_0^{(j)} &= \int_\Omega \Lambda_{\boldsymbol{\sigma}}^{\mathrm{D}} : Q\mathrm{d}\Omega \end{cases}$$

L'assemblage du système nous donne le système $\mathbf{A}\underline{\lambda} = \underline{b}$ à résoudre. La matrice $\mathbf{A}$ du schéma est symétrique définie positive et l'assemblage du problème se fait pas de temps par pas de temps pour avoir une structure $(2m+1)$-diagonale.

La résolution se fait également par factorisation de Cholesky de $\mathbf{A}$. Étant donné que les fonctions spatiales ne sont plus modifiées après leur génération, il est possible d'avoir uniquement à mettre à jour la factorisation à chaque nouveau mode spatial.

Un nouveau mode est ajouté dans le cas où

$$\max_{j\in[|1,m|]} \frac{\left\langle \lambda_j - \tilde{\lambda}_j, \lambda_j - \tilde{\lambda}_j \right\rangle}{1/2(\langle \lambda_j, \lambda_j \rangle + \left\langle \tilde{\lambda_j}, \tilde{\lambda_j} \right\rangle)} \geq \nu_U \tag{5.42}$$

avec $\nu_U = 1\%$.

5.1.5 Critère d'erreur

De multiples critères d'erreur peuvent être employés dans ce cas étant donné la génération de champs spatio-temporels complets [Passieux, 2008]. Le choix couramment effectué consiste à choisir un critère de stagnation des couples $(\boldsymbol{\varepsilon}, \boldsymbol{\sigma})$ et $(\hat{\boldsymbol{\varepsilon}}, \hat{\boldsymbol{\sigma}})$:

$$\nu_{Latin} : [(\boldsymbol{\varepsilon}, \boldsymbol{\sigma}), (\hat{\boldsymbol{\varepsilon}}, \hat{\boldsymbol{\sigma}})] \mapsto \frac{(\hat{\boldsymbol{\varepsilon}} - \boldsymbol{\varepsilon}, \hat{\boldsymbol{\sigma}} - \boldsymbol{\sigma})_{\mathbf{K}_e}}{1/2(\hat{\boldsymbol{\varepsilon}}, \hat{\boldsymbol{\sigma}})_{\mathbf{K}_e} + 1/2(\boldsymbol{\varepsilon}, \boldsymbol{\sigma})_{\mathbf{K}_e}} \tag{5.43}$$

avec $(\bullet, \bullet)_{\mathbf{K}_e} : (\boldsymbol{\varepsilon}, \boldsymbol{\sigma}) \mapsto \frac{1}{2}\int_{I\times\Omega} \boldsymbol{\varepsilon} : \mathbf{K}_e\boldsymbol{\varepsilon} \, \mathrm{d}\Omega\mathrm{d}t + \frac{1}{2}\int_{I\times\Omega} \boldsymbol{\sigma} : \mathbf{K}_e^{-1}\boldsymbol{\sigma} \, \mathrm{d}\Omega\mathrm{d}t$

5.1.6 Analyse de l'approche à variables internes

La section actuelle a présenté les développements techniques nécessaires à l'implémentation de l'approche à variables internes. La formulation de l'étape globale nécessite d'assembler et résoudre un problème pour la génération d'une nouvelle fonction spatiale et une nouvelle fonction temporelle, ce qui en fait une méthode coûteuse en temps de calcul. Cette méthode a malgré tout été implémentée pour la réalisation des premiers cas-tests. Dans le cadre du stage de Master de Ronan Scanff en 2018, l'implémentation de la formulation fonctionnelle a également été réalisée. Celle-ci a présenté une vitesse de convergence intéressante pour une algorithmie réduite en comparaison à l'approche à variables internes.

5.2 Approche fonctionnelle de la LATIN

Cette approche de la formulation LATIN est la version historique. La convergence de celle-ci n'est pas prouvée contrairement à la formulation précédente. Ici, cette approche sera présentée dans sa forme la plus simple, à savoir avec l'utilisation d'une direction de recherche constante.

Dans cette approche, les variables mises en jeu pour le calcul sont la déformation et la contrainte $(\boldsymbol{\varepsilon}, \boldsymbol{\sigma})$ à l'instar des codes de calculs commerciaux.

L'étape locale est conservée et est donc strictement la même que l'étape locale présentée en section 5.1.1. La différence se fait au niveau de l'étape globale.

5.2.1 Formulation de l'étape globale

On rappelle que l'étape globale consiste à trouver une correction $\Delta\boldsymbol{\varepsilon}$ sous une forme à variable séparée telle que l'admissibilité statique et cinématique à zéro soient vérifiées, ainsi que l'équation d'équilibre.

Le lien avec la variété Γ est fait à partir de la direction de recherche :

$$(\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}}) - \boldsymbol{\Theta}^-(\boldsymbol{\varepsilon} - \hat{\boldsymbol{\varepsilon}}) = \mathbf{0} \tag{5.44}$$

L'équation d'équilibre est telle que $\forall \mathrm{U}^* \in \mathcal{U}^*$,

$$\int_{I\times\Omega} \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\mathrm{U}^*)\, \mathrm{d}\Omega\mathrm{d}t = \int_{I\times\Omega} \underline{\mathrm{f}}_d \cdot \mathrm{U}^*\, \mathrm{d}\Omega\mathrm{d}t + \int_{I\times\partial_2\Omega} \underline{\mathrm{F}}_d \cdot \mathrm{U}^*\, \mathrm{d}s\mathrm{d}t = F_{ext} \tag{5.45}$$

On insère la direction de recherche (5.44) dans l'équation d'équilibre (5.45) pour obtenir l'équation suivante :

$$\forall \mathrm{U}^* \in \mathcal{U}^*,\ \int_{I\times\Omega} \boldsymbol{\Theta}^-(\boldsymbol{\varepsilon}) : \boldsymbol{\varepsilon}(\mathrm{U}^*)\, \mathrm{d}\Omega\mathrm{d}t = \mathcal{Q}(\mathrm{U}^*) \tag{5.46}$$

avec le résidu $\mathcal{R}(\mathrm{U}^*) = \int_{I\times\Omega} \hat{\boldsymbol{\sigma}} : \boldsymbol{\varepsilon}(\mathrm{U}^*)\, \mathrm{d}\Omega\mathrm{d}t - \int_{I\times\Omega} \underline{\mathrm{f}}_d\, \mathrm{U}^*\, \mathrm{d}\Omega\mathrm{d}t - \int_{I\times\partial_2\Omega} \underline{\mathrm{F}}_d\, \mathrm{U}^*\, \mathrm{d}s\mathrm{d}t$
et $\mathcal{Q}(\mathrm{U}^*) = \mathcal{R}(\mathrm{U}^*) - \int_{I\times\Omega} \boldsymbol{\Theta}^-(\tilde{\boldsymbol{\varepsilon}} - \hat{\boldsymbol{\varepsilon}}) : \boldsymbol{\varepsilon}(\mathrm{U}^*)\, \mathrm{d}\Omega\mathrm{d}t$.

La recherche de la solution sous forme PGD impose de relâcher la condition sur la direction de recherche pour éviter un problème sur-contraint. Par minimisation de la semi-norme associée à $\tilde{\boldsymbol{\Theta}}^-$, on obtient les deux problèmes couplés suivant :

$\forall (\Lambda^*, \lambda^*) \in \mathcal{U}^0 \times \mathcal{I}$,

$$\int_{I\times\Omega} \left[\boldsymbol{C}(\Lambda^*)\lambda - \boldsymbol{\Theta}^-(\Lambda^*\lambda) + Q\right] : \tilde{\boldsymbol{\Theta}}^- \left[\boldsymbol{C}(\Lambda^*)\lambda - \boldsymbol{\Theta}^-(\Lambda\lambda) + Q\right] \mathrm{d}\Omega\mathrm{d}t = 0 \tag{5.47}$$

$$\int_{I\times\Omega} \left[\boldsymbol{C}(\Lambda)\lambda^* - \boldsymbol{\Theta}^-(\Lambda\lambda^*) + Q\right] : \tilde{\boldsymbol{\Theta}}^- \left[\boldsymbol{C}(\Lambda)\lambda - \boldsymbol{\Theta}^-(\Lambda\lambda) + Q\right] \mathrm{d}\Omega\mathrm{d}t = 0 \tag{5.48}$$

La résolution du problème temporel (5.48) se résout plus simplement que dans le cas à variables internes grâce à l'absence de la dérivée temporelle de λ.

On note $\langle u, v\rangle_S = \int_\Omega u : \tilde{\boldsymbol{\Theta}}^-(v)\, \mathrm{d}\Omega$ $\langle u, v\rangle_T = \int_I u : \tilde{\boldsymbol{\Theta}}^-(v)\, \mathrm{d}t$ et on rappelle que $\boldsymbol{\Theta}^-$ et $\tilde{\boldsymbol{\Theta}}^-$ soit des opérateurs linéaires et qu'on a notamment $\tilde{\boldsymbol{\Theta}}^-(\boldsymbol{C}(\Lambda)\lambda) = \tilde{\boldsymbol{\Theta}}^-(\boldsymbol{C}(\Lambda))\lambda$.

5.2.2 Choix de la direction de recherche élastique

Une construction basée sur l'opérateur tangent peut être choisie comme direction de recherche. Néanmoins, ce choix nécessite une nouvelle fois de nombreuses factorisations d'opérateurs.

Un second choix serait de considérer l'opérateur de Hooke $\mathbf{\Theta}^- = \mathbf{K}_e$. Celui-ci est indépendant du temps et n'évolue pas au cours des itérations ce qui permet d'orthonormaliser les modes spatiaux vis-à-vis de cette norme. De plus, dans le cas d'étude de cette thèse consistant à faire de nombreuses résolutions en modifiant les paramètres, ce choix peut permettre de faire une unique factorisation pour la résolution de l'ensemble des problèmes.

Dans ce cas de figure, la résolution des problèmes (5.47), (5.48) devient beaucoup plus aisée, notamment par le fait que les modes spatiaux sont orthonormalisés selon la semi-norme spatiale : on a alors $\forall i, j \in [\![1, m]\!], \langle \Lambda_i, \Lambda_j \rangle_S = \delta_{ij}$.

La mise à jour des modes temporels correspond alors à la projection du second membre dans l'espace associé à chaque mode spatial et devient donc m problèmes découplés $\forall i, \lambda_i = \int_\Omega \mathrm{Q}_u : \Lambda_i \, \mathrm{d}\Omega$ avec la discrétisation du second membre associé à la mise à jour :

$$\mathrm{Q}_u = \mathbf{B}^{\mathrm{T}} \mathbf{\Theta}^- (\boldsymbol{\varepsilon}_{CA} - \hat{\boldsymbol{\varepsilon}}) - (\underbrace{\mathbf{B}^{\mathrm{T}} \mathbf{I}_{\mathbf{g}} \hat{\boldsymbol{\sigma}}}_{\underline{F}_{int}} - \underline{F}_{ext})$$

On note également le second membre discrétisé, associé à la recherche d'un nouveau mode $\mathrm{Q} = \mathbf{B}^{\mathrm{T}} \mathbf{\Theta}^- (\tilde{\boldsymbol{\varepsilon}} - \hat{\boldsymbol{\varepsilon}}) - (\underline{\tilde{F}}_{int} - \underline{F}_{ext})$. L'algorithme 7 détaille les étapes avec $\mathbf{K}$ la matrice de rigidité globale. Cette matrice étant symétrique définie positive et étant régulièrement utilisée, il est indispensable de réaliser une factorisation de Cholesky sur celle-ci dans le cas de problèmes ayant moins d'un million de degrés de liberté, ou d'utiliser un algorithme de gradient conjugué préconditionné.

Algorithm 7: Méthode LATIN fonctionnelle — Direction de recherche élastique

Initialisation : Approximation initiale $\mathbf{s}_0 \in A_d$
Boucle de convergence - Tant que $||\nu|| > tol$
- **Etape Locale** $\longrightarrow$ Algorithme 5
- **Etape *Update*** $\longrightarrow \forall i, \lambda_i \longleftarrow \int_\Omega \mathrm{Q}_u : \Lambda_i \, \mathrm{d}\Omega$
- **Etape Globale**
  - Recherche de la fonction spatiale : $\Lambda \longleftarrow \langle \lambda, \lambda \rangle_{\mathrm{T}}^{-1} \left(\mathbf{K} \setminus \int_I \mathrm{Q} \lambda \, \mathrm{dt} \right)$
  - Recherche de la fonction temporelle : $\lambda \longleftarrow \langle \Lambda, \Lambda \rangle_{\mathrm{S}}^{-1} \left(\int_\Omega \mathrm{Q} : \mathbf{K} \Lambda \, \mathrm{d}\Omega \right)$
- **Ajout du nouveau mode** $\lambda\Lambda$
- **& Orthonormalisation de la base de modes spatiaux suivant K**
- **Calcul de l'erreur LATIN** ν

La contrainte est calculée à l'étape locale. Il est également possible de calculer un champ contrainte vérifiant la loi de Hooke à partir de la déformation à l'étape globale $\boldsymbol{\sigma} = \mathbf{K}_e(\boldsymbol{\varepsilon} - \hat{\boldsymbol{\varepsilon}}_p)$.

5.3 Implémentation parallèle de la méthode

La méthode de résolution LATIN a déjà été mise en œuvre via de multiples démonstrateurs comme VISCOlatin (FORTRAN) [Cognard, 1989], LMTpp (C++) [Relun *et al.*, 2013], ou encore ROMlab (Matlab). Afin de pérenniser les développements réalisés, l'implémentation des méthodes a été faite à partir du démonstrateur ROMlab sous Matlab, initié par David Néron sur la base d'esbroufe3d de Jean-Charles Passieux, et développé par Matthieu Vitse pour la résolution multi-paramétrique d'un problème de fatigue d'une structure en béton armé [Vitse *et al.*, 2014]. Une attention particulière a été portée au temps de calcul dans le cadre de cette thèse. Afin de permettre la mise en œuvre de calculs de taille industrielle, le démonstrateur a été repensé afin de tirer parti des architectures multithreads actuelles.

5.3.1 Tensorisation des champs et des opérateurs

Usuellement en mécanique des structures, les champs spatio-temporels sont déployés sous la forme de vecteurs et les opérateurs sous la forme de matrices 6×6. Un autre formalisme dit « Tensor Mechanics » consiste à écrire les champs sous la forme de tenseurs n-dimensionnels. Ainsi, le champ de déformation $\boldsymbol{\varepsilon}(M)$ peut être discrétisé et stocké sous la forme d'un tenseur $3D$ $\boldsymbol{\varepsilon}_{cge}$.

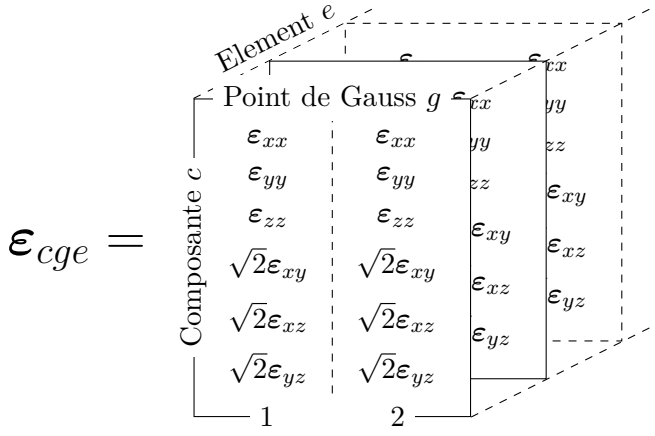


FIGURE 5.2 – Schématisation d'un tenseur

L'intégralité des champs et des opérateurs sont écrits sous cette forme.

5.3.2 Parallélisation des calculs

Cette tensorisation va permettre d'exploiter une technique d'accélération d'opérations nommée expansion implicite ou broadcasting, présente dans R, Matlab, et Python/Numpy. Cette technique consiste à autoriser les opérations élémentaires sur des matrices de tailles différentes, mais « compatibles ». Les dimensions singletons d'une matrice $\mathbf{A}$ sont implicitement

étendues lorsqu'elles sont multipliées par une matrice **B** dont la dimension en question est de taille supérieure à 1. Cette technique permet aux logiciels de calcul d'exploiter au mieux les nouveaux jeux d'instructions processeurs et de paralléliser les opérations.

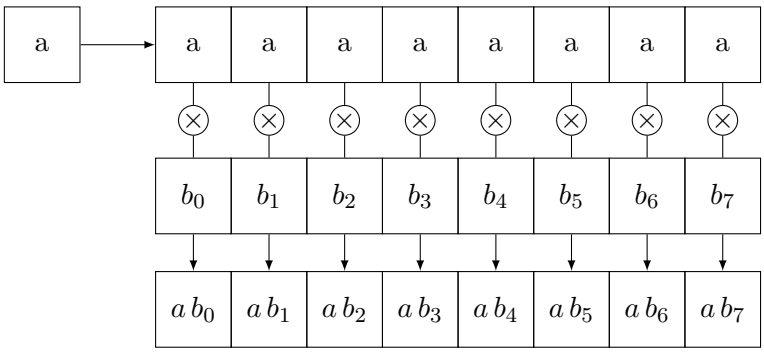


FIGURE 5.3 – Parallélisation des opérations lors d'une expansion implicite : exemple avec $c_i = a\,b_i$

Cette technique ouvre la porte à une parallélisation rapide à implémenter. Pour faciliter les opérations tout en garantissant une exécution rapide, les opérations de multiplications tensorielles (*inner product*, *outer product*) seront demandées en écrivant le calcul à effectuer en notation d'Einstein, à la manière de la fonction einsum dans Python/Numpy.

Ainsi, la multiplication matricielle entre les matrices $\mathbf{C}_{ij}$ et $\boldsymbol{\varepsilon}_{jget}$ pour tout pas de temps t, tout élément e et tout point de Gauss g, s'écrit $\boldsymbol{\sigma}_{iget} = C_{ij}\boldsymbol{\varepsilon}_{jget}$ et se calcule en parallèle avec l'opération einsum('ij,jget->iget',C,$\boldsymbol{\varepsilon}$).

$$A_{gc} = \omega_g \boldsymbol{\varepsilon}_c = \begin{bmatrix} \omega_1 & \omega_2 \\ \omega_1 & \omega_2 \\ \omega_1 & \omega_2 \\ \omega_1 & \omega_2 \\ \omega_1 & \omega_2 \\ \omega_1 & \omega_2 \end{bmatrix} .* \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xx} \\ \varepsilon_{yy} & \varepsilon_{yy} \\ \varepsilon_{zz} & \varepsilon_{zz} \\ \varepsilon_{xy} & \varepsilon_{xy} \\ \varepsilon_{xz} & \varepsilon_{xz} \\ \varepsilon_{yz} & \varepsilon_{yz} \end{bmatrix} = \begin{bmatrix} \omega_1\varepsilon_{xx} & \omega_2\varepsilon_{xx} \\ \omega_1\varepsilon_{yy} & \omega_2\varepsilon_{yy} \\ \omega_1\varepsilon_{zz} & \omega_2\varepsilon_{zz} \\ \omega_1\varepsilon_{xy} & \omega_2\varepsilon_{xy} \\ \omega_1\varepsilon_{xz} & \omega_2\varepsilon_{xz} \\ \omega_1\varepsilon_{yz} & \omega_2\varepsilon_{yz} \end{bmatrix}$$

FIGURE 5.4 – Schématisation des opérations lors d'une sommation d'Einstein. Les éléments en pointillé ne sont pas stockés en mémoire, mais répliqués par le jeu d'instructions processeur quand le calcul est demandé.

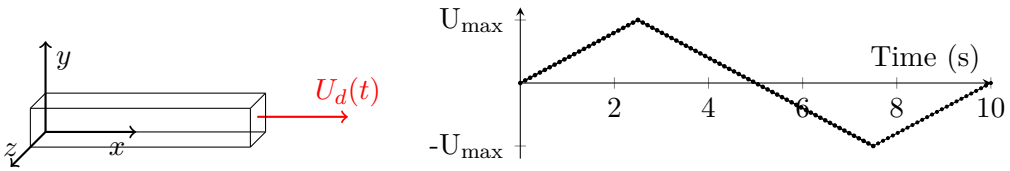
Cette mise en œuvre parallèle des calculs tensoriels au moyen de l'utilisation de la notation d'Einstein est d'ailleurs d'actualité en Machine Learning [Vasilache *et al.*, 2018] et est utilisé dans le module « Tensor Mechanics » du solveur multiphysique MOOSE [Gaston *et al.*, 2009].

5.4 Mise en œuvre de la méthode sur trois cas-tests

Pour tester la méthode et les deux approches proposées, trois problèmes seront testés sur une configuration Intel® Xeon® E5-2630 v4 @ 2.20GHz à 20 cœurs, 192 Go de RAM et la version MATLAB R2018b.

5.4.1 Cas-test 1 : Barre en traction-compression

Le premier est un exemple simple de poutre élasto-viscoplastique (figure 5.5a) sur laquelle des conditions de symétrie sont considérées sur les plans $x = 0$, $y = 0$ et $z = 0$. Les dimensions de la poutre sont de $100 \times 10 \times 10$ mm suivant (x, y, z). Le déplacement $U_d(t)$ suivant l'axe x est imposé sur le coté $x = 100$mm. L'évolution temporelle de $U_d(t)$ est indiqué sur la figure 5.5b avec $U_{max} = 1$ mm soit une déformation maximale $\boldsymbol{\varepsilon}_{zz} = 0.01$. Le matériau employé est un acier 316L à 800° C ayant les propriétés indiqués sur le tableau 5.1. La poutre est maillée avec des éléments tétraédriques d'ordre 1 et comporte 29631 degrés de liberté.



(a) Schéma du cas-test (b) Évolution temporelle du déplacement imposé $U_d(t)$

FIGURE 5.5 – Présentation du cas-test 1

E	ν	σ_y	n	K	C	γ	R_∞	b
137.6 GPa	0.3	8 MPa	14	150 MPa.$s^{1/n}$	$24.8\,10^3$	300	80	10

TABLE 5.1 – Coefficients associés à la loi élasto-viscoplastique employée

L'approche à variables internes avec direction de recherche tangente, et l'approche fonctionnelle avec direction de recherche élastique sont testées sur cet exemple.

La résolution de ce problème très simple génère un unique mode spatial et 5 ou 6 mises à jour de la fonction temporelle. La convergence est plus rapide avec l'approche à variables internes, mais pour un coût en temps de calcul beaucoup plus important du fait de la dépendance temporelle et spatiale de l'opérateur tangent.

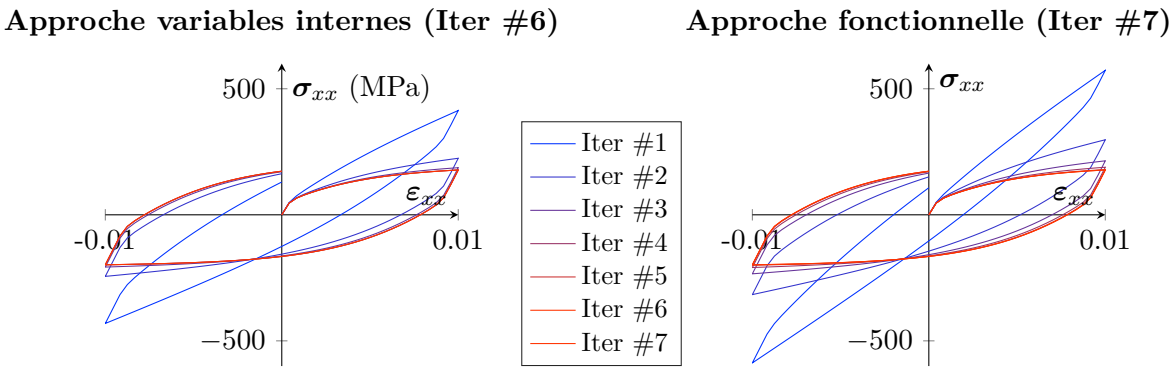


FIGURE 5.6 – Courbe contrainte-déformation pour un élément

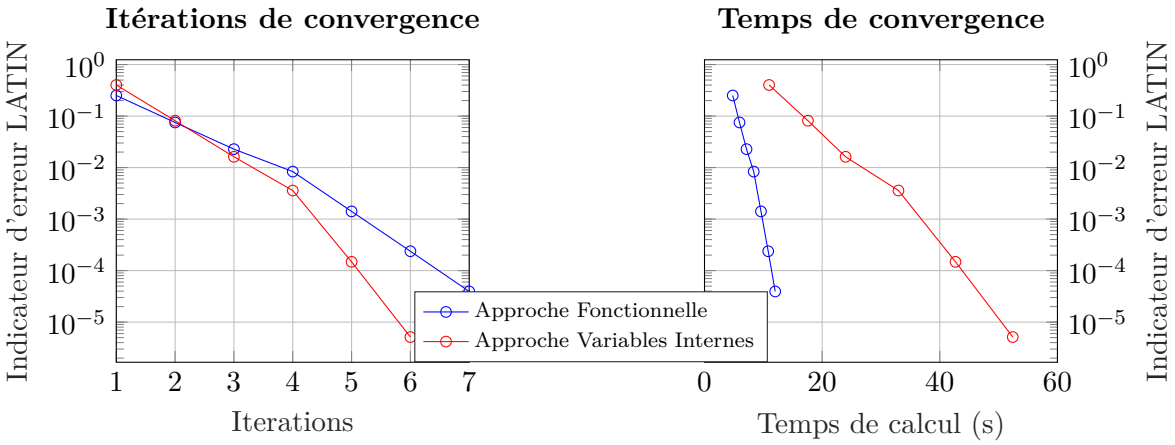
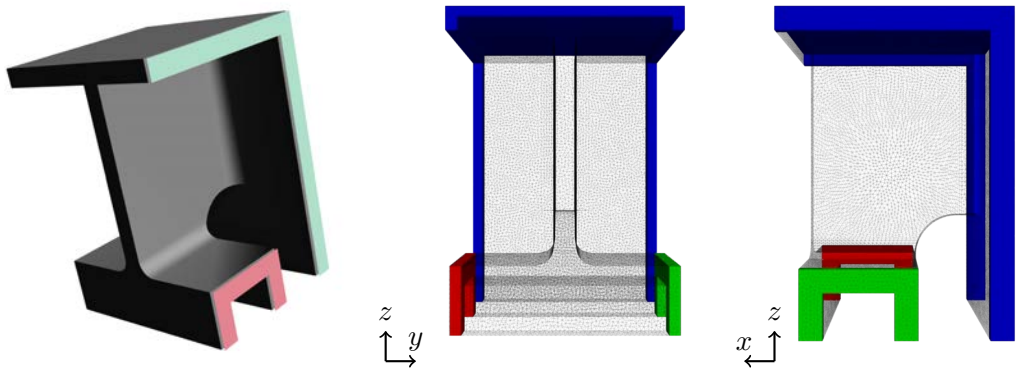


FIGURE 5.7 – Vitesse de convergence de l'algorithme

5.4.2 Cas-test 2 : Pièce d'amortissement

Le second problème est détaillé dans 7.3. La pièce est encastrée sur les côtés gauche et droit (parties bleues sur la figure 5.8b). Le nez de la turbine est visible en bas à gauche de la figure 5.8b. Celui-ci subit un déplacement imposé suivant la direction y sur les faces avant (en vert et rouge sur la figure). Il s'agit également d'un acier 316L à 800°C. Ce déplacement a lieu dans le sens $y+$, puis dans le sens $y-$ comme indiqué sur la figure 5.5b, avec $U_{max} = 0.1$ mm.

Les résultats de la figure 5.9 montrent toute la difficulté à exploiter pleinement l'approche à variables internes pour des problèmes viscoplastiques « simples ». Sans simplification de la direction de recherche, l'espace mémoire nécessaire pour le calcul et les temps de calcul sont prohibitifs en comparaison de l'approche fonctionnelle. L'utilisation de méthodes d'hyper-réduction comme la méthode des points de référence [Capaldo *et al.*, 2017] est indispensable pour permettre la résolution du problème en un temps raisonné avec l'approche à variables internes.



(a) Vue 3D (b) Vue selon l'axe x (c) Vue selon l'axe y

FIGURE 5.8 – Vues du cas-test 2. La pièce est encastrée sur les côtés gauche et droit (parties bleues) et les parties rouge et verte sont chargées comme indiqué sur la figure 7.6

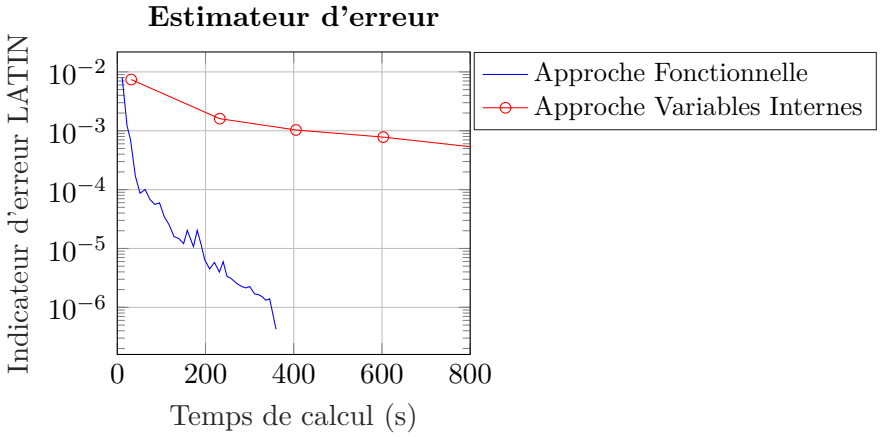


FIGURE 5.9 – Vitesse de convergence de l'algorithme

5.4.3 Cas-test 3 : Aube de turbine

Le dernier cas-test est une aube de turbine d'avion inspirée de celle présente sur le moteur M88 et détaillée dans 7.4. L'aube est encastrée sur ses zones de contact de pied et subit un chargement centrifuge maximum de 15000 tr/min et un chargement latéral en tête d'aube comme indiqué sur la figure 7.10. Le maillage considéré est un maillage tétraédrique quadratique TET10 de 1.7 MDOFs et sera effectué uniquement avec l'approche fonctionnelle.

La figure 5.10 présente l'analyse des résultats obtenus. On constate qu'avec les améliorations apportées, une bonne estimation avec $\nu_{LATIN} = 10^{-2}$ est obtenue en 47 minutes, et que le résultat convergé est obtenu en 1h27mn. Les figures 5.12 et 5.13 présentent eux les premiers modes de déplacement obtenus sous la forme PGD par l'algorithme, et les modes POD obtenus par un algorithme SVD sur la solution convergée. La visualisation des champs a été réalisée par un script de transfert utilisant XDMF3 et HDF5, permettant un rendu sous Paraview.

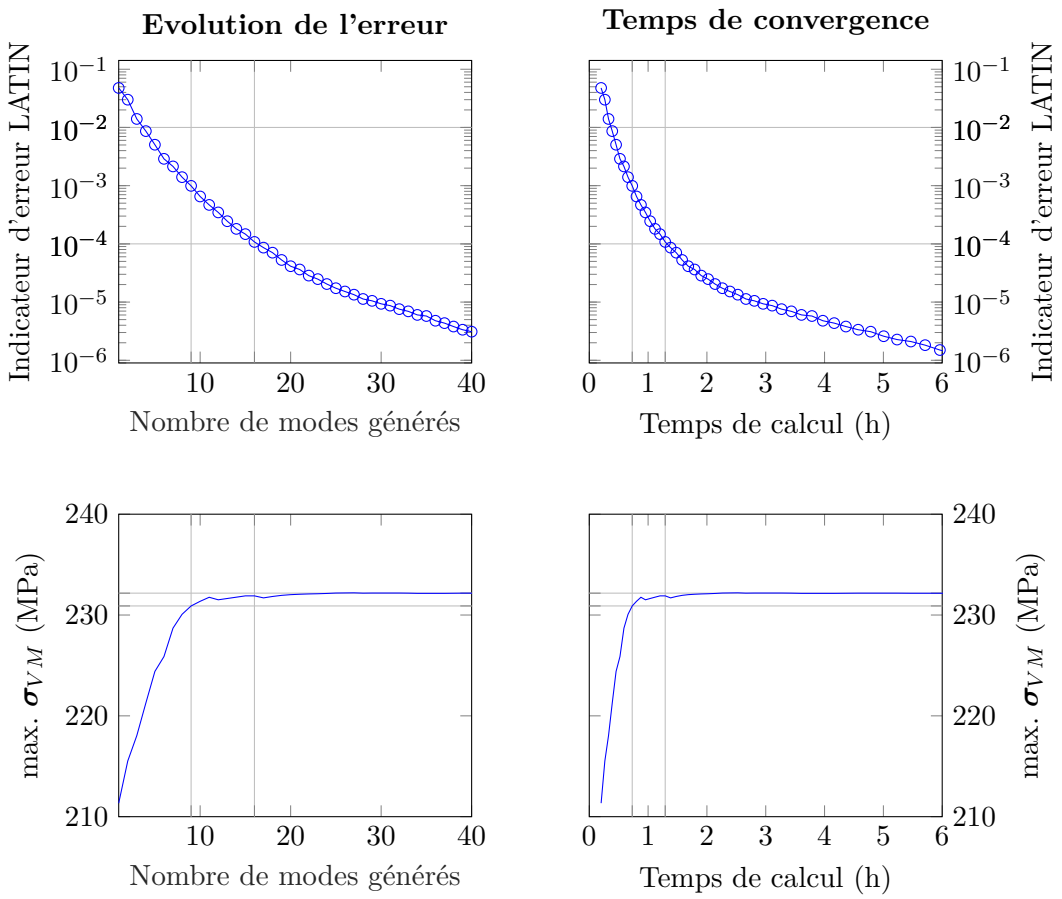
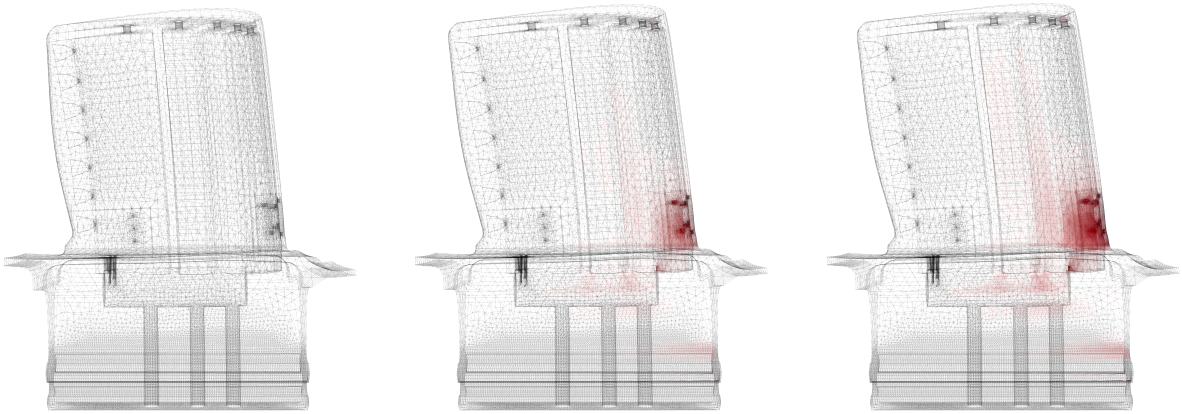


FIGURE 5.10 – Vitesse de convergence



(a) Initialisation élastique (Temps : 5')

(b) Solution approchée $\nu = 10^{-2}$ (Temps : 47')

(c) Solution convergée $\nu = 10^{-4}$ (Temps : 1h27)

FIGURE 5.11 – Taux de déformation plastique à la fin de la zone de chargement maximal

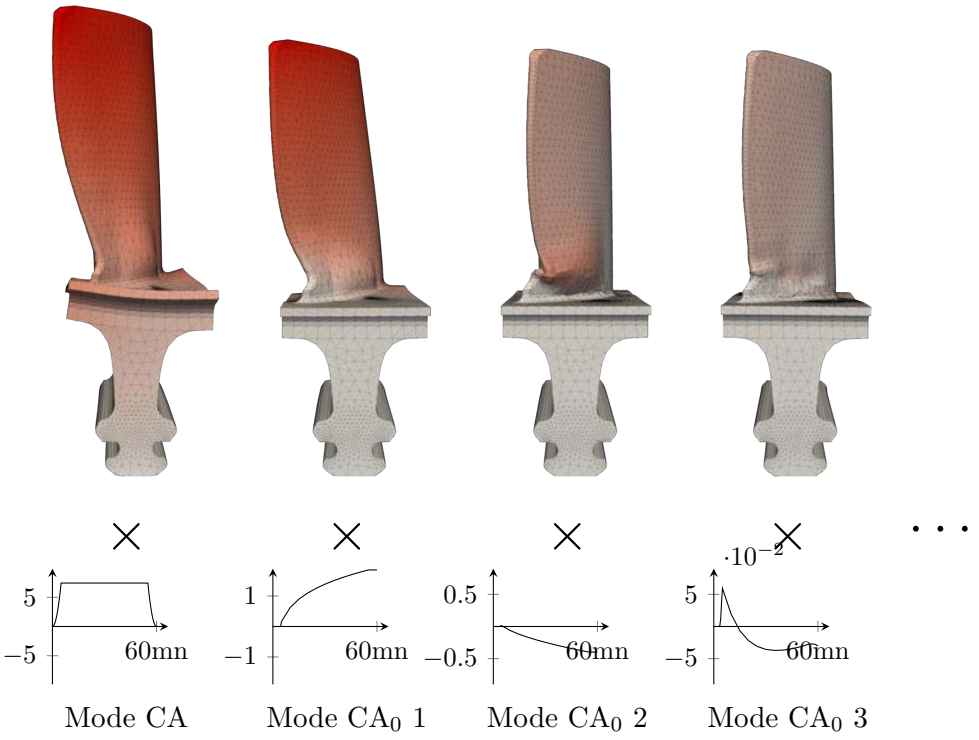


FIGURE 5.12 – Premiers modes PGD générés par le solveur LATIN-PGD

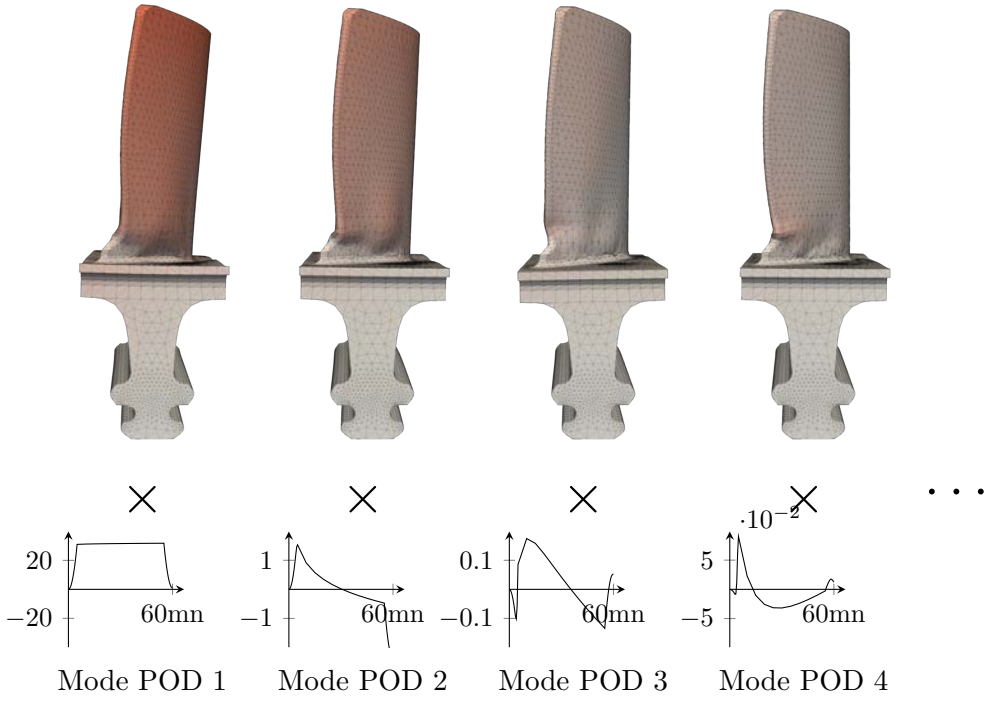


FIGURE 5.13 – Premiers modes POD associées à la solution convergée

Chapitre 6

Mise en œuvre d'une stratégie multiparamétrique de réexploitation des modes

Sommaire

6.1 Méthodes d'approximation du champ initial 86
6.1.1 Utilisation du champ le plus proche 86
6.1.2 Interpolation de la base spatiale dans la variété Grassmannienne 87
6.1.3 Interpolation des champs et de la base spatiale 88
6.2 Gain apporté par la méthode 90

6.1 Méthodes d'approximation du champ initial

Une des avantages déjà évoqué en section 1.3 est la possibilité d'initialiser la résolution du problème non-linéaire par une approximation de la solution. On se place dans le cadre de la LATIN fonctionnelle. La variable étudiée dans la suite sera le déplacement U. On suppose que $\mathrm{U}_1, \mathrm{U}_2, \ldots, \mathrm{U}_{j-1}$ ont déjà été calculés à partir des problèmes paramétrés par $\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_{j-1}$. On cherche la solution du problème paramétré par $\underline{x}_j$.

Deux éléments sont importants dans l'initialisation du calcul :

— la base de modes spatiaux CA_0 sur laquelle projeter la correction plastique. La pertinence de cette base permettera de limiter le nombre de nouveaux modes générés.

— le champ d'initialisation interpolé à partir des champs précedents. La justesse de ce champ va permettre d'obtenir à l'étape locale une solution dans Γ proche de la solution $\mathbf{A}_{\mathrm{d}j} \cap \Gamma$, ce qui limitera le nombre d'itérations.

6.1.1 Utilisation du champ le plus proche

Il s'agit de la méthode la plus simple mais présentant déjà des gains de temps estimables. Il s'agit d'initialiser la résolution à partir de la solution déjà calculée la plus proche dans

l'espace paramétrique au sens de la distance euclidienne. Dans ce cadre, il n'est pas nécessaire d'effectuer d'opérations supplémentaires sur la solution ou sur la base d'initialisation.

6.1.2 Interpolation de la base spatiale dans la variété Grassmannienne

Amsallem et Farhat [Amsallem et Farhat, 2008] ont montré que le point important dans le cas de l'interpolation de bases POD est le sous-espace vectoriel qu'engendre la base interpolée. L'ensemble des sous-espaces vectoriels de dimension fixée n d'un espace vectoriel de dimension $N > n$ a la structure d'une variété différentielle appelée variété Grassmannienne $\mathcal{G}(n, N)$ définissant une géodésique. Dans celle-ci, chaque base $\mathcal{S} \in \mathcal{G}(n, N)$ est associée à une base représentée par la matrice $\Psi \in \mathbb{R}^{N \times n}$ et le point dans l'espace paramétrique $\mu \in \mathcal{D}$. Pour chaque base $\mathcal{S}_0$, il existe un espace tangent associé $\mathcal{T}_{\mathcal{S}_0}$ pour lequel les méthodes d'interpolation usuelles sont possibles. Tout point de l'espace tangent $\mathbf{X} \in \mathcal{T}_{\mathcal{S}_0}$ représenté par la matrice $\mathbf{H} \in \mathbb{R}^{N \times n}$ peut être projeté sur $\mathcal{G}(n, N)$ associée à $\Psi \in \mathbb{R}^{N \times n}$, au moyen de l'application exponentielle géodésique $Log_{\mathcal{S}_0} : \mathbf{X} \in \mathcal{T}_{\mathcal{S}_0} \mapsto \mathcal{S} \in \mathcal{G}(n, N)$ de telle façon que :

$$\begin{aligned} &\frac{\mathbf{H}}{||\Psi_0||} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad \text{(SVD fine)} \\ &\Psi = ||\Psi_0|| \left[\frac{\Psi_0}{||\Psi_0||} \mathbf{V} \cos \Sigma + \mathbf{U} \sin \mathbf{\Sigma} \right] \end{aligned} \tag{6.1}$$

avec $||\Psi||^2 = \sqrt{\Psi^{\mathrm{T}} \Psi}$.

De même, toute base $\mathcal{S} \in \mathcal{G}(n, N)$ peut être projetée dans cet espace tangent $\mathcal{T}_{\mathcal{S}_0}$ au moyen de l'application logarithme géodésique $Log_{\mathcal{S}_0} : \mathcal{S} \in \mathcal{G}(n, N) \mapsto \mathbf{X} \in \mathcal{T}_{\mathcal{S}_0}$ avec :

$$\begin{aligned} &\frac{\Psi(\Psi_0 \Psi)^{-1} - \Psi_0}{||\Psi_0||} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad \text{(SVD fine)} \\ &\mathbf{H} = ||\Psi_0|| \mathbf{U} \tan^{-1}(\mathbf{\Sigma}) \mathbf{V}^T \end{aligned} \tag{6.2}$$

Ainsi, une méthode d'interpolation des bases est possible via l'algorithme 9 :

En dimension 1, une classique interpolation de Lagrange peut être effectuée. Dans le cas multidimensionnel, une interpolation IDW, ou un krigeage peuvent être effectués. Pour plus de détails, il est possible de se référer aux travaux de Mosquera Meza [Meza, 2018].

La méthode nécessite d'avoir un nombre de modes n pour chaque base associée à chaque solution calculée. Or dans le cadre de la méthode, le nombre de modes n'est pas défini à l'avance et dépend à la fois de l'indicateur d'erreur choisi comme critère d'arrêt, et de la position dans l'espace paramétrique comme le montre la section 9.2. La méthode ne sera pas employée ici mais possède des propriétés à tester dans le cadre LATIN-PGD.

Algorithm 8: Interpolation Grassmannienne

Initialisation :

- $j-1$ bases $\Psi_1, \ldots, \Psi_{j-1} \in \mathbb{R}^{N,m}$ respectivement associées à $\mu_1, \ldots, \mu_{j-1} \in \mathcal{D}$
- Nouveau paramètre $\mu_j \in \mathcal{D}$

Choix du point de référence ($\mathcal{S}_0 \in \mathcal{G}(n,N)$, $\Psi_0 \in \mathbb{R}^{N,m}$) avec r tel que
$\mu_r = \text{argmin}_{i \in [\![1,j-1]\!]} |\mu - \mu_i|$

Application $Log_{\mathcal{S}_0}$ — **Boucle sur les points interpolants** i

$[\mathbf{U}_i, \mathbf{\Sigma}_i, \mathbf{V}_i] \xleftarrow{\text{SVD fine}} (\Psi_i(\Psi_0^{\mathrm{T}}\Psi_i)^{-1} - \Psi_0)||\Psi_0||^{-1}$

$\mathbf{H}_i = \mathbf{U}_i \tan^{-1}(\mathbf{\Sigma}_i)\mathbf{V}_i^T$

Interpolation μ_j

$\mathbf{H}_j = \sum_{i=1}^{j-1} \omega_i(\mu_i)\mathbf{H}_i$

Application $Exp_{\mathcal{S}_0}$

$[\mathbf{U}_j, \mathbf{\Sigma}_j, \mathbf{V}_j] \xleftarrow{\text{SVD fine}} \mathbf{H}_j$

$\Psi = ||\Psi_0|| \, [\frac{\Psi_0}{||\Psi_0||}\mathbf{V}\cos\Sigma + \mathbf{U}\sin\Sigma]$

Sortie : Base adaptée Ψ

6.1.3 Interpolation des champs et de la base spatiale

Cette méthode permet d'exploiter les différentes solutions déjà calculées pour composer une première base de modes spatiaux $\mathbf{\Phi}_0$ et une solution $\mathbf{s}_0$.

Pour le calcul du paramètre $\underline{x}_j$, une interpolation de la solution $\mathbf{s} \in \mathbf{A}_\mathrm{d}$ est effectuée dans l'espace paramétrique. L'approche proposé consiste à exploiter une certaine distance entre les points dans l'espace paramétrique euclidien $\mathcal{D}$:

$$\mathrm{U}_j(t,M) \approx \sum_{k=0}^{j-1} \omega_k \, \mathrm{U}_k(t,M) \tag{6.3}$$

avec le vecteur de poids $\underline{\omega} \in \mathbb{R}^{j-1}$ dépendant d'une mesure d entre le point interpolé μ_j et les points d'interpolation : $\forall k \in [\![1, j-1]\!]$, $\omega_k = \phi(d(\underline{x}_j, \underline{x}_k))$.

Chaque champ U_k précedemment calculé se décompose en la solution élastique CA $\mathrm{U}_{k,0}$ et une correction CA_0 $\mathrm{U}_{k,c}$ sous forme PGD.

$$\mathrm{U}_k(t,M) = \mathrm{U}_{k,0}(t,M) + \mathrm{U}_{k,c}(t,M) = \mathrm{U}_{k,0}(t,M) + \sum_{l=1}^{m_k} \lambda_l(t)\,\Lambda_l(M) \tag{6.4}$$

Des points μ_k calculés, on se donne une base spatiale réduite CA_0 Ξ de taille m_s issue de la concaténation des modes spatiaux précedemment générés. Cette base sera notée $\Xi_\Lambda = [\Lambda_1^{(\Xi)}, \ldots, \Lambda_{m_s}^{(\Xi)}]$.

Le champ interpolé $\tilde{\mathrm{U}}_j$ et la base spatiale CA$_0$ Ξ_Λ sont définis, mais $\tilde{\mathrm{U}}_j$ n'est pas encore dans $\mathbf{A}_\mathrm{d}$. Pour se faire, la solution élastique CA associée au problème j $\mathrm{U}_{j,0}$ est calculée et le champ d'initialisation va être décomposé en le champ CA $\mathrm{U}_{j,0}$ et la projection CA0 de la différence entre $\tilde{\mathrm{U}}_j$ et $\mathrm{U}_{j,0}$ sur l'espace vectoriel générée par Ξ_Λ :

$$\mathrm{U}_j(t,M) \approx \underbrace{\mathrm{U}_{j,0}(t,M)}_{CA} + \sum_{i=1}^{m_s} \underbrace{\bar{\lambda}_i(t)\Lambda_i^{(\Xi)}(M)}_{CA_0} \tag{6.5}$$

avec $\forall i \in [\![1, m_s]\!]\,, \bar{\lambda}_i = \left\langle \Lambda_i^{(\Xi)},\, \tilde{\mathrm{U}}_j - \mathrm{U}_{j,0} \right\rangle_S$.

L'algorithme 9 récapitule les différentes étapes de la stratégie multi-paramétrique mise en place.

Algorithm 9: Stratégie multi-paramétrique

- **Initialisation** : n points à calculer $\mu_1, \ldots, \mu_n \in \mathcal{D}$
- **Pour chaque point** μ_j
  - **Pour chaque point précedent** μ_k
    - Calcul de la contribution du point : $\omega_k = \phi(\mathrm{d}(\mu_j, \mu_k))$
  - **Interpolation du champ** : $\mathrm{U}_j(t,M) \leftarrow \sum_{k=1}^{j-1} \omega_k \mathrm{U}_k(t,M)$
  - **Génération de la base commune spatiale**
    - Concaténation des modes spatiaux : $[\boldsymbol{\Xi}, \boldsymbol{\Sigma}, \mathbf{V}] \xleftarrow{\text{SVD fine}} [\boldsymbol{\Xi}_1, \ldots, \boldsymbol{\Xi}_{j-1}]$
  - **Calcul du champ élastique CA associé** : $\mathrm{U}_{j,0}(t,M)$
  - **Pour chaque mode spatial** $\Lambda_i^{(\Xi)}$
    - Calcul de la fonction temporelle $\bar{\lambda}_i \leftarrow \left\langle \Lambda_i^{(\Xi)},\, \tilde{\mathrm{U}}_j - \mathrm{U}_{j,0} \right\rangle_S$
  - **Correction LATIN** : $\mathrm{U}_j \in \mathbf{A}_{\mathrm{d}j} \cap \Gamma_j$
- **Sortie** : Champs mécaniques des n points

Il est nécessaire de caractériser les poids ω. Le poids des points voisins doit diminuer lorsque la distance augmente. Les plus grandes valeurs de p donnent une influence plus grande aux valeurs les plus proches du point interpolé. Ces poids peuvent être définis de plusieurs manières :

Méthode de Shepard

Cette méthode issue de [Shepard, 1968] est la plus connue avec : $\phi : \mathrm{d} \mapsto \dfrac{1}{\mathrm{d}^p}$, $p > 0$ et comme distance la distance canonique. Dans le cas où $p \to \infty$, on retrouve le cas d'interpolation au plus proche des voisins précedents.

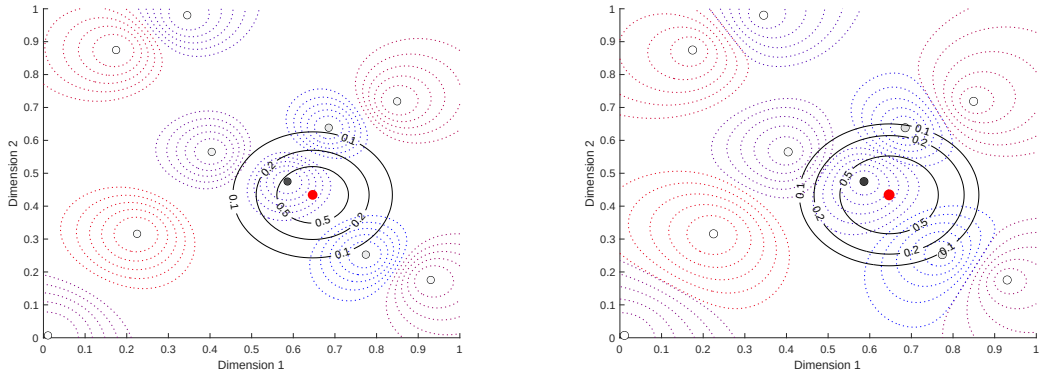
Méthode de Łukaszyk–Karmowski

Une autre méthode permettant de prendre en compte l'erreur sur les mesures se base sur le calcul de la distance à partir de la métrique de Łukaszyk–Karmowski [Łukaszyk, 2004]. La mise en place de cette technique n'a pas été abordée ici, mais pourrait avoir sa place étant donné la génération de données de différentes qualités.

Méthode basée sur les fonctions noyaux
Enfin, il est possible d'utiliser une fonction noyau comme la fonction exponentielle carrée $\phi : \phi : \mathrm{d} \mapsto \exp\left(-\gamma \mathrm{d}^2\right)$. On a alors $\phi : \mathrm{d} \mapsto \exp\left(-\gamma \mathrm{d}^2\right)$ et γ tel que $\sum_{k=0}^{j-1} \omega_k = 1$.

On compare la méthode de Shepard avec la méthode basée sur les fonctions noyaux. Le résultat est représenté sur les figures 6.1. On considère dans cet exemple un espace $\mathcal{D}$ 2D avec 10 points interpolants. Ces points possèdent une zone d'influence dans laquelle leur impact sur le point interpolé est majoritaire : cette zone d'influence est représentée par les zones concentriques en pointillées sur la figure. Un point interpolé (en rouge) est considéré. Autour de ce point, il est possible de tracer les iso-contour de poids : le point interpolant foncé est dans le cercle $\omega = 0.5$, son poids associé est supérieur à 0.5.

Les deux figures montrent des poids similaires alors que la méthode basée sur les fonctions noyaux nécessite une résolution non linéaire pour obtenir la valeur de γ. La méthode IDW classique sera donc employée par la suite.



(a) Méthode de Shepard (p=2)

(b) Méthode basée sur les fonctions noyaux

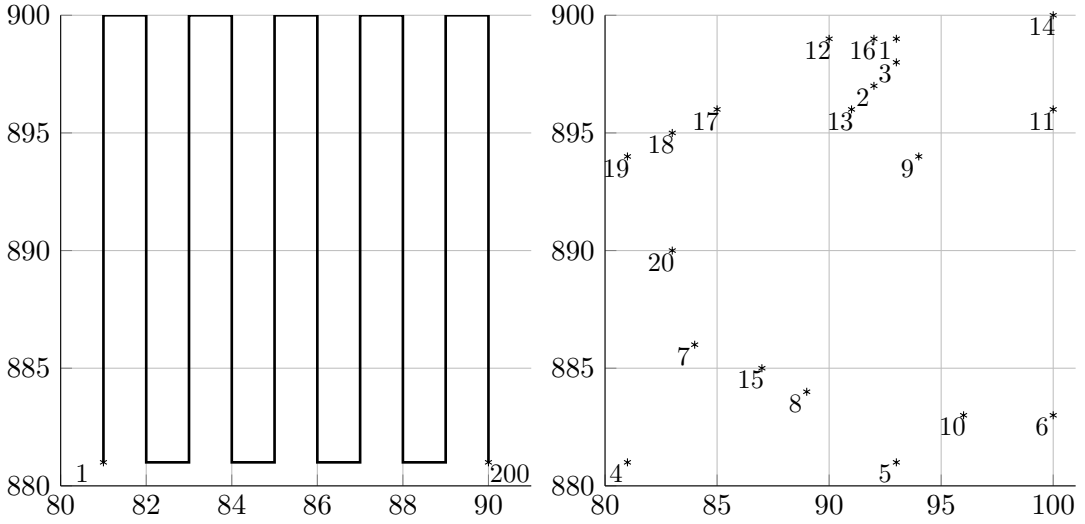
FIGURE 6.1 – Impact des points interpolants sur la construction du point interpolé
Légende : • ⟶ Point interpolé
• ⟶ Point interpolant (couleur selon poids)
○ ⟶ Zone d'influence
○ ⟶ Iso-contour de ω

6.2 Gain apporté par la méthode

La stratégie multi-paramétrique a été testée sur le cas-test de l'aube. Pour se faire, 400 points équidistants de l'espace paramétrique $[\alpha, T] = [81°, 100°] \times [881°\mathrm{C}, 900°\mathrm{C}]$ (cf. Section 7.4) ont été calculés sans la stratégie, et la stratégie a été testée pour plusieurs ordonancements de points :

— 200 points calculés par un chemin ordonné dans le sens de la variable T, puis de la variable α (cf. figure 6.2a)

— 15 ensemble de 200 points calculés par un chemin aléatoire (cf. figure 6.2b)



(a) Chemin ordonné

(b) Un des chemins aléatoires (20 premiers points)

Pour le chemin ordonné, le calcul est initialisé à partir de la solution précedente. La base de modes est obtenue par une SVD de la solution précedente, puis par une orthonormalisation suivant la matrice de rigidité. Le tableau 6.1 récapitule les gains vis-à-vis d'un calcul sans la stratégie.

Nombre de points calculés	**2**	**5**	**10**	**20**	**50**	**100**	**200**
Chemin régulier	4.25×	4.24×	4.46×	4.86×	4.95×	4.87×	5.05×

TABLE 6.1 – Gain de temps apporté par la stratégie multiparamétrique sur le cas-test de l'aube

Pour les chemins aléatoires, le calcul est initialisé à partir de l'interpolation IDW décrite par l'algorithme 9. Le tableau 6.2 récapitule les gains vis-à-vis d'un calcul sans la stratégie.

Nombre de points calculés	**2**	**5**	**10**	**20**	**50**	**100**	**200**
Cas optimal	2.70×	2.55×	2.64×	3.09×	3.68×	3.94×	4.12×
Moyenne	2.09×	2.13×	2.22×	2.56×	3.13×	3.51×	3.78×
Pire cas	0.91×	1.13×	1.34×	1.62×	1.86×	1.94×	2.02×

TABLE 6.2 – Gain de temps apporté par la stratégie multiparamétrique sur le cas-test de l'aube

La méthode peut apporter des gains bien plus importants dans le cas où les points à calculer sont encore plus proches.

Bilan

Les deux familles de méthodes jouent de concert pour apporter des gains de temps par l'utilisation de données multi-fidélités pour caractériser la QoI sur l'espace de conception, et sur la diminution du temps de calcul pour obtenir ces données. L'exploitation de données de qualité diverses peut s'effectuer de plusieurs manières : par l'exploitation des techniques de cokrigeage impactant la structure de covariance [4.2], par prise en compte de l'erreur sur les données approximées par ajout d'un bruit [4.3], ou par des techniques de krigeage récursif [4.4]. Ces méthodes présentent des difficultés techniques comme l'augmentation du nombre de paramètres à optimiser. La méthode Evofusion sera utilisée dans la suite pour exploiter les données multi-fidélité. La méthode a été améliorée par la prise en compte d'une variance sur les données basse fidélité corrigées.

Ces données multi-fidélité vont pouvoir être générée rapidement par l'utilisation du cadre LATIN-PGD. Un important travail a été effectué sur la méthode avec une importante contribution technique à l'approche dite à variables internes et à l'emploi d'un certain nombre des techniques de calcul pour rendre la méthode rapide et robuste [5.1]. En parallèle de celle-ci, l'approche fonctionnelle de la LATIN, plus simple à mettre en œuvre, a été implémentée pour comparer les temps de calcul [5.2]. L'implémentation des méthodes a été réalisée sous Matlab dans un formalisme adapté à la parallélisation des calculs permettant d'exploiter l'architecture matérielle multi-cœurs actuelle [5.3]. Les temps de calcul des deux approches ont été comparés sur deux cas-tests, permettant de mettre en évidence l'apport de l'approche fonctionnelle dans les cas étudiés. Cette approche a été ensuite testée sur un cas de calcul de 1.7 MDOFs permettant un calcul approché en 47 min, et un calcul exact en 1h27 [5.4].

Ces temps sont similaires à ceux obtenus au moyen de logiciels commerciaux, néanmoins la méthode LATIN présente l'avantage de permettre l'initialisation du calcul à partir de l'interpolation de résultats obtenus pour des configurations similaires. Plusieurs méthodes de réemploi d'informations ont été évoquées [6.1]. La méthode d'interpolation IDW a été choisie et testée sur l'aube paramétrée [6.2] pour un gain de l'ordre de 5× dans le cas où le parcours des points à calculer est ordonné, et de l'ordre de 3.8× pour un chemin aléatoire. Cette stratégie multiparamétrique permet un calcul plus rapide avec un résultat non faussé par la solution et la base de départ choisie, *a contrario* d'un calcul à partir d'une base POD. Elle mérite une étude plus poussée, notamment avec la mise en place d'une interpolation Grassmannienne de la base, et l'extension de cette stratégie pour la génération d'un modèle réduit multi-paramétrique d'erreur contrôlée.

Troisième partie

Mise en œuvre d'une stratégie de couplage robuste

Chapitre 7

Présentation de la campagne d'essais numériques

Sommaire

7.1 Mise en place de la campagne d'essais 95
7.2 Cas-test 1 : Plaque avec inclusion 96
7.2.1 Présentation du problème 96
7.2.2 Analyse de la réponse au problème 97
7.3 Cas-test 2 : Pièce d'amortissement 99
7.3.1 Présentation du problème 99
7.3.2 Analyse de la réponse au problème 100
7.4 Cas-test 3 : Aube de turbine 102
7.4.1 Présentation du problème 102
7.4.2 Analyse de la réponse au problème 103

7.1 Mise en place de la campagne d'essais

Neuf campagnes d'essais numériques vont être présentées dans la suite. Trois cas-tests paramétriques vont être étudiés dans un premier temps avec la réalisation d'un certain nombre de calculs permettant d'en extraire les temps de calcul, la réponse exacte de la quantité d'intérêt et l'évolution de l'erreur sur la quantité d'intérêt en fonction de l'indicateur LATIN ν. La génération d'un abaque virtuel de la QoI et l'optimisation globale de la QoI seront testées sur ces problèmes pour déterminer et valider les paramètres de la stratégie de couplage. Plus de 50000 points ont été calculés dans la première phase, puis plus de 10000 métamodèles nécessitant chacun entre 10 et 400 points auront été générés durant la thèse. Ces calculs auraient été complexes à mettre en œuvre sans les moyens de calcul du Mésocentre Fusion de CentraleSupélec et de l'École normale supérieure Paris-Saclay, et sans la mise en place d'un ensemble d'outils pour la génération et le stockage des 10 To de données générées. L'ensemble des calculs qui vont suivre ont été réalisés sur 8 cœurs d'un Intel® Xeon® CPU 6148 @ 2.40 GHz (Skylake).

7.2 Cas-test 1 : Plaque avec inclusion

7.2.1 Présentation du problème

Le premier cas-test est un huitième de plaque contenant une inclusion elliptique comme présentée sur la figure 7.1. Les dimensions de référence selon (x,y,z) sont $10 \times 1 \times 20$ mm. La loi de comportement des deux materiaux sont décrites par la loi de Chaboche, sans écrouissage dans ce cas de figure avec les paramètres matériaux indiqués Table 7.2.

La plaque est soumise à des conditions limites de symétrie sur les plans x=0, y=0, z=0 et à un déplacement imposé $\mathrm{U} = \underline{\mathrm{U}}_d$ sur la face supérieure z =20mm.
L'évolution temporelle du déplacement $\underline{\mathrm{U}}_d$ est représenté sur la figure 5.5b avec U_{max} = 0.03mm. Le matériau considéré possède une définition proche d'un acier 316 à 600 °K pour la plaque et l'inclusion, à ceci près que l'inclusion a un module d'Young $\boldsymbol{\alpha} E_{316}$ avec $\boldsymbol{\alpha} \in [0.1, 10]$ et que le coefficient puissance de la loi de Norton-Hoff associé à la plaque sera $\mathbf{n} \in [3, 7]$.

Le but ici est d'obtenir une approximation du maximum de la contrainte de Rankine pendant un cycle de chargement sur l'espace de conception 2D $(\alpha, n) \in [0.1, 10] \times [3, 7]$ comme décrit dans le Tableau 7.1.

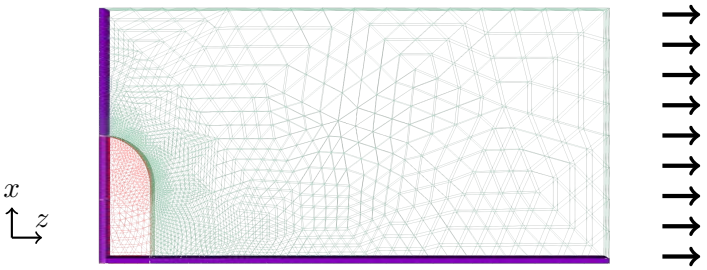


FIGURE 7.1 – Cas test d'une plaque avec inclusion avec des conditions de symétrie sur les parties violettes et un déplacement imposé sur la partie de droite. Une condition de symétrie est également présente au verso (non visible sur la figure).

Fonction objectif	$y(\underline{x}) = \max_{I \times \Omega} \sigma_{Rankine}$
Paramètres	$\underline{x} = (\boldsymbol{\alpha}, \mathbf{n}) \in \mathcal{D} = [0.1, 10] \times [3, 7]$
Type d'élément spatial	Triangulaire linéaire (8k DDLs)
Cycle de chargement	Un cycle (10s — 41 pas de temps)
Conditions limites	Conditions de symmétrie
	$\mathrm{U}_d = U_d(t)\underline{e_z}$ sur la surface supérieure

TABLE 7.1 – Paramètres du cas-test

part	E	$\boldsymbol{\sigma}_0$	n	K
Plaque	137.6 GPa	8 MPa	**n**	150 MPa.$s^{1/n}$
Inclusion	$\boldsymbol{\alpha} \times$ 137.6 GPa	8 MPa	5	150 MPa.$s^{1/n}$

TABLE 7.2 – Coefficients constitutifs élastiques-viscoplastiques pour les parties de cas de test d'inclusion (les symboles en gras sont les paramètres de conception)

7.2.2 Analyse de la réponse au problème

Les résultats sont issus de la première version du solveur avec l'approche LATIN à variables internes avec direction de recherche locale horizontale. Dans l'espace de conception indiqué précédemment, plus de 3000 points ont été générés pour permettre une étude de l'impact de l'indicateur LATIN sur trois critères : la quantité d'intérêt, le temps de calcul, et enfin le nombre de modes PGD générés.

Les figures 7.4 représentent l'évolution de la réponse de la QoI obtenue pour des calculs avec comme critère d'arrêt l'indicateur LATIN ν. Les surfaces pour $\nu = 10^{-2}$ et $\nu = 10^{-1}$ sont obtenues par krigeage avec estimation du bruit pour s'affranchir de valeurs aberrantes. On constate qu'un calcul fortement approximé donne déjà une bonne idée de la zone où se situe l'optimum.

L'erreur relative sur la QoI est récapitulée sur le graphique 7.2. Pour chaque valeur ν, la barre présente la plage de valeurs pour les 3000 calculs et le point donne la moyenne des valeurs.

Les données fortement approximées ont malgré tout l'avantage notable d'être peu coûteuses à calculer et de nécessiter peu de modes, comme le montre la figure 7.3.

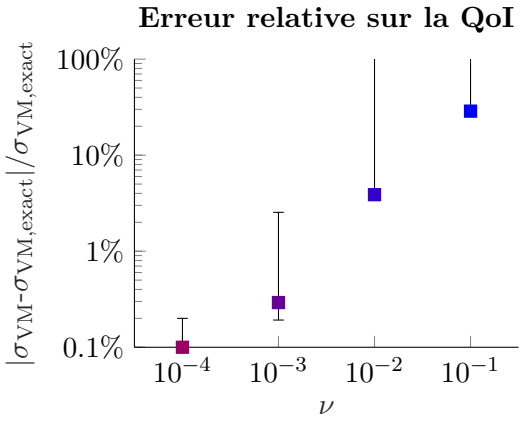
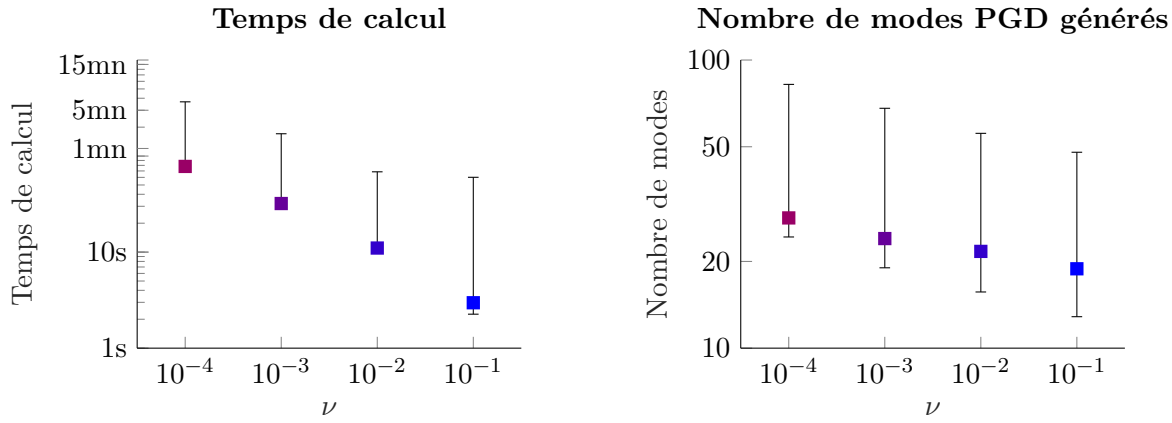


FIGURE 7.2 – Erreur sur la QoI



(a) Temps de calcul pour atteindre ν

(b) Nombre de modes PGD générés

FIGURE 7.3 – Influence de la valeur du critère d'arrêt

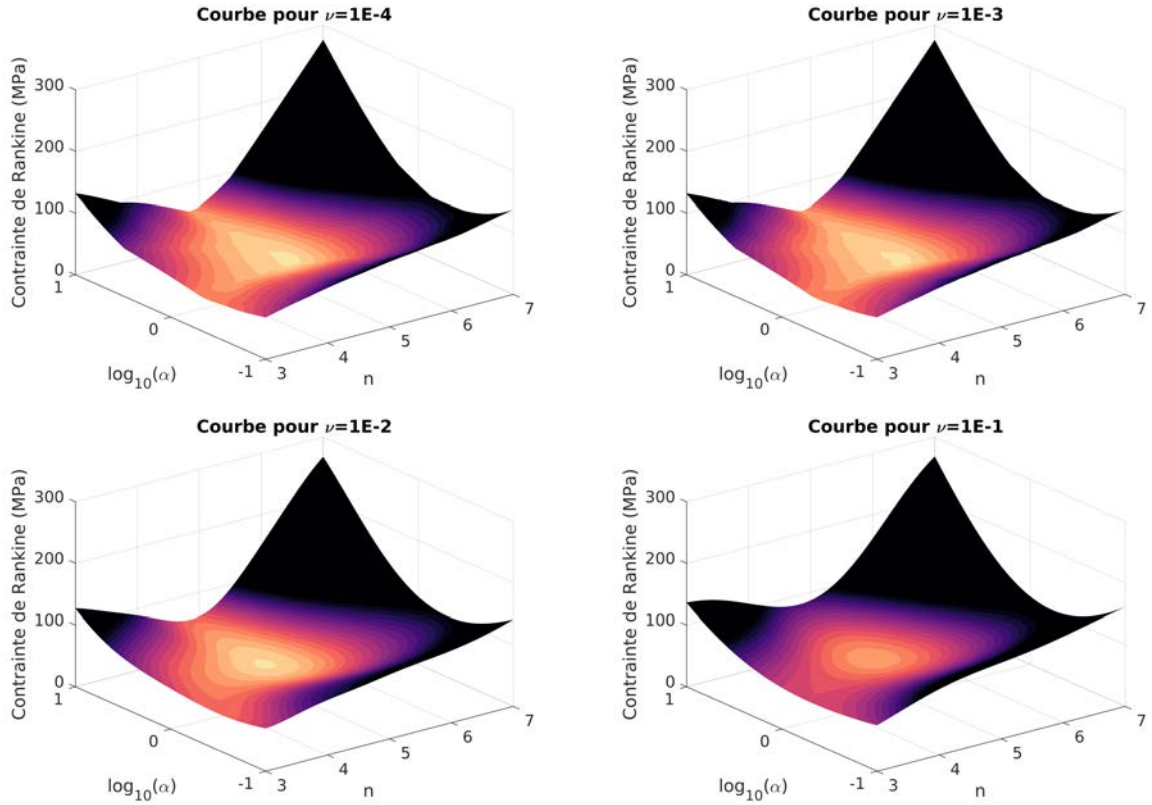


FIGURE 7.4 – Surfaces de réponse de la QoI en fonction de l'indicateur LATIN ν

7.3 Cas-test 2 : Pièce d'amortissement

7.3.1 Présentation du problème

Ce second test-case s'inspire très librement d'une pièce d'amortissement du moteur Vulcain présent dans le lanceur Ariane 5 [Néron *et al.*, 2015]. Les dimensions de référence suivant (x,y,z) sont $45 \times 70 \times 50$ mm. Le tableau 7.3 récapitule les caractéristiques de ce cas-test.

Fonction objective	$y(\underline{x}) = \max_{I\times\Omega} \sigma_{VonMises}$
Paramètres	$\underline{x} = (\theta_1, \phi_1, \theta_2, \phi_2) \in \mathcal{D} = [0°, 90°]^4$
Type d'élément spatial	Triangulaire linéaire (30k DOFs)
Cycle de chargement	Un cycle (10s — 41 pas de chargement)
Conditions limites	Encastrés sur les faces bleus
	$\underline{P}_1(t) = P(t)\,\underline{e}_1$ en vert
	$\underline{P}_2(t) = P(t)\,\underline{e}_2$ en rouge

TABLE 7.3 – Caractéristiques du cas-test 2

La pièce mécanique est supposée encastrée sur les côtés gauche et droit (parties bleues sur la figure 7.5). Le nez de de la pièce est visible en bas à gauche de la figure 7.5, et est chargé sur les faces avant et arrière (en vert et rouge sur la figure). Les deux pressions de charge ont la même intensité $P(t)$ décrite par la figure 7.6 avec $P_{max} = 80$ MPa, mais leur direction diffèrent.

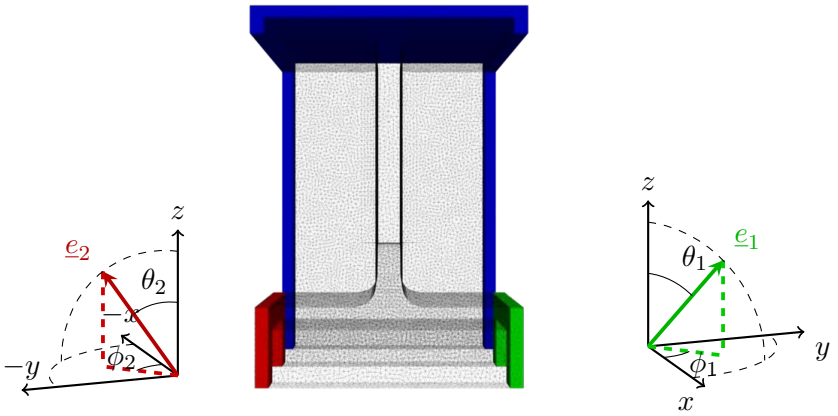
En effet, la direction de chaque chargement est pilotée par deux angles (θ, ϕ) qui peuvent varier entre $[0°, 90°]$. L'ensemble des possibilités de chargement forme un espace 4D $\mathcal{D} = [0°, 90°]^4$. La description des directions de chargement est visible sur la figure 7.5.

Concernant le matériau, il s'agit d'un acier 316 à 600 °K. Le comportement matériel est décrit par la loi constitutive de Chaboche. Le Tableau 7.4 récapitule les coefficients matériaux.

L'objectif ici est d'obtenir le cas de charge le plus défavorable pour la structure en considérant comme QoI le maximum de la contrainte Von Mises au cours d'un cycle de charge.

E	ν	n	K	σ_0	C	γ	R_∞	b
137.6 GPa	0.3	14	150 MPa.s$^{1/n}$	20 MPa	37.2 GPa	300	80 MPa	10

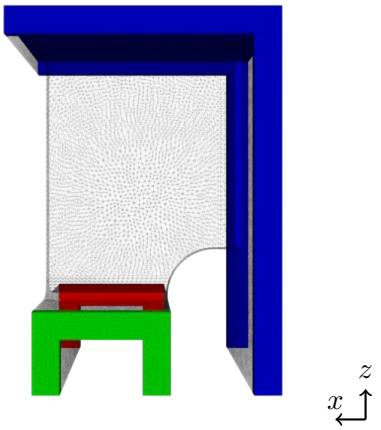
TABLE 7.4 – Coefficients de la loi viscoplastique associée au cas-test 2



(a) Direction de chargement associée à la partie rouge

(b) Vue suivant X

(c) Direction de chargement associée à la partie verte



(d) Vue suivant Y

FIGURE 7.5 – Descriptif du cas-test sur la pièce d'amortissement

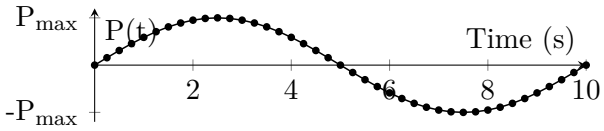


FIGURE 7.6 – Evolution de la pression P(t)

7.3.2 Analyse de la réponse au problème

L'ailette est un cas d'étude particulier dans la mesure où l'espace paramétrique est vaste et 4D. On représente la réponse à deux cas de chargement sur la figure 7.8. On constate sur ces deux cas que la zone de raccord entre le nez et le carter est la zone de concentration de contrainte ce qui est logique dans la mesure où il s'agit de la zone d'épaisseur la plus faible.

La complexité du chargement va modifier la localisation de la zone plastique mais celle-ci se situera toujours le long de l'arc.

Ici, la volonté est de trouver le cas de chargement le plus défavorable au sens du maximum de la contrainte de Von Mises, dans un espace de recherche vaste $\mathcal{D} = [0, 90]^4$. On trace sur la figure 7.7 la fonction de répartition des valeurs de la quantité d'intérêt. Le maximum de la quantité d'intérêt est de 118 MPa, 10% de l'espace de conception possèdent une valeur supérieure à 110 MPa dans de multiples zones non connexes et 1% une valeur supérieure à 113 MPa. L'objectif de recherche de l'optimum global sera considéré comme satisfait lorsque le maximum calculé sera supérieur à 113 MPa.

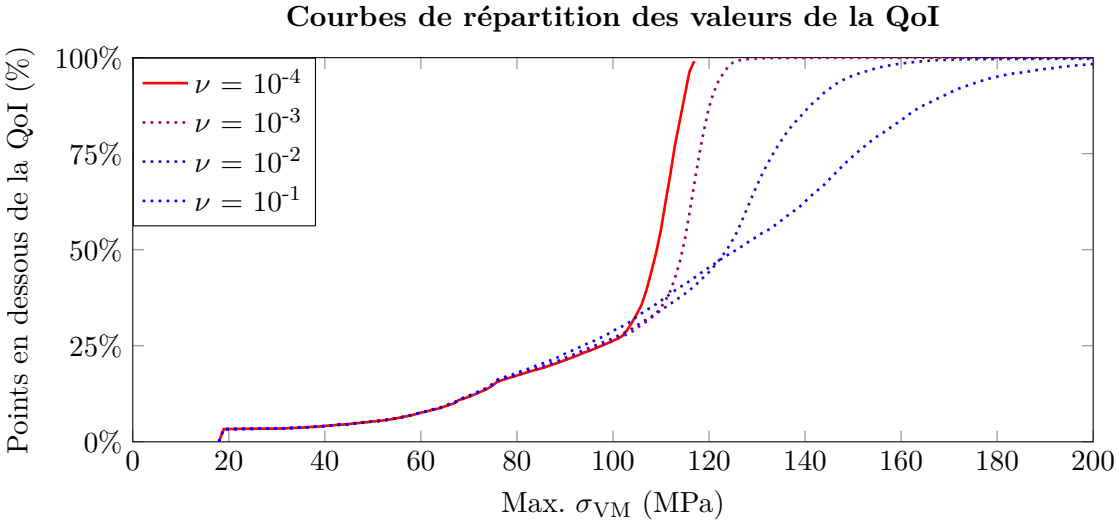
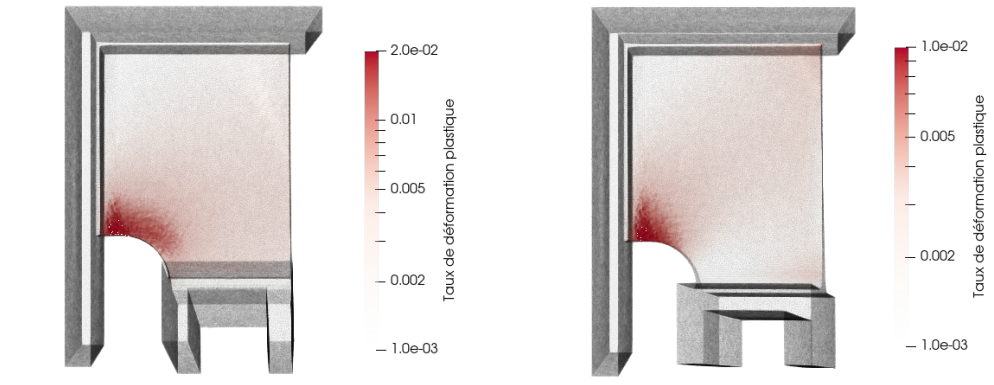


FIGURE 7.7 – Fonction de répartition des valeurs de la QoI



(a) $(\theta_1, \phi_1, \theta_2, \phi_2) = (45°, 0°, 0°, 0°)$ (b) $(\theta_1, \phi_1, \theta_2, \phi_2) = (0°, 0°, 0°, 60°)$

FIGURE 7.8 – Cas de chargement possible pour le cas-test

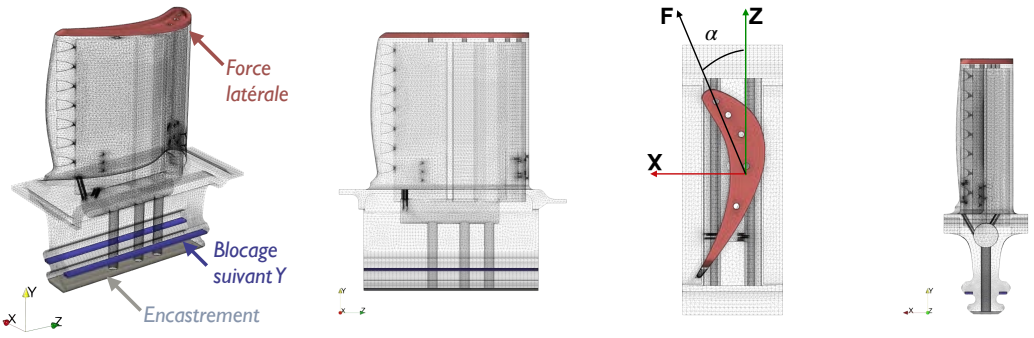
7.4 Cas-test 3 : Aube de turbine

7.4.1 Présentation du problème

Le dernier cas test est une aube de turbine d'avion inspirée du moteur M88 équipant l'avion de chasse Rafale. Le maillage est issu des travaux de [Blanchard, 2018]. Le cas-test représente un cas d'occurence extrême lié à une déformation du carénage. Dans ce cas, le talon subit une force de contact modélisée ici par une force latérale uniforme.

L'aube est modélisée comme étant encastrée sur l'embase du pied de sapin, avec un blocage des déplacements suivant l'axe y au niveau du contact avec le stator (en violet sur la figure 7.9). Celui-ci a une charge centrifuge et une charge uniforme sur la face supérieure (en rouge sur la figure 7.9). La direction de charge est dans le plan de surface et est pilotée par l'angle $\alpha \in [0°, 180°]$ comme vu dans la figure 7.9c. L'intensité des charges est indiquée sur la figure 7.10. Le matériau de ce cas de test est un Inconel 601 à température $T \in [800°\text{C}, 900°\text{C}]$. Les coefficients de la loi de comportement associée à la température T sont interpolés à partir des valeurs du tableau 7.5.

Le but ici est d'obtenir $\max_{I\times\Omega} \sigma_{VonMises} = 180$ MPa. L'optimisation est faite sur l'espace de conception 2D $(\alpha, T) \in [0°, 180°] \times [800°\text{C}, 900°\text{C}]$.



(a) Vue 3D (b) Vue suivant X (c) Vue suivant Y (d) Vue suivant Z

FIGURE 7.9 – Différentes vues de l'aube

Température T	n	K	σ_0	C	γ	R_∞	b
800°C	14	630 MPa.s$^{1/n}$	80 MPa	615 GPa	1.53.10^6 s^{-1}	80 MPa	300
850°C	11	560 MPa.s$^{1/n}$	71 MPa	497 GPa	1.36.10^6 s^{-1}	70 MPa	300
900°C	9	490 MPa.s$^{1/n}$	60 MPa	362 GPa	1.2.10^{6}s^{-1}	60 MPa	200

TABLE 7.5 – Coefficients de la loi viscoplastique associée (avec $E = 210$ GPa et $\nu = 0.28$)

Fonction objective	$y(\underline{x}) = \lvert\max_{I\times\Omega} \sigma_{VonMises} - 180\rvert$
Paramètres	$\underline{x} = (\alpha, T) \in [0°, 180°] \times [800°\text{C}, 900°\text{C}]$
Type d'élément spatial	Triangulaire quadratique (330kDDLs)
Conditions limites	Chargement en tête d'aube piloté angulairement
	Chargement centrifuge
	Encastrement en pied d'aube
	Blocage suivant Y sur les zones de contact

TABLE 7.6 – Caractéristiques du cas-test Aube

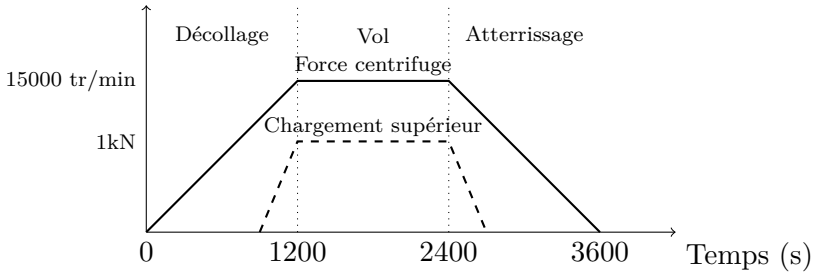


FIGURE 7.10 – Evolution en temps des efforts

7.4.2 Analyse de la réponse au problème

Une première analyse a été effectuée avec 11000 points dans l'espace paramétrique 2D. Ces points ont été calculés avec une précision maximum $\nu = 10^{-4}$ pour obtenir les courbes de réponse 7.11.

Pour aller plus loin dans le cadre de ce cas-test, on considère l'espace paramétrique 1D associé à l'angle de chargement $\alpha \in [\![0°, 180°]\!]$. On impose $T = 850°\text{C}$.

181 cas de chargement équirépartis sont testés pour quantifier l'influence de l'indicateur d'erreur sur le temps de calcul, le nombre de modes PGD générés et l'erreur sur la QoI pour ce problème. Les résultats sont présentés dans la Figure 7.12. La différence avec les autres cas-tests est la résolution du calcul pour un indicateur permettant de considérer la solution convergée. Cette solution est calculée avec un indicateur LATIN $\nu = 10^{-8}$.

Les courbes et l'erreur sur la quantité d'intérêt justifient l'utilisation de l'indicateur d'erreur LATIN $\nu = 10^{-4}$ comme critère d'arrêt pour le champ convergé. En effet, l'erreur sur la quantité d'intérêt issue de la résolution devient suffisamment faible pour que les erreurs de discrétisation et de modèle deviennent prépondérantes.

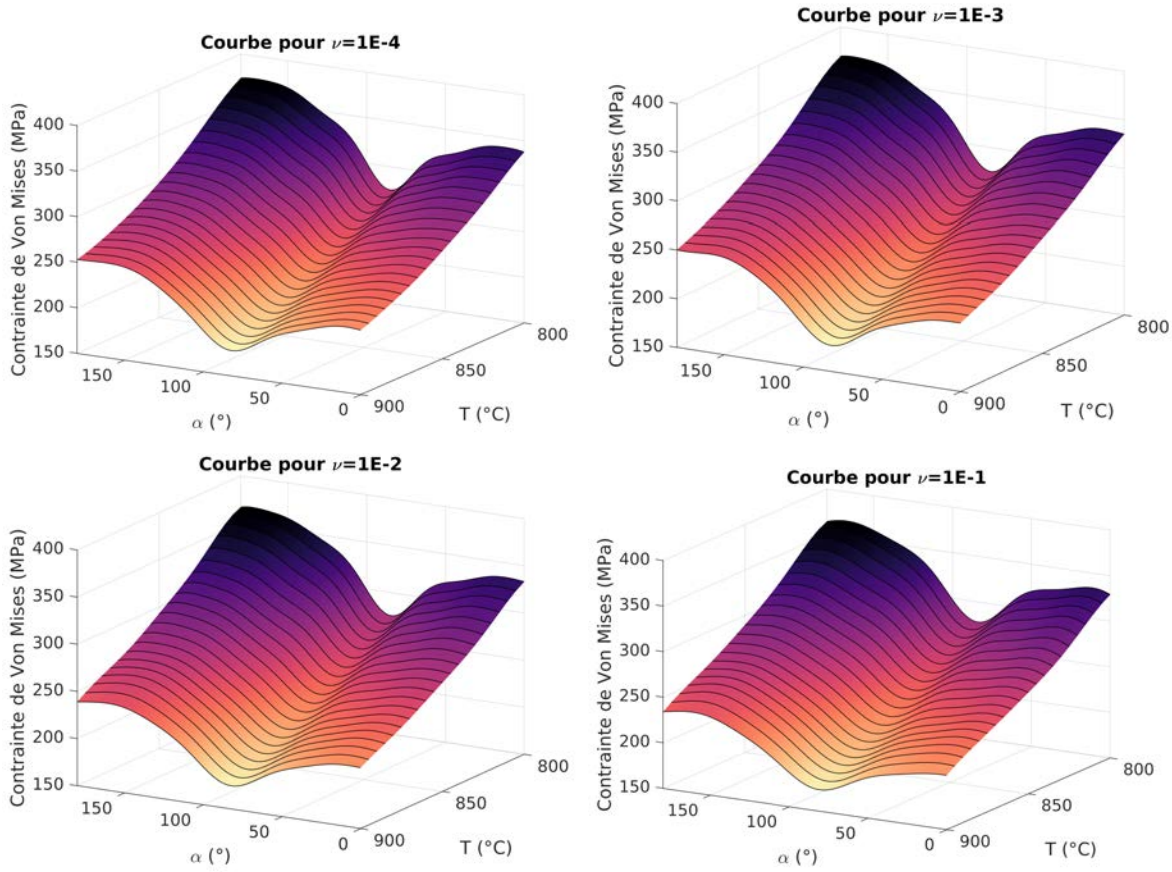
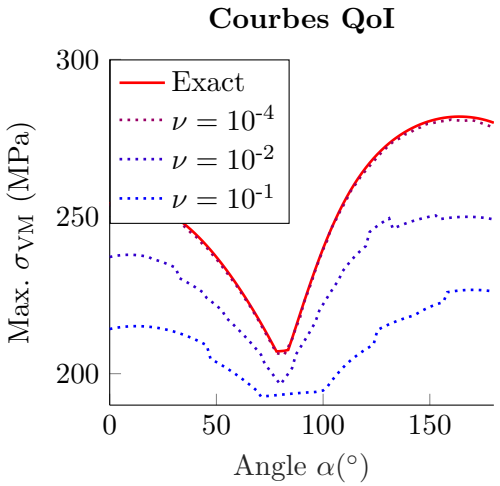
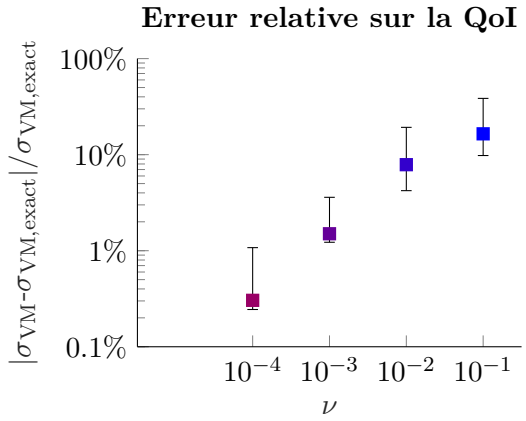


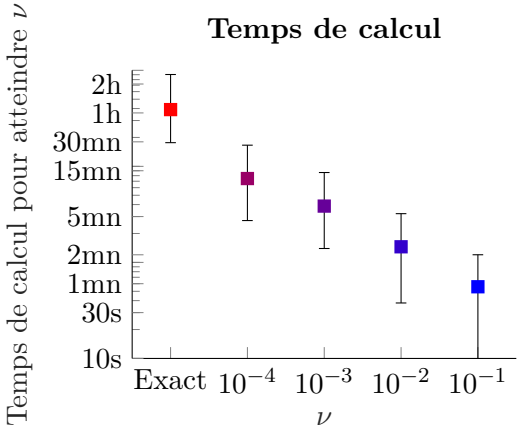
FIGURE 7.11 – Surfaces de réponse de la QoI en fonction de l'indicateur LATIN ν



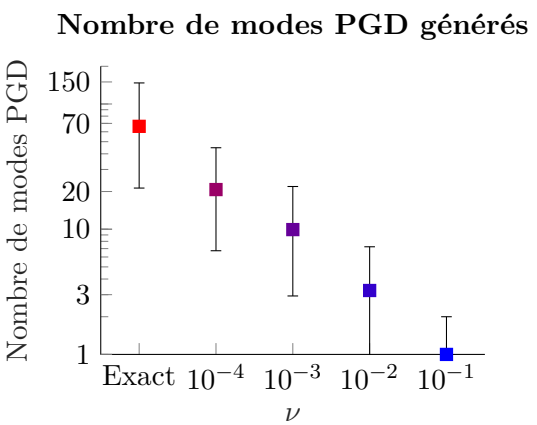
(a) Courbes de la QoI pour $T = 850°C$



(b) Erreur sur la QoI (En comparaison avec l'exact)



(c) Temps de calcul pour atteindre la valeur de l'indicateur d'erreur



(d) Nombre de modes PGD générés

FIGURE 7.12 – Influence de la valeur du critère d'arrêt

Chapitre 8

Génération rapide d'un abaque virtuel de la quantité d'intérêt

Sommaire

8.1 Généralisation de la stratégie au krigeage multi-fidélité 106
8.2 Détermination des paramètres de la stratégie 107
8.2.1 Génération complète du métamodèle — Cas-test 1 107
8.2.2 Génération complète du métamodèle — Cas-test 2 108
8.3 Validation sur le cas-test de l'aube 109

8.1 Généralisation de la stratégie au krigeage multi-fidélité

On a vu dans en section 2.1 la stratégie de génération d'un métamodèle par enrichissement. Une des difficultés non évoquée jusqu'à présent concerne le critère d'enrichissement (*Infill criteria*). Dans le cadre de la génération d'un abaque virtuel, l'objectif est d'avoir une approximation de la QoI sur tout l'espace. Le critère doit donc limiter l'erreur de modèle tout en minimisant le nombre d'observations.

Le krigeage possède déjà un puissant outil pour déterminer les zones à enrichir qu'est la variance du métamodèle également appelé erreur aux moindres carrées. D'autres critères plus efficients ont été développés comme l'Integrated MSE (IMSE) [Bates *et al.*, 1996] qui consiste en un point à intégrer la variance associée au métamodèle enrichi par la valeur estimée en ce point. D'autres critères sont plus performants dans le cas de données bruités comme l'IMSE empirique [Sacks *et al.*, 1989] ou encore les méthodes basées sur la validation croisée [Kleijnen et van Beers, 2004].

Dans le cadre de ces travaux, le choix a été d'exploiter uniquement la variance du métamodèle comme critère d'enrichissement dans un premier temps afin d'observer les gains déjà apporté par la stratégie. Le critère d'arrêt choisi sera un critère de Bravers-Person $r_{ccc} > 0.95$ lors de la comparaison du métamodèle généré à un métamodèle de référence obtenu par une grille régulière fine.

Dans le cadre du krigeage multi-fidélité, il est nécessaire de déterminer à chaque étape d'en-

richissement la qualité de la réponse voulue. La figure 8.1 présente le changement dans l'algorithme. Dans le cadre présenté dans ces travaux, le choix s'est porté sur la génération d'un premier métamodèle avec un certain nombre de points n_{BF} basse-fidélité, ayant comme indicateur d'erreur associé ν_{BF}, puis de points haute-fidélité jusqu'à atteindre la critère d'arrêt du métamodèle.

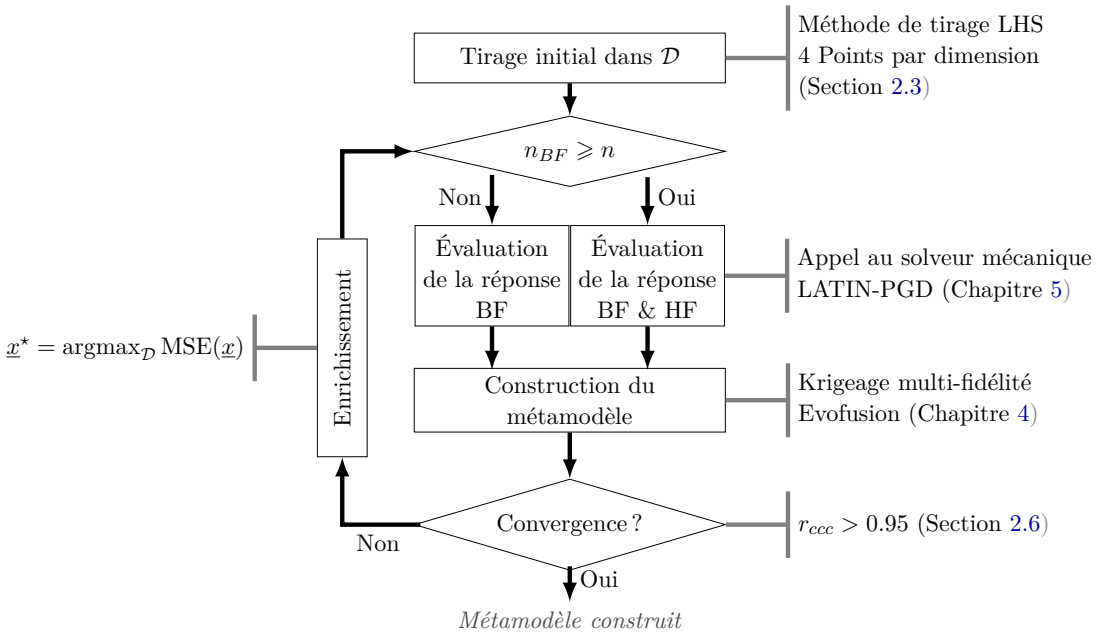


FIGURE 8.1 – Représentation schématique de la stratégie multi-fidélité mise en œuvre

Les deux paramètres (n_{BF}, ν_{BF}) sont à déterminer empiriquement. Pour se faire, les deux premiers cas-tests ont été exploités.

8.2 Détermination des paramètres de la stratégie

8.2.1 Génération complète du métamodèle — Cas-test 1

La première campagne porte sur la génération complète du métamodèle pour le cas-test de la plaque avec inclusion.

La campagne de tests donnent les résultats présentés sur le tableau 8.1. La meilleure stratégie obtenue est celle obtenue pour un critère d'arrêt en basse-fidelité $\nu_{BF} = 10^{-2}$ et le calcul de 2×12 points basse fidélité. En effet, dans ce cadre d'étude, les données basse fidélité suffisent à caractériser avec une précision suffisante le métamodèle. Le critère d'arrêt est donc rapidement atteint. Dans ce cas de figure, un gain de 5.2× issu de l'emploi du multi-fidélité a été obtenu. L'utilisation de la stratégie multi-paramétrique du solveur a permis quant à elle d'obtenir un gain de 1.5×. En effet, la génération du métamodèle prend en moyenne 1h11 avec des données exactes sans stratégie multi-paramétrique, soit un gain total de 7.8× entre

la meilleure stratégie et le cas d'un krigeage avec des données convergées issues d'un solveur classique. Néanmoins, on constate également le risque que peut engendrer l'utilisation de données basse-fidélité trop grossières comme dans le cas $\nu_{BF} = 10^{-1}$. Dans ce cas de figure, l'information apporté par les points basse-fidélité dégrade très fortement le métamodèle : le métamodèle généré est moins bon que le métamodèle obtenu dans le cas avec uniquement des données haute-fidelité.

Le tableau 8.2 présente la variance normalisée de chaque configuration présentée. On constate une grande disparité des temps de calcul en fonction du tirage initial et plus particulièrement pour les configurations à faible temps de génération. Néanmoins, la comparaison des temps maximaux pour chaque couple de paramètres nous donne le même résultat.

n_{BF} \ ν_{BF}	10^{-3}	10^{-2}	10^{-1}
4/dim	27'	24'	1h09'
8/dim	15'	8'57	46'
12/dim	13'	**5'30**	45'

TABLE 8.1 – Moyenne des temps de calcul pour obtenir $r_{ccc} > 0.95$ (Uniquement des données exactes : 24')

n_{BF} \ ν_{BF}	10^{-3}	10^{-2}	10^{-1}
4/dim	26%	27%	13%
8/dim	30%	40%	16%
12/dim	22%	32%	16%

TABLE 8.2 – Variance normalisée des temps de calcul pour générer un métamodèle complet

8.2.2 Génération complète du métamodèle — Cas-test 2

La seconde campagne porte sur la génération complète du métamodèle pour le cas-test de la pièce d'amortissement.

La campagne de tests donnent les résultats présentés sur le tableau 8.3. La meilleure stratégie obtenue est celle obtenue pour un critère d'arrêt en basse-fidelité $\nu_{BF} = 10^{-2}$ et le calcul de 4×12 points basse fidélité. Dans ce cas de figure, un gain de 11.8× issu de l'emploi du multi-fidélité a été obtenu. L'utilisation de la stratégie multi-paramétrique dans ce cadre un gain de 3.2× permettant d'estimer un gain de 37.8× entre la meilleure stratégie et le cas d'un krigeage avec des données convergées issues d'un solveur classique.

On constate donc que la génération complète du métamodèle par la stratégie de couplage permet des gains de 7.8× sur le cas-test 1 bi-matériau et de 37.8× sur le cas-test 2 comparé à une stratégie de génération avec une unique source de données et un solveur sans stratégie multi-paramétrique. Le meilleur choix de paramètres pour la stratégie correspond à une erreur basse-fidélité $\nu_{BF} = 10^{-2}$ et 12 points basse fidélité par dimension générés. Ceci concorde à la règle empirique de 10 points par dimension courrament présente dans la littérature [Courrier *et al.*, 2016]. La même étude avec 10 et 14 points basse fidélité par dimension est à réaliser pour s'assurer de ce paramètre.

ν_{BF} / n_{BF}	10^{-3}	10^{-2}	10^{-1}
4/dim	3h17	2h55	2h29
8/dim	1h45	1h35	2h06
12/dim	1h14	**34'**	36'

TABLE 8.3 – Moyenne des temps de calcul pour obtenir $r_{ccc} > 0.95$ (Uniquement des données exactes : 3h10)

ν_{BF} / n_{BF}	10^{-3}	10^{-2}	10^{-1}
4/dim	26%	27%	13%
8/dim	30%	40%	16%
12/dim	22%	32%	16%

TABLE 8.4 – Variance normalisée des temps de calcul pour générer un métamodèle complet

8.3 Validation sur le cas-test de l'aube

Ce cas-test va permettre de mesurer sur un cas industriel à la fois le gain apporté par les données multi-fidélité, et le gain apporté par la stratégie multi-paramétrique. Pour se faire, une première phase de calcul notée ϕ_0 avec uniquement des données exactes sans la stratégie multi-paramétrique, puis la phase ϕ_1 avec la stratégie multi-paramétrique, et enfin la phase ϕ_2 avec la stratégie proposée dans ce manuscrit en calculant initialement 10 points par dimension avec une erreur $\nu_{BF} = 10^{-2}$. 20 tirages initiaux seront testés afin de limiter l'impact des tirages dans la mesure du temps de calcul.

La phase ϕ_0 est réalisée en moyenne en 12h03, la phase ϕ_1 en 6h18 et enfin la phase ϕ_2 en 1h46. La stratégie multiparamétrique apporte un gain de 1.9× et la stratégie multi-fidélité un gain de 4.2× pour un gain total de 6.9×.

Chapitre 9

Optimisation bayésienne de la quantité d'intérêt

Sommaire

9.1 Généralisation de la stratégie au krigeage multi-fidélité 110
9.1.1 Critères d'enrichissement 111
9.1.2 Amélioration de la stratégie dans le cas du krigeage multi-fidélité 113
9.2 Détermination des paramètres de la stratégie 114
9.2.1 Recherche du minimum global — Cas-test 1 114
9.2.2 Recherche du maximum global — Cas-test 2 115
9.3 Validation sur le cas-test de l'aube 117

9.1 Généralisation de la stratégie au krigeage multi-fidélité

L'optimisation bayésienne [Močkus, 1975, Brochu *et al.*, 2010] est équivalente à l'algorithme EGO [Jones *et al.*, 1998] ou encore la méthode SMBO [Hutter *et al.*, 2011]. Cette méthode d'optimisation globale est de plus en plus employé, notamment en machine learning où elle est utilisée pour la détermination des hyperparamètres optimaux [Van Aken *et al.*, 2017]. Elle a notamment été étendue à l'optimisation multi-objective [Swersky *et al.*, 2013], à des problèmes à grand nombre de paramètres [Wang *et al.*, 2012] et de nombreux travaux ont portés sur son utilisation dans un environnement parallèle [Janusevskis *et al.*, 2012a, Contal *et al.*, 2013].

Elle correspond à un enrichissement séquentiel du métamodèle par maximisation de la fonction d'amélioration espérée (*expected improvement* — EI). Dans le cadre de données multi-fidélité, il est nécessaire d'étudier l'ensemble des fonctionelles d'enrichissement pour choisir la plus adaptée à notre cas d'étude. On notera dans la suite l'optimum parmi les données observées $\underline{x}^\star = \min_{\underline{x} \in \boldsymbol{X}} y(\underline{x})$ avec $y(\underline{x}^\star) = y^\star$.

9.1.1 Critères d'enrichissement

Probabilité d'amélioration (Probability of improvement — PI)

La probabilité d'amélioration est le premier critère imaginé [Kushner, 1964]. Il s'appuie sur la maximisation de la probabilité d'améliorer le minimum observé. On considère la fonction d'amélioration suivante :

$$\mathrm{I}(\underline{x}) = \max\{0, \hat{y}(\underline{x}) - y^\star\} \tag{9.1}$$

On peut alors calculer la probabilité d'amélioration :

$$\mathrm{PI}(\underline{x}) = \int_{I=0}^{I=\infty} \frac{1}{\sqrt{2\pi}\sigma(\underline{x})} \exp\left(-\frac{(\mu(\underline{x}) - y^\star - I)^2}{2\,\sigma^2(\underline{x})}\right) = \Phi\left(\frac{\mu(\underline{x}) - y^\star}{\sigma(\underline{x})}\right) \tag{9.2}$$

avec $\hat{y}$ le processus gaussien $N\left(\mu(\underline{x}), \sigma(\underline{x})\right)$ modélisant y, $y^\star$ le minimum des valeurs observées et Φ la fonction de répartition de la distribution normale centrée réduite.

Amélioration espérée (Expected improvement — EI)

La fonction d'amélioration espérée [Mockus *et al.*, 1978] est la plus utilisée dans la littérature. Elle consiste à non pas calculer la probabilité mais la moyenne de la fonction d'amélioration en intégrant la fonction de densité de I :

$$\mathrm{EI}(\underline{x}) = \int_{I=0}^{I=\infty} I \frac{1}{\sqrt{2\pi}\sigma(\underline{x})} \exp\left(-\frac{(\mu(\underline{x}) - y^\star - I)^2}{2\,\sigma^2(\underline{x})}\right) dI \tag{9.3}$$

Cette équation s'écrit analytiquement :

$$\mathrm{EI}(\underline{x}) = (\mu(\underline{x}) - y^\star)\Phi\left(\frac{\mu(\underline{x}) - y^\star}{\sigma(\underline{x})}\right) + \sigma(\underline{x})\phi\left(\frac{\mu(\underline{x}) - y^\star}{\sigma(\underline{x})}\right) \tag{9.4}$$

avec ϕ la fonction de densité de la distribution normale centrée réduite. On observe deux termes dans l'expression analytique (9.4) : un premier terme d'exploration dont la contribution sera importante dans les zones non observées de $\mathcal{D}$, et un terme d'exploitation dont la contribution sera importante au voisinage de l'optimum observé.

Amélioration espérée pondérée (Weighted Expected improvement — WEI)

Les contributions de l'EI peuvent être pondérées par un certain coefficient ω pour privilégier un terme par rapport à l'autre dans le cadre de l'amélioration espérée pondérée [Sóbester *et al.*, 2005] :

$$\mathrm{WEI}(\underline{x}) = \omega(\mu(\underline{x}) - y^\star)\Phi\left(\frac{\mu(\underline{x}) - y^\star}{\sigma(\underline{x})}\right) + (1-\omega)\sigma(\underline{x})\phi\left(\frac{\mu(\underline{x}) - y^\star}{\sigma(\underline{x})}\right) \tag{9.5}$$

Dans le cas d'un krigeage interpolant (sans variance associée aux observations), $\mathrm{EI}(\underline{x}) = 0$. La fonction étant positive, les points déjà échantillonnées ne seront jamais réévalués. Pour cette

raison, il est indispensable que les données corrigés de la méthode Evofusion ait une variance associée. De plus, le minimum observé n'est pas nécessairement exact dans le cas où il est issu des données basse fidélité.

Amélioration espérée avec ajout (Expected improvement with plug-in — PI50)

Pour remédier au dernier point, il est possible de ne pas considérer le minimum observé $y^\star$ mais le minimum de la moyenne ou un certain quantile du processus gaussien aux points observés, noté $\mu^\star$. On a alors :

$$\mathrm{PI50}(\underline{x}) = (\mu(\underline{x}) - \mu^\star)\Phi\left(\frac{\mu(\underline{x}) - \mu^\star}{\sigma(\underline{x})}\right) + \sigma(\underline{x})\phi\left(\frac{\mu(\underline{x}) - \mu^\star}{\sigma(\underline{x})}\right) \tag{9.6}$$

Amélioration espérée augmentée (Augmented Expected Improvement — AEI)

Dans le cas où la variance des données basse-fidélité est connue, [Huang *et al.*, 2006] propose la modulation du critère par un terme de pénalisation favorisant l'exploration :

$$\mathrm{AEI}(\underline{x}) = \left(1 - \frac{\tau}{\sqrt{\sigma(\underline{x})^2 + \tau^2}}\right)\mathrm{PI50}(\underline{x}) \tag{9.7}$$

Amélioration espérée augmentée (Approximate knowledge gradient — AKG)

Cette méthode s'appuie sur la stratégie du gradient de connaissance (*knowledge gradient policy*) [Scott *et al.*, 2011] estimant l'apport d'une observation supplémentaire au métamodèle, à la manière du IMSE. On considère la fonction d'amélioration suivante :

$$\mathrm{I}(\underline{x}) = \min_{\underline{x}\in\boldsymbol{X}^+} \hat{y}(\underline{x}) - \min_{\underline{x}\in\boldsymbol{X}^+} \hat{y}^+(\underline{x}) \tag{9.8}$$

avec $\boldsymbol{X}^+$ l'ensemble des points observés enrichi du point $\underline{x}$ à estimer et $\hat{y}^+$ le métamodèle généré à partir de $\mathcal{T} \cup \{\underline{x}, \hat{y}(\underline{x})\}$.

On s'intéresse alors à la moyenne de la fonction d'amélioration en intégrant la fonction de densité de I. Le problème nécessite en l'état la génération d'un nouveau métamodèle pour chaque point testé. Néanmoins, une astuce technique permet de limiter le coût de calcul du critère.

Borne de confiance supérieure (Upper confidence bound — UCB)

Dans le cadre de la maximisation d'une fonctionnelle pour le choix des hyperparamètres, la méthode GP-UCB [Srinivas *et al.*, 2012] maximise la borne de confiance supérieure du processus.

$$\mathrm{UCB} = \mu(\underline{x}) + \beta\sigma(\underline{x}) \tag{9.9}$$

avec β le paramètre de définition de l'enveloppe. Usuellement, $\beta = 2$ est employée. Des travaux ont porté récemment sur ces méthodes [Contal, 2016], avec une application dans le cadre multi-fidélité [Kandasamy *et al.*, 2016].

Une grande partie de ces méthodes ont été comparées dans [Picheny *et al.*, 2013]. L'analyse faite amène à penser que la méthode AKG est la plus adaptée au cas d'observations bruitées. Néanmoins, elle présente l'inconvénient d'être complexe à implémenter. Pour débuter, seules les techniques EI, $PI50$ seront testées dans le cadre d'un métamodèle généré à partir de la méthode Evofusion avec variance sur les données basse-fidélité. L'ensemble des fonctions d'enrichissement proposées seront à expérimenter à l'avenir, notamment la fonction AKG.

9.1.2 Amélioration de la stratégie dans le cas du krigeage multi-fidélité

L'emploi de données basse fidélité peut fortement fausser le processus d'optimisation dans le cas où l'optimum a été calculé uniquement en basse-fidélité avec une correction réalisée par la méthode Evofusion. Dans ce cas de figure, il est intéressant de calculer cet optimum en haute-fidélité pour vérifier la solution obtenue comme représenté dans le schéma 9.1. On comparera cette stratégie notée S2 à la stratégie S1 sans la dernière étape (en rouge dans le schéma).

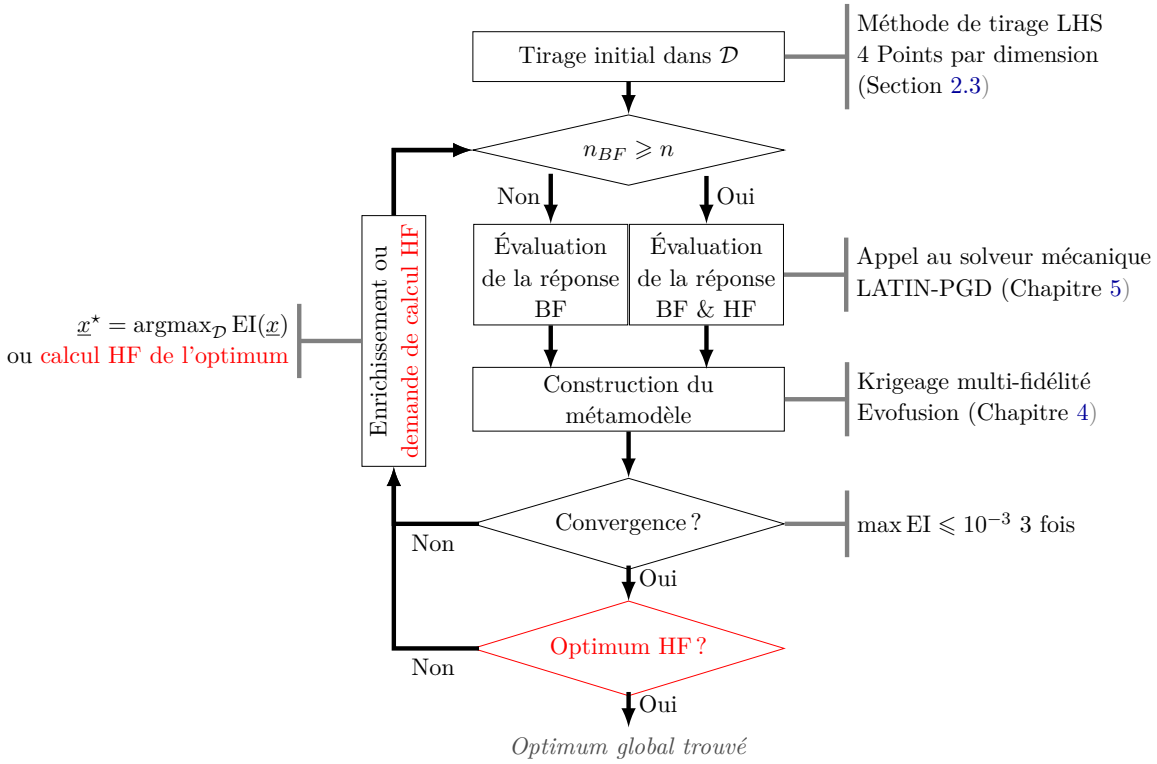


FIGURE 9.1 – Représentation schématique de la stratégie d'optimisation mise en œuvre. La stratégie notée S1 n'utilise pas cette étape en rouge, la stratégie notée S2 l'intègre.

9.2 Détermination des paramètres de la stratégie

On cherche dans cette section à estimer le couple de paramètres (n_{BF}, ν_{BF}) minimisant le temps de calcul pour l'optimisation globale. Pour se faire, le nombre de points par dimension n_{BF} va varier dans l'ensemble $\{6, 8, 10, 12\}$ et le critère d'arrêt BF du solveur basé sur l'indicateur LATIN ν_{BF} va varier dans l'ensemble $\{10^{-3}, 10^{-2}, 10^{-1}\}$. Pour chaqu'un des 12 couples, 20 optimisations seront réalisés avec différents tirages initiaux. Ces tirages sont les mêmes pour chaque couple.

9.2.1 Recherche du minimum global — Cas-test 1

La campagne d'essais qui suit caractérise le cas-test de la plaque avec inclusion avec un minimum localisé à $(log_{10}(\alpha), n) = (-0.08, 5.1)$. Ce cas est simple dans la mesure où les gradients sont conséquents autour du minimum.

Le critère de réussite de l'optimisation correspond au cas où l'écart entre la solution exacte calculée au minimum estimé, et le véritable minimum est inférieur à 2 MPa :

$$y_2(\underline{x}^\star) - y_2^\star \leqslant 2\text{MPa} \tag{9.10}$$

Le tableau 9.1 donne le taux de réussite dans le cadre de la stratégie S1 avec l'EI comme critère. Dans le cas de la stratégie S2, l'intégralité des solutions satisfont le critère de réussite.

ν_{BF} / n_{BF}	10^{-3}	10^{-2}	10^{-1}
6 / dim	90%	75%	50%
8 / dim	100%	85%	25%
10 / dim	100%	70%	40%
12 / dim	90%	55%	20%

TABLE 9.1 – Taux de réussite dans le cadre de la stratégie S1

ν_{BF} / n_{BF}	10^{-3}	10^{-2}	10^{-1}
6 / dim	100%	100%	100%
8 / dim	100%	100%	100%
10 / dim	100%	100%	100%
12 / dim	100%	100%	100%

TABLE 9.2 – Taux de réussite dans le cadre de la stratégie S2

Ces résultats valident le choix de confirmer l'optimum par un calcul haute-fidélité. Seuls les résultats de la stratégie S2 seront présentés dans la suite.

Deux critères d'enrichissement seront étudiés dans la suite : le critère EI et le crtière PI50. Les tableaux 9.3 et 9.4 permettent de comparer les méthodes et notamment les moyennes des temps de calcul pour chacune de ces méthodes. En terme de temps de calcul, la meilleure stratégie obtenue est celle obtenue pour un critère d'arrêt en basse-fidelité $\nu_{BF} = 10^{-2}$ et le calcul de 2×10 points basse-fidélité. En effet, on constate qu'une approximation des données basse-fidélité permet d'être encore plus rapide, mais que ceci se fait au détriment de la qualité

du minimum d'une part, et que le temps de calcul est très fortement dépendant du tirage initial.

n_{BF} \ ν_{BF}	10^{-3}	10^{-2}	10^{-1}
6 / dim	48'	33'	37'
8 / dim	50'	30'	38'
10 / dim	48'	37'	35'
12 / dim	48'	43'	40'

TABLE 9.3 – Moyenne des temps de calcul avec utilisation du critère EI - Seulement HF : 55'

n_{BF} \ ν_{BF}	10^{-3}	10^{-2}	10^{-1}
6 / dim	29'	26'	28'
8 / dim	29'	26'	27'
10 / dim	33'	**21'**	26'
12 / dim	41'	30'	29'

TABLE 9.4 – Moyenne des temps de calcul avec utilisation du critère PI50 - Seulement HF : 55'

Les différents tirages testés permettent également de juger de la sensibilité de la méthode au tirage initial. Le tableau 9.5 présente les variances normalisées de chaque configuration. On constate qu'elles sont semblables à celles obtenues dans le cas de la génération d'un abaque virtuel.

n_{BF} \ ν_{BF}	10^{-3}	10^{-2}	10^{-1}
6 / dim	21%	36%	20%
8 / dim	14%	27%	20%
10 / dim	17%	31%	18%
12 / dim	9%	12%	14%

TABLE 9.5 – Variance normalisée du temps de calcul (Cas PI50) - Seulement HF : 20%

9.2.2 Recherche du maximum global — Cas-test 2

La seconde campagne d'essais a été effectuée à partir d'une base de plus de 20000 points générées pour l'analyse de la QoI faite précedemment. Le gain apporté par le multi-paramétrique ne sera pas quantifiable, néanmoins un plus grand nombre de tirages initiaux seront testés. Pour chacun des couples (n_{BF}, ν_{BF}) testés, 100 optimisations sont réalisées avec différents tirages initiaux. Ces tirages sont les mêmes pour chaque couple. Un budget de 100 points haute-fidélité est fixé.

Le critère de réussite de l'optimisation correspond au cas où la solution exacte calculée au minimum estimé est supérieure à 113 MPa :

$$y_2(\underline{x}^\star) > 113\text{MPa} \tag{9.11}$$

Dans ce cas de figure, le critère d'arrêt n'est que trop rarement atteint. Le temps de calcul sera calculé par rapport à l'instant où le critère de réussite est atteint. Le tableau 9.6 monte le pourcentage de cas-tests ayant convergé vers la zone du minimum global. Les tableaux 9.7 et 9.8 permettent de juger des temps de calcul et de la sensibilité du couple de paramètres au tirage initial.

n_{BF} \ ν_{BF}	10^{-3}	10^{-2}	10^{-1}
6/dim	56%	83%	75%
10/dim	58%	84%	74%
12/dim	63%	87%	74%
14/dim	59%	87%	77%

TABLE 9.6 – Pourcentage de cas où la zone du minimum global a été trouvée

n_{BF} \ ν_{BF}	10^{-3}	10^{-2}	10^{-1}
6/dim	27'40	9'48	3'14
10/dim	36'02	10'09	3'23
12/dim	42'38	10'27	3'26
14/dim	43'22	11'14	3'38

TABLE 9.7 – Moyenne des temps de calcul — Seulement HF : 1h01

n_{BF} \ ν_{BF}	10^{-3}	10^{-2}	10^{-1}
6/dim	47%	87%	14%
10/dim	72%	73%	28%
12/dim	106%	78%	30%
14/dim	107%	91%	42%

TABLE 9.8 – Variance normalisée du temps de calcul — Seulement HF : 34%

En terme de temps de calcul, la meilleure stratégie obtenue est celle obtenue pour un critère d'arrêt en basse-fidelité $\nu_{BF} = 10^{-1}$ et le calcul de 4×6 points basse-fidélité. Ce résultat est guère surprenant. En effet, la courbe 7.7 montre que les données obtenues pour un indicateur ν_{BF} élevé sont plus discriminées que les données pour un indicateur ν_{BF} faible. Les premiers enrichissements avec des données basse-fidélité permettent d'obtenir le critère de réussite. Avec ce choix, dans cette configuration, le gain apporté par le multi-fidélité est de 19×. Néanmoins, en terme de réussite, l'utilisation de $\nu_{BF} = 10^{-2}$ semble être un bon compromis entre temps de calcul et réussite. Avec un gain estimé apporté par le multi-paramétrique de 2.5×, le gain total serait de 47.5×. Ce gain est à valider *a posteriori* en effectuant le calcul sans l'apport de la stratégie multi-paramétrique, avec uniquement des données haute-fidélité.

On constate dans cette section que l'optimisation du métamodèle par la stratégie de couplage et la méthode EGO permet des gains plus importants encore que dans le cas d'une génération d'un métamodèle juste sur tout l'espace de conception. La seule difficulté est sur le meilleur choix de paramètres pour la stratégie. Au vu des cas-tests précedents, on considérera pour la suite un critère d'arrêt en basse-fidelité $\nu_{BF} = 10^{-2}$ et le calcul de 10 points basse-fidélité par dimension. Une stratégie adaptative pour le choix du critère d'arrêt sur les données basse-

fidélité serait une solution pour s'affranchir de cette problématique.

9.3 Validation sur le cas-test de l'aube

Ce cas-test est plus simple que le précédent. L'idée de celui-ci est de pouvoir mesurer sur un cas industriel à la fois le gain apporté par les données multi-fidélité, et le gain apporté par la stratégie multi-paramétrique. Pour se faire, une première phase de calcul notée ϕ_0 avec uniquement des données exactes sans la stratégie multi-paramétrique, puis la phase ϕ_1 avec la stratégie multi-paramétrique, et enfin la phase ϕ_2 avec la stratégie proposée dans ce manuscrit en calculant initialement 10 points par dimension avec une erreur $\nu_{BF} = 10^{-2}$. 20 tirages initiaux seront testés afin de limiter l'impact des tirages dans la mesure du temps de calcul.

Le critère de réussite de l'optimisation correspond au cas où la solution exacte calculée au minimum estimé est inférieure à 180 MPa :

$$y_2(\underline{x}^\star) < 180\text{MPa} \tag{9.12}$$

Dans ce cas de figure, le critère de réussite est atteint à chaque fois. La phase ϕ_0 est réalisée en 6h05, la phase ϕ_1 en 2h51 et enfin la phase ϕ_2 en 44 minutes. La stratégie multi-paramétrique apporte un gain de 2.1× et le multi-fidélité un gain de 3.9× pour un gain total de 8.4×.

Bilan

Trois cas-tests ont été développés pour faire un choix des tester le couplage des méthodes. Le premier cas-test est une plaque avec une inclusion oblongue simple sur laquelle les paramètres de la stratégie ont été testés [7.2]. Les deux autres cas-tests [7.3,7.4] représentent des problèmes sur des structures industrielles à deux et quatre paramètres permettant de valider la stratégie choisie.

Dans un premier temps, la stratégie de couplage a été adaptée à la génération d'un métamodèle de la quantité d'intérêt sur tout l'espace de conception [8.2]. Dans ce cadre, la meilleure stratégie est de générer les $12\times d$ premiers points avec un critère d'arrêt à $\nu_{BF} = 10^{-2}$. Dans ce cas de figure, on observe des gains de $7.8\times$ sur le cas-test 1 bi-matériau [8.3]. Ces gains proviennent à la fois et de l'utilisation de données multi-fidélité (de l'ordre de $5.2\times$) et de la stratégie multi-paramétrique mise en place ($1.5\times$). Le second cas-test confirme les paramètres proposés et permet d'estimer un gain de $37.8\times$ entre la stratégie proposée et une statégie classique avec une unique source de données sans stratégie multi-paramétrique. Le choix des paramètres est à nuancer dans la mesure où une stratégie avec plus de points basse-fidélité pourraient également convenir, mais le choix du couplage métamodèle multi-fidélité et stratégie multi-paramétrique semble pertinante dans ce cas de figure.

Dans un second temps, la stratégie de couplage a été exploitée pour l'optimisation globale de la quantité d'intérêt. Plusieurs fonctions d'enrichissement adaptées à la technique multi-fidélité ont été proposées [9.2] et deux ont été testés [9.3]. Le choix s'est porté sur la méthode EI avec plug-in, notée PI50. La stratégie a permis de montrer des gains pouvant atteindre jusqu'à un facteur $47\times$. La difficulté porte dans ce cadre sur le choix des meilleurs paramètres pour la stratégie. Les deux premiers cas-tests mettent en évidence la nécessité de générer $10\times d$ premiers points avec un critère d'arrêt à $\nu_{BF} = 10^{-2}$. Dans le cadre de l'optimisation de l'aube, la stratégie multi-paramétrique apporte un gain de $2.1\times$ et la gestion de données multi-fidélité apporte un gain de $3.9\times$ pour un gain total de $8.4\times$ [7.4.2].

On constate au travers de cette étude deux problématiques à résoudre. La première concerne le critère d'arrêt de la stratégie. Dans le cas où le métamodèle de référence n'existe pas, il est nécessaire d'avoir un critère permettant de valider le métamodèle, ou d'utiliser dans le cas d'une optimisation globale une seconde phase utilisant un algorithme d'optimisation possédant de bonnes propriétés de convergence. La seconde problématique concerne le choix du critère d'arrêt du solveur. Les différents cas-tests ont convergé vers l'utilisation d'un critère basse-fidélité $\nu_{BF} = 10^{-2}$. Néanmoins, les informations à disposition ne sont pas suffisantes pour garantir du bon choix de ce critère. Une autre approche est possible : elle consiste à estimer l'erreur sur la QoI par quelques calculs haute-fidélité. Ces calculs permettent alors de faire un choix adapté au cas-test étudié.

Conclusion et perspectives

Les travaux présentés s'appliquent à réduire la phase d'optimisation paramétrique des structures mécaniques pour permettre l'utilisation de modèles plus complexes et plus justes au sein des bureaux d'études. Pour arriver à cette fin, la stratégie consistant à générer et exploiter un métamodèle de la quantité d'intérêt étudiée a été améliorée.

En effet, la stratégie a été généralisée à l'emploi de données multi-fidélité. L'étape de métamodélisation a été modifiée et le krigeage usuellement utilisé a été remplacé par un krigeage multi-fidélité. Ce manuscrit présente la génération du processus gaussien modèle par les deux approches observables dans la littérature : à savoir la construction d'un estimateur linéaire non-biaisé, et l'inférence bayésienne permettant d'obtenir un processus gaussien conditionné aux observations. L'exploitation de données de qualité diverses peut s'effectuer de plusieurs manières : par l'exploitation des techniques de cokrigeage impactant la structure de covariance, par prise en compte de l'erreur sur les données approximées par ajout d'une variance sur les données d'observation, ou par des techniques de krigeage récursif. Ces méthodes présentent des difficultés techniques comme l'augmentation du nombre de paramètres à optimiser, mais l'analyse de l'ensemble de ces méthodes a permis de choisir l'utilisation de la méthode Evofusion pour la prise en compte des données multi-fidélité, et d'améliorer celle-ci par une estimation de la variance associée aux données basse fidélité. Ainsi, les données basse fidélité corrigées ne sont plus interpolantes, mais une estimation de l'erreur de correction est prise en compte.

Les données multi-fidélité utilisées ne sont pas obtenues par l'appel à un modèle de qualité donné, mais par l'emploi d'un solveur permettant le calcul des champs mécaniques avec un niveau de fidélité adapté. Le solveur proposé se base sur la méthode LATIN qui présente le triple avantage de générer rapidement des champs spatio-temporels sous forme à variable séparée PGD, de donner une approximation de la solution complète et de la quantité d'intérêt à chaque itération, et d'autoriser l'initialisation d'un calcul à partir des solutions de problèmes similaires dans le cadre de la stratégie multi-paramétrique. Dans le cas où l'approximation calculée à partir de la base réduite est mauvaise, il est ainsi possible d'ajouter en temps réel de nouveaux modes permettant de réduire l'erreur de modèle, contrairement aux techniques classiques de génération d'une base *offline* pour un calcul *online*. Un important travail a été effectué sur la méthode dans le cadre d'un problème élasto-viscoplastique. De nombreuses contributions techniques ont été apportées à l'approche de la méthode dite *à variables internes*, permettant d'accélérer les calculs. En parallèle de celle-ci, l'approche fonctionnelle de la LATIN, plus simple à mettre en œuvre, a été implémentée afin de comparer les temps de calcul. L'implémentation des méthodes a été réalisée sous Matlab dans un formalisme adapté à la parallélisation. Cette approche a été ensuite testée sur l'étude d'un modèle d'aube de turbine à 1.7 MDOFs. Une bonne approximation des champs mécaniques est générée en 47 min. Le calcul exact quant à lui s'obtient en 1h27mn. Ces temps sont similaires à ceux obtenus au moyen de logiciels commerciaux. Néanmoins, la méthode LATIN permet l'initialisation du calcul à partir de l'interpolation de résultats obtenus pour des configurations similaires pré-

cédemment calculées. La comparaison de multiples méthodes d'interpolations multivariées a permis de faire un choix sur la méthode IDW. Cette stratégie multi-paramétrique a été testée sur l'aube paramétrée et a apporté un gain pouvant varier de 3× à 5× sur ce cas de figure à qualité de réponse identique.

Les deux familles de méthodes présentées vont ainsi travailler de concert pour apporter d'importants gains de temps sur la génération d'un abaque virtuel sur l'espace paramétrique étudié, ou sur l'optimisation d'une quantité d'intérêt sur cet espace. Pour se faire, le métamodèle sera généré à partir d'un tirage initial calculé en basse fidélité et enrichit au moyen de données basse fidélité, puis haute-fidélité. Le choix des points d'enrichissement est réalisé par maximisation d'une fonction estimant le gain d'information. La généralisation de la stratégie d'enrichissement nécessite la détermination du nombre de points basse fidélité calculés, et du critère d'arrêt du solveur dans ce cas basse fidélité. Quatre campagnes d'étude paramétrique ont été réalisées au moyen du mésocentre Fusion pour estimer ces deux paramètres. Dans le cadre de la génération d'un abaque virtuel, le choix de 10 points basse fidélité par dimension et d'un critère d'arrêt LATIN $\nu_{BF} = 10^{-2}$ se dégagent avec un gain de temps pour la génération du métamodèle pouvant atteindre 37×. Ce choix est moins évident pour une optimisation bayésienne de la quantité d'intérêt. Néanmoins, les études ont permis de fixer les paramètres de la stratégie à un choix de 10 points basse fidélité par dimension et d'un critère d'arrêt LATIN $\nu_{BF} = 10^{-2}$. Dans ce cadre, l'optimisation de l'aube a été obtenue en 44 minutes au lieu de 6 heures pour un gain de 8.4×.

Ces travaux se terminent sur des résultats prometteurs, mais ouvrant la porte à plusieurs pistes de recherche pour rendre la méthode encore plus rapide et robuste.

Perspectives à court terme

À court terme, la stratégie est à appliquer sur des problèmes d'optimisation géométrique. La principale difficulté concerne la stratégie multi-paramétrique et l'interpolation de champs provenant de géométries différentes. Cette difficulté peut être contournée par morphing autour d'un unique maillage [Courard *et al.*, 2016], par utilisation de méthodes XFEM [Noël *et al.*, 2016], ou par paramétrisation isogéométrique de la structure (Isogéométric Analysis — IGA) [Hughes *et al.*, 2005]. Dans ce dernier cas, les outils déjà développés peuvent être utilisés en utilisant une méthode de couplage IGA-FEM [Tirvaudey *et al.*, 2019a].

À ce sujet, la stratégie peut également être améliorée par un meilleur choix de la fonction d'enrichissement. Une comparaison des différentes fonctions d'enrichissement est à réaliser à la manière de [Picheny *et al.*, 2013] dans le cas de problèmes mécaniques. Plusieurs critères d'enrichissement, dont le critère AKG n'ont pas été comparés aux autres. Or il s'avère que la fonction AKG peut permettre d'obtenir plus rapidement l'optimum.

Enfin, un des points tout juste évoqués dans cette thèse concerne les modèles réduits multi-paramétriques permettant une approximation des champs sur tout l'espace de conception. La stratégie employée dans cette thèse consiste à générer le champ solution et la base associée pour un certain nombre de points et d'interpoler à partir de ces solutions sur l'espace

paramétrique. Cette stratégie de génération de modèles réduits à partir de *snapshots* PGD est à comparer en temps de calcul à la génération d'un modèle réduit par la PGD multi-paramétrique [Vitse *et al.*, 2014].

Perspectives à moyen terme

Si on s'intéresse à une échelle un peu plus longue en temps, la stratégie est à améliorer pour permettre le calcul en parallèle de plusieurs points de manière asynchrone, grâce à des fonctions d'enrichissement adaptées. De nombreux travaux récents portent sur le sujet [Janusevskis *et al.*, 2012b, Contal *et al.*, 2013, Kandasamy *et al.*, 2017] et permettraient ainsi d'exploiter un parallélisme hybride OpenMP/MPI et d'exploiter au maximum les ressources de calcul disponibles.

Un travail sur l'utilisation de niveaux de qualité continus dans la stratégie de couplage est également à faire. À partir d'un métamodèle liant l'indicateur LATIN et l'erreur sur la quantité d'intérêt, il serait possible de choisir la précision de la réponse pour chaque point calculé et de pouvoir allouer de manière optimale les ressources de calcul.

Pour aller plus loin sur l'erreur en QoI, des techniques d'adaptation de modèle notamment basée sur l'erreur en relation de comportement modifiée existent [Chamoin *et al.*, 2014, Li *et al.*, 2018]. Un découplage des erreurs de discrétisation, de modèle et de solveur peut être réalisé [Tirvaudey *et al.*, 2019b] pour trouver le bon compromis erreur sur la QoI/temps de calcul, et obtenir rapidement un abaque virtuel certifié.

Enfin, l'étape suivante pour la stratégie multi-paramétrique est d'améliorer la méthode proposée et de mettre en place une interpolation Grassmannienne de la base initiale pour le nouveau calcul pour quantifier les gains apportés.

Perspectives à long terme

À long terme, l'ensemble des travaux réalisés au cours des 20 dernières années sur la méthode LATIN sont à industrialiser. Les outils d'hyper-réduction [Capaldo *et al.*, 2017] sont à mettre en place dans le cadre de problèmes de très grande taille de même que les méthodes de résolution adaptées à ce type de problème [Marchand et Quilici, 2019]. Une implémentation de la méthode dans les logiciels industriels est indispensable. Pour se faire, l'approche à variables internes peut exploiter les logiciels de génération automatique de modèles éléments-finis (Automated Computational Modelling — ACM) comme AceGen [Korelc, 1997] ou Fenics [Kirby et Logg, 2006]. L'approche fonctionnelle est très proche de briques déjà présentes dans les codes industriels et permet notamment à Siemens d'envisager l'intégration de la méthode dans leur logiciel, dans le cadre de la thèse de Ronan Scanff [Scanff *et al.*, 2019].

Bibliographie

[Allaire, 2007] Allaire, G. (2007). *Conception optimale des structures*. Numéro 58 de Mathématiques & applications. Springer, Berlin. 8, 26

[Allix et Ladevèze, 1992] Allix, O. et Ladevèze, P. (1992). Interlaminar interface modelling for the prediction of delamination. *Composite Structures*, 22(4):235–242. 21

[Ammar *et al.*, 2012] Ammar, A., Chinesta, F., Cueto, E. et Doblaré, M. (2012). Proper generalized decomposition of time-multiscale models. *International Journal for Numerical Methods in Engineering*, 90(5):569–596. 10

[Amsallem et Farhat, 2008] Amsallem, D. et Farhat, C. (2008). Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA journal*, 46(7):1803–1813. 87

[Ancelet *et al.*, 2012] Ancelet, S., Bousquet, N. et Parent, E. (2012). Optimisation bayésienne de plans d'expérience physiques. 28

[Arzt, 1994] Arzt, M. (1994). *Approche Des Phénomènes Cycliques Par La Méthode à Grand Incrément de Temps*. Cachan, Ecole normale supérieure. 61

[Bates *et al.*, 1996] Bates, R. A., Buck, R. J., Riccomagno, E. et Wynn, H. P. (1996). Experimental Design and Observation for Large Systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):77–94. 106

[Bendsøe, 1995] Bendsøe, M. P. (1995). *Optimization of Structural Topology, Shape and Material*. Springer-Verlag, Berlin Heidelberg. 9

[Bertram et Zimmermann, 2018] Bertram, A. et Zimmermann, R. (2018). Theoretical investigations of the new Cokriging method for variable-fidelity surrogate modeling : Well-posedness and maximum likelihood training. *Advances in Computational Mathematics*, 44(6):1693–1716. 53

[Bhattacharyya *et al.*, 2018] Bhattacharyya, M., Fau, A., Nackenhorst, U., Néron, D. et Ladevèze, P. (2018). A multi-temporal scale model reduction approach for the computation of fatigue damage. *Computer Methods in Applied Mechanics and Engineering*, 340:630–656. 47, 71

[Blanchard, 2018] Blanchard, M. (2018). *Méthode global/local non-intrusive pour les simulations cycliques non-linéaires*. Thèse de doctorat, Université Paris-Saclay. 102

[Boisse, 1987] Boisse, P. (1987). *Nouvel Algorithme à Grand Incrément de Temps Pour Le Calcul Des Structures Élastoplastiques*. Paris 6. 61

[Bordeu *et al.*, 2011] Bordeu, F., Chinesta, F., Leygue, A., Cueto, E. et Niroomandi, S. (2011). Réduction de modèle par PGD appliqué la simulation en temps réel de solide déformables. *In 10e Colloque National En Calcul Des Structures*, Giens, France. 45

[Boucard et Ladevèze, 1999] Boucard, P.-A. et Ladevèze, P. (1999). Une application de la méthode LATIN au calcul multirésolution de structures non linéaires. *Revue Européenne des Éléments Finis*, 8(8):903 – 920. 10

[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. 45

[Box et Wilson, 1951] Box, G. E. P. et Wilson, K. (1951). On the Experimental Attainment of Optimum Conditions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 13(1):1–45. 32

[Breiman, 2001] Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32. 32

[Brochu *et al.*, 2010] Brochu, E., Cora, V. M. et de Freitas, N. (2010). A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. *arXiv :1012.2599 [cs]*. 110

[Broomhead et Lowe, 1988] Broomhead, D. S. et Lowe, D. (1988). Radial basis functions, multi-variable functional interpolation and adaptive networks. 10

[Broyden, 1970] Broyden, C. G. (1970). The Convergence of a Class of Double-rank Minimization Algorithms. *IMA Journal of Applied Mathematics*, 6(1):76–90. 26

[Capaldo *et al.*, 2017] Capaldo, M., Guidault, P. A., Néron, D. et Ladevèze, P. (2017). The Reference Point Method, a "hyperreduction" technique : Application to PGD-based nonlinear model reduction. *Computer Methods in Applied Mechanics and Engineering*, 322:483–514. 45, 82, 121

[Chamoin *et al.*, 2014] Chamoin, L., Ladevèze, P. et Waeytens, J. (2014). Goal-oriented updating of mechanical models using the adjoint framework. *Computational Mechanics*, 54(6):1415–1430. 121

[Champaney, 1996] Champaney, L. (1996). *Une Nouvelle Approche Modulaire Pour l'analyse d'assemblages de Structures Tridimentionnelles*. Thèse de doctorat, ENS-Cachan, Cachan, France. 21

[Champaney *et al.*, 1997] Champaney, L., Cognard, J. Y., Dureisseix, D. et Ladevèze, P. (1997). Large scale applications on parallel computers of a mixed domain decomposition method. *Computational Mechanics*, 19(4):253–263. 21

[Chinesta *et al.*, 2010] Chinesta, F., Ammar, A. et Cueto, E. (2010). Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. *Arch Comput methods Eng*, 17:327–350. 10

[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. 45

[Chinesta *et al.*, 2014] Chinesta, F., Keunings, R. et Leygue, A. (2014). *The Proper Generalized Decomposition for Advanced Numerical Simulations*. SpringerBriefs in Applied Sciences and Technology. Springer International Publishing, Cham. 43

[Ciarlet, 1988] Ciarlet, P. G. (1988). *Mathematical Elasticity. Volume 1, Volume 1,*. North-Holland ; Sole distributors for the U.S.A. and Canada, Elsevier Science Pub. Co., Amsterdam ; New York ; New York, N.Y., U.S.A. OCLC : 312444936. 15

[Cognard, 1989] Cognard, J.-Y. (1989). *Une Nouvelle Approche Des Problèmes de Plasticité et Viscoplasticitépar La Méthode à Grand Incrément de Temps*. Thèse de doctorat, Université Pierre et Marie Curie, Paris, France. 61, 79

[Collombier, 1995] Collombier, D. (1995). *Plans d'expérience factoriels : Construction et propriétés des fractions de plans*. Springer Science & Business Media. 28

[Contal, 2016] Contal, E. (2016). *Statistical Learning Approaches for Global Optimization*. Thèse de doctorat, Université Paris-Saclay. 113

[Contal *et al.*, 2013] Contal, E., Buffoni, D., Robicquet, A. et Vayatis, N. (2013). Parallel Gaussian Process Optimization with Upper Confidence Bound and Pure Exploration. *In* Blockeel, H., Kersting, K., Nijssen, S. et Železný, F., éditeurs : *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pages 225–240. Springer Berlin Heidelberg. 110, 121

[Courard *et al.*, 2016] Courard, A., Néron, D., Ladevèze, P. et Ballere, L. (2016). Integration of PGD-virtual charts into an engineering design process. *Computational Mechanics*, 57(4):637–651. 120

[Courrier *et al.*, 2016] Courrier, N., Boucard, P.-A. et Soulier, B. (2016). Variable-fidelity modeling of structural analysis of assemblies. *Journal of Global Optimization*, 64(3): 577–613. 11, 59, 108

[Davidon, 1959] Davidon, W. C. (1959). Variable metric method for minimization. Rapport technique, ANL-5990 Rev., Argonne National Laboratories. 26

[De Lozzo, 2015] De Lozzo, M. (2015). Substitution de modèle et approche multifidélité en expérimentation numérique. *Journal de la Société Française de Statistique*, 156(4):21–55. 10

[Dempster, 2008] Dempster, A. P. (2008). Upper and lower probabilities induced by a multivalued mapping. *In Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 57–72. Springer. 52

[Descamps *et al.*, 1997] Descamps, J., Cao, H. L. et Potier-Ferry, M. (1997). An asymptotic numerical method to solve large strain viscoplastic problems. *Computational plasticity, Fundamentals and Applications, CIMNE*, 1:393–400. 47

[Desmorat *et al.*, 2018] Desmorat, R., Desmorat, B., Olive, M. et Kolev, B. (2018). Micromechanics based framework with second-order damage tensors. *European Journal of Mechanics - A/Solids*, 69:88–98. 65

[Di Pietro et Ern, 2012] Di Pietro, D. A. et Ern, A. (2012). *Mathematical Aspects of Discontinuous Galerkin Methods*, volume 69 de *Mathématiques et Applications*. Springer Berlin Heidelberg, Berlin, Heidelberg. 74

[Dréo, 2004] Dréo, J. (2004). *Adaptation de La Méthode Des Colonies de Fourmis Pour l'optimisation En Variables Continues. Application En Génie Biomedical*. Thèse de doctorat, Université Paris XII - Val de Marne. 27

[Dubois et Prade, 2012] Dubois, D. et Prade, H. (2012). *Possibility Theory*. Springer. 52

[Duda *et al.*, 2000] Duda, R. O., Hart, P. E. et Stork, D. G. (2000). *Pattern Classification (2Nd Edition)*. Wiley-Interscience, New York, NY, USA. 33

[Duysinx, 1996] Duysinx, P. (1996). Introduction à l'optimisation topologique. Rapp. tech. OF-43. LTAS, Université de Liége. 8

[Efron *et al.*, 2004] Efron, B., Hastie, T., Johnstone, I. et Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32(2):407–499. 32

[Fang *et al.*, 2005] Fang, K.-T., Li, R. et Sudjianto, A. (2005). *Design and Modeling for Computer Experiments*. CRC Press. 25

[Fernández-Godino *et al.*, 2016] Fernández-Godino, M. G., Park, C., Kim, N.-H. et Haftka, R. T. (2016). Review of multi-fidelity models. *arXiv :1609.07196 [stat]*. 59

[Fink et Rheinboldt, 1983] Fink, J. P. et Rheinboldt, W. C. (1983). On the Error Behavior of the Reduced Basis Technique for Nonlinear Finite Element Approximations. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 63(1):21–28. 45

[Fletcher, 1970] Fletcher, R. (1970). A New Approach to Variable Metric Algorithms. *Computer Journal*, 13(3):317–322. 26

[Fletcher et Powell, 1963] Fletcher, R. et Powell, M. J. D. (1963). A rapidly Convergent Descent Method for Minimization. *Computer Journal*, 6(2):163–168. 26

[Forrester *et al.*, 2006] Forrester, A. I., Bressloff, N. W. et Keane, A. J. (2006). Optimization using surrogate models and partially converged computational fluid dynamics simulations. *Proceedings of the Royal Society A : Mathematical, Physical and Engineering Sciences*, 462(2071):2177–2204. 11, 58, 59

[Forrester *et al.*, 2007] Forrester, A. I. J., Sóbester, A. et Keane, A. J. (2007). Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society of London A : Mathematical, Physical and Engineering Sciences*, 463(2088):3251–3269. 54

[Freund et Schapire, 1997] Freund, Y. et Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139. 32

[Gaston *et al.*, 2009] Gaston, D., Newman, C., Hansen, G. et Lebrun-Grandié, D. (2009). MOOSE : A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*, 239(10):1768–1778. 80

[Ghnatios *et al.*, 2012] Ghnatios, C., Masson, F., Huerta, A., Leygue, A., Cueto, E. et Chinesta, F. (2012). Proper Generalized Decomposition based dynamic data-driven control of thermal processes. *Computer Methods in Applied Mechanics and Engineering*, 213-216:29–41. 45

[Gneiting *et al.*, 2010] Gneiting, T., Kleiber, W. et Schlather, M. (2010). Matérn Cross-Covariance Functions for Multivariate Random Fields. *Journal of the American Statistical Association*, 105(491):1167–1177. 53, 54

[Goldberg, 1989] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA. 27

[Goldfarb, 1970] Goldfarb, D. (1970). A Family of Variable Metric Updates Derived by Variational Means. *Mathematics of Computation*, 24(109):23–26. 26

[Golub et Van Loan, 1983] Golub, G. H. et Van Loan, C. F. (1983). *Matrix Computations*, volume 3. JHU press. 44

[Gratiet, 2012] Gratiet, L. L. (2012). Recursive co-kriging model for Design of Computer experiments with multiple levels of fidelity with an application to hydrodynamic. *arXiv preprint arXiv :1210.0686*. 55

[Gunn *et al.*, 1998] Gunn, S. R. *et al.* (1998). Support vector machines for classification and regression. *ISIS technical report*, 14:85–86. 32

[Halton, 1964] Halton, J. H. (1964). Algorithm 247 : Radical-inverse Quasi-random Point Sequence. *Commun. ACM*, 7(12):701–702. 29

[Han *et al.*, 2012] Han, Z., Zimmerman, R. et Görtz, S. (2012). Alternative Cokriging Method for Variable-Fidelity Surrogate Modeling. *AIAA Journal*, 50(5):1205–1210. 11

[Han et Görtz, 2012] Han, Z.-H. et Görtz, S. (2012). Hierarchical kriging model for variable-fidelity surrogate modeling. *AIAA journal*, 50(9):1885–1896. 57

[Hebb, 2005] Hebb, D. O. (2005). *The Organization of Behavior : A Neuropsychological Theory*. Psychology Press. 10

[Heyberger *et al.*, 2013] Heyberger, C., Boucard, P. A. et Néron, D. (2013). A rational strategy for the resolution of parametrized problems in the PGD framework. *Computer Methods in Applied Mechanics and Engineering*, 259:40–49. 45

[Hoerl et Kennard, 1970] Hoerl, A. E. et Kennard, R. W. (1970). Ridge regression : Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67. 32

[Holland, 1975] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The University of Michigan Press. 27

[Huang *et al.*, 2006] Huang, D., Allen, T. T., Notz, W. I. et Zeng, N. (2006). Global optimization of stochastic black-box via sequential kriging meta-models. *Journal of Global Optimization*, 34(3):441–466. 112

[Hughes *et al.*, 2005] Hughes, T., Cottrell, J. et Bazilevs., Y. (2005). Isogeometric analysis : CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4135–4195. 120

[Hutter *et al.*, 2011] Hutter, F., Hoos, H. H. et Leyton-Brown, K. (2011). Sequential Model-Based Optimization for General Algorithm Configuration. *In* Coello, C. A. C., éditeur : *Learning and Intelligent Optimization*, Lecture Notes in Computer Science, pages 507–523. Springer Berlin Heidelberg. 39, 110

[Janusevskis *et al.*, 2012a] Janusevskis, J., Le Riche, R., Ginsbourger, D. et Girdziusas, R. (2012a). Expected Improvements for the Asynchronous Parallel Global Optimization of Expensive Functions : Potentials and Challenges. *In* Hamadi, Y. et Schoenauer, M., éditeurs : *Learning and Intelligent Optimization*, Lecture Notes in Computer Science, pages 413–418. Springer Berlin Heidelberg. 110

[Janusevskis *et al.*, 2012b] Janusevskis, J., Riche, R. L., Ginsbourger, D. et Girdziusas, R. (2012b). Expected Improvements for the Asynchronous Parallel Global Optimization of Expensive Functions : Potentials and Challenges. *In Learning and Intelligent Optimization*, Lecture Notes in Computer Science, pages 413–418. Springer, Berlin, Heidelberg. 121

[Jolliffe, 2002] Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer Series in Statistics. Springer-Verlag, New York, 2 édition. 44

[Jones *et al.*, 1998] Jones, D. R., Schonlau, M. et Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492. 27, 110

[Joyot *et al.*, 2016] Joyot, P., Bur, N., Ghnatios, C., Villon, P., Cueto, E. et Chinesta, F. (2016). On the use of model order reduction for simulating automated fibre placement processes. *Advanced Modeling and Simulation in Engineering Sciences*, 3(1):4. 45

[Kachanov, 1993] Kachanov, M. (1993). Elastic Solids with Many Cracks and Related Problems. *In* Hutchinson, J. W. et Wu, T. Y., éditeurs : *Advances in Applied Mechanics*, volume 30, pages 259–445. Elsevier. 65

[Kandasamy *et al.*, 2016] Kandasamy, K., Dasarathy, G., Oliva, J., Schneider, J. et Póczos, B. (2016). Gaussian Process Bandit Optimisation with Multi-fidelity Evaluations. page 23. 113

[Kandasamy *et al.*, 2017] Kandasamy, K., Krishnamurthy, A., Schneider, J. et Poczos, B. (2017). Asynchronous Parallel Bayesian Optimisation via Thompson Sampling. *arXiv :1705.09236 [cs, stat]*. 121

[Karhunen, 1947] Karhunen, K. (1947). *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*. Thèse de doctorat, [Kirjapaino oy. sana], Helsinki. OCLC : 8986938. 44

[Kennedy et Eberhart, 1995] Kennedy, J. et Eberhart, R. C. (1995). Particle swarm optimization. *In Proceedings of IEEE International Conference on Neural Networks IV*, pages 1942–1948. Piscataway : IEEE. 27

[Kennedy et O'Hagan, 2000] Kennedy, M. C. et O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13. 54

[Kirby et Logg, 2006] Kirby, R. C. et Logg, A. (2006). A compiler for variational forms. *ACM Transactions on Mathematical Software*, 32(3):417–444. 121

[Kleijnen et van Beers, 2004] Kleijnen, J. P. C. et van Beers, W. C. M. (2004). Application-driven sequential designs for simulation experiments : Kriging metamodelling. *Journal of the Operational Research Society*, 55(8):876–883. 106

[Korelc, 1997] Korelc, J. (1997). Automatic generation of finite-element code by simultaneous optimization of expressions. *Theoretical Computer Science*, 187(1):231–248. 121

[Krige, 1951] Krige, D. G. (1951). A statistical approach to some mine valuation and allied problems on the Witwatersrand : By DG Krige. 29, 33

[Kushner, 1964] Kushner, H. J. (1964). A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Basic Engineering*, 86(1):97–106. 27, 111

[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. 9, 21, 45, 61

[Ladevèze, 1999] Ladevèze, P. (1999). *Nonlinear Computational Structural Mechanics : New Approaches and Non-Incremental Methods of Calculation*. Mechanical Engineering Series. Springer Verlag. 21, 43

[Ladevèze *et al.*, 2002] Ladevèze, P., Nouy, A. et Loiseau, O. (2002). A multiscale computational approach for contact problems. *Computer Methods in Applied Mechanics and Engineering*, 191(43):4869–4891. 21

[Lancaster et Salkauskas, 1981] Lancaster, P. et Salkauskas, K. (1981). Surfaces generated by moving least squares methods. *Mathematics of computation*, 37(155):141–158. 32

[Le Gratiet, 2013a] Le Gratiet, L. (2013a). Bayesian Analysis of Hierarchical Multifidelity Codes. *SIAM/ASA Journal on Uncertainty Quantification*, 1(1):244–269. 54

[Le Gratiet, 2013b] Le Gratiet, L. (2013b). Recursive co-kriging model for Design of Computer experiments with multiple levels of fidelity with an application to hydrodynamic. 56

[Lemaitre et Chaboche, 1994] Lemaitre, J. et Chaboche, J.-L. (1994). *Mechanics of Solid Materials*. Cambridge university press. 17

[Lemoussu *et al.*, 2002] Lemoussu, H., Boucard, P. A. et Ladevèze, P. (2002). A 3D shock computational strategy for real assembly and shock attenuator. *Advances in Engineering Software*, 33(7):517–526. 21

[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(1-2):191–207. 26

[Li *et al.*, 2018] Li, H., Garg, V. V. et Willcox, K. (2018). Model adaptivity for goal-oriented inference using adjoints. *Computer Methods in Applied Mechanics and Engineering*, 331:1–22. 121

[Lieu *et al.*, 2006] Lieu, T., Farhat, C. et Lesoinne, M. (2006). Reduced-order fluid/structure modeling of a complete aircraft configuration. *Computer Methods in Applied Mechanics and Engineering*, 195(41):5730–5742. 9

[Liggins II *et al.*, 2017] Liggins II, M., Hall, D. et Llinas, J. (2017). *Handbook of Multisensor Data Fusion : Theory and Practice*. CRC press. 51

[Lin, 1989] Lin, L. I.-K. (1989). A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, 45:255–268. 39

[Lindley, 1956] Lindley, D. V. (1956). On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, pages 986–1005. 28

[Loeve, 1948] Loeve, M. (1948). *Probability Theory II*. Springer Science & Business Media. 44

[Lopez-Caballero et Khalil, 2018] Lopez-Caballero, F. et Khalil, C. (2018). Vulnerability Assessment for Earthquake Liquefaction–Induced Settlements of an Embankment Using Gaussian Processes. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A : Civil Engineering*, 4(2):04018010. 8

[Łukaszyk, 2004] Łukaszyk, S. (2004). A new concept of probability metric and its applications in approximation of scattered data sets. *Computational Mechanics*, 33(4):299–304. 89

[Maday et Ronquist, 2004] Maday, Y. et Ronquist, E. (2004). The Reduced Basis Element Method : Application to a Thermal Fin Problem. *SIAM Journal on Scientific Computing*, 26(1):240–258. 9, 45

[Marchand et Quilici, 2019] Marchand, B. et Quilici, S. (2019). Préconditionneur grille grossière pour la résolution par solveur ité- ratif de problème éléments-finis fortement non-linéaires de grandes tailles. *In 14e colloque national en calcul des structures*, page 8, Giens, France. CSMA. 121

[Matheron, 1963] Matheron, G. (1963). Principles of geostatistics. *Economic geology*, 58(8): 1246–1266. 10, 33

[McCulloch et Pitts, 1943] McCulloch, W. S. et Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133. 10, 32

[Metropolis et Ulam, 1949] Metropolis, N. et Ulam, S. (1949). The monte carlo method. *Journal of the American statistical association*, 44(247):335–341. 28

[Meza, 2018] Meza, R. M. (2018). Interpolation sur les variétés grassmanniennes et applications à la réduction de modèles en mécanique. 87

[Miller *et al.*, 2012] Miller, R. L., Harding, L. B., Davis, M. J. et Gray, S. K. (2012). Bi-fidelity fitting and optimization. *The Journal of Chemical Physics*, 136. 54

[Močkus, 1975] Močkus, J. (1975). On Bayesian Methods for Seeking the Extremum. *In* Marchuk, G. I., éditeur : *Optimization Techniques IFIP Technical Conference : Novosibirsk, July 1–7, 1974*, Lecture Notes in Computer Science, pages 400–404. Springer Berlin Heidelberg, Berlin, Heidelberg. 110

[Mockus *et al.*, 1978] Mockus, J., Tiesis, V. et Zilinskas, A. (1978). The application of Bayesian methods for seeking the extremum. *Towards global optimization*, 2(117-129):2. 111

[Moore, 1920] Moore, E. H. (1920). On the reciprocal of the general algebraic matrix. *Bull. Am. Math. Soc.*, 26:394–395. 65

[Myers, 1982] Myers, D. E. (1982). Matrix Fomulation of Co-Kriging. *Mathematical Geology*, 14(3). 32

[Nealen, 2004] Nealen, A. (2004). An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation. *URL : http ://www. nealen. com/projects*, 130(150):25. 32

[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. 10, 99

[Nguyen, 2012] Nguyen, T. L. (2012). *La décomposition propre generalisée pour la résolution de problèmes multiphysiques transitolres couples dédiés à la mécanique des matériaux - maillage adaptatif et couplage avec la MAN.* Thèse de doctorat, ISAE-ENSMA Ecole Nationale Supérieure de Mécanique et d'Aérotechique - Poitiers. 47

[Nickerson, 1997] Nickerson, C. A. E. (1997). A Note on "A Concordance Correlation Coefficient to Evaluate Reproducibility". *Biometrics*, 53:1503–1507. 39

[Nocedal et Wright, 2006] Nocedal, J. et Wright, S. J. (2006). *Numerical Optimization.* Springer, 2nd édition. 25

[Noël *et al.*, 2016] Noël, L., Miegroet, L. V. et Duysinx, P. (2016). Analytical sensitivity analysis using the extended finite element method in shape optimization of bimaterial structures. *International Journal for Numerical Methods in Engineering*, 107(8):669–695. 120

[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):4521–4537. 45

[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(23-24):1603–1626. 43, 46

[O'Hagan, 1998] O'Hagan, A. (1998). *A Markov Property for Covariance Structures*. 55

[Paillet *et al.*, 2018] Paillet, C., Néron, D. et Ladevèze, P. (2018). A door to model reduction in high-dimensional parameter space. *Comptes Rendus Mécanique*, 346(7):524–531. 45

[Park *et al.*, 2017] Park, C., Haftka, R. T. et Kim, N. H. (2017). Remarks on multi-fidelity surrogates. *Structural and Multidisciplinary Optimization*, 55(3):1029–1050. 59

[Passieux, 2008] Passieux, J.-C. (2008). *Approximation radiale et méthode LATIN multiéchelle en temps et en espace*. Thèse de doctorat, École normale supérieure de Cachan - ENS Cachan. 11, 61, 71, 74, 76

[Passieux et Périé, 2012] Passieux, J.-C. et Périé, J.-N. (2012). High resolution digital image correlation using proper generalized decomposition : PGD-DIC. *International Journal for Numerical Methods in Engineering*, 92(6):531–550. 45

[Pearson, 1901] Pearson, K. (1901). LIII. 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. 44

[Penrose, 1955] Penrose, R. (1955). A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413. 65

[Picheny *et al.*, 2013] Picheny, V., Wagner, T. et Ginsbourger, D. (2013). A benchmark of kriging-based infill criteria for noisy optimization. *Structural and Multidisciplinary Optimization*, 48(3):607–626. 113, 120

[Pronzato et Müller, 2012] Pronzato, L. et Müller, W. G. (2012). Design of computer experiments : Space filling and beyond. *Statistics and Computing*, 22(3):681–701. 28

[Rasmussen et Williams, 2006] Rasmussen, C. E. et Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass. OCLC : ocm61285753. 10, 33, 39

[Relun, 2011] Relun, N. (2011). *Stratégie multiparamétrique pour la conception robuste en fatigue*. Thèse de doctorat, École normale supérieure de Cachan - ENS Cachan. 62, 67, 71

[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. 11, 61, 79

[Riedel, 1992] Riedel, K. (1992). A Sherman–Morrison–Woodbury Identity for Rank Augmenting Matrices with Application to Centering. *SIAM Journal on Matrix Analysis and Applications*, 13(2):659–662. 65

[Robbins et Monro, 1951] Robbins, H. et Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407. 26

[Robert, 2007] Robert, C. (2007). *The Bayesian Choice : From Decision-Theoretic Foundations to Computational Implementation*. Springer Science & Business Media. 36

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386. 32

[Rozza *et al.*, 2007] Rozza, G., Huynh, D. B. P. et Patera, A. T. (2007). Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):1. 45

[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. 9

[Ryckelynck *et al.*, 2006] Ryckelynck, D., Chinesta, F., Cueto, E. et Ammar, A. (2006). On thea priori model reduction : Overview and recent developments. *Archives of Computational Methods in Engineering*, 13(1):91–128. 9, 47

[Sacks *et al.*, 1989] Sacks, J., Welch, W. J., Mitchell, T. J. et Wynn, H. P. (1989). Design and Analysis of Computer Experiments. *Statistical Science*, 4(4):409–423. 106

[Scanff *et al.*, 2019] Scanff, R., Néron, D. et Ladevèze, P. (2019). Sur une vision non intrusive de la méthode LaTIn-PGD en non linéaire. *In 24e Congrès Français de Mécanique*, Brest, France. AFM. 121

[Schmidt *et al.*, 2011] Schmidt, F., Pirc, N., Mongeau, M. et Chinesta, F. (2011). Efficient mold cooling optimization by using model reduction. *International Journal of Material Forming*, 4(1):73–82. 45

[Scott *et al.*, 2011] Scott, W., Frazier, P. et Powell, W. (2011). The Correlated Knowledge Gradient for Simulation Optimization of Continuous Parameters using Gaussian Process Regression. *SIAM Journal on Optimization*, 21(3):996–1026. 112

[Shanno, 1970] Shanno, D. F. (1970). Conditioning of Quasi-Newton Methods for Function Minimization. *Mathematics of Computation*, 24(111):647–656. 26

[Shepard, 1968] Shepard, D. (1968). A Two-dimensional Interpolation Function for Irregularly-spaced Data. *In Proceedings of the 1968 23rd ACM National Conference*, ACM '68, pages 517–524, New York, NY, USA. ACM. 89

[Signorini, 1943] Signorini, A. (1943). Trasformazioni termoelastiche finite. *Annali di Matematica Pura ed Applicata*, 22(1):33–143. 47

[Simo et Taylor, 1986] Simo, J. C. et Taylor, R. L. (1986). A return mapping algorithm for plane stress elastoplasticity. *International Journal for Numerical Methods in Engineering*, 22(3):649–670. 63

[Sóbester *et al.*, 2005] Sóbester, A., Leary, S. J. et Keane, A. J. (2005). On the Design of Optimization Strategies Based on Global Response Surface Approximation Models. *Journal of Global Optimization*, 33(1):31–59. 111

[Sobol, 1967] Sobol, I. M. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112. 29

[Srinivas *et al.*, 2012] Srinivas, N., Krause, A., Kakade, S. M. et Seeger, M. (2012). Gaussian Process Optimization in the Bandit Setting : No Regret and Experimental Design. *IEEE Transactions on Information Theory*, 58(5):3250–3265. 112

[Sudret, 2012] Sudret, B. (2012). Meta-models for structural reliability and uncertainty quantification. *arXiv :1203.2062 [stat]*. 8

[Swersky *et al.*, 2013] Swersky, K., Snoek, J. et Adams, R. P. (2013). Multi-Task Bayesian Optimization. *In* Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z. et Weinberger, K. Q., éditeurs : *Advances in Neural Information Processing Systems 26*, pages 2004–2012. Curran Associates, Inc. 110

[Tibshirani, 1996] Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288. 32

[Tirvaudey *et al.*, 2019a] Tirvaudey, M., Bouclier, R., Passieux, J.-C. et Chamoin, L. (2019a). Non-invasive implementation of nonlinear Isogeometric Analysis in an industrial FE software. *Engineering Computations*, In press. 120

[Tirvaudey *et al.*, 2019b] Tirvaudey, M., Chamoin, L., Bouclier, R. et Passieux, J.-C. (2019b). A posteriori error estimation and adaptivity in non-intrusive couplings between concurrent models. *Computer Methods in Applied Mechanics and Engineering*, submitted. 121

[Van Aken *et al.*, 2017] Van Aken, D., Pavlo, A., Gordon, G. J. et Zhang, B. (2017). Automatic Database Management System Tuning Through Large-scale Machine Learning. *In Proceedings of the 2017 ACM International Conference on Management of Data - SIGMOD '17*, pages 1009–1024, Chicago, Illinois, USA. ACM Press. 110

[Vapnik, 1963] Vapnik, V. (1963). Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24:774–780. 33

[Vapnik *et al.*, 1996] Vapnik, V., Golowich, S. E. et Smola, A. (1996). Support vector method for function approximation, regression estimation, and signal processing. *Advances in neural information processing systems 9*. 10, 32, 33

[Vasilache *et al.*, 2018] Vasilache, N., Zinenko, O., Theodoridis, T., Goyal, P., DeVito, Z., Moses, W. S., Verdoolaege, S., Adams, A. et Cohen, A. (2018). Tensor Comprehensions : Framework-Agnostic High-Performance Machine Learning Abstractions. *arXiv :1802.04730 [cs]*. 80

[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. 79, 121

[Wang *et al.*, 2012] Wang, Z., Zoghi, M., Hutter, F., Matheson, D. et de Freitas, N. (2012). Bayesian Optimization in High Dimensions via Random Embeddings. page 7. 110

[Wright, 1996] Wright, M. H. (1996). Direct search methods : Once scorned, now respectable. *In* Griffiths, D. et Watson, G., éditeurs : *Numerical Analysis : Proceedings of the 1995 Dundee Biennial Conference in Numerical Analysis*, pages 191–208. Addison Wesley Logman (Harlow), U.K. 26

[Zimmerman et Holland, 2005] Zimmerman, D. L. et Holland, D. M. (2005). Complementary co-kriging : Spatial prediction using data combined from several environmental monitoring networks. *Environmetrics*, 16:219–234. 11

[Zimmermann, 2013] Zimmermann, R. (2013). Gradient-enhanced surrogate modeling based on proper orthogonal decomposition. *Journal of Computational and Mathematics*, 237:403–418. 43

[Zimmermann et Han, 2010] Zimmermann, R. et Han, Z.-H. (2010). Simplified Cross-Correlation Estimation for Multi-Fidelity Surrogate Cokriging Models. *Advances and Applications in Mathematical Sciences*, 7(2):181–201. 54

[Zou et Hastie, 2017] Zou, H. et Hastie, T. (2017). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, pages 301–320. 32

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMAG)

Titre : Optimisation de structures viscoplastiques par couplage entre métamodèle multi-fidélité et modèles réduits

Mots clés : Optimisation, Métamodèle, Réduction de modèles, Multi-fidélité, Viscoplasticité

Résumé : Les phases de conception et de validation de pièces mécaniques nécessitent des outils de calculs rapides et fiables, permettant de faire des choix technologiques en un temps court. Dans ce cadre, il n'est pas possible de calculer la réponse exacte pour l'ensemble des configurations envisageables. Les métamodèles sont alors couramment utilisés mais nécessitent un grand nombre de réponses, notamment dans le cas où celles-ci sont non-linéaires. Une solution est alors d'exploiter plusieurs sources de données de qualité diverses pour générer un métamodèle multi-fidélité plus rapide à calculer pour une précision équivalente. Ces données multi-fidélité peuvent être extraites de modèles réduits. Les travaux présentés proposent une méthode de génération de métamodèles multi-fidélité pour l'optimisation de structures mécaniques par la mise en place d'une stratégie d'enrichissement adaptatif des informations sur la réponse de la structure, par utilisation de données issues d'un solveur LATIN-PGD permettant de générer des données de qualités adaptées, et d'accélérer le calcul par la réutilisation des données précédemment calculées. Un grand nombre de données basse-fidélité sont calculées avant un enrichissement intelligent par des données haute-fidélité. Ce manuscrit présente les contributions aux métamodèles multi-fidélité et deux approches de la méthode LATIN-PGD avec la mise en place d'une stratégie multi-paramétrique pour le réemploi des données précédemment calculées. Une implémentation parallèle des méthodes a permis de tester la méthode sur trois cas-tests, pour des gains pouvant aller jusqu'à $37\times$.

Title : Structural design optimization by coupling multi-fidelity metamodels and reduced-order models

Keywords : Structural design, Surrogate Models, Reduced-Order Models, Multi-fidelity, Viscoplasticity

Abstract : Engineering simulation provides the best design products by allowing many design options to be quickly explored and tested, but fast-time-to-results requirement remains a critical factor to meet aggressive time-to-market requirements. In this context, using high-fidelity direct resolution solver is not suitable for (virtual) charts generation for engineering design and optimization. Metamodels are commonly considered to explore design options without computing every possibility, but if the behavior is nonlinear, a large amount of data is still required. A possibility is to use further data sources to generate a multi-fidelity surrogate model by using model reduction. Model reduction techniques constitute one of the tools to bypass the limited calculation budget by seeking a solution to a problem on a reduced order basis (ROB). The purpose of the present work is an online method for generating a multi-fidelity metamodel nourished by calculating the quantity of interest from the basis generated on-the-fly with the LATIN-PGD framework for elasto-viscoplastic problems. Low-fidelity fields are obtained by stopping the solver before convergence, and high-fidelity information is obtained with converged solution. In addition, the solver ability to reuse information from previously calculated PGD basis is exploited. This manuscript presents the contributions to multi-fidelity metamodels and the LATIN-PGD method with the implementation of a multi-parametric strategy. This coupling strategy was tested on three test cases for calculation time savings of more than $37\times$.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France