



Optimisation multidisciplinaire et réduction d'ordre de modèle

Gaspard Berthelin

► To cite this version:

Gaspard Berthelin. Optimisation multidisciplinaire et réduction d'ordre de modèle. Physique [physics]. ISAE - Institut Supérieur de l'Aéronautique et de l'Espace, 2022. Français. NNT: . tel-04012880

HAL Id: tel-04012880

<https://hal.science/tel-04012880>

Submitted on 3 Mar 2023

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

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



THÈSE

**En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE
Délivré par l'Institut Supérieur de l'Aéronautique et de l'Espace**

**Présentée et soutenue par
Gaspard BERTHELIN**

Le 22 juin 2022

Optimisation multidisciplinaire et réduction d'ordre de modèle

Ecole doctorale : **AA - Aéronautique, Astronautique**

Spécialité : **Mathématiques et Applications**

Unité de recherche :

ISAE-ONERA MOIS MODélisation et Ingénierie des Systèmes

Thèse dirigée par

Michel SALAÜN et Sylvain DUBREUIL

Jury

M. Michael KOKKOLARAS, Rapporteur

M. Rodolphe LE RICHE, Rapporteur

Mme Mathilde CHEVREUIL PLESSIS, Examinatrice

M. Michel SALAÜN, Directeur de thèse

M. Sylvain DUBREUIL, Co-directeur de thèse

Mme Nathalie BARTOLI, Co-encadrante de thèse

M. Christian GOGU, Co-encadrant de thèse

M. Pierre-Alain BOUCARD, Président

Table des matières

Table des matières	iii
Résumé	v
Abstract	vii
Remerciements	ix
Liste des figures	xi
Liste des tableaux	xvii
Introduction	1
1 État de l'art	9
1.1 Quelques notions en optimisation	11
1.2 Optimisation multidisciplinaire	27
1.3 Réduction d'ordre de modèle	51
1.4 Utilisation de la réduction d'ordre de modèle dans un contexte <i>multi-query</i>	61
1.5 Synthèse du chapitre	68
1.6 Références	69
2 Optimisation monodisciplinaire combinée à la réduction d'ordre de modèle.	83
2.1 Introduction	85
2.2 Approximation de la quantité d'intérêt	87
2.3 Estimation de l'erreur commise	90
2.4 Utilisation du modèle réduit dans le cadre de l'optimisation	101
2.5 Application	106
2.6 Bilan du Chapitre	116
2.7 Références	117
3 Optimisation multidisciplinaire combinée à la réduction d'ordre de modèle.	119
3.1 Introduction	121
3.2 Description de la résolution de la MDA par DPOD+I	125
3.3 Application sur un cas aéroélastique	146
3.4 Application sur un deuxième cas aéroélastique	164
3.5 Optimisation multidisciplinaire	175
3.6 Bilan du Chapitre	187
3.7 Références	189
Conclusion et perspectives	193

A	Annexes	I
A.1	Approximation de $\hat{f}^{obj}$ par Expansion par chaos polynomial, ou <i>Polynomial Chaos Expansion</i> (PCE)	I
A.2	Expansion Karhunen-Loève	I
A.3	Interpolation des termes de l'expansion <i>Karhunen-Loève</i> (KL) par des GPs .	II
A.4	Illustration de la résolution d'un problème Optimisation multidisciplinaire, ou <i>MultiDisciplinary Design Optimization</i> (MDO) par <i>Efficient Global Multidisciplinary Optimization</i> (EGMDO)	III
A.5	Références	VII
B	Liste des acronymes	IX

Résumé

Ces dernières décennies, les concepteurs avant-projet en aéronautique ont porté un intérêt particulier sur l'interaction entre les différents experts impliqués dans la modélisation des performances d'un avion. Ces experts utilisent des codes de simulation numériques dits disciplinaires pour modéliser leur discipline (par exemple l'aérodynamique, la structure ou la propulsion) et échangent des informations afin de prendre en compte l'influence des autres disciplines sur la leur. Le système couplé via ces informations est nommé analyse multidisciplinaire (MDA). La recherche de la meilleure configuration, nommée optimisation multidisciplinaire, consiste à identifier les variables de conception optimisant une fonction de performance calculée à partir de la solution de la MDA.

La résolution de ce problème d'optimisation soulève plusieurs défis. Premièrement, l'évaluation de chaque modèle disciplinaire est coûteuse et minimiser les appels à ces solveurs lors de l'optimisation représente un des principaux objectifs de la thèse. Deuxièmement, nous considérons que les informations échangées entre les solveurs disciplinaires sont de grande dimension (typiquement des champs discrétisés dont la dimension excède 1000). Finalement, la méthode de résolution doit être la plus générique possible : non intrusive afin de pouvoir utiliser des solveurs disciplinaires industriels et pouvant s'adapter à des MDA faisant intervenir un nombre variable de disciplines impliquées dans le processus de conception. Plusieurs formulations ont été proposées pour la résolution de ce type de problème mais aucune ne permet de relever entièrement ces défis. Une des propositions les plus prometteuses est de remplacer chacun des solveurs disciplinaires par des processus gaussiens (GP) et de coupler ces solveurs approchés pour obtenir une approximation de la MDA. L'idée est alors d'utiliser la variance des GP pour estimer les incertitudes sur la MDA et d'enrichir les GP lorsque cela s'avère nécessaire. Cependant, cette méthode ne peut être appliquée lorsque les informations échangées sont de grande dimension. Une autre voie de réduction du coût numérique est basée sur les méthodes de réduction d'ordre de modèle par projection. Pour cela, les sorties des modèles sont projetées sur une base réduite de faible dimension. Une des limitations de cette approximation est l'évaluation de l'erreur commise par la projection. L'objectif de la thèse est alors de coupler GP et réduction d'ordre de modèle afin de proposer une méthode de résolution d'un problème d'optimisation multidisciplinaire permettant de relever les défis proposés. Pour cela, l'idée est de remplacer chaque solveur disciplinaire par une approximation combinant projection et interpolation par GP. Cependant, afin de certifier que l'approximation est fidèle, l'erreur commise devra être estimée afin d'affiner l'approximation si celle-ci est trop importante.

Une première partie de nos travaux consiste à utiliser la projection de Petrov-Galerkin comme technique de réduction d'ordre de modèle et d'estimer l'erreur de projection par analyse statistique et préconditionnement. Une application est proposée sur l'optimisation d'une quantité d'intérêt via un couplage avec un algorithme bayésien. Plus précisément, l'estimation de l'erreur est utilisée pour déterminer si la solution réduite peut être utilisée à la place de la solution exacte. Ce couplage testé sur le dimensionnement d'une

aile d'avion montre un fort potentiel de réduction des temps de calcul.

Dans un second temps, nous avons utilisé les GP pour interpoler les coefficients de la base réduite et estimé l'erreur d'approximation grâce à la variance des GP. Une méthode d'entraînement spécifique au contexte multidisciplinaire est proposée pour la construction des bases réduites et des GP. Les erreurs induites par l'approximation de chaque solveur sont alors propagées afin d'obtenir une estimation de l'erreur sur la MDA. Un algorithme de résolution de MDA est proposé par enrichissement des GP disciplinaires et des bases réduites jusqu'à ce que les incertitudes liées à l'interpolation soient considérées comme suffisamment faibles. Cet algorithme a été testé dans un contexte aéroélastique modélisant une aile d'avion. Par la suite, plusieurs couplages avec des algorithmes de résolution d'optimisation multidisciplinaire ont montré l'intérêt de la méthode en permettant de fortement réduire le budget de l'optimisation.

Abstract

In recent decades, designers in aeronautics have been specifically interested in the interaction between the various experts involved in the modeling of the performance of aircraft. These experts use so-called disciplinary numerical solvers to model their discipline (for example aerodynamic, structure or propulsion) and exchange information to take into account the influence of other disciplines. The system coupled by this information is called multidisciplinary analysis (MDA). The search of the best design, called multidisciplinary optimization, consists in identifying the design variables optimizing a performance function derived from the MDA solution.

Solving this optimization problem raises several challenges. First, the evaluation of each disciplinary solver is numerically expensive and minimizing the calls to these solvers through the optimization represents one of the main objectives of the thesis. Secondly, it is assumed that the data exchanged by the disciplinary solvers is of high dimensionality (typically discretized field whose dimension exceeds 1000). Finally, the resolution method must be as generic as possible : non-intrusive in order to use industrial disciplinary solvers and adaptable to MDA involving various number of disciplines. Several formulations have been proposed for the resolution of this type of problem, but none of them can fully overcome these challenges. One of the most promising ideas is to replace each disciplinary solver by Gaussian processes (GP) and couple these surrogate models to approximate the MDA. The idea is then to use the variance of the GP to estimate the uncertainties on the MDA and to enrich the GP if necessary. However, this method cannot be applied when the dimension of the exchanged information is too high. Another way of reducing the numerical cost is based on model order reduction methods by projection. To do so, the outputs of the models are projected onto a reduced basis of lower dimension. One of the limitations of this approximation is the evaluation of the error made by the projection. The objective of the thesis is the coupling of GP and model order reduction to propose a method for solving a multidisciplinary optimization problem overcoming the proposed challenges. To this end, the idea is to replace each disciplinary solver by an approximation combining projection and interpolation by GP. However, in order to certify that the approximation is accurate, the error made shall be estimated in order to refine the approximation if needed.

A first part of our work consists in using the Petrov-Galerkin projection as a model order reduction technique and estimating the projection error by statistical analysis and preconditioning. An application is proposed on the optimization of a quantity of interest using Bayesian optimization. More precisely, the error is estimated to determine if the approximated solution shall be used instead of the exact solution. This coupling is tested on the preliminary design of an aircraft wing and shows a strong potential for reducing computational cost.

Secondly, we use GP to interpolate the coefficients on the reduced basis and estimate the error made using the variance of the GP. A training method for the construction of the reduced basis and the GP is proposed in the context of multidisciplinary analysis.

The errors caused by the approximation of each solver are then propagated to obtain an estimate of the error on the MDA. An MDA solver is proposed by enrichment of the disciplinary GP and reduced basis until the uncertainties caused by the interpolation are considered sufficiently low. This algorithm was tested in an aeroelastic context modeling an airplane wing. Subsequently, several couplings with multidisciplinary optimization algorithms showed the interest of the method by greatly lowering the optimization budget.

Remerciements

Je souhaite tout d'abord remercier mes directeurs et encadrants de thèse, Michel Salaün, Sylvain Dubreuil, Nathalie Bartoli et Christian Gogu pour m'avoir encadré durant ces presque quatre années. Merci de m'avoir soutenu et accompagné pour l'achèvement de cette thèse. Je remercie particulièrement Sylvain Dubreuil pour nos longues discussions sur tous les aspects scientifiques et pour tous ses conseils.

J'exprime ma gratitude à Michael Kokkolaras et Rodolphe Le Riche pour avoir accepté de rapporter ma thèse et à Pierre-Alain Boucard et Mathilde Chevreuil Plessis pour avoir accepté d'examiner ma thèse et pour leurs nombreuses questions pertinentes lors de la soutenance et des pré-rapports.

Je remercie également tous les membres de l'ISAE et de l'ONERA qui m'ont accueilli lors de ces trois années. Je pense plus particulièrement aux autres doctorants et post-doctorants avec qui j'ai partagé pauses cafés et parties de tarot. Ainsi je dis un grand merci en particulier à Rémy, Gabriel, Morgane, Maxime, Julien, Ines, Charles ainsi qu'à Pierre, Alessandro, Tiphaine, Vincent, Marco, Vincenzo, Marie, Paul, Camille, Florent et Samuel.

Je remercie tous mes amis qui m'ont aidé pendant cette thèse que ce soit en me permettant de m'évader le temps d'une soirée ou en me soutenant quand j'en ai eu le plus besoin. Je ne peux malheureusement pas citer tout le monde mais merci particulièrement à Louis, Margot, Florian, Thomas, Neige, Cyril, Adèle et Guillaume.

J'en profite aussi pour remercier le TUC badminton pour les échappées sportives.

Mes derniers remerciements vont à ma famille qui a tout fait pour m'aider, qui m'a soutenu et surtout supporté dans tout ce que j'ai entrepris.

Liste des figures

1	Exemples de nouveaux concepts d'avions.	2
2	Illustration de la MDA sur un problème avec deux disciplines : aérodynamique et structure où les variables de couplage sont $\mathbf{u}$ et $\mathbf{f}$	2
3	Illustration de la MDO sur un problème avec deux disciplines : aérodynamique et structure où les variables de couplage sont $\mathbf{u}$ et $\mathbf{f}$ et les variables de design sont représentées par le vecteur $\mathbf{x}$. Les variables de couplage convergées sont nommées $\mathbf{u}_*(\mathbf{x})$ et $\mathbf{f}_*(\mathbf{x})$ et permettent d'obtenir la fonction de performance f^{obj}	3
1.1	Illustration d'une régression par Processus gaussien, ou <i>Gaussian process</i> (GP) sur une fonction à une variable définie par FORRESTER et collab. [2008] où la moyenne μ du GP approche la fonction f avec un Plan d'expérience, ou <i>Design of Experiment</i> (DoE) composé de 5 points d'apprentissage. Un Intervalle de Confiance (IC) sur l'interpolation est calculé à partir de la variance du GP (en prenant $\mu(\mathbf{x}) \pm 3\sqrt{k(\mathbf{x}, \mathbf{x})}$).	20
1.2	Illustration de l'utilisation de méta-modèle pour l'optimisation globale sur une fonction à une variable définie par FORRESTER et collab. [2008].	22
1.3	Illustration de la non-convergence de l'optimisation globale par méta-modèle sur le même modèle mais avec des données d'apprentissage mal réparties.	22
1.4	Itérations de l'algorithme <i>Efficient Global Optimization</i> (EGO) sur un exemple 1D proposé par FORRESTER et collab. [2008] où l'amélioration espérée est maximisée pour obtenir le prochain point d'enrichissement. Après 6 itérations, le minimum global est identifié.	25
1.5	Illustration de la Analyse multidisciplinaire, ou <i>MultiDisciplinary Analysis</i> (MDA) faisant interagir deux disciplines et deux variables de couplage y^1 et y^2	28
1.6	Illustration de la FSI dans le cas de l'aéroélasticité où $\mathbf{u}_s$ est le déplacement de la structure, $\mathbf{f}_a$ sont les efforts aérodynamiques, $\mathcal{M}_a$ est le modèle aérodynamique et $\mathcal{M}_s$ est le modèle structure.	29
1.7	Illustration de la MDA pour le cas test analytique avec $x = -3$ où les deux solveurs disciplinaires sont définis par l'Eq. (1.33). Les deux réponses des solveurs disciplinaires sont illustrées en fonction des variables de couplage y^1 et y^2 . La solution de la MDA est représentée par une étoile rouge.	31
1.8	Illustration de la fonction objectif du cas test analytique obtenue pour x variant entre -5 et 5 . On peut observer que le minimum global de cette fonction est atteint en $x \approx -3$	34
1.9	Classification des principales formulations MDO	36
1.10	Illustration de la formulation MDF sur un système bi-disciplinaire.	37
1.11	Illustration de la formulation IDF sur un système bi-disciplinaire.	38

1.12	Illustration de la méthode EGMDO où la MDA déterministe est remplacée par une MDA aléatoire en substituant les deux solveurs disciplinaires coûteux par des approximations par GP.	39
1.13	Illustration des GP obtenus après entraînement sur le cas analytique présenté en Section 1.2.2.1. La moyenne et l'intervalle de confiance sont représentés ainsi que les disciplines exactes. La solution de la MDA est représentée en rouge, son approximation obtenue à partir de la moyenne des GP est représentée en orange.	40
1.14	Illustration de la génération de MDA aléatoires sur un cas avec uniquement deux disciplines. Les traits pleins violets et verts représentent la moyenne des GP, l'espace entre les traits pointillés est l'intervalle de confiance à 99%. Les traits fins vert clair, cyan et orange sont des trajectoires des GP parfaitement corrélés comme décrit dans l'Eq. (1.47). Les croix noires aux intersections des trajectoires sont des solutions de la MDA aléatoire.	42
1.15	Illustration de la fonction objectif aléatoire $\hat{f}^{obj}$ obtenue à partir des MDA aléatoires sur le cas test analytique. L'espérance et l'intervalle de confiance de $\hat{f}^{obj}$ sont obtenus en tout point de l'espace de design $\mathcal{X}$ à partir d'un échantillon de solutions de la MDA aléatoire de taille 2000.	43
1.16	Illustration du DoE de la fonction $\hat{f}^{obj}$ utilisé pour l'approximation de la fonction objectif aléatoire. L'espérance et l'intervalle de confiance de $\hat{f}^{obj}$ sont similaires à ceux de la Figure 1.15. 25 évaluations de $\hat{f}^{obj}$ sont représentées par des sphères bleues pour chaque point de DoE_{UQ} . L'estimation de la Densité de probabilité, ou <i>Probability Density Function</i> (PDF) de la fonction objectif aléatoire est obtenue à partir d'un échantillon de taille 2000.	44
1.17	Illustration de l'approximation f_x^{PCE} de $\hat{f}^{obj}$ par PCE entraînée sur les 5 points de DoE_{UQ} illustrés dans la Figure 1.16. 25 évaluations de f_x^{PCE} sont représentées par des sphères bleues pour chaque point de DoE_{UQ} et l'estimation de la PDF est obtenue à partir d'un échantillon de taille 1000. L'espérance et l'intervalle de confiance de $\hat{f}^{obj}$ sont similaires à ceux de la Figure 1.15.	44
1.18	Illustration de l'approximation $\tilde{f}^{obj}$ sur le cas analytique où la moyenne ainsi que l'intervalle de confiance à 99% sont apparents. $\tilde{f}^{obj}$ est entraînée sur le DoE illustré sur la Figure 1.16. La moyenne et l'intervalle de confiance sont obtenus à partir d'un échantillon de taille 10^7	45
1.19	Illustration de la résolution de la MDO en utilisant la méthode EGMDO.	48
2.1	Algorithme d'optimisation avec ou sans réduction de modèle	86
2.2	Histogrammes lissés de l'erreur relative pour l'approche primale, duale et primale-duale sur une échelle logarithmique obtenue sur un problème <i>Vortex Lattice Method</i> (VLM) paramétré.	89
2.3	Évolution du quantile de $ \cos $ à 90%, 95%, 99% et 99,9% en fonction de la dimension	93
2.4	Illustration de l'erreur commise sur la quantité d'intérêt $ s(x) - s_r^{\text{PD}}(x) $ et la borne de l'erreur sur un problème à une dimension. $\Delta_1(x)$ est la borne obtenue sans considérer le quantile ($q_k(1) = 1$).	93
2.5	Illustration du lien entre la norme de l'erreur $\ \mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})\ _2$ et la norme du résidu $\ \mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})\ _2$ préconditionné de 3 façons différentes : aucun, constant et obtenu par interpolation d'opérateurs. Ces calculs sont effectués sur le même cas 1D présenté dans la Section 2.3.1.	99

2.6	Illustration du cas où la réduction de modèle ne permet pas de conclure sur la non-optimalité de $\mathbf{x}_{\text{EGO}}$, un appel au solveur sera nécessaire pour calculer $s(\mathbf{x}_{\text{EGO}})$	103
2.7	Illustration du cas où la réduction de modèle permet d'assurer que $\mathbf{x}_{\text{EGO}}$ n'est pas optimal, $s_r(\mathbf{x}_{\text{EGO}})$ sera utilisé comme approximation de $s(\mathbf{x}_{\text{EGO}})$	104
2.8	Maillage de l'aile avec ou sans peau mettant en évidence les 3 sections et les différents composants : nervures (vert), longeron avant (gris), longeron arrière (rouge) et peau (bleu)	106
2.9	Histogrammes de $e_p(\mathbf{x}, \boldsymbol{\lambda}(\mathbf{x}))$ défini dans l'Eq. (2.67) pour le Préconditionnement par interpolation d'opérateurs (PIO) obtenu pour les différents coefficients ainsi que sans préconditionnement pour le cas 1D.	108
2.10	Histogrammes de $e_p(\boldsymbol{\lambda}(\mathbf{x}))$ défini dans l'Eq. (2.67) pour le PIO obtenu pour les différents coefficients ainsi que sans préconditionnement pour le cas 4D.	109
2.11	Histogrammes de $e_p(\boldsymbol{\lambda}(\mathbf{x}))$ défini dans l'Eq. (2.67) pour le PIO obtenu pour les différents coefficients ainsi que sans préconditionnement pour le cas 12D.	109
2.12	Temps de calcul des différentes stratégies en fonction de la dimension N du problème et pour 4 valeurs du taux d'échec p_s . Les calculs des différents coefficients pour l'interpolation sont détaillés dans les Sections 2.3.3.2 et 2.3.3.3.	111
3.2	Illustration de la FSI où $\mathbf{u}_s$ est le déplacement de la structure, $\mathbf{f}_a$ sont les efforts aérodynamiques, $\mathcal{M}_a$ est le modèle aérodynamique et $\mathcal{M}_s$ est le modèle structure.	123
3.3	Illustration de la méthode EGMDO où la MDA déterministe est remplacée par une MDA aléatoire en substituant les deux solveurs disciplinaires coûteux par des approximations par GP.	124
3.4	Illustration de la méthode mise en oeuvre pour l'interpolation des disciplines où la dimension des variables de couplage est réduite par POD.	124
3.5	Illustration des principales étapes de la méthodologie proposée. S_i correspond aux étapes de la méthode et O_i correspond aux sorties des différentes étapes.	126
3.6	Illustration de l'analyse multidisciplinaire réduite utilisant les variables de couplage projetées obtenues par Décomposition Orthogonale aux valeurs propres, ou <i>Proper Orthogonal Decomposition</i> (POD).	129
3.7	Distribution des entrées de chaque solveur disciplinaire lors de la résolution de 10 MDA associées à 10 paramètres de design. Seuls les premiers coefficients de la POD (α_1^1 and α_1^2) sont représentés. Chaque MDA est représentée par une couleur différente, la résolution de chacune des 10 MDA a nécessité quatre appels à chaque solveur disciplinaire.	131
3.8	Coefficients de la POD sur la base structure obtenus à partir des itérations nécessaires à la résolution de 200 MDA.	132
3.9	Distribution des premiers coefficients α_1^1 et α_1^2 des bases POD de chaque discipline et de la première variable de design x_1 donnée par l'algorithme d'entraînement contenant 50 solutions des solveurs disciplinaires.	135
3.10	Illustration de l'analyse multidisciplinaire réduite sur des bases POD limitées à un vecteur. Les solveurs disciplinaires réduits sont illustrés par des courbes en pointillé et la solution de l'analyse multidisciplinaire pour le paramètre $\mathbf{x}_*$ par une étoile rouge.	136

3.11 Illustration de l'analyse multidisciplinaire réduite sur des bases POD limitées à un vecteur. Les disciplines exactes sont illustrées en traits en pointillés et les GP disciplinaires sont illustrés avec des traits pleins.	137
3.12 Illustration de l'analyse multidisciplinaire réduite sur des bases POD limitées à un vecteur. Sur la gauche, trois solutions de la MDA aléatoire ainsi que les trajectoires des GP parfaitement corrélés associées. Sur la droite, un échantillon de 75 solutions de la MDA aléatoire.	138
3.13 Illustration de l'analyse multidisciplinaire réduite sur des bases POD limitées à un vecteur. Trois itérations de l'algorithme d'enrichissement sont effectuées : i) solutions obtenues après la phase d'entraînement décrite dans la Section 3.2.3 ii) MDA approchée après un enrichissement de la première discipline iii) MDA approchée après un nouvel enrichissement, cette fois de la deuxième discipline : le modèle approché est suffisamment précis.	140
3.14 Illustration de l'analyse multidisciplinaire réduite sur des bases POD limitées à un vecteur. Les disciplines exactes sont illustrées en traits en pointillés et les GP disciplinaires sont illustrés avec des traits pleins.	141
3.15 Maillage de la structure de l'aile sans les lisses et la peau supérieure.	148
3.16 Maillage aérodynamique de l'aile.	148
3.17 Illustration de la MDA sur un problème FSI décrit dans l'Eq (3.23) où les variables de couplage sont transférées grâce à la matrice de transfert $\mathbf{H}$	148
3.18 Illustration des amplitudes des vortex résultant du calcul aérodynamique et déplacements obtenus à partir du solveur structure en trois points particuliers : le centre de l'espace paramétrique et les points du DoE minimisant ou maximisant le déplacement en bout d'aile.	150
3.19 Illustration des quatre premiers modes POD obtenus sur le solveur structure grâce à l'algorithme d'entraînement. Les nuances de couleur représentent les variations de la norme du déplacement, le bleu correspondant à un déplacement nul et le rouge à un déplacement maximal.	152
3.20 Illustration des quatre premiers modes POD obtenus sur le solveur aérodynamique grâce à l'algorithme d'entraînement. Les nuances de couleur correspondent aux variations des amplitudes des vortex.	152
3.21 Problème à 4 variables de design, histogramme de l'erreur relative entre la solution de la MDA et son approximation par POD : e_{DoE}^1 et e_{DoE}^2 sur l'ensemble de test.	153
3.22 Problème à 4 variables de design, histogramme et quantile à 90% de $\hat{q}$ au cours des itérations de l'Algorithme 13 lors de la résolution de la MDA $\mathbf{x} = \mathbf{x}_1^{DoE}$ avec le solveur DPOD+I.	153
3.23 Problème à 4 variables de design, nombre d'appels aux solveurs disciplinaires nécessaires à la résolution des 100 MDA de l'ensemble de test en utilisant le solveur DPOD+I.	154
3.24 Problème à 4 variables de design, histogramme de l'erreur relative entre la solution MDA exacte et la solution obtenue par le solveur DPOD+I sur l'ensemble de test.	154
3.25 Problème à 4 variables de design, histogramme de l'erreur relative sur la structure et l'aérodynamique sur l'ensemble de test avec l'approche DPOD+I et l'approche POD+I classique.	156
3.26 Problème à 4 variables de design, étude de robustesse. Nombre d'appels aux solveurs disciplinaires au cours des résolutions sur l'ensemble de test avec r fixé à 5, 10, 15 et 20.	159

3.27	Problème à 8 variables de design, histogramme de l'erreur relative entre la solution de la MDA et son approximation par POD : e_{DoE}^1 et e_{DoE}^2 sur l'ensemble de test.	160
3.28	Problème à 8 variables de design, histogramme et quantile à 90% de $\hat{q}$ au cours des itérations de l'Algorithme 13 lors de la résolution de la MDA $\mathbf{x} = \mathbf{x}_1^{DoE}$ avec le solveur DPOD+I.	161
3.29	Problème à 8 variables de design, nombre d'appels aux solveurs disciplinaires nécessaires à la résolution des 100 MDA de l'ensemble de test en utilisant le solveur DPOD+I.	161
3.30	Problème à 8 variables de design, histogramme de l'erreur relative entre la solution MDA exacte et la solution obtenue par le solveur DPOD+I sur l'ensemble de test.	162
3.31	Problème à 8 variables design, histogramme de l'erreur relative sur la structure et l'aérodynamique sur l'ensemble de test avec l'approche DPOD+I et l'approche POD+I classique.	163
3.32	Maillage extérieur de l'aile utilisé pour la résolution des équations d'Euler avec le solveur SU2.	166
3.33	Illustration de la pression (solution stationnaire) sur la surface supérieure de l'aile obtenue à partir de la résolution des équations d'Euler au centre de l'espace de design.	166
3.34	Illustration de la pression (solution stationnaire) sur la surface inférieure de l'aile obtenue à partir de la résolution des équations d'Euler au centre de l'espace de design.	167
3.35	Illustration de la résolution de la MDO en utilisant la méthode EGMDO avec des modèles DPOD+I. Les changements effectués par rapport à la méthode EGMDO sont affichées en vert.	179
A.1	Illustration de EGMDO sur un cas analytique présentant les MCS de $\hat{f}^{obj}(\mathbf{x}, \Xi)$ obtenu en tout point de DoE_{UQ} ainsi qu'une estimation de la PDF de $\hat{f}^{obj}(\mathbf{x}, \Xi)$. On notera que la loi de $\hat{f}^{obj}$ n'est effectivement pas gaussienne.	IV
A.2	Illustration de EGMDO sur un cas analytique présentant la fonction à optimiser $f(\mathbf{x}) = f^{obj}(\mathbf{x}, y_*^1(\mathbf{x}), y_*^2(\mathbf{x}))$ et les approximations par PCE $\hat{f}_x^{PCE}(\Xi)$, $\forall \mathbf{x} \in DoE_{UQ}$ dont une estimation de la PDF est générée.	IV
A.3	Illustration de EGMDO sur un cas analytique présentant la fonction à optimiser $f(\mathbf{x}) = f^{obj}(\mathbf{x}, y_*^1(\mathbf{x}), y_*^2(\mathbf{x}))$ et l'approximation du champ de la fonction objectif aléatoire $\tilde{f}^{obj}(\mathbf{x}, \Xi, \eta)$ dont la moyenne et l'intervalle de confiance sont représentés.	V
A.4	Illustration de EGMDO sur un cas analytique présentant l'amélioration espérée définie dans l'Eq. (1.53) ainsi que son maximum obtenu par optimisation numérique.	V
A.5	Illustration de EGMDO sur un cas analytique présentant la convergence de $P_{min}(\mathbf{x}_{(i)})$, $\mathbf{x}_{(i)} \in DoE_{UQ}$ grâce aux enrichissements successifs.	V
A.6	Illustration de EGMDO sur un cas analytique présentant les MCS de $\hat{f}^{obj}(\mathbf{x}, \Xi)$ obtenus en tout point de DoE_{UQ} ainsi qu'une estimation de la PDF de $\hat{f}^{obj}(\mathbf{x}, \Xi)$ générée à partir de la MCS.	VI
A.7	Illustration de EGMDO sur un cas analytique présentant la fonction à optimiser $f(\mathbf{x}) = f^{obj}(\mathbf{x}, y_*^1(\mathbf{x}), y_*^2(\mathbf{x}))$ et l'approximation du champ de la fonction objectif aléatoire $\tilde{f}^{obj}(\mathbf{x}, \Xi)$ dont la moyenne et l'intervalle de confiance sont représentés après trois itérations de l'Algorithme 7.	VI

A.8	Illustration de EGMDO sur un cas analytique présentant la fonction à optimiser $f(\mathbf{x}) = f^{obj}(\mathbf{x}, y_*^1(\mathbf{x}), y_*^2(\mathbf{x}))$ et le champ aléatoire sur lequel sera déterminé le minimum de la MDO défini en supprimant l'incertitude liée à l'interpolation de l'expansion KL : $\tilde{f}^{obj}(\mathbf{x}, \Xi, 0)$	VI
-----	---	----

Liste des tableaux

2.1	Cas test 1D. Moyenne et coefficient de variation des erreurs relatives sur le problème primal et dual obtenues par POD sur les 50 itérations de l’Algorithme 10.	112
2.2	Cas test 1D. Moyenne et coefficient de variation de l’erreur relative sur la quantité d’intérêt ainsi que l’évaluation de la performance des bornes de l’erreur définie dans l’Eq. (2.68) sur les 50 itérations de l’Algorithme 10. . . .	112
2.3	Cas test 4D. Moyenne et coefficient de variation des erreurs relatives sur le problème primal et dual obtenues par POD sur les 150 itérations de l’Algorithme 10.	113
2.4	Cas test 4D. Moyenne et coefficient de variation de l’erreur relative sur la quantité d’intérêt ainsi que l’évaluation de la performance des bornes de l’erreur définie dans l’Eq. (2.68) sur les 150 itérations de l’Algorithme 10. . .	113
2.5	Cas test 12D. Moyenne et coefficient de variation des erreurs relatives sur le problème primal obtenues par POD en fonction du nombre d’enrichissements de la base effectués lors des 300 itérations de l’Algorithme 10.	114
2.6	Cas test 12D. Moyenne et coefficient de variation des erreurs relatives sur le problème dual obtenues par POD en fonction du nombre d’enrichissements de la base effectués lors des 300 itérations de l’Algorithme 10.	114
2.7	Cas test 12D. Moyenne de l’erreur relative sur la quantité d’intérêt ainsi que l’évaluation de la performance des bornes de l’erreur définie dans l’Eq. (2.68) sur les 300 itérations de chaque réalisation de l’Algorithme 10.	114
2.8	Cas test 12D. Nombre d’appels au solveur complet et p_s correspondant. . .	115
2.9	Cas test 12D. Nombre d’appels au solveur complet et p_s correspondant en prenant en compte la contrainte.	115
2.10	Cas test 12D. Solutions du problème d’optimisation en utilisant l’Algorithme 10 (couplant EGO et MOR) et en utilisant l’algorithme EGO sans réduction d’ordre de modèle.	115
2.11	Cas test 12D. Solutions du problème d’optimisation en utilisant l’Algorithme 10 (couplant EGO et MOR) et erreur relative avec la référence.	116
2.12	Cas test 12D. Solutions du problème d’optimisation en utilisant l’algorithme EGO sans réduction d’ordre de modèle et erreur relative avec la référence. .	116
3.1	Problème à 4 paramètres comprenant l’angle d’incidence, ou <i>Angle of Incidence</i> (AoI), la vitesse et les épaisseurs (les paramètres sont normalisés entre 0 et 1).	149
3.2	Problème à 8 paramètres comprenant l’AoI, la vitesse et les épaisseurs (les paramètres sont normalisés entre 0 et 1).	149

3.3	Problème à 4 variables de design, minimum, maximum et moyenne de l'erreur relative pour la structure en utilisant la DPOD+I et la POD+I classique sur l'ensemble de test.	155
3.4	Problème à 4 variables design, minimum, maximum et moyenne de l'erreur relative pour l'aérodynamique en utilisant la DPOD+I et la POD+I classique sur l'ensemble de test.	155
3.5	Problème à 4 variables de design, nombre d'appels aux solveurs disciplinaires nécessaires à la résolution des MDA sur l'ensemble de test pour le solveur MDA classique, l'approche DPOD+I et l'approche POD+I enrichie.	156
3.6	Problème à 4 variables de design, influence de r sur le coût de l'algorithme d'entraînement : moyenne et Coefficient de Variation (CV) des appels aux solveurs disciplinaires sur 10 réalisations.	157
3.7	Problème à 4 variables de design, influence de r sur le coût de l'algorithme d'entraînement : nombre moyen de modes POD conservés et erreurs relatives de projection sur l'ensemble de test lors des 10 réalisations.	157
3.8	Problème à 4 variables de design, influence de r sur le coût de calcul total : nombre moyen d'appels et Coefficient de Variation (CV) pour chaque solveur disciplinaire après les phases d'entraînement et d'enrichissement sur l'ensemble de test en utilisant le solveur Décomposition orthogonale aux valeurs propres et interpolation disciplinaire <i>Disciplinary Proper Orthogonal Decomposition and Interpolation</i> (DPOD+I) sur les 10 réalisations.	158
3.9	Problème à 4 variables de design, influence de r sur la précision du solveur DPOD+I : erreur relative moyenne sur les deux disciplines et CV associé sur l'ensemble de tests sur les 10 réalisations.	158
3.10	Problème à 8 variables design, minimum, maximum et moyenne de l'erreur relative pour la structure en utilisant la DPOD+I et la POD+I classique sur l'ensemble de test.	162
3.11	Problème à 8 variables de design, minimum, maximum et moyenne de l'erreur relative pour l'aérodynamique en utilisant la DPOD+I et la POD+I classique sur l'ensemble de test.	163
3.12	Problème à 8 variables de design, nombre d'appels aux solveurs disciplinaires nécessaires à la résolution des MDA sur l'ensemble de test pour le solveur MDA classique, l'approche DPODI et l'approche PODI enrichie.	163
3.13	Erreurs relatives entre la solution de la MDA et son approximation par POD : e_{DoE}^1 et e_{DoE}^2 sur l'ensemble de test.	171
3.14	Erreurs relatives entre la solution de la MDA et la solution donnée par la moyenne des GP disciplinaires pour les deux disciplines sur chaque point de l'ensemble de test avant enrichissement.	171
3.15	Nombre d'appels aux solveurs disciplinaires pour la résolution de la MDA exacte et en utilisant le solveur DPOD+I Jacobi présenté dans l'Algorithme 16.	172
3.16	Erreurs relatives entre la solution de la MDA et la solution après résolution par l'Algorithme 16 pour les deux disciplines sur chaque point de l'ensemble de test.	172
3.17	Nombre d'appels au solveur structure pour la résolution de la MDA exacte et en utilisant la nouvelle méthode de résolution présentée dans l'Algorithme 13.	173
3.18	Nombre d'appels au solveur aérodynamique pour la résolution de la MDA exacte et en utilisant la nouvelle méthode de résolution présentée dans l'Algorithme 13.	173

3.19 Erreurs relatives entre la solution de la MDA et la solution après résolution par l'Algorithme 13 pour les deux disciplines sur chaque point de l'ensemble de test.	174
3.20 Résolution MDF. Nombre d'appels aux solveurs disciplinaires lors de la résolution de la MDO sans approximation.	184
3.21 Résolution MDF. Espérance et coefficient de variation sur les 10 optimisations des paramètres de design et de la fonction objectif comparés aux valeurs de référence.	184
3.22 Couplage MDF. Nombre d'appels aux solveurs disciplinaires lors de la résolution de la MDO avec le couplage MDF pour 10 optimisations.	185
3.23 Couplage MDF. Espérance et coefficient de variation sur les 10 optimisations des paramètres de design et de la fonction objectif comparés aux valeurs de référence.	185
3.24 Couplage SBO. Nombre d'appels aux solveurs disciplinaires lors de la résolution de la MDO avec le couplage SBO pour 10 optimisations.	186
3.25 Couplage SBO. Espérance et coefficient de variation sur les 10 optimisations des paramètres de design et de la fonction objectif comparés aux valeurs de référence.	186
3.26 Couplage EGMDO. Nombre d'appels aux solveurs disciplinaires lors de la résolution de la MDO avec le couplage EGMDO pour 10 optimisations.	187
3.27 Couplage EGMDO. Espérance et coefficient de variation sur les 10 optimisations des paramètres de design et de la fonction objectif comparés aux valeurs de référence.	187

Liste des algorithmes

1	Newton	13
2	Trust region	15
3	EGO	24
4	Non linear block Jacobi algorithm	32
5	Non linear block Gauss Seidel algorithm	33
6	Uncertainty Reduction of the disciplinary GP	47
7	EGMDO	49
8	Greedy procedure for reduced basis	58
9	Initialization of EGO coupled with MOR	102
10	EGO coupled with MOR	105
11	Compute r samples of coupling variables.	132
12	Compute a POD basis and a training set for the GP.	134
13	Compute MDA solution using DPOD+I approximation.	142
14	Enrichment of the initial model through the resolutions of the MDA.	143
15	Compute MDA solution using POD+I.	145
16	Compute MDA solution using a DPOD+I Jacobi method.	170
17	Couplage MDF	176
18	Couplage SBO	178
19	Uncertainty Reduction of the DPOD+I	182
20	Modified EGMDO	183

Introduction

Cadre

Cette thèse a été supervisée par l'École Doctorale Aéronautique et Astronautique (ED-AA) et financée par l'Office National d'Études et de Recherche Aérospatiale (ONERA) et l'Institut Supérieur de l'Aéronautique et de l'Espace (ISAE-SUPAERO). Le travail de recherche s'est déroulé entre l'unité Méthodes Multidisciplinaires, Concepts Intégrés (M2CI) du Département Traitement de l'Information Systèmes (DTIS) de l'ONERA et l'unité Mathématiques Appliquées du Département d'Ingénierie des Systèmes Complexes (DISC) de l'ISAE. Cette thèse a été dirigée par Michel Salaün professeur à l'ISAE-SUPAERO et Sylvain Dubreuil docteur à l'ONERA et encadrée par Christian Gogu maître de conférence HDR à l'Université Paul Sabatier et Nathalie Bartoli maître de recherche HDR à l'ONERA.

Contexte et formulation du problème

Les nouveaux enjeux sociétaux et écologiques amènent à repenser l'aviation et notamment la conception des avions. Que ce soit au travers des technologies de propulsion (par exemple électrique ou hydrogène) ou de la forme des surfaces portantes (par exemple des ailes haubanées ou à fort allongement ou encore des ailes volantes) des concepts novateurs sont en développement ces dernières années. De telles configurations sont illustrées sur la Figure 1 présentant un concept d'aile volante (Figure 1a) et un concept de propulsion électrique distribuée (Figure 1b). L'étude et l'optimisation de ces nouveaux concepts reposent en grande partie sur la simulation numérique. En effet cette dernière permet une vaste exploration des différents designs pour un coût relativement faible. L'objectif est alors de déterminer les configurations d'avion permettant d'atteindre une performance optimale (masse de l'avion, consommation d'énergie) tout en respectant certaines contraintes (nombre de passagers à transporter, distance franchissable). Formellement ces concepts sont paramétrés par un ensemble de variables, dites de design, qui définissent, par exemple, la géométrie de l'aile, les conditions de vol, le nombre de moteurs, le nombre de passagers etc.

Cependant, la modélisation du comportement d'un avion est souvent complexe, d'autant plus lors de l'analyse de nouveaux concepts où peu de données expérimentales ou de modèles empiriques sont disponibles. Dans ce cas, plusieurs experts interagissent pour approcher les comportements des différentes disciplines impliquées dans la modélisation des performances. Par exemple, un expert peut être spécialiste de l'aérodynamique de l'aile, un autre de la structure ou encore de la propulsion de l'avion. En pratique, ces experts utilisent des solveurs numériques dits solveurs disciplinaires. Lors des dernières décennies, un intérêt particulier a émergé pour la prise en compte des interactions entre les différentes disciplines dès la phase de conception de l'avion. Les disciplines, modélisées par des solveurs disciplinaires, échangent des informations formant un système



(a) Illustration d'une aile volante développée à l'ONERA (projet CICA).



(b) Illustration d'avion électrique (projet DRAGON) développé en partenariat avec l'ONERA.

FIGURE 1 – Exemples de nouveaux concepts d'avions.

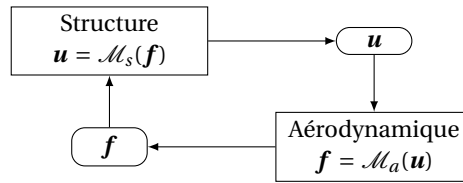


FIGURE 2 – Illustration de la MDA sur un problème avec deux disciplines : aérodynamique et structure où les variables de couplage sont u et f .

couplé interconnecté. Ces informations circulant entre les disciplines sont nommées variables de couplage. Typiquement, l'aérodynamicien s'intéresse à la forme de l'aile suite à la déformation causée par les efforts aérodynamiques ou par les effets de la propulsion sur l'écoulement de l'air. Nous avons donc un système où plusieurs disciplines sont interdépendantes à travers des variables de couplage. Ce système couplé nommé *Analyse multidisciplinaire, ou MultiDisciplinary Analysis (MDA)* est formellement modélisé par un problème de point fixe. Un tel système est illustré sur la Figure 2 où deux disciplines (aérodynamique et structure) sont couplées. Les variables de couplage sont le déplacement u et les efforts aérodynamiques f . La fonction de performance ou toute quantité d'intérêt (objectif et/ou contraintes) sont obtenues à partir des variables de couplage convergées, c'est à dire qui respectent l'équilibre défini par la MDA. Les variables de design ont une influence sur le résultat de chacune de ces disciplines et donc également sur les variables de couplage convergées. Ainsi, le problème induit par la conception avion combine la recherche de variables de design optimales tout en cherchant l'équilibre de la MDA et est nommé *Optimisation multidisciplinaire, ou MultiDisciplinary Design Optimization (MDO)*. La MDO agit alors sur les variables de design et les variables de couplage rendant sa résolution complexe. Ce problème est illustré sur la Figure 3 pour la MDA (présentée sur la Figure 2) faisant intervenir des variables de design x .

Dans le cadre de notre étude, plusieurs défis seront abordés pour la résolution de la MDO :

1. Les solveurs disciplinaires sont gourmands en temps de calcul. Minimiser les appels à ces solveurs lors de la résolution de la MDO représente un des principaux objectifs de la thèse.
2. De nombreux problèmes font intervenir des variables de couplage de grande dimension : typiquement, les champs discrétisés ont une dimension qui excède 10^3 . La méthode proposée devra être capable de traiter cette dimension.

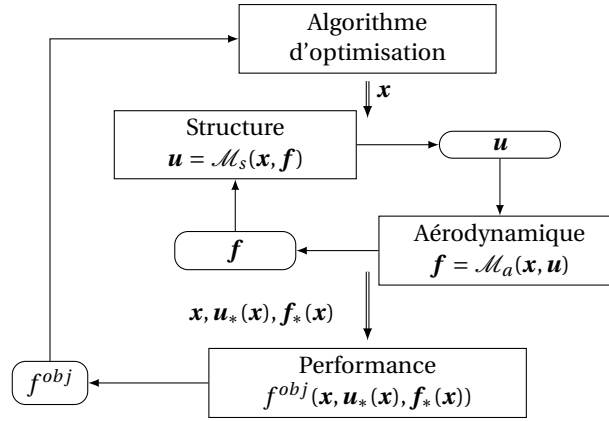


FIGURE 3 – Illustration de la MDO sur un problème avec deux disciplines : aérodynamique et structure où les variables de couplage sont $\mathbf{u}$ et $\mathbf{f}$ et les variables de design sont représentées par le vecteur $\mathbf{x}$. Les variables de couplage convergées sont nommées $\mathbf{u}_*(\mathbf{x})$ et $\mathbf{f}_*(\mathbf{x})$ et permettent d’obtenir la fonction de performance f^{obj} .

3. Dans de nombreux domaines, il est important que les méthodes restent peu intrusives au sein des solveurs disciplinaires afin de pouvoir utiliser comme solveurs numériques des logiciels commerciaux. Dans cette thèse, nous supposons que les solveurs disciplinaires sont des boîtes noires et que les dérivées des quantités d’intérêt ne sont pas accessibles.
4. Même si de nombreuses méthodologies MDO sont développées avec deux disciplines, il est important qu’elles puissent être facilement étendues à plus de disciplines. Nous nous placerons également dans ce cadre.

Plusieurs formulations de la MDO ont été proposées dans la littérature. Cependant, celles-ci ne permettent pas de répondre pleinement aux différents défis évoqués. L’une des formulations les plus communes est l’**Optimisation multidisciplinaire faisable, ou Multidisciplinary Feasible (MDF)** (CRAMER et collab. [1994]) considérant la fonction de performance comme une boîte noire. L’idée de la MDF est alors d’utiliser un algorithme d’optimisation, pour la recherche des variables de design optimales, et un solveur de système non-linéaire, pour déterminer l’équilibre de la MDA. Cependant la formulation MDF nécessite de résoudre une MDA en chaque point proposé par l’algorithme d’optimisation ce qui induit un grand nombre d’appels aux solveurs disciplinaires et donc un coût de calcul important. La MDF ne permet donc pas de surmonter le premier défi.

Une autre formulation, appelée **Individual Discipline Feasible (IDF)** (ALEXANDROV et LEWIS [2004]), cherche à résoudre le problème d’optimisation et la MDA simultanément afin d’éviter de résoudre la MDA dans des zones qui ne sont pas intéressantes du point de vue de l’optimisation. La résolution est donc effectuée par un algorithme d’optimisation manipulant conjointement les variables de design et les variables de couplage. Cependant, les algorithmes d’optimisation ont plus de difficultés à résoudre des problèmes avec un grand nombre de variables, surtout lorsque les dérivées ne sont pas connues. Ceci limite la formulation IDF à un faible nombre de variables de couplage (non compatible avec notre deuxième défi) ou à la connaissance des gradients des solveurs disciplinaires (non compatible avec notre troisième défi).

Finalement, les formulations classiques de résolution de la MDO sont difficiles à utiliser à cause des problèmes que nous venons d’évoquer. En effet, soit ces méthodes sont trop coûteuses, soit elles sont difficiles à mettre en oeuvre avec des variables de couplage de grande dimension sans la connaissance des gradients. Une autre approche consiste à

remplacer les solveurs disciplinaires, coûteux, par des modèles approchés.

La méthode *Efficient Global Multidisciplinary Optimization* (EGMDO) proposée par DUBREUIL et collab. [2020] remplace chaque solveur disciplinaire par un *Processus gaussien*, ou *Gaussian process* (GP) interpolant (WILLIAMS et RASMUSSEN [2006]). Ce GP permet d'obtenir une approximation de chaque solveur via sa moyenne et une information sur l'erreur commise via sa variance. Les GP disciplinaires sont alors couplés afin d'obtenir une approximation de la MDA. Une propagation des incertitudes est également effectuée afin de quantifier les incertitudes liées à ces interpolations. Cette incertitude est utilisée pour définir un enrichissement adaptatif et permet d'affiner les prédictions aux points les plus susceptibles d'être optimaux. Ainsi, les méta-modèles sont uniquement enrichis dans les zones intéressantes pour l'optimisation et cela permet de réduire le nombre d'appels aux solveurs disciplinaires. Cependant, l'une des limitations de cette méthode est qu'elle n'est pas adaptée lorsque les variables de couplage sont de très grande dimension et n'est donc pas compatible avec notre deuxième défi. En effet, les variables de couplage sont les entrées des GP disciplinaires rendant leur apprentissage difficile dans ce cas.

Une autre piste permettant de réduire le coût numérique des solveurs disciplinaires est la *Réduction d'ordre de modèle*, ou *Model Order Reduction* (MOR) (LADEVÈZE [1985]; SIROVICH [1987]). En effet, les solveurs disciplinaires sont généralement obtenus par intégration d'un système d'équations aux dérivées partielles dont la résolution est souvent coûteuse. L'idée principale de cette méthode est de diminuer le coût de calcul par la création et l'utilisation d'un modèle réduit.

Plusieurs méthodologies ont été proposées dont les méthodes de MOR par projection : plutôt que d'inverser un système linéaire de grande dimension, celui-ci est préalablement projeté sur une base réduite de faible dimension diminuant fortement le coût de la résolution de ce système. La base réduite peut être obtenue par *snapshot Décomposition Orthogonale aux valeurs propres*, ou *Proper Orthogonal Decomposition* (POD) (SIROVICH [1987]) ou par des méthodes gloutonnes (VEROY et collab. [2003]). Elle est généralement construite afin de minimiser une certaine erreur sur l'ensemble du domaine ou sur une quantité d'intérêt (méthodes dites *goal-oriented*). Une autre méthodologie, nommée *Décomposition orthogonale aux valeurs propres et interpolation*, ou *Proper Orthogonal Decomposition and Interpolation* (POD+I), consiste à interpoler les coordonnées sur la base réduite. Lors d'une phase d'entraînement, des appels au solveur sont effectués afin de construire une base par POD puis les sorties sont projetées afin de construire un *Plan d'expérience*, ou *Design of Experiment* (DoE) pour l'apprentissage des coefficients. Cette méthode permet de réduire encore le coût de l'approximation puisque le système linéaire n'est pas projeté. Ces deux méthodes sont donc très intéressantes pour relever notre premier défi puisque elles permettent de remplacer les solveurs disciplinaires par des approximations moins coûteuses. Néanmoins, l'erreur commise lors de cette approximation doit être contrôlée.

Deux stratégies sont généralement considérées pour la gestion de l'erreur. Dans la première, une base réduite précise sur tout l'ensemble de design est entraînée puis seule l'approximation est utilisée. Cette stratégie, nommée *Offline-Online*, nécessite une phase d'entraînement extrêmement coûteuse puisque obtenir une base réduite précise sur l'ensemble du domaine de définition est une tâche complexe. Ainsi, cette stratégie ne permet pas de relever le premier défi. Une autre stratégie est d'enrichir la base réduite au fur et à mesure afin qu'elle soit précise uniquement dans les zones jugées intéressantes et de limiter ainsi le coût de l'entraînement. Cependant, cette méthode nécessite de pouvoir estimer l'erreur commise par la MOR afin de déterminer s'il est nécessaire d'enrichir la base réduite. Dans le cas de la POD, l'erreur faite sur la projection est estimée grâce au ré-

sidu du système linéaire et affinée grâce à des méthodes de préconditionnement (ZAHM et NOUY [2016]). Dans le cas de la POD+I, aucune estimation de l'erreur de projection ne peut être calculée et une méthodologie doit être mise en oeuvre pour contrôler cette erreur.

Problématique et plan de la thèse

L'objectif de la thèse est de développer une méthode de résolution d'un problème MDO, non intrusive, avec un couplage fort entre les disciplines permettant de minimiser le nombre d'appels aux solveurs disciplinaires, abordant tous les défis présentés dans la section précédente. Les travaux existants sur la MOR seront utilisés et adaptés au contexte de la MDO afin de réduire la dimension des variables de couplage. L'idée est de proposer une nouvelle formulation, couplant MOR et EGMDO, pour la résolution d'un problème MDO avec un couplage fort. Cependant, l'un des verrous principaux à la construction d'approximations des solveurs disciplinaires par MOR est le couplage entre les disciplines. En effet, la majorité des méthodes de MOR repose sur un échantillonnage des entrées du solveur. Or, dans le contexte de la MDO, ces entrées sont à la fois les variables de design et les variables de couplage ce qui fait émerger deux difficultés : les espaces liés aux variables de couplage sont inconnus et leur dimension est grande.

Ce manuscrit de thèse se décompose en trois chapitres. Le premier correspond à un état de l'art de ces problématiques et les deux suivants décrivent les contributions de la thèse.

Le chapitre 1 a pour objectif de présenter différents outils et méthodes numériques utilisés dans cette thèse. Dans un premier temps, des notions d'optimisation seront introduites avec des outils numériques tels que les GP. Dans un second temps, les méthodes de résolution de la MDO seront présentées et comparées afin de mettre en évidence les apports et les manques de ces méthodes. Les méthodes de MOR seront ensuite analysées afin de déterminer quelles seront les méthodes les plus adaptées à notre contexte. Dans un dernier temps, une revue des méthodes couplant MOR et optimisation sera dressée.

Le chapitre 2 se focalise sur une simplification du problème lorsqu'une seule discipline est impliquée. L'objectif est alors de développer un couplage entre la MOR et l'optimisation bayésienne. Une estimation de l'erreur sera proposée par analyse statistique et préconditionnement. Cette estimation sera utilisée lors de l'optimisation afin de certifier les approximations faites par la MOR permettant d'assurer que le résultat de l'optimisation n'est pas impacté par la MOR. Les méthodes seront alors testées sur un problème de dimensionnement d'une aile d'avion.

Le chapitre 3 développe une adaptation de la méthode EGMDO faisant intervenir des variables de grande dimension. Afin d'atteindre cet objectif une approximation des solveurs disciplinaires par POD+I adaptée au contexte multidisciplinaire sera proposée. Cette approximation sera utilisée pour définir un solveur de MDA par une stratégie d'enrichissement adaptatif. Une méthodologie sera mise en oeuvre pour prendre en compte l'erreur de projection de la POD+I. Dans un second temps, des couplages entre ce solveur et l'optimisation seront proposés afin de résoudre un problème MDO et des applications numériques seront présentées.

Contributions

Publication acceptée

- G. BERTHELIN, S. DUBREUIL, M. SALAÜN, N. BARTOLI et C. GOGU, "Disciplinary Proper Orthogonal Decomposition and Interpolation for the resolution of parametrized Multidisciplinary Analysis", *International Journal for Numerical Methods in Engineering*, 2022.

Communication à des conférences sur résumé étendu

- G. BERTHELIN, S. DUBREUIL, M. SALAÜN, C. GOGU et N. BARTOLI, "Couplage Optimisation et Réduction de Modèles par Projection. Application à la conception avion avant-projet", 14ème Colloque National en Calcul des Structures, 2019.

Communication à des conférences sur résumé

- G. BERTHELIN, S. DUBREUIL, M. SALAÜN, C. GOGU et N. BARTOLI, "Model Order Reduction coupled with Efficient Global Optimization for multidisciplinary Optimization", WCCM-ECCOMAS 2020.

Communication à des conférences sur article

- S. DUBREUIL, N. BARTOLI, G. BERTHELIN, O. CHANVRE VILA, C. GOGU, T. LEFEBVRE, J. MORLIER et M. SALAÜN, "Development of MDO formulations based on disciplinary surrogate models by Gaussian Processes", *AeroBest*, 2021.

Références

- ALEXANDROV, N. et R. LEWIS. 2004, «Reconfigurability in MDO problem synthesis, part 1», dans *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, doi: 10.2514/6.2004-4307. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2004-4307>. 3
- BERTHELIN, G., S. DUBREUIL, M. SALAÜN, C. GOGU et N. BARTOLI. 2019, «Couplage Optimisation et Réduction de Modèles par Projection Application à la conception avion avant-projet», dans *14ème Colloque National en Calcul des Structures*, Giens, France. URL <https://hal.archives-ouvertes.fr/hal-02147411>. 6
- BERTHELIN, G., S. DUBREUIL, M. SALAÜN, N. BARTOLI et C. GOGU. 2022, «Disciplinary proper orthogonal decomposition and interpolation for the resolution of parameterized multidisciplinary analysis», *International Journal for Numerical Methods in Engineering*, doi: <https://doi.org/10.1002/nme.6981>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6981>. 6
- CRAMER, E., J. DENNIS, P. FRANK, R. LEWIS et G. R. SHUBIN. 1994, «Problem formulation for multidisciplinary optimization», *SIAM Journal on Optimization*, vol. 4, p. 754–776. 3
- DUBREUIL, S., N. BARTOLI, G. BERTHELIN, O. CHANDRE VILA, C. GOGU, T. LEFEBVRE, J. MORLIER et M. SALAÜN. 2021, «Development of MDO formulations based on disciplinary surrogate models by Gaussian processes», dans *AeroBest*, Lisbonne (on line), Portugal. URL <https://hal.archives-ouvertes.fr/hal-03280698>. 6
- DUBREUIL, S., N. BARTOLI, C. GOGU et T. LEFEBVRE. 2020, «Towards an efficient global multidisciplinary design optimization algorithm», *Structural and Multidisciplinary Optimization*, vol. 62, n° 4, doi: 10.1007/s00158-020-02514-6, p. 1739–1765, ISSN 1615-1488. URL <https://doi.org/10.1007/s00158-020-02514-6>. 4

- LADÈVÈZE, P. 1985, «New algorithms : mechanical framework and development», *Comptes rendus de l'Académie des Sciences*, vol. 300, n° 2, p. 41–44. [4](#)
- SIROVICH, L. 1987, «Turbulence and the dynamics of coherent structures. i. coherent structures», *Quarterly of applied mathematics*, vol. 45, n° 3, p. 561–571. [4](#)
- VEROY, K., C. PRUD'HOMME, D. ROVAS et A. PATERA. 2003, «A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations», dans *16th AIAA Computational Fluid Dynamics Conference*, doi: 10.2514/6.2003-3847. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2003-3847>. [4](#)
- WILLIAMS, C. K. et C. E. RASMUSSEN. 2006, *Gaussian processes for machine learning*, vol. 2, MIT press Cambridge, MA. [4](#)
- ZAHM, O. et A. NOUY. 2016, «Interpolation of inverse operators for preconditioning parameter-dependent equations», *SIAM Journal on Scientific Computing*, vol. 38, n° 2, p. A1044–A1074. [5](#)

Chapitre 1

État de l'art

« Le mathématicien est à la recherche de quelque chose qui n'existe pas mais qu'il doit néanmoins expliquer à quelques collègues. »

Laurent Lemire

Sommaire

1.1 Quelques notions en optimisation	11
1.1.1 Cadre	11
1.1.2 Algorithmes par modèles locaux	12
1.1.2.1 Algorithme de Newton et de quasi-Newton	12
1.1.2.2 Méthode de région de confiance	14
1.1.2.3 Convergence des algorithmes locaux	14
1.1.2.4 Bilan sur les algorithmes par modèles locaux	16
1.1.3 Algorithmes par modèles globaux	16
1.1.3.1 Généralités sur les méta-modèles	16
1.1.3.2 Les processus gaussiens	17
1.1.3.3 Optimisation par exploitation du méta-modèle	21
1.1.4 Bilan sur les méthodes d'optimisation	24
1.2 Optimisation multidisciplinaire	27
1.2.1 Introduction	27
1.2.1.1 Définition de la MDA paramétrée	27
1.2.1.2 Exemple à deux disciplines	28
1.2.1.3 Approche monolithique et partitionnée	28
1.2.1.4 Les variables de couplage	29
1.2.2 Résolution du système couplé	30
1.2.2.1 Cas test analytique	30
1.2.2.2 Méthode de Jacobi	31
1.2.2.3 Méthode de Gauss-Seidel	32
1.2.2.4 Méthode de Newton	33
1.2.2.5 Bilan sur les méthodes de résolution de l'analyse multidisciplinaire	33
1.2.3 Résolution du problème d'optimisation multidisciplinaire	33

1.2.3.1	Cadre	34
1.2.3.2	Cas test analytique	34
1.2.4	Formulations d'optimisation multidisciplinaire	34
1.2.4.1	Optimisation multidisciplinaire faisable	36
1.2.4.2	<i>Individual Discipline Feasible</i>	37
1.2.5	<i>Efficient Global Multidisciplinary Optimization</i>	38
1.2.5.1	Entraînement des solveurs disciplinaires	39
1.2.5.2	Résolution de la MDA aléatoire	40
1.2.5.3	Fonction objectif aléatoire	42
1.2.5.4	Approximation de la fonction objectif aléatoire	42
1.2.5.5	Amélioration espérée modifiée	45
1.2.5.6	Réduction des incertitudes	46
1.2.5.7	Résolution de la MDO grâce à EGMDO	47
1.2.5.8	Bilan sur la méthode EGMDO	47
1.2.6	Bilan sur l'optimisation multidisciplinaire	49
1.3	Réduction d'ordre de modèle	51
1.3.1	Introduction	51
1.3.2	Obtention des coefficients sur la base réduite	53
1.3.2.1	Projection de Petrov-Galerkin	53
1.3.2.2	Décomposition affine des éléments	54
1.3.2.3	Interpolation des coefficients	55
1.3.3	Génération de l'espace affine V_r^a	56
1.3.3.1	<i>Proper Orthogonal Decomposition</i>	56
1.3.3.2	Approche gloutonne	57
1.3.4	Réduction de modèle avec extraction d'une valeur d'intérêt linéaire	59
1.3.4.1	Problème dual	59
1.3.4.2	Correction Duale	59
1.3.4.3	Majoration de l'erreur issue de la correction duale	60
1.3.5	Bilan sur la réduction d'ordre de modèle	60
1.4	Utilisation de la réduction d'ordre de modèle dans un contexte <i>multi-query</i>	61
1.4.1	Cadre	61
1.4.2	Approche <i>Offline-Online</i>	61
1.4.2.1	Cas d'un solveur linéaire avec quantité d'intérêt linéaire	62
1.4.2.2	Cas d'un solveur linéaire avec quantité d'intérêt non-linéaire	62
1.4.2.3	Cas d'un solveur non-linéaire	62
1.4.2.4	Cas d'un solveur multidisciplinaire	63
1.4.2.5	Bilan des approches <i>Offline-Online</i>	65
1.4.3	Correction de la base réduite en phase <i>Online</i>	65
1.4.3.1	Méthode <i>on-the-fly</i>	65
1.4.3.2	Méthode par région de confiance	66
1.4.4	Bilan sur la réduction d'ordre de modèle dans un contexte <i>multi-query</i>	67
1.5	Synthèse du chapitre	68
1.6	Références	69

Objectifs du chapitre

- Introduire des algorithmes d'optimisation.
- Introduire le problème d'optimisation multidisciplinaire ainsi que certaines méthodes de résolution.
- Introduire des méthodes de réduction d'ordre de modèle.
- Passer en revue des méthodologies couplant réduction d'ordre de modèle et optimisation.

1.1 Quelques notions en optimisation

Les objectifs de cette partie sont

- D'introduire un problème d'optimisation non contraint qui servira d'exemple pour présenter les différentes méthodes.
- Fournir un rappel sur quelques méthodes utilisées pour résoudre ce type de problème :
 - * les méthodes par gradient notamment l'algorithme de Newton qui peuvent être vues comme des méthodes utilisant des méta-modèles locaux.
 - * les méthodes par méta-modèles globaux, dont notamment l'algorithme *Efficient Global Optimization* (EGO).

Dans un premier temps nous allons nous intéresser à des rappels sur l'optimisation afin d'introduire la problématique du sujet ainsi que des notions et outils qui seront utilisés dans la suite de la thèse. L'objectif est d'illustrer l'importance de la prise en compte de l'incertitude liée aux méta-modèles. Pour cela, nous nous intéresserons particulièrement au *Processus gaussien*, ou *Gaussian process* (GP) car il permet, à travers l'information fournie par la variance, d'estimer l'erreur commise lors de l'interpolation. Nous étudierons alors comment l'algorithme *Efficient Global Optimization* (EGO) exploite cette incertitude, ce qui nous servira de brique de base pour les méthodes développées dans cette thèse.

1.1.1 Cadre

Nous allons nous intéresser aux problèmes d'optimisation non-contraints, l'extension aux problèmes sous contraintes ne sera pas détaillée ici mais le lecteur peut se référer à [BYRD et collab. \[1987\]](#) ; [SCHONLAU et collab. \[1998\]](#) ; [SASENA et collab. \[2002\]](#). Un problème d'optimisation non contraint est défini de la façon suivante :

$$\mathbf{x}_* \in \arg \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \quad (1.1)$$

où $f : \mathcal{X} \subset \mathbb{R}^q \mapsto \mathbb{R}$ est nommée fonction objectif et $\mathcal{X}$ est l'espace des paramètres aussi noté espace de design, $\mathbf{x}$ est le vecteur des paramètres de conception (appelé encore paramètres de design). Nous prenons en compte, à travers l'espace $\mathcal{X}$, certaines contraintes

de bornes sur les paramètres de design. En pratique, $\mathcal{X}$ est souvent défini par un hypercube de $\mathbb{R}^q$. Dans le cadre de ces travaux de thèse les hypothèses suivantes sont faites sur la fonction objectif :

- l'évaluation de f est réalisée par un code de calcul numérique. Ce calcul est coûteux en terme de temps de calcul CPU.
- la fonction f peut avoir plusieurs minima locaux.
- le gradient de f n'est pas disponible et sera uniquement accessible par approximation de type différences finies.

Ce cadre particulier nommé *blackbox optimization* (plusieurs applications sont détaillées par [ALARIE et collab. \[2021\]](#)) empêche la mise en oeuvre de certaines méthodes. La première hypothèse limite l'utilisation d'algorithmes basés sur des échantillons comme le recuit simulé [KIRKPATRICK \[1984\]](#); [KIRKPATRICK et collab. \[1983\]](#), les algorithmes génétiques [KATOCH et collab. \[2021\]](#) car ils requièrent de nombreux appels à la fonction objectif. Un focus sur les méthodes proposant la construction d'un modèle d'approximation de la fonction objectif est fait. Pour cela, une distinction est faite entre les modèles locaux de la fonction objectif et les modèles globaux.

1.1.2 Algorithmes par modèles locaux

Le choix est fait ici de présenter d'abord les algorithmes utilisant le gradient, reconnus pour leur efficacité. L'objectif est de créer un modèle approché de la fonction objectif à partir du développement de Taylor autour d'un point initial $\mathbf{x}_0 \in \mathcal{X}$ détaillé dans l'Eq. (1.2)

$$f(\mathbf{x}) = f(\mathbf{x}_0) + (\nabla f(\mathbf{x}_0))^T (\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \mathbf{H}(f)(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) + \mathcal{O}(\|\mathbf{x} - \mathbf{x}_0\|_2^3) \quad (1.2)$$

où ∇f est le gradient de la fonction $f(\mathbf{x})$ et $\mathbf{H}(f)$ sa matrice Hessienne. Cependant, dans notre cas d'étude, le gradient de la fonction n'est pas calculable directement. Il est évalué par différences finies. Par exemple, l'approximation du gradient en un point $\mathbf{x}_0 \in \mathcal{X}$ est calculée à partir d'un pas $h \ll 1$:

$$\frac{\delta f}{\delta \mathbf{x}_i}(\mathbf{x}_0) \approx \frac{f(\mathbf{x}_0 + h\mathbf{e}_i) - f(\mathbf{x}_0)}{h}, \quad i = 1, \dots, q \quad (1.3)$$

où $\{\mathbf{e}_1, \dots, \mathbf{e}_q\}$ sont les vecteurs de la base canonique de $\mathbb{R}^q$. Ainsi, chaque calcul approché du gradient requiert $q + 1$ calculs, ce qui devient rapidement prohibitif en terme de coût numérique lorsque la fonction objectif f est évaluée par un solveur numérique et que la dimension q est grande. La matrice Hessienne peut elle aussi être approchée par différence finies.

L'idée est alors de créer une suite de points $\mathbf{x}_1, \dots, \mathbf{x}_n$ de $\mathcal{X}$, à partir d'un point $\mathbf{x}_1$ défini par l'utilisateur, convergeant vers la solution $\mathbf{x}_*$ du problème (1.1) :

$$\lim_{n \rightarrow \infty} \mathbf{x}_n = \mathbf{x}_* \quad (1.4)$$

Cela donne lieu à différents algorithmes d'optimisation selon la façon dont est construite la suite de points $\mathbf{x}_1, \dots, \mathbf{x}_n$.

1.1.2.1 Algorithme de Newton et de quasi-Newton

Parmi tous ces algorithmes certains utilisent l'information apportée par la Hessienne de la fonction comme les méthodes de Newton, qui cherchent le minimum de la forme quadratique du développement de Taylor d'ordre 2.

Pour cela, un modèle local $m_n(\mathbf{x}_n + \mathbf{s})$ d'ordre 2, appelé modèle quadratique, est construit à partir du développement de Taylor obtenu grâce à l'Eq. (1.2) :

$$m_n(\mathbf{x}_n + \mathbf{s}) = f(\mathbf{x}_n) + (\nabla f(\mathbf{x}_n))^T \mathbf{s} + \frac{1}{2} \mathbf{s}^T \mathbf{H}(f)(\mathbf{x}_n) \mathbf{s} \quad (1.5)$$

Le nouveau point $\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{s}_n$ est alors déterminé grâce au pas $\mathbf{s}_n$ solution du problème suivant :

$$\mathbf{s}_n = \underset{\mathbf{s}}{\operatorname{argmin}} m_n(\mathbf{x}_n + \mathbf{s}) \quad (1.6)$$

où $\mathbf{x}_n$ est le point obtenu à l'itération précédente. Notons que si la Hessienne est définie positive, la solution de ce problème est analytique et est obtenue en annulant le gradient de m_n aboutissant à la résolution du système suivant :

$$\mathbf{H}(f)(\mathbf{x}_n) \mathbf{s}_n = -\nabla f(\mathbf{x}_n) \quad (1.7)$$

En pratique, la solution de l'Eq. (1.7) n'est pas utilisée telle quelle mais permet de définir une direction de recherche du nouveau pas. La norme du pas est alors obtenue par une méthode de recherche linéaire, ou *line-search* (GRIPPO et collab. [1989]). Malheureusement, la Hessienne d'une fonction est rarement disponible dans le cas où f est évaluée par un solveur numérique. Pour résoudre cette difficulté, l'algorithme *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) proposé par BROYDEN [1970]; FLETCHER [1970]; GOLD-FARB [1970]; SHANNO [1970] définit une approximation de l'inverse de la matrice Hessienne à partir de l'information fournie par le gradient. Finalement, l'optimisation prend fin lorsqu'un certain critère d'arrêt, ou *stopping criterion* est atteint. Celui-ci peut être défini sur la norme du gradient, sur la norme du pas ou sur la valeur de la fonction objectif :

$$\frac{\|\nabla f(\mathbf{x}_n)\|}{\|\nabla f(\mathbf{x}_1)\|} \leq \epsilon_g \quad \frac{|f(\mathbf{x}_{n+1}) - f(\mathbf{x}_n)|}{|f(\mathbf{x}_1)|} \leq \epsilon_f \quad \frac{\|\mathbf{s}_n\|}{\|\mathbf{x}_1\|} \leq \epsilon_x \quad (1.8)$$

où $\epsilon_g > 0$, $\epsilon_f > 0$ et $\epsilon_x > 0$ sont des tolérances définies par l'utilisateur. Les étapes de l'algorithme de Newton sont détaillées dans l'Algorithme 1.

Algorithm 1: Newton

```

input :  $\mathbf{x}_1$ 
Compute  $f(\mathbf{x}_1)$ ;
 $n \leftarrow 1$ ;
conv  $\leftarrow$  False;
while conv do
    Compute the gradient  $\nabla f(\mathbf{x}_n)$  and the Hessian matrix  $\mathbf{H}(f)(\mathbf{x}_n)$  ;
    Compute the new step  $\mathbf{s}_n$  by solving Eq. (1.7) combined with a line-search;
     $\mathbf{x}_{n+1} \leftarrow \mathbf{x}_n + \mathbf{s}_n$ ;
    Compute  $f(\mathbf{x}_{n+1})$ ;
    Compute the stopping criteria;
    if stopping criteria then
        | conv  $\leftarrow$  True
    end
     $n \leftarrow n + 1$ 
end
output:  $\mathbf{x}_n$ 

```

Finalement, bien que cette méthode soit efficace, un problème de robustesse peut apparaître lorsque le point $\mathbf{x}_{n+1}$ est loin du point $\mathbf{x}_n$ puisque le modèle local perd en précision ce qui peut ralentir ou empêcher la convergence de l'algorithme. Pour y remédier, une revue est proposée sur les méthodes de *Région de confiance*, ou *Trust Region* (TR).

1.1.2.2 Méthode de région de confiance

Ces méthodes partent du postulat que, comme le modèle proposé par l'Eq. (1.2) est local, la recherche du meilleur paramètre devra se faire sur une région proche du point actuel, dite de confiance (POWELL [1970]). Parmi toutes les façons de définir les régions de confiance (DENG et collab. [1993]; NOCEDAL et YUAN [1998]), nous nous focalisons sur une sphère caractérisée par son rayon ρ (SORENSEN [1982]; FLETCHER [2000]). Ici, nous essayons de donner l'idée générale sans prendre en compte les différences entre les formulations proposées dans la littérature. Ainsi, on définit le calcul du nouveau pas $\mathbf{s}_n$:

$$\mathbf{s}_n = \arg \min_{\|\mathbf{s}\| \leq \rho} m_n(\mathbf{x}_n + \mathbf{s}) \quad (1.9)$$

dépendant du rayon ρ de la région de confiance. Ce paramètre est critique : si ρ est trop petit, le nombre d'itérations va être très important, si ρ est trop grand, des problèmes de robustesse peuvent advenir. Pour y remédier, il est proposé de faire varier ρ à chaque itération définissant une suite ρ_n au fur et à mesure des itérations de l'algorithme d'optimisation. L'évolution de ρ est paramétrée par cinq constantes η_1 , η_2 , γ_1 , γ_2 et γ_3 telles que

$$0 < \eta_1 < \eta_2 < 1 \text{ et } 0 < \gamma_1 < \gamma_2 < 1 \leq \gamma_3 \quad (1.10)$$

L'actualisation du rayon de confiance et l'acceptation du pas sont alors obtenues à partir de l'efficacité du modèle local déterminé par le rapport suivant :

$$r_n = \frac{f(\mathbf{x}_n) - f(\mathbf{x}_n + \mathbf{s}_n)}{m_n(\mathbf{x}_n) - m_n(\mathbf{x}_n + \mathbf{s}_n)} \quad (1.11)$$

- Si $r_n \geq \eta_2$ l'étape est un succès, $\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{s}_n$ et le modèle local est précis, ρ_n est donc augmenté : $\rho_{n+1} \in [\rho_n, \gamma_3 \rho_n]$
- Si $\eta_1 \leq r_n \leq \eta_2$ l'étape est un succès, $\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{s}_n$ mais le modèle local n'est pas précis, ρ_n est donc diminué : $\rho_{n+1} \in [\gamma_2 \rho_n, \rho_n]$
- Si $r_n < \eta_1$ l'étape est un échec, $\mathbf{x}_{n+1} = \mathbf{x}_n$ et le modèle local n'est pas précis, ρ_n est donc diminué : $\rho_{n+1} \in [\gamma_1 \rho_n, \gamma_2 \rho_n]$

À partir d'un choix initial du rayon de confiance ρ_0 , la dynamique proposée permet de s'adapter au problème d'optimisation. Si le modèle local est précis, la région de confiance est agrandie afin de limiter le nombre d'itérations et dans le cas contraire, la région est diminuée afin d'assurer la convergence. Les étapes de l'algorithme de région de confiance sont détaillées dans l'Algorithme 2.

Finalement, en analysant l'erreur commise par le modèle local, il est possible de définir une dynamique sur le rayon de la région de confiance afin d'améliorer la convergence de la méthode.

1.1.2.3 Convergence des algorithmes locaux

Les algorithmes d'optimisation basés sur le gradient convergent (lorsque le pas des itérations est bien réglé) vers un minimum local mais pas nécessairement vers le minimum global de la fonction objectif. Or, comme l'unicité du minimum n'est pas assurée, il est nécessaire d'effectuer une stratégie *multi-start* c'est à dire lancer plusieurs optimisations par gradient en partant de différents points de départ. Ceci permet de converger vers certains minima locaux de la fonction objectif f puis de choisir le meilleur. Notons que, sans information sur f , le nombre de points de départ nécessaire est difficile à estimer et son choix est souvent fait selon le budget de calcul alloué.

Algorithm 2: Trust region

input : $\mathbf{x}_1, \rho_1, \eta_1, \eta_2, \gamma_1, \gamma_2$ and γ_3
Compute $f(\mathbf{x}_1)$;
 $n \leftarrow 1$;
 $\text{conv} \leftarrow \text{False}$;
while conv **do**
 Compute the gradient $\nabla f(\mathbf{x}_n)$ and the Hessian matrix $\mathbf{H}(f)(\mathbf{x}_n)$;
 Compute the new step $\mathbf{s}_n$ by solving Eq. (1.9);
 Compute $f(\mathbf{x}_n + \mathbf{s}_n)$;
 Compute r_n using Eq. (1.9);
 if $r_n > \eta_2$ **then**
 $\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{s}_n$;
 $\rho_{n+1} \in [\rho_n, \gamma_3 \rho_n]$;
 end
 if $\eta_1 \leq r_n \leq \eta_2$ **then**
 $\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{s}_n$;
 $\rho_{n+1} \in [\gamma_2 \rho_n, \rho_n]$;
 end
 if $r_n < \eta_1$ **then**
 $\mathbf{x}_{n+1} = \mathbf{x}_n$;
 $\rho_{n+1} \in [\gamma_1 \rho_n, \gamma_2 \rho_n]$;
 end
 Compute the stopping criteria;
 if stopping criteria **then**
 $\text{conv} \leftarrow \text{True}$
 end
 $n \leftarrow n + 1$
end
output: $\mathbf{x}_n$

1.1.2.4 Bilan sur les algorithmes par modèles locaux

Dans cette section, nous avons présenté un certain nombre de méthodes classiques d'optimisation par modèles locaux. Ces méthodes s'appuient sur des bases théoriques solides, prouvant leur convergence dans un certain cadre. Dans le cas où la connaissance du gradient est acquise, ces méthodes sont particulièrement adaptées. Notons que plusieurs méthodes ont été développées pour résoudre ce type de problème avec des codes de calcul numériques notamment dans le cadre de l'optimisation multidisciplinaire (HWANG et collab. [2014] KENWAY et collab. [2014]; BONCORAGLIO et collab. [2021]). Les méthodes par gradient ont été utilisées plus généralement pour l'optimisation du profil d'une aile d'avion pour des solveurs aérodynamiques (JAKOBSSON et AMOIGNON [2007]; CARRIER et collab.) ou le dimensionnement d'une structure (KENNEDY et MARTINS [2014] SJØLUND et LUND [2018]). Bien que les différences finies augmentent le coût numérique de l'optimisation, celles-ci sont utilisées pour la résolution de problèmes d'optimisation multidisciplinaire comme dans COELHO et collab. [2009] où un modèle approché de l'optimisation multidisciplinaire est construit pour en réduire le coût. De fait, les méthodes par gradient seront utilisées comme référence afin de valider les différentes approches qui seront développées par la suite. Notons que ce calcul dit de référence sera d'autant plus coûteux qu'il sera associé à un *multi-start* pour trouver le minimum global.

Bien que certaines méthodes utilisent des modèles locaux qui ne sont pas basés sur le gradient, notamment l'algorithme Cobyla (POWELL [1994]) qui construit des approximations linéaires successives de la fonction objectif f , nous allons par la suite plutôt nous intéresser aux méthodes utilisant des méta-modèles globaux de type *Efficient Global Optimization*.

1.1.3 Algorithmes par modèles globaux

Dans cette section, une étude des algorithmes d'optimisation utilisant des modèles globaux est proposée. À cette fin, la première étape est de proposer une approximation de la fonction objectif en utilisant des méta-modèles. La seconde étape consiste à exploiter l'approximation de la fonction afin de résoudre le problème d'optimisation.

1.1.3.1 Généralités sur les méta-modèles

1.1.3.1.1 Introduction La méta-modélisation consiste à approcher une fonction, par exemple la fonction objectif f du problème d'optimisation, par un modèle mathématique ou modèle de substitution noté $\hat{f}$. Dans notre cadre, nous nous intéressons uniquement aux approximations construites à partir d'un échantillon de points. Dans ce cas on parle souvent de méta-modèle construit par apprentissage supervisé. L'apprentissage du modèle s'effectue sur un [Plan d'expérience, ou Design of Experiment \(DoE\)](#) composé de réalisations de la fonction : un ensemble de d points $\mathbf{x}_{\text{DoE}} = \{\mathbf{x}_1, \dots, \mathbf{x}_d\} \in \mathcal{X}^d$ et leurs sorties correspondantes $\mathbf{f}_{\text{DoE}} = \{f_1, \dots, f_d\} \in \mathbb{R}^d$ avec $f_i = f(\mathbf{x}_i) \in \mathbb{R}$. L'intérêt des méta-modèles dans le cadre de notre thèse est qu'ils permettent à partir d'un plan d'expérience de construire une approximation d'une fonction de type boîte noire. Ceci permet alors d'interagir avec un méta-modèle analytique à la place d'un solveur numérique. De plus, l'obtention du plan d'expérience est non intrusive puisque uniquement des appels au solveur pour différentes entrées sont effectués.

Générer ce DoE est une étape cruciale de la construction du méta-modèle. En effet, un DoE mal réparti dans l'espace des paramètres d'entrée peut engendrer l'imprécision du méta-modèle dans certaines zones d'intérêt pour l'optimisation. Cependant, l'usage d'un

plan factoriel complet (tensorisation d'une discrétisation uni-dimensionnelle) souffre de la malédiction de la dimension (BELLMAN [2021]). Pour éviter cette difficulté plusieurs méthodes d'échantillonnage existent telles que

- des méthodes déterministes : *Space Filling* (DAMBLIN et collab. [2013]), les suites à discrétion faible (NIEDERREITER [1988]) comprenant par exemple les suites de Sobol (SOBOL' [1967]).
- des méthodes aléatoires : *Monte Carlo* (MC) ou *Simple Random Sampling*, *Échantillonnage par Hypercube Latin*, ou *Latin Hypercube Sampling* (LHS) (MCKAY et collab. [1979]; DAMBLIN et collab. [2013]) pouvant être optimisées selon un certain critère (par exemple l'*Enhanced Stochastic Evolutionary* proposé par JIN et collab. [2003]) ou non, *Stratified Sampling* (MCKAY et collab. [1979]).

Parmi toutes ces méthodes nous privilégierons les LHS pour leur simplicité à générer des points.

À partir de ce DoE, le modèle mathématique $\hat{f}_\theta$ est ajusté à l'aide d'hyperparamètres notés θ afin d'obtenir la meilleure approximation de la fonction possible. Les paramètres du modèle peuvent par exemple être estimés par apprentissage statistique (maximum de vraisemblance, *leave one out*, moindre carré) (VAPNIK [1999]; HASTIE et collab. [2009]; BERVEILLER et collab. [2006]). Le nombre de ces hyperparamètres varie en fonction du méta-modèle choisi et est généralement sensible à la dimension de l'espace d'entrée.

1.1.3.1.2 Différents méta-modèles De nombreux méta-modèles ont été proposés dans la littérature parmi lesquels les modèles polynomiaux (MYERS et collab. [2012]) avec notamment les interpolations quadratiques (KAUFMAN et collab. [1996]), le chaos polynomial (GHANEM et SPANOS [2003]; DUTKA-MALEN et collab. [2009]), les modèles de régression par *Processus gaussien*, ou *Gaussian process* (GP) aussi dénommés krigeage, ou *kriging* (KRIGE [1951]; MATHERON [1963]) détaillés dans le livre de WILLIAMS et RASMUSSEN [2006], les modèles de *Fonctions à base radiale*, ou *Radial Basis Function* (RBF) (BROOMHEAD et LOWE [1988]), les *Réseaux de neurones artificiels*, ou *Artificial Neural Network* (ANN) (McCULLOCH et PITTS [1943]; DREYFUS et collab. [2002]; HASTIE et collab. [2009]). Parmi les méta-modèles disponibles dans la littérature, l'interpolation par *Processus gaussien*, ou *Gaussian process* (GP) apporte un avantage majeur : elle permet d'obtenir une approximation de l'erreur commise par l'interpolation, donnée par la variance du GP. Cette information est utilisée en optimisation pour permettre l'exploration de zones de $\mathcal{X}$ qui sont mal représentées par le méta-modèle mais potentiellement intéressantes. Cette incertitude sur l'interpolation sera par la suite exploitée dans ces travaux de thèse afin de mettre en place un processus d'enrichissement adaptatif (cf Section 1.2.5). Nous rappelons donc dans la section suivante le principe de construction des GP.

1.1.3.2 Les processus gaussiens

Les GP ont été utilisés pour la première fois dans le cadre de la régression par Daniel Krige KRIGE [1951] en géostatistique, puis cette méthode a été formalisée par le géologue et mathématicien français George Matheron (MATHERON [1963]). Plusieurs contributions ont été proposées pour répondre à différentes problématiques. La problématique de la grande dimension des variables d'entrée a été traitée par l'utilisation d'une réduction de dimension par moindres carrés partiels (BOUHLEL et collab. [2016]) ou par *Décomposition Orthogonale aux valeurs propres*, ou *Proper Orthogonal Decomposition* (POD) (BRACONNIER et collab. [2011]; ROY et collab. [2018]). La combinaison de différentes fidélités

de modèle a été traitée par cokrigeage (KENNEDY et O'HAGAN [2000] ; LE GRATIET [2013]). Finalement, les GP ont été utilisés dans de nombreuses applications comme l'optimisation (JONES et collab. [1998] ; FORRESTER et collab. [2008]) ou encore l'analyse de fiabilité (PICHENY et collab. [2010] ; ECHARD et collab. [2011]).

Dans cette partie, l'objectif est d'introduire brièvement ces modèles en se basant sur la présentation de WILLIAMS et RASMUSSEN [2006]. Dans la suite de la thèse, ils seront utilisés pour résoudre le problème d'optimisation à travers la méthode EGO puis comme modèles de substitution pour l'optimisation multidisciplinaire dans la suite de la thèse.

Un GP $\mathcal{G}$ est une collection infinie de variables aléatoires définies sur le même espace de probabilités $(\Omega_p, \mathcal{A}, \mathbb{P})$ dont chaque sous-ensemble fini suit une loi normale multivariée. L'idée de la régression par GP conditionné est d'identifier la fonction à approcher par un GP indexé par les variables de design (ou par les paramètres d'entrée). L'hypothèse principale de la régression par GP est de supposer que cette fonction est la réalisation d'un GP. Tout l'enjeu sera alors de trouver le GP qui soit le plus à même d'expliquer cette réalisation. Pour cela, l'objectif est de créer un GP $\mathcal{G}_a$ dit a priori caractérisé par

— sa moyenne :

$$\begin{aligned} \mu_a : \quad \mathbf{x} &\rightarrow \mathbb{E}[\mathcal{G}(\mathbf{x})] \\ \mathcal{X} &\mapsto \mathbb{R} \end{aligned} \quad (1.12)$$

— son noyau de covariance :

$$\begin{aligned} k_a : \quad \mathbf{x}, \mathbf{x}' &\rightarrow \text{CoVar}(\mathcal{G}(\mathbf{x}), \mathcal{G}(\mathbf{x}')) \\ \mathcal{X} \times \mathcal{X} &\mapsto \mathbb{R} \end{aligned} \quad (1.13)$$

où $\text{CoVar}(\cdot, \cdot)$ est la covariance entre deux variables aléatoires.

Par la suite $\mathcal{G}_a$ est conditionné pour qu'il prenne en compte les données connues à savoir le DoE afin d'obtenir le GP conditionné $\mathcal{G}_c$. L'objectif de la régression par GP est alors de paramétrer les deux fonctions μ_a et k_a par des hyperparamètres afin d'ajuster au mieux la régression.

Dans notre cas, nous allons nous intéresser au modèle de krigeage universel (PICHENY et collab. [2010] ; ROUSTANT et collab. [2012]) où le processus gaussien s'exprime sous la forme :

$$\mathcal{G}_a(\mathbf{x}) = \underbrace{\mathbf{e}(\mathbf{x})^T}_{\mu_a(\mathbf{x})} \boldsymbol{\beta} + Z(\mathbf{x}) \quad (1.14)$$

avec $\mathbf{e}(\mathbf{x})$ le vecteur des fonctions de base, $\boldsymbol{\beta}$ les coefficients associés et un GP $Z \sim \mathcal{N}(0, k_a)$ où k_a est le noyau de covariance de $\mathcal{G}_a$. Ce noyau de covariance s'exprime comme suit :

$$k_a(\mathbf{x}, \mathbf{x}') = \sigma_Z^2 r_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') \quad (1.15)$$

où σ_Z^2 est la variance de Z et $r_{\boldsymbol{\theta}}$ est une fonction de corrélation définie par un ensemble d'hyperparamètres $\boldsymbol{\theta}$. À l'instar des fonctions de base permettant de construire la moyenne, la fonction de corrélation $r_{\boldsymbol{\theta}}$ est supposée connue et dépendante des hyperparamètres $\boldsymbol{\theta}$. Cette construction a priori est ensuite conditionnée pour interpoler les données du DoE afin de définir le GP $\mathcal{G}_c$ dit a posteriori :

$$\hat{\mathcal{G}}_c = \mathcal{G}_a | \{\mathbf{x}_1, \dots, \mathbf{x}_d\}, \{f_1, \dots, f_d\} \quad (1.16)$$

Ce GP est défini par sa moyenne μ et son noyau de covariance k différents de μ_a et k_a . Le passage entre le GP a priori et le GP a posteriori pour tout point $\mathbf{x} \in \mathcal{X}$ est obtenu à partir

de la matrice de corrélation $\mathbf{K}_{cor}$ du processus a priori évaluée aux points du DoE :

$$\mathbf{K}_{cor} = \begin{pmatrix} k_a(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k_a(\mathbf{x}_1, \mathbf{x}_d) \\ \vdots & \ddots & \vdots \\ k_a(\mathbf{x}_d, \mathbf{x}_1) & \cdots & k_a(\mathbf{x}_d, \mathbf{x}_d) \end{pmatrix} \quad (1.17)$$

et du vecteur $\boldsymbol{\kappa}$ de corrélation entre le point $\mathbf{x}$ et les données du DoE :

$$\boldsymbol{\kappa}(\mathbf{x}) = \begin{pmatrix} k_a(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ k_a(\mathbf{x}, \mathbf{x}_d) \end{pmatrix} \quad (1.18)$$

La moyenne et la covariance du GP conditionné sont alors données par :

$$\mu(\mathbf{x}) = \mathbf{e}(\mathbf{x})^T \boldsymbol{\beta} + \boldsymbol{\kappa}(\mathbf{x})^T \mathbf{K}_{cor}^{-1} (\mathbf{F}_{DoE} - \mathbf{e}_x^T \boldsymbol{\beta}) \quad (1.19)$$

avec $\mathbf{e}_x = \{\mathbf{e}(\mathbf{x}_1), \dots, \mathbf{e}(\mathbf{x}_d)\}$.

$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}, \mathbf{x}') - (\boldsymbol{\kappa}(\mathbf{x})^T \mathbf{e}(\mathbf{x})^T) \begin{pmatrix} \mathbf{K}_{cor} & \mathbf{e}_x^T \\ \mathbf{e}_x & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{\kappa}(\mathbf{x}') \\ \mathbf{e}(\mathbf{x}') \end{pmatrix} \quad (1.20)$$

Maintenant que le passage du GP a priori au GP a posteriori (ou conditionné) a été détaillé, il faut nous intéresser à l'identification des meilleurs hyperparamètres $\boldsymbol{\theta}$, des coefficients permettant de définir la moyenne $\boldsymbol{\beta}$ et la variance σ_Z . Lorsque les hyperparamètres $\boldsymbol{\theta}$ sont donnés, les coefficients de la moyenne $\boldsymbol{\beta}$ et la variance σ_Z peuvent être estimés analytiquement par maximum de vraisemblance (FORRESTER et collab. [2008]) :

$$\hat{\boldsymbol{\beta}} = (\mathbf{e}_x^T \mathbf{K}_{cor}^{-1} \mathbf{e}_x)^{-1} \mathbf{e}_x^T \mathbf{K}_{cor}^{-1} \mathbf{F}_{DoE} \quad (1.21)$$

$$\hat{\sigma}_Z^2 = \frac{1}{d} (\mathbf{F}_{DoE} - \hat{\boldsymbol{\beta}} \mathbf{e}_x)^T \mathbf{K}_{cor}^{-1} (\mathbf{F}_{DoE} - \hat{\boldsymbol{\beta}} \mathbf{e}_x) \quad (1.22)$$

Par la suite, les hyperparamètres $\boldsymbol{\theta}$ sont ajustés par une optimisation numérique. Différents critères permettant de définir les hyperparamètres optimaux ont été proposés dans la littérature comme par maximum de vraisemblance (JONES [2001]), par maximum a posteriori (WILLIAMS et RASMUSSEN [2006]) ou par validation croisée (FUGLSTAD et collab. [2019]). Une illustration du résultat d'une régression par GP est proposée sur la Figure 1.1 en prenant l'exemple de fonction 1D issue de FORRESTER et collab. [2008].

Enfin, les noyaux de covariance $r_{\boldsymbol{\theta}}$ permettent de modéliser la corrélation entre les valeurs du GP. Ces noyaux permettent de définir le noyau du GP a priori $\mathcal{G}_a$ par le biais de la variance σ_Z dont l'estimateur est défini dans l'Eq. (1.22). Plusieurs fonctions de corrélation ont été proposées et analysées dans la littérature, par exemple :

— Le noyau gaussien, particulièrement populaire, est défini par

$$r_{\boldsymbol{\theta}}^G(\mathbf{x}, \mathbf{x}') = \exp \left[-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{\boldsymbol{\theta}^2} \right] \quad (1.23)$$

— Le noyau exponentiel est défini par

$$r_{\boldsymbol{\theta}}^E(\mathbf{x}, \mathbf{x}') = \exp \left[-\frac{\|\mathbf{x} - \mathbf{x}'\|_2}{\boldsymbol{\theta}} \right] \quad (1.24)$$

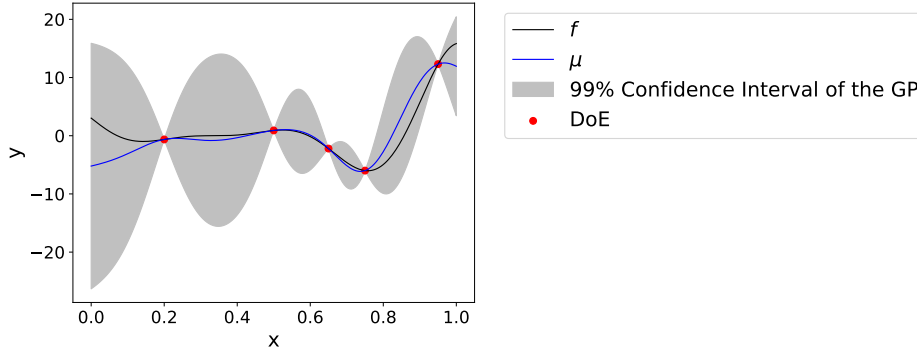


FIGURE 1.1 – Illustration d'une régression par GP sur une fonction à une variable définie par **FORRESTER et collab. [2008]** où la moyenne μ du GP approche la fonction f avec un DoE composé de 5 points d'apprentissage. Un Intervalle de Confiance (IC) sur l'interpolation est calculé à partir de la variance du GP (en prenant $\mu(\mathbf{x}) \pm 3\sqrt{k(\mathbf{x}, \mathbf{x})}$).

— Le noyau Matern $3/2$ qui s'exprime par :

$$r_{\boldsymbol{\theta}}^{M^{3/2}}(\mathbf{x}, \mathbf{x}') = \left(1 + \frac{\sqrt{3}\|\mathbf{x} - \mathbf{x}'\|_2}{\boldsymbol{\theta}}\right) \exp\left[-\frac{\sqrt{3}\|\mathbf{x} - \mathbf{x}'\|_2}{\boldsymbol{\theta}}\right] \quad (1.25)$$

— Le noyau Matern $5/2$ qui s'exprime par :

$$r_{\boldsymbol{\theta}}^{M^{5/2}}(\mathbf{x}, \mathbf{x}') = \left(1 + \frac{\sqrt{5}\|\mathbf{x} - \mathbf{x}'\|_2}{\boldsymbol{\theta}} + \frac{5\|\mathbf{x} - \mathbf{x}'\|_2^2}{3\boldsymbol{\theta}^2}\right) \exp\left[-\frac{\sqrt{5}\|\mathbf{x} - \mathbf{x}'\|_2}{\boldsymbol{\theta}}\right] \quad (1.26)$$

Dans le cadre de nos travaux, l'intérêt se porte sur des noyaux stationnaires c'est à dire que ceux-ci dépendent uniquement de la distance entre $\mathbf{x}$ et $\mathbf{x}'$, les noyaux non-stationnaires ne seront pas abordés dans cette thèse. La régularité de l'approximation est définie par celle du noyau soulignant l'importance du choix du noyau lors de la construction du GP. Par exemple, la moyenne obtenue à partir d'un noyau gaussien est infiniment différentiable alors que celle obtenue par un noyau Matern $3/2$ n'est dérivable que deux fois. D'autre part, lorsque la dimension q des entrées du GP devient importante et que les données du DoE sont trop proches, la matrice de corrélation $\mathbf{K}_{cor}$ peut être mal conditionnée (**ZIMMERMANN [2015]**). Cela amène à une imprécision sur l'inversion de la matrice de corrélation voire même une impossibilité de l'inverser. Or, ce conditionnement est influencé par le choix du noyau. Lors de nos expériences numériques, il s'avère que le noyau Matern $5/2$ se montre plus robuste en terme de conditionnement de la matrice de corrélation $\mathbf{K}_{cor}$. De plus, la précision sur la moyenne du GP obtenue est souvent meilleure en utilisant ce noyau. Pour cette raison, le noyau que nous privilégierons dans la suite sera le noyau Matern $5/2$.

Finalement, grâce à la régression par GP, il est possible de définir une approximation de la fonction objectif f à travers la moyenne conditionnée μ . De plus, une information supplémentaire est apportée par le GP : la variance qui permet d'estimer l'erreur d'interpolation comme l'illustre l'intervalle de confiance de la Figure 1.1. Notons qu'il s'agit uniquement d'une estimation de l'erreur et non d'un intervalle de confiance exact. En effet, la validité de l'intervalle de confiance est conditionnée par le respect des hypothèses, notamment celle supposant que la fonction f est la réalisation d'un GP.

Dans la suite, il est proposé de conserver les notations suivantes : un GP conditionné sera par la suite simplement noté $\mathcal{G}$ et sera défini par sa moyenne $\mu(\mathbf{x})$, sa covariance $k(\mathbf{x}, \mathbf{x}')$ et sa variance $\sigma^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x})$.

1.1.3.3 Optimisation par exploitation du méta-modèle

Grâce à la moyenne du GP μ , il est possible d'obtenir une approximation d'une fonction objectif f que l'on cherche à minimiser. Plusieurs solutions sont alors possibles pour résoudre le problème d'optimisation à l'aide du méta-modèle :

- Une méthode où le méta-modèle n'est pas amélioré, la solution du problème d'optimisation est évaluée sur l'information fournie par le méta-modèle uniquement.
- Une méthode où le méta-modèle est enrichi à l'optimum fourni par la méthode précédente et on itère ensuite jusqu'à convergence. Cette méthode sera nommée *Surrogate Based Optimization (SBO)* par la suite.
- Une méthode où l'enrichissement est effectué sur la base d'un compromis entre exploration du domaine de conception $\mathcal{X}$ (ou espace de design) et exploitation du méta-modèle. Pour cela, l'information fournie par la variance du GP est utilisée et nous nous limiterons à la méthode qui s'appelle *Efficient Global Optimization (EGO)*.

1.1.3.3.1 Optimisation directe du méta-modèle Une première idée naïve serait de remplacer le problème d'optimisation, rendu difficile par le coût numérique de f , par un problème simplifié en substituant f avec son approximation $\hat{f} = \mu$. On se retrouve dans ce cas avec le problème d'optimisation suivant :

$$\hat{\mathbf{x}}_* \in \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmin}} \mu(\mathbf{x}) \quad (1.27)$$

Comme la moyenne du GP est obtenue à partir d'un calcul analytique, son évaluation est peu coûteuse ce qui permet d'utiliser une multitude de méthodes d'optimisation comme par exemple les algorithmes basés sur le gradient présentés dans la Section 1.1.2. Ce problème simplifié est illustré sur la Figure 1.2. Cependant, l'utilisation de cette solution requiert de s'assurer que l'on obtienne une bonne approximation de la fonction f grâce à la moyenne μ sur l'ensemble du domaine de conception $\mathcal{X}$. Une mauvaise approximation de la fonction objectif peut entraîner la convergence de l'optimisation sur un minimum local comme cela est illustré sur la Figure 1.3. De plus, on observe sur la Figure 1.2 que même si le méta-modèle propose un point proche de la solution exacte, l'obtention d'une valeur précise nécessite un méta-modèle très précis sur l'ensemble du domaine $\mathcal{X}$. Les zones mal représentées par le méta-modèle peuvent être déterminées par l'information fournie par la variance du GP ou en utilisant des méthodes de validation croisée. Ces informations pourront servir à améliorer la précision du GP en ajoutant des points au DoE initial. Le coût de calcul de cette approche peut toutefois être important car on cherche à valider la précision du méta-modèle sur l'ensemble de l'espace $\mathcal{X}$. Pour pallier cette difficulté, il est proposé d'enrichir le méta-modèle sur la base d'un critère que nous allons définir par la suite.

1.1.3.3.2 Surrogate Based Optimization L'idée de la SBO est de calculer la fonction objectif au nouveau point $\hat{\mathbf{x}}_*$ solution du problème (1.27), et d'enrichir le méta-modèle grâce à cette nouvelle information. Notons que ce point est obtenu par l'information de la moyenne, sans prendre en compte les incertitudes liées à l'interpolation. Ce processus est réitéré jusqu'à convergence. Cependant, une mauvaise approximation initiale de la fonction objectif peut nuire à la convergence de l'algorithme. Par exemple, le méta-modèle obtenu sur la Figure 1.3 propose d'enrichir autour du minimum local et non autour du

minimum global amenant l'algorithme à converger vers le mauvais minimum de la fonction. Il est donc nécessaire d'avoir initialement une bonne approximation de la fonction objectif sur l'ensemble du domaine paramétrique $\mathcal{X}$.

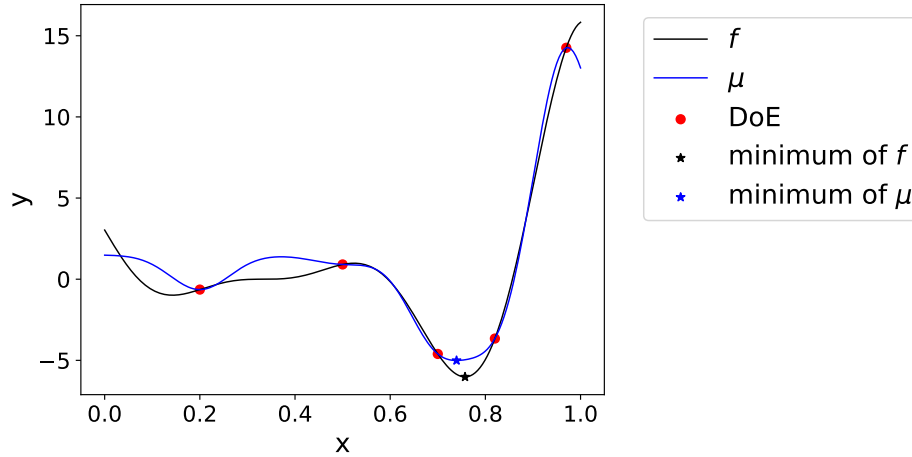


FIGURE 1.2 – Illustration de l'utilisation de méta-modèle pour l'optimisation globale sur une fonction à une variable définie par FORRESTER et collab. [2008].

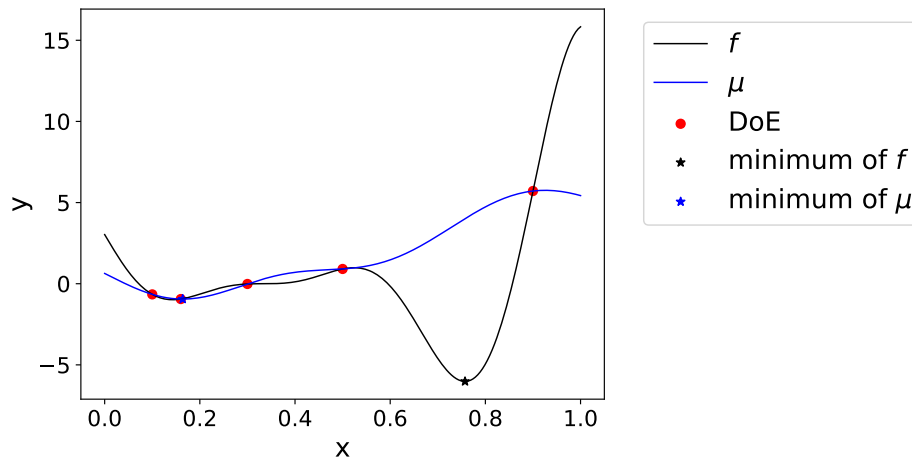


FIGURE 1.3 – Illustration de la non-convergence de l'optimisation globale par méta-modèle sur le même modèle mais avec des données d'apprentissage mal réparties.

Finalement, ne pas prendre en compte l'incertitude du méta-modèle lors de l'enrichissement conduit à la génération d'un DoE de trop grande taille pour représenter l'ensemble du domaine. Une autre méthode, EGO, optimise la position des points du DoE en combinant les aspects liés à l'exploration de l'espace paramétrique et l'exploitation du méta-modèle. Cela permet d'atténuer la sensibilité du résultat de l'optimisation au choix du DoE initial.

1.1.3.3.3 Efficient Global Optimization Une solution proposée dans la littérature pour parvenir à équilibrer l'exploration et l'exploitation est d'utiliser le critère d'Amélioration espérée, ou Expected Improvement (EI) proposé par MOČKUS [1975]; JONES et collab. [1998]; FORRESTER et collab. [2008]. Cette idée est développée au sein de l'optimisation bayésienne avec de nombreux ajouts tel que des krigeages avec des mélanges d'experts (BETTEBGHOR et collab. [2011]), des adaptations permettant de prendre en compte les

contraintes (PICHENY et collab. [2016]; PRIEM et collab. [2020]) ou encore des appels groupés à la fonction objectif (GINSBOURGER et collab. [2010]). Ces méthodes mettent en oeuvre une fonction d'acquisition (ici l'EI) qui sera optimisée afin de définir le point qui sera ajouté dans le DoE. Ceci permet de réduire les incertitudes liées à l'interpolation au fur et à mesure des itérations. D'autres fonctions d'acquisition ont été proposées dans la littérature comme *Watson Barnes 2* (WB2) (WATSON et BARNES [1995]), *Watson Barnes 2 normalisée*, ou *Scaled Watson Barnes 2* (WB2S) (BARTOLI et collab. [2019]).

L'EI repose sur le fait que le GP définit, en tout point $\mathbf{x} \in \mathcal{X}$, une loi gaussienne caractérisée par sa moyenne $\mu(\mathbf{x})$ et sa variance $\sigma(\mathbf{x})$. L'amélioration est une variable aléatoire qui cherche à estimer l'amélioration de la valeur du minimum courant dans le DoE en tout point $\mathbf{x} \in \mathcal{X}$. Elle est définie par :

$$\mathcal{J}(\mathbf{x}) = \max(f_{\min} - \mathcal{G}(\mathbf{x}), 0) \quad (1.28)$$

où $f_{\min}$ est le minimum de la fonction obtenu sur le DoE : $f_{\min} = \min_{i=1,\dots,d} f_i$.

Il est alors possible de calculer explicitement l'espérance de cette quantité, nommée EI, à partir de la moyenne et de la variance du processus gaussien :

$$\mathbb{E}(\mathcal{J}(\mathbf{x})) = \begin{cases} (f_{\min} - \mu(\mathbf{x}))\Phi\left(\frac{f_{\min} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right) + \sigma(\mathbf{x})\varphi\left(\frac{f_{\min} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right) & \text{si } \sigma(\mathbf{x}) > 0 \\ 0 & \text{si } \sigma(\mathbf{x}) = 0 \end{cases} \quad (1.29)$$

où $\Phi(\cdot)$ et $\varphi(\cdot)$ sont respectivement la fonction de répartition et la densité de probabilité d'une loi normale centrée réduite. Dans cette expression, on peut identifier le compromis entre exploitation et exploration. En effet, $(f_{\min} - \mu(\mathbf{x}))\Phi\left(\frac{f_{\min} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right)$ est prédominant lorsque $\mu(\mathbf{x}) \ll f_{\min}$ privilégiant l'exploitation et $\sigma(\mathbf{x})\varphi\left(\frac{f_{\min} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right)$ est prédominant lorsque la variance $\sigma(\mathbf{x})$ est importante et donc lorsque l'incertitude du méta-modèle est importante. Ce critère permet de définir le point le plus intéressant du point de vue de l'optimisation qui permettra d'enrichir le méta-modèle. Pour résoudre le problème d'optimisation, des enrichissements successifs sont effectués au cours des itérations. Néanmoins, il est difficile de déterminer un critère d'arrêt pour ce type d'algorithme. En effet, une idée serait de se baser sur le maximum de l'EI, seulement, ce critère est difficile à estimer. En effet, comme on peut l'observer sur l'itération 6 de la Figure 1.4, la forme de l'EI se rapproche de plus en plus de pics isolés au fur et à mesure des itérations. Optimiser une telle fonction devient alors extrêmement coûteux car cela requiert d'utiliser des algorithmes par méta-modèles locaux avec un grand nombre de *multi-start*, d'autant plus quand le nombre de variables de design est grand. Ainsi, une faible valeur du maximum de l'EI peut simplement être donnée par un échec de son optimisation. Le critère d'arrêt privilégié est donc un nombre d'itérations maximal n_{iter}^{max} qui permet de définir un budget sur le nombre d'appels à f possible.

Les différentes étapes de l'algorithme EGO, noté Algorithme 3, sont les suivantes :

1. À l'initialisation, un DoE est construit à partir d'un échantillon $\mathbf{x}_{DoE}$ et de leurs images par la fonction $f : \mathbf{x}_{DoE}$.
2. À partir du DoE, un GP est conditionné pour obtenir un méta-modèle de la fonction objectif.
3. La moyenne et la variance du GP permettent de calculer l'amélioration espérée qui est maximisée. Ce point maximum est considéré comme le plus intéressant à explorer d'après le méta-modèle et permet d'enrichir le GP.

4. Pour cela, la vraie valeur de la fonction objectif est calculée en ce point et un GP est construit à partir du DoE enrichi.
5. Tant que le budget défini n'est pas atteint, l'algorithme réitère à partir de l'étape 2.

Algorithm 3: EGO

```

input :  $\mathbf{x}_{\text{DoE}}, f_{\text{DoE}}, n_{\text{iter}}^{\text{max}}$ 
 $f_{\min} \leftarrow \min f_{\text{DoE}};$ 
 $n_{\text{iter}} \leftarrow 1$  while  $n_{\text{iter}} \leq n_{\text{iter}}^{\text{max}}$  do
   $\hat{\mu}, \hat{\sigma} \leftarrow \text{Gaussian Process } (\mathbf{x}_{\text{DoE}}, f_{\text{DoE}});$ 
   $\mathbf{x}_{n_{\text{iter}}} \leftarrow \underset{\mathbf{x} \in \mathcal{X}}{\text{argmax}} \mathbb{E}(\mathcal{J}(\mathbf{x}));$ 
  Add  $\mathbf{x}_{n_{\text{iter}}}$  to  $\mathbf{x}_{\text{DoE}};$ 
  Add  $f(\mathbf{x}_{n_{\text{iter}}})$  to  $f_{\text{DoE}};$ 
   $f_{\min} \leftarrow \min_{i=1, \dots, d} f_i;$ 
end
output:  $f_{\min}$ 

```

Finalement, la convergence de cet algorithme est illustrée sur le cas à une dimension proposé par FORRESTER et collab. [2008] où le DoE initial est le même que celui utilisé pour montrer la non-convergence de la méthode SBO de la Figure 1.3. Lors de la première itération, l'EI est plus importante autour du minimum local soulignant la capacité d'EGO à exploiter l'information fournie par le méta-modèle. Seulement, à partir de la deuxième itération, l'EI devient très importante autour du minimum global à cause de la forte incertitude liée au peu d'information disponible dans cette zone de l'espace paramétrique. Ainsi, la convergence du problème d'optimisation est assurée par la réduction des incertitudes d'interpolation grâce à l'exploration du domaine paramétrique permettant de s'affranchir d'une construction coûteuse d'un DoE initial riche.

Finalement, nous avons observé comment EGO parvient à diminuer la sensibilité du DoE initial par un enrichissement lié à une succession de phases d'exploration et d'exploitation. Néanmoins, cet algorithme a plusieurs inconvénients : premièrement, les critères d'arrêt, facilement définissables dans les algorithmes à gradient ne peuvent être étendus car aucune information n'est disponible sur le gradient ou sur le pas effectué. Deuxièmement, ces méthodes sont limitées aux fonctions dont une bonne approximation peut être obtenue par méta-modèle. Ceci limite donc le nombre de paramètres de design puisque qu'une fonction comportant un grand nombre d'entrées est plus difficile à estimer. Plusieurs recherches ont été proposées pour résoudre ce problème comme celles basées sur la *Partial Least Squares Regression* (BOUHLEL et collab. [2016]) ou sur la POD (BRACONNIER et collab. [2011]; ROY et collab. [2018]).

1.1.4 Bilan sur les méthodes d'optimisation

Dans cette section nous avons passé en revue plusieurs approches permettant de résoudre un problème d'optimisation par l'utilisation de méta-modèles locaux ou globaux. L'un des inconvénients majeurs des méta-modèles locaux est qu'il est nécessaire de construire le modèle approché à chaque itération. Or, dans le cadre de notre étude, la fonction objectif est obtenue à partir d'un code numérique coûteux, sans information sur le gradient et son évaluation par différences finies alourdit encore le coût numérique. En revanche, ces méthodes, très utilisées dans la littérature nous servons à définir des

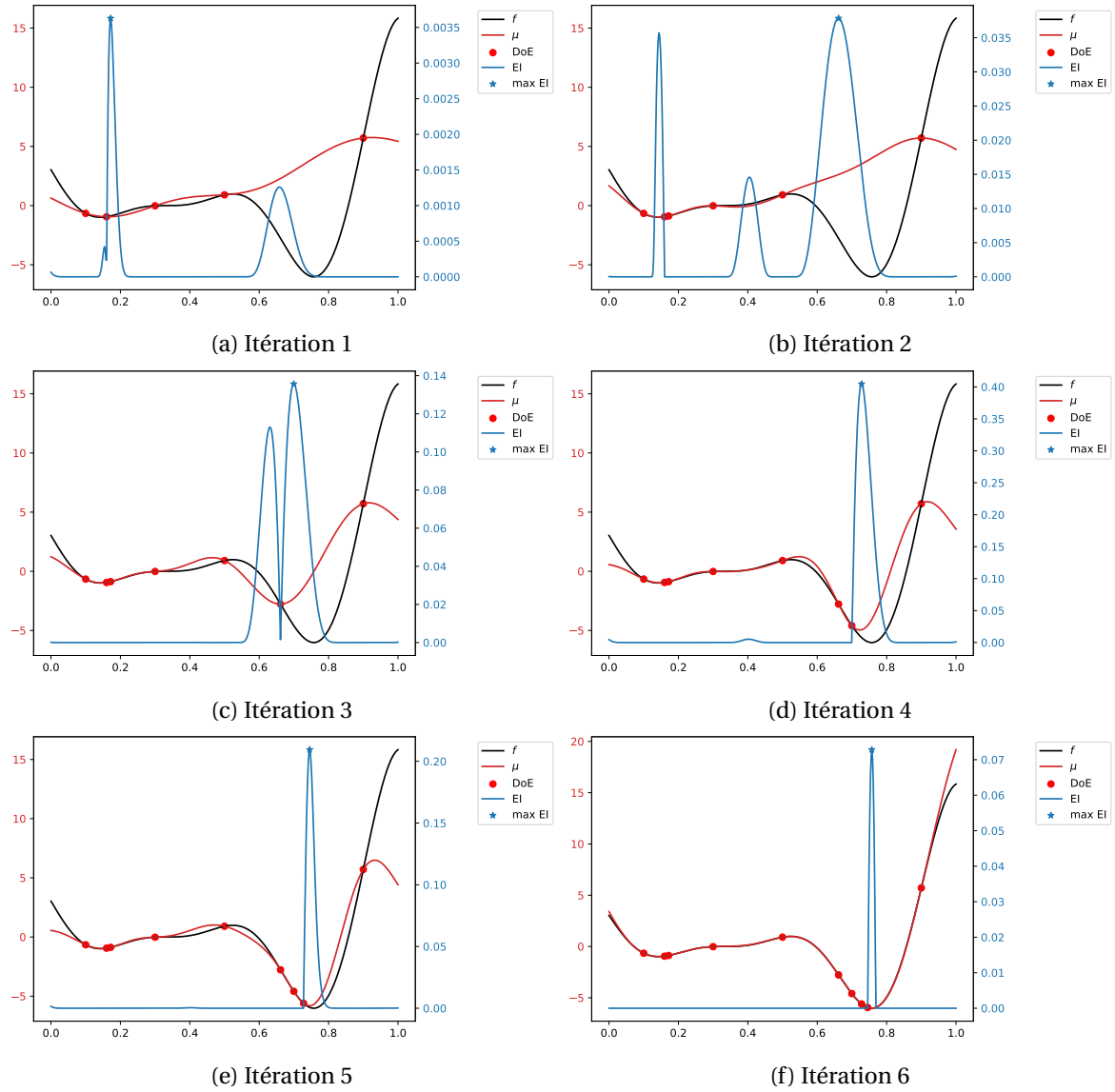


FIGURE 1.4 – Itérations de l'algorithme EGO sur un exemple 1D proposé par FORRESTER et collab. [2008] où l'amélioration espérée est maximisée pour obtenir le prochain point d'enrichissement. Après 6 itérations, le minimum global est identifié.

résultats de référence. D'autre part, les méta-modèles globaux permettent de fortement réduire le coût numérique de l'optimisation en limitant les appels à la fonction objectif (dans le cadre de l'optimisation sans connaissance du gradient). Cependant, nous avons observé la sensibilité de ces méthodes au DoE sur lequel le méta-modèle est construit. Pour y remédier, des critères tels que l'amélioration espérée proposent un compromis entre exploitation et exploration afin d'enrichir le méta-modèle au fur et à mesure du processus d'optimisation. Ces méthodes semblent donc être les plus intéressantes pour notre étude. L'un des objectifs de la thèse est de proposer une adaptation de ces méthodes pour l'optimisation multidisciplinaire, que nous allons détailler dans la prochaine partie.

1.2 Optimisation multidisciplinaire

Les objectifs de cette section sont :

- définir le cadre de l'Optimisation multidisciplinaire, ou *MultiDisciplinary Design Optimization* (MDO) en détaillant l'Analyse multidisciplinaire, ou *MultiDisciplinary Analysis* (MDA) paramétrée et les interactions entre les différentes disciplines.
- étudier les différentes solutions apportées dans la littérature pour résoudre ce type de problème MDO.
- présenter de manière approfondie une méthode récente, développée par DUBREUIL et collab. [2020], nommée *Efficient Global Multidisciplinary Optimization* (EGMDO) qui servira de socle pour nos développements futurs. En effet, l'un des objectifs de cette thèse est d'adapter cette méthode aux variables de couplage de grande dimension.

L'objectif est de définir la MDO et à travers elle la MDA paramétrée. La MDA est une méthode permettant de prendre en compte l'interaction entre plusieurs modèles physiques afin d'explicitier des phénomènes aéroélastiques comme le comportement d'une aile d'avion en vol (COELHO et collab. [2008]; SCHOLTEN et HARTL [2021]), d'une turbomachine (CAMPBELL et PATERSON [2011]) d'un parachute (TEZDUYAR et OSAWA [2001]). D'autres interactions ont été proposées afin de simuler la performance d'un avion complet (DUBREUIL et collab. [2018b]).

Dans un premier temps, nous allons détailler la MDA paramétrée ainsi que des méthodes de résolution des problèmes couplés. Par la suite, nous détaillerons la MDO qui a pour objectif d'optimiser une fonction de performance obtenue à partir de la solution de la MDA. Nous passerons en revue plusieurs méthodes adaptées à la résolution de ce problème d'optimisation particulier. L'objectif est de montrer quelles sont les méthodes adaptées au cadre de la thèse. Nous nous focaliserons plus particulièrement sur la méthode *Efficient Global Multidisciplinary Optimization* (EGMDO), non conforme à notre cadre, mais dont l'adaptation est une des contributions de la thèse.

1.2.1 Introduction

1.2.1.1 Définition de la MDA paramétrée

L'analyse multidisciplinaire met en oeuvre l'interaction de plusieurs modèles physiques modélisée par un système d'équations couplées à résoudre. Ces différents modèles physiques, aussi nommés disciplines, au nombre de n_d , sont résolus par des solveurs f^i , $i = 1, \dots, n_d$. Ils dépendent de paramètres globaux $\mathbf{x}^g \in \mathcal{X}^g \subset \mathbb{R}^{q^g}$ et de paramètres locaux $\mathbf{x}^i \in \mathcal{X}^i \subset \mathbb{R}^{q^i}$ propres à chaque discipline où q^g est le nombre de paramètres globaux et q^i est le nombre de paramètres locaux affectés à la discipline i . L'ensemble des paramètres globaux et locaux $\{\mathbf{x}^g, \mathbf{x}^1, \dots, \mathbf{x}^{n_d}\}$ est noté $\mathbf{x}$ dans la suite et l'on suppose que $\mathcal{X} = \mathcal{X}^g \times \prod_{i=1}^{n_d} \mathcal{X}^i$.

Ces paramètres vont influencer la solution du système non-linéaire, aussi appelé MDA, suivant :

$$\left\{ \mathbf{y}^i = f^i(\mathbf{x}^g, \mathbf{x}^i, \mathbf{y}^{c^{(i)}}), i = 1, \dots, n_d \quad \forall \mathbf{x} = \{\mathbf{x}^g, \mathbf{x}^1, \dots, \mathbf{x}^{n_d}\} \in \mathcal{X} \right. \quad (1.30)$$

où f^i est le solveur modélisant la $i^{\text{ème}}$ discipline, $\mathbf{y}^i$ sont les solutions de ces solveurs aussi appelées variables d'état et $\mathbf{y}^{c^{(i)}}$ sont les variables de couplage intervenant sur la $i^{\text{ème}}$ discipline. Ainsi, $c^{(i)}$ est un sous-ensemble de $\{1, \dots, n_d\}$ dont i ne fait pas partie. Ces

variables de couplage sont définies dans des espaces $\mathcal{C}^i \subset \mathbb{R}^{d^i}$ où d^i est la dimension de la variable de couplage y^i . Nous soulignons ici la complexité de la variété $\mathcal{C}^i$ qui dépend du solveur f^i , de l'espace paramétrique $\mathcal{X} \times \mathcal{X}^i$ ainsi que du couplage entre les différentes disciplines et donc des variétés $\mathcal{C}^j$, $j \in \mathbf{c}^{\{i\}}$. La solution de ce système non linéaire est indiquée par une étoile en indice : $\{y_*^1(x), \dots, y_*^{n_d}(x)\}$. Une illustration de ce système est proposée en Figure 1.5 où le système est simplifié pour faire intervenir uniquement deux disciplines. Dans ce cas, les ensembles de couplage $\mathbf{c}^{\{i\}}$ sont simples car $\mathbf{c}^{\{1\}} = \{2\}$ et $\mathbf{c}^{\{2\}} = \{1\}$.

Dans nos travaux, plusieurs contraintes ont été définies :

- Nous souhaitons rester le moins intrusif possible afin d'obtenir une méthode pouvant être adaptée à différents problèmes multidisciplinaires. Nous supposons que les disciplines f^i sont résolues par des solveurs boîtes noires limitant l'utilisation de méthodes intrusives. En revanche, nous supposons que le couplage peut être modifié.
- Nous supposons que les appels aux boîtes noires sont coûteux : minimiser leurs appels sera un des objectifs de la thèse.
- Les solutions des solveurs sont des champs discrétisés donc des vecteurs de grande dimension (en pratique, $d^i > 1000$, $i = 1, \dots, n_d$).

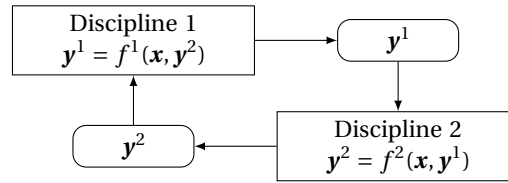


FIGURE 1.5 – Illustration de la MDA faisant interagir deux disciplines et deux variables de couplage y^1 et y^2 .

1.2.1.2 Exemple à deux disciplines

Un exemple que nous pouvons proposer est celui de l'*Interaction fluide-structure*, ou *Fluid-structure interaction (FSI)* pour simuler le comportement d'une aile d'avion en vol. C'est un cadre courant que l'on retrouve notamment dans COELHO et collab. [2008]; COELHO et collab. [2009]; BONCORAGLIO et collab. [2021]; SCHOLTEN et HARTL [2021]. Ce problème fait intervenir deux disciplines : l'aérodynamique et la structure. L'objectif est d'obtenir l'équilibre aéroélastique de l'aile, c'est à dire la déformée de la structure et les efforts aérodynamiques qui s'équilibrent. Dans ce contexte, les paramètres globaux peuvent être par exemple des variables influençant la géométrie de l'aile. Les paramètres locaux peuvent être des caractéristiques mécaniques pour le solveur structure ou l'angle d'attaque de l'aile pour l'aérodynamique. On observe alors un couplage entre ces deux disciplines illustré sur la Figure 1.6.

1.2.1.3 Approche monolithique et partitionnée

Nous allons passer en revue différentes solutions possibles pour la résolution numérique du problème (1.30). Une solution proposée dans la littérature est l'approche monolithique comme cela peut être appliqué dans le cas de la FSI (BATHE et collab. [1995]; BATHE et ZHANG [2004]). Dans cette approche, les équations physiques des différents modèles disciplinaires sont résolues par un même solveur. Ce solveur monolithique doit

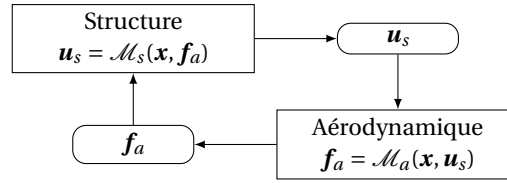


FIGURE 1.6 – Illustration de la FSI dans le cas de l'aéroélasticité où u_s est le déplacement de la structure, f_a sont les efforts aérodynamiques, $\mathcal{M}_a$ est le modèle aérodynamique et $\mathcal{M}_s$ est le modèle structure.

alors prendre en compte les différentes interactions et la résolution des équations physiques associées. L'avantage étant que la circulation des informations est prise en compte par le solveur et la résolution s'effectue sur un maillage unique. Néanmoins, lorsque l'on souhaite changer un des modèles physiques (par exemple, pour l'aérodynamique, passer des équations d'Euler aux équations de Navier-Stokes) ou ajouter un modèle physique supplémentaire (par exemple en prenant en compte l'interaction entre la propulsion et l'aérodynamique dans la FSI), il est nécessaire d'adapter le solveur monolithique.

Une autre approche repose sur le partitionnement des modèles (DEGROOTE [2013]). Cela permet de découpler le modèle couplé complexe en une combinaison de plusieurs modèles disciplinaires utilisant des solveurs disciplinaires dédiés. Ceci a plusieurs avantages : dans un premier temps, cela permet d'adapter un solveur dédié à chaque modèle et de moduler ainsi la précision et les caractéristiques du solveur facilement en intervenant avec un autre solveur. Dans un second temps, cela permet de faire intervenir des experts de chaque modèle afin de proposer le solveur le plus adapté. Ainsi, il est possible de réutiliser des solveurs ayant déjà fait l'objet d'un développement dédié pour les rendre fiables et optimisés au lieu de développer un solveur monolithique spécifique. Par contre, cette approche rend l'interaction entre les disciplines difficile à modéliser car l'information doit pouvoir circuler entre chaque discipline. Par exemple, dans le cas de la FSI, le maillage du solveur aérodynamique est déformé à partir du déplacement obtenu sur le maillage du solveur structure. De la même façon les efforts aérodynamiques sont obtenus sur le maillage du solveur aérodynamique, un transfert sur le maillage du solveur structure est alors requis. Ce transfert peut, par exemple, être défini par une interpolation entre les maillages par RBF [LOMBARDI et collab., 2013; RENDALL et ALLEN, 2008].

L'objectif de la thèse étant de développer une méthode de résolution de la MDO, et donc de la MDA, générique, nous choisissons de nous intéresser à l'approche partitionnée car celle-ci sera adaptable à plusieurs problèmes multiphysiques. Cette approche partitionnée permet de plus d'utiliser des solveurs disciplinaires en tant que boîtes noires conformément au cadre de nos travaux. Nous allons alors passer en revue les approches permettant de transférer les données entre les modèles disciplinaires.

1.2.1.4 Les variables de couplage

Plusieurs solutions sont possibles pour permettre le transfert des informations entre les modèles disciplinaires. Un premier cas se présente si les variables de couplage sont les sorties des solveurs disciplinaires $y^{c(i)}$, $i = 1, \dots, n_d$ qui sont des champs discrétisés sur les maillages correspondants. Dans ce cas, comme les solveurs disciplinaires utilisent généralement des maillages différents, une étape d'interpolation entre les différents maillages est nécessaire. Dans le deuxième cas, les variables de couplage sont obtenues en extrayant une partie de l'information des champs discrétisés à travers une fonction explicite. Cela impose donc une étape supplémentaire où l'information de chaque solveur discipli-

naire $\mathbf{y}^i$ est extraite à partir d'une fonction explicite $p^i : \mathbf{z}^i = p^i(\mathbf{y}^i)$ où la dimension de $\mathbf{z}^i$ est choisie afin d'être bien inférieure à celle de $\mathbf{y}^i$. La MDA serait alors la solution du problème suivant :

$$\{\mathbf{z}^i = p^i \circ f^i(\mathbf{x}^g, \mathbf{x}^i, \mathbf{z}^{c^{(i)}}), i = 1, \dots, n_d \quad \forall \mathbf{x} = \{\mathbf{x}^g, \mathbf{x}^1, \dots, \mathbf{x}^{n_d}\} \in \mathcal{X} \quad (1.31)$$

où les variables de couplage sont désormais les $\mathbf{z}^i$, $i = 1, \dots, n_d$ supposées de faible dimension.

Dans l'exemple de l'aéroélasticité statique sur une aile d'avion, la différence entre l'utilisation des champs discrétisés ou des représentations peut s'illustrer de la façon suivante. L'information fournie par un solveur structure peut être soit le champ de déplacement sur le maillage structure (variable de couplage de très grande dimension) soit quelques variables représentatives du déplacement tel que la torsion et le déplacement en bout d'aile en modélisant la structure de l'aile par une poutre rectiligne uniforme.

Dans le cadre de notre étude, nous rappelons que l'objectif est d'utiliser des modèles haute-fidélité où un couplage avec les variables d'état permet de gagner en précision par rapport à une représentation dont le choix est complexe. Par exemple, dans l'exemple précédent, bien que le calcul des déplacements puisse être effectué avec un modèle de structure précis, restreindre les informations transmises à l'aérodynamique à quelques variables engendre une perte d'information. Cette perte est difficile à estimer et ne permet pas de représenter parfaitement le couplage mis en oeuvre. Toutefois, peu de méthodes permettent de résoudre un problème couplé avec de nombreuses variables de couplage, c'est pourquoi nous allons aussi présenter des méthodes limitées à un faible nombre de variables de couplage.

1.2.2 Résolution du système couplé

Plusieurs solutions existent pour parvenir à calculer la solution d'un tel système non-linéaire. En effet, la MDA peut s'écrire sous la forme :

$$\mathbf{y}^1, \dots, \mathbf{y}^{n_d} = F(\mathbf{x}, \mathbf{y}^1, \dots, \mathbf{y}^{n_d}) \quad (1.32)$$

où F est une fonction dont les composantes sont les fonctions f^i avec les variables de couplage correspondantes. Ce système de point fixe peut être résolu par une méthode itérative adaptée. L'objectif est de créer une suite de variables de couplage $\mathbf{y}_j^1, \dots, \mathbf{y}_j^{n_d}$, $j = 1, \dots, n_{iter}$ convergeant vers la solution de la MDA : $\mathbf{y}_*^1, \dots, \mathbf{y}_*^{n_d}$. La différence principale par rapport à un système de point fixe vectoriel classique est que celui-ci peut être décomposé en plusieurs blocs comme dans l'Eq (1.30) et des méthodes peuvent exploiter cette particularité. Ces méthodes sont dénommées variantes non linéaires par bloc des méthodes de point fixe linéaires (MARTINS et NING [2021]).

1.2.2.1 Cas test analytique

Afin de mettre en évidence les différentes méthodes, nous allons introduire un modèle de MDA paramétré analytique. Pour cela, un seul paramètre de design $x \in [-5; 5]$ est considéré. Dans ce cas, $\mathcal{X}^g = [-5; 5]$, et il n'y a pas de paramètres locaux $\mathbf{x}^i$. La solution de la MDA est donnée par l'équation couplée suivante :

$$\begin{cases} y^1 &= (x)^2 - \cos\left(\frac{y^2}{2}\right) \\ y^2 &= -\frac{(x)^2}{2} + y^1 \end{cases} \quad (1.33)$$

La solution de la MDA est illustrée pour $x = -3$ sur la Figure 1.7. La solution de la MDA est le point d'intersection des deux solveurs disciplinaires soit l'étoile rouge.

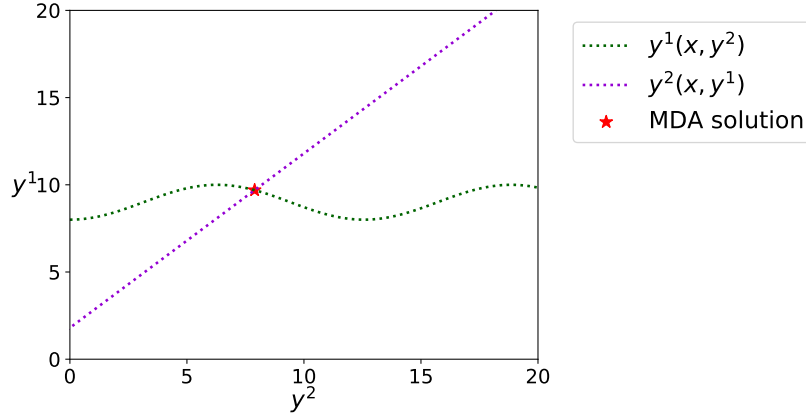


FIGURE 1.7 – Illustration de la MDA pour le cas test analytique avec $x = -3$ où les deux solveurs disciplinaires sont définis par l'Eq. (1.33). Les deux réponses des solveurs disciplinaires sont illustrées en fonction des variables de couplage y^1 et y^2 . La solution de la MDA est représentée par une étoile rouge.

1.2.2.2 Méthode de Jacobi

La méthode non linéaire de Jacobi permet de résoudre ce système en effectuant des appels répétés à la fonction F :

$$\mathbf{y}_{j+1}^1, \dots, \mathbf{y}_{j+1}^{n_d} = F(\mathbf{x}, \mathbf{y}_j^1, \dots, \mathbf{y}_j^{n_d}) \quad (1.34)$$

où $\mathbf{y}_j^1, \dots, \mathbf{y}_j^{n_d}$ est le vecteur des variables de couplage à l'itération j . L'avantage de cette méthode est qu'elle est peu intrusive mais requiert d'avoir à disposition des variables de couplage initiales c.a.d. $\mathbf{y}_1^1, \dots, \mathbf{y}_1^{n_d}$ permettant d'amorcer le processus itératif. Le critère d'arrêt de cet algorithme est généralement basé sur l'erreur e_j sur les variables de couplage entre deux itérations successives :

$$e_j = \max_{i=1, \dots, n_d} \frac{\|\mathbf{y}_{j+1}^i - \mathbf{y}_j^i\|_2}{\|\mathbf{y}_{j+1}^i\|_2} \quad (1.35)$$

La solution obtenue est suffisamment proche de la solution du problème couplé lorsque $e_j < \epsilon_{\text{MDA}}$ où ϵ_{MDA} est une tolérance choisie par l'utilisateur.

Les étapes de l'algorithme de Jacobi sont détaillées dans l'Algorithme 4. À partir de candidats initiaux pour l'ensemble des variables de couplage $\mathbf{y}_0^i$, $i = 1, \dots, n_d$, un processus itératif est initialisé : les solveurs disciplinaires sont appelés en utilisant comme entrées les variables de couplage obtenues à l'itération précédente. Notons qu'il est possible de lancer le calcul des différents solveurs en parallèle puisque les variables de couplage utilisées pour chacun des solveurs sont celles utilisées pour l'itération j et sont donc communes entre l'ensemble des solveurs.

Notons que la convergence de cet algorithme est démontrée dans le cas où F est une application contractante par le théorème de Picard-Banach :

Théorème 1 Soit $(\mathcal{E}, d)$ un espace métrique complet, et $F : \mathcal{E} \rightarrow \mathcal{E}$ une application contractante, c'est-à-dire qu'il existe $k_F \in [0, 1[$ tel que pour tout $(\mathbf{y}', \mathbf{y}) \in \mathcal{E}$, $d((F(\mathbf{y}'), F(\mathbf{y}))) \leq k_F d(\mathbf{y}', \mathbf{y})$

Alors F possède un unique point fixe $\mathbf{I}$. De plus, toute suite définie par $\mathbf{y}_0, \mathbf{y}_{n+1} = F(\mathbf{y}_n)$, converge vers cet unique point fixe, et on a les estimations suivantes :

$$\begin{aligned} d(\mathbf{y}_n, \mathbf{I}) &\leq k_F^n d(\mathbf{y}_0, \mathbf{I}) \\ d(\mathbf{y}_n, \mathbf{I}) &\leq \frac{k_F}{1-k_F} d(\mathbf{y}_n, \mathbf{y}_{n-1}) \end{aligned} \quad (1.36)$$

Néanmoins, démontrer que l'application F est contractante est très difficile et intrusif lorsque celle-ci est obtenue à partir de solveurs numériques. Nous supposons donc que les hypothèses de ce théorème sont valides même si celles-ci ne seront pas vérifiées dans la suite.

Algorithm 4: Non linear block Jacobi algorithm

```

input :  $\mathbf{x}, \mathbf{y}_0^i, i = 1, \dots, n_d$ 
 $j \leftarrow 0$ 
 $e_j \leftarrow 1 + \epsilon_{\text{MDA}}$ 
while  $e_j > \epsilon_{\text{MDA}}$  do
  for  $i = 1, \dots, n_d$  do
     $\mathbf{y}_{j+1}^i \leftarrow f^i(\mathbf{x}^g, \mathbf{x}^i, \mathbf{y}_k^{c\{i\}})$ 
  end
   $e_j \leftarrow \max_{i=1, \dots, n_d} \frac{\|\mathbf{y}_{j+1}^i - \mathbf{y}_j^i\|_2}{\|\mathbf{y}_{j+1}^i\|_2}$ 
   $j \leftarrow j + 1$ 
end
output:  $\mathbf{y}_j^1, \dots, \mathbf{y}_j^{n_d}$ 
```

Finalement, cette méthode permet d'obtenir la solution du problème couplé en étant très peu intrusive. Seulement, cette méthode a une vitesse de convergence assez faible : un grand nombre d'itérations est nécessaire pour obtenir une solution de la [MDA](#). Des méthodes ont été proposées afin de réduire le nombre d'itérations en intervertissant l'ordre d'évaluation des disciplines ([BLOEBAUM \[1995\]](#)), ou d'accélérer la convergence comme la méthode d'Aitken ([AITKEN \[1927\]](#)).

1.2.2.3 Méthode de Gauss-Seidel

La méthode non linéaire par bloc de Gauss-Seidel est très similaire à celle de Jacobi, la seule différence est que, dès qu'une variable de couplage est modifiée, la nouvelle variable de couplage obtenue est utilisée par les autres solveurs. Cela permet d'utiliser une variable de couplage qui serait plus proche de la solution de la [MDA](#) puisqu'elle est obtenue après une itération de plus par rapport à la méthode de Jacobi. L'autre avantage de cette méthode est qu'il n'est pas nécessaire d'avoir un candidat pour l'ensemble des variables de couplage, uniquement les variables correspondants aux indices $\mathbf{c}^{(0)}$ sont nécessaires. Cependant, cette méthode ne permet plus d'appeler les solveurs en parallèle car au sein d'une même itération, les variables de couplage sont dépendantes les unes des autres. Cette méthode est donc à utiliser lorsqu'il n'est pas possible, notamment par une limitation en terme de moyens numériques, d'effectuer les calculs en parallèle. Le critère d'arrêt est le même que celui de la méthode de Jacobi. Cette méthode est détaillée dans l'Algorithme 5.

Algorithm 5: Non linear block Gauss Seidel algorithm

```

input :  $\mathbf{x}, \mathbf{y}_0^{c^{(1)}}$ 
 $j \leftarrow 0$ 
 $e_j \leftarrow 1 + \epsilon_{\text{MDA}}$ 
while  $e_j > \epsilon_{\text{MDA}}$  do
  for  $i = 1, \dots, n_d$  do
     $\mathbf{y}_{j+1}^i \leftarrow f^i(\mathbf{x}^g, \mathbf{x}^i, \mathbf{y}_{\text{GS}}^{c^{(i)}})$  with  $\forall k \in c^{(i)}, \mathbf{y}_{\text{GS}}^k = \begin{cases} \mathbf{y}_{j+1}^k & \text{if } k < i \\ \mathbf{y}_j^k & \text{else} \end{cases}$ 
  end
   $e_j \leftarrow \max_{i=1, \dots, n_d} \frac{\|\mathbf{y}_{j+1}^i - \mathbf{y}_j^i\|_2}{\|\mathbf{y}_{j+1}^i\|_2}$ 
   $k \leftarrow k + 1$ 
end
output:  $\mathbf{y}_j^1, \dots, \mathbf{y}_j^{n_d}$ 

```

1.2.2.4 Méthode de Newton

Finalement, la résolution du système (1.32) peut s'effectuer par une méthode de Newton faisant intervenir la jacobienne $\mathbf{J}_F$ de la fonction F . Dans ce cas, le processus itératif est défini de la façon suivante :

$$(\mathbf{J}_F(\mathbf{x}, \mathbf{Y}_j) - \mathbf{I})(\mathbf{Y}_{j+1} - \mathbf{Y}_j) = \mathbf{Y}_j - F(\mathbf{x}, \mathbf{Y}_j) \quad (1.37)$$

en notant $\mathbf{Y}_j$ le vecteur d'état des variables de couplage à l'itération j : $\mathbf{Y}_j = [y_j^1, \dots, y_j^{n_d}]$ et $\mathbf{I}$ la matrice identité. Dans ce cas, il est nécessaire d'obtenir la jacobienne de F à chaque itération et de résoudre le système linéaire associé. Les méthodes de quasi-Newton (**FLETCHER et POWELL [1963]**; **LIU et NOCEDAL [1989]**; **DAVIDON [1991]**) peuvent être utilisées pour obtenir une approximation de la jacobienne ou de son inverse.

1.2.2.5 Bilan sur les méthodes de résolution de l'analyse multidisciplinaire

Dans cette section, nous avons passé en revue plusieurs méthodes permettant de résoudre une MDA. Bien que la méthode de Newton soit la plus intéressante car elle donne une convergence très rapide vers la solution du problème couplé quand elle converge, celle-ci requiert d'obtenir la jacobienne dont le calcul peut être complexe si cette donnée n'est pas disponible. De plus, dans le cas d'un grand nombre de variables de couplage, le calcul de la jacobienne et de son inverse est délicat. Les méthodes de Gauss Seidel ou de Jacobi sont alors à privilégier puisqu'elles n'utilisent pas l'information de la jacobienne, en revanche leur convergence est plus lente bien que certaines solutions existent pour améliorer la vitesse de convergence (**AITKEN [1927]**; **BLOEBAUM [1995]**).

1.2.3 Résolution du problème d'optimisation multidisciplinaire

Nous proposons de passer en revue certaines méthodes proposées pour la résolution de problèmes d'optimisation multidisciplinaire dans une approche partitionnée. Les différentes solutions proposées dans la littérature sont présentées et un exemple analytique, utilisé par la suite, est introduit.

1.2.3.1 Cadre

L'analyse multidisciplinaire permet d'obtenir, à partir d'un paramètre global $\mathbf{x}^g \in \mathcal{X}^g \subset \mathbb{R}^{q^g}$ et des paramètres locaux $\mathbf{x}^i \in \mathcal{X}^i \subset \mathbb{R}^{q^i}$, la solution d'un problème couplé. A présent nous introduisons une fonction de performance $f^{obj} : \mathcal{X}^g \times \prod_{i=1}^{n_d} \mathcal{X}^i \times \prod_{i=1}^{n_d} \mathcal{C}^i \mapsto \mathbb{R}$ qui, à partir de la solution de la MDA, définit la fonction objectif f du problème d'optimisation :

$$f(\mathbf{x}) = f^{obj}(\mathbf{x}, y_*^1(\mathbf{x}), \dots, y_*^{n_d}(\mathbf{x})) \quad (1.38)$$

L'enjeu de la MDO est de résoudre ce problème en essayant de limiter le nombre d'appels aux solveurs disciplinaires f^i dont l'évaluation est coûteuse. Nous supposons que l'évaluation de la fonction f^{obj} requiert en revanche peu de ressources numériques.

1.2.3.2 Cas test analytique

Il est proposé de continuer sur le cas analytique proposé dans la Section 1.2.2.1 en lui ajoutant la fonction objectif donnée par :

$$f(x) = f^{obj}(x, y_*^1(x), y_*^2(x)) = \cos\left(\frac{y_*^1(x) + e^{-y_*^2(x)}}{\pi}\right) + \frac{x}{20} \quad (1.39)$$

où $y_*^1(x)$ et $y_*^2(x)$ sont la solution du système défini par l'Eq. (1.31). Cette fonction, dépendante de x est illustrée sur la Figure 1.8. Le minimum de la fonction est atteint en $x \approx -3$.

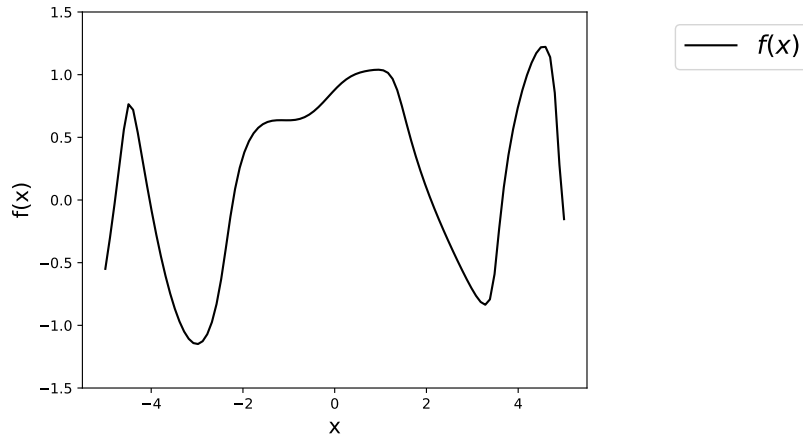


FIGURE 1.8 – Illustration de la fonction objectif du cas test analytique obtenue pour x variant entre -5 et 5 . On peut observer que le minimum global de cette fonction est atteint en $x \approx -3$.

1.2.4 Formulations d'optimisation multidisciplinaire

De nombreuses formulations ont été proposées dans la littérature pour résoudre ce type de problème. Plusieurs revues de ces méthodes ont aussi été proposées notamment dans BALLING et SOBIESZCZANSKI-SOBIESKI [1996]; MARTINS et LAMBE [2013]; BREITKOPF et COELHO [2013]; MARTINS et NING [2021] ainsi que plusieurs analyses sur les performances numériques et qualitatives de ces méthodes (YI et collab. [2008]; TEDFORD et MARTINS [2010]; DELBECQ et collab. [2020]). Nous allons classer ces différentes formulations selon deux critères :

- La façon dont est résolue la **MDA** : les approches couplées où la solution de la **MDA** est obtenue à partir d'une méthode itérative détaillée dans la Section 1.2.1.1 par opposition aux approches découplées où l'objectif est d'obtenir la solution de la **MDA** uniquement pour le paramètre optimal de la **MDO**. L'approche devra alors résoudre le problème couplé tout en cherchant le paramètre optimal. Par exemple, des contraintes d'égalité des variables de couplage peuvent être ajoutées au problème d'optimisation afin que celles-ci vérifient les équations de la **MDA**.
- Le nombre de niveaux d'optimisation avec les formulations à un niveau, où un seul optimiseur est utilisé pour la résolution de la **MDO** par opposition aux formulations multi-niveaux où des problèmes d'optimisation sur les disciplines sont ajoutés à celui de la recherche des paramètres de design optimal. L'objectif est de coupler ces problèmes afin de distribuer la complexité du problème. Plusieurs optimiseurs sont alors mis en place.

À partir de ces deux critères, nous identifions quatre groupes de formulation :

- La formulation **Optimisation multidisciplinaire faisable, ou Multidisciplinary Feasible (MDF)** (CRAMER et collab. [1994] ; BALLING et SOBIESZCZANSKI-SOBIESKI [1996]) qui consiste à utiliser un seul algorithme d'optimisation sur les paramètres $\mathbf{x}$ du système. La fonction objectif est alors estimée à partir de la solution de la **MDA** obtenue par un solveur dédié pour chaque choix de paramètres $\mathbf{x}$.
- La formulation **Individual Discipline Feasible (IDF)** (CRAMER et collab. [1994] ; ALEXANDROV et LEWIS [2004]), cherche à trouver la solution de la **MDO** et la solution de la **MDA** simultanément. Pour cela, des contraintes d'égalité sont ajoutées pour que les résidus des solveurs disciplinaire soient nuls. La formulation impose donc de manipuler les paramètres de design et les variables de couplage en même temps. La formulation **Modular Analysis and Unified Derivatives (MAUD)** HWANG et MARTINS [2018], dérivée de la formulation **IDF**, utilise des méthodes de linéarisation du problème couplé par le calcul des dérivées des résidus des solveurs disciplinaires afin de définir une direction pour la formulation **IDF**. Une autre extension de la formulation **IDF** est proposée via la formulation **Simultaneous Analysis and Design (SAND)** (HAFTKA [1985]).
- Les formulations **Concurrent SubSpace Optimization (CSSO)** (SOBIESZCZANSKI-SOBIESKI [1988]), **Bilevel Integrated System Synthesis (BLISS)** (SOBIESZCZANSKI-SOBIESKI et collab. [2000]), **Asymmetric Subspace Optimization (ASO)** (CHITTICK et MARTINS [2009]) définissent plusieurs niveaux de problème mais la solution de la **MDA** est donnée par un algorithme dédié pour chaque paramètre. Ce sont des approches **MDF** distribuées.
- Les approches **Collaborative Optimization (CO)** (BRAUN [1996] ; BRAUN et collab. [1996] ; ALLISON et collab. [2005]), **Analytical Target Cascading (ATC)** (KOKKOLARAS et collab. [2002] ; TOSSERAMS et collab. [2008]) sont des approches **IDF** distribuées.

Les principales méthodes sont synthétisées sur la Figure 1.9 mettant en avant les différentes catégories de formulations existantes. Des méthodes ont aussi été développées afin d'accélérer l'optimisation en utilisant un calcul du gradient par une méthode par adjoint (MARTINS et collab. [2003] ; MARTINS et collab. [2005] ; MARTINS et HWANG [2013]). Seulement ces méthodes nécessitent de calculer l'adjoint des solveurs disciplinaires ce qui est intrusif et n'est donc pas considéré dans la suite. Notons que les principales méthodes à plusieurs niveaux ne résolvent pas le même problème d'optimisation et qu'il n'y a pas de preuve de convergence vers la solution du problème initial. Pour cette raison, celles-ci ne seront pas développées dans la suite de la thèse.

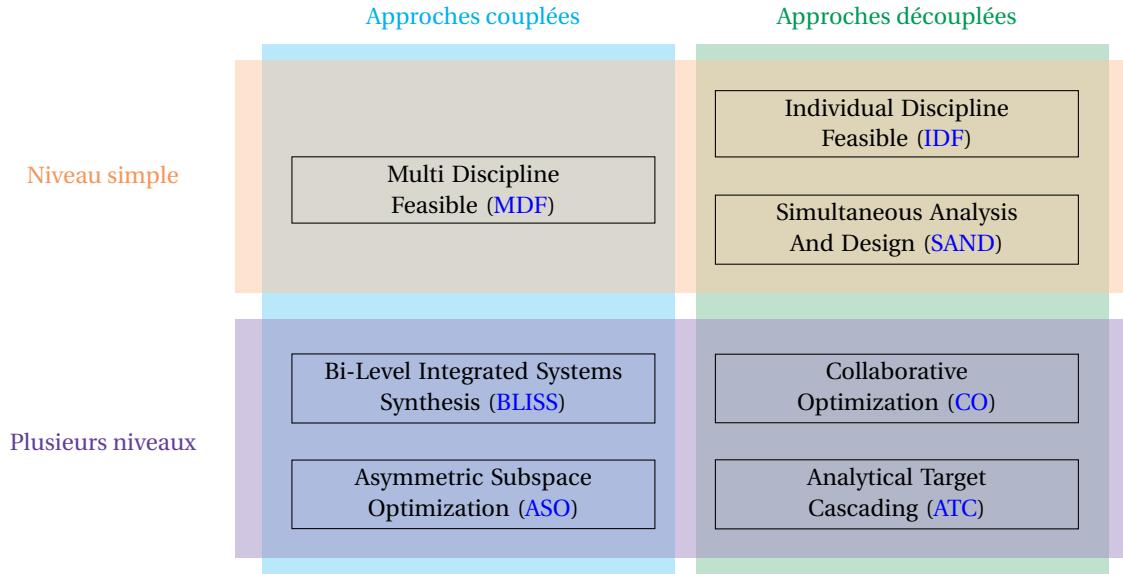


FIGURE 1.9 – Classification des principales formulations MDO

Dans cette étude nous allons nous concentrer sur trois méthodes en particulier : l'approche MDF, assez simple et qui est peu sensible au nombre de variables de couplage, la méthode IDF et la méthode EGMDO (DUBREUIL et collab. [2018b]; DUBREUIL et collab. [2020]) dont l'efficacité pour limiter le nombre d'appels aux solveurs disciplinaires a été démontrée sur un cas aéroélastique (DUBREUIL et collab. [2018b]).

1.2.4.1 Optimisation multidisciplinaire faisable

La formulation MDF initialement décrite par CRAMER et collab. [1994]; BALLING et SOBIESZCZANSKI-SOBIESKI [1996] repose sur le découplage entre le processus d'optimisation et la résolution du problème couplé. Cette formulation est surtout utilisée lorsque les solveurs de la MDA sont des boîtes noires ou lorsque la MDA est résolue par un solveur monolithique. En effet, cette méthode limite l'intrusion au sein des interactions interdisciplinaires puisque la MDA est résolue par un solveur externe. Cela revient à résoudre un problème d'optimisation similaire au système (1.1) où les variables de couplage sont uniquement gérées par le solveur externe; l'algorithme d'optimisation prend en compte uniquement les paramètres de design :

$$\operatorname{argmin}_x f(\mathbf{x}) \quad (1.40)$$

Ainsi, à chaque fois que l'algorithme d'optimisation nécessitera d'obtenir la fonction objectif en un point $\mathbf{x} = \{\mathbf{x}^g, \mathbf{x}^1, \dots, \mathbf{x}^{n_d}\} \in \mathcal{X}$, la solution de la MDA $\{\mathbf{y}_*^1(\mathbf{x}), \dots, \mathbf{y}_*^{n_d}(\mathbf{x})\}$ est calculée via le solveur externe puis la fonction objectif est calculée : $f(\mathbf{x}) = f^{obj}(\mathbf{x}, \mathbf{y}_*^1(\mathbf{x}), \dots, \mathbf{y}_*^{n_d}(\mathbf{x}))$. Cette formulation a l'avantage d'être facile à implémenter car l'algorithme d'optimisation peut être choisi indépendamment du solveur externe. De plus, la solution finale sera forcément compatible avec la MDA. Par contre, cette formulation requiert, à chaque itération de l'optimisation, de calculer la solution de la MDA ce qui peut être très coûteux. Ainsi, cette formulation est souvent la plus coûteuse en terme d'appels aux solveurs disciplinaires mais reste une des plus robustes pour trouver le minimum de la fonction objectif (HULME et BLOEBAUM [2000]). La solution de la MDF définie dans l'Eq. (1.40) peut être obtenue par plusieurs méthodes d'optimisation comme par exemple les algorithmes par gradient. Cependant, ces algorithmes nécessitent de calculer le gradient par différences finies requérant une estimation précise de la fonction objectif f et donc de la MDA.

Or, lorsqu'une méthode de Gauss-Seidel ou de Jacobi est utilisée pour la résolution de la **MDA**, une précision importante peut nécessiter un nombre très important d'itérations et donc être très coûteuse. Une autre solution peut être d'utiliser des algorithmes d'optimisation par méta-modèles (**SHI et collab. [2018]**; **XU et collab. [2018]**) comme **EGO**. **EGO** est une solution intéressante car le bruit sur la fonction objectif peut être pris en compte en utilisant un **GP** bruité (**FORRESTER et collab. [2008]**). La stratégie **MDF** est illustrée sur la Figure 1.10 dans le cas d'un système avec deux disciplines.

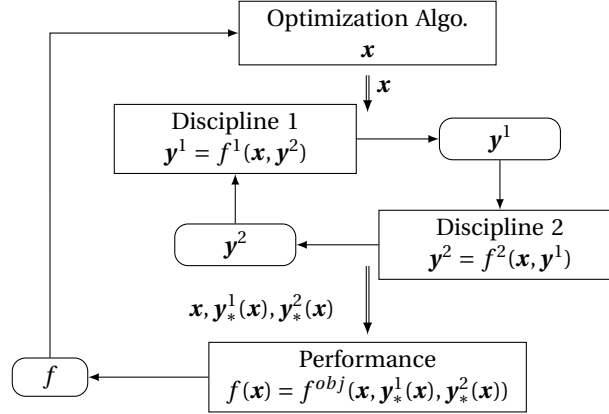


FIGURE 1.10 – Illustration de la formulation **MDF** sur un système bi-disciplinaire.

Notons que plusieurs formulations multi-niveaux de cette formulation ont été proposées. Notamment la formulation **BLISS** basée sur la même stratégie que les méthodes **TR** introduites dans la Section 1.1.2.2. Des approximations disciplinaires linéaires sont calculées en utilisant le gradient de la fonction objectif pour trouver une direction dans l'espace paramétrique. Un pas est alors proposé dans une certaine région de confiance.

Cette formulation permet de résoudre des problèmes de **MDO** en étant peu intrusive, seulement, il a été mis en évidence par **HULME et BLOEBAUM [2000]** que celle-ci reste coûteuse en terme de temps de calcul. D'autres formulations ont été proposées pour résoudre la **MDO** en laissant l'algorithme d'optimisation trouver la solution de la **MDA** afin de réduire le nombre d'appels aux solveurs disciplinaires, en particulier les méthodes **IDF** introduites ci-dessous.

1.2.4.2 Individual Discipline Feasible

L'**IDF** est une formulation découplée, à un seul niveau. La formulation ajoute donc des variables au problème d'optimisation correspondants aux variables de couplage de la **MDA**. Le problème d'optimisation est alors le suivant :

$$x_* \in \arg \min_{x, y^1, \dots, y^{n_d}} f^{obj}(x, y^1, \dots, y^{n_d}) \quad t.q. \quad r^i = y^i - f^i(x, y^{c(i)}) = 0, \quad i = 1, \dots, n_d \quad (1.41)$$

où le paramètre de design est $x \in \mathcal{X}$ et les variables de couplages $y^i \in \mathcal{C}^i$, $i = 1, \dots, n_d$ sont des variables du problème d'optimisation. Cette formulation est illustrée sur la Figure 1.11 sur un système bi-disciplinaire.

Cette formulation permet, grâce à l'approche découplée, de réduire le nombre d'appels aux solveurs disciplinaires. En effet, idéalement, cette formulation permet d'utiliser l'information d'une **MDA** non convergée afin d'orienter la recherche du paramètre de design. Cela permet de converger vers la solution de la **MDO** sans avoir à résoudre la **MDA** à chaque itération. La solution convergée obtenue vérifie cependant les contraintes d'égalité dans la limite des tolérances imposées à l'algorithme d'optimisation. De plus, cette

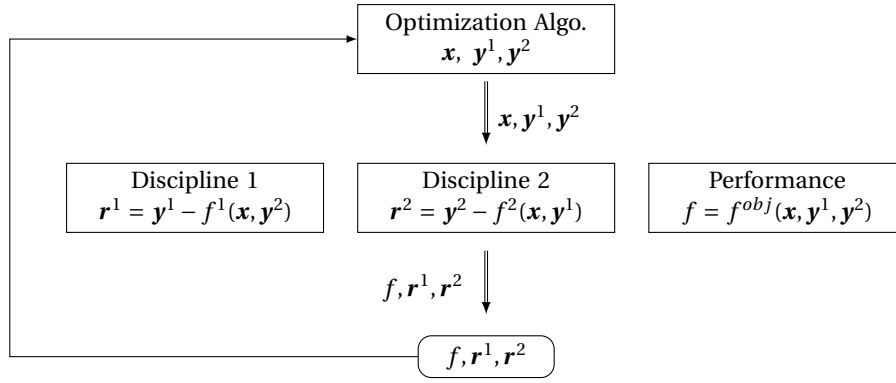


FIGURE 1.11 – Illustration de la formulation IDF sur un système bi-disciplinaire.

méthode a un inconvénient majeur : l'augmentation du nombre de variables d'optimisation peut rendre l'optimisation délicate. En effet, les méthodes par méta-modèles globaux sont difficiles à mettre en oeuvre dans ce cadre car elles sont limitées à un faible nombre de variables. Les méthodes par gradient, quant à elles, sont coûteuses lorsque le gradient n'est pas fourni par les solveurs disciplinaires.

Notons que la formulation CO (BRAUN [1996] ; BRAUN et collab. [1996] ; ALLISON et collab. [2005]), basée sur l'IDF, est décomposée en deux niveaux. Une sous-optimisation sur chaque discipline permet de minimiser le résidu disciplinaire r^i , $i = 1, \dots, n_d$ en agissant sur les paramètres de design x afin de s'assurer de la convergence de la MDA. L'optimiseur principal cherchera alors à minimiser la fonction objectif tout en annulant les résidus optimisés obtenus par les sous-optimisations, en agissant sur le paramètre de design x et sur les variables de couplage $y^1, \dots, y^{n_d}$. Notons qu'une pénalisation ainsi qu'une région de confiance sont imposées pour éviter que les paramètres générés par les sous-problèmes ne soient trop éloignés du paramètre défini par l'optimiseur principal. Finalement, nous avons observé dans cette formulation une façon de réduire le nombre d'appels aux solveurs disciplinaires. Plutôt que de limiter le nombre d'appels aux solveurs disciplinaires, il est proposé de remplacer les solveurs disciplinaires par des méta-modèles dans l'approche suivante.

1.2.5 Efficient Global Multidisciplinary Optimization

L'idée de la méthode EGMDO est basée sur l'approximation des solveurs disciplinaires par des GP permettant de remplacer les solveurs disciplinaires, coûteux, par des méta-modèles afin de réduire le coût de l'optimisation. Des méthodes, basées sur l'approximation des solveurs disciplinaires ont déjà été proposées. Une étude de PAIVA et collab. [2010] compare l'utilisation des interpolations quadratiques, des GP et des ANN pour remplacer les solveurs disciplinaires. Cette étude permet de valider l'utilisation des GP dans un cadre multidisciplinaire. D'autres études proposent des approximations par méta-modèles des solveurs disciplinaires (ZHANG et collab. [2017] ; WANG et collab. [2017] SCHOLTEN et HARTL [2021]). Dans ces études, seule l'approximation par méta-modèle est utilisée pour résoudre le problème d'optimisation. Il est alors nécessaire de créer un modèle de chaque discipline qui soit suffisamment précis sur l'ensemble du domaine paramétrique comme cela a été évoqué dans la Section 1.1.3.3. Or, l'obtention de méta-modèles précis est très coûteuse surtout dans un cadre multidisciplinaire où les variables de couplage vivent dans des espaces $\mathcal{C}^i$ non connus. Découpler les solveurs disciplinaires afin de rendre l'apprentissage des disciplines plus efficace est une des problématiques exposées par SCHOLTEN et HARTL [2021].

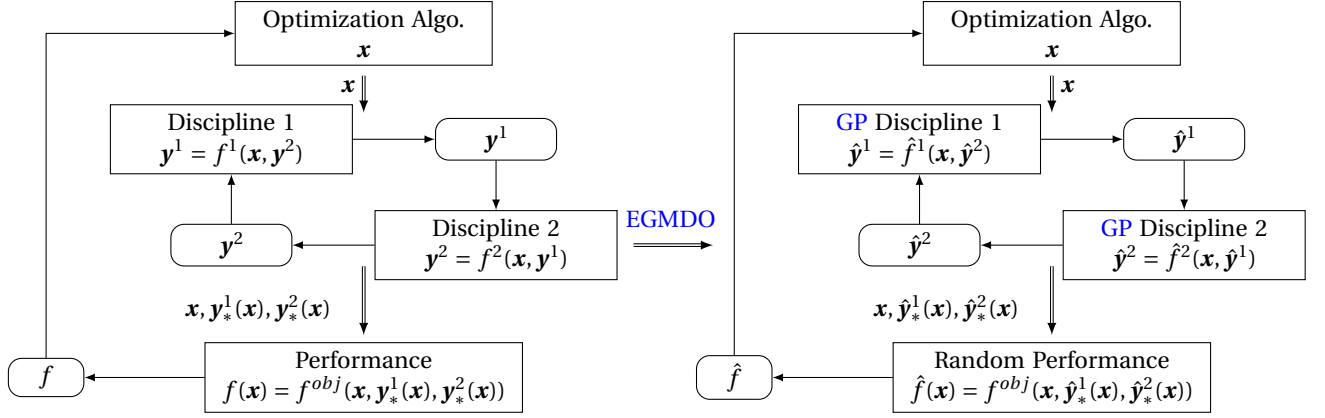


FIGURE 1.12 – Illustration de la méthode **EGMDO** où la **MDA** déterministe est remplacée par une **MDA** aléatoire en substituant les deux solveurs disciplinaires coûteux par des approximations par **GP**.

L'objectif de **EGMDO** est d'utiliser l'incertitude propre à chaque **GP** disciplinaire, donnée par la variance des **GP**, et de propager ensuite cette incertitude à la fonction objectif. Pour cela, une **MDA** aléatoire sera introduite où chaque discipline est rendue aléatoire puisqu'elle est remplacée par un **GP**. Dans cette méthode, la fonction objectif sera alors aléatoire comme dans l'algorithme **EGO**. Estimer la distribution de la fonction objectif aléatoire est un des points clefs de cette méthode car, contrairement au cadre de l'algorithme **EGO**, la fonction objectif aléatoire ne suit pas une loi gaussienne. À partir de la distribution de la fonction objectif, une fonction d'acquisition est construite et utilisée comme critère d'enrichissement pour l'algorithme d'optimisation. Dans une deuxième étape, une réduction des incertitudes liées aux méta-modèles est proposée afin de minimiser le nombre d'appels aux solveurs disciplinaires. De cette façon, les méta-modèles disciplinaires seront enrichis au fur et à mesure des itérations dans les zones intéressantes pour la résolution de l'optimisation. Pour cela, de façon similaire à ce qui a été présenté pour l'algorithme **EGO** (Section 1.1.3.3.3), l'objectif est de trouver un compromis entre exploration du domaine de conception et exploitation des méta-modèles via la fonction d'acquisition. Pour les détails de cette méthode, nous allons nous restreindre à des variables de couplage scalaires afin de ne pas alourdir les notations. Cependant, il est tout à fait possible d'adapter cette méthode avec des variables de couplage vectorielles comme cela a été montré dans **DUBREUIL et collab. [2020]**. Bien qu'il soit possible d'appliquer cette méthode à des variables de couplage scalaires et vectorielles, cette méthode est limitée à un nombre de variables de couplage faible (inférieur à 20 en pratique) : adapter cette méthode à des variables de couplage en grande dimension (définies par des champs discrétisés) est l'un des objectifs de la thèse.

1.2.5.1 Entraînement des solveurs disciplinaires

La première étape de **EGMDO** est l'entraînement des **GP** permettant d'approcher les solveurs disciplinaires. Pour cela, chaque solveur disciplinaire

$$f^i: \begin{matrix} \mathcal{X}^g \times & \mathcal{X}^i \times & \prod_{j \in c^{(i)}} \mathcal{C}^j & \rightarrow \mathcal{C}^i \\ \mathbf{x}^g, & \mathbf{x}^i, & \mathbf{y}^{c^{(i)}} & \mapsto \mathbf{y}^i \end{matrix} \quad (1.42)$$

est remplacé par un GP, entraîné sur un DoE comme cela est proposé dans la Section 1.1.3.1. Ainsi, un échantillon sur $\mathcal{X}^g \times \mathcal{X}^i \times \prod_{j \in c^{(i)}} \mathcal{C}^j$ est généré et la solution des solveurs disciplinaires pour chacun des points du DoE devra être calculée. Échantillonner selon les paramètres globaux et locaux du système est une tâche souvent aisée car les espaces paramétriques $\mathcal{X}^g$ et $\mathcal{X}^i$ sont généralement des hypercubes de $\mathbb{R}^{q^g}$ et de $\mathbb{R}^{q^i}$ dont les bornes sont connues. Ainsi, une méthode d'échantillonnage classique telle que LHS (MCKAY et col-lab. [1979]) peut être utilisée. Pour les variables de couplage, la tâche est plus complexe car aucune information n'est disponible sur les espaces de couplage $\mathcal{C}^i$ dans le cadre non-intrusif dans lequel nous travaillons. Dans ce cas une entorse à ce cadre est faite : un avis d'expert est nécessaire afin d'estimer les bornes λ_-^i et λ_+^i des espaces paramétriques telles que $\mathcal{C}^i \subset [\lambda_-^i; \lambda_+^i]$. Notons que cette hypothèse est commune à la formulation IDF. Dans ce cas spécifique, toutes les entrées du solveur disciplinaire sont considérées comme appartenant à un hypercube de dimension $q^g + q^i + \sum_{j \in c^{(i)}} d^j$ dont des échantillons peuvent être obtenus facilement. Grâce au DoE, un GP $\hat{f}^i$ de chaque solveur disciplinaire est construit :

$$f^i(\mathbf{x}^g, \mathbf{x}^i, y^{c^{(i)}}) \text{ remplacé par } \hat{f}^i(\mathbf{x}^g, \mathbf{x}^i, y^{c^{(i)}}) = \mu^i(\mathbf{x}^g, \mathbf{x}^i, y^{c^{(i)}}) + \epsilon^i(\mathbf{x}^g, \mathbf{x}^i, y^{c^{(i)}}) \quad (1.43)$$

où μ^i est la moyenne du GP et ϵ^i est un GP de moyenne nulle dont la covariance est celle de $\hat{f}^i$ notée k^i . Après entraînement, les GP obtenus sont illustrés sur le cas analytique au point $x = -3$ sur la Figure 1.13.

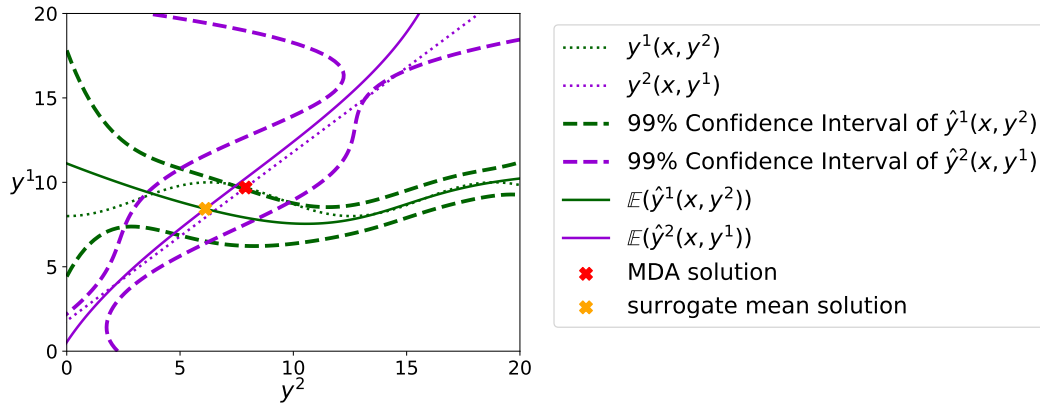


FIGURE 1.13 – Illustration des GP obtenus après entraînement sur le cas analytique présenté en Section 1.2.2.1. La moyenne et l'intervalle de confiance sont représentés ainsi que les disciplines exactes. La solution de la MDA est représentée en rouge, son approximation obtenue à partir de la moyenne des GP est représentée en orange.

Dans ce cas, la résolution de la MDA n'est plus déterministe car une composante aléatoire est introduite : ϵ^i caractérisée par sa covariance k^i . La solution du problème couplé sera donc elle aussi aléatoire. Par la suite, cette information est utilisée pour estimer l'incertitude de la solution de la MDA obtenue par les GP.

1.2.5.2 Résolution de la MDA aléatoire

Le remplacement des disciplines, déterministes, par des GP, aléatoires, amène à résoudre une MDA dite aléatoire. Comme la sortie de la MDA aléatoire n'est pas gaussienne, son analyse sera basée sur des échantillons. Nous allons donc détailler comment générer

des échantillons de cette MDA aléatoire. Notons que dans le cadre de cette méthode les disciplines sont remplacées par des GP peu coûteux. Des méthodes ont été développées notamment par BREVAULT et collab. [2015] pour gérer des disciplines aléatoires et coûteuses en approchant les disciplines par *Expansion par chaos polynomial, ou Polynomial Chaos Expansion (PCE)*. Ici, une analyse par MC est possible car les GP disciplinaires sont suffisamment peu coûteux et la MDA aléatoire peut être résolue sans effectuer d'approximations.

Pour générer une solution aléatoire pour un paramètre $\mathbf{x}$ des trajectoires t^i sont générées pour chaque GP en $\mathbf{x} = \{\mathbf{x}^g, \mathbf{x}^1, \dots, \mathbf{x}^{n_d}\}$:

$$t^i(\mathbf{x}^g, \mathbf{x}^i, y^{c[i]}) \text{ trajectoire de } \hat{f}^i(\mathbf{x}^g, \mathbf{x}^i, y^{c[i]}) \text{ en } \mathbf{x} \quad (1.44)$$

L'idée étant alors de déterminer la solution de la MDA aléatoire suivante :

$$\{\hat{y}^i = t^i(\mathbf{x}^g, \mathbf{x}^i, y^{c[i]}), \quad i = 1, \dots, n_d \quad (1.45)$$

afin d'obtenir la solution aléatoire $\hat{y}_*^1(\mathbf{x}, t^1, \dots, t^{n_d}), \dots, \hat{y}_*^{n_d}(\mathbf{x}, t^1, \dots, t^{n_d})$. Notons qu'ici, la solution dépend des trajectoires générées. Seulement, comme le couplage entre les disciplines est non-linéaire, la solution aléatoire $\hat{y}_*^1(\mathbf{x}, t^1, \dots, t^{n_d}), \dots, \hat{y}_*^{n_d}(\mathbf{x}, t^1, \dots, t^{n_d})$ n'est pas gaussienne : estimer la distribution de cette loi n'est pas trivial. Dans ce cas, il est proposé de caractériser la distribution de la solution par *Échantillonnage par Monte Carlo, ou Monte Carlo Sampling (MCS)* en générant des échantillons de solutions des MDA aléatoires obtenues à partir de différentes trajectoires pour les GP disciplinaires. La trajectoire d'un GP conditionné peut être obtenue à partir d'un GP non conditionné puis en conditionnant les trajectoires (WILLIAMS et RASMUSSEN [2006]; VILLEMONTAIX et collab. [2009]; LE GRATIET et collab. [2014]). Cependant, les trajectoires des GP sont difficiles à estimer quand le nombre d'entrées des GP disciplinaires est important. De plus, les trajectoires complètes sont caractérisées par un nombre infini de variables aléatoires ce qui n'est pas adapté pour la suite. Notons qu'en pratique, la trajectoire n'est simulée que sur un nombre fini, mais généralement très grand de points.

Ainsi, DUBREUIL et collab. [2020] proposent une modélisation particulière en utilisant une trajectoire de GP parfaitement corrélé. L'idée est de modéliser les solveurs disciplinaires par des GP conditionnés avec une fonction de corrélation constante et égale à 1. Les trajectoires des GP parfaitement corrélés sont données par :

$$t^i(y^{c[i]}) = \mu^i(\mathbf{x}^g, \mathbf{x}^i, y^{c[i]}) + \xi^i \sigma^i(\mathbf{x}^g, \mathbf{x}^i, y^{c[i]}) \quad (1.46)$$

où ξ^i est une loi normale centrée réduite et μ^i, σ^i sont la moyenne et l'écart-type conditionnés du $i^{\text{ème}}$ GP disciplinaire dont le calcul est détaillé dans la Section 1.1.3.2. Cette modélisation est comparée dans DUBREUIL et collab. [2016] à l'utilisation de trajectoires classiques de GP : la différence entre les solutions obtenues n'est pas significative. Avec cette modélisation, plusieurs solutions aléatoires peuvent être obtenues en tirant plusieurs réalisations des variables aléatoires normales ξ^i . Pour obtenir un échantillon de MDA aléatoires, on génère un échantillon aléatoire de taille n_{MC} : $(\xi_k^i)_{k=1, \dots, n_{MC}}$. Soit $\mathbf{x} = \{\mathbf{x}^g, \mathbf{x}^1, \dots, \mathbf{x}^{n_d}\}$, pour chaque tirage $\Xi_k = \xi_k^1, \dots, \xi_k^{n_d}$ il est nécessaire de résoudre le système couplé suivant :

$$\{\hat{y}_k^i = \mu^i(\mathbf{x}^g, \mathbf{x}^i, \hat{y}_k^{c[i]}) + \xi_k^i \sigma^i(\mathbf{x}^g, \mathbf{x}^i, \hat{y}_k^{c[i]}), \quad i = 1, \dots, n_d, \quad k = 1, \dots, n_{MC} \quad (1.47)$$

dont la solution est notée $\hat{y}_{*,k}^1(\mathbf{x}, \Xi_k), \dots, \hat{y}_{*,k}^{n_d}(\mathbf{x}, \Xi_k)$, on notera ici que la solution d'un tel système dépend du paramètre $\mathbf{x}$ et du tirage $\Xi_k = \xi_k^1, \dots, \xi_k^{n_d}$. Cette idée est illustrée dans la Figure 1.14 pour le cas test analytique. Des échantillons de MDA aléatoires sont générés en utilisant des trajectoires des GP.

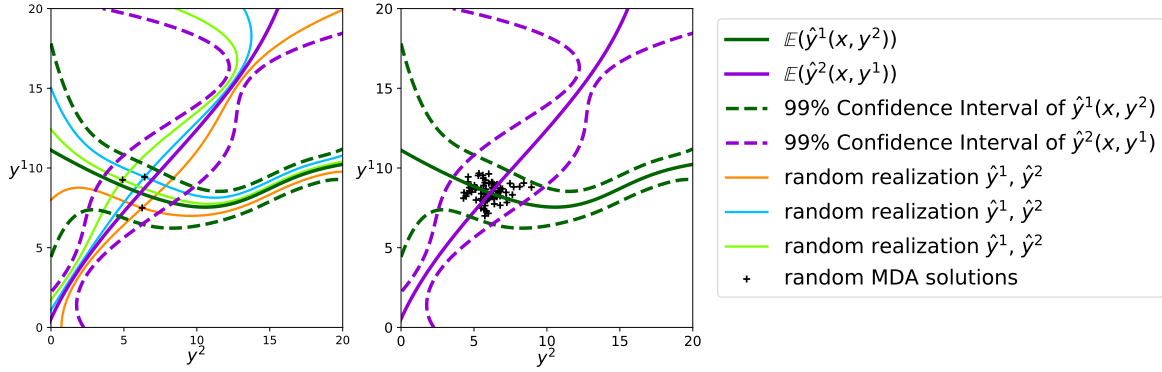


FIGURE 1.14 – Illustration de la génération de MDA aléatoires sur un cas avec uniquement deux disciplines. Les traits pleins violets et verts représentent la moyenne des GP, l'espace entre les traits pointillés est l'intervalle de confiance à 99%. Les traits fins vert clair, cyan et orange sont des trajectoires des GP parfaitement corrélés comme décrit dans l'Eq. (1.47). Les croix noires aux intersections des trajectoires sont des solutions de la MDA aléatoire.

1.2.5.3 Fonction objectif aléatoire

Nous allons à présent essayer de calculer la distribution de la fonction objectif aléatoire à partir des échantillons de MDA aléatoires. Pour cela, nous allons définir la fonction objectif aléatoire via la solution de la MDA aléatoire :

$$\hat{f}^{obj}(\mathbf{x}, \Xi) = f^{obj}(\mathbf{x}, \hat{y}_*^1(\mathbf{x}, \Xi), \dots, \hat{y}_*^{n_d}(\mathbf{x}, \Xi)) \quad (1.48)$$

La fonction objectif aléatoire ainsi définie est illustrée sur la Figure 1.15. Notons que la fonction objectif aléatoire obtenue à partir des solutions des MDA aléatoires dans le cadre de la modélisation proposée par DUBREUIL et collab. [2020] est une variable aléatoire dépendante de la variable aléatoire $\Xi = \xi^1, \dots, \xi^{n_d}$ obtenue à partir des Eq. (1.47) et Eq. (1.48). Nous définirons, dans la suite, une amélioration espérée à l'aide de cette fonction objectif. Notons que plusieurs problèmes sont à étudier dans ce cas :

- La loi des $\hat{y}_*^i(\mathbf{x}, \Xi)$ n'est pas normale même si la loi de Ξ est composée de lois normales. Ceci est dû au couplage non linéaire entre les disciplines. A fortiori, la loi de $\hat{f}^{obj}(\mathbf{x}, \Xi)$ ne sera pas normale. Ainsi, il n'est pas possible de définir une amélioration espérée analytique comme celle définie pour l'algorithme EGO (cf Eq. (1.29)).
- Même si obtenir un échantillon de $\hat{f}^{obj}(\mathbf{x}, \Xi)$ ne fait appel qu'aux moyennes et variances des GP qui sont analytiques l'optimisation de l'amélioration espérée reste trop coûteuse si elle est évaluée en tout point $\mathbf{x}$ par une simulation MC. Par conséquent, une approximation de cette variable aléatoire est proposée.

Nous avons, dans cette partie, introduit la fonction objectif aléatoire définie par la modélisation proposée par DUBREUIL et collab. [2020]. Comme la définition d'une fonction d'acquisition sur cette fonction objectif est rendue difficile par le coût numérique nécessaire à la résolution des MDA aléatoires, il est proposé de construire une approximation de la fonction objectif aléatoire.

1.2.5.4 Approximation de la fonction objectif aléatoire

L'objectif ici est de définir une approximation de la variable aléatoire suivante :

$$\begin{aligned} \hat{f}^{obj} : \mathcal{X} \times \mathbb{R}^{n_d} &\rightarrow \mathbb{R} \\ \mathbf{x} \quad , \quad \Xi &\mapsto f^{obj}(\mathbf{x}, \hat{y}_*^1(\mathbf{x}, \Xi), \dots, \hat{y}_*^{n_d}(\mathbf{x}, \Xi)) \end{aligned} \quad (1.49)$$

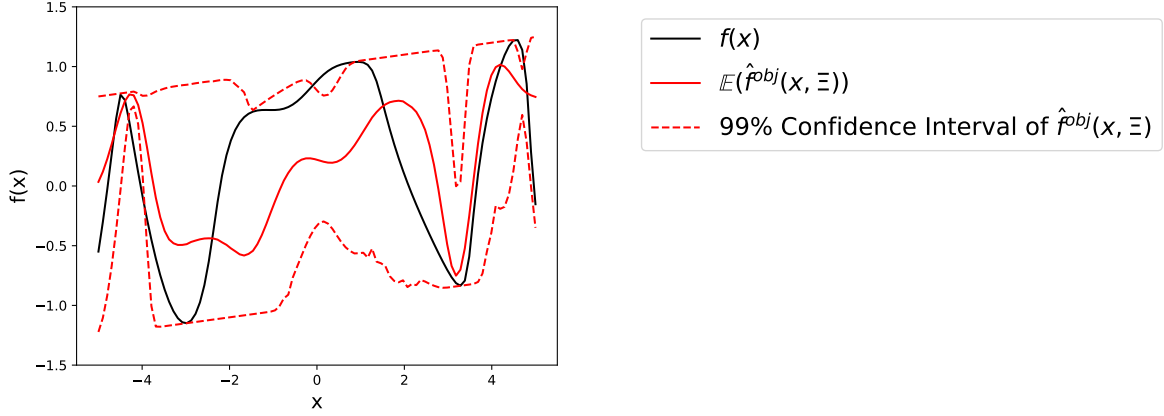


FIGURE 1.15 – Illustration de la fonction objectif aléatoire $\hat{f}^{obj}$ obtenue à partir des MDA aléatoires sur le cas test analytique. L'espérance et l'intervalle de confiance de $\hat{f}^{obj}$ sont obtenus en tout point de l'espace de design $\mathcal{X}$ à partir d'un échantillon de solutions de la MDA aléatoire de taille 2000.

où Ξ est un vecteur aléatoire de loi normale centrée réduite non corrélée de dimension n_d . Cette approximation se fera en deux temps. En premier lieu, une approximation de $\hat{f}$ est obtenue sur un DoE de $\mathcal{X}$ noté DoE_{UQ}. Pour tout $\mathbf{x} \in \text{DoE}_{\text{UQ}}$, la fonction aléatoire $\hat{f}$ est approchée par PCE :

$$\begin{aligned} \hat{f}_{\mathbf{x}}^{\text{PCE}} : \mathbb{R}^{n_d} &\rightarrow \mathbb{R} \\ \Xi &\mapsto \hat{f}_{\mathbf{x}}^{\text{PCE}}(\mathbf{x}, \hat{y}_*^1(\mathbf{x}, \Xi), \dots, \hat{y}_*^{n_d}(\mathbf{x}, \Xi)) \end{aligned} \quad , \quad \forall \mathbf{x} \in \text{DoE}_{\text{UQ}} \quad (1.50)$$

Le calcul de la PCE est détaillé dans l'Annexe A.1. Le DoE nécessaire à l'approximation de la fonction objectif est illustré dans la Figure 1.16 dans le cas analytique où plusieurs échantillons de $\hat{f}^{obj}$ sont générés sur un échantillon de l'espace de design $\mathcal{X}$ noté DoE_{UQ}. Dans cette figure, une estimation de la Densité de probabilité, ou Probability Density Function (PDF) de $\hat{f}^{obj}$ pour chaque $\mathbf{x} \in \text{DoE}_{\text{UQ}}$ est calculée et illustrée. Notons que le caractère non gaussien de la fonction objectif aléatoire est remarquable. L'approximation de $\hat{f}^{obj}$, pour chaque $\mathbf{x} \in \text{DoE}_{\text{UQ}}$, par PCE est illustrée sur la Figure 1.17. Cette figure permet alors de comparer les différences entre la fonction aléatoire $\hat{f}^{obj}$ obtenue à partir des solutions de la MDA aléatoire illustrée sur la Figure 1.16 et son approximation par PCE $\hat{f}_{\mathbf{x}}^{\text{PCE}}$ illustrée sur la Figure 1.17.

Dans un second temps, l'approximation sera étendue à tout point $\mathbf{x} \in \mathcal{X}$ en combinant une expansion de Karhunen-Loève (KL) (cf Annexe A.2) et une interpolation par GP (cf Annexe A.3) (ARNST et collab. [2012]). Notons qu'une solution admissible serait d'interpoler les coefficients des PCE sur le domaine $\mathcal{X}$ par GP. Néanmoins, il a été montré dans DUBREUIL et collab. [2018a] que l'expansion de KL permet d'améliorer la précision de l'approximation obtenue tout en diminuant le nombre de quantités à interpoler par GP. Ceci permet alors de définir une fonction $\tilde{f}$:

$$\hat{f}^{obj}(\mathbf{x}, \Xi) \approx \tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta}) \quad (1.51)$$

où la variable $\boldsymbol{\eta}$ représente l'incertitude liée à l'approximation par GP prenant en compte la variance des GP.

Cette approximation est illustrée sur la Figure 1.18. Nous pouvons observer des différences entre la fonction $\hat{f}^{obj}(\mathbf{x}, \Xi)$ illustrée sur la Figure 1.15 et $\tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta})$. En effet, les moyennes sont assez similaires bien que les zones autour de $x \approx 0$ et de $x \approx 2$ soient mal représentées puisque aucun point de DoE_{UQ} n'est présent dans ces zones. De plus,

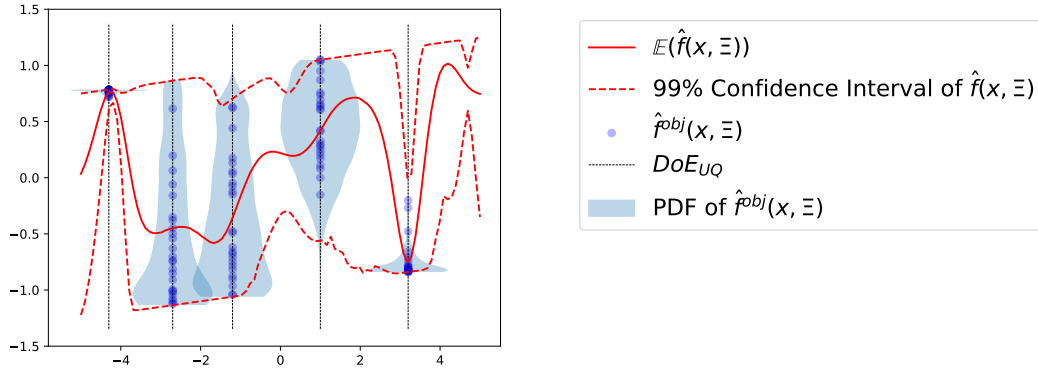


FIGURE 1.16 – Illustration du DoE de la fonction $\hat{f}^{obj}$ utilisé pour l'approximation de la fonction objectif aléatoire. L'espérance et l'intervalle de confiance de $\hat{f}^{obj}$ sont similaires à ceux de la Figure 1.15. 25 évaluations de $\hat{f}^{obj}$ sont représentées par des sphères bleues pour chaque point de DoE_{UQ} . L'estimation de la PDF de la fonction objectif aléatoire est obtenue à partir d'un échantillon de taille 2000.

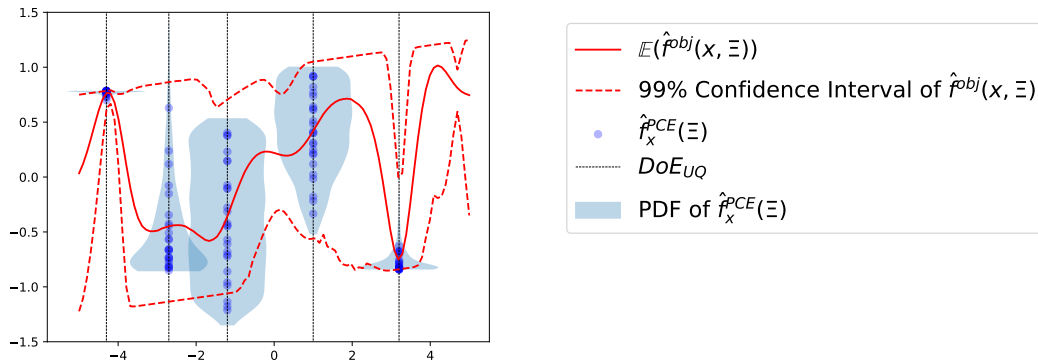


FIGURE 1.17 – Illustration de l'approximation f_x^{PCE} de $\hat{f}^{obj}$ par PCE entraînée sur les 5 points de DoE_{UQ} illustrés dans la Figure 1.16. 25 évaluations de f_x^{PCE} sont représentées par des sphères bleues pour chaque point de DoE_{UQ} et l'estimation de la PDF est obtenue à partir d'un échantillon de taille 1000. L'espérance et l'intervalle de confiance de $\hat{f}^{obj}$ sont similaires à ceux de la Figure 1.15.

l'intervalle de confiance à 99% de $\tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta})$ est plus étendu puisque les incertitudes liées à l'interpolation de l'expansion **KL** sont prises en compte. Ainsi, $\hat{f}^{obj}(\mathbf{x}, \Xi)$ est comprise entre -1 et 1 avec un niveau de confiance à 99% alors que $\tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta})$ est comprise entre -2 et 2 avec le même niveau de confiance. Pour tout $\mathbf{x} \in \text{DoE}_{\text{UQ}}$, l'incertitude d'interpolation des coefficients de la **PCE** réduits est nulle. Les différences sur les intervalles de confiance en ces points sont donc dues aux deux étapes d'approximation préalables : la **PCE** et la réduction de la dimension des coefficients par expansion de **KL**.

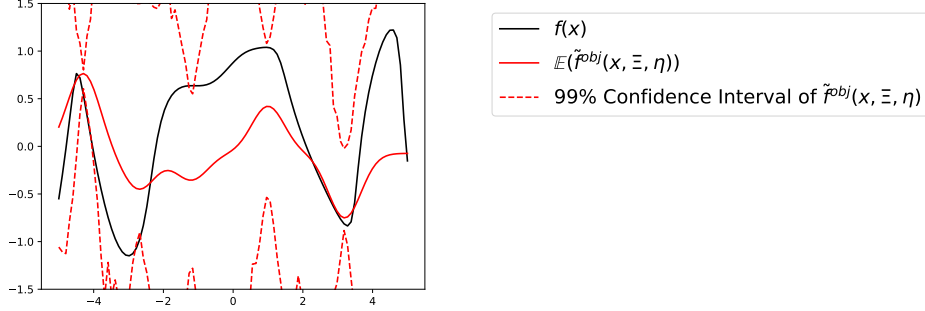


FIGURE 1.18 – Illustration de l'approximation $\tilde{f}^{obj}$ sur le cas analytique où la moyenne ainsi que l'intervalle de confiance à 99% sont apparents. $\tilde{f}^{obj}$ est entraînée sur le **DoE** illustré sur la Figure 1.16. La moyenne et l'intervalle de confiance sont obtenus à partir d'un échantillon de taille 10^7 .

1.2.5.5 Amélioration espérée modifiée

Grâce au modèle $\tilde{f}^{obj}(\mathbf{x}, \Xi)$ de la loi de la fonction objectif et la connaissance de $\hat{f}^{obj}(\mathbf{x}, \Xi)$ sur un échantillon DoE_{UQ} , il est possible de définir une amélioration :

$$I(\mathbf{x}, \Xi, \boldsymbol{\eta}) = \left(\hat{f}_{\min}^{\text{PCE}}(\Xi) - \tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta}) \right) \times \mathbb{1}_{\tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta}) \leq \hat{f}_{\min}^{\text{PCE}}(\Xi)} \quad (1.52)$$

où $\hat{f}_{\min}^{\text{PCE}}(\Xi) = \min_{\mathbf{x} \in \text{DoE}_{\text{UQ}}} \hat{f}^{\text{PCE}}(\Xi)$. L'amélioration espérée dans **EGMDO** est alors définie par :

$$\text{EI}(\mathbf{x}) = \mathbb{E} \left[\left(\hat{f}_{\min}^{\text{PCE}}(\Xi) - \tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta}) \right) \times \mathbb{1}_{\tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta}) \leq \hat{f}_{\min}^{\text{PCE}}(\Xi)} \right] \quad (1.53)$$

où

$$\begin{aligned} \mathbb{1}_{\tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta}) \leq \hat{f}_{\min}^{\text{PCE}}(\Xi)} &= 0 \text{ si } \tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta}) > \hat{f}_{\min}^{\text{PCE}}(\Xi) \\ \mathbb{1}_{\tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta}) \leq \hat{f}_{\min}^{\text{PCE}}(\Xi)} &= 1 \text{ si } \tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta}) \leq \hat{f}_{\min}^{\text{PCE}}(\Xi) \end{aligned} \quad (1.54)$$

En pratique, un échantillon $\boldsymbol{\eta}_1, \dots, \boldsymbol{\eta}_{n_{\text{MC}}^\eta} = \mathbf{H}_{\text{MC}}$ de taille n_{MC}^η et un échantillon $\Xi_1, \dots, \Xi_{n_{\text{MC}}^\Xi} = \mathbf{X}_{\text{MC}}$ de taille n_{MC}^Ξ sont générés permettant d'estimer l'**EI**.

En chaque point $\mathbf{x} \in \mathcal{X}$ l'amélioration espérée est approchée par **Estimation par Monte Carlo**, ou *Monte Carlo Estimation* (**MCE**) :

$$\text{EI}(\mathbf{x}) \approx \frac{1}{n_{\text{MC}}^\Xi} \frac{1}{n_{\text{MC}}^\eta} \sum_{i=1}^{n_{\text{MC}}^\Xi} \sum_{j=1}^{n_{\text{MC}}^\eta} I(\mathbf{x}, \Xi_i, \boldsymbol{\eta}_j) \quad (1.55)$$

Cette amélioration espérée permet de définir le point qui sera ajouté dans DoE_{UQ} :

$$\mathbf{x}_* = \arg \max_{\mathbf{x} \in \mathcal{X}} \text{EI}(\mathbf{x}) \quad (1.56)$$

L'optimisation de l'EI est obtenue en utilisant des algorithmes d'optimisation par modèles locaux tels que Coby (POWELL [1994]) ou BFGS (BROYDEN [1970]; FLETCHER [1970]; GOLDFARB [1970]; SHANNO [1970]). Suite à l'ajout de ce point sur le DoE_{UQ}, l'approximation par PCE de la fonction objectif aléatoire est de nouveau calculée sur tous les points de DoE_{UQ}. Pour cela un nouvel échantillon MC Ξ_{MC} est généré et les MDA aléatoires correspondantes sont calculées comme cela est détaillé dans l'Eq. (1.47). L'optimisation de la fonction d'acquisition présentée ici permet de résoudre le problème d'optimisation en recherchant le paramètre de design optimal. Toutefois, il est nécessaire d'enrichir les GP disciplinaires afin de réduire les incertitudes sur la fonction objectif aléatoire $\hat{f}^{obj}$ aux points de DoE_{UQ}.

1.2.5.6 Réduction des incertitudes

Pour réduire les incertitudes liées aux différentes approximations, il est proposé d'utiliser la probabilité qu'un point $\mathbf{x}$ de DoE_{UQ} soit le point minimisant $\hat{f}^{PCE}$ sur DoE_{UQ}. Cette quantité est définie par :

$$P_{min}(\mathbf{x}) = \mathbb{P} \left(\min_{\mathbf{v} \in \text{DoE}_{UQ}} \hat{f}_{\mathbf{v}}^{PCE}(\Xi) = \hat{f}_{\mathbf{x}}^{PCE}(\Xi) \right), \quad \forall \mathbf{x} \in \text{DoE}_{UQ} \quad (1.57)$$

Comme il est facile de générer un échantillon de $\hat{f}_{\mathbf{x}}^{PCE}$ via la PCE, cette probabilité est estimée par MCE :

$$P_{min}(\mathbf{x}) \approx \frac{1}{n_{MC}^{\Xi}} \sum_{i=1}^{n_{MC}^{\Xi}} \mathbb{1}_{\min_{\mathbf{v} \in \text{DoE}_{UQ}} \hat{f}_{\mathbf{v}}^{PCE}(\Xi_i) = \hat{f}_{\mathbf{x}}^{PCE}(\Xi_i)} \quad (1.58)$$

Cette probabilité permet de définir un critère d'enrichissement des solveurs disciplinaires aux points de DoE_{UQ}. Les incertitudes sont alors réduites aux points de DoE_{UQ} vérifiant le critère suivant : $P_{min}(\mathbf{x}_{(i)}) > \frac{1}{n_{UQ}}$ où n_{UQ} est le nombre de points de DoE_{UQ}. Les solveurs disciplinaires seront enrichis au point $\mathbf{x}$ et ce jusqu'à ce que les GP soient considérés comme suffisamment précis. Les solveurs disciplinaires au point $\mathbf{x}$ sont suffisamment précis lorsque le Coefficient de Variation (CV) de $\hat{f}^{PCE}(\mathbf{x}, \Xi)$:

$$CV(\hat{f}^{PCE}(\mathbf{x}, \Xi)) = \frac{\sqrt{\text{Var}(\hat{f}_{\mathbf{x}}^{PCE}(\Xi))}}{\mathbb{E}(\hat{f}_{\mathbf{x}}^{PCE}(\Xi))} \quad (1.59)$$

est inférieur à une tolérance ϵ_{CV} définie par l'utilisateur. La réduction des incertitudes est synthétisée dans l'Algorithme 6 dont les étapes sont les suivantes :

1. Calcul de la PCE permettant d'obtenir l'approximation $\hat{f}_{\mathbf{x}}^{PCE}(\Xi)$, $\forall \mathbf{x} \in \text{DoE}_{UQ}$.
2. Cette approximation est utilisée pour calculer $P_{min}(\mathbf{x})$, $\forall \mathbf{x} \in \text{DoE}_{UQ}$ via l'Eq. (1.58) en utilisant une MCS.
3. Les points satisfaisant le critère d'enrichissement sont ajoutés dans $\mathbf{Z}_{enrich}$ dans l'ordre de probabilité P_{min} croissant.
4. Tant que le critère de convergence n'est pas atteint : les solveurs disciplinaires sont enrichis. L'enrichissement des GP au point $\mathbf{x}$ nécessite une estimation de la solution de la MDA au point $\mathbf{x}$. En effet, comme rappelé dans l'Eq. (1.42), les entrées de f^i sont $\mathbf{x}^g$, $\mathbf{x}^i$ et $\mathbf{y}^{c^i}$. L'estimation de la solution de la MDA est obtenue en résolvant la MDA en utilisant la moyenne des GP :

$$\left\{ \bar{y}_k^i = \mu^i(\mathbf{x}_*^g, \mathbf{x}_*^i, \bar{y}_k^{c^i}), \quad i = 1, \dots, n_d, \quad k = 1, \dots, n_{MC} \right. \quad (1.60)$$

5. Comme les approximations disciplinaires ont changé, la **PCE** est calculée à nouveau permettant de redéfinir $P_{min}(\mathbf{x})$, $\forall \mathbf{x} \in DoE_{UQ}$. Ainsi, les étapes [3-6] sont répétées jusqu'à ce que le coefficient de variation $CV(\hat{f}_x^{PCE}(\mathbf{x}, \Xi))$ soit inférieur à ϵ_{CV} pour tout $\mathbf{x} \in DoE_{UQ}$ tel que $P_{min}(\mathbf{x}) > \frac{1}{n_{UQ}}$.

Algorithm 6: Uncertainty Reduction of the disciplinary **GP**

```

input :  $\epsilon_{CV}$ 
Compute the PCE approximation to obtain  $\hat{f}_x^{PCE}(\Xi)$ ,  $\forall \mathbf{x} \in DoE_{UQ}$ ;
Compute  $P_{min}(\mathbf{x})$ ,  $\forall \mathbf{x} \in DoE_{UQ}$  using Eq. (1.58);
while  $\exists \mathbf{x} \in DoE_{UQ}$  such that  $P_{min}(\mathbf{x}) > \frac{1}{n_{UQ}}$  and  $CV(\hat{f}_x^{PCE}(\Xi)) > \epsilon_{CV}$  do
     $\mathbf{Z}_{enrich} \leftarrow \left\{ \mathbf{x} \mid P_{min}(\mathbf{x}) \geq \frac{1}{n_{UQ}} \right\}$ ;
    Sort  $\mathbf{Z}_{enrich}$  in decreasing order with respect to  $P_{min}(\mathbf{x})$ ;
    convergence  $\leftarrow$  False;
     $k \leftarrow 0$ ;
    while  $k < card(\mathbf{Z}_{enrich})$  and convergence==False do
        convergence  $\leftarrow$  True;
         $\mathbf{x} \leftarrow \mathbf{Z}_{enrich}[k]$ ;
        if  $CV(\hat{f}_x^{PCE}(\Xi)) > \epsilon_{CV}$  then
            Enrich the disciplinary GP on  $\mathbf{x}$ ;
            Compute the PCE approximation to obtain  $\hat{f}_x^{PCE}(\Xi)$ ,  $\forall \mathbf{x} \in DoE_{UQ}$ ;
            Compute  $P_{min}(\mathbf{x})$ ,  $\forall \mathbf{x} \in DoE_{UQ}$  using Eq. (1.58);
            Update  $\mathbf{Z}_{enrich}$ ;
            convergence  $\leftarrow$  False;
        else
             $k \leftarrow k + 1$ 
        end
    end
end

```

1.2.5.7 Résolution de la **MDO** grâce à **EGMDO**

Les itérations de l'algorithme **EGMDO** sont détaillées dans l'Algorithme 7. Notons que la sortie de l'algorithme **EGMDO** est une loi aléatoire dont les incertitudes ont été réduites suffisamment autour du minimum. Cette loi est donnée par l'approximation par expansion **KL** et interpolation par **GP** dont les incertitudes selon $\boldsymbol{\eta}$ sont quasiment éliminées dans les zones d'intérêt pour l'optimisation : $\tilde{f}^{obj}(\mathbf{x}, \Xi, 0)$. Les itérations de l'algorithme sont illustrées dans l'Annexe A.4. Les étapes de la résolution sont synthétisées sur la Figure 1.19.

1.2.5.8 Bilan sur la méthode **EGMDO**

Dans cette section, nous avons passé en revue une méthode basée sur la construction de méta-modèles disciplinaires pour résoudre des problèmes **MDO** menant à une approximation de la **MDA**. Contrairement à ce qui est proposé dans la littérature, l'incertitude liée aux **GP** disciplinaires est utilisée et propagée jusqu'à la fonction objectif afin de définir, de manière similaire à la méthode **EGO**, une fonction d'acquisition. Cette

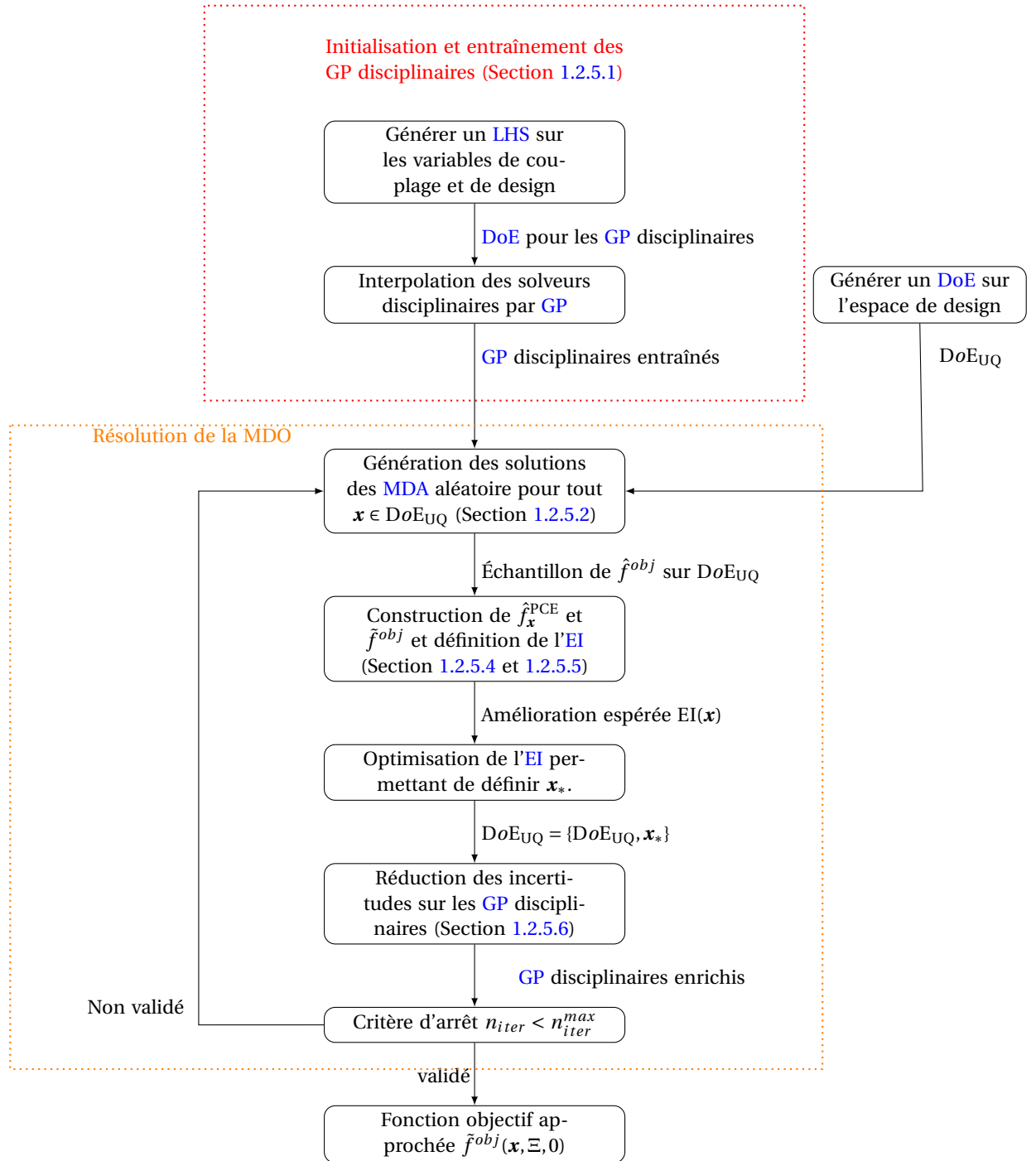


FIGURE 1.19 – Illustration de la résolution de la MDO en utilisant la méthode EGMDO.

Algorithm 7: EGMDO

input : A GP approximation $\hat{f}^i$ of the disciplinary solver f_i as proposed in Section 1.2.5.1
DoE_{UQ} $\leftarrow$ DoE over $\mathcal{X}$ obtained by LHS;
 $\Xi_{MC} \leftarrow$ MC sample from normal laws;
Compute the random MDA solution for each element of DoE_{UQ} and Ξ_{MC} ;
Compute the PCE approximation $\forall \mathbf{x} \in \text{DoE}_{UQ}$;
 $n_{iter} \leftarrow 0$ **while** $n_{iter} < n_{iter}^{max}$ **do**
 Compute the combination of KL expansion and GP interpolation to get $\tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta})$;
 Find $\mathbf{x}_* = \arg \max_{\mathbf{x} \in \mathcal{X}} \text{EI}(\mathbf{x})$;
 Add $\mathbf{x}_*$ to DoE_{UQ};
 Enrich the GP using Algorithm 6;
end
output: $\tilde{f}^{obj}(\mathbf{x}, \Xi, 0)$

fonction d'acquisition permet d'identifier le paramètre de design où les modèles disciplinaires doivent potentiellement être enrichis. Une réduction des incertitudes permet alors d'enrichir les GP disciplinaires aux points d'intérêt, c'est à dire avec une forte probabilité d'être optimaux. Cette méthode permet de fortement diminuer le nombre d'appels aux solveurs disciplinaires nécessaires à la résolution de la MDO en construisant et en exploitant des approximations sur $\mathcal{Z} \times \mathcal{C}^i$ et non pas uniquement sur $\mathcal{Z}$ ce qui est le cas des approches traditionnelles. Toutefois, cette méthode suppose que les variables de couplage sont de faible dimension car il est nécessaire d'entraîner des GP disciplinaires dont les entrées sont à la fois les paramètres de design et les variables de couplage correspondantes. Or, comme nous l'avons abordé dans la Section 1.1.3.1, dans le cas d'un grand nombre d'entrées, l'entraînement des GP est délicat. De plus, certains problèmes liés à l'incertitude des GP peuvent faire échouer la méthode. En effet, aucune certitude n'est disponible sur l'existence d'une solution pour la MDA aléatoire définie par l'Eq. (1.47). Finalement, cette méthode requiert de connaître les bornes des espaces des variables de couplage $\mathcal{C}^i$, $i = 1, \dots, n_d$. Or, cette information peut être difficile à obtenir. Dans les travaux de cette thèse, une méthode d'entraînement des GP disciplinaires ne prenant pas en compte cette information sera développée.

1.2.6 Bilan sur l'optimisation multidisciplinaire

Nous avons présenté dans cette section plusieurs méthodes permettant de résoudre le problème MDO. La méthode faisable MDF respecte notre volonté d'être le moins intrusif possible et est facilement utilisable lorsque qu'un grand nombre de variables de couplage est impliqué dans la résolution de la MDA. Malheureusement, cette méthode reste très coûteuse. Les méthodes découplées de type IDF permettent de réduire le nombre d'appels aux solveurs disciplinaires puisque l'information d'une MDA non convergée peut être utilisée. En dernier lieu, les méthodes basées sur l'enrichissement itératif d'approximations disciplinaires (DUBREUIL et collab. [2020, 2018b]) semblent très prometteuses pour réduire le nombre d'évaluations des solveurs disciplinaires. Cependant, cette méthode n'est pas adaptée à un grand nombre de variables de couplage car cela induirait un trop grand nombre de variables en entrée des GP disciplinaires. Une amélioration possible proposée dans ces travaux de thèse permet d'adapter cette méthode à un grand

nombre de variables de couplage en s'inspirant des méthodes de réduction d'ordre de modèle présentées dans la suite de ce chapitre.

1.3 Réduction d'ordre de modèle

Les objectifs de cette section sont :

- Définir le cadre considéré ici pour la [Réduction d'ordre de modèle, ou Model Order Reduction \(MOR\)](#), à savoir les problèmes linéaires paramétrés. Ces solveurs peuvent être identifiés aux solveurs disciplinaires définis dans la [MDA](#).
- Définir les approximations par projection qui ont été proposées dans la littérature.
- Trouver les méthodes qui seront les plus efficaces dans le cadre multidisciplinaire.

Dans cette partie, nous allons travailler sur des problèmes linéaires paramétrés obtenus à partir de l'intégration d'une équation aux dérivées partielles sur un maillage. Les solveurs disciplinaires, introduits dans la Section 1.2.1.1, sont solutions de ce type de problème. Or, la résolution de tels systèmes, où l'intégration est effectuée sur un maillage fin induisant un système linéaire de grande dimension, peut être coûteuse. De plus, ces solveurs sont utilisés dans un contexte *multi-query* (requêtes multiples), comme la [MDO](#), où de nombreux appels au solveur sont effectués pour différents paramètres. Une des solutions proposées par la littérature est la [MOR](#) dont l'objectif est de créer une approximation du système qui sera moins coûteuse. Dans cette étude, nous allons nous focaliser sur les méthodes de projection, permettant d'exprimer la solution d'un tel solveur sur un espace vectoriel de faible dimension. Cela suppose que les solutions d'un solveur physique paramétré peuvent être décrites par un faible nombre de modes empiriques. Ainsi, l'enjeu de cette section va être d'obtenir ces modes et de calculer l'approximation des solveurs à partir de ces modes.

1.3.1 Introduction

Dans un premier temps, nous allons présenter la résolution d'un problème par intégration des équations physiques sur un maillage. Ce problème fait intervenir un ensemble de paramètres de design $\mathbf{x}$ ayant une influence sur les caractéristiques physiques du problème. Afin de décrire un tel système, l'exemple de l'élasticité tridimensionnelle est proposé. Dans ce cadre, les paramètres peuvent influencer la géométrie des solides, les caractéristiques des matériaux telles que le coefficient de Poisson, le module d'Young... Le domaine d'étude $\Omega \in \mathbb{R}^3$ possède une frontière $\Gamma = \partial\Omega$. Le modèle élastique stationnaire est régi par le système d'équations sur le domaine Ω :

$$\left\{ \begin{array}{ll} \text{div } \boldsymbol{\sigma} + \mathbf{f}_v &= \mathbf{0} & \text{Équilibre local} \\ \boldsymbol{\sigma} &= \mathbf{C}\boldsymbol{\epsilon} & \text{loi de Hooke} \\ \boldsymbol{\epsilon} &= \frac{1}{2}(\nabla \mathbf{u} + \nabla^T \mathbf{u}) = \mathbf{L}(\mathbf{u}) & \text{relation cinématique} \end{array} \right. \quad (1.61)$$

où $\boldsymbol{\sigma}$ est le tenseur des contraintes, $\mathbf{f}_v$ est le vecteur des efforts volumiques appliqués sur le solide, $\mathbf{C}$ est le tenseur des compliances élastiques, $\boldsymbol{\epsilon}$ est le tenseur de déformation linéarisé et $\mathbf{u}$ est le déplacement du solide. Ce modèle physique est contraint par un certain nombre de conditions aux limites :

$$\begin{array}{ll} \mathbf{u} &= \mathbf{u}_D \quad \text{sur } \Gamma_D \quad \text{condition de Dirichlet} \\ \boldsymbol{\sigma} \cdot \mathbf{n} &= \boldsymbol{\sigma}_N \quad \text{sur } \Gamma_N \quad \text{condition de Neumann} \end{array} \quad (1.62)$$

avec $\mathbf{u}_D$ un déplacement imposé, $\boldsymbol{\sigma}_N$ des contraintes imposées et $\{\Gamma_D, \Gamma_N\}$ une partition de Γ . La formulation variationnelle du problème est définie pour toute fonction-test v nulle sur le bord de Dirichlet et s'exprime selon :

$$\int_{\Omega} \mathbf{L}(v) : \mathbf{CL}(\mathbf{u}) - \int_{\Omega} \mathbf{f}_v v - \int_{\Gamma_N} \boldsymbol{\sigma}_N v = 0 \quad (1.63)$$

Cette formulation variationnelle est résolue sur un espace discret via l'utilisation d'une méthode d'éléments finis. Après intégration de l'Eq. (1.63) sur les éléments du maillage, la formulation variationnelle discrète se met sous la forme d'un système matriciel :

$$\mathbf{K}(\mathbf{x})\mathbf{u}(\mathbf{x}) = \mathbf{F}(\mathbf{x}) \quad (1.64)$$

où $\mathbf{K} \in \mathbb{R}^{N \times N}$ est la matrice de rigidité du solide, $\mathbf{F} \in \mathbb{R}^N$ le vecteur des efforts extérieurs et la solution du système est le vecteur des déplacements $\mathbf{u} \in V \subset \mathbb{R}^N$ dépendant des paramètres $\mathbf{x}$ du problème. Le nombre de degrés de liberté N du problème est lié à la discrétisation de la formulation variationnelle et représente la dimension de l'espace discret dans lequel le déplacement discrétisé $\mathbf{u}$ est obtenu. De nombreux développements ont été proposés dans la littérature afin de rendre la discrétisation la plus précise possible soit en augmentant le nombre de degré de liberté N en raffinant le maillage afin d'obtenir une approximation plus fidèle de Ω , soit en augmentant le degré des fonctions de forme. Cette augmentation de N se traduit par un système matriciel linéaire Eq. (1.64) de très grande dimension dont le temps de résolution peut devenir important.

Ainsi, la MOR par projection propose de construire une base B_r caractérisée par sa matrice de projection orthonormée $\Phi \in \mathbb{R}^{N \times n_r}$ afin d'obtenir une solution dans un sous-espace affine V_r^a de dimension bien inférieure : $n_r \ll N$. L'objectif est alors d'obtenir une approximation $\mathbf{u}_r \in V_r^a$ du déplacement :

$$\mathbf{u}(\mathbf{x}) \approx \mathbf{u}_r(\mathbf{x}) = \Phi_0 + \sum_{i=1}^{n_r} \alpha_i(\mathbf{x}) \Phi_i \quad (1.65)$$

où Φ_i , $i = 1, \dots, n_r$ sont les colonnes de Φ et Φ_0 est un élément de V_r^a . Nous noterons V_r l'espace vectoriel engendré par la matrice Φ , la seule différence entre l'espace affine V_r^a et l'espace vectoriel V_r est la translation caractérisée par le vecteur Φ_0 . Ainsi, deux étapes sont nécessaires pour la réduction d'ordre de modèle par projection : le calcul du sous-espace V_r et la construction des coefficients α_i sur les vecteurs de la base de projection Φ_i .

De nombreuses autres méthodes de réduction d'ordre de modèle ont été proposées. Parmi celles-ci, certaines utilisent les modes de vibration de la structure (ROULEAU et col-lab. [2014]), d'autres exploitent la sous-structuration du problème (GUYAN [1965]). Une autre solution a été très développée : la *Décomposition propre généralisée, ou Proper Generalized Decomposition* (PGD) (LADEVÈZE [1985]; LADEVÈZE et collab. [2010], CHINESTA et collab. [2011]; LADEVÈZE [2012]; CHEVREUIL et NOUY [2012]; NARGIL et collab. [2014]; ALLIER et collab. [2015]) permet de construire a priori la base réduite en partant directement des équations d'équilibre. Cette méthode est très intrusive bien que certains travaux récents (ZOU et collab. [2018]; LEON et collab. [2019]) essayent de rendre cette méthode non-intrusive afin d'être appliquée à des solveurs industriels. Cependant, ces méthodes supposent que l'espace des entrées du solveur est borné et que les bornes sont connues. Or, nous souhaitons appliquer cette méthode aux solveurs disciplinaires dont les bornes des paramètres de la MDA $\mathbf{x}$ sont connues mais pas celles des variables de couplage. Ainsi, cette méthode ne sera pas développée dans la suite.

Pour notre étude, nous allons nous intéresser en premier lieu au calcul des coefficients. Dans un second temps, nous nous intéresserons à la génération des bases de projection par POD ou en utilisant les méthodes gloutonnes. Une extension de la MOR aux quantités d'intérêt linéaires sera étudiée.

1.3.2 Obtention des coefficients sur la base réduite

Différentes méthodes permettant d'obtenir les coefficients sur la base réduite α_i , $i = 1, \dots, n_r$ sont présentées. Nous nous focaliserons sur deux méthodes en particulier : la projection de Petrov-Galerkin qui est une méthode intrusive et l'interpolation des coefficients qui est une méthode non-intrusive.

1.3.2.1 Projection de Petrov-Galerkin

L'objectif de cette projection est de déterminer la projection orthogonale de $\mathbf{u}(\mathbf{x})$ sur l'espace V_r^a :

$$\mathbf{u}_r = \min_{\mathbf{v} \in V_r^a} \|\mathbf{u}(\mathbf{x}) - \mathbf{v}\| \quad (1.66)$$

où $\|\cdot\|$ est une norme que l'on définira dans la suite. Dans le cas particulier où la matrice $\mathbf{K}(\mathbf{x})$ est symétrique et définie positive, la solution de l'Eq. (1.66), pour la norme $\|\mathbf{x}\|_{\mathbf{K}(\mathbf{x})}^2 = \mathbf{x}^T \mathbf{K}(\mathbf{x}) \mathbf{x} = \langle \mathbf{x}, \mathbf{K}(\mathbf{x}) \mathbf{x} \rangle$ où $\langle \cdot, \cdot \rangle$ est le produit scalaire de $\mathbb{R}^N$, s'obtient en résolvant un système projeté. Ce cas particulier est appelé projection de Galerkin. La solution de l'Eq. (1.66) est obtenue par la résolution d'un système linéaire de faible dimension en projetant les données de l'Eq. (1.64), $\mathbf{K}(\mathbf{x})$ et $\mathbf{F}(\mathbf{x})$, dans V_r définissant l'opérateur réduit $\mathbf{K}_r(\mathbf{x})$ et les efforts projetés $\mathbf{F}_r(\mathbf{x})$:

$$\begin{aligned} \mathbf{K}_r(\mathbf{x}) &= \Phi^T \mathbf{K}(\mathbf{x}) \Phi \\ \mathbf{F}_r(\mathbf{x}) &= \Phi^T (\mathbf{F}(\mathbf{x}) - \mathbf{K}(\mathbf{x}) \phi_0) \end{aligned} \quad (1.67)$$

Les coefficients de la base réduite $\alpha_i(\mathbf{x})$, $i = 1, \dots, n_r$ sont solutions du système suivant :

$$\mathbf{K}_r(\mathbf{x}) \boldsymbol{\alpha}(\mathbf{x}) = \mathbf{F}_r(\mathbf{x}) \quad (1.68)$$

où $\boldsymbol{\alpha}(\mathbf{x})$ est le vecteur contenant les coefficients $\alpha_i(\mathbf{x})$, $i = 1, \dots, n_r$. Cette solution $\mathbf{u}_r(\mathbf{x}) \in V_r^a$ est caractérisée par :

$$\langle \mathbf{K}(\mathbf{x}) \mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x}), \Phi \boldsymbol{\omega} \rangle = 0, \quad \forall \boldsymbol{\omega} \in \mathbb{R}^{n_r} \quad (1.69)$$

Dans le cas où la matrice du système linéaire n'est pas symétrique (par exemple dans le cas d'un calcul *Vortex Lattice Method* (VLM) que l'on présentera dans la suite de la thèse) ou que l'on souhaite obtenir une approximation qui soit optimale selon une norme différente de $\|\mathbf{x}\|_{\mathbf{K}(\mathbf{x})}$, il convient de définir un espace vectoriel de test W_r engendré par la matrice $\Psi \in \mathbb{R}^{N \times n_r}$ définissant une nouvelle solution dans l'espace réduit caractérisée par :

$$\langle \mathbf{K}(\mathbf{x}) \mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x}), \mathbf{v} \rangle = 0, \quad \forall \mathbf{v} \in W_r \quad (1.70)$$

Le calcul pratique de cette nouvelle solution s'effectue à partir de la matrice de projection Ψ sur W_r dans ce cas, les données projetées sont définies par

$$\begin{aligned} \mathbf{K}_r(\mathbf{x}) &= \Psi^T \mathbf{K}(\mathbf{x}) \Phi \\ \mathbf{F}_r(\mathbf{x}) &= \Psi^T (\mathbf{F}(\mathbf{x}) - \mathbf{K}(\mathbf{x}) \phi_0) \end{aligned} \quad (1.71)$$

Dans la suite, nous supposerons que la matrice de rigidité est symétrique et que l'espace de test est engendré par Φ ($\Psi = \Phi$). Bien que cette solution permette d'obtenir exactement les coefficients sur la base réduite à partir des données du système linéaire, cette méthode nécessite d'assembler, pour chaque $\mathbf{x}$ pour lequel on souhaite obtenir une approximation de la solution de l'Eq. (1.64), les données $\mathbf{K}(\mathbf{x})$ et $\mathbf{F}(\mathbf{x})$. Or l'assemblage de ces données peut être coûteux. Pour y remédier, une hypothèse supplémentaire est nécessaire sur le solveur utilisé : la décomposition affine des éléments.

1.3.2.2 Décomposition affine des éléments

Dans les faits, la résolution de l'Eq. (1.64) par projection de Petrov-Galerkin permet de fortement diminuer le temps de résolution du système linéaire. En revanche, l'assemblage des données est nécessaire pour projeter les opérateurs sur la base réduite. La décomposition affine des éléments permet de projeter des quantités lors d'une phase dite *Offline* afin d'éviter d'avoir à assembler et à projeter les opérateurs lorsque l'on souhaite obtenir $\mathbf{u}_r(\mathbf{x})$ en un point $\mathbf{x}$ appelé phase *Online*.

La décomposition affine des éléments est définie par :

$$\begin{aligned}\mathbf{K}(\mathbf{x}) &= \sum_{i=1}^{n_K} \kappa_i(\mathbf{x}) \mathbf{K}_i \\ \mathbf{F}(\mathbf{x}) &= \sum_{i=1}^{n_F} f_i(\mathbf{x}) \mathbf{F}_i\end{aligned}\tag{1.72}$$

où κ_i , $i = 1, \dots, n_K$ et f_i , $i = 1, \dots, n_F$ sont des fonctions de $\mathcal{X} \mapsto \mathbb{R}$ qui représentent les variations de $\mathbf{K}(\mathbf{x})$ et $\mathbf{F}(\mathbf{x})$ selon le paramètre $\mathbf{x}$. $\mathbf{K}_i \in \mathbb{R}^{N \times N}$, $i = 1, \dots, n_K$ sont des matrices qui seront toutes projetées dans la base réduite. De la même façon, les vecteurs $\mathbf{F}_i \in \mathbb{R}^N$, $i = 1, \dots, n_F$ seront eux aussi projetés. Ainsi, lorsque le système Eq. (1.68) sera résolu, toutes les opérations seront effectuées dans l'espace réduit de dimension n_r :

$$\begin{aligned}\mathbf{K}_r(\mathbf{x}) &= \sum_{i=1}^{n_K} \kappa_i(\mathbf{x}) \underbrace{\Phi^T \mathbf{K}_i \Phi}_{\text{calculé en phase Offline}} \\ \mathbf{F}_r(\mathbf{x}) &= \sum_{i=1}^{n_F} f_i(\mathbf{x}) \underbrace{\Phi^T (\mathbf{F}_i - \mathbf{K}(\mathbf{x}) \Phi_0) \Phi}_{\text{calculé en phase Offline}}\end{aligned}\tag{1.73}$$

Notons néanmoins que cette décomposition affine des éléments n'est pas toujours disponible. Ainsi, plusieurs méthodes ont été développées pour définir une approximation de celle-ci. Les premiers termes de la série de Taylor peuvent être utilisés (VEROY et collab. [2003] ; BUI-THANH et collab. [2008]). Pour pallier l'assemblage des données, la *gappy POD* développée par EVERSON et SIROVICH [1995] pour la reconnaissance faciale permet l'approximation d'un vecteur paramétré $\mathbf{F}(\mathbf{x})$ en créant une base réduite pour cette quantité (nommée base *POD* que nous introduirons dans la suite). En revanche, les coefficients sur la base réduite ne sont pas obtenus par projection mais sont définis par la concordance entre l'approximation sur la base réduite et certaines composantes dispersées du vecteur $\mathbf{F}(\mathbf{x})$. Notons qu'il est alors nécessaire de connaître les composantes dispersées du vecteur $\mathbf{F}(\mathbf{x})$, $\mathbf{x} \in \mathcal{X}$ ce qui est délicat. Par exemple, si $\mathbf{F}(\mathbf{x})$ représente les efforts extérieurs obtenus sur le maillage, les composantes dispersées seront obtenues en assemblant les efforts sur un maillage creux. Les *Gauss Newton with Approximated Tensors* (GNAT) (CARLBERG et collab. [2013]) ont étendu cette idée pour obtenir une approximation de la matrice de rigidité et du vecteur des efforts par *gappy POD* afin de définir une décomposition affine des éléments dans un contexte de solveurs non-linéaires. Un des désavantages de cette méthode est l'estimation de l'erreur commise sur l'apprentissage par *gappy POD* des données $\mathbf{F}(\mathbf{x})$ et $\mathbf{K}(\mathbf{x})$. Enfin, la *Discrete Empirical Interpolation Method* (DEIM) développée par CHATURANTABUT et SORENSEN [2010] est utilisée afin de déterminer les éléments de $\mathbf{F}(\mathbf{x})$ et $\mathbf{K}(\mathbf{x})$ les plus intéressants pour l'interpolation selon une méthode proche de la *gappy POD* utilisée par AMSALLEM et collab. [2015] ; NEGRI et collab. [2015]. CHATURANTABUT et SORENSEN [2010] proposent aussi une évaluation de l'erreur commise qui sera affinée par WIRTZ et collab. [2012]. Cependant, cette méthode requiert de pouvoir assembler la matrice de rigidité et le vecteur des efforts sur un maillage creux : seul un faible nombre de mailles est alors considéré, mais cette possibilité n'est pas toujours accessible sur les solveurs industriels (HE et collab. [2021]). Dans le

cadre de notre étude, la décomposition affine des éléments n'est pas disponible ou peut conduire à des erreurs importantes (DAL SANTO et MANZONI [2019]). De plus, une estimation de l'erreur sur cette hyper-réduction est difficile à obtenir bien que certains travaux s'y intéressent (WIRTZ et collab. [2012]; XIAO [2019]; NURTAJ HOSSAIN et GHOSH [2020]). Or, même si une approximation de l'erreur sur la norme matricielle de $\mathbf{K}(\mathbf{x})$ est calculée, la propagation de l'erreur sur la solution du système linéaire $\mathbf{u}(\mathbf{x})$ n'est pas triviale et la solution la plus utilisée pour valider la décomposition affine des éléments par DEIM est la validation croisée (AMSALLEM et collab. [2015]). Néanmoins, obtenir une validation croisée précise requiert un DoE de grande taille et vient donc alourdir le coût numérique nécessaire à la construction du modèle MOR. Un DoE de plus faible dimension peut être utilisé mais l'erreur peut alors être sous-estimée.

1.3.2.3 Interpolation des coefficients

La *Décomposition orthogonale aux valeurs propres et interpolation*, ou *Proper Orthogonal Decomposition and Interpolation* (POD+I) (COELHO et collab. [2009]; FARZAM FAR et collab. [2018]; HESTHAVEN et UBBIALI [2018]; RIPEPI et collab. [2018]) cherche à interpoler les coefficients de la base réduite par des méta-modèles comme ceux présentés dans la Section 1.1.3.1. L'objectif est de construire un méta-modèle des coefficients :

$$\alpha_i(\mathbf{x}) \approx \hat{\alpha}_i(\mathbf{x}) \quad (1.74)$$

où le méta-modèle peut être obtenu par PCE ou par GP par exemple. L'apprentissage des coefficients s'effectue sur un DoE constitué de $\alpha_i(\mathbf{x}_j)$, $i = 1, \dots, n_r$, $j = 1, \dots, n_x$ où n_x est la taille du DoE. Pour cela, la solution la plus simple est de partir d'un DoE de solutions de l'Eq. (1.64) $\mathbf{u}(\mathbf{x}_j)$, $j = 1, \dots, n_x$ puis de projeter chaque solution sur la base réduite :

$$\alpha_1(\mathbf{x}_j), \dots, \alpha_{n_r}(\mathbf{x}_j) = \Phi^T(\mathbf{u}(\mathbf{x}_j) - \Phi_0) \quad (1.75)$$

Grâce à l'approximation des coefficients, une approximation $\hat{\mathbf{u}}(\mathbf{x})$ peut être obtenue très facilement :

$$\mathbf{u}(\mathbf{x}) \approx \hat{\mathbf{u}}(\mathbf{x}) = \Phi_0 + \sum_{i=1}^{n_r} \hat{\alpha}_i(\mathbf{x}) \Phi_i \quad (1.76)$$

Remarques :

- Cette méthode est très peu intrusive car l'apprentissage des coefficients s'effectue sur des solutions du système matriciel sans avoir besoin d'obtenir et de projeter les données de l'Eq. (1.64) comme cela est nécessaire pour la méthode de Petrov-Galerkin.
- Les coefficients obtenus dans l'Eq. (1.75) sont différents de ceux obtenus par projection de Galerkin puisqu'ils sont la solution du problème :

$$\langle \hat{\mathbf{u}}(\mathbf{x}) - \mathbf{u}(\mathbf{x}), \Phi \omega \rangle = 0, \quad \forall \omega \in \mathbb{R}^{n_r} \quad (1.77)$$

Pour retrouver ces coefficients optimaux par projection de Petrov-Galerkin, il est nécessaire de définir W_r comme étant l'image de V_r par l'opérateur $\mathbf{K}^{-T}(\mathbf{x})$.

- Cette méthode permet aussi d'approcher des modèles non-linéaires puisque les coefficients sont obtenus en projetant les solutions du solveur sur la base réduite. Néanmoins, il sera généralement plus difficile de représenter la solution d'un solveur non-linéaire sur un espace affine ce qui engendrera une base réduite de plus grande dimension. De plus, l'apprentissage des coefficients par méta-modèles sera plus difficile.

1.3.3 Génération de l'espace affine V_r^a

Maintenant que la construction des coefficients sur la base réduite a été introduite, nous allons détailler la génération de l'espace vectoriel V_r et la translation Φ_0 caractérisant V_r^a . Les deux méthodes que nous passerons en revue sont la **POD** construite sur des *snapshots* et la méthode gloutonne.

1.3.3.1 Proper Orthogonal Decomposition

Une des premières solutions proposées pour construire l'espace vectoriel V_r est la **POD** (**SIROVICH [1987]**) très utilisée dans la littérature que ce soit sous le nom de **POD** en **MOR**, **Analyse en composantes principales**, ou **Principal Component Analysis (PCA)** en analyse statistique ou encore expansion **KL** en théorie des processus stochastiques. Celle-ci vise à déterminer la base optimale selon un critère que nous allons définir dans la suite. Dans cette partie nous allons détailler la *snapshot POD* où la base est apprise sur un **DoE** composé de solutions du solveur physique défini dans l'Eq. (1.64), aussi dénommées *snapshots*.

Pour construire la base **POD**, un échantillon de solutions de l'Eq. (1.64) est généré à partir d'un **DoE** de paramètres $\{\mathbf{x}_1^{\text{POD}}, \dots, \mathbf{x}_m^{\text{POD}}\}$ obtenu par une méthode d'échantillonnage (**LHS** par exemple) où m est le nombre de *snapshots*. Notons qu'une solution serait d'utiliser tous les *snapshots* afin de définir une base réduite par une méthode de Gram-Schmidt. Seulement, utiliser l'ensemble des vecteurs n'est pas efficace car cela ne prend pas en compte la redondance des informations de la sortie des solveurs. L'idée est alors d'extraire les modes principaux qui sont définis par les directions permettant d'exprimer le maximum de variance des coordonnées.

Nous supposons dans la suite que cette base est créée à partir d'un faible nombre de *snapshots*, c'est à dire $m \ll N$. Pour chacun des paramètres du **DoE** $\{\mathbf{x}_1^{\text{POD}}, \dots, \mathbf{x}_m^{\text{POD}}\}$ la solution de l'Eq. (1.64) est calculée en utilisant le solveur physique. Ceci permet de définir les *snapshots* $\{\mathbf{u}(\mathbf{x}_1^{\text{POD}}), \dots, \mathbf{u}(\mathbf{x}_m^{\text{POD}})\}$ à partir desquels la base réduite sera construite. La constante Φ_0 de l'approximation définie dans l'Eq. (1.65) est la moyenne empirique des *snapshots* :

$$\Phi_0 = \sum_{k=1}^m \mathbf{u}(\mathbf{x}_k^{\text{POD}}) \quad (1.78)$$

Les solutions centrées sont stockées dans une matrice notée **D** de taille $N \times m$ où pour rappel, N est la dimension de $\mathbf{u}(\mathbf{x})$.

$$\mathbf{D} = [\mathbf{u}(\mathbf{x}_0^{\text{POD}}) - \Phi_0, \dots, \mathbf{u}(\mathbf{x}_m^{\text{POD}}) - \Phi_0] \quad (1.79)$$

Par la suite, une **Décomposition en valeurs singulières**, ou **Singular Value Decomposition (SVD)** est réalisée sur la matrice des *snapshots* centrés. Ceci permet de définir la décomposition suivante :

$$\mathbf{D} = \mathbf{U} \mathbf{S} \mathbf{V}^T \quad (1.80)$$

où **U** est une matrice orthogonale de taille $N \times N$ contenant les vecteurs singuliers dits "de sortie", **S** est une matrice de dimension $N \times m$ dont les valeurs de la diagonale sont les valeurs singulières σ_i , $i = 1, \dots, m$ et dont tous les éléments non diagonaux sont nuls, **V** est une matrice orthogonale de taille $m \times m$ contenant les vecteurs singuliers dits "d'entrée". Les valeurs singulières sont rangées dans l'ordre décroissant : $\sigma_1 \geq \dots \geq \sigma_m \geq 0$. La matrice de projection Φ sera alors engendrée par les n_r premières colonnes de **U**. Le nombre n_r sera choisi selon les propriétés de la **SVD** suivantes :

- Pour tout $1 \leq n_r \leq m$, la matrice de projection Φ de taille n_r minimisant l'erreur de projection au sens des moindres carrées :

$$\sum_{i=1}^m \|\mathbf{u}(\mathbf{x}_i^{\text{POD}}) - \mathbf{P}^\Phi(\mathbf{u}(\mathbf{x}_i^{\text{POD}}))\|_2^2 \quad (1.81)$$

est donnée par les n_r premières colonnes de $\mathbf{V}$ où

$$\mathbf{P}^\Phi(\mathbf{u}(\mathbf{x}_i^{\text{POD}})) = \Phi_0 + \sum_{k=1}^{n_r} \langle \mathbf{u}(\mathbf{x}_i^{\text{POD}}) - \Phi_0, \Phi_k \rangle \Phi_k \quad (1.82)$$

représente la reconstruction de $\mathbf{u}(\mathbf{x}_i^{\text{POD}})$ sur l'espace V_r^a avec Φ_k les vecteurs de la matrice de projection Φ . Ainsi, la [SVD](#) permet de générer la meilleure base pour une taille donnée en minimisant l'erreur de projection sur les *snapshots* au sens des moindres carrés. Il ne reste plus qu'à déterminer le nombre n_r de modes [POD](#) retenus.

- Le choix de n_r est déterminé par le lien existant entre les valeurs singulières de la [SVD](#) et l'erreur de projection :

$$\sum_{i=1}^m \|\mathbf{u}(\mathbf{x}_i^{\text{POD}}) - \mathbf{P}^\Phi(\mathbf{u}(\mathbf{x}_i^{\text{POD}}))\|_2^2 = \sum_{k=n_r+1}^m \sigma_k^2 \quad (1.83)$$

En pratique, un pourcentage η est défini (typiquement 99.99%), le nombre n_r de modes conservés est alors :

$$n_r = \min_s \quad \text{sous contrainte} \quad \frac{\sum_{k=1}^s \sigma_k^2}{\sum_{k=1}^m \sigma_k^2} > \eta \quad (1.84)$$

Ce pourcentage η permet alors de définir l'erreur relative de projection au sens des moindres carrés atteinte par la base [POD](#).

Dans cette section, une méthode permettant de définir la meilleure base selon le critère de l'erreur de projection au sens des moindres carrés a été décrite. Son principal inconvénient est qu'elle est basée uniquement sur les *snapshots* qui sont sélectionnés par une méthode d'échantillonnage sur l'espace paramétrique $\mathcal{X}$. En revanche, il n'est pas possible d'estimer l'erreur de projection sur le domaine non couvert par les *snapshots*. Pour obtenir une base précise sur l'ensemble du domaine, la taille de l'échantillon de *snapshots* devient importante dans le cas d'un grand nombre de paramètres, ou d'un modèle physique particulièrement sensible aux paramètres.

1.3.3.2 Approche gloutonne

Une autre solution appelée approche gloutonne, ou *greedy procedure* permet de certifier la base réduite sur l'ensemble de l'espace paramétrique par l'utilisation d'une borne d'erreur ([VEROY et collab. \[2003\]](#); [ZAHM et collab. \[2017\]](#); [CHEN et collab. \[2017\]](#)). Grâce à cette borne de l'erreur, l'erreur de projection sera contrôlée afin que le modèle réduit soit considéré comme suffisamment précis.

Pour cette méthode, une base Φ initiale est nécessaire, celle-ci peut être obtenue par une méthode [POD](#) ou plus simplement par la base engendrée par le vecteur $\mathbf{u}(\bar{\mathbf{x}})$ où $\bar{\mathbf{x}}$ est un point particulier de $\mathcal{X}$ (son centre s'il existe par exemple). L'objectif est alors de déterminer le paramètre où la solution du solveur est la moins bien représentée :

$$\mathbf{x}_*^\Phi = \arg \max_{\mathbf{x} \in \mathcal{X}} \|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\| \quad (1.85)$$

Par la suite, le vecteur $\mathbf{u}(\mathbf{x}_*)^\Phi$ sera ajouté à la base Φ et la base sera alors orthogonalisée. La solution du problème (1.85) sera de nouveau cherchée et le vecteur ajouté à la matrice de projection Φ jusqu'à ce que

$$\max_{\mathbf{x} \in \mathcal{X}} \|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\| \leq \zeta \quad (1.86)$$

où ζ est une précision que l'on souhaite obtenir sur la base réduite. Cependant, la solution du problème (1.85) est inaccessible car cela demanderait trop d'appels au solveur.

Pour y remédier, une estimation de l'erreur $\Delta^\Phi(\mathbf{x})$ est définie par :

$$c_\Delta(\mathbf{x}) \|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\| \leq \Delta^\Phi(\mathbf{x}) \leq C_\Delta(\mathbf{x}) \|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\| \quad (1.87)$$

où les variables $0 < c_\Delta(\mathbf{x}) \leq C_\Delta(\mathbf{x}) < \infty$ mesurent l'efficacité de l'estimation de l'erreur. Les variables $c_\Delta(\mathbf{x})$ et $C_\Delta(\mathbf{x})$ sont généralement obtenues à partir d'une analyse de la matrice de rigidité $\mathbf{K}(\mathbf{x})$, par exemple en utilisant les valeurs propres. Le problème (1.85) est alors remplacé par le problème suivant :

$$\mathbf{x}_*^\Phi = \operatorname{argmax}_{\mathbf{x} \in \mathcal{X}} \Delta^\Phi(\mathbf{x}) \quad (1.88)$$

Les étapes de la procédure permettant la construction dite gloutonne de la base réduite sont résumées dans l'Algorithme 8.

Algorithm 8: Greedy procedure for reduced basis

input : $\mathbf{K}$, $\mathbf{F}$, an initial reduced basis Φ , a target precision ζ
while $\max_{\mathbf{x} \in \mathcal{X}} \|\mathbf{u}(\mathbf{x}) - \mathbf{P}^\Phi(\mathbf{u}(\mathbf{x}))\| \leq \zeta$ **do**
 $\mathbf{x}_*^\Phi = \operatorname{argmax}_{\mathbf{x} \in \mathcal{X}} \Delta^\Phi(\mathbf{x})$;
 Compute $\mathbf{u}(\mathbf{x}_*^\Phi)$;
 Add $\mathbf{u}(\mathbf{x}_*^\Phi)$ to the reduced basis Φ ;
 Update the error estimation $\Delta^\Phi(\mathbf{x})$;
end
output: Φ

Dans le cas simplifié d'une matrice $\mathbf{K}$ symétrique réelle définie positive, l'estimation de l'erreur est donnée par le résidu :

$$\|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\| \approx \|\mathbf{K}(\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x}))\| = \|\mathbf{F}(\mathbf{x}) - \mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x})\| \quad (1.89)$$

De plus, il est possible de définir :

$$\gamma_{V_r}(\mathbf{x}) \leq \frac{\|\mathbf{K}(\mathbf{x})\mathbf{u}\|}{\|\mathbf{u}\|} \leq \Gamma_{V_r}(\mathbf{x}), \quad \forall \mathbf{u} \in V_r \setminus \{0\} \quad (1.90)$$

Plusieurs propositions ont été faites dans la littérature afin de définir $\gamma_{V_r}(\mathbf{x})$ et $\Gamma_{V_r}(\mathbf{x})$ par analyse des valeurs propres de la matrice de rigidité ou par construction d'une estimation au cours de la procédure de l'Algorithme 8 (HUYNH et collab. [2007] ; CHEN et collab. [2017]) ou par des méthodes aléatoires (JANON et collab. [2016] ; BALABANOV et NOUY [2019] ; SMETANA et collab. [2019]). Des recherches ont aussi été menées pour améliorer l'efficacité des estimations de l'erreur afin d'obtenir $c_\Delta(\mathbf{x}) \sim 1$ et $C_\Delta(\mathbf{x}) \sim 1$ par préconditionnement (ZAHM et NOUY [2016]).

Bien que cette méthode soit extrêmement performante pour obtenir une base réduite précise sur l'ensemble du domaine $\mathcal{X}$, celle-ci ne peut être utilisée que lorsqu'une décomposition affine des éléments est disponible ; sans cela, le résidu du problème réduit

nécessite l'assemblage d'opérateurs dont la complexité est celle du problème d'origine de taille N . De plus, ceci nécessite d'effectuer une optimisation sur les paramètres d'entrée du solveur ce qui est inapplicable dans le cas d'un système multidisciplinaire avec un grand nombre de variables de couplage.

1.3.4 Réduction de modèle avec extraction d'une valeur d'intérêt linéaire

À présent, nous ne nous intéressons plus à la solution du système linéaire (1.64) mais à une valeur d'intérêt scalaire dépendante linéairement de $\mathbf{u}(\mathbf{x})$. Dans ce cadre dit *goal-oriented*, la quantité d'intérêt est définie par :

$$s(\mathbf{x}) = \mathbf{L}(\mathbf{x})\mathbf{u}(\mathbf{x}) \quad (1.91)$$

où $\mathbf{L}(\mathbf{x})$ est un opérateur permettant d'extraire la quantité d'intérêt à partir de la solution du solveur $\mathbf{u}(\mathbf{x})$. De nombreux exemples sont disponibles avec ce type de quantité d'intérêt comme la moyenne de la température dans le cas du refroidissement de composants électriques (ZAHM et collab. [2017]), ou l'extraction de certaines valeurs particulières de $\mathbf{u}(\mathbf{x})$ (JANON et collab. [2016]; ZAHM et collab. [2017]). Dans le cadre de l'optimisation, cette quantité d'intérêt peut par exemple correspondre à la fonction objectif. Bien que la méthode que nous allons détailler permette d'approcher un ensemble de quantités d'intérêt ($s(\mathbf{x})$ serait alors un vecteur), nous allons traiter le cas d'une quantité d'intérêt scalaire afin de simplifier les notations. Cette quantité d'intérêt peut alors être corrigée par la solution du problème dual détaillé par la suite.

1.3.4.1 Problème dual

Le problème dual est défini par :

$$\mathbf{K}(\mathbf{x})\mathbf{Q}(\mathbf{x}) = \mathbf{L}^T(\mathbf{x}) \quad (1.92)$$

où $\mathbf{Q}(\mathbf{x})$ est la solution du problème dual. La résolution de ce système linéaire peut être approchée, de la même façon que le système (1.64), par des approches MOR via l'utilisation d'une base d'approximation $\mathbf{B}_r^Q$. Ceci permet alors d'obtenir une approximation $\mathbf{Q}_r(\mathbf{x})$ de $\mathbf{Q}(\mathbf{x})$.

1.3.4.2 Correction Duale

La correction duale permet d'obtenir une meilleure approximation de la quantité à partir de la solution du problème dual. Plusieurs approches ont été proposées dans la littérature avec notamment l'approche primale-duale (MACHIELS et collab. [1998] MADAY et collab. [2002]; CUONG et collab. [2005]; JANON et collab. [2016]; ZAHM et collab. [2017]) ou l'approche *saddle point* (DAHMEN et collab. [2014]; ZAHM et collab. [2017]). Ici nous nous intéresserons à l'approche primale-duale car il a été montré dans ZAHM et collab. [2017] que les résultats sont similaires à ceux obtenus avec l'approche *saddle point* mais que l'obtention de l'approximation avec l'approche *saddle point* est plus coûteuse. La solution de l'approximation primale-duale est définie par :

$$s_r^{\text{PD}}(\mathbf{x}) = \mathbf{L}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{Q}_r^T(\mathbf{x}) (\mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})) \quad (1.93)$$

L'approximation primale $\mathbf{L}(\mathbf{x})\mathbf{u}_r(\mathbf{x})$ est alors corrigée en utilisant le résidu du problème à la place de $\mathbf{L}(\mathbf{x})$ dans le problème dual. L'avantage de cette méthode est que résoudre les

deux problèmes par des modèles MOR est plus performant que d'utiliser un modèle MOR deux fois plus performant. En effet, quelques vecteurs obtenus par POD par exemple, permettent de générer une bonne approximation d'un solveur. Cependant, doubler le nombre de vecteurs de la POD ne permet pas un gain significatif. Ainsi, il est plus efficace de combiner les deux modèles.

Finalement, grâce à la correction duale, une approximation de la Quantité d'intérêt (QI) linéaire peut être obtenue. Nous allons à présent nous intéresser à l'estimation de l'erreur sur la QI avec la correction duale.

1.3.4.3 Majoration de l'erreur issue de la correction duale

Une majoration de l'erreur sur la quantité d'intérêt est possible en définissant :

$$\beta_r(\mathbf{x}) = \max_{\mathbf{v} \in V \setminus \{0\}} \frac{|(\mathbf{L}(\mathbf{x}) - \mathbf{Q}_r^T(\mathbf{x})\mathbf{K}(\mathbf{x})) \mathbf{v}|}{\|\mathbf{v}\|} \quad (1.94)$$

La majoration de l'erreur est alors donnée par :

$$|s(\mathbf{x}) - s_r^{\text{PD}}(\mathbf{x})| \leq \beta_r(\mathbf{x}) \|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\|_{W_r} \quad (1.95)$$

Il est alors possible d'adapter l'approche gloutonne définie dans la Section 1.3.3.2 avec ce type de méthode afin d'obtenir une base certifiée sur la quantité d'intérêt.

Notons que pour être efficace cette méthode nécessite la résolution d'un problème dual sur l'opérateur $\mathbf{L}(\mathbf{x})$ défini par l'Eq. (1.92). Cette solution du problème dual est peu coûteuse à obtenir si une factorisation de la matrice de rigidité a préalablement été calculée. Cela élimine donc l'utilisation de méthodes itératives pour la résolution du système linéaire primal puisque cela reviendrait à doubler le coût numérique de la construction de la base réduite. En contrepartie, la correction duale permet de fortement diminuer l'erreur commise sur la quantité d'intérêt à taille de base équivalente. Cette méthode est donc à privilégier lorsque cela est possible.

1.3.5 Bilan sur la réduction d'ordre de modèle

La MOR permet d'obtenir une approximation d'un solveur disciplinaire en cherchant la solution d'un système linéaire de grande dimension (cf Eq. (1.64)) sur un espace de dimension fortement réduite V_r^a . Ceci permet de fortement diminuer le coût numérique de l'inversion du système linéaire de grande dimension. Pour être encore plus efficace, la décomposition affine des éléments permet d'éviter la projection des données $\mathbf{K}$ et $\mathbf{F}$. Cependant, le calcul exact de cette décomposition est intrusif et son approximation induit une erreur difficile à estimer. La prise en compte d'un modèle réduit sur la quantité d'intérêt linéaire permet d'obtenir une meilleure approximation. Finalement, l'approche POD+I permet d'être peu intrusif mais il est difficile d'estimer l'erreur commise et de certifier le modèle réduit obtenu. Notons que de nombreuses recherches ont été également menées afin de réduire le coût de calcul d'un solveur non-linéaire en utilisant des méthodes MOR. Ceci peut être obtenu par projection des itérations de Newton sur la base réduite ou par des méthodes d'interpolation telles que la DEIM.

Dans la suite, nous allons observer comment la MOR a été utilisée au sein d'un processus d'optimisation, mono ou multidisciplinaire, afin de valider l'intérêt de cette méthode.

1.4 Utilisation de la réduction d'ordre de modèle dans un contexte *multi-query*

Les objectifs de cette section sont :

- de détailler les utilisations de la MOR dans un contexte *multi-query*.
- de détailler l'adaptation des méthodes de MOR au contexte multidisciplinaire
- d'aborder l'utilisation de l'estimation de l'erreur fournie.

Dans cette partie, plusieurs méthodes utilisant la MOR vont être présentées dans le cadre *multi-query*, c'est-à-dire dans le cas où de nombreux appels au code de calcul devront être effectués pour des paramètres différents. Le cadre défini ici inclut donc l'optimisation, l'analyse de fiabilité ou encore l'analyse de sensibilité. L'objectif est alors de déterminer l'intérêt des méthodes développées d'un point de vue réduction du coût numérique tout en précisant leurs limitations. Des méthodes ont aussi été développées afin de produire un modèle de MOR dans le cadre de la multi-fidélité (BENAMARA et collab. [2017]). Cependant, l'objectif étant de prendre en compte l'incertitude du modèle approché plutôt que de l'utiliser comme un modèle de plus basse fidélité, nous ne détaillerons pas ces méthodes.

1.4.1 Cadre

Dans la suite de la thèse nous nous intéressons à un problème d'optimisation non-constrait :

$$\mathbf{x}_* \in \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmin}} s(\mathbf{x}) \quad (1.96)$$

où la QI $s(\mathbf{x})$ est obtenue à partir de la solution d'un solveur $\mathbf{u}(\mathbf{x})$:

$$s(\mathbf{x}) = l(\mathbf{x}, \mathbf{u}(\mathbf{x})) \quad (1.97)$$

où l est un opérateur linéaire ou non mais supposé explicite qui permet d'extraire la QI. Dans cette partie, le solveur définissant le vecteur d'état $\mathbf{u}(\mathbf{x})$ peut être linéaire comme dans l'Eq. (1.64), non-linéaire ou fourni plus particulièrement par un modèle multidisciplinaire en considérant $\mathbf{u}(\mathbf{x}) = \{\mathbf{y}_*^1, \dots, \mathbf{y}_*^{n_d}\}$ la solution du système multidisciplinaire de l'Eq. (1.30). Comme l'algorithme d'optimisation devra faire appel plusieurs fois au code de calcul pour calculer la QI $s(\mathbf{x})$, le coût cumulé de ces appels peut être prohibitif. Ces appels répétés définissent le contexte *multi-query* et la réduction d'ordre de modèle est utilisée pour diminuer ce coût. Ici, l'objectif est donc de présenter les différentes approches permettant d'utiliser la MOR dans ce cadre.

1.4.2 Approche *Offline-Online*

La méthode la plus utilisée dans le cadre du couplage de la MOR et de l'optimisation est l'approche *Offline-Online*. L'objectif est alors de calculer un méta-modèle par MOR puis de remplacer le solveur coûteux par son approximation. Ainsi, les appels répétés se feront sur un modèle moins coûteux. Cela revient dans le cadre de l'optimisation à découpler l'apprentissage du méta-modèle du processus d'optimisation comme proposé lors de l'optimisation directe du méta-modèle (Section 1.1.3.3.1).

La première étape de cette méthode est de construire une approximation par MOR de la variable d'état afin d'obtenir :

$$\mathbf{u}_r(\mathbf{x}) \approx \mathbf{u}(\mathbf{x}) \quad (1.98)$$

Cette première étape, coûteuse puisque la construction de l'approximation nécessite généralement des appels aux solveurs comme dans le cas de la *snapshot* **POD** ou de l'approche gloutonne, est nommée phase *Offline*. Généralement, l'approche gloutonne est préférée dans le cas de la méthode *Offline-Online* car cela permet de s'assurer que l'erreur commise par la **MOR** est contrôlée par le critère d'arrêt de l'approche gloutonne. Une fois que cette approximation est construite et considérée comme fiable, une approximation de la **QI** peut être obtenue :

$$s_r(\mathbf{x}) = l(\mathbf{x}, \mathbf{u}_r(\mathbf{x})) \approx s(\mathbf{x}) \quad (1.99)$$

Comme cette quantité approchée $s_r(\mathbf{x})$, obtenue par **MOR** est supposée être moins coûteuse numériquement, il est alors proposé de l'utiliser à la place de la quantité d'intérêt exacte $s(\mathbf{x})$.

Dans le cadre de l'optimisation, cela revient à résoudre le problème suivant :

$$\mathbf{x}_* \in \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmin}} s_r(\mathbf{x}) \quad (1.100)$$

Cette étape est nommée phase *Online*. L'idée de l'approche *Offline-Online* est de ne pas estimer l'erreur due à la **MOR** lors de la phase *Online* car l'on suppose que la phase *Offline* permet d'obtenir une approximation suffisamment précise. Plusieurs cas de figure sont considérés dans la suite en fonction de la linéarité ou non du solveur et de la quantité d'intérêt étudiée.

1.4.2.1 Cas d'un solveur linéaire avec quantité d'intérêt linéaire

Le cas où le solveur et la **QI** sont tous les deux linéaires est l'un des plus propices à l'approche *Offline-Online*. Dans ce cas,

$$l(\mathbf{x}, \mathbf{u}(\mathbf{x})) = \mathbf{L}(\mathbf{x})\mathbf{u}(\mathbf{x}) \quad (1.101)$$

et il est possible de définir une approximation de la **QI** par correction duale comme cela a été détaillé dans la Section 1.3.4. Ceci a été appliqué pour l'analyse de sensibilité par **JANON et collab.** [2016]. Grâce à la borne de l'erreur sur la **QI**, il est possible d'obtenir un modèle **MOR** qui permet d'approcher la **QI** selon une certaine précision définie par l'utilisateur grâce à l'Algorithme 8. Cette approche est rendue possible par l'estimation de l'erreur commise sur la **QI**. Seulement, pour être efficace, une décomposition affine des éléments est nécessaire. Lorsque celle-ci n'est pas disponible une approximation supplémentaire, via généralement la **DEIM**, est construite. L'erreur sur cette approximation est difficile à obtenir ce qui limite l'intérêt de l'approche gloutonne car l'erreur induite par **DEIM** est généralement obtenue à partir d'une validation croisée.

1.4.2.2 Cas d'un solveur linéaire avec quantité d'intérêt non-linéaire

Lorsque le solveur est linéaire mais que la **QI** ne l'est pas, l'Algorithme 8 permet de valider un modèle **MOR** suffisamment précis pour la solution $\mathbf{u}(\mathbf{x})$, c'est à dire tel que $\|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\| \leq \zeta$. Seulement, il n'est pas possible à partir de cette information d'obtenir une estimation fiable de l'erreur sur la quantité d'intérêt sans information sur la fonction l . Le même problème est présent pour la décomposition affine des éléments.

1.4.2.3 Cas d'un solveur non-linéaire

Lorsque le solveur n'est pas linéaire, il est difficile d'obtenir une méthode d'estimation de l'erreur produite par projection des itérations de Newton sur une base réduite. De

plus, sans décomposition affine des éléments de la Jacobienne du problème, la réduction d'ordre de modèle n'est pas suffisamment efficace. [AMSALLEM et collab. \[2015\]](#) utilisent l'hyper-réduction de modèle par [DEIM](#) pour obtenir une approximation de la décomposition affine des éléments. La certification du modèle approché est alors effectuée par validation croisée. Cela souligne la difficulté d'obtenir une estimation de l'erreur fiable sans avoir de décomposition affine des éléments exacte. La [POD+I](#) peut aussi être utilisée car aucune décomposition n'est nécessaire. À nouveau, comme l'erreur de l'interpolation est difficile à calculer, une validation croisée est nécessaire. La [POD+I](#) a été appliquée dans le cas de l'optimisation multidisciplinaire d'une aile 3D par [COELHO et collab. \[2008\]](#) en considérant le problème multidisciplinaire comme un modèle non-linéaire monolithique et par [RIPEPI et collab. \[2018\]](#) dans le cas d'un calcul non-linéaire aérodynamique.

1.4.2.4 Cas d'un solveur multidisciplinaire

Dans le cadre d'un solveur multidisciplinaire, l'idée est d'exploiter la structure de la [MDA](#) pour proposer des approximations des solveurs disciplinaires. Pour simplifier les notations et le discours, nous allons nous placer dans un cas couplant uniquement deux disciplines comme illustré sur la Figure 1.5. La [MDA](#) est alors définie par le système couplé suivant :

$$\begin{cases} \mathbf{y}^1 &= f^1(\mathbf{x}, \mathbf{y}^2) \\ \mathbf{y}^2 &= f^2(\mathbf{x}, \mathbf{y}^1) \end{cases} \quad (1.102)$$

où f^1 et f^2 sont les solveurs disciplinaires, $\mathbf{y}^1$ et $\mathbf{y}^2$ sont les variables de couplage. Lorsque cela est possible les deux solveurs disciplinaires peuvent être approchés par des méthodes de [MOR](#). Par exemple dans le cas d'un solveur linéaire $\mathbf{y}^1(\mathbf{x}, \mathbf{y}^2)$ solution de :

$$\mathbf{K}^1(\mathbf{x}, \mathbf{y}^2) \mathbf{y}^1(\mathbf{x}, \mathbf{y}^2) = \mathbf{F}^1(\mathbf{x}, \mathbf{y}^2) \quad (1.103)$$

une matrice de projection Φ^1 de taille n_r^1 associée à un vecteur Φ_0^1 peut être générée par [POD](#) afin de réduire le coût de calcul de l'inversion. On définit alors $\alpha^1(\mathbf{x}, \mathbf{y}^2)$ la solution de :

$$\mathbf{K}_r^1(\mathbf{x}, \mathbf{y}^2) \alpha^1(\mathbf{x}, \mathbf{y}^2) = \mathbf{F}_r^1(\mathbf{x}, \mathbf{y}^2) \quad (1.104)$$

où $\mathbf{K}_r^1(\mathbf{x}, \mathbf{y}^2)$ et $\mathbf{F}_r^1(\mathbf{x}, \mathbf{y}^2)$ sont définis par l'Eq. (1.71). On obtient alors une approximation du solveur disciplinaire :

$$\mathbf{y}_r^1(\mathbf{x}, \mathbf{y}^2) = f_r^1(\mathbf{x}, \mathbf{y}^2) = \Phi_0^1 + \sum_{i=1}^{n_r^1} \alpha_i^1(\mathbf{x}, \mathbf{y}^2) \Phi_i^1 \quad (1.105)$$

De la même façon en définissant une matrice de projection Φ^2 de taille n_r^2 associée à un vecteur Φ_0^2 on définit une approximation $\mathbf{y}_r^2(\mathbf{x}, \mathbf{y}^1)$ obtenue via le solveur réduit $f_r^2(\mathbf{x}, \mathbf{y}^1)$. Une approximation de la [MDA](#) est obtenue en résolvant le système suivant :

$$\begin{cases} \mathbf{y}_r^1 &= f_r^1(\mathbf{x}, \mathbf{y}_r^2) \\ \mathbf{y}_r^2 &= f_r^2(\mathbf{x}, \mathbf{y}_r^1) \end{cases} \quad (1.106)$$

Ceci a été appliqué entre autre par [LEGRESLEY et ALONSO \[2000, 2004\]](#) pour l'optimisation du design d'une aile d'avion. Seulement, cette approche a plusieurs inconvénients :

1. La génération des *snapshots* est moins triviale que dans le cas présenté dans la Section 1.3.3.1. En effet, il est nécessaire ici d'échantillonner selon les entrées du solveur f^1 à savoir selon $\mathbf{x} \in \mathcal{X}$ et $\mathbf{y}^2 \in \mathcal{C}^2$. Or si échantillonner dans $\mathcal{X}$ est simple, $\mathcal{C}^2$

est un espace inconnu de grande dimension. Il est alors nécessaire d'utiliser des solutions obtenues par l'algorithme permettant de résoudre la **MDA** présentée dans la Section 1.2.2. Un échantillon $\{\mathbf{x}_1, \dots, \mathbf{x}_d\}$ contenant d éléments de $\mathcal{X}$ est généré par une méthode d'échantillonnage classique de type **LHS**. Pour chacun de ces paramètres, la **MDA** est résolue ce qui permet d'obtenir des échantillons des variables de couplage. Ensuite, les solutions des solveurs disciplinaires obtenues lors des itérations nécessaires à la résolution de la **MDA** peuvent être utilisées pour la génération de la base réduite par **POD**. Pour générer les bases, des **MDA** non convergées (**COELHO et collab. [2009]**) peuvent aussi être utilisées en relâchant le critère d'arrêt ou en effectuant un nombre limité d'itérations sur le solveur de **MDA**.

2. La décomposition affine des éléments est rarement disponible. Par exemple, dans le cas aéroélastique, le déplacement influe sur le maillage de l'aérodynamique et son interaction sera difficile à obtenir sur les données du solveur aérodynamique. De plus, les méthodes d'hyper-réduction sont difficiles à appliquer puisque la **DEIM** nécessite d'effectuer une optimisation sur les entrées du solveur à savoir $\mathbf{x} \in \mathcal{X}$ et $\mathbf{y}^2 \in \mathcal{C}^2$ ce qui n'est pas possible car l'espace $\mathcal{C}^2$ n'est pas défini explicitement.

Enfin, **COELHO et collab. [2009]** proposent de remplacer un des solveurs disciplinaires (par exemple le premier solveur) en utilisant une méthode de **POD+I**. Pour y parvenir, la première étape est de générer une base réduite pour les 2 disciplines afin de réduire la dimension des variables de couplage. Pour la suite, un méta-modèle d'une des disciplines est obtenu par apprentissage sur les variables de couplage réduites. La **MDA** réduite est définie par le système couplé mettant en oeuvre les variables de couplage réduites avec l'équation suivante :

$$\begin{cases} \boldsymbol{\alpha}^1 &= f_{\text{POD}}^1(\mathbf{x}, \boldsymbol{\alpha}^2) \\ \boldsymbol{\alpha}^2 &= f_{\text{POD}}^2(\mathbf{x}, \boldsymbol{\alpha}^1) \end{cases} \quad (1.107)$$

où :

$$\begin{aligned} f_{\text{POD}}^1(\mathbf{x}, \boldsymbol{\alpha}_r^2) &= (\boldsymbol{\Phi}^2)^T \left[f^1 \left(\mathbf{x}, \boldsymbol{\Phi}_0^1 + \sum_{i=1}^{n_r^1} \alpha_i^1(\mathbf{x}, \mathbf{y}^2) \boldsymbol{\Phi}_i^1 \right) - \boldsymbol{\Phi}_0^2 \right] \\ f_{\text{POD}}^2(\mathbf{x}, \boldsymbol{\alpha}_r^1) &= (\boldsymbol{\Phi}^1)^T \left[f^2 \left(\mathbf{x}, \boldsymbol{\Phi}_0^2 + \sum_{i=1}^{n_r^2} \alpha_i^2(\mathbf{x}, \mathbf{y}^1) \boldsymbol{\Phi}_i^2 \right) - \boldsymbol{\Phi}_0^1 \right] \end{aligned} \quad (1.108)$$

dont la solution est $\boldsymbol{\alpha}_*^1(\mathbf{x})$ et $\boldsymbol{\alpha}_*^2(\mathbf{x})$. Notons que le nombre de variables de couplage dans ce nouveau système a fortement diminué grâce à la **POD**. On considère la discipline réduite suivante :

$$\begin{aligned} f_{\text{POD}}^1: \mathcal{X} \times \mathcal{C}_r^2 &\rightarrow \mathcal{C}_r^1 \\ \mathbf{x}, \quad \boldsymbol{\alpha}^2 &\mapsto (\boldsymbol{\Phi}^2)^T \left(f_r^1(\mathbf{x}, \boldsymbol{\Phi}_0^1 + \sum_{i=1}^{n_r^1} \alpha_i^1(\mathbf{x}, \mathbf{y}^2) \boldsymbol{\Phi}_i^1) - \boldsymbol{\Phi}_0^2 \right) \end{aligned} \quad (1.109)$$

où $\mathcal{C}_r^1$ et $\mathcal{C}_r^2$ sont les images par projection des espaces $\mathcal{C}^1$ et $\mathcal{C}^2$. Comme le nombre d'entrées et de sorties est faible, un méta-modèle de cette fonction peut être généré :

$$f_{\text{POD}}^1(\mathbf{x}, \boldsymbol{\alpha}^2) \approx \hat{f}_{\text{POD}}^1(\mathbf{x}, \boldsymbol{\alpha}^2) \quad (1.110)$$

La **MDA** approchée suivante est alors obtenue :

$$\begin{cases} \boldsymbol{\alpha}^1 &= \hat{f}_{\text{POD}}^1(\mathbf{x}, \boldsymbol{\alpha}^2) \\ \boldsymbol{\alpha}^2 &= f_{\text{POD}}^2(\mathbf{x}, \boldsymbol{\alpha}^1) \end{cases} \quad (1.111)$$

dont la solution est $\hat{\boldsymbol{\alpha}}_*^1(\mathbf{x})$ et $\hat{\boldsymbol{\alpha}}_*^2(\mathbf{x})$. Cette solution est obtenue par un solveur de problème couplé comme Gauss-Seidel (cf Section 1.2.2.3). Bien que cette méthode soit très intéressante pour réduire le coût numérique d'un des solveurs lors de la phase *Online*, ceci

entraîne une augmentation du nombre d'appels à l'autre solveur. En effet, f_{POD}^2 dépend de f^2 . En utilisant ce type d'approximation, l'estimation de l'erreur commise est difficile à obtenir. Pour la diminuer, [COELHO et collab. \[2009\]](#) proposent d'effectuer une itération de Gauss-Seidel à partir de la solution $\hat{\alpha}_*^1(\mathbf{x})$ et $\hat{\alpha}_*^2(\mathbf{x})$. Enfin, bien que les espaces $\mathcal{C}_r^1$ et $\mathcal{C}_r^2$ soient de faible dimension, échantillonner dans ces espaces reste difficile. Il est alors nécessaire d'utiliser des solutions de [MDA](#) : [COELHO et collab. \[2009\]](#) proposent d'utiliser des [MDA](#) non convergées en faisant une seule itération de l'algorithme de Gauss-Seidel pour obtenir les bases et entraîner le méta-modèle.

1.4.2.5 Bilan des approches *Offline-Online*

Si on fait le bilan, les approches *Offline-Online* permettent de fortement réduire le coût numérique de l'optimisation en approchant la fonction objectif par une méthode [MOR](#). Dans ce cas, l'erreur commise par la [MOR](#) n'est pas calculée lors de la phase *Online* ce qui nécessite de certifier la [MOR](#) lors de la phase *Offline*. Or, cette borne de l'erreur est soit impossible à obtenir et nécessite une validation croisée, soit numériquement difficile à calculer entraînant une phase *Offline* trop pénalisante.

1.4.3 Correction de la base réduite en phase *Online*

Une autre solution proposée dans la littérature est d'utiliser l'estimation de l'erreur sur la phase *Online* afin de corriger la base réduite si celle-ci se révèle imprécise. Cette méthode couple donc le processus *multi-query* avec la [MOR](#) et sera donc uniquement présentée sur le cas de l'optimisation.

1.4.3.1 Méthode *on-the-fly*

La première idée présentée par [GOGU et PASSIEUX \[2013\]](#) ; [GOGU \[2015\]](#) est de construire la base réduite au fur et à mesure de la phase *Online* en utilisant une méthode à la volée, *on-the-fly*. Par la suite, il est proposé d'utiliser l'estimation de l'erreur commise par la [MOR](#) notée $\Delta^\Phi(\mathbf{x})$ qui sera donnée par le résidu comme critère de précision de la base réduite. Le cadre défini ici est celui d'un solveur linéaire défini par l'Eq. (1.64). Cette méthode a été appliquée notamment à l'optimisation topologique ([GOGU \[2015\]](#)) et à l'approche [EGO](#) ([SOILAHODINE et collab. \[2017\]](#)).

La première étape est de calculer une matrice de projection initiale $\Phi = [\phi_1, \dots, \phi_{n_r}]$ (et une moyenne ϕ_0) qui sera enrichie par la suite. [GOGU et PASSIEUX \[2013\]](#) ; [GOGU \[2015\]](#) ; [SOILAHODINE et collab. \[2017\]](#) proposent d'utiliser comme base initiale des *snapshot* orthonormalisés. L'enrichissement de la base est régi par le critère de précision sur la base réduite suivant :

$$\Delta^\Phi(\mathbf{x}) = \|\mathbf{K}\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})\| \leq \zeta \quad (1.112)$$

où ζ est un critère de précision défini par l'utilisateur similaire à celui utilisé dans l'Algorithme 8. Ainsi, à chaque évaluation du solveur défini par l'Eq. (1.64) pour un paramètre $\mathbf{x} \in \mathcal{X}$, l'approximation $\mathbf{u}_r(\mathbf{x})$ puis le résidu $\mathbf{K}\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})$ sont calculés. Si le critère défini dans l'Eq. (1.112) est respecté, la [MOR](#) est considérée comme suffisamment précise, on peut alors approcher $\mathbf{u}(\mathbf{x})$ par son approximation $\mathbf{u}_r(\mathbf{x})$. Dans le cas où le critère n'est pas respecté, $\mathbf{u}(\mathbf{x})$ est calculé par inversion du système (1.64). L'information obtenue en $\mathbf{x}$ est utilisée pour améliorer la base de projection :

$$\Phi = \left[\phi_1, \dots, \phi_{n_r}, \frac{\mathbf{u}(\mathbf{x}) - \phi_0 - \sum_{k=1}^{n_r} \langle \mathbf{u}(\mathbf{x}) - \phi_0, \phi_k \rangle \phi_k}{\|\mathbf{u}(\mathbf{x}) - \phi_0 - \sum_{k=1}^{n_r} \langle \mathbf{u}(\mathbf{x}) - \phi_0, \phi_k \rangle \phi_k\|} \right] \quad (1.113)$$

permettant d'obtenir une base orthogonale de taille $n_r + 1$. Afin que la base réduite ne soit pas trop importante, il est aussi proposé par GOGU [2015]; GOGU et PASSIEUX [2013] de supprimer le premier vecteur Φ_1 de la matrice de projection au fur et à mesure des enrichissements à partir d'un certain rang.

Cette méthode permet d'obtenir une base réduite précise (selon le critère défini par l'Eq. (1.112)) autour des paramètres $\mathbf{x}$ qui sont proposés lors de la phase *Online*. Lors d'un processus d'optimisation par exemple, seuls les points intéressants pour l'optimisation seront visités. Cela permet ainsi d'obtenir une base réduite précise uniquement dans les zones où la solution du problème d'optimisation est supposée être contrairement à l'approche *Offline-Online* dont la base doit être précise sur l'ensemble de l'espace de design $\mathcal{X}$ et sera alors de grande taille. La conséquence étant qu'une base réduite de grande taille alourdit la phase *Online*. La phase *Offline* est elle aussi moins coûteuse puisqu'il n'est pas nécessaire d'obtenir une base précise sur tout l'espace de design sachant que celle-ci sera enrichie lors de la phase *Online*.

1.4.3.2 Méthode par région de confiance

Une autre solution proposée dans la littérature s'inspire de la méthode de région de confiance qui a été détaillée dans la Section 1.1.2.2 et est nommée *Trust-Region Proper Orthogonal Decomposition* (TRPOD) (ARIAN et collab. [2001]; FAHL [2004]). L'idée est alors d'utiliser comme modèle local, le modèle donné par la MOR et d'utiliser de la même façon une région de confiance paramétrée par un rayon ρ_n dont l'évolution sera similaire à celle de la Section 1.1.2.2.

Soit $\mathbf{x}_n$ le paramètre de design à l'itération n de la TRPOD, on définit l'approximation suivante obtenue par POD :

$$m_n(\mathbf{x}_n + \mathbf{s}) = l(\mathbf{x}_n + \mathbf{s}, \mathbf{u}_r(\mathbf{x}_n + \mathbf{s})) \quad (1.114)$$

où l est la vraie fonction permettant d'extraire la fonction objectif et $\mathbf{u}_r(\mathbf{x})$ est l'approximation obtenue par POD. Le calcul du nouveau pas $\mathbf{s}_n$ est obtenu via l'équation suivante :

$$\mathbf{s}_n \in \arg \min_{\|\mathbf{s}\| \leq \rho_n} m_n(\mathbf{x}_n + \mathbf{s}) \quad (1.115)$$

L'efficacité du modèle est elle aussi prédite par le même rapport :

$$r_n = \frac{l(\mathbf{x}, \mathbf{u}(\mathbf{x})) - l(\mathbf{x}_n + \mathbf{s}_n, \mathbf{u}(\mathbf{x}_n + \mathbf{s}_n))}{m_n(\mathbf{x}_n) - m_n(\mathbf{x}_n + \mathbf{s}_n)} \quad (1.116)$$

Notons que pour obtenir r_n , il est nécessaire de calculer $l(\mathbf{x}_n + \mathbf{s}_n, \mathbf{u}(\mathbf{x}_n + \mathbf{s}_n))$ et donc la solution du solveur $\mathbf{u}(\mathbf{x}_n + \mathbf{s}_n)$ au nouveau point $\mathbf{x}_n + \mathbf{s}_n$. L'évolution du rayon sera alors la même que dans la Section 1.1.2.2 via les constantes $\eta_1, \eta_2, \gamma_1, \gamma_2$ et γ_3 .

Une proposition a été faite pour améliorer la TRPOD en utilisant l'estimation de l'erreur, fournie par la POD comme proposé dans la Section 1.3.3.2, nommée *Error-based Trust Region* (ETR) (YUE et MEERBERGEN [2013]) où la région de confiance ne sera plus une sphère dans l'espace paramétrique mais définie par l'estimation de l'erreur $\Delta^\Phi(\mathbf{x}_n + \mathbf{s})$. La contrainte du problème de l'Eq. (1.115) est alors modifiée selon :

$$\|\mathbf{s}\| \leq \rho_n \text{ remplacé par } \frac{\Delta^\Phi(\mathbf{x}_n + \mathbf{s})}{m_n(\mathbf{x}_n + \mathbf{s})} \leq \epsilon_L \quad (1.117)$$

avec ϵ_L une borne d'erreur choisie par l'utilisateur. ZAHR et FARHAT [2015] proposent une approche assez similaire où la région de confiance est définie par :

$$\Delta^\Phi(\mathbf{x}_n + \mathbf{s}) \leq \epsilon_L \quad (1.118)$$

mais où une évolution de la borne de l'erreur ϵ_L similaire à celle développée pour les approches TR est proposée. Enfin, une autre solution nommée *Error-based Penalty* (EP) (YUE et MEERBERGEN [2013]; Zahr et collab.) ajoute une pénalisation à $m_n(\mathbf{x}_n + \mathbf{s})$ lors de la résolution du sous-problème (1.115) dépendant de l'estimation de l'erreur $\Delta^\Phi(\mathbf{x}_n + \mathbf{s})$.

1.4.4 Bilan sur la réduction d'ordre de modèle dans un contexte *multi-query*

Dans cette partie, nous avons pu voir comment les méthodes de réduction d'ordre de modèle ont été utilisées dans la littérature pour l'optimisation. Plusieurs stratégies co-existent : une approche *Offline-Online* basée sur un découplage de l'optimisation et de la MOR et des méthodes basées sur l'estimation de l'erreur dans la phase *Online*. L'avantage de la stratégie *Offline-Online* est la réduction drastique du coût numérique de la phase *Online*, même si ceci se fait au détriment de la phase *Offline* où obtenir une MOR robuste est coûteux et incertain. L'approche *on-the-fly* est prometteuse car cela permet d'enrichir la MOR au fur et à mesure de l'optimisation. Toutefois, cela requiert de calculer la borne de l'erreur lors de la phase *Online* ce qui peut être coûteux. Dans la suite, nous proposons une évolution de cette méthode dans le but d'améliorer l'estimation de l'erreur.

1.5 Synthèse du chapitre

Dans ce chapitre nous avons présenté diverses approches qui nous ont paru utiles à la compréhension des développements qui seront menés dans la suite de cette thèse. Dans un premier temps, nous avons détaillé plusieurs méthodes permettant de minimiser une fonction f selon certains paramètres de design $\mathbf{x}$. Nous avons mis en évidence les limites des méthodes basées sur les modèles locaux car la convergence vers le minimum global n'est pas assurée. De plus, lorsque le gradient est inconnu, la construction du méta-modèle local est une étape coûteuse. Pour y remédier, les approches exploitant les méta-modèles globaux sont des méthodes intéressantes dans notre contexte puisque le gradient n'est pas nécessaire et que le méta-modèle global est construit au fur et à mesure du processus d'optimisation par des enrichissements successifs. De plus, ces méthodes permettent, en partie, de répondre à l'écueil des minima locaux lorsque l'erreur de prédiction commise est estimée comme cela a été mis en évidence en comparant les approches **SBO** et **EGO**. À travers l'algorithme d'optimisation **EGO**, nous avons mis en avant une méthode permettant de faire un compromis entre exploitation et exploration via les **GP**. En effet, les **GP** présentent l'avantage de fournir intrinsèquement une estimation de l'incertitude sur la prédiction puisque la variance du **GP** peut servir d'indicateur d'erreur. Néanmoins cet indicateur est basé sur l'hypothèse que la fonction interpolée est la réalisation d'un **GP** et dépend en pratique des caractéristiques du **GP** choisies telles que le noyau, le terme de régression... Pour autant, les méthodes par méta-modèle utilisant une estimation de l'erreur commise sont particulièrement intéressantes dans le cadre de l'optimisation car cela permet d'enrichir le méta-modèle au fur et à mesure dans les zones les plus intéressantes pour l'optimisation. C'est sur ce résultat essentiel que seront articulées certaines décisions prises par la suite.

Puis, plusieurs méthodologies ont été proposées pour la résolution d'un problème **MDO**. La **MDO** met en oeuvre l'optimisation d'une fonction coûteuse dont la valeur dépend d'un système non linéaire couplé noté **MDA**. Dans un premier temps, nous avons différencié les approches partitionnées des approches monolithiques en soulignant que l'avantage des méthodes partitionnées est de pouvoir utiliser des solveurs disciplinaires qui ont été optimisés et testés par des experts du domaine. Toutefois, la problématique de cette méthode est la communication des informations entre les solveurs disciplinaires. En pratique, les informations ou variables de couplage circulant entre les solveurs sont des champs discrétisés de très grande dimension. Pour résoudre la **MDO**, des méthodes ont été développées, exploitant la structure de ce problème particulier afin de limiter le nombre d'appels aux solveurs disciplinaires. Cependant, ces méthodes exploitent souvent le gradient des solveurs disciplinaires ou nécessitent un nombre limité de variables de couplage. Parmi celles-ci, **EGMDO** propose de remplacer les solveurs disciplinaires coûteux par des **GP**. L'idée est alors de propager les incertitudes des solveurs disciplinaires à la fonction objectif. La fonction objectif aléatoire obtenue est utilisée pour construire un critère d'enrichissement qui sera utilisé dans l'esprit de l'algorithme **EGO**. À nouveau, nous pouvons observer l'importance de la prise en compte de l'erreur lors du processus d'optimisation. Néanmoins, construire des **GP** des solveurs disciplinaires nécessite un nombre de variables de couplage réduit ce qui est incompatible avec les hypothèses de notre problème. Pour répondre à cette difficulté, une analyse des méthodes issues de la réduction d'ordre de modèle a été proposée.

La **MOR** par projection a été développée pour diminuer le coût des solveurs physiques en proposant une approximation de la solution du solveur sur un espace de dimension bien inférieur. Celle-ci peut alors être utilisée pour réduire le coût numérique des solveurs

disciplinaires. La projection de Galerkin permet d'obtenir les coefficients sur la base à partir de la projection des équations physiques du solveur. Toutefois, cette méthode nécessite une hypothèse contraignante : la décomposition affine des éléments afin d'éviter des calculs dont la complexité serait celle du problème d'origine pour la construction du problème projeté. Cette décomposition peut être estimée par plusieurs méthodes et nécessite une intrusion dans le code du solveur ou une approximation supplémentaire. De plus, l'erreur obtenue sur cette approximation est très difficile à obtenir sans validation croisée. Notons que la projection de Galerkin permet de construire une estimation de l'erreur qui pourrait être utilisée dans un contexte d'optimisation. L'approche **POD+I** est très intéressante car il est possible grâce à des méta-modèles d'obtenir une approximation très peu coûteuse du solveur sans nécessiter de décomposition affine des éléments mais aucune estimation d'erreur n'est disponible. Enfin, une correction duale a été présentée afin de proposer l'approximation d'une quantité d'intérêt linéaire.

Finalement, nous nous sommes intéressés à l'application de ces méthodes dans un contexte *multi-query*. Dans une première stratégie, la **MOR** est utilisée pour créer une approximation de la **QI**. Cette stratégie est alors décomposée en deux phases : la phase *Offline* et la phase *Online*. Lors de la phase *Offline*, une approximation précise sur l'ensemble du domaine paramétrique est obtenue, généralement à l'aide d'une approche gloutonne. Cette construction nécessite des appels répétés au solveur et la projection des termes de la décomposition affine des éléments (et son estimation si cela est nécessaire). L'évaluation lors du contexte *multi-query* sera alors faite uniquement sur l'approximation lors de la phase *Online*. Toutefois, comme nous l'avons vu dans la première partie, l'exploitation pure du méta-modèle peut entraîner d'importantes erreurs lors de l'optimisation. Nous noterons une idée intéressante proposée par **COELHO et collab. [2009]** sur l'application de la **MOR** dans le contexte multidisciplinaire où la **MOR** est utilisée pour limiter le nombre de variables de couplage permettant ainsi d'utiliser la **POD+I** dans ce contexte. Dans une seconde stratégie, plusieurs méthodes ont été proposées pour évaluer l'erreur commise lors de la phase *Online* afin de limiter l'impact numérique de la phase *Offline* tout en améliorant la **MOR** lorsque cela est nécessaire. Parmi ces méthodes on retrouve la méthode *on-the-fly* et la méthode par région de confiance adaptée à la **POD**.

De nombreuses méthodes ont été développées afin de résoudre des problèmes d'optimisation dans le cadre général ou dans le cadre multidisciplinaire. De plus, la **MOR** permet de réduire le coût de calcul des solveurs physiques. Seulement, peu de méthodes utilisent l'estimation de l'erreur sur la **QI** lors du processus d'optimisation. La première contribution de cette thèse sera d'utiliser l'estimation de l'erreur sur la **QI** en la couplant avec un algorithme d'optimisation globale de type **EGO**. Dans un second temps, il sera proposé d'étendre la méthode **EGMDO** aux variables de couplage de grande dimension (modélisées par des champs discrétisés) tout en faisant l'hypothèse que les espaces des variables de couplage ne sont pas connus. Pour cela, une adaptation des travaux de **COELHO et collab. [2009]** est proposée afin de remplacer chaque solveur disciplinaire par une approximation par **POD+I**. De plus, cette approximation sera couplée à la méthode **EGMDO** afin de résoudre le problème **MDO** en utilisant l'incertitude sur les méta-modèles disciplinaires.

1.6 Références

AITKEN, A. C. 1927, «XXV.—On Bernoulli's numerical solution of algebraic equations», *Proceedings of the Royal Society of Edinburgh*, vol. 46, p. 289–305. [32](#), [33](#)

- ALARIE, S., C. AUDET, A. E. GHERIBI, M. KOKKOLARAS et S. LE DIGABEL. 2021, «Two decades of blackbox optimization applications», *EURO Journal on Computational Optimization*, vol. 9, doi: <https://doi.org/10.1016/j.ejco.2021.100011>, p. 100 011, ISSN 2192-4406. URL <https://www.sciencedirect.com/science/article/pii/S2192440621001386>. 12
- ALEXANDROV, N. et R. LEWIS. 2004, «Reconfigurability in MDO problem synthesis, part 1», dans *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, doi: 10.2514/6.2004-4307. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2004-4307>. 35
- ALLIER, P.-E., L. CHAMOIN et P. LADEVÈZE. 2015, «Proper generalized decomposition computational methods on a benchmark problem : introducing a new strategy based on constitutive relation error minimization», *Advanced Modeling and Simulation in Engineering Sciences*, vol. 2, n° 1, doi: 10.1186/s40323-015-0038-4, p. 17, ISSN 2213-7467. URL <https://doi.org/10.1186/s40323-015-0038-4>. 52
- ALLISON, J., M. KOKKOLARAS, M. ZAWISLAK et P. Y. PAPALAMBROS. 2005, «On the use of analytical target cascading and collaborative optimization for complex system design», dans *6th World Congress on Structural and Multidisciplinary Optimization Rio de Janeiro*, vol. 30, Citeseer. 35, 38
- AMSALLEM, D., M. ZAHR, Y. CHOI et C. FARHAT. 2015, «Design optimization using hyper-reduced-order models», *Structural and Multidisciplinary Optimization*, vol. 51, n° 4, doi: 10.1007/s00158-014-1183-y, p. 919–940, ISSN 1615-1488. URL <https://doi.org/10.1007/s00158-014-1183-y>. 54, 55, 63
- ARIAN, E., M. FAHL et E. SACHS. 2001, «Trust-region proper orthogonal decomposition for flow control», *Institute for computer applications in science and engineering Hampton VA*. 66
- ARNST, M., R. GHANEM, E. PHIPPS et J. RED-HORSE. 2012, «Dimension reduction in stochastic modeling of coupled problems», *International Journal for Numerical Methods in Engineering*, vol. 92, n° 11, doi: <https://doi.org/10.1002/nme.4364>, p. 940–968. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4364>. 43
- BALABANOV, O. et A. NOUY. 2019, «Randomized linear algebra for model reduction. Part I : Galerkin methods and error estimation», *Advances in Computational Mathematics*, vol. 45, p. 2969–3019. URL <https://hal.archives-ouvertes.fr/hal-01899836>. 58
- BALLING, R. J. et J. SOBIESZCZANSKI-SOBIESKI. 1996, «Optimization of coupled systems-a critical overview of approaches», *AIAA journal*, vol. 34, n° 1, p. 6–17. 34, 35, 36
- BARTOLI, N., T. LEFEBVRE, S. DUBREUIL, R. OLIVANTI, R. PRIEM, N. BONIS, J. R. R. A. MARTINS et J. MORLIER. 2019, «Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design», *Aerospace Science and Technology*, vol. 90, doi: <https://doi.org/10.1016/j.ast.2019.03.041>, p. 85–102, ISSN 1270-9638. URL <https://www.sciencedirect.com/science/article/pii/S1270963818306011>. 23
- BATHE, K., H. ZHANG et M. WANG. 1995, «Finite element analysis of incompressible and compressible fluid flows with free surfaces and structural interactions», *Computers & Structures*, vol. 56, n° 2-3, p. 193–213. 28

- BATHE, K.-J. et H. ZHANG. 2004, «Finite element developments for general fluid flows with structural interactions», *International Journal for Numerical Methods in Engineering*, vol. 60, n° 1, p. 213–232. 28
- BELLMAN, R. E. 2021, *Dynamic Programming*, Princeton University Press, ISBN 9781400835386, doi: doi:10.1515/9781400835386. URL <https://doi.org/10.1515/9781400835386>. 17
- BENAMARA, T., P. BREITKOPF, I. LEPOT, C. SAINVITU et P. VILLON. 2017, «Multi-fidelity POD surrogate-assisted optimization : Concept and aero-design study», *Structural and Multidisciplinary Optimization*, vol. 56, n° 6, doi: 10.1007/s00158-017-1730-4, p. 1387–1412, ISSN 1615-1488. URL <https://doi.org/10.1007/s00158-017-1730-4>. 61
- BERVEILLER, M., B. SUDRET et M. LEMAIRE. 2006, «Stochastic finite element : a non intrusive approach by regression», *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, vol. 15, n° 1-3, p. 81–92. 17
- BETTEBGHOR, D., N. BARTOLI, S. GRIHON, J. MORLIER et M. SAMUELIDES. 2011, «Surrogate modeling approximation using a mixture of experts based on EM joint estimation», *Structural and Multidisciplinary Optimization*, vol. 43, n° 2, doi: 10.1007/s00158-010-0554-2, p. 243–259. URL <https://hal.archives-ouvertes.fr/hal-01852300>. 22
- BLOEBAUM, C. L. 1995, «Coupling strength-based system reduction for complex engineering design», *Structural optimization*, vol. 10, n° 2, doi: 10.1007/BF01743538, p. 113–121, ISSN 1615-1488. URL <https://doi.org/10.1007/BF01743538>. 32, 33
- BONCORAGLIO, G., C. FARHAT et C. BOU-MOSLEH. 2021, «Model reduction framework with a new take on active subspaces for optimization problems with linearized fluid-structure interaction constraints», *International Journal for Numerical Methods in Engineering*, vol. 122, n° 19, p. 5450–5481. 16, 28
- BOUHLEL, M. A., N. BARTOLI, A. OTSMANE et J. MORLIER. 2016, «Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction», *Structural and Multidisciplinary Optimization*, vol. 53, n° 5, p. 935–952. 17, 24
- BRACONNIER, T., M. FERRIER, J.-C. JOUHAUD, M. MONTAGNAC et P. SAGAUT. 2011, «Towards an adaptive POD/SVD surrogate model for aeronautic design», *Computers & Fluids*, vol. 40, n° 1, p. 195–209. 17, 24
- BRAUN, R., P. GAGE, I. KROO et I. SOBIESKI. 1996, «Implementation and performance issues in collaborative optimization», 6th Symposium on Multidisciplinary Analysis and Optimization, doi: 10.2514/6.1996-4017. URL <https://arc.aiaa.org/doi/abs/10.2514/6.1996-4017>. 35, 38
- BRAUN, R. D. 1996, *Collaborative Optimization : An Architecture for Large-Scale Distributed Design*, thèse de doctorat, Stanford, CA, USA. UMI Order No. GAX96-30279. 35, 38
- BREITKOPF, P. et R. F. COELHO. 2013, *Multidisciplinary design optimization in computational mechanics*, John Wiley & Sons. 34

- BREVAULT, L., M. BALESDENT, N. BÉREND et R. LE RICHE. 2015, «Decoupled MDO formulation for interdisciplinary coupling satisfaction under uncertainty», *AIAA Journal*, vol. 54, n° 1, doi: 10.2514/1.J054121, p. 186 – 205. URL <https://hal-emse.ccsd.cnrs.fr/emse-01185241>. 41
- BROOMHEAD, D. S. et D. LOWE. 1988, «Radial basis functions, multi-variable functional interpolation and adaptive networks», Cahier de recherche, Royal Signals and Radar Establishment Malvern (United Kingdom). 17
- BROYDEN, C. G. 1970, «The convergence of a class of double-rank minimization algorithms 1. general considerations», *IMA Journal of Applied Mathematics*, vol. 6, n° 1, p. 76–90. 13, 46
- BUI-THANH, T., K. WILLCOX et O. GHATTAS. 2008, «Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications», *AIAA Journal*, vol. 46, n° 10, doi: 10.2514/1.35850, p. 2520–2529. URL <https://doi.org/10.2514/1.35850>. 54
- BYRD, R. H., R. B. SCHNABEL et G. A. SHULTZ. 1987, «A trust region algorithm for nonlinearly constrained optimization», *SIAM Journal on Numerical Analysis*, vol. 24, n° 5, p. 1152–1170. 11
- CAMPBELL, R. et E. PATERSON. 2011, «Fluid–structure interaction analysis of flexible turbomachinery», *Journal of Fluids and Structures*, vol. 27, n° 8, doi: <https://doi.org/10.1016/j.jfluidstructs.2011.08.010>, p. 1376–1391, ISSN 0889-9746. URL <https://www.sciencedirect.com/science/article/pii/S0889974611001319>. 27
- CARLBERG, K., C. FARHAT, J. CORTIAL et D. AMSALLEM. 2013, «The GNAT method for nonlinear model reduction : Effective implementation and application to computational fluid dynamics and turbulent flows», *Journal of Computational Physics*, vol. 242, doi: <https://doi.org/10.1016/j.jcp.2013.02.028>, p. 623–647, ISSN 0021-9991. URL <https://www.sciencedirect.com/science/article/pii/S0021999113001472>. 54
- CARRIER, G., D. DESTARAC, A. DUMONT, M. MEHEUT, I. S. E. DIN, J. PETER, S. B. KHELIL, J. BREZILLON et M. PESTANA. «Gradient-based aerodynamic optimization with the elsA software», dans *52nd Aerospace Sciences Meeting*, doi: 10.2514/6.2014-0568. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2014-0568>. 16
- CHATURANTABUT, S. et D. C. SORENSSEN. 2010, «Nonlinear model reduction via discrete empirical interpolation», *SIAM Journal on Scientific Computing*, vol. 32, n° 5, p. 2737–2764. 54
- CHEN, P., A. QUARTERONI et G. ROZZA. 2017, «Reduced basis methods for uncertainty quantification», *SIAM Journal on Uncertainty Quantification*, vol. 5, n° 1, p. 813–869. 57, 58
- CHEVREUIL, M. et A. NOUY. 2012, «Model order reduction based on proper generalized decomposition for the propagation of uncertainties in structural dynamics», *International Journal for Numerical Methods in Engineering*, vol. 89, n° 2, p. 241–268. 52
- CHINESTA, F., A. AMMAR, A. LEYGUE et R. KEUNINGS. 2011, «An overview of the proper generalized decomposition with applications in computational rheology», *Journal of Non-Newtonian Fluid Mechanics*, vol. 166, n° 11, p. 578–592. 52

- CHITTICK, I. R. et J. R. R. A. MARTINS. 2009, «An asymmetric suboptimization approach to aerostructural optimization», *Optimization and Engineering*, vol. 10, n° 1, p. 133–152. [35](#)
- COELHO, R. F., P. BREITKOPF et C. KNOPF-LENOIR. 2008, «Model reduction for multidisciplinary optimization - Application to a 2D wing», *Structural and Multidisciplinary Optimization*, vol. 37, n° 1, doi: 10.1007/s00158-007-0212-5, p. 29–48, ISSN 1615-1488. URL <https://doi.org/10.1007/s00158-007-0212-5>. [27](#), [28](#), [63](#)
- COELHO, R. F., P. BREITKOPF, C. KNOPF-LENOIR et P. VILLON. 2009, «Bi-level model reduction for coupled problems», *Structural and Multidisciplinary Optimization*, vol. 39, n° 4, doi: 10.1007/s00158-008-0335-3, p. 401–418. [16](#), [28](#), [55](#), [64](#), [65](#), [69](#)
- CRAMER, E., J. DENNIS, P. FRANK, R. LEWIS et G. R. SHUBIN. 1994, «Problem formulation for multidisciplinary optimization», *SIAM Journal on Optimization*, vol. 4, p. 754–776. [35](#), [36](#)
- CUONG, N. N., K. VEROY et A. T. PATERA. 2005, «Certified real-time solution of parametrized partial differential equations», dans *Handbook of materials modeling*, Springer, p. 1529–1564. [59](#)
- DAHMEN, W., C. PLESKEN et G. WELPER. 2014, «Double greedy algorithms : Reduced basis methods for transport dominated problems», *ESAIM : Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, vol. 48, n° 3, doi: 10.1051/m2an/2013103, p. 623–663. URL <http://www.numdam.org/articles/10.1051/m2an/2013103/>. [59](#)
- DAL SANTO, N. et A. MANZONI. 2019, «Hyper-reduced order models for parametrized unsteady Navier-Stokes equations on domains with variable shape», *Advances in Computational Mathematics*, vol. 45, n° 5, doi: 10.1007/s10444-019-09722-9, p. 2463–2501. [55](#)
- DAMBLIN, G., M. COUPLET et B. IOOSS. 2013, «Numerical studies of space-filling designs : optimization of latin hypercube samples and subprojection properties», *Journal of Simulation*, vol. 7, n° 4, p. 276–289. [17](#)
- DAVIDON, W. C. 1991, «Variable metric method for minimization», *SIAM Journal on Optimization*, vol. 1, n° 1, p. 1–17. [33](#)
- DEGROOTE, J. 2013, «Partitioned simulation of fluid-structure interaction», *Archives of Computational Methods in Engineering*, vol. 20, n° 3, doi: 10.1007/s11831-013-9085-5, p. 185–238, ISSN 1886-1784. URL <https://doi.org/10.1007/s11831-013-9085-5>. [29](#)
- DELBECQ, S., M. BUDINGER et A. REYSSET. 2020, «Benchmarking of monolithic MDO formulations and derivative computation techniques using OpenMDAO», *Structural and Multidisciplinary Optimization*, vol. 62, n° 2, p. 645–666. [34](#)
- DENG, N., Y. XIAO et F. ZHOU. 1993, «Nonmonotonic trust region algorithm», *Journal of optimization theory and applications*, vol. 76, n° 2, p. 259–285. [14](#)
- DREYFUS, G., J. MARTINEZ, M. SAMUELIDES, M. GORDON, F. BADRAN, S. THIRIA et L. HÉRAULT. 2002, *Réseaux de neurones*, vol. 39, Eyrolles Paris. [17](#)

- DUBREUIL, S., N. BARTOLI, C. GOGU et T. LEFEBVRE. 2016, «Propagation of modeling uncertainty by polynomial chaos expansion in multidisciplinary analysis», *Journal of Mechanical Design*, vol. 138, n° 11, p. 111–141. 41
- DUBREUIL, S., N. BARTOLI, C. GOGU et T. LEFEBVRE. 2020, «Towards an efficient global multidisciplinary design optimization algorithm», *Structural and Multidisciplinary Optimization*, vol. 62, n° 4, doi: 10.1007/s00158-020-02514-6, p. 1739–1765, ISSN 1615-1488. URL <https://doi.org/10.1007/s00158-020-02514-6>. 27, 36, 39, 41, 42, 49
- DUBREUIL, S., N. BARTOLI, C. GOGU, T. LEFEBVRE et J. MAS COLOMER. 2018a, «Extreme value oriented random field discretization based on an hybrid polynomial chaos expansion - Kriging approach», *Computer Methods in Applied Mechanics and Engineering*, vol. 332, doi: 10.1016/j.cma.2018.01.009, p. 540–571. URL <https://hal.archives-ouvertes.fr/hal-01712026>. 43
- DUBREUIL, S., N. BARTOLI, T. LEFEBVRE et C. GOGU. 2018b, «Efficient global multidisciplinary optimization based on surrogate models», dans *2018 Multidisciplinary Analysis and Optimization Conference*, doi: 10.2514/6.2018-3745. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2018-3745>. 27, 36, 49
- DUTKA-MALEN, I., R. LEBRUN, B. SAASSOUH et B. SUDRET. 2009, «Implementation of a polynomial chaos toolbox in OpenTURNS and applications to structural reliability and sensitivity analyses», *The 10th International Conference on Structural Safety and Reliability ICOSSAR 2009*. 17
- ECHARD, B., N. GAYTON et M. LEMAIRE. 2011, «AK-MCS : an active learning reliability method combining kriging and Monte Carlo simulation», *Structural Safety*, vol. 33, n° 2, p. 145–154. 18
- EVERSON, R. et L. SIROVICH. 1995, «Karhunen–Loeve procedure for gappy data», *JOSA A*, vol. 12, n° 8, p. 1657–1664. 54
- FAHL, M. 2004, *Trust-region Methods for Flow Control based on Reduced Order Modelling*, doctoral thesis, Universität Trier, doi: 10.25353/ubtr-xxxx-908b-671d. 66
- FARZAM FAR, M., F. MARTIN, A. BELAHSEN, L. MONTIER et T. HENNERON. 2018, «Orthogonal interpolation method for order reduction of a synchronous machine model», *IEEE Transactions on Magnetics*, vol. 54, n° 2, doi: 10.1109/TMAG.2017.2768328, p. 1–6. 55
- FLETCHER, R. 1970, «A new approach to variable metric algorithms», *The computer journal*, vol. 13, n° 3, p. 317–322. 13, 46
- FLETCHER, R. 2000, «Restricted step methods», dans *Practical Methods of Optimization*, chap. 5, John Wiley & Sons, Ltd, ISBN 9781118723203, p. 95–109, doi: <https://doi.org/10.1002/9781118723203.ch5>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118723203.ch5>. 14
- FLETCHER, R. et M. J. POWELL. 1963, «A rapidly convergent descent method for minimization», *The computer journal*, vol. 6, n° 2, p. 163–168. 33
- FORRESTER, A. I. J., A. SOBESTER et A. J. KEANE. 2008, *Engineering Design via Surrogate Modelling*, John Wiley and Sons Ltd. xi, 18, 19, 20, 22, 24, 25, 37

- FUGLSTAD, G.-A., D. SIMPSON, F. LINDGREN et H. RUE. 2019, «Constructing priors that penalize the complexity of Gaussian random fields», *Journal of the American Statistical Association*, vol. 114, n° 525, p. 445–452. [19](#)
- GHANEM, R. G. et P. D. SPANOS. 2003, *Stochastic finite elements : a spectral approach*, Courier Corporation. [17](#)
- GINSBOURGER, D., R. L. RICHE et L. CARRARO. 2010, «Kriging is well-suited to parallelize optimization», dans *Computational intelligence in expensive optimization problems*, Springer, p. 131–162. [23](#)
- GOGU, C. 2015, «Improving the efficiency of large scale topology optimization through on-the-fly reduced order model construction», *International Journal for Numerical Methods in Engineering*, vol. 101, n° 4, p. 281–304. [65](#), [66](#)
- GOGU, C. et J.-C. PASSIEUX. 2013, «Efficient surrogate construction by combining response surface methodology and reduced order modeling», *Structural and Multidisciplinary Optimization*, vol. 47, n° 6, doi: 10.1007/s00158-012-0859-4, p. 821–837, ISSN 1615-1488. URL <https://doi.org/10.1007/s00158-012-0859-4>. [65](#), [66](#)
- GOLDFARB, D. 1970, «A family of variable-metric methods derived by variational means», *Mathematics of computation*, vol. 24, n° 109, p. 23–26. [13](#), [46](#)
- GRIPPO, L., F. LAMPARIELLO et S. LUCIDI. 1989, «A truncated newton method with non-monotone line search for unconstrained optimization», *Journal of Optimization Theory and Applications*, vol. 60, n° 3, p. 401–419. [13](#)
- GUYAN, R. J. 1965, «Reduction of stiffness and mass matrices», *AIAA Journal*, vol. 3, n° 2, doi: 10.2514/3.2874, p. 380. URL <https://hal.archives-ouvertes.fr/hal-01711552>. [52](#)
- HAFTKA, R. T. 1985, «Simultaneous analysis and design», *AIAA Journal*, vol. 23, n° 7, doi: 10.2514/3.9043, p. 1099–1103. URL <https://doi.org/10.2514/3.9043>. [35](#)
- HASTIE, T., R. TIBSHIRANI, J. H. FRIEDMAN et J. H. FRIEDMAN. 2009, *The elements of statistical learning : data mining, inference, and prediction*, vol. 2, Springer. [17](#)
- HE, P., R. HALDER, K. FIDKOWSKI, K. MAKI et J. R. MARTINS. 2021, «An efficient nonlinear reduced-order modeling approach for rapid aerodynamic analysis with OpenFOAM», dans *AIAA Scitech 2021 Forum*, p. 1476, doi: 10.2514/6.2021-1476. [54](#)
- HESTHAVEN, J. et S. UBBIALI. 2018, «Non-intrusive reduced order modeling of nonlinear problems using neural networks», *Journal of Computational Physics*, vol. 363, p. 55–78. [55](#)
- HULME, K. et C. BLOEBAUM. 2000, «Simulation-based comparison of multidisciplinary design optimization solution strategies using cascade», *Structural and Multidisciplinary Optimization*, vol. 19, doi: 10.1007/s001580050083, p. 17–35. [36](#), [37](#)
- HUYNH, D. B. P., G. ROZZA, S. SEN et A. T. PATERA. 2007, «A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants», *Comptes Rendus Mathématique*, vol. 345, n° 8, p. 473–478. [58](#)

- HWANG, J. T., D. Y. LEE, J. W. CUTLER et J. R. R. A. MARTINS. 2014, «Large-scale multidisciplinary optimization of a small satellite's design and operation», *Journal of Spacecraft and Rockets*, vol. 51, n° 5, p. 1648–1663. 16
- HWANG, J. T. et J. R. R. A. MARTINS. 2018, «A computational architecture for coupling heterogeneous numerical models and computing coupled derivatives», *ACM Trans. Math. Softw.*, vol. 44, n° 4, doi: 10.1145/3182393, ISSN 0098-3500. URL <https://doi.org/10.1145/3182393>. 35
- JAKOBSSON, S. et O. AMOIGNON. 2007, «Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization», *Computers & Fluids*, vol. 36, n° 6, p. 1119–1136. 16
- JANON, A., M. NODET et C. PRIEUR. 2016, «Goal-oriented error estimation for the reduced basis method, with application to sensitivity analysis», *Journal of Scientific Computing*, vol. 68, n° 1, p. 21–41. 58, 59, 62
- JIN, R., W. CHEN et A. SUDJANTO. 2003, «An efficient algorithm for constructing optimal design of computer experiments», dans *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 37009, p. 545–554. 17
- JONES, D. R. 2001, «A taxonomy of global optimization methods based on response surfaces», *Journal of Global Optimization*, vol. 21, n° 4, p. 345–383. 19
- JONES, D. R., M. SCHONLAU et W. J. WELCH. 1998, «Efficient global optimization of expensive black-box functions», *Journal of Global Optimization*, vol. 13, n° 4, doi: 10.1023/A:1008306431147, p. 455–492, ISSN 1573-2916. URL <http://dx.doi.org/10.1023/A:1008306431147>. 18, 22
- KATOCH, S., S. S. CHAUHAN et V. KUMAR. 2021, «A review on genetic algorithm : past, present, and future», *Multimedia Tools and Applications*, vol. 80, n° 5, doi: 10.1007/s11042-020-10139-6, p. 8091–8126, ISSN 1573-7721. URL <https://doi.org/10.1007/s11042-020-10139-6>. 12
- KAUFMAN, M., V. BALABANOV, A. GIUNTA, B. GROSSMAN, W. MASON, S. BURGEE, R. HAFTKA et L. T. WATSON. 1996, «Variable-complexity response surface approximations for wing structural weight in HSCT design», *Computational Mechanics*, vol. 18, n° 2, p. 112–126. 17
- KENNEDY, G. J. et J. R. R. A. MARTINS. 2014, «A parallel finite-element framework for large-scale gradient-based design optimization of high-performance structures», *Finite Elements in Analysis and Design*, vol. 87, p. 56–73. 16
- KENNEDY, M. C. et A. O'HAGAN. 2000, «Predicting the output from a complex computer code when fast approximations are available», *Biometrika*, vol. 87, n° 1, p. 1–13. 18
- KENWAY, G., G. KENNEDY et J. R. R. A. MARTINS. 2014, «Aerostructural optimization of the common research model configuration», dans *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, doi: 10.2514/6.2014-3274. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2014-3274>. 16

- KIRKPATRICK, S. 1984, «Optimization by simulated annealing : Quantitative studies», *Journal of Statistical Physics*, vol. 34, n° 5, doi: 10.1007/BF01009452, p. 975–986, ISSN 1572-9613. URL <https://doi.org/10.1007/BF01009452>. 12
- KIRKPATRICK, S., C. D. GELATT et M. P. VECCHI. 1983, «Optimization by simulated annealing», *Science*, vol. 220, n° 4598, doi: 10.1126/science.220.4598.671, p. 671–680, ISSN 0036-8075. URL <https://science.sciencemag.org/content/220/4598/671>. 12
- KOKKOLARAS, M., R. FELLINI, H. KIM, N. MICHELENA et P. PAPALAMBROS. 2002, «Extension of the Target Cascading formulation to the design of product families», *Structural and Multidisciplinary Optimization*, vol. 24, doi: 10.1007/s00158-002-0240-0, p. 293–301. 35
- KRIGE, D. G. 1951, «A statistical approach to some basic mine valuation problems on the witwatersrand», *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 52, n° 6, p. 119–139. 17
- LADEVÈZE, P. 1985, «New algorithms : mechanical framework and development», *Comptes rendus de l'Académie des Sciences*, vol. 300, n° 2, p. 41–44. 52
- LADEVÈZE, P. 2012, *Nonlinear computational structural mechanics : new approaches and non-incremental methods of calculation*, Springer Science & Business Media. 52
- LADEVÈZE, P., J.-C. PASSIEUX et D. NÉRON. 2010, «The latin multiscale computational method and the proper generalized decomposition», *Computer Methods in Applied Mechanics and Engineering*, vol. 199, n° 21-22, p. 1287–1296. 52
- LE GRATIET, L. 2013, *Multi-fidelity Gaussian process regression for computer experiments*, Theses, Université Paris-Diderot - Paris VII. URL <https://tel.archives-ouvertes.fr/tel-00866770>. 18
- LE GRATIET, L., C. CANNAMELA et B. IOOSS. 2014, «A bayesian approach for global sensitivity analysis of (multifidelity) computer codes», *SIAM Journal on Uncertainty Quantification*, vol. 2, n° 1, p. 336–363. 41
- LEGRESLEY, P. et J. ALONSO. 2000, «Airfoil design optimization using reduced order models based on proper orthogonal decomposition», dans *Fluids 2000 Conference and Exhibit*, doi: 10.2514/6.2000-2545. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2000-2545>. 63
- LEGRESLEY, P. et J. ALONSO. 2004, «Improving the performance of design decomposition methods with POD», dans *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, doi: 10.2514/6.2004-4465. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2004-4465>. 63
- LEON, A., S. MUELLER, P. DE LUCA, R. SAID, J.-L. DUVAL et F. CHINESTA. 2019, «Non-intrusive proper generalized decomposition involving space and parameters : application to the mechanical modeling of 3d woven fabrics», *Advanced Modeling and Simulation in Engineering Sciences*, vol. 6, n° 1, doi: 10.1186/s40323-019-0137-8, p. 13, ISSN 2213-7467. URL <https://doi.org/10.1186/s40323-019-0137-8>. 52
- LIU, D. C. et J. NOCEDAL. 1989, «On the limited memory BFGS method for large scale optimization», *Mathematical programming*, vol. 45, n° 1, p. 503–528. 33

- LOMBARDI, M., N. PAROLINI et A. QUARTERONI. 2013, «Radial basis functions for inter-grid interpolation and mesh motion in FSI problems», *Computer Methods in Applied Mechanics and Engineering*, vol. 256, p. 117–131. 29
- MACHIELS, L., A. T. PATERA, J. PERAIRE et Y. MADAY. 1998, «A general framework for finite element a posteriori error control : application to linear and nonlinear convection-dominated problems», dans *Proceedings of Institute for Computational Fluid Dynamics Conference, Oxford*. 59
- MADAY, Y., A. T. PATERA et D. V. ROVAS. 2002, «A blackbox reduced-basis output bound method for noncoercive linear problems», dans *Nonlinear partial differential equations and their applications : Collège de France seminar. Volume XIV*, Elsevier, p. 533–569. 59
- MARTINS, J. R. R. A., J. J. ALONSO et J. J. REUTHER. 2005, «A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design», *Optimization and Engineering*, vol. 6, n° 1, doi: 10.1023/B:OPTE.0000048536.47956.62, p. 33–62, ISSN 1573-2924. URL <https://doi.org/10.1023/B:OPTE.0000048536.47956.62>. 35
- MARTINS, J. R. R. A. et J. T. HWANG. 2013, «Review and unification of methods for computing derivatives of multidisciplinary computational models», *AIAA Journal*, vol. 51, n° 11, doi: 10.2514/1.J052184, p. 2582–2599. URL <https://doi.org/10.2514/1.J052184>. 35
- MARTINS, J. R. R. A. et A. LAMBE. 2013, «Multidisciplinary design optimization : A survey of architectures», *AIAA Journal*, vol. 51, doi: 10.2514/1.J051895, p. 2049–2075. 34
- MARTINS, J. R. R. A. et A. NING. 2021, *Engineering Design Optimization*, Cambridge University Press. 30, 34
- MARTINS, J. R. R. A., P. STURDZA et J. J. ALONSO. 2003, «The complex-step derivative approximation», *ACM Trans. Math. Softw.*, vol. 29, n° 3, doi: 10.1145/838250.838251, p. 245–262, ISSN 0098-3500. URL <https://doi.org/10.1145/838250.838251>. 35
- MATHERON, G. 1963, «Principles of geostatistics», *Economic Geology*, vol. 58, n° 8, doi: 10.2113/gsecongeo.58.8.1246, p. 1246–1266, ISSN 0361-0128. URL <https://doi.org/10.2113/gsecongeo.58.8.1246>. 17
- MCCULLOCH, W. S. et W. PITTS. 1943, «A logical calculus of the ideas immanent in nervous activity», *The Bulletin of Mathematical Biophysics*, vol. 5, n° 4, doi: 10.1007/BF02478259, p. 115–133, ISSN 1522-9602. URL <https://doi.org/10.1007/BF02478259>. 17
- MCKAY, M., R. BECKMAN et W. CONOVER. 1979, «A comparison of three methods for selecting values of input variables in the analysis of output from a computer code», *Technometrics*, vol. 21, doi: 10.1080/00401706.1979.10489755, p. 239–245. 17, 40
- MOČKUS, J. 1975, «On bayesian methods for seeking the extremum», dans *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, édité par G. I. Marchuk, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-540-37497-8, p. 400–404. 22
- MYERS, R. H., D. C. MONTGOMERY, G. G. VINING et T. J. ROBINSON. 2012, *Generalized linear models : with applications in engineering and the sciences*, John Wiley & Sons. 17

- NARGIL, P., F. LOUF et P.-A. BOUCARD. 2014, «A proper generalized decomposition approach for low frequency dynamic problems», dans *The Twelfth International Conference on Computational Structures Technology*, doi: 10.4203/ccp.106.161. URL <https://hal.archives-ouvertes.fr/hal-01696439>. 52
- NEGRI, F., A. MANZONI et D. AMSALLEM. 2015, «Efficient model reduction of parametrized systems by matrix discrete empirical interpolation», *Journal of Computational Physics*, vol. 303, doi: 10.1016/j.jcp.2015.09.046, p. 431–454. 54
- NIEDERREITER, H. 1988, «Low-discrepancy and low-dispersion sequences», *Journal of number theory*, vol. 30, n° 1, p. 51–70. 17
- NOCEDAL, J. et Y.-X. YUAN. 1998, «Combining trust region and line search techniques», dans *Advances in nonlinear programming*, Springer, p. 153–175. 14
- NURTAJ HOSSAIN, M. et D. GHOSH. 2020, «Adaptive reduced order modeling for nonlinear dynamical systems through a new a posteriori error estimator : Application to uncertainty quantification», *International Journal for Numerical Methods in Engineering*, vol. 121, n° 15, doi: <https://doi.org/10.1002/nme.6365>, p. 3417–3441. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6365>. 55
- PAIVA, R. M., A. R. D. CARVALHO, C. CRAWFORD et A. SULEMAN. 2010, «Comparison of surrogate models in a multidisciplinary optimization framework for wing design», *AIAA Journal*, vol. 48, n° 5, doi: 10.2514/1.45790, p. 995–1006. URL <https://doi.org/10.2514/1.45790>. 38
- PICHENY, V., D. GINSBOURGER, O. ROUSTANT, R. T. HAFTKA et N.-H. KIM. 2010, «Adaptive Designs of Experiments for Accurate Approximation of a Target Region», *Journal of Mechanical Design*, vol. 132, n° 7, doi: 10.1115/1.4001873, ISSN 1050-0472. URL <https://doi.org/10.1115/1.4001873>, 071008. 18
- PICHENY, V., R. B. GRAMACY, S. WILD et S. L. DIGABEL. 2016, «Bayesian optimization under mixed constraints with a slack-variable augmented lagrangian», dans *Proceedings of the 30th International Conference on Neural Information Processing Systems*, p. 1443–1451. 23
- POWELL, M. J. 1970, «A new algorithm for unconstrained optimization», dans *Nonlinear programming*, Elsevier, p. 31–65. 14
- POWELL, M. J. D. 1994, «A direct search optimization method that models the objective and constraint functions by linear interpolation», dans *Advances in optimization and numerical analysis*, Springer, p. 51–67. 16, 46
- PRIEM, R., N. BARTOLI, Y. DIOUANE et A. SGUEGLIA. 2020, «Upper trust bound feasibility criterion for mixed constrained bayesian optimization with application to aircraft design», *Aerospace Science and Technology*, vol. 105, p. 105 980. 23
- RENDALL, T. C. et C. B. ALLEN. 2008, «Unified fluid–structure interpolation and mesh motion using radial basis functions», *International journal for numerical methods in engineering*, vol. 74, n° 10, p. 1519–1559. 29
- RIPEPI, M., M. J. VERVELD, N. KARCHER, T. FRANZ, M. ABU-ZURAYK, S. GÖRTZ et T. KIER. 2018, «Reduced-order models for aerodynamic applications, loads and MDO», *CEAS Aeronautical Journal*, vol. 9, n° 1, p. 171–193. 55, 63

- ROULEAU, L., J.-F. DEÛ et A. LEGAY. 2014, «Review of reduction methods based on modal projection for highly damped structures», dans *11th World Congress on Computational Mechanics, WCCM XI*, Barcelona, Spain. URL <https://hal.archives-ouvertes.fr/hal-03179154>. 52
- ROUSTANT, O., D. GINSBOURGER et Y. DEVILLE. 2012, «Dicekriging, diceoptim : Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization», *Journal of statistical software*, vol. 51, n° 1, p. 1–55. 18
- ROY, P. T., N. EL MOÇAYD, S. RICCI, J.-C. JOUHAUD, N. GOUTAL, M. DE LOZZO et M. C. ROCHOUX. 2018, «Comparison of polynomial chaos and Gaussian process surrogates for uncertainty quantification and correlation estimation of spatially distributed open-channel steady flows», *Stochastic environmental research and risk assessment*, vol. 32, n° 6, p. 1723–1741. 17, 24
- SASENA, M. J., P. PAPALAMBROS et P. GOOVAERTS. 2002, «Exploration of metamodeling sampling criteria for constrained global optimization», *Engineering optimization*, vol. 34, n° 3, p. 263–278. 11
- SCHOLTEN, W. et D. HARTL. 2021, «Uncoupled method for static aeroelastic analysis», *Journal of Fluids and Structures*, vol. 101, doi: <https://doi.org/10.1016/j.jfluidstructs.2021.103221>, p. 103 221, ISSN 0889-9746. URL <https://www.sciencedirect.com/science/article/pii/S0889974621000049>. 27, 28, 38
- SCHONLAU, M., W. J. WELCH et D. R. JONES. 1998, «Global versus local search in constrained optimization of computer models», *Lecture Notes-Monograph Series*, p. 11–25. 11
- SHANNO, D. F. 1970, «Conditioning of quasi-Newton methods for function minimization», *Mathematics of computation*, vol. 24, n° 111, p. 647–656. 13, 46
- SHI, R., L. LIU, T. LONG, Y. WU et G. G. WANG. 2018, «Multidisciplinary modeling and surrogate assisted optimization for satellite constellation systems», *Structural and Multidisciplinary Optimization*, vol. 58, n° 5, p. 2173–2188. 37
- SIROVICH, L. 1987, «Turbulence and the dynamics of coherent structures. i. coherent structures», *Quarterly of applied mathematics*, vol. 45, n° 3, p. 561–571. 56
- SJØLUND, J. et E. LUND. 2018, «Structural gradient based sizing optimization of wind turbine blades with fixed outer geometry», *Composite Structures*, vol. 203, doi: <https://doi.org/10.1016/j.compstruct.2018.07.031>, p. 725–739, ISSN 0263-8223. URL <https://www.sciencedirect.com/science/article/pii/S0263822318306883>. 16
- SMETANA, K., O. ZAHM et A. T. PATERA. 2019, «Randomized residual-based error estimators for parametrized equations», *SIAM Journal on Scientific Computing*, vol. 41, n° 2, doi: 10.1137/18M120364X, p. A900–A926. URL <https://hal.archives-ouvertes.fr/hal-01851462>. 58
- SOBIESZCZANSKI-SOBIESKI, J. 1988, *Optimization by decomposition : a step from hierarchic to non-hierarchic systems*, National Aeronautics and Space Administration, Langley Research Center. 35
- SOBIESZCZANSKI-SOBIESKI, J., J. S. AGTE et R. R. SANDUSKY. 2000, «Bilevel integrated system synthesis», *AIAA Journal*, vol. 38, n° 1, doi: 10.2514/2.937, p. 164–172. URL <https://doi.org/10.2514/2.937>. 35

- SOBOL', I. M. 1967, «On the distribution of points in a cube and the approximate evaluation of integrals», *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, vol. 7, n° 4, p. 784–802. 17
- SOILAHOU DINE, M., C. GOGU et C. BES. 2017, «Accelerated adaptive surrogate-based optimization through reduced-order modeling», *AIAA Journal*, vol. 55, n° 5, doi: 10.2514/1.J055252, p. 1681–1694. URL <https://hal.archives-ouvertes.fr/hal-01943030>. 65
- SORENSEN, D. 1982, «Newton's method with a model trust region modification», *SIAM Journal on Numerical Analysis*, vol. 19, p. 409–426. 14
- TEDFORD, N. P. et J. R. R. A. MARTINS. 2010, «Benchmarking multidisciplinary design optimization algorithms», *Optimization and Engineering*, vol. 11, n° 1, p. 159–183. 34
- TEZDUYAR, T. et Y. OSAWA. 2001, «Fluid–structure interactions of a parachute crossing the far wake of an aircraft», *Computer Methods in Applied Mechanics and Engineering*, vol. 191, n° 6, doi: [https://doi.org/10.1016/S0045-7825\(01\)00311-5](https://doi.org/10.1016/S0045-7825(01)00311-5), p. 717–726, ISSN 0045-7825. URL <https://www.sciencedirect.com/science/article/pii/S0045782501003115>, minisymposium on Methods for Flow Simulation and Modeling. 27
- TOSSE RAMS, S., L. F. P. ETMAN et J. E. ROODA. 2008, «Augmented lagrangian coordination for distributed optimal design in MDO», *International Journal for Numerical Methods in Engineering*, vol. 73, n° 13, doi: <https://doi.org/10.1002/nme.2158>, p. 1885–1910. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2158>. 35
- VAPNIK, V. N. 1999, «An overview of statistical learning theory», *IEEE transactions on neural networks*, vol. 10, n° 5, p. 988–999. 17
- VEROY, K., C. PRUD'HOMME, D. ROVAS et A. PATERA. 2003, «A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations», dans *16th AIAA Computational Fluid Dynamics Conference*, doi: 10.2514/6.2003-3847. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2003-3847>. 54, 57
- VILLEMONTAIX, J., E. VAZQUEZ et E. WALTER. 2009, «An informational approach to the global optimization of expensive-to-evaluate functions», *Journal of Global Optimization*, vol. 44, n° 4, p. 509–534. 41
- WANG, X., M. LI, Y. LIU, W. SUN, X. SONG et J. ZHANG. 2017, «Surrogate based multidisciplinary design optimization of lithium-ion battery thermal management system in electric vehicles», *Structural and Multidisciplinary Optimization*, vol. 56, n° 6, p. 1555–1570. 38
- WATSON, A. G. et R. J. BARNES. 1995, «Infill sampling criteria to locate extremes», *Mathematical Geology*, vol. 27, n° 5, doi: 10.1007/BF02093902, p. 589–608, ISSN 1573-8868. URL <https://doi.org/10.1007/BF02093902>. 23
- WILLIAMS, C. K. et C. E. RASMUSSEN. 2006, *Gaussian processes for machine learning*, vol. 2, MIT press Cambridge, MA. 17, 18, 19, 41

- WIRTZ, D., D. SORENSEN et B. HAASDONK. 2012, «A posteriori error estimation for DEIM reduced nonlinear dynamical systems», *SIAM Journal on Scientific Computing*, vol. 36, doi: 10.1137/120899042, p. A311–A338. 54, 55
- XIAO, D. 2019, «Error estimation of the parametric non-intrusive reduced order model using machine learning», *Computer Methods in Applied Mechanics and Engineering*, vol. 355, doi: <https://doi.org/10.1016/j.cma.2019.06.018>, p. 513–534, ISSN 0045-7825. URL <https://www.sciencedirect.com/science/article/pii/S0045782519303615>. 55
- XU, C.-Z., Z.-H. HAN, K.-S. ZHANG et W. SONG. 2018, «Surrogate-based optimization method applied to multidisciplinary design optimization architectures», dans *31st congress of the International Council Of The Aeronautical Sciences (ICAS 2018)*. 37
- YI, S.-I., J.-K. SHIN et G. PARK. 2008, «Comparison of MDO methods with mathematical examples», *Structural and Multidisciplinary Optimization*, vol. 35, n° 5, p. 391–402. 34
- YUE, Y. et K. MEERBERGEN. 2013, «Accelerating optimization of parametric linear systems by model order reduction», *SIAM Journal Optimization*, vol. 23, p. 1344–1370. 66, 67
- ZAHM, O., M. BILLAUD-FRIESS et A. NOUY. 2017, «Projection-based model order reduction methods for the estimation of vector-valued variables of interest», *SIAM Journal on Scientific Computing*, vol. 39, n° 4, p. A1647–A1674. 57, 59
- ZAHM, O. et A. NOUY. 2016, «Interpolation of inverse operators for preconditioning parameter-dependent equations», *SIAM Journal on Scientific Computing*, vol. 38, n° 2, p. A1044–A1074. 58
- ZAHR, M. J., D. AMSALLEM et C. FARHAT. «Construction of parametrically-robust CFD-based reduced-order models for PDE-constrained optimization», dans *21st AIAA Computational Fluid Dynamics Conference*, doi: 10.2514/6.2013-2845. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2013-2845>. 67
- ZAHR, M. J. et C. FARHAT. 2015, «Progressive construction of a parametric reduced-order model for PDE-constrained optimization», *International Journal for Numerical Methods in Engineering*, vol. 102, n° 5, p. 1111–1135. 66
- ZHANG, M., W. GOU, L. LI, F. YANG et Z. YUE. 2017, «Multidisciplinary design and multi-objective optimization on guide fins of twin-web disk using kriging surrogate model», *Structural and Multidisciplinary Optimization*, vol. 55, n° 1, doi: 10.1007/s00158-016-1488-0, p. 361–373, ISSN 1615-1488. URL <https://doi.org/10.1007/s00158-016-1488-0>. 38
- ZIMMERMANN, R. 2015, «On the condition number anomaly of Gaussian correlation matrices», *Linear Algebra and Its Applications*, vol. 466, p. 512–526. 20
- ZOU, X., M. CONTI, P. DÍEZ et F. AURICCHIO. 2018, «A nonintrusive proper generalized decomposition scheme with application in biomechanics», *International Journal for Numerical Methods in Engineering*, vol. 113, n° 2, doi: <https://doi.org/10.1002/nme.5610>, p. 230–251. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.5610>. 52

Chapitre 2

Optimisation monodisciplinaire combinée à la réduction d'ordre de modèle.

*« Tout ce que nous avons à décider,
c'est ce que nous devons faire du
temps qui nous est imparti. »*

JRR Tolkien

Sommaire

2.1 Introduction	85
2.1.1 Cadre	85
2.1.2 Concept général de la méthodologie proposée	86
2.1.3 Contributions	87
2.1.4 Organisation du chapitre	87
2.2 Approximation de la quantité d'intérêt	87
2.2.1 Approche primale	88
2.2.2 Approche duale	88
2.2.3 Correction Duale	89
2.2.4 Estimation de l'erreur issue de la correction duale	90
2.2.5 Bilan sur l'estimation de la quantité d'intérêt	90
2.3 Estimation de l'erreur commise	90
2.3.1 Analyse statistique de l'inégalité de Cauchy-Schwarz	91
2.3.2 Estimation de l'erreur du problème primal	93
2.3.2.1 Encadrement de l'erreur grâce au résidu	94
2.3.2.2 Lien entre l'encadrement et le conditionnement de la matrice	95
2.3.2.3 Définition d'un préconditionneur	95
2.3.3 Préconditionnement	96
2.3.3.1 Préconditionneur constant	96
2.3.3.2 Préconditionneur par interpolation d'opérateurs	96
2.3.3.3 Changement du calcul des coefficients de l'interpolation	99
2.3.4 Bilan sur l'estimation de l'erreur	100
2.4 Utilisation du modèle réduit dans le cadre de l'optimisation	101

2.4.1	Initialisation	101
2.4.2	Utilisation de la réduction d'ordre de modèle lors de l'optimisation	102
2.5	Application	106
2.5.1	Description du cas d'application	106
2.5.2	Analyse des différents préconditionneurs	107
2.5.3	Analyse de la stratégie d'optimisation	108
2.5.4	Optimisation sur différents cas test	111
2.5.4.1	Cas test 1D	111
2.5.4.2	Cas test 4D	113
2.5.4.3	Cas 12D	114
2.6	Bilan du Chapitre	116
2.7	Références	117

Objectifs du chapitre

- Utiliser la projection de Galerkin pour l'approximation d'une quantité d'intérêt scalaire.
- Estimer la borne de l'erreur due à la projection.
- Proposer une méthode couplant un algorithme d'optimisation bayésienne et la projection de Galerkin prenant en compte l'approximation de l'erreur.
- Appliquer cette méthode sur un cas de dimensionnement d'une aile d'avion.

2.1 Introduction

Après avoir passé en revue, dans le Chapitre 1, des techniques d'optimisation, de méta-modélisation et de réduction de modèles, nous proposons dans ce chapitre un couplage entre la réduction d'ordre de modèle et l'optimisation. Pour cela, nous nous éloignons de l'approche *Offline-Online* présentée dans la Section 1.4.2 pour nous inspirer de l'approche *on-the-fly*. Pour rappel, l'approche *Offline-Online* consiste à construire un modèle réduit précis du solveur disciplinaire via la réduction d'ordre de modèle. La phase de construction du modèle, nommée phase *Offline*, recherche les points de l'espace de design où le modèle réduit est le moins précis afin de réduire l'erreur commise sur l'ensemble du domaine de conception. Par la suite, lors de l'optimisation (phase *Online*), seul le modèle réduit est utilisé afin d'obtenir l'optimum du problème à faible coût. Cette approche requière donc de générer un modèle réduit précis sur l'ensemble du domaine de conception $\mathcal{X}$ ce qui peut conduire à une phase *Offline* très coûteuse. Ainsi, la méthodologie qui sera mise en oeuvre sera inspirée de l'approche *on-the-fly* détaillée dans la Section 1.4.3.1. Le cadre d'étude dans lequel nous nous plaçons est celui d'une quantité d'intérêt linéaire obtenue à partir d'un solveur linéaire comme introduit en Section 1.3.4. Ceci permet de tester les méthodes de MOR et d'estimer l'erreur commise par la MOR lors de l'optimisation. Nous proposons alors d'étudier l'intérêt qu'aurait un contrôle plus précis de l'erreur dans une approche *on-the-fly*. Ainsi, un premier apport sera d'améliorer l'estimation de l'erreur. Par la suite, nous proposerons un couplage avec l'optimisation prenant en compte l'estimation de l'erreur commise.

2.1.1 Cadre

Nous nous intéressons ici aux problèmes résolus par la méthode des éléments finis dans le cadre de la mécanique des milieux continus (cf Section 1.3.1). Ces problèmes s'écrivent sous la forme d'un système caractérisé par un paramètre $\mathbf{x} \in \mathcal{X}$:

$$\mathbf{K}(\mathbf{x})\mathbf{u}(\mathbf{x}) = \mathbf{F}(\mathbf{x}) \quad (2.1)$$

où $\mathbf{K}(\mathbf{x}) \in \mathbb{R}^{N \times N}$ est la matrice de rigidité du solide, $\mathbf{F}(\mathbf{x}) \in \mathbb{R}^N$ le vecteur des efforts extérieurs et la solution $\mathbf{u}(\mathbf{x}) \in \mathbb{R}^N$ est le vecteur des déplacements dépendant des paramètres $\mathbf{x}$ du problème. Les déplacements ainsi obtenus permettent de définir une fonction de

performance s :

$$s(\mathbf{x}) = \mathbf{L}(\mathbf{x})\mathbf{u}(\mathbf{x}) \quad (2.2)$$

où $\mathbf{L}(\mathbf{x})$ est un opérateur matriciel permettant d'extraire la quantité d'intérêt à partir de $\mathbf{u}(\mathbf{x})$ et est modélisé par un vecteur ligne. Le problème d'optimisation que nous cherchons à résoudre est :

$$\min_{\mathbf{x} \in \mathcal{X}} s(\mathbf{x}) \quad (2.3)$$

2.1.2 Concept général de la méthodologie proposée

Puisque des appels répétés à la fonction objectif s , et donc à la solution du système (2.1), seront nécessaires pour la résolution du problème d'optimisation, nous choisissons d'utiliser la MOR pour réduire le coût de calcul. Pour cela, nous proposons d'évaluer la précision du modèle réduit lors de l'optimisation (phase *Online*). La Figure 2.1 illustre les stratégies avec ou sans prise en compte de la précision du modèle approché. Dans le cas où la MOR n'est pas utilisée, chaque fois que l'algorithme nécessitera de faire appel à la fonction objectif pour un paramètre $\mathbf{x}$, la solution du système linéaire (2.1) sera calculée comme cela est illustré dans la Figure 2.1a. La stratégie que nous proposons repose sur une estimation de l'erreur commise lors de la phase d'optimisation, ou phase *Online* permettant de déterminer si le modèle réduit est suffisamment précis. Cette stratégie est illustrée dans la Figure 2.1b.

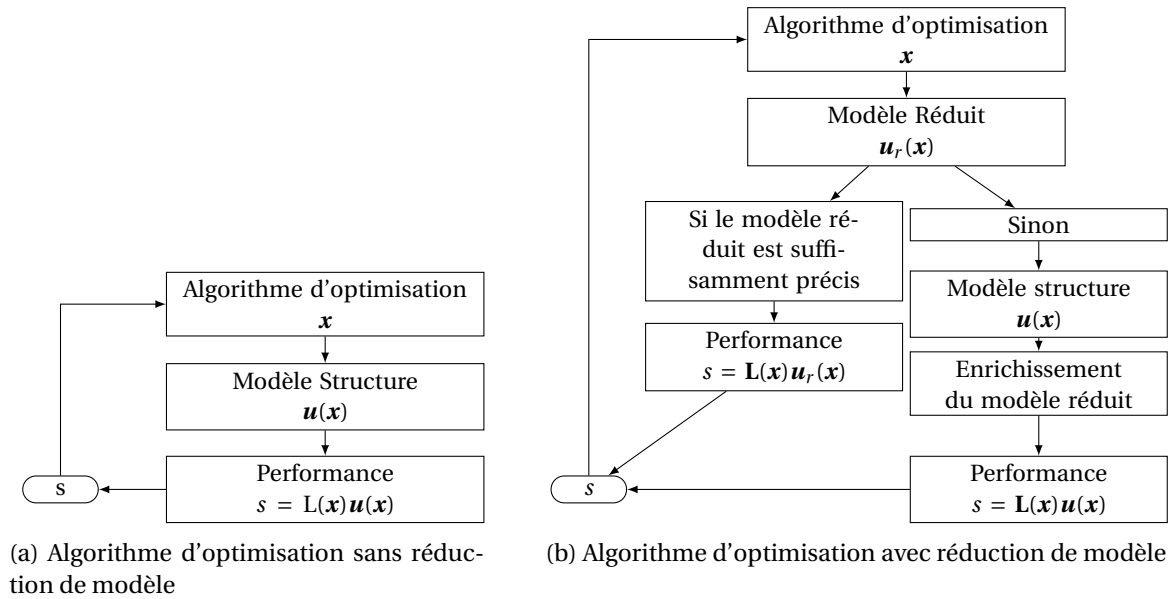


FIGURE 2.1 – Algorithme d'optimisation avec ou sans réduction de modèle

Nous supposons que la décomposition affine exacte des éléments n'est pas disponible sans être intrusif au sein du solveur et que son approximation présentée en Section 1.3.2.2 introduirait une erreur trop difficile à estimer pour être utilisée dans nos travaux.

L'idée principale de notre étude est d'utiliser l'estimation de l'erreur sur la quantité d'intérêt pour certifier les approximations obtenues par la MOR comme cela a été proposé dans la méthode *on-the-fly*. Cependant, dans notre étude, nous ne souhaitons pas faire d'approximation sur le résultat du problème (2.3). Nous cherchons à utiliser la MOR pour assister la recherche du paramètre optimal mais le résultat de l'optimisation ne devra pas être une approximation. Pour un algorithme d'optimisation de type EGO, cela revient à utiliser la MOR lors des phases d'exploration et à utiliser le modèle complet lors de

l'exploitation du modèle. Seulement, il est difficile d'estimer clairement ces phases, nous proposerons donc un critère basé sur l'optimalité des points proposés par EGO que nous allons définir par la suite. Pour rappel, l'algorithme EGO ajoute des points au DoE initial, ces points sont obtenus par maximisation d'une fonction d'acquisition. Le minimum courant est alors défini par le minimum de la fonction objectif sur le DoE. Nous cherchons alors à estimer l'optimalité du nouveau point, c'est-à-dire si le point proposé par EGO permet d'améliorer le minimum courant, grâce à l'information fournie par le modèle réduit. Cette estimation de l'optimalité du nouveau point sera obtenue grâce à l'approximation de la QI et à l'approximation de l'erreur commise sur cette même quantité permettant de définir un intervalle de confiance sur la QI. Dans le cas où la non optimalité est certifiée par le modèle réduit, le modèle réduit est utilisé pour estimer la QI. Sinon, le modèle complet est utilisé et le modèle réduit est enrichi. Ceci permet de d'assurer que le minimum courant est toujours donné par le modèle complet.

2.1.3 Contributions

Dans notre étude, nous proposons une amélioration de l'estimation de l'erreur commise par la MOR. En effet, nous allons voir comment la borne de l'erreur proposée dans la littérature (DAHMEN et collab. [2014]) a tendance à surestimer l'erreur commise par l'utilisation de l'inégalité de Cauchy-Schwarz. Nous proposons alors une analyse statistique de cette inégalité prenant en compte la grande dimension des vecteurs considérés. De plus, nous proposons d'adapter le préconditionnement par interpolation d'opérateurs proposé par ZAHM et NOUY [2016] au cadre *on-the-fly* sans décomposition affine des éléments. Finalement, nous adapterons la méthode *on-the-fly* dans le cas particulier où un algorithme d'optimisation de type EGO est utilisé en exploitant l'estimation de l'erreur obtenue.

2.1.4 Organisation du chapitre

Dans un premier temps, la réduction d'ordre de modèle pour une quantité d'intérêt linéaire sera détaillée. À cette fin, nous présenterons le problème dual permettant de définir une correction sur la quantité d'intérêt. Nous définirons par la suite une estimation de l'erreur commise sur cette approximation corrigée. Cette erreur sera améliorée par le préconditionnement du problème ainsi que par la considération d'une borne de l'erreur statistique. Enfin, un couplage avec l'algorithme d'optimisation EGO sera proposé. Finalement, les approches que nous aurons présentées seront testées sur un exemple aéro-structure.

2.2 Approximation de la quantité d'intérêt

De nombreuses approximations sont disponibles sur des quantités d'intérêt linéaires (JANON et collab. [2016]; ZAHM et collab. [2017]). Bien que la méthode que nous allons détailler permette d'approcher un ensemble de quantités d'intérêt ($s(\mathbf{x})$ serait alors un vecteur), nous allons traiter le cas d'une quantité d'intérêt scalaire afin de simplifier les notations.

2.2.1 Approche primale

La première solution pour obtenir une approximation du problème (2.2) est de générer une approximation du problème primal (Eq. (2.1)) $\mathbf{u}_r(\mathbf{x})$ et de l'utiliser pour obtenir une approximation de $s(\mathbf{x})$:

$$s(\mathbf{x}) \approx s_r^P(\mathbf{x}) = \mathbf{L}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) \quad (2.4)$$

Pour définir une approximation du problème primal, une base réduite $\mathbf{B}_r^P$ est construite par **POD**. Ainsi, un échantillon de l'espace paramétrique $\mathcal{X}$ est généré par **LHS** : $\mathbf{x}_1, \dots, \mathbf{x}_m$ où m est la taille de l'échantillon. Pour chacun de ces paramètres, la solution du système linéaire (2.1) est calculée : $\mathbf{u}(\mathbf{x}_1), \dots, \mathbf{u}(\mathbf{x}_m)$. Ces *snapshots* obtenus sont alors utilisés pour construire une matrice de projection Φ^P et une moyenne Φ_0^P par **POD** (cf Section 1.3.3.1). Pour rappel, dans le cas de la projection de Galerkin, la solution approchée est définie par la résolution du système projeté :

$$\mathbf{K}_r^P(\mathbf{x})\boldsymbol{\alpha}^P(\mathbf{x}) = \mathbf{F}_r^P(\mathbf{x}) \quad (2.5)$$

où $\mathbf{K}_r^P(\mathbf{x})$ et $\mathbf{F}_r^P(\mathbf{x})$ sont les grandeurs du système projeté :

$$\begin{aligned} \mathbf{K}_r^P(\mathbf{x}) &= (\Phi^P)^T \mathbf{K}(\mathbf{x}) \Phi \\ \mathbf{F}_r^P(\mathbf{x}) &= (\Phi^P)^T (\mathbf{F}(\mathbf{x}) - \mathbf{K}(\mathbf{x})\Phi_0^P) \end{aligned} \quad (2.6)$$

$\mathbf{u}_r(\mathbf{x})$ est alors obtenu à partir des coordonnées sur la base réduite $\boldsymbol{\alpha}^P(\mathbf{x})$:

$$\mathbf{u}_r(\mathbf{x}) = \Phi_0 + \sum_{i=1}^{n_P} \alpha_i^P(\mathbf{x}) \Phi_i^P \quad (2.7)$$

où Φ_i^P , $i = 1, \dots, n_P$ sont les vecteurs de la matrice de projection Φ^P .

En posant, $\beta(\mathbf{x}) = \sup_{\mathbf{v} \in V \setminus \{0\}} \frac{|\mathbf{L}(\mathbf{x})\mathbf{v}|}{\|\mathbf{v}\|_2}$ on obtient une borne de l'erreur de la **QI** :

$$|s(\mathbf{x}) - s_r^P(\mathbf{x})| \leq \beta(\mathbf{x}) \|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\|_2 \quad (2.8)$$

Notons que l'ensemble $V = \{\mathbf{u}(\mathbf{x}), \mathbf{x} \in \mathcal{X}\} \subset \mathbb{R}^N$ est généralement inconnu conduisant à majorer $\beta(\mathbf{x})$ par $\|\mathbf{L}(\mathbf{x})\|_2$. La borne de l'erreur est alors :

$$|s(\mathbf{x}) - s_r^P(\mathbf{x})| \leq \|\mathbf{L}(\mathbf{x})\|_2 \|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\|_2 \quad (2.9)$$

Néanmoins, en considérant le problème dual que nous allons détailler, il est possible d'affiner cette approximation.

2.2.2 Approche duale

Une autre solution pour résoudre le système (2.2) peut être définie par l'introduction du problème dual :

$$\mathbf{K}(\mathbf{x})\mathbf{Q}(\mathbf{x}) = \mathbf{L}^T(\mathbf{x}) \quad (2.10)$$

où $\mathbf{Q}(\mathbf{x})$ est la solution du problème dual. La **QI** peut être obtenue à partir de cette solution :

$$s(\mathbf{x}) = \mathbf{Q}^T(\mathbf{x})\mathbf{F}(\mathbf{x}) \quad (2.11)$$

La résolution du système (2.10) peut être approchée, de la même façon que le système (1.64), par des approches **MOR** via l'utilisation d'une nouvelle base d'approximation $\mathbf{B}_r^D$. La base d'approximation sera définie par une translation Φ_0^D et par un espace vectoriel

V_r^D généré par la matrice de projection Φ^D . Ceci permet alors de définir une approximation $Q_r(x)$ de $Q(x)$ sur la base réduite B_r^D . L'approximation de la QI par l'approximation duale $Q_r^T(x)$ est donnée par :

$$s_r^Q(x) = Q_r^T(x)F(x) \quad (2.12)$$

Pour obtenir la base réduite duale B_r^D , la même méthodologie que pour l'approche primale est utilisée. À partir du même échantillon $x_1, \dots, x_m$, les solutions du problème dual sont calculées : $Q(x_1), \dots, Q(x_m)$. Ces snapshots permettent le calcul de la base réduite par POD. Notons que si la factorisation de la matrice $K(x_i)$, $i = 1, \dots, m$ a déjà été calculée, l'obtention de la solution duale via le système linéaire (2.10) est peu coûteuse.

2.2.3 Correction Duale

La solution de l'approximation primale-duale (MACHIELS et collab. [1998]; MADAY et collab. [2002]; CUONG et collab. [2005]; JANON et collab. [2016]; ZAHM et collab. [2017]) est définie par :

$$s_r^{PD}(x) = L(x)u_r(x) - Q_r^T(x)(K(x)u_r(x) - F(x)) \quad (2.13)$$

L'approximation primale est alors corrigée en utilisant le résidu du problème à la place de $L(x)$ dans le problème dual. Nous rappelons que l'avantage de cette méthode est lié au fait qu'il est plus performant de corriger un modèle MOR par un autre.

Les différences de performance sont illustrées sur un calcul VLM (KATZ et PLOTKIN [2001]) paramétré sur une aile d'avion où la quantité d'intérêt est la moyenne des vitesses sur le bord d'attaque. La VLM est basée sur la théorie des flux potentiels où les effets de viscosité, de turbulence, de dissipation, de couche limite sont négligés et sera détaillée dans le Chapitre 3. Le nombre de vecteurs conservés pour l'approche primale est de 6, le nombre de vecteurs conservé pour l'approche duale est 6, le nombre de vecteurs conservés pour l'approche primale-duale est de 3 pour chaque base. Les histogrammes lissés des erreurs relatives pour chacune des approches sont représentés sur la Figure 2.2 avec une échelle logarithmique. On remarque que l'approche primale-duale permet de diminuer de quasiment deux ordres de grandeur l'erreur commise.

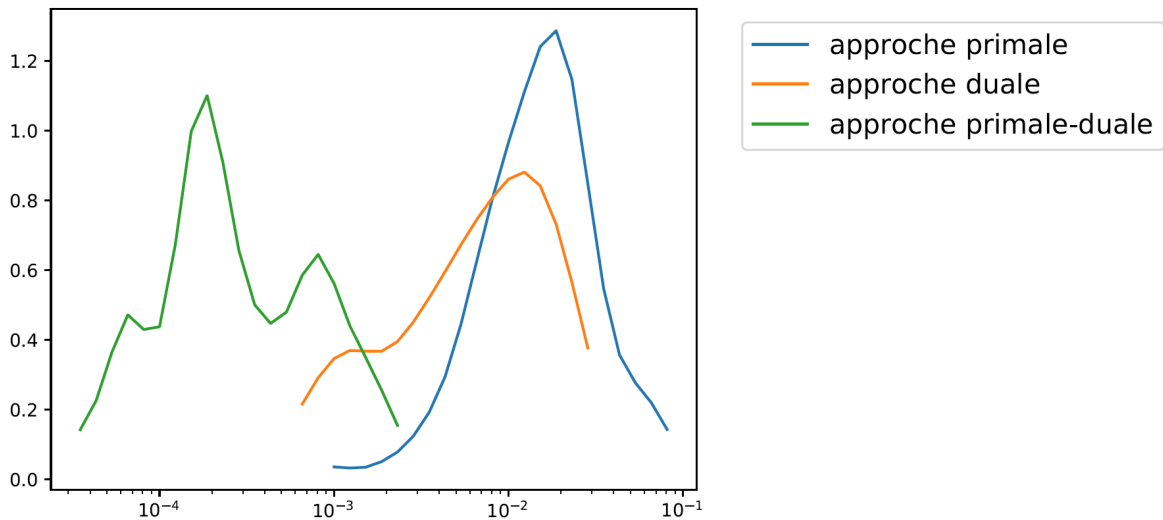


FIGURE 2.2 – Histogrammes lissés de l'erreur relative pour l'approche primale, duale et primale-duale sur une échelle logarithmique obtenue sur un problème VLM paramétré.

Finalement, grâce à la correction duale, une approximation de la QI linéaire peut être

obtenue. Nous allons à présent nous intéresser à l'estimation de l'erreur sur la **QI** avec la correction duale.

2.2.4 Estimation de l'erreur issue de la correction duale

Dans le cas d'une correction duale, une estimation de l'erreur commise sur la quantité d'intérêt peut être définie à partir des équations suivantes :

$$\begin{aligned} s(\mathbf{x}) - s_r^{\text{PD}}(\mathbf{x}) &= \mathbf{L}(\mathbf{x})\mathbf{u}(\mathbf{x}) - \mathbf{L}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) + \mathbf{Q}_r^{\text{T}}(\mathbf{x}) (\mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})) \\ &= \mathbf{L}(\mathbf{x}) (\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})) - \mathbf{Q}_r^{\text{T}}(\mathbf{x})\mathbf{K}(\mathbf{x}) (\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})) \\ &= (\mathbf{L}(\mathbf{x}) - \mathbf{Q}_r^{\text{T}}(\mathbf{x})\mathbf{K}(\mathbf{x})) (\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})) \end{aligned} \quad (2.14)$$

On remarque alors que l'erreur est donnée par le produit scalaire entre deux quantités : le résidu du problème dual $\mathbf{L}(\mathbf{x}) - \mathbf{Q}_r^{\text{T}}(\mathbf{x})\mathbf{K}(\mathbf{x})$ et l'erreur du problème primal $\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})$. Le résidu du problème dual est facile à estimer car cela ne requiert pas d'inverser la matrice de rigidité $\mathbf{K}(\mathbf{x})$, néanmoins, l'erreur du problème primal est inconnue. Dans la littérature, il est proposé d'utiliser l'inégalité de Cauchy-Schwarz afin de définir une borne de l'erreur :

$$|s(\mathbf{x}) - s_r^{\text{PD}}(\mathbf{x})| \leq \|\mathbf{L}(\mathbf{x}) - \mathbf{Q}_r^{\text{T}}(\mathbf{x})\mathbf{K}(\mathbf{x})\|_2 \|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\|_2 = \Delta_1(\mathbf{x}) \quad (2.15)$$

Nous remarquons que la borne sur l'erreur ainsi définie dépend de l'erreur du problème primal $\|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\|$. Il sera donc nécessaire de définir une approximation de cette erreur par la suite.

2.2.5 Bilan sur l'estimation de la quantité d'intérêt

Finalement, des méthodes basées sur l'approche primale-duale permettent d'obtenir une approximation de la quantité d'intérêt plus fine qu'en considérant uniquement le problème primal dans le cas linéaire. Notons que pour être efficace cette méthode nécessite la résolution d'un problème dual sur l'opérateur $\mathbf{L}(\mathbf{x})$ défini par l'Eq. (2.10). Cette solution du problème dual est peu coûteuse à obtenir si une factorisation de la matrice de rigidité a préalablement été calculée. Sous ces hypothèses, la correction duale permet de diminuer l'erreur commise sur la quantité d'intérêt à taille de base équivalente. Enfin, nous avons mis en évidence un moyen de majorer l'erreur commise par la **MOR**. Néanmoins, cette estimation de l'erreur est peu précise et dépend de l'erreur du problème primal $\|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\|$ qui est inconnue. Nous allons donc voir dans la suite comment nous pouvons estimer et affiner cette erreur.

2.3 Estimation de l'erreur commise

L'objectif de cette partie est d'estimer l'erreur commise par la réduction d'ordre de modèle à savoir $|s(\mathbf{x}) - s_r^{\text{PD}}(\mathbf{x})|$. Or, la borne de l'erreur définie dans l'Eq. (2.15) a deux principaux inconvénients :

1. L'erreur sur la quantité d'intérêt est obtenue en utilisant l'inégalité de Cauchy-Schwarz sur le produit scalaire. Or, ceci peut engendrer une surestimation de l'erreur commise. Nous proposons donc une analyse statistique pour réduire cette surestimation.

2. L'erreur du problème primal $\|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\|$ n'est pas connue. Pour l'estimer, nous proposons d'utiliser le résidu : $\mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})$ comme cela est proposé dans [JANON et collab. \[2016\]](#) ; [ZAHM et collab. \[2017\]](#). Néanmoins, lorsque la matrice de rigidité est mal conditionnée, la corrélation entre le résidu et l'erreur commise est faible. Pour y remédier, nous proposons d'utiliser un préconditionnement du problème dual basé sur l'interpolation d'opérateurs proposée par [ZAHM et collab. \[2017\]](#). Une adaptation de ce préconditionnement au cadre de notre étude est proposée par la suite.

2.3.1 Analyse statistique de l'inégalité de Cauchy-Schwarz

On remarque que la valeur absolue de l'erreur définie dans l'Eq. (2.14) s'écrit :

$$|s(\mathbf{x}) - s_r^{\text{PD}}(\mathbf{x})| = |(\mathbf{L}(\mathbf{x}) - \mathbf{Q}_r^T(\mathbf{x})\mathbf{K}(\mathbf{x}))(\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x}))| \quad (2.16)$$

$$= \|\mathbf{L}(\mathbf{x}) - \mathbf{Q}_r^T(\mathbf{x})\mathbf{K}(\mathbf{x})\| \|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\| |\cos(\iota(\mathbf{x}))| \quad (2.17)$$

où $\iota(\mathbf{x})$ est l'angle entre les deux vecteurs $\mathbf{L}^T(\mathbf{x}) - \mathbf{K}(\mathbf{x})\mathbf{Q}_r(\mathbf{x})$ et $\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})$.

Dans la littérature, la valeur du cosinus est majorée par 1. Cependant, si on considère deux vecteurs aléatoires $\mathbf{a}$ et $\mathbf{b}$ de dimension très grande, la probabilité que $\mathbf{a}$ et $\mathbf{b}$ soient alignés est d'autant plus faible que la dimension est élevée. Ainsi, la valeur absolue du cosinus de l'angle entre $\mathbf{a}$ et $\mathbf{b}$ converge en probabilité vers 0 lorsque la dimension des vecteurs tend vers l'infini (dans le cas où la direction des vecteurs est uniforme et que les directions des vecteurs sont indépendantes). Ce résultat sera démontré dans la suite. Pour cette raison, nous souhaitons estimer la distribution de ce cosinus afin de calculer un quantile du cosinus plutôt que de le majorer par 1. En pratique, nous souhaitons estimer le cosinus de deux quantités déterministes : $\mathbf{L}^T(\mathbf{x}) - \mathbf{K}(\mathbf{x})\mathbf{Q}_r(\mathbf{x})$ et $\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})$. Néanmoins, comme nous n'avons pas d'information sur l'erreur du problème primal, nous proposons une analyse statistique du cosinus.

On note F la fonction de répartition de $|\cos(\iota(\mathbf{x}))|$ dans $[0, 1]$. L'objectif est alors de déterminer un quantile de niveau α de la variable $|\cos(\iota(\mathbf{x}))| : \inf\{t \in \mathbb{R} : F(t) > 1 - \alpha\}$ que l'on notera $q_k(\alpha)$. Ceci permet alors d'écrire :

$$\mathbb{P} \left(\frac{|(\mathbf{L}(\mathbf{x}) - \mathbf{Q}_r^T(\mathbf{x})\mathbf{K}(\mathbf{x}))(\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x}))|}{\|\mathbf{L}(\mathbf{x}) - \mathbf{Q}_r^T(\mathbf{x})\mathbf{K}(\mathbf{x})\| \|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\|} \geq q_k(\alpha) \right) \leq \alpha \quad (2.18)$$

et donc de définir une borne d'erreur probabiliste $\Delta_\alpha(\mathbf{x})$ de niveau de confiance α calculée via :

$$\Delta_\alpha(\mathbf{x}) = \|\mathbf{L}(\mathbf{x}) - \mathbf{Q}_r^T(\mathbf{x})\mathbf{K}(\mathbf{x})\| \|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\| q_k(\alpha) \quad (2.19)$$

Lorsque la matrice $\mathbf{K}(\mathbf{x})$ est symétrique définie positive, $\mathbf{u}_r(\mathbf{x}) - \boldsymbol{\phi}_0$ est la projection orthogonale de $\mathbf{u}(\mathbf{x}) - \boldsymbol{\phi}_0$ sur V_r^P par rapport au produit scalaire défini par $\mathbf{K}(\mathbf{x})$. Le terme $\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})$ est donc à valeurs dans $(V_r^P)^\perp$. De même, le terme $\mathbf{Q}(\mathbf{x}) - \mathbf{Q}_r(\mathbf{x})$ est à valeurs dans $(V_r^Q)^\perp$, donc $\mathbf{L}^T(\mathbf{x}) - \mathbf{K}(\mathbf{x})\mathbf{Q}_r(\mathbf{x})$ est à valeurs dans un espace vectoriel W de dimension $N - \dim(V_r^Q)$ (W est l'image de $(V_r^Q)^\perp$ par la fonction $\mathbf{Q} \mapsto \mathbf{K}(\mathbf{x})\mathbf{Q}$). Comme ces deux quantités sont issues de calculs différents (le premier étant le résidu du problème dual et le second étant l'erreur faite sur le problème primal), on suppose qu'elles sont indépendantes. Notons que ce raisonnement n'est pas exact puisque les deux problèmes sont liés. Néanmoins, nous pouvons supposer que, comme il s'agit des erreurs commises sur les bases de projection (donc les modes [POD](#) les moins représentatifs du système), nous

pouvons appliquer cette hypothèse en première approximation. Comme aucune autre information n'est disponible sur la distribution du vecteur $\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})$, on supposera que sa direction est tirée uniformément sur la sphère unité.

Par rotation du repère, tirer uniformément le vecteur $\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})$ dans $(V_r^P)^\perp$ et estimer le quantile du cosinus entre $\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})$ et $\mathbf{L}^T(\mathbf{x}) - \mathbf{K}(\mathbf{x})\mathbf{Q}_r(\mathbf{x})$ revient à estimer le quantile de la distribution de $|\cos \iota|$, où ι est l'angle entre une direction fixe $\mathbf{u}_0$ et un vecteur uniformément distribué sur la sphère unité de dimension $\dim((V_r^P)^\perp \cap W)$. Notons que nous connaissons la dimension de $(V_r^P)^\perp$ et de W mais nous ne connaissons pas la dimension de $(V_r^P)^\perp \cap W$. Nous proposons alors simplement de minimiser sa dimension : $\dim((V_r^P)^\perp \cap W) \geq N - \dim(V_r^P) - \dim(V_r^Q) = k$.

Le calcul du quantile s'effectue à partir des coordonnées hypersphériques. Soit $(\mathbf{h}_1, \dots, \mathbf{h}_{k-1}, \mathbf{u}_0)$ une base orthogonale de $\mathbb{R}^k$ et $\rho, \theta_1, \dots, \theta_{k-1} \in \mathbb{R}^+ \times [0; \pi]^{k-2} \times [0; 2\pi]$ les coordonnées hypersphériques associées. Par construction, on a θ_1 qui correspond à l'angle entre $\mathbf{u}_0$ et un vecteur tiré uniformément sur la sphère unité, ainsi, estimer la distribution de θ_1 permettra de répondre à notre problématique. On cherche alors à obtenir la distribution des $\theta_1, \dots, \theta_{k-1}$, permettant d'obtenir une distribution uniforme sur la sphère, afin d'obtenir la distribution de ι . Soit $f(\theta_1, \dots, \theta_{k-1})$ la densité de probabilité du tirage uniforme sur la sphère unité. Soit S_k un élément de surface infinitésimal de la sphère unité, sa surface dS_k (obtenue en calculant la jacobienne de la transformation entre la base cartésienne et la base hypersphérique) est donnée par :

$$dS_k = \prod_{i=1}^{k-1} \sin^{k-1-i} \theta_i d\theta_i \quad (2.20)$$

Soit S la surface de la sphère unité, on a :

$$\mathbb{P}(x \in S_k) = f(\theta_1, \dots, \theta_{k-1}) d\theta_1 \dots d\theta_{k-1} = \frac{dS_k}{S} = \frac{\prod_{i=1}^{k-1} \sin^{k-1-i} \theta_i d\theta_i}{S} \quad (2.21)$$

Ainsi, il vient : $f(\theta_1, \dots, \theta_{k-1}) = \prod_{i=1}^{k-1} f_i(\theta_i)$, avec

$$f_i(\theta_i) = \frac{\sin^{k-1-i} \theta_i}{\int_0^\pi \sin^{k-1-i} x dx} \quad \text{pour } 1 \leq i \leq k-2 \quad (2.22)$$

et

$$f_{k-1}(\theta_{k-1}) = \frac{1}{2\pi}$$

On en déduit que les variables $(\theta_i)_{1 \leq i \leq k-1}$ sont indépendantes et l'équation (2.22) permet d'obtenir les lois de probabilité f_i de chacune des coordonnées θ_i . On peut alors numériquement estimer la distribution de $\iota = \theta_1$ permettant de calculer le quantile. On remarque qu'avec nos hypothèses, le résultat ne dépend que de la dimension de V_r^P et de V_r^Q . Les résultats numériques obtenus pour les quantiles d'ordre α , notés $q_k(\alpha)$, sont donnés sur la Figure 2.3 et permettent d'observer une décroissance en $\frac{1}{\sqrt{k}}$. Nous remarquons que dans un calcul linéaire avec 100 000 degrés de liberté, le quantile à 99.9% est d'environ 10^{-2} . Ainsi, ne pas prendre en compte le fait que la probabilité que les vecteurs soient alignés conduit à surestimer de deux ordres de grandeur l'estimation de l'erreur.

L'importance de cette correction de l'erreur est illustrée sur un cas structure à une variable de design qui sera développé dans la Section 2.5.1. L'erreur entre la quantité d'intérêt et son approximation $|s(x) - s_r^{PD}(x)|$ est illustrée sur la Figure 2.4. Sur cette figure, les

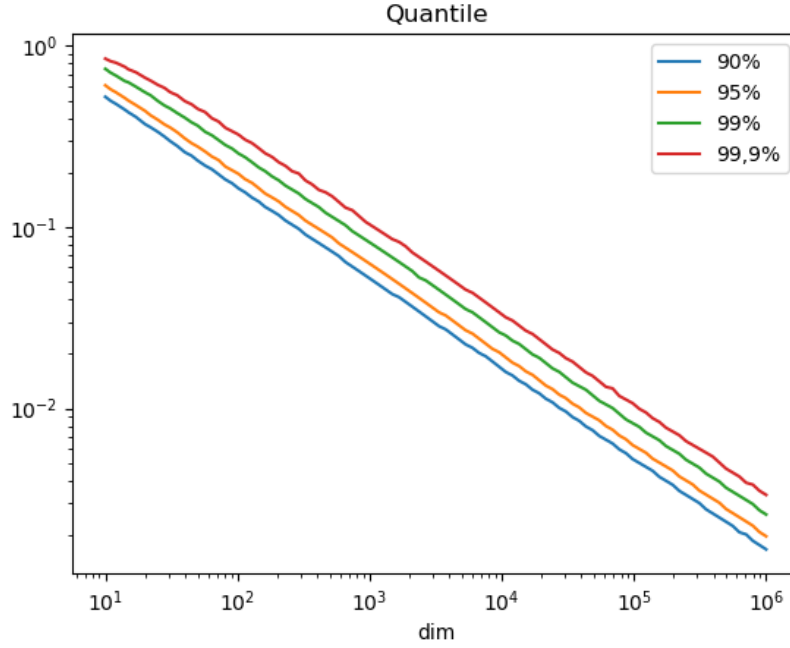


FIGURE 2.3 – Évolution du quantile de $|\cos t|$ à 90%, 95%, 99% et 99,9% en fonction de la dimension

bornes de l'erreur $\Delta_1(x)$ et $\Delta_{0.95}(x)$ sont illustrées. Notons que la borne de l'erreur, non probabiliste, $\Delta_1(x)$ est obtenue en choisissant $q_k(1) = 1$ dans l'Eq. (2.19). Cette illustration montre que, ne pas considérer la faible probabilité que les deux vecteurs de grande dimension intervenant dans le calcul de l'erreur soient parallèles, conduit à une surestimation de l'erreur commise.

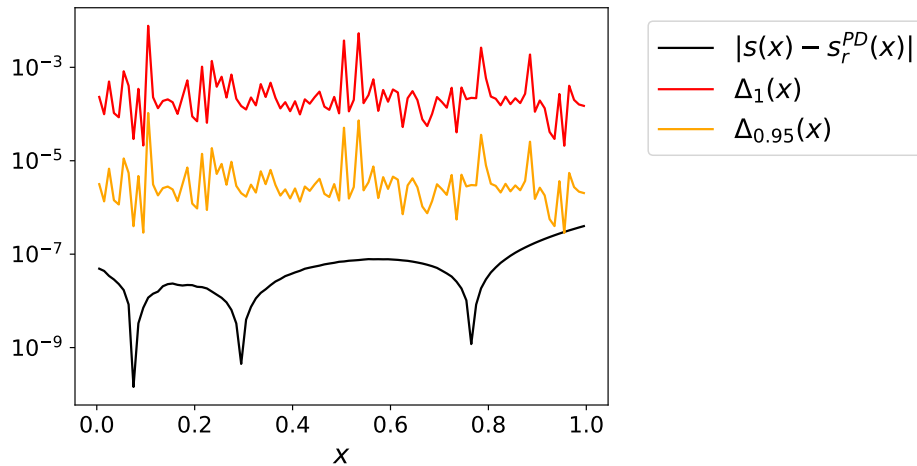


FIGURE 2.4 – Illustration de l'erreur commise sur la quantité d'intérêt $|s(x) - s_r^{PD}(x)|$ et la borne de l'erreur sur un problème à une dimension. $\Delta_1(x)$ est la borne obtenue sans considérer le quantile ($q_k(1) = 1$).

2.3.2 Estimation de l'erreur du problème primal

Dans cette partie, l'objectif est d'approcher le second terme de la majoration de l'erreur $\|\mathbf{u}(x) - \mathbf{u}_r(x)\|_2$ grâce à l'information peu coûteuse apportée par le résidu $\mathbf{K}(x)\mathbf{u}_r(x) -$

$\mathbf{F}(\mathbf{x})$. En effet, une corrélation existe entre ce résidu et la vraie erreur. Cependant, cette corrélation est détériorée si le conditionnement de la matrice $\mathbf{K}(\mathbf{x})$ n'est pas suffisamment bon. Ainsi, il est proposé de préconditionner le résidu afin d'améliorer l'estimation de l'erreur.

Si l'estimation de l'erreur doit être répétée plusieurs fois comme c'est le cas au cours de problèmes d'optimisation tel que celui que nous présentons dans la Section 2.4, alors il est nécessaire de réduire le plus possible le coût numérique de cette opération.

2.3.2.1 Encadrement de l'erreur grâce au résidu

Dans un premier temps, nous allons définir un encadrement de l'erreur grâce au résidu en nous basant sur des propriétés de la matrice $\mathbf{K}(\mathbf{x})$ et sur l'ensemble V . En effet, la matrice $\mathbf{K}(\mathbf{x})$ est symétrique et définie positive. Il existe donc $c(\mathbf{x}) > 0$ et $C(\mathbf{x})$ tels que :

$$c(\mathbf{x}) = \inf_{v \in V \setminus \{0\}} \frac{\langle \mathbf{K}(\mathbf{x}) v, v \rangle}{\|v\|_2^2} \quad (\text{définie positive}) \quad (2.23)$$

et

$$C(\mathbf{x}) = \sup_{v \in V \setminus \{0\}} \sup_{u \in V \setminus \{0\}} \frac{\langle \mathbf{K}(\mathbf{x}) v, u \rangle}{\|v\|_2 \|u\|_2} \quad (\text{opérateur linéaire donc continu}) \quad (2.24)$$

Proposition 1 *Il est possible à partir de ces deux quantités de définir un encadrement de l'erreur :*

$$\frac{1}{C(\mathbf{x})} \|\mathbf{K}(\mathbf{x}) \mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})\|_2 \leq \|\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})\|_2 \leq \frac{1}{c(\mathbf{x})} \|\mathbf{K}(\mathbf{x}) \mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})\|_2 \quad (2.25)$$

Démonstration:

En effet, en prenant $v = \mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})$ dans l'Eq. (2.23) :

$$\frac{\langle \mathbf{K}(\mathbf{x})(\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})), \mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x}) \rangle}{\|\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})\|_2^2} \geq c(\mathbf{x}) \quad (2.26)$$

Or, par l'application de l'inégalité de Cauchy-Schwarz, on obtient :

$$\langle \mathbf{K}(\mathbf{x})(\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})), \mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x}) \rangle \leq \|\mathbf{K}(\mathbf{x})(\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x}))\|_2 \|\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})\|_2 \quad (2.27)$$

ce qui permet donc d'écrire :

$$\begin{aligned} \frac{\langle \mathbf{K}(\mathbf{x})(\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})), \mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x}) \rangle}{\|\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})\|_2^2} &\geq c(\mathbf{x}) \\ \frac{\|\mathbf{K}(\mathbf{x})(\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x}))\|_2 \|\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})\|_2}{\|\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})\|_2^2} &\geq c(\mathbf{x}) \\ \frac{\|\mathbf{K}(\mathbf{x})(\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x}))\|_2}{\|\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})\|_2} &\geq c(\mathbf{x}) \\ \frac{1}{c(\mathbf{x})} \|\mathbf{K}(\mathbf{x}) \mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})\|_2 &\geq \|\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})\|_2 \end{aligned}$$

D'autre part, à partir de l'Eq. (2.24), en prenant $u = \mathbf{K}(\mathbf{x}) \mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})$ et $v = \mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})$,

on obtient :

$$\begin{aligned} \frac{\langle \mathbf{K}(\mathbf{x})(\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})), \mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x}) \rangle}{\|\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})\|_2 \|\mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})\|_2} &\leq C(\mathbf{x}) \\ \frac{\langle \mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x}), \mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x}) \rangle}{\|\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})\|_2 \|\mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})\|_2} &\leq C(\mathbf{x}) \\ \frac{\|\mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})\|_2}{\|\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})\|_2} &\leq C(\mathbf{x}) \\ \frac{1}{C(\mathbf{x})} \|\mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})\|_2 &\leq \|\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})\|_2 \end{aligned}$$

Ces deux inégalités permettent de définir l'encadrement de la norme de l'erreur $\|\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})\|_2$ proposé dans l'Eq. (2.25). ■

Cependant, nous avons remarqué qu'en pratique cet encadrement est d'un intérêt limité puisque les bornes de l'erreur ont plusieurs ordres de grandeur de différence avec la vraie erreur. Il est alors difficile d'utiliser ces bornes puisque l'erreur sera fortement surestimée. Cette observation peut être expliquée par le lien entre le conditionnement de la matrice de rigidité et l'encadrement de l'erreur.

2.3.2.2 Lien entre l'encadrement et le conditionnement de la matrice

Nous allons voir que le conditionnement et plus spécifiquement les valeurs propres extrêmes de la matrice $\mathbf{K}(\mathbf{x})$ ont une influence directe sur la qualité de l'encadrement défini dans la section précédente. En effet, le conditionnement κ de la matrice symétrique $\mathbf{K}(\mathbf{x})$ est :

$$\text{cond}(\mathbf{K}(\mathbf{x})) = \kappa = \frac{\lambda_{\max}(\mathbf{x})}{\lambda_{\min}(\mathbf{x})} \quad (2.28)$$

où $\lambda_{\max}(\mathbf{x})$ et $\lambda_{\min}(\mathbf{x})$ sont respectivement la plus grande et la plus petite valeur propre de $\mathbf{K}(\mathbf{x})$. De plus, de part la définition de l'Eq. (2.23) et de l'Eq. (2.24), on a :

$$\lambda_{\min}(\mathbf{x}) \leq c(\mathbf{x}) \quad (2.29)$$

et,

$$\lambda_{\max}(\mathbf{x}) \geq C(\mathbf{x}) \quad (2.30)$$

Ainsi, plus κ est proche de 1, plus l'encadrement de l'erreur sera fin et plus la norme du résidu sera un bon indicateur de la norme de l'erreur. Or, dans la plupart des problèmes de mécanique, le conditionnement de la matrice de rigidité est très élevé ($\kappa > 10^6$). L'idée est donc de proposer une solution permettant d'améliorer le conditionnement de la matrice $\mathbf{K}(\mathbf{x})$ en utilisant un préconditionneur.

Remarque :

Les inégalités dans les Eq. (2.29) et (2.30) sont dues au fait que $V \subset \mathbb{R}^N$.

2.3.2.3 Définition d'un préconditionneur

Un préconditionneur est une matrice $\mathbf{P}(\mathbf{x}) \in \mathbb{R}^{N \times N}$ permettant d'approcher l'inverse de la matrice de rigidité : $\mathbf{P}(\mathbf{x}) \approx \mathbf{K}^{-1}(\mathbf{x})$. De la même façon, il est possible de définir un encadrement de l'erreur en utilisant la matrice de rigidité préconditionnée :

$$\frac{1}{C'(\mathbf{x})} \|\mathbf{P}(\mathbf{x})(\mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x}))\|_2 \leq \|\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})\|_2 \leq \frac{1}{c'(\mathbf{x})} \|\mathbf{P}(\mathbf{x})(\mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x}))\|_2 \quad (2.31)$$

avec

$$C'(\mathbf{x}) = \sup_{v \in V \setminus \{0\}} \sup_{u \in V \setminus \{0\}} \frac{\langle \mathbf{P}(\mathbf{x}) \mathbf{K}(\mathbf{x}) v, u \rangle}{\|v\|_2 \|u\|_2} \text{ et } c'(\mathbf{x}) = \inf_{v \in V \setminus \{0\}} \frac{\langle \mathbf{P}(\mathbf{x}) \mathbf{K}(\mathbf{x}) v, v \rangle}{\|v\|_2^2} \quad (2.32)$$

Cependant, la matrice $\mathbf{P}(\mathbf{x}) \mathbf{K}(\mathbf{x})$ est supposée être proche de l'identité impliquant que $C'(\mathbf{x}) \approx 1$ et $c'(\mathbf{x}) \approx 1$. Dans ces conditions, le résidu conditionné est une bonne approximation de l'erreur :

$$\|\mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\|_2 \approx \|\mathbf{P}(\mathbf{x}) (\mathbf{K}(\mathbf{x}) \mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x}))\|_2 \quad (2.33)$$

Nous allons à présent introduire plusieurs préconditionnements de la littérature.

2.3.3 Préconditionnement

Plusieurs solutions ont été proposées dans la littérature pour définir un préconditionneur d'un problème linéaire. Dans notre étude, nous allons nous intéresser au cadre spécifique des préconditionneurs calculés sur un problème paramétré. En effet, l'une des caractéristiques principales de ce cadre est que des inverses des matrices $\mathbf{K}(\mathbf{x}_1), \dots, \mathbf{K}(\mathbf{x}_m)$ sont disponibles pour un DoE de paramètres utilisés pour la construction des bases de projection définies dans la Section 2.2.2. Notons que ces inverses ne sont pas disponibles explicitement mais sous la forme de factorisations. Cependant, nous supposons que calculer l'image par $\mathbf{K}^{-1}(\mathbf{x}_i)$, $i = 1, \dots, m$ d'un vecteur est peu coûteux même si nous nuancerons cette hypothèse dans la suite. Nous allons à présent détailler plusieurs méthodes qui exploitent l'information fournie par ces factorisations.

Remarques :

- 1 Nous ne prenons pas en compte les aspects de taille de mémoire vive dans notre étude : plusieurs factorisations de plusieurs matrices de rigidité peuvent être stockées simultanément. Néanmoins, le nombre de factorisations conservées dépend de la taille de la mémoire vive disponible ce qui peut être contraignant.
- 2 Cette méthode de préconditionnement empêche l'utilisation de méthodes itératives pour la résolution des systèmes linéaires ce qui peut être contraignant.

2.3.3.1 Préconditionneur constant

La première idée suppose que la matrice $\mathbf{K}(\mathbf{x})$ varie très peu selon les paramètres de design $\mathbf{x}$. Le préconditionnement est obtenu à partir de l'inverse de la matrice en un point particulier : $\mathbf{P}_0(\mathbf{x}) = \mathbf{K}^{-1}(\mathbf{x}_0)$ où $\mathbf{x}_0$ est le centre de l'espace de design $\mathcal{X}$ (GHANEM et KRUGER [1996]).

En pratique, la matrice de rigidité varie fortement en fonction des paramètres de design, une autre solution permet alors d'exploiter la connaissance des différentes factorisations des matrices $\mathbf{K}(\mathbf{x}_1), \dots, \mathbf{K}(\mathbf{x}_m)$ pour affiner le préconditionnement.

2.3.3.2 Préconditionneur par interpolation d'opérateurs

2.3.3.2.1 Interpolation d'opérateurs Le Préconditionnement par interpolation d'opérateurs (PIO), proposé par ZAHM et NOUY [2016], est une combinaison linéaire des inverses des matrices factorisées :

$$\mathbf{P}_I(\mathbf{x}, \boldsymbol{\lambda}) = \sum_{i=1}^m \lambda_i(\mathbf{x}) \mathbf{K}^{-1}(\mathbf{x}_i) = \sum_{i=1}^m \lambda_i(\mathbf{x}) \mathbf{P}^i \quad (2.34)$$

où $\boldsymbol{\lambda} = \{\lambda_i(\mathbf{x}), i = 1, \dots, m\}$ sont les coefficients de l'interpolation et en notant $\mathbf{P}^i$ l'inverse de $\mathbf{K}(\mathbf{x}_i)$. Plusieurs solutions ont alors été proposées pour déterminer ces coefficients.

2.3.3.2.2 Première solution La première solution proposée par GROTE et HUCKLE [1997] est obtenue en minimisant la quantité suivante :

$$\lambda_1^F(\mathbf{x}), \dots, \lambda_m^F(\mathbf{x}) = \arg \min_{\lambda_1, \dots, \lambda_m} \left\| \mathbf{I}_N - \sum_{i=1}^m \lambda_i(\mathbf{x}) \mathbf{P}^i \mathbf{K}(\mathbf{x}) \right\|_F \quad (2.35)$$

où $\|\cdot\|$ est la norme de Frobenius d'une matrice et $\mathbf{I}_N$ est la matrice identité de taille N .

Dans ce cas, la solution de ce problème est analytique : en posant $\boldsymbol{\lambda}^F(\mathbf{x}) = (\lambda_1^F(\mathbf{x}), \dots, \lambda_m^F(\mathbf{x}))$, les coordonnées sur les opérateurs sont données par :

$$\mathbf{M}(\mathbf{x}) \boldsymbol{\lambda}^F(\mathbf{x}) = \mathbf{b}(\mathbf{x}) \quad (2.36)$$

avec

$$\mathbf{M}_{i,j}(\mathbf{x}) = \text{trace}(\mathbf{K}(\mathbf{x}) \mathbf{P}^i \mathbf{P}^j \mathbf{K}(\mathbf{x})) \quad , \quad i = 1, \dots, m, \quad j = 1, \dots, m \quad (2.37)$$

et

$$\mathbf{b}_i(\mathbf{x}) = \text{trace}(\mathbf{P}^i \mathbf{K}(\mathbf{x})) \quad , \quad i = 1, \dots, m \quad (2.38)$$

Cependant, l'assemblage des données de ce système linéaire $\mathbf{M}$ et $\mathbf{b}$ est extrêmement coûteux. En effet, même si l'on suppose qu'obtenir $\mathbf{P}^i \mathbf{u}$ où $\mathbf{u} \in \mathbb{R}^N$ est peu coûteux, puisqu'une factorisation est disponible, les données de ce système requièrent de calculer la matrice $(\mathbf{P}^i \mathbf{K}(\mathbf{x}))$ ce qui revient à calculer l'inverse de N vecteurs pour chacune des m factorisations. ZAHM et NOUY [2016] proposent de simplifier le problème en projetant la matrice $\mathbf{K}(\mathbf{x})$ sur une matrice aléatoire.

Remarque :

Notons qu'un lien a été mis en évidence par GONZÁLEZ [2006] entre la norme de Frobenius et le conditionnement de la matrice $\mathbf{P}_1(\mathbf{x}) \mathbf{K}(\mathbf{x})$:

$$\text{cond}(\mathbf{P}_1(\mathbf{x}) \mathbf{K}(\mathbf{x})) \leq \frac{\sqrt{N - (N-1)(1 - \|\mathbf{I}_N - \mathbf{P}_1(\mathbf{x}) \mathbf{K}(\mathbf{x})\|_F)^2}}{1 - \|\mathbf{I}_N - \mathbf{P}_1(\mathbf{x}) \mathbf{K}(\mathbf{x})\|_F} \quad (2.39)$$

Ainsi, minimiser $\|\mathbf{I}_N - \mathbf{P}_1(\mathbf{x}) \mathbf{K}(\mathbf{x})\|_F$ permet de minimiser le conditionnement de la matrice $\mathbf{P}_1(\mathbf{x}) \mathbf{K}(\mathbf{x})$.

2.3.3.2.3 Deuxième solution La deuxième solution proposée par ZAHM et NOUY [2016] pour pallier ce problème est de multiplier la matrice $\mathbf{P}_1(\mathbf{x}) \mathbf{K}(\mathbf{x})$ à droite par une matrice aléatoire $\mathbf{Z} \in \mathbb{R}^{N \times K^Z}$, où K^Z est le nombre de colonnes retenues. Une approximation de la norme de Frobenius est alors définie :

$$\|(\mathbf{P}_1(\mathbf{x}) \mathbf{K}(\mathbf{x}))\|_F \approx \|(\mathbf{P}_1(\mathbf{x}) \mathbf{K}(\mathbf{x})) \mathbf{Z}\|_F \quad (2.40)$$

Cette approximation revient à considérer une semi-norme de Frobenius aléatoire que nous allons utiliser pour obtenir une approximation des coefficients $\boldsymbol{\lambda}^F(\mathbf{x})$ qui seront notés $\boldsymbol{\lambda}^Z(\mathbf{x})$ et sont solutions du système suivant :

$$\mathbf{M}^Z(\mathbf{x}) \boldsymbol{\lambda}^Z(\mathbf{x}) = \mathbf{b}^Z(\mathbf{x}) \quad (2.41)$$

où les données $\mathbf{M}^Z(\mathbf{x})$ et $\mathbf{b}^Z(\mathbf{x})$ sont obtenues par projection sur l'espace engendré par $\mathbf{Z}$:

$$\mathbf{M}_{i,j}^Z(\mathbf{x}) = \text{trace}(\mathbf{Z}^T \mathbf{K}(\mathbf{x}) \mathbf{P}^i \mathbf{P}^j \mathbf{K}(\mathbf{x}) \mathbf{Z}) \quad , \quad i = 1, \dots, m, \quad j = 1, \dots, m \quad (2.42)$$

et

$$\mathbf{b}_i^Z(\mathbf{x}) = \text{trace}(\mathbf{Z}^T \mathbf{P}^i \mathbf{K}(\mathbf{x}) \mathbf{Z}) \quad , \quad i = 1, \dots, m \quad (2.43)$$

Ainsi, pour l'obtention des données du système linéaire, il n'est plus nécessaire d'inverser la matrice $\mathbf{K}(\mathbf{x})$ mais uniquement sa projection sur la matrice aléatoire $\mathbf{K}(\mathbf{x}) \mathbf{Z}$. Ceci permet alors d'inverser K^Z vecteurs au lieu des N vecteurs du problème original.

2.3.3.2.4 Génération de $\mathbf{Z}$ À présent, nous allons détailler l'obtention de la matrice aléatoire $\mathbf{Z}$ ainsi que sa taille K^Z . Soit $\mathcal{D}$ la distribution permettant de générer $\mathbf{Z}$, $\mathcal{D}$ est choisi afin de respecter la (ϵ_Z, δ_Z) -concentration pour toute matrice $\mathbf{B} \in \mathbb{R}^{N \times N}$:

$$\mathbb{P}(\|\mathbf{B}\mathbf{Z}\|_F^2 - \|\mathbf{B}\|_F^2 \geq \epsilon_Z \|\mathbf{B}\|_F^2) \leq \delta_Z \quad (2.44)$$

où ϵ_Z et δ_Z sont des tolérances définies par l'utilisateur. Cette propriété est satisfaite en choisissant $K^Z \geq K_{lim}^Z(\epsilon_Z, \delta_Z)$. Plusieurs distributions $\mathcal{D}$ sont alors proposées par **ZAHM et NOUY [2016]** :

- la distribution de Rademacher (**AVRON et TOLEDO [2011]**) où $\mathbf{Z}$ est composé de variables indépendantes et identiquement distribuées avec $Z_{i,j} = \pm \frac{1}{\sqrt{K^Z}}$ avec une probabilité de $\frac{1}{2}$. Dans ce cas, le minimum de colonnes retenues est :

$$K_{lim}^Z(\epsilon_Z, \delta_Z) = 6\epsilon^{-2} \ln\left(\frac{2N}{\delta}\right) \quad (2.45)$$

- la distribution d'un sous-échantillon d'une matrice de Hadamard aléatoire (**BEKAS et collab. [2007]**) où $\mathbf{Z}$ est obtenu à partir d'une matrice de Hadamard. Une matrice de Hadamard $\mathbf{H}_N$ vérifie $\mathbf{H}_N^T \mathbf{H}_N = N\mathbf{I}_N$. $\mathbf{Z}$ est défini par :

$$\mathbf{Z} = \frac{1}{\sqrt{N}}(\mathbf{R}\mathbf{H}_N\mathbf{D})^T \quad (2.46)$$

où

- $\mathbf{D} \in \mathbb{R}^{N \times N}$ est une matrice diagonale dont les composantes sont des variables aléatoires indépendantes identiquement distribuées telles que $D_{i,i} = \pm 1$ avec une probabilité de $\frac{1}{2}$.
- $\mathbf{R} \in \mathbb{R}^{K \times N}$ est un sous ensemble de K colonnes de la matrice identité $\mathbf{I}_N$ choisi aléatoirement, sans remplacement.

Dans ce cas, le minimum de colonnes est :

$$K_{lim}^Z(\epsilon_Z, \delta_Z) = \frac{2}{\epsilon_Z^2 - \frac{\epsilon_Z^3}{3}} \ln\left(\frac{4}{\delta_Z}\right) \left(1 + \sqrt{8 \ln\left(\frac{4N}{\delta_Z}\right)}\right)^2 \quad (2.47)$$

Dans la suite, nous choisirons la distribution la plus avantageuse c'est-à-dire celle dont K_{lim}^Z sera le plus petit pour N , ϵ_Z , δ_Z donnés. En pratique, les sous-échantillons des matrices de Hadamard sont plus performants et permettent d'obtenir une taille K plus petite.

Remarque :

Dans le cas où $N = 15000$, $\epsilon = 0.1$ et $\delta = 0.2$, l'analyse probabiliste proposée par **ZAHM et NOUY [2016]** impose un K_{lim}^Z de plus de 6000 pour les deux distributions. Calculer l'inversion dans ce cas est donc toujours impossible car cela dépasserait le coût nécessaire à la factorisation d'une matrice. En pratique, nous proposons de rester à $K_{lim}^Z \leq 200$ afin de limiter le coût de la méthode. Notons que cette simplification a été étudiée par **ZAHM et NOUY [2016]**, qui ont conclu qu'une estimation des coefficients de l'interpolation à 6% près pouvait être obtenue avec $K = 200$.

L'apport de l'utilisation d'une interpolation est illustré sur le cas 1D sur la Figure 2.5. Sur cette figure, on peut observer la corrélation entre la norme du résidu $\|\mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})\|_2$ préconditionné et la norme de l'erreur $\|\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})\|_2$. Plusieurs préconditionneurs sont comparés : aucun ($\mathbf{P}(\mathbf{x}) = \mathbf{I}_N$), constant ($\mathbf{P}(\mathbf{x}) = \mathbf{P}_0$) et obtenu par interpolation d'opérateurs ($\mathbf{P}(\mathbf{x}) = \sum_{i=1}^m \lambda_i^Z(\mathbf{x}) \mathbf{K}^{-1}(\mathbf{x}_i)$). On remarque que sans préconditionneur, la

norme du résidu est bien supérieure à la norme de l'erreur. Le préconditionneur constant permet de limiter l'écart entre la vraie erreur et le résidu, cependant, la norme du résidu a tendance à sous-estimer la norme de l'erreur. On remarque que le [Préconditionnement par interpolation d'opérateurs \(PIO\)](#) permet d'estimer précisément l'inverse de la matrice $\mathbf{K}(\mathbf{x})$ conduisant à une corrélation quasiment exacte entre la norme du résidu et la norme de l'erreur. Ainsi, avec l'utilisation de ce type de préconditionneur, le résidu permet d'estimer l'erreur du problème primal commise. Dans la suite de la thèse, nous considérerons donc cette méthode pour le préconditionnement. Néanmoins, comme le calcul des coefficients peut se révéler trop important, nous allons étudier deux façons de simplifier leurs obtentions.

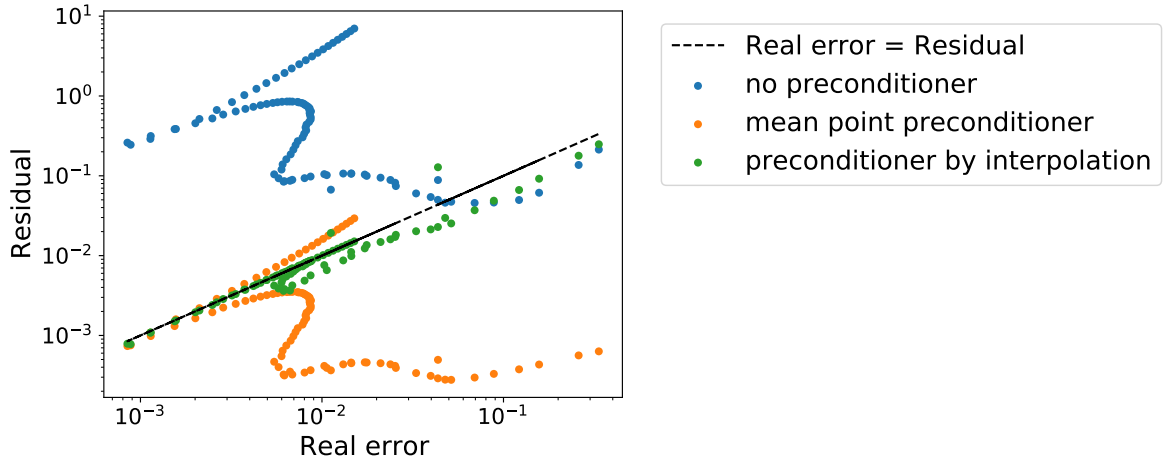


FIGURE 2.5 – Illustration du lien entre la norme de l'erreur $\|\mathbf{u}_r(\mathbf{x}) - \mathbf{u}(\mathbf{x})\|_2$ et la norme du résidu $\|\mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})\|_2$ préconditionné de 3 façons différentes : aucun, constant et obtenu par interpolation d'opérateurs. Ces calculs sont effectués sur le même cas 1D présenté dans la Section 2.3.1.

2.3.3.3 Changement du calcul des coefficients de l'interpolation

Dans un premier temps, nous avons décidé de changer la façon de calculer les coefficients du [PIO](#). En effet, bien que la projection sur la matrice aléatoire de dimension K permette de se limiter à mK^Z inversions, ce calcul reste malheureusement plus coûteux que l'obtention du vecteur $\mathbf{u}(\mathbf{x})$ dans certain cas. Dans la littérature, ces méthodes sont appliquées lorsqu'une décomposition affine des éléments est disponible. Dans ce cas,

$$\mathbf{P}^i \mathbf{K}(\mathbf{x}) \mathbf{Z} = \sum_{i=1}^{n_K} k_i(\mathbf{x}) \mathbf{P}^i \mathbf{K}_i \mathbf{Z} \quad (2.48)$$

et l'ensemble de ces opérations élémentaires peut être effectué lors de la phase *Offline* afin d'éviter ce calcul lors de la phase *Online*. Cependant, nous supposons dans le cadre de nos travaux que cette décomposition n'est pas disponible. Pour y remédier, nous proposons deux méthodes d'approximation des coefficients de l'interpolation basées sur les diagonales des matrices $\mathbf{K}(\mathbf{x}_i)$, $i = 1, \dots, m$. En effet, la diagonale d'une matrice, souvent utilisée comme préconditionneur dans la littérature est reconnue pour son efficacité. Nous allons alors exploiter la diagonale de la matrice de rigidité afin de déterminer les coefficients de la [PIO](#).

Dans un premier temps, nous proposons d'approximer le calcul des $\mathbf{P}^i \mathbf{K}(\mathbf{x}) \mathbf{Z}$:

$$\mathbf{P}^i \mathbf{K}(\mathbf{x}) \mathbf{Z} \approx (\mathbf{D}^i)^{-1} \mathbf{K}(\mathbf{x}) \mathbf{Z} \quad (2.49)$$

où $\mathbf{D}^i$ est une matrice diagonale dont la diagonale est celle de $\mathbf{K}(\mathbf{x}_i)$. Notons que l'inversion d'un système est remplacée par un simple produit avec une matrice diagonale. Les $\Lambda^D(\mathbf{x})$ de l'interpolation sont alors solutions du système :

$$\mathbf{M}^D(\mathbf{x})\boldsymbol{\Lambda}^D(\mathbf{x}) = \mathbf{b}^D(\mathbf{x}) \quad (2.50)$$

avec

$$\mathbf{M}_{i,j}^D(\mathbf{x}) = \text{trace}(\mathbf{Z}^T \mathbf{K}(\mathbf{x}) (\mathbf{D}^i)^{-1} (\mathbf{D}^j)^{-1} \mathbf{K}(\mathbf{x}) \mathbf{Z}) \quad , \quad i = 1, \dots, m, \quad j = 1, \dots, m \quad (2.51)$$

et

$$\mathbf{b}_i^D(\mathbf{x}) = \text{trace}(\mathbf{Z}^T (\mathbf{D}^i)^{-1} \mathbf{K}(\mathbf{x}) \mathbf{Z}) \quad , \quad i = 1, \dots, m \quad (2.52)$$

Le principale avantage de cette méthode est que le calcul des coefficients de l'interpolation ne requiert plus d'inverser de système à partir des factorisations puisque celles-ci sont remplacées par les diagonales inversées des matrices. Pour obtenir le résidu conditionné, seules m inversions sont nécessaires pour le calcul de $\mathbf{P}(\mathbf{x})(\mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x}))$.

Dans un second temps, nous décidons de nous éloigner des travaux de ZAHM et NOUY [2016] en exploitant le lien entre les matrices de rigidité et leurs diagonales. En effet, dans la plupart de nos problèmes, les paramètres ont une forte influence sur les diagonales des matrices de rigidité. Ainsi, déterminer les coefficients à partir des diagonales des matrices de rigidité semble approprié. Les coefficients seront alors déterminés par la meilleure représentation de la diagonale de la matrice $\mathbf{K}(\mathbf{x})$ sur les diagonales des matrices $\mathbf{K}(\mathbf{x}_i)$, $i = 1, \dots, m$. Nous cherchons ainsi les coefficients solutions du problème suivant :

$$\lambda_1^I(\mathbf{x}), \dots, \lambda_m^I(\mathbf{x}) = \arg \min_{\lambda_1, \dots, \lambda_m} \left\| \mathbf{d}(\mathbf{x}) - \sum_{i=1}^m \lambda_i \mathbf{d}^i \right\| \quad (2.53)$$

où $\mathbf{d}(\mathbf{x}) \in \mathbb{R}^N$ est le vecteur contenant la diagonale de $\mathbf{K}(\mathbf{x})$ et $\mathbf{d}^i$ est le vecteur contenant la diagonale de $\mathbf{K}(\mathbf{x}_i)$. La solution de ce système est analytique et solution du système :

$$\mathbf{M}^I(\mathbf{x})\boldsymbol{\Lambda}^I(\mathbf{x}) = \mathbf{b}^I(\mathbf{x}) \quad (2.54)$$

avec

$$\mathbf{M}_{i,j}^I(\mathbf{x}) = \langle \mathbf{d}^i, \mathbf{d}^j \rangle \quad , \quad i = 1, \dots, m, \quad j = 1, \dots, m \quad (2.55)$$

et

$$\mathbf{b}_i^I(\mathbf{x}) = \langle \mathbf{d}^i, \mathbf{d}(\mathbf{x}) \rangle \quad , \quad i = 1, \dots, m \quad (2.56)$$

Avec cette méthode, nous supposons que les systèmes (2.36) et (2.54) sont similaires et que les solutions sont proches. Ce lien, difficile à mettre en évidence, repose principalement sur le fait que la diagonale de la matrice de rigidité représente en grande partie la matrice elle-même. En effet, dans le cas de la matrice de rigidité d'un problème structure, la matrice de rigidité est à diagonale quasiment dominante (elle est à diagonale dominante sans prendre en compte les conditions aux limites du problème). Nous souhaitons donc exploiter la diagonale plutôt que le produit avec une matrice aléatoire comme cela est proposé par ZAHM et NOUY [2016]. Cette méthode sera analysée numériquement par la suite.

2.3.4 Bilan sur l'estimation de l'erreur

Dans le cadre de nos travaux, nous avons essayé de développer une estimation de l'erreur basée sur une analyse statistique du cosinus entre des vecteurs aléatoires ainsi

que sur l'estimation de l'erreur primale par préconditionnement du résidu. Ceci conduit à la borne de l'erreur probabiliste :

$$\mathbb{P}(|s(\mathbf{x}) - s_r^{\text{PD}}(\mathbf{x})| \leq \Delta_\alpha(\mathbf{x})) \leq \alpha \quad (2.57)$$

où la borne d'erreur $\Delta_\alpha(\mathbf{x})$ de niveau de confiance α est estimée à partir du résidu préconditionné :

$$\Delta_\alpha(\mathbf{x}) \approx \|\mathbf{L}(\mathbf{x}) - \mathbf{Q}_r^T(\mathbf{x})\mathbf{K}(\mathbf{x})\| \|\mathbf{P}(\mathbf{x})(\mathbf{K}(\mathbf{x})\mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x}))\| q_k(\alpha) \quad (2.58)$$

Nous avons alors détaillé plusieurs méthodes permettant de définir le préconditionneur $\mathbf{P}(\mathbf{x})$, que ce soit des préconditionnements constants ou par interpolation d'opérateurs. Ces **PIO** sont obtenus via la définition de plusieurs coefficients d'interpolation dont le coût numérique peut être très différent. Nous analyserons ainsi les simplifications qui ont été proposées afin de réduire ce coût. Finalement, cette nouvelle estimation statistique de l'erreur sera utilisée lors d'un problème d'optimisation que nous allons détailler par la suite.

2.4 Utilisation du modèle réduit dans le cadre de l'optimisation

Dans cette section, nous allons utiliser les outils développés précédemment pour proposer un nouvel algorithme d'optimisation couplant la réduction d'ordre de modèle et l'algorithme **EGO** (Section 1.1.3.3.3).

Dans le cadre de nos travaux, nous cherchons la solution du problème d'optimisation suivant :

$$\mathbf{x}_* \in \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmin}} s(\mathbf{x}) \quad (2.59)$$

où la fonction de performance $s(\mathbf{x})$ est obtenue via la solution d'un solveur $\mathbf{u}(\mathbf{x})$ détaillé dans l'Eq. (2.2). Nous cherchons la solution de ce problème d'optimisation via l'algorithme d'optimisation **EGO**. Pour rappel, l'algorithme **EGO** est initialisé sur un **DoE** de paramètres $\mathbf{x}_{\text{DoE}} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\} \in \mathcal{X}^m$ où la fonction de performance est évaluée pour chacun de ces paramètres. À partir des images $\{s(\mathbf{x}_1), \dots, s(\mathbf{x}_m)\}$, un **GP** est construit afin d'obtenir une approximation de la fonction objectif. Par la suite, des points, choisis par maximisation de l'amélioration espérée (définie dans l'Eq. (1.29)), sont ajoutés au **DoE**. Nous allons à présent détailler le couplage entre la réduction d'ordre de modèle et **EGO** lors de la phase d'initialisation et d'optimisation.

2.4.1 Initialisation

Lors de la phase d'initialisation de **EGO**, un **DoE** sur la fonction objectif est utilisé. Lors du calcul de ces points, toutes les informations nécessaires pour la réduction d'ordre de modèle sont calculées puis stockées. Les vecteurs $\mathbf{u}(\mathbf{x}_1), \dots, \mathbf{u}(\mathbf{x}_m)$ sont utilisés pour créer une base **POD** pour le problème primal comme cela est détaillé dans la Section 2.2.1. De la même façon, lors du calcul du **DoE**, les solutions $\mathbf{Q}(\mathbf{x}_1), \dots, \mathbf{Q}(\mathbf{x}_m)$ du problème dual (cf Eq. (2.10)) sont calculées et stockées pour obtenir une base **POD**. Enfin, les diagonales et les factorisations des matrices $\mathbf{K}(\mathbf{x}_1), \dots, \mathbf{K}(\mathbf{x}_m)$ sont stockées pour le calcul du préconditionnement. En pratique, nous ne conservons pas les m factorisations puisque la mémoire vive nécessaire serait trop importante. Nous ne conservons que $m' < m$ factorisations (typiquement $m' = 5$). Dans ce cas, seules les m' factorisations les plus intéressantes du point de vue de l'optimisation sont conservées. Notons que cette phase *Offline* n'est

pas pénalisée par la réduction d'ordre de modèle, hormis par le calcul de la solution du problème dual, puisque cette initialisation est imposée par l'algorithme EGO. Cette procédure est synthétisée dans l'Algorithme 9.

Algorithm 9: Initialization of EGO coupled with MOR

```

input :  $\mathbf{x}_{\text{DoE}}, m'$ 
 $\mathbf{x}_{\text{fact}} \leftarrow []$ ;
for  $\mathbf{x} \in \mathbf{x}_{\text{DoE}}$  do
    Assemble  $\mathbf{K}(\mathbf{x})$  and  $\mathbf{F}(\mathbf{x})$ ;
    Compute the factorisation of  $\mathbf{K}(\mathbf{x})$ ;
    Compute the solution  $\mathbf{u}(\mathbf{x})$  and  $\mathbf{Q}(\mathbf{x})$  (Eq. (2.1) and Eq. (2.10)) using the
        factorization;
    Compute  $s(\mathbf{x})$  using Eq. (2.2);
    if  $\text{len}(\mathbf{x}_{\text{fact}}) < m'$  then
        Add  $\mathbf{x}$  to  $\mathbf{x}_{\text{fact}}$ ;
    else if  $s(\mathbf{x}) < \max_{\mathbf{x} \in \mathbf{x}_{\text{fact}}} s(\mathbf{x})$  then
        Remove  $\arg \max_{\mathbf{x} \in \mathbf{x}_{\text{fact}}} s(\mathbf{x})$  from  $\mathbf{x}_{\text{fact}}$ ;
        Add  $\mathbf{x}$  to  $\mathbf{x}_{\text{fact}}$ ;
    end
    Compute POD basis  $\mathbf{B}_r^{\text{P}}$  from  $\{\mathbf{u}(\mathbf{x}), \mathbf{x} \in \mathbf{x}_{\text{DoE}}\}$ ;
    Compute POD basis  $\mathbf{B}_r^{\text{D}}$  from  $\{\mathbf{Q}(\mathbf{x}), \mathbf{x} \in \mathbf{x}_{\text{DoE}}\}$ ;
    Store the factorizations and diagonals of  $\{\mathbf{K}(\mathbf{x}), \mathbf{x} \in \mathbf{x}_{\text{fact}}\}$ ;
 $\mathbf{s}_{\text{DoE}} \leftarrow \{s(\mathbf{x}), \mathbf{x} \in \mathbf{x}_{\text{DoE}}\}$ ;
output:  $\mathbf{B}_r^{\text{P}}, \mathbf{B}_r^{\text{D}}$ , stored factorizations and diagonals at  $\mathbf{x}_{\text{fact}}, \mathbf{s}_{\text{DoE}}$ 

```

2.4.2 Utilisation de la réduction d'ordre de modèle lors de l'optimisation

Le but est de détailler comment utiliser l'approximation de la QI lors de l'optimisation. L'idée est alors d'estimer l'optimalité du point ajouté au DoE par EGO en utilisant la MOR.

De notre point de vue, EGO définit un point $\mathbf{x}_{\text{EGO}}$ par maximisation de l'amélioration espérée, puis la fonction objectif $s(\mathbf{x}_{\text{EGO}})$ est calculée. Le nouveau point est alors ajouté au DoE : $\mathbf{x}_{\text{DoE}} = \mathbf{x}_{\text{DoE}} \cup \{\mathbf{x}_{\text{EGO}}\}$, $\mathbf{s}_{\text{DoE}} = \mathbf{s}_{\text{DoE}} \cup \{s(\mathbf{x}_{\text{EGO}})\}$. Le GP est alors enrichi sur le nouveau DoE et EGO proposera alors un nouveau point $\mathbf{x}_{\text{EGO}}$. Cependant, nous rappelons que EGO cherche à combiner exploitation et exploration du domaine $\mathcal{X}$. Il arrive donc que le point proposé par EGO ne permette pas d'améliorer le minimum actuel :

$$s_{\min} = \min \mathbf{s}_{\text{DoE}} \quad (2.60)$$

soit $s(\mathbf{x}_{\text{EGO}}) > s_{\min}$. Dans ce cas, l'évaluation de $s(\mathbf{x}_{\text{EGO}})$ nécessite une précision moins importante puisque la zone autour de $\mathbf{x}_{\text{EGO}}$ n'est pas optimale. Cela permet d'utiliser, dans ces zones, des solutions approximatives issues de la MOR comme proposé par SOI-LAHOUDINE et collab. [2017]. C'est cette idée que nous allons également exploiter pour le couplage que nous proposons par la suite. Lorsque EGO va définir un nouveau point d'enrichissement, nous calculons l'approximation de la quantité d'intérêt $s_r^{\text{PD}}(\mathbf{x})$ et la borne d'erreur $\Delta_\alpha(\mathbf{x})$. L'estimation de la QI et la borne de l'erreur sont alors utilisées pour tester l'optimalité du point proposé par EGO $\mathbf{x}_{\text{EGO}}$. Deux cas sont alors possibles :

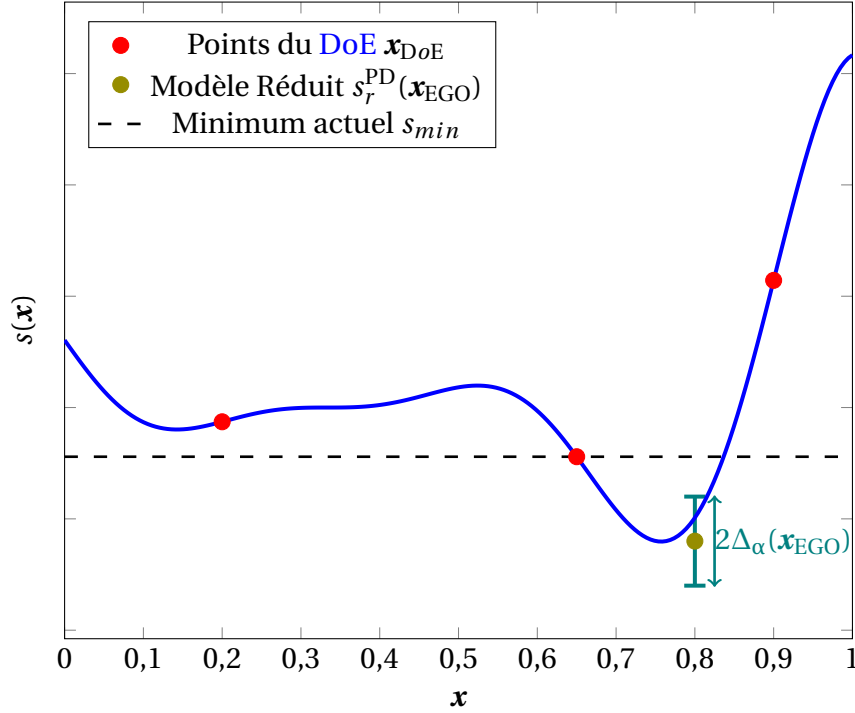


FIGURE 2.6 – Illustration du cas où la réduction de modèle ne permet pas de conclure sur la non-optimalité de $\mathbf{x}_{\text{EGO}}$, un appel au solveur sera nécessaire pour calculer $s(\mathbf{x}_{\text{EGO}})$.

- $s_r^{\text{PD}}(\mathbf{x}_{\text{EGO}}) - \Delta_\alpha(\mathbf{x}_{\text{EGO}}) < s_{\min}$: dans ce cas, le modèle réduit ne permet pas de conclure au niveau de confiance fixé sur la non-optimalité de $s(\mathbf{x}_{\text{EGO}})$. Ce cas est illustré sur la Figure 2.6. Dans ce cas, le modèle complet est évalué, permettant d'obtenir $s(\mathbf{x}_{\text{EGO}})$, $\mathbf{u}(\mathbf{x}_{\text{EGO}})$ et $\mathbf{Q}(\mathbf{x}_{\text{EGO}})$. On rajoute alors $\text{vect}(\mathbf{u}(\mathbf{x}_{\text{EGO}}))$ (resp. $\text{vect}(\mathbf{Q}(\mathbf{x}_{\text{EGO}}))$) à l'espace de projection V_r^{P} (resp. V_r^{D}) si l'erreur relative de reconstruction est supérieure à une tolérance ϵ_r définie par l'utilisateur. De plus, s'il s'avère que $s(\mathbf{x}_{\text{EGO}}) < s_{\min}$, l'opérateur $\mathbf{K}(\mathbf{x}_{\text{EGO}})^{-1}$ remplace l'opérateur correspondant au paramètre le moins optimal dans le calcul du préconditionnement.
- $s_r^{\text{PD}}(\mathbf{x}_{\text{EGO}}) - \Delta_\alpha(\mathbf{x}_{\text{EGO}}) > s_{\min}$: dans ce cas, il est fortement probable que le point $\mathbf{x}$ ne soit pas optimal (risque géré par le quantile de niveau α et la précision du préconditionneur). Il est alors inutile de calculer la vraie valeur de $s(\mathbf{x}_{\text{EGO}})$. Ce cas est illustré sur la Figure 2.7.

L'approche d'optimisation proposée, couplant la MOR et EGO est détaillée dans l'Algorithme 10 :

- Similairement à un algorithme EGO classique, le minimum actuel est calculé et un GP de la fonction objectif est construit.
- Le point $\mathbf{x}_{\text{EGO}}$ devant être ajouté au DoE est obtenu par maximisation de l'amélioration espérée définie dans l'Eq. (1.29).
- Contrairement à un algorithme sans réduction de modèle, l'approximation de la QI $s_r^{\text{PD}}(\mathbf{x}_{\text{EGO}})$ est calculée ainsi que l'estimation de l'erreur $\Delta_\alpha(\mathbf{x}_{\text{EGO}})$.
- Si le modèle ne permet pas de conclure sur la non-optimalité de $s(\mathbf{x}_{\text{EGO}})$ ($s_r^{\text{PD}}(\mathbf{x}_{\text{EGO}}) - \Delta_\alpha(\mathbf{x}_{\text{EGO}}) < s_{\min}$), la QI exacte est calculée et les bases réduites sont enrichies si cela est jugé nécessaire. Afin de décider si la base doit être enrichie avec le nouveau vecteur, nous évaluons l'erreur de reconstruction sur la base POD. La reconstruction

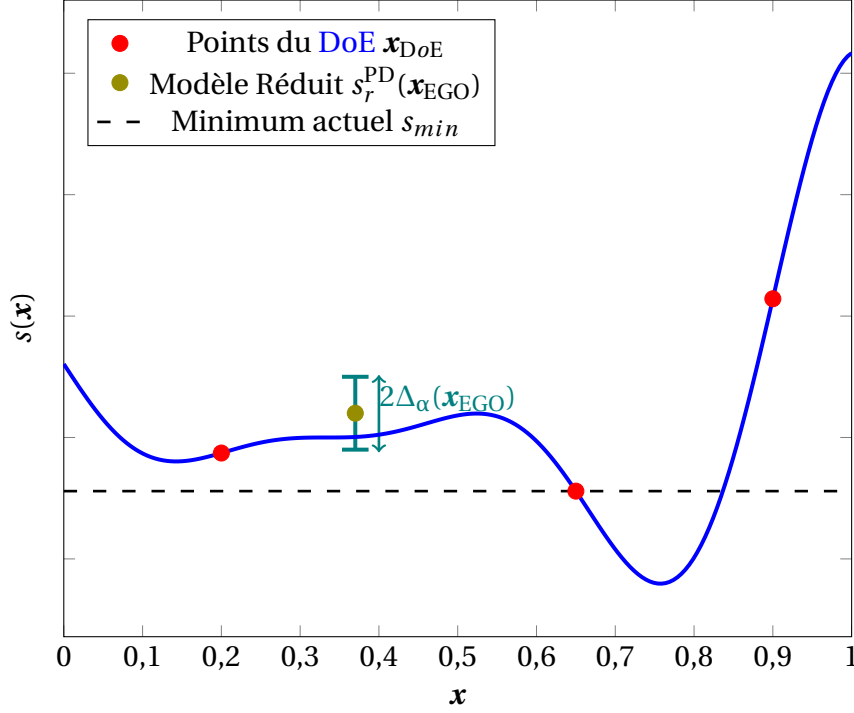


FIGURE 2.7 – Illustration du cas où la réduction de modèle permet d'assurer que $\mathbf{x}_{EGO}$ n'est pas optimal, $s_r(\mathbf{x}_{EGO})$ sera utilisé comme approximation de $s(\mathbf{x}_{EGO})$.

est définie par :

$$\mathbf{P}^{\Phi^P}(\mathbf{u}(\mathbf{x}_{EGO})) = \Phi_0^P + \sum_{k=1}^{n_r^P} \langle \mathbf{u}(\mathbf{x}_{EGO}) - \Phi_0^P, \Phi_k^P \rangle \Phi_k^P \quad (2.61)$$

pour la solution du problème primal et

$$\mathbf{P}^{\Phi^D}(\mathbf{Q}(\mathbf{x}_{EGO})) = \Phi_0^D + \sum_{k=1}^{n_r^D} \langle \mathbf{Q}(\mathbf{x}_{EGO}) - \Phi_0^D, \Phi_k^D \rangle \Phi_k^D \quad (2.62)$$

pour la solution du problème dual.

Si l'erreur de reconstruction relative est supérieure à une tolérance ϵ_r définie par l'utilisateur, la base **POD** correspondante est enrichie. En pratique, ajouter $\text{vect}(\mathbf{u}(\mathbf{x}_{EGO}) - \Phi_0^P)$ à V_r^P revient à ajouter la direction de $\mathbf{u}(\mathbf{x}_{EGO}) - \Phi_0^P$ orthogonale à Φ^P , après normalisation, à la matrice de projection Φ^P . De plus, la liste des factorisations utilisées pour le préconditionnement est actualisée de la même façon que lors de l'initialisation.

- Si $s_r^{PD}(\mathbf{x}_{EGO}) - \Delta_\alpha(\mathbf{x}_{EGO}) \geq s_{min}$, la **QI** sera approchée par le modèle réduit.
- Lorsque le budget n_{iter}^{max} est atteint, le minimum trouvé par **EGO** couplé à la réduction d'ordre de modèle est donné par s_{min} . Notons que la stratégie permet, par construction, de s'assurer que s_{min} est obtenue par une solution exacte du problème.

Algorithm 10: EGO coupled with MOR

input : $\mathbf{x}_{DoE}$, $\mathbf{s}_{DoE}$, n_{iter}^{max} , $\mathbf{x}_{fact}$, B_r^P , B_r^D , α , ϵ_r
 $s_{min} \leftarrow \min \mathbf{s}_{DoE}$;
 $n_{iter} \leftarrow 1$ **while** $n_{iter} \leq n_{iter}^{max}$ **do**
 $\hat{\mu}, \hat{s} \leftarrow$ Gaussian Process ($\mathbf{x}_{DoE}$, $\mathbf{s}_{DoE}$);
 $\mathbf{x}_{EGO} \leftarrow \arg\max_{\mathbf{x} \in \mathcal{X}} \mathbb{E}(\mathcal{J}(\mathbf{x}))$;
 Compute $s_r^P(\mathbf{x}_{EGO})$ using B_r^P and B_r^D (cf Eq. (2.13));
 Compute $\Delta_\alpha(\mathbf{x}_{EGO})$ using the preconditioning on the factorizations of $\mathbf{x}_{fact}$;
 Add $\mathbf{x}_{EGO}$ to $\mathbf{x}_{DoE}$;
 if $s_r^{PD}(\mathbf{x}_{EGO}) - \Delta_\alpha(\mathbf{x}_{EGO}) < s_{min}$ **then**
 Compute $\mathbf{u}(\mathbf{x}_{EGO})$, $\mathbf{Q}(\mathbf{x}_{EGO})$ and $s(\mathbf{x}_{EGO})$;
 if $\frac{\|\mathbf{u}(\mathbf{x}_{EGO}) - P^{\Phi^P}(\mathbf{u}(\mathbf{x}_{EGO}))\|}{\|\mathbf{u}(\mathbf{x}_{EGO})\|} \geq \epsilon_r$ **then**
 Add $vect(\mathbf{u}(\mathbf{x}_{EGO}) - \Phi_0^P)$ to V_r^P ;
 end
 if $\frac{\|\mathbf{Q}(\mathbf{x}_{EGO}) - P^{\Phi^D}(\mathbf{Q}(\mathbf{x}_{EGO}))\|}{\|\mathbf{u}(\mathbf{x}_{EGO})\|} \geq \epsilon_r$ **then**
 Add $vect(\mathbf{Q}(\mathbf{x}_{EGO}) - \Phi_0^D)$ to V_r^D ;
 end
 if $s(\mathbf{x}_{EGO}) < \max_{\mathbf{x} \in \mathbf{x}_{fact}} s(\mathbf{x})$ **then**
 Remove $\arg\max_{\mathbf{x} \in \mathbf{x}_{fact}} s(\mathbf{x})$ from $\mathbf{x}_{fact}$;
 Add $\mathbf{x}_{EGO}$ to $\mathbf{x}_{fact}$;
 end
 Add $s(\mathbf{x}_{EGO})$ to $\mathbf{s}_{DoE}$;
 else
 Add $s_r^{PD}(\mathbf{x}_{EGO})$ to $\mathbf{s}_{DoE}$;
 end
 output: s_{min}

2.5 Application

2.5.1 Description du cas d'application

Il est proposé de tester ces méthodes sur un cas de dimensionnement d'aile d'avion. La structure de l'aile est modélisée par un modèle éléments finis avec 2552 éléments plaques (DKT), soit 15 312 degrés de liberté. L'aile est composée de nervures (vert), d'un longeron avant (gris), d'un longeron arrière (rouge) et d'une peau (bleu) comme l'illustre la Figure 2.8. Chacun de ces éléments est caractérisé par des propriétés physiques communes (module d'Young $E = 72.4\text{GPa}$, coefficient de Poisson $\nu = 0.3$) mais par des épaisseurs différentes. Les douze épaisseurs forment les paramètres $\mathbf{x}$ de notre modèle et sont définies sur 4 groupes de composants de l'aile :

- la peau supérieure et inférieure
- les nervures
- le longeron avant
- le longeron arrière

De plus, l'aile est divisée, suivant l'envergure, en 3 sections. Le découpage de l'aile est illustrée sur la Figure 2.8a où les 3 sections sont représentées avec des nuances de bleu différentes pour la peau supérieure. Les quatre groupes, divisés en 3 sections de l'aile, permettent de définir 12 régions. Les paramètres $\mathbf{x}$ sont alors composés des épaisseurs de ces 12 régions.

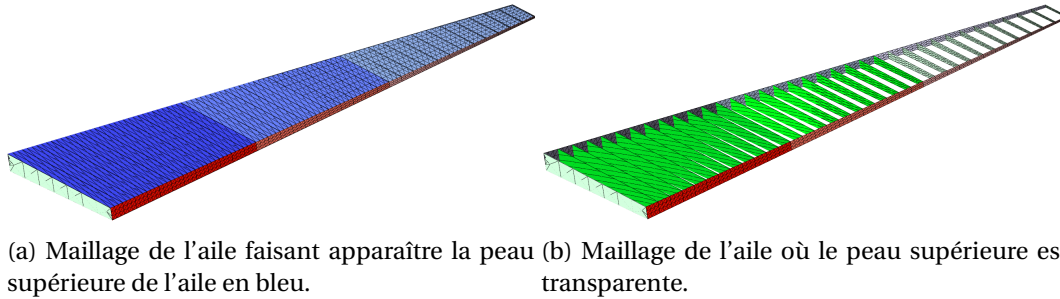


FIGURE 2.8 – Maillage de l'aile avec ou sans peau mettant en évidence les 3 sections et les différents composants : nervures (vert), longeron avant (gris), longeron arrière (rouge) et peau (bleu)

Le chargement est constant et calculé comme étant le résultat d'un calcul d'aéroélasticité où l'aérodynamique de l'aile est obtenue par une méthode *Vortex Lattice Method* (VLM) (KATZ et PLOTKIN [2001]) dans le cadre d'une certification (coefficient de charge à 2.5). Cette méthode sera décrite plus précisément dans la suite de la thèse puisque ce cas test sera utilisé comme application pour les développements réalisés sur l'optimisation multidisciplinaire. Ici, le chargement ne sera pas recalculé pour chaque nouveau paramètre mais sera considéré comme représentatif du calcul aéroélastique. En posant $\mathbf{K}(\mathbf{x})$ la matrice de rigidité de l'aile et $\mathbf{F}$ le chargement (supposé constant), le déplacement des noeuds de l'aile est obtenu en résolvant le système

$$\mathbf{K}(\mathbf{x})\mathbf{u}(\mathbf{x}) = \mathbf{F} \quad (2.63)$$

L'objectif est de minimiser le déplacement vertical maximal en un certain point du bout de l'aile :

$$s(\mathbf{x}) = u_z(\mathbf{x}) \quad (2.64)$$

où u_z est le déplacement vertical obtenu en extrayant le coefficient correspondant de $\mathbf{u}(\mathbf{x})$. La minimisation du déplacement vertical maximal est effectuée sous contrainte de masse :

$$\mathbf{x}_* \in \arg \min_{\mathbf{x} \in \mathcal{X}} s(\mathbf{x}) \quad \text{s.c. } M(\mathbf{x}) \leq M_{max} \quad (2.65)$$

où $M(\mathbf{x})$ est la masse de l'aile et M_{max} est la masse maximale admissible. La quantité $\mathbf{L}(\mathbf{x}) = \mathbf{L}$ ne dépend alors pas de $\mathbf{x}$ et est composée de zéros sauf en un point correspondant au noeud où le déplacement vertical est calculé. Le point particulier qui nous intéresse dans cette étude est au bout de l'aile. La masse de l'aile dépend uniquement de la géométrie initiale de l'aile et des épaisseurs $\mathbf{x}$. La masse est une fonction linéaire des paramètres, elle est obtenue analytiquement à partir des aires des différentes sections. Le coût de calcul de la masse est alors négligeable et ne sera pas considéré dans la suite.

2.5.2 Analyse des différents préconditionneurs

Dans un premier temps, nous souhaitons comparer les différentes méthodes de calcul des coefficients du **PIO**. Nous allons analyser ces différences en terme d'efficacité et de temps de calcul.

L'analyse sera effectuée sur différents problèmes, tous issus du cas présenté dans la Section 2.5.1. Les différences entre les problèmes résident dans le paramétrage du problème. Dans le premier cas, un seul paramètre est considéré. Ainsi, les douze épaisseurs présentées dans la Section 2.5.1 sont influencées par le même paramètre. Ceci permet de définir un cas 1D dont l'approximation et le conditionnement seront plus faciles à étudier. Le deuxième cas est un problème 4D obtenu en faisant varier les groupes de composants (nervures, longerons avant et arrière ainsi que la peau) sur l'aile complète plutôt que sur chacune des trois sections. Enfin, le problème 12D sera aussi utilisé.

Pour rappel, le **PIO** est défini par :

$$\mathbf{P}(\mathbf{x}) = \sum_{i=1}^{m'} \lambda_i(\mathbf{x}) \mathbf{P}^i \quad (2.66)$$

où m' est le nombre de préconditionneurs retenus pour l'interpolation, son choix résulte d'un compromis entre coût et précision du préconditionnement. Les coefficients $\lambda(\mathbf{x})$ sont égaux aux :

- coefficients $\lambda^Z(\mathbf{x})$ proposés par **ZAHM et collab. [2017]** définis dans l'Eq. (2.41).
- coefficients $\lambda^D(\mathbf{x})$ obtenus en remplaçant les factorisations par l'inverse des diagonales dont l'obtention est détaillée dans l'Eq. (2.50).
- coefficients $\lambda^I(\mathbf{x})$ obtenus par projection de la diagonale de $\mathbf{K}(\mathbf{x})$ sur les diagonales $\mathbf{K}(\mathbf{x}_1), \dots, \mathbf{K}(\mathbf{x}_{m'})$ définis dans l'Eq. (2.54).

Dans un premier temps, nous allons entraîner les différents modèles. La première étape est de générer un **DoE** sur l'espace paramétrique $\mathcal{X}$ pour construire les bases par **POD** : un échantillon de paramètres de taille 20 est généré par **LHS**. Les 20 solutions du problème primal et dual sont utilisées pour construire les bases **POD** correspondantes et 5 vecteurs sont conservés pour chacune des bases. De plus, les factorisations des matrices associées aux $m' = 10$ meilleurs points pour l'optimisation sont conservées pour les différents préconditionneurs. Afin de conserver un coût de calcul raisonnable, nous fixons K^V , définissant la matrice de projection aléatoire $\mathbf{Z}$ à 200. Les différents **PIO** sont alors comparés sur le critère suivant :

$$e_p(\mathbf{x}, \lambda(\mathbf{x})) = \frac{\left\| \left(\sum_{i=1}^{m'} \lambda_i(\mathbf{x}) \mathbf{P}^i \right) (\mathbf{K}(\mathbf{x}) \mathbf{u}_r(\mathbf{x}) - \mathbf{F}(\mathbf{x})) \right\|_2}{\| \mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x}) \|_2} \quad (2.67)$$

Notons que plus ce critère est proche de 1, plus le préconditionneur est considéré comme performant. Ce critère est calculé pour 100 points différents des espaces paramétriques de dimension 1, 4 et 12. Cet échantillon de test est généré par LHS. Les histogrammes de $e_p(\mathbf{x}, \boldsymbol{\lambda}(\mathbf{x}))$ pour les 100 points du LHS sont illustrés sur les Figures 2.9, 2.10 et 2.11 pour les différents modes de calcul des coefficients du PIO et sans préconditionnement. Nous remarquons que, dans le cas 12D, le conditionnement de la matrice de rigidité $\mathbf{K}(\mathbf{x})$ est très important. Cela se traduit par une norme du résidu de 6 ordres de grandeur plus grande que la norme de l'erreur. On peut remarquer que lorsque le nombre de paramètres est inférieur à 4, les coefficients du PIO définis par ZAHM et NOUY [2016] sont très performants et permettent d'obtenir une excellente estimation de l'erreur à partir du résidu préconditionné. Cependant, cette méthode se détériore lorsque la complexité du problème augmente comme on peut l'observer sur la Figure 2.11. Dans ce cas, le résidu préconditionné sous-estime de deux ordres de grandeur l'erreur. Les coefficients obtenus en remplaçant les factorisations par les inverses des diagonales notés $\boldsymbol{\lambda}^D(\mathbf{x})$ ne sont pas très performants sur l'ensemble des cas et on observe une différence de deux ordres de grandeur entre le résidu et l'erreur. Finalement, bien que l'interpolation par projection sur les diagonales des matrices de rigidité sous-estime plus souvent l'erreur lorsque le nombre de paramètres est inférieur à 4 comparé à $\boldsymbol{\lambda}^Z$, celle-ci permet d'approcher fidèlement le préconditionnement dans le cas où le conditionnement de la matrice de rigidité est détérioré. C'est donc ce préconditionnement que nous considérerons dans la suite.

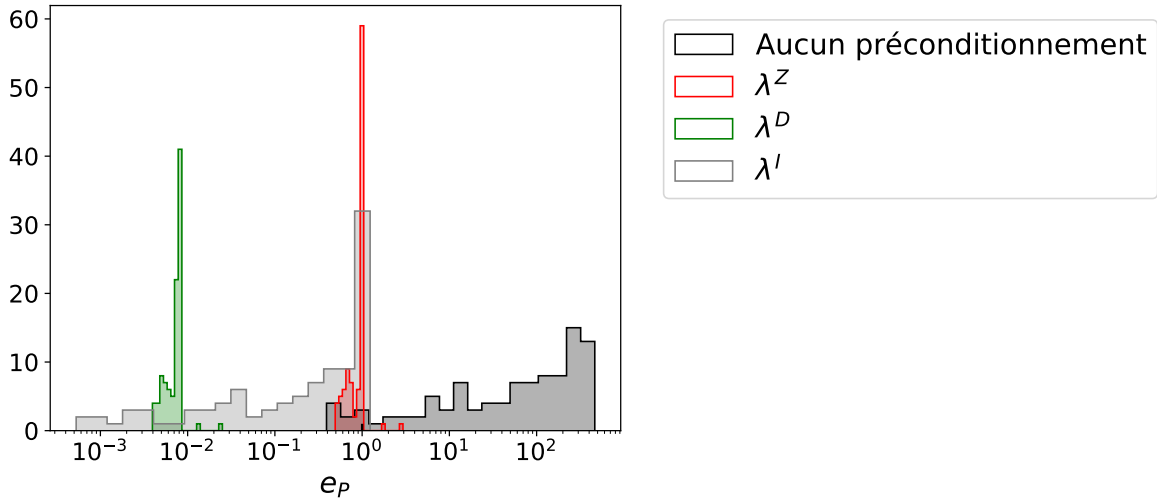


FIGURE 2.9 – Histogrammes de $e_p(\mathbf{x}, \boldsymbol{\lambda}(\mathbf{x}))$ défini dans l'Eq. (2.67) pour le PIO obtenu pour les différents coefficients ainsi que sans préconditionnement pour le cas 1D.

2.5.3 Analyse de la stratégie d'optimisation

Dans cette section, nous allons analyser la stratégie proposée en terme de coût numérique. En effet, plusieurs points sont contraignants :

- 1) la méthode impose l'assemblage des données puisque nous n'avons pas utilisé de décomposition affine des éléments comme cela est souvent utilisé dans la littérature.
- 2) le préconditionnement du résidu peut se révéler être une étape coûteuse comme nous allons le voir dans la suite.

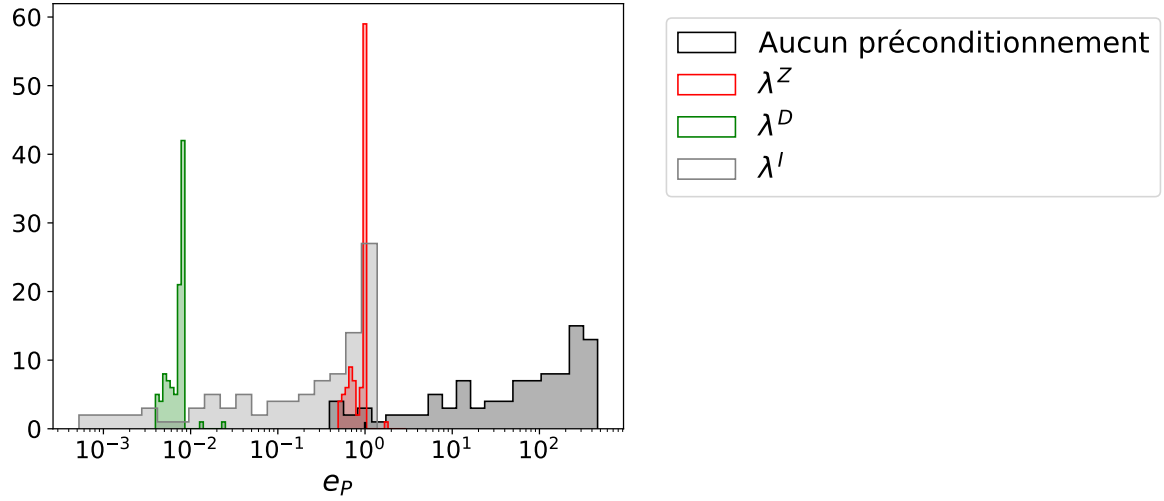


FIGURE 2.10 – Histogrammes de $e_p(\lambda(\mathbf{x}))$ défini dans l'Eq. (2.67) pour le PIO obtenu pour les différents coefficients ainsi que sans préconditionnement pour le cas 4D.

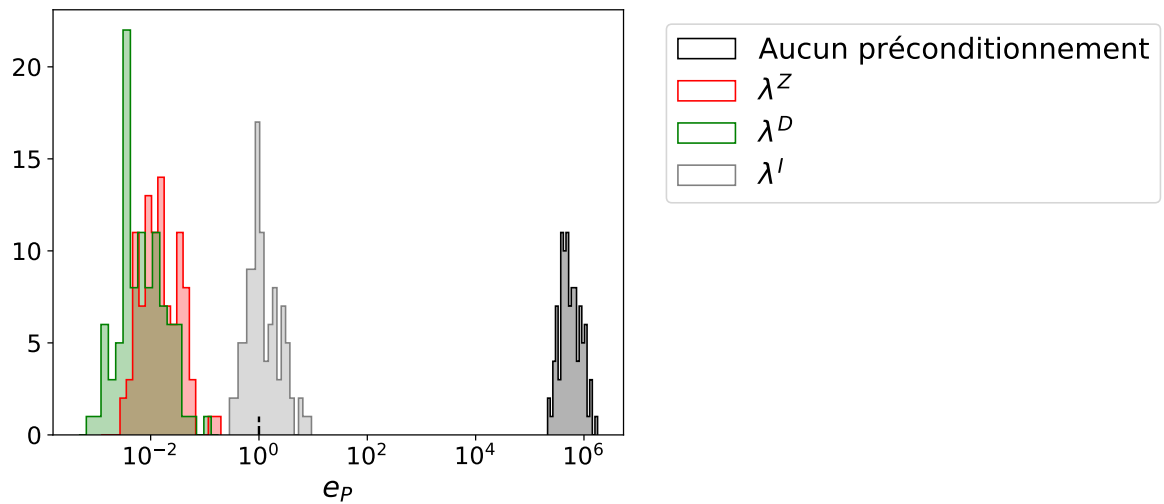


FIGURE 2.11 – Histogrammes de $e_p(\lambda(\mathbf{x}))$ défini dans l'Eq. (2.67) pour le PIO obtenu pour les différents coefficients ainsi que sans préconditionnement pour le cas 12D.

- 3) même si la réduction d'ordre de modèle est beaucoup plus efficace que la résolution du système linéaire complète, notre algorithme nécessite dans certains cas d'effectuer la résolution complète. La fréquence de cette résolution peut fortement impacter l'efficacité de notre méthode.

En ce qui concerne les temps de calcul, nous allons présenter une estimation de ceux-ci. Pour cela, nous allons additionner les temps nécessaires à la résolution des différents calculs dominants. Pour simplifier, nous ne considérons pas le coût numérique associé au produit scalaire entre deux vecteurs mais nous considérons :

- le temps d'assemblage de la matrice de rigidité $\mathbf{K}(\mathbf{x})$, du vecteur des efforts extérieurs $\mathbf{F}(\mathbf{x})$, de l'opérateur donnant la $\mathbf{QI} \mathbf{L}(\mathbf{x})$.
- le temps de factorisation de la matrice de rigidité $\mathbf{K}(\mathbf{x})$.
- les produits matriciels entre la matrice de rigidité $\mathbf{K}(\mathbf{x})$ et des vecteurs.

Les opérations sont effectuées sur 4 maillages de taille différente avec un nombre de degrés de liberté égal à 15 312, 64 680, 266 232 et 1 080 600. Ces différents maillages sont obtenus par raffinement (via le logiciel GMSH ([GEUZAIN et REMACLE \[2009\]](#))) du maillage initial de taille 15 312. Chaque opération a été réalisée 10 fois sur un serveur de calcul doté de processeurs Intel Xeon E5-2690 cadencés à 2,6GHz. Chaque calcul est effectué sur 8 coeurs en parallèle et le temps d'exécution est obtenu en sommant la participation de chaque processeur lors de l'opération.

Pour rappel, le préconditionnement par interpolation d'opérateurs est évalué à partir de la semi-norme de Frobenius calculée à partir d'une matrice aléatoire $\mathbf{V}$ de taille $K = 200$. Nous supposons que les bases réduites contiennent 20 vecteurs pour le problème primal et dual. Le nombre d'opérateurs retenus pour l'interpolation est fixé à 10. Finalement, la performance de notre méthode dépend fortement du nombre de fois où le modèle complet est utilisé, soit le nombre de fois où le modèle réduit n'a pas permis de conclure sur la non-optimalité du point proposé par [EGO](#) comme détaillé dans la Section 2.4.2. Nous allons modéliser ce choix par un paramètre p_s qui exprime le pourcentage d'échecs de la réduction d'ordre de modèle, soit la probabilité que lors d'une itération de l'Algorithme 10, le modèle réduit ne suffise pas pour prédire la quantité d'intérêt. Les temps de calcul pour 4 valeurs de p_s différentes sont illustrés sur la Figure 2.12. Comme nous pouvons l'observer, en utilisant le préconditionnement proposé par [ZAHM et NOUY \[2016\]](#) sans décomposition affine des éléments, le coût numérique nécessaire avec notre stratégie dépasse systématiquement le coût numérique du problème d'origine avec le problème de dimension $N = 15\,312$. Notons que dans la cas où $p_s = 0.75$, donc dans le cas où la [MOR](#) ne permet d'approcher la [QI](#) qu'une fois sur quatre, cette stratégie n'est pas rentable quelle que soit la dimension. Ceci illustre donc la nécessité de simplifier l'obtention des coefficients de l'interpolation d'opérateurs. Nous pouvons observer que les autres stratégies proposées permettent, quelle que soit la dimension du problème et pour chaque p_s utilisé, de réduire le temps de calcul. Néanmoins, lorsque $p_s = 0.75$, le gain est anecdotique. Ainsi, une des limitations de la performance de l'algorithme proposé réside dans le fait que p_s doit être petit. Cependant, il est extrêmement difficile de prédire cette donnée puisqu'elle dépend de la capacité de la [MOR](#) à approcher la [QI](#), de l'estimation de l'erreur commise et de l'algorithme d'optimisation utilisé. Supposons que l'algorithme [EGO](#) parvienne à trouver, à chaque itération, un paramètre $\mathbf{x}_{\text{EGO}}$ tel que $s(\mathbf{x}_{\text{EGO}}) \leq s_{\min}$ et que l'estimation de la borne de l'erreur commise soit correcte $|s_r^{\text{PD}}(\mathbf{x}_{\text{EGO}}) - s(\mathbf{x}_{\text{EGO}})| \leq \Delta_\alpha(\mathbf{x})$. Dans ce cas, $s_r^{\text{PD}}(\mathbf{x}_{\text{EGO}}) - \Delta_\alpha(\mathbf{x}) \leq s_{\min}$ donc la [MOR](#) ne permet pas de conclure sur la non-optimalité de $s(\mathbf{x}_{\text{EGO}})$. Dans ce cas, le modèle complet sera utilisé à chaque itération. Finalement, avec la nouvelle méthode de calcul des coefficients

du **PIO** ($\lambda^I(x)$), il est possible de réduire de 16% à 84% le coût numérique pour $N = 15\,312$ et de 23% à 97% pour $N = 1\,080\,600$. Notons que cette méthode est la moins coûteuse ce qui justifie son utilisation dans la suite.

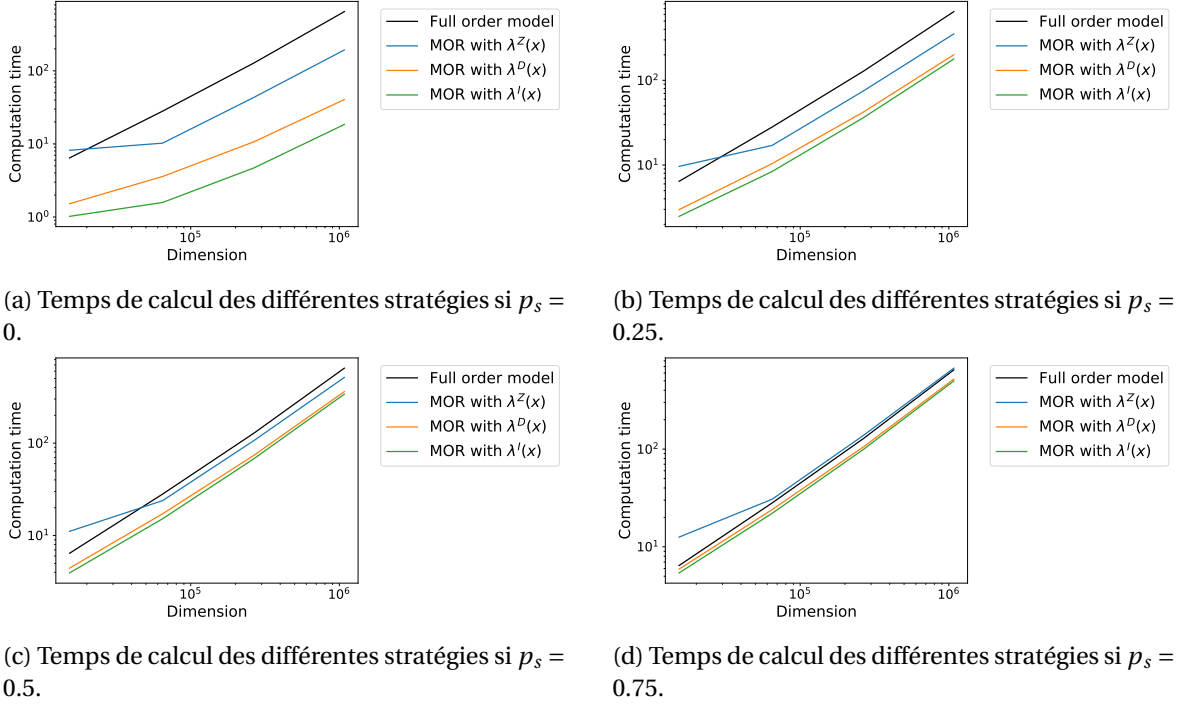


FIGURE 2.12 – Temps de calcul des différentes stratégies en fonction de la dimension N du problème et pour 4 valeurs du taux d'échec p_s . Les calculs des différents coefficients pour l'interpolation sont détaillés dans les Sections 2.3.3.2 et 2.3.3.3.

2.5.4 Optimisation sur différents cas test

Nous allons à présent nous intéresser à la résolution du problème d'optimisation afin de valider l'approche présentée dans la Section 2.4.2. Nous allons ainsi détailler les résultats obtenus sur les problèmes avec les différentes paramétrisations. Notons que dans notre analyse, nous sommes plus intéressés par la précision de la réduction d'ordre de modèle et l'efficacité de notre algorithme que par le résultat de l'optimisation. Nous analyserons donc les résultats de l'optimisation uniquement sur le cas 12D. Le budget alloué au **DoE** est donc paramétré afin d'obtenir une base réduite précise et le budget alloué à l'optimisation est choisi en fonction du nombre de variables de design du problème. Notons que lors de l'optimisation, la solution du solveur est calculée même lorsque cela n'est pas jugé nécessaire afin d'évaluer la performance de la **POD** et de l'estimation de l'erreur dans tous les cas.

2.5.4.1 Cas test 1D

Dans un premier temps nous allons détailler le cas 1D obtenu en faisant varier l'ensemble des épaisseurs via un seul paramètre de design. Nous fixons dans ce cas le nombre de points pour le **DoE** à 10 et le nombre d'itérations de l'algorithme d'**EGO** à 50. Nous fixons l'erreur relative de projection au sens des moindres carrés, permettant de définir le nombre de modes **POD** retenus (défini dans l'Eq. (1.84)), à 0.001%. Le nombre de modes conservés est alors de 2 pour la base primale et de 5 pour la base duale. De plus,

TABLEAU 2.1 – Cas test 1D. Moyenne et coefficient de variation des erreurs relatives sur le problème primal et dual obtenues par **POD** sur les 50 itérations de l'Algorithme 10.

	$\frac{\ u(x) - u_r(x)\ _2}{\ u(x)\ _2}$	$\frac{\ Q(x) - Q_r(x)\ _2}{\ Q(x)\ _2}$
Moyenne	1.43×10^{-5}	4.15×10^{-5}
Coefficient de variation (%)	29	5

TABLEAU 2.2 – Cas test 1D. Moyenne et coefficient de variation de l'erreur relative sur la quantité d'intérêt ainsi que l'évaluation de la performance des bornes de l'erreur définie dans l'Eq. (2.68) sur les 50 itérations de l'Algorithme 10.

	$\frac{ s(x) - s_r^{PD}(x) }{ s(x) }$	$e_s(x, 1)$	$e_s(x, 0.95)$
Moyenne	1.1×10^{-10}	1.4×10^{-4}	9.9×10^{-3}
Coefficient de variation (%)	86	101	101

5 factorisations sont conservées pour générer le **PIO**. Lors des itérations de l'optimisation avec l'Algorithme 10 couplant **MOR** et **EGO**, deux appels au modèle complet ont été effectués : $p_s = \frac{2}{50} = 4\%$. Ceci permet effectivement de valider notre approche puisque peu d'appels au modèle complet sont nécessaires. Néanmoins, notons que le cas 1D est aisé à approcher que ce soit du point de vue de la réduction d'ordre de modèle, comme cela est démontré par le peu de modes **POD** conservés, ou du point de vue de l'optimisation puisque la contrainte est linéaire et la fonction objectif est décroissante. L'intérêt de ce cas est principalement de montrer comment la réduction de modèle se comporte sur un cas simple. Ainsi, les erreurs relatives obtenues sur les approximations des problèmes primal et dual par **POD** sont présentées dans la Table 2.1. Notons que dans ce cas, aucun enrichissement d'aucune des bases comme proposé dans l'Algorithme 10 n'a été effectué puisque les erreurs relatives obtenues sont très faibles.

Dans un second temps, nous allons analyser la précision obtenue sur la quantité d'intérêt ainsi que la qualité de l'estimation de la borne de l'erreur et de l'erreur du problème primal par **POD**. Pour cela, nous nous intéressons à la moyenne de l'erreur relative entre la vraie quantité d'intérêt $s(x)$ et son approximation $s_r^{PD}(x)$. L'analyse de la borne de l'erreur sera faite sur le rapport entre la vraie erreur et la borne de l'erreur :

$$e_s(x, \alpha) = \frac{|s(x) - s_r^{PD}(x)|}{\Delta_\alpha(x)} \quad (2.68)$$

Notons que plus cette quantité est proche de 1, plus la borne de l'erreur est fidèle et plus elle est proche de 0 plus la borne surestime l'erreur. Nous considérons deux valeurs de α , $\alpha = 1$ ce qui revient à considérer le quantile égal à 1 dans la Section 2.3.1 et $\alpha = 95\%$. Les moyennes et **CV** de ces quantités sont affichées dans la Table 2.2. L'erreur relative sur la quantité d'intérêt est de l'ordre de 10^{-10} démontrant la capacité d'approximation de la **POD** dans ce cas. De plus, la borne de l'erreur statistique proposée dans la Section 2.3.1 permet de surestimer d'un facteur 100 l'erreur contrairement à la borne de l'erreur proposée par la littérature dont la surestimation est de 4 ordres de grandeur plus importante que l'erreur réelle.

Sur ce cas, la **POD** permet d'approcher très fidèlement la **QI** permettant de résoudre le problème d'optimisation avec un nombre fortement réduit d'appels au solveur. Cependant, ce cas est très simple, du point de vue de l'optimisation comme de la réduction d'ordre de modèle et nous allons donc observer comment notre algorithme se comporte lorsque la complexité du modèle augmente. Néanmoins, nous pouvons observer que l'algorithme converge bien vers la solution du problème.

TABEAU 2.3 – Cas test 4D. Moyenne et coefficient de variation des erreurs relatives sur le problème primal et dual obtenues par POD sur les 150 itérations de l'Algorithme 10.

	$\frac{\ \mathbf{u}(\mathbf{x}) - \mathbf{u}_r(\mathbf{x})\ _2}{\ \mathbf{u}(\mathbf{x})\ _2}$	$\frac{\ \mathbf{Q}(\mathbf{x}) - \mathbf{Q}_r(\mathbf{x})\ _2}{\ \mathbf{Q}(\mathbf{x})\ _2}$
Moyenne	1.57×10^{-4}	9.43×10^{-4}
Coefficient de variation (%)	72	235

TABEAU 2.4 – Cas test 4D. Moyenne et coefficient de variation de l'erreur relative sur la quantité d'intérêt ainsi que l'évaluation de la performance des bornes de l'erreur définie dans l'Eq. (2.68) sur les 150 itérations de l'Algorithme 10.

	$\frac{ s(\mathbf{x}) - s_r^{\text{PD}}(\mathbf{x}) }{ s(\mathbf{x}) }$	$e_s(\mathbf{x}, 1)$	$e_s(\mathbf{x}, 0.95)$
Moyenne	2.6×10^{-5}	5.1×10^{-4}	3.7×10^{-2}
Coefficient de variation (%)	112	77	77

2.5.4.2 Cas test 4D

Pour le cas 4D, nous fixons dans ce cas le nombre de points pour le DoE à 20 et le nombre d'itérations de l'algorithme d'EGO à 150. Nous fixons l'erreur relative de projection au sens des moindres carrés à 0.001%. Le nombre de modes conservés est alors de 8 pour la base primale et de 18 pour la base duale. Notons que seulement 2 vecteurs sur 20 n'ont pas été gardés pour la base du problème dual. Ceci est justifié par le fait qu'obtenir une base pour ce problème n'est pas trivial sachant que le second membre est un vecteur particulier dont les coefficients sont tous nuls sauf 1. De plus, 5 factorisations sont conservées pour générer le PIO. Lors des itérations de l'optimisation avec l'Algorithme 10 couplant MOR et EGO, 98 appels au modèle complet ont été effectués : $p_s = \frac{98}{150} = 61\%$. Le nombre d'appels au solveur a beaucoup augmenté comparé au cas 1D. Notons qu'une des limites de notre algorithme est que lorsque EGO propose un point où la contrainte n'est pas respectée et que la fonction objectif est plus petite que $s_{\min}$ (cf Eq. (2.60)) le modèle complet est appelé. En conséquence, lorsque la contrainte est plus difficile à apprendre, le nombre de points où le modèle complet est appelé peut augmenter. Les erreurs relatives obtenues sur les approximations des problèmes primal et dual par POD sont présentées dans la Table 2.3. Notons que dans ce cas aussi, aucun enrichissement d'aucune des bases n'a été effectué.

Dans un second temps, nous allons analyser la précision obtenue sur la quantité d'intérêt ainsi que la qualité de l'estimation de la borne de l'erreur et de l'erreur du problème primal. Les moyennes et CV de ces quantités sont affichées dans la Table 2.4. L'erreur relative sur la quantité d'intérêt est de l'ordre de 10^{-5} démontrant à nouveau la capacité d'approximation de la POD. Néanmoins, nous pouvons observer que la précision a diminué puisque dans ce cas, l'erreur relative est de 5 ordres supérieure à celle du cas 1D. La surestimation de l'erreur dans les cas 1D et 4D reste assez stable avec un ordre d'environ deux. Cependant, dans ce cas, l'influence de la borne statistique est importante. En effet, 21 appels au solveur sur les 92 ont été évités en considérant la borne statistique de l'erreur. Ce cas correspond à

$$s_r^{\text{PD}}(\mathbf{x}) - \Delta_1(\mathbf{x}) < s(\mathbf{x}) < s_r^{\text{PD}}(\mathbf{x}) - \Delta_{0.95}(\mathbf{x}) \quad (2.69)$$

TABLEAU 2.5 – Cas test 12D. Moyenne et coefficient de variation des erreurs relatives sur le problème primal obtenues par **POD** en fonction du nombre d'enrichissements de la base effectués lors des 300 itérations de l'Algorithme 10.

Nombre enrichissements	0	0	0	0	0
Moyenne	4.78×10^{-5}	1.51×10^{-4}	3.26×10^{-5}	1.46×10^{-4}	6.72×10^{-5}
Coefficient de variation	60	48	185	79	122

TABLEAU 2.6 – Cas test 12D. Moyenne et coefficient de variation des erreurs relatives sur le problème dual obtenues par **POD** en fonction du nombre d'enrichissements de la base effectués lors des 300 itérations de l'Algorithme 10.

Nombre enrichissements	7	0	0	1	3
Moyenne	4.97×10^{-3}	7.66×10^{-4}	2.82×10^{-3}	2.56×10^{-3}	2.86×10^{-3}
Coefficient de variation	63	86	66	238	112

2.5.4.3 Cas 12D

Nous allons à présent nous intéresser au cas le plus complet utilisant les douze variables de design du problème. Nous avons choisi 36 points pour le **DoE**. L'optimisation a été effectuée 5 fois afin de rendre compte de la variance du résultat en fonction du **DoE**. Ainsi, 5 **DoE** de taille 36 ont été générés par **LHS** et la résolution de l'optimisation en utilisant l'Algorithme 10 est réalisée pour chacun de ces **DoE**.

Le nombre de modes **POD** conservés est de 20 (3 fois) ou 21 (2 fois) pour la base primale. Pour la base duale, 35 modes **POD** sur 36 sont conservés : la corrélation entre les vecteurs de la solution duale est faible. Par la suite, 300 itérations de l'Algorithme 10 sont effectuées. La base primale ne sera pas réenrichie puisque l'erreur relative est inférieure à 1% pour chaque calcul. Cependant, la base duale sera enrichie entre 0 et 7 fois. Les erreurs relatives de projection moyenne ainsi que leurs coefficients de variation sont affichés dans la Table 2.5 pour le problème primal et dans la Table 2.6 pour le problème dual. Nous observons que le problème dual est plus difficile à approcher, l'erreur relative sur le problème dual étant beaucoup plus importante que celle sur le problème primal. Ceci explique donc les enrichissements nécessaires pour la base duale. Notons que le nombre d'enrichissements a peu d'influence sur l'erreur moyenne relative. Nous affichons de même l'erreur relative sur la **QI** ainsi que les bornes avec et sans correction dans la Table 2.7. Notons que l'erreur sur la quantité d'intérêt reste de l'ordre de 10^{-5} alors que l'erreur relative sur le problème dual a augmenté entre le cas 4D et le cas 12D. Ainsi, la correction duale permet d'obtenir une bonne approximation de la **QI** même lorsque l'un des problèmes est difficile à approcher par **POD**.

Les variations du nombre d'appels au solveur complet et de p_s sur les 5 optimisations sont fournies dans la Table 2.8. Nous pouvons observer que dans ce cas, le nombre

TABLEAU 2.7 – Cas test 12D. Moyenne de l'erreur relative sur la quantité d'intérêt ainsi que l'évaluation de la performance des bornes de l'erreur définie dans l'Eq. (2.68) sur les 300 itérations de chaque réalisation de l'Algorithme 10.

Optimisation	1	2	3	4	5
$\frac{ s(\mathbf{x}) - s_r^{\text{PD}}(\mathbf{x}) }{ s(\mathbf{x}) }$	1.8×10^{-5}	3.1×10^{-5}	3.3×10^{-5}	5.5×10^{-5}	8.5×10^{-5}
$e_s(\mathbf{x}, 1)$	2.2×10^{-4}	1×10^{-4}	4.9×10^{-4}	1.3×10^{-4}	5×10^{-4}
$e_s(\mathbf{x}, 0.95)$	1.6×10^{-2}	7.6×10^{-3}	3.6×10^{-2}	9.8×10^{-3}	3.7×10^{-2}

TABLEAU 2.8 – Cas test 12D. Nombre d'appels au solveur complet et p_s correspondant.

nombre d'appels	230	232	239	247	206
p_s (%)	77	77	80	82	69

TABLEAU 2.9 – Cas test 12D. Nombre d'appels au solveur complet et p_s correspondant en prenant en compte la contrainte.

nombre d'appels	105	85	59	101	70
p_s (%)	35	28	26	34	23

d'appels au solveur est plus important, le gain en temps de calcul est donc moins intéressant dans ce cas puisque en moyenne 231 appels au modèle complet sont effectués. Le p_s moyen est de 77% ce qui permet de réduire de 13% le coût numérique de l'optimisation. Néanmoins, on peut noter l'utilité de la borne statistique de l'erreur puisque celle-ci a permis d'éviter en moyenne 47 appels au solveur (cas détaillé dans l'Eq. (2.69)). Ainsi, sans cette correction, le nombre moyen d'appels au solveur serait de 278 soit un p_s de 93%.

Finalement notons que, dans le cadre de nos études, comme la contrainte est indépendante de la solution du solveur il est possible de l'évaluer avant d'évaluer la réponse du solveur. Ainsi, dans le cas où le point proposé par EGO ne respecte pas la contrainte, une amélioration de l'Algorithme 10 serait d'utiliser le modèle réduit dans ce cas. En effet, dans ces zones, la précision sur la quantité d'intérêt n'est pas nécessaire puisque la zone n'est pas faisable. Le nombre d'appels et le p_s correspondant en prenant en compte cette modification et en négligeant l'influence de l'erreur commise sur la quantités d'intérêt sont affichés dans la Table 2.9. Le p_s moyen est de 30% conduisant à une réduction de 65% du temps de calcul.

Nous allons à présent valider les résultats de l'optimisation. Pour cela, nous proposons de comparer les résultats obtenus avec et sans réduction de modèle. Les valeurs optimales obtenues par les algorithmes d'optimisation sont détaillées dans la Table 2.10. Nous n'observons pas de différence significative entre les résultats des deux méthodes. Notons que le résultat est plus influencé par le DoE initial qui est le même pour les deux méthodes. Notons que l'optimisation par EGO est rendue difficile par le fait que l'optimum se situe quasiment dans un coin du domaine de définition des variables de design et par le nombre de variables de design.

Par la suite, la solution de référence du problème est obtenue en utilisant une méthode par gradient, dont le point de départ est fixé au point central de l'espace de design. Comme le gradient de la fonction objectif n'est pas connu, il est estimé par différences finies. Cette méthode conduit à plus de 1000 appels au solveur. Les erreurs relatives obtenues entre l'optimisation par gradient et le résultat de l'Algorithme 10 sont affichées dans la Table 2.11. Nous avons de la même façon détaillé les résultats obtenus sans réduction d'ordre de modèle dans la Table 2.12. Finalement, aucune différence significative n'est observable entre les deux méthodes permettant de valider le fait que notre méthode ne

TABLEAU 2.10 – Cas test 12D. Solutions du problème d'optimisation en utilisant l'Algorithme 10 (couplant EGO et MOR) et en utilisant l'algorithme EGO sans réduction d'ordre de modèle.

Optimisation	1	2	3	4	5
s_* avec MOR	1.67	1.74	1.81	1.66	1.82
s_* sans MOR	1.64	1.87	1.78	1.63	1.91

TABLEAU 2.11 – Cas test 12D. Solutions du problème d'optimisation en utilisant l'Algorithme 10 (couplant EGO et MOR) et erreur relative avec la référence.

Optimisation	1	2	3	4	5
s_*	1.67	1.74	1.81	1.66	1.82
erreur relative	7%	12%	16%	6%	17%

TABLEAU 2.12 – Cas test 12D. Solutions du problème d'optimisation en utilisant l'algorithme EGO sans réduction d'ordre de modèle et erreur relative avec la référence.

Optimisation	1	2	3	4	5
s_*	1.64	1.87	1.78	1.63	1.91
erreur relative	5%	20%	14%	4%	23%

détériorer pas le résultat de l'optimisation. Les principaux écarts avec la solution de référence sont dus à l'utilisation d'EGO. Cependant, notre méthode a permis de remplacer en moyenne 23% des calculs complets par des calculs projetés sans prendre en compte la contrainte et 71% en la prenant en compte.

2.6 Bilan du Chapitre

Dans ce chapitre, l'objectif était d'appliquer la réduction d'ordre de modèle sur un cas d'optimisation monodisciplinaire linéaire avec une QI linéaire en développant une méthode proche de l'approche *on-the-fly*. Pour cela, nous avons en premier lieu détaillé une correction de l'estimation de la QI proposée dans la littérature permettant d'affiner l'estimation par rapport à l'utilisation seule du problème primal. Comme nous ne souhaitions pas effectuer d'approximation de la QI lors de l'optimisation, nous avons décidé de certifier l'approximation par POD en utilisant une estimation de l'erreur commise. Il est alors nécessaire de rendre l'estimation de l'erreur la plus proche possible de la véritable erreur. Pour cela, deux apports ont été proposés pour améliorer cette estimation. La première contribution concerne le produit scalaire entre deux vecteurs de grande dimension. En effet, dans la littérature, ce produit scalaire était généralement borné par le produit des normes par application de la formule de Cauchy-Schwarz. Néanmoins, plus la dimension du système linéaire est importante, plus la probabilité que les deux vecteurs soient alignés, et donc d'obtenir l'égalité dans la formule de Cauchy-Schwarz, est faible. Pour pallier cette difficulté, nous avons décidé d'effectuer une analyse statistique basée sur l'hypothèse que la direction de l'erreur du problème primal est distribuée uniformément sur la sphère unité. Nous avons alors défini un coefficient basé sur un indice de confiance α permettant de réduire l'écart entre l'estimation de l'erreur et la vraie erreur. La deuxième contribution implique l'estimation de la norme de l'erreur du problème primal par préconditionnement. Pour cela, nous avons utilisé le PIO proposé par ZAHM et NOUY [2016]. Cependant, le coût de calcul associé à l'obtention des coefficients de l'interpolation était jugé trop important sans décomposition affine des éléments. Nous avons alors proposé deux façons d'obtenir les coefficients plus adaptés à l'approche *on-the-fly*. Lors d'une dernière étape, une méthode d'optimisation couplant l'algorithme EGO et la MOR a été proposée. L'idée principale est d'utiliser la MOR pour approcher la QI aux points les moins intéressants lors de l'optimisation. Finalement, nous avons analysé les différentes méthodes sur un cas structure de dimension croissante.

Les améliorations proposées sur l'estimation de l'erreur ont par la suite été validées

sur le cas test. Nous avons remarqué que, sans la correction probabiliste sur la **QI**, la méthode couplant **EGO** et **MOR** est beaucoup moins efficace. Cette correction, simple à mettre en oeuvre permet alors, en fonction de la dimension du problème, de ne pas sur-évaluer l'estimation de l'erreur. Le **PIO** a été utilisé et permet, avec précision, d'estimer l'erreur du problème primal. Néanmoins, cette méthode requiert de calculer et d'utiliser la factorisation de la matrice de rigidité et de sauvegarder ces factorisations. Ceci est une des limites de cette méthode puisque cette opération est très intrusive. Nous avons vu que les calculs des coefficients proposés pour le **PIO**, notamment par projection des diagonales, permettent de réduire fortement le coût de calcul tout en étant plus stable lors de l'augmentation du nombre de variables de design. Finalement, le couplage entre **MOR** et **EGO** conduit à une réduction de 13% du coût numérique sans prendre en compte la contrainte et de 65% si elle est prise en compte dans le cas 12D. Les hypothèses de cette méthode restreignent la capacité d'adaptabilité de cette méthode. En effet, les problèmes d'optimisation de **QI** linéaire sont particuliers et une des perspectives de nos travaux serait d'adapter cette méthode aux problèmes non linéaires.

En conclusion, nous avons développé des méthodes permettant d'accélérer le temps de calcul sans utiliser de décomposition affine des éléments et sans faire d'approximation sur le résultat de l'optimisation. L'un des objectifs de ces travaux était d'étudier la possibilité d'adapter cette méthode à l'approximation d'une **MDA** présentée dans la Section 1.2.1.1. Comme l'erreur relative moyenne faite sur la quantité d'intérêt est de l'ordre de 10^{-5} , la **MOR** permettrait d'approcher fidèlement les solveurs disciplinaires. Ainsi, il serait possible d'appliquer cette méthode pour l'approximation de chacune de ces disciplines afin d'obtenir une approximation de la **MDA**. Cependant, même s'il est possible de calculer l'erreur sur une quantité d'intérêt linéaire (et donc potentiellement sur les variables de couplage), cette erreur ne pourrait être propagée sur une **MDA** à cause du couplage non linéaire entre les disciplines. De plus, sans décomposition affine des éléments, l'assemblage de la matrice de rigidité est nécessaire et nous avons observé que son coût est non négligeable. Cette remarque est d'autant plus pertinente sur les calculs aérodynamiques utilisant une méthode **VLM** où le temps d'assemblage est plus important que le temps d'inversion du système. Dans la suite, nous allons donc utiliser une autre approche permettant d'obtenir une approximation sans assemblage des données : la **POD+I** présentée en Section 1.3.2.3. Dans ce cas, il ne sera pas possible de calculer une borne de l'erreur théorique sur l'erreur de projection. Néanmoins, nous allons développer une estimation de l'erreur due à l'interpolation dans un contexte multidisciplinaire.

2.7 Références

- AVRON, H. et S. TOLEDO. 2011, «Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix», *Journal of the ACM (JACM)*, vol. 58, n° 2, p. 1–34. [98](#)
- BEKAS, C., E. KOKIOPOULOU et Y. SAAD. 2007, «An estimator for the diagonal of a matrix», *Applied numerical mathematics*, vol. 57, n° 11-12, p. 1214–1229. [98](#)
- CUONG, N. N., K. VEROY et A. T. PATERA. 2005, «Certified real-time solution of parametrized partial differential equations», dans *Handbook of materials modeling*, Springer, p. 1529–1564. [89](#)
- DAHMEN, W., C. PLESKEN et G. WELPER. 2014, «Double greedy algorithms : Reduced basis methods for transport dominated problems», *ESAIM : Mathematical Modelling and*

- Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, vol. 48, n° 3, doi: 10.1051/m2an/2013103, p. 623–663. URL <http://www.numdam.org/articles/10.1051/m2an/2013103/>. 87
- GEUZAIN, C. et J.-F. REMACLE. 2009, «Gmsh : A 3-D finite element mesh generator with built-in pre-and post-processing facilities», *International journal for numerical methods in engineering*, vol. 79, n° 11, p. 1309–1331. 110
- GHANEM, R. G. et R. M. KRUGER. 1996, «Numerical solution of spectral stochastic finite element systems», *Computer methods in applied mechanics and engineering*, vol. 129, n° 3, p. 289–303. 96
- GONZÁLEZ, L. 2006, «Orthogonal projections of the identity : spectral analysis and applications to approximate inverse preconditioning», *SIAM review*, vol. 48, n° 1, p. 66–75. 97
- GROTE, M. J. et T. HUCKLE. 1997, «Parallel preconditioning with sparse approximate inverses», *SIAM Journal on Scientific Computing*, vol. 18, n° 3, p. 838–853. 97
- JANON, A., M. NODÉ et C. PRIEUR. 2016, «Goal-oriented error estimation for the reduced basis method, with application to sensitivity analysis», *Journal of Scientific Computing*, vol. 68, n° 1, p. 21–41. 87, 89, 91
- KATZ, J. et A. PLOTKIN. 2001, *Low-speed aerodynamics*, vol. 13, Cambridge university press. 89, 106
- MACHIELS, L., A. T. PATERA, J. PERAIRE et Y. MADAY. 1998, «A general framework for finite element a posteriori error control : application to linear and nonlinear convection-dominated problems», dans *Proceedings of Institute for Computational Fluid Dynamics Conference, Oxford*. 89
- MADAY, Y., A. T. PATERA et D. V. ROVAS. 2002, «A blackbox reduced-basis output bound method for noncoercive linear problems», dans *Nonlinear partial differential equations and their applications : Collège de France seminar. Volume XIV*, Elsevier, p. 533–569. 89
- SOILAHOU DINE, M., C. GOGU et C. BES. 2017, «Accelerated adaptive surrogate-based optimization through reduced-order modeling», *AIAA Journal*, vol. 55, n° 5, doi: 10.2514/1.J055252, p. 1681–1694. URL <https://hal.archives-ouvertes.fr/hal-01943030>. 102
- ZAHM, O., M. BILLAUD-FRIESS et A. NOUY. 2017, «Projection-based model order reduction methods for the estimation of vector-valued variables of interest», *SIAM Journal on Scientific Computing*, vol. 39, n° 4, p. A1647–A1674. 87, 89, 91, 107
- ZAHM, O. et A. NOUY. 2016, «Interpolation of inverse operators for preconditioning parameter-dependent equations», *SIAM Journal on Scientific Computing*, vol. 38, n° 2, p. A1044–A1074. 87, 96, 97, 98, 100, 108, 110, 116

Chapitre 3

Optimisation multidisciplinaire combinée à la réduction d'ordre de modèle.

« On ne peut comprendre un processus en l'interrompant. La compréhension doit rejoindre le cheminement du processus et cheminer avec lui. »

Frank Herbert

Sommaire

3.1 Introduction	121
3.1.1 Objectifs	122
3.1.2 Description globale de la méthode	123
3.1.3 Organisation du Chapitre	123
3.2 Description de la résolution de la MDA par DPOD+I	125
3.2.1 Aperçu de la méthodologie proposée	125
3.2.2 Approximation des solveurs disciplinaires par POD+I	127
3.2.2.1 Approximation par POD et MDA réduite	127
3.2.2.2 Interpolation des solveurs réduits par Processus Gaussiens	129
3.2.3 Entraînement des processus gaussiens	129
3.2.3.1 Initialisation de l'algorithme d'entraînement des processus gaussiens	131
3.2.3.2 Algorithme d'entraînement	132
3.2.4 Résolution de la MDA à partir des GP disciplinaires	135
3.2.4.1 Générer des échantillons de la MDA aléatoire réduite	137
3.2.4.2 Évaluation de la dispersion des solutions de la MDA approchée	138
3.2.5 Enrichissement de l'analyse multidisciplinaire approchée	139
3.2.5.1 Enrichissement local des GP disciplinaires	139
3.2.5.2 Enrichissement du modèle initial	141
3.2.6 Comparaison avec la POD+I	143
3.3 Application sur un cas aéroélastique	146

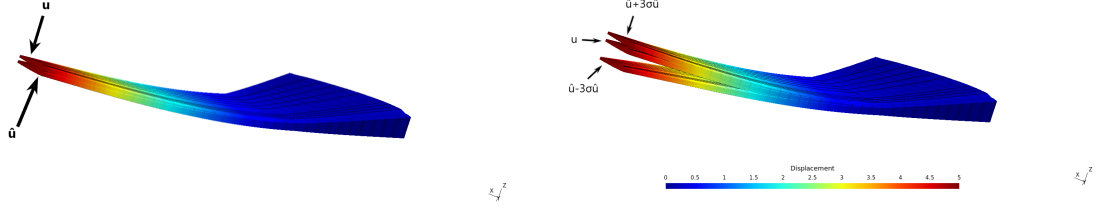
3.3.1	Description de la MDA	146
3.3.1.1	Description des solveurs disciplinaires	146
3.3.1.2	Description du couplage entre les disciplines	147
3.3.1.3	Description de l'avion	147
3.3.1.4	Paramétrisation du problème	149
3.3.1.5	Réglage des paramètres de la DPOD+I	150
3.3.2	Problème à 4 variables de design	151
3.3.2.1	Résolution avec le solveur DPOD+I	151
3.3.2.2	Comparaison avec la POD+I classique	155
3.3.2.3	Comparaison avec la POD+I enrichie	156
3.3.2.4	Robustesse de l'algorithme d'entraînement	156
3.3.3	Problème à 8 variables de design	160
3.3.3.1	Résolution avec le solveur DPOD+I	160
3.3.3.2	Comparaison avec la POD+I classique	162
3.3.3.3	Comparaison avec la POD+I enrichie	163
3.3.4	Conclusion sur le premier cas test	164
3.4	Application sur un deuxième cas aéroélastique	164
3.4.1	Description du modèle des équations d'Euler	164
3.4.2	Paramétrisation du solveur aérostructure	165
3.4.3	Maillages	165
3.4.4	Difficultés du modèle	166
3.4.5	Convergence de la MDA aléatoire	168
3.4.5.1	Limitation des sources d'incertitude de la MDA aléatoire	168
3.4.5.2	Modification de l'algorithme de résolution	169
3.4.6	Résultats	171
3.4.6.1	Analyse de l'entraînement	171
3.4.6.2	Analyse du solveur DPOD+I Jacobi	171
3.4.6.3	Analyse du solveur DPOD+I	173
3.4.7	Conclusion sur le second cas test aéroélastique	174
3.5	Optimisation multidisciplinaire	175
3.5.1	Couplage MDF	175
3.5.2	Couplage SBO	177
3.5.3	Couplage EGMDO	177
3.5.3.1	Fonction objectif aléatoire	178
3.5.3.2	Approximation de la fonction objectif aléatoire	180
3.5.3.3	Amélioration espérée modifiée	180
3.5.3.4	Réduction des incertitudes des modèles DPOD+I	180
3.5.3.5	Résultat de l'optimisation	181
3.5.4	Application aérostructure	181
3.5.4.1	Résultats sans approximation	184
3.5.4.2	Résultats couplage MDF	184
3.5.4.3	Résultats couplage SBO	185
3.5.4.4	Résultats couplage EGMDO	186
3.6	Bilan du Chapitre	187
3.7	Références	189

Objectifs du chapitre

- Proposer un méta-modèle de la MDA en approchant chaque solveur disciplinaire par projection et interpolation.
- Proposer une méthode permettant de générer un DoE pour ces approximations disciplinaires.
- Proposer une réduction des incertitudes des méta-modèles par enrichissement des approximations disciplinaires.
- Proposer une méthode de résolution de MDA.
- Proposer une méthode de résolution d'un problème MDO.
- Appliquer ces méthodes sur des cas test aérostructure.

3.1 Introduction

Nous allons à présent nous intéresser à des problèmes d'optimisation multidisciplinaire et adapter la méthode EGMDO lorsque un grand nombre de variables de couplage intervient. Pour cela, nous proposons de coupler l'optimisation avec la MOR mais sur un cas multidisciplinaire. Notons que nous souhaitons rester dans un cadre différent de l'approche *Offline-Online* afin d'éviter une phase *Offline* trop coûteuse. Ainsi, un enrichissement adaptatif à la volée basé sur une estimation de l'erreur sera proposé. Néanmoins, nous avons vu, dans le chapitre précédent, que l'erreur théorique fournie par la réduction d'ordre de modèle est difficile à utiliser : cette estimation de l'erreur est très intrusive et son calcul pénalise fortement le coût numérique. De plus, l'estimation de la sortie du solveur par projection de Petrov-Galerkin nécessite d'assembler et d'utiliser les données du système linéaire associé rendant cette méthode intrusive et coûteuse. Ainsi, une solution proposée dans la littérature permettant d'obtenir une approximation de la solution d'un tel solveur est d'interpoler les coefficients de la POD afin d'éviter l'assemblage des données (matrice de rigidité et vecteur des efforts extérieurs dans le cas d'un solveur structure). Cependant, l'interpolation rend la définition de l'erreur commise sur la sortie du solveur difficile à calculer. Afin d'illustrer nos propos, la Figure 3.1a représente les déplacements issus d'un solveur structure et une approximation des déplacements en utilisant une stratégie d'interpolation des coefficients sur la base réduite, nommée POD+I. Notons que, comme nous l'avons observé dans le chapitre précédent, l'erreur de projection est souvent très faible et la source de l'erreur sera principalement due à l'interpolation. Cependant, sans estimation de l'erreur commise sur l'interpolation, il n'est pas possible de définir une stratégie d'enrichissement. Pour y pallier, nous proposons d'utiliser les GP conditionnés pour interpoler les coefficients de la POD puisque ceux-ci permettent, à travers la variance du GP, d'obtenir une approximation de l'erreur commise. Cette idée est illustrée dans la Figure 3.1b où un intervalle de confiance est créé à partir de la variance des GP. Nous verrons alors dans ce chapitre, comment l'incertitude sur l'interpolation peut être exploitée afin de proposer une stratégie d'enrichissement. L'une des problématique de ce chapitre concerne les difficultés liées au cadre multidisciplinaire lors de la construction de la base POD et de la construction du DoE nécessaire à l'approximation des coefficients.



(a) Illustration des déplacements issus d'un solveur structure u et son approximation $\hat{u}$ par **POD+I**.
(b) Illustration des déplacements u et de l'encadrement des déplacements issus d'un solveur structure obtenu à partir de la variance des GP.

3.1.1 Objectifs

La **MDO** cherche à résoudre un problème d'optimisation basée sur une **MDA** paramétrée. Ces paramètres, ou variables de design dépendent de paramètres globaux $\mathbf{x}^g \in \mathcal{X}^g \subset \mathbb{R}^{q^g}$ et de paramètres locaux $\mathbf{x}^i \in \mathcal{X}^i \subset \mathbb{R}^{q^i}$ propres à chaque discipline où q^g est le nombre de paramètres globaux et q^i est le nombre de paramètres locaux affectés à la discipline i . L'ensemble des paramètres globaux et locaux $\mathbf{x} = \{\mathbf{x}^g, \mathbf{x}^1, \dots, \mathbf{x}^{n_d}\}$ est noté $\mathbf{x}$ dans la suite et on pose $\mathcal{X} = \mathcal{X}^g \times \prod_{i=1}^{n_d} \mathcal{X}^i$.

L'objectif de ce chapitre est de résoudre le problème d'optimisation multidisciplinaire suivant :

$$\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) = f^{obj}(\mathbf{x}, \mathbf{y}_*^1(\mathbf{x}), \dots, \mathbf{y}_*^{n_d}(\mathbf{x})) \quad (3.1)$$

où $\mathbf{y}_*^1(\mathbf{x}), \dots, \mathbf{y}_*^{n_d}(\mathbf{x})$ est la solution du système couplé coûteux, nommé **MDA** :

$$\left\{ \mathbf{y}^i = f^i(\mathbf{x}^g, \mathbf{x}^i, \mathbf{y}^{c[i]}), i = 1, \dots, n_d \quad \forall \mathbf{x} = \{\mathbf{x}^g, \mathbf{x}^1, \dots, \mathbf{x}^{n_d}\} \in \mathcal{X} \right. \quad (3.2)$$

où f^i est le solveur modélisant la $i^{\text{ème}}$ discipline, $\mathbf{y}^i$ sont les solutions de ces solveurs aussi appelées variables d'état et $\mathbf{y}^{c[i]}$ sont les variables de couplage intervenant sur la $i^{\text{ème}}$ discipline. Ainsi, $\mathbf{c}^{[i]}$ est un sous-ensemble de $\{1, \dots, n_d\}$ dont i ne fait pas partie. Ces variables de couplage sont définies dans des espaces $\mathcal{C}^i \subset \mathbb{R}^{d^i}$ où d^i est la dimension de la variable de couplage $\mathbf{y}^i$. Dans le cas d'un couplage fort, les variables de couplage sont des sorties de grande dimension (champs discrétisés). Rappelons que l'une des difficultés de cette formulation particulière qu'est la **MDO** est que chaque appel à la fonction objectif définie dans l'Eq. (3.1) nécessite de résoudre un système couplé coûteux (**MDA**) défini dans l'Eq. (3.2). Or, notre objectif ici est de résoudre ce problème en minimisant le nombre d'appels aux solveurs disciplinaires dont le coût numérique est prédominant. Pour cela, plusieurs formulations ont été proposées comme la **MDF** (**CRAMER et collab. [1994]**; **BALLING et SOBIESZCZANSKI-SOBIESKI [1996]**), l'**IDF** (**CRAMER et collab. [1994]**; **ALEXANDROV et LEWIS [2004]**). Cependant, nous avons vu dans le Chapitre 1 que ces approches ne permettent pas de relever les défis abordés dans l'introduction. Dans notre étude, nous souhaitons adapter la méthode **EGMDO** en remplaçant les solveurs disciplinaires coûteux par des approximations par **GP** car cette méthode nécessite moins d'appels aux solveurs disciplinaires que les autres lorsque le nombre de variables de couplage est faible. Cependant, cette méthode n'est pas applicable directement puisque le nombre de variables de couplage (la dimension de $\mathbf{y}^i$) est trop important. En effet, les solutions des solveurs sont supposées être des champs discrétisés donc des vecteurs de grande dimension (en pratique, $d^i > 1000$, $i = 1, \dots, n_d$).

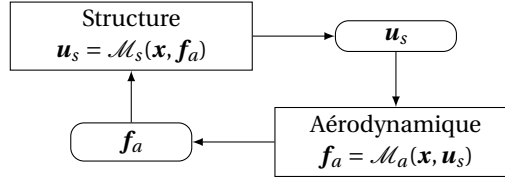


FIGURE 3.2 – Illustration de la FSI où u_s est le déplacement de la structure, f_a sont les efforts aérodynamiques, $\mathcal{M}_a$ est le modèle aérodynamique et $\mathcal{M}_s$ est le modèle structure.

3.1.2 Description globale de la méthode

Pour simplifier, nous considérons dans la suite une MDA composée uniquement de deux disciplines. Notons que les méthodes développées dans la suite peuvent prendre en compte un nombre plus important de disciplines mais nos applications seront limitées à deux disciplines. En effet, le cas d'application de notre étude sera la FSI introduite en Section 1.2.1.2. Dans ce cas, deux disciplines interviennent : la structure permettant le calcul des déplacements et l'aérodynamique permettant le calcul des efforts. À partir du déplacement de la structure, il est possible de définir la géométrie de l'aile déformée permettant d'obtenir les efforts sur l'aile à l'aide d'un modèle aérodynamique. Ces efforts, appliqués sur la structure de l'aile, vont avoir une influence sur le déplacement de l'aile que l'on peut obtenir par un modèle structure. On observe alors un couplage entre ces deux disciplines illustré sur la Figure 3.2.

Le problème d'optimisation devient alors :

$$\min_{x \in \mathcal{X}} f(x) = f^{obj}(x, y_*^1(x), y_*^2(x)) \quad (3.3)$$

où $y_*^1(x)$ et $y_*^2(x)$ sont les solutions de :

$$\begin{cases} y^1 = f^1(x^g, x^1, y^2) \\ y^2 = f^2(x^g, x^2, y^1) \end{cases} \quad (3.4)$$

La méthode proposée est inspirée de la méthode EGMDO, illustrée sur la Figure 3.3, où les disciplines exactes sont remplacées par des GP disciplinaires. Cependant, comme le nombre de variables de couplage est très élevé et ces variables sont des entrées des GP disciplinaires, il n'est pas possible d'entraîner directement ces GP facilement. Une étape supplémentaire de réduction de dimension des variables de couplage est proposée. Le Chapitre 2 ayant mis en évidence la capacité de la POD (Section 1.3.3.1) à approcher un solveur disciplinaire par un nombre réduit de coefficients sur une base réduite, nous proposons d'utiliser cette méthode pour la réduction de dimension. Cette idée est illustrée sur la Figure 3.4 où la première étape consiste à projeter les disciplines sur les bases POD afin de réduire la dimension des variables de couplage et où la deuxième étape est d'interpoler les coefficients des disciplines réduites. Ces modèles approchés de chaque discipline, nommés *Décomposition orthogonale aux valeurs propres et interpolation disciplinaire* *Disciplinary Proper Orthogonal Decomposition and Interpolation* (DPOD+I), sont obtenus en projetant les entrées et sorties des solveurs disciplinaires (les variables de couplage) et en interpolant les coefficients des bases POD.

3.1.3 Organisation du Chapitre

Dans un premier temps, nous allons laisser de côté le problème d'optimisation pour nous concentrer sur la construction des bases POD et sur l'interpolation des coefficients.

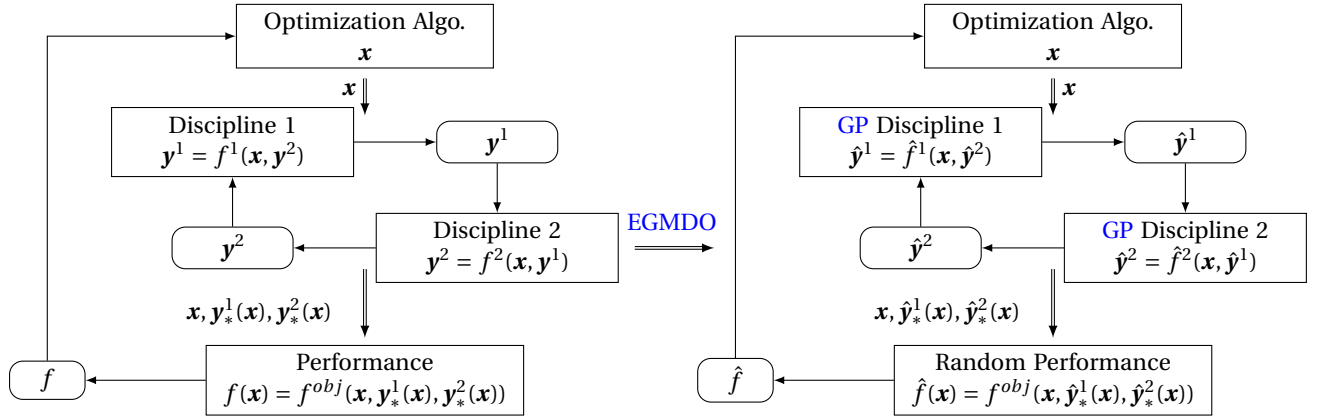


FIGURE 3.3 – Illustration de la méthode **EGMDO** où la **MDA** déterministe est remplacée par une **MDA** aléatoire en substituant les deux solveurs disciplinaires coûteux par des approximations par **GP**.

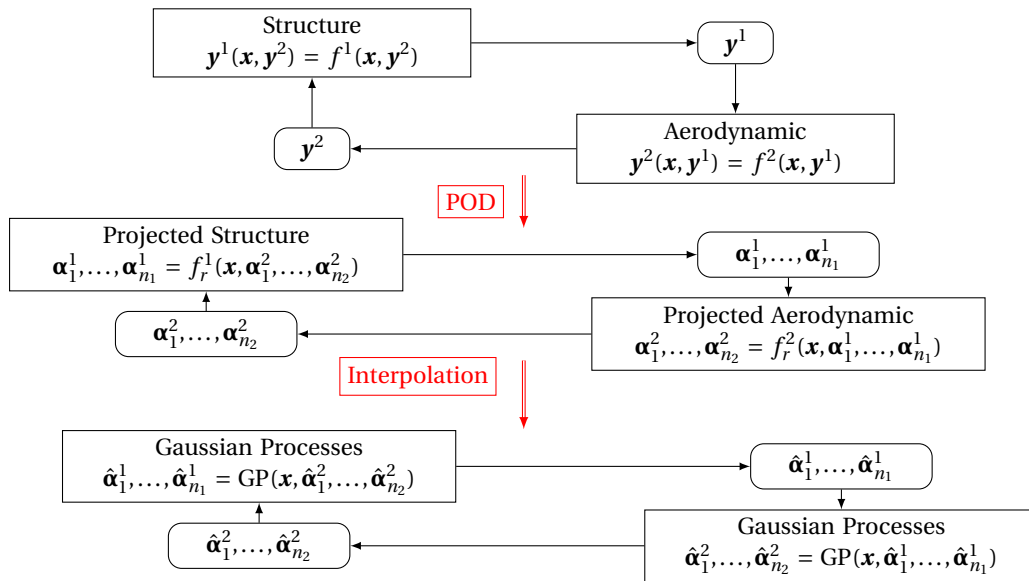


FIGURE 3.4 – Illustration de la méthode mise en oeuvre pour l'interpolation des disciplines où la dimension des variables de couplage est réduite par **POD**.

Pour cela, nous proposons d'approcher chaque discipline grâce à une approximation par **POD** et d'interpoler les coefficients par **GP**. Ces modèles **DPOD+I** seront couplés afin d'obtenir une approximation de la **MDA**. Nous proposerons alors une méthode permettant d'entraîner les approximations disciplinaires. Par la suite, la précision de la **MDA** approchée sera obtenue par propagation des incertitudes induites par les approximations disciplinaires. Cette incertitude sera réduite par une stratégie d'enrichissement adaptatif jusqu'à ce que l'approximation par **DPOD+I** soit considérée comme suffisamment précise. Nous analyserons alors les performances de cette méthode sur deux cas tests **FSI**.

Dans un second temps, nous verrons comment nous allons coupler la méthode d'approximation de la **MDA** dans le cadre de l'optimisation multidisciplinaire. Pour cela, nous proposerons notamment d'utiliser la méthode **EGMDO** puisque le nombre de variables de couplage aura été réduit grâce à la **DPOD+I** et d'exploiter le modèle ainsi défini pour résoudre le problème d'optimisation. Nous appliquerons alors la méthode ainsi développée dans le cadre de la **FSI**.

3.2 Description de la résolution de la **MDA** par **DPOD+I**

Dans un premier temps, un focus sur la résolution de la **MDA** paramétrée grâce aux modèles approchés est proposé. La résolution de la **MDO** sera présentée dans la suite de la thèse.

3.2.1 Aperçu de la méthodologie proposée

Cette section présente un aperçu global des différentes étapes impliquées dans la méthode d'approximation proposée. Ensuite, dans les sections suivantes, toutes les étapes seront décrites plus en détails. La Figure 3.5 illustre ces différentes étapes, S_i représente les étapes de calcul et O_i représente les sorties correspondantes.

Comme mentionné dans l'introduction, la méthode proposée vise à résoudre la **MDA** pour différentes valeurs de variables de design (par exemple générées par un optimiseur dans le cadre de la **MDO** où par **MCS** pour la fiabilité). Dans la Figure 3.5, cet ensemble de valeurs est indiqué par $\mathbb{X}_{DoE} = \{\mathbf{x}_i, i = 1, \dots, n_{sim}\}$. L'originalité de l'approche proposée réside dans la construction et l'enrichissement de méta-modèles disciplinaires. Ces méta-modèles disciplinaires, nommés **DPOD+I**, sont basés sur une combinaison de **POD** et d'interpolation par **GP**.

La Figure 3.5 est composée de deux blocs faisant référence aux deux parties principales de l'approche. Le premier, nommé "Initialisation et apprentissage des modèles **DPOD+I**", sera détaillé dans les Sections 3.2.2 et 3.2.3. Le second, nommé "Résolution de la **MDA** pour l'ensemble des variables de design $\mathbf{x} \in \mathbb{X}_{DoE}$ ", sera détaillé dans les Sections 3.2.4 et 3.3.

Comme présenté dans la Figure 3.5, la première étape de la méthode proposée est la construction des bases **POD** disciplinaires et du **DoE** correspondant. Cette étape est désignée par S_1 dans la Figure 3.5 et décrite en détail dans la Section 3.2.3. La difficulté de l'entraînement réside dans le couplage entre les solveurs disciplinaires, par conséquent une procédure d'apprentissage dédiée est proposée.

Les sorties de cette première étape sont des méta-modèles disciplinaires globaux **DPOD+I** (notés O_1 sur la Figure 3.5). Ces méta-modèles sont appelés *globaux* car ils sont destinés à être utilisés pour tous les $\mathbf{x}_i$ sur tout l'espace de design $\mathcal{X}$. Il est à noter que nous choisissons d'utiliser une base **POD** globale car les exemples numériques présentés ci-après sont

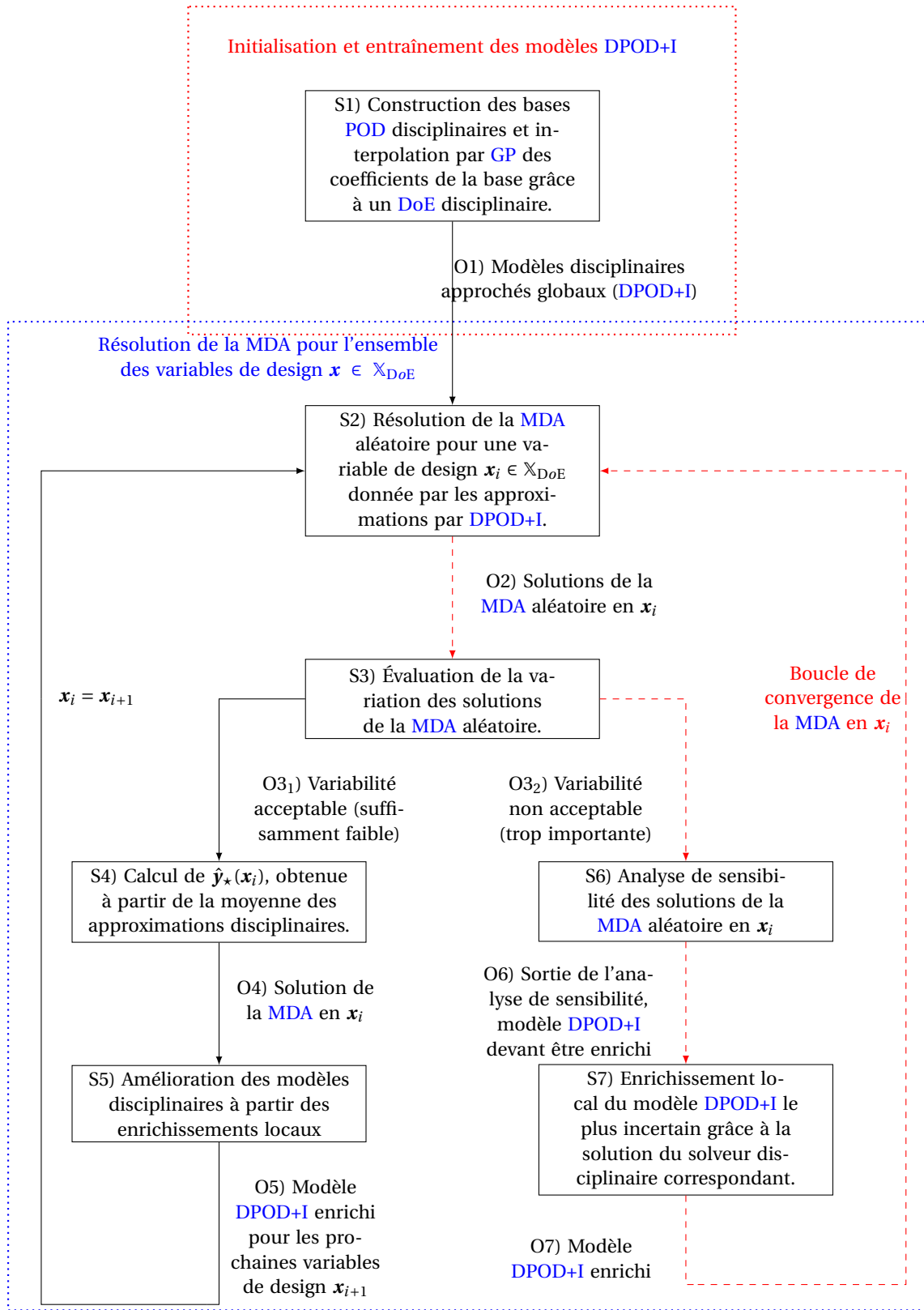


FIGURE 3.5 – Illustration des principales étapes de la méthodologie proposée. S_i correspond aux étapes de la méthode et O_i correspond aux sorties des différentes étapes.

bien approchés par une approche globale. De plus, cette base **POD** globale est plus adaptée aux approximations **POD+I** car chaque observation peut être utilisée pour entraîner les modèles d'interpolation. À noter que pour des simulations plus complexes, on peut remplacer cette approximation globale par des bases **POD** locales et une interpolation comme proposée par exemple dans **DUPUIS et collab. [2018]**. La Section 3.2.2 présente les méta-modèles disciplinaires utilisés dans ce travail et la **MDA** réduite obtenue.

Une fois que ces modèles **DPOD+I** initiaux sont construits, ils sont utilisés pour résoudre la **MDA** en un point $\mathbf{x}_i \in \mathbb{X}_{\text{DoE}}$. Cette résolution, notée S2 dans la Figure 3.5, fait intervenir une **MDA** aléatoire. Le caractère aléatoire provient du fait que l'interpolation par **GP** est utilisée pour les méta-modèles. Les détails sur cette **MDA** aléatoire et sa résolution sont fournis dans la Section 3.2.4.1.

La sortie de S2, notée O2, est une solution aléatoire du problème **MDA** en $\mathbf{x}_i$. Cette solution aléatoire est analysée à l'étape S3 afin de décider si les modèles **DPOD+I** sont suffisamment précis pour approcher la solution de la **MDA** en $\mathbf{x}_i$. Si c'est le cas, étape O3₁ dans la Figure 3.5, alors la solution de la **MDA** est obtenue en résolvant la **MDA** approchée utilisant la valeur moyenne des modèles **DPOD+I** (étape S4). Si les méta-modèles disciplinaires ne sont pas assez précis, étape O3₂ dans la Figure 3.5, alors la variabilité de la solution aléatoire est importante et les méta-modèles disciplinaires doivent être améliorés. Le critère pour quantifier la dispersion de la solution aléatoire est présenté dans la Section 3.2.4.2.

Dans ce dernier cas, il est proposé d'effectuer une analyse de sensibilité de la solution de la **MDA** aléatoire, lors de l'étape S6 de la Figure 3.5, afin de décider quel modèle **DPOD+I** doit être enrichi (sortie O6). Cette analyse de sensibilité est suivie de l'étape d'enrichissement S7. Cet enrichissement est noté *local* car il est effectué pour réduire l'incertitude de la solution aléatoire en $\mathbf{x}_i$. Le processus d'enrichissement proposé est décrit dans la Section 3.2.5.1. Après avoir ajouté un point au **DoE** disciplinaire concerné, il est prévu que le modèle **DPOD+I** correspondant soit amélioré (sortie O7 dans la Figure 3.5). Il est à noter que les bases **POD** disciplinaires peuvent également être améliorées lors de cette étape. Ensuite, comme illustré dans la Figure 3.5, la **MDA** aléatoire peut être résolue à nouveau en $\mathbf{x}_i$. Cette boucle de convergence (lignes rouges en pointillé sur la Figure 3.5) est effectuée au point de design $\mathbf{x}_i$ jusqu'à ce que la variabilité de la solution **MDA** aléatoire soit suffisamment faible.

Une fois que la variabilité de la solution est considérée comme acceptable à l'étape S3, la solution de la **MDA** est calculée en utilisant la valeur moyenne des modèles **DPOD+I** (étape S4). Les modèles **DPOD+I** disciplinaires ayant évolué (par enrichissement) au cours de la boucle de convergence pour $\mathbf{x}_i$, il est intéressant de mettre à jour les modèles **DPOD+I** qui seront utilisés pour le prochain point de design $\mathbf{x}_{i+1}$. Cependant, afin d'augmenter la robustesse de l'approche et d'éviter les problèmes de mauvais conditionnement lors de la construction du **GP**, il est nécessaire de filtrer le **DoE** comme présenté à l'étape S5. Cette étape de filtrage génère des modèles **DPOD+I** améliorés (c'est-à-dire enrichis à l'aide d'une partie de solutions d'enrichissement locales) qui peuvent être utilisés pour résoudre la **MDA** au point de design suivant et sera détaillée dans la Section 3.2.5.2.

3.2.2 Approximation des solveurs disciplinaires par **POD+I**

3.2.2.1 Approximation par **POD** et **MDA** réduite

Les variables de couplage dans le cadre de notre étude sont les sorties de solveurs numériques représentant des grandeurs physiques et une réduction de leurs dimensions est nécessaire. Pour cela, la réduction d'ordre de modèle que nous avons présenté dans le

chapitre précédent est utilisée. Nous avons vu dans ce cas que via un très faible nombre de modes **POD** il est possible d'approcher la sortie d'un solveur physique réduisant drastiquement la dimension des sorties. De la même façon que dans le chapitre précédent nous allons utiliser la **POD** car celle-ci a démontré son efficacité sur des problèmes mécanique et aérodynamique, linéaire (**ZAHM et collab. [2017]**) ou non-linéaire (**SIROVICH [1987]**; **WANG et collab. [2011]**).

L'utilisation de la **POD** sera décrite sur la première discipline. Cependant, la même méthodologie sera appliquée sur la deuxième discipline. Pour rappel, la **POD** cherche à obtenir une approximation de chaque solveur disciplinaire sur un espace affine :

$$\mathbf{y}^1(\mathbf{x}, \mathbf{y}^2) \approx \tilde{\mathbf{y}}^1(\mathbf{x}, \mathbf{y}^2) = \boldsymbol{\Phi}_0^1 + \sum_{j=1}^{n^1} \alpha_j^1(\mathbf{x}, \mathbf{y}^2) \boldsymbol{\Phi}_j^1 \quad (3.5)$$

où $\tilde{\mathbf{y}}^1 \in \mathbb{R}^{d_1}$ désigne l'approximation du vecteur $\mathbf{y}^1 \in \mathcal{C}^1$ par **POD**, $\boldsymbol{\Phi}_j^1 \in \mathbb{R}^{d_1}$ est le $j^{\text{ème}}$ vecteur de la matrice $\boldsymbol{\Phi}^1$ et $\boldsymbol{\Phi}_0^1$ est un vecteur constant. La base **POD** pour le premier solveur est caractérisée par la matrice de projection $\boldsymbol{\Phi}^1$ et le vecteur $\boldsymbol{\Phi}_0^1$. Les coefficients $\alpha_j^1 \in \mathbb{R}$ associés au vecteur $\mathbf{y}^1$ sont obtenus par projection : $\alpha_j^1 = \langle \mathbf{y}^1 - \boldsymbol{\Phi}_0^1, \boldsymbol{\Phi}_j^1 \rangle$. Le nombre n^1 de termes est choisi selon le critère défini pour la construction des bases **POD** décrit dans la Section 1.3.3.1.

Pour simplifier les notations, nous allons définir les applications de projection et de reconstruction sur la base **POD** permettant la réduction de la dimension du problème. L'application p^1 permettant d'obtenir les coefficients sur la base **POD** est définie par :

$$\begin{aligned} p^1 : \mathcal{C}^1 \subset \mathbb{R}^{d_1} &\rightarrow \tilde{\mathcal{C}}^1 \subset \mathbb{R}^{n^1} \\ \mathbf{y}^1 &\mapsto p^1(\mathbf{y}^1) = \{\alpha_1^1, \dots, \alpha_{n^1}^1\} = (\boldsymbol{\Phi}^1)^T (\mathbf{y}^1 - \boldsymbol{\Phi}_0^1) \end{aligned} \quad (3.6)$$

où $\tilde{\mathcal{C}}^1$ est l'image de $\mathcal{C}^1$ par p^1 . L'application réciproque de p^1 , notée p^{-1} , permettant la reconstruction à partir des coordonnées sur la base **POD** est définie par :

$$\begin{aligned} p^{-1} : \tilde{\mathcal{C}}^1 &\rightarrow \mathcal{C}^1 \\ \alpha_1^1, \dots, \alpha_{n^1}^1 &\mapsto p^{-1}(\alpha_1^1, \dots, \alpha_{n^1}^1) = \boldsymbol{\Phi}_0^1 + \sum_{j=1}^{n^1} \alpha_j^1 \boldsymbol{\Phi}_j^1 \end{aligned} \quad (3.7)$$

Par la suite, la même méthodologie est appliquée pour la seconde discipline permettant de définir une base **POD** pour la deuxième discipline. Cette base est caractérisée par une matrice de projection $\boldsymbol{\Phi}^2$ contenant les n^2 vecteurs $\{\boldsymbol{\Phi}_j^2\}_{j=1}^{n^2}$ et par un vecteur moyen $\boldsymbol{\Phi}_0^2$. Il est alors possible de définir les applications p^2 et p^{-2} de la même façon :

$$\begin{aligned} p^2 : \mathcal{C}^2 \subset \mathbb{R}^{d_2} &\rightarrow \tilde{\mathcal{C}}^2 \subset \mathbb{R}^{n^2} \\ \mathbf{y}^2 &\mapsto p^2(\mathbf{y}^2) = \{\alpha_1^2, \dots, \alpha_{n^2}^2\} = (\boldsymbol{\Phi}^2)^T (\mathbf{y}^2 - \boldsymbol{\Phi}_0^2) \end{aligned} \quad (3.8)$$

où $\tilde{\mathcal{C}}^2$ est l'image de $\mathcal{C}^2$ par p^2 et

$$\begin{aligned} p^{-2} : \tilde{\mathcal{C}}^2 &\rightarrow \mathcal{C}^2 \\ \alpha_1^2, \dots, \alpha_{n^2}^2 &\mapsto p^{-2}(\alpha_1^2, \dots, \alpha_{n^2}^2) = \boldsymbol{\Phi}_0^2 + \sum_{j=1}^{n^2} \alpha_j^2 \boldsymbol{\Phi}_j^2 \end{aligned} \quad (3.9)$$

Nous pouvons ainsi, grâce aux bases **POD**, définir les variables de couplage réduites $\boldsymbol{\alpha}^1 \in \mathbb{R}^{n^1}$ et $\boldsymbol{\alpha}^2 \in \mathbb{R}^{n^2}$ via :

$$\begin{aligned} \boldsymbol{\alpha}^1 &= (\alpha_1^1, \dots, \alpha_{n^1}^1)^T \\ \boldsymbol{\alpha}^2 &= (\alpha_1^2, \dots, \alpha_{n^2}^2)^T \end{aligned} \quad (3.10)$$

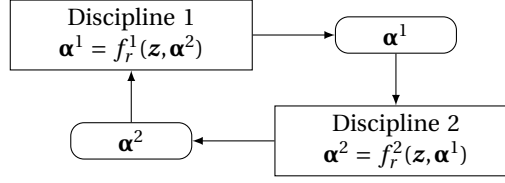


FIGURE 3.6 – Illustration de l'analyse multidisciplinaire réduite utilisant les variables de couplage projetées obtenues par **POD**.

En résumé, l'utilisation de la **POD** dans ce contexte permet de remplacer les variables de couplage de grande dimension $\mathbf{y}^1$ et $\mathbf{y}^2$ par des variables de couplage de faible dimension α^1 et α^2 . Notons que la dimension des variables de couplage réduites (égales aux nombres de modes **POD** conservés) est souvent inférieure à 20.

Nous allons alors introduire la **MDA** réduite utilisant les variables de couplage réduites :

$$\begin{cases} \alpha^1(\mathbf{x}, \alpha^2) = f_r^1(\mathbf{x}, \alpha^2) \\ \alpha^2(\mathbf{x}, \alpha^1) = f_r^2(\mathbf{x}, \alpha^1) \end{cases} \quad (3.11)$$

où $f_r^1(\mathbf{x}, \alpha^2) = p^1(f^1(\mathbf{x}, p^{-2}(\alpha^2)))$ et $f_r^2(\mathbf{x}, \alpha^1) = p^2(f^2(\mathbf{x}, p^{-1}(\alpha^1)))$ sont les solveurs disciplinaires réduits. Cette **MDA** réduite est illustrée dans la Figure 3.6.

3.2.2.2 Interpolation des solveurs réduits par Processus Gaussiens

Dans cette section, nous allons introduire l'interpolation des solveurs disciplinaires réduits f_r^1 et f_r^2 définis dans la **MDA** réduite (cf. Figure 3.6).

Afin de distinguer les variables aléatoires des variables déterministes, nous allons utiliser des accents de type chapeau $\hat{\cdot}$ pour désigner les variables aléatoires. Ainsi, chaque coefficient construit grâce à la base **POD** est approché à l'aide d'un **GP** :

$$\begin{aligned} \hat{\alpha}_i^1(\mathbf{x}, \alpha^2) & \text{ pour } \alpha_i^1(\mathbf{x}, \alpha^2) \quad \forall i = 1, \dots, n^1 \\ \hat{\alpha}_j^2(\mathbf{x}, \alpha^1) & \text{ pour } \alpha_j^2(\mathbf{x}, \alpha^1) \quad \forall j = 1, \dots, n^2 \end{aligned} \quad (3.12)$$

où $\hat{\alpha}_i^1$ et $\hat{\alpha}_j^2$ sont des **GP** dont une présentation a été effectuée dans la Section 3.2.3. Ces **GP** sont caractérisés par leurs moyennes et leurs covariances :

$$\begin{aligned} \hat{\alpha}_i^1(\mathbf{x}, \alpha^2) &= \mu_i^1(\mathbf{x}, \alpha^2) + \hat{\epsilon}_i^1(\mathbf{x}, \alpha^2) \quad \forall i = 1, \dots, n^1 \\ \hat{\alpha}_j^2(\mathbf{x}, \alpha^1) &= \mu_j^2(\mathbf{x}, \alpha^1) + \hat{\epsilon}_j^2(\mathbf{x}, \alpha^1) \quad \forall j = 1, \dots, n^2 \end{aligned} \quad (3.13)$$

où μ_i^1 et μ_j^2 sont les moyennes des **GP** ; $\hat{\epsilon}_i^1$ et $\hat{\epsilon}_j^2$ sont des **GP** de moyenne nulle et dont la fonction de covariance est celle du **GP** correspondant.

Cependant, la construction de ces **GP** représente un challenge. En effet, pour construire un **DoE** pour l'entraînement des interpolations des coefficients de la **POD**, il est nécessaire d'obtenir un échantillon sur l'espace de design et sur l'espace des variables de couplage. Néanmoins, échantillonner sur les espaces des variables de couplage est une tâche ardue puisqu'aucune information sur ces espaces n'est disponible sans effectuer des résolutions de **MDA**. De plus, ces espaces étant de grande dimension, une approximation de ces espaces est difficilement envisageable. Ainsi, une méthode d'échantillonnage, exploitant la structure de la **MDA**, est proposée dans la Section 3.2.3.

3.2.3 Entraînement des processus gaussiens

Dans cette section, nous allons détailler la méthodologie permettant d'obtenir les informations utilisées pour la construction des **GP** interpolant les coefficients de la **POD**.

Dans un premier temps, nous allons expliquer pourquoi générer les données à partir de solutions de la MDA exacte ne permet pas de répondre à cette problématique. Pour appliquer cette méthodologie, un DoE contenant p^{DoE} éléments sur l'espace paramétrique $\mathcal{X}$ est généré : $\mathbf{x}_i$, $i = 1, \dots, p^{\text{DoE}}$. Notons que cette étape est triviale dans le cadre de notre étude puisque l'espace de design est un hypercube et plusieurs méthodes (notamment les LHS MCKAY et collab. [1979]) permettent de générer des points dans cet espace. Par la suite, les solutions des MDA pour chacun de ces paramètres sont calculées à l'aide d'un solveur (type Jacobi comme cela est détaillé dans la Section 1.2.1.1). Pour chacun de ces paramètres, plusieurs itérations et donc appels aux solveurs disciplinaires sont nécessaires. Ainsi, les informations apportées par chaque appel sont utilisées pour générer les bases POD ainsi que les plans d'expérience pour l'entraînement des GP. Après la construction des bases POD, en projetant les entrées et les sorties de chaque solveur disciplinaire, on définit un point d'entraînement pour l'apprentissage de la discipline réduite. Cependant, cette méthode est peu efficace pour obtenir un DoE bien réparti sur l'espace des variables de couplage. En effet, lors de la résolution d'une MDA, la variable de design est constante (égale à $\mathbf{x}_i$) et les variables de couplage $\mathbf{y}^1$ et $\mathbf{y}^2$ convergent vers la solution de la MDA. Ainsi, à chaque itération, les variables de couplage sont de plus en plus proches de la solution de la MDA. La répartition des variables de couplage et des variables de design obtenues en utilisant cette méthode est illustrée sur la Figure 3.7 où la première coordonnée de chaque base POD ainsi que la première variable de design $\mathbf{x}_1$ sont représentées. Une possible alternative, notamment proposée par COELHO et collab. [2008], est d'utiliser des MDA non convergées. Néanmoins, cette solution rend les premiers calculs difficiles à utiliser. En effet, les conditions initiales étant inchangées dans les algorithmes de résolution des MDA, cela impose que le premier calcul disciplinaire (dans le cas d'un algorithme de Gauss-Seidel) ou tous les premiers calculs disciplinaires (dans le cas d'un algorithme de Jacobi) ont des variables de couplage fixes. Ainsi, pour entraîner des méta-modèles, il est nécessaire d'effectuer plus d'une itération rendant cette méthode moins efficace. De plus, comme le DoE n'est pas uniformément réparti, cela pourrait conduire à un GP mal conditionné. En effet, lorsque les points d'apprentissage sont trop proches les uns des autres, la matrice de covariance associée au noyau de corrélation k (voir Eq. (1.20)) n'est plus numériquement inversible car certaines colonnes sont quasiment identiques (voir MOHAMMADI et collab. [2016] pour plus de détails).

Pour y remédier, nous proposons une méthodologie spécifique pour l'entraînement des méta-modèles exploitant une décorrélation entre les variables de design et les variables de couplage. Ainsi, l'objectif sera d'obtenir un DoE propice à l'entraînement des méta-modèles, c'est à dire un DoE dont les points sont bien répartis. En pratique, l'obtention d'un tel DoE est rendu difficile par plusieurs points. Premièrement, les espaces des variables de couplage $\mathcal{C}^1$ et $\mathcal{C}^2$ sont inconnus donc il est impossible d'utiliser des méthodes classiques d'échantillonnage telles que les LHS. De plus, les espaces des variables de couplage sont de très grande dimension rendant la construction d'une approximation de ceux-ci très difficile. Une solution serait d'approcher les espaces des variables de couplage projetées. Néanmoins, ces espaces ne sont pas triviaux et il existe des structures de corrélations entre les coefficients de la POD comme nous pouvons l'observer dans la Figure 3.8. Pour cette illustration, 200 MDA ont été résolues, tous les calculs disciplinaires ont alors été utilisés pour la génération de la base POD. Les coefficients, obtenus par projection, sont affichés sur cette figure. Nous pouvons notamment observer une forte structure de dépendance entre le premier et le deuxième coefficient même si le coefficient de corrélation entre ces deux variables est nul (ceci est une des propriétés de la SVD). Ainsi, il sera difficile de construire une approximation de ces espaces réduits. Pour résoudre ce

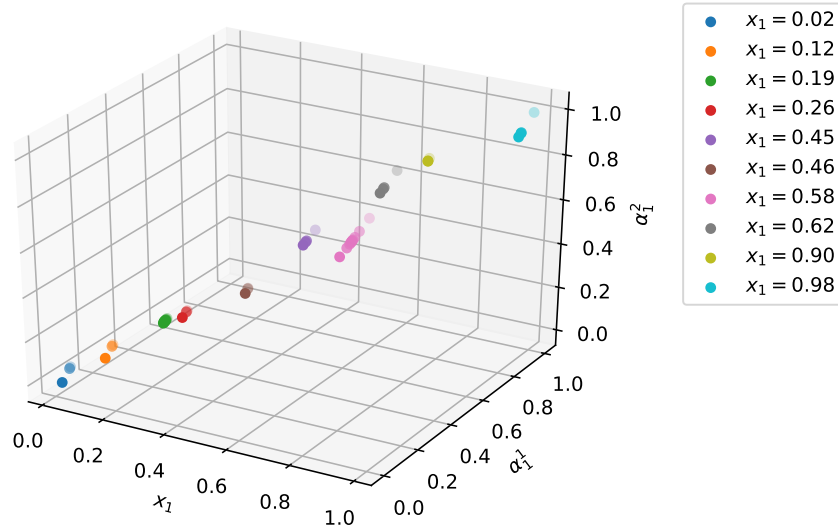


FIGURE 3.7 – Distribution des entrées de chaque solveur disciplinaire lors de la résolution de 10 MDA associées à 10 paramètres de design. Seuls les premiers coefficients de la POD (α_1^1 and α_1^2) sont représentés. Chaque MDA est représentée par une couleur différente, la résolution de chacune des 10 MDA a nécessité quatre appels à chaque solveur disciplinaire.

problème, une nouvelle méthodologie, inspirée des algorithmes itératifs de résolution de la MDA, va être mise en oeuvre. Elle est détaillée dans les Sections 3.2.3.1 et 3.2.3.2.

3.2.3.1 Initialisation de l'algorithme d'entraînement des processus gaussiens

En premier lieu, rappelons que pour effectuer un appel à un solveur disciplinaire, une valeur initiale des solutions des autres solveurs est nécessaire. Ainsi, le premier enjeu de cette méthode est de générer un échantillon de valeurs initiales de taille r . Pour cela, nous proposons de tirer ces valeurs initiales dans une variété de dimension 1 à partir de la valeur initiale de l'algorithme de la résolution des MDA notée $\mathbf{y}_0^1$ (solution de la MDA au centre de $\mathcal{X}$). Cette variété est définie par deux bornes λ^- et λ^+ : $Y^1 = \{\lambda \mathbf{y}_0^1, \lambda \in [\lambda^-; \lambda^+]\}$. Ces bornes sont choisies de telle sorte que chaque élément de Y^1 est susceptible d'être un élément de $\mathcal{C}^1$. Pour cela, une analyse physique du problème est nécessaire. Par exemple, dans le cas d'un problème aérostructure, le déplacement maximal d'une aile est compris entre zéro et une valeur donnée par une considération physique sur la structure de l'aile. Par la suite, l'échantillon de $\mathbf{y}_0^1$ est utilisé pour générer un échantillon pour l'autre discipline comme cela est proposé dans l'Algorithme 11. Les étapes de cet algorithme sont :

1. Générer r échantillons dans l'intervalle $[\lambda^-; \lambda^+]$ en utilisant une méthode d'échantillonnage classique.
2. Les variables de couplage issues de la première discipline sont alors obtenues en multipliant les échantillons de $[\lambda^-; \lambda^+]$ par $\mathbf{y}_0^1$. Les variables de couplage ainsi obtenues sont notées $(\mathbf{y}^1)_i^0 = \lambda_i \mathbf{y}_0^1$, $i = 1, \dots, r$.
3. Pour obtenir un échantillon de solutions de la deuxième discipline, les variables de couplage préalablement calculées sont utilisées. En ce qui concerne les variables de design associées $\mathbf{x}_i$, elles sont générées par une méthode d'échantillonnage type LHS. On obtient alors les entrées du deuxième solveur en combinant aléatoirement les variables de couplage et les variables de design : $(\mathbf{y}^2)_i^0 = f_2(\mathbf{x}_i, (\mathbf{y}^1)_i^0)$.

Remarque :

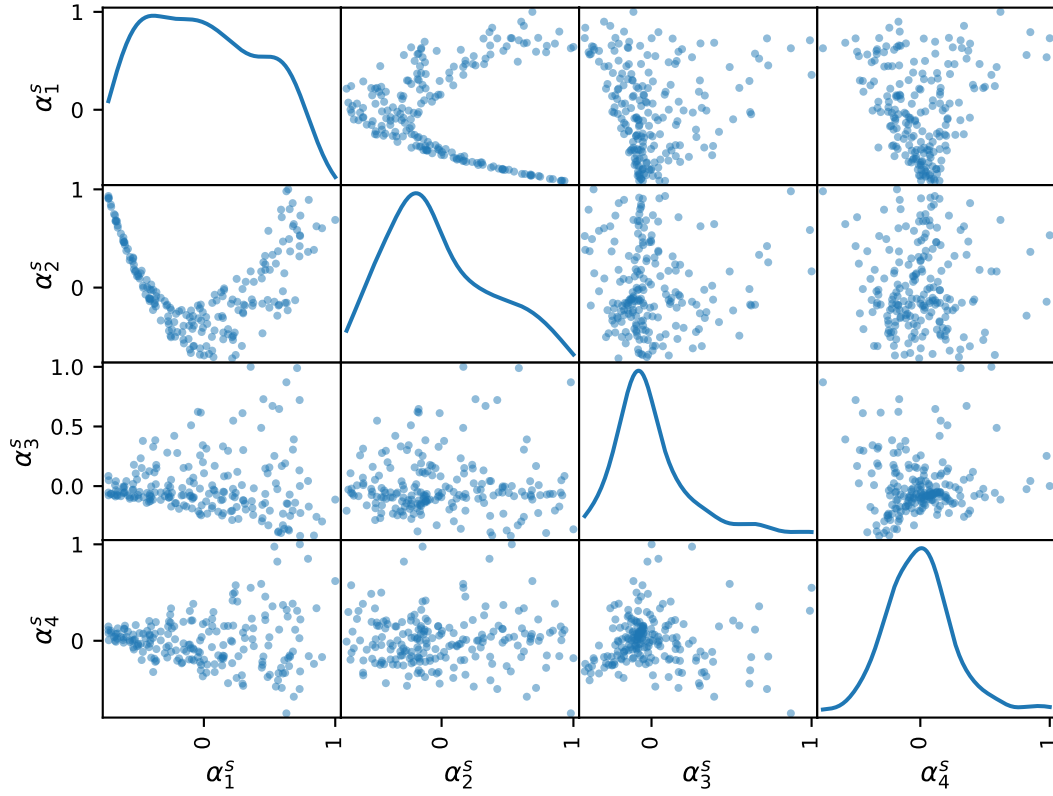


FIGURE 3.8 – Coefficients de la **POD** sur la base structure obtenus à partir des itérations nécessaires à la résolution de 200 **MDA**.

- On peut penser que l'impact des premières valeurs initiales sera négligeable sur le résultat de l'entraînement global à cause des itérations présentes dans la suite du processus, qui sera décrit plus loin.
- Les appels aux solveurs effectués lors de cet algorithme ne seront pas utilisés pour l'entraînement des **GP** puisque leur pertinence est incertaine.

Algorithm 11: Compute r samples of coupling variables.

input : An initial guess $\mathbf{y}_0^1$, a number $r > 0$, bounds λ^- and λ^+
 $\{\lambda_i, i = 1, \dots, r\} \leftarrow r$ samples from $[\lambda^-; \lambda^+]$ obtained by LHS;
for $i = 1, \dots, r$ **do**
 $(\mathbf{y}^1)_i^0 \leftarrow \lambda_i \mathbf{y}_0^1$;
end
 $\{\mathbf{x}_i, i = 1, \dots, r\} \leftarrow$ containing r samples from $\mathcal{X}$ obtained by LHS;
for $i = 1, \dots, r$ **do**
 $(\mathbf{y}^2)_i^0 \leftarrow f_2(\mathbf{x}_i, (\mathbf{y}^1)_i^0)$;
end
output: $\{(\mathbf{y}^1)_i^0, i = 1, \dots, r\}, \{(\mathbf{y}^2)_i^0, i = 1, \dots, r\}$

3.2.3.2 Algorithme d'entraînement

Les objectifs sont désormais de construire, à partir des valeurs initiales, une base **POD** et un **DoE** pour les **GP** disciplinaires. Il est alors proposé de générer des groupes de solutions disciplinaires à l'aide des solutions issues de l'Algorithme 11. Dans le cadre de nos

travaux, l'objectif est principalement d'obtenir une base **POD** performante puisque les **GP** disciplinaires seront enrichis lors d'une phase de résolution. Cette performance sera évaluée sur des solutions indépendantes de celles utilisées pour sa construction. La base réduite sera considérée comme suffisamment performante lorsque l'erreur relative de projection (similaire à celle utilisée dans l'Algorithme 10) est inférieure à une tolérance ϵ_{PE} choisie par l'utilisateur.

Les étapes de cette méthode, appelée Algorithme 12 dans la suite, sont :

1. Initialisation de l'erreur de projection, du compteur d'itérations n et des bases **POD** à partir des valeurs initiales générées via l'Algorithme 11. Notons que la base **POD** associée à la première discipline n'a qu'un seul mode puisque chaque élément de $\{(\mathbf{y}^1)_i^0, i = 1, \dots, r\}$ est colinéaire à $\mathbf{y}_0^1$. Il est donc peu probable que l'algorithme converge en une itération puisque cette base ne sera pas suffisamment précise.
2. Calcul des échantillons de chaque discipline jusqu'à ce que l'erreur relative de projection moyenne soit inférieure à ϵ_{PE} . À cette fin, un échantillon de variables de design est généré par **LHS** : $\{(\mathbf{x}^1)_i^n, i = 1, \dots, r\}$ et associé à un échantillon $\{(\mathbf{y}^2)_i^{n-1}, i = 1, \dots, r\}$ afin d'obtenir les entrées du premier solveur. Pour chacune des r entrées, la solution exacte du solveur est calculée. L'erreur de projection sur la base **POD** de $(\mathbf{y}^1)_i^n$ est calculée avant l'enrichissement de la base **POD**. La moyenne de l'erreur sur la première discipline est calculée afin d'évaluer le critère d'arrêt de l'algorithme.
3. Dans un second temps, des solutions pour le deuxième solveur sont générées à partir d'un échantillon de variables de design $\{(\mathbf{x}^2)_i^n, i = 1, \dots, r\}$ généré par **LHS**. Chaque variable de design est associée à la solution du premier solveur calculée à l'étape précédente $\{(\mathbf{y}^1)_i^n, i = 1, \dots, r\}$. L'erreur relative de projection est calculée pour chaque solution du solveur afin de calculer la moyenne des erreurs relatives pour la deuxième discipline.
4. Les bases **POD** enrichies sont alors générées par **SVD** en utilisant toutes les solutions préalablement calculées à l'exception des variables de couplage initiales $\{(\mathbf{y}^1)_i^0, i = 1, \dots, r\}$ et $\{(\mathbf{y}^2)_i^0, i = 1, \dots, r\}$.
5. Si le critère sur l'erreur relative de projection sur les deux disciplines est atteint, l'algorithme est arrêté et l'on suppose que les bases **POD** obtenues sont suffisamment robustes. Notons qu'un critère lors de la résolution de la **MDA** permettra d'améliorer la base réduite si cela est jugé nécessaire. Les entrées et sorties de chaque solveur sont projetées sur les bases **POD** afin de définir un **DoE** pour chaque discipline réduite f_r^1 et f_r^2 . $\{(\mathbf{x}^1)_i^j, p^2((\mathbf{y}^2)_i^{j-1})\} \rightarrow p^1((\mathbf{y}^1)_i^j), i = 1, \dots, r, j = 1, \dots, n$ sont des observations de f_r^1 et $\{(\mathbf{x}^2)_i^j, p^1((\mathbf{y}^1)_i^j)\} \rightarrow p^2((\mathbf{y}^2)_i^j), i = 1, \dots, r, j = 1, \dots, n$ sont des observations de f_r^2 .

Cette méthode permet d'obtenir une base **POD** pour chaque discipline et un **DoE** pour l'entraînement des **GP** disciplinaires. La répartition des données du **DoE** via cette méthode est illustrée dans la Figure 3.9 où les entrées pour chaque discipline sont représentées. Seuls la première variable de design et le premier coefficient de chaque base **POD** figurent. Notons que, par comparaison avec la Figure 3.7, le **DoE** obtenu est mieux réparti permettant d'améliorer les **GP** disciplinaires.

Remarque :

Aucune **MDA** exacte n'a été calculée lors de cet algorithme d'entraînement. Ainsi, il est impossible de certifier la robustesse de la base sur les espaces de définition des variables de couplage et de la même façon sur les solutions de la **MDA**. Un critère d'enrichissement des bases sera proposé dans le contexte de la recherche de la solution au problème **MDA** pour s'assurer de la performance des bases **POD**.

Algorithm 12: Compute a **POD** basis and a training set for the **GP**.

input : $\{(\mathbf{y}^1)_i^0, i = 1, \dots, r\}, \{(\mathbf{y}^2)_i^0, i = 1, \dots, r\}$, a tolerance over the projection error ϵ_{PE}

projection error $\leftarrow 1 + \epsilon_{\text{PE}}$;
 $p^1 \leftarrow$ **POD** basis built from $\{(\mathbf{y}^1)_i^0, i = 1, \dots, r\}$;
 $p^2 \leftarrow$ **POD** basis built from $\{(\mathbf{y}^2)_i^0, i = 1, \dots, r\}$;
 $n \leftarrow 1$;
while projection error $> \epsilon_{\text{PE}}$ **do**
 sample $\{(\mathbf{x}^1)_i^n, i = 1, \dots, r\}$ obtained by **LHS** in $\mathcal{X}$;
 for $i = 1, \dots, r$ **do**
 $(\mathbf{y}^1)_i^n = f^1((\mathbf{x}^1)_i^n, (\mathbf{y}^2)_i^{n-1})$;
 $(e^1)_i^n \leftarrow \frac{\|(\mathbf{y}^1)_i^n - p^{-1}(p^1((\mathbf{y}^1)_i^n))\|_2}{\|(\mathbf{y}^1)_i^n\|_2}$;
 end
 $e^1 \leftarrow \frac{1}{r} \sum_{i=1}^r (e^1)_i^n$;
 sample $\{(\mathbf{x}^2)_i^n, i = 1, \dots, r\}$ obtained by **LHS** in $\mathcal{X}$;
 for $i = 1, \dots, r$ **do**
 $(\mathbf{y}^2)_i^n \leftarrow f^2((\mathbf{x}^2)_i^n, (\mathbf{y}^1)_i^n)$;
 $(e^2)_i^n \leftarrow \frac{\|(\mathbf{y}^2)_i^n - p^{-2}(p^2((\mathbf{y}^2)_i^n))\|_2}{\|(\mathbf{y}^2)_i^n\|_2}$;
 end
 $e^2 \leftarrow \frac{1}{r} \sum_{i=1}^r (e^2)_i^n$;
 $p^1 \leftarrow$ **POD** basis built from $\{(\mathbf{y}^1)_i^j, i = 1, \dots, r, j = 1, \dots, n\}$;
 $p^2 \leftarrow$ **POD** basis built from $\{(\mathbf{y}^2)_i^j, i = 1, \dots, r, j = 1, \dots, n\}$;
 projection error $\leftarrow \max(e^1, e^2)$;
 $n \leftarrow n + 1$;
end
output: $p^1, p^2, \{(\mathbf{y}^1)_i^j, i = 1, \dots, r, j = 1, \dots, n\}, \{(\mathbf{y}^2)_i^j, i = 1, \dots, r, j = 0, \dots, n\},$
 $\{(\mathbf{x}^1)_i^j, i = 1, \dots, r, j = 1, \dots, n\}, \{(\mathbf{x}^2)_i^j, i = 1, \dots, r, j = 1, \dots, n\}$

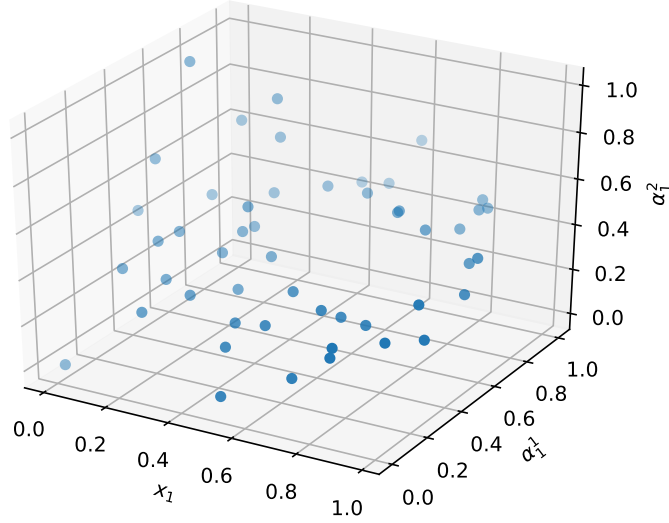


FIGURE 3.9 – Distribution des premiers coefficients α_1^1 et α_1^2 des bases **POD** de chaque discipline et de la première variable de design x_1 donnée par l'algorithme d'entraînement contenant 50 solutions des solveurs disciplinaires.

3.2.4 Résolution de la **MDA** à partir des **GP** disciplinaires

L'objectif de cette section est d'utiliser un solveur **MDA**, appelé solveur **DPOD+I**, résolvant la **MDA** grâce à une stratégie d'enrichissement basée sur les travaux présentés par **DUBREUIL et collab. [2020]**. Dans la section précédente, nous avons défini un **DoE** pour l'entraînement des **GP** disciplinaires. Ceux-ci permettent alors de définir un modèle initial de la **MDA**. Cependant, ce modèle est potentiellement imprécis que ce soit au niveau des bases **POD** ou de l'interpolation par **GP**. Ainsi, nous proposons une stratégie d'enrichissement du modèle initial obtenu à la fin de la phase d'entraînement (comme détaillé dans la Section 3.2.3.2). Pour résoudre la **MDA** en un paramètre de design donné $\mathbf{x} = \mathbf{x}_*$, un modèle dit local est généré à partir du modèle initial et localement enrichi. Ensuite, les données utilisées pour faire converger le modèle local sont filtrées et utilisées pour enrichir le modèle initial afin que ce modèle soit de plus en plus performant au fur et à mesure des résolutions des **MDA**. Cependant, les données utilisées pour l'enrichissement du modèle initial doivent être filtrées car si les **GP** disciplinaires sont entraînés sur des **DoE** mal répartis, cela engendre des problèmes de conditionnement (cf Section 3.2.3).

La première étape que nous allons détailler est l'enrichissement du modèle local jusqu'à ce qu'il soit considéré comme suffisamment précis afin d'obtenir une approximation de la solution de la **MDA**. Les méthodes qui seront présentées par la suite seront illustrées sur le cas analytique présenté dans la Section 1.2.2.1. Cela revient alors à considérer des bases **POD** contenant uniquement un mode. De cette façon, chaque **GP** disciplinaire pourra être illustré puisqu'il ne dépend plus que d'une seule variable de couplage. La **MDA** réduite est ainsi illustrée sur la Figure 3.10 où la ligne pointillée verte représente la réponse de la première discipline selon la variable de couplage α_1^2 pour un paramètre de design fixé $\mathbf{x}_*$. La ligne pointillée violette correspond quand à elle à la réponse de la deuxième discipline selon la variable de couplage α_1^1 . La solution de la **MDA** est alors le

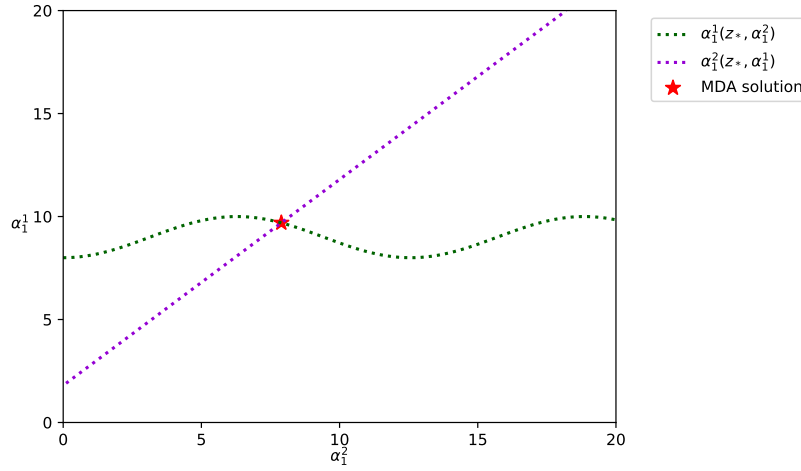


FIGURE 3.10 – Illustration de l'analyse multidisciplinaire réduite sur des bases **POD** limitées à un vecteur. Les solveurs disciplinaires réduits sont illustrés par des courbes en pointillé et la solution de l'analyse multidisciplinaire pour le paramètre $\mathbf{x}_*$ par une étoile rouge.

point d'intersection entre les deux courbes illustrée par une étoile rouge.

Grâce aux étapes d'entraînement, un modèle approché est initialisé :

$$\begin{cases} \hat{\alpha}_i^1 &= \mu_i^1(\mathbf{x}_*, \hat{\alpha}^2) + \hat{e}_i^1(\mathbf{x}_*, \hat{\alpha}^2) \quad \forall i = 1, \dots, n^1 \\ \hat{\alpha}_j^2 &= \mu_j^2(\mathbf{x}_*, \hat{\alpha}^1) + \hat{e}_j^2(\mathbf{x}_*, \hat{\alpha}^1) \quad \forall j = 1, \dots, n^2 \end{cases} \quad (3.14)$$

où μ_i^1 et μ_j^2 sont les moyennes des **GP** disciplinaires, $\hat{e}_i^1$ et $\hat{e}_j^2$ sont des variables aléatoires gaussiennes centrées dont l'écart type σ_i^1 ou σ_j^2 est celui du **GP** correspondant. Notons que la solution de ce système est aléatoire puisqu'il fait intervenir des variables aléatoires définies par la covariance des **GP** disciplinaires. Il est possible de définir une approximation de la **MDA** en considérant uniquement les moyennes des **GP** disciplinaires :

$$\begin{cases} \bar{\alpha}_i^1 &= \mu_i^1(\mathbf{x}_*, \bar{\alpha}^2) \quad \forall i = 1, \dots, n^1 \\ \bar{\alpha}_j^2 &= \mu_j^2(\mathbf{x}_*, \bar{\alpha}^1) \quad \forall j = 1, \dots, n^2 \end{cases} \quad (3.15)$$

où $\bar{\alpha}^1 = \{\bar{\alpha}_1^1, \dots, \bar{\alpha}_{n^1}^1\}$ et $\bar{\alpha}^2 = \{\bar{\alpha}_1^2, \dots, \bar{\alpha}_{n^2}^2\}$. Dans la suite, les solutions $\bar{\alpha}_*^1(\mathbf{x})$ et $\bar{\alpha}_*^2(\mathbf{x})$ sont appelées solutions données par la moyenne des **GP** disciplinaires.

Cette solution est illustrée par une étoile orange sur la Figure 3.11. C'est le point d'intersection des **GP** disciplinaires représentés par des lignes continues. Les réponses des disciplines exactes sont toujours représentées par des lignes pointillées. Sur cette figure, la solution de la **MDA** est représentée par une étoile rouge. Ainsi, il est possible d'obtenir une approximation de la **MDA**. Cependant, si les **GP** disciplinaires ne sont pas suffisamment précis, cela peut engendrer une erreur importante sur l'approximation de la solution de la **MDA**. L'étape suivante est donc d'évaluer l'incertitude liée à l'interpolation en introduisant l'information fournie par les fonctions de covariance des **GP** disciplinaires. Cela revient donc à considérer une **MDA** aléatoire, même si, à l'origine le problème est déterministe. Cette situation est proche de celle observée dans l'algorithme **EGO** où une fonction déterministe est interpolée par **GP** dont la variance est utilisée pour définir une amélioration espérée. Néanmoins, dans notre cadre, la solution de la **MDA** aléatoire n'est pas gaussienne. En effet, même si les approximations disciplinaires sont gaussiennes, le couplage non linéaire rend la solution non gaussienne. Il est donc proposé d'analyser les incertitudes de la solution de la **MDA** aléatoire sur des échantillons. Pour cela, nous allons

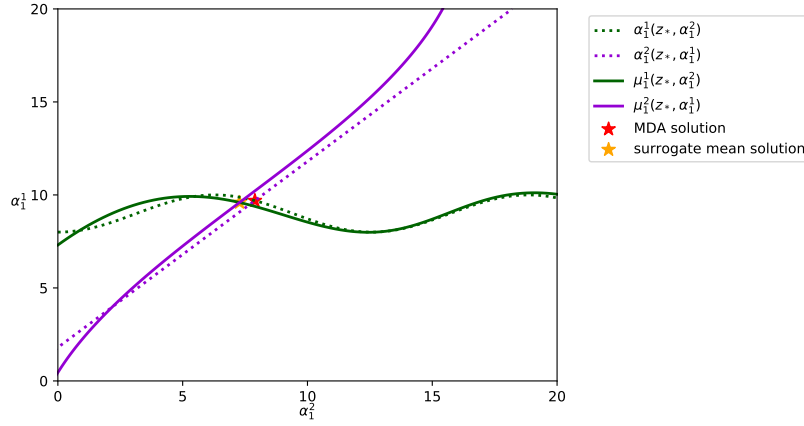


FIGURE 3.11 – Illustration de l'analyse multidisciplinaire réduite sur des bases **POD** limitées à un vecteur. Les disciplines exactes sont illustrées en traits en pointillés et les **GP** disciplinaires sont illustrés avec des traits pleins.

en premier lieu détailler l'obtention de ces échantillons puis nous proposerons une stratégie d'enrichissement pour améliorer les **GP** disciplinaires dans le cas où les incertitudes sont trop importantes.

3.2.4.1 Générer des échantillons de la **MDA** aléatoire réduite

L'objectif de cette section est de propager les incertitudes liées aux **GP** disciplinaires à la solution de la **MDA** aléatoire. Pour cela, nous allons détailler la génération de solutions de la **MDA** aléatoire à partir de trajectoires des **GP** disciplinaires. Notons que les hypothèses faites sur l'interpolation par **GP** dans ce cadre sont les mêmes que celles exposées dans la Section 1.2.5.2 (en particulier l'utilisation de **GP** parfaitement corrélés) et permettent de définir les trajectoires suivantes :

$$\begin{aligned} \hat{t}_i^1(\mathbf{x}_*, \hat{\alpha}^2, \xi_i^1) &= \hat{\xi}_i^1 \sigma_i^1(\mathbf{x}_*, \hat{\alpha}^2), \quad \forall i = 1, \dots, n^1 \\ \hat{t}_j^2(\mathbf{x}_*, \hat{\alpha}^1, \xi_j^2) &= \hat{\xi}_j^2 \sigma_j^2(\mathbf{x}_*, \hat{\alpha}^1), \quad \forall j = 1, \dots, n^2 \end{aligned} \quad (3.16)$$

où les variables aléatoires $\hat{\xi}_i^1 \sim \mathcal{N}(0, 1)$, $i = 1, \dots, n^1$ et $\hat{\xi}_j^2 \sim \mathcal{N}(0, 1)$, $j = 1, \dots, n^2$ sont utilisées pour caractériser les processus et σ_i^1 , σ_j^2 sont les écarts-type conditionnés des **GP** disciplinaires. Ces variables sont par la suite assemblées en deux vecteurs aléatoires : $\hat{\xi}^1 = \{\hat{\xi}_1^1, \dots, \hat{\xi}_{n^1}^1\} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n^1})$ et $\hat{\xi}^2 = \{\hat{\xi}_1^2, \dots, \hat{\xi}_{n^2}^2\} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n^2})$ correspondant à chacune des deux disciplines.

À partir d'un tirage de $\hat{\xi}^1$ et $\hat{\xi}^2$ de respectivement $\hat{\xi}^1$ et $\hat{\xi}^2$, la solution du système suivant est trouvée :

$$\begin{cases} \alpha_i^1 &= \mu_i^1(\mathbf{x}_*, \alpha^2) + \hat{\xi}_i^1 \sigma_i^1(\mathbf{x}_*, \alpha^2) \quad \forall i = 1, \dots, n^1 \\ \alpha_j^2 &= \mu_j^2(\mathbf{x}_*, \alpha^1) + \hat{\xi}_j^2 \sigma_j^2(\mathbf{x}_*, \alpha^1) \quad \forall j = 1, \dots, n^2 \end{cases} \quad (3.17)$$

Cette solution, caractérisée par la variable de design $\mathbf{x}_*$ et le tirage $\hat{\xi}^1$ et $\hat{\xi}^2$, sera notée $\alpha_*^1(\mathbf{x}_*, \hat{\xi}^1, \hat{\xi}^2)$ et $\alpha_*^2(\mathbf{x}_*, \hat{\xi}^1, \hat{\xi}^2)$ dans la suite. Ces variables aléatoires, caractérisant les solutions de la **MDA** aléatoire seront rassemblées en un vecteur : $\Xi = \{\hat{\xi}^1, \hat{\xi}^2\}$ lorsque la distinction n'est pas nécessaire. Notons que la fonction $\mathbf{x}_*, \hat{\xi}^1, \hat{\xi}^2 \rightarrow \alpha_*^1(\mathbf{x}_*, \hat{\xi}^1, \hat{\xi}^2)$ (ainsi que $\mathbf{x}_*, \hat{\xi}^1, \hat{\xi}^2 \rightarrow \alpha_*^2(\mathbf{x}_*, \hat{\xi}^1, \hat{\xi}^2)$) est déterministe. Ce processus est illustré dans la Figure 3.12. À gauche de la Figure 3.12 trois solutions de la **MDA** aléatoire sont générées. Ces solutions sont les points d'intersection des trajectoires qui sont illustrées en traits fins. À droite, la

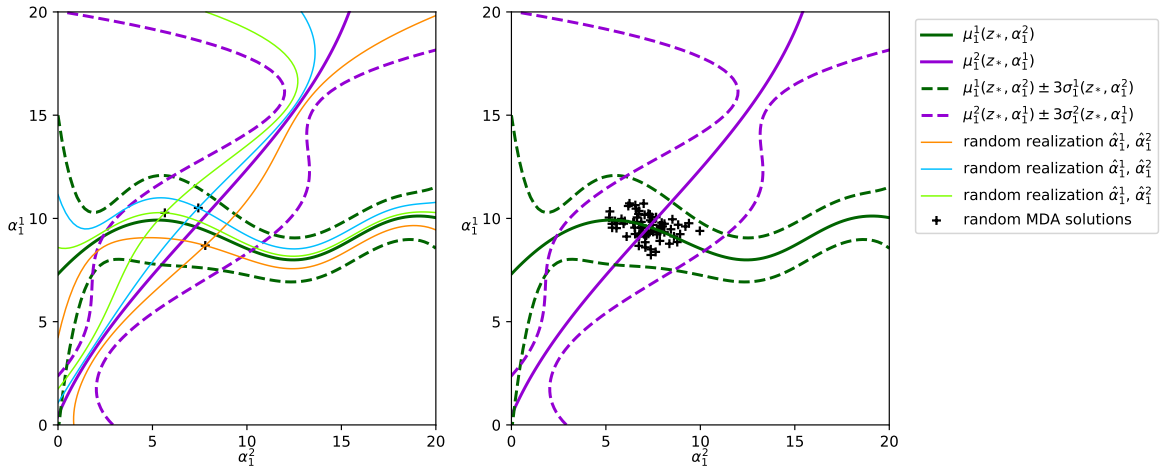


FIGURE 3.12 – Illustration de l'analyse multidisciplinaire réduite sur des bases POD limitées à un vecteur. Sur la gauche, trois solutions de la MDA aléatoire ainsi que les trajectoires des GP parfaitement corrélés associées. Sur la droite, un échantillon de 75 solutions de la MDA aléatoire.

même méthodologie est utilisée pour générer un échantillon de MDA aléatoires de taille 75 afin d'échantillonner la distribution des solutions.

3.2.4.2 Évaluation de la dispersion des solutions de la MDA approchée

Pour simplifier les notations dans la suite de cette section, la dépendance des quantités selon la variable de design $\mathbf{x}_*$ est sous-entendue : l'intérêt sera porté sur la dépendance des quantités aux variables ξ^1 et ξ^2 .

L'objectif de cette section est de quantifier la précision du modèle approché en évaluant la dispersion des solutions de la MDA aléatoire. L'idée est d'estimer cette dispersion en comparant la solution donnée par la moyenne des GP disciplinaires (étoile orange sur la Figure 3.11) et les solutions de la MDA aléatoire (croix noire sur la Figure 3.12). Pour rappel, la solution donnée par la moyenne des GP disciplinaires ($\bar{\alpha}_*^1, \bar{\alpha}_*^2$) est solution de l'Eq. (3.15) et les solutions de la MDA aléatoire ($\hat{\alpha}_*^1(\xi^1, \xi^2), \hat{\alpha}_*^2(\xi^1, \xi^2)$) sont solutions de l'Eq. (3.17). Nous choisissons dans notre étude d'évaluer l'erreur relative entre les solutions reconstruites via les opérateurs p^{-1} et p^{-2} :

$$\begin{aligned} \gamma^1(\hat{\alpha}_*^1, \bar{\alpha}_*^1) &= \frac{\|p^{-1}(\hat{\alpha}_*^1) - p^{-1}(\bar{\alpha}_*^1)\|_2}{\|p^{-1}(\bar{\alpha}_*^1)\|_2} \\ \gamma^2(\hat{\alpha}_*^2, \bar{\alpha}_*^2) &= \frac{\|p^{-2}(\hat{\alpha}_*^2) - p^{-2}(\bar{\alpha}_*^2)\|_2}{\|p^{-2}(\bar{\alpha}_*^2)\|_2} \end{aligned} \quad (3.18)$$

Notons que plus cette quantité est proche de zéro, plus les solutions de la MDA aléatoire sont proches de la solution donnée par la moyenne des GP disciplinaires. Pour rappel, les trajectoires des GP parfaitement dépendants sont égales à la moyenne μ plus une variable normale multipliée par l'écart type : $\xi\sigma$ (cf Eq. (3.16)). Les solutions des MDA aléatoires sont donc proches de la solution donnée par la moyenne des GP disciplinaires lorsque les écarts type des GP disciplinaires sont négligeables par rapport à la moyenne autour de la solution de la MDA. Nous supposons dans notre étude que lorsque l'écart type d'un GP conditionné en un point est négligeable par rapport à la moyenne alors la moyenne du GP est une bonne approximation de la fonction interpolée (exacte) en ce point. Ainsi, la précision de la MDA approchée est obtenue à partir des quantités aléatoires suivantes :

$$\begin{aligned} \hat{q}^1(\hat{\xi}^1, \hat{\xi}^2) &= \gamma^1(\hat{\alpha}_*^1(\hat{\xi}^1, \hat{\xi}^2), \bar{\alpha}_*^1) \\ \hat{q}^2(\hat{\xi}^1, \hat{\xi}^2) &= \gamma^2(\hat{\alpha}_*^2(\hat{\xi}^1, \hat{\xi}^2), \bar{\alpha}_*^2) \end{aligned} \quad (3.19)$$

dépendantes de $\hat{\xi}^1$ et $\hat{\xi}^2$. La **MDA** approchée sera considérée comme suffisamment précise lorsque les τ -quantiles q_τ^1 et q_τ^2 de $\hat{q}^1$ et $\hat{q}^2$ sont inférieurs à ϵ_q où ϵ_q est la tolérance de la résolution (typiquement égale à ϵ_{MDA} qui est la tolérance des algorithmes de résolution de la **MDA** présentés en Section 1.2.1.1) et τ est typiquement choisi à 0.9. Si ce critère est atteint, cela signifie qu'une proportion τ des solutions de la **MDA** aléatoire sont suffisamment proches de la solution moyenne. Ces quantiles seront estimés par **MCS**. Ce critère est similaire à celui des algorithmes de résolution de **MDA** classique où la convergence est atteinte lorsque l'erreur relative entre les variables de couplage entre deux itérations successives est inférieure à ϵ_{MDA} . Dans le cas où le critère n'est pas atteint, un enrichissement du modèle local sera effectué jusqu'à ce que ce critère soit atteint.

3.2.5 Enrichissement de l'analyse multidisciplinaire approchée

3.2.5.1 Enrichissement local des GP disciplinaires

Dans le cas où le critère sur ϵ_q n'est pas atteint, une stratégie d'enrichissement adaptatif est proposée. Une solution naïve serait d'enrichir les deux solveurs disciplinaires. Cependant, cette méthode n'est pas optimale puisqu'un seul des **GP** disciplinaires peut manquer de précision. Nous proposons donc d'enrichir un seul **GP**. L'objectif est donc de déterminer quel est le **GP** disciplinaire le plus imprécis. Pour cela, nous proposons d'évaluer les indices de sensibilité de Sobol suivants :

$$\begin{aligned} S^1 &= \frac{\text{Var}(\mathbb{E}(\hat{q}|\hat{\xi}^1))}{\text{Var}(\mathbb{E}(\hat{q}))} \\ S^2 &= \frac{\text{Var}(\mathbb{E}(\hat{q}|\hat{\xi}^2))}{\text{Var}(\mathbb{E}(\hat{q}))} \end{aligned} \quad (3.20)$$

avec

$$\hat{q}(\hat{\xi}^1, \hat{\xi}^2) = \hat{q}^1(\hat{\xi}^1, \hat{\xi}^2) + \hat{q}^2(\hat{\xi}^1, \hat{\xi}^2) \quad (3.21)$$

Ces indices sont estimés grâce à une approximation par **PCE** (**SUDRET [2008]**). La discipline associée à l'indice le plus important est alors enrichie. De plus, un enrichissement de la base **POD** peut être nécessaire dans le cas où la solution du solveur disciplinaire est mal représentée sur la base **POD**. Ainsi, si l'erreur relative de projection est trop importante, le vecteur sera ajouté à la base **POD** afin de maintenir une erreur de projection faible sur le modèle local.

Le processus d'enrichissement sera alors répété jusqu'à ce que le critère sur les quantiles soit atteint. L'objectif est alors de réduire la dispersion des solutions de la **MDA** aléatoire par des enrichissements successifs. Le processus est décrit dans l'Algorithme 13 et les principales étapes sont :

1. La solution donnée par la moyenne des **GP** disciplinaires définie dans l'Eq (3.15) est calculée. Elle sera utilisée comme référence pour évaluer la dispersion des solutions de la **MDA** aléatoire et pour l'enrichissement.
2. En premier lieu, un échantillon de ξ_k^1 , $k = 1, \dots, n_{\text{MC}}$ et ξ_k^2 , $k = 1, \dots, n_{\text{MC}}$ est généré. Rappelons que cet échantillonnage est trivial puisque $\hat{\xi}^1$ et $\hat{\xi}^2$ suivent des lois normales centrées réduites.
3. Les solutions de la **MDA** aléatoire définies dans l'Eq. (3.17) associées à chacune des trajectoires sont calculées. Notons que sans enrichissement, les solutions sont très dispersées comme cela est illustré sur la Figure 3.13 i).
4. Les quantiles des quantités $\hat{q}^1$ et $\hat{q}^2$ définies dans l'Eq. (3.19) sont estimés.

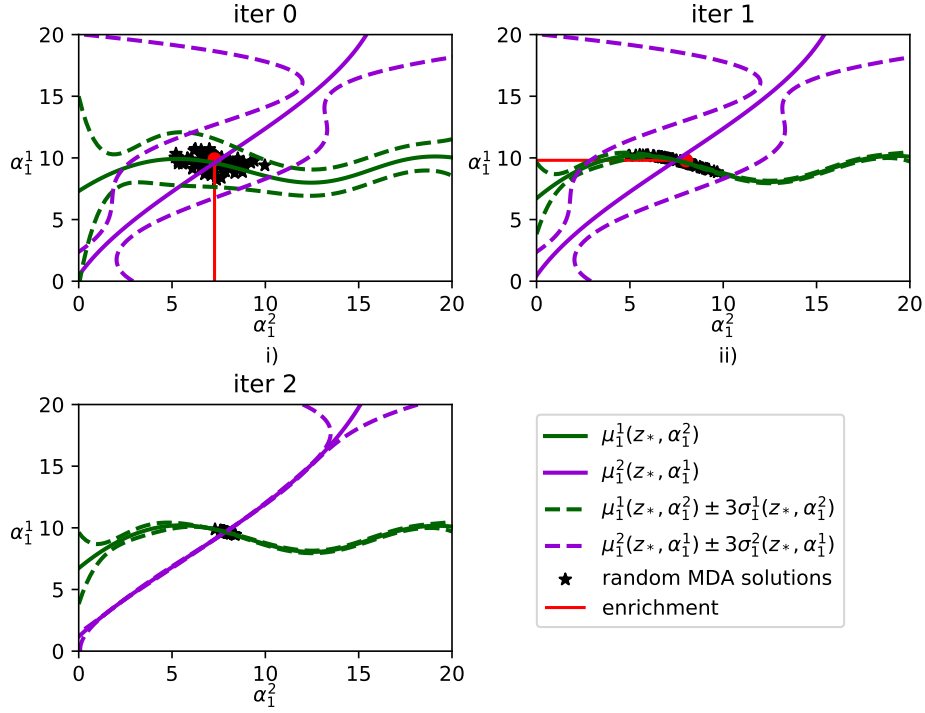


FIGURE 3.13 – Illustration de l'analyse multidisciplinaire réduite sur des bases **POD** limitées à un vecteur. Trois itérations de l'algorithme d'enrichissement sont effectuées : i) solutions obtenues après la phase d'entraînement décrite dans la Section 3.2.3 ii) **MDA** approchée après un enrichissement de la première discipline iii) **MDA** approchée après un nouvel enrichissement, cette fois de la deuxième discipline : le modèle approché est suffisamment précis.

5. Si les quantiles de niveau τ sont supérieurs à ϵ_q , l'un des **GP** disciplinaires sera enrichi. Afin de déterminer la discipline la plus influente sur la quantité $\hat{q}$ définie dans l'Eq. (3.21) les indices de Sobol définis dans l'Eq. (3.20) sont évalués. Les entrées du solveur disciplinaire nécessaires à l'enrichissement sont alors obtenues grâce à la solution donnée par la moyenne des **GP** disciplinaires. Les entrées de la première discipline sont $\mathbf{x}_*$ et la solution reconstruite $p^{-2}(\bar{\alpha}_*^2)$. Les entrées de la deuxième disciplines sont $\mathbf{x}_*$ et la solution reconstruite $p^{-1}(\bar{\alpha}_*^1)$. La **POD** est enrichie si l'erreur relative de projection est supérieure à ϵ_{PE} . Nous pouvons observer sur la Figure 3.13 ii) qu'après l'enrichissement de la première discipline, le **GP** disciplinaire associé est beaucoup plus précis. Ceci se traduit par une moyenne plus fidèle à la vraie discipline ainsi qu'une variance plus faible. Sur la Figure 3.13 iii), après un enrichissement de la deuxième discipline, les variances des deux **GP** disciplinaires sont faibles entraînant une réduction de la dispersion des solutions de la **MDA** aléatoire.
6. Si les quantiles de niveau τ sont inférieurs à ϵ_q , le modèle approché de la **MDA** est suffisamment précis et la solution donnée par la moyenne des **GP** disciplinaires permet une bonne approximation de la vraie solution de la **MDA** comme cela est illustré sur la Figure 3.14 où les solutions exacte et approchée de la **MDA** sont comparées. Cette figure est à comparer à la Figure 3.11 de l'approximation initiale obtenue après l'algorithme d'entraînement.

Remarques :

- Cette stratégie d'enrichissement nécessite de sauvegarder l'ensemble des solutions calculées précédemment, notamment lors de l'ajout d'un vecteur dans la base **POD**.
- Les enrichissements successifs des **GP** disciplinaires peuvent amener à ajouter des données mal réparties sur le **DoE** induisant des **GP** mal conditionnés. En pratique,

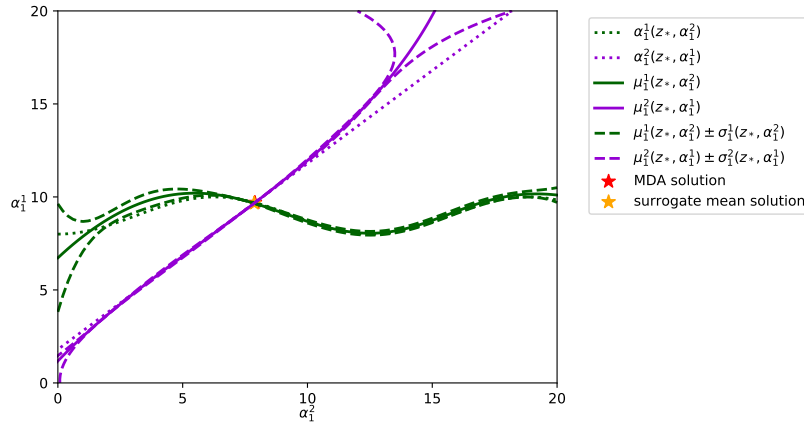


FIGURE 3.14 – Illustration de l'analyse multidisciplinaire réduite sur des bases **POD** limitées à un vecteur. Les disciplines exactes sont illustrées en traits en pointillés et les **GP** disciplinaires sont illustrés avec des traits pleins.

ce cas est identifié en analysant le nombre de solutions de la **MDA** aléatoire qui convergent. Lorsque quasiment aucune solution de la **MDA** aléatoire ne converge, deux solutions sont possibles. Soit nous admettons que le modèle approché a échoué et utilisons un solveur de **MDA** classique pour cette valeur de variable de design particulière. Soit nous faisons abstraction de la stratégie d'enrichissement adaptatif et tentons d'enrichir les deux disciplines simultanément et d'analyser l'impact de cet enrichissement sur la convergence des solutions de la **MDA** aléatoire.

- Notons que l'un des avantages de cette méthode, comparée à une approche *Offline-Online*, est qu'il n'est pas nécessaire que les **GP** disciplinaires soient précis sur l'ensemble des espaces des variables de couplage. En effet, comme on peut l'observer sur la Figure 3.14, les **GP** disciplinaires sont très précis autour de la solution de la **MDA** mais peu précis sur les bords du domaine de définition des variables de couplage. Les enrichissements permettent d'obtenir une bonne estimation de la valeur de la **MDA** sans ajouter d'enrichissement peu pertinent (soit sur les zones éloignées des solutions de la **MDA**).
- Cette méthode permet d'obtenir une bonne approximation de la **MDA** réduite (obtenue en projetant les solveurs disciplinaires sur les bases **POD**). Cependant, aucune information n'est disponible sur l'erreur commise avec la solution de la vraie **MDA** (sans approximation par **POD**). Nous supposons ici que le maintien de l'erreur de projection en dessous de ϵ_{PE} , grâce à la stratégie d'enrichissement des bases **POD**, suffit pour négliger l'erreur entre la **MDA** réduite et la vraie **MDA**. Notons que le lien entre l'erreur commise par la **POD** sur les solveurs disciplinaires et l'erreur commise sur la **MDA** n'est pas trivial à cause du couplage non-linéaire entre les disciplines.

3.2.5.2 Enrichissement du modèle initial

Dans cette section, l'objectif est d'ajouter les informations issues de l'enrichissement local (cf Section 3.2.5.1) au modèle initial (initialisé à partir de l'algorithme d'entraînement : Algorithme 12). Pour rappel, si l'ensemble des informations de l'enrichissement local était ajouté au modèle initial, ceci engendrerait un **DoE** mal réparti avec des groupes de points localisés autour des **MDA** résolues grâce au modèle approché. Cette difficulté nous a donc amené à filtrer les informations issues des enrichissements locaux afin de ne

Algorithm 13: Compute MDA solution using DPOD+I approximation.

```

input :  $\mathbf{x}_*$ ,  $n_{\text{MC}}$ 
while  $\tau$ -quantile of  $\hat{q} > \epsilon_q$  do
     $\bar{\alpha}_*^1, \bar{\alpha}_*^2 \leftarrow$  solution of Eq. (3.15);
     $\xi_k^1, k = 1, \dots, n_{\text{MC}} \leftarrow n^1 \times n_{\text{MC}}$  independent standard Gaussian samples;
     $\xi_k^2, k = 1, \dots, n_{\text{MC}} \leftarrow n^2 \times n_{\text{MC}}$  independent standard Gaussian samples;
    for  $k = 1, \dots, n_{\text{MC}}$  do
         $\hat{\alpha}_*^1(\xi_k^1, \xi_k^2), \hat{\alpha}_*^2(\xi_k^1, \xi_k^2) \leftarrow$  solution of Eq. (3.17) associated to  $\xi_k^1$  and  $\xi_k^2$ ;
         $q_k^1 \leftarrow \Upsilon^1(\hat{\alpha}_*^1(\xi_k^1, \xi_k^2), \bar{\alpha}_*^1)$ ;
         $q_k^2 \leftarrow \Upsilon^2(\hat{\alpha}_*^2(\xi_k^1, \xi_k^2), \bar{\alpha}_*^2)$ ;
         $q_k \leftarrow q_k^1 + q_k^2$ ;
    end
     $q_l^1 \leftarrow \tau$ -quantile of  $(q_k^1)_{k=1}^{n_{\text{MC}}}$ ;
     $q_l^2 \leftarrow \tau$ -quantile of  $(q_k^2)_{k=1}^{n_{\text{MC}}}$ ;
    if  $q_l^1 > \epsilon_q$  or  $q_l^2 > \epsilon_q$  then
         $\tilde{S}^1 \leftarrow$  estimation of  $S^1$  by PCE;
         $\tilde{S}^2 \leftarrow$  estimation of  $S^2$  by PCE;
        if  $\tilde{S}^1 > \tilde{S}^2$  then
            Enrich the first disciplinary surrogate model;
             $\bar{\mathbf{y}}^1 \leftarrow f^1(\mathbf{x}^*, p^{-2}(\bar{\alpha}_*^2))$ ;
            If needed also enrich the first POD basis;
            if  $\frac{\|\bar{\mathbf{y}}^1 - p^{-1}(p(\bar{\mathbf{y}}^1))\|_2}{\|\bar{\mathbf{y}}^1\|_2} > \epsilon_{\text{PE}}$  then
                 $\Phi^1 \leftarrow \left\{ \Phi^1, \frac{\bar{\mathbf{y}}^1 - p^{-1}(p(\bar{\mathbf{y}}^1))}{\|\bar{\mathbf{y}}^1 - p^{-1}(p(\bar{\mathbf{y}}^1))\|_2} \right\}$ ;
            end
        else
            Enrich the second disciplinary surrogate model;
             $\bar{\mathbf{y}}^2 \leftarrow f^2(\mathbf{x}^*, p^{-1}(\bar{\alpha}_*^1))$ ;
            If needed also enrich the second POD basis;
            if  $\frac{\|\bar{\mathbf{y}}^2 - p^{-2}(p^2(\bar{\mathbf{y}}^2))\|_2}{\|\bar{\mathbf{y}}^2\|_2} > \epsilon_{\text{PE}}$  then
                 $\Phi^2 \leftarrow \left\{ \Phi^2, \frac{\bar{\mathbf{y}}^2 - p^{-2}(p^2(\bar{\mathbf{y}}^2))}{\|\bar{\mathbf{y}}^2 - p^{-2}(p^2(\bar{\mathbf{y}}^2))\|_2} \right\}$ ;
            end
        end
    end
end
output:  $p^{-1}(\bar{\alpha}_*^1), p^{-2}(\bar{\alpha}_*^2)$ 

```

pas obtenir un modèle local mal conditionné. Le modèle local permet, pour un paramètre donné $\mathbf{x}_*$, d'effectuer des enrichissements successifs afin d'obtenir une approximation fidèle de la MDA. L'idée est alors d'ajouter certaines informations issues des enrichissements locaux dans le but d'améliorer le modèle initial au fur et à mesure des résolutions tout en conservant un DoE bien réparti. Cet enrichissement du modèle initial permettra de réduire le nombre d'appels aux solveurs disciplinaires.

La solution que nous avons adoptée est de n'utiliser que le dernier appel à chaque solveur disciplinaire lors des enrichissements. De plus, nous avons décidé de ne pas enrichir la POD obtenue après l'algorithme d'entraînement. Les bases ne seront donc enrichies que lors de la résolution locale de la MDA dans l'Algorithme 13. Notons que les vecteurs ajoutés lors des enrichissements locaux représentent des phénomènes locaux qui ne sont pas forcément pertinent pour le modèle initial. Limiter le nombre de vecteurs permet également de limiter le nombre d'entrées des GP disciplinaires qui seront donc plus faciles à apprendre. Cette procédure est synthétisée dans l'Algorithme 14.

Algorithm 14: Enrichment of the initial model through the resolutions of the MDA.

```

input :  $\mathbf{x}^1, \dots, \mathbf{x}^m, n_{MC}$ 
Initial model obtained through Algorithm 12 ;
for  $k = 1, \dots, m$  do
    Local model  $\leftarrow$  copy of initial model ;
    Solve the MDA using Algorithm 13;
    if Enrichement of the first discipline during Algorithm 13 then
        | Enrich the initial model using last call of the first discipline solver.
    end
    if Enrichement of the second discipline during Algorithm 13 then
        | Enrich the initial model using last call of the second discipline solver.
    end
end
output: initial model

```

3.2.6 Comparaison avec la POD+I

Afin de pouvoir comparer l'approche que nous avons proposée, appelée DPPOD+I, nous présentons également dans cette section l'approche classique POD+I, qui vise à approcher directement les solutions MDA exactes $\mathbf{y}_*^1(\mathbf{x})$ and $\mathbf{y}_*^2(\mathbf{x})$ comme cela est proposé par COELHO et collab. [2008]. Pour entraîner la POD+I, seules les solutions exactes de la MDA doivent être utilisées. Tout d'abord, un DoE est nécessaire pour entraîner les bases et les méta-modèles. Pour ce faire, n_{DoE} différentes variables de design sont obtenues par LHS : $\{\mathbf{x}_i^{DoE}, i = 1, \dots, n_{DoE}\}$. Les solutions de la MDA $\{\mathbf{y}_*^1(\mathbf{x}_i^{DoE}), i = 1, \dots, n_{DoE}\}$ et $\{\mathbf{y}_*^2(\mathbf{x}_i^{DoE}), i = 1, \dots, n_{DoE}\}$ associées à chaque variable de design sont obtenues à l'aide d'un solveur MDA. Ensuite, une base POD est créée pour chaque discipline. Un DoE pour les méta-modèles de chaque discipline est obtenu en projetant chaque solution MDA sur la base POD. Les méta-modèles utilisés sont des GP conditionnés :

$$\begin{cases} (\hat{\alpha}_*)_i(z) &= \mu_j^1(z) + \hat{\epsilon}_i^1(z) \quad \forall i = 1, \dots, n^1 \\ (\hat{\alpha}_*)_j(z) &= \mu_j^2(z) + \hat{\epsilon}_j^2(z) \quad \forall j = 1, \dots, n^2 \end{cases} \quad (3.22)$$

où $\hat{\epsilon}_i^1$ et $\hat{\epsilon}_j^2$ sont des GP de moyenne nulle dont l'écart type est celui du GP conditionné correspondant (i.e. respectivement σ_i^1 et σ_j^2). A partir de cet entraînement *Offline*, il est

possible de comparer les performances de la **POD+I** en analysant l'approximation *On-line* donnée par les valeurs moyennes des **GP**.

Pour réaliser une comparaison équitable avec notre approche **DPOD+I**, nous adaptons davantage l'approche **POD+I** classique en intégrant une stratégie d'enrichissement similaire à celle proposée pour notre approche décrite dans la section 3.2.5.1. Cette méthode est appelée **POD+I** enrichie dans la suite. À noter que la différence majeure qui subsiste entre la **DPOD+I** et la **POD+I** enrichie réside sur la quantité approchée : les solveurs disciplinaires pour la **DPOD+I** ou les variables de couplage convergées pour la **POD+I**.

- La **DPOD+I** résout la **MDA** grâce à des **GP** disciplinaires, dont les entrées sont les variables de design et les variables de couplage. Les **GP** disciplinaires sont alors enrichis jusqu'à ce que les solutions de la **MDA** aléatoire soient suffisamment concentrées. Notons que l'enrichissement d'un de ces **GP** disciplinaires nécessite un seul appel à un solveur disciplinaire.
- La **POD+I** enrichie résout quant à elle la **MDA** en utilisant une approximation directe des solutions de la **MDA**. Ainsi, si les solutions de la **POD+I** sont trop dispersées, il est nécessaire de résoudre la **MDA** à partir d'un solveur externe puisque la **POD+I** est entraînée sur les solutions de la **MDA**. Chaque enrichissement nécessite autant d'appels aux solveurs disciplinaires que le nombre d'itérations du solveur de la **MDA**.

L'algorithme d'enrichissement pour la **POD+I** enrichie est décrit dans l'Algorithme 15 selon les étapes suivantes :

- Les valeurs moyennes des **GP** donnent une approximation de la solution. Il convient de noter que la solution est plus facile à obtenir car la **POD+I** estime la solution du système couplé plutôt que la solution de chaque solveur disciplinaire.
- Une **MCS** est générée en utilisant les valeurs de la variance des **GP**.
- Une analyse similaire est faite sur la variation des solutions aléatoires autour de la solution moyenne.

Algorithm 15: Compute [MDA](#) solution using [POD+I](#).

```

input :  $\mathbf{x}$ ,  $n_{\text{MC}}$ 
 $(\bar{\alpha}_*^1)_i \leftarrow \mu_i^1(\mathbf{x})$ ,  $i = 1, \dots, n^1$ ;
 $(\bar{\alpha}_*^2)_j \leftarrow \mu_j^2(\mathbf{x})$ ,  $j = 1, \dots, n^2$ ;
 $\xi_k^1$ ,  $k = 1, \dots, n_{\text{MC}} \leftarrow n^1 \times n_{\text{MC}}$  independent standard Gaussian samples;
 $\xi_k^2$ ,  $k = 1, \dots, n_{\text{MC}} \leftarrow n^2 \times n_{\text{MC}}$  independent standard Gaussian samples;
for  $k = 1, \dots, n_{\text{MC}}$  do
     $((\alpha_*^1)_i)_k \leftarrow \mu_i^1(\mathbf{x}) + (\xi_k^1)_i \sigma_i^1(\mathbf{x})$ ,  $i = 1, \dots, n^1$ ;
     $((\alpha_*^2)_j)_k \leftarrow \mu_j^2(\mathbf{x}) + (\xi_k^2)_j \sigma_j^2(\mathbf{x})$ ,  $j = 1, \dots, n^2$ ;
     $q_k^1 \leftarrow \gamma^1(\hat{\alpha}_*^1(\xi_k^1, \xi_k^2), \bar{\alpha}_*^1)$ ;
     $q_k^2 \leftarrow \gamma^2(\hat{\alpha}_*^2(\xi_k^1, \xi_k^2), \bar{\alpha}_*^2)$ ;
     $q_k \leftarrow q_k^1 + q_k^2$ ;
end
 $q_l^1 \leftarrow \tau\text{-quantile of } (q_k^1)_{k=1}^{n_{\text{MC}}}$ ;
 $q_l^2 \leftarrow \tau\text{-quantile of } (q_k^2)_{k=1}^{n_{\text{MC}}}$ ;
if  $q_l^1 > \epsilon_q$  or  $q_l^2 > \epsilon_q$  then
    Then, the surrogate model is not accurate enough and the exact MDA is
    computed;
    The GP are enriched accordingly;
     $\mathbf{y}^1 \leftarrow \mathbf{y}_*^1(\mathbf{x})$ ;
     $\mathbf{y}^2 \leftarrow \mathbf{y}_*^2(\mathbf{x})$ ;
else
    The surrogate model is accurate enough;
     $\mathbf{y}^1 \leftarrow p^{-1}(\bar{\alpha}_*^1)$ ;
     $\mathbf{y}^2 \leftarrow p^{-2}(\bar{\alpha}_*^2)$ ;
end
output:  $\mathbf{y}^1$ ,  $\mathbf{y}^2$ 

```

3.3 Application sur un cas aéroélastique

Dans la suite, nous allons appliquer les différentes méthodes que nous avons détaillées sur un cas couplant deux disciplines : aérodynamique et structure. Dans un premier temps, nous allons décrire le problème physique. Dans un second temps, nous testerons et comparerons les algorithmes présentés.

3.3.1 Description de la MDA

Les problèmes aéroélastiques statiques sont des cas tests très répandus en analyse multidisciplinaire (COELHO et collab. [2008]; COELHO et collab. [2009]; BONCORAGLIO et collab. [2021]; SCHOLTEN et HARTL [2021]) aussi connus sous le nom de FSI. Ce problème multi-disciplinaire fait intervenir deux disciplines : les efforts sur la structure sont calculés à partir de la géométrie de l'aile déformée et la déformation de l'aile est obtenue à partir des efforts appliqués sur la structure. Ce couplage est modélisé par le système suivant :

$$\begin{cases} \mathbf{u} = \mathcal{M}_s(\mathbf{x}, \mathbf{f}) \\ \mathbf{f} = \mathcal{M}_a(\mathbf{x}, \mathbf{u}) \end{cases} \quad (3.23)$$

où $\mathbf{u}$ est le vecteur de déplacement de l'aile, $\mathbf{f}$ est le vecteur des efforts aérodynamiques, $\mathbf{x}$ est le vecteur contenant les variables de design, $\mathcal{M}_a$ est un solveur aérodynamique et $\mathcal{M}_s$ un solveur structure.

3.3.1.1 Description des solveurs disciplinaires

Dans cette étude, la structure de l'aile est supposée élastique linéaire et l'équation régissant le déplacement est résolue à l'aide d'une méthode d'éléments finis, voir Section 2.5.1. Le modèle éléments finis de l'aile est composé d'éléments de plaques minces pour les peaux, les nervures et les longerons et d'éléments de poutre pour les lisses rigidifiant la peau. Le maillage est présenté sur la Figure 3.15. La structure de l'aile est composée de 3 longerons modélisés en forme de poutre en T avec des éléments plaque. Le revêtement de l'aile est rigidifié par neuf lisses sur les revêtements supérieur et inférieur modélisées par des éléments de poutre (sur la Figure 3.15 le revêtement supérieur de l'aile est retiré pour plus de clarté). Le nombre total de degrés de liberté du modèle de structure est de $d^1 = 43\,416$. Enfin, le solveur structure utilisé est Code Aster EDF [1989–2017] comme décrit dans la Section 2.5.1.

L'aérodynamique est décrite par la théorie de l'écoulement potentiel résolue en utilisant la *Vortex Lattice Method* (VLM) (KATZ et PLOTKIN [2001]). La VLM modélise les surfaces portantes de l'aile comme une feuille infiniment mince composée de tourbillons discrets pour calculer la portance et la traînée induite. L'influence de l'épaisseur et de la viscosité est négligée. Cette méthode permet de simuler, avec une géométrie simplifiée (typiquement un quadrangle), une aile d'avion. L'aile est alors discrétisée en un certain nombre de panneaux. Pour chacun de ces panneaux, un vortex en forme de fer à cheval est placé au quart de la corde et un point de contrôle est positionné au trois-quarts de la corde. L'inconnue qui doit être déterminée est l'amplitude de ce vortex noté Γ . Pour cela on considère la vecteur de vitesse total qui, par hypothèse de conservation du flux, est donné par

$$\mathbf{v} = \mathbf{v}_\infty + \nabla \phi_v \quad (3.24)$$

où $\mathbf{v}_\infty$ est la vitesse du fluide à l'infini et $\nabla\Phi_v$ est une perturbation de la vitesse. Pour un problème à n_{panneaux} panneaux, la perturbation au point de contrôle est obtenue en sommant la contribution de l'ensemble des vortex :

$$(\nabla\Phi_v)_i = \sum_{j=1}^{n_{\text{panneaux}}} \mathbf{w}_{ij} \Gamma_j, i = 1 \dots, d^2 \quad (3.25)$$

où $\mathbf{w}_{ij}$ sont les **Coefficients d'influence aérodynamiques, ou Aerodynamic Influence Coefficients (AIC)** dont le calcul est détaillé par KATZ et PLOTKIN [2001]. De plus, le fluide étant non visqueux, la vitesse au point de contrôle respecte la condition de Neumann suivante :

$$\mathbf{v}_i \cdot \mathbf{n}_i = 0, i = 1 \dots, n_{\text{panneaux}} \quad (3.26)$$

où $\mathbf{v}_i$ est la vitesse au point de contrôle du panneau i et $\mathbf{n}_i$ est la normale au panneau au point de contrôle. Ainsi, en projetant l'Eq. (3.24) sur $\mathbf{n}_i$, on obtient :

$$0 = \mathbf{v}_\infty \cdot \mathbf{n}_i + \sum_{j=1}^{n_{\text{panneaux}}} (\mathbf{w}_{ij} \cdot \mathbf{n}_i) \Gamma_j, i = 1 \dots, n_{\text{panneaux}} \quad (3.27)$$

Ces n_{panneaux} équations peuvent être assemblées en un système linéaire de dimension n_{panneaux} :

$$\mathbf{AIC} \, \Gamma = \mathbf{b} \quad (3.28)$$

où la matrice **AIC** contient les coefficients projetés $(\mathbf{w}_{ij} \cdot \mathbf{n}_i)$ et $\mathbf{b}$ contient les projections de la vitesse du fluide à l'infini sur les normales $\mathbf{v}_\infty \cdot \mathbf{n}_i$. Les efforts aérodynamiques peuvent alors être obtenus à partir des forces des vortex Γ sans avoir à résoudre de système linéaire supplémentaire. Notons que la dimension des efforts est trois fois supérieure au nombre de panneaux retenus. Dans nos travaux, nous fixons n_{panneaux} à 700, donc $d^2 = 2100$ degrés de liberté pour le solveur aérodynamique. Le maillage aérodynamique de l'aile est présenté sur la Figure 3.16 et la résolution est assurée par un solveur interne à l'ONERA.

3.3.1.2 Description du couplage entre les disciplines

Le transfert des efforts et des déplacements entre le maillage aérodynamique et structure se fait par interpolation par fonctions à base radiale RENDALL et ALLEN [2007]. Une matrice d'interpolation $\mathbf{H} \in \mathbb{R}^{d^2 \times d^1}$ est construite en utilisant l'interpolation par fonctions à base radiale. Ensuite, le transfert du déplacement sur le maillage aérodynamique $\mathbf{u}_a$ au maillage structure est calculé par $\mathbf{u}_a = \mathbf{H}\mathbf{u}$. De même, les charges aérodynamiques sur le maillage structure $\mathbf{f}_s$ sont transférés au maillage structure via $\mathbf{f}_s = \mathbf{H}^T \mathbf{f}$ afin d'imposer la conservation des efforts (voir RENDALL et ALLEN [2007] pour plus de détails) :

$$\delta w = \delta \mathbf{u}_s^T \mathbf{f}_s = \delta \mathbf{u}_a^T \mathbf{f}_a \quad (3.29)$$

avec δw les efforts virtuels et $\delta \mathbf{u}$ les déplacement virtuels. La MDA obtenue est décrite dans la Figure 3.17 où une étape de transfert sur les variables de couplage est effectué.

3.3.1.3 Description de l'avion

Nous considérons dans notre étude un avion de ligne de type A350 dont les caractéristiques géométriques sont les suivantes. L'envergure totale de l'avion est de 65m, l'aile ayant une envergure de 29m. La cassure (visible sur les Figures 3.15 et 3.16) se situe à 40%

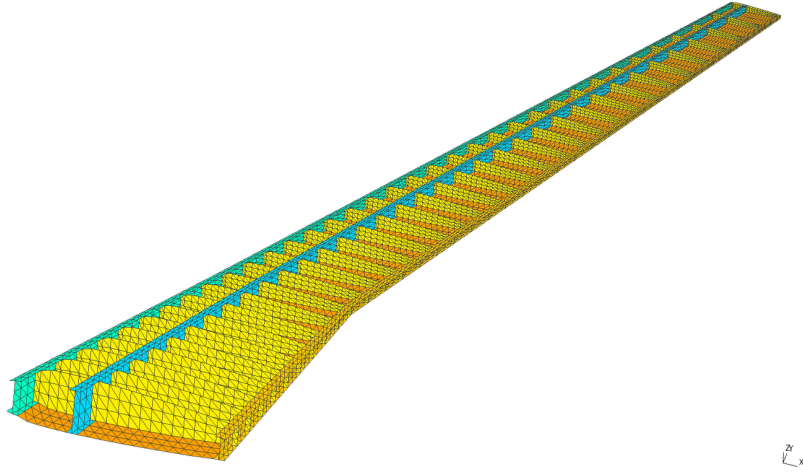


FIGURE 3.15 – Maillage de la structure de l'aile sans les lisses et la peau supérieure.

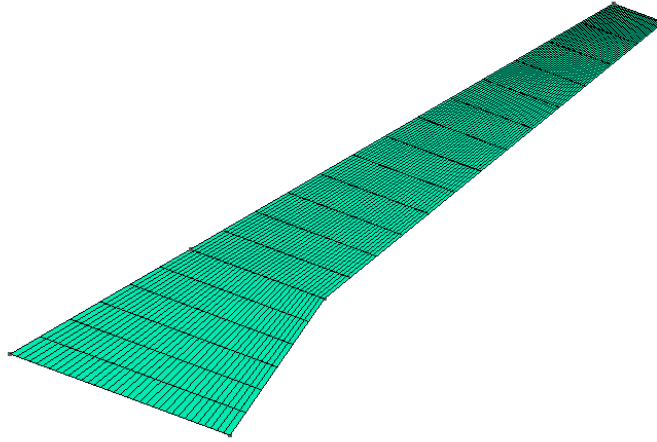


FIGURE 3.16 – Maillage aérodynamique de l'aile.

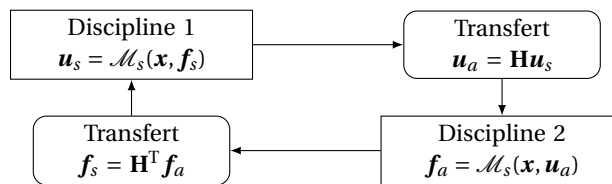


FIGURE 3.17 – Illustration de la **MDA** sur un problème **FSI** décrit dans l'Eq (3.23) où les variables de couplage sont transférées grâce à la matrice de transfert **H**.

TABEAU 3.1 – Problème à 4 paramètres comprenant l'AoI, la vitesse et les épaisseurs (les paramètres sont normalisés entre 0 et 1).

Variable de design	aérodynamique		structure	
	AoI	vitesse	épaisseurs	
espace de définition	[0,1]	[0,1]	longerons [0,1]	peaux [0,1]

TABEAU 3.2 – Problème à 8 paramètres comprenant l'AoI, la vitesse et les épaisseurs (les paramètres sont normalisés entre 0 et 1).

Variable de design	aérodynamique		structure		
	AoI	vitesse	épaisseurs		
espace de définition	[0,1]	[0,1]	longerons avant/central/arrière [0,1] × [0,1] × [0,1]	peaux supérieure/inférieure [0,1] × [0,1]	nervures [0,1]

de l'envergure. La corde a pour longueur 13m à l'implanture, 6m au niveau de la cassure et 3m en bout d'aile. La surface portante totale est de 443m². L'épaisseur de la structure de l'aile décroît le long de l'aile et est comprise entre 1m et 20cm. Le dièdre de l'aile est fixé à 5°, la flèche est constante le long du bord d'attaque et vaut 35°. Pour le calcul aérodynamique, les conditions de vol sont celles d'un vol de croisière à une altitude d'environ 11km et une masse volumique de l'air de 0.36 kg.m⁻³. Le mach et l'angle d'incidence de l'avion seront des variables de design de la MDA. Pour le calcul de structure, nous supposons que l'aile est composée uniquement d'aluminium avec un module d'élasticité de 73MPa et un coefficient de poisson de 0.3 pour une masse volumique de 2 770 kg.m⁻³. Les lisses sont modélisées par des poutres en forme de T composées d'aluminium. Le poids propre de l'aile est pris en compte. Les épaisseurs des éléments de la structure seront des variables de design de la MDA.

3.3.1.4 Paramétrisation du problème

Deux problèmes paramétriques légèrement différents sont maintenant introduits. Ils utilisent les solveurs disciplinaires précédemment décrits mais un nombre différent de variables de design. Le premier compte 4 paramètres décrits dans la Table 3.1 et le second 8 paramètres décrits dans la Table 3.2.

La différence entre les deux problèmes concerne le nombre de paramètres utilisés pour représenter l'épaisseur des pièces de structure. Dans le problème à 4 paramètres, la même épaisseur est appliquée aux 3 longerons et l'épaisseur des peaux inférieure et supérieure est également supposée égale. Dans le problème à 8 paramètres, 4 nouvelles variables de design sont introduites pour différencier les épaisseurs de ces pièces structurelles. Les variables de design liées à l'aérodynamique : l'angle d'incidence, ou *Angle of Incidence* (AoI) et la vitesse sont inchangées entre les deux problèmes.

Il convient de noter que les paramètres de design sont mis à l'échelle pour prendre des valeurs dans l'intervalle [0, 1] (voir Table 3.1 et Table 3.2). Cependant, les valeurs d'échelle ont été choisies de manière à ce que la plage de variation obtenue par rapport aux sorties de la MDA soit suffisamment large pour évaluer les performances de la méthode proposée. A titre d'exemple, le déplacement en mètre du bout d'aile obtenu appartient à [0.06, 3.95].

La tolérance du solveur MDA ϵ_{MDA} est fixée à 0.01. Le point de départ est donné par

la solution pour le paramètre au centre de l'espace paramétrique $\mathcal{X}$. La référence pour tester la méthode est faite sur un DoE contenant 100 solutions MDA définies par 100 variables de design différentes obtenues via un LHS $\{\mathbf{x}_i^{\text{DoE}}, i = 1, \dots, 100\}$ et est notée ensemble de test. La résolution de la MDA nécessite en moyenne 4.3 appels à chaque solveur disciplinaire pour le problème avec 4 paramètres et 3.9 pour le problème avec 8 paramètres. Les déplacements et amplitudes des vortex solutions de la MDA obtenus en trois points particuliers de l'espace de design sont représentés sur la Figure 3.18.

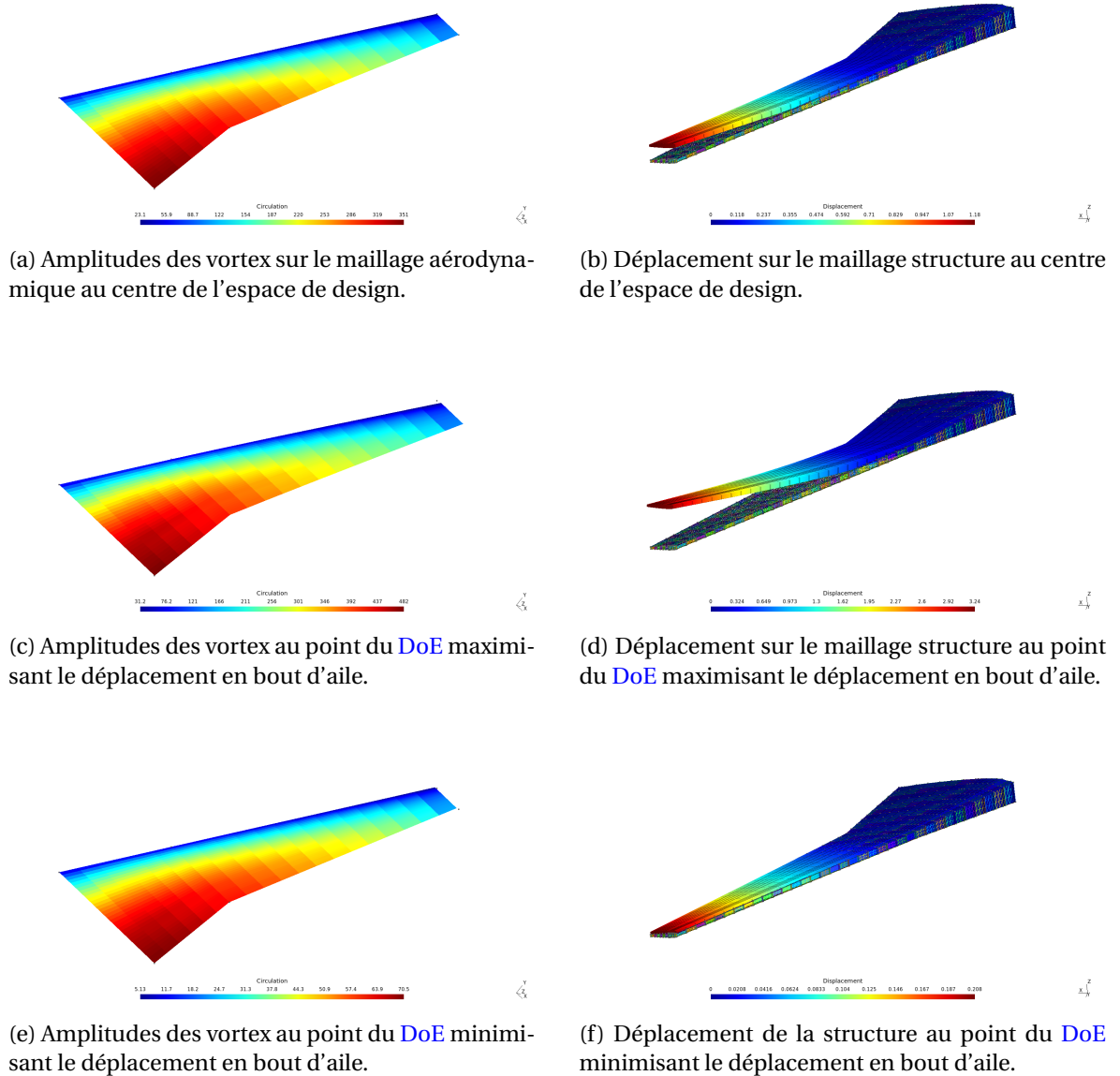


FIGURE 3.18 – Illustration des amplitudes des vortex résultant du calcul aérodynamique et déplacements obtenus à partir du solveur structure en trois points particuliers : le centre de l'espace paramétrique et les points du DoE minimisant ou maximisant le déplacement en bout d'aile.

3.3.1.5 Réglage des paramètres de la DPOD+I

Les bornes λ^+ et λ^- décrites dans la Section 3.2.3.1 sont données dans ce cas par une analyse physique : le déplacement maximum de l'aile est supposé compris entre 0. et 3.6

mètres. Ces valeurs ne sont pas celles données dans la Section 3.3.1 (0.06 et 3.95) mais des valeurs approximatives pour garantir que la précision sur ces limites n'est pas requise. La tolérance sur l'erreur de projection moyenne est fixée au même seuil que le solveur MDA : $\epsilon_{PE} = 0.01$. Ce paramètre permet de gérer l'erreur entre la MDA exacte et la MDA réduite. La matrice de projection est obtenue par SVD en utilisant la bibliothèque scikit-learn PEDREGOSA et collab. [2011]. Le pourcentage η d'énergie captée est fixé à 99.995%. Les GP sont entraînés à l'aide de la bibliothèque SMT BOUHLEL et collab. [2019]. La fonction de corrélation utilisée est une Matérn avec $\nu = \frac{5}{2}$ WILLIAMS et RASMUSSEN [2006]. Pour la résolution de la MDA, deux paramètres sont à régler : τ qui définit le quantile retenu et ϵ_q qui est le seuil à atteindre par ce quantile. Ici et pour le reste de l'étude, τ est fixé à 0.9 et ϵ_q à 0.01. Ces paramètres permettent de gérer l'erreur commise entre la MDA réduite et la MDA approchée. Pour évaluer le quantile, le nombre n_{MC} de MDA aléatoires générées est fixé à 2000.

Remarque :

L'AoI et la vitesse ne concernent que l'aérodynamique alors que les épaisseurs ne concernent que la structure cf Tables 3.1 et 3.2. Par conséquent, pour le problème à 4 paramètres, le nombre d'entrées pour les GP disciplinaires du solveur aérodynamique est de $n^1 + 2$ et $n^2 + 2$ pour les GP disciplinaires du solveur structure avec n^1 le nombre de modes POD pour la structure et n^2 pour l'aérodynamique. Pour le problème avec 8 paramètres, le nombre d'entrées pour les GP disciplinaires du solveur aérodynamique est $n^1 + 2$ et $n^2 + 6$ pour les GP disciplinaires du solveur structure.

3.3.2 Problème à 4 variables de design

3.3.2.1 Résolution avec le solveur DPOD+I

Tout d'abord, des modèles de DPOD+I sont générés en fixant r , la taille de l'échantillon initial des variables de couplage décrites dans la Section 3.2.3.1, à 10. L'algorithme d'entraînement (voir Algorithme 12) se termine après $n = 5$ itérations conduisant à 50 appels au solveur structure et 60 appels au solveur aérodynamique. La taille de la base POD est de $n^1 = 6$ pour la structure et de $n^2 = 5$ pour l'aérodynamique. Certains modes POD obtenus sont représentés sur la Figure 3.19 pour la structure et la Figure 3.20 pour l'aérodynamique. Dans un premier temps, la qualité des bases POD obtenues est analysée au moyen des erreurs relatives entre la solution de la MDA exacte et son approximation par POD :

$$\begin{aligned} (e^1_{DoE})_i &= \frac{\|y^1(x_i^{DoE}) - p^{-1}(p^1(y^1(x_i^{DoE})))\|_2}{\|y^1(x_i^{DoE})\|_2}, \quad i = 1, \dots, 100 \\ (e^2_{DoE})_i &= \frac{\|y^2(x_i^{DoE}) - p^{-2}(p^2(y^2(x_i^{DoE})))\|_2}{\|y^2(x_i^{DoE})\|_2}, \quad i = 1, \dots, 100 \end{aligned} \quad (3.30)$$

On cherche ici à vérifier la capacité de l'Algorithme 12 à créer des bases POD précises sur les variables de couplage convergées. Les histogrammes de ces quantités sont présentés dans la Figure 3.21. L'erreur relative pour le modèle POD de la structure se situe entre 0.16 % et 3.8 % avec une moyenne de 0.7 %. L'erreur relative pour le modèle POD aérodynamique se situe entre 0.03 % et 0.7 % avec une moyenne de 0.16 %. L'approximation POD est performante pour ce type de solveurs disciplinaires. Notons que ce résultat est important pour la DPOD+I. En effet, les coefficients de la POD sont des entrées des GP disciplinaires. Ainsi, une grande base de POD compliquerait l'apprentissage de ces GP disciplinaires. Cela confirme également que, sur cet exemple, une base POD globale est capable d'approcher avec précision les solutions des solveurs disciplinaires.

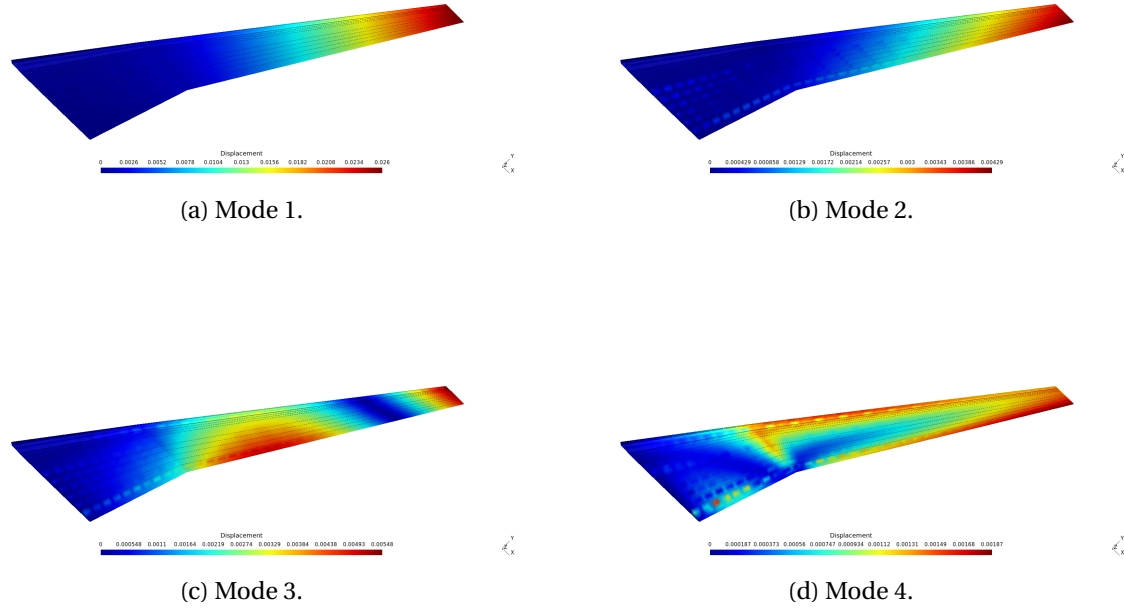


FIGURE 3.19 – Illustration des quatre premiers modes **POD** obtenus sur le solveur structure grâce à l'algorithme d'entraînement. Les nuances de couleur représentent les variations de la norme du déplacement, le bleu correspondant à un déplacement nul et le rouge à un déplacement maximal.

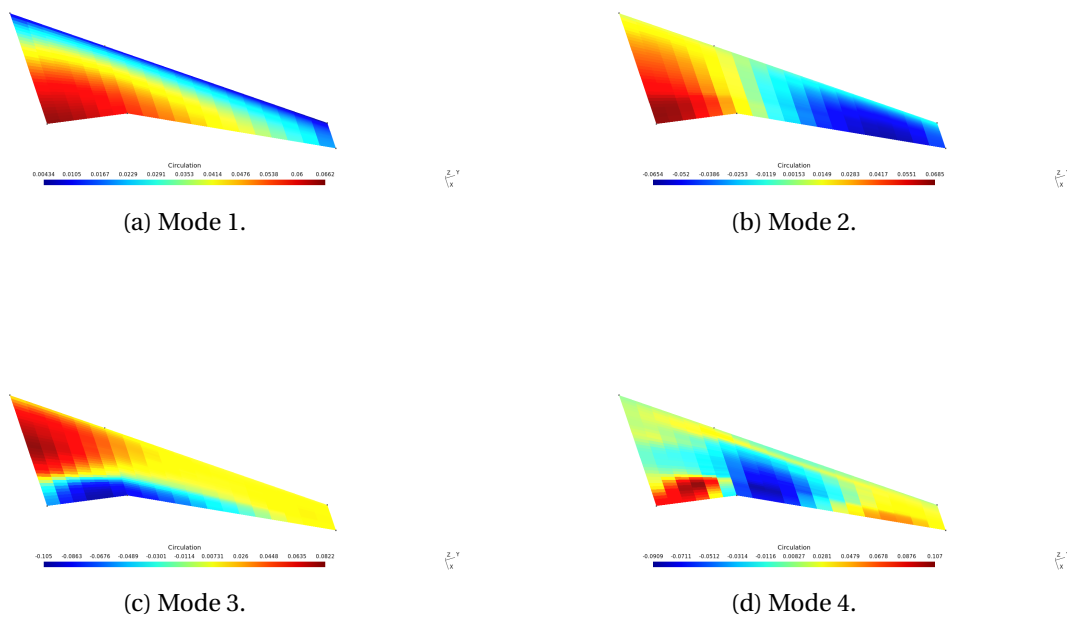


FIGURE 3.20 – Illustration des quatre premiers modes **POD** obtenus sur le solveur aérodynamique grâce à l'algorithme d'entraînement. Les nuances de couleur correspondent aux variations des amplitudes des vortex.

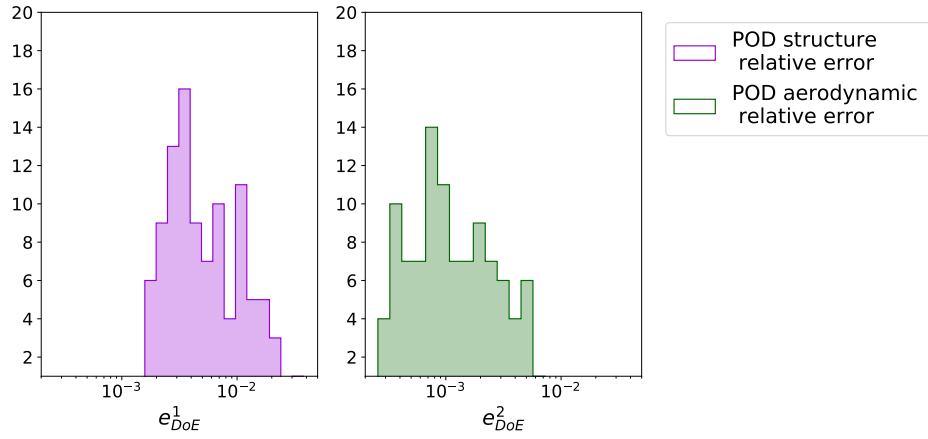


FIGURE 3.21 – Problème à 4 variables de design, histogramme de l'erreur relative entre la solution de la MDA et son approximation par POD : e^1_{DoE} et e^2_{DoE} sur l'ensemble de test.

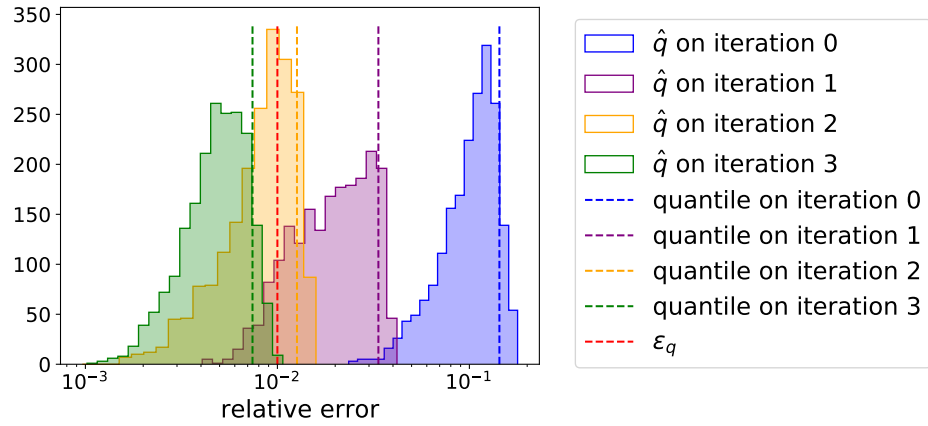


FIGURE 3.22 – Problème à 4 variables de design, histogramme et quantile à 90% de $\hat{q}$ au cours des itérations de l'Algorithme 13 lors de la résolution de la MDA $\mathbf{x} = \mathbf{x}_1^{DoE}$ avec le solveur DPOD+I.

Les informations obtenues au cours de l'étape d'entraînement sont utilisées pour le conditionnement des GP disciplinaires comme expliqué dans la Section 3.2.3.2. Comme première illustration, le solveur DPOD+I est utilisé pour résoudre la MDA au premier point de design de l'ensemble de test noté $\mathbf{x}_1^{DoE}$. L'histogramme et l'estimation du τ -quantile de la quantité $\hat{q}$ définis par Eq. (3.19) à travers les itérations sont illustrés sur la Figure 3.22. L'estimation du quantile est obtenue par MCS à partir d'un échantillon de taille 2000 soit autant de solutions de la MDA aléatoire générées. Sur cette figure, on remarque la convergence du τ -quantile de $\hat{q}$ vers le seuil souhaité ϵ_q . L'algorithme se termine après 4 itérations, ce qui implique 3 étapes d'enrichissement. Ces enrichissements sont un enrichissement structure suivi d'un enrichissement aérodynamique et enfin structure. Cela montre que la convergence vers la solution MDA est plus efficace qu'avec un solveur de MDA (type Jacobi) où 4 appels à chaque solveur disciplinaire sont requis. Seule la base POD de la structure a été améliorée lors de la dernière itération de l'algorithme.

Par la suite, la résolution pour les 100 variables de design de l'ensemble de test est obtenue à l'aide de l'Algorithme 13. Le nombre d'appels à chaque solveur disciplinaire nécessaire lors de la résolution de la MDA est indiqué sur la Figure 3.23. La résolution de la MDA avec le solveur DPOD+I nécessite, en moyenne, moins d'une évaluation d'un des solveurs disciplinaires permettant de réduire drastiquement le coût de calcul. Le nombre total d'appels aux solveurs disciplinaires nécessaires pour l'entraînement et la résolution

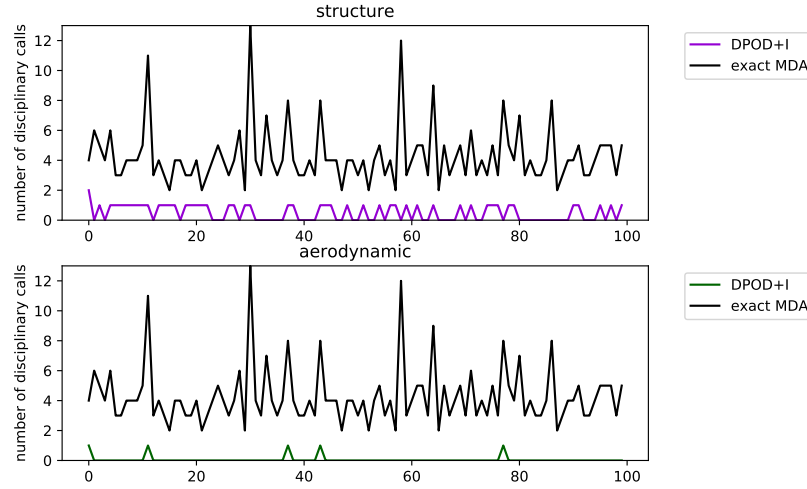


FIGURE 3.23 – Problème à 4 variables de design, nombre d'appels aux solveurs disciplinaires nécessaires à la résolution des 100 MDA de l'ensemble de test en utilisant le solveur DPOD+I.

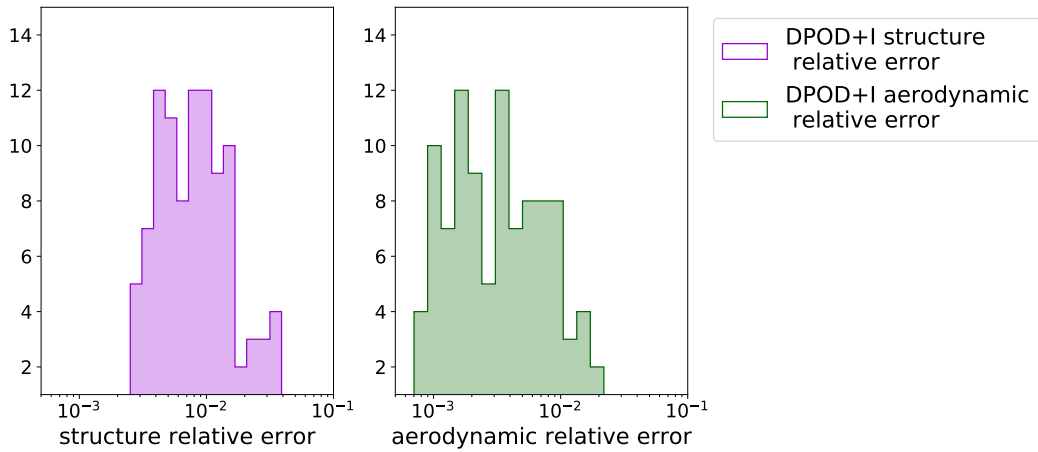


FIGURE 3.24 – Problème à 4 variables de design, histogramme de l'erreur relative entre la solution MDA exacte et la solution obtenue par le solveur DPOD+I sur l'ensemble de test.

est de 99 pour la structure et de 65 pour l'aérodynamique à comparer aux 429 appels à chaque solveur pour obtenir la solution des MDA exacte.

Finalement, une analyse sur la précision de la solution obtenue via le solveur DPOD+I est proposée. Pour ce faire, un histogramme de l'erreur relative entre la MDA exacte et la solution donnée par le solveur DPOD+I est donné dans la Figure 3.24. L'erreur relative pour la structure se situe entre 0.25 % et 4.8 % avec une moyenne de 1.1 %. L'erreur relative pour l'aérodynamique se situe entre 0.07 % et 2.1 % avec une moyenne de 0.46 %. Premièrement, la moyenne de l'erreur est proche de l'erreur cible qui était de 1%. Ensuite, il apparaît que certaines solutions MDA sont plus difficiles à approcher car leur erreur relative est supérieure à 2%, ceci est potentiellement dû à l'erreur entre la solution MDA réduite (obtenue par projection des solveurs disciplinaires sur les bases POD) et la solution de la MDA exacte. Le seuil de 2% est justifié par le fait que ϵ_{MDA} et ϵ_q sont fixés à 1% et correspond donc à deux erreurs opposées. De plus, la MDA exacte est résolue avec un critère d'arrêt fixé à 1%.

TABEAU 3.3 – Problème à 4 variables de design, minimum, maximum et moyenne de l'erreur relative pour la structure en utilisant la **DPOD+I** et la **POD+I** classique sur l'ensemble de test.

approche	min (%)	moyenne (%)	max (%)
DPOD+I	0.25	1.1	4.8
POD+I	0.4	3.5	28

TABEAU 3.4 – Problème à 4 variables design, minimum, maximum et moyenne de l'erreur relative pour l'aérodynamique en utilisant la **DPOD+I** et la **POD+I** classique sur l'ensemble de test.

approche	min (%)	moyenne (%)	max (%)
DPOD+I	0.07	0.46	2.2
POD+I	0.12	1.6	13

3.3.2.2 Comparaison avec la **POD+I** classique

Une comparaison avec une **POD+I** classique est maintenant présentée pour étudier la précision des deux méthodes lorsqu'un nombre égal de données est utilisé pour entraîner chaque modèle. Rappelons que, dans ce contexte, la **POD+I** estime directement la solution de la **MDA** tandis que la **DPOD+I** vise à créer un méta-modèle de chaque solveur disciplinaire et à obtenir une approximation de la solution de la **MDA** en couplant ces **GP** disciplinaires. Les détails de l'approche **POD+I** sont donnés dans la Section 3.2.6.

La comparaison est ensuite faite sur leur capacité à prédire la solution exacte de la **MDA**. Les bases **POD** et les **GP**, pour la méthode **DPOD+I**, sont entraînés sur un échantillon de solutions de la **MDA** de taille 25. Le nombre de solutions **MDA** utilisé (25) est choisi de manière à ce que le nombre d'appels aux solveurs disciplinaires utilisés pour la méthode **DPOD+I** corresponde au nombre d'appels aux solveurs disciplinaires utilisés pour la résolution des **MDA**. Ici, les résolutions des 25 **MDA** impliquent 99 appels à chaque solveur disciplinaire (à comparer aux 50 + 49 appels au solveur structure utilisé par la méthode **DPOD+I**). Ensuite, l'erreur relative entre la solution exacte et la solution donnée par la **POD+I** classique est calculée sur l'ensemble de test. La moyenne, le minimum et le maximum de l'erreur relative obtenue pour les 100 variables de design de l'ensemble de test sont donnés dans la Table 3.3 pour la discipline structure et dans la Table 3.4 pour la discipline aérodynamique. L'erreur moyenne sur les modèles structure et aérodynamique a été réduite d'un facteur 3. Le minimum de l'erreur relative est plus faible pour la stratégie **DPOD+I**. L'histogramme de l'erreur relative pour les deux disciplines est donné par la Figure 3.25. Pour la discipline aérodynamique, seules deux solutions dépassent le seuil de 1% d'erreur (soit ϵ_{MDA}). Pour la discipline structure, environ la moitié des erreurs sont supérieures à 1% et une dizaine sont supérieures à 2%.

Dans cette étude, nous avons montré que l'utilisation de solveurs disciplinaires permet, à enrichissement comparable, d'améliorer la précision des méta-modèles. L'idée principale est que l'entraînement via des solutions de la **MDA** est peu efficace pour l'approximation par méta-modèles car beaucoup d'informations ne sont pas utilisées (seule la dernière itération est utilisée pour la **POD+I**). Notre approche permet d'utiliser une plus grande proportion d'appels aux solveurs disciplinaires. Néanmoins, les résultats de cette analyse sont à nuancer car, même si le nombre d'appels aux solveurs est équivalent, la moitié des appels aux solveurs pour la méthode **DPOD+I** sont effectués lors de la résolution sur l'ensemble de test. Ces calculs sont donc adaptés à l'ensemble de test ce qui n'est pas le cas de la méthode **POD+I** classique. Dans la suite, une comparaison est proposée entre notre approche et une méthode **POD+I** à enrichissement.

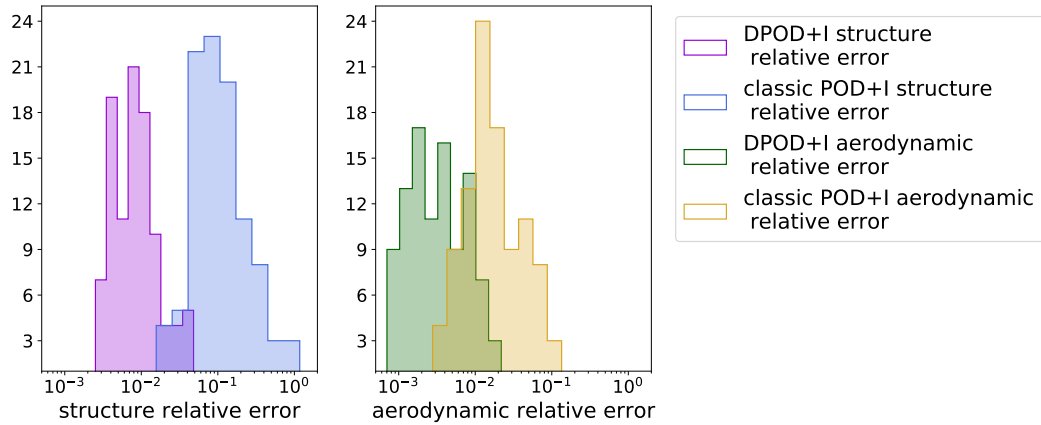


FIGURE 3.25 – Problème à 4 variables de design, histogramme de l'erreur relative sur la structure et l'aérodynamique sur l'ensemble de test avec l'approche **DPOD+I** et l'approche **POD+I** classique.

TABEAU 3.5 – Problème à 4 variables de design, nombre d'appels aux solveurs disciplinaires nécessaires à la résolution des **MDA** sur l'ensemble de test pour le solveur **MDA** classique, l'approche **DPOD+I** et l'approche **POD+I** enrichie.

méthode	appels au solveur structure		appels au solveur aérodynamique		Total
	<i>Offline</i>	<i>Online</i>	<i>Offline</i>	<i>Online</i>	
MDA exacte	0	429	0	429	858
DPOD+I	50	49	60	5	164
POD+I enrichie	99	251	99	251	700

3.3.2.3 Comparaison avec la **POD+I** enrichie

L'idée ici est de comparer le gain apporté par l'enrichissement pour chaque méthode. Un critère d'enrichissement similaire est utilisé pour les deux méthodes et elles sont comparées sur le budget nécessaire à la résolution des **MDA** sur l'espace de test. L'approximation **POD+I** initiale est égale à la **POD+I** classique présentée dans la Section 3.3.2.2 et est enrichie en utilisant l'Algorithme 15.

La comparaison est faite sur le nombre d'appels aux solveurs disciplinaires et les résultats sont donnés par la Table 3.5. La résolution des 100 **MDA** aux 100 points de l'espace de test via le solveur **MDA** a nécessité 429 appels à chaque solveur disciplinaire, la **POD+I** enrichie permet de réduire d'environ 20% le nombre d'appels au solveur disciplinaire (700 au lieu de 858). En revanche, le solveur **DPOD+I** réduit par trois le nombre d'appels au solveur structure et par cinq ceux du solveur aérodynamique par rapport à la **POD+I** enrichie.

Finalement, nous avons montré dans cette première étude que la **DPOD+I** est une méthode prometteuse par rapport à la **POD+I** sur ce problème à 4 variables de design. Cependant, comme la méthode proposée est probabiliste, une étude de robustesse est réalisée dans la suite.

3.3.2.4 Robustesse de l'algorithme d'entraînement

Dans la suite, nous proposons d'étudier l'influence du nombre r d'échantillons retenus dans l'Algorithme 11 et l'Algorithme 12. Bien que ce paramètre puisse être choisi par des contraintes logistiques comme le nombre d'appels aux solveurs disciplinaires pouvant être réalisés en parallèle, il influence la qualité du modèle initial. Comme le pro-

TABEAU 3.6 – Problème à 4 variables de design, influence de r sur le coût de l'algorithme d'entraînement : moyenne et Coefficient de Variation (CV) des appels aux solveurs disciplinaires sur 10 réalisations.

r	appels au solveur structure	CV (%)	appels au solveur aérodynamique	CV (%)
5	24.5	37	29.5	31
10	49	14	59	12
15	61.5	30	76.5	24
20	78	47	98	37

TABEAU 3.7 – Problème à 4 variables de design, influence de r sur le coût de l'algorithme d'entraînement : nombre moyen de modes POD conservés et erreurs relatives de projection sur l'ensemble de test lors des 10 réalisations.

r	nombre moyen de modes		erreur relative moyenne de projection	
	$\frac{1}{10} \sum_{j=1}^{10} (n^1)^j$	$\frac{1}{10} \sum_{j=1}^{10} (n^2)^j$	$\frac{1}{1000} \sum_{j=1}^{10} \sum_{i=1}^{100} (e_{DoE}^1)_i^j$	$\frac{1}{1000} \sum_{j=1}^{10} \sum_{i=1}^{100} (e_{DoE}^2)_i^j$
5	4.9	4.4	1.06%	0.23%
10	5.3	5.	0.74%	0.17%
15	5.3	5.	0.81%	0.17%
20	5.3	5.	0.78%	0.17%

cessus est aléatoire, 10 réalisations de l'algorithme d'apprentissage sont effectuées. Les valeurs testées pour r sont 5, 10, 15 et 20. La Table 3.6 présente le nombre moyen d'appels disciplinaires pour atteindre la convergence des étapes d'entraînement pour chaque valeur de r et leur CV associé. Tout d'abord, il apparaît qu'une valeur r élevée tend à augmenter le nombre d'appels disciplinaires pour l'algorithme d'apprentissage. Ainsi, cette valeur r doit être maintenue aussi petite que possible. Cependant, si la base POD obtenue et les GP ne sont pas assez entraînés, cela se traduit par une phase d'enrichissement plus coûteuse. Il est à noter que la valeur 10 semble être la meilleure valeur en terme de stabilité car elle minimise la variation des appels de solveurs disciplinaires nécessaires. Avec r fixé à 20, la variation est la plus élevée en raison des phases d'entraînement qui nécessitent jusqu'à 200 appels pour le solveur structure et 210 pour l'aérodynamique.

Pour chaque réalisation j de l'algorithme d'entraînement, nous notons $(n^1)^j$ et $(n^2)^j$ le nombre de vecteurs POD retenus pour chaque discipline. La moyenne de $(n^1)^j$ et $(n^2)^j$ sur les 10 réalisations est affichée dans la Table 3.7 en fonction du paramètre r . Ensuite, les erreurs relatives de projection sur chaque base $(e_{DoE}^1)_i^j$, $i = 1, \dots, 100$, $j = 1, \dots, 10$ et $(e_{DoE}^2)_i^j$, $i = 1, \dots, 100$, $j = 1, \dots, 10$ sont calculées sur l'ensemble de test, pour chaque réalisation. Leurs moyennes sont présentées dans la Table 3.7. Il apparaît que la valeur de r n'influence pas la qualité de la base POD sauf pour la plus petite valeur $r = 5$. Dans ce cas, la base POD manque d'informations pour être suffisamment précise. L'influence de ce manque de précision initiale sera analysée dans la suite lors de la résolution des MDA sur l'espace de test.

Pour chaque valeur r , 10 réalisations de la phase d'entraînement ont été générées. Ensuite, pour chaque modèle entraîné, la résolution de la MDA en utilisant le solveur DPOD+I sur l'ensemble de test est effectuée. Pour chaque réalisation, le nombre d'appels totaux de chaque solveur disciplinaire comptant les appels issus des phases d'entraînement et d'enrichissement est calculé. La moyenne et le CV des appels aux solveurs disciplinaires sur les 10 réalisations sont présentés dans la Table 3.8. Le nombre total d'appels au solveur structure n'est que faiblement impacté par le paramètre r alors que le

TABEAU 3.8 – Problème à 4 variables de design, influence de r sur le coût de calcul total : nombre moyen d'appels et CV pour chaque solveur disciplinaire après les phases d'entraînement et d'enrichissement sur l'ensemble de test en utilisant le solveur DPOD+I sur les 10 réalisations.

r	appels au solveur structure	CV (%)	appels au solveur aérodynamique	CV (%)
5	88.8	15	45.8	21
10	89.5	8	66	10
15	92.5	10	82.1	22
20	105.9	27	107.7	35

TABEAU 3.9 – Problème à 4 variables de design, influence de r sur la précision du solveur DPOD+I : erreur relative moyenne sur les deux disciplines et CV associé sur l'ensemble de tests sur les 10 réalisations.

r	erreur relative moyenne sur la structure (%)	CV (%)	erreur relative moyenne sur l'aérodynamique (%)	CV (%)
5	1.3	10	0.61	17
10	1.1	6	0.42	10
15	1.2	14	0.43	17
20	1.1	15	0.38	15

nombre total d'appels au solveur aérodynamique augmente significativement avec r . En effet, nous avons observé que, sur cet exemple, pour le solveur aérodynamique, le nombre d'appels disciplinaires pour la phase d'enrichissement est négligeable par rapport à la phase d'entraînement qui est fortement influencée par le paramètre r comme le montre la Table 3.6. Au contraire, pour le solveur structure, plus de points sont ajoutés lors de la phase d'enrichissement ce qui lisse l'impact de la phase d'entraînement initiale.

Concernant la précision sur la solution obtenue par le solveur DPOD+I, l'erreur relative sur l'ensemble de test est calculée pour chaque discipline. La moyenne et CV des erreurs sur les 10 réalisations sur l'ensemble de test sont affichés dans la Table 3.9. Le manque de précision sur les bases POD avec r fixé à 5 a un impact sur la précision des solutions données par le solveur DPOD+I. Ainsi, même si les bases sont enrichies au fur et à mesure des itérations de l'Algorithme 12, la précision des bases obtenues après l'entraînement a un impact sur la précision du solveur. Pour les autres valeurs de r , la différence reste négligeable. En ce qui concerne le CV sur l'erreur moyenne commise, la valeur 10 reste la plus intéressante.

Enfin, une des remarques faites en Section 3.2.5.1 concernait le fait que les solutions de la MDA aléatoire ne convergent pas lorsque les GP sont mal entraînés. Avec r défini à 5, le nombre d'échecs de la résolution de la MDA est d'environ 1%, ce qui est acceptable. Pour les autres, les échecs sont trop rares pour être remarqués : moins de 0.2%.

Pour terminer cette analyse, le nombre moyen d'appels disciplinaires pour chaque valeur de r est affiché dans la Figure 3.26 en fonction du nombre de résolutions de la MDA. Nous pouvons observer sur cette figure que le nombre d'appels au solveur aérodynamique est très peu important : les sauts visibles sur ces courbes sont uniquement dus à des échecs de la résolution de la MDA où le solveur itératif a été utilisé. Nous pouvons observer que quelque soit la valeur de r , les échecs sont présents aux points 37 et 66. Pour la discipline structure, on peut observer que le nombre d'appels au solveur pour la résolution de chaque MDA diminue au fur et à mesure que le nombre de résolutions de la MDA augmente grâce à l'enrichissement du modèle initial. Moins de 5 appels au solveur structure sont effectués lors de la résolution des 20 dernières MDA.

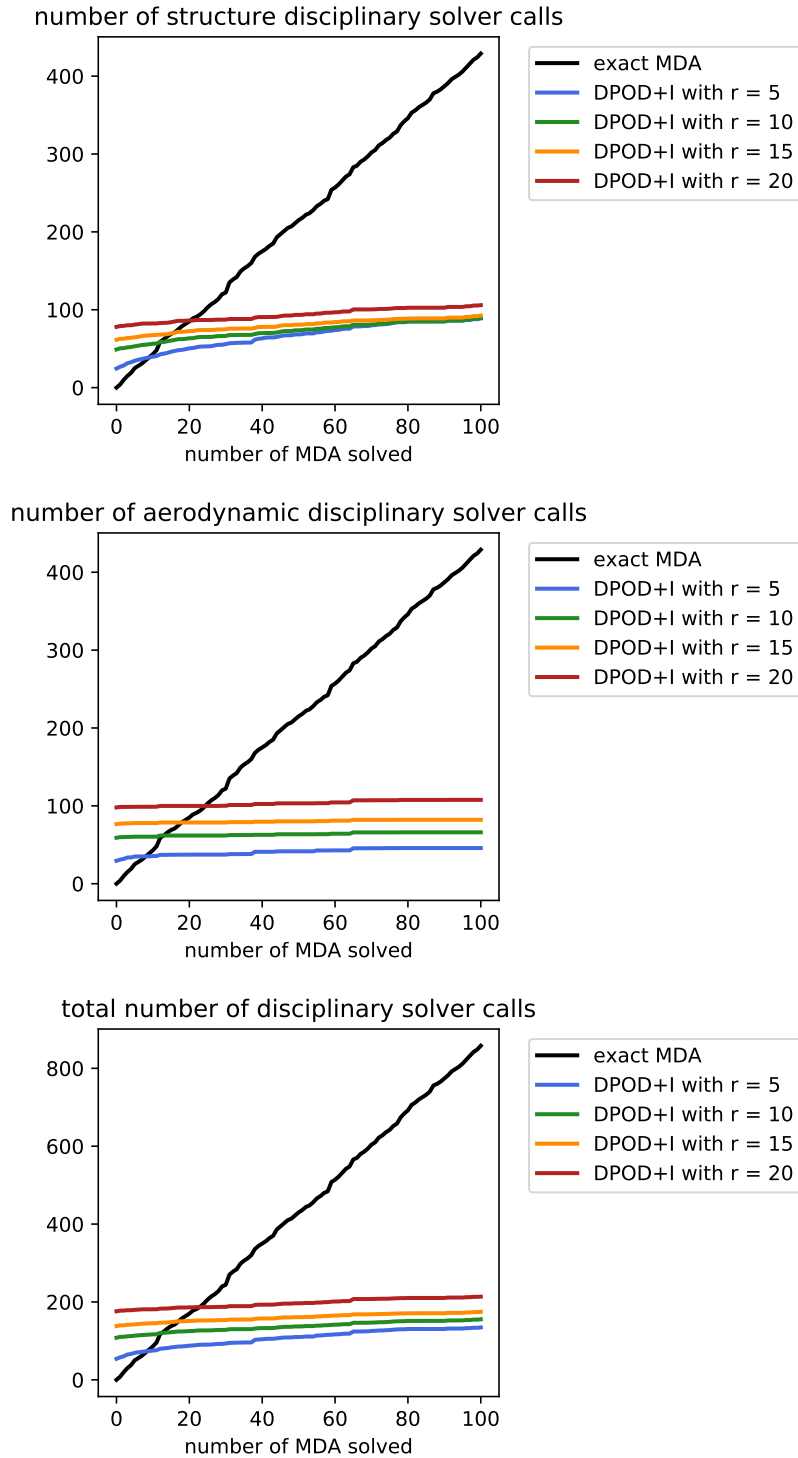


FIGURE 3.26 – Problème à 4 variables de design, étude de robustesse. Nombre d'appels aux solveurs disciplinaires au cours des résolutions sur l'ensemble de test avec r fixé à 5, 10, 15 et 20.

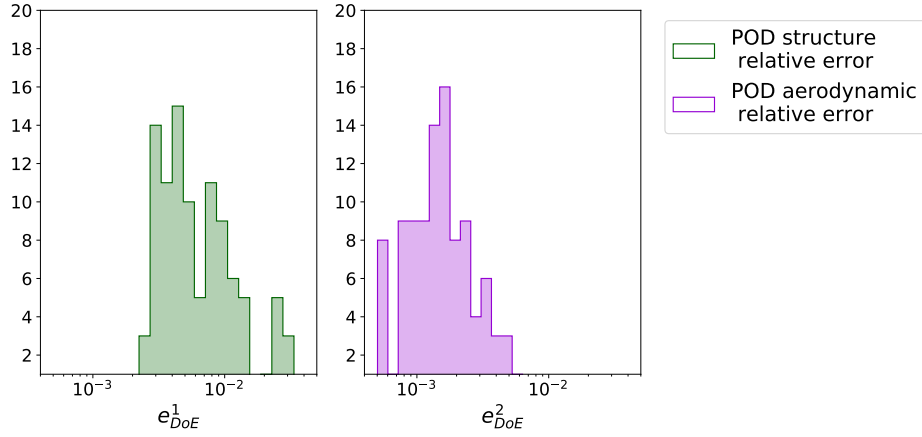


FIGURE 3.27 – Problème à 8 variables de design, histogramme de l'erreur relative entre la solution de la MDA et son approximation par POD : e^1_{DoE} et e^2_{DoE} sur l'ensemble de test.

3.3.3 Problème à 8 variables de design

3.3.3.1 Résolution avec le solveur DPOD+I

La même analyse est proposée pour le problème à 8 variables de design (cf Table 3.2) afin de tester notre approche sur un cas test plus complexe. Le nombre r est toujours fixé à 10. L'algorithme d'entraînement se termine également après $n = 5$ itérations, ce qui implique 50 appels au solveur structure et 60 appels au solveur aérodynamique. Les tailles des bases POD sont de $n^1 = 11$ pour la structure et de $n^2 = 5$ pour l'aérodynamique. Le nombre de modes POD pour l'aérodynamique reste constant mais augmente pour la structure par rapport au problème précédent. En effet, la complexité de la réponse du solveur structure est augmentée en raison d'un nombre plus élevé de variables de design associées. Dans un premier temps, la qualité de la base POD obtenue est analysée au moyen des erreurs relatives entre la solution de la MDA exacte et sa projection définie dans l'Eq. (3.30). Les histogrammes de ces quantités sont présentés dans la Figure 3.27. L'erreur relative pour le modèle POD de la structure se situe entre 0.22 % et 3.4 % avec une moyenne de 0.83 %. L'erreur relative pour le modèle POD aérodynamique se situe entre 0.05 % et 0.6 % avec une moyenne de 0.18 %. L'approximation POD est toujours précise même avec l'augmentation du nombre de variables de design.

Après l'étape d'entraînement, les itérations de la méthode DPOD+I sont illustrées sur la première variable de design du DoE x_1^{DoE} . L'histogramme et le τ -quantile de la quantité $\hat{q}$ sur les 2000 solutions de la MDA aléatoire au cours des itérations sont montrés sur la Figure 3.28. Sur cette figure, on observe la convergence du τ -quantile de $\hat{q}$ jusqu'à atteindre le critère ϵ_q . L'algorithme se termine après 3 itérations, ce qui implique 2 étapes d'enrichissement qui correspondent à un enrichissement du solveur structure suivi d'un enrichissement du solveur aérodynamique.

Ensuite, la résolution pour les 100 variables de design de l'espace de test est obtenue à l'aide du solveur DPOD+I. Le nombre d'appels à chaque solveur disciplinaire nécessaire à la résolution de chaque MDA est indiqué dans la Figure 3.29. La résolution des MDA avec le solveur DPOD+I nécessite moins d'un appel à chaque solveur disciplinaire à l'exception d'une résolution qui en a nécessité 2. Même avec un nombre plus élevé de variables de design, le nombre d'appels disciplinaires par résolution reste faible. Le nombre total d'appels de solveurs disciplinaires nécessaires pour l'entraînement et les résolutions est de 128 pour la structure et de 66 pour l'aérodynamique.

Pour analyser l'erreur faite par le solveur DPOD+I, l'histogramme de l'erreur relative

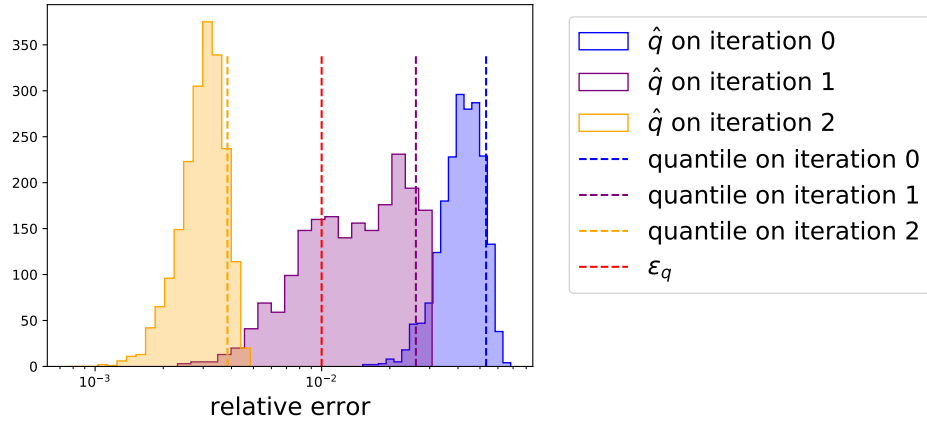


FIGURE 3.28 – Problème à 8 variables de design, histogramme et quantile à 90% de $\hat{q}$ au cours des itérations de l'Algorithme 13 lors de la résolution de la MDA $\mathbf{x} = \mathbf{x}_1^{DoE}$ avec le solveur DPOD+I.

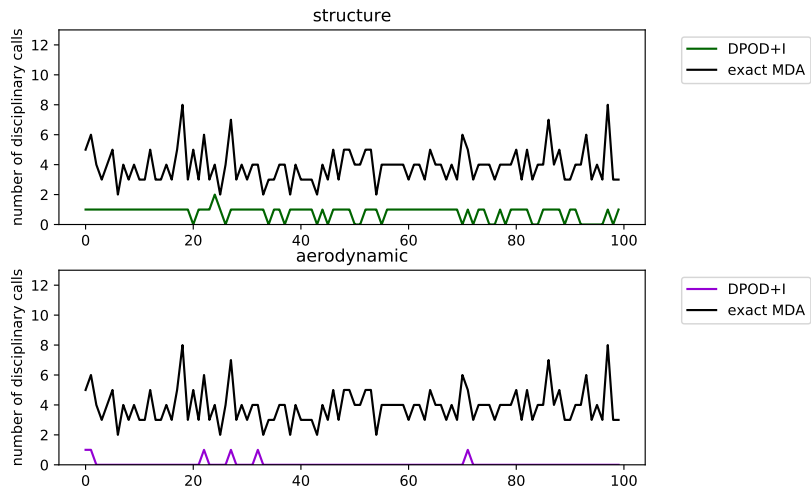


FIGURE 3.29 – Problème à 8 variables de design, nombre d'appels aux solveurs disciplinaires nécessaires à la résolution des 100 MDA de l'ensemble de test en utilisant le solveur DPOD+I.

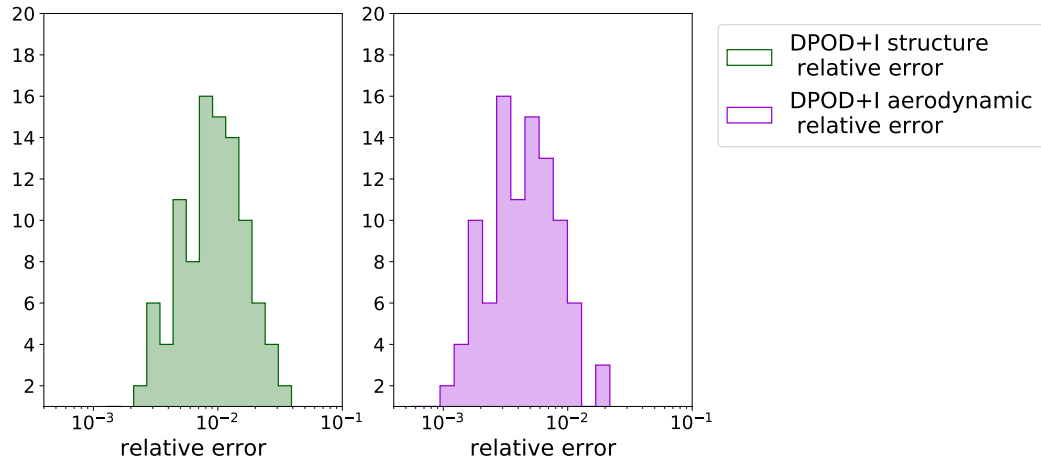


FIGURE 3.30 – Problème à 8 variables de design, histogramme de l'erreur relative entre la solution **MDA** exacte et la solution obtenue par le solveur **DPOD+I** sur l'ensemble de test.

TABLEAU 3.10 – Problème à 8 variables design, minimum, maximum et moyenne de l'erreur relative pour la structure en utilisant la **DPOD+I** et la **POD+I** classique sur l'ensemble de test.

approche	min (%)	moyenne (%)	max (%)
DPOD+I	0.13	1.1	3.4
POD+I	1.6	16	117

entre la solution de la **MDA** exacte et la solution donnée par le solveur **DPOD+I** est présenté dans la Figure 3.30. L'erreur relative sur la discipline structure se situe entre 0.13% et 3.4% avec une moyenne de 1.1%. L'erreur relative sur la discipline aérodynamique se situe entre 0.05% et 2.2% avec une moyenne de 0.55%. La moyenne de l'erreur est toujours proche ou inférieure à de l'erreur cible de 1%.

3.3.3.2 Comparaison avec la **POD+I** classique

La même méthodologie que précédemment est appliquée afin de comparer notre approche avec la **POD+I** classique. Dans le problème à 8 variables de design, 30 solutions de la **MDA** sont utilisées car le nombre total d'appels aux solveurs disciplinaires nécessaires pour le solveur **DPOD+I** sur le nouveau problème est de 128. Ici, la résolution des 30 **MDA** implique 121 appels aux solveurs disciplinaires. L'erreur relative entre la solution exacte et la solution donnée par les valeurs moyennes des **GP** de la **POD+I** classique est calculée sur l'ensemble de test. La moyenne, le minimum et le maximum de l'erreur relative obtenue lors de l'approximation des **MDA** sur l'espace de test sont indiqués dans la Table 3.10 pour la structure et dans la Table 3.11 pour l'aérodynamique. L'erreur moyenne sur l'aérodynamique est réduite d'un facteur 5 et d'un facteur 15 pour la structure. Le minimum de l'erreur relative est plus faible pour la stratégie **DPOD+I**. Cet exemple montre les limites de la stratégie **POD+I** lorsque le nombre de variables de design augmente. En effet, un plus grand nombre de données est nécessaire pour entraîner les **GP** de la **POD+I**. La **DPOD+I** est moins sensible à ce phénomène. L'historique de l'erreur relative pour les deux disciplines sur l'ensemble de test est donné par la Figure 3.31.

TABEAU 3.11 – Problème à 8 variables de design, minimum, maximum et moyenne de l'erreur relative pour l'aérodynamique en utilisant la DPOD+I et la POD+I classique sur l'ensemble de test.

approche	min (%)	moyenne (%)	max (%)
DPOD+I	0.05	0.55	2.2
POD+I	0.3	2.6	13

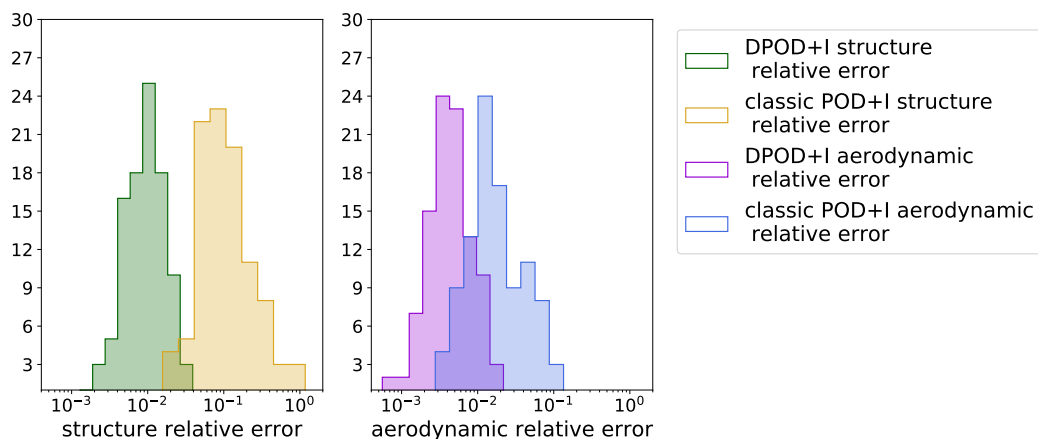


FIGURE 3.31 – Problème à 8 variables design, histogramme de l'erreur relative sur la structure et l'aérodynamique sur l'ensemble de test avec l'approche DPOD+I et l'approche POD+I classique.

3.3.3.3 Comparaison avec la POD+I enrichie

Nous allons à présent comparer la stratégie développée dans cette thèse avec la POD+I enrichie. Le nombre d'enrichissements nécessaires est donné par la Table 3.12. Pour l'approximation des MDA sur l'espace de test, le solveur MDA nécessite 392 appels à chaque solveur disciplinaire, la POD+I enrichie ne permet pas de réduire le nombre d'appels aux solveurs disciplinaires. En effet, il n'y a que quatre points de l'ensemble de test pour lesquelles les critères quantiles sont remplis. Le coût de calcul total est donc plus élevé en raison de la phase d'apprentissage. Par contre, le solveur DPOD+I réduit d'un facteur 1.5 le nombre d'appels au solveur structure et d'un facteur 6 le nombre d'appels au solveur aérodynamique par rapport au solveur MDA. Enfin, avec cet exemple impliquant un plus grand nombre de variables de design, nous avons montré les limites de la POD+I enrichie par rapport au solveur DPOD+I. La précision est améliorée lors de l'utilisation de méta-modèles disciplinaires plutôt que l'utilisation de méta-modèles sur les variables de couplage convergées.

TABEAU 3.12 – Problème à 8 variables de design, nombre d'appels aux solveurs disciplinaires nécessaires à la résolution des MDA sur l'ensemble de test pour le solveur MDA classique, l'approche DPODI et l'approche PODI enrichie.

approche	appels au solveur structure		appels au solveur aérodynamique		Total
	Offline	Online	Offline	Online	
MDA exacte	0	392	0	392	784
DPOD+I	50	78	60	6	194
POD+I enrichie	121	378	121	378	998

3.3.4 Conclusion sur le premier cas test

Nous avons présenté une nouvelle méthodologie adaptative basée sur des méta-modèles disciplinaires pour résoudre des MDA faisant intervenir des variables de couplage de grande dimension. Des approximations POD disciplinaires ont été proposées pour réduire leurs dimensions ce qui conduit à une MDA réduite définie par le problème (3.11). Ensuite, l'idée est d'interpoler les solveurs disciplinaires réduits par des GP. Cette DPOD+I construit des méta-modèles indépendants pour chaque solveur disciplinaire permettant de découpler la MDA, ce qui peut être avantageux dans un contexte industriel. L'un des défis de cette approche est de découpler l'entraînement des modèles DPOD+I car l'utilisation de MDA exactes conduit à un DoE mal distribué comme l'a montré la Figure 3.7. Pour ce faire, une nouvelle stratégie d'entraînement a été développée sans connaissance initiale sur les plages de variation des variables de couplage. Enfin, un nouveau solveur MDA est développé pour résoudre le problème couplé par des enrichissements adaptatifs successifs. Ce solveur, comprend une estimation de l'erreur. Si cette erreur est trop élevée, un enrichissement de la discipline qui a le plus d'influence sur la quantité d'intérêt, selon l'estimation des indices de sensibilité de Sobol, est effectué.

Ces approches ont été testées sur un cas test aéroélastique et confirment l'intérêt de la méthodologie proposée. La nouvelle méthode DPOD+I est comparée à l'approche POD+I classique. Avec un nombre comparable d'évaluations de solveurs disciplinaires pour l'entraînement, le DPOD+I surpasse la POD+I en terme de précision. Lorsque la même stratégie d'enrichissement est appliquée à la méthodologie POD+I, la DPOD+I surpasse la POD+I enrichie en termes de nombre d'évaluations des solveurs disciplinaires. En effet, cette nouvelle stratégie permet l'enrichissement de chaque méta-modèle disciplinaire impliquant une réduction du coût d'enrichissement. L'un des défis est la détermination des paramètres de la DPOD+I. La tolérance sur l'approximation POD ϵ_{PE} et la tolérance sur l'interpolation ϵ_q sont liées à la tolérance cible de la résolution de la MDA notée ϵ_{MDA} . Le paramètre le plus difficile à fixer, a priori, est le paramètre d'apprentissage r mais nous avons montré que son influence sur la précision de la méthode DPOD+I est lissée par la phase d'enrichissement. Enfin, la DPOD+I a montré une bonne robustesse lors de l'augmentation du nombre de variables de conception contrairement à la POD+I.

3.4 Application sur un deuxième cas aéroélastique

Il est proposé maintenant de tester la DPOD+I sur un cas aéroélastique plus complexe faisant intervenir un solveur aérodynamique non linéaire. Le solveur utilisé pour le calcul des efforts aérodynamiques utilise les équations d'Euler résolues par volumes finis à la place de la méthode VLM détaillée sur le cas précédent. Ce cas test a été en partie développé par David Perez Sancha, stagiaire en M1 de l'école ISAE-Supaero de février à juillet 2021 sous la supervision de Sylvain Dubreuil, Thierry Lefebvre et moi-même pour un cas d'analyse mission. Cependant, la méthode proposée pour le cas précédent ne pourra pas être mise en oeuvre directement et des modifications seront alors proposées afin de palier aux difficultés rencontrées.

3.4.1 Description du modèle des équations d'Euler

Les équations d'Euler sont une simplification des équations de Navier-Stokes aux fluides compressibles lorsque la viscosité et la conductivité thermique sont négligées. On cherche

alors la solution $\{\rho, \rho \mathbf{v}, \rho E\}$ où ρ est la densité du fluide, $\mathbf{v} \in \mathbb{R}^3$ est la vitesse du fluide et E est la quantité d'énergie par unité de masse respectant les équations suivantes :

— l'équation de continuité :

$$\frac{\partial \rho}{\partial t} + \text{div}(\rho \mathbf{v}) = 0 \quad (3.31)$$

où $\text{div}(\cdot)$ est l'opérateur divergence.

— l'équation de bilan de la quantité de mouvement :

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \rho (\mathbf{v} \otimes \mathbf{v}) = -\nabla p + \rho \mathbf{g} \quad (3.32)$$

où p est la pression statique et $\otimes$ le produit dyadique.

— l'équation de bilan de l'énergie :

$$\frac{\partial \rho E}{\partial t} + \text{div}(\rho E \mathbf{v}) = -\text{div}(\rho \mathbf{v}) + \rho \mathbf{g} \mathbf{v} \quad (3.33)$$

Ces équations sont alors résolues par le solveur volumes finis *open source* SU2 v5.0 "Raven" (ECONOMON et collab. [2016]). Notons que nous nous intéresserons à la solution stationnaire de cette équation.

3.4.2 Paramétrisation du solveur aérostructure

Dans ce cas test, nous allons prendre en compte plusieurs paramètres permettant de modéliser une mission de vol classique. Ainsi, nous allons faire varier l'angle d'attaque, la quantité de carburant restant dans l'avion ainsi que la vitesse de l'avion. Il est alors possible, en faisant varier ces trois paramètres de calculer la performance de l'avion sur un vol de croisière. Notons que l'altitude est aussi un des paramètres du modèle mais nous avons décidé de simplifier ce cas test en supposant une altitude constante (croisière).

Afin de prendre en compte la quantité de carburant, le poids du carburant est ajouté au chargement sur le solveur structure. Pour simplifier, nous supposons qu'il est uniformément réparti dans l'aile en fonction du volume disponible et que le chargement est réparti uniformément sur les longerons et les nervures en fonction de la surface de ces éléments. Le poids des moteurs est lui aussi pris en compte.

3.4.3 Maillages

Le maillage du solveur structure reste inchangé. Le maillage aérodynamique est modifié pour passer d'un maillage en deux dimensions à un maillage en trois dimensions décrivant l'air autour de l'aile. Ce maillage contient environ 7 millions d'éléments. Le maillage de la surface de l'aile est illustré sur la Figure 3.32 et fait intervenir 148 604 noeuds conduisant à des efforts aérodynamiques de dimension $d^2 = 445\,812$. Le géométrie de l'aile a été calquée sur le modèle de recherche u-CRM (BROOKS et collab. [2017]). Un profil d'aile supercritique est utilisé pour maintenir un nombre de Mach plus faible sur sa surface supérieure qu'une voilure conventionnelle induisant une moindre onde de choc. La discrétisation en espace pour la résolution des équations d'Euler est effectuée par une méthode de Volumes Finis (HIRSCH [2007]).

La pression stationnaire obtenue sur la surface de l'aile au centre de l'espace de design est illustrée sur l'extrados sur la Figure 3.33 et sur la Figure 3.34 pour l'intrados. Nous pouvons observer sur la Figure 3.33 une décompression très importante au niveau du

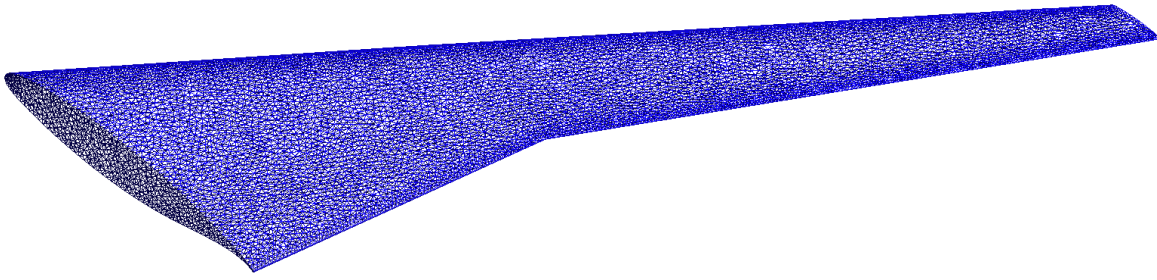


FIGURE 3.32 – Maillage extérieur de l'aile utilisé pour la résolution des équations d'Euler avec le solveur SU2.

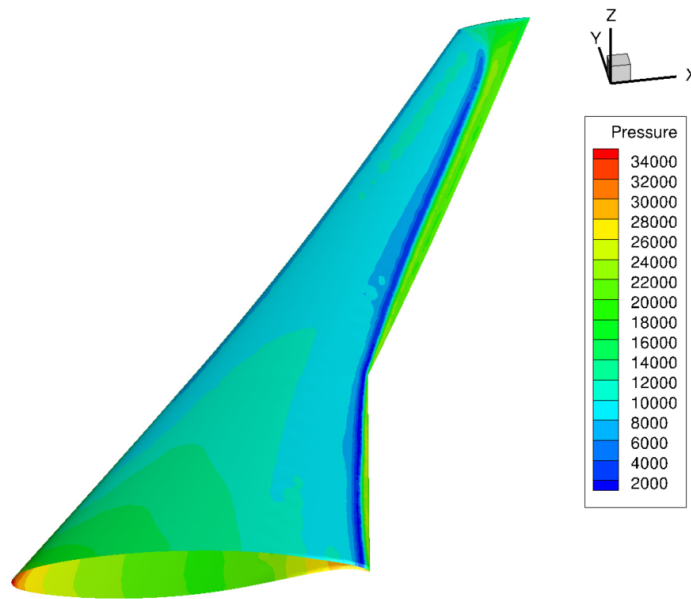


FIGURE 3.33 – Illustration de la pression (solution stationnaire) sur la surface supérieure de l'aile obtenue à partir de la résolution des équations d'Euler au centre de l'espace de design.

bord de fuite sur l'extrados. Cette décompression semble anormalement importante pour un profil supercritique et peut être causée par une déformation trop importante du profil de l'aile au niveau du bord de fuite. Cette déformation rend le calcul aérodynamique très complexe et peut expliquer la difficulté de l'approximation par **POD** de cette discipline comme nous le verrons dans la suite.

3.4.4 Difficultés du modèle

Plusieurs difficultés interviennent lors de la résolution de ce modèle. Dans un premier temps, le temps de calcul est très important avec environ 20 min pour un appel au solveur aérodynamique sur 48 coeurs Broadwell Intel Xeon E5-2650 cadencés à 2.20GHz. Ceci nous a fortement limité dans notre étude. En effet, la résolution d'une **MDA** nécessitant entre 5 et 20 itérations, cela représentait entre 1h40min et 6h40min de temps de calcul. Ceci rend les tests permettant de valider notre approche très difficiles. De plus, nous nous sommes aperçus que l'algorithme de Gauss-Seidel n'avait pas convergé pour certaines variables de design. La condition initiale est l'un des facteurs sur lequel nous nous sommes appuyés afin de pouvoir faire converger l'algorithme. Le vecteur initialisant l'Algorithme de Gauss Seidel y_0^1 a été modifié dans les cas où cela était nécessaire. De

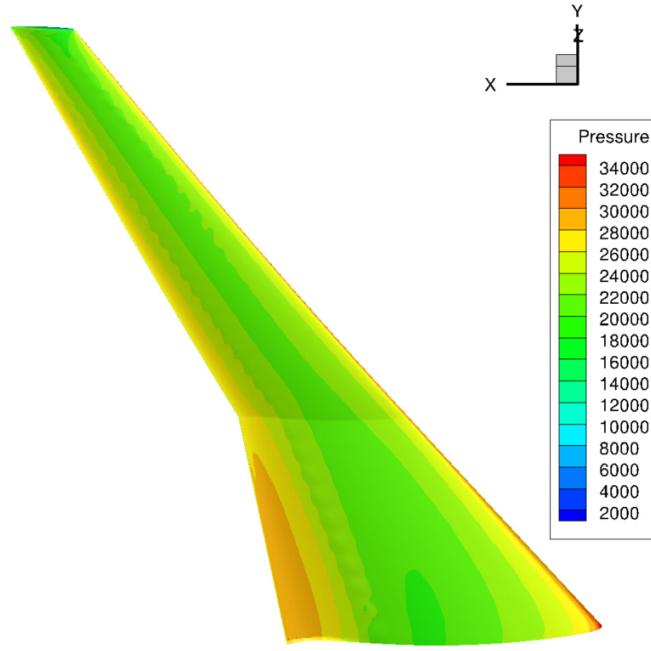


FIGURE 3.34 – Illustration de la pression (solution stationnaire) sur la surface inférieure de l'aile obtenue à partir de la résolution des équations d'Euler au centre de l'espace de design.

plus, la convergence de la MDA est lente et il est difficile d'obtenir une solution convergée avec $\epsilon_{\text{MDA}} = 1\%$ comme cela a été proposé dans le cas test précédent. Nous décidons donc de relâcher ce critère en changeant ϵ_{MDA} à 3%.

Nous avons analysé des conditions de vol transsonique (Mach compris entre 0.74 et 0.86) conduisant à des phénomènes très difficiles à approcher par POD. Nous allons voir pourquoi la difficulté d'approcher les solveurs disciplinaires par POD est un challenge pour la DPOD+I. Premièrement, la base POD est plus difficile à apprendre ce qui augmenter le coût de l'algorithme d'entraînement (cf Algorithme 12). En effet, pour obtenir une erreur moyenne de 1% sur la POD, en fixant le pourcentage η d'énergie captée pour la construction de la base POD à 99.99%, 10 itérations sont nécessaires imposant 100 appels au solveur structure et 110 appels au solveur aérodynamique, soit plus de 35 heures de temps de calcul. Deuxièmement, le nombre de modes pour la base POD est très important avec $n^1 = 77$ vecteurs pour la discipline aérodynamique et $n^2 = 9$ pour la discipline structure. Ainsi, le nombre d'entrées de la discipline structure est de $77 + 1 = 78$ rendant cette discipline difficile à approcher. Pour y parvenir, nous utilisons une extension des GP basée sur une projection des entrées pour réduire leur dimension par une méthode Moindres carrés partiels, ou *Partial Least Square* (PLS) (BOUHLEL et collab. [2016]).

Remarque : Une solution serait de limiter le nombre de vecteurs de la base POD mais en pratique cela peut conduire à des difficultés. En effet, soit α^1 des coordonnées sur la base POD de la première discipline (l'aérodynamique par exemple). Le vecteur reconstitué est donné par :

$$\mathbf{y}^1 = \Phi_0^1 + \sum_{j=1}^{n^1} \alpha_j^1 \Phi_j^1$$

Soit à présent un paramètre de design $\mathbf{x}$ quelconque et $\delta \mathbf{y}^1 \in \mathbb{R}^{d^1}$ un vecteur orthogonal à la base POD de la première discipline ($\langle \delta \mathbf{y}^1, \Phi_j^1 \rangle = 0, 1 \leq j \leq n^1$). Les projections des vecteurs sur la base POD de $\mathbf{y}^1$ et $\mathbf{y}^1 + \delta \mathbf{y}^1$ sont similaires puisque la différence est orthogonale à la base POD. Nous définissons alors $\mathbf{y}^2 = f^2(\mathbf{x}, \mathbf{y}^1)$ et $\mathbf{y}_b^2 = f^2(\mathbf{x}, \mathbf{y}^1 + \delta \mathbf{y}^1)$.

Les projections sur la deuxième base **POD** des solutions des solveurs sont respectivement α^2 et α_b^2 . Comme les entrées du solveur sont différentes, on peut supposer que $\alpha^2 \neq \alpha_b^2$. Supposons maintenant que ces deux appels au solveur soient utilisés pour l'entraînement des **GP** disciplinaires. Dans ce cas, le premier point d'apprentissage (issu de y^1) sera $x, \alpha^1 \rightarrow \alpha^2$, le deuxième point (issu de $y^1 + \delta^1$) sera $x, \alpha^1 \rightarrow \alpha_b^2$. Ainsi, la fonction réduite que le **GP** disciplinaire essaie d'apprendre n'est pas une fonction puisque deux entrées similaires peuvent avoir des sorties différentes. Ce cas, très particulier, n'est jamais atteint exactement. Cependant, lorsque nous avons choisi un nombre réduit de modes dans la base **POD**, l'apprentissage des **GP** disciplinaires lors de la résolution via l'Algorithme 13, conduit à des solutions de la **MDA** approchée aberrantes. Nous avons donc dû conserver un nombre important de modes **POD** afin de limiter l'erreur commise sur les entrées des **GP** disciplinaires.

3.4.5 Convergence de la **MDA** aléatoire

Une des difficultés que nous avons observées en utilisant la **DPD+I** sur ce problème est la convergence de la **MDA** aléatoire. En effet, le nombre important de modes **POD** conduit à beaucoup de sources d'incertitude lors de la génération des solutions de la **MDA** aléatoire. Pour rappel, une solution de la **MDA** aléatoire est donnée par

$$\begin{cases} \alpha_i^1 = \mu_i^1(x_*, \alpha^2) + \xi_i^1 \sigma_i^1(x_*, \alpha^2) & \forall i = 1, \dots, n^1 \\ \alpha_j^2 = \mu_j^2(x_*, \alpha^1) + \xi_j^2 \sigma_j^2(x_*, \alpha^1) & \forall j = 1, \dots, n^2 \end{cases} \quad (3.34)$$

où $\xi^1 \in \mathbb{R}^{n^1}$ et $\xi^2 \in \mathbb{R}^{n^2}$ sont de la dimension des bases **POD** correspondantes. Faire converger le coefficient de variation des solutions de la **MDA** aléatoire nécessite des variances des **GP** disciplinaires d'autant plus faibles que le nombre de vecteurs dans la base **POD** augmente. Nous avons observé lors de notre étude que, même si la solution donnée par la moyenne des **GP** disciplinaires (Eq. (3.15)) était proche de la solution de la **MDA** projetée, le coefficient de variation de $\hat{q}$ est très important. De plus, l'écart de la taille des deux disciplines biaise l'analyse de sensibilité. En effet, comme le nombre de variables aléatoires utilisées pour évaluer l'incertitude d'une discipline est égal à la taille de la base, plus le nombre de modes **POD** est important, plus la discipline contient de sources d'incertitude. La discipline aérodynamique, au vu de la différence de taille des bases **POD** était systématiquement enrichie. Nous avons alors proposé deux méthodes permettant de pallier ce problème.

3.4.5.1 Limitation des sources d'incertitude de la **MDA** aléatoire

La première idée est de limiter les dimensions de ξ^1 et ξ^2 utilisées pour générer les solutions de la **MDA** aléatoire. Pour cela, nous proposons, dans un souci d'équité entre les sources aléatoires utilisées pour représenter les disciplines, de se limiter, pour les deux disciplines à $n = \min(n^1, n^2)$ variables aléatoires pour chaque discipline. Supposons que $n^2 > n^1$ (la deuxième discipline est donc l'aérodynamique), les solutions de la **MDA** aléatoire sont alors données par :

$$\begin{cases} \alpha_i^1 = \mu_i^1(x_*, \alpha^2) + \xi_i^1 \sigma_i^1(x_*, \alpha^2) & \forall i = 1, \dots, n^1 \\ \alpha_j^2 = \mu_j^2(x_*, \alpha^1) + \xi_j^2 \sigma_j^2(x_*, \alpha^1) & \forall j = 1, \dots, n^1 \\ \alpha_j^2 = \mu_j^2(x_*, \alpha^1) & \forall j = n^1 + 1, \dots, n^2 \end{cases} \quad (3.35)$$

Cette solution permet alors de limiter le nombre de sources d'incertitude de la **MDA** aléatoire et d'égaliser ce nombre pour chacune des disciplines. Notons que ce choix est logique puisque les premiers modes **POD** sont prépondérants dans la représentation des

solveurs disciplinaires. Néanmoins, la solution donnée par la moyenne des GP disciplinaires dépend toujours de l'ensemble des modes POD. Ainsi, la précision sur la solution donnée par la DPOD+I reste inchangée.

3.4.5.2 Modification de l'algorithme de résolution

Une idée alternative consiste à modifier l'algorithme de résolution en proposant un algorithme plus proche de la méthode de Jacobi détaillée dans l'Algorithme 4. Cette nouvelle méthode ne prend pas en compte l'estimation de l'erreur commise par la DPOD+I en utilisant les solutions de la MDA aléatoire. À la place, nous proposons d'utiliser uniquement l'information de la solution donnée par la moyenne des GP disciplinaires :

$$\begin{cases} \bar{\alpha}_i^1 &= \mu_i^1(\mathbf{x}_*, \bar{\alpha}^2) \quad \forall i = 1, \dots, n^1 \\ \bar{\alpha}_j^2 &= \mu_j^2(\mathbf{x}_*, \bar{\alpha}^1) \quad \forall j = 1, \dots, n^2 \end{cases} \quad (3.36)$$

et d'itérer sur cette solution. L'objectif est alors d'enrichir les modèles disciplinaires jusqu'à ce que la solution donnée par la moyenne des GP disciplinaires et les données de l'enrichissement soient suffisamment proches. L'algorithme se décompose donc en deux étapes : une étape d'enrichissement et une étape d'évaluation de la précision des modèles DPOD+I. Cette méthode sera nommée solveur DPOD+I Jacobi dans la suite.

Concernant l'enrichissement, comme les solutions de la MDA aléatoire ne sont pas calculées, aucune analyse de sensibilité sur les GP disciplinaires n'est effectuée et les deux disciplines sont donc enrichies à chaque itération.

Concernant l'évaluation de la précision des modèles DPOD+I, les modèles sont considérés comme suffisamment précis lorsque l'écart relatif entre la solution donnée par la moyenne des GP disciplinaires et les données de l'enrichissement est inférieur à ϵ_{MDA} . Ainsi, pour un paramètre fixé $\mathbf{x}_*$, le critère d'arrêt est donné par :

$$\begin{aligned} \frac{\|p^{-1}(\bar{\alpha}^1(\mathbf{x}_*)) - f^1(\mathbf{x}_*, p^{-2}(\bar{\alpha}^2(\mathbf{x}_*)))\|_2}{\|f^1(\mathbf{x}_*, p^{-2}(\bar{\alpha}^2(\mathbf{x}_*)))\|_2} &\leq \epsilon_{\text{MDA}} \\ \frac{\|p^{-2}(\bar{\alpha}^2(\mathbf{x}_*)) - f^2(\mathbf{x}_*, p^{-1}(\bar{\alpha}^1(\mathbf{x}_*)))\|_2}{\|f^2(\mathbf{x}_*, p^{-1}(\bar{\alpha}^1(\mathbf{x}_*)))\|_2} &\leq \epsilon_{\text{MDA}} \end{aligned} \quad (3.37)$$

où $\bar{\alpha}^1(\mathbf{x}_*)$, $\bar{\alpha}^2(\mathbf{x}_*)$ est calculée en utilisant les moyennes des GP. Nous allons à présent justifier que ce critère est validé si les GP disciplinaires sont de bonnes approximations des solveurs réduits. Notons que la DPOD+I cherche à estimer la solution de la MDA réduite solution de :

$$\begin{cases} \alpha^1 &= f_r^1(\mathbf{x}_*, \alpha^2) \\ \alpha^2 &= f_r^2(\mathbf{x}_*, \alpha^1) \end{cases} \quad (3.38)$$

où f_r^1 et f_r^2 sont les solveurs réduits. Ainsi, si les GP disciplinaires permettent de bien approcher les solveurs réduits :

$$\begin{aligned} f_r^1(\mathbf{x}_*, \bar{\alpha}^2(\mathbf{x}_*)) &\approx \mu^1(\mathbf{x}_*, \bar{\alpha}^2(\mathbf{x}_*)) \\ \Rightarrow f_r^1(\mathbf{x}_*, \bar{\alpha}^2(\mathbf{x}_*)) &\approx \bar{\alpha}^1(\mathbf{x}_*) && \text{car } \bar{\alpha}^1(\mathbf{x}_*) \text{ solution de l'Eq. (3.15)} \\ \Rightarrow p^{-1}(f_r^1(\mathbf{x}_*, \bar{\alpha}^2(\mathbf{x}_*))) &\approx p^{-1}(\bar{\alpha}^1(\mathbf{x}_*)) \\ \Rightarrow f(\mathbf{x}_*, p^{-2}(\bar{\alpha}^2(\mathbf{x}_*))) &\approx p^{-1}(\bar{\alpha}^1(\mathbf{x}_*)) && \text{par définition des solveurs réduits} \end{aligned} \quad (3.39)$$

de même,

$$\begin{aligned} f_r^2(\mathbf{x}_*, \bar{\alpha}^1(\mathbf{x}_*)) &\approx \mu^2(\mathbf{x}_*, \bar{\alpha}^1(\mathbf{x}_*)) \\ \Rightarrow f_r^2(\mathbf{x}_*, \bar{\alpha}^1(\mathbf{x}_*)) &\approx \bar{\alpha}^2(\mathbf{x}_*) && \text{car } \bar{\alpha}^2(\mathbf{x}_*) \text{ solution de l'Eq. (3.15)} \\ \Rightarrow p^{-2}(f_r^2(\mathbf{x}_*, \bar{\alpha}^1(\mathbf{x}_*))) &\approx p^{-2}(\bar{\alpha}^2(\mathbf{x}_*)) \\ \Rightarrow f^2(\mathbf{x}_*, p^{-1}(\bar{\alpha}^1(\mathbf{x}_*))) &\approx p^{-2}(\bar{\alpha}^2(\mathbf{x}_*)) && \text{par définition des solveurs réduits} \end{aligned} \quad (3.40)$$

car pour rappel, les solveurs disciplinaires réduits sont définis par $f_r^1(\mathbf{x}, \boldsymbol{\alpha}^2) = p^1(f^1(\mathbf{x}, p^{-2}(\boldsymbol{\alpha}^2)))$ et $f_r^2(\mathbf{x}, \boldsymbol{\alpha}^1) = p^2(f^2(\mathbf{x}, p^{-1}(\boldsymbol{\alpha}^1)))$.

Dans ce cas, le nouvel algorithme de résolution est noté Algorithme 16 et les étapes sont :

1. Calculer la solution donnée par la moyenne des GP disciplinaires $\bar{\boldsymbol{\alpha}}_*^1, \bar{\boldsymbol{\alpha}}_*^2$.
2. Effectuer les appels aux solveurs disciplinaires en utilisant la solution donnée par la moyenne des GP disciplinaires.
3. Enrichir les bases POD si cela est nécessaire.
4. Évaluer l'erreur commise définie dans l'Eq. (3.37).

Algorithm 16: Compute MDA solution using a DPOD+I Jacobi method.

```

input :  $\mathbf{x}_*, n_{MC}$ 
err  $\leftarrow 1 + \epsilon_{MDA}$ ;
while err <  $\epsilon_{MDA}$  do
     $\bar{\boldsymbol{\alpha}}_*^1, \bar{\boldsymbol{\alpha}}_*^2 \leftarrow$  solution of Eq. (3.15);
    Enrich the first disciplinary surrogate model;
     $\bar{\mathbf{y}}^1 \leftarrow f^1(\mathbf{x}^*, p^2(\bar{\boldsymbol{\alpha}}_*^2))$ ;
    If needed also enrich the first POD basis;
    if  $\frac{\|\bar{\mathbf{y}}^1 - p^1(p^{-1}(\bar{\mathbf{y}}^1))\|_2}{\|\bar{\mathbf{y}}^1\|_2} > \epsilon_{PE}$  then
         $\Phi^1 \leftarrow \left\{ \Phi^1, \frac{\bar{\mathbf{y}}^1 - p^1(p^{-1}(\bar{\mathbf{y}}^1))}{\|\bar{\mathbf{y}}^1 - p^1(p^{-1}(\bar{\mathbf{y}}^1))\|_2} \right\}$ ;
    end
    Enrich the second disciplinary surrogate model;
     $\bar{\mathbf{y}}^2 \leftarrow f^2(\mathbf{x}^*, p^1(\bar{\boldsymbol{\alpha}}_*^1))$ ;
    If needed also enrich the second POD basis;
    if  $\frac{\|\bar{\mathbf{y}}^2 - p^2(p^{-2}(\bar{\mathbf{y}}^2))\|_2}{\|\bar{\mathbf{y}}^2\|_2} > \epsilon_{PE}$  then
         $\Phi^2 \leftarrow \left\{ \Phi^2, \frac{\bar{\mathbf{y}}^2 - p^2(p^{-2}(\bar{\mathbf{y}}^2))}{\|\bar{\mathbf{y}}^2 - p^2(p^{-2}(\bar{\mathbf{y}}^2))\|_2} \right\}$ ;
    end
    err1  $\leftarrow \frac{\|p^{-1}(\bar{\boldsymbol{\alpha}}^1) - \bar{\mathbf{y}}^1\|_2}{\|\bar{\mathbf{y}}^1\|_2}$ ;
    err2  $\leftarrow \frac{\|p^{-2}(\bar{\boldsymbol{\alpha}}^2) - \bar{\mathbf{y}}^2\|_2}{\|\bar{\mathbf{y}}^2\|_2}$ ;
    err  $\leftarrow \max(\text{err}_1, \text{err}_2)$ ;
end
output:  $\bar{\mathbf{y}}^1, \bar{\mathbf{y}}^2$ 

```

Remarques :

- Une des améliorations par rapport à la méthode présentée dans l'Algorithme 13 est que la solution donnée par cette méthode est issue d'un appel aux solveurs disciplinaires permettant d'améliorer la précision des résultats.
- Comme seule la moyenne des GP disciplinaires est utilisée, d'autres méthodes d'interpolation pourraient être utilisées. Cependant, nous n'utiliserons pas d'autres méthodes puisque l'amélioration utilisant la PLS permet de gérer un nombre important d'entrées des approximations disciplinaires.
- L'un des désavantages de cette méthode est que chaque résolution nécessite au moins un appel à chaque solveur disciplinaire. Cela rend cette méthode moins intéressante que la précédente si le but est de minimiser le nombre d'appels aux solveurs disciplinaires.

TABEAU 3.13 – Erreurs relatives entre la solution de la **MDA** et son approximation par **POD** : e_{DoE}^1 et e_{DoE}^2 sur l'ensemble de test.

calcul	1	2	3	4	5	6	7	8	9	10	moyenne
e_{DoE}^1 (%)	0.2	0.3	0.4	0.3	0.2	0.3	0.3	0.5	0.3	0.16	0.3
e_{DoE}^2 (%)	0.7	1	0.9	0.7	0.9	0.8	1	1.1	1.3	0.7	0.9

TABEAU 3.14 – Erreurs relatives entre la solution de la **MDA** et la solution donnée par la moyenne des **GP** disciplinaires pour les deux disciplines sur chaque point de l'ensemble de test avant enrichissement.

calcul	1	2	3	4	5	6	7	8	9	10	moyenne
err_1 (%)	1.7	11	6.9	3.3	1.2	3.8	13	1.5	4.3	0.9	4.7
err_2 (%)	3.6	8.2	8.4	4.2	4.1	6.3	11	10	7.5	6.6	7

3.4.6 Résultats

Nous allons faire une analyse sur la résolution en utilisant la **DPOD+I** en utilisant 10 points de l'espace de design. Nous analyserons tout d'abord les résultats donnés par l'algorithme d'entraînement puis les résultats obtenus avec le nouvel algorithme de résolution (Algorithme 16). Enfin, nous présenterons les résultats obtenus avec l'Algorithme de résolution initial (Algorithme 13).

3.4.6.1 Analyse de l'entraînement

Dans un premier temps, nous allons analyser les performances de la base **POD** obtenue afin de valider sa capacité à approcher des solutions de **MDA**. Pour cela, nous nous basons sur les erreurs de projection e_{DoE}^1 et e_{DoE}^2 définies dans l'Eq. (3.30). Les erreurs obtenues sur l'ensemble de test sont affichées sur la Table 3.13. Nous pouvons observer que pour la discipline structure, l'erreur relative de projection est très peu importante soulignant la performance de l'algorithme d'entraînement, l'erreur relative pour la discipline aérodynamique est plus importante avec un maximum à 1.3% mais reste contenue. De plus, un enrichissement de la base **POD** peut être effectué lors de l'enrichissement afin de réduire cette erreur.

Dans un second temps, nous étudions la performance des **GP** disciplinaires avant enrichissement. Pour cela, nous évaluons l'erreur relative entre la solution de la **MDA** et la solution donnée par la moyenne des **GP** disciplinaires (solution de l'Eq. (3.15)). L'erreur relative, avant enrichissement est peu importante (< 5% pour les deux disciplines) pour 3 solutions de l'ensemble de test. Seules deux solutions ont une erreur relative maximale plus importante que 10%. Les **GP** disciplinaires sont donc représentatifs des solutions et permettent de valider l'approche d'entraînement dans ce cas complexe.

3.4.6.2 Analyse du solveur **DPOD+I** Jacobi

Notons que dans ce cas, seules les moyennes des **GP** disciplinaires sont utilisées. Dans un premier temps, une analyse du nombre d'appels aux solveurs disciplinaires pour la **MDA** exacte et le solveur **DPOD+I** Jacobi est proposée. Ces nombres sont affichés pour chacune des 10 résolutions dans la Table 3.15. Le nombre total d'appels pour la **MDA** exacte est de 90 et le nombre d'appels pour le solveur **DPOD+I** Jacobi est de 52. Néanmoins, pour le solveur **DPOD+I** Jacobi, 110 appels au solveur aérodynamique et 100 appels au solveur structure ont été nécessaires lors de la phase d'entraînement. Ainsi, la

TABLEAU 3.15 – Nombre d'appels aux solveurs disciplinaires pour la résolution de la MDA exacte et en utilisant le solveur DPOD+I Jacobi présenté dans l'Algorithme 16.

calcul	1	2	3	4	5	6	7	8	9	10	moyenne
MDA exacte	6	17	5	13	5	6	6	5	20	7	9
DPOD+I Jacobi	1	2	13	4	2	15	3	2	1	1	4.4

TABLEAU 3.16 – Erreurs relatives entre la solution de la MDA et la solution après résolution par l'Algorithme 16 pour les deux disciplines sur chaque point de l'ensemble de test.

calcul	1	2	3	4	5	6	7	8	9	10	moyenne
err_1 (%)	1.5	2.3	3.5	1.8	0.9	0.4	3.1	2.1	0.3	2.4	1.8
err_2 (%)	1.7	1.5	2.2	1.3	1.6	1.3	2.1	1.4	2.2	6.1	2.1

nouvelle méthode n'est pas efficace d'un point de vue numérique pour la résolution de 10 MDA mais deviendrait intéressante pour la résolution d'au moins 30 MDA si l'on ne considère pas l'amélioration du modèle initial au fur et à mesure des itérations.

Nous allons à présent analyser les erreurs faites sur la solution de la MDA. Nous nous intéresserons aux quantités :

$$err_1 = \frac{\|y^1(x_*) - \bar{y}^1(x_*)\|_2}{\|y^1(x_*)\|_2} \quad (3.41)$$

et

$$err_2 = \frac{\|y^2(x_*) - \bar{y}^2(x_*)\|_2}{\|y^2(x_*)\|_2} \quad (3.42)$$

où x_* est un paramètre de design et $\bar{y}^1(x_*)$, $\bar{y}^2(x_*)$ sont les solutions données par le solveur DPOD+I Jacobi. Ces quantités, pour les 10 variables de l'espace de test sont affichées dans la Table 3.16. Nous pouvons observer que dans 9 cas sur 10, l'erreur relative sur la solution de la MDA reste inférieure ou proche de 3% validant la méthode dans la plupart des cas. Néanmoins, dans le dernier cas, l'erreur est supérieure à 3%. Notons que nous cherchons à approcher, à l'aide de chaque solveur, la solution de la MDA avec une tolérance de 3%. Ainsi, une erreur de 6% peut être expliquée par le cas extrême où les deux erreurs commises par les deux algorithmes de résolution sont de 3%. L'entraînement permet de réduire fortement l'erreur sur les calcul n°2, 3 et 7. Cette erreur est peu améliorée par rapport à l'erreur obtenue après la phase d'entraînement pour le calcul n°10 : l'erreur sur la discipline structure augmente alors qu'un enrichissement de chaque discipline a été effectué. Ceci peut être expliqué par le fait qu'améliorer la précision des GP disciplinaires ne conduit pas forcément à une amélioration de la précision sur la MDA, d'autant plus lorsque le point d'enrichissement est éloigné de la solution de la MDA ce qui est le cas ici. Ceci peut aussi être dû à deux phénomènes. En premier lieu, même si nous avons observé que pour des GP disciplinaires précis le critère converge, la réciproque n'est pas vraie : une mauvaise estimation de la variance des GP disciplinaires peut faire converger le critère. Deuxièmement, la non linéarité de la MDA rend la propagation des erreurs impossible, une erreur inférieure à 3% sur les deux solveurs disciplinaires n'implique pas une erreur sur la solution de la MDA de 3%.

Finalement, nous avons vérifié que même avec ce cas complexe, il est possible d'approcher les solveurs disciplinaires en utilisant la DPOD+I. Les GP disciplinaires permettent, par la stratégie d'enrichissement proposée dans l'Algorithme 16, d'accélérer le calcul des MDA.

TABEAU 3.17 – Nombre d'appels au solveur structure pour la résolution de la MDA exacte et en utilisant la nouvelle méthode de résolution présentée dans l'Algorithme 13.

calcul	1	2	3	4	5	6	7	8	9	10	moyenne
MDA exacte	6	17	5	13	5	6	6	5	20	7	9
solveur DPOD+I	0	0	0	2	0	4 (+6)	1	2	1	0	1.6

TABEAU 3.18 – Nombre d'appels au solveur aérodynamique pour la résolution de la MDA exacte et en utilisant la nouvelle méthode de résolution présentée dans l'Algorithme 13.

calcul	1	2	3	4	5	6	7	8	9	10	moyenne
MDA exacte	6	17	5	13	5	6	6	5	20	7	9
solveur DPOD+I	1	3	3	2	2	6 (+6)	3	15	2	1	4.4

3.4.6.3 Analyse du solveur DPOD+I

Nous allons à présent analyser la méthode présentée dans l'Algorithme 13. Notons que, afin que les solutions de la MDA aléatoire convergent, les dimensions de ξ^1 et ξ^2 sont limitées à 7 (ceci correspond à la taille de la base structure). La tolérance pour la base ϵ_{PE} est fixé à 1%, ϵ_q est fixée, comme ϵ_{MDA} , à 3%.

Nous avons donc décidé, pour ce cas test, que si les solutions de la MDA aléatoire ne convergent pas, nous enrichissons les deux disciplines afin d'observer l'impact de cet enrichissement sur la MDA approchée. Cependant, lorsque les solutions de la MDA aléatoires ne convergent pas suite à 3 enrichissements successifs des deux disciplines, le processus est abandonné.

Le nombre d'appels nécessaires à la résolution des 10 MDA de l'espace de test est affiché dans la Table 3.17 pour la structure et dans la Table 3.18 pour l'aérodynamique. Le calcul n°6 est un cas où l'algorithme 13 n'a pas fonctionné. Notons que pour l'algorithme précédent, ce point était lui aussi difficile à approcher comme le démontre le nombre d'itérations effectuées en ce point (cf Table 3.15). La solution de la MDA exacte est alors nécessaire conduisant à 6 appels à chaque solveur disciplinaire. Nous analyserons le calcul n°6 dans la suite. Nous pouvons observer que cette méthode permet de fortement réduire le nombre d'appels à la discipline structure en passant d'un total de 44 appels (Table 3.15) à 16 appels (Table 3.17). Ceci souligne l'importance et l'efficacité de l'enrichissement adaptatif proposé permettant de privilégier l'enrichissement d'une certaine discipline. De plus, cela n'a pas d'impact sur le nombre d'appels à la discipline aérodynamique puisque 44 appels sont utilisés dans les deux cas. L'analyse de sensibilité permet donc de diminuer le nombre d'appels aux solveurs disciplinaires en enrichissant la discipline la plus influente.

Dans un second temps, nous étudions les erreurs commises lors de la résolution. Ces erreurs sont affichées dans la Table 3.19. Le calcul n°6 n'est pas répertorié puisque nous considérons qu'il a échoué. Nous pouvons observer que deux résolutions approchées conduisent à une erreur plus importante que l'erreur cible (3%). De plus, le calcul n° 10 est mieux approché par cette méthode que par la méthode utilisée dans la section précédente. Pourtant, un enrichissement de la discipline structure de moins a été effectué et un nombre équivalent pour la discipline aérodynamique a été réalisé. Cela souligne la dépendance de la méthode aux enrichissements précédents, via l'enrichissement du modèle initial. Finalement, nous pouvons tout de même observer que la moyenne des erreurs relatives sur les deux disciplines est inférieure à l'erreur cible de 3%. Enfin, le comportement de la solution donnée par la moyenne des GP disciplinaires n'est pas

TABLEAU 3.19 – Erreurs relatives entre la solution de la MDA et la solution après résolution par l'Algorithme 13 pour les deux disciplines sur chaque point de l'ensemble de test.

calcul	1	2	3	4	5	6	7	8	9	10	moyenne
err_1 (%)	0.7	0.6	1.2	1.6	0.5	NA	4.7	1.6	3.0	1.2	1.7
err_2 (%)	1.9	0.9	1.2	2.3	1.6	NA	3.4	1.9	4.0	3.4	2.3

forcement stable aux enrichissements. En effet, ajouter de la précision sur un solveur disciplinaire peut avoir un impact négatif sur la solution donnée par la moyenne des GP disciplinaires. Ceci est notamment dû au couplage non linéaire entre les GP disciplinaires.

Dans un dernier temps, nous avons tenté de résoudre la MDA au 6^{ème} point sans interrompre le solveur mais en nous limitant à 30 itérations. Nous observons que, pour ce point, au bout de 13 itérations de l'Algorithme 13, soit 9 enrichissements de l'aérodynamique et 4 enrichissements de la structure, l'erreur commise sur la solution de la MDA est de 3% pour les deux disciplines. Cependant, les solutions des MDA aléatoires ne convergent pas. Comme le critère d'arrêt de l'Algorithme 13 est défini à partir de ces solutions, la résolution continue. L'erreur relative de l'approximation sur la solution est inférieure à 2% au bout de la 28^{ème} itération, mais les solutions de la MDA aléatoire ne convergent pas à cette itération. Nous observons un phénomène assez problématique pour notre méthode : lorsque les GP disciplinaires sont précis, les solutions de la MDA aléatoire convergent moins souvent. En effet, lorsque les points d'enrichissement sont de plus en plus proches les uns des autres, l'optimisation des hyperparamètres des GP est rendue difficile. Nous nous apercevons alors que les hyperparamètres atteignent les bornes imposées lors de l'optimisation rendant l'estimation de la variance des GP imprécise.

3.4.7 Conclusion sur le second cas test aéroélastique

Dans cette section, nous avons testé les modèles DPOD+I sur un cas faisant intervenir un solveur disciplinaire non linéaire. La première difficulté a été l'augmentation importante du nombre de modes POD nécessaires à la représentation des sorties d'un solveur non linéaire. Les sources d'incertitude de la MDA aléatoire étant trop importantes, deux solutions ont été proposées. La première consiste à limiter la dimension de ξ^1 et ξ^2 . La seconde est une adaptation de l'algorithme de Jacobi à l'approximation par DPOD+I permettant d'éviter l'estimation de l'erreur par la DPOD+I via la MDA aléatoire. Cependant, sans enrichissement adaptatif, le nombre d'enrichissements de la discipline la moins influente est plus important. Cela souligne donc l'impact de cet enrichissement adaptatif. Finalement, bien que l'approximation par la solution donnée par la moyenne des GP disciplinaires puisse être précise grâce aux enrichissements, les solutions de la MDA aléatoire peuvent ne pas converger. Une perspective de la thèse serait d'adapter la construction des GP disciplinaires afin de contraindre l'existence de solutions de la MDA aléatoires. Des GP respectant des contraintes d'inégalité ont notamment été proposés par MAATOUK et BAY [2017]. De plus, l'une des limitations de notre méthode est que le nombre de modes POD est trop important. Ceci est principalement causé par la présence de phénomènes locaux dans le cadre transsonique qui sont difficiles à approcher avec une seule base POD. Une solution serait d'utiliser plusieurs bases POD (WASHABAUGH et collab. [2012]; DANIEL et collab. [2020]) afin de prendre en compte ces phénomènes locaux et donc de réduire le nombre de vecteurs dans les bases. Néanmoins, ceci limiterait la taille du DoE utilisé pour l'entraînement des interpolations sur les bases POD. Une autre so-

lution serait d'utiliser d'autres méthodes d'approximation de solveurs non-linéaires utilisant une interpolation comme l'Isomap adaptée à l'apprentissage de variétés non-linéaires (?; HALDER et collab. [2022]). Ceci permettrait alors de ne pas restreindre la taille du DoE pour l'entraînement des GP.

3.5 Optimisation multidisciplinaire

Cette section est dédiée à la résolution d'un problème d'optimisation grâce à la DPOD+I que nous avons testée dans les sections précédentes. L'idée est alors de coupler un algorithme d'optimisation avec la méthode DPOD+I. Pour rappel, l'objectif est de résoudre un problème d'optimisation multidisciplinaire :

$$\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) = f^{obj}(\mathbf{x}, y_*^1(\mathbf{x}), y_*^2(\mathbf{x})) \quad (3.43)$$

où $y_*^1(\mathbf{x}), y_*^2(\mathbf{x})$ sont la solution de la MDA :

$$\begin{cases} \mathbf{y}^1 = f^1(\mathbf{x}^g, \mathbf{x}^1, \mathbf{y}^2) \\ \mathbf{y}^2 = f^2(\mathbf{x}^g, \mathbf{x}^2, \mathbf{y}^1) \end{cases} \quad (3.44)$$

L'objectif sera donc d'exploiter l'approximation par DPOD+I de la MDA lors de l'optimisation. À cette fin, trois méthodes différentes vont être proposées :

- Un couplage simple où un algorithme d'optimisation va diriger la recherche des variables de design optimales et la résolution de la MDA sera effectuée par la méthode DPOD+I. Cette méthode sera similaire à l'approche MDF détaillée en Section 1.2.4.1 en remplaçant le solveur de la MDA et sera donc nommée couplage MDF dans la suite de la thèse.
- Un couplage où la solution donnée par la moyenne des GP disciplinaires sera utilisée pour obtenir les paramètres optimaux. Cette méthode sera proche de la méthode SBO présentée en Section 1.1.3.3.2 et sera nommée couplage SBO.
- Un couplage entre la méthode EGMDO et l'approximation par DPOD+I. La principale différence entre la méthode SBO et cette méthode sera l'exploitation de l'incertitude liée à l'interpolation par GP et sera nommée couplage EGMDO.

3.5.1 Couplage MDF

Dans un premier temps, un couplage simple entre l'algorithme d'optimisation en adoptant une stratégie MDF est décrit. Cette méthode sera extrêmement proche de la méthode que nous avons détaillée pour la résolution des MDA. En effet, nous avons détaillé cette méthodologie pour un DoE $\mathbb{X}_{DoE}$ qui avait été donné par un tirage par LHS dans notre application (cf Section 3.3.2 et 3.3.3). Cependant, la méthode peut être appliquée sur n'importe quel DoE et notamment si celui-ci est généré par un algorithme d'optimisation. Dans le cadre de la méthode MDF la résolution est couplée avec l'algorithme d'optimisation EGO.

Ce couplage de l'algorithme EGO avec l'approximation par DPOD+I nommé Algorithme 17 suit les étapes suivantes :

1. La première étape est d'initialiser les modèles DPOD+I grâce aux Algorithmes 11 et 12 dont nous avons testé la robustesse dans la Section 3.3.2.4.

2. Grâce aux modèles **DPOD+I** de chaque discipline, nous résolvons la **MDA** pour chaque point du **DoE** en utilisant l'Algorithme 13. Notons que l'une des différences clef entre cette méthode et l'Algorithme 10 présenté dans le Chapitre 2 est que le **DoE** d'initialisation de la méthode **EGO** n'est pas utilisé dans l'entraînement des modèles approchés. En effet, comme nous l'avons vu dans la Section 3.2.3, le plan d'expérience généré par cette méthode ne permet pas d'entraîner efficacement les **GP** disciplinaires. Les informations sont ensuite filtrées afin que le modèle soit de plus en plus performant lors de l'optimisation. Cette remarque est d'autant plus pertinente que les points trouvés par la méthode **EGO** sont susceptible d'être proches.
3. Par la suite, pour chaque appel à la fonction objectif, la résolution de la **MDA** sera faite grâce à l'Algorithme 13.

Algorithm 17: Couplage MDF

input : $\mathbf{x}_{\text{DoE}}, n_{\text{iter}}^{\text{max}}$
 Compute initial models for the **DPOD+I** using Algorithms 11 and 12;
for $\mathbf{x} \in \mathbf{x}_{\text{DoE}}$ **do**
 Solve the **MDA** for $\mathbf{x}$ using Algorithm 13 to obtain $p^{-1}(\bar{\alpha}_*^1)$ and $p^{-2}(\bar{\alpha}_*^2)$;
 $f(\mathbf{x}) \leftarrow f^{\text{obj}}(\mathbf{x}, p^{-1}(\bar{\alpha}_*^1), p^{-2}(\bar{\alpha}_*^2))$;
 Filter the information added to the initial model through Algorithm 14;
end
 $n_{\text{iter}} \leftarrow 1$ **while** $n_{\text{iter}} \leq n_{\text{iter}}^{\text{max}}$ **do**
 $\hat{\mu}, \hat{\sigma} \leftarrow \text{Gaussian Process}(\mathbf{x}_{\text{DoE}}, f_{\text{DoE}})$;
 $\mathbf{x}_{n_{\text{iter}}} \leftarrow \underset{\mathbf{x} \in \mathcal{X}}{\text{argmax}} \mathbb{E}(\mathcal{J}(\mathbf{x}))$;
 Add $\mathbf{x}_{n_{\text{iter}}}$ to $\mathbf{x}_{\text{DoE}}$;
 Solve the **MDA** for $\mathbf{x}_{n_{\text{iter}}}$ using Algorithm 13 to obtain $p^{-1}(\bar{\alpha}_*^1)$ and $p^{-2}(\bar{\alpha}_*^2)$;
 $f(\mathbf{x}_{n_{\text{iter}}}) \leftarrow f^{\text{obj}}(\mathbf{x}, p^{-1}(\bar{\alpha}_*^1), p^{-2}(\bar{\alpha}_*^2))$;
 Add $f(\mathbf{x}_{n_{\text{iter}}})$ to f_{DoE} ;
 Filter the information added to the initial model through Algorithm 14;
 $f_{\min} \leftarrow \min_{i=1, \dots, d} f_i$;
end
output: $f_{\min}$

Remarques :

- Dans une approche **MDF** utilisant par exemple une optimisation par gradient il est nécessaire de calculer la **MDA** avec une grande précision. Cependant, lors de nos tests, le couplage avec un algorithme par gradient n'a pas été concluant puisque notre méthode ne permet pas de résoudre la **MDA** avec une précision suffisante (notamment à cause de l'approximation par **POD**). Même si les premières itérations convergent vers le minimum local de la fonction objectif puisque la valeur du gradient est assez importante, le bruit sur son calcul par différences finies, induit par l'approximation, devient prédominant lorsque sa valeur diminue (soit lorsque l'on se rapproche du minimum local). Lors de nos tests, nous n'avons pas réussi à enrichir les modèles **DPOD+I** suffisamment précisément pour réussir à estimer le gradient sans avoir de problèmes de stabilité dus à la génération des solutions de la **MDA** aléatoire. Ceci représente une des limitations de notre méthode.
- De plus, la méthode, sans développement particulier n'est pas applicable en tant que telle. Aucun enrichissement ne doit être effectué lors de l'évaluation du gra-

dient par différences finies afin de pas modifier l'estimation de la solution aux points préalablement obtenus. Cependant, une extension possible de la méthode serait d'estimer le gradient analytique donné par le couplage des modèles **DPOD+I** (donc donné par la moyenne des **GP** disciplinaires). Cela permettrait de calculer, grâce à la stratégie d'enrichissement, la solution de la **MDA** en un point, d'estimer le gradient analytiquement en supposant que si la **MDA** a été résolue suffisamment précisément le calcul du gradient par les modèles **DPOD+I** serait précis.

3.5.2 Couplage **SBO**

Dans cette deuxième alternative, l'idée est d'utiliser la moyenne des **GP** disciplinaires afin de résoudre le problème d'optimisation. À cette fin, nous allons utiliser le point minimisant la fonction objectif approchée :

$$\mathbf{x}_{\text{SBO}} \in \arg \min_{\mathbf{x} \in \mathcal{X}} \tilde{f}(\mathbf{x}) = f^{\text{obj}}(\mathbf{x}, p^{-1}(\tilde{\alpha}_*^1), p^{-2}(\tilde{\alpha}_*^2)) \quad (3.45)$$

avec $p^{-1}(\tilde{\alpha}_*^1)$, $p^{-2}(\tilde{\alpha}_*^2)$ la solution donnée par la moyenne des **GP** disciplinaires. Les modèles **DPOD+I** seront alors enrichis en ce point à l'aide de l'Algorithme 13. Le problème (3.45) sera de nouveau résolu en prenant en compte l'information fournie par l'enrichissement. Ceci permet d'affiner l'approximation de $\tilde{f}$ au fur et à mesure des itérations autour des points donnés par la **SBO**.

Cependant, dans cette méthode, les données filtrées pour l'enrichissement du modèle initial doivent être changées. En effet, les bases **POD** doivent être enrichies au fur et à mesure des itérations afin de s'assurer que la fonction $\tilde{f}$ prenne en compte les modifications des bases **POD**. Néanmoins, pour l'enrichissement des **GP** disciplinaires, nous conservons la même stratégie afin d'éviter les problèmes de conditionnement de la matrice de corrélation.

Les étapes de la méthode données par l'Algorithme 18 sont :

1. La première étape est d'initialiser les modèles **DPOD+I** grâce aux Algorithmes 11 et 12.
2. À chaque itération, le calcul du paramètre optimal selon la solution donnée par la moyenne des **GP** disciplinaires (cf Eq. (3.45)) est effectué pour trouver le nouveau point d'enrichissement.
3. Les modèles disciplinaires sont alors enrichis au nouveau point pour obtenir l'approximation de la **MDA**.
4. Les bases **POD** sont enrichies grâce à l'information fournie par l'enrichissement local. Les **GP** disciplinaires sont enrichis en utilisant l'information du dernier appel au solveur disciplinaire comme cela est proposé dans l'Algorithme 14.

3.5.3 Couplage **EGMDO**

Dans cette troisième approche, nous allons présenter un couplage avec l'algorithme **EGMDO**. L'idée principale du couplage est d'utiliser les travaux proposés par **DUBREUIL et collab. [2018]** sur l'approximation de la fonction aléatoire. Par contre, l'apprentissage et l'enrichissement des **GP** disciplinaires seront effectués grâce à la méthode présentée dans ce chapitre (cf Section 3.2.3.2 et 3.2.5.1).

Les modèles **DPOD+I** sont créés à l'aide des Algorithmes 11 et 12. Nous obtenons alors une **MDA** approchée faisant intervenir un nombre réduit de variables de couplage comme

Algorithm 18: Couplage SBO

input : n_{iter}^{max}
 Compute initial models for the DPOD+I using Algorithms 11 and 12;
 $n_{iter} \leftarrow 1$;
while $n_{iter} \leq n_{iter}^{max}$ **do**
 $\mathbf{x}_{SBO} \leftarrow \arg\min_{\mathbf{x} \in \mathcal{X}} f^{obj}(\mathbf{x}, p^{-1}(\bar{\alpha}_*^1), p^{-2}(\bar{\alpha}_*^2))$;
 Solve the MDA for $\mathbf{x}_{SBO}$ using Algorithm 13 to obtain $p^{-1}(\bar{\alpha}_*^1)$ and $p^{-2}(\bar{\alpha}_*^2)$;
 $f_{SBO} \leftarrow f^{obj}(\mathbf{x}_{SBO}, p^{-1}(\bar{\alpha}_*^1), p^{-2}(\bar{\alpha}_*^2))$;
 Filter the information added to the initial model through Algorithm 14;
 Enrich the POD basis if needed;
end
output: f_{SBO}

cela était nécessaire pour la méthode EGMDO présentée dans la Section 1.2.5. Ainsi, le cadre est le même à la fin de l'initialisation des modèles DPOD+I que celui de la méthode EGMDO (cf Section 1.2.5.1) où chaque solveur est approché par des GP.

L'objectif est d'utiliser l'approximation de la fonction objectif détaillée dans la Section 1.2.5 et dans les Annexes A.1, A.2 et A.3 pour déterminer les points les plus intéressants du point de vue de l'optimisation. Notons que cette détermination ne prend pas en compte l'incertitude liée à la projection sur les bases POD. Nous supposons que cette erreur est faible et uniformément distribuée sur l'espace de design $\mathcal{X}$. Néanmoins, nous verrons que lors de la réduction des incertitudes, les bases POD seront enrichies si cela est jugé nécessaire.

Par la suite, une méthode de réduction des incertitudes adaptée aux modèles DPOD+I est proposée. Notons que les étapes de résolution de la MDO ne sont pas changées, seules les étapes d'entraînement et de réduction des incertitudes sont modifiées. Ces changements sont illustrés en couleur verte sur la Figure 3.35 illustrant la méthode EGMDO pour la résolution de la MDO en utilisant des modèles DPOD+I à la place de l'interpolation par GP.

3.5.3.1 Fonction objectif aléatoire

La fonction objectif aléatoire $\hat{f}^{obj}$ dans le cas de modèles DPOD+I est donnée par :

$$\hat{f}^{obj}(\mathbf{x}, \hat{\xi}^1, \hat{\xi}^2) = f^{obj}(\mathbf{x}, p^{-1}(\alpha_*^1(\mathbf{x}, \hat{\xi}^1, \hat{\xi}^2)), p^{-2}(\alpha_*^2(\mathbf{x}, \hat{\xi}^1, \hat{\xi}^2))) \quad (3.46)$$

où les variables aléatoires $\hat{\xi}^1$ et $\hat{\xi}^2$ caractérisent les trajectoires comme introduit en Section 3.2.4.1. Notons que, comparé à la fonction aléatoire sans projection des disciplines, la seule différence est que les données de la MDA aléatoire doivent être reconstruites grâce aux opérateurs p^{-1} et p^{-2} . Désormais, nous souhaitons construire une approximation de cette fonction objectif aléatoire. En effet, même si le calcul des solutions de la MDA aléatoire ne fait intervenir que des moyennes et variances de GP, son temps de calcul n'est pas négligeable. Comme nous allons baser notre stratégie d'enrichissement sur une amélioration obtenue à partir de cette fonction objectif aléatoire non gaussienne, une estimation de son espérance par MCS sera nécessaire. Ainsi, lors de l'optimisation de l'amélioration espérée, plus de 10^5 appels à l'amélioration seront nécessaires pour l'évaluation de son espérance en chaque variable de design. Optimiser directement cette fonction objectif aléatoire serait trop coûteux.

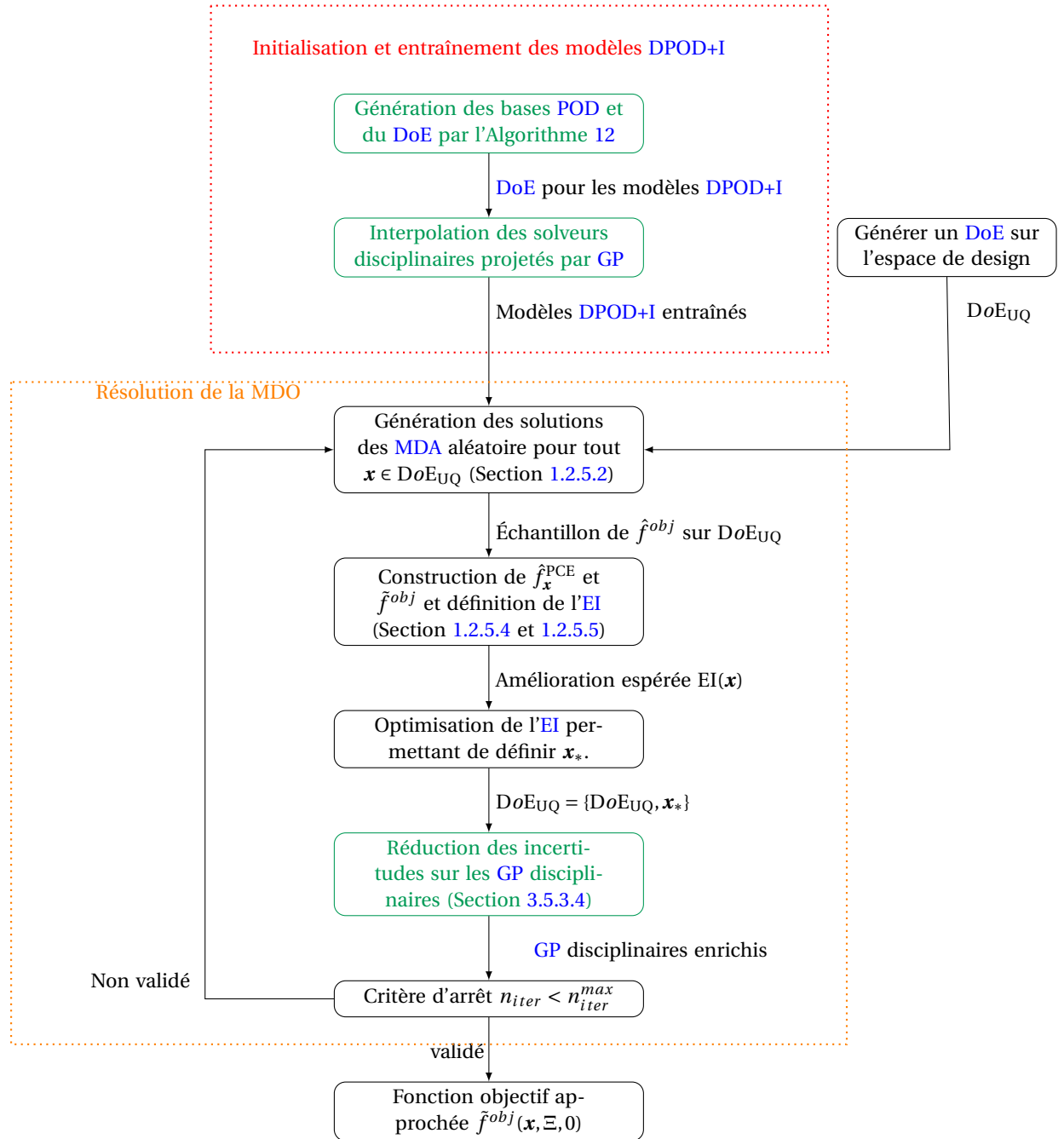


FIGURE 3.35 – Illustration de la résolution de la MDO en utilisant la méthode EGMDO avec des modèles DPOD+I. Les changements effectués par rapport à la méthode EGMDO sont affichés en vert.

Cependant, nous considérons qu'une fois qu'une solution de la MDA aléatoire est obtenue, le calcul de $f^{obj}(\mathbf{x}, p^{-1}(\boldsymbol{\alpha}_*^1(\mathbf{x}, \hat{\xi}^1, \hat{\xi}^2)), p^{-2}(\boldsymbol{\alpha}_*^2(\mathbf{x}, \hat{\xi}^1, \hat{\xi}^2)))$ est négligeable. Cette hypothèse peut être contraignante, d'autant plus que cette fonction fait appel à des vecteurs en grande dimension. Une extension possible de nos travaux serait alors d'approcher la fonction f^{obj} par interpolation par GP. Cela rajouterait une source d'incertitude sur la fonction aléatoire $\hat{f}^{obj}$ qui pourrait être prise en compte. Une stratégie d'enrichissement adaptée serait alors possible afin de limiter le nombre d'appels à f^{obj} .

3.5.3.2 Approximation de la fonction objectif aléatoire

L'approximation de la fonction objectif aléatoire est détaillée dans la Section 1.2.5.4. Nous rappelons dans la suite les principales étapes :

1. Définir les points DoE_{UQ} où les MDA aléatoires seront calculées.
2. En chaque point de DoE_{UQ}, calculer les solutions de la MDA aléatoire. Ceci permet d'obtenir, en plusieurs points un échantillon de la fonction aléatoire $\hat{f}^{obj}$. Ces échantillons de $\hat{f}^{obj}$ vont être utilisés pour générer une approximation $\tilde{f}^{obj}$.
3. Grâce à cet échantillon, une approximation $\hat{f}_x^{PCE}$ de $\hat{f}^{obj}$ est générée par PCE en tout point $\mathbf{x} \in \text{DoE}_{UQ}$. Le détail de l'approximation par PCE est présenté dans l'Annexe A.1.
4. L'approximation sur DoE_{UQ} est étendue à tout point de $\mathcal{X}$ en combinant une expansion KL (cf Annexe A.2) et une interpolation par GP (cf Annexe A.3) afin d'obtenir

$$\hat{f}^{obj}(\mathbf{x}, \xi^1, \xi^2) \approx \tilde{f}^{obj}(\mathbf{x}, \xi^1, \xi^2, \boldsymbol{\eta}) \quad (3.47)$$

où la variable $\boldsymbol{\eta}$ représente l'incertitude liée à l'approximation par GP en prenant en compte la variance des GP.

3.5.3.3 Amélioration espérée modifiée

Grâce à l'approximation $\tilde{f}^{obj}$, une amélioration I est définie dans l'Eq. (1.53) (cf Section 1.2.5.5) afin de déterminer le point le plus intéressant du point de vue de l'optimisation en prenant en compte l'incertitude des différentes interpolations. L'amélioration espérée EI est alors approchée par MCS (cf Eq. (1.55)). Le point $\mathbf{x}_*$ maximisant l'EI sera ajouté à DoE_{UQ} :

$$\mathbf{x}_* = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}} \text{EI}(\mathbf{x}) \quad (3.48)$$

Notons que cette amélioration espérée ne prend pas en compte l'erreur de projection par POD. Nous supposons que cette incertitude sera suffisamment réduite lors de la phase de réduction des incertitudes.

3.5.3.4 Réduction des incertitudes des modèles DPOD+I

Dans cette section, nous proposons une stratégie de réduction des incertitudes liées aux modèles DPOD+I à l'aide de la stratégie décrite dans les Algorithmes 6 et 13. Notons que la décision d'enrichir en un point $\mathbf{x}$ sera déterminée par la probabilité qu'un point $\mathbf{x} \in \text{DoE}_{UQ}$ minimise $\hat{f}_x^{PCE}$ comme cela est détaillé dans la Section 1.2.5.6. Cependant, l'enrichissement sera effectué selon la méthode développée dans l'Algorithme 13. De plus, le critère d'arrêt de l'enrichissement sera celui défini dans la Section 3.2.4.2 au lieu du

coefficient de variation de $\hat{f}_x^{obj}$ comme cela est proposé initialement dans EGMDO. L'objectif de cet algorithme est de réduire les incertitudes liées aux approximations par GP aux points de DoEUQ probablement optimaux. Cet algorithme permet donc d'enrichir les DoE disciplinaires sur lesquels sont conditionnés les GP disciplinaires ainsi que les bases POD si cela est nécessaire.

La réduction des incertitudes est synthétisée dans l'Algorithme 19 dont les étapes sont les suivantes :

1. Calcul de la PCE permettant d'obtenir l'approximation $\hat{f}_x^{PCE}(\xi^1, \xi^2)$, $\forall \mathbf{x} \in \text{DoEUQ}$.
2. Cette approximation est utilisée pour calculer $P_{min}(\mathbf{x})$, $\forall \mathbf{x} \in \text{DoEUQ}$ via l'Eq. (1.58) en utilisant une MCS.
3. Les points satisfaisant le critère d'enrichissement sont ajoutés dans $\mathbf{Z}_{enrich}$ dans l'ordre de probabilité P_{min} croissant.
4. Tant que les incertitudes de tous les points de $\mathbf{Z}_{enrich}$ ne sont pas réduites, les modèles DPOD+I sont enrichis. Si les modèles DPOD+I ne sont pas considérés comme suffisamment précis en un point de $\mathbf{Z}_{enrich}$, l'un des méta-modèles disciplinaires sera enrichi. Afin de déterminer la discipline la plus influente sur la quantité $\hat{q}$ définie dans l'Eq. (3.21) les indices de Sobol définis dans l'Eq. (3.20) sont évalués. Les entrées du solveur disciplinaire nécessaires à l'enrichissement sont alors données par la solution moyenne. La POD est enrichie si l'erreur relative de projection est supérieure à ϵ_{PE} . Notons que cette étape d'enrichissement est équivalente à une itération de l'Algorithme 13.
5. Comme les approximations disciplinaires ont changé, la PCE est calculée à nouveau permettant de redéfinir $P_{min}(\mathbf{x})$, $\forall \mathbf{x} \in \text{DoEUQ}$ et $\mathbf{Z}_{enrich}$. Ainsi, les étapes [3-6] sont répétées jusqu'à ce que le critère de convergence soit atteint pour tout point de $\mathbf{Z}_{enrich}$.

3.5.3.5 Résultat de l'optimisation

Les itérations de l'Algorithme EGMDO modifié sont détaillées dans l'Algorithme 20. Notons que la sortie de l'Algorithme 20 est similaire à celle donnée par l'Algorithme 7, soit l'approximation par expansion KL et interpolation par GP : $\tilde{f}^{obj}(\mathbf{x}, \Xi, 0)$. Comparé à l'Algorithme 7, les étapes modifiées sont l'entraînement des modèles disciplinaires ainsi que leur enrichissement. Le résultat de l'optimisation ne prend donc pas en compte l'erreur due à la projection par POD. La méthode EGMDO identifie le résultat du problème MDO :

$$\mathbf{x}_*^{\text{EGMDO}} = \arg \max_{\mathbf{x} \in \text{DoEUQ}} P_{min}(\mathbf{x}) \quad (3.49)$$

mais aussi une approximation de la fonction objectif aléatoire $\tilde{f}^{obj}(\mathbf{x}, \Xi, 0)$ obtenue en négligeant les incertitudes de l'interpolation des coefficients de la PCE réduits pouvant être utilisée à d'autres fins.

3.5.4 Application aérostructure

Nous allons à présent appliquer ces trois méthodes et les comparer sur un cas aérostructure. Le problème est le même que celui présenté pour la résolution des MDA avec 4 paramètres. Le paramétrage est lui aussi similaire. Ceci permet de réutiliser les mêmes données d'entraînement utilisées pour l'analyse de robustesse présentée dans la

Algorithm 19: Uncertainty Reduction of the DPOD+I

input : ϵ_q
 Compute the PCE approximation to obtain $\hat{f}_x^{\text{PCE}}(\Xi)$, $\forall \mathbf{x} \in \text{DoE}_{\text{UQ}}$;
 Compute $P_{\min}(\mathbf{x})$, $\forall \mathbf{x} \in \text{DoE}_{\text{UQ}}$ using Eq. (1.58);
while $\exists \mathbf{x} \in \text{DoE}_{\text{UQ}}$ such that $P_{\min}(\mathbf{x}) > \frac{1}{n_{\text{US}}}$ and $\text{CV}(\hat{f}_x^{\text{PCE}}(\Xi)) > \epsilon_{\text{CV}}$ **do**
 $\mathbf{Z}_{\text{enrich}} \leftarrow \left\{ \mathbf{x} \mid P_{\min}(\mathbf{x}) \geq \frac{1}{n_{\text{US}}} \right\}$;
 Sort $\mathbf{Z}_{\text{enrich}}$ in decreasing order with respect to $P_{\min}(\mathbf{x})$;
 for $\mathbf{x} \in \mathbf{Z}_{\text{enrich}}$ **do**
 Compute the quantiles q_l^1 and q_l^2 (cf Eq. (3.19));
 $q_l^1 \leftarrow \tau$ -quantile of $(q_k^1)_{k=1}^{n_{\text{MC}}}$;
 $q_l^2 \leftarrow \tau$ -quantile of $(q_k^2)_{k=1}^{n_{\text{MC}}}$;
 if $q_l^1 > \epsilon_q$ or $q_l^2 > \epsilon_q$ **then**
 $\tilde{S}^1 \leftarrow$ estimation of S^1 by PCE;
 $\tilde{S}^2 \leftarrow$ estimation of S^2 by PCE;
 if $\tilde{S}^1 > \tilde{S}^2$ **then**
 Enrich the first disciplinary surrogate model;
 $\bar{\mathbf{y}}^1 \leftarrow f^1(\mathbf{x}^*, p^{-2}(\bar{\boldsymbol{\alpha}}_*^2))$;
 If needed also enrich the first POD basis;
 if $\frac{\|\bar{\mathbf{y}}^1 - p^{-1}(p(\bar{\mathbf{y}}^1))\|_2}{\|\bar{\mathbf{y}}^1\|_2} > \epsilon_{\text{PE}}$ **then**
 $\Phi^1 \leftarrow \left\{ \Phi^1, \frac{\bar{\mathbf{y}}^1 - p^{-1}(p(\bar{\mathbf{y}}^1))}{\|\bar{\mathbf{y}}^1 - p^{-1}(p(\bar{\mathbf{y}}^1))\|_2} \right\}$;
 end
 else
 Enrich the second disciplinary surrogate model;
 $\bar{\mathbf{y}}^2 \leftarrow f^2(\mathbf{x}^*, p^{-1}(\bar{\boldsymbol{\alpha}}_*^1))$;
 If needed also enrich the second POD basis;
 if $\frac{\|\bar{\mathbf{y}}^2 - p^{-2}(p^2(\bar{\mathbf{y}}^2))\|_2}{\|\bar{\mathbf{y}}^2\|_2} > \epsilon_{\text{PE}}$ **then**
 $\Phi^2 \leftarrow \left\{ \Phi^2, \frac{\bar{\mathbf{y}}^2 - p^{-2}(p^2(\bar{\mathbf{y}}^2))}{\|\bar{\mathbf{y}}^2 - p^{-2}(p^2(\bar{\mathbf{y}}^2))\|_2} \right\}$;
 end
 end
 end
 Compute the PCE approximation to obtain $\hat{f}_x^{\text{PCE}}(\Xi)$, $\forall \mathbf{x} \in \text{DoE}_{\text{UQ}}$;
 Compute $P_{\min}(\mathbf{x})$, $\forall \mathbf{x} \in \text{DoE}_{\text{UQ}}$ using Eq. (1.58);
 Update $\mathbf{Z}_{\text{enrich}}$;
 end
end

Algorithm 20: Modified EGMDO

input : A DPOD+I approximation of the MDA obtained through Algorithm 12
 $\text{DoE}_{\text{UQ}} \leftarrow \text{DoE}$ over $\mathcal{X}$ obtained by LHS;
 $\Xi_{\text{MC}} \leftarrow \text{MC}$ sample from normal laws;
 Compute the random MDA solution for each element of DoE_{UQ} and Ξ_{MC} ;
 Compute the PCE approximation $\forall \mathbf{x} \in \text{DoE}_{\text{UQ}}$;
 $n_{\text{iter}} \leftarrow 0$ **while** $n_{\text{iter}} < n_{\text{iter}}^{\text{max}}$ **do**
 Compute the combination of KL expansion and GP interpolation to get $\tilde{f}^{\text{obj}}(\mathbf{x}, \Xi, \boldsymbol{\eta})$;
 Find $\mathbf{x}_* = \arg\max_{\mathbf{x} \in \mathcal{X}} \text{EI}(\mathbf{x})$;
 Add $\mathbf{x}_*$ to DoE_{UQ} ;
 Enrich the GP using Algorithm 19;
end
output: $\tilde{f}^{\text{obj}}(\mathbf{x}, \Xi, 0)$

Section 3.3.2.4. De plus, les paramètres de la DPOD+I sont les mêmes que ceux présentés dans la Section 3.3.1.5.

Le problème d'optimisation sera un problème de type problème inverse. L'objectif est de retrouver les paramètres définissant des déplacements et forces aérodynamiques donnés que nous noterons $\mathbf{y}_{\text{ref}}^1$ et $\mathbf{y}_{\text{ref}}^2$. Ce problème a été notamment résolu dans la littérature afin de retrouver le design de l'aile *Modèle de Recherche Commun, ou Common Research Model (CRM)* (KENWAY et collab. [2014]) non déformée à partir des informations fournies par l'aile déformée. Le problème devient alors :

$$\min_{\mathbf{x} \in \mathcal{X}} \frac{\|\mathbf{y}_*^1(\mathbf{x}) - \mathbf{y}_{\text{ref}}^1\|_2}{\|\mathbf{y}_{\text{ref}}^1\|_2} + \frac{\|\mathbf{y}_*^2(\mathbf{x}) - \mathbf{y}_{\text{ref}}^2\|_2}{\|\mathbf{y}_{\text{ref}}^2\|_2} \quad (3.50)$$

où $\mathbf{y}_*^1(\mathbf{x})$ et $\mathbf{y}_*^2(\mathbf{x})$ sont les solutions de la MDA pour la variable de design $\mathbf{x}$.

Afin de tester la méthode, nous proposons de choisir comme $\mathbf{y}_{\text{ref}}^1$ et $\mathbf{y}_{\text{ref}}^2$ solution de la MDA en un paramètre de design particulier $\mathbf{x}_{\text{ref}}$ afin de comparer le paramètre donné par la résolution de la MDO et $\mathbf{x}_{\text{ref}}$. Ainsi,

$$\begin{aligned} \mathbf{y}_{\text{ref}}^1 &= \mathbf{y}_*^1(\mathbf{x}_{\text{ref}}) \\ \mathbf{y}_{\text{ref}}^2 &= \mathbf{y}_*^2(\mathbf{x}_{\text{ref}}) \end{aligned} \quad (3.51)$$

Le paramètre $\mathbf{x}_{\text{ref}}$ est choisi comme étant un point où la MDA est la plus difficile à résoudre au sens du nombre d'itérations de l'algorithme de Gauss-Seidel. Les épaisseurs de la structure sont minimales et l'angle d'attaque ainsi que la vitesse de l'aile sont maximisés. Ce point conduit au maximum du déplacement en bout d'aile et maximise le nombre d'itérations lors de la résolution avec l'algorithme de Gauss-Seidel. Le nombre d'itérations de l'algorithme nécessaire à la résolution de la MDA en ce paramètre est de 38 ce qui implique 38 appels aux solveurs disciplinaires.

Les modèles DPOD+I sont initialisés via l'Algorithme 12 avec $r = 10$ et sont issus de l'analyse de robustesse proposée dans la Section 3.3.2.4. Notons que sur ce problème, les modèles DPOD+I initiaux sont extrêmement précis sur l'ensemble du domaine paramétrique. En effet, la résolution de chaque MDA nécessite moins de 1 appel au solveur structure et quasiment aucun appel au solveur aérodynamique. Ceci implique que peu d'appels aux solveurs disciplinaires seront effectués lors de la résolution de la MDO quelque soit le couplage avec les modèles DPOD+I considéré.

TABLEAU 3.20 – Résolution MDF. Nombre d'appels aux solveurs disciplinaires lors de la résolution de la MDO sans approximation.

n° calcul	1	2	3	4	5	6	7	8	9	10	moyenne
n^s, n^a	3002	3609	2401	2996	1892	3886	3290	3459	3400	2422	3036

TABLEAU 3.21 – Résolution MDF. Espérance et coefficient de variation sur les 10 optimisations des paramètres de design et de la fonction objectif comparés aux valeurs de référence.

	x_1	x_2	x_3	x_4	$f(x)$
Référence	1	1	0	0	0
$\mathbb{E}$	1.	0.859	4×10^{-4}	3×10^{-4}	0.04
CV (%)	0.004	20	280	234	119

3.5.4.1 Résultats sans approximation

Nous allons analyser la résolution de la MDO sans approximation par DP0D+I sur ce cas test. Afin de prendre en compte des variables de couplage de grande dimension, seule la formulation MDF peut être utilisée. Ainsi, la MDA est résolue à l'aide d'un solveur itératif et l'optimisation est effectuée via un algorithme d'optimisation bayésien : EGO. La taille du DoE initial de l'algorithme EGO est de 20 et nous effectuons 200 itérations. Afin d'évaluer l'impact du DoE initial sur le résultat de l'optimisation, 10 optimisations seront effectuées avec 10 DoE initiaux différents.

Les nombres n^s et n^a d'appels aux solveurs disciplinaires effectués lors de la résolution sont affichés sur la Table 3.20. Environ 3000 appels à chaque solveur sont nécessaires pour la résolution de la MDO et le temps de calcul nécessaire est d'environ 16 heures par optimisation. Notons que le budget de l'optimisation est alourdi par le fait que l'optimum x_{ref} se situe dans une zone de l'espace de design nécessitant un nombre important d'itérations (38).

Par la suite, les résultats de l'optimisation sont analysés. Les moyennes et coefficients des paramètres optimaux x_* obtenus lors de l'optimisation ainsi que la fonction objectif sont affichés sur la Table 3.21. Le point optimal est déterminé avec une précision d'environ 10^{-5} pour environ la moitié des optimisations. Néanmoins, sur la moitié des optimisations, le minimum n'est pas déterminé avec précision, principalement sur la deuxième variable de design. Nous avons observé que les variations de cette variable de design, correspondante à la vitesse de l'avion, ont moins d'impact sur la fonction objectif.

3.5.4.2 Résultats couplage MDF

Nous allons analyser le couplage MDF sur ce cas test. La taille du DoE initial de l'algorithme EGO est de 20 et nous effectuons 200 itérations.

Premièrement, le nombre d'appels aux solveurs disciplinaires nécessaire à l'entraînement et à la résolution de la MDO grâce à l'Algorithme 17 est analysé. Nous noterons n_{train}^s et n_{train}^a le nombre d'appels aux solveurs structure et aérodynamique effectués lors de l'entraînement des modèles DP0D+I. De même, nous noterons n_{MDO}^s et n_{MDO}^a le nombre d'appels aux solveurs disciplinaires effectués lors de la résolution de la MDO. Les nombres d'appels nécessaires à la phase d'entraînement et de résolution sont affichés dans la Table 3.22. Le nombre moyen d'appels au solveur structure pour l'entraînement est de 49 et de 59 pour l'aérodynamique. Le nombre moyen d'appels au solveur structure et aérodynamique pour la résolution de la MDO est de 19 pour la structure et de 6 pour

TABLEAU 3.22 – Couplage MDF. Nombre d'appels aux solveurs disciplinaires lors de la résolution de la MDO avec le couplage MDF pour 10 optimisations.

n° calcul	1	2	3	4	5	6	7	8	9	10	moyenne
n_{train}^s	50	50	60	40	50	40	50	60	40	50	49
n_{train}^a	60	60	70	50	60	50	60	70	50	60	59
n_{MDO}^s	20	20	17	21	19	34	13	10	24	12	19
n_{MDO}^a	25	1	1	4	4	11	2	4	7	2	6

TABLEAU 3.23 – Couplage MDF. Espérance et coefficient de variation sur les 10 optimisations des paramètres de design et de la fonction objectif comparés aux valeurs de référence.

	x_1	x_2	x_3	x_4	$f(x)$
Référence	1	1	0	0	0
$\mathbb{E}$	0.998	0.907	0.002	0.001	0.07
CV (%)	0.2	14	140	316	59

l'aérodynamique. Ainsi, le nombre moyen total d'appels à la discipline structure est de 78 et de 65 pour l'aérodynamique. Cela représente, au point x_{ref} , le budget nécessaire à la résolution de deux MDA. Le budget nécessaire à la résolution de la MDO est donc réduit de plus de 97% comparé à la MDF sans approximation.

Par la suite, les résultats de l'optimisation sont analysés. Les moyennes et coefficients des paramètres optimaux x_* obtenus lors de l'optimisation ainsi que la fonction objectif approchée $f^{obj}(x_*, p^{-1}(\bar{\alpha}_*^1), p^{-2}(\bar{\alpha}_*^2))$ sont affichés sur la Table 3.23. Ceux-ci sont comparés à x_{ref} et $f(x_{ref}) = 0$. Les valeurs x_1 , x_3 et x_4 sont particulièrement bien retrouvées par l'algorithme d'optimisation puisque les valeurs moyennes sont quasiment les valeurs cibles. Les CV des dernières composantes sont plus importantes puisque la moyenne est quasiment nulle. Néanmoins, la deuxième variable de design correspondant à la vitesse de l'aile est plus difficilement retrouvée. La fonction objectif approchée est d'environ 0.07 alors que la cible est 0. Notons que la fonction objectif fait intervenir la somme des erreurs des deux disciplines et correspond donc à une erreur sur chaque discipline divisée par deux.

3.5.4.3 Résultats couplage SBO

La stratégie effectuant un couplage SBO est testée sur ce problème. Le nombre d'itérations de l'algorithme est fixé à 10. Le minimum de la fonction objectif est souvent déterminé dès la première itération de l'Algorithme 18.

Les nombres d'appels nécessaires à la phase d'entraînement et de résolution sont affichés dans la Table 3.24. Le nombre moyen d'appels au solveur structure et aérodynamique pour la résolution de la MDO est de 2.4 pour la structure et de 2.6 pour l'aérodynamique. Pour le cas particulier du calcul n°4, plus de 10 appels aux solveurs disciplinaires ont été effectués lors de la résolution de la MDO. Ceci est dû au fait que plusieurs fois lors de la résolution de la MDO avec ce DoE, les solutions de la MDA aléatoire n'ont pas convergé. Ceci a impliqué un plus grand nombre d'appels aux solveurs disciplinaires puisque les deux disciplines sont enrichies si l'analyse de sensibilité échoue. Comparé au couplage MDF nous pouvons observer une forte diminution des appels nécessaires à la résolution de la MDO.

Par la suite, les résultats de l'optimisation sont analysés. Les moyennes et coefficients des paramètres optimaux x_* obtenus lors de l'optimisation ainsi que la fonction objectif

TABLEAU 3.24 – Couplage SBO. Nombre d'appels aux solveurs disciplinaires lors de la résolution de la MDO avec le couplage SBO pour 10 optimisations.

n° calcul	1	2	3	4	5	6	7	8	9	10	moyenne
n_{train}^s	50	50	60	40	50	40	50	60	40	50	49
n_{train}^a	60	60	70	50	60	50	60	70	50	60	59
n_{MDO}^s	1	1	1	14	1	2	1	1	1	1	2
n_{MDO}^a	2	2	1	12	1	2	2	2	1	1	3

TABLEAU 3.25 – Couplage SBO. Espérance et coefficient de variation sur les 10 optimisations des paramètres de design et de la fonction objectif comparés aux valeurs de référence.

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$f(\mathbf{x})$
Référence	1	1	0	0	0
$\mathbb{E}$	0.998	0.990	0.004	2×10^{-20}	0.038
CV (%)	0.3	2	83	136	46

approchée $f^{obj}(\mathbf{x}_*, p^{-1}(\bar{\alpha}_*^1), p^{-2}(\bar{\alpha}_*^2))$ sont affichés sur la Table 3.25. Les variables de design moyennes sont très proches de celles de référence. Le niveau de précision est équivalent à celui obtenu sur le couplage MDF sur les variables $\mathbf{x}_1$, $\mathbf{x}_3$ et $\mathbf{x}_4$. Cependant, avec cette méthode, nous pouvons observer que la 2^{ème} variable est elle aussi estimée avec précision. La moyenne de la fonction objectif obtenue est légèrement élevée même si elle ne dépasse pas les 4%. Ceci est dû au fait que, en $\mathbf{x}_{ref}$, l'approximation par DPOD+I est complexe et entraîne une erreur assez importante. Notons que les erreurs sur les paramètres optimaux sont très peu importantes puisque les variations de la fonction objectif dominent les erreurs dues à l'approximation par DPOD+I.

3.5.4.4 Résultats couplage EGMDO

Nous allons à présent tester la méthode EGMDO sur ce problème. Nous fixons le nombre d'itérations à 10, de la même façon, l'optimum est souvent déterminé dès la première itération de l'algorithme.

Les nombres d'appels nécessaires à la phase d'entraînement et de résolution sont affichés dans la Table 3.26. Le nombre moyen d'appels aux solveurs structure et aérodynamique pour la résolution de la MDO est de 2 pour les deux disciplines. On peut donc observer que quasiment aucun enrichissement n'est effectué lors de la résolution de la MDO soulignant l'efficacité de notre méthode. Nous pouvons de plus observer que même lors de la résolution de la MDO, les entraînements peu efficaces comme c'est le cas avec le calcul n° 9 sont compensés, lors de la phase de résolution par un nombre d'enrichissements plus important. Notons que dans de nombreux cas, le nombre d'enrichissements nécessaires est de 1 pour chaque discipline. Ceci correspond souvent au nombre de points d'enrichissements nécessaires pour la résolution de la MDA au point $\mathbf{x}_{ref}$ qui est obtenu dès la première itération de EGMDO.

Les moyennes et coefficients des paramètres optimaux $\mathbf{x}_*$ obtenus lors de l'optimisation ainsi que la fonction objectif approchée $f^{obj}(\mathbf{x}_*, p^{-1}(\bar{\alpha}_*^1), p^{-2}(\bar{\alpha}_*^2))$ sont affichés sur la Table 3.27. Nous n'observons pas de différences entre le couplage SBO et le couplage EGMDO concernant la précision des résultats. Ce couplage est lui-aussi extrêmement performant. En effet, les modèles DPOD+I, à l'issue de la phase d'entraînement, sont extrêmement précis et les incertitudes liées à l'interpolation sont trop faibles pour avoir d'impact sur l'EI limitant les phases d'exploration de l'algorithme EGMDO.

TABEAU 3.26 – Couplage **EGMDO**. Nombre d'appels aux solveurs disciplinaires lors de la résolution de la **MDO** avec le couplage **EGMDO** pour 10 optimisations.

n° calcul	1	2	3	4	5	6	7	8	9	10	moyenne
n_{train}^s	50	50	60	40	50	40	50	60	40	50	49
n_{train}^a	60	60	70	50	60	50	60	70	50	60	59
n_{MDO}^s	4	1	1	3	1	2	1	1	5	1	2
n_{MDO}^a	4	1	1	1	1	2	3	2	4	1	2

TABEAU 3.27 – Couplage **EGMDO**. Espérance et coefficient de variation sur les 10 optimisations des paramètres de design et de la fonction objectif comparés aux valeurs de référence.

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$f(\mathbf{x})$
Référence	1	1	0	0	0
$\mathbb{E}$	0.999	0.987	0.006	3×10^{-7}	0.044
CV (%)	0.3	2	90	316	44

3.6 Bilan du Chapitre

Dans ce chapitre, nous avons détaillé une nouvelle méthode d'approximation de **MDA** avec des variables de couplage de grande dimension. Cette méthode d'approximation est basée sur des modèles approchés de chaque discipline qui sont par la suite couplés afin de générer une approximation de la **MDA**. Comme les disciplines ont un nombre très important d'entrées et de sorties (les variables de couplage) la première étape est de projeter les variables de couplage sur des bases **POD** afin de diminuer drastiquement la dimension de ces variables. Par la suite ces disciplines réduites sont remplacées par des méta-modèles. Dans le cadre de notre étude, nous avons choisi d'interpoler chaque coefficient de la **POD** par des **GP** conditionnés. Ces modèles disciplinaires sont nommés **DPOD+I** et sont par la suite couplés afin de générer une approximation de la **MDA**. Deux informations sont alors utilisées : la solution donnée par la moyenne des **GP** pour obtenir une approximation de la **MDA** et la **MDA** aléatoire afin de modéliser les incertitudes liées aux interpolations par **GP**. Notre première contribution concerne l'entraînement des modèles **DPOD+I**. En effet, nous avons observé que le **DoE** obtenu sur un échantillon d'itérations de résolution de **MDA** par des algorithmes de type Gauss-Seidel ou Jacobi génère des données mal réparties sur les espaces des variables de couplage. Ce phénomène a été mis en évidence sur la Figure 3.7. Pour y pallier, nous avons proposé une nouvelle stratégie d'enrichissement où un découplage entre les variables de design et les variables de couplage est mis en oeuvre. Ce découplage permet de répartir les données plus uniformément comme cela est illustré sur la Figure 3.9 limitant l'obtention des **GP** mal-conditionnés. Cette stratégie d'entraînement dépendant d'un paramètre r correspondant aux nombres d'appels aux solveurs pouvant être effectués en parallèle a été analysée en Section 3.3.2.4 montrant que l'impact de ce paramètre est lissé au cours de la phase d'enrichissement. Cette stratégie a été par la suite testée sur un cas plus complexe avec une discipline non linéaire démontrant que même si la discipline est beaucoup plus difficile à approcher par **POD**, ce qui entraîne un budget d'entraînement plus important, elle est efficace. La performance des bases **POD** obtenues a été testée sur des solutions de **MDA** et même si aucune résolution de **MDA** n'a été effectuée lors de l'entraînement, les bases **POD** restent performantes sur ces solutions.

La première application de ce méta-modèle de la **MDA** a été de développer un solveur **MDA** qui cherche la solution en affinant localement la solution donnée par la moyenne

des GP disciplinaires grâce à des enrichissements successifs. La précision du modèle approché est estimée à partir de la dispersion des solutions de la MDA aléatoire. Si les solutions sont considérées comme trop dispersées, un enrichissement d'une des disciplines est proposé afin de réduire les incertitudes liées à l'interpolation. Cet enrichissement adaptatif est basé sur une estimation des indices de Sobol estimant la part de variance sur la dispersion des solutions due à l'incertitude liée à une discipline. Afin d'effectuer une analyse de sensibilité liée à l'impact des incertitudes des GP sur la dispersion des solutions de la MDA aléatoire, une modélisation initialement proposée par DUBREUIL et collab. [2018] a été mise en oeuvre à travers l'utilisation de GP parfaitement corrélés. Cette modélisation permet de représenter les trajectoires de chaque GP par une variable aléatoire (ξ) et d'étudier l'impact de cette variable sur la dispersion de la MDA aléatoire à partir d'une estimation des indices de Sobol par PCE (SUDRET [2008]). L'enrichissement est ensuite effectué à partir de la solution donnée par la moyenne des GP disciplinaires et la base POD est enrichie si cela est jugé nécessaire. Les modèles globaux sont enrichis au fur et à mesure des enrichissements locaux afin de rendre les résolutions des MDA de plus en plus efficaces. Cependant, afin d'éviter l'écueil des GP mal conditionnés, nous proposons de filtrer les données ajoutées aux modèles globaux. Cette stratégie d'enrichissement a été testée sur un premier cas aérostructure et a démontré la capacité du solveur DPOD+I. Comparé à un solveur de MDA classique, ce solveur DPOD+I permet de diminuer d'un facteur 5 le nombre d'appels aux solveurs disciplinaires dans le cas 4D et d'un facteur 4 pour le cas 8D. La nouvelle méthode DPOD+I est ensuite comparée à l'approche POD+I classique. Avec un nombre comparable d'appels aux solveurs disciplinaires, la DPOD+I surpasse la POD+I en terme de précision. Lorsque la même stratégie d'enrichissement est appliquée, la DPOD+I surpasse la POD+I enrichie en termes d'appels aux solveurs disciplinaires. En effet, cette nouvelle stratégie permet l'enrichissement de chaque DPOD+I impliquant indépendamment une réduction du coût d'enrichissement. Ce solveur DPOD+I a par la suite été testé sur un cas faisant intervenir un solveur non linéaire. Sans développement supplémentaire, nous avons observé que les solutions de la MDA aléatoire étaient trop dispersées et que le critère d'arrêt de l'algorithme d'enrichissement local (Algorithme 12) n'était jamais atteint. Ceci est principalement dû au fait que le nombre de sources d'incertitude dans la MDA aléatoire est égal au nombre de vecteurs dans la base POD. Nous avons donc proposé de limiter le nombre de sources d'incertitude de la MDA aléatoire. Néanmoins, nous continuons à observer la non convergence des MDA aléatoires pour une résolution de la MDA. Ainsi, un autre solveur basé sur les modèle DPOD+I inspiré de l'algorithme de Jacobi a été proposé. Ce solveur utilise comme critère d'arrêt la différence entre la prédiction de la solution donnée par les GP disciplinaires et les données de l'enrichissement. Comme aucune analyse de sensibilité n'est effectuée, les deux disciplines sont enrichies simultanément augmentant le nombre d'appels à la discipline la moins incertaine (soit la structure dans cet exemple) comparé au solveur DPOD+I.

Finalement, les modèles DPOD+I ont été utilisés dans un contexte d'optimisation multidisciplinaire. Plusieurs couplages ont alors été proposés. Un couplage MDF où un algorithme d'optimisation pilote la recherche des paramètres et la résolution de la MDA est effectué grâce au solveur DPOD+I. Cette solution permet de réduire fortement le nombre d'appels au solveur disciplinaire sur un exemple de problème inverse. Néanmoins, une des variables de design n'était pas obtenue avec suffisamment de précision. Un autre couplage basé sur une stratégie SBO utilisant la solution donnée par la moyenne des GP disciplinaires a été proposé. Cette méthode permet de réduire encore le nombre d'appels aux solveurs disciplinaires (équivalent au budget nécessaire à la résolution de deux MDA

à l'optimum). Par la suite, un couplage avec l'algorithme **EGMDO** a été proposé. En effet, les modèles **DPOD+I** couplés sont équivalents aux interpolations des disciplines (dans un cas avec un faible nombre de variables de couplage) par **GP** proposés par **DUBREUIL et collab.** [2020] si l'erreur de projection sur les bases **POD** est négligée. Ainsi, l'**EI** obtenue à partir de la **MDA** aléatoire permet de définir le point d'enrichissement de l'algorithme. Les entraînements et réduction des incertitudes ont été adaptés à la nouvelle méthode d'approximation et intégrés à l'algorithme **EGMDO**. Les couplages **SBO** et **EGMDO** sont similaires en terme de performance. Ces performances similaires sur la résolution de la **MDO** sont principalement dues à la précision des modèles **DPOD+I** à la fin de l'algorithme d'entraînement.

3.7 Références

- ALEXANDROV, N. et R. LEWIS. 2004, «Reconfigurability in MDO problem synthesis, part 1», dans *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, doi: 10.2514/6.2004-4307. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2004-4307>. 122
- BALLING, R. J. et J. SOBIESZCZANSKI-SOBIESKI. 1996, «Optimization of coupled systems-a critical overview of approaches», *AIAA journal*, vol. 34, n° 1, p. 6–17. 122
- BONCORAGLIO, G., C. FARHAT et C. BOU-MOSLEH. 2021, «Model reduction framework with a new take on active subspaces for optimization problems with linearized fluid-structure interaction constraints», *International Journal for Numerical Methods in Engineering*, vol. 122, n° 19, p. 5450–5481. 146
- BOUHLEL, M. A., N. BARTOLI, A. OTSMANE et J. MORLIER. 2016, «Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction», *Structural and Multidisciplinary Optimization*, vol. 53, n° 5, p. 935–952. 167
- BOUHLEL, M. A., J. T. HWANG, N. BARTOLI, R. LAFAGE, J. MORLIER et J. R. R. A. MARTINS. 2019, «A python surrogate modeling framework with derivatives», *Advances in Engineering Software*, doi: <https://doi.org/10.1016/j.advengsoft.2019.03.005>, p. 102 662, ISSN 0965-9978. 151
- BROOKS, T. R., G. K. KENWAY et J. R. R. A. MARTINS. 2017, *Undelected Common Research Model (uCRM) : An Aerostructural Model for the Study of High Aspect Ratio Transport Aircraft Wings*, doi: 10.2514/6.2017-4456. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2017-4456>. 165
- COELHO, R. F., P. BREITKOPF et C. KNOPF-LENOIR. 2008, «Model reduction for multidisciplinary optimization - Application to a 2D wing», *Structural and Multidisciplinary Optimization*, vol. 37, n° 1, doi: 10.1007/s00158-007-0212-5, p. 29–48, ISSN 1615-1488. URL <https://doi.org/10.1007/s00158-007-0212-5>. 130, 143, 146
- COELHO, R. F., P. BREITKOPF, C. KNOPF-LENOIR et P. VILLON. 2009, «Bi-level model reduction for coupled problems», *Structural and Multidisciplinary Optimization*, vol. 39, n° 4, doi: 10.1007/s00158-008-0335-3, p. 401–418. 146
- CRAMER, E., J. DENNIS, P. FRANK, R. LEWIS et G. R. SHUBIN. 1994, «Problem formulation for multidisciplinary optimization», *SIAM Journal on Optimization*, vol. 4, p. 754–776. 122

- DANIEL, T., F. CASENAVE, N. AKKARI et D. RYCKELYNCK. 2020, «Model order reduction assisted by deep neural networks (rom-net)», *Advanced Modeling and Simulation in Engineering Sciences*, vol. 7, n° 1, doi: 10.1186/s40323-020-00153-6, p. 16, ISSN 2213-7467. URL <https://doi.org/10.1186/s40323-020-00153-6>. 174
- DUBREUIL, S., N. BARTOLI, C. GOGU et T. LEFEBVRE. 2020, «Towards an efficient global multidisciplinary design optimization algorithm», *Structural and Multidisciplinary Optimization*, vol. 62, n° 4, doi: 10.1007/s00158-020-02514-6, p. 1739–1765, ISSN 1615-1488. URL <https://doi.org/10.1007/s00158-020-02514-6>. 135, 189
- DUBREUIL, S., N. BARTOLI, T. LEFEBVRE et C. GOGU. 2018, «Efficient global multidisciplinary optimization based on surrogate models», dans *2018 Multidisciplinary Analysis and Optimization Conference*, doi: 10.2514/6.2018-3745. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2018-3745>. 177, 188
- DUPUIS, R., J.-C. JOUHAUD et P. SAGAUT. 2018, «Surrogate modeling of aerodynamic simulations for multiple operating conditions using machine learning», *AIAA Journal*, vol. 56, n° 9, p. 3622–3635. 127
- ECONOMON, T. D., F. PALACIOS, S. R. COPELAND, T. W. LUKACZYK et J. J. ALONSO. 2016, «SU2 : An open-source suite for multiphysics simulation and design», *AIAA Journal*, vol. 54, n° 3, p. 828–846. 165
- EDF. 1989–2017, «Finite element *code_aster*, analysis of structures and thermomechanics for studies and research», Open source on www.code-aster.org. 146
- HALDER, R., K. FIDKOWSKI et K. MAKI. 2022, *Local Non-Intrusive Reduced Order Modeling Using Isomap*, p. 0081, doi: 10.2514/6.2022-0081. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2022-0081>. 175
- HIRSCH, C. 2007, *Numerical computation of internal and external flows : The fundamentals of computational fluid dynamics*, Elsevier. 165
- KATZ, J. et A. PLOTKIN. 2001, *Low-speed aerodynamics*, vol. 13, Cambridge university press. 146, 147
- KENWAY, G., G. KENNEDY et J. R. R. A. MARTINS. 2014, «Aerostructural optimization of the common research model configuration», dans *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, doi: 10.2514/6.2014-3274. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2014-3274>. 183
- MAATOUK, H. et X. BAY. 2017, «Gaussian process emulators for computer experiments with inequality constraints», *Mathematical Geosciences*, vol. 49, n° 5, p. 557–582. 174
- MCKAY, M., R. BECKMAN et W. CONOVER. 1979, «A comparison of three methods for selecting values of input variables in the analysis of output from a computer code», *Technometrics*, vol. 21, doi: 10.1080/00401706.1979.10489755, p. 239–245. 130
- MOHAMMADI, H., R. L. RICHE, N. DURRANDE, E. TOUBOUL et X. BAY. 2016, «An analytic comparison of regularization methods for Gaussian processes», *arXiv preprint arXiv:1602.00853*. 130

- PEDREGOSA, F., G. VAROQUAUX, A. GRAMFORT, V. MICHEL, B. THIRION, O. GRISEL, M. BLONDEL, P. PRETTENHOFER, R. WEISS, V. DUBOURG, J. VANDERPLAS, A. PASSOS, D. COURNAPEAU, M. BRUCHER, M. PERROT et E. DUCHESNAY. 2011, «Scikit-learn : Machine learning in Python», *Journal of Machine Learning Research*, vol. 12, p. 2825–2830. 151
- RENDALL, T. C. S. et C. B. ALLEN. 2007, «Unified fluid–structure interpolation and mesh motion using radial basis functions», *International Journal for Numerics Methods in Engineering*, vol. 78, p. 1519–1559. 147
- SCHOLTEN, W. et D. HARTL. 2021, «Uncoupled method for static aeroelastic analysis», *Journal of Fluids and Structures*, vol. 101, doi: <https://doi.org/10.1016/j.jfluidstructs.2021.103221>, p. 103 221, ISSN 0889-9746. URL <https://www.sciencedirect.com/science/article/pii/S0889974621000049>. 146
- SIROVICH, L. 1987, «Turbulence and the dynamics of coherent structures. I. Coherent structures», *Quarterly of applied mathematics*, vol. 45, n° 3, p. 561–571. 128
- SUDRET, B. 2008, «Global sensitivity analysis using polynomial chaos expansions», *Reliability engineering & system safety*, vol. 93, n° 7, p. 964–979. 139, 188
- WANG, Z., I. AKHTAR, J. BORGGAARD et T. ILIESCU. 2011, «Two-level discretizations of non-linear closure models for proper orthogonal decomposition», *Journal of Computational Physics*, vol. 230, n° 1, p. 126–146. 128
- WASHABAUGH, K., D. AMSALLEM, M. ZAHR et C. FARHAT. 2012, «Nonlinear model reduction for CFD problems using local reduced-order bases», dans *42nd AIAA Fluid Dynamics Conference and Exhibit*, p. 2686. 174
- WILLIAMS, C. K. et C. E. RASMUSSEN. 2006, *Gaussian processes for machine learning*, vol. 2, MIT press Cambridge, MA. 151
- ZAHM, O., M. BILLAUD-FRIESS et A. NOUY. 2017, «Projection-based model order reduction methods for the estimation of vector-valued variables of interest», *SIAM Journal on Scientific Computing*, vol. 39, n° 4, p. A1647–A1674. 128

Conclusion et perspectives

Conclusion

Dans cette thèse, nous nous sommes intéressés au développement de nouvelles méthodes permettant de réduire le coût numérique de l'optimisation, pour des problèmes monodisciplinaires et multidisciplinaires. Dans un premier temps, nous avons adapté les méthodes d'estimation de l'erreur due à la projection de Petrov-Galerkin à des problèmes dits *goal-oriented* afin de développer un couplage entre [POD](#) et optimisation bayésienne. Dans un second temps, nous avons développé une méthode d'approximation de solveurs disciplinaires dans un contexte multidisciplinaire basée sur la [POD+I](#). Cette approximation a permis de résoudre des [MDA](#) par enrichissements successifs puis des problèmes [MDO](#) sur différents cas d'application.

Dans le Chapitre 2, un focus sur la résolution d'un problème d'optimisation monodisciplinaire a été effectué, l'objectif étant d'utiliser la [MOR](#) dans un cadre *goal-oriented* afin d'approcher une quantité d'intérêt linéaire puis de trouver les paramètres optimisant cette quantité. Cette approximation est obtenue par projection sur une base réduite qui est ajustée au fur et à mesure de l'optimisation. L'erreur de projection est estimée à partir du résidu et plusieurs contributions ont été apportées sur l'amélioration de celle-ci. Premièrement, comme l'estimation de l'erreur est usuellement bornée grâce à l'inégalité de Cauchy-Schwarz et que les vecteurs intervenant dans l'inégalité sont rarement colinéaires (car de grande dimension), l'erreur est souvent surestimée. Pour pallier ce phénomène, une estimation d'un quantile du cosinus entre ces deux vecteurs a été proposée. Deuxièmement, l'erreur du problème primal est estimée par préconditionnement par [PIO](#) du résidu qui se révèle coûteux lorsque aucune décomposition affine des éléments n'est disponible. Deux méthodes ont été proposées pour réduire ce coût en se basant sur la diagonale des opérateurs interpolés. Un couplage entre un algorithme d'optimisation bayésienne ([EGO](#)) et la [MOR](#) a été proposé. Plus précisément, l'estimation de l'erreur est utilisée pour déterminer si la solution réduite peut être utilisée à la place de la solution exacte dans les zones jugées non optimales. Nous avons alors testé ces méthodes sur un cas de dimensionnement de la structure d'une aile d'avion sur 3 paramétrages composés de 1, 4 et 12 variables de design. L'analyse statistique permet de diminuer de deux ordres de grandeur la surestimation de l'erreur. Notons que cette diminution est cruciale pour la méthode développée puisque le critère permettant de déterminer si l'utilisation du modèle complet est nécessaire ou non est basé sur l'estimation de cette erreur et permet de réduire d'environ 20% le nombre d'appels au solveur complet. Dans un second temps, nous avons vérifié que les estimations des coefficients de la [PIO](#) sont précises sur ce problème. Les deux contributions ont alors été couplées et ont permis de réduire le coût numérique de l'optimisation de 13% lorsque la contrainte du problème d'optimisation n'est pas prise en compte et de 65% sinon. Un des objectifs de cette recherche était d'étendre le cadre d'application de la projection de Petrov-Galerkin et du calcul de l'erreur

aux problèmes multidisciplinaires. Cependant, nous nous sommes aperçu que le calcul du système projeté et de cette erreur sont trop coûteux. De plus, la projection sur la base **POD** est une opération intrusive qui ne respecte pas le troisième défi de notre étude. Pour ces raisons, nous avons privilégié dans la suite une méthode d'interpolation des coefficients sur la base réduite (**POD+I**).

Dans le Chapitre 3 nous avons développé une méthode d'approximation des solveurs disciplinaires appelée **DPOD+I**. Cette méthode consiste à projeter les variables de couplage sur des bases **POD** puis à interpoler les solveurs disciplinaires, dont les dimensions des entrées et des sorties sont réduites, grâce à des **GP** conditionnés. Une des premières difficultés a été l'entraînement de ces **DPOD+I** puisque un **DoE** de solutions de **MDA** conduit à des problèmes de conditionnement des **GP** et a un coût trop élevé. Une méthode d'entraînement, basée sur un découplage des variables de design et des variables de couplage, a été proposée et testée. Les approximations par **DPOD+I** sont alors couplées afin d'obtenir un méta-modèle de la **MDA**. Comme ce méta-modèle n'est pas suffisamment précis après la phase d'entraînement, l'idée a été d'estimer l'erreur commise par ce méta-modèle afin de l'enrichir si cela s'avère nécessaire. Les incertitudes sur la **MDA** sont évaluées par propagation des incertitudes liées aux **GP** disciplinaires, définissant une **MDA** aléatoire dont la génération des solutions a été développée. La précision du méta-modèle de la **MDA** est alors évaluée à partir de la dispersion des solutions de la **MDA** aléatoire. Lorsque la **MDA** n'est pas considérée comme suffisamment précise, un enrichissement d'une des disciplines est proposé. Ce choix est effectué par estimation de l'influence de l'incertitude liée à chaque discipline sur la dispersion des **MDA** aléatoires. La première application de ces approximations disciplinaires a été de développer un solveur de **MDA** par exploitation des enrichissements successifs afin d'être testée sur deux cas aérostructures. Dans le premier cas, utilisant un solveur aérodynamique fondé sur une méthode **VLM** et une méthode d'éléments finis pour la structure, la méthode a été testée sur deux variantes faisant intervenir respectivement 4 et 8 variables de design. Ce cas d'application respecte bien les trois premiers défis abordés dans l'introduction. Les solveurs numériques utilisés sont des boîtes noires coûteuses dont l'information sur le gradient n'est pas connue et les variables de couplage sont de grande dimension ($d^1 = 43\,416$ et $d^2 = 2100$). La méthode proposée a permis de réduire le coût numérique nécessaire à la résolution de 100 **MDA** de 75% (variante à 4 variables) ou de 80% (variante à 8 variables). Pour les deux variantes, l'erreur relative moyenne est d'environ 1% correspondant à l'erreur cible imposée par l'erreur sur la **POD** et l'erreur sur le quantile. La méthode a par la suite été testée sur un second cas test utilisant la même discipline structure mais une intégration des équations d'Euler est utilisée pour le calcul des efforts aérodynamiques induisant des variables de couplage de plus grande dimension ($d^2 = 445\,812$). Ce nouveau cas a été plus difficile à approcher par **DPOD+I** puisque la sortie du solveur aérodynamique n'est pas représentable sur une base **POD** de faible dimension. En conséquence, les **MDA** aléatoires ne convergent pas avec la méthode initialement développée et deux solutions ont alors été proposées. Dans la première, une réduction des sources d'incertitude de la **MDA** aléatoire est proposée. La seconde repose sur le développement d'une nouvelle méthode de résolution de la **MDA**, appelée solveur **DPOD+I** Jacobi, afin de définir un critère d'arrêt basé sur l'enrichissement plutôt que sur la convergence des **MDA** aléatoires. La résolution par **DPOD+I** permet de réduire d'environ 66% le nombre d'appels aux solveurs pour la résolution d'une **MDA** en négligeant l'impact de l'entraînement. L'erreur relative moyenne est d'environ 2% pour les deux disciplines ce qui reste en dessous de la valeur cible de 3%. Finalement l'approximation par **DPOD+I** a été utilisée pour la résolution d'un problème **MDO**. Plusieurs couplages ont alors été proposés : un couplage **MDF**, un couplage **SBO** et

un couplage avec l'algorithme **EGMDO**. Ces couplages ont été comparés à la formulation **MDF** de référence sur un problème inverse utilisant la variante à 4 variables de design du premier cas aérostructure. Le nombre d'appels aux solveurs nécessaires à la **MDO** a été diminué d'environ 97% tout en améliorant la précision de l'optimisation.

Finalement, la méthode proposée a permis de résoudre un problème **MDO** permettant de répondre aux différents défis abordés lors de l'introduction. La **POD+I** a permis d'approcher des solveurs disciplinaires coûteux sans être intrusif et un algorithme d'entraînement spécifique au contexte multidisciplinaire a été proposé afin de limiter le nombre d'appels à ces solveurs. La **POD** a été utilisée pour la réduction des dimensions des variables de couplage et a permis le couplage avec la méthode **EGMDO**. Ce couplage a permis de réduire encore le nombre d'appels aux solveurs disciplinaires pour la résolution d'un problème **MDO**. Finalement, même si la méthode est adaptable à un plus grand nombre de disciplines, sa mise en oeuvre reste une des perspectives de la thèse.

Perspectives

Perspectives sur l'optimisation monodisciplinaire

Une des limitations de la méthode développée concerne son cadre d'application. En effet, elle est restreinte aux problèmes linéaires, avec une quantité d'intérêt linéaire. Des méthodes ont été proposées pour l'approximation de solutions de problèmes non-linéaires, néanmoins, l'estimation de l'erreur commise sur la projection est plus difficile à estimer à cause des non-linéarités. Par exemple, la **DEIM** (**CHATURANTABUT et SORENSEN [2010]**) permet d'obtenir une approximation d'un solveur non-linéaire et des travaux ont été proposés pour l'estimation de cette erreur (**WIRTZ et collab. [2012]**; **XIAO [2019]**; **NUR-TAJ HOSSAIN et GHOSH [2020]**). Ceci permettrait d'étendre la méthode à une plus grande classe de problèmes.

En particulier, l'un des objectifs du Chapitre 2 était de tester les capacités de la projection de Petrov-Galerkin pour des problèmes multidisciplinaires. La projection étant performante sur des cas structure et aérodynamique, la **MDA** peut être approchée en remplaçant les solveurs disciplinaires par des approximations par **POD**. Cependant, l'estimation de l'erreur commise est difficile à utiliser. En effet, le couplage non-linéaire ne permet pas de propager l'erreur sur la solution du problème couplé puisque seule la norme de l'erreur est estimée. Une solution serait d'apprendre le lien entre les erreurs sur les deux disciplines et l'erreur sur le problème couplé. Par exemple, **DROHMANN et CARLBERG [2015]** construisent un méta-modèle apprenant le lien entre l'estimation du problème primal et la vraie erreur associée.

Finalement, notons que les performances de notre méthode pourrait être significativement améliorées lorsqu'une décomposition affine des éléments est disponible. Même si cette décomposition est difficilement approchable dans le cas multidisciplinaire, à cause du nombre de variables de couplage, le contexte monodisciplinaire est bien approprié. De nombreuses méthodes ont été proposées permettant notamment de construire plusieurs approximations pour différentes zones de l'espace de design (**DANIEL et collab. [2020]**) afin d'appliquer ce type de méthode à de fortes non-linéarités sur l'espace de design. La décomposition permet alors de réduire drastiquement le coût de calcul et d'autres méthodes de **PIO** pourraient s'avérer plus intéressantes.

Perspectives sur l'optimisation multidisciplinaire

Notre méthode est basée sur l'approximation des solveurs disciplinaires par projection des variables de couplage sur une base **POD** globale puis par interpolation des coefficients. Cependant, d'autres techniques pourraient être mises en oeuvre pour la réduction de dimension des variables de couplage comme par exemple l'utilisation de plusieurs bases **POD** afin de prendre en compte des phénomènes locaux (**WASHABAUGH et collab.** [2012]; **DANIEL et collab.** [2020]). Une autre solution serait d'utiliser des méthodes d'approximation adaptées aux solveurs non-linéaires comme l'Isomap (?) notamment combinée avec des techniques d'interpolation par **HALDER et collab.** [2022].

Nous avons observé sur le deuxième cas test aérodynamique qu'une réduction des sources d'incertitude est nécessaire afin d'assurer la convergence des **MDA** aléatoires. En effet, lors du calcul d'une solution de la **MDA** aléatoire, les trajectoires de chaque **GP** disciplinaire sont calculées via la somme de la moyenne du **GP** et d'une estimation de l'erreur commise via la variance. Additionner toutes ces petites erreurs sur un problème couplé peut engendrer une forte erreur sur la solution couplée. Si les erreurs commises sur chaque **GP** disciplinaire sont trop importantes, il est possible que le problème couplé n'ait pas de solution. Pour pallier ce problème la construction des **GP** disciplinaires pourrait être adaptée afin d'imposer la convergence des solutions de la **MDA** aléatoire. Des **GP** respectant des contraintes d'inégalité (**MAATOUK et BAY** [2017]) ou une divergence nulle (**JIDLING et collab.** [2017]) ont notamment été proposés.

Finalement, la méthode a été développée sur un cas faisant intervenir deux disciplines. Cependant, il est possible d'appliquer cette méthode avec un plus grand nombre de disciplines. L'une des difficultés sera alors l'augmentation du nombre de variables de couplage en entrée des solveurs disciplinaires. En effet, chaque variable de couplage peut être projetée sur une base **POD** adaptée ou par une autre méthode de **MOR**. Cependant, les **GP** disciplinaires auront en entrée les coefficients de chaque variable de couplage projetée (sauf celle correspondante à la sortie de cette discipline). La méthode d'entraînement des approximations disciplinaires devra elle-aussi être adaptée puisque des valeurs initiales pour chaque variable de couplage (sauf une) devront être générées. De plus, il serait intéressant de pouvoir effectuer des approximations de quantités qui ne sont pas liées au problème couplé mais dépendantes des variables de couplage convergées comme par exemple la fonction objectif ou les contraintes. Plusieurs sources d'incertitude seraient alors ajoutées à la fonction objectif aléatoire et une stratégie d'enrichissement pourrait être mise en oeuvre pour chaque approximation.

Références

- CHATURANTABUT, S. et D. C. SORENSSEN. 2010, «Nonlinear model reduction via discrete empirical interpolation», *SIAM Journal on Scientific Computing*, vol. 32, n° 5, p. 2737–2764. 195
- DANIEL, T., F. CASENAVE, N. AKKARI et D. RYCKELYNCK. 2020, «Model order reduction assisted by deep neural networks (rom-net)», *Advanced Modeling and Simulation in Engineering Sciences*, vol. 7, n° 1, doi: 10.1186/s40323-020-00153-6, p. 16, ISSN 2213-7467. URL <https://doi.org/10.1186/s40323-020-00153-6>. 195, 196
- DROHMANN, M. et K. CARLBERG. 2015, «The ROMES method for statistical modeling of reduced-order-model error», *SIAM/ASA Journal on Uncertainty Quantification*, vol. 3, n° 1, p. 116–145. 195

- HALDER, R., K. FIDKOWSKI et K. MAKI. 2022, *Local Non-Intrusive Reduced Order Modeling Using Isomap*, p. 0081, doi: 10.2514/6.2022-0081. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2022-0081>. 196
- JIDLING, C., N. WAHLSTROM, A. WILLS et T. B. SCHÖN. 2017, «Linearly constrained Gaussian processes», dans *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, ISBN 9781510860964, p. 1215–1224. 196
- MAATOUK, H. et X. BAY. 2017, «Gaussian process emulators for computer experiments with inequality constraints», *Mathematical Geosciences*, vol. 49, n° 5, p. 557–582. 196
- NURTAJ HOSSAIN, M. et D. GHOSH. 2020, «Adaptive reduced order modeling for nonlinear dynamical systems through a new a posteriori error estimator : Application to uncertainty quantification», *International Journal for Numerical Methods in Engineering*, vol. 121, n° 15, doi: <https://doi.org/10.1002/nme.6365>, p. 3417–3441. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.6365>. 195
- WASHABAUGH, K., D. AMSALLEM, M. ZAHR et C. FARHAT. 2012, «Nonlinear model reduction for CFD problems using local reduced-order bases», dans *42nd AIAA Fluid Dynamics Conference and Exhibit*, p. 2686. 196
- WIRTZ, D., D. SORENSEN et B. HAASDONK. 2012, «A posteriori error estimation for DEIM reduced nonlinear dynamical systems», *SIAM Journal on Scientific Computing*, vol. 36, doi: 10.1137/120899042, p. A311–A338. 195
- XIAO, D. 2019, «Error estimation of the parametric non-intrusive reduced order model using machine learning», *Computer Methods in Applied Mechanics and Engineering*, vol. 355, doi: <https://doi.org/10.1016/j.cma.2019.06.018>, p. 513–534, ISSN 0045-7825. URL <https://www.sciencedirect.com/science/article/pii/S0045782519303615>. 195

Annexe A

Annexes

A.1 Approximation de $\hat{f}^{obj}$ par PCE

Un échantillon de $\mathcal{X}$ dénoté DoE_{UQ} de taille n_{UQ} contenant les éléments $\{\mathbf{x}_{(1)}, \dots, \mathbf{x}_{(n_{UQ})}\}$ est généré par LHS. De plus, un échantillon noté Ξ_{UQ} de taille n_{MC} contenant les éléments $\{\Xi_{(1)}, \dots, \Xi_{(n_{MC})}\}$ est généré. À partir des Eq. (1.47) et Eq. (1.48), la fonction objectif aléatoire est évaluée pour chaque paramètre et pour chaque échantillon de Ξ_{UQ} : $\hat{f}^{obj}(\mathbf{x}_{[j]}, \Xi_{(k)})$, $j = 1, \dots, n_{UQ}$, $k = 1, \dots, n_{MC}$. En chaque point $\mathbf{x}_{[j]}$, une PCE est construite à partir des échantillons de la fonction aléatoire :

$$\hat{f}^{obj}(\mathbf{x}, \Xi) \approx \hat{f}_x^{PCE}(\Xi) = \sum_{l=1}^P a_l(\mathbf{x}) H_l(\Xi), \quad \forall \mathbf{x} \in DoE_{UQ} \quad (A.1)$$

où H_l , $l = 1, \dots, P$ sont les polynômes de Hermite à n_d variables, P est le nombre de polynômes choisis pour l'approximation et $a_l(\mathbf{x})$, $l = 1, \dots, P$ sont les coefficients de la PCE. Généralement, le nombre de termes P retenus est $P = \frac{(n_d + deg)!}{n_d! deg!}$ où deg est le degré maximal des polynômes de Hermite choisi. Le calcul des coefficients est effectué par régression (BERVEILLER et collab. [2006], SUDRET [2008]).

A.2 Expansion Karhunen-Loève

L'objectif est d'étendre ici l'approximation $\hat{f}_x^{PCE}$ obtenue pour tout $\mathbf{x}$ dans DoE_{UQ} à tout l'espace $\mathcal{X}$. Notons qu'une solution admissible serait d'interpoler les P coefficients de la PCE $a_l(\mathbf{x})$ définis dans l'Eq. (A.1) sur le domaine $\mathcal{X}$ par GP. Seulement, il a été montré par DUBREUIL et collab. [2018] que la méthode que l'on va exposer est plus intéressante. En effet, l'expansion KL a été utilisée par ARNST et collab. [2012] pour réduire la dimension du problème stochastique pour la résolution de problèmes couplés non linéaires. Nous allons utiliser cette réduction afin de limiter le nombre de quantités à interpoler et ainsi de réduire les sources d'incertitude.

Cette réduction est effectuée par le calcul de l'expansion KL (KARHUNEN [1947]; LOEVE [1948]) d'un champ aléatoire. En premier lieu nous allons rappeler quelques notions sur l'expansion KL ou décomposition KL aussi nommée *Proper Orthogonal Decomposition* dans le cadre de la réduction d'ordre de modèle que nous avons détaillée dans la Section 1.3.3.1. Soit $Y(\mathbf{x}, \xi)$ un champ aléatoire réel aléatoire d'ordre deux, dont la moyenne et sa covariance sont notées respectivement $\mu_Y : \mathcal{X} \rightarrow \mathbb{R}$ et $k_Y : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. Son expansion

KL est définie par :

$$Y(\mathbf{x}, \xi) = \mu_Y(\mathbf{x}) + \sum_{i=0}^{\infty} \gamma_i(\xi) \sqrt{\lambda_i} \phi_i(\mathbf{x}) \quad (\text{A.2})$$

convergeant au sens de la moyenne quadratique. γ_i sont des variables centrées et réduites, mutuellement décorélées vérifiant :

$$\gamma_i(\xi) = \frac{1}{\sqrt{\lambda_i}} \int_{\mathcal{X}} (Y(\mathbf{x}, \xi) - \mu_Y(\mathbf{x})) \phi_i(\mathbf{x}) d\mathbf{x} \quad (\text{A.3})$$

où λ_i et ϕ_i sont respectivement les valeurs propres et vecteurs propres de la fonction de corrélation k_Y solutions de l'équation de Fredholm du second ordre :

$$\int_{\mathcal{X}} k_Y(\mathbf{x}, \mathbf{x}') \phi_i(\mathbf{x}') d\mathbf{x}' = \lambda_i \phi_i(\mathbf{x}) \quad (\text{A.4})$$

Notons qu'en pratique, des méthodes d'intégration numérique (BETZ et collab. [2014]) sont utilisées pour la résolution de l'Eq. (A.4). Dans le cas particulier où Y est un GP, les variables aléatoires γ_i sont gaussiennes, centrées et réduites. Dans ce cas, le fait qu'elles soient décorélées impliquent qu'elles sont même indépendantes. Sinon, ces variables aléatoires sont obtenues grâce à l'Eq. (A.3).

Dans notre étude, nous nous intéressons à la décomposition KL d'un vecteur aléatoire particulier :

$$\hat{f}_{DoEUQ}^{PCE}(\Xi) = \{\hat{f}_{\mathbf{x}}^{PCE}(\Xi), \forall \mathbf{x} \in DoEUQ\} \quad (\text{A.5})$$

Dans ce cas particulier, où le vecteur aléatoire est obtenu à partir d'une PCE, il a été montré par ARNST et collab. [2012] que l'expansion KL de $\hat{f}^{PCE}$ peut être calculée à partir des coefficients de la PCE :

$$\hat{f}_{DoEUQ}^{PCE}(\Xi) \approx \tilde{f}_{DoEUQ}^{PCE}(\Xi) = \mu_{\hat{f}^{PCE}} + \sum_{k=1}^{n_{UQ}} \left(\sum_{l=2}^P \mathbf{a}_l^T \boldsymbol{\varphi}_k H_l(\Xi) \right) \boldsymbol{\varphi}_k \quad (\text{A.6})$$

où $\mathbf{a}_l = \{a_l(\mathbf{x}_{\{1\}}), \dots, a_l(\mathbf{x}_{\{n_{UQ}\}})\}$, $l = 1, \dots, P$, $\mu_{\hat{f}^{PCE}} = \mathbf{a}_1$ et $\boldsymbol{\varphi}_k$ sont les n_{UQ} vecteurs propres de la matrice de covariance $\mathbf{K}_{PCE} = \sum_{l=2}^P \mathbf{a}_l \mathbf{a}_l^T$. En pratique, tous les vecteurs propres de la matrice de covariance de la PCE $\mathbf{K}_{PCE}$ ne sont pas conservés. En effet, on remarque que le champ aléatoire $\hat{f}^{PCE}$ est fortement corrélé, seuls les m vecteurs propres associés aux m valeurs propres les plus importantes sont conservés. Ceci permet de fortement réduire le nombre de vecteurs propres ($m \ll n_{UQ}$) sans pour autant diminuer la précision de l'expansion KL tronquée. La valeur de m est choisie de telle sorte que la somme cumulative des m valeurs propres soit plus grande que $1 - \epsilon_{KL}$ où ϵ_{KL} est une tolérance définie par l'utilisateur (typiquement 10^{-6}). L'expansion KL tronquée est alors :

$$\hat{f}_{DoEUQ}^{PCE}(\Xi) \approx \tilde{f}_{DoEUQ}^{PCE}(\Xi) = \mu_{\hat{f}^{PCE}} + \sum_{k=1}^m \left(\sum_{l=2}^P \mathbf{a}_l^T \boldsymbol{\varphi}_k H_l(\Xi) \right) \boldsymbol{\varphi}_k \quad (\text{A.7})$$

A.3 Interpolation des termes de l'expansion KL par des GPs

Notons que grâce à l'expansion KL, le vecteur aléatoire suivant a été défini :

$$\tilde{f}_{DoEUQ}^{PCE}(\Xi) = \mu_{\hat{f}_{DoEUQ}^{PCE}} + \sum_{k=1}^m \left(\sum_{l=2}^P \mathbf{a}_l^T \boldsymbol{\varphi}_k H_l(\Xi) \right) \boldsymbol{\varphi}_k \quad (\text{A.8})$$

Les composantes de ce vecteur sont des représentations des composantes de $\hat{f}_{DoEUQ}^{PCE}$ définies par $\{\hat{f}_{\mathbf{x}}^{PCE}(\Xi), \forall \mathbf{x} \in DoEUQ\}$. Ainsi, $(\hat{f}_{DoEUQ}^{PCE}(\Xi))_i, i = 1, \dots, n_{UQ}$ est une approximation de $\hat{f}_{\mathbf{x}_{(i)}}^{PCE}(\Xi)$. Le **DoE** $\{\mathbf{x}_{(1)}, \dots, \mathbf{x}_{(n_{UQ})}\} \rightarrow \{(\mu_{\hat{f}_{DoEUQ}^{PCE}})_1, \dots, (\mu_{\hat{f}_{DoEUQ}^{PCE}})_{n_{UQ}}\}$ est utilisé comme plan d'expérience sur lequel un **GP** de la moyenne sera entraîné. On obtient alors une approximation $\hat{\mu}_{\hat{f}_{DoEUQ}^{PCE}}(\mathbf{x}, \boldsymbol{\eta})$. De même, les informations obtenues sur les vecteurs propres $\boldsymbol{\varphi}_k$ sont interpolées par **GP** afin d'obtenir $\hat{\boldsymbol{\varphi}}_k(\mathbf{x}, \boldsymbol{\eta})$. Dans ces deux approximations, la variable $\boldsymbol{\eta}$ représente l'incertitude liée à l'approximation par **GP** prenant en compte la variance du **GP**. Ainsi, en tout point $\mathbf{x}$ une approximation de $\hat{f}^{obj}(\mathbf{x}, \Xi)$ est définie :

$$\hat{f}^{obj}(\mathbf{x}, \Xi) \approx \tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta}) = \hat{\mu}_{\hat{f}_{DoEUQ}^{PCE}}(\mathbf{x}, \boldsymbol{\eta}) + \sum_{k=1}^m \left(\sum_{l=2}^P \mathbf{a}_l^T \hat{\boldsymbol{\varphi}}_k(\mathbf{x}, \boldsymbol{\eta}) \phi_l(\Xi) \right) \hat{\boldsymbol{\varphi}}_k(\mathbf{x}, \boldsymbol{\eta}) \quad (\text{A.9})$$

Ceci permet d'obtenir une approximation de la loi de la fonction objectif que l'on va maintenant utiliser sur un critère inspiré de l'amélioration espérée introduite en Section 1.1.3.3.3.

A.4 Illustration de la résolution d'un problème **MDO** par **EGMDO**

L'illustration des différentes étapes de **EGMDO** est faite sur le cas test analytique proposé en Section 1.2.3.2.

Dans un premier temps, des **GP** de f^1 et f^2 sont construits à partir de deux **LHS** sur l'espace de design et sur l'espace des variables de couplage. Ici, il est supposé que $y^1 \in [-5, 24]$ et $y^2 \in [1, 50]$. Cette information est primordiale pour générer un **LHS** et reste un des inconvénients de **EGMDO** même si une mauvaise estimation des espaces des variables de couplage peut être compensée lors des itérations de **EGMDO**. De plus, un **LHS** dénommé $DoEUQ$ est généré permettant de définir les points où seront calculés les **PCE**. Nous allons dans la suite illustrer les étapes suivantes :

1. Constituer une **MCS** de **MDA** aléatoires obtenues via l'Eq. (1.47) permettant de générer une **MCS** de $\hat{f}^{obj}(\mathbf{x}, \Xi)$ pour chaque point de $DoEUQ$ comme cela est illustré sur la Figure A.1.
2. Calculer une **PCE** à partir des **MCS** afin de caractériser les champs $\hat{f}^{obj}(\mathbf{x}, \Xi), \forall \mathbf{x} \in DoEUQ$ par un nombre fini de variables : les coefficients de la **PCE**. L'approximation obtenue $\hat{f}_{\mathbf{x}}^{PCE}(\Xi)$ est représentée sur la Figure A.2.
3. Interpoler l'approximation $\hat{f}_{\mathbf{x}}^{PCE}(\Xi), \forall \mathbf{x} \in DoEUQ$ à tout $\mathbf{x} \in \mathcal{X}$ par un mélange d'expansion **KL** et d'interpolation par **GP** permettant de générer une approximation de la fonction objectif aléatoire $\tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta})$. Ce champ est illustré dans la Figure A.3. Après l'initialisation, ce champ est très incertain, des enrichissements seront nécessaires pour parvenir à en obtenir une approximation fidèle.
4. À partir de $\tilde{f}^{obj}(\mathbf{x}, \Xi, \boldsymbol{\eta})$, une amélioration espérée est calculée par l'Eq. (1.53). L'amélioration espérée obtenue ainsi que son maximum sont représentés sur la Figure A.4. Ceci permet de définir le point $\mathbf{x}_*$ qui sera ajouté à $DoEUQ$.
5. L'enrichissement des solveurs disciplinaires est ensuite effectué à partir de l'Algorithme 6. Ainsi, tous les points $\mathbf{x}_{(i)} \in DoEUQ$ avec $P_{min}(\mathbf{x}_{(i)}) > \frac{1}{n_{UQ}}$ seront enrichis si l'incertitude liée à l'interpolation des **GP** n'est pas suffisamment faible ($CV(\hat{f}_{\mathbf{x}_{(i)}}^{obj}(\Xi)) > \epsilon_{CV}$).

Grâce aux enrichissements, l'approximation par GP permet de définir avec certitude quel est le point de DoE_{UQ} le plus intéressant pour l'optimisation ainsi qu'une approximation fidèle en ce point.

6. La Figure A.6 reprend l'étape 1 avec les GP enrichis et le nouveau DoE_{UQ} . On remarque que l'Algorithme 6 permet d'obtenir une approximation sans incertitude au deuxième point du DoE_{UQ} où le minimum est situé.
7. Finalement, après trois itérations de l'Algorithme 7, le champ $\tilde{f}^{obj}(\mathbf{x}, \Xi)$ est illustré sur la Figure A.8. L'information autour du minimum est très précise et aucune incertitude liée aux GP ne subsiste après les différents enrichissements. Cela permet alors de définir l'emplacement probable du minimum ainsi qu'une bonne approximation du problème MDO. Le champ aléatoire sur lequel sera déterminé le minimum de la MDO défini en supprimant l'incertitude liée à l'interpolation de l'expansion KL : $\tilde{f}^{obj}(\mathbf{x}, \Xi, 0)$ est illustré sur la Figure A.8.

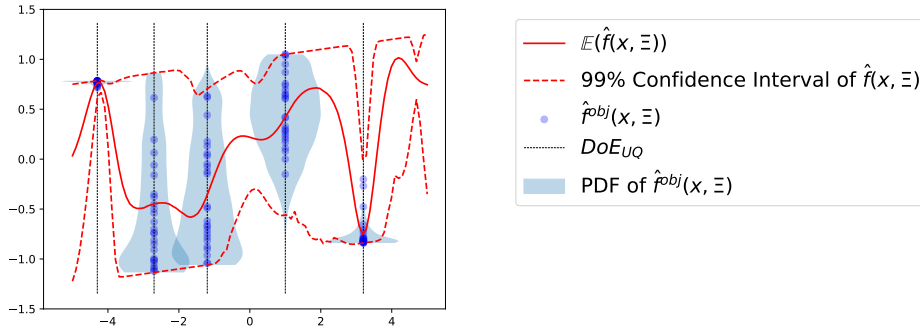


FIGURE A.1 – Illustration de EGMDO sur un cas analytique présentant les MCS de $\hat{f}^{obj}(\mathbf{x}, \Xi)$ obtenu en tout point de DoE_{UQ} ainsi qu'une estimation de la PDF de $\hat{f}^{obj}(\mathbf{x}, \Xi)$. On notera que la loi de $\hat{f}^{obj}$ n'est effectivement pas gaussienne.

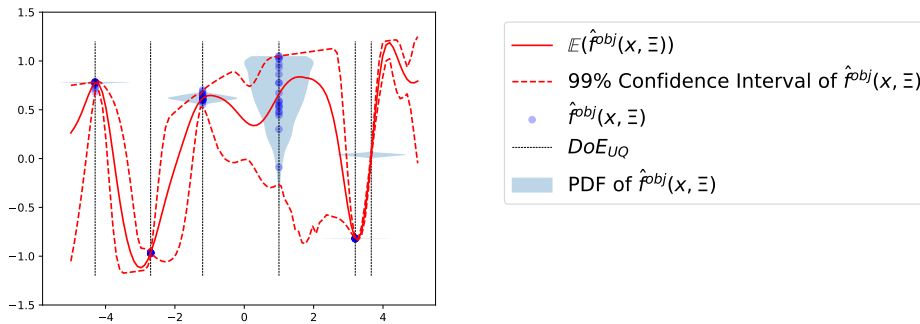


FIGURE A.2 – Illustration de EGMDO sur un cas analytique présentant la fonction à optimiser $f(\mathbf{x}) = f^{obj}(\mathbf{x}, y_*^1(\mathbf{x}), y_*^2(\mathbf{x}))$ et les approximations par PCE $\hat{f}_x^{PCE}(\Xi)$, $\forall \mathbf{x} \in DoE_{UQ}$ dont une estimation de la PDF est générée.

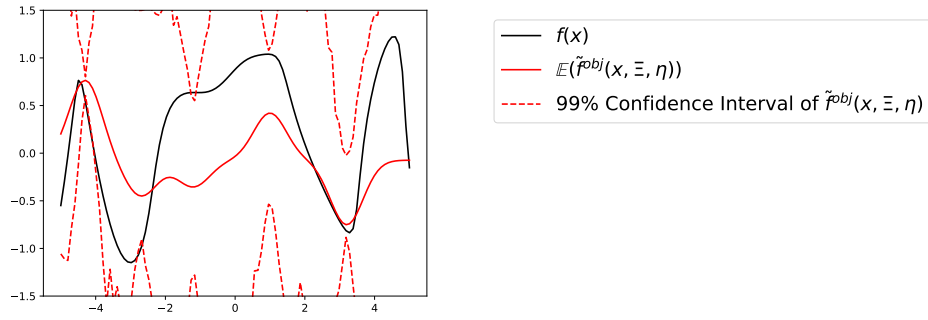


FIGURE A.3 – Illustration de EGMDO sur un cas analytique présentant la fonction à optimiser $f(\mathbf{x}) = f^{obj}(\mathbf{x}, y_*^1(\mathbf{x}), y_*^2(\mathbf{x}))$ et l'approximation du champ de la fonction objectif aléatoire $\tilde{f}^{obj}(\mathbf{x}, \Xi, \eta)$ dont la moyenne et l'intervalle de confiance sont représentés.

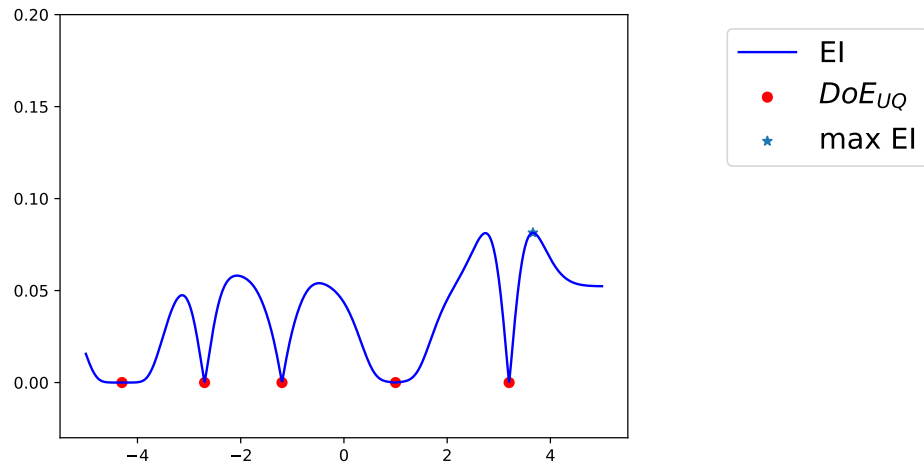
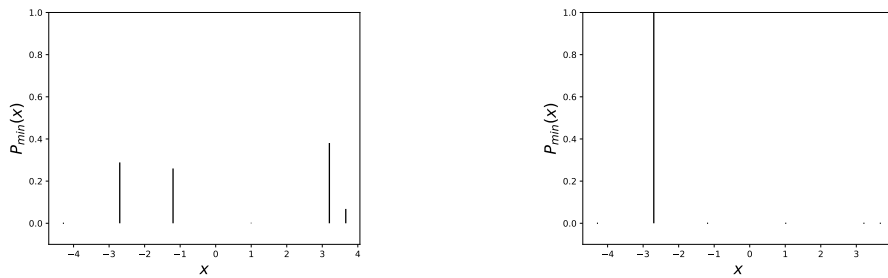


FIGURE A.4 – Illustration de EGMDO sur un cas analytique présentant l'amélioration espérée définie dans l'Eq. (1.53) ainsi que son maximum obtenu par optimisation numérique.



(a) Illustration de EGMDO sur un cas analytique présentant $P_{min}(\mathbf{x}_{(i)})$, $\mathbf{x}_{(i)} \in DoE_{UQ}$ avant enrichissement des solveurs disciplinaires. (b) Illustration de EGMDO sur un cas analytique présentant $P_{min}(\mathbf{x}_{(i)})$, $\mathbf{x}_{(i)} \in DoE_{UQ}$ après enrichissement des solveurs disciplinaires.

FIGURE A.5 – Illustration de EGMDO sur un cas analytique présentant la convergence de $P_{min}(\mathbf{x}_{(i)})$, $\mathbf{x}_{(i)} \in DoE_{UQ}$ grâce aux enrichissements successifs.

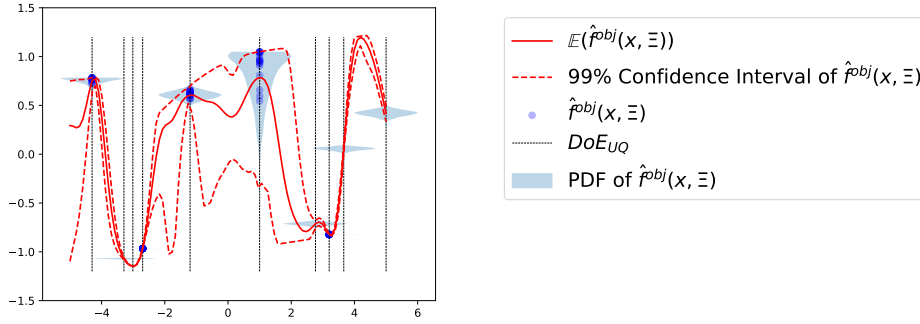


FIGURE A.6 – Illustration de EGMDO sur un cas analytique présentant les MCS de $\hat{f}^{obj}(\mathbf{x}, \Xi)$ obtenus en tout point de DoE_{UQ} ainsi qu’une estimation de la PDF de $\hat{f}^{obj}(\mathbf{x}, \Xi)$ générée à partir de la MCS.

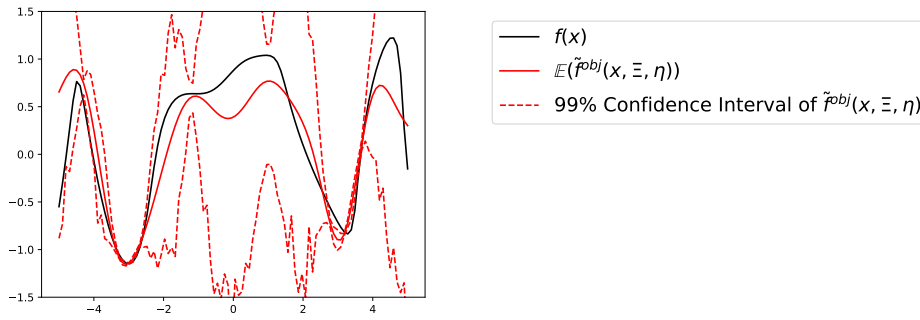


FIGURE A.7 – Illustration de EGMDO sur un cas analytique présentant la fonction à optimiser $f(\mathbf{x}) = f^{obj}(\mathbf{x}, y_*^1(\mathbf{x}), y_*^2(\mathbf{x}))$ et l’approximation du champ de la fonction objectif aléatoire $\tilde{f}^{obj}(\mathbf{x}, \Xi)$ dont la moyenne et l’intervalle de confiance sont représentés après trois itérations de l’Algorithme 7.

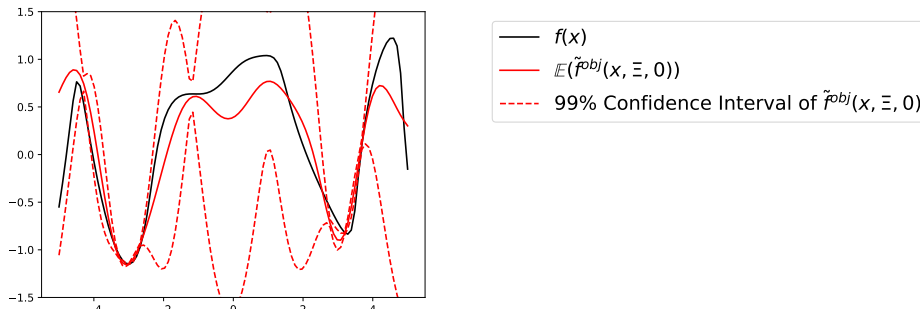


FIGURE A.8 – Illustration de EGMDO sur un cas analytique présentant la fonction à optimiser $f(\mathbf{x}) = f^{obj}(\mathbf{x}, y_*^1(\mathbf{x}), y_*^2(\mathbf{x}))$ et le champ aléatoire sur lequel sera déterminé le minimum de la MDO défini en supprimant l’incertitude liée à l’interpolation de l’expansion KL : $\tilde{f}^{obj}(\mathbf{x}, \Xi, 0)$.

A.5 Références

- ARNST, M., R. GHANEM, E. PHIPPS et J. RED-HORSE. 2012, «Dimension reduction in stochastic modeling of coupled problems», *International Journal for Numerical Methods in Engineering*, vol. 92, n° 11, doi: <https://doi.org/10.1002/nme.4364>, p. 940–968. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4364>. I, II
- BERVEILLER, M., B. SUDRET et M. LEMAIRE. 2006, «Stochastic finite element : a non intrusive approach by regression», *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, vol. 15, n° 1-3, p. 81–92. I
- BETZ, W., I. PAPAIOANNOU et D. STRAUB. 2014, «Numerical methods for the discretization of random fields by means of the karhunen–loève expansion», *Computer Methods in Applied Mechanics and Engineering*, vol. 271, p. 109–129. II
- DUBREUIL, S., N. BARTOLI, C. GOGU, T. LEFEBVRE et J. MAS COLOMER. 2018, «Extreme value oriented random field discretization based on an hybrid polynomial chaos expansion - Kriging approach», *Computer Methods in Applied Mechanics and Engineering*, vol. 332, doi: 10.1016/j.cma.2018.01.009, p. 540–571. URL <https://hal.archives-ouvertes.fr/hal-01712026>. I
- KARHUNEN, K. 1947, «Über lineare methoden in der wahrscheinlichkeitsrechnung», *Amer. Acad. Sci., Ser. A, I*, vol. 37, p. 3–79. I
- LOEVE, M. 1948, «Fonctions aléatoires du second ordre», *Processus Stochastique et Mouvement Brownien*. I
- SUDRET, B. 2008, «Global sensitivity analysis using polynomial chaos expansions», *Reliability engineering & system safety*, vol. 93, n° 7, p. 964–979. I

Annexe B

Liste des acronymes

- AIC** Coefficients d'influence aérodynamiques, ou *Aerodynamic Influence Coefficients*. 147
- ANN** Réseaux de neurones artificiels, ou *Artificial Neural Network*. 17, 38
- AoI** angle d'incidence, ou *Angle of Incidence*. xvii, 149, 151
- ASO** *Asymmetric Subspace Optimization*. 35
- ATC** *Analytical Target Cascading*. 35
- BFGS** *Broyden-Fletcher-Goldfarb-Shanno*. 13, 46
- BLISS** *Bilevel Integrated System Synthesis*. 35, 37
- CO** *Collaborative Optimization*. 35, 38
- CRM** Modèle de Recherche Commun, ou *Common Research Model*. 183
- CSSO** *Concurrent SubSpace Optimization*. 35
- CV** Coefficient de Variation. xviii, 46, 112, 113, 157, 158, 184–187
- DEIM** *Discrete Empirical Interpolation Method*. 54, 55, 60, 62–64, 195
- DoE** Plan d'expérience, ou *Design of Experiment*. xi, xii, xiv, 4, 16–20, 22–24, 26, 40, 43–45, 48, 49, 55, 56, 87, 96, 101–104, 107, 111, 113–115, 121, 125–127, 129, 130, 132, 133, 135, 140, 141, 143, 150, 160, 164, 174–176, 179, 181, 183–185, 187, 194, III
- DPOD+I** Décomposition orthogonale aux valeurs propres et interpolation disciplinaire *Disciplinary Proper Orthogonal Decomposition and Interpolation*. xviii, 123, 125–127, 135, 143, 144, 151, 153–158, 160, 162–164, 167–169, 171–181, 183, 184, 186–189, 194
- EGMDO** *Efficient Global Multidisciplinary Optimization*. iv, xii, xiii, 4, 5, 27, 36, 38, 39, 45, 47, 68, 69, 121–125, 175, 177, 178, 181, 186, 189, 195, III
- EGO** *Efficient Global Optimization*. xi, xvii, 11, 18, 21–25, 37, 39, 42, 47, 65, 68, 69, 86, 87, 101–104, 110–113, 115–117, 136, 175, 176, 184, 193
- EI** Amélioration espérée, ou *Expected Improvement*. 22–24, 45, 46, 48, 179, 186, 189
- EP** *Error-based Penalty*. 67
- ETR** *Error-based Trust Region*. 66
- FSI** Interaction fluide-structure, ou *Fluid-structure interaction*. 28, 29, 123, 125, 146
- GNAT** *Gauss Newton with Approximated Tensors*. 54

- GP** Processus gaussien, ou *Gaussian process*. xi–xiv, xxi, 4, 5, 11, 17–21, 23, 24, 37–43, 46–49, 55, 68, 101–103, 119, 121–127, 129, 130, 132, 133, 135–141, 143–145, 151, 153, 155, 157, 158, 162, 164, 167–181, 183, 187–189, 194, 196, I–IV
- IC** Intervalle de Confiance. xi, 20
- IDF** *Individual Discipline Feasible*. 3, 35–38, 40, 49, 122
- KL** *Karhunen-Loève*. iv, 43, 45, 47, 49, 56, 180, 181, 183, I–IV
- LHS** Échantillonnage par Hypercube Latin, ou *Latin Hypercube Sampling*. 17, 40, 48, 49, 56, 64, 88, 107, 108, 114, 130, 131, 133, 134, 143, 150, 175, 183, I, III
- MAUD** *Modular Analysis and Unified Derivatives*. 35
- MC** *Monte Carlo*. 17, 41, 42, 46, 49, 183
- MCE** Estimation par Monte Carlo, ou *Monte Carlo Estimation*. 45, 46
- MCS** Échantillonnage par Monte Carlo, ou *Monte Carlo Sampling*. 41, 46, 125, 139, 144, 153, 178, 180, 181, III
- MDA** Analyse multidisciplinaire, ou *MultiDisciplinary Analysis*. xi–xv, 2–5, 27–43, 46–49, 51, 52, 63–65, 68, 117, 119, 121–127, 129–133, 135–141, 143–145, 147, 149–151, 153–158, 160, 162–164, 166–169, 171–181, 183–189, 193–196, III
- MDF** Optimisation multidisciplinaire faisable, ou *Multidisciplinary Feasible*. 3, 35–37, 49, 122, 175, 176, 184–186, 188, 194, 195
- MDO** Optimisation multidisciplinaire, ou *MultiDisciplinary Design Optimization*. iv, 2, 3, 5, 27, 29, 34, 35, 37, 47, 49, 51, 68, 69, 121, 122, 125, 178, 181, 183–186, 189, 193–195, III, IV
- MOR** Réduction d'ordre de modèle, ou *Model Order Reduction*. 4, 5, 51, 52, 55, 56, 59–63, 65–69, 85–90, 102, 103, 110, 112, 113, 115–117, 121, 193, 196
- PCA** Analyse en composantes principales, ou *Principal Component Analysis*. 56
- PCE** Expansion par chaos polynomial, ou *Polynomial Chaos Expansion*. iv, 41, 43, 45–47, 49, 55, 139, 142, 180–183, 188, I–III
- PDF** Densité de probabilité, ou *Probability Density Function*. xii, 43, 44
- PGD** Décomposition propre généralisée, ou *Proper Generalized Decomposition*. 52
- PIO** Préconditionnement par interpolation d'opérateurs. xiii, 96, 99, 101, 107–109, 111–113, 116, 117, 193, 195
- PLS** Moindres carrés partiels, ou *Partial Least Square*. 167, 170
- POD** Décomposition Orthogonale aux valeurs propres, ou *Proper Orthogonal Decomposition*. xiii–xv, 4, 17, 24, 52, 54, 56, 57, 60, 62–64, 66, 69, 88, 89, 91, 101–104, 107, 111–114, 116, 121, 123, 125–137, 139–143, 151–155, 157, 158, 160, 164, 166–171, 174, 176–182, 187–189, 193–196
- POD+I** Décomposition orthogonale aux valeurs propres et interpolation, ou *Proper Orthogonal Decomposition and Interpolation*. 4, 5, 55, 60, 63, 64, 69, 117, 121, 127, 143, 144, 155, 156, 162–164, 188, 193–195
- QI** Quantité d'intérêt. 60–62, 69, 87–90, 102–104, 110, 112, 114, 116, 117
- RBF** Fonctions à base radiale, ou *Radial Basis Function*. 17, 29

SAND *Simultaneous Analysis and Design*. [35](#)

SBO *Surrogate Based Optimization*. [21](#), [24](#), [68](#), [175](#), [177](#), [185](#), [186](#), [188](#), [189](#), [194](#)

SVD Décomposition en valeurs singulières, ou *Singular Value Decomposition*. [56](#), [57](#), [130](#), [133](#), [151](#)

TR Région de confiance, ou *Trust Region*. [13](#), [37](#), [67](#)

TRPOD *Trust-Region Proper Orthogonal Decomposition*. [66](#)

VLM *Vortex Lattice Method*. [xii](#), [53](#), [89](#), [106](#), [117](#), [146](#), [164](#), [194](#)

WB2 *Watson Barnes 2*. [23](#)

WB2S *Watson Barnes 2 normalisée*, ou *Scaled Watson Barnes 2*. [23](#)