



HAL
open science

Analyse d'incertitudes et de robustesse pour les modèles à entrées et sorties fonctionnelles.

Mohamed Reda El Amri

► **To cite this version:**

Mohamed Reda El Amri. Analyse d'incertitudes et de robustesse pour les modèles à entrées et sorties fonctionnelles.. Mathématiques [math]. Université Grenoble Alpes, 2019. Français. NNT: . tel-02433324

HAL Id: tel-02433324

<https://theses.hal.science/tel-02433324>

Submitted on 9 Jan 2020

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

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Mathématiques Appliquées**

Arrêté ministériel : 25 mai 2016

Présentée par

Mohamed Reda El Amri

Thèse dirigée par **Clémentine Prieur**
et codirigée par **Céline Helbert**

préparée au sein du **Laboratoire Jean Kuntzmann**
et de l'**École Doctorale Mathématiques, Sciences et Technologies
de l'information, Informatique**

Analyse d'incertitudes et de robustesse pour les modèles à entrées et sorties fonctionnelles.

Thèse soutenue publiquement le **29/04/2019**,
devant le jury composé de :

Hervé Monod

Directeur de recherche, INRA/MaIAGE, Président

Gilles Pagès

Professeur, Sorbonne Universités, Rapporteur

Josselin Garnier

Professeur, Ecole Polytechnique, Rapporteur

Julien Bect

Maître de conférences, CentraleSupélec, Examineur

Clémentine Prieur

Professeur, Université Grenoble Alpes/LJK, Directrice de thèse

Céline Helbert

Maître de conférences, Ecole Centrale de Lyon - ICJ, Co-Directrice de thèse

Delphine Sinoquet

Ingénieur de recherche, IFP Énergies Nouvelles, Co-Encadrante de thèse

Miguel Munoz Zuniga

Ingénieur de recherche, IFP Énergies Nouvelles, Co-Encadrant de thèse

Olivier Lepreux

Ingénieur de recherche, IFP Énergies Nouvelles, Co-Encadrant de thèse



Remerciements

Tout d'abord je voudrais remercier Gilles Pagès et Josselin Garnier d'avoir accepté de relire cette thèse et d'en être rapporteurs. La version finale de ce mémoire a bénéficié de leur lecture attentive et de leurs précieuses remarques. Je tiens à remercier Hervé Monod d'avoir accepté d'être président du jury. Je remercie également tous les membres du jury d'avoir accepté d'assister à la présentation de ce travail.

Je voudrais remercier Clémentine Prieur et Céline Helbert qui m'ont dirigé tout au long de ces trois années de thèse. Elles ont toujours été disponibles, à l'écoute de mes questions, et se sont toujours intéressées à l'avancée de mes travaux. Les nombreuses discussions que nous avons eues ainsi que leurs conseils sont pour beaucoup dans le résultat final de ce travail. Merci aussi à Delphine Sinoquet, Miguel Munoz Zuniga et Olivier Lepreux d'avoir co-encadré ce travail de thèse. Ils ont toujours été là pour moi, et ont manifesté un souci constant de valoriser notre travail et de communiquer à son sujet au sein de l'IFPEN. Nos échanges m'ont permis de découvrir une grande variété de problèmes statistiques issus de la R&D. Enfin, je remercie tous mes encadrants pour leurs nombreuses relectures et corrections de cette thèse qui ont été très appréciables. Pour tout cela merci.

Durant cette thèse, j'ai essentiellement travaillé au Laboratoire Jean Kuntzmann à Grenoble. Je salue donc l'équipe Airsea qui m'a accueilli avec bienveillance et plus particulièrement Laurent Debreu pour sa générosité. Merci à Anne Pierson et à Annie Simon pour leurs disponibilité et gentillesse.

La chaire OQUAIDO et le Groupement De Recherche MASCOT-NUM ont joué un rôle de catalyseur durant cette thèse et je salue les membres de ces communautés avec qui j'ai pu échanger. Je pense à Luc Pronzato, David Ginsbourger, Julien Bect, Bertrand Iooss et Olivier Roustant.

Je profite de cette occasion pour remercier deux de mes anciens professeurs, dont je garde un agréable souvenir : Sophie Mercier, à l'université de Pau, et François Pantigny, en Spé MP.

Un immense merci à mes parents, mes frères ainsi qu'au reste de ma famille pour leur soutien indéfectible.

Résumé

L'objectif de cette thèse est de résoudre un problème d'inversion sous incertitudes de fonctions coûteuses à évaluer dans le cadre du paramétrage du contrôle d'un système de dépollution de véhicules.

L'effet de ces incertitudes est pris en compte au travers de l'espérance de la grandeur d'intérêt. Une difficulté réside dans le fait que l'incertitude est en partie due à une entrée fonctionnelle connue à travers d'un échantillon donné. Nous proposons deux approches basées sur une approximation du code coûteux par processus gaussiens et une réduction de dimension de la variable fonctionnelle par une méthode de Karhunen-Loève.

La première approche consiste à appliquer une méthode d'inversion de type SUR (Stepwise Uncertainty Reduction) sur l'espérance de la grandeur d'intérêt. En chaque point d'évaluation dans l'espace de contrôle, l'espérance est estimée par une méthode de quantification fonctionnelle gloutonne qui fournit une représentation discrète de la variable fonctionnelle et une estimation séquentielle efficace à partir de l'échantillon donné de la variable fonctionnelle.

La deuxième approche consiste à appliquer la méthode SUR directement sur la grandeur d'intérêt dans l'espace joint des variables de contrôle et des variables incertaines. Une stratégie d'enrichissement du plan d'expériences dédiée à l'inversion sous incertitudes fonctionnelles et exploitant les propriétés des processus gaussiens est proposée.

Ces deux approches sont comparées sur des fonctions jouets et sont appliquées à un cas industriel de post-traitement des gaz d'échappement d'un véhicule. La problématique est de déterminer les réglages du contrôle du système permettant le respect des normes de dépollution en présence d'incertitudes, sur le cycle de conduite.

Mots Clés : inversion stochastique, incertitude fonctionnelle, modèles de substitution, réduction de dimension

Abstract

This thesis deals with the inversion problem under uncertainty of expensive-to-evaluate functions in the context of the tuning of the control unit of a vehicule depollution system.

The effect of these uncertainties is taken into account through the expectation of the quantity of interest. The problem lies in the fact that the uncertainty is partly due to a functional variable only known through a given sample. We propose two approaches to solve the inversion problem, both methods are based on Gaussian Process modelling for expensive-to-evaluate functions and a dimension reduction of the functional variable by the Karhunen-Loève expansion.

The first methodology consists in applying a Stepwise Uncertainty Reduction (SUR) method on the expectation of the quantity of interest. At each evaluation point in the control space, the expectation is estimated by a greedy functional quantification method that provides a discrete representation of the functional variable and an effective sequential estimate from the given sample.

The second approach consists in applying the SUR method directly to the quantity of interest in the joint space. Devoted to inversion under functional uncertainties, a strategy for enriching the experimental design exploiting the properties of Gaussian processes is proposed.

These two approaches are compared on toy analytical examples and are applied to an industrial application for an exhaust gas post-treatment system of a vehicle. The objective is to identify the set of control parameters that leads to meet the pollutant emission norms under uncertainties on the driving cycle.

Keywords : stochastic inversion, functional uncertainty, surrogate models, dimension reduction

Table des matières

Introduction	1
1 Estimation d'un ensemble d'excursion par modèle de substitution	5
1.1 Modélisation par processus gaussien	7
1.1.1 Définitions et propriétés de base	7
1.1.2 Processus gaussiens	8
1.1.3 Modélisation par processus gaussiens	10
1.1.4 Synthèse et objectif	11
1.2 Estimation et quantification de l'incertitude associée à un ensemble d'excursion	12
1.2.1 Espérance d'ensembles aléatoires fermés	12
1.2.2 Estimation d'un ensemble d'excursion et l'incertitude associée	15
1.3 Stratégie de réduction séquentielle d'incertitudes	16
1.3.1 La stratégie SUR	16
1.3.2 Exemple d'application	18
2 Intégration numérique	21
2.1 Cas vectoriel	22
2.1.1 Approches Monte-Carlo et quasi-Monte Carlo	22
2.1.2 La quantification vectorielle	23
2.1.2.1 Introduction	23
2.1.2.2 Définitions et propriétés	24
2.1.2.3 Algorithmes existants	26
2.1.2.4 Application en intégration numérique	27
2.2 Cas fonctionnel	28
2.2.1 Réduction de dimension	29
2.2.2 La quantification fonctionnelle	30
2.2.2.1 Introduction	30
2.2.2.2 Constructions gloutonnes	31
2.3 Applications	34
3 Data-driven stochastic inversion via functional quantization	39
3.1 Introduction	40
3.2 Functional data reduction	41
3.2.1 Context	41
3.2.2 The Karhunen-Loève expansion	43
3.2.3 Greedy Functional Quantization	43
3.3 Numerical integration	46
3.4 Background on SUR strategies	51

3.4.1	Random closed set and bayesian framework	52
3.4.2	Vorob'ev approach	52
3.4.3	SUR strategies	53
3.4.4	SUR strategy adapted to noisy observations	54
3.5	Algorithm coupling SUR and functional quantization	54
3.6	Numerical tests	55
3.6.1	Analytical example	56
3.6.2	IFPEN test case	62
3.7	Conclusions	65
3.8	Complément de l'article : un nouvel exemple	65
4	Inversion stochastique sous incertitudes fonctionnelles	69
4.1	Introduction	69
4.2	Problem formulation	70
4.2.1	Finite dimensional subproblem	71
4.2.2	Gaussian process modelling	71
4.2.3	GP model of the unobservable response	72
4.3	Infill strategy for stochastic inversion in presence of functional uncertainty	72
4.3.1	Minimization of the Vorob'ev deviation : choice of next \mathbf{x}	73
4.3.2	Minimization of the variance : choice of next \mathbf{u}	73
4.4	Implementation	75
4.5	Numerical tests	76
4.6	Conclusion	79
5	Application au système de dépollution SCR	83
5.1	La réduction catalytique sélective des oxydes d'azote	83
5.2	Le simulateur numérique : une boîte noire	85
5.2.1	Les entrées du simulateur numérique	85
5.2.2	Les sorties du simulateur numérique	86
5.2.3	Formalisation du problème	87
5.3	Visualisation de données fonctionnelles	88
5.3.1	HDR boxplot	88
5.3.2	Le boxplot fonctionnel	89
5.3.2.1	Notion de profondeur	89
5.3.2.2	Construction du boxplot fonctionnel	90
5.3.3	Application aux cycles de conduite	91
5.4	Réglage des stratégies de contrôle pour respecter les normes de dépollution en présence d'incertitudes	93
5.4.1	Application à un sous problème	94
5.4.1.1	Inversion par quantification fonctionnelle	94
5.4.1.2	Inversion par métamodélisation dans l'espace joint	97
5.4.1.3	Analyse comparative des deux méthodes	99
5.4.2	Application : 9 variables de contrôle et 1 variable fonctionnelle incertaine	100
	Conclusions et perspectives	103
A	Conception du générateur du cycle de conduite	106

Table des figures

1.1	Construction d'un modèle de substitution à partir d'échantillons d'un simulateur numérique coûteux.	6
1.2	Fonctions de covariance classiques (à gauche) et des réalisations des processus gaussiens associés (à droite). Les paramètres sont fixés à $(\sigma^2, \theta) = (1, 0.3)$	9
1.3	A gauche : la fonction inconnue g (courbe noire) et cinq réalisations du processus gaussien conditionné défini par les équations (1.13), intervalle de confiance à 95% (bande en bleu clair). A droite : Approximation de la fonction g (en trait noir) par la moyenne conditionnelle m_n (trait bleu) obtenue à partir de $n = 8$ points d'expériences, intervalle de confiance à 95% (bande en bleu clair).	11
1.4	A gauche : 5 réalisations du processus gaussien conditionnel avec $n = 7$. A droite : la moyenne conditionnelle m_n (trait bleu), la bande de confiance (en bleu clair) et l'estimateur $\Gamma_{\text{plug-in}}$	12
1.5	L'espérance de Vorob'ev Q_{n, α_n^*} (en vert), l'estimateur <i>plug-in</i> $\Gamma_{\text{plug-in}}$ (en bleu), les $n = 8$ observations (points rouges), la moyenne conditionnelle m_n (courbe bleue) et la bande de confiance (bande bleu clair).	16
1.6	Exemple analytique. en haut à gauche : Le vrai ensemble d'excursion (ensemble vert de frontière mauve). En bas à droite : le plan initial (triangles noirs), les points ajoutés par la stratégie SUR (points rouges), la fonction de couverture et l'espérance de Vorob'ev (ensemble vert de frontière bleue) à l'étape $n = 50$. En haut à droite : l'évolution du logarithme de la <i>déviatio</i> n de Vorob'ev en fonction des itérations. En bas à droite : le plan initial (triangles noirs), les points ajoutés par la suite de Sobol (points rouges), la fonction de couverture et l'espérance de Vorob'ev (ensemble vert de frontière bleue) à l'étape $n = 50$	19
2.1	Une suite à faible discrédance (Sobol) et une suite aléatoire dans $[0, 1]^2$ ($l = 500$). 23	
2.2	Une quantification de Voronoi de la loi $\mathcal{N}(0, I_2)$ où $l = 200$	27
2.3	L'ensemble de 100 réalisations Ξ de la variable \mathbf{V} (à gauche), la projection de l'ensemble Ξ sur les deux premières composantes principales $m = 2$ (à droite).	31
2.4	Le quantifieur de taille 10 obtenu à partir de l'ensemble \mathcal{G} (à gauche), le quantifieur associé dans l'ensemble Ξ (à droite).	32
2.5	Les quantifieurs de taille 10 obtenus à partir de l'ensemble \mathcal{G} par la méthode maximin (en rouge) et en réduisant l'erreur de quantification (en bleu) (à gauche), le quantifieur associé dans l'ensemble Ξ par la méthode maximin (à droite).	33
2.6	Temps de calcul en fonction de la taille du quantifieur par l'algorithme 2.25 en bleu et l'algorithme 2.26 en rouge.	33
2.7	Exemple analytique 1. Estimation séquentielle de l'espérance en fonction du nombre de courbes. Les lignes représentent les moyennes. Les intervalles de confiance (Monte Carlo (en vert) and maximin-GFQ (en rouge)).	35

2.8	Exemple analytique 2. Estimation séquentielle de l'espérance en fonction du nombre de courbes. Les lignes représentent les moyennes. Les intervalles de confiance (Monte Carlo (en vert) and maximin-GFQ (en rouge)).	36
2.9	Exemple analytique 2. Estimation séquentielle de l'espérance en fonction du nombre de courbes. Les lignes représentent les moyennes. Les intervalles de confiance (Monte Carlo (en mauve) and maximin-GFQ (en rouge)).	37
3.1	Ξ is a sample of 100 realizations of \mathbf{V} (left) and \mathcal{G} the corresponding representation in the truncated space of coefficients with $m_{KL} = 2$ (right).	44
3.2	Sequential design of 10 points (bold points) in the set of the coefficients \mathcal{G} (left) and their corresponding bold curves in Ξ (right).	45
3.3	Left : Two designs of $l = 10$ points in the 2D-coefficients set \mathcal{G} . Maximin-GFQ (red circle points) and greedy \mathbb{L}^2 -GFQ (blue square points). Right : The corresponding red curves for the Maximin-GFQ procedure (right).	46
3.4	Application 1. Sequential expectation estimation. Lines denote the average estimates, and coloured bands mark the 25-th and 75-th quantiles (Monte Carlo (in green) and maximin-GFQ (in red)). Horizontal axis denotes the number of curves l used for the expectation estimation.	48
3.5	Application 2. Sequential expectation estimation by maximin-GFQ and \mathbb{L}^2 -GFQ methods.	49
3.6	Application 2. Sequential expectation estimation. Lines denote the average estimates, and coloured bands mark the 25-th and 75-th quantiles (Monte Carlo (in green) and maximin-GFQ (in red)). Horizontal axis denotes the number of curves l used for the expectation estimation.	49
3.7	Application 2. Sequential expectation estimation. Lines denote the average estimates, and coloured bands mark the 25-th and 75-th quantiles (Fpca (in blue) and maximin-GFQ (in red)). Horizontal axis denotes the number of curves l used for the expectation estimation.	50
3.8	Application 2. Computation time (in seconds) of the two GFQ methods as a function of point set size (1).	51
3.9	Analytical example. Contour plot of the function g , the set of interest (green) with boundary (red line), the initial design of experiments (black triangles). . .	56
3.10	Analytical example. The relative error obtained by the two GFQ methods for different values of l_0 and ϵ as a function of the number of calls to the function f . . .	57
3.11	Analytical example. Results for $(l_0, \epsilon) = (4, 5.e^{-3})$. Lines denote the average, and coloured bands mark the 25-th and 75-th quantiles (Fpca (in red) and Crude MC (in green)). Top : The Vorob'ev deviation. Bottom : The set estimation error $\mu(\Gamma^* \Delta Q_{n_{\text{last}}, \alpha_{n_{\text{last}}}^*}) / \mu(\Gamma^*)$	59
3.12	Analytical example. Results based on the initial DoE in the case $m_{KL} = 2$. The coverage function, the boundary of the true set (red), the estimated sets (green). . .	61
3.13	Analytical example. Results at the last iteration in the case $m_{KL} = 2$. The coverage function, the boundary of the true set (red), the estimated sets (green). . .	62
3.14	Automotive test case. Sample of 7 real driving cycles.	63
3.15	Automotive test case. Top : Decrease of the Vorob'ev deviation at each iteration when new points are added. Bottom : Evolution of the absolute error Eq. (3.31) and the red line represents the stopping criterion.	64
3.16	Automotive test case. Top : Coverage probability function (grey scale), estimate set (green) after 62 added points and 1575 function evaluations, initial DoE (black triangles), the sequentially added points (red circles). Bottom : number of driving cycles used to estimate expectation at each added point.	64

3.17	Résultats d'estimation pour $m_{KL} = 2$ et pour des critères d'arrêt fixés à $(l_0, \epsilon) = (4, 10^{-2})$ et $(l_0^{\text{SUR}}, \epsilon^{\text{SUR}}) = (4, 5 \cdot 10^{-3})$. L'ensemble estimé par la méthode \mathbb{L}^2 -GFQ (haut à gauche), la méthode maximin-GFQ à l'itération 14 (haut à droite) et par la méthode "Fpca" à l'itération 21.	68
4.1	10 realisations of the Brownian motion (left) and Max-stable process (right).	76
4.2	Analytical example 1 [Brownian motion (left), Max-stable process (right)]. Convergence of Algorithm 8 for $m_{KL} = \{2, 4, 8\}$. 25th and 75th percentiles of the symmetric difference vs. number of simulators calls. The curves are based on 10 random DoE for every run.	77
4.3	Analytical example 2 [Brownian motion (left), Max-stable process (right)]. Convergence of Algorithm 8 for $m_{KL} = \{2, 4, 8\}$. 25th and 75th percentiles of the symmetric difference vs. number of simulators calls. The curves are based on 10 random DoE for every runs.	78
4.4	Analytical example 1 [Brownian motion (left), Max-stable (right)]. Convergence of Algorithm 8 for $m_{KL} = 8$ and Algorithm 9 for $l = \{10, 20\}$. 25th and 75th percentiles of the symmetric difference vs. number of simulators calls. The curves are based on 10 random DoE for every runs.	78
4.5	Analytical example 2 [Brownian motion (left), Max-stable (right)]. Convergence of Algorithm 8 for $m_{KL} = 8$ and Algorithm 9 for $l = \{10, 20\}$. 25th and 75th percentiles of the symmetric difference vs. number of simulators calls. The curves are based on 10 random DoE for every runs.	79
4.6	The computation time (sec.) needed to provide the next evaluation point as a function of iterations.	80
5.1	Injection d'urée (Adblue) en amont du SCR	84
5.2	Fonctionnement SCR et système de contrôle	85
5.3	Le simulateur numérique du cas d'application.	85
5.4	Une illustration de deux cycles de conduite visualisés sur une durée de 1200s (les cycles étudiés durent 5400s).	87
5.5	Bagplot bivarié pour les températures de surface de l'océan pacifique sur une année (à gauche) et sa représentation fonctionnelle (à droite), extrait de [Hyndman and Shang, 2010]	88
5.6	(De gauche à droite) Exemples de bandes (en vert) délimitées par (de gauche à droite) : 2 courbes ; 2 courbes qui se croisent ; 3 courbes.	90
5.7	boxplot fonctionnel pour les températures de surface de l'océan pacifique sur une année	91
5.8	L'échantillon des 1000 cycles de conduite	92
5.9	HDR boxplot de l'échantillon de cycles de conduite	92
5.10	Boxplot fonctionnel de l'échantillon de cycles de conduite	93
5.11	Part de variance expliquée par l'ACP en fonction de la taille de la base	93
5.12	Pic d'ammoniac. La fonction de couverture, le plan d'expériences initial (triangles noirs) et l'estimation initiale de l'ensemble recherché Γ_2^* (en vert).	95
5.13	Pic d'ammoniac. A gauche : La fonction de couverture, le plan d'expériences initial (triangles noirs), l'estimation initiale de l'ensemble recherché Γ_2^* (en vert) et le point proposé par la stratégie SUR (en rouge). A droite : La mise à jour de l'estimation de Γ_2^* (en vert) après une itération.	96

5.14	Pic d'ammoniac. En haut : La <i>déviatiion de Vorob'ev</i> en fonction des itérations SUR. En bas : la fonction de couverture, le plan d'expériences initial (triangles noirs) et l'estimation de l'ensemble recherché Γ_2^* (en vert) après l'ajout de 62 points (en rouge) et 1575 évaluations au total (évaluations pour l'estimation de l'espérance incluses).	96
5.15	Score DeNO _x . A gauche : La <i>déviatiion de Vorob'ev</i> en fonction des itérations SUR. A droite : la fonction de couverture, le plan d'expériences initial (triangles noirs) et l'estimation de l'ensemble recherché Γ_1^* (en vert) après l'ajout de 12 points (en rouge) et 288 évaluations au total (évaluations pour l'estimation de l'espérance incluses).	97
5.16	Le résultat final pour l'estimation de l'intersection $\Gamma_1^* \cap \Gamma_2^*$	97
5.17	Pic d'ammoniac. Le plan d'expériences initial (triangles noirs) et l'estimation initiale de l'ensemble recherché Γ_2^* (en vert).	98
5.18	Pic d'ammoniac. A gauche : La fonction de couverture, le plan d'expériences initial (triangles noirs), l'estimation initiale de l'ensemble recherché Γ_2^* (en vert) et les deux points proposés par la stratégie SUR (en rouge). A droite : La mise à jour de l'estimation de Γ_2^* (en vert) après une itération.	99
5.19	Pic d'ammoniac. A gauche : La <i>déviatiion de Vorob'ev</i> en fonction du nombre d'appels au simulateur f . A droite : la fonction de couverture, le plan d'expériences initial (triangles noirs) et l'estimation de l'ensemble recherché Γ_2^* (en vert) après l'ajout de 400 points (en rouge).	99
5.20	Résultats pour le pic d'ammoniac et pour un nombre d'évaluations du simulateur fixé à 510. L'ensemble d'excursion estimé par la première méthode (ensemble vert de frontière bleue) et la frontière rouge de l'ensemble estimé par la deuxième méthode.	100
5.21	Nuages de points appartenant à l'estimateur de l'intersection de toutes les paires des variables de contrôle (les points rouges). Le plan d'expériences initial à 45 points (les triangles noirs).	101
A.1	Le cycle de conduite NEDC	106
A.2	Les 13 cycles enregistrés concaténés	107
A.3	Histogramme et densité de probabilité des accélérations a_{k+1} sachant ($a_k = 0, v_k = 30$)	107
A.4	Un schéma décrivant la génération des cycles de conduite.	108

Liste des tableaux

1.1	Noyaux de covariance usuels.	9
2.1	Les erreurs de quantification en fonction de la taille du quantifieur optimal pour le mouvement brownien.	31
3.1	Analytical example. <i>Stopping criteria</i> parameters used in the estimation of expectation.	57
3.2	Analytical example. (Left) The relative error obtained by the two GFQ methods for different values of l_0 and ϵ . (Right) The cumulative number of calls to the function f (in brackets are the number of iterations required to reach the stopping criterion in the SUR strategy).	57
3.3	Analytical example. The average relative error obtained by the crude MC, Fpca for different values of l_0 and ϵ (in brackets are the standard deviation for the crude MC and Fpca methods).	58
3.4	Analytical example. The average cumulative number of calls to the function f (written in brackets are the number of iterations required to reach the stopping criterion in the SUR strategy).	58
3.5	Analytical example. (Top) The average set estimation error obtained for different sample size and methods and $m_{KL} = 7$ (in brackets are the standard deviation). (Bottom) The average cumulative number of calls to the function f (in brackets are the number of iterations required to reach the stopping criterion in the SUR strategy).	59
3.6	Analytical example. (Top) The average set estimation error obtained by the Fpca and the \mathbb{L}^2 -GFQ methods for different values of m (in brackets are the standard deviation). (Bottom) The average cumulative number of calls to the function f (in brackets are the number of iterations required to reach the stopping criterion in the SUR strategy).	60
3.7	Analytical example. The explained variance in function of m_{KL}	60
3.8	La part de variance expliquée en fonction de l'argument de troncature m_{KL}	66
3.9	L'erreur d'estimation ou sa moyenne (dans le cas des méthodes "Fpca" et maximin-GFQ) obtenue par les 3 méthodes pour différentes valeurs de m_{KL} (entre parenthèses, l'écart type). La moyenne du nombre total d'appels à la fonction f (entre parenthèses, le nombre d'itérations nécessaires pour atteindre le critère d'arrêt dans la stratégie SUR).	66
4.1	The explained variance in function of m_{KL} for two types of uncertainties.	76
4.2	Analytical example 2 [Max stable process]. Number of simulator calls in function of the estimation error for the two methods presented respectively in Chapter 3 (\mathbb{L}^2 -GFQ) and in Chapter 4 (Kriging in joint space). Bottom : CPU time at iteration 100 and 200 for the methods.	79

5.1	Les 9 paramètres de contrôle.	86
5.2	Les paramètres incertains en entrée du code et leurs distributions de probabilité associées	86
5.3	les domaines de définition et les distributions de probabilité associés aux entrées du code de calcul. Certaines données (*) sont masquées car confidentielles. . . .	94

Introduction

Contexte de l'étude

En Europe, de nouvelles normes se succèdent pour revoir à la baisse les seuils d'émissions des principaux polluants réglementés. Ces normes régulent les émissions de 4 polluants : les particules, le monoxyde de carbone (CO), les hydrocarbures imbrûlés (HC) et les oxydes d'azote (NO_x). Concernant ce dernier, l'ajout d'un filtre à l'échappement s'est avéré indispensable puisqu'il n'était pas possible de réduire ces rejets par des réglages sur le moteur. A l'heure actuelle un des systèmes les plus utilisés pour faire face à cette problématique est le SCR (Selective Catalytic Redution). Ce système se charge de transformer les NO_x en azote et vapeur d'eau. Cette technologie est fiable et a été utilisée depuis plusieurs années dans le traitement des émissions polluantes des installations industrielles. Mettant en jeu un ensemble de réactions très complexes, ce système permet de réduire les NO_x à l'intérieur de la ligne d'échappement à l'aide d'une solution à base d'urée. Cette solution est convertie en NH₃ et ce dernier réduit les NO_x en azote (N₂).

Le système de dépollution, faisant intervenir des réactions physico-chimiques complexes, dépend d'un grand nombre de paramètres à calibrer. Il est souvent représenté par un modèle numérique qui joue un rôle prépondérant dans la conception. Par exemple la simulation numérique est utilisée pour calibrer les paramètres du modèle afin d'optimiser le rendement et donc de réduire plus efficacement les émissions de polluants. Décrivant fidèlement les phénomènes physiques qui gouvernent le fonctionnement des systèmes, ces simulateurs numériques permettent d'analyser et de comparer différentes configurations envisagées avant de mettre en œuvre des expériences réelles plus onéreuses, surtout dans un contexte où les budgets sont de plus en plus restreints. Cependant, le nombre de simulations numériques qu'il est envisageable de réaliser en un temps d'étude raisonnable reste tout de même limité, car ces simulateurs numériques s'appuient sur des codes de calcul généralement complexes dont l'exécution est assez lourde.

Ce travail de thèse a été motivé et financé par l'IFP Énergies Nouvelles dont l'un des objectifs a été de développer des méthodologies et des algorithmes efficaces pour le paramétrage du contrôle de système de dépollution. Nos résultats ont permis de proposer une procédure de configuration du système SCR dans le but de répondre aux normes instaurées par l'UE. Ces résultats sont présentés dans le chapitre 5.

La suite de cette introduction est dédiée à une brève description du problème à résoudre. Nous énonçons les problématiques sous-adjacentes à nos travaux de recherche et détaillons la composition du manuscrit.

Prise en compte des incertitudes

Le système de dépollution étudié dans cette thèse est défini par des paramètres de deux natures différentes :

- des paramètres de contrôle (paramètres de la stratégie de contrôle du système de dépollution)
- des entrées incertaines.

Parmi les entrées incertaines on trouve par exemple les biais des capteurs : l'incertitude associée peut provenir des tolérances de fabrication. Le fonctionnement du système de dépollution dépend aussi du profil de vitesse du véhicule (cycle de conduite), on parle alors d'incertitude environnementale. On écrit ainsi le simulateur numérique sous la forme suivante :

$$y = f(\mathbf{x}, \boldsymbol{\xi}),$$

où

- f est le simulateur numérique,
- \mathbf{x} est le vecteur des paramètres de contrôle du système, encore appelées variables de décision qui caractérisent le fonctionnement du contrôle,
- $\boldsymbol{\xi}$ regroupe les entrées incertaines du système,
- y est la variable de sortie ou quantité d'intérêt.

Dans le cadre de notre travail, nous n'envisageons pas le cas des sorties spatiales ou fonctionnelles. Le simulateur numérique f renvoie la quantité de NO_x rejeté, encore appelé le score DeNO_x . Le vecteur \mathbf{x} regroupe les variables de contrôle du système de dépollution. L'entrée incertaine $\boldsymbol{\xi}$ correspond aux variables incertaines scalaires (biais de capteurs) et fonctionnelle (cycle de conduite).

Les variables incertaines sont modélisées par des variables aléatoires auxquelles sont attribuées des lois de probabilité. Ces lois de probabilité peuvent être choisies par avis d'expert (constructeur des capteurs) ou construites par inférence à partir d'observations, comme c'est le cas pour la variable fonctionnelle dans notre cas d'étude où seul un échantillon de réalisations (courbes temporelles) est mis à notre disposition.

Ensemble d'excursion : problème d'inversion

Dans ce travail de recherche, on s'intéresse à la problématique d'identification de paramètres admissibles pour des systèmes complexes. Plus précisément, on cherche à construire l'ensemble des valeurs des variables de contrôle admissibles par rapport à un seuil prédéfini : dans l'application qui nous intéresse dans la thèse, l'ensemble des valeurs des paramètres de contrôle du système SCR permettant d'être sous les seuils d'émission de polluants imposés par les normes de l'UE. On peut être confronté à un tel problème également dans l'ingénierie de la fiabilité où la sécurité d'un système est souvent modélisée avec un simulateur numérique qui prend en entrée un ensemble de mesures physiques et renvoie le niveau de fiabilité du système. Dans un tel contexte, une quantité d'intérêt est l'ensemble de tous les paramètres physiques qui conduisent à un niveau de sécurité prédéfini. On parle alors de problèmes d'inversion.

Dans notre cas d'étude, les problèmes d'inversion à résoudre doivent tenir compte de l'incertitude induite par les variables incertaines. On parle alors de problèmes d'inversion stochastiques. On souhaite ainsi concevoir des systèmes dont les performances sont robustes en

un sens à préciser. Dans un tel cadre, la prise en compte des incertitudes dans le problème d'inversion peut aboutir à différentes formulations. On pourrait s'intéresser à la résolution du problème en grande probabilité,

$$\Gamma^\pi = \{\mathbf{x}, \mathbb{P}(f(\mathbf{x}, \boldsymbol{\xi}) \leq \text{seuil}) \geq \pi\},$$

où la probabilité $1-\pi$ reflète le risque que l'on est prêt à assumer. Une autre possibilité consiste à prendre l'espérance comme mesure de robustesse. Il s'agit de la formulation considérée dans le cadre de cette thèse. Mathématiquement, cela se traduit par l'identification de l'ensemble d'excursion suivant :

$$\Gamma^* = \{\mathbf{x}, \mathbb{E}_\xi[f(\mathbf{x}, \boldsymbol{\xi})] \leq \text{seuil}\}. \quad (1)$$

Dans ce cas, on cherche les paramètres de contrôle tels que la sortie soit, en moyenne, inférieure au seuil. On justifiera ce choix plus loin dans le manuscrit.

Réduction de modèle et stratégie d'enrichissement

Résoudre le problème (1) ci-dessus ne peut se faire directement sur le simulateur numérique trop coûteux en calcul.

Une solution consiste à remplacer le simulateur numérique par un modèle approché (métamodèle) plus rapide à exécuter, appelé modèle de substitution. Un tel modèle est construit à partir d'un nombre limité d'observations évaluées avec le simulateur coûteux, et permet ensuite de prédire les performances de n'importe quelle configuration du système de façon quasi-instantanée. Ces modèles de substitution sont définis indépendamment du système étudié, ainsi et de par leur nature générique, ils sont employés dans des domaines très variés. Parmi les modèles les plus répandus, on peut citer les approximations polynomiales, les réseaux de neurones ou encore la modélisation par processus gaussiens. Le contrôle de l'erreur de prédiction est primordiale puisqu'il permet le développement de plusieurs techniques d'échantillonnage adaptatif, suivant le problème à résoudre.

Dans le cadre de notre travail, on s'est intéressé à la modélisation par processus gaussiens (krigeage) qui donne accès à une estimation de l'erreur de prédiction ainsi utilisée pour construire un plan d'expériences adaptatif dédié à l'inversion. Cette stratégie de construction de plans d'expériences permet de limiter les évaluations du simulateur en certaines zones de l'espace des entrées permettant ainsi de préciser le modèle aux endroits stratégiques.

Problématiques

Ainsi pour résoudre le problème d'inversion (1) nous proposons d'étendre les techniques existantes d'inversion par processus gaussien ([Rasmussen and Williams, 2006]) au contexte incertain, notamment à la présence d'une variable aléatoire fonctionnelle dont la distribution n'est connue qu'au travers un échantillon donné.

Nos contributions portent sur la quantification des incertitudes fonctionnelles et les problèmes d'inversion stochastique. Ce travail de thèse a pour objectifs de répondre aux problématiques suivantes :

- quantifier l'incertitude associée à une variable fonctionnelle (cycles de conduite) connue à travers un échantillon de réalisations,

- proposer des méthodes de résolution d'un problème d'inversion en présence d'incertitude fonctionnelle.

Organisation du manuscrit

La thèse comporte 5 chapitres organisés comme suit.

Dans le chapitre 1, un bref rappel sur la modélisation par processus gaussiens est présenté. Ensuite, on donne une stratégie pour un choix judicieux des points d'évaluations dans le cas de fonctions déterministes. Dédiée à l'inversion, cette stratégie repose sur des notions d'ensembles aléatoires fermés.

Dans le chapitre 2, on commence par rappeler quelques méthodes élémentaires pour l'estimation de l'espérance dans le cas de variables aléatoires vectorielles ou fonctionnelles. Ensuite, on propose deux méthodes d'estimation séquentielle dans un cadre fonctionnel et où la variable fonctionnelle n'est connue qu'à travers un échantillon de réalisations. Ces deux méthodes reposent sur une réduction de dimension de la variable fonctionnelle par une méthode de Karunhen Loeve. Leur originalité réside dans le fait que l'estimation de l'espérance est faite de manière séquentielle. Les performances de ces méthodes sera illustrée sur des cas analytiques pour lesquels on montre l'efficacité des estimations et ce, à faible coût en comparaison des méthodes Monte Carlo classiques.

Dans le chapitre 3, on propose une méthodologie permettant de résoudre un problème d'inversion sous incertitudes fonctionnelles. Cette approche combine les notions définies dans les chapitres 1 et 2.

Le chapitre 4 propose une nouvelle construction de plan d'expériences adaptatif dédiée à l'inversion sous incertitude fonctionnelle. Cette méthode repose sur une modélisation par processus gaussien et la capacité à calculer semi-analytiquement une mesure de robustesse (l'espérance). Cette approche est appliquée à deux exemples analytiques et comparée à la méthode présentée dans le chapitre 3.

Dans le chapitre 5, on commence par introduire le problème industriel qui a motivé les travaux de cette thèse. Ensuite, on présente des outils de visualisation de données fonctionnelles et une application de ces derniers pour les cycles de conduite. On applique les deux méthodologies développées dans les chapitres précédents pour identifier l'ensemble d'excursion menant à respecter les normes en moyenne. Une analyse comparative des deux méthodes est également présentée.

Enfin, nous concluons ce manuscrit par un rappel sur les méthodologies proposées dans ce document, ainsi que par une description des perspectives envisagées.

Chapitre 1

Estimation d'un ensemble d'excursion par modèle de substitution

Outlines

Ce chapitre a pour objectif d'introduire les méthodes pour la résolution d'un problème d'inversion pour des simulateurs déterministes et coûteux en temps de calcul. On présentera une stratégie séquentielle d'évaluation du simulateur pour identifier l'ensemble recherché avec le moins d'appels au code.

Contents

1.1	Modélisation par processus gaussien	7
1.1.1	Définitions et propriétés de base	7
1.1.2	Processus gaussiens	8
1.1.3	Modélisation par processus gaussiens	10
1.1.4	Synthèse et objectif	11
1.2	Estimation et quantification de l'incertitude associée à un ensemble d'excursion	12
1.2.1	Espérance d'ensembles aléatoires fermés	12
1.2.2	Estimation d'un ensemble d'excursion et l'incertitude associée	15
1.3	Stratégie de réduction séquentielle d'incertitudes	16
1.3.1	La stratégie SUR	16
1.3.2	Exemple d'application	18

Les simulateurs numériques, ou codes de calcul, permettent de modéliser un phénomène physique complexe. Lorsque l'utilisateur n'a accès qu'à des couples entrées/sorties du simulateur, on parle de boîte noire. Plus précisément, l'utilisateur n'a accès qu'aux couples $(\mathbf{x}_i, g(\mathbf{x}_i))$ où

$$g : \begin{cases} \mathbb{X} \subset \mathbb{R}^p & \rightarrow \mathbb{R} \\ \mathbf{x} & \mapsto g(\mathbf{x}), \end{cases} \quad (1.1)$$

pour n valeurs de paramètres d'entrée $\mathbf{x}_1, \dots, \mathbf{x}_n$. Un large éventail de problèmes peut être formulé à partir de simulateurs de type boîte noire. Par exemple, on peut être intéressé par les valeurs d'entrée pour lesquelles une sortie atteint son maximum ou son minimum (problèmes d'optimisation). Alternativement, nous pourrions rechercher l'ensemble du domaine de définition des entrées pour lesquelles la sortie est dans un ensemble d'intérêt (problème

d'inversion). Cela se traduit par l'identification de l'ensemble suivant

$$\Gamma^* = \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) \leq c\}, \quad (1.2)$$

où $c \in \mathbb{R}$ est un seuil prédéfini. Les simulateurs numériques ont souvent des temps d'exécution de plusieurs minutes voire plusieurs heures, ce qui exclut l'application de techniques gourmandes en évaluations. Par exemple des méthodes de recherche aléatoire (recuit simulé, stratégies évolutionnaires, etc.), très gourmands en évaluations, sont écartées. Il est donc primordial d'apporter une solution raisonnable au problème considéré en utilisant peu d'évaluations du simulateur. Cela implique de choisir, selon une stratégie bien définie, les différents points où évaluer le simulateur.

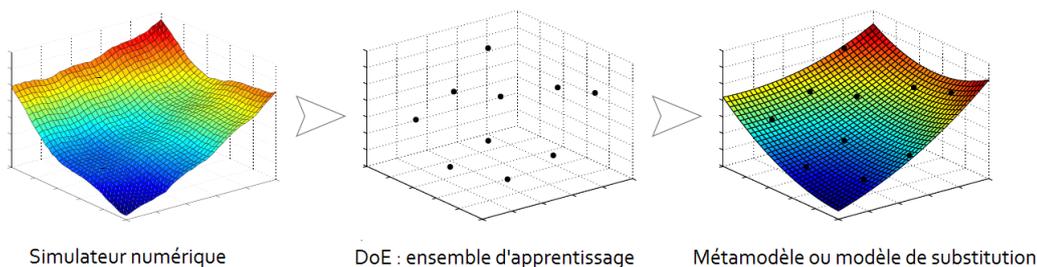


FIGURE 1.1: Construction d'un modèle de substitution à partir d'échantillons d'un simulateur numérique coûteux.

Une approche très courante pour élaborer des stratégies économiques d'évaluation du simulateur repose sur l'utilisation des métamodèles aussi appelés modèles de substitution. Un métamodèle est une représentation déterministe ou probabiliste de la relation entre les entrées et la sortie du simulateur construite à partir d'un nombre limité d'évaluations. Plus précisément, dans le cadre de la thèse, la classe de métamodèles retenue est la régression par processus gaussiens

(voir, e.g, [Matheron, 1973, Rasmussen and Williams, 2006, Chiles and Delfiner, 1999]) dont l'utilisation est très répandue et dont l'application à l'optimisation ou à l'inversion a déjà fait ses preuves. La métamodélisation par processus gaussien, également appelée krigeage, est une technique issue de la géostatistique. Son principe est le suivant : on suppose que la fonction à approcher g est la réalisation d'un processus gaussien $(Y_{\mathbf{x}})_{\mathbf{x} \in \mathbb{X}}$ conditionné par les points évalués. Cette hypothèse a notamment pour intérêt de fournir, en plus d'une prédiction de la fonction inconnue g , l'erreur de prédiction commise par ce métamodèle en tout point de l'espace $\mathbf{x} \in \mathbb{X}$. Notons que le processus $(Y_{\mathbf{x}})_{\mathbf{x} \in \mathbb{X}}$ peut être considéré autre que gaussien (voir, e.g, [Palacios and Steel, 2006]), mais cette hypothèse facilite grandement la définition du processus lorsque de nouvelles observations sont assimilées. En effet, un processus gaussien conditionné par de nouvelles observations demeure gaussien et les calculs en sont facilités.

Dans le cadre de la thèse et parmi l'éventail des problèmes possibles, on s'intéresse au problème d'inversion, ou l'identification d'un ensemble d'excursion, à partir d'un ensemble d'évaluations. Ce problème fait l'objet d'une littérature abondante. Des méthodes d'estimation non paramétrique ont été proposées dans [Devroye and Wise, 1980] et [Baillo et al., 2000], ces dernières fournissent des estimateurs convergents, sous certaines hypothèses sur l'ensemble recherché. Il existe aussi une technique directe qui consiste à prendre un estimateur *plug-in* ([Molchanov, 1998, Cuevas et al., 2006]). Dans un tel contexte, cette approche consiste à remplacer la fonction inconnue g par un estimateur approprié, et de considérer l'ensemble d'excursion de l'estimateur comme estimation de l'ensemble recherché ([Ranjan et al., 2008]). Bien que l'approche *plug-in* soit pratique, elle ne fournit pas une quantification des incertitudes

associées à l'estimateur de l'ensemble. La stratégie expliquée dans ce chapitre vient pallier ce problème.

L'objectif de ce chapitre est de définir une stratégie adaptative d'ajout de simulations afin de résoudre un problème d'inversion. On propose d'adopter la stratégie de [Chevalier et al., 2014a] qui repose sur des techniques de modélisation par processus gaussien et sur la théorie des ensembles aléatoires fermés. En guise de préliminaire, les deux premières sections sont dédiées au rappel de quelques fondamentaux sur les variables et processus aléatoires, la modélisation par processus gaussien ainsi que la théorie des ensembles aléatoires fermés. La troisième section est dédiée à présenter la stratégie retenue qui combine les notions introduites auparavant pour l'identification d'un ensemble d'excursion.

1.1 Modélisation par processus gaussien

1.1.1 Définitions et propriétés de base

On considère un espace probabilisé $(\Omega, \mathcal{F}, \mathbb{P})$ et un espace d'état mesuré $(S, \mathcal{B}(S))$. Cet espace peut éventuellement être \mathbb{R}^p ($p \in \mathbb{N}^*$) muni de sa tribu borélienne notée $\mathcal{B}(\mathbb{R}^p)$.

On appelle *variable aléatoire* toute application mesurable $Y : (\Omega, \mathcal{F}, \mathbb{P}) \rightarrow (S, \mathcal{B}(S))$. Lorsque $(S, \mathcal{B}(S)) = (\mathbb{R}^p, \mathcal{B}(\mathbb{R}^p))$, on parle de *vecteur aléatoire réel* (V.a.r.), ou encore de *variable aléatoire réelle* (v.a.r.) dans le cas où $p = 1$. L'espérance d'une v.a.r. Y , notée $\mathbb{E}[Y]$, est l'intégrale de Y par rapport à la mesure de probabilité \mathbb{P} :

$$\mathbb{E}[Y] = \int_{\Omega} Y(w) d\mathbb{P}(w). \quad (1.3)$$

Pour $s \geq 1$, on note $L^s(\mathbb{P})$ l'espace des v.a.r. telles que $\mathbb{E}[|Y^s|] < \infty$, et les nombres $\mathbb{E}[|Y^s|]$ sont appelés *les moments d'ordre s* de la v.a.r. Y . Dans le cas où $s = 2$, $L^2(\mathbb{P})$ est un espace de Hilbert, espace des variables aléatoires de carré intégrables (i.e. $\mathbb{E}[|Y^2|] < \infty$). La covariance entre $Y_1, Y_2 \in L^2(\mathbb{P})$ est définie par :

$$\text{Cov}(Y_1, Y_2) = \mathbb{E}[Y_1 Y_2] - \mathbb{E}[Y_1] \mathbb{E}[Y_2] \quad (1.4)$$

Pour un vecteur aléatoire $\mathbf{Y} \in \mathbb{R}^p$, on appelle matrice de covariance de \mathbf{Y} , notée $\text{Cov}(\mathbf{Y})$, la matrice symétrique réelle de taille $p \times p$ de terme général $\text{Cov}(Y_i, Y_j)$. Cette matrice est clairement symétrique et de plus on a :

$$\begin{aligned} \forall a = (a_1, \dots, a_p)^\top \in \mathbb{R}^p, \quad a^\top \text{Cov}(\mathbf{Y}) a &= \sum_{i=1}^p \sum_{j=1}^p a_i a_j \text{Cov}(Y_i, Y_j) \\ &= \text{var}\left(\sum_{i=1}^p a_i Y_i\right) \geq 0, \end{aligned} \quad (1.5)$$

ce qui signifie que la matrice $\text{Cov}(\mathbf{Y})$ est une matrice symétrique positive.

Les vecteurs gaussiens constituent une classe particulièrement intéressante de vecteurs aléatoires. Rappelons qu'une v.a.r. est dite *gaussienne standard*, ou encore de loi $\mathcal{N}(0, 1)$, lorsqu'elle admet pour densité de probabilité la fonction $\varphi_{\mathcal{N}(0,1)}(y) = \frac{1}{\sqrt{2\pi}} e^{-y^2/2}$. Une v.a.r. est dite *gaussienne* s'il existe $m \in \mathbb{R}$, $\sigma \in [0, \infty[$, et $N \sim \mathcal{N}(0, 1)$ tels que $Y = m + \sigma N$. On a alors $Y \in L^2$ avec $\mathbb{E}[Y] = m$ et $\text{Var}(Y) = \sigma^2$ et on notera $Y \sim \mathcal{N}(m, \sigma^2)$. On appelle *vecteur gaussien standard* un V.a.r. \mathbf{N} dont les composantes sont indépendantes et de loi $\mathcal{N}(0, 1)$. De manière plus générale, on dit que \mathbf{Y} est un vecteur gaussien si :

$$\mathbf{Y} = \mathbf{m} + \mathbf{A}\mathbf{N}, \quad (1.6)$$

où $\mathbf{m} \in \mathbb{R}^p$ et $A \in \mathbb{R}^{p \times p}$. Dans ce cas, $\mathbb{E}[\mathbf{Y}] = \mathbf{m}$ et il vient alors par calcul direct que $\Sigma := \text{Cov}(\mathbf{Y}) = AA^\top$, et on note $\mathbf{Y} \sim \mathcal{N}(\mathbf{m}, \Sigma)$. Dans le cas où Σ est inversible, on a la densité de probabilité gaussienne multivariée suivante :

$$\varphi_{\mathbf{m}, \Sigma}(\mathbf{y}) = \frac{1}{|\Sigma|^{1/2} (2\pi)^{d/2}} e^{-\frac{1}{2}(\mathbf{y}-\mathbf{m})^\top \Sigma^{-1}(\mathbf{y}-\mathbf{m})}, \quad \mathbf{y} \in \mathbb{R}^p, \quad (1.7)$$

où $|\Sigma|$ est le déterminant de la matrice Σ .

Une particularité remarquable des vecteurs gaussiens concerne le conditionnement d'un vecteur gaussien par rapport à un autre vecteur gaussien. Si $\mathbf{Y} = (\mathbf{Y}_1, \mathbf{Y}_2)^\top$ est un vecteur gaussien ([Petersen et al., 2008]) tel que :

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}\right),$$

et Σ_{22} est inversible, alors pour toute réalisation \mathbf{y}_2 du vecteur gaussien \mathbf{Y}_2 , on a :

$$\mathbf{Y}_1 | \mathbf{Y}_2 = \mathbf{y}_2 \sim \mathcal{N}(\mathbf{m}_1 + \Sigma_{12} \Sigma_{22}^{-1}(\mathbf{y}_2 - \mathbf{m}_2), \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}). \quad (1.8)$$

1.1.2 Processus gaussiens

Soit $(\Omega, \mathcal{F}, \mathbb{P})$ un espace probabilisé, $(S, \mathcal{B}(S)) = (\mathbb{R}, \mathcal{B}(\mathbb{R}))$ et $\mathbb{X} \subset \mathbb{R}^p$, où $p \geq 1$. $(Y_{\mathbf{x}})_{\mathbf{x} \in \mathbb{X}}$ est appelé *processus gaussien* ou encore *champs gaussien* lorsque toutes ses loi finies-dimensionnelles sont gaussiennes, i.e., $\forall k \geq 1, \forall \mathbf{x}_1, \dots, \mathbf{x}_k \in \mathbb{X}$, le V.a.r. $(Y_{\mathbf{x}_1}, \dots, Y_{\mathbf{x}_k})$ est gaussien. Cela implique que quel que soit $\mathbf{x} \in \mathbb{X}$, la v.a.r. $Y_{\mathbf{x}}$ suit une loi gaussienne. Il en découle que le processus $(Y_{\mathbf{x}})_{\mathbf{x} \in \mathbb{X}}$ admet des moments d'ordre 2, notons les comme suit :

$$\begin{aligned} m(\mathbf{x}) &:= \mathbb{E}[Y_{\mathbf{x}}] \\ k(\mathbf{x}, \mathbf{x}') &:= \text{Cov}(Y_{\mathbf{x}}, Y_{\mathbf{x}'}), \quad (\mathbf{x}, \mathbf{x}') \in \mathbb{X}^2 \end{aligned} \quad (1.9)$$

La moyenne, appelée aussi *tendance*, notée m et le noyau de covariance, noté k , déterminent complètement la loi du processus, dans la suite de ce chapitre, on utilisera la notation $Y \sim \mathcal{PG}(m, k)$.

Dire qu'un processus est stationnaire à l'ordre 1 signifie que m est constante, i.e., $\forall \mathbf{x} \in \mathbb{X}, m(\mathbf{x}) = c$. La stationnarité à l'ordre deux s'obtient par l'existence et la stationnarité du noyau de covariance k et cela signifie que quel que soit $(\mathbf{x}, \mathbf{x}') \in \mathbb{X}^2$, $k(\mathbf{x}, \mathbf{x}')$ ne dépend que de $\mathbf{h} = \mathbf{x} - \mathbf{x}'$, dans ce cas, on parle de processus fortement stationnaire. Cela illustre le rôle prépondérant que joue le noyau de covariance k dans la définition d'un processus gaussien. Donnons un panel de fonctions couramment utilisées pour le noyau de covariance.

Choix du noyau

Nous avons vu que toute fonction définie positive peut être prise comme fonction de covariance, sauf que la condition de positivité est difficilement vérifiable. Ainsi on a souvent recours à des fonctions de covariance bien connues. Dans le cas où $\mathbb{X} = \mathbb{R}$, l'exemple classique d'une telle fonction est le noyau exponentiel de degré 2 stationnaire suivant :

$$k(x, x') = \sigma^2 e^{-\frac{(x-x')^2}{2\theta^2}}, \quad (1.10)$$

où σ est un paramètre d'échelle et θ un paramètre de portée. Ce choix de noyau conduit à des fonctions de classe \mathcal{C}^∞ pour les trajectoires du processus associé. Pour le noyau exponentiel défini par

$$k(x, x') = \sigma^2 e^{-\frac{|x-x'|}{2\theta}}, \quad (1.11)$$

les trajectoires du processus ne sont pas dérivables mais seulement continues (de classe \mathcal{C}^0). Parmi les noyaux les plus couramment utilisés pour la modélisation par processus gaussien, on trouve aussi les noyaux de Matern. La Table 1.1 classe par ordre de régularité les noyaux usuels.

Nom	Expression	Classe
Gaussien	$\sigma^2 \exp\left(-\frac{(x-x')^2}{2\theta^2}\right)$	\mathcal{C}^∞
Matérn 5/2	$\sigma^2 \left(1 + \frac{\sqrt{5} x-x' }{\theta} + \frac{5(x-x')^2}{3\theta^2}\right) \exp\left(\frac{-\sqrt{5} x-x' }{\theta}\right)$	\mathcal{C}^2
Matérn 3/2	$\sigma^2 \left(1 + \frac{\sqrt{3} x-x' }{\theta}\right) \exp\left(\frac{-\sqrt{3} x-x' }{\theta}\right)$	\mathcal{C}^1
Exponentiel	$\sigma^2 \exp\left(-\frac{ x-x' }{\theta}\right)$	\mathcal{C}^0

TABLE 1.1: Noyaux de covariance usuels.

La stationnarité est la particularité de ces noyaux puisque leur expression dépend uniquement de $|x - x'|$. De plus, pour chacun de ces noyaux, $k(x, x')$ est strictement décroissante en $|x - x'|$. Il en découle que la connaissance d'un processus Y en un point x aura une influence locale sur la connaissance de Y . Sur la figure 1.2, on représente quelques noyaux classiques et des réalisations des processus gaussiens associés.

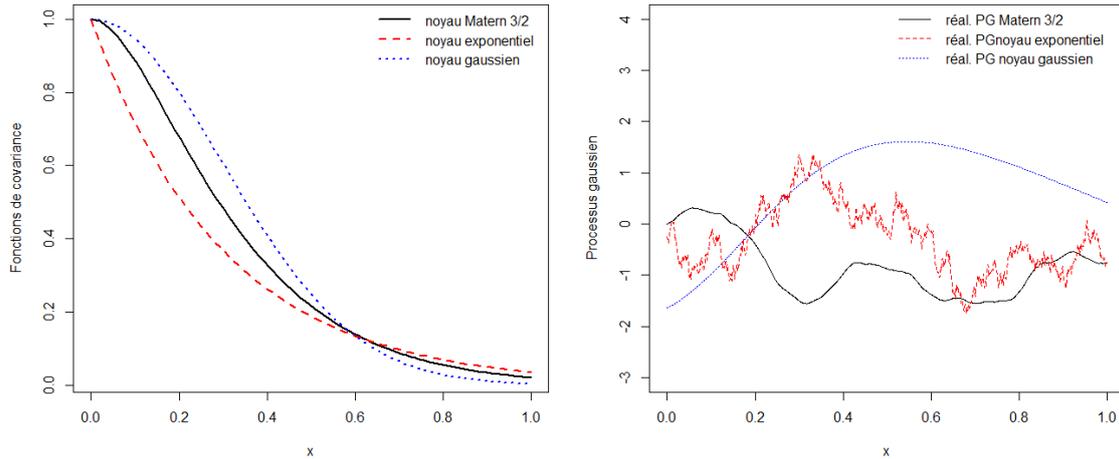


FIGURE 1.2: Fonctions de covariance classiques (à gauche) et des réalisations des processus gaussiens associés (à droite). Les paramètres sont fixés à $(\sigma^2, \theta) = (1, 0.3)$.

Dans le cas où les variables d'entrée sont des vecteurs de \mathbb{R}^d , les noyaux usuels sont construits comme le produit tensoriel de noyaux unidimensionnels k_i . Pour $\mathbf{x} = (x_1, \dots, x_p)^\top \in \mathbb{R}^p$ et $\mathbf{x}' = (x'_1, \dots, x'_p)^\top \in \mathbb{R}^p$,

$$k(\mathbf{x}, \mathbf{x}') = \prod_{i=1}^p k_i(x_i, x'_i). \quad (1.12)$$

Dans la section 1.1.3, on verra le rôle prépondérant que joue le noyau de covariance dans la modélisation par processus gaussiens. Pour le choix du noyau optimal pour traiter un problème donné, il existe des techniques de validation croisée mais elles sont coûteuses. Si on a des a priori sur la régularité de g ou sur certaines de ses propriétés, on choisira un noyau en conséquence.

1.1.3 Modélisation par processus gaussiens

Nous considérons dans cette section que la fonction à modéliser g est définie sur un compact $\mathbb{X} \subset \mathbb{R}^p$, continue et qu'elle est à valeurs réelles. Comme il a été dit dans l'introduction de ce chapitre, on suppose que l'on dispose de n points $\mathcal{X}_n = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{X}^n$ pour lesquels la valeur de g est connue, et on note par $\mathbf{g}_{\mathcal{X}_n} = (g(\mathbf{x}_1), \dots, g(\mathbf{x}_n)) \in \mathbb{R}^n$ les valeurs de la fonction g aux points \mathcal{X}_n .

L'hypothèse fondamentale de la modélisation par processus gaussien est que la fonction g est une trajectoire d'un processus gaussien Y indexé par \mathbb{X} . L'objectif va donc être de prendre en compte dans la modélisation de g l'information partielle fournie par les évaluations $\mathbf{g}_{\mathcal{X}_n}$. La loi du processus Y est caractérisée par ses deux premiers moments (cf. section 1.1.2), on peut s'intéresser à la distribution du processus Y sachant qu'il prend les valeurs de la fonction g aux points du plan $\mathcal{X}_n : Y_{\mathcal{X}_n} = (Y_{\mathbf{x}_1}, \dots, Y_{\mathbf{x}_n}) = \mathbf{g}_{\mathcal{X}_n}$. Le processus conditionnel $Y|Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}$ est gaussien, de plus la moyenne et la covariance conditionnelles sont obtenues directement en utilisant la proposition 1.8 :

$$\begin{aligned} Y|Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n} &\sim \mathcal{GP}(m_n, k_n) \quad \text{où} \\ m_n(\mathbf{x}) &= m(\mathbf{x}) + \mathbf{k}_n(\mathbf{x})^\top \mathbf{K}_n^{-1}(\mathbf{g}_{\mathcal{X}_n} - \mathbf{m}(\mathcal{X}_n)) \\ k_n(\mathbf{x}, \mathbf{x}') &= k(\mathbf{x}, \mathbf{x}') - \mathbf{k}_n(\mathbf{x})^\top \mathbf{K}_n^{-1} \mathbf{k}_n(\mathbf{x}') \end{aligned} \quad (1.13)$$

avec $\mathbf{k}_n(\mathbf{x}) = (k(\mathbf{x}, \mathbf{x}_1), \dots, k(\mathbf{x}, \mathbf{x}_n))$, $\mathbf{k}_n(\mathbf{x}') = (k(\mathbf{x}', \mathbf{x}_1), \dots, k(\mathbf{x}', \mathbf{x}_n))$ et \mathbf{K}_n la matrice de covariance aux points \mathcal{X}_n , i.e., $\mathbf{K}_n = (k(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i, j \leq n}$. La quantité m_n interpole les valeurs de la fonction g aux points \mathcal{X}_n . De plus $m_n(\mathbf{x})$ est le meilleur prédicteur (au sens de l'erreur quadratique moyenne), linéaire en $\mathbf{g}_{\mathcal{X}_n}$ sans biais de $Y_{\mathbf{x}}$. La variance conditionnelle $k_n(\mathbf{x}, \mathbf{x})$ permet de calculer un intervalle de prédiction en chaque point \mathbf{x} .

Estimation des paramètres

Comme le montrent les équations (1.13), l'expression de m_n dépend de la structure de covariance donnée par k_n et donc des paramètres qui la définissent. Il est donc important de bien estimer ces derniers. En pratique, on trouve deux grandes méthodes d'estimation dans la littérature. Une première méthode, par maximum de vraisemblance, consiste à déterminer les paramètres maximisant la vraisemblance, définie comme étant la densité de probabilité des observations. La deuxième méthode, par validation croisée, a pour objectif de quantifier la capacité de généralisation d'un modèle en se basant uniquement sur les données d'apprentissage (se référer à [Ginsbourger et al., 2009, Santner et al., 2013] pour plus de détails). Dans la suite de ce manuscrit, nous nous limiterons à la première approche.

Exemple d'application

Afin d'illustrer la modélisation par processus gaussien, nous utilisons un des cas présentés dans l'article de [Roustant et al., 2012]. On définit la fonction g comme suit

$$g : \begin{cases} [0, 1] & \rightarrow \mathbb{R} \\ x & \mapsto (\sin(10x)/(1+x) + 2 \cos(5x)x^3 + 0.841)/1.6, \end{cases} \quad (1.14)$$

la modélisation correspondante est donnée sur la figure 1.3. Les points rouges représentent l'ensemble d'apprentissage utilisé pour modéliser la réponse. Le noyau de covariance retenue est de type Matérn 5/2 (voir table 1.1) et les paramètres (θ, σ) de ce dernier sont estimés par maximum de vraisemblance. La prédiction obtenue est donnée sur la figure de droite 1.3 et l'incertitude associée à cet estimateur est donnée par la bande de confiance à 95%. On remarque bien sur cet exemple que la prédiction est interpolante.

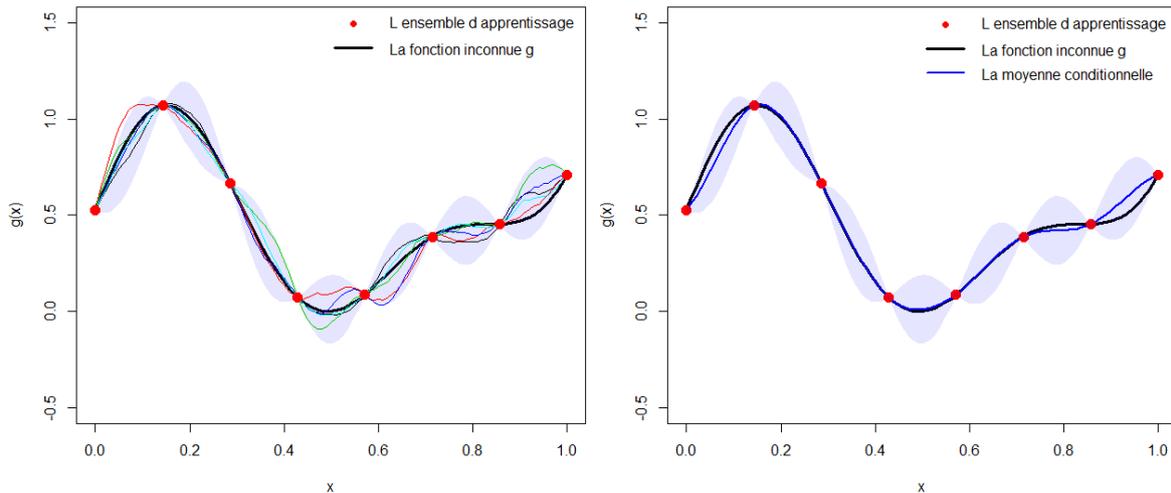


FIGURE 1.3: A gauche : la fonction inconnue g (courbe noire) et cinq réalisations du processus gaussien conditionné défini par les équations (1.13), intervalle de confiance à 95% (bande en bleu clair). A droite : Approximation de la fonction g (en trait noir) par la moyenne conditionnelle m_n (en trait bleu) obtenue à partir de $n = 8$ points d'expériences, intervalle de confiance à 95% (bande en bleu clair).

1.1.4 Synthèse et objectif

Nous avons présenté dans cette section une méthode probabiliste pour modéliser une fonction inconnue supposée continue, g , que l'on a observée en un nombre limité de points. On a donc cherché à mettre à profit ces connaissances pour construire un modèle qui approche au mieux g . On a vu que la modélisation par processus gaussien fournit un estimateur de la fonction g donné par la moyenne conditionnelle m_n ainsi que l'incertitude associée à cet estimateur fournie par la variance conditionnelle k_n . On rappelle qu'on s'intéresse à la résolution d'un problème d'inversion. En d'autres termes, on cherche à identifier l'ensemble

$$\Gamma^* = \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) \leq c\}, \quad (1.15)$$

où $c \in \mathbb{R}$ est un seuil prédéfini. Comme la fonction g est coûteuse en temps de calcul, le but est d'estimer l'ensemble Γ^* avec le moins d'appels possible au simulateur g . Toujours dans la même configuration où la valeur de g est connue en certains points du plan \mathcal{X}_n , il est raisonnable d'exploiter cette information dans l'estimation de Γ^* . Dans un cadre bayésien (cf. section 1.1.3), et dans la logique de l'approche *plug-in* introduite précédemment, on peut approcher la fonction g par la moyenne conditionnelle m_n et ainsi estimer l'ensemble Γ^* par

$$\Gamma_{\text{plug-in}} = \{\mathbf{x} \in \mathbb{X}, m_n(\mathbf{x}) \leq c\}, \quad (1.16)$$

La figure 1.4 montre l'estimateur *plug-in* pour l'exemple introduit précédemment où le seuil est fixé à $c = 0.25$. Bien que cet estimateur soit facile à implémenter, cette technique ne fournit pas une quantification de l'incertitude. Le but de la section 1.2 est de pallier ce problème et donc de fournir un estimateur de l'ensemble Γ^* ainsi qu'une quantification de l'incertitude associée.

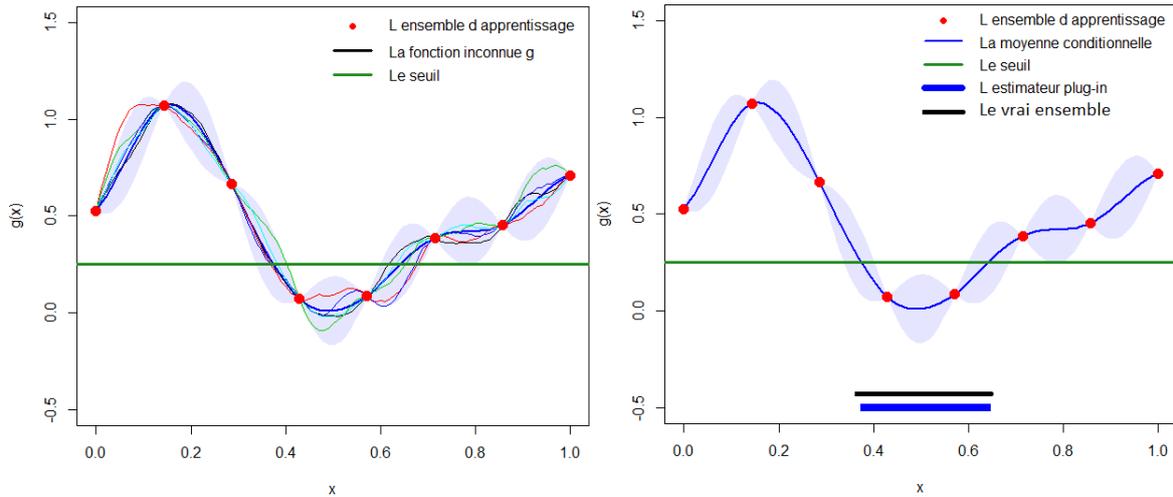


FIGURE 1.4: A gauche : 5 réalisations du processus gaussien conditionnel avec $n = 7$. A droite : la moyenne conditionnelle m_n (trait bleu), la bande de confiance (en bleu clair) et l'estimateur $\Gamma_{\text{plug-in}}$.

1.2 Estimation et quantification de l'incertitude associée à un ensemble d'excursion

Dans cette partie, deux problèmes issus de la géométrie aléatoire seront abordés. Dans la Section 1.2.1, nous considérons un ensemble aléatoire fermé Γ et étudions l'espérance de cet ensemble aléatoire, au sens de Vorob'ev. Puis dans la section 1.2.2, nous introduisons un estimateur numériquement implémentable de l'ensemble d'excursion Γ^* , en nous inspirant des travaux de [Chevalier et al., 2014a].

1.2.1 Espérance d'ensembles aléatoires fermés

La définition de l'espérance d'un ensemble aléatoire Γ est une question naturelle. Elle peut émerger dans divers problèmes tels que ceux présentés dans [Tran, 2014]. Soit un compact $\mathbb{X} \subset \mathbb{R}^p$ et Γ un ensemble aléatoire fermé de \mathbb{X} . Ce fermé aléatoire est une fonction de l'espace de probabilité $(\Omega, \mathcal{F}, \mathbb{P})$ dans l'ensemble \mathcal{C} des compacts de \mathbb{X} mesurable au sens où pour tout compact $C \in \mathcal{C}$, $\{w \in \Omega, \Gamma(w) \cap C \neq \emptyset\} \in \mathcal{F}$ (voir [Molchanov, 2006], chapitre 1.1).

Des définitions de l'espérance de Γ abondent dans la littérature. Ici, on s'intéresse à une définition s'appuyant sur le principe de *linéarisation*. Pour une revue plus approfondie des autres méthodes, on renvoie le lecteur à l'ouvrage [Molchanov, 2006].

La *notion de linéarisation* consiste à linéariser l'espace \mathcal{C} où l'ensemble aléatoire Γ est défini et où la notion d'espérance n'est pas triviale, pour se ramener à un espace \mathbb{Y} où l'espérance d'une variable aléatoire γ_Γ est facile à calculer. Une fois cette espérance calculée, l'étape suivante consiste à trouver l'ensemble $C \in \mathcal{C}$ pour lequel γ_C soit suffisamment proche de $\mathbb{E}[\gamma_\Gamma]$. Cette étape fait appel à une pseudo-distance δ qui évalue l'écart entre $\mathbb{E}[\gamma_\Gamma]$ et γ_C . Ainsi l'espérance de Γ est définie par

$$\mathbb{E}[\Gamma] \in \arg \min_{C \in \mathcal{C}} \delta(\mathbb{E}[\gamma_\Gamma], \gamma_C). \quad (1.17)$$

Le problème d'optimisation (1.17) est difficile en pratique, et au lieu de le résoudre dans tout l'espace \mathcal{C} , on restreint la recherche à une famille paramétrique $\mathcal{C}_\alpha \subset \mathcal{C}$.

La notion de *linéarisation* peut être illustrée par le diagramme suivant :

$$\begin{array}{ccc}
 \Gamma \in \mathcal{C} & \xrightarrow{\gamma} & \gamma_\Gamma \in \mathbb{Y} \\
 & & \downarrow \\
 \mathbb{E}[\Gamma] \in \mathcal{C} & \xleftarrow{\text{proximité}} & \mathbb{E}[\gamma_\Gamma] \in \mathbb{Y}
 \end{array}$$

Cette technique dépend alors des ingrédients suivants :

- un espace \mathbb{Y} ,
 - une fonction $\gamma : \mathcal{C} \rightarrow \mathbb{Y}$,
 - une pseudo-distance δ entre deux éléments de \mathbb{Y} (*proximité*),
 - une famille paramétrique d'ensembles fermés \mathcal{C}_α candidats pour l'espérance $\mathbb{E}[\Gamma]$,
- ainsi en définissant proprement ces éléments, on obtient différentes définitions de l'espérance d'un ensemble aléatoire fermé.

Ici, on s'intéresse à l'espérance de Vorob'ev qui reflète davantage une notion d'occupation moyenne et d'ensembles de niveau. L'espérance de Vorob'ev est une des applications directes du principe de *linéarisation*. Pour plus de détails, on renvoie le lecteur à l'ouvrage [Molchanov, 2006].

Soit la fonction de couverture de Γ , notée p

$$p : \begin{cases} \mathbb{X} & \rightarrow [0, 1] \\ \mathbf{x} & \mapsto \mathbb{P}(\mathbf{x} \in \Gamma). \end{cases} \quad (1.18)$$

On définit la famille paramétrique $\mathcal{C}_\alpha = \{Q_\alpha : \alpha \in [0, 1]\}$, où

$$Q_\alpha = \{\mathbf{x} \in \mathbb{X} : p(\mathbf{x}) = \mathbb{P}(\mathbf{x} \in \Gamma) \geq \alpha\}. \quad (1.19)$$

Cette famille paramétrique d'ensembles est aussi appelée *quantiles de Vorob'ev*.

Définition 1 (Espérance de Vorob'ev ([Vorob'ev, 1984])). L'espérance de Vorob'ev d'un ensemble aléatoire fermé est l'ensemble $Q_{\alpha^*} = \{\mathbf{x} \in \mathbb{X} : p(\mathbf{x}) \geq \alpha^*\}$ tel que α^* est déterminé à partir de l'équation $\mathbb{E}[\mu(\Gamma)] = \mu(Q_{\alpha^*})$ si cette dernière admet une solution ou à partir de la condition

$$\mu(Q_\alpha) < \mathbb{E}[\mu(\Gamma)] \leq \mu(Q_{\alpha^*}), \quad (1.20)$$

pour tout $\alpha > \alpha^*$ et μ la mesure de Lebesgue.

L'unicité de l'espérance de Vorob'ev n'est assurée que si la fonction $\alpha \mapsto \mu(Q_\alpha)$ est continue en α^* . Dans le cadre gaussien de notre étude, cette condition est vérifiée par la continuité du processus gaussien étudié et qui définit Q_α , les *quantiles de Vorob'ev* (cf. section 1.2.2).

D'un point de vue calculatoire, le calcul de l'espérance de Vorob'ev est facile en pratique si l'espérance du volume de Γ est facile à calculer. Dans le cas gaussien, ce calcul sera présenté dans la section suivante 1.2.2.

A ce stade, on introduit une notion de distance entre ensembles aléatoires, cette distance nous sera utile dans la suite. Dans la littérature, pour les ensembles aléatoires fermés, différentes distances existent. Cette distance peut être utilisée pour quantifier l'éloignement entre les *quantiles de Vorob'ev*, Q_α , et l'ensemble aléatoire correspondant Γ .

Définition 2 (Distance moyenne pour ensembles aléatoires fermés). Soit deux ensembles aléatoires fermés $\Gamma_1, \Gamma_2 \subset \mathbb{X}$. La distance moyenne entre Γ_1 et Γ_2 est donnée par

$$d_\mu(\Gamma_1, \Gamma_2) = \mathbb{E}[\mu(\Gamma_1 \Delta \Gamma_2)], \quad (1.21)$$

où $\Gamma_1 \Delta \Gamma_2 = (\Gamma_1 \setminus \Gamma_2) \cup (\Gamma_2 \setminus \Gamma_1)$.

Cette distance quantifie l'écart entre deux ensembles en mesurant la partie où ces derniers ne se chevauchent pas. Si les deux ensembles sont égaux presque sûrement alors $d_\mu(\Gamma_1, \Gamma_2) = 0$ et si $\Gamma_1 \cap \Gamma_2 = \emptyset$ alors $d_\mu(\Gamma_1, \Gamma_2) = \mathbb{E}[\mu(\Gamma_1)] + \mathbb{E}[\mu(\Gamma_2)]$. Cette distance est très utile dans les problèmes d'estimation d'ensembles. Le résultat suivant montre que, parmi tous les ensembles de volume égale au volume moyen, l'espérance de Vorob'ev est l'ensemble qui minimise cette distance.

Théorème 1 (Le problème de minimisation ([Molchanov, 2006] ; Théorème 2.3)). Si $\alpha^* \geq 1/2$ alors pour tout ensemble mesurable M vérifiant $\mu(M) = \mathbb{E}[\mu(\Gamma)]$, on a

$$\mathbb{E}[\mu(\Gamma \Delta Q_{\alpha^*})] \leq \mathbb{E}[\mu(\Gamma \Delta M)]. \quad (1.22)$$

Démonstration 1. Soit (Ω, \mathcal{F}, P) un espace probabilisé, soit un ensemble M tel que $\mu(M) = \mathbb{E}[\mu(\Gamma)]$, on pose $F = Q_{\alpha^*}$. Pour tout $\omega \in \Omega$, on a :

$$\begin{aligned} \mu(M \Delta \Gamma(\omega)) - \mu(F \Delta \Gamma(\omega)) &= \mu[(M \Delta \Gamma(\omega)) \setminus (F \Delta \Gamma(\omega))] - \mu[(F \Delta \Gamma(\omega)) \setminus (M \Delta \Gamma(\omega))] \\ &= \mu[(M \setminus F) \cap \Gamma(\omega)^c] + \mu[(F \setminus M) \cap \Gamma(\omega)] - \mu[(F \setminus M) \cap \Gamma(\omega)^c] \\ &\quad - \mu[(M \setminus F) \cap \Gamma(\omega)] \\ &= \mu(M \setminus F) - \mu(F \setminus M) + 2(\mu[(F \setminus M) \cap \Gamma(\omega)] - \mu[(M \setminus F) \cap \Gamma(\omega)]) \\ &= \mu(M) - \mu(F) + 2(\mu[(F \setminus M) \cap \Gamma(\omega)] - \mu[(M \setminus F) \cap \Gamma(\omega)]) \end{aligned}$$

On introduit l'espérance des deux cotés, on obtient :

$$\begin{aligned} \mathbb{E}[\mu(M \Delta \Gamma) - \mu(F \Delta \Gamma)] &= \mu(M) - \mu(F) + \mathbb{E}[2(\mu[(F \setminus M) \cap \Gamma] - \mu[(M \setminus F) \cap \Gamma])] \\ &= \mu(M) - \mu(F) + 2 \int_{F \setminus M} p(\mathbf{x}) \mu(d\mathbf{x}) - 2 \int_{M \setminus F} p(\mathbf{x}) \mu(d\mathbf{x}) \end{aligned}$$

Puisque $\forall \mathbf{x} \in F \setminus M$, $p(\mathbf{x}) \geq \alpha^*$ et que, $\forall \mathbf{x} \in M \setminus F$, $p(\mathbf{x}) < \alpha^*$, alors :

$$\begin{aligned} \mu(M) - \mu(F) + 2 \left[\int_{F \setminus M} p(\mathbf{x}) \mu(d\mathbf{x}) - \int_{M \setminus F} p(\mathbf{x}) \mu(d\mathbf{x}) \right] &\geq \mu(M) - \mu(F) + 2\alpha^* [\mu(F \setminus M) - \mu(M \setminus F)] \\ &= (2\alpha^* - 1) [\mu(F) - \mu(M)] \\ &\geq 0 \end{aligned}$$

Ce résultat intéressant nous conforte dans le choix de la famille des *quantiles de Vorob'ev* comme famille d'ensembles fermés candidats pour définir l'espérance. La quantité $\mathbb{E}[\mu(\Gamma \Delta Q_{\alpha^*})]$ est appelée la *déviatoin de Vorob'ev*.

Dans la Section 1.2.2, on utilise cette grandeur pour mesurer la variabilité associée à un estimateur de Γ . Puis pour conclure ce chapitre, on s'appuie sur l'incertitude induite par cette distance afin de définir une stratégie séquentielle d'évaluation du simulateur pour réduire cette incertitude et ainsi identifier l'ensemble d'excursion souhaité.

1.2.2 Estimation d'un ensemble d'excursion et l'incertitude associée

Dans un contexte de modélisation par processus gaussien, la fonction inconnue g , supposée continue, est vue comme une réalisation d'un processus gaussien de moyenne m et de noyau de covariance k , noté $Y \sim \mathcal{GP}(m, k)$. Ainsi, l'ensemble d'excursion à identifier $\Gamma^* = \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) \leq c\}$, où $c \in \mathbb{R}$ est un seuil prédéfini, peut être vu comme une réalisation d'un ensemble aléatoire fermé $\Gamma = \{\mathbf{x} \in \mathbb{X}, Y_{\mathbf{x}} \leq c\}$. Pour la modélisation de g , on s'est appuyé sur l'ensemble d'apprentissage $(\mathcal{X}_n, \mathbf{g}_{\mathcal{X}_n})$ ainsi que sur les deux premiers moments pour construire un estimateur de g donné par m_n et une variance de prédiction k_n ; il en est de même dans le cas d'estimation d'un ensemble d'excursion, on va prendre comme estimateur l'espérance de Vorob'ev et la distance moyenne donnée par l'équation (1.21) comme critère d'incertitude.

Dans ce contexte bayésien et en conditionnant le processus gaussien Y par les évaluations de la fonction g aux points \mathcal{X}_n , la fonction de couverture p devient

$$\begin{aligned} p_n(\mathbf{x}) &= \mathbb{P}(\mathbf{x} \in \Gamma \mid Y_{\mathcal{X}_n} = \mathbf{g}_n) \\ &= \mathbb{P}(Y_{\mathbf{x}} \leq c \mid Y_{\mathcal{X}_n} = \mathbf{g}_n) \\ &= \Phi\left(\frac{c - m_n(\mathbf{x})}{\sqrt{k_n(\mathbf{x}, \mathbf{x})}}\right), \end{aligned} \tag{1.23}$$

où $\Phi(\cdot)$ est la fonction de répartition de la loi normale centrée réduite. Les ensembles *quantiles de Vorob'ev* générés par la fonction de couverture p_n sont obtenus comme suit

$$\begin{aligned} Q_{n,\alpha} &= \{\mathbf{x} \in \mathbb{X} : p_n(\mathbf{x}) \geq \alpha\} \\ &= \{\mathbf{x} \in \mathbb{X} : \Phi\left(\frac{c - m_n(\mathbf{x})}{\sqrt{k_n(\mathbf{x}, \mathbf{x})}}\right) \geq \alpha\}. \end{aligned} \tag{1.24}$$

Ainsi l'espérance de Vorob'ev peut être calculée en déterminant le paramètre α_n^* tel que la condition (1.20) soit vérifiée, en remarquant que

$$\mathbb{E}[\mu(\Gamma) \mid Y_{\mathcal{X}_n} = \mathbf{g}_n] = \int_{\mathbb{X}} p_n(\mathbf{x}) d\mu(\mathbf{x}) \tag{1.25}$$

D'un point de vue computationnel, l'évaluation de la fonction de couverture en un point dépend de l'approximation de la fonction de répartition de la loi normale centrée réduite. Sous le langage de programmation **R**, une approximation rapide est obtenue à l'aide de la fonction `pnorm`. Pour le choix du paramètre α_n^* et ainsi l'espérance de Vorob'ev, on peut s'appuyer sur une simple recherche dichotomique dans l'intervalle $[0, 1]$ et une discrétisation de l'espace \mathbb{X} . Sous **R**, cette procédure est implémentée sous la fonction `vorob_threshold` du package `KrigInv` (cf. [Chevalier et al., 2014b]).

Une fois l'espérance de Vorob'ev déterminée, le calcul de la *déviatio*n de Vorob'ev se fait de manière directe. En effet

$$\begin{aligned} \mathbb{E}_n[\mu(\Gamma \Delta Q_{n,\alpha_n^*})] &= \mathbb{E}_n[\mu(\Gamma \cap Q_{n,\alpha_n^*}^c) + \mu(Q_{n,\alpha_n^*} \cap \Gamma^c)] \\ &= \mathbb{E}_n\left[\int_{\mathbb{X}} 1_{\{\mathbf{x} \in \Gamma, \mathbf{x} \notin Q_{n,\alpha_n^*}\}} + 1_{\{\mathbf{x} \notin \Gamma, \mathbf{x} \in Q_{n,\alpha_n^*}\}} \mu(d\mathbf{x})\right] \\ &= \int_{Q_{n,\alpha_n^*}^c} \mathbb{E}_n[1_{\{\mathbf{x} \in \Gamma\}}] \mu(d\mathbf{x}) + \int_{Q_{n,\alpha_n^*}} \mathbb{E}_n[1_{\{\mathbf{x} \notin \Gamma\}}] \mu(d\mathbf{x}) \\ &= \int_{Q_{n,\alpha_n^*}^c} p_n(\mathbf{x}) \mu(d\mathbf{x}) + \int_{Q_{n,\alpha_n^*}} (1 - p_n(\mathbf{x})) \mu(d\mathbf{x}), \end{aligned} \tag{1.26}$$

où $\mathbb{E}_n[\cdot] = \mathbb{E}[\cdot \mid Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}]$. La figure 1.5 montre l'espérance et la *déviatio*n de Vorob'ev obtenues pour l'exemple introduit précédemment.

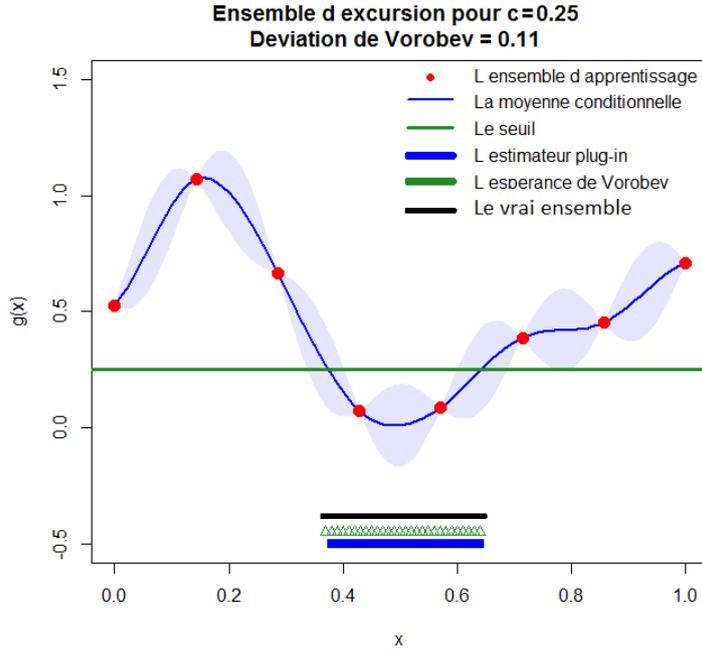


FIGURE 1.5: L'espérance de Vorob'ev Q_{n,α_n^*} (en vert), l'estimateur *plug-in* $\Gamma_{\text{plug-in}}$ (en bleu), les $n = 8$ observations (points rouges), la moyenne conditionnelle m_n (courbe bleue) et la bande de confiance (bande bleu clair).

1.3 Stratégie de réduction séquentielle d'incertitudes

Dans les deux sections précédentes, on s'est placé dans un cadre bayésien dans lequel on a défini les principales méthodes utilisées pour fournir une estimation d'un ensemble d'excursion ainsi qu'une quantification des incertitudes associées. Le but de cette section est de définir une **stratégie séquentielle** d'évaluation du simulateur afin de réduire l'incertitude associée à cette estimation.

1.3.1 La stratégie SUR

Les stratégies séquentielles guidées par des métamodèles sont aussi utilisées pour l'estimation des lignes de niveaux (cf. [Ranjan et al., 2008]), d'une région cible (cf. [Picheny et al., 2010]) ou plus récemment un ensemble d'excursion (cf. [Chevalier et al., 2014a]). L'idée derrière ces stratégies est de sélectionner le prochain point d'évaluation de façon pertinente en utilisant toute l'information disponible grâce aux évaluations précédentes. Toutes ces stratégies peuvent être regroupées en un seul nom : la stratégie *Stepwise Uncertainty Reduction* (SUR). L'objectif de la stratégie SUR est de sélectionner le prochain point d'évaluation afin de réduire une fonction *incertitude*.

Cette fonction *incertitude* dépend du problème qu'on cherche à résoudre. Par exemple dans [Vazquez and Bect, 2009], l'objectif est d'estimer une probabilité de défaillance, ainsi la fonction *incertitude* considérée n'est autre que la variance de l'estimateur de cette probabilité. Dans notre contexte, on cherche à identifier un ensemble d'excursion, ce dernier est estimé par l'espérance de Vorob'ev Q_{n,α_n^*} et l'incertitude associée à cet estimateur est évaluée via la *déviatoin de Vorob'ev* égale à $\mathbb{E}_n[\mu(\Gamma \Delta Q_{n,\alpha_n^*})]$, on note la fonction *incertitude* par

$$\mathcal{H}_n^{\text{uncert}} = \mathbb{E}[\mu(\Gamma \Delta Q_{n,\alpha_n^*}) \mid Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}] \quad (1.27)$$

Ainsi à l'étape n , pour chaque point $\mathbf{x} \in \mathbb{X}$, on évalue quelle serait la *déviatiion de Vorob'ev* à l'étape $(n+1)$ si on prenait ce point comme $(n+1)$ ème point. Or pour calculer cette incertitude nous avons besoin de l'évaluation de g au point \mathbf{x} . Pour combler ce manque d'information, nous utilisons la loi de $Y_{\mathbf{x}}$ sachant $(\mathcal{X}_n, \mathbf{g}_{\mathcal{X}_n})$ qui est connue. L'équation à l'étape $(n+1)$ devient donc

$$\mathcal{H}_{n+1}^{\text{uncert}}(\mathbf{x}) = \mathbb{E}[\mu(\Gamma \Delta Q_{n+1, \alpha_{n+1}^*}) \mid Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}, Y_{\mathbf{x}}]. \quad (1.28)$$

Le but est de trouver le point \mathbf{x}_{n+1} qui minimise la quantité 1.28 en moyenne, et ce en intégrant selon la loi de $Y_{\mathbf{x}}$ sachant $(\mathcal{X}_n, \mathbf{g}_{\mathcal{X}_n})$. On a alors :

$$\begin{aligned} \mathbf{x}_{n+1} &\in \arg \min_{\mathbf{x} \in \mathbb{X}} \mathbb{E}_{n, \mathbf{x}}[\mathcal{H}_{n+1}^{\text{uncert}}(\mathbf{x})] \\ &:= \arg \min_{\mathbf{x} \in \mathbb{X}} \mathcal{J}_n(\mathbf{x}) \end{aligned} \quad (1.29)$$

où $\mathbb{E}_{n, \mathbf{x}}$ est l'espérance sachant $Y_{\mathbf{x}} \mid Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}$. Pour tout $\mathbf{x} \in \mathbb{X}$, le critère \mathcal{J}_n s'écrit comme

$$\begin{aligned} \mathcal{J}_n(\mathbf{x}) &= \mathbb{E}_{n, \mathbf{x}}[\mathcal{H}_{n+1}^{\text{uncert}}(\mathbf{x})] \\ &= \mathbb{E}_{n, \mathbf{x}} \left[\int_{\mathbb{X}} \left(p_{n+1}(\mathbf{x}) \mathbb{1}_{(p_{n+1}(\mathbf{x}) < \alpha_{n+1}^*)} + (1 - p_{n+1}(\mathbf{x})) \mathbb{1}_{(p_{n+1}(\mathbf{x}) \geq \alpha_{n+1}^*)} \right) \mu(d\mathbf{x}) \right]. \end{aligned} \quad (1.30)$$

Grâce à des formules de mise à jour de la moyenne conditionnelle et du noyau de covariance conditionnelle, on obtient la proposition suivante :

Proposition 1. Le critère \mathcal{J}_n peut être exprimé sous forme explicite (la démonstration est donnée dans [Chevalier and Ginsbourger, 2013], chapitre 4.2),

$$\begin{aligned} \mathcal{J}_n(\mathbf{x}) &= \int_{\mathbb{X}} \left(2\Phi_2 \left(\left(\begin{bmatrix} a_{n+1}(u) \\ \Phi^{-1}(\alpha_n^*) - a_{n+1}(u) \end{bmatrix}; \begin{bmatrix} 1 + \gamma_{n+1}(u) & -\gamma_{n+1}(u) \\ -\gamma_{n+1}(u) & \gamma_{n+1}(u) \end{bmatrix} \right) \right) \right. \\ &\quad \left. + p_n(u) + \Phi \left(\frac{a_{n+1}(u) - \Phi^{-1}(\alpha_n^*)}{\sqrt{\gamma_{n+1}(u)}} \right) \right) \mu(du) \end{aligned} \quad (1.31)$$

avec

$$\begin{aligned} \mathbf{x} &= (x^{(1)}, x^{(2)}, \dots, x^{(q)}) \in \mathbb{X}^q \\ a_{n+1}(u) &= \frac{c - m_n(u)}{\sqrt{k_{n+1}(u, u)}} \\ \gamma_{n+1}(u) &= \mathbf{b}_{n+1}^t(u) \mathbf{b}_{n+1}(u) \\ \mathbf{b}_{n+1}(u) &= \frac{k_n(\mathbf{x}, u)}{\sqrt{k_{n+1}(u, u)}}, \quad u \in \mathbb{X} \end{aligned}$$

où $\Phi_2(\cdot, \Sigma)$ est la fonction de répartition de la loi gaussienne centrée bivariée de matrice de covariance Σ .

Pour plus de perspectives théoriques sur la stratégie SUR, on renvoie le lecteur au papier [Bect et al., 2016] qui traite de la convergence d'une telle stratégie dans différents cas d'applications. Dans le cas où *l'incertitude* est définie par la *déviatiion de Vorob'ev*, la convergence n'a pas encore été prouvée. La stratégie SUR appliquée à un problème d'inversion peut être résumée par l'algorithme suivant :

Algorithm 1 Stratégie séquentielle d'évaluation du simulateur pour l'estimation d'un ensemble d'excursion

```

1: Initial DoE : le plan d'expériences initial  $\mathcal{X}_n$  et les simulations  $\mathbf{g}_{\mathcal{X}_n}$ 
2: while le budget alloué  $r$  n'est pas atteint, i.e.,  $n \leq r$  do
3:    $Y|Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n} \leftarrow$  la modélisation par processus gaussien
4:    $\mathbf{x}_{n+1} \leftarrow$  l'optimisation du critère  $\mathcal{J}_n$ 
5:    $g(\mathbf{x}_{n+1}) \leftarrow$  l'évaluation de la fonction  $g$  au point  $\mathbf{x}_{n+1}$ 
6:    $n \leftarrow n + 1$ 
7: end while
8:  $Q_{n,\alpha_n^*} \leftarrow$  l'espérance de Vorob'ev comme estimateur de l'ensemble d'excursion
9:  $\mathbb{E}_n[\mu(\Gamma \Delta Q_{n,\alpha_n^*})] \leftarrow$  la déviation de Vorob'ev comme incertitude associée à l'estimation
10: end

```

1.3.2 Exemple d'application

Dans cette section, on considère la fonction Branin-Hoo définie comme suit

$$g : \begin{cases} \mathbb{X} = [0, 1]^2 & \rightarrow \mathbb{R} \\ \mathbf{x} = (x_1, x_2) & \mapsto \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10, \end{cases} \quad (1.32)$$

L'objectif est d'estimer l'ensemble $\Gamma^* = \{\mathbf{x} \in [0, 1]^2, g(\mathbf{x}) \leq c\}$, où le seuil est fixé à $c = 80$. La fonction g est évaluée sur un plan d'expériences de taille $n = 10$, on obtient ainsi un ensemble d'apprentissage $(\mathcal{X}_n, \mathbf{g}_{\mathcal{X}_n})$. Une modélisation par processus gaussien est alors faite, tenant compte des observations comme cela a été décrit dans la section 1.1.3, on prend un noyau de covariance de type Matérn 5/2 et les paramètres de ce noyau sont estimés par maximum de vraisemblance. Ce processus gaussien conditionnel définit un ensemble aléatoire fermé Γ sur lequel on s'appuie pour fournir un estimateur pour l'ensemble d'excursion Γ^* ainsi qu'une quantification d'incertitude associée.

Dans cet exemple, on se fixe un budget de $r = 40$ points et on va mettre en compétition deux stratégies : la première consiste à remplir l'espace indépendamment du modèle et de l'objectif d'inversion. On a choisi une suite de Sobol. La deuxième est la stratégie SUR dont le but est de minimiser la *déviatio*n de Vorob'ev. A chaque itération, la fonction g est évaluée au nouveau point proposé et le processus gaussien est mis à jour via la ré-estimation des paramètres du noyau de covariance.

La figure 1.6 (haut à droite) montre l'évolution de la *déviatio*n de Vorob'ev en fonction des points ajoutés au plan. On remarque que la stratégie SUR mène à une décroissance plus rapide de la *déviatio*n de Vorob'ev et que l'espérance de Vorob'ev obtenue après 40 points ajoutés se confond parfaitement avec le vrai ensemble Γ^* . On remarque aussi que les points proposés par la stratégie SUR se trouvent dans le voisinage de la frontière de l'ensemble. C'est en effet dans cette zone qu'on attend le plus d'incertitude.

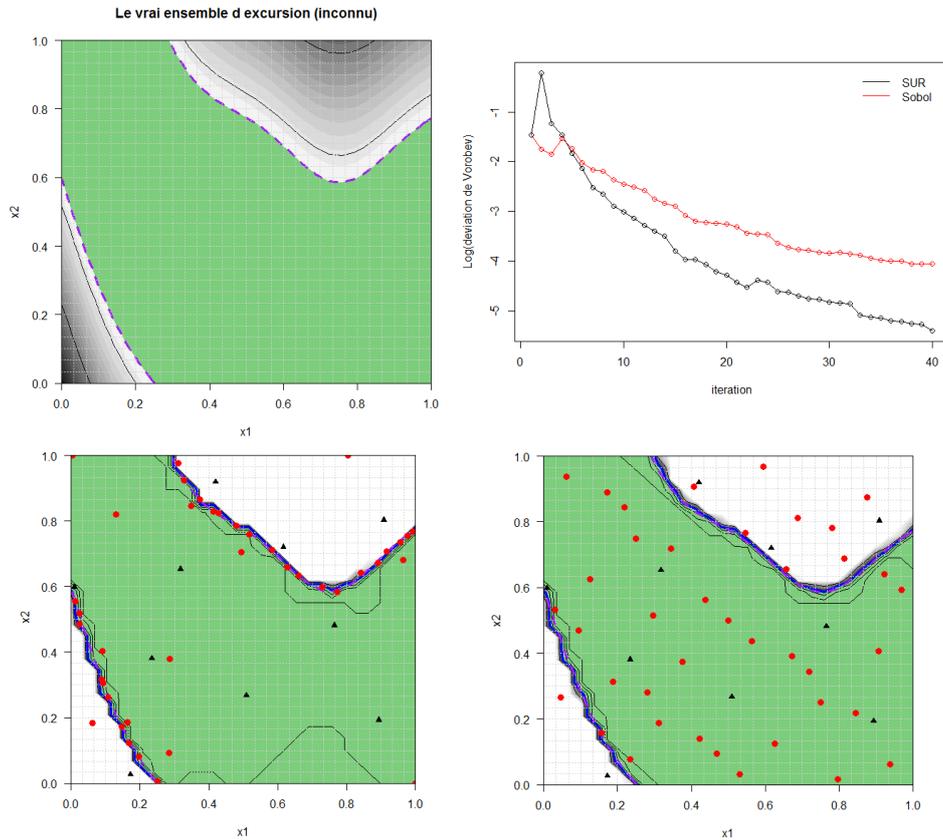


FIGURE 1.6: Exemple analytique. en haut à gauche : Le vrai ensemble d'excursion (ensemble vert de frontière mauve). En bas à droite : le plan initial (triangles noirs), les points ajoutés par la stratégie SUR (points rouges), la fonction de couverture et l'espérance de Vorob'ev (ensemble vert de frontière bleue) à l'étape $n = 50$. En haut à droite : l'évolution du logarithme de la *déviaton de Vorob'ev* en fonction des itérations. En bas à droite : le plan initial (triangles noirs), les points ajoutés par la suite de Sobol (points rouges), la fonction de couverture et l'espérance de Vorobev (ensemble vert de frontière bleue) à l'étape $n = 50$.

Synthèse : L'objectif de ce chapitre est d'introduire les méthodes existantes pour la résolution d'un problème d'inversion pour des simulateurs coûteux en temps de calcul. On a présenté une stratégie séquentielle d'évaluation du simulateur afin d'identifier l'ensemble recherché à moindre coût. Cette stratégie s'appuie sur les deux notions suivantes :

- la modélisation par processus gaussiens,
- la théorie des ensembles aléatoires fermés.

On s'est placé dans un cadre bayésien pour la modélisation du simulateur. En effet un tel cadre permet de formaliser l'estimation de l'ensemble d'excursion et de quantifier l'incertitude associée. Enfin, l'objectif de la stratégie SUR est de minimiser cette incertitude en proposant des points judicieusement choisis.

L'objectif du chapitre qui suit est de présenter deux méthodes originales pour estimer de manière séquentielle l'espérance de la sortie du code en un point de contrôle fixé quand cette sortie dépend d'une variable fonctionnelle incertaine dont la distribution n'est connue qu'au travers un échantillon de réalisations. Le couplage de ces méthodes

avec une approche de type SUR (objet du chapitre 3) permettra de résoudre des problèmes d'inversion sous incertitude fonctionnelle. La combinaison de ces approches et la stratégie présentée dans ce premier chapitre donnera naissance à une méthodologie pour résoudre des problèmes d'inversion sous incertitude fonctionnelle. Cette dernière est présentée dans le chapitre 3.

Chapitre 2

Intégration numérique

Outlines

L'objectif de ce chapitre est de proposer deux méthodes d'estimation séquentielle de l'espérance dans le cas de variable aléatoire fonctionnelle. On commencera par rappeler les approches existantes pour l'estimation de l'espérance dans le cadre vectoriel. On étendra quelques notions au cadre fonctionnel. Par la suite, on introduira deux méthodes découlant du concept de la quantification pour l'estimation gloutonne de l'espérance. Ces notions seront combinées aux outils présentés dans le chapitre 1 pour introduire une méthodologie permettant de résoudre un problème d'inversion sous incertitude fonctionnelle. La méthodologie globale sera présentée dans le chapitre 3.

Contents

2.1	Cas vectoriel	22
2.1.1	Approches Monte-Carlo et quasi-Monte Carlo	22
2.1.2	La quantification vectorielle	23
2.1.2.1	Introduction	23
2.1.2.2	Définitions et propriétés	24
2.1.2.3	Algorithmes existants	26
2.1.2.4	Application en intégration numérique	27
2.2	Cas fonctionnel	28
2.2.1	Réduction de dimension	29
2.2.2	La quantification fonctionnelle	30
2.2.2.1	Introduction	30
2.2.2.2	Constructions gloutonnes	31
2.3	Applications	34

On s'intéresse dans ce chapitre à l'estimation de l'espérance d'une fonctionnelle h d'une variable aléatoire \mathbf{U} à partir de l réalisations u_1, \dots, u_l de la variable \mathbf{U} . Dans la section 2.1, on rappelle les méthodes usuelles de Monte Carlo et quasi-Monte Carlo pour l'estimation de l'espérance dans un cadre fini-dimensionnel. On introduira dans la même partie le concept de la quantification vectorielle. Dans la section 2.2, on se place dans un cadre fonctionnel. On commence par présenter le concept et les outils de quantification fonctionnelle et on étudie comment ces outils peuvent être adaptés au contexte de la thèse où on dispose d'un échantillon u_1, \dots, u_N , à partir duquel on souhaite estimer $\mathbb{E}[h(\mathbf{U})]$. Nous proposons un algorithme glouton original, exploitant les outils de quantification ainsi qu'une variante moins coûteuse. La section 2.3 est consacrée à la comparaison des méthodes à travers différents exemples analytiques.

Remarque : Une partie de ce chapitre (Sections 2.2 et 2.3) est redondante avec le chapitre 3 qui réutilise la quantification fonctionnelle gloutonne pour répondre au problème d'inversion sous incertitudes. On tient à faire un chapitre qui regroupe l'ensemble des outils de la thèse qui porte sur la quantification.

2.1 Cas vectoriel

2.1.1 Approches Monte-Carlo et quasi-Monte Carlo

Soit \mathbf{U} une variable aléatoire de loi de probabilité $\mathbb{P}_{\mathbf{U}}$ sur \mathbb{R}^m . La méthode de Monte Carlo consiste à estimer l'espérance $\mathbb{E}[h(\mathbf{U})]$ où h est une fonction telle que $h(\mathbf{U})$ est intégrable. Cette approximation est obtenue en générant $\mathbf{u}_1, \dots, \mathbf{u}_l$ l copies i.i.d. selon la loi $\mathbb{P}_{\mathbf{U}}$ et à calculer

$$\mathbb{E}[h(\mathbf{U})] \approx \frac{1}{l} \sum_{i=1}^l h(\mathbf{u}_i). \quad (2.1)$$

Il est bien connu que par la loi des grands nombres, cette quantité converge presque sûrement quand l tend vers l'infini vers $\mathbb{E}[h(\mathbf{U})]$. De plus avec le théorème central limite, on a une vitesse de convergence en probabilité en $O(l^{-1/2})$ indépendante de la dimension m .

Dans l'objectif de réduire la variance de l'estimateur Monte Carlo, différentes techniques ont été mises au point, parmi lesquelles nous citons : la méthode des variables antithétiques, des variables de contrôle, d'échantillonnage préférentiel, d'échantillonnage stratifié. En particulier, l'échantillonnage par hypercube latin est une forme d'échantillonnage stratifié (voir [Owen, 2013]).

Un mot sur les méthodes quasi-Monte Carlo

L'écueil principal des méthodes de Monte Carlo est leur taux de convergence en $O(l^{-1/2})$. Il se trouve que, notamment dans le calcul d'intégrales de dimension raisonnable, on peut sous certaines conditions faire sensiblement mieux avec la méthode dite de quasi-Monte Carlo. On peut en effet montrer que, sous réserve que les u_i soient judicieusement choisis (il s'agira en général d'une suite déterministe, mais irrégulière de l -uplets), la convergence peut atteindre des vitesses en $O(l^{-1})$ pour $m = 1$ et $O(l^{-1} \log(l)^{m-1})$ pour $m > 1$, sous réserve de certaines propriétés de régularité de h .

Définition 3 ([Niederreiter, 1992] Définition 2.1). Soit $\{u_i\}_{i>0}$ une suite points de $[0, 1]^m$; pour tout ensemble $B \subset [0, 1]^m$, on désigne par $\lambda_l(B)$ la proportion des l premiers points de cette suite appartenant à B . La discrédance des l premiers points de la suite est :

$$\mathcal{D}_l^*(\mathbf{U}) = \sup_B |\lambda_l(B) - \lambda(B)|, \quad \lambda_l(B) = \frac{1}{l} \sum_{i=1}^l \mathbf{1}_B(u_i),$$

où λ est la mesure de Lebesgue et le sup est pris sur tous les B de la forme $B = \prod_{i=1}^m [0, x_i]$, $x_i \leq 1$.

La discrédance compare donc la fonction de répartition empirique à celle de la mesure de Lebesgue. Il existe d'autres mesures de discrédance dont on ne parlera pas ici (voir [Franco, 2008]).

Définition 4. Soit $h : [0, 1]^m \rightarrow \mathbb{R}$. La variation totale de h au sens de Hardy et Krause est définie par

$$V(h) = \sum_{k=1}^m \sum_{i_1 \leq \dots \leq i_k} \int_{[0,1]^k} \left| \frac{\partial^k h}{\partial x_{i_1} \dots \partial x_{i_k}} \right| dx_{i_1} \dots dx_{i_k}.$$

Théorème 2 (Inégalité de Koksma-Hlawka [Niederreiter, 1992] Théorème 2.11). Pour toute fonction $h : [0, 1]^m \rightarrow \mathbb{R}$ à variation bornée, on obtient la borne sur l'erreur d'approximation suivante

$$\left| \frac{1}{l} \sum_{i=1}^l h(u_i) - \int_{[0,1]^m} h(u) du \right| \leq V(h) \mathcal{D}_l^*(\mathbf{U}).$$

Il s'agit maintenant de trouver des suites $\{u_i\}_{i>0}$ telles que $(\mathcal{D}_l^*(\mathbf{U}))_{l>0}$ décroisse le plus vite possible. Il existe des méthodes de construction arithmétiques de suites (Sobol (cf. Figure 2.1), Halton, ...) telles que

$$\mathcal{D}_l^*(\mathbf{U}) = O(l^{-1} \log(l)^m), \text{ pour } m > 1.$$

Pour $m = 1$, l'ensemble des points milieux $u_i = \frac{2i+1}{2l}$ est optimale avec $\forall l > 0, \mathcal{D}_l^*(\mathbf{U}) = \frac{1}{2l}$ (voir [Lemieux, 2009]).

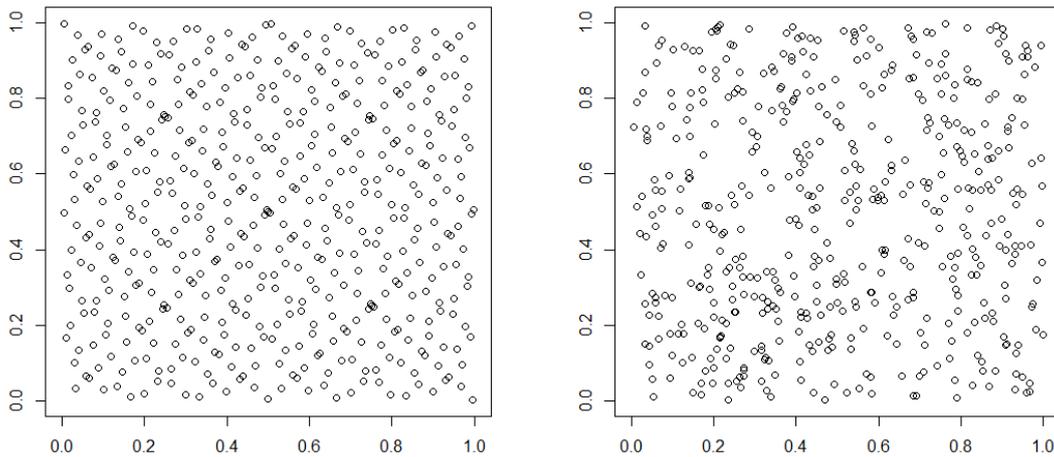


FIGURE 2.1: Une suite à faible discrédance (Sobol) et une suite aléatoire dans $[0, 1]^2$ ($l = 500$).

2.1.2 La quantification vectorielle

Cette section est dédiée à la quantification vectorielle alternative aux suites à faible discrédance et à la méthode de Monte Carlo pour évaluer une espérance. Après une brève introduction des origines de la quantification, on va donner précisément les propriétés et fournir quelques résultats concernant l'intégration numérique et la convergence des algorithmes.

2.1.2.1 Introduction

Le terme quantification prend naissance au milieu des années 50 où il est employé en théorie de l'information et du signal. Dans ce contexte, la quantification signifie la discrétisation d'un signal continu en un ensemble fini de points, appelé quantifieur. Le but est d'avoir une transmission efficace et parcimonieuse du signal. Par conséquent, la taille et la géométrie du quantifieur doivent être optimisées à cet effet.

Ce concept a également connu un intérêt important dans le domaine de la reconnaissance de formes et la classification. Dans ce contexte, le problème de la quantification optimale consiste à trouver la meilleure approximation de la distribution continue $\mathbb{P}_{\mathbf{U}}$ d'un vecteur aléatoire \mathbf{U} par une variable aléatoire discrète prenant un nombre fini de valeurs. La quantification a été traitée dans un cadre probabiliste notamment par [Zador, 1963] ou [Pagès, 1998] et a été intensivement utilisée en probabilité numérique, en finance et en intégration numérique par [Pagès and Printems, 2003], [Pages et al., 2004] et [Bally et al., 2005].

2.1.2.2 Définitions et propriétés

Soit $(\Omega, \mathcal{F}, \mathbb{P})$ un espace probabilisé et \mathbf{U} une variable aléatoire à valeurs dans \mathbb{R}^m telle que $\mathbb{E}|\mathbf{U}|^2 < \infty$. On note $\|\mathbf{U}\|_2 = (\mathbb{E}|\mathbf{U}|^2)^{1/2}$ et $\mathbb{P}_{\mathbf{U}}$ la loi de probabilité de \mathbf{U} . La quantification optimale consiste à étudier la meilleure approximation de \mathbf{U} par une variable aléatoire $\mathbf{U}'_l = \pi(\mathbf{U})$ prenant au plus l valeurs dans \mathbb{R}^m . L'objectif alors est de trouver le vecteur aléatoire qui minimise l'erreur de quantification $\|\mathbf{U} - \pi(\mathbf{U})\|_2$.

Définition 5 (Quantification de Voronoi). Soit $A = \{u_1, \dots, u_l\}$ un l -uplet dans \mathbb{R}^m et $C_{u_i}(A), i = 1, \dots, l$, une partition de Voronoi de A définie par :

$$C_{u_i}(A) \subset \{u \in \mathbb{R}^m : |u_i - u| = \min_{j=1, \dots, l} |u_j - u|\} = \bar{C}_{u_i}(A). \quad (2.2)$$

On appelle quantifieur de Voronoi de \mathbf{U} , la variable aléatoire \mathbf{U}'_l à valeurs dans la grille A , définie par :

$$\mathbf{U}'_l = \pi_A(\mathbf{U}) := \sum_{i=1}^l u_i \mathbf{1}_{C_{u_i}}(\mathbf{U}), \quad (2.3)$$

et donc de loi de probabilité discrète $\mathbb{P}_{\mathbf{U}'_l}$ caractérisée par :

$$p_i := \mathbb{P}(\mathbf{U}'_l = u_i) = \mathbb{P}(\mathbf{U} \in C_{u_i}(A)), \quad i = 1, \dots, l. \quad (2.4)$$

En d'autres termes, \mathbf{U}'_l est la projection selon le plus proche voisin de la variable aléatoire \mathbf{U} sur la grille A . Les p_i sont appelés aussi masses des cellules de Voronoi. L'erreur résultante au carré de la quantification est appelée distorsion (quadratique) et s'écrit donc par définition :

$$\begin{aligned} D_l^{\mathbf{U}}(A) &= \|\mathbf{U} - \mathbf{U}'_l\|_2^2 = \|\mathbf{U} - \pi_A(\mathbf{U})\|_2^2 \\ &= \mathbb{E} \left[\left| \mathbf{U} - \sum_{i=1}^l u_i \mathbf{1}_{C_{u_i}}(\mathbf{U}) \right|^2 \right] \\ &= \sum_{i=1}^l \mathbb{E} \left[|\mathbf{U} - u_i|^2 \mathbf{1}_{C_{u_i}}(\mathbf{U}) \right] \\ &= \mathbb{E} \left[\min_{i=1, \dots, l} |\mathbf{U} - u_i|^2 \right] \\ &= \int_{\mathbb{R}^m} \min_{i=1, \dots, l} |u - u_i|^2 \mathbb{P}_{\mathbf{U}}(du) = \int_{\mathbb{R}^m} d_l(A, u) \mathbb{P}_{\mathbf{U}}(du), \end{aligned} \quad (2.5)$$

où $d_l : (\mathbb{R}^m)^l \times \mathbb{R}^m \rightarrow \mathbb{R}_+$ est la distorsion locale définie par :

$$d_l(A, u) = \min_{i=1, \dots, l} |u - u_i|^2 = \sum_{i=1}^l |u - u_i|^2 \mathbf{1}_{C_{u_i}}(u). \quad (2.6)$$

Le problème de la quantification optimale consiste à minimiser la distorsion sur les l -uplets A . On dira que $\hat{\mathbf{U}}_l$ est un l -quantifieur optimal de \mathbf{U} si :

$$\mathbb{E}[|\mathbf{U} - \hat{\mathbf{U}}_l|^2] = \inf_{A \in (\mathbb{R}^m)^l} D_l^{\mathbf{U}}(A) = \underline{D}_l^{\mathbf{U}}.$$

Ces définitions nous poussent à nous poser les questions suivantes : est ce que la fonction distorsion $D_l^{\mathbf{U}}(\cdot)$ atteint son infimum ? Comment ce dernier se comporte quand l tend vers l'infini ?

En effet, on remarque que la distorsion ne dépend que du l -uplet A et de la loi de probabilité $\mathbb{P}_{\mathbf{U}}$ de \mathbf{U} . Un premier résultat nous garantit que la fonction $A \rightarrow D_l^{\mathbf{U}}(A)$ atteint son infimum, et ce dernier sera noté \hat{A} . Soit $\mathcal{D}_l = \{A \in (\mathbb{R}^m)^l : u_i \neq u_j \text{ ssi } i \neq j\}$, on note par $C := \text{conv}(S_{\mathbf{U}})$ l'enveloppe convexe fermée du support topologique de $\mathbb{P}_{\mathbf{U}}$ sur \mathbb{R}^m .

Proposition 2 ([Pagès, 1998], Prop. 5). La fonction $A \rightarrow D_l^{\mathbf{U}}(A)$ est continue sur $(\mathbb{R}^m)^l$ et atteint son minimum dans l'ensemble C . De plus $\text{argmin}_{A \in (\mathbb{R}^m)^l} D_l^{\mathbf{U}}(A) \subset \mathcal{D}_l$ et la suite $l \rightarrow \min_{(C)^l} D_l^{\mathbf{U}}(A)$ tend vers 0.

La preuve est donnée dans [Pagès, 1998]. Cette proposition implique que la fonction $A \rightarrow D_l^{\mathbf{U}}(A)$ atteint son minimum dans $(\mathbb{R}^m)^l$ en un certain point $\hat{A} \in (\mathbb{R}^m)^l$, et que ce minimum a des composantes distinctes, i.e., $|\hat{\mathbf{U}}_l(\Omega)| = l$ si $|S_{\mathbf{U}}| \geq l$. De plus, l'erreur de quantification tend vers zéro quand l tend vers l'infini. Cependant, la question sur le taux de convergence est un problème assez difficile. Le taux de convergence de l'erreur de quantification est donné par le théorème de Zador ([Zador, 1963]) ; voir aussi [Graf and Luschgy, 2000] pour la preuve.

Théorème 3. (a) **Théorème de Zador** [Zador, 1963]. Soit $\mathbb{P}_{\mathbf{U}} = \mathbb{P}_a + \mathbb{P}_s$ la décomposition de $\mathbb{P}_{\mathbf{U}}$ par rapport à la mesure de Lebesgue λ , où \mathbb{P}_a est la partie absolument continue et \mathbb{P}_s la partie singulière de $\mathbb{P}_{\mathbf{U}}$. Supposons que $\mathbb{E}[|\mathbf{U}|^{2+\zeta}] < \infty$ pour un $\zeta > 0$. Alors

$$\lim_{l \rightarrow +\infty} l^{2/m} D_l^{\mathbf{U}}(A) = Q_2(\mathbb{P}_{\mathbf{U}}),$$

avec $Q_2(\mathbb{P}_{\mathbf{U}}) = J_{(2,m)} \left(\int_{\mathbb{R}^m} f^{\frac{m+2}{m}} d\lambda \right)^{\frac{m}{m+2}}$, où $f = \frac{d\mathbb{P}_a}{d\lambda}$ et $J_{(2,m)} \in [0, \infty)$.

(b) **Borne supérieure non-asymptotique** [Luschgy et al., 2008]. Il existe $C_{m,2,\zeta} \in [0, \infty)$ tel que pour toute variable aléatoire \mathbf{U} à valeurs dans \mathbb{R}^m ,

$$\forall l \geq 1, \quad (D_l^{\mathbf{U}}(A))^{1/2} \leq C_{m,2,\zeta} \|\mathbf{U}\|_{2+\zeta} l^{-1/m}.$$

Ce résultat est amélioré dans [Pagès, 2018] où $\|\mathbf{U}\|_{2+\zeta}$ est remplacée par $\inf_{u \in \mathbb{R}^m} \|\mathbf{U} - u\|_{2+\zeta}$.

Remarques

- La vraie valeur de la constante $J_{(2,m)}$ est inconnue pour $m \geq 3$ mais on sait que $J_{(2,m)} \sim \frac{m}{2\pi e}$ lorsque $m \rightarrow +\infty$ et que $J_{(2,1)} = \frac{1}{2^2(2+1)}$.
- Lorsque $\mathbb{P}_a = 0$ alors $Q_2(\mathbb{P}_{\mathbf{U}}) = 0$. Dans ce cas, $D_l^{\mathbf{U}}(A) = o(l^{-1/m})$ et la vitesse de convergence est étudiée dans [Graf and Luschgy, 2000] pour un grand nombre de classes de probabilité singulières.

Une question naturelle se pose : comment obtenir le quantifieur optimal ? le but de la partie qui suit est de fournir quelques algorithmes pour résoudre ce problème d'optimisation.

2.1.2.3 Algorithmes existants

Il n'existe pas de formes analytiques pour déterminer le quantifieur optimal, sauf dans certains cas comme pour la loi uniforme unidimensionnelle. Cependant, il existe différents algorithmes permettant d'obtenir "une approximation" du quantifieur optimal. Soit \mathbf{U} une variable aléatoire à valeurs dans \mathbb{R}^m et de distribution de probabilité notée $\mathbb{P}_{\mathbf{U}}$. Soit $l \geq 1$, on rappelle que l'erreur de quantification associée à $A = \{u_1, \dots, u_l\} \in (\mathbb{R}^m)^l$, notée $D_l^{\mathbf{U}}(A)$, est donnée par :

$$D_l^{\mathbf{U}}(A) = \mathbb{E}[\min_{i=1, \dots, l} |\mathbf{U} - u_i|^2] = \int_{\mathbb{R}^m} \min_{i=1, \dots, l} |u - u_i|^2 \mathbb{P}_{\mathbf{U}}(du).$$

Dans cette partie, on se focalisera sur le problème d'optimisation suivant :

$$\begin{cases} \text{Trouver un } l\text{-uplets } \hat{A} = \{\hat{u}_1, \dots, \hat{u}_l\} \text{ t.q.} \\ D_l^{\mathbf{U}}(\hat{A}) \leq D_l^{\mathbf{U}}(A), \quad \forall A = \{u_1, \dots, u_l\} \in (\mathbb{R}^m)^l \end{cases}$$

L'algorithme de Lloyd I. Historiquement, cet algorithme a été proposé pour résoudre le problème d'optimisation en dimension $m = 1$. Cette méthode, aussi appelée *méthode du point fixe de Lloyd*, est une méthode itérative. Elle consiste à induire la grille $A^{(s+1)}$ de taille l sachant $A^{(s)}$ de même taille comme suit :

$$A^{(s+1)} = \mathbb{E}[\mathbf{U} | \pi_{A^{(s)}}(\mathbf{U})](\Omega) = (\mathbb{E}[\mathbf{U} | \mathbf{U} \in C_{u_i}(A^{(s)})](\Omega))_{1 \leq i \leq l}, \quad (2.7)$$

en d'autres termes, la méthode démarre par un quantifieur initial, $A^{(0)}$, et génère successivement une mise à jour, $A^{(s+1)}$, avec les nouveaux centroides, $u_i^{(s+1)}$, des régions associées à l'étape précédente $A^{(s)}$, et on écrit :

$$u_i^{(s+1)} = \frac{\int_{C_{u_i}(A^{(s)})} u \mathbb{P}_{\mathbf{U}}(du)}{\int_{C_{u_i}(A^{(s)})} \mathbb{P}_{\mathbf{U}}(du)} \quad \text{pour } i = 1, \dots, l. \quad (2.8)$$

Cependant en grande dimension, cette approche présente deux inconvénients. Premièrement, il s'agit d'une procédure locale puisqu'elle n'explore pas tout l'espace. En effet, si la méthode converge, il est possible qu'on soit piégé dans un minimum local. Le deuxième inconvénient est numérique, en grande dimension cette méthode nécessite le calcul d'intégrales m -dimensionnelles. Cependant, quand on sait générer un échantillon selon $\mathbb{P}_{\mathbf{U}}$, le calcul d'intégrales peut être approché par Monte Carlo. Plus de détails à propos de cette approche peuvent être trouvés dans [Pagès and Printems, 2003].

L'algorithme du gradient stochastique. La méthode du gradient stochastique est basée sur le fait que le gradient de la distorsion admet une représentation intégrale par rapport à la loi de probabilité de \mathbf{U} :

$$\nabla D_l^{\mathbf{U}}(A) = \int_{\mathbb{R}^m} \nabla_u d_l(A, u) \mathbb{P}_{\mathbf{U}}(du) = \mathbb{E}[\nabla_u d_l(A, \mathbf{U})], \quad (2.9)$$

où \mathbf{U} est une variable aléatoire de loi de probabilité $\mathbb{P}_{\mathbf{U}}$ et $\nabla_u d_l : (\mathbb{R}^m)^l \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ est donné par :

$$\frac{1}{2} \nabla_u d_l(A, u) = ((u_i - u) \mathbf{1}_{C_{u_i}(A)})_{1 \leq i \leq l}. \quad (2.10)$$

Grâce à cette représentation intégrale, on peut ainsi implémenter une descente du gradient stochastique pour approcher un minimum local de la distorsion $D_l^{\mathbf{U}}(A)$. En grande dimension,

cette approche devient irréalisable puisqu'elle nécessite le calcul d'intégrales multidimensionnelles. On renvoie le lecteur à [Pagès and Printems, 2003] pour plus de détails à propos de cette méthode.

L'algorithme glouton. Un algorithme glouton consiste dans notre contexte à choisir à l'étape l un nouveau point u_l réduisant l'erreur de quantification du quantifieur $\{u_1, \dots, u_{l-1}\} \cup \{u_l\}$ de façon optimale. Un résultat important est que l'erreur de quantification induite par la **suite** de quantifieurs $\{u_i\}_{1 \leq i \leq l}$ tend vers zéro à une vitesse en $O(l^{-1/m})$, qui est l'ordre optimal pour un l -quantifieur optimal en dimension m . La preuve de ce résultat est donnée dans [Luschgy and Pagès, 2015]. Si on fixe $\{u_1, \dots, u_{l-1}\}, l > 1$ alors

$$\forall l > 1, \quad u_l \in \arg \min_{u \in \mathbb{R}^m} D_l^{\mathbf{U}}(\{u_1, \dots, u_{l-1}\} \cup u). \quad (2.11)$$

L'intérêt de l'approche gloutonne est qu'elle permet d'enrichir itérativement le quantifieur.

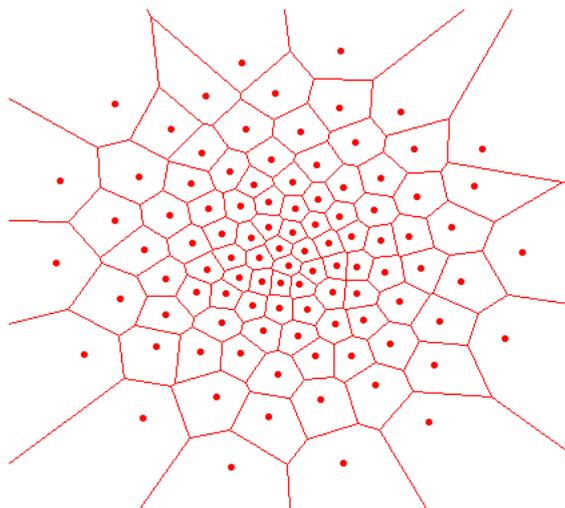


FIGURE 2.2: Une quantification de Voronoi de la loi $\mathcal{N}(0, I_2)$ où $l = 200$.

2.1.2.4 Application en intégration numérique

Soit \mathbf{U} une variable aléatoire à valeurs dans \mathbb{R}^m et soit h une fonction intégrable par rapport à $\mathbb{P}_{\mathbf{U}}$. On s'intéresse ici à l'estimation de la quantité

$$\mathbb{E}[h(\mathbf{U})] = \int_{\mathbb{R}^m} h(u) \mathbb{P}_{\mathbf{U}}(du). \quad (2.12)$$

L'idée est d'approcher la loi de probabilité $\mathbb{P}_{\mathbf{U}}$ par la loi de probabilité de la variable quantifiée $\hat{\mathbf{U}}_l$, notée $\mathbb{P}_{\hat{\mathbf{U}}_l}$, ainsi on approche (2.12) par

$$\mathbb{E}[h(\hat{\mathbf{U}}_l)] = \int_{\mathbb{R}^m} h(u) \mathbb{P}_{\hat{\mathbf{U}}_l}(du) = \sum_{i=1}^l h(u_i) \mathbb{P}(\mathbf{U} \in C_{u_i}(A)), \quad (2.13)$$

où $A = \{u_1, \dots, u_l\}$ est un quantifieur si possible optimal. Il s'agit donc d'une formule consistant à pondérer des valeurs de la fonction h en une famille de points de \mathbb{R}^m . On rappelle ici

quelques résultats élémentaires liés à l'erreur induite par cette approximation.

On a le résultat suivant pour les fonctions lipschitziennes. On note pour toute fonction h lipschitzienne sur \mathbb{R}^m :

$$[h]_{\text{lip}} = \sup_{y \neq z \in \mathbb{R}^m} \frac{|h(y) - h(z)|}{|y - z|} < \infty. \quad (2.14)$$

Proposition 3. Pour tout $h \in \mathbb{L}^1(\mathbb{P}_{\mathbf{U}})$ et h lipschitzienne, on a :

$$\begin{aligned} |\mathbb{E}[h(\mathbf{U})] - \mathbb{E}[h(\hat{\mathbf{U}}_l)]| &\leq \mathbb{E}[|h(\mathbf{U}) - h(\hat{\mathbf{U}}_l)|] \\ &\leq [h]_{\text{lip}} \mathbb{E}[|\mathbf{U} - \hat{\mathbf{U}}_l|] \leq [h]_{\text{lip}} \sqrt{D_l^{\mathbf{U}}(A)}. \end{aligned}$$

Par conséquent, cette proposition montre que si A est un quantifieur optimal alors $|\mathbb{E}[h(\mathbf{U})] - \mathbb{E}[h(\hat{\mathbf{U}}_l)]|$ converge vers 0 quand l tend vers l'infini avec une vitesse de convergence en $O(l^{-1/m})$ d'après le théorème de Zador,

$$|\mathbb{E}[h(\mathbf{U})] - \mathbb{E}[h(\hat{\mathbf{U}}_l)]| = O(l^{-1/m}).$$

Si la fonction h possède un peu plus de régularité, on peut améliorer la borne d'erreur.

Proposition 4. Pour tout $h \in \mathbb{L}^1(\mathbb{P}_{\mathbf{U}})$ telle que h soit de classe \mathcal{C}^1 avec ∇h lipschitzienne, on a :

$$|\mathbb{E}[h(\mathbf{U})] - \mathbb{E}[h(\hat{\mathbf{U}}_l)]| \leq \frac{1}{2} [\nabla h]_{\text{lip}} D_l^{\mathbf{U}}(A).$$

Ainsi si A est un quantifieur optimal alors

$$|\mathbb{E}[h(\mathbf{U})] - \mathbb{E}[h(\hat{\mathbf{U}}_l)]| = O(l^{-2/m}).$$

Pour $h \in \mathbb{L}^1(\mathbb{P}_{\mathbf{U}})$, lipschitzienne, l'erreur d'intégration numérique par quasi-MC est en $O(\mathcal{D}_l^*(\mathbf{U})^{1/m})$ où $\mathcal{D}_l^*(\mathbf{U})$ est la discrédance à l'origine, c'est à dire, en $O(\frac{\ln(l)}{l^{1/m}})$ pour les suites "classiques". Des grilles de quantifieurs optimaux et les poids (mais aussi les erreurs de quantification) associés peuvent être stockés offline. On trouvera ainsi des grilles de quantifieurs quadratiques et leurs poids associés pour la loi $\mathcal{N}(0, I_m)$ sur le site internet

www.quantize.maths-fi.com

pour $m \in \{1, \dots, 10\}$ et $l \in \{1, \dots, 5000\}$.

Après un rappel sur les méthodes Monte-Carlo, quasi-Monte-Carlo et les notions de quantification dans le cas vectoriel, la partie qui suit a pour but de travailler dans un cadre fonctionnel.

2.2 Cas fonctionnel

Soit \mathbf{V} une variable aléatoire fonctionnelle définie dans l'espace $\mathcal{H} = \mathbb{L}^2(\Omega, \mathcal{F}, \mathbb{P}; \mathcal{V})$, dont une réalisations $v = \mathbf{V}(\cdot, \omega)$ est définie dans l'espace des fonctions de carré intégrable sur $[0, T]$, noté $\mathcal{V} = \mathbb{L}^2([0, T])$ et muni du produit scalaire et de la norme

$$\langle v, v' \rangle = \int_0^T v_t v'_t dt \quad \text{et} \quad \|v\| = \left(\int_0^T v_t^2 dt \right)^{1/2}, \quad (2.15)$$

pour tout $v, v' \in \mathcal{V}$. La variable aléatoire $\mathbf{V}(t, \cdot) = \boldsymbol{\eta} \in \mathbb{L}^2(\Omega, \mathbb{P})$, muni du produit scalaire et de la norme

$$\langle \boldsymbol{\eta}, \boldsymbol{\eta}' \rangle_{\mathbb{L}^2(\Omega, \mathbb{P})} = \int_{\Omega} \boldsymbol{\eta} \boldsymbol{\eta}' d\mathbb{P} = \mathbb{E}[\boldsymbol{\eta} \boldsymbol{\eta}'] \quad \text{et} \quad \|\boldsymbol{\eta}\|_{\mathbb{L}^2(\Omega, \mathbb{P})} = \left(\int_{\Omega} \boldsymbol{\eta}^2 d\mathbb{P} \right)^{1/2} = (\mathbb{E}[\boldsymbol{\eta}^2])^{1/2}, \quad (2.16)$$

pour tout $\boldsymbol{\eta}, \boldsymbol{\eta}' \in \mathbb{L}^2(\Omega)$. Tout au long de ce manuscrit, on va considérer des variables aléatoires fonctionnelles définies dans l'espace de Hilbert $\mathcal{H} = \mathbb{L}^2(\Omega, \mathcal{F}, \mathbb{P}; \mathcal{V})$ muni du produit scalaire et de la norme

$$\langle \mathbf{V}, \mathbf{V}' \rangle_{\mathbb{L}^2} = \mathbb{E}[\langle \mathbf{V}, \mathbf{V}' \rangle] = \mathbb{E} \left[\int_0^T \mathbf{V} \mathbf{V}' dt \right] \quad \text{et} \quad \|\mathbf{V}\|_{\mathbb{L}^2} = (\mathbb{E}[\|\mathbf{V}\|^2])^{1/2} = \left(\mathbb{E} \left[\int_0^T \mathbf{V}^2 dt \right] \right)^{1/2}, \quad (2.17)$$

pour tout $\mathbf{V}, \mathbf{V}' \in \mathcal{H}$.

Dans un cadre où la variable fonctionnelle \mathbf{V} n'est connue qu'à travers un échantillon Ξ , on va utiliser ce dernier pour estimer l'espérance $\mathbb{E}[h(\mathbf{V})]$. Pour cela on approche la loi de probabilité $\mathbb{P}_{\mathbf{V}}$ par la mesure discrète $\frac{1}{\#\Xi} \sum_{i=1}^{\#\Xi} \delta_{v_i}$ et on approche l'espérance par Monte Carlo avec cette loi discrète. Plusieurs alternatives à cette approche ont été développées dans la littérature. Nous commencerons par présenter une approche classique de projection : la variable fonctionnelle est projetée sur un espace de dimension finie (section 2.2.1). Nous établirons ensuite un état de l'art des approches de quantification fonctionnelle (section 2.2.2). Nous présenterons enfin les approches gloutonnes (section 2.2.3), notamment deux algorithmes originaux développés dans cette thèse.

2.2.1 Réduction de dimension

Soit \mathbf{V} une variable aléatoire fonctionnelle à valeurs dans \mathcal{H} centrée de fonction de covariance $C(t, s)$. On peut représenter \mathbf{V} sous la forme

$$\mathbf{V}(t) = \sum_{j=1}^{\infty} \mathbf{u}_j \psi_j(t), \quad t \in [0, T], \quad (2.18)$$

où les $\{\psi_j\}_{j=1}^{\infty}$ sont fonctions propres de la fonction de covariance, i.e., solution de l'équation de Fredholm suivante

$$\lambda_j \psi_j(t) = \int_0^T C(t, s) \psi_j(s) ds. \quad (2.19)$$

Les $\{\mathbf{u}_j\}_{j=1}^{\infty}$ sont des variables aléatoires non corrélées et de variance $\{\lambda_j\}_{j=1}^{\infty}$. On note que l'utilisation d'une telle décomposition est conditionnée par une connaissance à priori de la fonction de covariance. Néanmoins, cette fonction peut être estimée à partir de l'échantillon initial de courbes Ξ . Si on note $\Xi = \{v_1, \dots, v_N\}$, la fonction de covariance $C(t, s)$ peut être estimée par

$$C^N(t, s) = \frac{1}{N} \sum_{k=1}^N v_k(s) v_k(t). \quad (2.20)$$

Ainsi on résout le problème (2.19) en remplaçant C par C^N . Des résultats de convergence sont donnés par [Cardot et al., 1999].

On représente ensuite \mathbf{V} sur la base $\{\psi_j^{(N)}\}_{j=1}^m$ tronquée à m éléments (le choix de m sera discuté plus tard),

$$\mathbf{V}(t) \approx \sum_{j=1}^m \mathbf{u}_j \psi_j^{(N)}(t), \quad t \in [0, T]. \quad (2.21)$$

La variable fonctionnelle est alors représentée par les coefficients $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_m)^\top \in \mathbb{R}^m$. Ces coefficients sont des variables aléatoires caractérisées par une densité de probabilité multivariée. Parmi les méthodes d'estimation pour la densité de variables aléatoires vectorielles, on peut citer : **l'estimation de densité à noyau** [Rosenblatt, 1956] qui est une méthode non-paramétrique d'estimation ; **l'estimation par mélange de gaussienne**, le but est d'approcher la distribution de probabilité par mélange de lois normales et d'estimer les paramètres par *Expectation-Maximisation* [Dempster et al., 1977].

Ainsi, une fois placé en dimension finie, et la loi de probabilité de la variable aléatoire \mathbf{U} estimée, on peut appliquer les méthodes introduites dans la section 2.1 pour estimer l'espérance $\mathbb{E}[h(\mathbf{V})]$.

2.2.2 La quantification fonctionnelle

2.2.2.1 Introduction

De même que pour le cas vectoriel, la quantification fonctionnelle a pour but d'approcher la variable aléatoire fonctionnelle \mathbf{V} par une variable aléatoire discrète $\hat{\mathbf{V}}_l$ à valeurs dans l'ensemble $\hat{A}_l = \{\hat{v}_1, \dots, \hat{v}_l\} \in \mathcal{V}^l$. Le problème consiste à minimiser l'erreur de quantification :

$$D_l^{\mathbf{Y}}(\hat{A}_l) = \inf\{(\mathbb{E}\|\mathbf{V} - \hat{\mathbf{V}}_l\|^2)^{1/2}, \hat{\mathbf{V}}_l : \Omega \rightarrow \mathcal{V}, \text{card}(\hat{\mathbf{V}}_l(\Omega)) \leq l\}, \quad (2.22)$$

où $\hat{\mathbf{V}}_l = \sum_{i=1}^l \hat{v}_i \mathbf{1}_{C_{\hat{v}_i}}(\mathbf{V})$ et $C_{\hat{v}_i}$ la partition de Voronoi induite par \hat{v}_i et définie par $C_{\hat{v}_i} = \{v \in \mathcal{V} \mid \forall h \in \{1, \dots, l\}, \|v - \hat{v}_i\| \leq \|v - \hat{v}_h\|\}$.

Le cas des processus gaussiens. [Pages and Printems, 2009] proposent différents algorithmes pour obtenir des quantifieurs optimaux dans le cas du mouvement brownien ou encore du pont brownien.

Théorème 4 ([Luschgy and Pagès, 2002]). Soit \hat{A}_l un quantifieur optimal de \mathbf{V} , $U = \text{span}(\hat{A}_l)$ et $m = \text{dim}(U)$.

Alors $\mathbb{E}[\|\mathbf{V} - \Pi_U(\mathbf{V})\|^2] = \sum_{i \geq m+1} \lambda_i$ où Π_U est la projection orthogonale de \mathcal{H} sur U et $\lambda_1 \geq \lambda_2 \geq \dots > 0$ sont les valeurs propres de l'opérateur associé à la fonction de covariance C . On en déduit la représentation finale du carré de l'erreur de quantification :

$$\begin{aligned} D_l^{\mathbf{Y}}(\hat{A}_l) = D_l(\mathbf{V}) &= \sum_{i \geq m+1} \lambda_i + D_l\left(\bigotimes_{j=1}^m \mathcal{N}(0, \lambda_j)\right) \quad \text{pour } m \geq d_l(\mathbf{V}), \\ D_l^{\mathbf{Y}}(\hat{A}_l) = D_l(\mathbf{V}) &< \sum_{i \geq m+1} \lambda_i + D_l\left(\bigotimes_{j=1}^m \mathcal{N}(0, \lambda_j)\right) \quad \text{pour } 1 \leq m < d_l(\mathbf{V}), \end{aligned} \quad (2.23)$$

où $d_l(\mathbf{V})$ désigne la dimension de la quantification au niveau l pour le problème de quantification de \mathbf{V} . Ainsi on écrit :

$$D_l^{\mathbf{Y}}(\hat{A}_l) = D_l(\mathbf{V}) = \sum_{i \geq d_l(\mathbf{V})+1} \lambda_i + D_l\left(\bigotimes_{j=1}^{d_l(\mathbf{V})} \mathcal{N}(0, \lambda_j)\right) \quad (2.24)$$

Une conséquence directe de ce théorème est la possibilité de définir la dimension de l'espace tronqué pour obtenir le quantifieur optimal pour \mathbf{V} en fonction de la taille souhaitée.

	$d_l(\mathbf{V})$	$(\mathbb{E}\ \mathbf{V} - \hat{\mathbf{V}}_l\ ^2)^{1/2}$
$l = 10$	2	0.0921
$l = 50$	3	0.0558
$l = 100$	4	0.0475
$l = 500$	5	0.0353

TABLE 2.1: Les erreurs de quantification en fonction de la taille du quantifieur optimal pour le mouvement brownien.

L'algorithme de Lloyd. Dans le cas des processus non gaussiens, [Miranda and Bocchini, 2013] proposent une application de l'algorithme de Lloyd directement sur les courbes discrétisées, comme dans le cas vectoriel. Cependant, cette méthode nécessite de savoir simuler la variable aléatoire fonctionnelle \mathbf{V} .

Dans le cadre de notre travail, seul un échantillon de réalisations de \mathbf{V} est mis à notre disposition. Notre objectif est d'estimer l'espérance par quantification gloutonne. Nos deux approches s'appuient sur une réduction de dimension.

2.2.2.2 Constructions gloutonnes

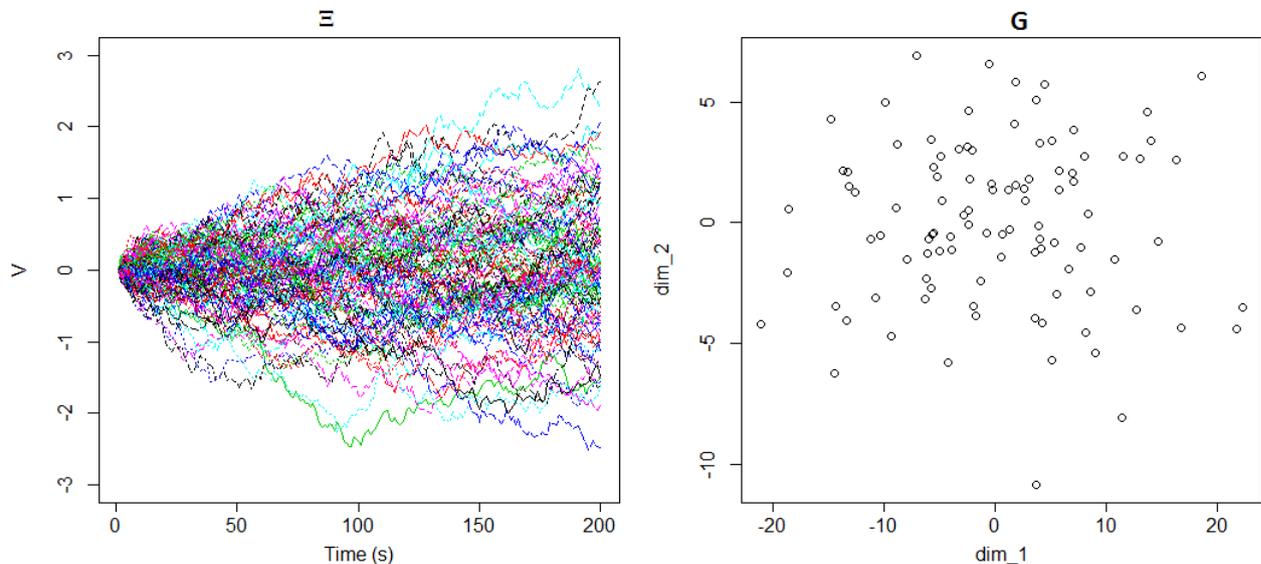


FIGURE 2.3: L'ensemble de 100 réalisations Ξ de la variable \mathbf{V} (à gauche), la projection de l'ensemble Ξ sur les deux premières composantes principales $m = 2$ (à droite).

Une première approche consiste à faire une quantification gloutonne dans l'espace tronqué des coefficients de la décomposition KL (cf. Figure 2.3). Nous nous sommes donc ramenés au cas vectoriel (voir [Luschgy and Pagès, 2015]). D'après l'équation (2.21), on a \mathbf{U} une variable aléatoire discrète dans $\mathcal{G} = \{(\langle v, \psi_1 \rangle, \dots, \langle v, \psi_m \rangle)^\top, v \in \Xi\} = \{u_i\}_{i=1}^N$, avec $u_i = (\langle v_i, \psi_1 \rangle, \dots, \langle v_i, \psi_m \rangle)^\top$ et \mathbf{U}_l est une l -quantification induite par $\{\hat{u}_1, \dots, \hat{u}_{l-1}\} \cup \{u\}$.

L'algorithme est ainsi donné par

$$\begin{aligned}
\hat{D}_1 &= \{\hat{u}_1\} \text{ où } \hat{u}_1 \text{ est solution de } D_1^{\mathbf{U}}(u) \\
\forall l \geq 2, \hat{D}_l &= \hat{D}_{l-1} \cup \{\hat{u}_l\} \\
\text{où } \hat{u}_l &\in \arg \min_{\mathbf{u} \in \mathcal{G}} (\mathbb{E} \|\mathbf{U} - \mathbf{U}_l\|^2)^{1/2}, \\
l &= l + 1.
\end{aligned} \tag{2.25}$$

La deuxième étape consiste à récupérer les courbes correspondantes aux points sélectionnés par l'algorithme (cf. Figure 2.4).

$$\hat{D}_l = \{\hat{u}_i\}_{i=1}^l \subset \mathcal{G} \longrightarrow \hat{A}_l = \{\hat{v}_i\}_{i=1}^l \subset \Xi$$

La figure 2.4 montre bien que la quantification cherche à capter le comportement moyen de la variable \mathbf{V} , puisqu'elle commence par sélectionner les points centraux et par la suite les points sur les bords.

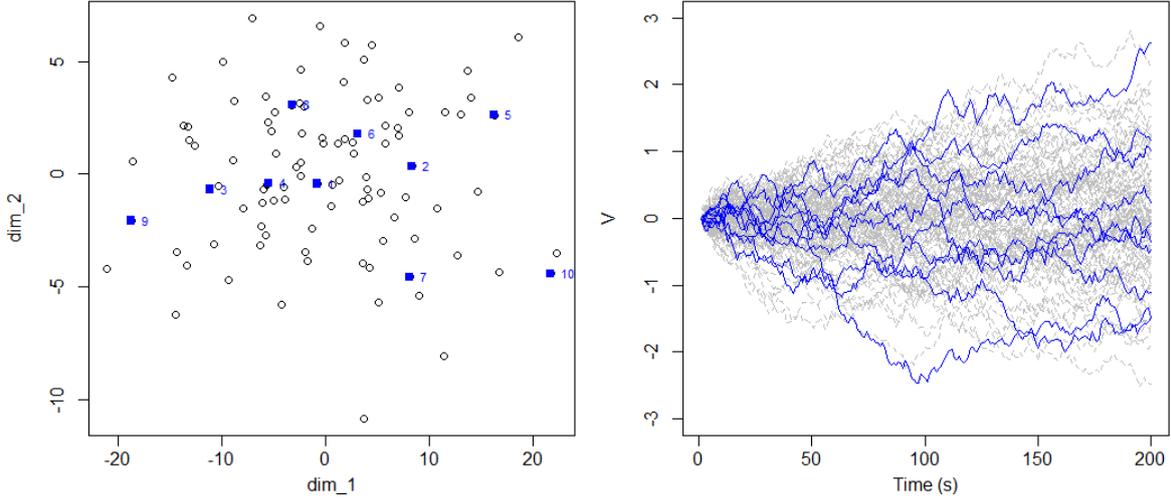


FIGURE 2.4: Le quantifieur de taille 10 obtenu à partir de l'ensemble \mathcal{G} (à gauche), le quantifieur associé dans l'ensemble Ξ (à droite).

On propose une deuxième construction gloutonne qui s'appuie sur une notion de remplissage optimale de l'espace. Un des plans d'expériences connus ayant de bonnes propriétés de remplissage de l'espace est le plan "maximin". La configuration est obtenue en maximisant la plus petite distance entre deux points du plan. Ici le but est de réutiliser ce critère (plus petite distance entre deux points) pour la quantification c'est à dire pour sélectionner les l meilleurs points parmi les N disponibles. Ainsi, un plan **maximin** est un plan qui maximise $\phi_{Mm}(D) = \min_{i \neq j} |u_i - u_j|$, où $D = \{u_1, \dots, u_l\}$.

$$\begin{aligned}
\text{Initialisation : } \hat{D}_1 &= \{\hat{u}_1\} \text{ où } \hat{u}_1 \text{ est choisi aléatoirement de } \mathcal{G} \\
\forall l \geq 2, \hat{D}_l &= \hat{D}_{l-1} \cup \{\hat{u}_l\} \\
\text{où } \hat{u}_l &\in \arg \max_{u \in \mathcal{G}} \phi_{Mm}(\hat{D}_{l-1} \cup \{u\}), \\
l &= l + 1.
\end{aligned} \tag{2.26}$$

L'algorithme est initialisé en un point choisi aléatoirement de l'ensemble \mathcal{G} , le point suivant est sélectionné afin de maximiser le critère maximin. En plus de la séquentialité, les points sont choisis afin d'explorer au mieux la variabilité de \mathbf{V} comme le montre la figure 2.5.

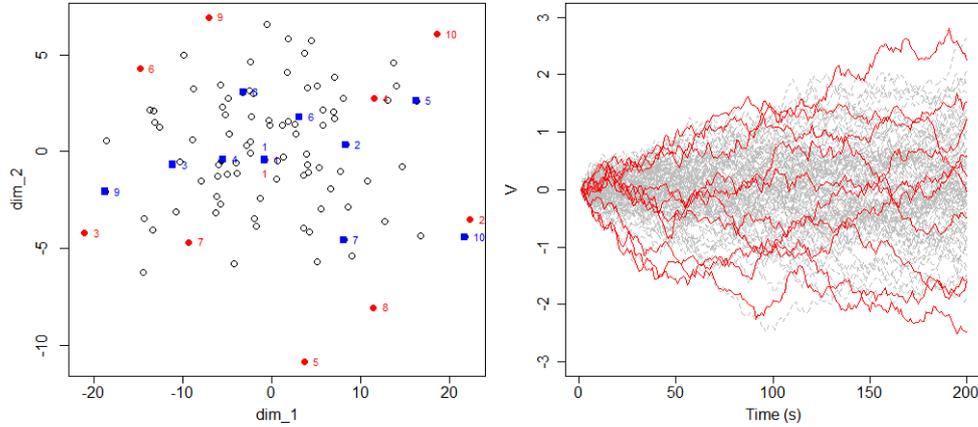


FIGURE 2.5: Les quantifieurs de taille 10 obtenus à partir de l'ensemble \mathcal{G} par la méthode maximin (en rouge) et en réduisant l'erreur de quantification (en bleu) (à gauche), le quantifieur associé dans l'ensemble Ξ par la méthode maximin (à droite).

Remarque 1. On précise que les points obtenus par les constructions gloutonnes présentées dans cette partie sont solutions d'optimisation discrète. A chaque itération, l'algorithme 2.25 calcule pour chaque point candidat une moyenne empirique en considérant les N points à disposition, et donc il a une complexité $O(l \times N^2 \times m)$. Contrairement à l'algorithme 2.26 qui a une complexité égale à $O(l \times N \times m)$. La figure 2.6 montre le temps de calcul des deux algorithmes en fonction de la taille du quantifieur pour $N = 200$.

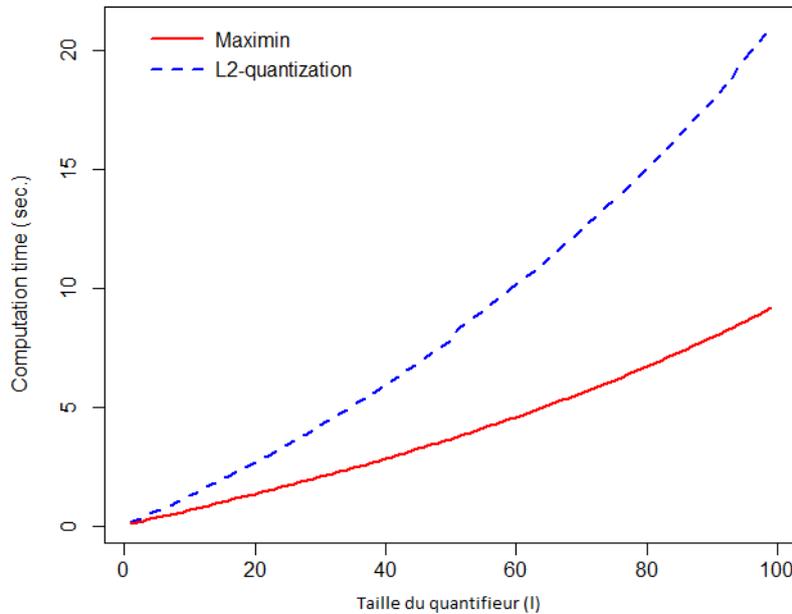


FIGURE 2.6: Temps de calcul en fonction de la taille du quantifieur par l'algorithme 2.25 en bleu et l'algorithme 2.26 en rouge.

Le temps de calcul de l'algorithme 2.25 peut être réduit en s'appuyant sur la capacité de stocker le calcul effectué à l'étape $l - 1$. En effet si on développe l'erreur de quantification à l'itération l , on obtient :

$$\begin{aligned} \frac{1}{N} \sum_{i=1}^N \min_{1 \leq j \leq l} |u_i - u_j|^2 &= \frac{1}{N} \sum_{i=1}^N |u_i|^2 + \frac{1}{N} \sum_{i=1}^N \underbrace{\min_{1 \leq j \leq l} \left(|u_j|^2 - 2\langle u_j, u_i \rangle \right)}_{\Upsilon_{ij}}, \\ &= \underbrace{\frac{1}{N} \sum_{i=1}^N |u_i|^2}_{\text{stocké}} + \frac{1}{N} \sum_{i=1}^N \min \left(\Upsilon_{il}, \underbrace{\min_{1 \leq j \leq l-1} \Upsilon_{ij}}_{\text{stocké}} \right). \end{aligned}$$

Ainsi le calcul des différences des normes se ramène à des produits scalaires. Le premier terme est calculé une seule fois alors que le second se résume à une somme de $i = 1, \dots, N$ du minimum entre Υ_{il} et le $\min_{1 \leq j \leq l-1} \Upsilon_{ij}$ préalablement effectué et stocké.

Application à l'estimation de l'espérance. Une fois la grille \hat{A}_l obtenue par une des deux constructions proposées, on définit une l -quantification de la variable aléatoire fonctionnelle \mathbf{V} par

$$\hat{\mathbf{V}}_l = \sum_{i=1}^l \hat{v}_i \mathbf{1}_{C_{\hat{v}_i}}(\mathbf{V}), \quad (2.27)$$

où $\{C_{\hat{v}_i}(\hat{A}_l) : \hat{v}_i \in \hat{A}_l\}$ est la partition de Voronoi de Ξ induite par \hat{A}_l . Ainsi il est naturel d'approcher $\mathbb{E}[h(\mathbf{V})]$ par $\mathbb{E}[h(\hat{\mathbf{V}}_l)]$

$$\mathbb{E}[h(\hat{\mathbf{V}}_l)] = \sum_{i=1}^l h(\hat{v}_i) \mathbb{P}(\hat{\mathbf{V}}_l = \hat{v}_i) \quad (2.28)$$

où la distribution $(\mathbb{P}(\hat{\mathbf{V}}_l = \hat{v}_i))_{i=1:l}$ de $\hat{\mathbf{V}}_l$ peut être approchée empiriquement par $(\text{card}(C_{\hat{v}_i}(\hat{A}_l) \cap \Xi) / \text{card}(\Xi))_{i=1:l}$.

2.3 Applications

Les deux méthodes gloutonnes présentées dans ce chapitre, et résumées par l'algorithme 2, seront appliquées à deux exemples analytiques. Ces derniers seront comparés à l'approche Monte Carlo.

Algorithm 2 Intégration numérique gloutonne par **maximin** ou par **quantification**

- 1: **Inputs** : Échantillon initial (Ξ) , argument de troncature (m) et la taille du quantifieur (l) .
 - 2: $\mathcal{G} \leftarrow$ Réduction de la dimension par KL
 - 3: Construction de l'ensemble des points \tilde{D}_l soit par maximin 2.26 ou quantification 2.25
 - 4: Induire de \tilde{D}_l l'ensemble des courbes représentantes \hat{A}_l
 - 5: Définir $\hat{\mathbf{V}}_l \leftarrow \hat{A}_l$ -quantification de \mathbf{V} par l'équation 2.27
 - 6: Calculer l'espérance $\mathbb{E}[h(\hat{\mathbf{V}}_l)]$ par l'équation 2.28
-

Exemple analytique 1. On considère la fonction h définie comme suit

$$h : \mathbf{V} \mapsto (x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7) + \int_0^1 e^{\mathbf{V}^t} dt,$$

où $\mathbf{x} = (x_1, x_2) = (50, -80)$ et \mathbf{V} un mouvement brownien sur $[0, T] = [0, 1]$. On suppose que \mathbf{V} n'est connue qu'à travers un échantillon de $N = 200$ réalisations $\Xi = \{v_1, \dots, v_N\}$. L'argument de troncature m est fixé à 2, valeur pour laquelle 95% de la variance du processus est expliquée. On rappelle que la méthode gloutonne à base de quantification quadratique (\mathbb{L}^2 -GFQ) est bien déterministe. La méthode gloutonne à base de maximin (maximin-GFQ) est aléatoire par construction. En effet cette nature stochastique est due au choix aléatoire du point de départ. Pour prendre en compte cet aléa dans la comparaison de ces deux méthodes à l'estimation par Monte Carlo, on va répéter le test 200 fois.

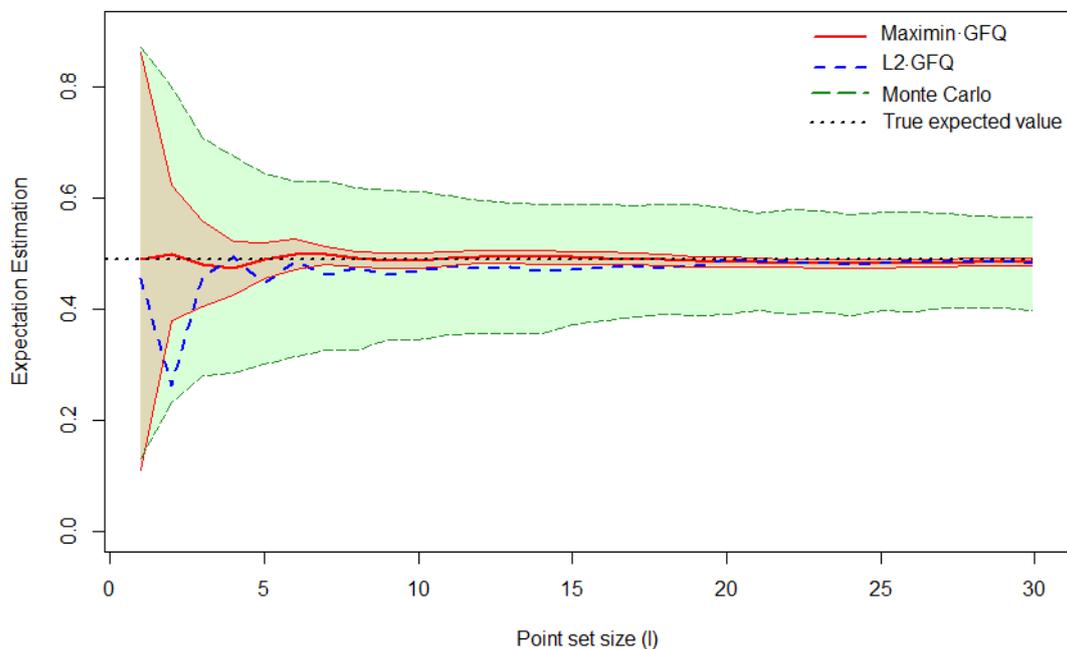


FIGURE 2.7: Exemple analytique 1. Estimation séquentielle de l'espérance en fonction du nombre de courbes. Les lignes représentent les moyennes. Les intervalles de confiance (Monte Carlo (en vert) and maximin-GFQ (en rouge)).

La figure 2.7 montre le résultat des tests. On observe que quel que soit l , l'erreur d'intégration induite par les deux méthodes gloutonnes est significativement très faible par rapport à la méthode Monte Carlo. Pour tout $l \geq 25$, on remarque que la méthode maximin-GFQ est moins sensible au point de départ. On note aussi que pour un faible nombre de courbes $l \leq 5$, la méthode maximin-GFQ dépend beaucoup du point initial pour les faibles valeurs de l , ce qui induit une grande incertitude dans l'estimation. A partir de $l \geq 10$, l'estimation est plus stable puisque les courbes centrales et extrêmes sont sélectionnées (voir figure 2.5).

Exemple analytique 2. Comme deuxième exemple, on considère h une fonction définie comme suit

$$h : \mathbf{V} \mapsto \max_t \mathbf{V}_t \cdot |0.1 \cos(x_1 \max_t \mathbf{V}_t) \sin(x_2) \cdot (x_1 + x_2 \min_t \mathbf{V}_t)^2| \cdot \int_0^T (30 + \mathbf{V}_t)^{-\frac{x_1 \cdot x_2}{20}} dt,$$

where $\mathbf{x} = (x_1, x_2) = (2.95, 3.97)$ et comme pour l'exemple précédent, on suppose que \mathbf{V} est un mouvement brownien et que cette variable n'est connue qu'à travers un échantillon de courbes Ξ de taille $N = 200$. On fixe l'argument de troncature à $m = 2$ et pour comparer les deux approches gloutonnes à l'estimation par Monte Carlo, on va répéter le test 30 fois.

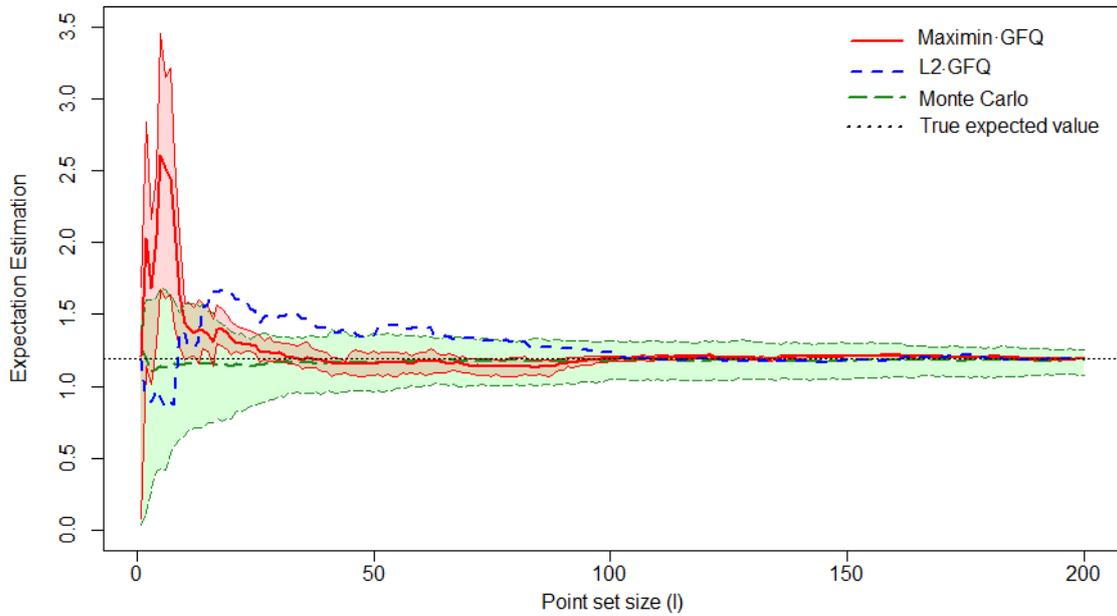


FIGURE 2.8: Exemple analytique 2. Estimation séquentielle de l'espérance en fonction du nombre de courbes. Les lignes représentent les moyennes. Les intervalles de confiance (Monte Carlo (en vert) and maximin-GFQ (en rouge)).

La figure 2.8 conduit à faire deux commentaires : les deux méthodes gloutonnes proposées font mieux en terme d'erreur d'intégration par rapport à Monte Carlo. A partir de $l \geq 20$, la méthode maximin-GFQ est moins sensible au point de départ dans la construction.

On rappelle que nos deux méthodes sont basées sur une réduction de dimension. Puisque on récupère les courbes et ainsi leurs structures, on espère robustifier nos approches par rapport à cette approximation. Pour voir l'impact de la réduction de dimension sur l'estimation de l'espérance, on va comparer nos deux approches à la méthode introduite dans la section 2.2.1. La figure 2.9 montre le résultat après avoir répété le test 30 fois. On remarque que la méthode "Fpca" mène à une estimation biaisée de l'espérance. Ce résultat attendu est expliqué par la perte d'information dû à la réduction de dimension.

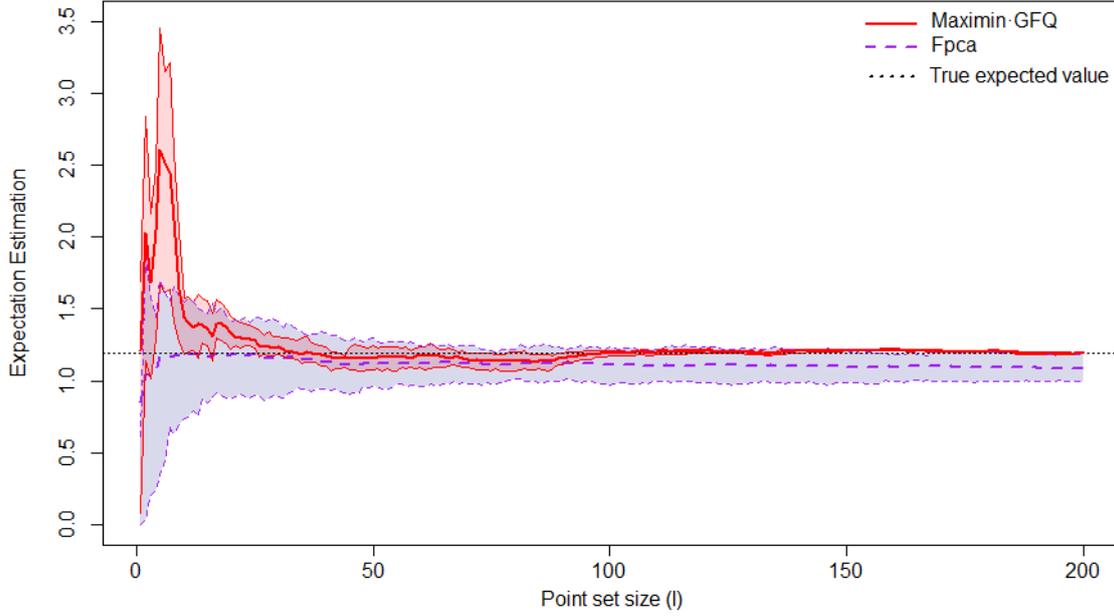


FIGURE 2.9: Exemple analytique 2. Estimation séquentielle de l'espérance en fonction du nombre de courbes. Les lignes représentent les moyennes. Les intervalles de confiance (Monte Carlo (en mauve) and maximin-GFQ (en rouge)).

Synthèse : L'objectif de ce chapitre est de proposer deux nouvelles méthodes pour l'estimation séquentielle de l'espérance. Ces méthodes sont basées sur une quantification fonctionnelle gloutonne. En guise de préliminaire et dans un cadre vectoriel, on a présenté les approches standards pour l'estimation de l'espérance, comme les méthodes de Monte-Carlo ou les méthodes impliquant la quantification vectorielle. Par la suite, on a présenté l'extension de la notion de quantification ainsi que les différents algorithmes pour la construction des quantifieurs au cadre fonctionnel. On a défini deux nouvelles méthodes pour construire de manière gloutonne des quantifieurs emboîtés. Ces deux méthodes s'appuient sur une réduction de dimension ainsi que sur les deux notions suivantes :

- la notion de quantification vectorielle,
- le remplissage optimal de l'espace.

A travers deux exemples analytiques, on a mis en exergue la robustesse de ces deux méthodes par rapport aux approches standards.

L'objectif du chapitre suivant est d'apporter une solution au problème d'inversion suivant :

$$\Gamma^* := \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})] \leq c\}, \quad (2.29)$$

où $c \in \mathbb{R}$ est un seuil prédéfini. La méthodologie proposée combine les outils des chapitres 1 et 2.

Chapitre 3

Data-driven stochastic inversion via functional quantization

Outlines

L'objectif de ce chapitre est de proposer une méthodologie pour tenir compte des incertitudes fonctionnelles dans la résolution des problèmes d'inversion. Cette méthode est une combinaison des outils présentés dans les chapitres 2 et 1.

Les résultats de ce chapitre ont donné lieu à un article soumis pour publication [El Amri et al., 2018]. L'objectif de ce chapitre est d'apporter une réponse à la problématique soulevée en introduction portant sur la résolution d'un problème d'inversion en présence d'incertitudes fonctionnelles. Ce chapitre étant un article complet, certains résultats sont similaires au chapitre 2.

Dans ce chapitre, nous proposons une nouvelle méthodologie pour traiter une entrée fonctionnelle incertaine dans les problèmes d'inversion. Dans notre contexte, le code prend en entrée deux types paramètres : les variables de contrôle déterministes et une variable aléatoire fonctionnelle. On note que le code étudié est caractérisé par deux éléments : le premier est le coût de calcul élevé des simulations, ce qui rend l'inversion en présence d'incertitudes coûteuse avec les approches de la littérature. Le deuxième élément est que la densité de probabilité de l'entrée fonctionnelle \mathbf{V} n'est pas connue, seul un échantillon de réalisations est à notre disposition. La méthode proposée implique deux tâches imbriquées. Une première tâche basée sur une approche bayésienne vise à choisir judicieusement les nouveaux points de l'espace de contrôle (cf. chapitre 1), où l'espérance sera évaluée, qui permettront d'estimer l'ensemble d'excursion en le minimum d'itérations. La deuxième tâche vise à estimer efficacement l'espérance par rapport à la variable aléatoire fonctionnelle en chaque point de l'étape précédente (cf. chapitre 2). Notre méthode est illustrée et calibrée sur un exemple analytique. Il est ensuite appliqué au cas SCR où l'objectif est d'identifier l'ensemble des paramètres de contrôle conduisant à respecter les normes d'émissions de polluants.

Contents

3.1	Introduction	40
3.2	Functional data reduction	41
3.2.1	Context	41
3.2.2	The Karhunen-Loève expansion	43
3.2.3	Greedy Functional Quantization	43
3.3	Numerical integration	46
3.4	Background on SUR strategies	51
3.4.1	Random closed set and bayesian framework	52

3.4.2	Vorob'ev approach	52
3.4.3	SUR strategies	53
3.4.4	SUR strategy adapted to noisy observations	54
3.5	Algorithm coupling SUR and functional quantization	54
3.6	Numerical tests	55
3.6.1	Analytical example	56
3.6.2	IFPEN test case	62
3.7	Conclusions	65
3.8	Complément de l'article : un nouvel exemple	65

Title : Data-driven stochastic inversion via functional quantization.

Auteur : Mohamed Reda El Amri^{¶,§}, Céline Helbert[†], Olivier Lepreux[‡], Miguel Munoz Zuniga[¶], Clémentine Prieur[§], Delphine Sinoquet[¶]

¶ IFPEN, Rueil-Malmaison, France

§ Université Grenoble Alpes, Laboratoire Jean Kuntzmann, Grenoble, France

† Université de Lyon, UMR 5208, École Centrale de Lyon, Institut Camille Jordan

‡ IFPEN, Lyon, France

3.1 Introduction

In recent years, computer models are omnipresent in engineering and sciences, because the corresponding physical experimentation is costly or impossible to execute. Indeed, numerical simulations are often used to replace physical experiment which can be costly or even impossible to perform as underlined in [Bect et al., 2012, Chevalier et al., 2014a]. Practitioners are not only interested in the response of their model for a given set of inputs (forward problem) but also in recovering the set of input values leading to a prescribed value or range for the output of interest. The problem of estimating such a set is called hereafter inversion problem. We will consider a system that evolves in an uncertain environment, the uncertainties appear for example due to manufacturing tolerances or environmental conditions. The numerical simulator modelling the system, denoted f , takes two types of input variables : a set of control variables $\mathbf{x} \in \mathbb{X}$, and a set of uncertain variables $\mathbf{v} \in \mathcal{V}$. Given any distributional assumptions for the uncertain variable \mathbf{v} , robust inversion consists in seeking the set of control variables $\mathbf{x} \in \mathbb{X}$ such that $\sup_{\mathbf{v} \in \mathcal{V}} f(\mathbf{x}, \mathbf{v})$ is smaller than a threshold c . Then, the difficulty of solving the robust inversion problem strongly depends on the uncertainty set \mathcal{V} . In our setting, \mathcal{V} is a functional space, and we consider instead the inversion problem under uncertainty as a stochastic inversion problem, assuming that the uncertainty has a probabilistic description. Let \mathbf{V} denote the associated random variable, valued in \mathcal{V} , modelling the uncertainty. In our framework, we are interested in recovering the set $\Gamma^* := \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})] \leq c\}$, with $c \in \mathbb{R}$, and the functional random variable \mathbf{V} is only known from a set of realizations and the expectation has to be estimated. Moreover, the simulations are time consuming and thus the usual Monte Carlo method to estimate the expectation ought to be avoided. Many reviews have been published to address this issue. Among the numerous techniques, the paper will focus on the ones based on the choice of a finite representative set of realizations of \mathbf{V} , among the ones available. More precisely these approaches aim at minimizing the expected distance between a random draw from the probability distribution of \mathbf{V} and this finite set. In the case of vector-valued random variables, this type of methods, also known as principal points and

introduced in [Flury, 1990], was employed in various statistical applications, including quantizer design [Pagès, 2014] and stratified sampling. It is increasingly used for many engineering applications, where we are often faced with the challenge of working with *big data*. It is then necessary to reduce big data to manageable data. In the case of functional random variables, various studies have been done in the Gaussian case (see [Pagès and Printems, 2005] and references therein). Here we work in the special case where the functional variable \mathbf{V} could be non Gaussian and the latter is known through a finite sample of realizations. This paper propose two new methodologies to perform this data reduction, or quantization, for functional random variable and we investigate their performance in terms of integration errors.

Inversion problems have already been carried out in many applications, notably reliability engineering (see, e.g., [Bect et al., 2012], [Chevalier et al., 2014a]), climatology (see, e.g., [Bolin and Lindgren, 2015], [French et al., 2013]) and many other fields. In the literature, one way to solve the problem is to adopt a sequential sampling strategy based on Gaussian process emulators. The underlying idea is that Gaussian process emulators, which capture prior knowledge about the regularity of the unknown function $g : \mathbf{x} \mapsto \mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})]$, make it possible to assess the uncertainty about Γ^* given a set of evaluations of g . More specifically, these sequential strategies for the estimation of an excursion set are closely related to the field of Bayesian global optimization (see, e.g., [Chevalier and Ginsbourger, 2013]). In the case of inversion problems, Stepwise Uncertainty Reduction (SUR) strategies based on set measures were introduced in [Vazquez and Bect, 2009]. More recently, a parallel implementation of these strategies has been proposed in [Chevalier et al., 2014a] and applied to the problem of recovery of an excursion set. Briefly, the strategy SUR gives sequentially the next location in the control space where to estimate the function g in order to minimize an uncertainty function. The key contribution of the present paper is to propose a data-driven adaptation of that procedure in the presence of functional uncertainties.

The paper is divided into five sections. Following this introduction, Section 3.2 is devoted to the introduction of two new adaptive methods to choose the finite representative set of the functional random variable for a reliable expectation estimation. In Section 3.3, We highlight the integration performance of our methods comparing to the standard Monte Carlo and to an existing method based on a probabilistic modelling with truncated principal component analysis (PCA). In Section 3.4, we introduce the Bayesian framework and fundamental notions for Stepwise Uncertainty Reduction (SUR) infill strategies in the context of computationally costly simulations. In Section 3.5, we introduce the new proposed data-driven methodology for stochastic inversion under functional uncertainties and describe our algorithm. Finally, in Section 3.6, we illustrate the overall procedure on an analytical example and then apply it to an industrial test case.

3.2 Functional data reduction

In this section, we introduce new data reduction strategies for functional data in a greedy fashion. The first one is based on notion coming from functional quantization. The second one is related to notion of space filling design. In this paper, data reduction aims at reducing the integration error when computing $\mathbb{E}[f(\mathbf{x}, \mathbf{V})]$. Therefore, we focus in Section 3.3 on the performance in terms of integration error of our strategies, comparing to standard procedures.

3.2.1 Context

We consider the space $\mathcal{H} = \mathbb{L}^2(\Omega, \mathcal{F}, \mathbb{P}; \mathcal{V})$ of random processes \mathbf{V} with realizations $\mathbf{V}(\cdot, \omega) = \mathbf{v}$ in the space of deterministic square-integrable functions defined on $[0, T]$ denoted

with $\mathcal{V} = \mathbb{L}^2([0, T])$ and equipped with norm $\|\mathbf{v}\| = (\int_0^T \mathbf{v}(t)^2 dt)^{1/2}$. The random variables $\mathbf{V}(t, \cdot) = \eta$ lie in $\mathbb{L}^2(\Omega, \mathbb{P})$, the space of random variables with finite mean and variance, defined on $(\Omega, \mathcal{F}, \mathbb{P})$ and equipped with norm $\|\eta\|_{\mathbb{L}^2(\Omega, \mathbb{P})} = (\int_{\Omega} \eta^2 d\mathbb{P})^{1/2}$. All random processes discussed in this paper lie in \mathcal{H} which is equipped with norm

$$\|\mathbf{V}\|_{\mathbb{L}^2} = (\mathbb{E}[\|\mathbf{V}\|^2])^{1/2} = \left(\mathbb{E} \left[\int_0^T \mathbf{V}(t)^2 dt \right] \right)^{1/2}, \quad (3.1)$$

for any $\mathbf{V} \in \mathcal{H}$. The vast majority of realistic engineering problems can be addressed within this set of assumptions. Without loss of generality, we will consider a centred stochastic process with finite variance. We aim at summarizing the distribution of \mathbf{V} through a finite collection of deterministic functions $\{\mathbf{v}_j\}_{j=1}^l$ and corresponding weights $\{w_j\}_{j=1}^l$. Many reviews have been done on *functional quantization* [Pagès and Printems, 2005, Luschgy et al., 2010, Pages and Printems, 2009]. For instance, Luschgy, Pagès and Wilbertz [Luschgy et al., 2010] propose different strategies for Gaussian processes.

An optimal quantization of \mathbf{V} consists in finding the subset $A \subset \mathcal{V}$ with $\text{card}(A) \leq l$ that minimizes

$$\left\| \min_{\mathbf{a} \in A} \|\mathbf{V} - \mathbf{a}\| \right\|_{\mathbb{L}^2} = \left(\mathbb{E} \left[\min_{\mathbf{a} \in A} \|\mathbf{V} - \mathbf{a}\|^2 \right] \right)^{1/2}. \quad (3.2)$$

Such a set is called an optimal *l-quantizer*. Lets us denote it as $A = \{\mathbf{a}_1, \dots, \mathbf{a}_l\}$. We define a neighbour projection associated to A as :

$$\pi_A := \sum_{i=1}^l \mathbf{a}_i \mathbf{1}_{C_{\mathbf{a}_i}(A)}, \quad (3.3)$$

where $\forall i \in \{1, \dots, l\}$ $C_{\mathbf{a}_i}(A)$ is the Voronoi partition induced by A and associated with \mathbf{a}_i :

$$C_{\mathbf{a}_i}(A) = \{\mathbf{v} \in \mathcal{V} \mid \forall h \in \{1, \dots, l\}, \|\mathbf{v} - \mathbf{a}_i\| \leq \|\mathbf{v} - \mathbf{a}_h\|\}. \quad (3.4)$$

The *A-quantization* of \mathbf{V} is defined by :

$$\hat{\mathbf{V}}_l := \pi_A(\mathbf{V}). \quad (3.5)$$

Voronoi partition is optimal in the sense that, for any random variable $\mathbf{V}'_l : \Omega \rightarrow A$, one has $\mathbb{E}\|\mathbf{V} - \mathbf{V}'_l\|^2 \geq \mathbb{E}\|\mathbf{V} - \hat{\mathbf{V}}_l\|^2$ (see, [Pages and Printems, 2009]). Finally the *l-quantization* error of \mathbf{V} is defined by

$$e_l(\mathbf{V}) = \inf \{ (\mathbb{E}\|\mathbf{V} - \hat{\mathbf{V}}_l\|^2)^{1/2}, \hat{\mathbf{V}}_l : \Omega \rightarrow \mathcal{V}, \text{card}(\hat{\mathbf{V}}_l(\Omega)) \leq l \}. \quad (3.6)$$

From a computational point of view, the cost of minimizing the error defined in Eq. (3.6) is not negligible. Even in the finite-dimensional space \mathbb{R}^m , the numerical search of an optimal solution leads to an increasing computational cost when l or m grows (see [Levrard, 2014]). Luschgy, Pagès propose in [Luschgy and Pagès, 2015] a greedy version of the \mathbb{L}^2 -quantization problem for \mathbf{U} an \mathbb{R}^m valued random vector. The greedy-quantization is easier to compute in terms of complexity but provides a possible sub-optimal quantizer $\{\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_l\}$. The authors in [Luschgy and Pagès, 2015] prove that the \mathbb{L}^2 -quantization error at level l induced by $\{\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_l\}$ goes to 0 at rate $l^{-1/m}$. The idea of such a procedure is to determine sequentially the sequence $(\hat{\mathbf{u}}_l)_{l \geq 1}$. The first vector $\hat{\mathbf{u}}_1$ achieves the error $e_1(\mathbf{U})$. Then, for $l \geq 2$,

$$\forall l \geq 2, \quad \hat{\mathbf{u}}_l \in \arg \min_{\mathbf{u} \in \mathbb{R}^m} (\mathbb{E}\|\mathbf{U} - \hat{\mathbf{U}}_l\|^2)^{1/2}, \quad (3.7)$$

where $\hat{\mathbf{U}}_l$ is the l -quantization induced by $\{\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_{l-1}\} \cup \{\mathbf{u}\}$.

In the present work, we propose a sequential strategy in an infinite-dimensional setting under the assumption that the random process \mathbf{V} may not be Gaussian. In this framework, Miranda and Bocchini [Miranda and Bocchini, 2013, Miranda and Bocchini, 2015] propose a one-shot algorithm that produces an optimal functional quantizer but which depends on a simulation procedure for \mathbf{V} . In the following, we propose a greedy algorithm to compute a l -quantization of \mathbf{V} . In our framework, the functional random variable is only known through a finite set of realizations. The specificity of our procedure is first that it does not require a simulation algorithm of the unknown process \mathbf{V} (which is known only from a finite set of realizations), and secondly our quantizer can be sequentially increased in a greedy fashion. One ingredient in our methodology is the PCA decomposition of \mathbf{V} (also known as Karhunen-Loève expansion). Let us briefly recall the Karhunen-Loève expansion which is the most commonly employed method to reduce the statistical complexity of random fields indexed over bounded intervals, with continuous covariance function.

3.2.2 The Karhunen-Loève expansion

Let $\mathbf{V} \in \mathcal{H}$ be a random process with zero mean and continuous covariance function $C(t, s)$. Then

$$\mathbf{V}(t) = \sum_{i=1}^{\infty} u_i \psi_i(t), \quad t \in [0, T], \quad (3.8)$$

where $\{\psi_i\}_{i=1}^{\infty}$ are orthogonal and normalized eigenfunctions of the integral operator corresponding to C :

$$\lambda_i \psi_i(t) = \int_0^T C(t, s) \psi_i(s) ds. \quad (3.9)$$

The $\{u_i\}_{i=1}^{\infty}$ denotes a set of orthogonal random variables with zero mean and variance λ_i , where λ_i is the eigenvalue corresponding to the eigenfunction ψ_i . Equation (3.8) can be truncated after m -terms :

$$\mathbf{V}(t) \simeq \sum_{i=1}^{m_{KL}} u_i \psi_i(t). \quad (3.10)$$

Computational method for functional PCA The covariance structure of the process is unknown and has to be estimated from the data. More precisely, $C(s, t)$ is estimated from $\Xi = \{\mathbf{v}_i\}_{i=1}^N$ by :

$$C^N(t, s) = \frac{1}{N} \sum_{i=1}^N \mathbf{v}_i(s) \mathbf{v}_i(t). \quad (3.11)$$

The eigenvalue problem defined by Eq. (3.9) is then solved by replacing C by C^N (see, e.g., [Cardot et al., 1999] for convergence results). That approximated eigenvalue problem is solved, e.g., by discretizing the trajectories $\{\mathbf{v}_i\}_{i=1, \dots, N}$ on $[0, T] : \{v_i(t_j)\}_{i=1, \dots, N}^{j=1, \dots, N_T}$ (see, e.g., [Ramsay, 2006] for a detailed overview on the subject).

3.2.3 Greedy Functional Quantization

L²-Greedy Functional Quantization (L²-GFQ). Now we aim at optimally exploring the range of variations of \mathbf{V} with a few elements in Ξ . Let

$\mathcal{G} = \{(\langle \mathbf{v}, \psi_1 \rangle, \dots, \langle \mathbf{v}, \psi_{m_{KL}} \rangle)^\top, \mathbf{v} \in \Xi\} = \{\mathbf{u}_i\}_{i=1}^N$, with $\mathbf{u}_i = (\langle \mathbf{v}_i, \psi_1 \rangle, \dots, \langle \mathbf{v}_i, \psi_{m_{KL}} \rangle)^\top$, be the set of the first m coefficients in KL expansion (see Fig. 3.1). Since we place ourselves in a finite space $\mathcal{G} \subset \mathbb{R}^{m_{KL}}$, a first step consists in an efficient and sequential strategy for the selection of a vectorial l -quantizer. A first solution is to apply the greedy

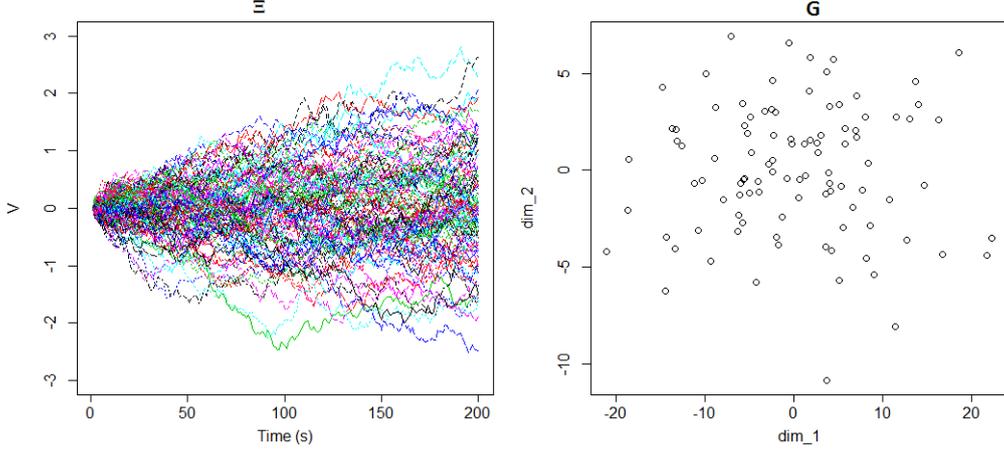


FIGURE 3.1: Ξ is a sample of 100 realizations of \mathbf{V} (left) and \mathcal{G} the corresponding representation in the truncated space of coefficients with $m_{KL} = 2$ (right).

vectorial quantization procedure described by Equation (3.7). The sequential construction is detailed below

$$\begin{aligned}
 \hat{D}_1 &= \{\hat{\mathbf{u}}_1\} \text{ where } \hat{\mathbf{u}}_1 \text{ is a solution of } e_1(\mathbf{U}) \text{ from Eq. (3.6)} \\
 \forall l \geq 2, \hat{D}_l &= \hat{D}_{l-1} \cup \{\hat{\mathbf{u}}_l\} \\
 \text{where } \hat{\mathbf{u}}_l &\in \arg \min_{\mathbf{u} \in \mathcal{G}} (\mathbb{E} \|\mathbf{U} - \mathbf{U}_l\|^2)^{1/2}, \\
 l &= l + 1,
 \end{aligned} \tag{3.12}$$

\mathbf{U} is a random vector with discrete uniform distribution on \mathcal{G} and \mathbf{U}_l is the l -quantization induced by $\{\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_{l-1}\} \cup \{\mathbf{u}\}$.

A collection of representative curves associated to our functional random variable \mathbf{V} is obtained by recovering the curves in the initial sample Ξ that correspond to the selected points \hat{D}_l :

$$\hat{D}_l = \{\hat{\mathbf{u}}_i\}_{i=1}^l \subset \mathcal{G} \longrightarrow \Theta'_l = \{\theta'_i\}_{i=1}^l \subset \Xi \tag{3.13}$$

This step is important as it allows to recover the functional variability of \mathbf{V} and not only the variability of its first m_{KL} coefficients. Figure 3.2 shows the algorithm up to step $l = 10$ on the example of Figure 3.1. Note that, from this figure, we can see that \mathbb{L}^2 -quantization leads to selecting central points and so the sample $\Theta'_{l=10}$ is mainly representative of the mean behaviour of \mathbf{V} . We also note that the extreme points are not chosen in the beginning of the construction. To address this issue, we propose a method that aims at exploring the range of variations of \mathbf{V} by selecting central points and also extreme points. One way to do so consists in the construction of a space filling design in $\mathbb{R}^{m_{KL}}$. Let \tilde{D} be this design. As the distribution of \mathbf{V} is only known through Ξ , it is natural to search \tilde{D} included in \mathcal{G} such that the chosen points are well spread all over the space. Let us recall the notion of space-filling design, from a purely model-free stance.

Space-Filling Design and criterion. Let us define $D = \{\mathbf{u}_1, \dots, \mathbf{u}_l\}$ a collection of l points. We denote by $\text{dist}_{ij} = \|\mathbf{u}_i - \mathbf{u}_j\|$ the euclidean distance between two design points \mathbf{u}_i and \mathbf{u}_j of D . One must then attempt to make the smallest distance between neighboring points in D as large as possible. We call a design that maximizes $\phi_{Mm}(D) = \min_{i \neq j} \text{dist}_{ij}$, a **maximin-distance design** (see Johnson et al [Johnson et al., 1990]). There are several other intrinsic criteria in literature such as discrepancy that measures whether the distribution of the

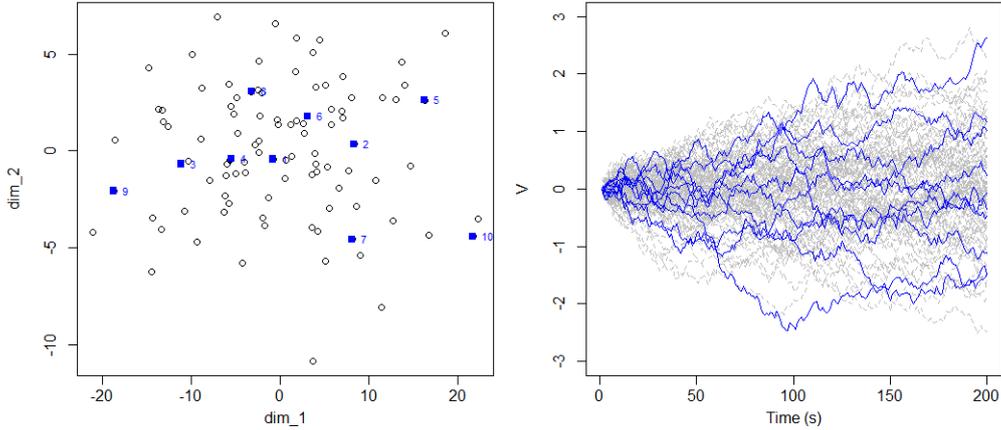


FIGURE 3.2: Sequential design of 10 points (bold points) in the set of the coefficients \mathcal{G} (left) and their corresponding bold curves in Ξ (right).

points of D is close to a uniform distribution. See Pronzato et al. [Pronzato and Müller, 2012] for a detailed overview on the subject. In the following, we consider the maximin-distance criterion to construct our design, and since we want to select points from the set of coefficients \mathcal{G} , the design \tilde{D} can be obtained by finding the design of l points among N , that maximizes the criterion ϕ_{Mm} .

Maximin-Greedy Functional Quantization (Maximin-GFQ). Finding the design \tilde{D} is a computationally difficult task. We could adapt one of the optimal design algorithms used in the literature such as simulated annealing (see Morris et al. [Morris and Mitchell, 1995]) and stochastic evolutionary algorithm (see Jin et al. [Jin et al., 2005]) for our purpose. Here we propose a one-point-at-time greedy algorithm for the generation of our design. The sequential construction is described below

$$\begin{aligned}
 &\text{Initialization : } \tilde{D}_1 = \{\tilde{\mathbf{u}}_1\} \text{ where } \tilde{\mathbf{u}}_1 \text{ is randomly chosen} \\
 &\forall l \geq 2 \text{ , } \tilde{D}_l = \tilde{D}_{l-1} \cup \{\tilde{\mathbf{u}}_l\} \\
 &\text{where } \tilde{\mathbf{u}}_l \in \arg \max_{\mathbf{u} \in \mathcal{G}} \phi_{Mm}(\tilde{D}_{l-1} \cup \{\mathbf{u}\}), \\
 &l = l + 1.
 \end{aligned} \tag{3.14}$$

The algorithm starts with a random point $\tilde{\mathbf{u}}_1$, the next point is chosen among the points in \mathcal{G} in order to maximize the maximin-distance criterion. Besides the sequentiality as for the \mathbb{L}^2 -GFQ method, the points are now chosen in order to explore the range of variations of \mathbf{V} at each step using a distance criterion. Meaning that at each step, the exploration of the domain is reasonable. The technique to recover the curves remains the same :

$$\tilde{D}_l = \{\tilde{\mathbf{u}}_i\}_{i=1}^l \subset \mathcal{G} \longrightarrow \Theta_l'' = \{\theta_i''\}_{i=1}^l \subset \Xi \tag{3.15}$$

In order to make a comparison, we start the Maximin-GFQ method Eq. (3.15) with the same point as the \mathbb{L}^2 -GFQ method Eq. (3.13), i.e., $\tilde{\mathbf{u}}_1 = \hat{\mathbf{u}}_1$. Figure 3.3 shows the results up to step $l = 10$ of both procedures. One can observe that the greedy maximin method covers well the range of variation of \mathbf{V} contrary to the \mathbb{L}^2 -GFQ method, which provides a well distributed points only on the first component. The \mathbb{L}^2 -GFQ seems to be more influenced by the higher order KL expansion. In the following, in order to improve the readability, we adopt the simplified notation Θ_l that refers to one of the two constructions Θ_l'' and Θ_l' . In this way

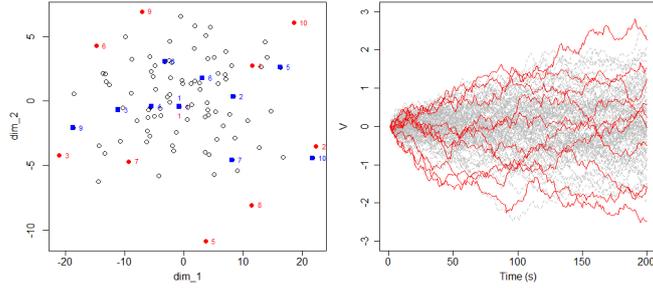


FIGURE 3.3: Left : Two designs of $l = 10$ points in the 2D-coefficients set \mathcal{G} . Maximin-GFQ (red circle points) and greedy \mathbb{L}^2 -GFQ (blue square points). Right : The corresponding red curves for the Maximin-GFQ procedure (right).

and in the same spirit as before (see equations (3.3),(3.5)), we define the Θ_l -quantization of the stochastic process \mathbf{V} as

$$\hat{\mathbf{V}}_l = \sum_{i=1}^l \theta_i \mathbf{1}_{C_{\theta_i}}(\mathbf{V}), \quad (3.16)$$

where $\{C_{\theta_i}(\Theta_l) : \theta_i \in \Theta_l\}$ is a Voronoi partition of Ξ induced by Θ_l as defined in Equation (3.4).

In this section, we have introduced two data-driven greedy original procedures for functional quantization, quantization being an alternative to Monte Carlo methods for numerical integration. In the next section we highlight the performance of these procedures through two analytical examples.

3.3 Numerical integration

Let $h : \mathcal{V} = \mathbb{L}^2([0, T]) \rightarrow \mathbb{R}$ be a continuous function, and let $\hat{\mathbf{V}}_l$ be a l -quantization. It is natural to approximate $\mathbb{E}[h(\mathbf{V})]$ by $\mathbb{E}[h(\hat{\mathbf{V}}_l)]$. This quantity $\mathbb{E}[h(\hat{\mathbf{V}}_l)]$ is simply the finite weighted sum :

$$\mathbb{E}[h(\hat{\mathbf{V}}_l)] = \sum_{i=1}^l h(\theta_i) \mathbb{P}(\hat{\mathbf{V}}_l = \theta_i), \quad (3.17)$$

where the distribution $(\mathbb{P}(\hat{\mathbf{V}}_l = \theta_i))_{i=1:l}$ of $\hat{\mathbf{V}}_l$ can be approximated empirically by $(\text{card}(C_{\theta_i}(\Theta_l) \cap \Xi) / \text{card}(\Xi))_{i=1:l}$. This proportion acts as a weight in the computation of the expectation. Assigning appropriate weights can bring a significant improvement (see L'Ecuyer et al. [L'Ecuyer and Lemieux, 2003]).

Remark. Under regularity assumptions, the integration error can be bounded by the quantization error. E.g., if h is Lipschitz in the sense that $\forall \mathbf{v}, \mathbf{v}' \in \mathcal{V}, |h(\mathbf{v}) - h(\mathbf{v}')| \leq [h]_{\text{lip}} \|\mathbf{v} - \mathbf{v}'\|$, then

$$\begin{aligned} |\mathbb{E}[h(\mathbf{V})] - \mathbb{E}[h(\hat{\mathbf{V}}_l)]| &\leq \mathbb{E}|h(\mathbf{V}) - h(\hat{\mathbf{V}}_l)| \\ &\leq [h]_{\text{lip}} \mathbb{E}\|\mathbf{V} - \hat{\mathbf{V}}_l\| \\ &\leq [h]_{\text{lip}} (\mathbb{E}\|\mathbf{V} - \hat{\mathbf{V}}_l\|^2)^{1/2}. \end{aligned} \quad (3.18)$$

Returning to our original notation $\mathbb{E}[f(\mathbf{x}, \mathbf{V})]$, the proposed methodologies to efficiently estimate the expectation over a functional random variable are summarized in Algorithms 3 and 4.

Algorithm 3 maximin-GFQ : Numerical integration

- 1: **Inputs** : initial sample (Ξ), truncation argument (m_{KL}), \mathbf{x} value where the expectation will be evaluated and set size of the quantization (l).
 - 2: $\mathcal{G} \leftarrow$ truncation of the KL expansion after m_{KL} -terms.
 - 3: Construct the set of representative points \tilde{D}_l defined by Eq. (3.14).
 - 4: Induce from \tilde{D}_l the corresponding set of representative curves Θ_l'' Eq. (3.15).
 - 5: Define $\hat{\mathbf{V}}_l \leftarrow \Theta_l''$ -quantization of \mathbf{V} Eq. (3.16).
 - 6: Perform the computation of the expectation $\mathbb{E}[f(\mathbf{x}, \hat{\mathbf{V}}_l)]$ Eq. (3.17) :
 - 7: $\mathbb{E}[f(\mathbf{x}, \mathbf{V})] \approx \mathbb{E}[f(\mathbf{x}, \hat{\mathbf{V}}_l)] = \sum_{i=1}^l f(\mathbf{x}, \theta_i'') \mathbb{P}(\hat{\mathbf{V}}_l = \theta_i'')$.
-

Algorithm 4 \mathbb{L}^2 -GFQ : Numerical integration

- 1: **Inputs** : initial sample (Ξ), truncation argument (m_{KL}), \mathbf{x} value where the expectation will be evaluated and set size of the quantization (l).
 - 2: $\mathcal{G} \leftarrow$ truncation of the KL expansion after m_{KL} -terms.
 - 3: Construct the set of representative points \hat{D}_l defined by Eq. (3.12).
 - 4: Induce from \hat{D}_l the corresponding set of representative curves Θ_l' Eq. (3.13).
 - 5: Define $\hat{\mathbf{V}}_l \leftarrow \Theta_l'$ -quantization of \mathbf{V} Eq. (3.16).
 - 6: Perform the computation of the expectation $\mathbb{E}[f(\mathbf{x}, \hat{\mathbf{V}}_l)]$ Eq. (3.17) :
 - 7: $\mathbb{E}[f(\mathbf{x}, \mathbf{V})] \approx \mathbb{E}[f(\mathbf{x}, \hat{\mathbf{V}}_l)] = \sum_{i=1}^l f(\mathbf{x}, \theta_i') \mathbb{P}(\hat{\mathbf{V}}_l = \theta_i')$.
-

In the sequel, we compare these two algorithms to the standard Monte Carlo, whose steps are outlined in Algorithm 5. The estimation of the expectation is sequentially calculated in the same vein as the GFQ procedures.

Algorithm 5 Crude MC : Numerical integration

- 1: **Inputs** : initial sample (Ξ), \mathbf{x} value where the expectation will be evaluated and set size of the quantization (l).
 - 2: Sample $\bar{\mathbf{V}}_l = \{\bar{\mathbf{v}}_i\}_{i=1}^l$ where $\bar{\mathbf{v}}_1, \dots, \bar{\mathbf{v}}_l \stackrel{i.i.d.}{\sim} \mathbb{U}_\Xi$, where \mathbb{U}_Ξ a discrete uniform distribution on Ξ .
 - 3: Estimate the expectation $\mathbb{E}[f(\mathbf{x}, \mathbf{V})]$ by l MC runs to $f(\mathbf{x}, \bar{\mathbf{v}})$:
 - 4: $\mathbb{E}[f(\mathbf{x}, \mathbf{V})] \approx \frac{1}{l} \sum_{i=1}^l f(\mathbf{x}, \bar{\mathbf{v}}_i)$.
-

We consider two analytical examples to highlight the integration performance of the two Greedy Functional Quantization methods (GFQ) in comparison with crude Monte Carlo. The first example is defined as an additive Lipschitz function, i.e., sum of the 2D Bohachevsky function and uncertainties. The second example does not check the Lipschitz assumptions to mimic real applications.

Application 1. We consider a functional f defined as

$$f : (\mathbf{x}, \mathbf{v}) \mapsto (x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7) + \int_T e^{\mathbf{v}^t} dt,$$

where $\mathbf{x} = (x_1, x_2) = (50, -80)$ and \mathbf{V} is a standard Brownian motion on \mathbb{R} and $[0, T] = [0, 1]$. We suppose that a sample Ξ of $N = 200$ realizations of \mathbf{V} is available and that the probability distribution of \mathbf{V} is unknown. In this example, we fix the truncation argument at 2 which leads to the explanation of 90% of the variance. Because of the random choice of the starting curve, the maximin-GFQ methods have a stochastic behaviour like the Monte Carlo method.

To account for these variabilities in the test, the performance of the method is averaged over 200 independent runs.

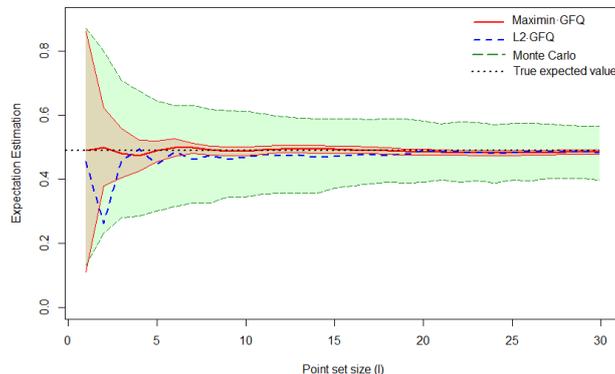


FIGURE 3.4: Application 1. Sequential expectation estimation. Lines denote the average estimates, and coloured bands mark the 25-th and 75-th quantiles (Monte Carlo (in green) and maximin-GFQ (in red)). Horizontal axis denotes the number of curves l used for the expectation estimation.

The results are shown in Figure 3.4. We observe that for any choice of l , the integration error induced by any of both GFQ methods is significantly smaller than the standard Monte Carlo. One can note that the maximin-GFQ method is less sensitive to the starting point from set size $l \geq 25$. We also remark that for a small size $l \leq 5$, the maximin-GFQ method is not yet stable implying more uncertainties in the estimation. From $l \geq 10$ (see Fig. 3.3), stability is reached since different kind of curves (centered and extreme) have been selected.

Application 2. We define a function f by mixing control variables and uncertainties. This function involves $\max(\mathbf{v})$ and $\min(\mathbf{v})$, so catching the variability of \mathbf{V} becomes important. The function f is given by

$$f : (\mathbf{x}, \mathbf{v}) \mapsto \max_t \mathbf{v}_t \cdot |0.1 \cos(x_1 \max_t \mathbf{v}_t) \sin(x_2) \cdot (x_1 + x_2 \min_t \mathbf{v}_t)^2|. \int_0^T (30 + \mathbf{v}_t)^{\frac{x_1 \cdot x_2}{20}} dt,$$

where $\mathbf{x} = (x_1, x_2) = (2.95, 3.97)$ and \mathbf{V} is a standard Brownian motion on \mathbb{R} and $[0, T] = [0, 1]$. To mimic real applications, we assume in the procedure that the probability distribution of \mathbf{V} is unknown. We suppose that a sample Ξ of $N = 200$ realizations of \mathbf{V} is available. We note that the two GFQ methods depend on the truncation argument m_{KL} . In this example, we fix it at 2 to explain 90% of the variance (results are similar for other truncation argument and \mathbf{x} values, and are omitted for brevity).

First, we compare two GFQ methods by fixing the starting point for maximin-GFQ at the first curve obtained by \mathbb{L}^2 -GFQ. One can see from Figure 3.5 that the obtained results are quite similar. For a small set size, the \mathbb{L}^2 -GFQ method leads to a smaller estimation error, this is due to the fact that the maximin-GFQ algorithm selects the points at the corners (exploration phase) as shown in Figure 3.3 which leads to a slower decrease of the error in the first steps. Due to the stochastic nature of the Monte Carlo and the maximin-GFQ methods, the performance of the method is averaged over 30 independent runs. The results of the integration algorithms are shown in Figure 3.6. The lines indicate average estimate and the coloured bands mark the area between the 25-th and 75-th quantiles. Here two observations can be made. First, for any choice of l , the integration error induced by any of both GFQ methods is significantly smaller than the standard Monte Carlo. Secondly, for maximin-GFQ method, the variability induced by the choice of the starting point is weak from set size $l \geq 20$.

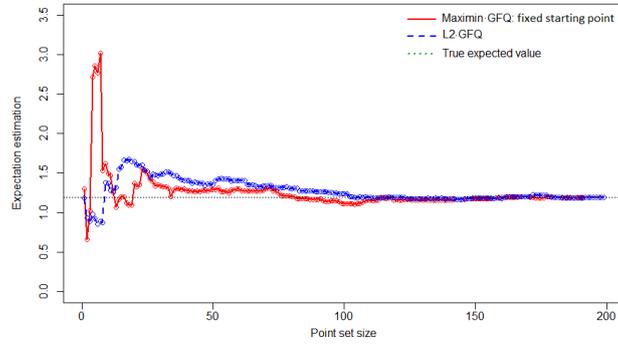


FIGURE 3.5: Application 2. Sequential expectation estimation by maximin-GFQ and L^2 -GFQ methods.

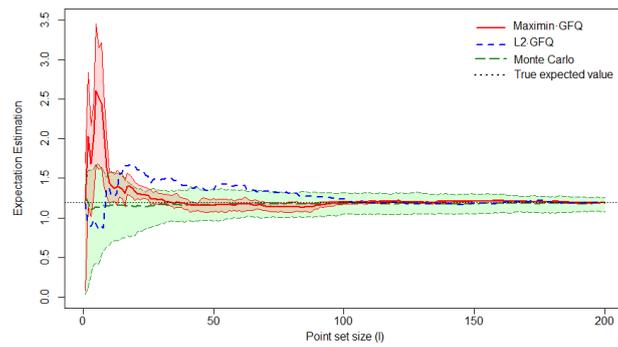


FIGURE 3.6: Application 2. Sequential expectation estimation. Lines denote the average estimates, and coloured bands mark the 25-th and 75-th quantiles (Monte Carlo (in green) and maximin-GFQ (in red)). Horizontal axis denotes the number of curves l used for the expectation estimation.

Note that our procedure is based on a dimension reduction. However, once the space filling

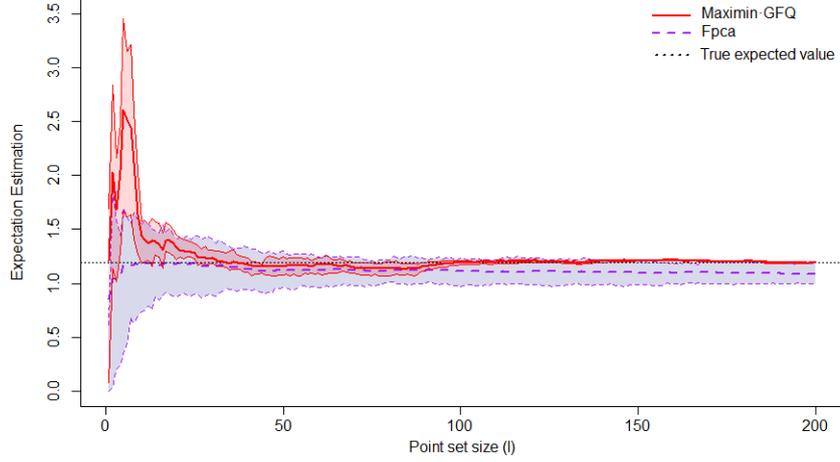


FIGURE 3.7: Application 2. Sequential expectation estimation. Lines denote the average estimates, and coloured bands mark the 25-th and 75-th quantiles (Fpca (in blue) and maximin-GFQ (in red)). Horizontal axis denotes the number of curves l used for the expectation estimation.

design has been built in $\mathbb{R}^{m_{KL}}$, we go back to the infinite-dimensional space \mathcal{V} by selecting the corresponding curves. We expect from such a procedure to be robust to the dimension reduction. To illustrate this intuition, we compare the maximin-GFQ algorithm to an existing method, called hereafter Fpca method. This latter consists in sampling independently the KL random variables $\mathbf{U} \in \mathbb{R}^{m_{KL}}$ whose probability distribution is estimated beforehand and denoted $\mathbb{P}_{\mathbf{U}}$ (see, e.g., [Nanty et al., 2016] for a detailed overview on the subject). Then we obtain the desired curves using the linear combination of Equation (3.10). The steps of the Fpca method are summarized in Algorithm 6. The results of the comparison are shown in Figure 3.7. We note that the Fpca method leads to a biased estimation due to the regularization induced by reducing the dimension.

Algorithm 6 Fpca : Numerical integration

- 1: **Inputs** : initial sample (Ξ), truncation argument (m_{KL}), \mathbf{x} value where the expectation will be evaluated and set size of the quantization (l).
 - 2: $\mathcal{G} \leftarrow$ truncation of the KL expansion after m_{KL} -terms.
 - 3: $\mathbb{P}_{\mathbf{U}} \leftarrow$ estimation of the joint distribution of the truncated-KL random variables $\mathbf{U} \in \mathbb{R}^{m_{KL}}$ from \mathcal{G} .
 - 4: Sample $\mathbf{U}_l^{\text{pca}} = \{\mathbf{u}_i^{\text{pca}}\}_{i=1}^l$ where $\mathbf{u}_1^{\text{pca}}, \dots, \mathbf{u}_l^{\text{pca}} \stackrel{i.i.d.}{\sim} \mathbb{P}_{\mathbf{U}}$.
 - 5: Construct the associated curves $\mathbf{V}_l^{\text{pca}} = \{\mathbf{v}_i^{\text{pca}}\}_{i=1}^l$ to $\mathbf{U}_l^{\text{pca}}$ defined by Equation (3.10).
 - 6: Estimate the expectation $\mathbb{E}[f(\mathbf{x}, \mathbf{V})]$ by l MC runs to $f(\mathbf{x}, \mathbf{v}^{\text{pca}})$:
 - 7: $\mathbb{E}[f(\mathbf{x}, \mathbf{V})] \approx \frac{1}{l} \sum_{i=1}^l f(\mathbf{x}, \mathbf{v}_i^{\text{pca}})$.
-

In summary, these simulations show that the two GFQ methods benefit improved performances over Monte Carlo in numerical accuracy. Regarding computation time, Figure 3.8 shows the time needed to compute the whole Greedy Functional Quantization as a function of point set size l . The KL expansion is done in a prior unaccounted step. These algorithms are implemented in the software R and all computations are performed on a 8-core Intel 2.80 Ghz processor. From this figure, the running time of the maximin-GFQ method grows much

more slowly than the \mathbb{L}^2 -GFQ one. Indeed, the discrete optimization in the \mathbb{L}^2 -GFQ method (Eq.(3.12)) involves an empirical estimation of the expectation at each step, thus the algorithm has complexity $O(N^2 \times l \times m_{KL})$. The latter becomes time consuming as l increases comparing to the maximin-GFQ algorithm which has complexity $O(N \times l \times m_{KL})$.

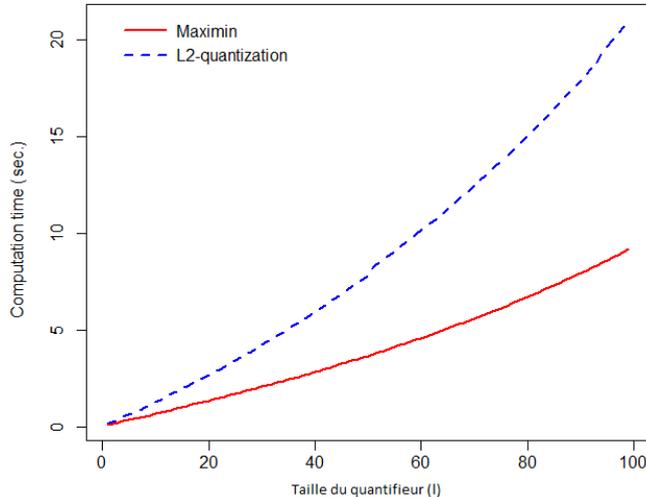


FIGURE 3.8: Application 2. Computation time (in seconds) of the two GFQ methods as a function of point set size (l).

In this section, we presented a methodology to efficiently estimate the expectation over \mathbf{V} at a point \mathbf{x} in the control space \mathbb{X} . In the next section, we recall an existing method to address the inversion problem in the control variable space in the context of computationally costly simulations. This strategy is defined in a Bayesian framework based on the so-called Stepwise Uncertainty Reduction strategy (SUR). Let us start with some presentation of the SUR paradigm.

3.4 Background on SUR strategies

Let $f : \mathbb{X} \times \mathcal{V} \rightarrow \mathbb{R}$ denote a real-valued continuous function, where \mathbb{X} is a bounded subset of \mathbb{R}^p , $p \geq 1$, and \mathcal{V} a functional space on which a random variable \mathbf{V} is defined. Moreover, we suppose that a finite set of N independent and identically distributed realizations of the functional random variable \mathbf{V} is available. In the following, we consider the expectation over the distribution of the functional random variable and we are interested in characterizing the set of control variables which leads a system a safe behaviour :

$$\Gamma^* := \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})] \in C\} \quad \text{with} \quad C = (\infty, c], \quad c \in \mathbb{R}. \quad (3.19)$$

While the function f depends on two separate types of inputs (control and uncertain variables), our objective function g depends only on the control variables, i.e., for each setting of control variables, the objective function is the mean of f over the unknown distribution of the uncertain variable.

The estimation of Γ^* by evaluating the function g at each grid point of the discretized version of \mathbb{X} requires far too many evaluations of g . Therefore, statistical methods based on a reduced number of evaluation points are widely used to overcome this latter difficulty by focusing the evaluations on the 'promising' subregion of the control space.

These methods usually begin by an exploration phase, during which the output of the code is computed on an experimental design of size n . This initial design is then sequentially expanded by adding new goal oriented points. These sequential strategies have been used in recent years for many purposes, such as the failure probability estimation ([Bect et al., 2012]) and target regions ([Picheny et al., 2010]) whose main idea is to decrease the variance kriging at the points where the kriging mean is close to the threshold c . Unlike the two cited methods, we are interested in the whole excursion set. In ([Chevalier et al., 2013],[Chevalier et al., 2014b]) the sampling criterion is based on concept of random closed sets and applied to identify the set Γ^* . In this work we adopt this strategy and the procedure is introduced below.

3.4.1 Random closed set and bayesian framework

In a Bayesian framework, we assume that g is a realization of an almost surely continuous Gaussian process $Y \sim GP(m, k)$ with a mean structure m , defined as, $m(\mathbf{x}) = \mathbb{E}[Y_{\mathbf{x}}]$, $\mathbf{x} \in \mathbb{X}$, and a covariance kernel k , defined as, $k(\mathbf{x}, \mathbf{x}') := \text{Cov}(Y_{\mathbf{x}}, Y_{\mathbf{x}'})$, $\mathbf{x}, \mathbf{x}' \in \mathbb{X}$. Due to the stochastic nature of $(Y_{\mathbf{x}})_{\mathbf{x} \in \mathbb{X}}$, the associated excursion set,

$$\Gamma := \{\mathbf{x} \in \mathbb{X}, Y_{\mathbf{x}} \in C\} \quad (3.20)$$

is a random closed set. From the assumption that g is a realization of Y , the true unknown set Γ^* can be seen as a realization of the random closed set Γ . Therefore, we implement a Stepwise Uncertainty Reduction strategy (SUR) that aims at reducing uncertainty on Γ by adding new evaluation points step by step as proposed by [Chevalier et al., 2013]. In this context such strategy relies on the notion of uncertainty for random sets. This latter will be characterized by the Vorob'ev expectation and deviation for random sets introduced in Subsection 3.4.2. The principle of SUR strategies are also recalled in Subsection 3.4.3.

3.4.2 Vorob'ev approach

Γ defines a random closed set. Several ways to define the expectation of a random set have been developed in the literature (see e.g. [Molchanov, 2006]). In this paper, we focus on Vorob'ev expectation, closely related to quantiles and level-sets which have been the subject of many developments. Let us define *the coverage probability function* of a random set Γ as

$$p(\mathbf{x}) = \mathbb{P}(\mathbf{x} \in \Gamma), \quad \mathbf{x} \in \mathbb{X}. \quad (3.21)$$

The coverage function p is not always available in an analytical closed form. For instance [Cuevas et al., 2006], [Cuevas and Fraiman, 1997] proposed to replace p by an empirical counterpart and established consistency of plug-in estimators under weak assumptions. Here thanks to the Gaussianity assumption we will work with a closed form criterion. For $\alpha \in [0, 1]$, the α -level set of $p(\mathbf{x})$, also known as the *Vorob'ev Quantile*, is

$$Q_{\alpha} = \{\mathbf{x} \in \mathbb{X} : p(\mathbf{x}) \geq \alpha\}. \quad (3.22)$$

Let μ be a Borel σ -finite measure defined on \mathbb{X} , the Vorob'ev expectation is defined as Q_{α^*} with α^* such that the volume of Q_{α^*} matches the mean volume of Γ in the following sense :

$$\forall \beta > \alpha^*, \mu(Q_{\beta}) < \mathbb{E}[\mu(\Gamma)] \leq \mu(Q_{\alpha^*}). \quad (3.23)$$

Finally we introduce a notion of variability, based on the concept of expected distance between two random sets. First, let us consider two random closed sets A, B , the expected distance between A, B with respect to the measure μ is defined as

$$d_{\mu}(A, B) = \mathbb{E}[\mu(A \Delta B)], \quad (3.24)$$

where $A\Delta B$ is the symmetric difference of A and B . The quantity $\mathbb{E}[\mu(\Gamma\Delta Q_{\alpha^*})]$ is called *Vorob'ev deviation*.

In the following, we use the Vorob'ev expectation and deviation to quantify the variability of Γ conditionally to available observations. Let us denote the initial design points as $\mathcal{X}_n = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \in \mathbb{X}^n$ and the responses at these points as $\mathbf{g}_{\mathcal{X}_n} = (g(\mathbf{x}_1), g(\mathbf{x}_2), \dots, g(\mathbf{x}_n)) \in \mathbb{R}^n$. We note

$Y_{\mathcal{X}_n} = (Y_{\mathbf{x}_1}, Y_{\mathbf{x}_2}, \dots, Y_{\mathbf{x}_n})$ the random vector associated to the random process Y considered at \mathcal{X}_n . The main object of interest is then the conditional probability distribution of the random closed set Γ given the n observations. We know that the posterior distribution of the process Y given the n available observations remains Gaussian and is characterized by the posterior mean $m_n(\mathbf{x}) = \mathbb{E}[Y_x \mid Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}]$, $\mathbf{x} \in \mathbb{X}$, and the posterior covariance kernel, $k_n(\mathbf{x}, \mathbf{x}') := \text{Cov}(Y_{\mathbf{x}}, Y_{\mathbf{x}'} \mid Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n})$. The latter are given by the following equations :

$$\begin{aligned} m_n(\mathbf{x}) &= m(\mathbf{x}) + \mathbf{k}_n(\mathbf{x})^\top \mathbf{K}_n^{-1}(\mathbf{g}_{\mathcal{X}_n} - m(\mathcal{X}_n)), \\ s_n^2(\mathbf{x}) &= k_n(\mathbf{x}, \mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_n(\mathbf{x})^\top \mathbf{K}_n^{-1} \mathbf{k}_n(\mathbf{x}), \end{aligned} \quad (3.25)$$

where $\mathbf{k}_n(\mathbf{x}) = (k(\mathbf{x}, \mathbf{x}_1), \dots, k(\mathbf{x}, \mathbf{x}_n))^\top$, $\mathbf{K}_n = (k(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i, j \leq n}$.

Given the n observations, the *coverage probability function* p_n and *Vorob'ev Quantile* $Q_{n, \alpha}$ can easily be computed as presented in [Chevalier et al., 2013]. The Vorob'ev expectation Q_{n, α_n^*} can be determined by tuning α to a level α_n^* such that $\mu(Q_{n, \alpha_n^*}) = \mathbb{E}[\mu(\Gamma) \mid Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}]$ as explained in Chapter 1. Once the Vorob'ev expectation is determined, the computation of Vorob'ev deviation $\mathbb{E}[\mu(\Gamma\Delta Q_{n, \alpha_n^*}) \mid Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}]$ can be expressed as a function of the posterior coverage probability function p_n and does not require simulations of Γ (for detailed formulas see [Chevalier et al., 2013]).

3.4.3 SUR strategies

The principle of stepwise uncertainty reduction (SUR) (see, e.g., [Bect et al., 2012]; [Chevalier et al., 2014a]) is to define an uncertainty measure, depending on the objective to be fulfilled, and to sequentially choose the points that decrease most this uncertainty. In other words, the aim of the SUR strategy is to construct a sequence of evaluation locations in order to reduce the *expected* uncertainty on a quantity of interest.

Here, we work in the setting where g is a sample path of a random process Y . The uncertainty function for an estimate of Γ is defined as a function $\mathcal{H}^{\text{uncert}}$ that associates to any finite sequence of observations $(\mathcal{X}_n, \mathbf{g}_{\mathcal{X}_n})$ a real value representing the uncertainty on the estimation of Γ . When n observations are available, we denote by $\mathcal{H}_n^{\text{uncert}}$ the uncertainty at step n . We assume that we have r evaluations left. The objective of the SUR strategy is to find r optimal locations $\mathbf{x}_{n+1}, \dots, \mathbf{x}_{n+r}$ such that the uncertainty $\mathcal{H}_{n+r}^{\text{uncert}}$ is as small as possible.

In what follows, we consider the Vorob'ev deviation as the uncertainty function, at step n ,

$$\mathcal{H}_n^{\text{uncert}} = \mathbb{E}[\mu(\Gamma\Delta Q_{n, \alpha_n^*}) \mid Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}] \quad (3.26)$$

One way of constructing the sequence $\mathbf{x}_{n+1}, \dots, \mathbf{x}_{n+r}$ is to choose at each step the point that gives the smallest uncertainty $\mathcal{H}_{n+1}^{\text{uncert}}$,

$$\mathcal{H}_{n+1}^{\text{uncert}}(\mathbf{x}) = \mathbb{E}[\mu(\Gamma\Delta Q_{n+1, \alpha_{n+1}^*}) \mid Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}, Y_{\mathbf{x}}] \quad (3.27)$$

We note that the future uncertainty $\mathcal{H}_{n+1}^{\text{uncert}}$ is function of $Y_{\mathbf{x}}$ given $Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}$. Therefore, at each step we choose the point that gives the smallest uncertainty in expectation, that is :

$$\begin{aligned} \mathbf{x}_{n+1} &\in \operatorname{argmin}_{\mathbf{x} \in \mathbb{X}} \mathbb{E}_{n, \mathbf{x}}[\mathcal{H}_{n+1}^{\text{uncert}}(\mathbf{x})] \\ &:= \operatorname{argmin}_{\mathbf{x} \in \mathbb{X}} \mathcal{J}_n(\mathbf{x}), \end{aligned} \quad (3.28)$$

where $\mathbb{E}_{n,\mathbf{x}}$ denotes the expectation with respect to $Y_{\mathbf{x}}|Y_{\mathcal{X}_n} = \mathbf{g}_{\mathcal{X}_n}$ (for detailed formula of $\mathcal{J}_n(\cdot)$ see [Chevalier et al., 2013]).

After having evaluated the function g at the optimal location \mathbf{x}_{n+1} , we update the parameters of the posterior mean and covariance, and we restart until the evaluation budget r is spent. Such strategy is called *one-step lookahead*, which means that we select the next evaluation point as if it were the last one. For more theoretical perspectives on the SUR strategies, see [Bect et al., 2016] and references therein.

3.4.4 SUR strategy adapted to noisy observations

In our context, we can not compute exact evaluations of the expectation over the probability distribution of \mathbf{V} . We propose in Section 3.2 sequential algorithms that efficiently approximate the value of $g(\mathbf{x})$ by $\mathbb{E}[f(\mathbf{x}, \hat{\mathbf{V}}_l)]$. On the design points, the n evaluations $\mathbf{g}(\mathcal{X}_n)$ are replaced by their approximation $\tilde{\mathbf{g}}(\mathcal{X}_n) = (\mathbb{E}[f(\mathbf{x}_1, \hat{\mathbf{V}}_l)], \dots, \mathbb{E}[f(\mathbf{x}_n, \hat{\mathbf{V}}_l)])$. For that reason, we do not want to build an exact interpolant at points $\mathbf{x}_1, \dots, \mathbf{x}_n$. We rather consider that $\tilde{\mathbf{g}}(\mathcal{X}_n)$ are realizations of a Gaussian vector $(\tilde{Y}_{\mathbf{x}_1}, \dots, \tilde{Y}_{\mathbf{x}_n})^T$ defined by $\tilde{Y}_{\mathbf{x}_i} := Y_{\mathbf{x}_i} + \epsilon_i$ where $\epsilon_1, \dots, \epsilon_n$ are independent centred Gaussian variables of variance $\tau_1^2, \dots, \tau_n^2$. Conditionally to $\tilde{Y}_{\mathbf{x}_1}, \dots, \tilde{Y}_{\mathbf{x}_n}$ the process Y is still Gaussian. Its conditional mean and variance functions are given by Equations (3.25), except that the covariance matrix \mathbf{K}_n is replaced by $\tilde{\mathbf{K}}_n = \mathbf{K}_n + \text{diag}([\tau_1^2 \dots \tau_n^2])\mathbf{I}_n$.

Remark. In our context, we estimate the expectation empirically by l calls to the function f . The well-known Monte Carlo methods (Crude MC, FPCA) allow us to quantify the noise of estimation and to integrate it into GP modelling (kriging with noisy observations). Consequently, we define the variance components $\{\tau_i^2\}_{i=1}^n$ as $\tau_i^2 = \text{var}(f(\mathbf{x}_i, \bar{\mathbf{V}}_l))$ for the crude MC method and $\text{var}(f(\mathbf{x}_i, \mathbf{V}_l^{\text{pca}}))$ for the FPCA method. We also note that the two deterministic GFQ methods are in the same spirit than the Quasi-MC methods. The error of estimation depends on the variation of f that is most often not tractable. Therefore we assume a negligible integration error and thus the observations are assumed noise-free.

3.5 Algorithm coupling SUR and functional quantization

The whole computational aspect is carried out in the **R** environment : we use `DiceKriging` package [Roustant et al., 2012] for Gaussian modelling and the sampling criterion \mathcal{J}_n Eq. (3.28), used in order to select the next evaluation \mathbf{x}_{n+1} of the function g , is already implemented in the `KrigInv` package [Chevalier et al., 2014b]. We exploit the kriging update formulas [Chevalier et al., 2015] for faster updates of posterior mean and covariance. The sequentiality of our estimation method of the expectation on \mathbf{x}_{n+1} leads us to define a *stopping criterion* on the expectation estimation \hat{m} . Thus, at each step in the estimation, we evaluate the absolute difference between two consecutive estimations of the expectation,

$$e_l(\mathbf{x}_{n+1}) = |\hat{m}_{l-1}(\mathbf{x}_{n+1}) - \hat{m}_l(\mathbf{x}_{n+1})|, \quad (3.29)$$

where $\hat{m}_i(\mathbf{x}_{n+1}) = \mathbb{E}[f(\mathbf{x}_{n+1}, \hat{\mathbf{V}}_i)]$, we denote by $|\cdot|$ the absolute value function. In the following, the *stopping criterion* is defined by the following relation,

$$\forall 0 \leq j \leq l_0, e_{l-j}(\mathbf{x}_{n+1}) \leq \epsilon \quad (3.30)$$

It ensures that the quantities e_l are smaller than a prescribed tolerance ϵ on the l_0 previous steps in the estimation. These two parameters are set by practitioners. It allows to use fewer curves without losing estimation accuracy.

Remark. The parameters (l_0, ϵ) are closely linked to the allocated budget. Moreover the parameter l_0 can be set in practice between $[2, 5]$ regarding the stability we want to achieve with the method. About the parameter ϵ , it will be intuitively calibrated depending both on the precision and on the scale of the outputs.

The strategy SUR could be stopped if the allocated number of simulations is reached. However, we define in this work an additional stopping criterion based on the Vorob'ev deviation and close to the one defined for the expectation estimate. Thus, the strategy is carried out until the following stopping criterion is verified

$$\forall 0 \leq j \leq l_0^{\text{SUR}}, e_{l-j}^{\text{SUR}}(\mathbf{x}_{n+1}) \leq \epsilon^{\text{SUR}} \quad (3.31)$$

where $e_i^{\text{SUR}} = | \mathbb{E}_{i-1}[\mu(\Gamma \Delta Q_{i-1, \alpha_{i-1}^*})] - \mathbb{E}_i[\mu(\Gamma \Delta Q_{i, \alpha_i^*})] |$ is the absolute error between two successive Vorob'ev deviations. The condition Eq. (3.31) tests if all the quantities are smaller than a tolerance ϵ^{SUR} on l_0^{SUR} consecutive steps.

The global methodology to perform inversion in presence of functional uncertainty proposed in this paper is summarized in Algorithm 8.

Algorithm 7 Data-driven stochastic inversion under functional uncertainties

- 1: Create an initial design of experiments (DoE) of n points in the control space \mathbb{X} .
 - 2: Alg $\mathcal{A} \leftarrow$ choose one of the Algorithms 3,4, 5 and 6.
 - 3: Estimate the expectation (Alg. \mathcal{A}).
 - 4: Deduce $\{\tau_i^2\}_{i=1}^n$ (if Crude MC or Fpca methods).
 - 5: **while** Stopping criterion Eq. (3.31) not met (SUR) **do**
 - 6: $\mathbf{x}_{n+1} \leftarrow$ Sampling criterion \mathcal{J}_n .
 - 7: Set $l = 1$.
 - 8: **while** Stopping criterion Eq. (3.30) not met (Expectation Estimation) **do**
 - 9: Approximate the expectation by $\mathbb{E}[f(\mathbf{x}_{n+1}, \hat{\mathbf{V}}_l)]$ (Alg. \mathcal{A}).
 - 10: Set $l = l + 1$.
 - 11: **end while**
 - 12: $\tau_{\mathbf{x}_{n+1}}^2 \leftarrow \text{var}(f(\mathbf{x}_{n+1}, \hat{\mathbf{V}}_l))$ (if Crude MC or Fpca methods).
 - 13: Update DoE.
 - 14: Set $n = n + 1$.
 - 15: **end while**
 - 16: **end**
-

Remark on stage 2 of Algorithm 8 Due to their sampling based nature, the Crude MC and Fpca methods are sensitive to the resulting estimation errors. Consequently, we consider the adaptation of the SUR strategy for noisy observations (see, Subsection 3.4.4).

3.6 Numerical tests

In this section we apply the proposed data-driven methodology for stochastic inversion under functional uncertainties to two test cases. On an analytical test case, we compare the methods combining the SUR strategy to the four approach to estimate the expectation. We then present in Subsection 3.6.2 an application to the industrial automotive test case which motivate our study.

3.6.1 Analytical example

In this example, we define the function f as follows :

$$f : (\mathbf{x}, \mathbf{v}) \mapsto \max_t \mathbf{v}_t \cdot |0.1 \cos(x_1 \max_t \mathbf{v}_t) \sin(x_2) \cdot (x_1 + x_2 \min_t \mathbf{v}_t)^2| \cdot \int_0^T (30 + \mathbf{v}_t)^{\frac{x_1 \cdot x_2}{20}} dt, \quad (3.32)$$

where the control variable \mathbf{x} lies in $\mathbb{X} = [1.5, 5] \times [3.5, 5]$, and \mathbf{V} is a standard Brownian motion. We suppose that a sample of N realizations of \mathbf{V} is available, denoted by Ξ , and these realizations are discretized uniformly on 100 points. The objective is to construct the set $\Gamma^* := \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})] \leq c\}$, where $c = 1.2$.

Here we consider a Gaussian process prior $(Y_{\mathbf{x}})_{\mathbf{x} \in \mathbb{X}} \sim GP(m, k)$, with constant mean function and Matérn covariance kernel with $\nu = 5/2$. The initial DoE consists of a 9 points LHS design optimized by maximin criterion. The hyper-parameters of the Gaussian process Y are estimated by Maximum Likelihood Estimation (MLE). Figure 3.9 shows the initial design of experiments and the target set Γ^* obtained from a 30×30 grid experiment, where at each grid point the expectation is approximated by a Monte Carlo Method over 5000 realizations of \mathbf{V} . We aim at estimating the set Γ^* using the SUR strategy to choose the next evaluation point as defined in Section 3.4, and the methods presented in Section 3.2 to provide an efficient estimation of the expectation. We proceed to add one point at each iteration of the

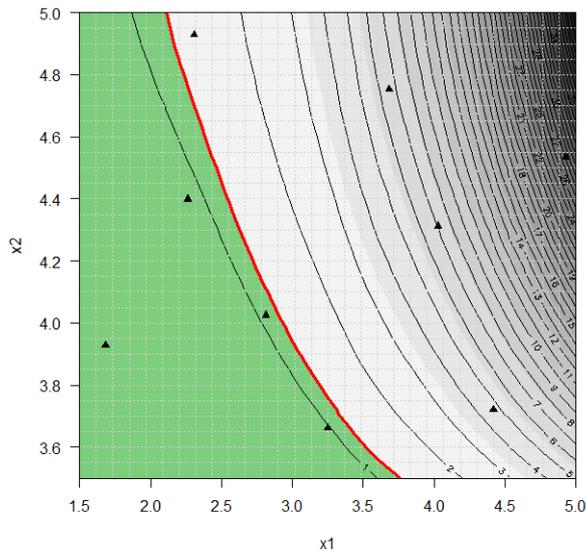


FIGURE 3.9: Analytical example. Contour plot of the function g , the set of interest (green) with boundary (red line), the initial design of experiments (black triangles).

SUR strategy until the condition Eq. (3.31) for $(l_0^{\text{SUR}}, \epsilon^{\text{SUR}}) = (4, 5.10^{-3})$ is reached. The covariance parameters are re-estimated at each step by MLE. Since this criterion is based on the Vorob'ev deviation, the objective is to reduce the uncertainty on the set estimate until stability. For the sequential estimation of the expectation, we test the sensitivity to the parameters (l_0, ϵ) of criterion Eq. (3.30) (see Table 3.1).

The estimation of the expectation at the proposed point by SUR is carried out with one of the methods detailed in Section 3.2 (Fpca, crude MC, maximin-GFQ, \mathbb{L}^2 -GFQ). As presented in Section 3.5, the estimation is done sequentially and it depends on the *stopping criteria* parameters l_0, ϵ and on the truncation argument m_{KL} . This latter is set at $m_{KL} = 7$ in order to explain 97% of the variance. The four expectation estimation methods are sequential as

detailed in Section 3.2 . Indeed, the two GFQ methods are sequential by definition. The crude MC method is sequential because at each step a curve is drawn with replacement from the available sample Ξ (see Algorithm 5). The same goes for the probabilistic approach (Fpca), at each step we add a new curve built as explained in Algorithm 6.

The first test consists in fixing the available sample of realizations of \mathbf{V} ($N=200$). For this fixed sample, we compare the obtained results for different l_0 and ϵ . Table 3.1 lists the parameters tested in this section.

l_0	4	2	3	4
ϵ	10^{-2}	5.10^{-3}	5.10^{-3}	5.10^{-3}

TABLE 3.1: Analytical example. *Stopping criteria* parameters used in the estimation of expectation.

To compare the performance of the various methods we use the ratio between the volume of the symmetric difference between the true set Γ^* and the estimated set at last iteration, $\mu(\Gamma^* \Delta Q_{n_{\text{last}}, \alpha_{n_{\text{last}}}})$ and the volume of the true set, $\mu(\Gamma^*)$. As shown earlier in Figure 3.6, the maximin-GFQ method is not very sensitive to the starting point. Thus, in the following test, we consider the deterministic version of the maximin-GFQ method by fixing the starting point to the one of \mathbb{L}^2 -GFQ method.

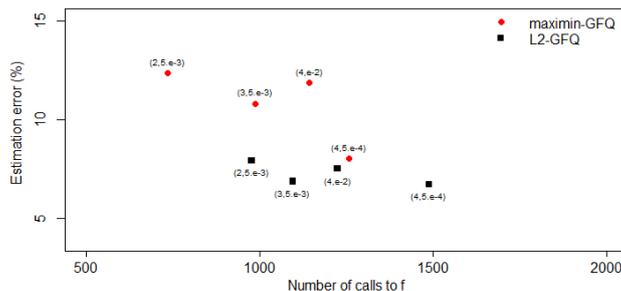


FIGURE 3.10: Analytical example. The relative error obtained by the two GFQ methods for different values of l_0 and ϵ as a function of the number of calls to the function f .

(l_0, ϵ)	$\mu(\Gamma^* \Delta Q_{n_{\text{last}}, \alpha_{n_{\text{last}}}}) / \mu(\Gamma^*)$		Cumulative number of calls to f	
	maximin-GFQ	\mathbb{L}^2 -GFQ	maximin-GFQ	\mathbb{L}^2 -GFQ
(4,1.e-2)	11.86 %	7.50 %	1144 (21)	1225 (42)
(2,5.e-3)	12.34 %	7.93 %	735 (21)	978 (22)
(3,5.e-3)	10.80 %	6.87 %	989 (18)	1096 (21)
(4,5.e-3)	8.02 %	6.79 %	1259 (19)	1489 (26)

TABLE 3.2: Analytical example. (Left) The relative error obtained by the two GFQ methods for different values of l_0 and ϵ . (Right) The cumulative number of calls to the function f (in brackets are the number of iterations required to reach the stopping criterion in the SUR strategy).

From the comparison results displayed in Table 3.2 and plotted on Figure 3.10, we note that the two GFQ methods are sensitive to the parameters l_0 and ϵ . The \mathbb{L}^2 -GFQ method performs well in set estimation error terms, the maximin-GFQ provides better results in terms of cost. In the following comparison tests, we consider only the \mathbb{L}^2 -GFQ method as it gives much better set estimation error for a reasonable number of calls to the function f .

Regarding the second test, the two expectation estimation methods (Crude MC and Fpca) have a stochastic behaviour because of the sampling steps. To account for these variabilities, the performance of each method is averaged over 30 independent runs. The results are summarized in Tables 3.4 and 3.3. The results indicate that the three methods are sensitive to the parameters l_0 and ϵ : larger is the parameter l_0 , i.e., when seeking to a stability of the estimation, smaller is the error but higher is the number of calls to the function f . The \mathbb{L}^2 -GFQ method performs well in terms of error and cost. The cumulative number of calls to f has been decreased by a factor greater than 3 in comparison with the two other methods.

	$\mu(\Gamma^* \Delta Q_{n_{\text{last}}, \alpha_{n_{\text{last}}}^*}) / \mu(\Gamma^*)$		
(l_0, ϵ)	Crude MC	Fpca	\mathbb{L}^2 -GFQ
(4,1.e-2)	9.53 % (4.12)	9.89 % (4.14)	7.50 %
(2,5.e-3)	9.84 % (3.24)	10.86 % (2.82)	7.93 %
(3,5.e-3)	9.54 % (3.81)	7.29 % (1.07)	6.87 %
(4,5.e-3)	8.98 % (2.62)	7.01 % (1.21)	6.79 %

TABLE 3.3: Analytical example. The average relative error obtained by the crude MC, Fpca for different values of l_0 and ϵ (in brackets are the standard deviation for the crude MC and Fpca methods).

	Cumulative number of calls to the function f		
(l_0, ϵ)	Crude MC	Fpca	\mathbb{L}^2 -GFQ
(4,1.e-2)	2849 (27)	2805 (24)	1225 (42)
(2,5.e-3)	2393 (26)	2670 (23)	978 (22)
(3,5.e-3)	3537 (23)	3661 (20)	1096 (21)
(4,5.e-3)	4400 (23)	4278 (20)	1489 (26)

TABLE 3.4: Analytical example. The average cumulative number of calls to the function f (written in brackets are the number of iterations required to reach the stopping criterion in the SUR strategy).

Figure 3.11 shows the set estimation error and the Vorob'ev deviation as a function of the iteration number for the three methods and $(l_0, \epsilon) = (4,5.e-3)$. For the crude MC and Fpca methods, the dotted lines indicate average error decay, and the coloured bands mark the area between the 25-th and 75-th error quantiles. Note that the three methods show a strong decrease in the set estimation error. The main observation that can be made is that, for a small total call to f (see, Table 3.4), the convergence rate for the proposed approach (\mathbb{L}^2 -GFQ) is better in comparison to the Crude MC and Fpca methods.

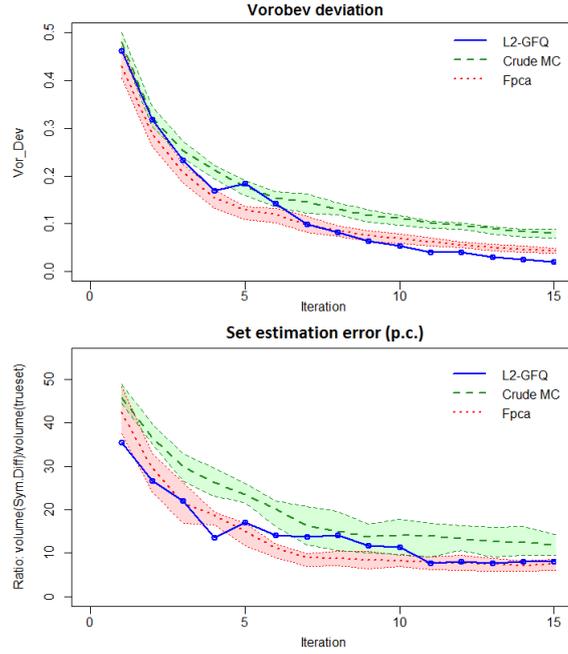


FIGURE 3.11: Analytical example. Results for $(l_0, \epsilon) = (4, 5.e^{-3})$. Lines denote the average, and coloured bands mark the 25-th and 75-th quantiles (Fpca (in red) and Crude MC (in green)). Top : The Vorob'ev deviation. Bottom : The set estimation error $\mu(\Gamma^* \Delta Q_{n_{\text{last}}, \alpha_{n_{\text{last}}}^*}) / \mu(\Gamma^*)$.

In the following, the stopping criteria for SUR ($l_0^{\text{SUR}} = 4, \epsilon^{\text{SUR}} = 5.10^{-3}$) and for the expectation estimation ($l_0 = 4, \epsilon = 5.10^{-3}$) are chosen because they offer a good compromise between the accuracy and the number of model evaluations.

Table 3.5 compares the sensitivity of the methods to the size of the available sample Ξ , denoted by N . In each cell of the table, we perform 20×20 independent runs. Indeed, for each value of N , we generate 20 training samples Ξ of size N and for each sample we perform 20 runs for each method. The table summarizes the results averaged over the 400 runs.

	$\mu(\Gamma^* \Delta Q_{n_{\text{last}}, \alpha_{n_{\text{last}}}^*}) / \mu(\Gamma^*)$		
	Crude MC	Fpca	\mathbb{L}^2 -GFQ
N=50	15.38 % (8.76)	13.25 % (5.09)	11.13 % (6.48)
N=100	9.60 % (4.65)	8.80 % (3.60)	9.08 % (5.41)
N=200	8.22 % (2.18)	7.71 % (2.35)	7.02 % (2.31)

	Cumulative number of calls to f		
	Crude MC	Fpca	\mathbb{L}^2 -GFQ
N=50	4281 (22)	4343 (21)	1044 (27)
N=100	4262 (22)	4313 (21)	1236 (25)
N=200	4152 (22)	4552 (21)	1262 (24)

TABLE 3.5: Analytical example. (Top) The average set estimation error obtained for different sample size and methods and $m_{KL} = 7$ (in brackets are the standard deviation). (Bottom) The average cumulative number of calls to the function f (in brackets are the number of iterations required to reach the stopping criterion in the SUR strategy).

We note that for a larger sample size, the recovering error is smaller. This can be explained

by the fact that with a large sample size, the available information on variable \mathbf{V} enables an effective estimation of the expectation.

We know that the \mathbb{L}^2 -GFQ and the probabilistic modelling (Fpca) depend on the truncation argument. To better understand the effect of the number of dimensions m , we fix the stopping criteria for the SUR strategy and expectation estimation, and we consider different values of $m_{KL} = \{2, 3, 4, 5, 6\}$. Each cell of Table 3.6 represents the result averaged over 14×20 independent runs. For each m , we generate 14 samples Ξ of size $N=200$, and for each of them we perform 20 runs of each method.

	$\mu(\Gamma^* \Delta Q_{n_{\text{last}}, \alpha_{n_{\text{last}}}^*}) / \mu(\Gamma^*)$	
	Fpca	\mathbb{L}^2 -GFQ
$m_{KL} = 2$	11.43 % (3.70)	8.90 % (3.71)
$m_{KL} = 3$	10.70 % (3.38)	7.72 % (3.38)
$m_{KL} = 4$	9.24 % (3.18)	7.40 % (3.13)
$m_{KL} = 5$	8.94 % (2.66)	7.05 % (5.09)
$m_{KL} = 6$	8.27 % (1.67)	6.96 % (3.32)

	Cumulative number of calls to f	
	Fpca	\mathbb{L}^2 -GFQ
$m_{KL} = 2$	3855 (18)	1286 (26)
$m_{KL} = 3$	4418 (24)	1139 (21)
$m_{KL} = 4$	4438 (21)	1236 (20)
$m_{KL} = 5$	4542 (21)	1214 (25)
$m_{KL} = 6$	4955 (19)	1142 (21)

TABLE 3.6: Analytical example. (Top) The average set estimation error obtained by the Fpca and the \mathbb{L}^2 -GFQ methods for different values of m (in brackets are the standard deviation). (Bottom) The average cumulative number of calls to the function f (in brackets are the number of iterations required to reach the stopping criterion in the SUR strategy).

Table 3.6 shows that for all values of m , the \mathbb{L}^2 -GFQ method outperforms the probabilistic Fpca modelling. As shown in Table 3.7, for high truncation argument, the explained variance increases, that explains the decrease of the estimation error for the probabilistic modelling (Fpca). In the other hand, the \mathbb{L}^2 -GFQ accuracy seems to be quite constant for $m_{KL} \geq 3$. This can be explained by the fact that the KL expansion is only used to define a space filling design, and the information lost by the truncation is recovered by tacking the corresponding curve in the set Ξ . On the contrary, the probabilistic modelling which is based on Fpca gives better results when m is higher. Figures 3.12 and 3.13 show the initial and the final results obtained by the two methods for $m_{KL} = 2$. However the errors in Table 3.6 seem to be bounded below. To go below that bound, we probably need to increase the size of Ξ .

m_{KL}	2	3	4	5	6
Explained variance	90.2 %	93.4 %	95.1 %	96 %	96.7%

TABLE 3.7: Analytical example. The explained variance in function of m_{KL}

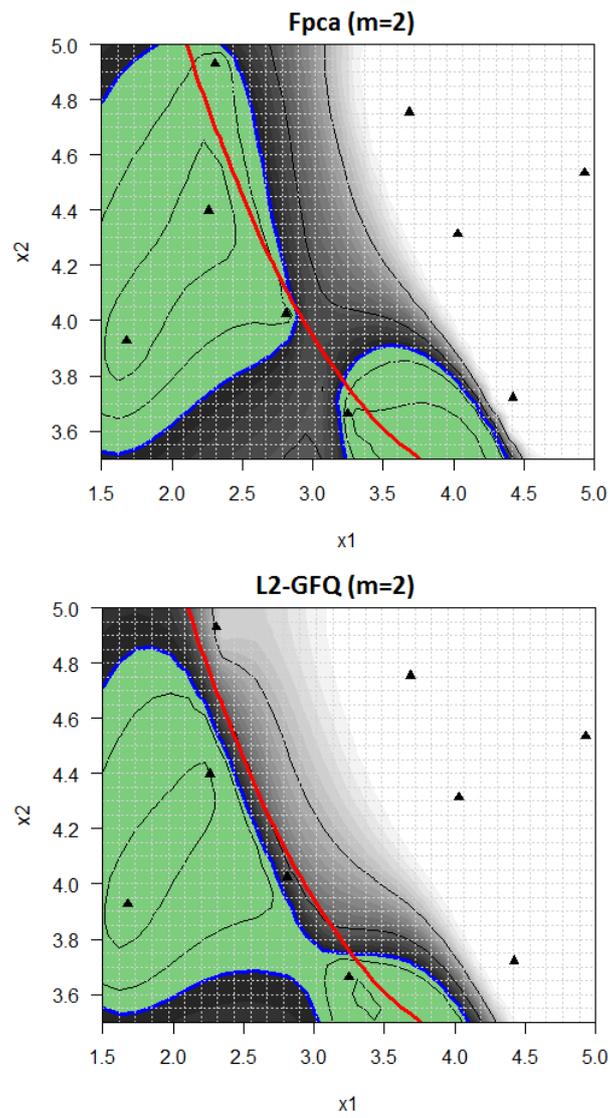


FIGURE 3.12: Analytical example. Results based on the initial DoE in the case $m_{KL} = 2$. The coverage function, the boundary of the true set (red), the estimated sets (green).

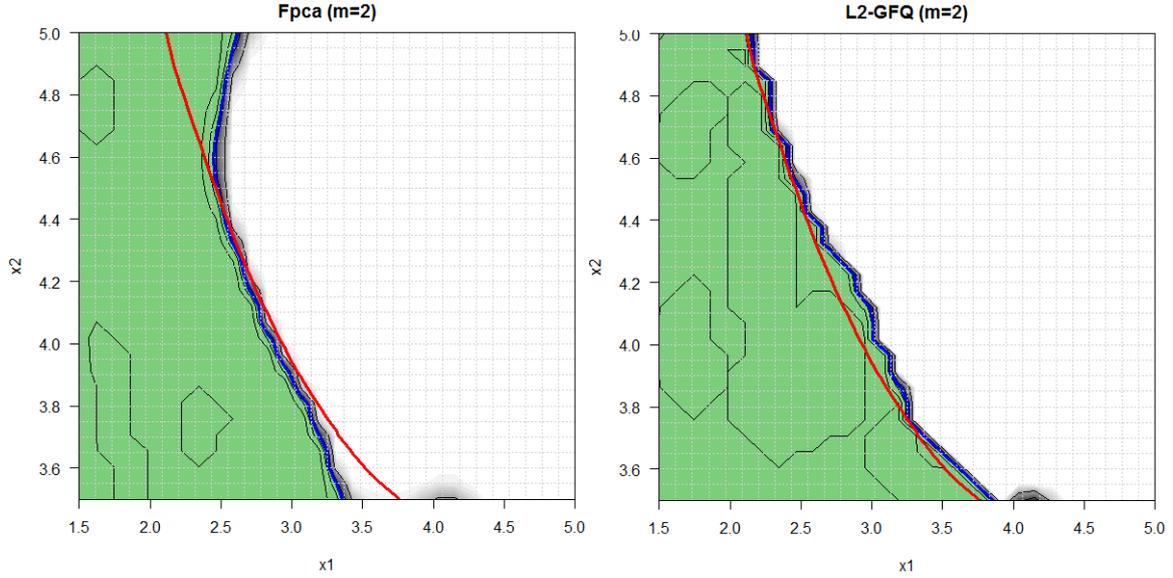


FIGURE 3.13: Analytical example. Results at the last iteration in the case $m_{KL} = 2$. The coverage function, the boundary of the true set (red), the estimated sets (green).

3.6.2 IFPEN test case

In this section we test the proposed method on an automotive test case from IFPEN. The problem concerns an after-treatment device of diesel vehicles, called Selective Catalytic Reduction (SCR). This latter consists on a basic process of chemical reduction of nitrogen oxides (NO_x) to diatomic nitrogen (N_2) and water (H_2O) by the reaction of NO_x and ammonia NH_3 . The reaction itself occurs in the SCR catalyst. Ammonia is provided by a liquid-reductant agent injected upstream of the SCR catalyst. The amount of ammonia introduced into the reactor is a critical quantity : overdosing causes undesirable ammonia slip downstream of the catalyst, whereas under-dosing causes insufficient NO_x reduction. In practice, ammonia slip is restricted to a prescribed threshold.

We use an emission-oriented simulator developed by IFPEN, which models the vehicle, its engine and the exhaust after-treatment system. It takes the vehicle driving cycle profile as input and provides the time-series of corresponding exhaust emissions as output. A realistic SCR control law is used in this simulator. See [Bonfils et al., 2012] for an example of such a control law. In this study, we choose two control variables as input and a functional one considered as random. The control variables are parameters of the SCR control law. They set the targeted level of NH_3 storage in the catalyst and then are indirectly related to the NH_3 injected. They lie in $\mathbb{X} = [0, 0.6]^2$. The functional random variable describes the evolution of vehicle speed on I , with $I = [0, 5400s]$. These functional uncertainties come from an available sample of 100 real driving cycles. Regarding the discretization of the real-driving cycle, we have one observation per second, so $\forall i \in \{1, \dots, 100\}$, $\mathbf{v}_i \in \mathbb{R}^{5400}$. A subset of that sample is represented in Figure 3.14.

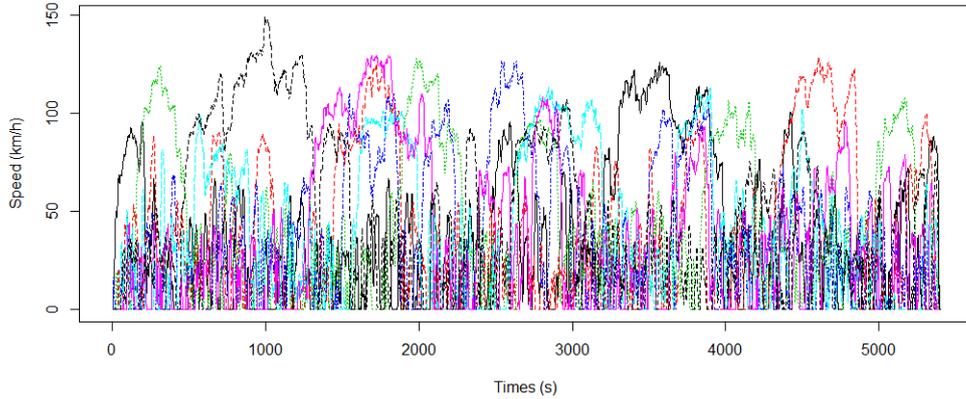


FIGURE 3.14: Automotive test case. Sample of 7 real driving cycles.

In short, the ammonia emissions peak during a driving cycle is modelled as a function,

$$\begin{aligned}
 f & : \mathbb{X} \times \mathcal{V} \rightarrow \mathbb{R} \\
 (\mathbf{x}, \mathbf{V}) & \mapsto f(\mathbf{x}, \mathbf{V}) = \max_{t \in I} \text{NH}_3^{\text{slip}}(t, \mathbf{x}, \mathbf{V})
 \end{aligned}$$

We are interested in recovering the set $\Gamma^* = \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})] \leq c\}$, with $c = 30 \text{ppm}$. Conducting this study on a full grid would consist in covering the space $[0, 0.6]^2$ with a fine mesh and evaluating the code 100 times at each point. Knowing that each simulation takes about two minutes, such study would require many computational hours, and thus the use of meta-models allows to tackle this computational issue.

Here we consider a Gaussian process prior $(Y_{\mathbf{x}})_{\mathbf{x} \in \mathbb{X}} \sim GP(m, k)$, with constant mean function and Matérn covariance kernel with $\nu = 5/2$. The initial DoE consists of a 8 points LHS design optimized with respect to the maximin criterion. The covariance kernel hyper-parameters are estimated by maximizing the likelihood.

As for the analytical example, we proceed to add one point at each iteration for the SUR strategy until the stopping criterion with $(l_0^{\text{SUR}}, \epsilon^{\text{SUR}}) = (4, 5 \cdot 10^{-3})$ is verified. Concerning the expectation estimation, we set the stopping criterion parameters at $(l_0, \epsilon) = (4, 10^{-2})$ and the truncation argument is set at $m_{KL} = 20$ in order to explain 80% of the variance. The algorithm was stopped at the 62-point design because the Vorob'ev deviation appears to have stabilized, in other words, the absolute error between the Vorob'ev deviations of the points 58-62 are smaller than 0.005, as shown in Figure 3.15. We note that for each additional point, the new observed response affects the estimation of the excursion set and its uncertainty. Thus, although the Vorob'ev deviation generally decreases, it is not a monotonic decreasing. The stopping criterion is constructed to check the stability of convergence by taking into account the last four iterations. In searching for the true set, the SUR algorithm heavily visits the boundary region of Γ^* and allows itself to explore also potentially interesting regions (cf. Fig. 3.16). In each added point, Figure 3.16 shows the number of necessary driving cycles to estimate the expectation. We remark that instead of taking the whole sample (100 driving cycles), it was sufficient to sequentially and wisely choose a reduced and representative number of driving cycles below 35. In the present case, the excursion domain Γ^* is well recovered by the algorithm. Actually, after 62 iterations (1575 evaluations) the whole domain \mathbb{X} has an excursion probability close to either 0 or 1.

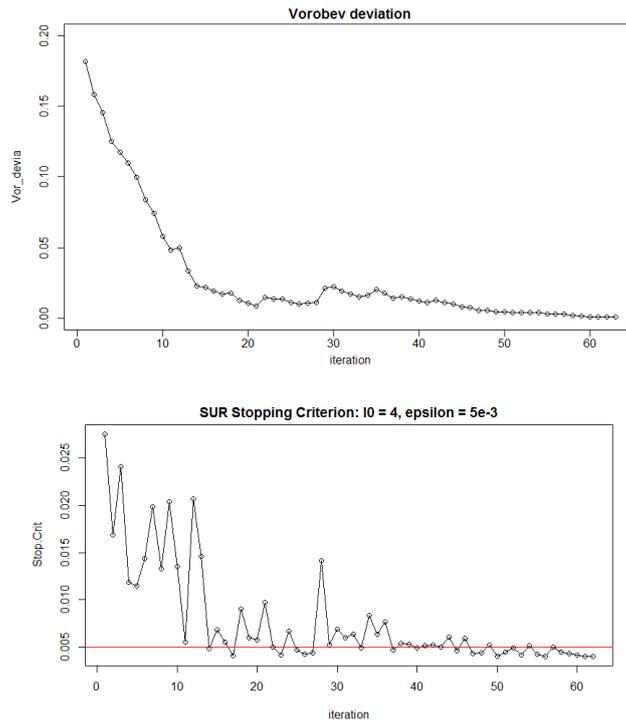


FIGURE 3.15: Automotive test case. Top : Decrease of the Vorob'ev deviation at each iteration when new points are added. Bottom : Evolution of the absolute error Eq. (3.31) and the red line represents the stopping criterion.

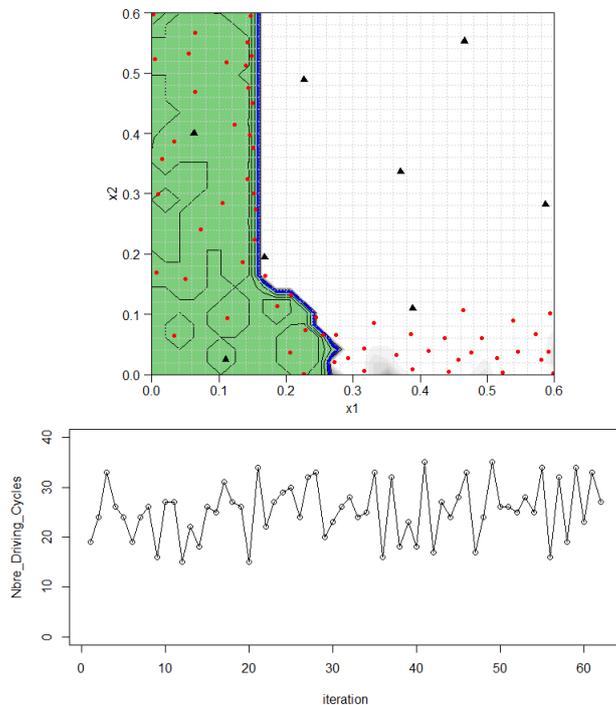


FIGURE 3.16: Automotive test case. Top : Coverage probability function (grey scale), estimate set (green) after 62 added points and 1575 function evaluations, initial DoE (black triangles), the sequentially added points (red circles). Bottom : number of driving cycles used to estimate expectation at each added point.

3.7 Conclusions

In this paper, a new method of inversion under uncertainty was proposed for problems where some of the input parameters are functional random variables with unknown probability distribution (only a sample is available). The objective is to recover the set of control variables leading to robustly ensure some constraints by taking into account the uncertainties. The method is composed of two steps : a sequential strategy to estimate the excursion set, and the modelling of functional uncertainties. To solve the first issue a kriging model in the control input space is built. It makes possible to assess the uncertainty on the set of interest given a sample of evaluations. Then a sequential strategy (SUR) proposed by [Chevalier et al., 2013] and based on the kriging model is used to sequentially and efficiently choose new evaluation points to improve the excursion set estimation. For the second issue, we consider the expectation to model uncertainties and we propose two sequential approaches to estimate the expectation at each point proposed by SUR. Each curve is represented by its coefficients in a truncated KL decomposition. The chosen points in the KL coefficients finite set, each one corresponding to a curve, are sequentially added and chosen either to approximate a maximin space filling design or to reduce the quantization error. This methodology leads to an efficient estimation of the expectation, as illustrated on the application on an analytical test case with two control inputs and a functional random one. The results illustrate significant enhancement in term of precision and number of calls to the simulator in comparison with the Monte Carlo based methods. We also applied this method to the automotive test case which motivated this research work. The obtained result agrees with the intuitions made from physics behind the simulator.

The paper focuses on the mean of $f(\mathbf{x}, \mathbf{V})$ and here we choose to construct a GP model for the unobservable integrated response g . In the optimization context and for discrete and continuous random variables, existing works deal with the case of unobservable response (see [Williams et al., 2000],[Janusevskis and Le Riche, 2013]). The authors propose to build a GP model for the simulator f and induce a new GP model by integrating the one on f over the distribution of the uncertain variables. The adaptation of these works in the context of inversion and functional random variables is an on-going work and will be presented in chapter 4. Other than the expectation other functionals of the output distribution may also be of great importance. For example, practitioners may be interested in ensuring a certain level of reliability, leading to consider a probabilistic constraint. The proposed method could be adapted to that case by seeing the probability as an expectation, at least for moderate risk levels.

3.8 Complément de l'article : un nouvel exemple

Ce complément de l'article présente un nouvel exemple d'application pour souligner la robustesse des algorithmes de quantification par rapport à l'argument de troncature de la décomposition de Karhunen-Loève dans le cadre de l'inversion stochastique. Pour ce faire, on considère le même exemple analytique introduit dans la section 3.6.1 :

$$f : (\mathbf{x}, \mathbf{v}) \mapsto \max_t \mathbf{v}_t \cdot |0.1 \cos(x_1 \max_t \mathbf{v}_t) \sin(x_2) \cdot (x_1 + x_2 \min_t \mathbf{v}_t)^2| \cdot \int_0^T (30 + \mathbf{v}_t)^{\frac{x_1 \cdot x_2}{20}} dt,$$

où \mathbf{x} est la variable de contrôle définie dans $\mathbb{X} = [1.5, 5] \times [3.5, 5]$ et \mathbf{V} un processus max-stable. On suppose qu'on dispose d'un échantillon de 200 réalisations de \mathbf{V} . L'objectif est d'identifier l'ensemble d'excursion $\Gamma^* := \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})] \leq 0.9\}$.

On suppose que la fonction g est estimée sur un plan d'expériences de taille $n = 9$. On fait une modélisation par processus gaussien tout en tenant compte des observations comme cela a été décrit dans la section 1.1.3. On prend un noyau de covariance de type Matérn 5/2 et les paramètres de ce noyau sont estimés par maximum de vraisemblance. On fixe les critères d'arrêt à $(l_0, \epsilon) = (4, 10^{-2})$ pour l'estimation de l'espérance et à $(l_0^{\text{SUR}}, \epsilon^{\text{SUR}}) = (4, 5 \cdot 10^{-3})$ pour la stratégie SUR.

Les algorithmes gloutons de quantification fonctionnelle ainsi que la méthode "Fpca" dépendent de l'argument de troncature utilisé dans la décomposition de Karhunen-Loève. Pour mieux comprendre l'effet de cette grandeur, on considère différentes valeurs pour m_{KL} (cf. tableau 3.8).

m_{KL}	2	4	8	10
Part de variance expliquée	62.2 %	67.2 %	74.1 %	76.9 %

TABLE 3.8: La part de variance expliquée en fonction de l'argument de troncature m_{KL} .

On rappelle que la méthode \mathbb{L}^2 -GFQ est déterministe contrairement aux approches maximin-GFQ et Fpca qui sont stochastiques par construction pour l'une et par nature pour l'autre. Ainsi compte tenu de cette aléa, on répète 30 fois la résolution du problème.

Le tableau 3.9 résume les résultats obtenus par les 3 méthodes. En ce qui concerne la méthode "Fpca", on note qu'elle très sensible à l'argument de troncature puisque plus on l'augmente plus on réduit l'erreur d'estimation. Mais cette réduction d'erreur se paie par une forte augmentation du nombre d'appels à la fonction f . On remarque aussi que les deux algorithmes gloutons sont beaucoup moins sensibles à l'argument de troncature tout en offrant des erreurs d'estimation très satisfaisantes et ce, à faible coût en évaluations de la fonction. On note qu'il existe toutefois un compromis entre les deux algorithmes gloutons (la précision et le coût en évaluations de la fonction). En effet avec moins d'appels à la fonction, la méthode maximin-GFQ fournit une erreur un peu moins bonne que la méthode \mathbb{L}^2 -GFQ.

m_{KL}	$\mu(\Gamma^* \Delta Q_{n_{\text{last}}, \alpha_{n_{\text{last}}}^*}) / \mu(\Gamma^*)$		
	Fpca	maximin-GFQ	\mathbb{L}^2 -GFQ
$m_{KL} = 2$	30.84 % (6.25)	2.51 % (0.97)	2.26 %
$m_{KL} = 4$	23.53 % (5.77)	2.28 % (0.66)	1.73 %
$m_{KL} = 8$	20.93 % (4.41)	1.76 % (0.53)	1.56 %
$m_{KL} = 10$	20.34 % (4.05)	1.83 % (0.81)	1.14 %
m_{KL}	Nombre total d'appels à la fonction f		
	Fpca	maximin-GFQ	\mathbb{L}^2 -GFQ
$m_{KL} = 2$	2121 (21)	675 (14)	1043 (26)
$m_{KL} = 4$	2891 (22)	656 (14)	886 (24)
$m_{KL} = 8$	4236 (23)	698 (15)	810 (23)
$m_{KL} = 10$	4699 (21)	692 (14)	839 (24)

TABLE 3.9: L'erreur d'estimation ou sa moyenne (dans le cas des méthodes "Fpca" et maximin-GFQ) obtenue par les 3 méthodes pour différentes valeurs de m_{KL} (entre parenthèses, l'écart type). La moyenne du nombre total d'appels à la fonction f (entre parenthèses, le nombre d'itérations nécessaires pour atteindre le critère d'arrêt dans la stratégie SUR).

La figure 3.17 montre le résultat d'estimation pour la méthode \mathbb{L}^2 -GFQ et aussi un exemple

d'estimation pour les deux autres méthodes en fixant les critères d'arrêt à $(l_0, \epsilon) = (4, 10^{-2})$ pour l'estimation de l'espérance et à $(l_0^{\text{SUR}}, \epsilon^{\text{SUR}}) = (4, 5.10^{-3})$ pour la stratégie SUR. Nous concluons de cet exemple que l'étape consacrée à la récupération des courbes associées aux coefficients dans l'espace réduit de la décomposition de Karhunen-Loève après une quantification dans l'espace réduit, rend les deux algorithmes gloutons robustes par rapport à la taille de l'espace réduit.

Synthèse : Dans ce chapitre, on a proposé une méthode permettant de résoudre un problème d'inversion sous incertitude fonctionnelle pour des simulateurs numériques coûteux en temps de calcul.

Dans un cadre bayésien, on s'est appuyé sur la stratégie SUR pour un enrichissement adaptatif de plans d'expériences dans l'espace de contrôle. Cette stratégie couplée aux algorithmes de quantification fonctionnelle gloutonne mène à une estimation efficace de l'ensemble recherché. A travers divers exemples, on a montré la robustesse de la méthodologie proposée par rapport aux méthodes standards. En termes de précision, la méthodologie basée sur l'algorithme \mathbb{L}^2 -GFQ fournit une estimation plus précise de l'ensemble recherché par rapport à la méthode maximin-GFQ. Par contre en terme de coût de simulations (nombre d'appels au simulateur), on a remarqué que cette dernière mène à une estimation satisfaisante avec moins d'appels à la fonction. Ainsi le choix entre les deux méthodologies peut se faire en fonction du budget de simulations consacré à la résolution du problème.

Le chapitre suivant propose une alternative à la méthode présentée dans ce chapitre. Cette méthode a pour but de réduire encore plus le nombre d'appels au simulateur numérique grâce à une stratégie d'enrichissement adaptatif du plans d'expériences dans l'espace de toutes les entrées (variables de contrôle et incertaines).

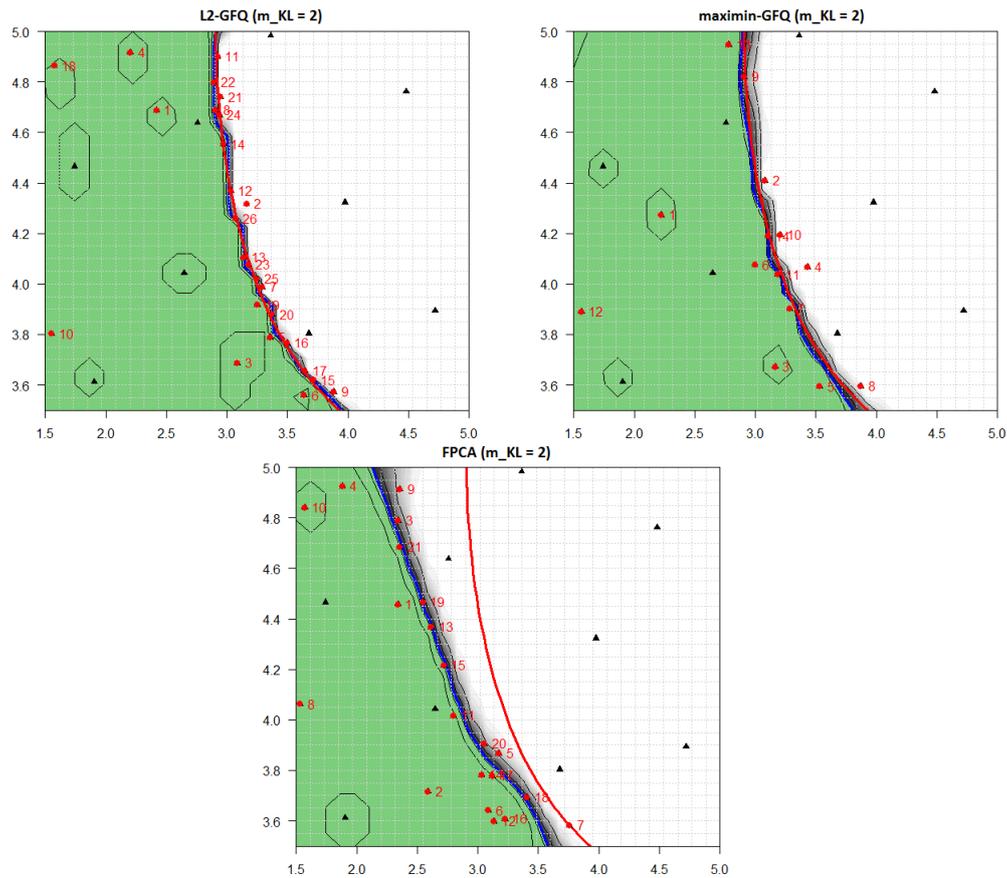


FIGURE 3.17: Résultats d'estimation pour $m_{KL} = 2$ et pour des critères d'arrêt fixés à $(l_0, \epsilon) = (4, 10^{-2})$ et $(l_0^{\text{SUR}}, \epsilon^{\text{SUR}}) = (4, 5.10^{-3})$. L'ensemble estimé par la méthode L^2 -GFQ (haut à gauche), la méthode maximin-GFQ à l'itération 14 (haut à droite) et par la méthode "Fpca" à l'itération 21.

Chapitre 4

Stochastic inversion under functional uncertainty

Outlines

In this chapter we propose a new method for solving an inversion problem under functional uncertainties with expensive-to-evaluate functions. The simulation inputs are divided into two sets, the deterministic control variables and a functional uncertain variable. Unlike modelling in the control space only, as presented in the previous chapter, the methodology we propose in this chapter to tackle stochastic inversion under functional uncertainty is based on Gaussian Process Emulation with respect to both deterministic control variables and stochastic functional variable. The method is compared with *greedy functional quantization* based approach (presented in Chapter 3) on two analytical test functions.

Contents

4.1	Introduction	69
4.2	Problem formulation	70
4.2.1	Finite dimensional subproblem	71
4.2.2	Gaussian process modelling	71
4.2.3	GP model of the unobservable response	72
4.3	Infill strategy for stochastic inversion in presence of functional uncertainty	72
4.3.1	Minimization of the Vorob'ev deviation : choice of next \mathbf{x}	73
4.3.2	Minimization of the variance : choice of next \mathbf{u}	73
4.4	Implementation	75
4.5	Numerical tests	76
4.6	Conclusion	79

4.1 Introduction

In the first part of this thesis we have introduced on the one hand, the concept of inversion through surrogate modelling and, on the other hand an original greedy methodology for expectation estimation over a functional random variable. In Chapter 3, the two items are combined for an efficient stochastic inversion under functional uncertainty. In other words, we proposed a method leading to identify the set $\Gamma^* = \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})] \leq c\}$, where $c \in \mathbb{R}$. This combined approach adopts an *adaptive design of experiments* based on surrogate models to replace the unobservable response g and a sequential expectation estimation via *functional quantization*.

In this chapter we introduce a new methodology to estimate the set Γ^* with the aim of further reducing the number of calls to the numerical simulator. The proposed methodology is based on a surrogate model of the simulator f and an adaptive enrichment of the design of experiment for the estimation of the set of interest Γ^* .

In the context of robust optimization, there are various methods (see [Janusevskis and Le Riche, 2013] or [Williams et al., 2000]) for minimizing with respect to the deterministic variables the expected objective where the expectation is taken over a vector-valued random variable. These methods are based on a kriging model built in the joint space of deterministic and uncertain variables. Then an averaged Gaussian process over the random variables is derived. This latter, "projected" Gaussian process, approximates the expected response. Finally an adaptive design of experiments is proposed for minimizing the expected objective

In the same spirit, we propose an original method to deal with a stochastic inversion problem. In this work a Gaussian process model is created to approximate the simulator f that takes as inputs controlled variables and an uncertain functional variable. The samples in the joint space of control and uncertain variables, \mathbf{x} and \mathbf{v} , are selected based on a two steps process. First the "projected" process which only depends upon \mathbf{x} is considered and the Vorob'ev deviation is minimized to determine the next evaluation point in the control space. Second, in the uncertain space, the next point is chosen such that it minimizes the standard error of the "projected" process evaluated at the proposed point in the first step. Compared to the method proposed in Chapter 3, this adaptive design of experiments defined in the joint space will lead us to further reduce the number of calls to the numerical simulator.

The chapter is structured as follows. Firstly, in Section 4.2, we recall the problem formulation and we extend the concept of Gaussian process modelling to the case where the inputs contain functional variable. In Section 4.3, an infill strategy define the next couple of (\mathbf{x}, \mathbf{v}) which is calculated by the simulator f . We describe the whole algorithm in Section 4.4. Finally, in Section 4.5, this method is compared to those proposed in Chapter 3 on two analytical test cases.

4.2 Problem formulation

We remind the problem to be solved : let

$$f : \mathbb{X} \times \mathcal{V} \rightarrow \mathbb{R}$$

denote the response of a numerical simulator. $\mathbb{X} \subset \mathbb{R}^p$ is the search space of the control variables and the stochasticity of the simulator is driven by a random variable defined in a functional space \mathcal{V} . We assume that the probability distribution of the functional input is only known through a finite set of realizations $\Xi = \{\mathbf{v}_1, \dots, \mathbf{v}_N\}$ and each evaluation of f involves a time consuming call to a deterministic simulator. In this study we are concerned with estimating the set

$$\Gamma^* = \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})] \leq c\}, \quad (4.1)$$

where $\mathbf{x} \in \mathbb{X}$ are the deterministic control variables, $\mathbb{E}_{\mathbf{V}}$ is the expectation with respect to the functional uncertain variable \mathbf{V} and $c \in \mathbb{R}$ is a threshold. Let's denote the initial design points as $\mathcal{X}_n = \{\mathbf{x}_{(1)}, \dots, \mathbf{x}_{(n)}\}$ and $\mathcal{V}_n = \{\mathbf{v}_{(1)}, \dots, \mathbf{v}_{(n)}\}$, where $\forall i \in \{1, \dots, n\}$, $\mathbf{v}_{(i)} \in \Xi$. The simulator responses at these design points are denoted by $\mathbf{Z} = \{f(\mathbf{x}_{(1)}, \mathbf{v}_{(1)}), \dots, f(\mathbf{x}_{(n)}, \mathbf{v}_{(n)})\}$.

The estimation of Γ^* by evaluating the function g at each grid point of the discretized version of \mathbb{X} requires far too many calls to the simulator f . The method proposed in Chapter

3 makes it possible to partially overcome this problem by using functional quantization techniques in order to reduce the cost of the expectation estimation. In the following we propose an alternative sampling strategy based on an adaptation of existing SUR strategies.

Among statistical models, the ones based on Gaussian processes have received increasing interests in recent years, due to many of their desirable properties, such as explicit formulation of the predictor and the associated uncertainty. However, in the literature the input variables involved in Gaussian process models are mostly univariate or multivariate. The purpose of this section is to extend the use of Gaussian process model to cases where the inputs contain infinite dimensional variables or functional data which are collected as curves. In this Section, we define an infill sampling criterion based on GP model in order to solve the stochastic inversion problem (4.1).

4.2.1 Finite dimensional subproblem

Several methods can be applied to reduce the dimension of the functional space \mathcal{V} (see Section 2.2.1). Here we adopt the Karhunen-Loève expansion which is the most commonly employed method to reduce the statistical complexity of random fields. A direct application of the KL expansion leads us to represent \mathbf{V} by the random vector $\mathbf{U} \in \mathbb{R}^{m_{KL}}$ defined by the first m_{KL} principal components of \mathbf{V} and we write

$$\mathbf{V}(t) \simeq \sum_{i=1}^{m_{KL}} \mathbf{u}_i \psi_i(t), \quad t \in [0, T],$$

where $\{\psi_i\}_{i=1}^{m_{KL}}$ are orthogonal and normalized eigenfunctions of the integral operator corresponding to C , the covariance function of \mathbf{V} . The $\{\mathbf{u}_i\}_{i=1}^{m_{KL}} = \{\langle \mathbf{V}, \psi_i \rangle\}_{i=1}^{m_{KL}}$ denotes a set of orthogonal random variables with zero mean and variances λ_i , where $\lambda_1 \geq \lambda_2 \geq \dots > 0$ are the ordered non-zero eigenvalues of C .

In this finite space, the uncertainties will be characterized by the random vector $\mathbf{U} \in \mathbb{R}^{m_{KL}}$. This latter point leads us to build a Gaussian process model in the joint space $\mathbb{X} \times \mathbb{R}^{m_{KL}}$.

4.2.2 Gaussian process modelling

Let $Z_{(\mathbf{x}, \mathbf{u})}$ denote the Gaussian process defined on $\mathbb{X} \times \mathbb{R}^{m_{KL}}$ with mean function m_Z and covariance function k_Z (see Chapter 1),

$$\begin{aligned} \mathbb{E}[Z_{(\mathbf{x}, \mathbf{u})}] &= m_Z(\mathbf{x}, \mathbf{u}), \\ \text{Cov}(Z_{(\mathbf{x}, \mathbf{u})}, Z_{(\mathbf{x}', \mathbf{u}')} &= k_Z(\mathbf{x}, \mathbf{u}; \mathbf{x}', \mathbf{u}'). \end{aligned} \quad (4.2)$$

We assume that $f(\mathbf{x}, \mathbf{v})$ can be approximated by $z(\mathbf{x}, \mathbf{u})$ which is a realization of $Z_{(\mathbf{x}, \mathbf{u})}$, where $\mathbf{u} = (\langle \mathbf{v}, \psi_1 \rangle, \dots, \langle \mathbf{v}, \psi_{m_{KL}} \rangle)^\top$. Let $Z_{(\mathbf{x}, \mathbf{u})}^n$ denote a GP conditioned on the n observations

$$Z_{(\mathbf{x}, \mathbf{u})}^n = [Z_{(\mathbf{x}, \mathbf{u})} | Z_{(\mathcal{X}_n, \mathcal{U}_n)} = \mathbf{Z}], \quad (4.3)$$

where $\mathcal{X}_n = \{\mathbf{x}_{(1)}, \dots, \mathbf{x}_{(n)}\}$, $\mathcal{U}_n = \{\mathbf{u}_{(1)}, \dots, \mathbf{u}_{(n)}\}$, $\mathbf{Z} = \{f(\mathbf{x}_{(1)}, \mathbf{v}_{(1)}), \dots, f(\mathbf{x}_{(n)}, \mathbf{v}_{(n)})\}$ and $\forall i \in \{1, \dots, n\}$, $\mathbf{u}_{(i)} = (\langle \mathbf{v}_{(i)}, \psi_1 \rangle, \dots, \langle \mathbf{v}_{(i)}, \psi_{m_{KL}} \rangle)^\top$. The conditional expectation and the conditional covariance are

$$\begin{aligned} \mathbb{E}[Z_{(\mathbf{x}, \mathbf{u})}^n] &= m_Z(\mathbf{x}, \mathbf{u}) + k_Z(\mathbf{x}, \mathbf{u}; \mathcal{X}_n, \mathcal{U}_n) k_Z(\mathcal{X}_n, \mathcal{U}_n; \mathcal{X}_n, \mathcal{U}_n)^{-1} (Z - m_Z(\mathcal{X}_n, \mathcal{U}_n)) \\ \text{Cov}(Z_{(\mathbf{x}, \mathbf{u})}^n, Z_{(\mathbf{x}', \mathbf{u}')}^n) &= k_Z(\mathbf{x}, \mathbf{u}; \mathbf{x}', \mathbf{u}') \\ &\quad - k_Z(\mathbf{x}, \mathbf{u}; \mathcal{X}_n, \mathcal{U}_n) k_Z(\mathcal{X}_n, \mathcal{U}_n; \mathcal{X}_n, \mathcal{U}_n)^{-1} k_Z(\mathcal{X}_n, \mathcal{U}_n; \mathbf{x}', \mathbf{u}'). \end{aligned}$$

These equations allow us to estimate the predicted kriging mean and the predicted kriging variance of $Z_{(\mathbf{x}, \mathbf{u})}^n$ at any unobserved new point (\mathbf{x}, \mathbf{u}) . In practice the covariance and mean functions of the Gaussian process are not known. Usually isotropic stationary kernel functions are used to model covariance and simple polynomial functions are used to model the mean function. The parameters of these functions are estimated from the data by maximizing the kriging model likelihood (see e.g. Section 1.1.3).

4.2.3 GP model of the unobservable response

In this section we take full advantage of the Gaussian process eq. (4.3) to identify the set Γ^* defined by the expectation eq. (4.1). Indeed it is possible to induce a Gaussian process by taking the expectation of the conditional process with respect to \mathbf{U}

$$Y_{\mathbf{x}}^n = \mathbb{E}_{\mathbf{U}}[Z_{(\mathbf{x}, \mathbf{U})}^n] = \int_{\mathbb{R}^{m_{KL}}} Z_{(\mathbf{x}, \mathbf{u})}^n d\rho(\mathbf{u}), \quad (4.4)$$

where $d\rho(u)$ is a probability measure on \mathbf{u} . We note that $Y_{\mathbf{x}}^n$ is a linear combination of Gaussian variables and therefore $(Y_{\mathbf{x}}^n)_{\mathbf{x} \in \mathbb{X}}$ is still a Gaussian process fully characterized by its mean and covariance functions which are given by

$$\mathbb{E}[Y_{\mathbf{x}}^n] = \int_{\mathbb{R}^{m_{KL}}} m_Z(\mathbf{x}, \mathbf{u}) d\rho(u) + \int_{\mathbb{R}^{m_{KL}}} k_Z(\mathbf{x}, \mathbf{u}; \mathcal{X}_n, \mathcal{U}_n) k_Z(\mathcal{X}_n, \mathcal{U}_n; \mathcal{X}_n, \mathcal{U}_n)^{-1} (Z - m_Z(\mathcal{X}_n, \mathcal{U}_n)) d\rho(\mathbf{u}), \quad (4.5)$$

$$\begin{aligned} \text{Cov}(Y_{\mathbf{x}}^n, Y_{\mathbf{x}'}^n) &= \int_{\mathbb{R}^{m_{KL}}} \int_{\mathbb{R}^{m_{KL}}} k_Z(\mathbf{x}, \mathbf{u}; \mathbf{x}', \mathbf{u}') d\rho(u) d\rho(u') \\ &\quad - \int_{\mathbb{R}^{m_{KL}}} \int_{\mathbb{R}^{m_{KL}}} k_Z(\mathbf{x}, \mathbf{u}; \mathcal{X}_n, \mathcal{U}_n) k_Z(\mathcal{X}_n, \mathcal{U}_n; \mathcal{X}_n, \mathcal{U}_n)^{-1} k_Z(\mathcal{X}_n, \mathcal{U}_n; \mathbf{x}', \mathbf{u}') d\rho(\mathbf{u}) d\rho(\mathbf{u}'). \end{aligned} \quad (4.6)$$

In the case where the uncertain variables \mathbf{U} are Gaussian as well as the covariance kernel, closed form solutions of the integrals in eq. (4.5) and eq. (4.6) are given in [Janusevskis and Le Riche, 2013]. In our context, no additional assumptions are made about the kernel or the distribution of the uncertain variables [no particular kernel or uncertain variables are assumed]. Thus standard Monte Carlo simulations are needed to approximate the equations in (4.5) and (4.6).

4.3 Infill strategy for stochastic inversion in presence of functional uncertainty

To solve the stochastic inversion problem 4.1, we follow a Stepwise Uncertainty Reduction strategy (see Section 1.3). We propose an infill sampling criterion that relies on the GP models of $f(\mathbf{x}, \mathbf{v})$ and $g(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})]$ given by the eq. (4.3) and eq. (4.4), respectively.

The proposed infill strategy consists in two steps. First we choose $\mathbf{x}_{(n+1)}$ by minimizing the Vorob'ev deviation (see Section 1.3.1) defined by the process $Y_{\mathbf{x}}^n$. Then we choose the point $\mathbf{u}_{(n+1)}$ that minimizes the variance of the process $Y_{\mathbf{x}}^n$ evaluated at the point $\mathbf{x}_{(n+1)}$. By doing so we expect an efficient estimation of the set Γ^* in terms of number of model evaluations.

4.3.1 Minimization of the Vorob'ev deviation : choice of next \mathbf{x}

The objective of the first step is to wisely choose the points in the control space \mathbb{X} in order to efficiently estimate the set $\Gamma^* = \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})] \leq c\}$. In this context, we adopt the strategy introduced in Section 1.3.1 which relies on the statistical model of the unobservable function g given by $Y_{\mathbf{x}}^n$. Indeed, due to the stochastic nature of $(Y_{\mathbf{x}}^n)_{\mathbf{x} \in \mathbb{X}}$, the associated excursion set,

$$\Gamma = \{\mathbf{x} \in \mathbb{X}, Y_{\mathbf{x}}^n \leq c\} \quad (4.7)$$

is a random closed set. From the assumption that g is a realisation of $Y_{\mathbf{x}}^n$, the true unknown set Γ^* can be seen as a realisation of the random closed set Γ .

Therefore, we implement a Stepwise Uncertainty Reduction strategy (SUR) that aims at reducing uncertainty on Γ by adding new evaluation points step by step as proposed in [Chevalier et al., 2013]. In this context such strategy relies on the notion of uncertainty for random sets. In what follows, we consider the Vorob'ev deviation as the uncertainty function, at step n ,

$$\mathcal{H}_n^{\text{uncert}} = \mathbb{E}[\mu(\Gamma \Delta Q_{n, \alpha_n^*}) \mid Z_{(\mathcal{X}_n, \mathcal{U}_n)} = \mathbf{Z}],$$

where μ is a Borel σ -finite measure defined on \mathbb{X} , the Vorob'ev quantiles are given by $Q_{n, \alpha} = \{\mathbf{x} \in \mathbb{X}, \mathbb{P}(Y_{\mathbf{x}}^n \leq c) \geq \alpha\}$, and the Vorob'ev expectation Q_{n, α_n^*} can be determined by tuning α to a level α^* such that $\mu(Q_{n, \alpha_n^*}) = \mathbb{E}[\mu(\Gamma) \mid Z_{(\mathcal{X}_n, \mathcal{U}_n)} = \mathbf{Z}]$.

The objective of the SUR strategy is to find r optimal locations $\{\mathbf{x}_{(n+1)}, \dots, \mathbf{x}_{(n+r)}\}$ such that the uncertainty $\mathcal{H}_{n+r}^{\text{uncert}}$ is as small as possible. One way to construct such a sequence is to choose at each step the point that gives the smallest uncertainty $\mathcal{H}_{n+1}^{\text{uncert}}$,

$$\mathcal{H}_{n+1}^{\text{uncert}}(\mathbf{x}) = \mathbb{E}[\mu(\Gamma \Delta Q_{n, \alpha_{n+1}^*}) \mid Z_{(\mathcal{X}_n, \mathcal{U}_n)} = \mathbf{Z}, Y_{\mathbf{x}}^n].$$

We note that the future $\mathcal{H}_{n+1}^{\text{uncert}}$ is a function of $Y_{(\mathbf{x})}^n$ given $Z_{(\mathcal{X}_n, \mathcal{U}_n)} = \mathbf{Z}$. Therefore we choose the point that minimizes most the uncertainty in *expectation*, that is :

$$\begin{aligned} \mathbf{x}_{(n+1)} &\in \operatorname{argmin}_{\mathbf{x} \in \mathbb{X}} \mathbb{E}_{n, \mathbf{x}}[\mathcal{H}_{n+1}^{\text{uncert}}(\mathbf{x})] \\ &:= \operatorname{argmin}_{\mathbf{x} \in \mathbb{X}} \mathcal{J}_n(\mathbf{x}), \end{aligned} \quad (4.8)$$

where $\mathbb{E}_{n, \mathbf{x}}$ denotes the expectation with respect to $Y_{(\mathbf{x})}^n \mid Z_{(\mathcal{X}_n, \mathcal{U}_n)} = \mathbf{Z}$ (for detailed formula of $\mathcal{J}_n(\cdot)$ see [Chevalier et al., 2013]).

Minimizing the Vorob'ev deviation gives a point $\mathbf{x}_{(n+1)}$ in the control space \mathbb{X} which aims at a better estimate of the unknown set Γ^* . However, the simulation of f requires also the choice of a value for the uncertain variables. The following step gives the next value $\mathbf{u} \in \mathbb{R}^{m_{KL}}$ and we induce the curve \mathbf{v} corresponding to that \mathbf{u} .

4.3.2 Minimization of the variance : choice of next \mathbf{u}

The process Y^n approximates $\mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})]$. It can be seen as a projection of Z^n from the joint space onto the control space and the variance of Y^n at a given \mathbf{x} depends on our knowledge about the simulator on the proximity of \mathbf{x} in the joint space. We propose to sample the point in the uncertain space such that it reduces the most the uncertainty of the estimated expectation at the point of interest $\mathbf{x}_{(n+1)}$. Therefore the next uncertain point $\mathbf{u}_{(n+1)}$ is taken such that

$$\mathbf{u}_{(n+1)} = \operatorname{arg} \min_{\mathbf{u} \in \mathbb{R}^{m_{KL}}} \operatorname{VAR}(Y_{\mathbf{x}_{(n+1)}}^n), \quad (4.9)$$

where the expression of the predicted variance of the process $Y_{\mathbf{x}(n+1)}^n$ is obtained from eq. (4.6)

$$\begin{aligned}
\text{VAR}(Y_{\mathbf{x}(n+1)}^n) &= \vartheta(\tilde{\mathbf{u}}) \\
&= \int_{\mathbb{R}^{m_{KL}}} \int_{\mathbb{R}^{m_{KL}}} k_Z(\mathbf{x}(n+1), \mathbf{u}; \mathbf{x}(n+1), \mathbf{u}') d\rho(\mathbf{u}) d\rho(\mathbf{u}') \\
&\quad - \int_{\mathbb{R}^{m_{KL}}} \int_{\mathbb{R}^{m_{KL}}} k_Z(\mathbf{x}(n+1), \mathbf{u}; \mathcal{X}_{n+1}, \mathcal{U}_{n+1}) \\
&\quad k_Z(\mathcal{X}_{n+1}, \mathcal{U}_{n+1}; \mathcal{X}_{n+1}, \mathcal{U}_{n+1})^{-1} k_Z(\mathcal{X}_{n+1}, \mathcal{U}_{n+1}; \mathbf{x}(n+1), \mathbf{u}') d\rho(\mathbf{u}) d\rho(\mathbf{u}'),
\end{aligned} \tag{4.10}$$

where $\mathcal{X}_{n+1} = (\mathcal{X}_n, \mathbf{x}(n+1))$, $\mathcal{U}_{n+1} = (\mathcal{U}_n, \tilde{\mathbf{u}})$ and $\mathcal{X}_n, \mathcal{U}_n$ are the already observed data points.

In our application context the functional random variable \mathbf{V} is only known through a sample of curves $\Xi = \{\mathbf{v}_1, \dots, \mathbf{v}_N\}$, and so after dimension reduction, the uncertain space is characterised by a set of points denoted by $\mathcal{G} \subset \mathbb{R}^{m_{KL}}$ and given by $\mathcal{G} = \{(\langle \mathbf{v}, \psi_1 \rangle, \dots, \langle \mathbf{v}, \psi_{m_{KL}} \rangle)^\top, \mathbf{v} \in \Xi\} = \{\mathbf{u}_i\}_{i=1}^N$, with $\mathbf{u}_i = (\langle \mathbf{v}_i, \psi_1 \rangle, \dots, \langle \mathbf{v}_i, \psi_{m_{KL}} \rangle)^\top$. In this setting, the next uncertain point \mathbf{u}^{n+1} is taken such that

$$\mathbf{u}_{(n+1)} = \arg \min_{\tilde{\mathbf{u}} \in \mathcal{G}} \text{VAR}(Y_{\mathbf{x}(n+1)}^n). \tag{4.11}$$

The sampling criterion of eq. (4.11) involves a discrete optimization, as well as the estimation of kriging hyper parameters and the minimization of Vorob'ev deviation eq. (4.8). More details are given in Section 4.4 .

The formula given by [Janusevskis and Le Riche, 2013] is detailed in the specific case where \mathbf{U} follows a Gaussian distribution. This is not our case : we do not know the distribution of \mathbf{U} but we have \mathcal{G} a sample from it. Therefore, we have to adapt this formula by replacing analytical integrals with standard Monte Carlo (MC) approximations.

4.4 Implementation

The complete approach to solve eq. (4.1) is summarized in Algorithm 8.

Algorithm 8 Stochastic inversion via joint space modelling

- 1: Create an initial DoE of n points in the joint space $(\mathbb{X}, \mathcal{V})$
 - 2: Calculate simulator responses at the design points
 - 3: **while** $n \leq \text{budget}$ **do**
 - 4: Fit a GP model $Z_{(\mathbf{x}, \mathbf{u})}^n$ of the simulator f
 - 5: Induce a GP model $Y_{\mathbf{x}}^n$ for the unobservable simulator g
 - 6: $\mathbf{x}_{(n+1)} \leftarrow$ Sampling criterion \mathcal{J}_n
 - 7: $\mathbf{u}_{(n+1)} \leftarrow \arg \min_{\hat{\mathbf{u}} \in \mathcal{G}} \text{VAR}(Y_{\mathbf{x}_{(n+1)}}^{n+1})$
 - 8: Simulator response at $(\mathbf{x}_{(n+1)}, \mathbf{v}_{(n+1)})$, where $\mathbf{v}^{n+1} \in \Xi$ is the associated curve to the $\mathbf{u}_{(n+1)}$
 - 9: Update DoE
 - 10: Set $n = n + 1$
 - 11: **end while**
 - 12: **end**
-

The algorithm involves 3 optimization subtasks which are solved successively at each iteration. We use `DiceKriging` package [Roustant et al., 2012] for the first optimization which consists in maximizing the likelihood to obtain covariance parameters. The function `genoud` is used to solve the second optimization (4.8). The last optimization (4.11) is discrete on a limited number of feasible values, thus it is not that time consuming.

To illustrate the performance of the proposed method, it is compared to the approach proposed in Chapter 3, where the expectation with respect to \mathbf{V} is estimated by *greedy functional quantization* using the simulator f , and the stochastic inversion is based on the kriging metamodel in the control space \mathbb{X} . The precision (fidelity level) of the estimation of the expectation is controlled by l the size of the quantizer (see Section 3.2.3). Like traditional SUR algorithms, our implementation minimizes at each iteration the Vorob'ev deviation calculated from a kriging model. This method is summarized in Algorithm 9. Note that the kriging model is built in p dimensions as $\mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})]$ is only a function of \mathbf{x} (contrarily to algorithm 8 where kriging is performed in a $p + m_{KL}$ dimensional space). We also note that each iteration leads to k calls to the simulator $f(\mathbf{x}, \mathbf{v})$.

Algorithm 9 Stochastic inversion via functional quantization

- 1: Create an initial DoE of n points in the control space \mathbb{X}
 - 2: Estimation by functional quantization using k calls to f
 - 3: **while** $n \leq \text{budget}$ **do**
 - 4: Fit a GP model $Y_{(\mathbf{x})}^n$ of the simulator in the control space \mathbb{X}
 - 5: $\mathbf{x}_{(n+1)} \leftarrow$ Sampling criterion \mathcal{J}_n
 - 6: Estimation by functional quantization $g(\mathbf{x}_{(n+1)}) \approx \mathbb{E}_{\hat{\mathbf{V}}_l}[f(\mathbf{x}_{(n+1)}, \hat{\mathbf{V}}_l)]$
 - 7: Update DoE
 - 8: Set $n = n + 1$
 - 9: **end while**
 - 10: **end**
-

4.5 Numerical tests

To illustrate the behaviour of the proposed algorithm, we consider two analytical examples. We suppose that a sample Ξ of $N = 200$ realizations of the functional random variable \mathbf{V} is available and its probability distribution is unknown. To highlight the robustness of our method regarding uncertainties, we consider two types of functional random variables : Brownian motion and Max-stable process. As Algorithm 8 depends on the truncation argument, we consider different values (see Table.4.1) to better understand the effect of the uncertain space dimension m_{KL} .

m_{KL}	2	4	8
\mathbf{V} : Brownian motion	90.1 %	95.2 %	97.6%
\mathbf{V} : Max-stable process	58.8 %	63.3 %	70%

TABLE 4.1: The explained variance in function of m_{KL} for two types of uncertainties.

For the following two analytical examples, we consider a Gaussian process prior $Z_{(\mathbf{x}, \mathbf{u})}$ with constant mean and Matèrn covariance kernel with $\nu = 5/2$. The Random Latin Hypercube design (RLHD) was used as an initial DoE for the two algorithms. The number of points of the initial DoE is $n = 5(p + m_{KL})$ and $n = 5p$ for the functional quantization based approach. The RLHD induces variability in the behaviour of the algorithms. To account for these variability in the tests, the performance of each method is averaged over 10 independent runs.

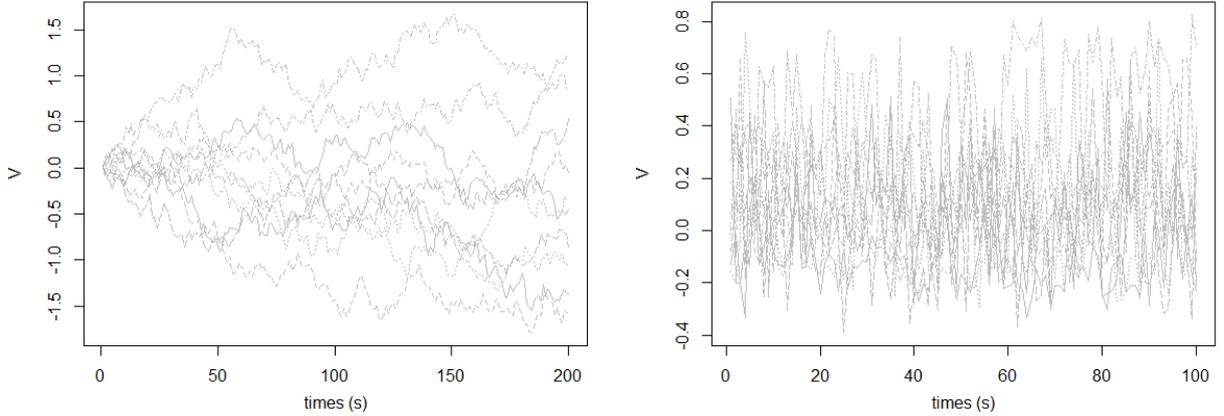


FIGURE 4.1: 10 realisations of the Brownian motion (left) and Max-stable process (right).

Analytical example 1. We consider an additive function, i.e., sum of the 2D Bohachevsky function and uncertainties defined as :

$$f : (\mathbf{x}, \mathbf{v}) \mapsto (x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7) + \int_0^T e^{\mathbf{v}t} dt,$$

where $\mathbf{x} \in \mathbb{X} = [-100, 100]^2$. The objective is to construct the sets $\Gamma^* = \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})] \leq 3500\}$ for two different uncertain functional variables (Brownian motion and Max-stable process).

Analytical example 2. For the second example we define a function that is not separable with respect to the control variables and uncertainties. The function involves the maximum

and the minimum of the function \mathbf{v} , so catching the whole variability of \mathbf{V} becomes important. The function f is given by

$$f : (\mathbf{x}, \mathbf{v}) \mapsto \max_t \mathbf{v}_t \cdot |0.1 \cos(x_1 \max_t \mathbf{v}_t) \sin(x_2) \cdot (x_1 + x_2 \min_t \mathbf{v}_t)^2|. \int_0^T (30 + \mathbf{v}_t)^{\frac{x_1 \cdot x_2}{20}} dt,$$

where the control variables lie in $\mathbb{X} = [1.5, 5] \times [3.5, 5]$ and as well as in the example above we consider two types of functional uncertainties (Brownian motion and Max-stable process). The objective is to construct the sets $\Gamma^* = \{\mathbf{x} \in \mathbb{X}, g(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})] \leq c\}$, where $c = 1.2$ and $c = 0.9$ for the Brownian motion and the Max-stable process respectively.

To compare the performance of the two algorithms we use the ratio of the volume of the symmetric difference between the true set Γ^* and the estimate set $Q_{n,\alpha^*} : \mu(\Gamma^* \Delta Q_{n,\alpha^*}) / \mu(\Gamma^*)$.

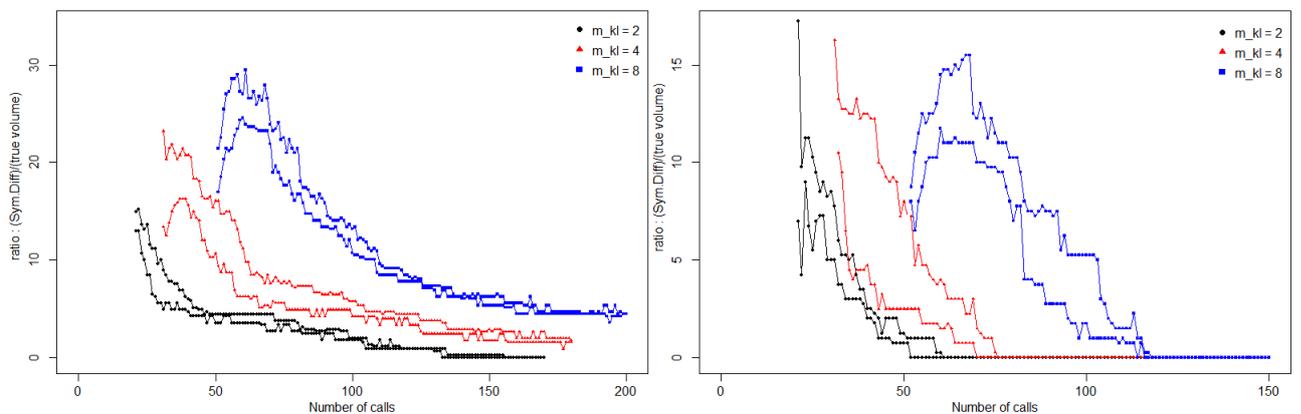


FIGURE 4.2: Analytical example 1 [Brownian motion (left), Max-stable process (right)]. Convergence of Algorithm 8 for $m_{KL} = \{2, 4, 8\}$. 25th and 75th percentiles of the symmetric difference vs. number of simulator calls. The curves are based on 10 random DoE for every run.

In Fig. 4.2 and 4.3 we show the 25th and 75th percentiles of convergence rates of Algorithm 8 for 10 runs of the complete approach for 3 values of the truncation argument m_{KL} . We notice that the smaller values of m_{KL} , the faster the convergence. This observation can be explained by the fact that, in higher dimensional joined space (due to larger values of m_{KL}), the Gaussian process model built to represent the simulator f may need much more points to learn an accurate GP model (more hyper parameters to determine). It may also be noted that even for 90% (for Brownian motion) or 58.8% (for Max-stable process) of explained variance with $m_{KL} = 2$ the proposed algorithm provides an efficient estimate of the true set Γ^* . Indeed, on stage 8 in Algorithm 8 we recover the associated curve $\mathbf{v}_{(n+1)} \in \Xi$ to $\mathbf{u}_{(n+1)}$, and this leads us to recover the information lost after the dimension reduction, thereby further robustifying our method. For the second analytical example and where the uncertainties are driven by a Max-stable process, we obtain a more accurate estimation for higher truncation argument ($m_{KL} = 8$). This is due to the fact that the simulator response is more sensitive to higher KL order.

The main conclusion from figures 4.4 4.5 is that in all test cases, the convergence rates of the proposed approach are better than those of the functional quantization based inversion (\mathbb{L}^2 -GFQ). Our algorithm is $\mathcal{O}(l)$ times less expensive in terms of simulations used per iteration. The precision of the functional quantization method is limited by l , the size of the

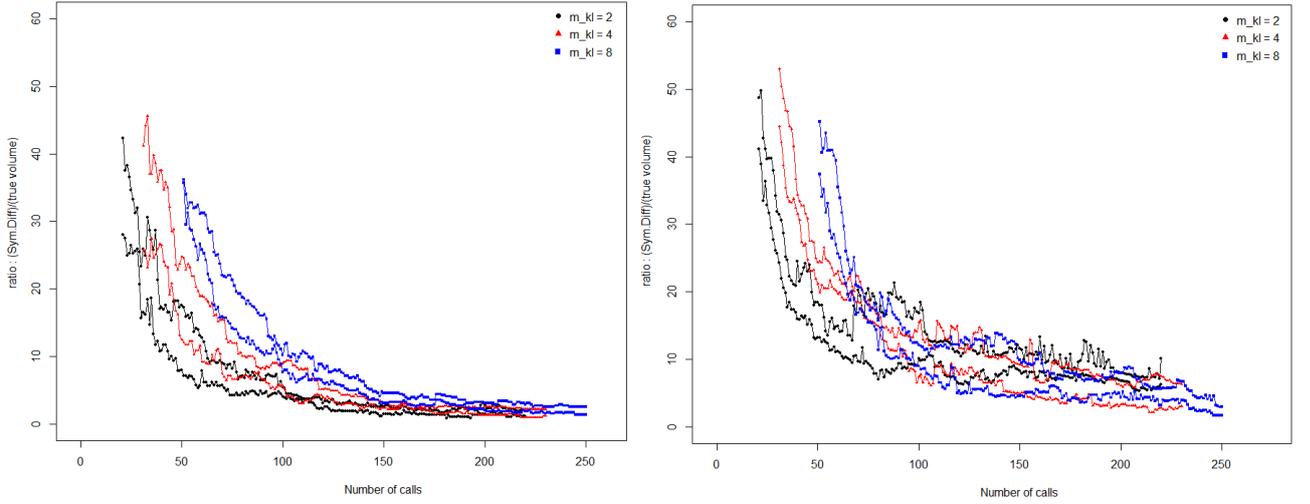


FIGURE 4.3: Analytical example 2 [Brownian motion (left), Max-stable process (right)]. Convergence of Algorithm 8 for $m_{KL} = \{2, 4, 8\}$. 25th and 75th percentiles of the symmetric difference vs. number of simulators calls. The curves are based on 10 random DoE for every runs.

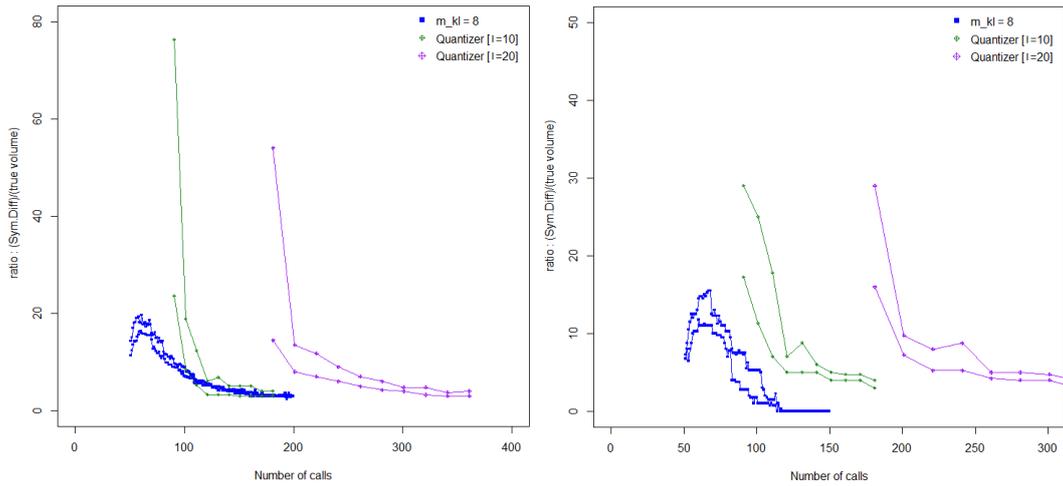


FIGURE 4.4: Analytical example 1 [Brownian motion (left), Max-stable (right)]. Convergence of Algorithm 8 for $m_{KL} = 8$ and Algorithm 9 for $l = \{10, 20\}$. 25th and 75th percentiles of the symmetric difference vs. number of simulators calls. The curves are based on 10 random DoE for every runs.

quantizer used to estimate the expectation (see Chapter 3). Indeed l controls the accuracy of the estimated expectation $\mathbb{E}_{\mathbf{V}}[f(\mathbf{x}, \mathbf{V})]$.

Because that is the closest example to our industrial case (see Chapter 5), we focus on the second analytical example with uncertainties defined by the max stable process. From Table 4.2, we note that the strategy defined in the joint space outperforms the one presented in the chapter 3. However, this strategy requires a relatively important amount of computational time for the choice of the next evaluation point. This is further explained by the need of integrals calculations by Monte Carlo methods (see section 4.3). As shown in Figure 4.6, the bigger the dimension of the problem is, the larger the computational time is. The

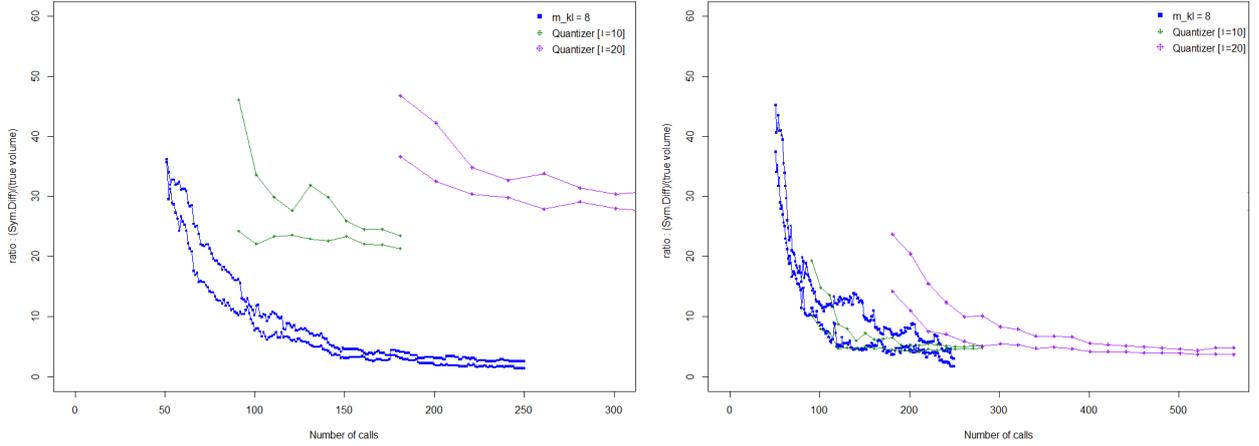


FIGURE 4.5: Analytical example 2 [Brownian motion (left), Max-stable (right)]. Convergence of Algorithm 8 for $m_{KL} = 8$ and Algorithm 9 for $l = \{10, 20\}$. 25th and 75th percentiles of the symmetric difference vs. number of simulator calls. The curves are based on 10 random DoE for every runs.

computational cost increases with the number of simulated points, and thus with the number of iterations, because of the cost of kriging approximation directly linked with the number of data points. For example at iteration 200, the method proposed in this chapter needs 140 seconds to perform the optimization and provide the next evaluation point.

Method : joint space	Est. Err. 5%	Est. Err. 2%	\mathbb{L}^2 -GFQ	Est. Err. 5%	Est. Err. 2%
$m_{KL} = 2$	not reached	not reached	$m_{KL} = 2$	≈ 150	> 500
$m_{KL} = 4$	≈ 190	not reached	$m_{KL} = 4$	≈ 150	> 500
$m_{KL} = 8$	≈ 150	≈ 250	$m_{KL} = 8$	≈ 100	> 500

Method : joint space	iteration 100	iteration 200
$m_{KL} = 2$	32 (sec.)	62 (sec.)
$m_{KL} = 4$	44 (sec.)	94 (sec.)
$m_{KL} = 8$	71 (sec.)	140 (sec.)

\mathbb{L}^2 -GFQ	iteration 100	iteration 200
$m_{KL} = 2, 4, 8$	15 (sec.)	35 (sec.)

TABLE 4.2: Analytical example 2 [Max stable process]. Number of simulator calls in function of the estimation error for the two methods presented respectively in Chapter 3 (\mathbb{L}^2 -GFQ) and in Chapter 4 (Kriging in joint space). Bottom : CPU time at iteration 100 and 200 for the methods.

4.6 Conclusion

In this chapter, a new stochastic inversion method has been proposed to address problems where some of the inputs are functional uncertain variables. After a dimension reduction of the functional space, the method creates a kriging model of the function of interest in the joint space of control and uncertain variables (\mathbf{x}, \mathbf{u}) . Then an averaged Gaussian process over the truncated functional random variable is derived, whereas the methods presented in Chapter 3

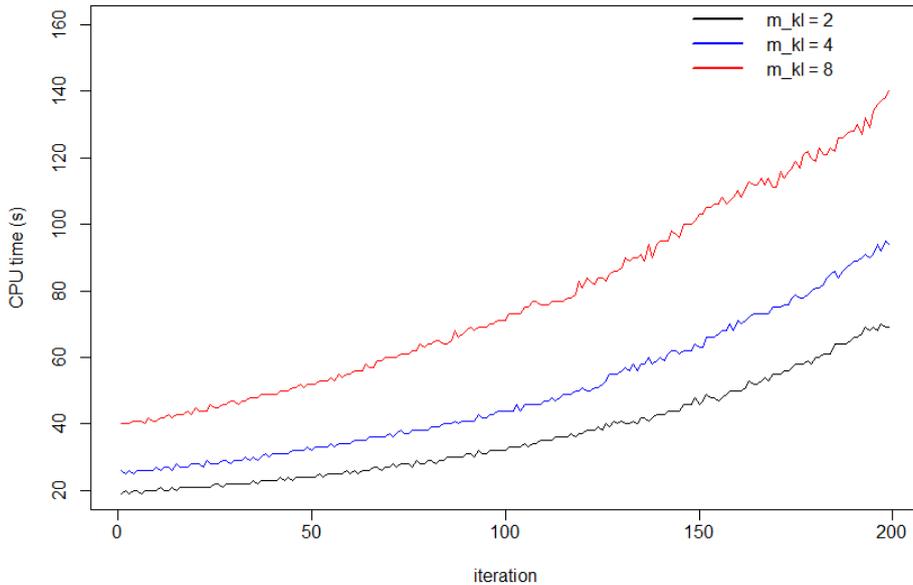


FIGURE 4.6: The computation time (sec.) needed to provide the next evaluation point as a function of iterations.

rely on GP models of the expectation of the function of interest in control variables space only. The induced projected Gaussian process approximates the expected response. The minimization of the Vorob'ev deviation and the induced variance allow to simultaneously efficiently estimate the set of interest Γ^* , and sample the uncertain variables \mathbf{v} .

The method has been compared to the SUR algorithm where the Monte Carlo simulations were performed on the true simulator. The results indicate significant enhancements in terms of precision and number of calls to the simulator on analytical test examples. Contrarily to the functional quantization method, the proposed approach has additional advantages : the points are not randomly drawn in the uncertain space but chosen by minimizing the kriging variance in high performance region ; the next control and uncertain points are chosen according to the previous added points.

Through analytical examples, we have highlighted the robustness of this approach and also its sensitivity to Karhunen-Loève decomposition. Indeed, it has been noted that for an efficient estimation of the excursion set, more components should be kept in the decomposition. The obtained results emphasize that the computational time needed to provide the next evaluation point in the SUR loop increases drastically with the size of the design of experiments and with the problem dimension (which includes the reduced dimension of the functional variable). This represents the major drawback of this method.

Synthèse : On a proposé dans ce chapitre une nouvelle méthode dédiée à l'inversion sous incertitudes fonctionnelles. Cette nouvelle approche consiste en un enrichissement adaptatif du plan d'expériences dans l'espace joint (variables de contrôle et variables incertaines). A travers des exemples analytiques, on a montré la robustesse de cette approche mais aussi sa sensibilité à la décomposition de Karhunen-Loève.

Comparée à la méthodologie proposée dans le chapitre 3, cette approche est beaucoup moins coûteuse en termes d'évaluation de la fonction. Concernant le temps de calcul à chaque itération, on a remarqué que ce dernier augmente en fonction de la taille du plan d'apprentissage dans la modélisation et aussi de la dimension du problème. En effet, ce temps de calcul conséquent est expliqué par : la modélisation par processus gaussien dans l'espace joint (dont la taille dépend directement de la taille choisie pour la décomposition Karhunen-Loève), les calculs d'intégrales associées au critère SUR ainsi que les étapes d'optimisation dans l'espace joint pour le choix du prochain point d'évaluation.

Dans le chapitre suivant, on présentera le cas industriel qui a motivé ces travaux de recherche. Un premier travail sur la visualisation des incertitudes fonctionnelles sera présenté. Ensuite on appliquera les méthodes proposées dans les chapitres 3 et 4 à ce problème industriel.

Chapitre 5

Application au système de dépollution SCR

Outlines

Ce chapitre présente une application des méthodes proposées pendant la thèse à un cas industriel. Nous décrivons en premier lieu le système de post-traitement des gaz d'échappement étudié et nous explicitons les problématiques statistiques qui découlent des objectifs opérationnels et la formulation choisie. Nous présentons ensuite un état de l'art des méthodes existantes pour la visualisation d'une variable aléatoire fonctionnelle. Enfin, Nous présentons les résultats d'inversion sous incertitudes.

Contents

5.1	La réduction catalytique sélective des oxydes d'azote	83
5.2	Le simulateur numérique : une boîte noire	85
5.2.1	Les entrées du simulateur numérique	85
5.2.2	Les sorties du simulateur numérique	86
5.2.3	Formalisation du problème	87
5.3	Visualisation de données fonctionnelles	88
5.3.1	HDR boxplot	88
5.3.2	Le boxplot fonctionnel	89
5.3.2.1	Notion de profondeur	89
5.3.2.2	Construction du boxplot fonctionnel	90
5.3.3	Application aux cycles de conduite	91
5.4	Réglage des stratégies de contrôle pour respecter les normes de dépollution en présence d'incertitudes	93
5.4.1	Application à un sous problème	94
5.4.1.1	Inversion par quantification fonctionnelle	94
5.4.1.2	Inversion par métamodélisation dans l'espace joint	97
5.4.1.3	Analyse comparative des deux méthodes	99
5.4.2	Application : 9 variables de contrôle et 1 variable fonctionnelle incertaine	100

5.1 La réduction catalytique sélective des oxydes d'azote

Dans le cadre de cette thèse, on s'intéresse à la dépollution des gaz d'échappement en sortie des moteurs, et plus particulièrement au paramétrage d'une stratégie de contrôle d'un

ystème de dépollution qui satisfait les normes européennes. En Europe, de nouvelles normes se succèdent pour revoir à la baisse les seuils d'émissions des principaux polluants réglementés. Les normes régulent les émissions de 4 polluants : NO_x , CO, HC et les particules. Par exemple, pour les véhicules Diesel, la norme Euro 6b impose un seuil égal à 80 mg/km pour les NO_x .

Dans le cadre de la thèse, on va s'intéresser aux oxydes d'azote NO_x . Une des techniques largement utilisées est la réduction catalytique sélective par réaction avec du NH_3 . En effet, le SCR (Selective Catalytic Reduction) permet de réduire les NO_x à l'intérieur de la ligne d'échappement à l'aide d'une solution à base d'urée injectée en amont du catalyseur (cf. Figure 5.1). L'urée est convertie en NH_3 par hydrolyse, et ce dernier réagit avec les NO_x pour former N_2 et H_2O , suivant les réactions chimiques suivantes :

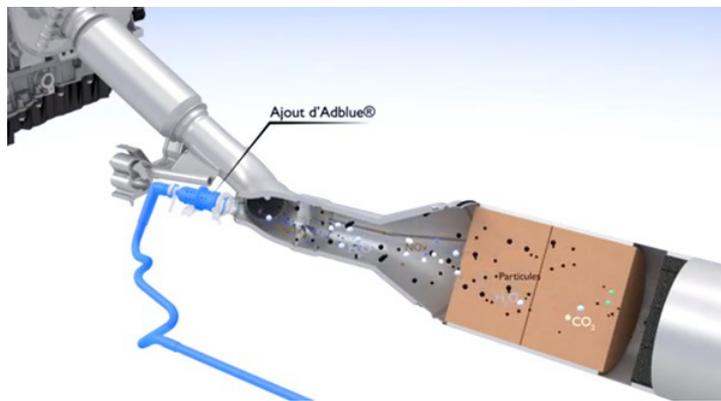
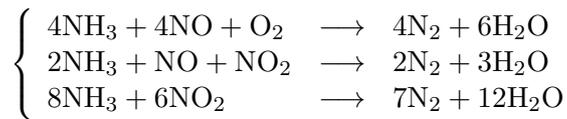


FIGURE 5.1: Injection d'urée (Adblue) en amont du SCR

Parmi les avantages à utiliser NH_3 comme agent réducteur, on notera particulièrement la sélectivité élevée de la réaction de ce dernier avec NO et NO_2 en présence d'oxygène. L'utilisation de NH_3 engendre cependant la question de son rejet. Ainsi la bonne calibration du système, i.e. le bon dosage de l'injection d'urée est critique : un sous dosage de l'urée mène au non respect des normes, alors qu'un surdosage sera suivi de relargage de NO_2 à l'échappement. La quantité d'urée à injecter est calculée en fonction de la proportion des NO_x présents en amont du catalyseur, ainsi que de la température, du débit d'air, d'oxygène ainsi que des états internes au système de contrôle. C'est ce calcul (stratégie de contrôle) qui est en charge d'éviter un sous-dosage ou un surdosage de la solution d'urée.

Le système de dépollution SCR évolue dans un environnement incertain. En effet, le système de contrôle dépend de différentes grandeurs qui peuvent être entachées d'erreur puisqu'elles sont fournies par des capteurs installés dans le véhicule. Le système de contrôle dépend de la nature du cycle de conduite (profil de vitesse du véhicule en fonction du temps), ainsi le comportement du contrôle doit nécessairement s'adapter d'un cycle à l'autre.

Résumé et schéma de fonctionnement : Le système de post-traitement étudié est, entre autres, constitué d'un modèle théorique défini par des équations décrivant les réactions physico-chimiques à l'intérieur du pot catalytique. En plus, on a un système de contrôle qui vient corriger le modèle théorique, soit pour injecter plus de solution d'urée (donc de NH_3)

pour ainsi réduire plus de NO_x , soit pour stopper l'injection et du coup limiter le dégagement d'ammoniac.

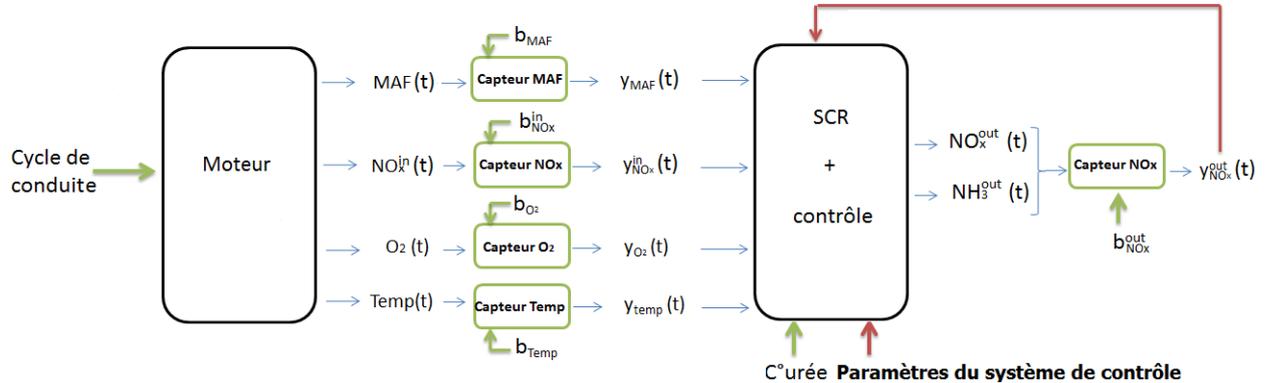


FIGURE 5.2: Fonctionnement SCR et système de contrôle

Comme le montre le schéma 5.2, la boîte **SCR + "contrôle"** (système de contrôle) prend en entrée différentes grandeurs, comme la quantité des NO_x , la température, l'oxygène et le débit d'air massique. Toutes ces grandeurs sont mesurées par des capteurs, et donc sont entachées d'erreurs liées aux capteurs, notés par la suite de manière générique "biais des capteurs" : $b_{\text{NO}_x^{\text{in}}}$, b_{MAF} , b_{O_2} et b_{Temp} . En sortie, on retrouve les NO_x non réduits, ainsi que l'ammoniac non réagi avec les NO_x . Ces sorties mesurées, à leur tour, vont être réintégrées dans le code, plus précisément dans la partie **contrôle**, afin de corriger l'injection de l'urée.

5.2 Le simulateur numérique : une boîte noire

Le simulateur numérique mis à disposition permettant de modéliser le système de post-traitement ne donne accès qu'à des couples entrées/sorties. On parle ainsi de modèle **boîte noire**. Dans la suite nous considérerons les variables d'entrées et les valeurs d'intérêts en sortie décrites dans le schéma 5.3. L'objectif de cette section est de préciser la nature de ces variables.

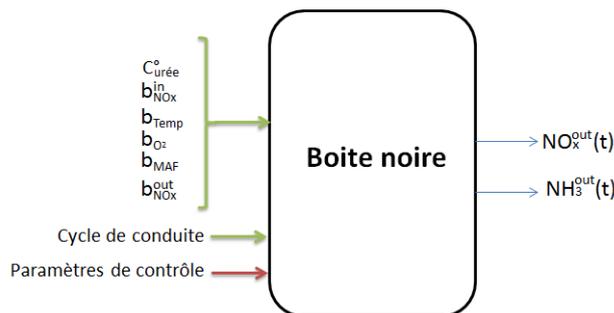


FIGURE 5.3: Le simulateur numérique du cas d'application.

Remarque. On note que toutes les entrées du simulateur sont supposées indépendantes.

5.2.1 Les entrées du simulateur numérique

Le simulateur numérique, utilisé pour modéliser les phénomènes physico-chimiques mis en jeu dans le cadre du post-traitement des gaz d'échappements, prend deux types d'entrée : des

paramètres de contrôle et des paramètres incertains. Le tableau 5.1 résume les 9 paramètres scalaires de contrôle ainsi que leurs domaines de définition. Certains de ces paramètres agissent sur la quantité de stockage de l'ammoniac dans la catalyseur et d'autres caractérisent un contrôle fonctionnel.

Type	Variable	Plage de variation
Scalaires	theta.sp.manual	[0.01, 0.6]
	theta.sp.min	[0.01, 0.6]
	C.NH3.in.max	[0.005, 0.02]
	carto.obs4.f1.x2	[0.05, 0.95]
	carto.obs4.f1.y2	[0, 1]
	carto.obs4.f1.y3	[0, 1]
	carto.obs4.f2.x2	[220, 300]
	carto.obs4.f2.y2	[0, 1]
	carto.obs4.f2.y3	[0, 1]

TABLE 5.1: Les 9 paramètres de contrôle.

L'objectif est de régler la stratégie de contrôle du système sachant que ce dernier évolue dans un environnement incertain induit par le cycle de vitesse ainsi que par les erreurs des capteurs. Dans ce contexte, les paramètres incertains, listés dans le tableau 5.2, sont représentés par des variables aléatoires et leur incertitude est modélisée par une distribution de probabilité. Pour réaliser cette modélisation, il est possible de s'appuyer sur des données disponibles et, dans certains cas, sur un avis d'expert. C'est le cas de nos entrées scalaires dont les distributions ont été établies en collaboration avec un expert moteur (cf. Table 5.2).

Type	Variable	Description	Loi
Fonctionnelle	V	Cycles de conduite	–
Scalaires	$C_{urée}$	Concentration d'urée	Uniforme
	$b_{NO_x^{in}}$	Biais du capteur amont NO_x	Normale
	b_{O_2}	Biais du capteur O_2	Normale
	b_{Temp}	Biais du capteur température	Normale
	b_{MAF}	Biais du capteur MAF	Normale
	$b_{NO_x^{out}}$	Biais du capteur aval NO_x	Normale

TABLE 5.2: Les paramètres incertains en entrée du code et leurs distributions de probabilité associées

La variable aléatoire fonctionnelle \mathbf{V} , modélisant les cycles de conduite, est connue à travers un échantillon de 100 courbes temporelles d'une durée de 90 min (voir figure 5.4). Ces cycles de vitesse sont obtenus à partir d'un générateur de cycles de type markovien dont les probabilités de transition sont apprises à partir de 13 cycles enregistrés (voir annexe A pour plus de détails sur la définition de ces cycles).

Remarque. Dans la génération des cycles de conduite, différentes contraintes doivent être prises en compte. Ces contraintes définissant un processus de rejet mènent à un temps de génération très important. Ainsi il a été choisi de limiter la taille de l'échantillon à 100 cycles de 90 minutes.

5.2.2 Les sorties du simulateur numérique

Le simulateur numérique renvoie deux sorties fonctionnelles : $NH_3^{out}(t)$ et $NO_x^{out}(t)$

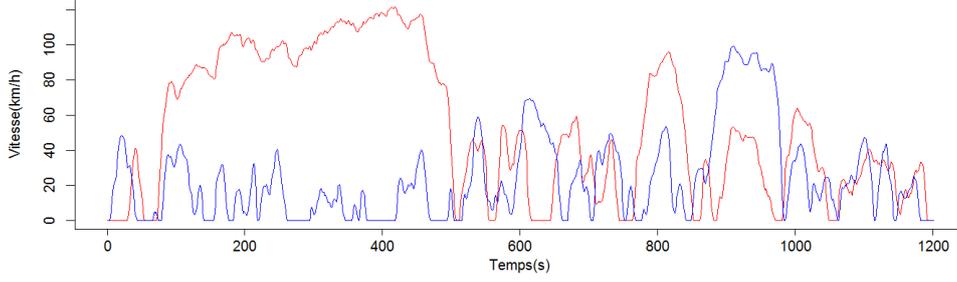


FIGURE 5.4: Une illustration de deux cycles de conduite visualisés sur une durée de 1200s (les cycles étudiés durent 5400s).

$$\text{Sorties fonctionnelles} \begin{cases} \text{NO}_x^{\text{out}}(t) & : \text{ les émissions des oxydes d'azote (ppm)} \\ \text{NH}_3^{\text{out}}(t) & : \text{ les émissions d'ammoniac (ppm)} \end{cases}$$

Cependant les normes sont instaurées sur les quantités scalaires suivantes :

$$\begin{cases} y_1 & = \frac{1}{\text{distance}} \int_0^T 10^3 \cdot \text{Flux}_{\text{NO}_x}(t) dt, \\ y_2 & = \max_{t \in [0, T]} \text{NH}_3^{\text{out}}(t), \\ \text{où } \text{Flux}_{\text{NO}_x} (g \cdot s^{-1}) & = \frac{10^3 \cdot 10^{-6} \cdot \text{Flux}_{\text{gas}} (kg \cdot h^{-1})}{3600 \cdot \text{Mmol}_{\text{gas}} \cdot \text{Mmol}_{\text{NO}_x} \cdot \text{NO}_x^{\text{out}} (\text{ppm})}. \end{cases}$$

y_1 représente le score de NO_x ($mg \cdot km^{-1}$) obtenu pour un cycle de conduite sur un intervalle de temps $[0, T]$ et cette grandeur doit être inférieure à **160** $mg \cdot km^{-1}$. La sortie y_2 représente le pic d'ammoniac associé au cycle et ne doit pas dépasser **30** ppm. La contrainte de 160 $mg \cdot km^{-1}$ correspond à la norme euro 6 de 80 mg/km pondérée par un facteur de conformité pour les roulages réels autour de 2, facteur en discussion lors de l'établissement de ces travaux de thèse. La limite de **30** ppm n'est pas donnée par la norme mais paraît une vraisemblable au vu des émissions autorisées pour les poids lourds. Ces limites sont régulièrement révisées avec l'évolution des normes.

5.2.3 Formalisation du problème

On définit f notre simulateur numérique par :

$$f : \begin{cases} \mathbb{X} \times \mathcal{V} \times \mathbb{R}^6 & \rightarrow \mathbb{R}^2 \\ (\mathbf{x}, \mathbf{V}, \boldsymbol{\xi}) & \mapsto (f_1(\mathbf{x}, \mathbf{V}, \boldsymbol{\xi}), f_2(\mathbf{x}, \mathbf{V}, \boldsymbol{\xi})) = (y_1, y_2), \end{cases} \quad (5.1)$$

où $\mathbb{X} \subset \mathbb{R}^9$ est le domaine de définition du paramètre de contrôle \mathbf{x} (Tableau 5.1), $\boldsymbol{\xi}$ est le vecteur aléatoire regroupant les 6 entrées scalaires incertaines et \mathcal{V} est l'espace fonctionnel où la variable aléatoire \mathbf{V} prend ses valeurs (Tableau 5.2).

Dans un tel contexte, nous nous sommes intéressés aux objectifs suivants :

- **visualiser l'incertitude de la variable aléatoire fonctionnelle**
- **résoudre un problème d'inversion** afin de construire l'ensemble des paramètres de contrôle garantissant le respect des normes (respect des seuils sur y_1 et y_2).

Visualiser les incertitudes de variables fonctionnelles peut être un moyen efficace pour explorer ces données et mettre en valeur certaines de leurs caractéristiques, en résumant graphiquement des informations complexes. Plusieurs techniques de visualisation de données fonctionnelles ont été récemment développées avec plusieurs objectifs : la détection de tendances et de courbes extrêmes dans les échantillons de fonctions, ou encore la construction de

domaines de confiance contenant la plupart des fonctions de l'échantillon. Ces travaux sont présentés dans la section 5.3 .

La résolution d'un problème d'inversion est motivée par la connaissance des seuils à respecter. On rappelle que pour répondre aux normes, il faut que y_1 , le score de DeNOx, soit inférieur à 160 mg.km^{-1} , et que y_2 , le pic d'ammoniac, ne dépasse pas 30 ppm . A partir de ces seuils, on définit le domaine admissible $D = [0, 160] \times [0, 30]$. Tout en tenant compte des différents types d'incertitude, le but est de trouver l'ensemble des paramètres de contrôle tels que, en moyenne, les sorties du code appartiennent à l'ensemble D . Ainsi cela se traduit par la résolution des problèmes d'inversion suivants

$$\begin{cases} \Gamma_1^* &= \{\mathbf{x} \in \mathbb{X}, g_1(\mathbf{x}) = \mathbb{E}_{(\mathbf{V}, \xi)}[f_1(\mathbf{x}, \mathbf{V}, \xi)] \leq 160\} \\ \Gamma_2^* &= \{\mathbf{x} \in \mathbb{X}, g_2(\mathbf{x}) = \mathbb{E}_{(\mathbf{V}, \xi)}[f_2(\mathbf{x}, \mathbf{V}, \xi)] \leq 30\}, \end{cases} \quad (5.2)$$

où l'espérance est prise sur les variables aléatoires scalaires et fonctionnelle. Les deux méthodes que nous avons introduites dans les chapitres 3 et 4 sont utilisées pour apporter une solution à ce problème. Les résultats sont présentés dans la section 5.4 .

5.3 Visualisation de données fonctionnelles

Cette section est consacrée à la visualisation d'un échantillon de courbes, i.e. d'un ensemble de réalisations $\{v_i\}_{i=1\dots N}$, $N \in \mathbb{N}$ d'une variable aléatoire fonctionnelle \mathbf{V} . Ces méthodes ont pour objectif d'aider à l'exploration des données et de mettre en valeur leurs caractéristiques statistiques. Les sections 5.3.1 et 5.3.2 présentent deux méthodes existantes de visualisation qui sont des généralisations du boxplot ([Tukey, 1977]) : le HDR boxplot et le boxplot fonctionnel. Le résultat de l'application de ces méthodes est présenté dans la section 5.3.3.

5.3.1 HDR boxplot

[Hyndman and Shang, 2010] proposent une méthode de visualisation de données fonctionnelles : le HDR (Highest Density Region) boxplot fonctionnel. Celle-ci s'appuie sur l'estimation de la densité des deux premières composantes issues de la décomposition en Analyse en Composantes Principales (ACP) des fonctions observées. Pour cela, ils utilisent un estimateur à noyau :

$$\hat{f}(u) = \frac{1}{N} \sum_{i=1}^N \mathcal{K}_{h_i}(u - U_i) \quad (5.3)$$

où $u \in \mathbb{R}^2$ et $U = \{U_1, \dots, U_N\} \subset \mathbb{R}^2$ est l'ensemble des couples de points issus de l'ACP. Une région de plus haute densité est définie comme $\mathcal{R}_\alpha = \{u \in \mathbb{R}^2 : \hat{f}(u) \geq f_\alpha\}$, où f_α est telle que $\int_{\mathcal{R}_\alpha} \hat{f}(u) du = 1 - \alpha$. Le HDR boxplot représente les 3 éléments suivants :

- Les régions de plus haute densité avec $\alpha = 50\%$ en gris foncé et $\alpha = 95\%$ en gris clair.
- Les valeurs extrêmes, définies comme étant les points qui n'appartiennent pas aux deux régions définies précédemment.
- Le mode : le point de l'échantillon de plus haute densité.

[Hyndman and Shang, 2010] définissent à la fois des versions bivariée et fonctionnelle du HDR boxplot, la version fonctionnelle est constituée de la courbe médiane (correspondant au mode), des courbes extrêmes (correspondant aux valeurs extrêmes) et les enveloppes fonctionnelles correspondant aux régions de plus haute densité. La figure 5.5 illustre le résultat obtenu en appliquant le HDR boxplot au jeu de données "Elnino" issu du package `rainbow`,

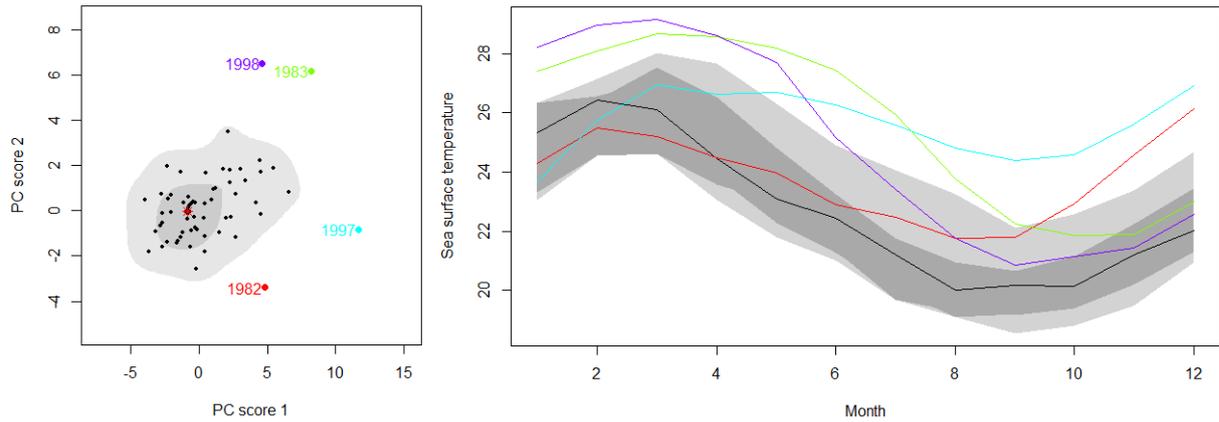


FIGURE 5.5: Bagplot bivarié pour les températures de surface de l’océan pacifique sur une année (à gauche) et sa représentation fonctionnelle (à droite), extrait de [Hyndman and Shang, 2010]

il s’agit d’un échantillon constitué de 62 courbes de températures de surface de l’océan pacifique relevées mensuellement entre 1951 et 2011, et chaque courbe représente une année et est discrétisée en 12 points. L’étoile rouge est le mode, la zone en gris foncé est la région de plus haute densité à 50%, la zone en gris clair est celle à 95%. Sur le graphique de droite, la courbe médiane est représentée en noir. Les courbes et points colorés sont les données extrêmes.

5.3.2 Le boxplot fonctionnel

Une autre représentation par boxplot est proposée par [Sun and Genton, 2011]. Celle-ci utilise la profondeur de bande (band depth) introduite par [López-Pintado and Romo, 2009] pour ordonner les fonctions. La profondeur de bande mesure la centralité des courbes : plus la profondeur de bande est grande, plus la courbe est centrale.

5.3.2.1 Notion de profondeur

Dans cette partie, on va s’intéresser à la notion de profondeur pour des données fonctionnelles, on va commencer par introduire les différentes définitions de la profondeur et on verra que cet outil de visualisation aide à mettre en lumière ou déceler des courbes ”outlier” éventuelles.

Dans la suite, la profondeur d’une courbe sera une valeur réelle positive et bornée. Soit un échantillon de N courbes $\Xi = \{v_1, v_2, \dots, v_n\}$ définies sur l’intervalle $[0, T]$, et soit v une courbe appartenant, ou non, à l’échantillon. On définit comme suit,

Le graphe d’une fonction :

$$G(v) = \{(t, v(t)) : t \in [0, T]\}$$

La bande délimitée par j courbes $v_{i_1}, v_{i_2}, \dots, v_{i_j}$

$$B(v_{i_1}, v_{i_2}, \dots, v_{i_j}) = \{(y, t) : t \in [0, T], \min_{r=1, \dots, j} v_{i_r}(t) \leq y \leq \max_{r=1, \dots, j} v_{i_r}(t)\}$$

Comme la bande est délimitée par j courbes et disposant de N courbes dans notre échantillon, on aura donc $\binom{N}{j}$ bandes possibles.

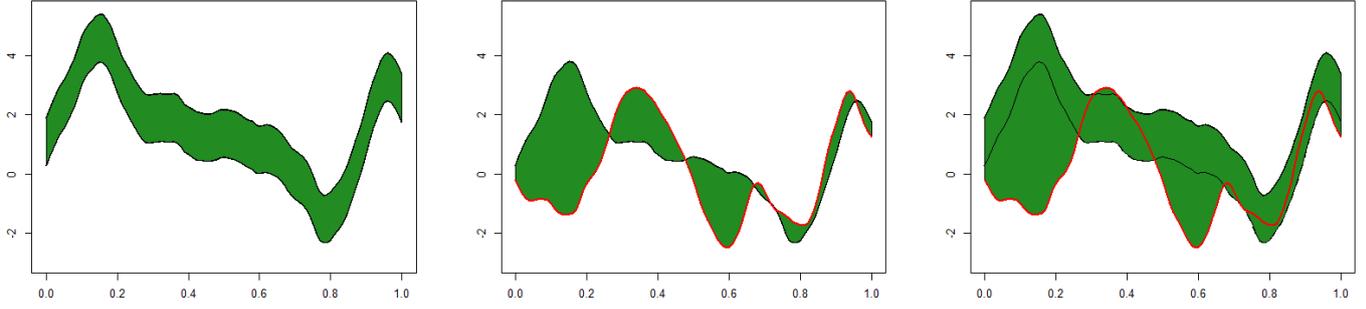


FIGURE 5.6: (De gauche à droite) Exemples de bandes (en vert) délimitées par (de gauche à droite) : 2 courbes ; 2 courbes qui se croisent ; 3 courbes.

La profondeur de bande :

La profondeur de bande, [López-Pintado and Jornsten, 2007], d'une courbe v est définie comme la proportion de toutes les bandes possibles contenant v .

$$S_{N,J}(v|\Xi) = \sum_{j \geq 2}^J S_N^{(j)}(v|\Xi), J \geq 2, \quad (5.4)$$

avec

$$S_N^{(j)}(v|\Xi) = \binom{N}{j}^{-1} \sum_{1 \leq i_1 < i_2 < \dots < i_j \leq N} I\{G(v) \in B(v_{i_1}, v_{i_2}, \dots, v_{i_j})\}, j \geq 2. \quad (5.5)$$

Si une courbe v a une forme, ou comportement, qui diffère de l'échantillon Ξ , alors peu de bandes la contiendront. Ainsi, une courbe peu ressemblante à l'échantillon se verra attribuer une faible profondeur de bande et vice-versa. Cette définition de profondeur de bande, donne des résultats peu convaincants surtout si les courbes se croisent, ainsi "la profondeur de bande modifiée" vient pallier ce problème.

La profondeur de bande modifiée :

La profondeur de bande modifiée, introduite par [López-Pintado and Jornsten, 2007], consiste à remplacer la fonction indicatrice dans l'équation 5.5 par la proportion de longueur que la courbe reste dans la bande.

$$GS_{N,J}(v|\Xi) = \sum_{j \geq 2}^J GS_N^{(j)}(v|\Xi), J \geq 2, \quad (5.6)$$

avec

$$GS_N^{(j)}(v|\Xi) = \binom{\Xi}{j}^{-1} \sum_{1 \leq i_1 < i_2 < \dots < i_j \leq N} \frac{\lambda(A(v_{i_1}, v_{i_2}, \dots, v_{i_j}))}{\lambda([0, T])}, j \geq 2 \quad (5.7)$$

$$A(v_{i_1}, v_{i_2}, \dots, v_{i_j}) = \left\{ t \in [0, T] : \min_{r=1, \dots, j} v_{i_r}(t) \leq v(t) \leq \max_{r=1, \dots, j} v_{i_r}(t) \right\}$$

5.3.2.2 Construction du boxplot fonctionnel

Pour la construction du boxplot fonctionnel, on ne retient que la profondeur de bande modifiée. Comme pour le boxplot ordinaire, on cherche à construire une boîte qui contiendra environ la moitié des courbes.

Soit un échantillon de N observations $\Xi = \{v_1, v_2, \dots, v_N\}$, on les ordonne selon leur profondeur de bande modifiée : $v_{[1]}, \dots, v_{[N]}$, où $v_{[1]}$ est la courbe la plus centrale (avec la plus grande profondeur de bande) et $v_{[N]}$ la courbe la plus extrême. On définit alors la région à 50%, contenant les $\lceil \frac{N}{2} \rceil$ courbes les plus centrales :

$$C_{0.5} = \left\{ (t, v(t)); \min_{r=1, \dots, \lceil \frac{N}{2} \rceil} v_{[r]}(t) \leq v(t) \leq \max_{r=1, \dots, \lceil \frac{N}{2} \rceil} v_{[r]}(t) \right\},$$

où $\lceil \frac{N}{2} \rceil$ est le plus petit entier supérieur à $\frac{n}{2}$, cette région à 50% représente le sac (qui contient la moitié des points), l'enveloppe (qui sépare les courbes extrêmes des autres) est déterminée en augmentant le sac d'un facteur **1.5**. Le facteur de proportionnalité, appelé facteur de dilatation, est pris égal à 1.5 par défaut, par analogie avec la taille des "moustaches" dans le boxplot usuel. Ainsi toute courbe en dehors de l'enveloppe est considérée comme extrême.

On peut considérer deux types de courbes extrêmes :

- Les courbes qui sont éloignées de la courbe centrale appelées "Magnitude outliers"
- Les courbes qui ont une forme ou un comportement différent des autres appelées Shape Outlier

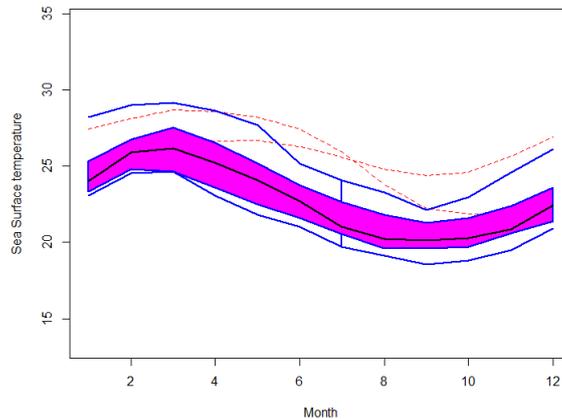


FIGURE 5.7: boxplot fonctionnel pour les températures de surface de l'océan pacifique sur une année

La méthode du boxplot fonctionnel est appliquée au jeu de données "ElNino" présenté dans le paragraphe précédent. La courbe noire est la médiane, la surface en violet représente le sac. La zone entre les courbes bleues représente l'enveloppe. Les courbes extrêmes sont représentées en pointillés rouges.

5.3.3 Application aux cycles de conduite

Les deux méthodes de visualisation présentées dans les sections 5.3.2 et 5.3.1 sont appliquées à l'échantillon de 1000 courbes de vitesse illustrées par la Figure 5.8. On note que pour cette étude on utilise des courbes de 1200 secondes. Les résultats sur l'inversion (Section 5.4) sont bien obtenus pour des durées de 90 min

Les résultats obtenus sont illustrés par les Figures 5.9, 5.10. On constate que ces outils de visualisations ont failli à leur tâche, qui est de montrer une certaine tendance ou de déceler des courbes extrêmes. Le HDR boxplot, Figure 5.9, est basé sur une projection des courbes sur les deux premières composantes ($m_{KL} = 2$), or cette projection, dans notre cas, n'explique que

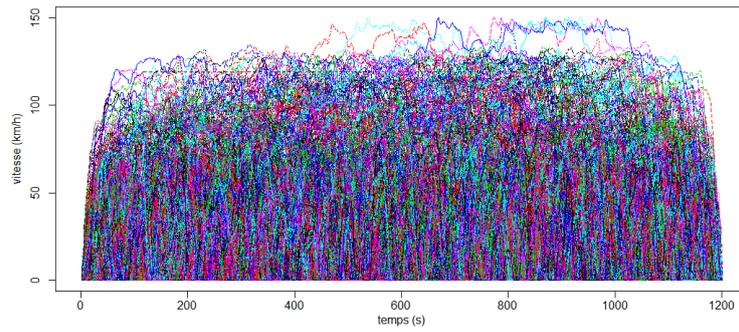


FIGURE 5.8: L'échantillon des 1000 cycles de conduite

24% de la variance, ainsi, il peut être intéressant d'utiliser une base de décomposition de plus de deux composantes. Cependant, l'estimateur à noyau, utilisé par [Hyndman and Shang, 2010] pour estimer la densité des coefficients, ne peut être appliqué qu'en petite dimension. Dans notre cas, et comme le montre la figure 5.11, il faut considérer les 80 premières composantes afin d'expliquer 97% de la variance.

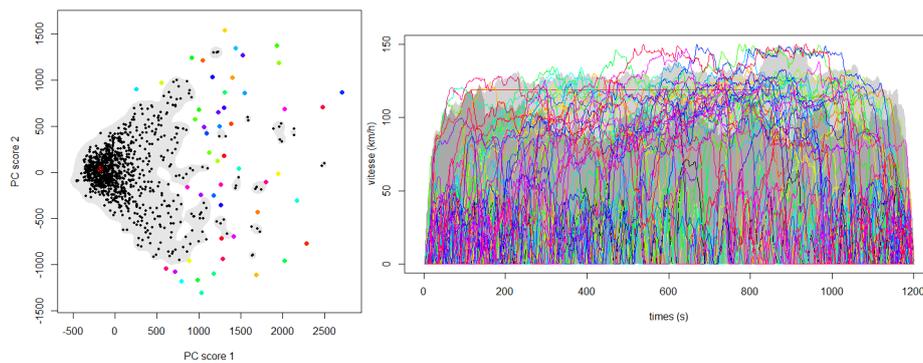


FIGURE 5.9: HDR boxplot de l'échantillon de cycles de conduite

Le boxplot fonctionnel, Figure 5.10, est basé sur des notions de profondeur. On remarque que tous les cycles ayant une partie autoroutière (Vitesses supérieures à 100 km/h) sont considérés comme des cycles extrêmes, et la courbe médiane (la courbe noire) représente un cycle de conduite urbaine. Ce résultat est expliqué par la proportion élevée de cycles urbains dans notre échantillon, ce qui implique une profondeur plus grande pour ces derniers et faible pour les cycles représentant d'autres types de conduite.

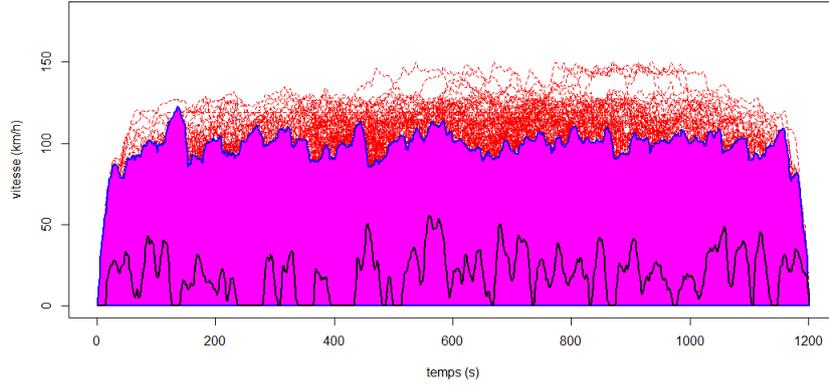


FIGURE 5.10: Boxplot fonctionnel de l'échantillon de cycles de conduite

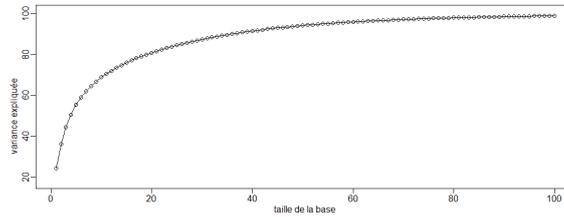


FIGURE 5.11: Part de variance expliquée par l'ACP en fonction de la taille de la base

5.4 Réglage des stratégies de contrôle pour respecter les normes de dépollution en présence d'incertitudes

Le problème d'inversion appliqué à la stratégie de contrôle d'un système de dépollution lors de la phase d'étude conceptuelle a été abordé dans la section 5.2 . On rappelle qu'il s'agit de définir le domaine de définition des variables de contrôle en vue de respecter, en moyenne, les deux normes sur les rejets de polluants instaurées par l'UE. Ces dernières sont évaluées à l'aide d'un code de calcul regroupant un ensemble de disciplines en interaction (la structure du véhicule, les réactions physico-chimiques dans le catalyseur, ...). Les variables de contrôle ainsi que les variables incertaines qui opèrent dans ce problème sont rappelées dans le tableau 5.3. En reprenant la définition du simulateur numérique introduite précédemment

$$f : \begin{cases} \mathbb{X} \times \mathcal{V} \times \mathbb{R}^6 & \rightarrow \mathbb{R}^2 \\ (\mathbf{x}, \mathbf{V}, \boldsymbol{\xi}) & \mapsto (f_1(\mathbf{x}, \mathbf{V}, \boldsymbol{\xi}), f_2(\mathbf{x}, \mathbf{V}, \boldsymbol{\xi})) = (y_1, y_2), \end{cases} \quad (5.8)$$

le problème d'inversion se traduit par l'estimation de l'intersection suivante $\Gamma_1^* \cap \Gamma_2^*$ où

$$\begin{cases} \Gamma_1^* & = \{\mathbf{x} \in \mathbb{X}, g_1(\mathbf{x}) = \mathbb{E}_{(\mathbf{V}, \boldsymbol{\xi})}[f_1(\mathbf{x}, \mathbf{V}, \boldsymbol{\xi})] \leq 160\} \\ \Gamma_2^* & = \{\mathbf{x} \in \mathbb{X}, g_2(\mathbf{x}) = \mathbb{E}_{(\mathbf{V}, \boldsymbol{\xi})}[f_2(\mathbf{x}, \mathbf{V}, \boldsymbol{\xi})] \leq 30\}. \end{cases}$$

Les deux contributions de cette thèse permettent d'estimer l'ensemble d'excursion quand le simulateur numérique renvoie une unique sortie. Ainsi on décide de procéder en deux temps. Tout d'abord, on résout séparément deux problèmes d'inversion pour estimer Γ_1^* et Γ_2^* . Ensuite, on considère l'intersection des deux estimateurs comme estimation de $\Gamma_1^* \cap \Gamma_2^*$.

Type	Notation	
Variables de contrôle (x)	theta.sp.manual	[0.01, 0.6]
	theta.sp.min	[0.01, 0.6]
	C.NH3.in.max	[0.005, 0.02]
	carto.obs4.f1.x2	[0.05, 0.95]
	carto.obs4.f1.y2	[0, 1]
	carto.obs4.f1.y3	[0, 1]
	carto.obs4.f2.x2	[220, 300]
	carto.obs4.f2.y2	[0, 1]
carto.obs4.f2.y3	[0, 1]	
Variable incertaine fonctionnelle	V	Uniforme discrète
Variable incertaine vectorielle (ξ)	C_{urée}	$\mathcal{U}([*, *])$
	b_{NO_xⁱⁿ}	$\mathcal{N}(0, *)$
	b_{O₂}	$\mathcal{N}(1, *)$
	b_{Temp}	$\mathcal{N}(0, *)$
	b_{MAF}	$\mathcal{N}(1, *)$
	b_{NO_x^{out}}	$\mathcal{N}(0, *)$

TABLE 5.3: les domaines de définition et les distributions de probabilité associés aux entrées du code de calcul. Certaines données (*) sont masquées car confidentielles.

5.4.1 Application à un sous problème

Dans cette partie, on considère que le contrôle du système de dépollution est caractérisé par les deux variables "theta.sp.manual" et "theta.sp.min". En effet, ces deux variables ont été identifiées par les gens du métier comme étant les principaux paramètres de réglage de la stratégie de contrôle. Nous avons donc choisi dans un premier temps de faire l'exercice avec ces deux variables. Par ailleurs, on suppose que les biais des capteurs sont fixés en leurs valeurs nominales, ou en d'autres termes, que les capteurs sont justes et fidèles. Ainsi l'incertitude affectant le simulateur numérique n'est régie que par la variable fonctionnelle **V**. Notre objectif est de nous focaliser sur cette incertitude fonctionnelle, qui intéresse particulièrement les gens du métier dans le cadre du passage aux normes en conditions réelles de conduite (RDE). Cette approche pourra (devra) par la suite être enrichie avec les incertitudes scalaires. Dans le cadre de l'évaluation des performances, la norme RDE de roulage en conditions réelles impose l'utilisation de cycles d'au moins 90 min. Le problème d'inversion se traduit par l'estimation de l'intersection $\Gamma_1^* \cap \Gamma_2^*$ où

$$\begin{cases} \Gamma_1^* &= \{\mathbf{x} \in [0.01, 0.6]^2, g_1(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f_1(\mathbf{x}, \mathbf{V})] \leq 160\} \\ \Gamma_2^* &= \{\mathbf{x} \in [0.01, 0.6]^2, g_2(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f_2(\mathbf{x}, \mathbf{V})] \leq 30\}. \end{cases}$$

5.4.1.1 Inversion par quantification fonctionnelle

On commence par appliquer la première contribution pour identifier l'ensemble suivant

$$\Gamma_2^* = \{\mathbf{x} \in [0.01, 0.6]^2, g_2(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f_2(\mathbf{x}, \mathbf{V})] \leq 30\}.$$

On considère un processus gaussien $Y_{\mathbf{x} \in \mathbb{X}} \sim \mathcal{GP}(m, k)$ de moyenne constante et de noyau de covariance de type Matérn 5/2. Les paramètres du noyau de covariance et la moyenne sont estimés par maximum de vraisemblance. On considère un plan d'expériences initial à $n = 8$ points, $\mathcal{X}_n = (\mathbf{x}_i)_{i=1}^n$, optimisé selon le critère maximin afin de garantir une bonne répartition des points dans l'espace $[10^{-2}, 0.6]^2$. En chaque point $\mathbf{x}_i \in \mathcal{X}_n$, on estime l'espérance

$g_2(\mathbf{x}_i) = \mathbb{E}_{\mathbf{V}}[f_2(\mathbf{x}_i, \mathbf{V})]$ par quantification fonctionnelle gloutonne (\mathbb{L}^2 -GFQ où $m_{KL} = 20$). Dans le but d'avoir une bonne estimation de l'espérance, on fixe les paramètres pour vérifier la stagnation de la convergence à $(4, 10^{-2})$. On rappelle que ces paramètres d'arrêt stoppent l'estimation séquentielle de l'espérance si au cours des 4 dernières itérations, l'erreur absolue entre deux estimations est inférieure à 10^{-2} .

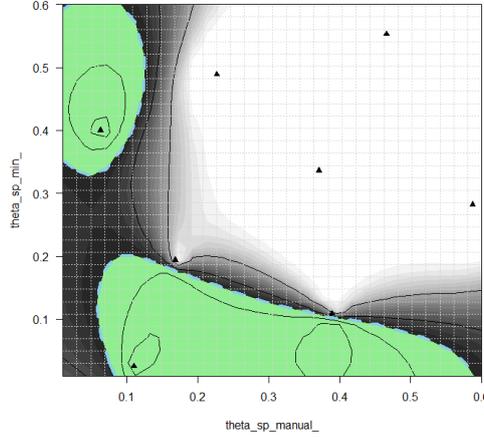


FIGURE 5.12: Pic d'ammoniac. La fonction de couverture, le plan d'expériences initial (triangles noirs) et l'estimation initiale de l'ensemble recherché Γ_2^* (en vert).

La figure 5.12 montre la fonction de couverture calculée à partir de l'information fournie par l'estimation de l'espérance aux points \mathcal{X}_n . Une estimation initiale de Γ_2^* est tracée en vert. Dans la figure 5.12, le niveau de gris représente la probabilité que l'observation dépasse le seuil, la zone blanche correspond à une zone où cette probabilité est très grande, et inversement.

On rappelle que la première contribution pour l'identification de l'ensemble d'excursion (cf. chapitre 3) combine deux méthodes : une stratégie de réduction des incertitudes induites par le métamodèle approchant la fonction g et une estimation de l'espérance par quantification fonctionnelle faite à partir d'une réduction de dimension. La première méthode donne un point dans l'espace de contrôle $[10^{-2}, 0.6]^2$ tel que si on ajoute ce point à l'ensemble d'apprentissage du méta-modèle, on réduit l'incertitude associée à l'estimation de l'ensemble recherché. La deuxième méthode mène à une estimation efficace séquentielle de l'espérance en ce point. Une itération de cette méthodologie est illustrée par la figure 5.13. On remarque que le point rouge proposé par la stratégie SUR se situe dans une zone où l'incertitude est grande. Après l'estimation de l'espérance en ce point (égale à $2.26 \leq 30$) et la mise à jour du processus gaussien en prenant compte cette nouvelle observation, on obtient une nouvelle estimation de Γ_2^* .

La figure de gauche dans 5.14 montre la décroissance de la *déviatio*n de Vorob'ev (l'incertitude) au cours des itérations, en effet la stratégie SUR est par définition proposée pour parvenir à ce but. Ainsi après 62 itérations, l'incertitude associée à l'estimation de Γ_2^* est quasiment nulle. En termes de coût de calcul, chaque itération nécessite une estimation de l'espérance en faisant appel au simulateur en moyenne 23 fois (cf. Figure 3.16). Ainsi la résolution de ce problème d'inversion a coûté 1575 évaluations du simulateur numérique.

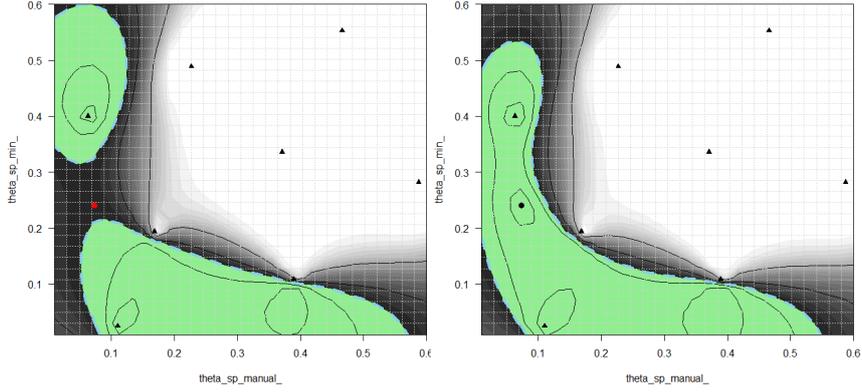


FIGURE 5.13: Pic d'ammoniac. A gauche : La fonction de couverture, le plan d'expériences initial (triangles noirs), l'estimation initiale de l'ensemble recherché Γ_2^* (en vert) et le point proposé par la stratégie SUR (en rouge). A droite : La mise à jour de l'estimation de Γ_2^* (en vert) après une itération.

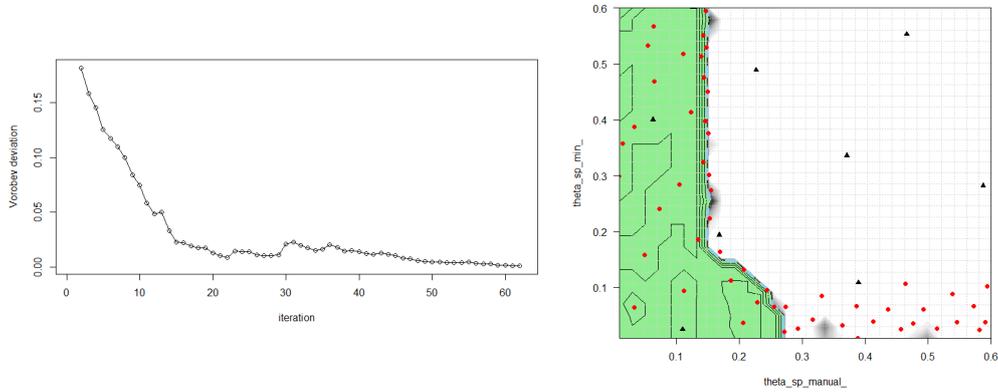


FIGURE 5.14: Pic d'ammoniac. En haut : La *dévi*ation de Vorob'ev en fonction des itérations SUR. En bas : la fonction de couverture, le plan d'expériences initial (triangles noirs) et l'estimation de l'ensemble recherché Γ_2^* (en vert) après l'ajout de **62** points (en rouge) et 1575 évaluations au total (évaluations pour l'estimation de l'espérance incluses).

On procède de la même manière pour identifier de l'ensemble Γ_1^* :

$$\Gamma_1^* = \{\mathbf{x} \in [0.01, 0.6]^2, g_1(\mathbf{x}) = \mathbb{E}_{\mathbf{V}}[f_2(\mathbf{x}, \mathbf{V})] \leq 160\}.$$

La figure 5.15 le résultat de l'estimation de l'ensemble Γ_1^* après l'ajout de 12 points (rouges) au plan d'expériences initial à 8 points (triangles noires). L'identification de cet ensemble n'a pas nécessité beaucoup de simulations par rapport au problème précédent. Pour résoudre ce problème, on a fait 288 appels au simulateur numérique.

L'objectif principal étant de répondre aux deux normes, on considère l'intersection des deux estimations comme résultat final de ce problème. La figure 5.16 montre l'estimation de $\Gamma_1^* \cap \Gamma_2^*$ représentée par l'ensemble de frontière bleue. On rappelle que les deux variables de contrôle agissent sur la quantité de solution d'urée permettant la production d'ammoniac pour réduire les oxydes d'azote. Ainsi l'ensemble identifié résulte d'un compromis entre un sur et sous dosage de cette solution. Par exemple, si on injecte une quantité minimale de solution (la zone proche du point $(10^{-2}, 10^{-2})$), on rejette une quantité excessive des oxydes d'azote et pour le coup, on obtient un score de DeNO_x bien supérieur au seuil.

Résumé : On a appliqué la méthodologie introduite dans le chapitre 3 pour identifier

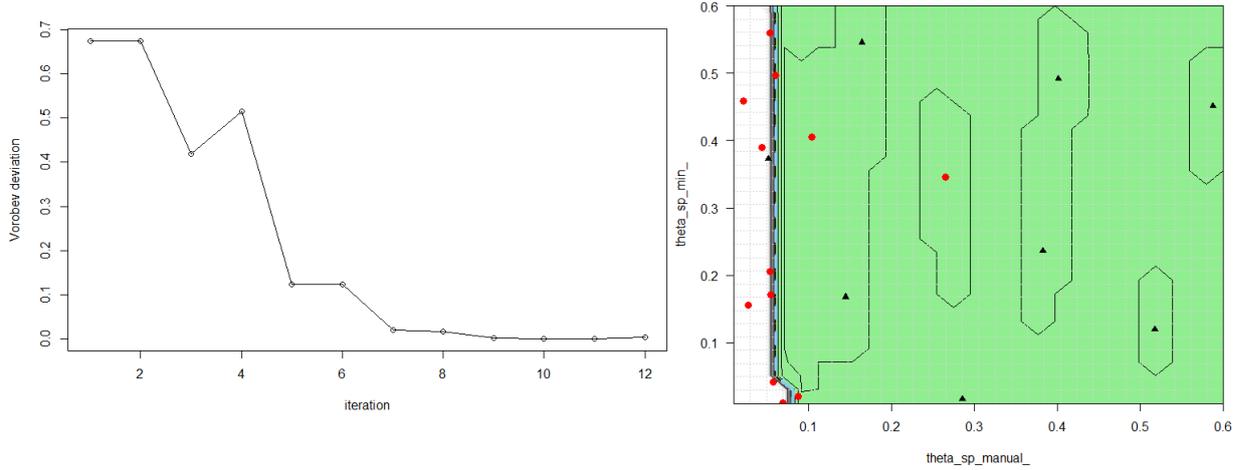


FIGURE 5.15: Score DeNO_x. A gauche : La *déviatiion de Vorob'ev* en fonction des itérations SUR. A droite : la fonction de couverture, le plan d'expériences initial (triangles noirs) et l'estimation de l'ensemble recherché Γ_1^* (en vert) après l'ajout de **12** points (en rouge) et 288 évaluations au total (évaluations pour l'estimation de l'espérance incluses).

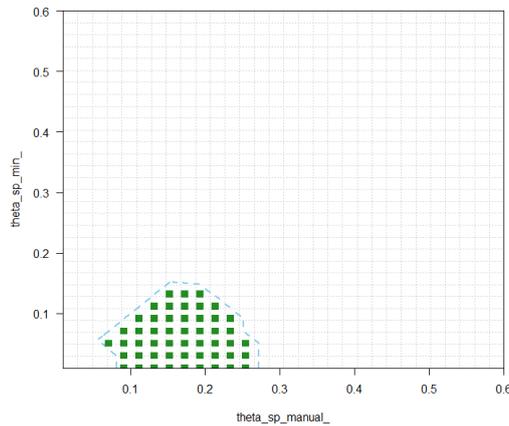


FIGURE 5.16: Le résultat final pour l'estimation de l'intersection $\Gamma_1^* \cap \Gamma_2^*$.

l'ensemble d'excursion au respect des deux normes. Cette approche permet d'avoir une estimation efficace de cet ensemble pour un coût de calcul dépendant de deux paramètres d'arrêt opérant à l'étape de l'estimation de l'espérance, soit ici 1575 simulations.

5.4.1.2 Inversion par métamodélisation dans l'espace joint

Dans cette partie, le même problème est résolu par la deuxième méthode présentée dans le chapitre 4. On rappelle que cette méthode repose sur un métamodèle de type processus gaussien défini dans l'espace des entrées du simulateur f (variables de contrôle et variable incertaine). Une première étape nécessaire est la caractérisation de la variable aléatoire fonctionnelle \mathbf{V} par un vecteur aléatoire \mathbf{U} , ce dernier est défini à l'aide d'une réduction de dimension et ce, en tronquant la décomposition KL de \mathbf{V} aux 20 premiers termes, afin d'expliquer 80% de la variance. On considère un processus gaussien défini donc en dimension 22 et noté $Z_{(\mathbf{x}, \mathbf{u})} \sim \mathcal{PG}(m, k)$ de moyenne constante et de noyau de covariance de type Matérn 5/2. Les paramètres du noyau et la moyenne sont estimés par maximum de vraisemblance. On considère un plan d'expériences initial à $n = 5 \times 22 = 110$ points dans $[0.01, 0.6]^2 \times \mathbb{R}^{20}$.

L'information disponible du simulateur f est donnée par l'évaluation de ce dernier aux points du plan d'expériences.

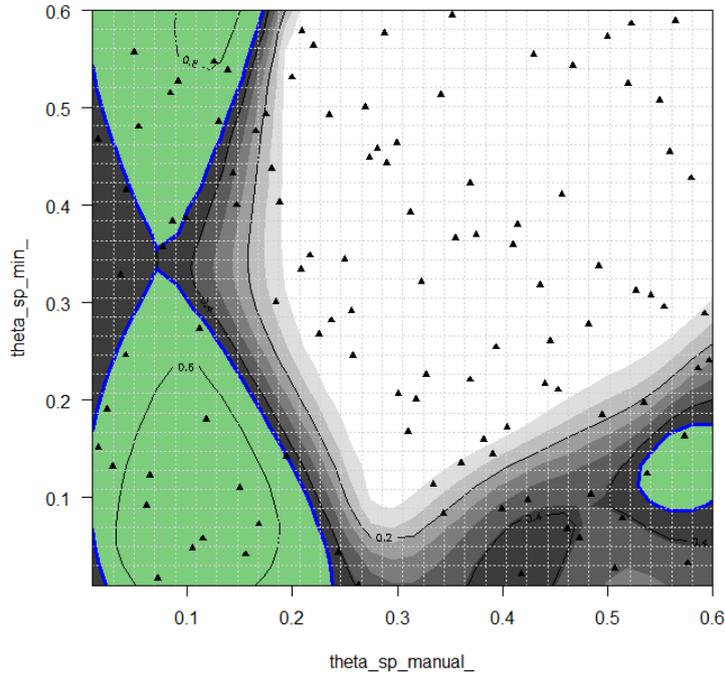


FIGURE 5.17: Pic d'ammoniac. Le plan d'expériences initial (triangles noirs) et l'estimation initiale de l'ensemble recherché Γ_2^* (en vert).

La figure 5.17 montre la fonction de couverture définie par le processus gaussien Y_x sachant l'information donnée par les évaluations aux points du plan d'expériences initial. L'estimation initiale de Γ_2^* est donnée par l'ensemble vert de frontière bleue. Le niveau de gris représente la probabilité que l'observation dépasse le seuil de 30 ppm.

On rappelle que cette méthode s'appuie sur le processus gaussien Y_x pour définir une stratégie de réduction séquentielle des incertitudes associée à l'estimation de l'ensemble d'excursion recherchée. Cette stratégie fournit donc le prochain point dans $[0.01, 0.6]^2$ et la prochaine courbe dans Ξ (cycle de conduite) où lancer simulateur f . Pour des raisons de coût de calcul liées aux problèmes d'optimisation cachés derrière cette stratégie (cf. chapitre 4), à chaque itération on cherche les deux points dans l'espace joint qui réduisent l'incertitude associée à l'estimation de l'ensemble Γ_2^* . Plus précisément, on procède en deux étapes. La première consiste à lancer la stratégie pour avoir le premier point. Ensuite et de la même manière, on cherche le second en s'appuyant sur un modèle de krigeage basé sur le plan d'apprentissage enrichi par le point de la première étape et ce, en remplaçant l'évaluation inconnue du simulateur numérique en ce point par la moyenne conditionnelle. On tient à noter que cette technique donne une solution sous-optimale mais économise du temps de calcul. Une itération de cette méthode est illustrée par la figure 5.18.

La figure de gauche dans 5.19 montre la décroissance de la *déviatio*n de Vorob'ev au cours des itérations. La figure de droite, l'estimation de l'ensemble recherché Γ_2^* est représentée par l'ensemble vert de frontière bleue et obtenue après avoir ajouté **400** points et le coût total pour résoudre le problème est de **510** évaluations du simulateur numérique.

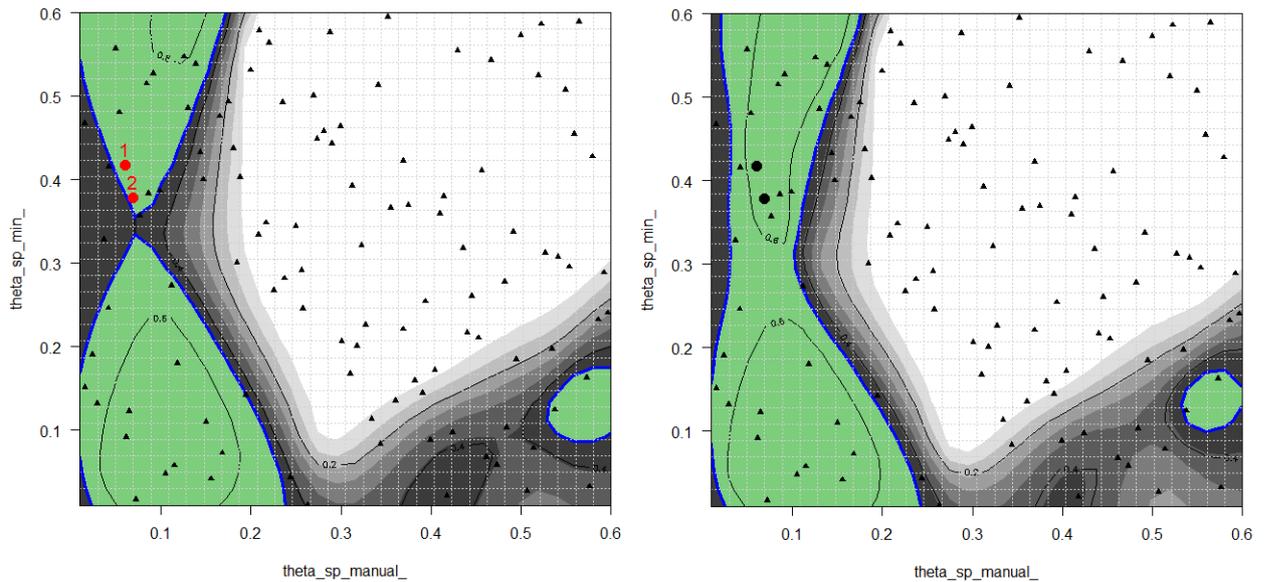


FIGURE 5.18: Pic d'ammoniac. A gauche : La fonction de couverture, le plan d'expériences initial (triangles noirs), l'estimation initiale de l'ensemble recherché Γ_2^* (en vert) et les deux points proposés par la stratégie SUR (en rouge). A droite : La mise à jour de l'estimation de Γ_2^* (en vert) après une itération.

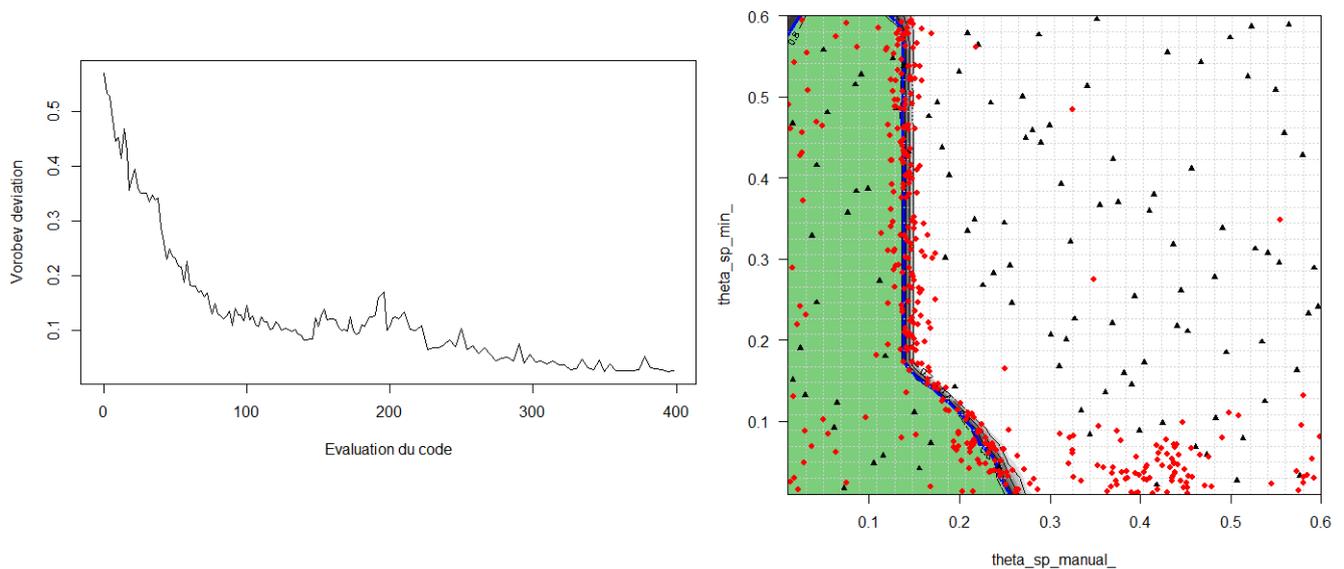


FIGURE 5.19: Pic d'ammoniac. A gauche : La *dévi*ation de Vorob'ev en fonction du nombre d'appels au simulateur f . A droite : la fonction de couverture, le plan d'expériences initial (triangles noirs) et l'estimation de l'ensemble recherché Γ_2^* (en vert) après l'ajout de 400 points (en rouge).

5.4.1.3 Analyse comparative des deux méthodes

La figure 5.20 montre les estimations obtenues pour le pic d'ammoniac par les deux méthodes à nombre d'évaluations du simulateur égal à 510.

On remarque que la méthode à base de modélisation dans l'espace joint (Chapitre 4) fournit une estimation beaucoup plus efficace que la méthode basée sur la quantification (Chapitre

3). En effet, pour un même budget d'évaluations fixé pour les 2 méthodes, on obtient, avec la méthode du Chapitre 3, une approximation de l'ensemble très proche de la solution finale obtenue avec la méthode 1 (voir la figure 5.14) alors que l'approximation obtenue par la méthode 1 est très éloignée de la solution.

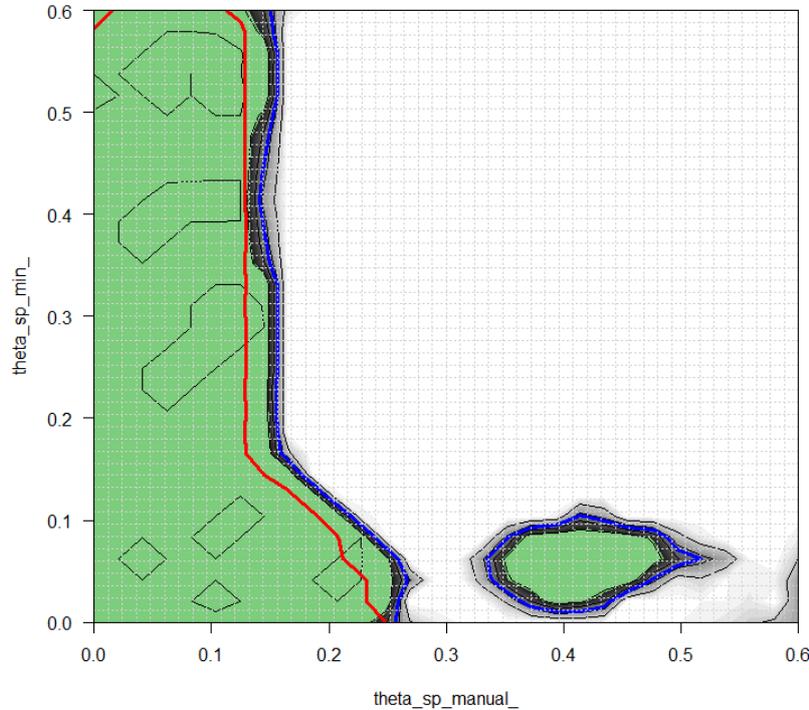


FIGURE 5.20: Résultats pour le pic d'ammoniac et pour un nombre d'évaluations du simulateur fixé à 510. L'ensemble d'excursion estimé par la première méthode (ensemble vert de frontière bleue) et la frontière rouge de l'ensemble estimé par la deuxième méthode.

5.4.2 Application : 9 variables de contrôle et 1 variable fonctionnelle incertaine

On a vu dans la section 5.4.1 que la deuxième méthode fournit une bonne estimation de l'ensemble recherché avec un faible nombre d'appels au simulateur. Néanmoins, dans le chapitre 4, nous avons illustré l'augmentation forte du coût de calcul de la méthode avec la dimension du problème et le nombre de points simulés.

Bien que la deuxième méthode soit efficace en terme de nombre de simulations, le temps de calcul associé à chaque itération nous pousse à appliquer la première méthode au problème d'inversion pour les 9 variables de contrôle.

On considère un processus gaussien $Y_{\mathbf{x} \in \mathbb{X}} \sim \mathcal{PG}(m, k)$ de moyenne constante et de noyau de covariance de type Matérn 5/2. Les paramètres du noyau et la moyenne sont estimés par maximum de vraisemblance. On considère un plan d'expériences initial à $n = 45$ points. En chaque point, on estime l'espérance séquentiellement par quantification fonctionnelle en fixant les paramètres d'arrêt à $(4, 10^{-2})$. La figure 5.21 montre la projection 2D des points de contrôle appartenant à l'estimation de l'intersection $\Gamma_1^* \cap \Gamma_2^*$. Ce résultat est obtenu après avoir ajouté 300 points de contrôle au plan d'expériences initial et 6924 évaluations du simulateur numérique. On remarque que le résultat obtenu fait apparaître certaines parties de l'espace de contrôle qui peuvent guider les experts à choisir des valeurs intéressantes pour certaines variables.

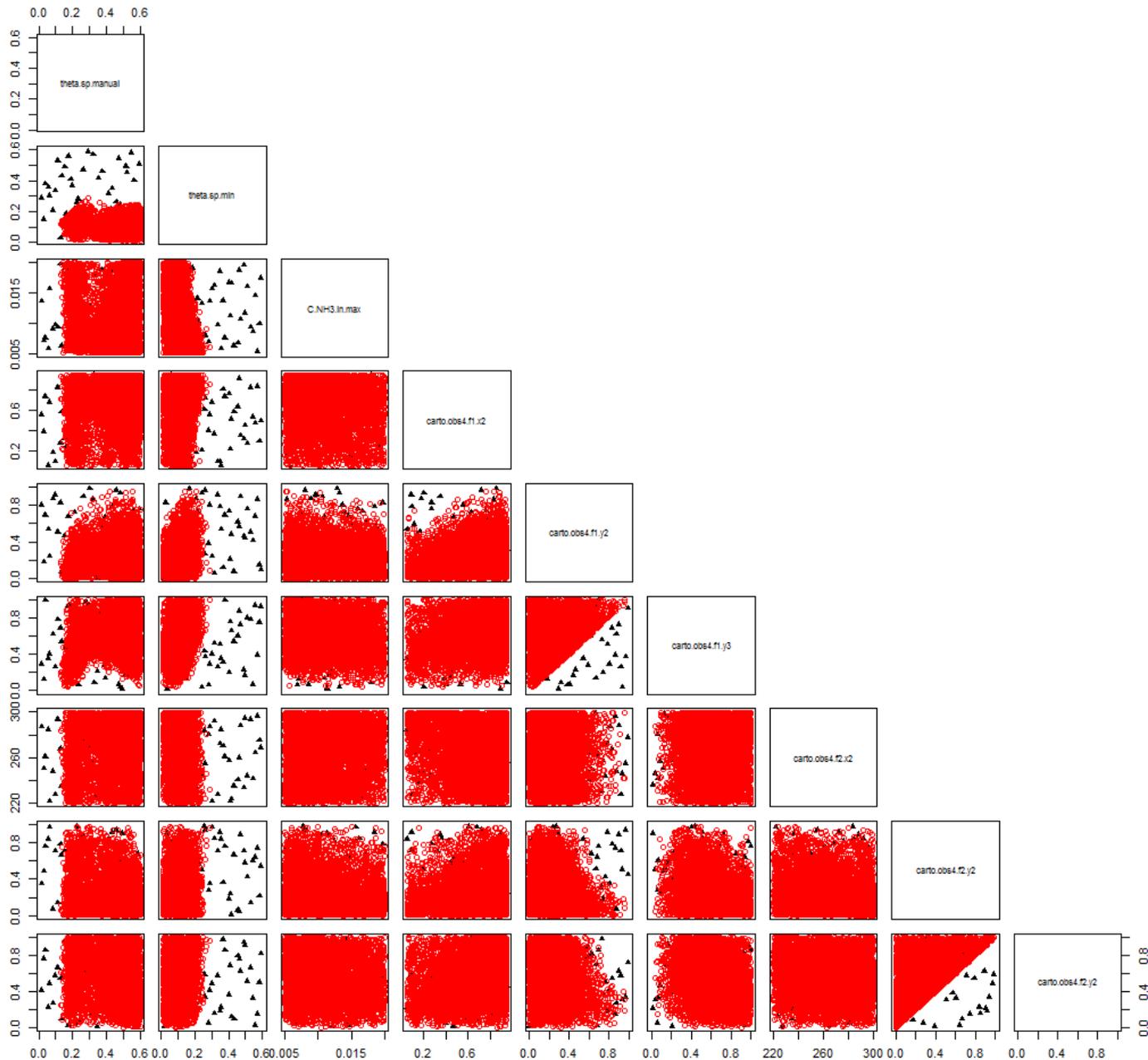


FIGURE 5.21: Nuages de points appartenant à l'estimateur de l'intersection de toutes les paires des variables de contrôle (les points rouges). Le plan d'expériences initial à 45 points (les triangles noirs).

Synthèse : Ce chapitre présente une application des méthodes introduites dans cette thèse pour ainsi répondre aux objectifs opérationnels au cas test industriel. Il s'agit de définir l'ensemble admissible des paramètres de contrôle du système de dépollution des gaz d'échappement afin de répondre aux normes européennes. La réponse apportée doit tenir compte de l'incertitude sur la conduite (cycle de vitesse).

Un premier travail a porté sur la visualisation des incertitudes fonctionnelles et ce, à travers divers outils. On a vu que ces derniers étaient adaptés aux variables fonctionnelles

bien approchées par les premières composantes de la décomposition KL.

Sur un cas d'application simplifié ne comprenant que 2 variables de contrôle, la méthode 2 (modélisation dans l'espace joint) donne de meilleurs résultats tant en précision qu'en nombre d'appels au simulateur. On a mis en évidence la robustesse de la deuxième approche en terme de précision et de nombre d'appels au simulateur numérique. En effet pour ce problème, cette méthode nous permet d'obtenir un résultat similaire à celui de la première approche avec trois fois moins d'appels au simulateur. Cependant, pour le cas d'application plus complexe défini à partir de 9 variables de contrôles, on a préféré la première approche à la deuxième pour des questions de temps de calcul du krigeage, les évaluations du code de calcul n'étant pas très onéreuses.

Conclusions et perspectives

Dans ce travail de thèse nous apportons des réponses au problème d'inversion de code de calcul coûteux en présence d'incertitudes, notamment d'entrées fonctionnelles incertaines. Ces incertitudes sont modélisées par des variables aléatoires dont la distribution est résumée par l'espérance. L'objectif est de trouver l'ensemble des paramètres de contrôle du système (ensemble d'excursion) tels que le système, en moyenne, satisfait les normes imposées. Nous proposons deux méthodes différentes basées sur une démarche de métamodélisation du code de calcul par processus gaussiens :

- la première approche consiste à proposer des méthodes efficaces d'approximation de l'espérance et à construire un métamodèle sur l'espérance,
- la deuxième approche propose une modélisation par processus gaussiens dans l'espace joint (paramètres de contrôle, entrées incertaines).

Ces deux méthodologies sont déployées dans le contexte spécifique d'entrées aléatoires fonctionnelles et sont couplées à une méthode de réduction de dimension.

Une étape importante de la première méthodologie a consisté à développer deux algorithmes gloutons de quantification fonctionnelle dans le cas où la variable aléatoire n'est connue qu'à travers un échantillon de réalisations (cf. chapitre 2). Ces deux algorithmes ont comme point commun une réduction de dimension dans le but de se placer dans un cadre de dimension finie. L'un repose sur le concept de la quantification fonctionnelle tandis que l'autre se réfère aux notions de remplissage optimal de l'espace (le critère maximin). On a montré que ces deux algorithmes sont robustes par rapport à la réduction de dimension et efficaces comparés aux méthodes standards (Monte Carlo). On a aussi constaté que l'approche basée sur la quantification fournit une estimation séquentielle de l'espérance plus efficace par rapport à l'approche basée sur un critère maximin. Dans cette première méthodologie dédiée à l'inversion, la résolution du problème sous incertitudes fonctionnelles repose sur la construction d'un métamodèle de l'espérance de la sortie. Une stratégie définie à partir de l'erreur de prédiction est ensuite utilisée pour choisir judicieusement les points dans l'espace de contrôle. Couplée aux algorithmes de quantification fonctionnelle, cette méthodologie fournit une estimation efficace à faible coût de l'ensemble d'excursion recherché (cf. chapitre 3). Cependant l'étape dédiée à l'enrichissement du plan d'apprentissage nécessite des appels au simulateur pour l'estimation de l'espérance.

La deuxième méthodologie que nous avons présentée au chapitre 4 a pour but de simplifier la première approche pour cette estimation de l'espérance. Cette méthode fournit une nouvelle stratégie d'enrichissement adaptatif de plans d'expériences. Cette dernière repose sur une modélisation jointe par processus gaussien et la possibilité d'en déduire une formule semi-analytique pour l'espérance. La stratégie pour le choix des points d'évaluations dans l'espace joint est ensuite guidée par l'erreur de prédiction du krigeage. Cette dernière approche

a été comparée à la première méthodologie à travers deux exemples analytiques. Les résultats obtenus sont concluants en terme de nombre de simulations et de précision. Cependant, ces résultats ont montré un inconvénient de la méthode lié à son coût de calcul intrinsèque. En effet, l'espace des entrées du processus gaussien comprend, en plus des paramètres de contrôle, le vecteur résultant de la réduction de la variable fonctionnelle. Comme la taille de l'échantillon d'apprentissage augmente au cours de l'enrichissement (des itérations), le coût de construction du krigeage ainsi que additionné au coût de la recherche du meilleur prochain point, due au calcul d'intégrales multidimensionnelles, devient très vite prohibitif.

Pour finir, ces deux méthodes ont été appliquées avec succès à un problème industriel (cf. chapitre 5). Il s'agit de régler, sous incertitudes fonctionnelles liées au cycle de conduite, la stratégie de contrôle d'un système de dépollution d'un véhicule afin de respecter des normes de pollution. Appliquées dans un premier temps à un problème défini en 2D (2 paramètres de contrôle et une variable incertaine fonctionnelle), les deux méthodes fournissent des résultats très similaires. Comparée à la première, la deuxième approche nécessite trois fois moins d'appels au simulateur numérique mais demande quelques jours de calcul pour les raisons évoquées dans le paragraphe précédent. On a donc décidé d'appliquer la première approche pour traiter le problème industriel initial défini en 9D (9 paramètres de contrôle et une variable incertaine fonctionnelle).

A travers divers exemples analytiques et le cas industriel, on a souligné le compromis entre le coût de calcul d'une simulation et le coût du choix de nouveau point à chaque itération. Ce compromis pourrait servir de guide pour le choix de la méthode à considérer. En effet, dans le cas où une simulation prend quelques jours, alors on préférera la deuxième méthode (Chapitre 4) à la première (Chapitre 3). Au contraire, si le coût d'une simulation n'excède pas quelques minutes, le calcul du nouveau point devient plus coûteux. Ainsi, dans un tel contexte, on préférera la première méthodologie.

Nous présentons quelques perspectives à ce travail de thèse. Ces perspectives portent sur la quantification fonctionnelle gloutonne pour certains processus gaussiens, l'optimisation sous incertitude, la prise en compte d'un mélange d'incertitudes (fonctionnelle et scalaire) et le cas où le simulateur renvoie des sorties vectorielles.

Dans le chapitre 2, nous avons développé des algorithmes gloutons de quantification fonctionnelle, en fixant a priori la dimension de la troncature. Or dans [Pages and Printems, 2009] et pour certains processus gaussiens, les auteurs donnent la dimension de troncature en fonction de la taille du quantifieur. Ainsi on pourrait envisager une quantification gloutonne en taille et dimension. Plus précisément, cela consisterait à construire séquentiellement un quantifieur de taille l dans un espace réduit défini par la décomposition de Karunhen-Loeve optimale à l fixé.

Les méthodologies présentées dans cette thèse sont dédiées à la prise en compte des incertitudes fonctionnelles lors de d'inversion, elles peuvent aussi être appliquées pour des problèmes d'optimisation sous incertitudes fonctionnelles. La première méthode proposée dans la thèse portant sur la quantification peut aisément être couplée à de nombreux problèmes portant sur l'optimisation stochastique. En ce qui concerne la deuxième approche proposée, il serait par exemple possible de repenser le critère du choix des points pour tenir compte des incertitudes fonctionnelles dans un algorithme de type EGO.

La première contribution à base de quantification gloutonne permet de tenir compte des incertitudes fonctionnelles. Dans le futur, il serait possible d'étendre cette méthode pour prendre en compte toutes les sources d'incertitudes (combinant variables scalaires et fonctionnelles). Pour la variable fonctionnelle, la construction du quantifieur est faite dans l'espace des coefficients. Il pourrait être intéressant d'augmenter cet espace en ajoutant des variables scalaires, ainsi on pourrait envisager d'adapter notre algorithme de quantification pour une construction séquentielle en ces deux espaces. Cela nécessiterait de définir une distance sur l'ensemble des deux espaces. En ce qui concerne la deuxième méthodologie, cette dernière pourrait être appliquée de manière directe.

Les deux méthodologies présentées dans cette thèse s'appliquent aux simulateurs numériques à sorties scalaires. Une extension intéressante serait d'étendre ces méthodes au cas des sorties vectorielles. Dans [Pourmohamad et al., 2016], les auteurs proposent une modélisation par processus gaussien tenant compte des éventuelles corrélations entre les sorties du simulateur. Des résultats satisfaisants ont été obtenus en comparant cette approche à celle consistant à créer un modèle de krigeage pour chaque sortie du simulateur. Ainsi pour les problèmes d'inversion présentés dans cette thèse, il serait intéressant d'appliquer cette modélisation et d'utiliser l'incertitude associée pour définir une stratégie de choix des points adaptée aux sorties vectorielles.

Enfin dans un cadre industriel et pour le cas d'application qui a motivé ces travaux de recherche, il serait intéressant de tenir compte des sources d'incertitudes scalaires dans le but d'améliorer le système de contrôle.

Une autre formulation du problème d'inversion peut être établie en considérant une mesure de robustesse autre que l'espérance, par exemple résoudre le problème en grande probabilité. Dans un tel contexte, en se fixant une probabilité faible, on fera certainement appel aux méthodes d'échantillonnage d'importance. On pourrait aussi voir la probabilité comme l'espérance d'une indicatrice. Cela ouvre la possibilité d'utiliser des algorithmes à gradient stochastique mais elle soulève également la difficulté de gérer une fonction à forte discontinuité, à savoir la fonction indicatrice.

Annexe A

Conception du générateur du cycle de conduite

Un mot sur la conception du générateur de cycles de conduite

Un cycle de conduite est une courbe de la vitesse du véhicule en fonction du temps sur un intervalle donné. Le plus souvent la courbe est discrétisée en un nombre fini de pas de temps. Le NEDC (New European Driving Cycle) est un cycle particulier sur lequel peuvent être comparées les émissions de polluants de différents véhicules et qui sert à l'attribution des homologations. Sauf que ce dernier est critiqué par les experts car il ne représente pas les conditions réelles de conduite, de ce fait, la conception d'un générateur de cycles réalistes de conduite s'avère nécessaire pour s'approcher des cycles réels sur lesquels les normes devront être satisfaites.

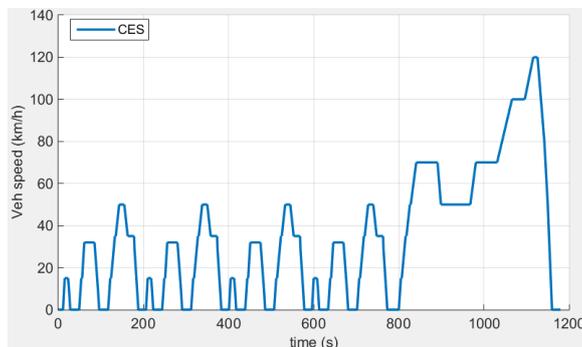


FIGURE A.1: Le cycle de conduite NEDC

La conception a nécessité l'enregistrement de 13 cycles de conduite en condition réelle. En chacun des 3.1×10^6 pas de temps de ces cycles, on peut définir une accélération et une vitesse. L'objectif est alors de chercher à identifier un modèle markovien d'un état $(a, v)_t$ à l'instant t vers un état $(a, v)_{t+1}$ à l'instant $t + 1$. Le modèle proposé est un modèle discret : les accélérations sont discrétisées avec un pas de 0.2 m/s^2 et de 5 km/h pour la vitesse.

Exemple : Pour le couple $(a_k = 0 \text{ m/s}^2, v_k = 30 \text{ km/h})$, on commence par extraire les différentes valeurs de la future accélération a_{k+1} , ensuite on estime les paramètres de la loi normale qui s'ajustent au mieux à ces observations. Ces paramètres (moyenne, variance) seront stockés dans une matrice **M**.

La génération des cycles de conduite

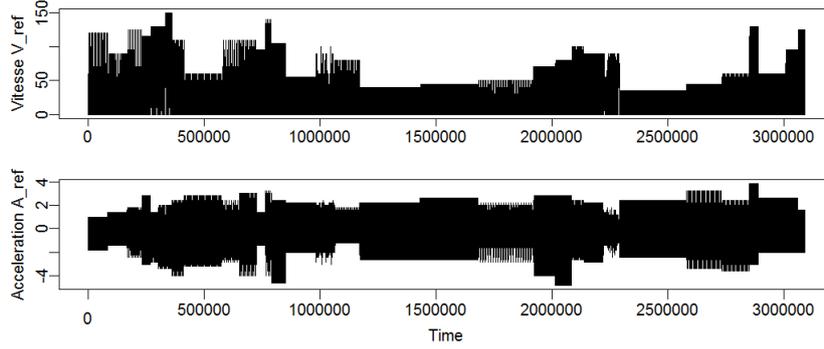


FIGURE A.2: Les 13 cycles enregistrés concaténés

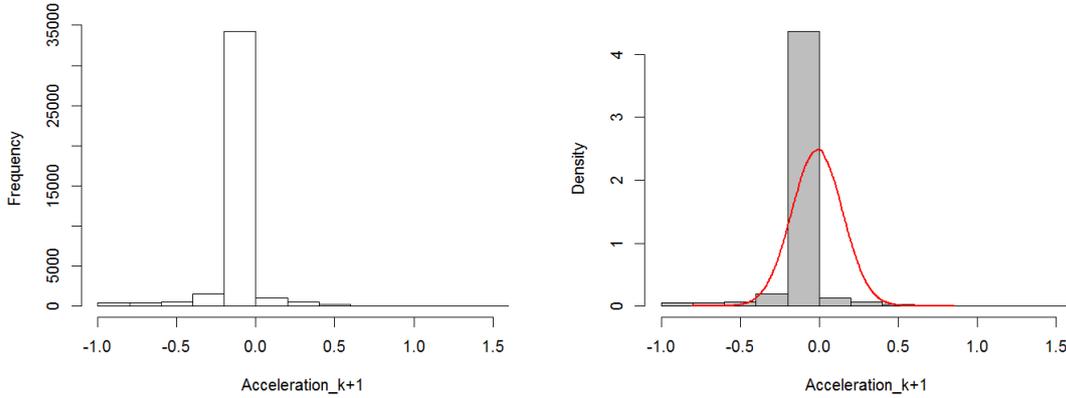


FIGURE A.3: Histogramme et densité de probabilité des accélérations a_{k+1} sachant ($a_k = 0, v_k = 30$)

Le schéma A.4 explique la méthode de construction implémentée dans le code. A l’instant $t = 0(s)$, on est à l’arrêt, i.e., à l’état $e_t = (a_t = 0, v_t = 0)$. On extrait de la matrice \mathbf{M} les paramètres de la loi lognormale qui modélise la future accélération sachant e_t , $\mathcal{L}(a_{t+1}|e_t)$, cette étape est représentée dans le schéma par le bloc **1**. Ensuite, un tirage suivant la loi uniforme sur $]0, 1[$, $p \sim \mathcal{U}(]0, 1[)$, est effectué, puis on calcule le quantile d’ordre p pour $\mathcal{L}(a_{t+1}|e_t)$ (bloc **2**). Ce quantile représente notre accélération a_{t+1} et on en déduit la vitesse v_{t+1} (bloc **3**)

$$v_{t+1} = v_t + \Delta t \times a_{t+1} , \Delta t : \text{le pas de discrétisation en temps}$$

A l’instant d’après, i.e., $t=1(s)$, on est à l’état $e_{t+1} = (a_{t+1}, v_{t+1})$. Si cet état n’est pas représenté dans la matrice \mathbf{M} , on récupère les paramètres de la loi normale modélisant $\mathcal{L}(a_{t+2}|e_{t+1})$ par interpolation (bloc **4**). On procède ensuite comme précédemment avec les étapes **4** et **5**.

On réitère (**3** \rightarrow **4** \rightarrow **5**) jusqu’à atteindre la durée souhaitée du cycle. Une fois la construction finie, ce dernier doit passer une étape de validation et pour cela, **un processus de rejet** est mis en place pour vérifier si le cycle généré est bien ”plausible”, ainsi on rejette les cycles vérifiant les critères listés ci-dessous :

- existence de longs instants à vitesse nulle au début et à la fin du cycle,
- la vitesse varie, entre 3 et 10 km/h, plus de 20 secondes,
- existence de longs instants d’accélérations ou de décélérations
- le cycle ne finit pas à l’arrêt.

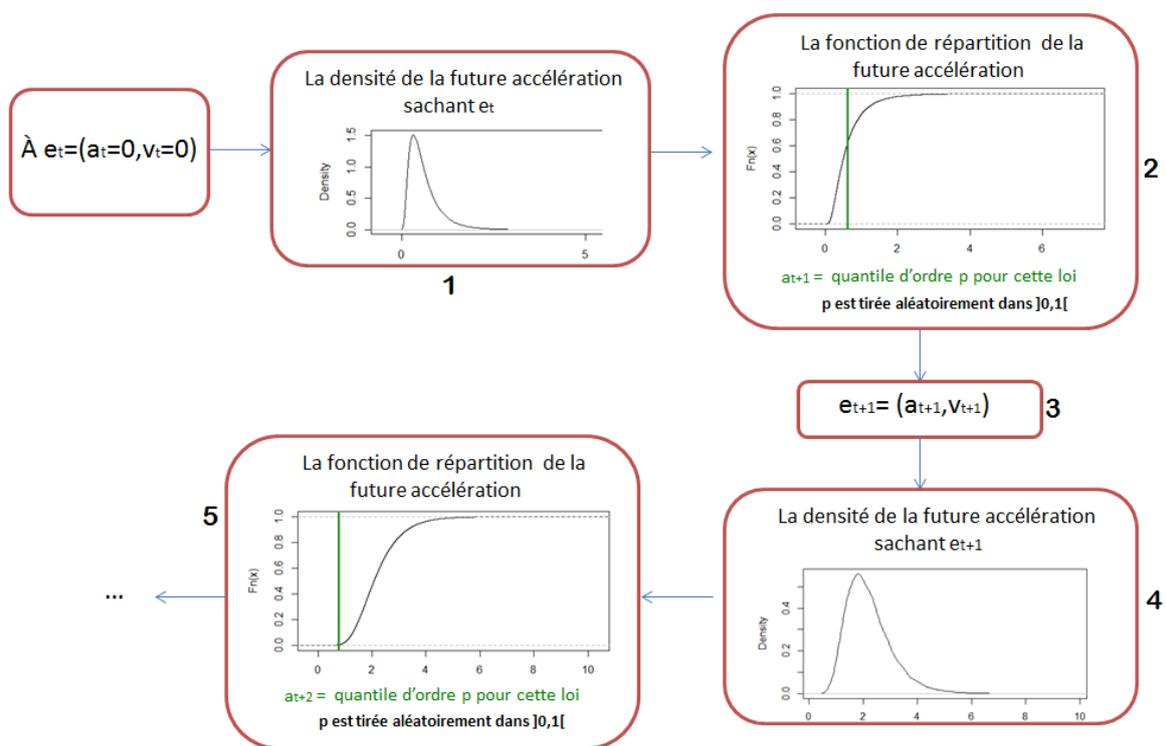


FIGURE A.4: Un schéma décrivant la génération des cycles de conduite.

Bibliographie

- [Baillo et al., 2000] Baillo, A., Cuevas, A., and Justel, A. (2000). Set estimation and nonparametric detection. *Canadian Journal of Statistics*, 28(4) :765–782.
- [Bally et al., 2005] Bally, V., Pagès, G., and Printems, J. (2005). A quantization tree method for pricing and hedging multidimensional american options. *Mathematical Finance : An International Journal of Mathematics, Statistics and Financial Economics*, 15(1) :119–168.
- [Bect et al., 2016] Bect, J., Bachoc, F., and Ginsbourger, D. (2016). A supermartingale approach to gaussian process based sequential design of experiments. *arXiv preprint arXiv :1608.01118*.
- [Bect et al., 2012] Bect, J., Ginsbourger, D., Li, L., Picheny, V., and Vazquez, E. (2012). Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3) :773–793.
- [Bolin and Lindgren, 2015] Bolin, D. and Lindgren, F. (2015). Excursion and contour uncertainty regions for latent gaussian models. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 77(1) :85–106.
- [Bonfils et al., 2012] Bonfils, A., Creff, Y., Lepreux, O., and Petit, N. (2012). Closed-loop control of a scr system using a nox sensor cross-sensitive to nh3. *IFAC Proceedings Volumes*, 45(15) :738–743.
- [Cardot et al., 1999] Cardot, H., Ferraty, F., and Sarda, P. (1999). Functional linear model. *Statistics & Probability Letters*, 45(1) :11–22.
- [Chevalier et al., 2014a] Chevalier, C., Bect, J., Ginsbourger, D., Vazquez, E., Picheny, V., and Richet, Y. (2014a). Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56(4) :455–465.
- [Chevalier et al., 2015] Chevalier, C., Emery, X., and Ginsbourger, D. (2015). Fast update of conditional simulation ensembles. *Mathematical Geosciences*, 47(7) :771–789.
- [Chevalier and Ginsbourger, 2013] Chevalier, C. and Ginsbourger, D. (2013). Fast computation of the multi-points expected improvement with applications in batch selection. In *International Conference on Learning and Intelligent Optimization*, pages 59–69. Springer.
- [Chevalier et al., 2013] Chevalier, C., Ginsbourger, D., Bect, J., and Molchanov, I. (2013). Estimating and quantifying uncertainties on level sets using the vorob’ev expectation and deviation with gaussian process models. In *mODa 10–Advances in Model-Oriented Design and Analysis*, pages 35–43. Springer.
- [Chevalier et al., 2014b] Chevalier, C., Picheny, V., and Ginsbourger, D. (2014b). Krigingv : An efficient and user-friendly implementation of batch-sequential inversion strategies based on kriging. *Computational statistics & data analysis*, 71 :1021–1034.
- [Chiles and Delfiner, 1999] Chiles, J.-P. and Delfiner, P. (1999). *Geostatistics : modeling spatial uncertainty*. Wiley, New York.

- [Cuevas and Fraiman, 1997] Cuevas, A. and Fraiman, R. (1997). A plug-in approach to support estimation. *The Annals of Statistics*, pages 2300–2312.
- [Cuevas et al., 2006] Cuevas, A., González-Manteiga, W., and Rodríguez-Casal, A. (2006). Plug-in estimation of general level sets. *Australian & New Zealand Journal of Statistics*, 48(1) :7–19.
- [Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.
- [Devroye and Wise, 1980] Devroye, L. and Wise, G. L. (1980). Detection of abnormal behavior via nonparametric estimation of the support. *SIAM Journal on Applied Mathematics*, 38(3) :480–488.
- [El Amri et al., 2018] El Amri, M. R., Helbert, C., Lepreux, O., Munoz Zuniga, M., Prieur, C., and Sinoquet, D. (2018). Data-driven stochastic inversion under functional uncertainties. working paper or preprint.
- [Flury, 1990] Flury, B. A. (1990). Principal points. *Biometrika*, 77(1) :33–41.
- [Franco, 2008] Franco, J. (2008). *Planification d’expériences numériques en phase exploratoire pour la simulation des phénomènes complexes*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne.
- [French et al., 2013] French, J. P., Sain, S. R., et al. (2013). Spatio-temporal exceedance locations and confidence regions. *The Annals of Applied Statistics*, 7(3) :1421–1449.
- [Ginsbourger et al., 2009] Ginsbourger, D., Dupuy, D., Badea, A., Carraro, L., and Roustant, O. (2009). A note on the choice and the estimation of kriging models for the analysis of deterministic computer experiments. *Applied Stochastic Models in Business and Industry*, 25(2) :115–131.
- [Graf and Luschgy, 2000] Graf, S. and Luschgy, H. (2000). *Foundations of quantization for probability distributions*. Springer.
- [Hyndman and Shang, 2010] Hyndman, R. J. and Shang, H. L. (2010). Rainbow plots, bag-plots, and boxplots for functional data. *Journal of Computational and Graphical Statistics*, 19(1) :29–45.
- [Janusevskis and Le Riche, 2013] Janusevskis, J. and Le Riche, R. (2013). Simultaneous kriging-based estimation and optimization of mean response. *Journal of Global Optimization*, 55(2) :313–336.
- [Jin et al., 2005] Jin, R., Chen, W., and Sudjianto, A. (2005). An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference*, 134(1) :268–287.
- [Johnson et al., 1990] Johnson, M. E., Moore, L. M., and Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of statistical planning and inference*, 26(2) :131–148.
- [L’Ecuyer and Lemieux, 2005] L’Ecuyer, P. and Lemieux, C. (2005). Recent advances in randomized quasi-monte carlo methods. In *Modeling uncertainty*, pages 419–474. Springer.
- [Lemieux, 2009] Lemieux, C. (2009). Quasi-monte carlo constructions. In *Monte Carlo and Quasi-Monte Carlo Sampling*, pages 1–61. Springer.
- [Levrard, 2014] Levrard, C. (2014). *High-dimensional vector quantization : convergence rates and variable selection*. PhD thesis, Université de Paris 11.
- [López-Pintado and Jornsten, 2007] López-Pintado, S. and Jornsten, R. (2007). Functional analysis via extensions of the band depth. *Lecture Notes-Monograph Series*, pages 103–120.

- [López-Pintado and Romo, 2009] López-Pintado, S. and Romo, J. (2009). On the concept of depth for functional data. *Journal of the American Statistical Association*, 104(486) :718–734.
- [Luschgy and Pagès, 2002] Luschgy, H. and Pagès, G. (2002). Functional quantization of gaussian processes. *Journal of Functional Analysis*, 196(2) :486–531.
- [Luschgy and Pagès, 2015] Luschgy, H. and Pagès, G. (2015). Greedy vector quantization. *Journal of Approximation Theory*, 198 :111–131.
- [Luschgy et al., 2008] Luschgy, H., Pagès, G., et al. (2008). Functional quantization rate and mean regularity of processes with an application to lévy processes. *The Annals of Applied Probability*, 18(2) :427–469.
- [Luschgy et al., 2010] Luschgy, H., Pagès, G., and Wilbertz, B. (2010). Asymptotically optimal quantization schemes for gaussian processes on hilbert spaces. *ESAIM : Probability and Statistics*, 14 :93–116.
- [Matheron, 1973] Matheron, G. (1973). The intrinsic random functions and their applications. *Advances in applied probability*, 5(3) :439–468.
- [Miranda and Bocchini, 2013] Miranda, M. and Bocchini, P. (2013). Functional quantization of stationary gaussian and non-gaussian random processes. *Safety, Reliability, Risk and Life-Cycle Performance of Structures and Infrastructures*, pages 2785–2792.
- [Miranda and Bocchini, 2015] Miranda, M. J. and Bocchini, P. (2015). A versatile technique for the optimal approximation of random processes by functional quantization. *Applied Mathematics and Computation*, 271 :935–958.
- [Molchanov, 2006] Molchanov, I. (2006). *Theory of random sets*. Springer Science & Business Media.
- [Molchanov, 1998] Molchanov, I. S. (1998). A limit theorem for solutions of inequalities. *Scandinavian Journal of Statistics*, 25(1) :235–242.
- [Morris and Mitchell, 1995] Morris, M. D. and Mitchell, T. J. (1995). Exploratory designs for computational experiments. *Journal of statistical planning and inference*, 43(3) :381–402.
- [Nanty et al., 2016] Nanty, S., Helbert, C., Marrel, A., Pérot, N., and Prieur, C. (2016). Sampling, metamodeling, and sensitivity analysis of numerical simulators with functional stochastic inputs. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1) :636–659.
- [Niederreiter, 1992] Niederreiter, H. (1992). *Random number generation and quasi-Monte Carlo methods*, volume 63. Siam.
- [Owen, 2013] Owen, A. B. (2013). *Monte Carlo theory, methods and examples*.
- [Pagès, 1998] Pagès, G. (1998). A space quantization method for numerical integration. *Journal of computational and applied mathematics*, 89(1) :1–38.
- [Pagès, 2014] Pagès, G. (2014). Introduction to optimal vector quantization and its applications for numerics.
- [Pagès, 2018] Pagès, G. (2018). *Numerical Probability : An Introduction with Applications to Finance*. Springer.
- [Pages et al., 2004] Pages, G., Pham, H., and Printems, J. (2004). An optimal markovian quantization algorithm for multi-dimensional stochastic control problems. *Stochastics and Dynamics*, 4(04) :501–545.
- [Pagès and Printems, 2003] Pagès, G. and Printems, J. (2003). Optimal quadratic quantization for numerics : the gaussian case. *Monte Carlo Methods and Applications mcma*, 9(2) :135–165.

- [Pagès and Printems, 2005] Pagès, G. and Printems, J. (2005). Functional quantization for numerics with an application to option pricing. *Monte Carlo Methods and Applications mcma*, 11(4) :407–446.
- [Pages and Printems, 2009] Pages, G. and Printems, J. (2009). Optimal quantization for finance : from random vectors to stochastic processes. In *Handbook of Numerical Analysis*, volume 15, pages 595–648. Elsevier.
- [Palacios and Steel, 2006] Palacios, M. B. and Steel, M. F. J. (2006). Non-gaussian bayesian geostatistical modeling. *Journal of the American Statistical Association*, 101(474) :604–618.
- [Petersen et al., 2008] Petersen, K. B., Pedersen, M. S., et al. (2008). The matrix cookbook. *Technical University of Denmark*, 7(15) :510.
- [Picheny et al., 2010] Picheny, V., Ginsbourger, D., Roustant, O., Haftka, R. T., and Kim, N.-H. (2010). Adaptive designs of experiments for accurate approximation of a target region. *Journal of Mechanical Design*, 132(7) :071008.
- [Pourmohamad et al., 2016] Pourmohamad, T., Lee, H. K., et al. (2016). Multivariate stochastic process models for correlated responses of mixed type. *Bayesian Analysis*, 11(3) :797–820.
- [Pronzato and Müller, 2012] Pronzato, L. and Müller, W. G. (2012). Design of computer experiments : space filling and beyond. *Statistics and Computing*, 22(3) :681–701.
- [Ramsay, 2006] Ramsay, J. O. (2006). *Functional data analysis*. Wiley Online Library.
- [Ranjan et al., 2008] Ranjan, P., Bingham, D., and Michailidis, G. (2008). Sequential experiment design for contour estimation from complex computer codes. *Technometrics*, 50(4) :527–541.
- [Rasmussen and Williams, 2006] Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian processes for machine learning*, volume 1. MIT press Cambridge.
- [Rosenblatt, 1956] Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, pages 832–837.
- [Roustant et al., 2012] Roustant, O., Ginsbourger, D., and Deville, Y. (2012). Dicekriging, diceoptim : Two r packages for the analysis of computer experiments by kriging-based metamodeling and optimization.
- [Santner et al., 2013] Santner, T. J., Williams, B. J., and Notz, W. I. (2013). *The design and analysis of computer experiments*. Springer Science & Business Media.
- [Sun and Genton, 2011] Sun, Y. and Genton, M. G. (2011). Functional boxplots. *Journal of Computational and Graphical Statistics*, 20(2) :316–334.
- [Tran, 2014] Tran, V. C. (2014). *Une ballade en forêts aléatoires*. PhD thesis, Université Lille 1.
- [Tukey, 1977] Tukey, J. W. (1977). *Exploratory data analysis*, volume 2. Reading, Mass.
- [Vazquez and Bect, 2009] Vazquez, E. and Bect, J. (2009). A sequential bayesian algorithm to estimate a probability of failure. *IFAC Proceedings Volumes*, 42(10) :546–550.
- [Vorob’ev, 1984] Vorob’ev, O. Y. (1984). Srednemernoje modelirovanie (mean-measure modelling).
- [Williams et al., 2000] Williams, B. J., Santner, T. J., and Notz, W. I. (2000). Sequential design of computer experiments to minimize integrated response functions. *Statistica Sinica*, pages 1133–1152.
- [Zador, 1963] Zador, P. (1963). Development and evaluation of procedures for quantizing multivariate distributions. Technical report, STANFORD UNIV CALIF.