université PARIS DIDEROT PARIS 7

Thèse

Pour obtenir le grade de

Docteur de l'Université de Paris-Diderot

Présentée par

– Loic Le Gratiet –

Multi-fidelity Gaussian process regression for computer experiments

Thèse dirigée par **Josselin Garnier**

Thèse rapportée par

Dave Higdon	Directeur de recherche, LANL
Olivier Roustant	Professeur agrégé, ENSM-SE

Thèse soutenue publiquement le 04 Octobre 2013 devant le jury composé de

Claire Cannamela	Ingénieur chercheur, CEA	Encadrante
Josselin Garnier	Professeur, Université Paris VII	Directeur de thèse
Bertrand Iooss	Chercheur senior, EDF R&D	Examinateur
Hervé Monod	Directeur de recherche, INRA	Examinateur
Dominique Picard	Professeur, Université Paris VII	Examinateur
Luc Pronzato	Directeur de recherche, CNRS	Examinateur
Olivier Roustant	Professeur agrégé, ENSM-SE	Rapporteur

À mon père, Jean,
à ma mère, Sophie,
à ma soeur, Claire,
à mon frère, Yann
et à ma nièce, Léa.

Remerciements

Je voudrais avant tout remercier mon directeur de thèse Josselin Garnier pour m'avoir guidé et encadré tout au long de ces trois années de thèse. En plus de ses conseils scientifiques de hauts niveaux, Josselin a su pour moi trouver l'équilibre parfait entre l'encadrement et le laisser-faire. Je suis conscient que sans cela ma thèse aurait été bien autre chose et surtout aurait pu être beaucoup moins agréable. Je tiens à souligner ses formidables qualités humaines qui pour ma part ont été d'une grande importance pour cette collaboration de trois ans.

Ensuite je tiens à remercier spécialement Claire Cannamela qui a été une encadrante idéale pendant ces trois ans. Sans elle je pense que j'aurais pris beaucoup moins de plaisir à me lever tous les matins pour aller travailler. Je suis également heureux que l'on ait pu troquer notre statut de collègue pour celui d'ami. Je ne suis pas prêt d'oublier ce magnifique voyage au Maroc que l'on a fait avec Alexia et qui m'a donné une bouffée d'air pendant ma longue et fastidieuse rédaction...

J'adresse également mes remerciements à Dave Higdon et Olivier Roustant d'avoir accepté de rapporter ma thèse et à Bertand Iooss, Hervé Monod, Dominique Picard et Luc Pronzato d'avoir accepté de faire partie de mon jury de thèse en tant qu'examinateurs. Je donne une mention spéciale à Olivier qui a été un professeur marquant durant mes études et à Bertrand qui a animé de manière significative deux semaines de ma thèse à l'occasion d'une école d'été mémorable. C'est aussi un peu grâce à lui que j'ai pu écrire le dernier paragraphe de ces remerciements.

Mes prochains remerciements seront pour Françoise Poggi qui a su se battre pour faire vivre un laboratoire traitant des incertitudes au sein du CEA et ainsi pu permettre le financement de ma thèse. Son énergie et sa motivation sont remarquables et remarquées...

Un grand merci également à tous mes collègues du CEA qui m'ont fait passer d'agréables moments et avec qui j'ai pu avoir d'intéressants échanges pendant mes nombreuses (mais pas trop quand même) pauses café (j'en profite aussi pour remercier ma machine Nespresso... what else !). Parmi ces collègues je pourrais citer Alexia qui a partagé mon bureau pendant de nom-

breux mois et qui me faisait monter sur les tables pour aller cueillir les araignées, Françoise pour avoir supporté mon bureau sale pendant deux ans, Sophie et Sophie pour leur apéros traquenards, Flavie pour ses excellents rapports sur les ragots du service, Marie, Marie, Yann, Gilles, Jean, Hervé, Michelle, Gaël et les différents habitants du couloir des thésards et stagiaires qui ont été mes collègues pendant mes derniers mois de thèse.

Je tiens bien entendu à remercier ma famille. Ma grand-mère Claudine, mon père Jean, ma mère Sophie, ma sœur Claire et mon frère Yann. Finalement ils sont tous directement responsables de qui je suis devenu et m'ont offert l'environnement d'un cocon familial rare. Je pense sincèrement que l'équilibre de vie qu'ils m'ont permis d'obtenir m'aide à réussir et à avancer tous les jours.

C'est maintenant le moment de saluer tous les amis qui m'ont accompagné pendant cette thèse. Parmi eux, il y a les vieux potes : Vincent avec ses conneries incessantes qui me permettent de me reposer le cerveau, Ugo la force tranquille toujours prêt à sauver les demoiselles en détresse, Rémi qui arrive avec humour à faire redescendre Vincent de sa montagne de confiance en soi, Loïc le beau gosse qui a un succès terrible avec les filles sauf lorsqu'elles sont serveuses... et je n'oublierai pas leurs compagnes Aude, Maude et Aurélie que j'ai toujours plaisir à voir. Après, il y a les amis chers que j'ai rencontré pendant mes années parisiennes. Estelle avec qui je prends chaque fois un grand plaisir à discuter et qui est d'un conseil précieux pour les soirées bourgeois-bohème parisiennes (mais les bobos ont bon goût), Valentine, Céline, Rosa et tous les gens des doctoriales avec qui j'ai pu passer des soirées parfois endiablées, Nathalie et ses discussions passionnantes sur ses nombreux voyages et aventures et Géraud avec ses magnifiques soirées chants. Je tiens aussi à remercier Sara avec qui j'ai pu m'ouvrir à d'autres domaines (grâce aussi à sa doctorante Estelle), je n'oublierai pas son invitation à la soirée Charles Dickens à l'hôtel de ville de Paris. Je tiens aussi à remercier les amis de partout en France et particulièrement ceux de Toulouse : Laurence, Malika, Tatiana, Mélanie, Benoît, Nil, Matthias, Claire, Ludovic et Nathalie. J'ai découvert là-bas un groupe de gens formidables, accueillants et chaleureux que j'apprécie de revoir.

La conclusion de ces remerciements ira naturellement à Gaëlle. C'est pour sûr la personne la plus importante que j'ai rencontrée durant cette thèse. Nous avons su nous soutenir et parfois nous supporter pendant ces trois ans. Malgré cela, nous avons passé des moments merveilleux ensemble qui ont toujours supplanté les moments de galère. Nous continuons donc notre aventure à deux et la prochaine étape sera asiatique...

Contents

Context **13**

I Introduction **21**

1 Gaussian process regression **23**
1.1 Gaussian processes . . . 23
1.2 Bayesian approach . . . 24
1.2.1 Kriging equations . . . 25
1.2.2 Bayesian kriging equations . . . 29
1.3 Model Selection . . . 34
1.3.1 Bayesian estimate . . . 34
1.3.2 Maximum likelihood estimates . . . 36
1.3.3 Cross-validation estimate . . . 39
1.4 Covariance kernels . . . 43
1.4.1 Relations between Gaussian process regularities and covariance kernels . 45
1.4.2 Stationary covariance functions . . . 47
1.4.3 Non-stationary covariance kernels . . . 51
1.4.4 Eigenfunction analysis . . . 52
1.5 Two others approaches . . . 56
1.5.1 The Best Linear Unbiased Predictor . . . 56
1.5.2 Regularization in a Reproducing Kernel Hilbert Space . . . 58

2 Co-kriging models **69**
2.1 Bayesian Kriging models for vectorial functions . . . 69
2.1.1 Simple co-kriging equations . . . 70
2.1.2 Co-kriging parameter estimation . . . 72
2.1.3 Universal co-kriging equations . . . 74
2.2 Co-kriging in geostatistics . . . 75
2.2.1 Simple co-kriging . . . 75
2.2.2 Universal co-kriging . . . 76

2.3 Admissible matrix-valued covariance kernels . . . 78
2.3.1 Linear transformation of a multivariate Gaussian process . . . 78
2.3.2 Spectral analysis of a multivariate covariance structure . . . 81
2.4 Co-kriging models using function derivatives . . . 83

II Contributions in Multi-fidelity Co-kriging models 87

3 The AR(1) multi-fidelity co-kriging model 89
3.1 Introduction . . . 89
3.2 Building a surrogate model based on a hierarchy of s levels of code . . . 90
3.3 Building a model with 2 levels of code . . . 92
3.3.1 Conditional distribution of the output . . . 92
3.3.2 Bayesian estimation of the parameters with 2 levels of code . . . 93
3.4 Bayesian prediction for a code with 2 levels . . . 94
3.4.1 Prior distributions and Bayesian estimation of the parameters . . . 95
3.4.2 Predictive distributions when $\boldsymbol{\beta}_2, \rho, \sigma_1^2$ and σ_2^2 are known . . . 96
3.4.3 Bayesian prediction . . . 97
3.4.4 Discussion about the numerical evaluations of the integrals . . . 97
3.5 Academic examples . . . 98
3.6 The case of s levels of code . . . 103
3.6.1 Bayesian estimation of parameters for s levels of code . . . 104
3.6.2 Reduction of computational complexity of inverting the covariance matrix $\mathbf{V}_s$ . . . 104
3.6.3 Numerical test on the reduction of computational complexity . . . 109
3.6.4 Academic example on the complexity reduction . . . 109
3.6.5 Comparison with existing methods on an academic example . . . 112
3.7 Example : Fluidized-Bed Process . . . 114
3.7.1 Building the 3-level co-kriging . . . 115
3.7.2 3-level co-kriging prediction: predictions when code output is available . 116
3.7.3 3-level co-kriging prediction: predictions when code output is not available 119
3.7.4 Comparison with existing methods . . . 120
3.8 Conclusion . . . 121

4 Multi-fidelity co-kriging model: recursive formulation 123
4.1 Introduction . . . 123
4.2 Multi-fidelity Gaussian process regression . . . 124
4.2.1 Recursive multi-fidelity model . . . 124
4.2.2 Complexity analysis . . . 128
4.2.3 Parameter estimation . . . 128
4.3 Universal co-kriging model . . . 129
4.4 Fast cross-validation for co-kriging surrogate models . . . 131
4.5 Illustration: hydrodynamic simulator . . . 135

4.5.1 Estimation of the hyper-parameters . . . 135
4.5.2 Comparison between kriging and multi-fidelity co-kriging . . . 136
4.5.3 Nested space filling design . . . 137
4.5.4 Multi-fidelity surrogate model for the dissipation factor *eps* . . . 139
4.5.5 Multi-fidelity surrogate model for the mixture characteristic length L_c . 142
4.6 The R CRAN package MuFiCokriging . . . 145
4.6.1 Nested Experimental design sets . . . 145
4.6.2 Building a multi-fidelity co-kriging models with MuFiCokriging R package 147
4.6.3 Predictive means and variances at new points . . . 150
4.6.4 Cross validation procedures . . . 152
4.7 Conclusion . . . 152

5 Sequential design for kriging and Multi-fidelity co-kriging models 155
5.1 Kriging models and sequential designs . . . 156
5.1.1 The Kriging model . . . 156
5.1.2 LOO-CV based strategies for kriging sequential design . . . 158
5.2 Sequential design in a multi-fidelity framework . . . 161
5.2.1 Multi-fidelity co-kriging models . . . 161
5.2.2 Sequential design for multi-fidelity co-kriging models . . . 162
5.3 Applications . . . 168
5.3.1 Comparison between sequential kriging criteria . . . 168
5.3.2 Spherical tank under internal pressure example . . . 170
5.4 Conclusion . . . 175

6 Multi-fidelity sensitivity analysis 177
6.1 Introduction . . . 177
6.2 Global sensitivity analysis: the method of Sobol . . . 178
6.2.1 Sobol variance-based sensitivity analysis . . . 178
6.2.2 Monte-Carlo Based estimations of Sobol indices . . . 180
6.3 Kriging-based sensitivity analysis: a first approach . . . 182
6.3.1 Kriging-based sensitivity indices . . . 182
6.3.2 Monte-Carlo estimations for the first approach . . . 184
6.4 Kriging-based sensitivity analysis: a second approach . . . 185
6.4.1 Kriging-based Sobol index estimation . . . 185
6.4.2 Determining the minimal number of Monte-Carlo particles m . . . 186
6.4.3 Sampling with respect to the kriging predictive distribution on large data sets . . . 187
6.5 Multi-fidelity co-kriging based sensitivity analysis . . . 190
6.5.1 Extension of the method of Oakley and O'Hagan for multi-fidelity co-kriging . . . 191
6.5.2 Extension of the second approach for multi-fidelity co-kriging models . . 191
6.6 Numerical illustrations on an academic example . . . 192
6.6.1 Comparison between the different methods . . . 193

6.6.2 Model building and Monte-Carlo based estimator . . . 194
6.6.3 Sensitivity index estimates when n increases . . . 196
6.6.4 Optimal Monte-Carlo resource when n increases . . . 197
6.6.5 Coverage rate of the suggested Sobol index estimator . . . 198
6.7 Application of multi-fidelity sensitivity analysis . . . 200
6.7.1 Multi-fidelity model building . . . 201
6.7.2 Multi-fidelity sensitivity analysis . . . 203
6.8 Conclusion . . . 205

III Contributions in noisy-kriging 207

7 Asymptotic analysis of the learning curve 213
7.1 Introduction . . . 213
7.2 Gaussian process regression . . . 214
7.3 Convergence of the learning curve for Gaussian process regression . . . 216
7.4 Examples of rates of convergence for the learning curve . . . 219
7.5 Applications of the learning curve . . . 223
7.5.1 Estimation of the budget required to reach a prescribed precision . . . 224
7.5.2 Optimal resource allocation for a given budget . . . 225
7.6 Industrial Case: code MORET . . . 226
7.6.1 Data presentation . . . 227
7.6.2 Model selection . . . 228
7.6.3 Convergence of the IMSE . . . 229
7.6.4 Resource allocation . . . 230
7.7 Proof of Theorem 7.1 . . . 231
7.7.1 Proof of Theorem 7.1: the degenerate case . . . 231
7.7.2 Proof of Theorem 7.1: the lower bound for $\sigma^2(x)$ . . . 231
7.7.3 Proof of Theorem 7.1: the upper bound for $\sigma^2(x)$ . . . 232
7.8 Conclusion . . . 239

8 Asymptotic normality of a Sobol index estimator in noisy kriging framework 241
8.1 Introduction . . . 241
8.2 Gaussian process regression for stochastic simulators . . . 242
8.2.1 Gaussian process regression with a large number of observations . . . 242
8.2.2 Idealized Gaussian process regression . . . 243
8.3 Asymptotic normality of a Sobol index estimator . . . 246
8.3.1 A Sobol index estimator . . . 246
8.3.2 Theorem on the asymptotic normality of the Sobol index estimator . . . 247
8.4 Proof of Theorem 8.1 . . . 248
8.4.1 The Skorokhod representation theorem . . . 248
8.4.2 Convergences with a fixed $\tilde{\omega}_Z \in \tilde{\Omega}_Z$ . . . 250
8.4.3 Convergence in the probability space $(\Omega_Z \times \Omega_X, \sigma(\mathcal{F}_Z \times \mathcal{F}_X), \mathbb{P}_Z \otimes \mathbb{P}_X)$. 255

8.5 Examples of asymptotic normality for Sobol's index . . . 256
8.5.1 Asymptotic normality with d-tensorised Matérn-ν kernels . . . 256
8.5.2 Asymptotic normality for d-dimensional Gaussian kernels . . . 257
8.5.3 Asymptotic normality for d-dimensional Gaussian kernels with a Gaussian measure $\mu(x)$ . . . 257
8.6 Numerical illustration . . . 258
8.6.1 Exact Sobol indices . . . 258
8.6.2 Model selection . . . 259
8.6.3 Convergence of IMSE_T . . . 260
8.6.4 Confidence intervals for the Sobol index estimations . . . 260
8.7 Conclusion . . . 261

Conclusion and perspectives **263**

IV Appendix **267**

A Chapter 3 supplementary materials **269**
A.1 A Markovian property for covariance structure . . . 269
A.2 The case of ρ depending on x . . . 270
A.2.1 Building a model with s levels of code . . . 270
A.2.2 Bayesian estimation of parameters for s levels of code . . . 271
A.2.3 Some important results about the covariance matrix $\mathbf{V}_s$ . . . 272
A.2.4 Bayesian prediction for a code with 2 levels . . . 272

B Extension of the recursive formulation (Chapter 4) **273**
B.1 Multi-fidelity co-kriging models without nested experimental design sets . . . 273
B.1.1 Building multi-fidelity co-kriging models when the design sets are not nested . . . 273
B.1.2 Parameter estimation for the multi-fidelity co-kriging model when the design sets are not nested . . . 274
B.2 Fast cross validation for co-kriging multi-fidelity models without nested experimental design sets . . . 276

C Multi-fidelity Co-kriging models and noisy-kriging (Introduction of Part III) **277**

D Optimal resource allocation (Chapter 7) **281**
D.1 Proof of Proposition 7.3 . . . 281
D.2 Numerical illustrations . . . 283

Context

The general framework of the manuscript is the approximation of a real-valued function $z(x)$:

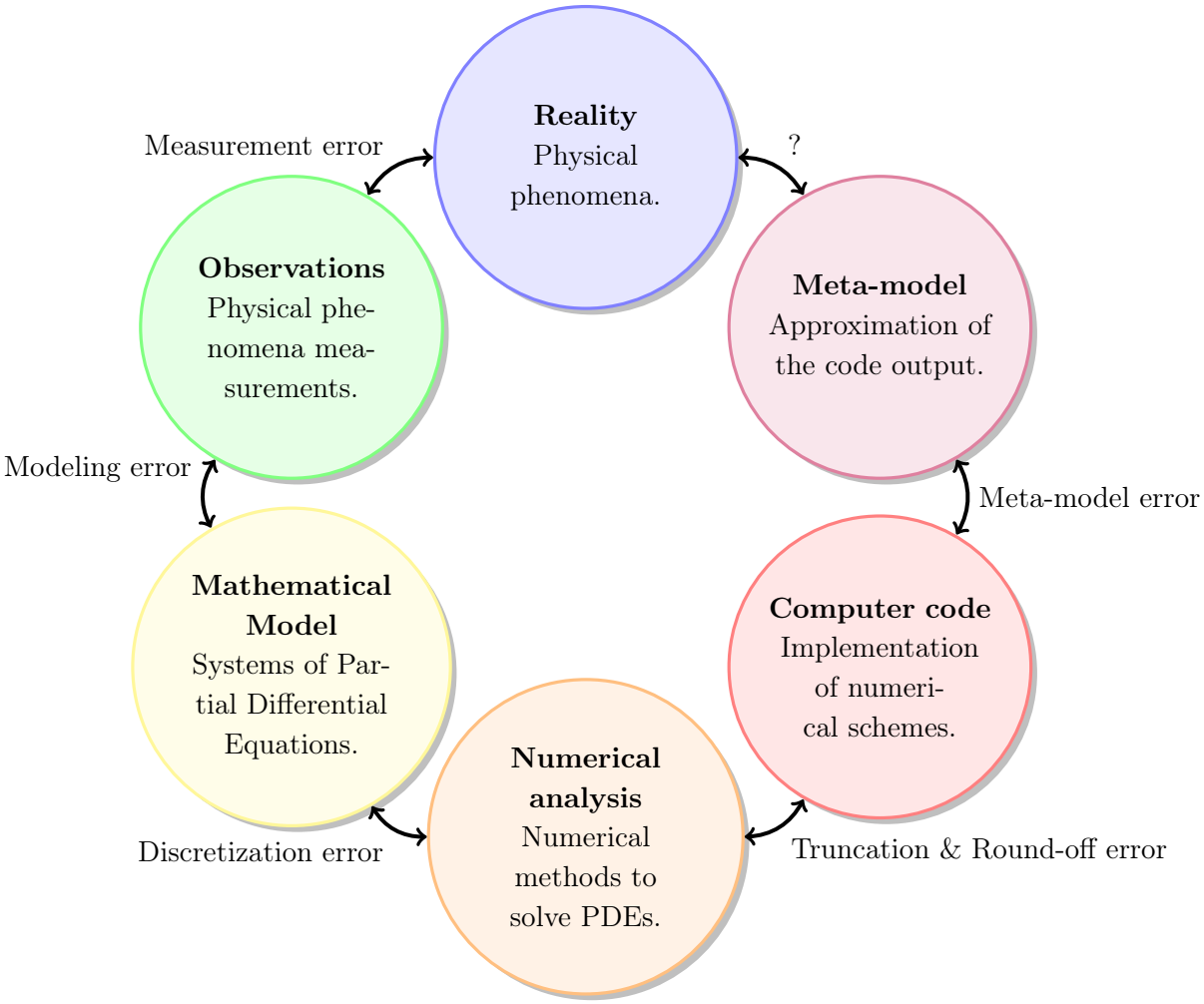
$$\begin{aligned} z : Q \subset \quad \mathbb{R}^d \quad &\to \mathbb{R} \\ x \quad &\mapsto z(x) \end{aligned}$$

from some of its values $\{z(x_1), \ldots, z(x_n)\}$, $(x_i)_{i=1,\ldots,n} \in Q$ where Q is a nonempty open set called the input parameter space. We suppose that we do not have any information about this function. Such a function is generally called in the computer experiments literature a black-box function and it represents the output of a computer code taking x as input parameters. Computer codes are widely used in science and engineering to describe physical phenomena. The term "Computer Experiments" refers to mathematical and statistical modeling using experiments performed via computer simulations. This kind of experiments is often called "experiments in silico".

To approximate the relation between the input variable x and the response variable $z(x)$, the only available information is the so-called experimental design set $\mathbf{D} = \{x_1, \ldots, x_n\}$ and the known outputs $\mathbf{z}^n = \{z(x_1), \ldots, z(x_n)\}$ of $z(x)$ at points in $\mathbf{D}$. Nevertheless, they are not sufficient to build a surrogate model for $z(x)$. Indeed, we also have to make some assumptions about the space where $z(x)$ lies.

A legitimate question that we can point out is the necessity to control the number n of observations. Indeed, a natural way to know the output $z(x)$ is to simulate the computer code with the input variable x. Nonetheless, advances in physics and computer science lead to increased complexity for the simulators. As a consequence, performing an uncertainty propagation, a sensitivity analysis or an optimization based on a complex computer code is extremely time-consuming since it requires a large number of computer simulations. Therefore, to avoid prohibitive computational costs, a fast approximation of the computer code - also called surrogate model or meta-model - is built with a restricted n.

The statistical approach is widely used for the analysis of computer experiments since there are many sources of uncertainty to consider. We summary them in the following graph.



Statistical analyses generally deal with the measurement, the modeling and the meta-modeling errors. The modeling error has two main sources of uncertainty. The first one is the mathematical approximation of the phenomena including physical simplifications and the second one is the uncertainty about the values of the physical parameters present in the model. The measurement error represents the uncertainty between the real phenomena and our observations of the phenomena. Finally, the meta-model error corresponds to the uncertainty due to the approximation of the code output. Since the meta-models are also implemented with computer codes, this part includes discretization, truncation and round-off errors.

We note that the discretization error is due to the transcription of the mathematical model - generally considering continuous functions - into a discrete model. Furthermore, the truncation error is due to the fact that computers can only deal with finite approximations and the round-off error arises because we can only represent a finite number of real numbers on a machine. We highlight that nowadays, we cannot handle all sources of uncertainty and thus the ones between the reality and the surrogate model remain unknown.

In this manuscript, we focus on the measurement and on the meta-modeling errors. In particular, we consider the Gaussian process regression - also called kriging model - as surrogate

model. It is a useful and very popular tool to approximate an objective function given some of its observations (see e.g [Sacks et al., 1989b], [Sacks et al., 1989a], [Currin et al., 1991], [Morris et al., 1993], [Laslett, 1994], [Koehler and Owen, 1996], [Schonlau, 1998], [Stein, 1999], [Kennedy and O'Hagan, 2001], [Santner et al., 2003], [Fang et al., 2006], [O'Hagan, 2006], [Conti and O'Hagan, 2010], [Bect et al., 2012] and [Gramacy and Lian, 2012]). It corresponds to a particular class of surrogate models which makes the assumption that the response of the complex code is a realization of a Gaussian process. A strength of this approach is that it provides a basis for statistical inference through the Gaussian assumption. It has originally been used in geostatistics by [Krige, 1951] to interpolate a random field at unobserved locations (see [Matheron, 1963], [Matheron, 1969], [Chilès and Delfiner, 1999], [Wackernagel, 2003], [Berger et al., 2001] and [Gneiting et al., 2010]) and it has been developed in many areas such as environmental and atmospheric sciences. It was then proposed in the field of computer experiments by [Sacks et al., 1989b]. During the last decades, this method has become widely used and investigated.

We introduce the Gaussian process regression in Part I. This chapter is inspired by the books of [Stein, 1999], [Santner et al., 2003] and [Rasmussen and Williams, 2006], the reader is referred to them for more detail about kriging model. In this part, we introduce in Chapter 1 the univariate kriging model, i.e. when the output of the objective function is a scalar. In this chapter, we present different approaches for the kriging model: from the Bayesian one in Section 1.2 to the original one introduced by [Krige, 1951] in Section 1.5. Furthermore, throughout Chapter 1 we present some methods to implement and use in practical way the kriging model. In particular, in Section 1.3 we present classical mathematical tools and recent advances about model selection in a Gaussian process regression context. Moreover, in Section 1.4 we discuss about covariance kernels which are an important element of kriging model. Finally, we give in Chapter 1 some theoretical insights about Gaussian process regression. More specifically, we deal with spectral representation of a Gaussian process in Section 1.4 and we propose a short introduction to reproducing kernel Hilbert spaces in Section 1.5.

Then, in Chapter 2, we present kriging models in a multivariate framework. The corresponding method is called co-kriging and is used when the output of the objective function is a vector with correlated components. First in Section 2.1, we extend the Bayesian kriging equations presented in Section 1.2 for the co-kriging models. Second, we present in Section 2.2 the original co-kriging model introduced in the geostatistical literature. We will see that the Bayesian and the geostatistical approaches are equivalent. Then, in Section 2.3 we discuss about matrix-valued covariance kernels which are an important ingredient of the method with a non-trivial definition. Finally, in Section 2.4, we give an example of a co-kriging model widely used in computer experiments which allows for taking into account the derivatives into the model building.

Sometimes low-fidelity versions of the computer code are available. They may be less accurate but they are computationally cheap. A question of interest is how to build a surrogate model using data from simulations of multiple levels of fidelity. The objective is hence to build a multi-fidelity surrogate model which is able to use the information obtained from the fast versions of the code. Such models have been presented in the literature [Craig et al.,

1998], [Kennedy and O'Hagan, 2000], [Higdon et al., 2004], [Forrester et al., 2007], [Qian and Wu, 2008] and [Cumming and Goldstein, 2009]. We propose in Part II some derivations and extensions to the model proposed by [Kennedy and O'Hagan, 2000] and investigated by [Higdon et al., 2004], [Forrester et al., 2007] and [Qian and Wu, 2008]. First of all, we present this model in Chapter 3 and we deal with some key issues that make difficult to use the suggested model for practical applications. In particular we propose in sections 3.3 and 3.6 an original approach for the parameter estimations which is effective even when the number of code levels is large. Furthermore, we propose in Section 3.4 a Bayesian formulation of the model which allows to consider prior information in the parameter estimations and integrates all the uncertainty due to the estimation of the parameters. We also proposed some tricks to reduce the computational complexity of the model. Comparisons have been performed between our model and the ones of [Kennedy and O'Hagan, 2000] and [Qian and Wu, 2008] on a academic example in Section 3.5 and on an application in Section 3.7. They show that our approach improves the former ones both in terms of prediction accuracy and computational costs.

Then, in Chapter 4, we suggest another approach to build multi-fidelity co-kriging models based on a recursive formulation. With this original formulation presented in Section 4.2, we obtain the same performance in terms of prediction accuracy and computational costs as the model proposed in Chapter 3 when we use the suggested improvements. However, it allows for extending classical results of kriging to the considered co-kriging model. In particular, we give Universal co-kriging equations in Section 4.3 which integrate the uncertainty due to the estimation of some parameters. Moreover, in Section 4.4 we give computational shortcuts to compute the cross-validation procedure for the suggested multi-fidelity co-kriging model. The efficiency of the recursive formulation of the model is emphasized on an application in Section 4.5. We also implement this model in a R CRAN package named "MuFiCokriging" (`http://cran.r-project.org/web/packages/MuFiCokriging`) and present it in Section 4.6. Another strength of the approach presented in Chapter 4 is that it allows for obtaining the contribution of each code level into the total model variance. We use this important property in Chapter 5 to propose sequential design strategies in a multi-fidelity framework.

In Chapter 5, we first propose original kriging-based sequential design strategies in Section 5.1. The novelty is that they take into account the model prediction capability into the sequential procedure and not only the estimated model variance. Then, we give in Section 5.2 a method to extend the kriging-based sequential design strategies to the multi-fidelity co-kriging model. We note that, in a multi-fidelity framework, the search for the best locations where to run the code is not the only point of interest. Indeed, once the best locations are determined, we also have to decide which code level is worth being run. In particular, the presented extensions take into account the computational time ratios between code versions and the part of each code into the model's variance. The performance of the given sequential strategies for kriging and co-kriging models are illustrated on applications in Section 5.3.

In many cases, computer codes have a large number d of input parameters. Global sensitivity analysis aims to identify those which have the most important impact on the output. A popular tool to perform global sensitivity analysis is the variance-based method coming

from the Hoeffding-Sobol decomposition [Hoeffding, 1948] and named as the Sobol method [Sobol, 1993]. Nevertheless, this method requires an important number of simulations. The codes being often extremely time-consuming, we use a surrogate model to handle with it. We present in Chapter 6 an original kriging-based global sensitivity analysis. In particular, it fixes important flaws present in the pioneering article of [Oakley and O'Hagan, 2004]. We present the principle of their method in Section 6.3 and give some improvements for it. Then, in Section 6.4 we suggest our original approach to perform kriging-based sensitivity analysis. Finally, the extensions of the two presented methods for the multi-fidelity co-kriging models are presented in Section 6.5.

We emphasize that in Chapter 6 Subsections 6.4.3 and 6.5.2 we propose two methods to generate samples with respect to the kriging and co-kriging predictive distributions on large data sets. In particular, we avoid numerical issues such that ill-conditioned matrices and high computational costs.

For many realistic cases, we do not have direct access to the function to be approximated but only to noisy versions of it. For example, if the objective function is the result of an experiment, the available responses can be tainted by measurement noise. Another example is Monte-Carlo based simulators - also called stochastic simulators - which use Monte-Carlo or Monte-Carlo Markov Chain methods to solve a system of partial differential equations through its probabilistic interpretation. Gaussian process regression can be easily adapted to the case of noisy observations. We deal with the framework of stochastic simulators in Part III.

First, we introduce at the beginning of Part III, the context of stochastic simulators. The important point is that in this framework the observation noise variance is inversely proportional to the number of particles used to the Monte-Carlo schemes. Furthermore, the amount of particles also controls the computational cost of the simulator. Therefore, in that framework, we have an explicit relation between the accuracy of an output and its computational cost. Another particularity is that an infinite number of code levels of increasing accuracy can be obtained. In particular, we consider the case of partially converged simulations, i.e. an accurate code output corresponds to a coarse one after continuing the Monte-Carlo convergence. We show in the introduction of Part III that using a multi-fidelity co-kriging model in such a context is equivalent to use a noisy-kriging considering uniquely the most accurate simulations.

Then, Chapter 7 deals with the learning curve describing the generalization error of the Gaussian process regression as a function of the training size. The main result of this chapter is the proof of a theorem giving the generalization error for a large class of correlation kernels and for any dimension when the number of observations is large. The theorem is presented in Section 7.3 and its proof is given in Section 7.7. The presented proof generalizes previous ones that were limited to special kernels or to small dimensions (one or two). From this result, we deduce in Section 7.4 the asymptotic behavior of the generalization error when the observation error is small. This is of interest since it provides a powerful tool for decision support. Indeed, from an initial experimental design set, it allows for predicting the additional computational budget necessary to reach a given desired accuracy. This result is applied successfully in Section 7.6 to a nuclear safety problem. Moreover, in Section 7.5 we deal with

the optimal resource allocation. If we consider as fixed the number of particles for the Monte-Carlo procedures and the number of simulations, then a question of interest is to find the particle repartition on the simulations which minimizes the model uncertainty. We provide a proposition giving an optimal allocation under restricted conditions. Furthermore, we observe in Appendix D that this allocation remains efficient in more general cases.

Finally, we address in Chapter 8 the problem of global sensitivity analysis for stochastic simulators. As seen previously, variance-based sensitivity methods require a large number of simulations. As the computer codes are time-consuming they are generally substituted by a surrogate model. Therefore, there are two sources of uncertainty in such analysis. The first one corresponds to the meta-model error (approximation error) and the second one corresponds to the error on the sensitivity index estimates of the meta-model (estimation error). To perform such analysis, we suggest a particular surrogate model in Section 8.2 which corresponds to a Gaussian process regression build from lot of simulations but with a large uncertainty. The main result of this chapter is a theorem presented in Section 8.3 which gives sufficient conditions to obtain the asymptotic normality for the suggested index estimators. The proof of this theorem is given in Subsection 8.4. From the theorem, we can derived asymptotic confidence intervals taking into account the uncertainty of both the meta-model approximation error and the index estimation error. We illustrate on an example the efficiency of our approach.

Notations

a.c.	absolutely continuous,
a.s.	almost surely,
a.e.	almost every,
BLUP	Best Linear Unbiased Predictor,
CV	Cross-Validation,
IMSE	Integrated Mean Squared Error,
LOO	Leave-One-Out,
MCMC	Monte-Carlo Markov Chain,
MLE	Maximum Likelihood Estimate,
MSE	Mean Squared Error,
RKHS	Reproducing Kernel Hilbert Space,
$z(x)$	Objective function to be approximated,
x	input parameter in a subset Q of $\mathbb{R}^d$,
Q	nonempty open subset of $\mathbb{R}^d$ representing the input parameter space,
d	number of dimensions of the input parameter space,
n	number of observations,
$\mathbf{z}^n$	the vector of the observed values of $z(x)$ in $\mathbf{D}$.
$\mathbf{D}$	the $n \times d$ experimental design set, the n lines represent the observation points in Q,
GP	Gaussian process,
$\mathcal{N}$	Multivariate or univariate Gaussian distribution,
$Z(x)$	Gaussian process of mean $m(x)$ and covariance structure $k(x, \tilde{x})$,
$\mathbf{Z}^n$	the Gaussian vector $Z(\mathbf{D})$,
$k(x, \tilde{x})$	covariance function or continuous positive definite kernel,
$\mathbf{k}(x)$	covariance vector between x and $\mathbf{D}$ with respect to $k(x, \tilde{x})$,
$\mathbf{K}$	covariance matrix of $\mathbf{D}$ with respect to $k(x, \tilde{x})$,
$\mathbf{V}(x, \tilde{x})$	matrix valued covariance kernel,
$r(x, \tilde{x})$	correlation kernel,
$\mathbf{r}(x)$	correlation vector between x and $\mathbf{D}$ with respect to $r(x, \tilde{x})$,
$\mathbf{R}$	correlation matrix of $\mathbf{D}$ with respect to $r(x, \tilde{x})$,
θ	hyper-parameters of the covariance or correlation structure,

σ^2	variance parameter,
$\mathbf{f}(x)$	vector of regressors of size p,
$\boldsymbol{\beta}$	regression parameter,
$\mathbf{F}$	design matrix corresponding to the values of $\mathbf{f}'(\mathbf{D})$,
Ω	sample space,
$\mathcal{F}$	a σ-algebra on Ω,
$\mathcal{B}$	the Borelian σ-algebra,
$\mathbb{P}$	a probability on $\mathcal{F}$,
μ	a probability measure on Q,
$p(x)$	probability density function,
$\mathbb{E}$	expectation,
cov	covariance,
$\stackrel{\mathcal{L}}{=}$	equality in distribution,
$:=$	an equality which acts as a definition,
$\mathbf{1}$	indicator function,
$\mathbf{I}$	the identity matrix,
$'$	matrix or vector transpose,
tr	trace of a matrix,
$\langle . \rangle$	scalar product,
$\|\|.\|\|$	euclidean norm,
$\delta_{x=\tilde{x}}$	Kronecker symbol,
$\mathrm{diag}\,(x)$	diagonal matrix with diagonal vector x,
$*$	convolution operator,
$\mathcal{H}$	a Hilbert space of real functions,
L^2_μ	space of square-integrable functions with respect to the measure μ.

Part I

Introduction

Chapter 1

An introduction to Gaussian process regression

Let us consider that we are interested in approximating an objective function $z(x) \in \mathbb{R}$ with $x \in Q \subset \mathbb{R}^d$ from few of its observations and where Q is a nonempty open set. In our framework, $z(x)$ represents the output of a code and x represents its input. Furthermore, we denote by $\mathbf{D} = \{x_1, \ldots, x_n\}$ with $x_i \in Q$ the experimental design set and $\mathbf{z}^n = z(\mathbf{D})$ the values of $z(x)$ at points in $\mathbf{D}$ - $\mathbf{z}^n$ is called the vector of observations. Gaussian process regression - also called kriging model - is a very popular tool to perform such approximation. Throughout, the manuscript, we will equivalently use the term kriging model or Gaussian process regression.

We present in this chapter the Gaussian process regression principle through different approaches. First, we introduce it with a Bayesian paradigm in Section 1.2. Then, we give two other approaches: the geostatistical one with the Best Linear Unbiased Predictor (BLUP) (Subsection 1.5.1) and the regularization one with the representer theorem in a Reproducing Kernel Hilbert Space (RKHS) (Subsection 1.5.2).

We also deal with two important points controlling the efficiency of the Gaussian process regression. The first one is about the model selection (Section 1.3) in which we present different ways to estimate the model parameters. The second one is the choice of the covariance kernel of the Gaussian process used in the model (Section 1.4). Over all, let us introduce in the next Section 1.1 the so-called Gaussian processes.

1.1 Gaussian processes: a short introduction

Let us consider a probability space $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$, a measurable space $(S, \mathcal{B}(S))$ and T an arbitrary set. A stochastic process $Z(x)$, $x \in T$, is a collection of random variables defined on $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$, indexed by T and with values in S. $Z(x)$ is Gaussian if and only if for any finite collection $C \subset T$, $Z(C)$ has a joint Gaussian distribution. In our work, we always have $S = \mathbb{R}$ and $T = Q \subset \mathbb{R}^d$ with d an integer representing the dimension of the input parameter space and Q a nonempty open set. A Gaussian process is completely specified by

its mean function $m(x) = \mathbb{E}_Z[Z(x)]$ and its covariance function $k(x, \tilde{x}) = \text{cov}_Z(Z(x), Z(\tilde{x})) = \mathbb{E}_Z[(Z(x) - \mathbb{E}_Z[Z(x)])(Z(\tilde{x}) - \mathbb{E}_Z[Z(\tilde{x})])]$.

The mean function $m(x)$ of a Gaussian process represents its trend. In a Gaussian process regression framework, we usually choose a mean function of the form $m(x) = \mathbf{f}'(x)\boldsymbol{\beta}$, with $\mathbf{f}'(x) = (f_1(x), \ldots, f_p(x))$ a vector of regressors generally including a constant function and $\boldsymbol{\beta}$ a $p \times 1$ vector of regression parameters.

The covariance function $k(x, \tilde{x})$ is a positive definite kernel, i.e. for all $(a_i)_{i=1,\ldots,N} \in \mathbb{R}$, $N \in \mathbb{N}^*$ and distinct $(x_i)_{i=1,\ldots,N} \in T$, it satisfies the following property:

$$\sum_{i,j=1}^{N} a_i a_j k(x_i, x_j) \geq 0$$

and $\sum_{i,j=1}^{N} a_i a_j k(x_i, x_j) = 0$ if and only if $a_i = 0$ for all $i = 1, \ldots, N$. Furthermore, we always consider in the manuscript that $k(x, \tilde{x})$ is continuous and $\sup_{x \in T} k(x, x) < \infty$. The covariance kernel describes the dependence structure of the Gaussian process $Z(x)$. In our framework, we often consider kernels of the form $k(x, \tilde{x}) = \sigma^2 r(x, \tilde{x}; \boldsymbol{\theta})$ where $r(x, \tilde{x}; \boldsymbol{\theta})$ is a correlation kernel parametrized with the vector $\boldsymbol{\theta}$ and σ^2 is the variance parameter. Furthermore, we generally consider a stationary kernel, i.e. $k(x, \tilde{x})$ is a function of $x - \tilde{x}$. Nonetheless, for some derivations - like in Chapter 7 - we consider any continuous positive definite kernel $k(x, \tilde{x})$ such that $\sup_{x \in T} k(x, x) < \infty$. The covariance kernel is certainly the most important ingredient of a Gaussian process regression. Indeed, it controls the smoothness of the Gaussian process (see Section 1.5) and thus the regularity of the approximation of the objective function $z(x)$.

A first example of covariance kernel. A popular covariance kernel is the isotropic squared exponential one defined as

$$k(x, \tilde{x}) = \sigma^2 \exp\left(-\frac{1}{2\theta^2}||x - \tilde{x}||^2\right), \tag{1.1}$$

where $||.||$ stands for the euclidean norm. It is parametrized by the hyper-parameter θ which is called the characteristic length-scale or correlation length. Roughly speaking, θ represents the distance for which the observations are strongly dependent. In general, the parameters of the covariance function are referred to hyper-parameters to highlight that they are parameters of a non-parametric model. We illustrate in Figure 1.1 some realizations of Gaussian processes with a squared exponential covariance kernel. We vary the formula of the mean and the value of the variance parameter σ^2 and the hyper-parameter θ. We observe in Figure 1.1 that the variance parameter σ^2 controls the range of variation of the Gaussian process, the hyper-parameter θ controls the oscillation frequencies and the mean controls the trend of the Gaussian process.

1.2 Kriging models : a Bayesian approach

In a kriging framework, we consider that the code $z(x)$ is a realization of a Gaussian process $Z(x)$. Usually, we consider a Gaussian process with mean of the form $m(x) = \mathbf{f}'(x)\boldsymbol{\beta}$, with

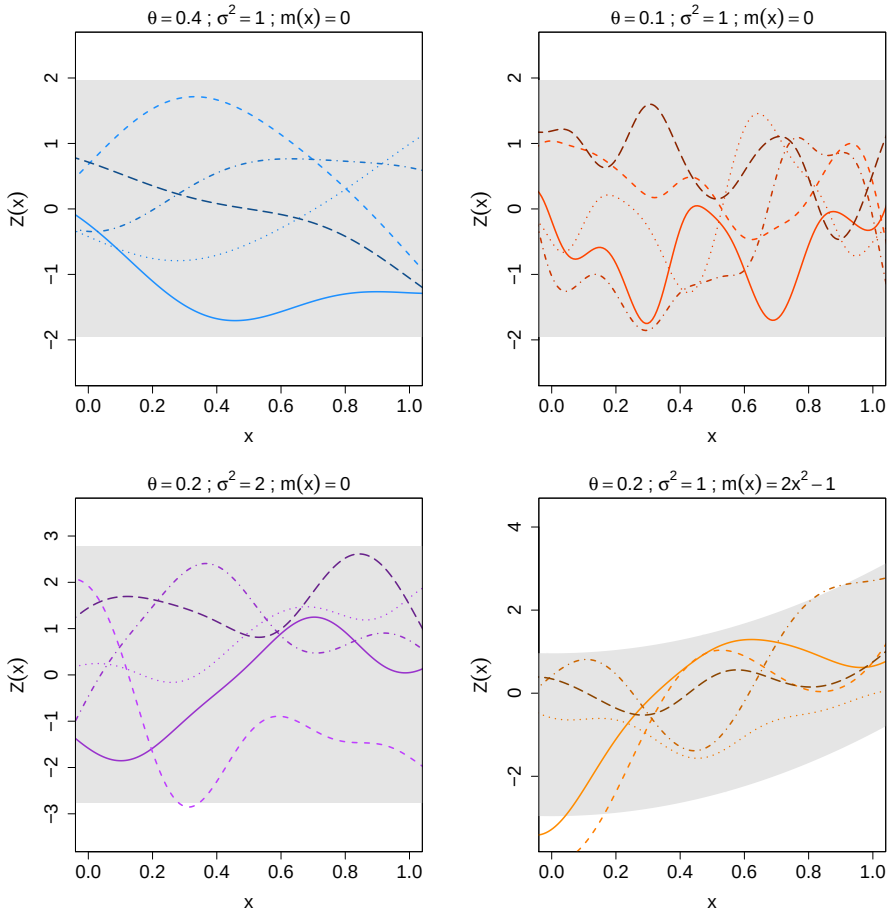


Figure 1.1: Realizations of Gaussian processes with squared exponential kernel with different parameter values and trend formulas. The shade area represents the point-wise mean plus and minus twice the standard deviation. It corresponds to 95% confidence intervals.

$\mathbf{f}'(x) = (f_1(x), \ldots, f_p(x))$ and with covariance function $k(x, \tilde{x}) = \sigma^2 r(x, \tilde{x}; \boldsymbol{\theta})$. The mean of the Gaussian process models the trend of the observations with respect to the input parameters and the covariance structure models the dependence between the different values of the objective function.

1.2.1 Kriging equations

We develop in this subsection the so-called kriging equations. The kriging mean provides the surrogate model that we use to approximate the objective function $z(x)$ and the kriging variance represents the uncertainty of the model. We derive two types of kriging models. In the first one, we consider that the observations are noisy-free. In the second one, we consider that the observations are tainted by a white noise.

The noisy-free case

We consider the random vector $\mathbf{Z}^n := Z(\mathbf{D})$ which is Gaussian since $Z(x)$ is a Gaussian process. We consider the problem of predicting the random variable $Z(x)$ for any $x \in Q$. Intuitively, we want to use the information contains in $\mathbf{Z}^n$ to predict $Z(x)$ and thus we consider the joint distribution of $Z(x)$ and $\mathbf{Z}^n$ given by:

$$\begin{pmatrix} Z(x) \\ \mathbf{Z}^n \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{f}'(x)\boldsymbol{\beta} \\ \mathbf{F}\boldsymbol{\beta} \end{pmatrix}, \sigma^2 \begin{pmatrix} 1 & \mathbf{r}'(x) \\ \mathbf{r}(x) & \mathbf{R} \end{pmatrix} \right), \tag{1.2}$$

where $'$ stands for transpose, $\mathbf{F} := \mathbf{f}'(\mathbf{D})$ is the design matrix, $\mathbf{r}'(x) = [r(x, x_i; \boldsymbol{\theta})]_{i=1,\dots,n}$ is the correlation vector between $Z(x)$ and the observations at points $(x_i)_{i=1,\dots,n}$ in $\mathbf{D}$ and $\mathbf{R} = [r(x_i, x_j; \boldsymbol{\theta})]_{i,j=1,\dots,n}$ is the correlation matrix between the observations at points in $\mathbf{D}$.

Then, the predictive distribution is defined by $[Z(x)|\mathbf{Z}^n = \mathbf{z}^n, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}]$. The notation $[A|B]$ stands for the distribution of A conditionally to B. Conditionally to $\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}$ the random vector $(Z(x), \mathbf{Z}^n)$ is Gaussian. Therefore, conditionally to these parameters, the conditional distribution $[Z(x)|\mathbf{Z}^n = \mathbf{z}^n, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}]$ is a Gaussian $\mathcal{N}\left(\hat{z}(x), s^2(x)\right)$ with :

$$\hat{z}(x) = \mathbf{f}'(x)\boldsymbol{\beta} + \mathbf{r}'(x)\mathbf{R}^{-1}\left(\mathbf{z}^n - \mathbf{F}\boldsymbol{\beta}\right) \tag{1.3}$$

and

$$s^2(x) = \sigma^2\left(1 - \mathbf{r}'(x)\mathbf{R}^{-1}\mathbf{r}(x)\right). \tag{1.4}$$

Equations (1.3) and (1.4) correspond to the Simple Kriging equations, i.e. when all parameters are considered as known. The kriging mean $\hat{z}(x)$ is the surrogate model that we use to approximate the objective function $z(x)$ and the kriging variance $s^2(x)$ represents the model mean squared error.

We illustrate in Figure 1.2 some realizations of a conditional Gaussian process distribution. We see in Figure 1.2 that the kriging mean interpolates the observations. This is an important property of kriging equations. Furthermore, we see that the kriging variance equals zero at points of the experimental design set. It means that we consider that the model error is null at these points. It is natural since the model is interpolating.

Then, we see in Equation (1.3) that the kriging mean does not depend on the variance parameter σ^2. In fact, this parameter - representing the range of variation of the function $z(x)$ - has just an impact on the kriging variance (1.4). Furthermore, we see that the kriging variance does not depend on the observations $\mathbf{z}^n$. This property can be useful to elaborate strategies to reduce the model uncertainty. Indeed, we can evaluate the reduction of uncertainty after adding some points into the experimental design set without simulating new observations. Nevertheless, this point is also a big flaw of the method. Since the Gaussian assumption cannot be verified, the kriging variance can poorly represent the model error. In fact, kriging variance is more a measure of the distance between the point x and the points in $\mathbf{D}$ than a measure of the prediction error at point x. Therefore, conception based uniquely on kriging

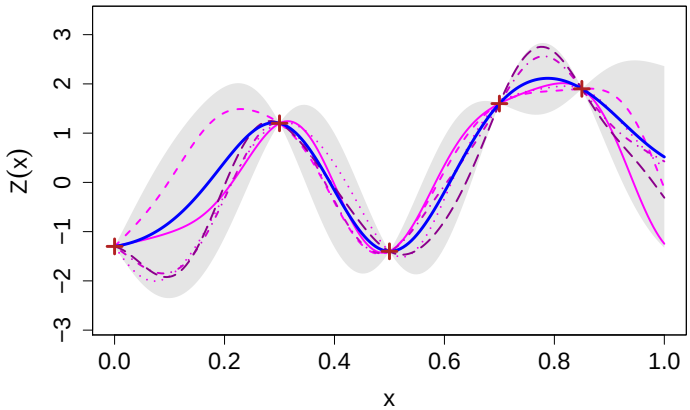


Figure 1.2: Realizations of a conditional Gaussian process distribution with squared exponential kernel, variance parameter $\sigma^2 = 1$, hyper-parameter $\theta = 0.1$, regressors function $\mathbf{f}'(x) = (1, x)$ and trend parameter $\boldsymbol{\beta} = (-1, 1)$. The thin purple lines represent the realizations, the crosses represent the observations, the thick blue line represents the kriging mean $\hat{z}(x)$ and the shade area represents the mean $\hat{z}(x)$ plus and minus twice the standard deviation $s(x)$. It corresponds to 95% confidence intervals.

variance could be inappropriate. We present in Chapter 5 an example of method which uses the model prediction capability to adjust the kriging variance.

Furthermore, if we denote by $Y(x) = Z(x) - \mathbf{f}'(x)\boldsymbol{\beta}$, $\mathbf{y}^n = \mathbf{y}^n - \mathbf{F}\boldsymbol{\beta}$ and $\hat{y}(x) = \hat{z}(x) - \mathbf{f}'(x)\boldsymbol{\beta}$, then $Y(x)$ is a Gaussian process with mean zero and the same covariance structure as $Z(x)$. Then we can rewrite Equation (1.3) with the two following forms:

$$\hat{y}(x) = \sum_{i=1}^{n} \alpha_i \mathbf{y}_i^n, \tag{1.5}$$

with $\alpha_i = [\mathbf{r}'(x)\mathbf{R}^{-1}]_i$, $i = 1, \ldots, n$ and

$$\hat{y}(x) = \sum_{i=1}^{n} \gamma_i k(x, x_i), \tag{1.6}$$

with $\gamma_i = \left[\mathbf{R}^{-1}\left(\mathbf{z}^n - \mathbf{F}\boldsymbol{\beta}\right)\right]_i$, $i = 1, \ldots, n$. These two equations introduce the two other approaches of the Gaussian process regression. In Equation (1.5) we notice that the predictor $\hat{y}(x)$ can be viewed as a linear predictor with respect to the observed values $\mathbf{y}^n$. This approach which refers to the Best Linear Unbiased Predictor (BLUP) is presented in Subsection 1.5.1. Then, in Equation (1.6), we see that the predictor can be written as a linear combination of the kernel $k(x, \tilde{x})$ centered onto the points of the experimental design set. This form - corresponding to the solution of a specific regularization problem in a Reproducing Kernel Hilbert Space (RKHS) - is presented in Subsection 1.5.2.

The noisy case

For many cases, we do not have direct access to the function to be approximated but only to a noisy version of it. For example, if the objective function is the result of an experiment, the observations are typically tainted by measurement noise. Let us suppose that we want to approximate an objective function $x \in Q \to f(x) \in \mathbb{R}$ from noisy observations at points $(x_i)_{i=1,\dots,n}$ in $\mathbf{D}$. Throughout the manuscript $f(x)$ designs a function for which we have noisy observations (see Part III). We assume an independent Gaussian observation noise with zero mean and variance $\sigma_\varepsilon^2(x)$. In the computer experiments literature, it is referred as the "nugget effect". Therefore, we have n observations of the form $z_i = f(x_i) + \sigma_\varepsilon(x_i)\varepsilon_i$ where $(\varepsilon_i)_{i=1,\dots,n}$ are independent and identically distributed with respect to a Gaussian distribution with zero mean and variance one. As in the noisy-free case, we assume that $f(x)$ is a realization of a Gaussian process $Z(x)$ of mean $m(x) = \mathbf{f}'(x)\boldsymbol{\beta}$ and covariance structure $k(x, \tilde{x}) = \sigma^2 r(x, \tilde{x}; \boldsymbol{\theta})$. Denoting by $\mathbf{Z}^n = Z(\mathbf{D}) + \boldsymbol{\varepsilon}^n$, with $\boldsymbol{\varepsilon}^n := [\sigma_\varepsilon(x_i)\varepsilon_i]_{i=1,\dots,n}$, we have the following covariances:

$$\operatorname{cov}\left(Z(x), \mathbf{Z}^n\right) = \mathbf{k}'(x),$$

with $\mathbf{k}'(x) = [k(x, x_i)]_{i=1,\dots,n}$ and

$$\operatorname{cov}\left(\mathbf{Z}^n, \mathbf{Z}^n\right) = \mathbf{K} + \boldsymbol{\Delta},$$

where $\mathbf{K} = [k(x_i, x_j)]_{i,j=1,\dots,n}$, $\boldsymbol{\Delta} = [\sigma_\varepsilon^2(x_i)\delta_{ij}]_{i,j=1,\dots,n}$ and δ_{ij} is the Kronecker delta which is one if $i = j$ and zero otherwise. Therefore, we have the following joint distribution:

$$\begin{pmatrix} Z(x) \\ \mathbf{Z}^n \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{f}'(x)\boldsymbol{\beta} \\ \mathbf{F}\boldsymbol{\beta} \end{pmatrix}, \begin{pmatrix} k(x,x) & \mathbf{k}'(x) \\ \mathbf{k}(x) & \mathbf{K} + \boldsymbol{\Delta} \end{pmatrix}\right). \tag{1.7}$$

Then, the predictive distribution $[Z(x)|\mathbf{Z}^n = \mathbf{z}^n, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}, \boldsymbol{\Delta}]$ is still a Gaussian distribution $\mathcal{N}\left(\hat{z}(x), s^2(x)\right)$ with :

$$\hat{z}(x) = \mathbf{f}'(x)\boldsymbol{\beta} + \mathbf{k}'(x)(\mathbf{K} + \boldsymbol{\Delta})^{-1}\left(\mathbf{z}^n - \mathbf{F}\boldsymbol{\beta}\right) \tag{1.8}$$

and

$$s^2(x) = k(x,x) - \mathbf{k}'(x)(\mathbf{K} + \boldsymbol{\Delta})^{-1}\mathbf{k}(x). \tag{1.9}$$

We note that in the noisy case, the predictor (1.8) can also be viewed as a linear predictor with respect to the observations or as a regularization problem solution in a RKHS. Furthermore, the mean $\hat{z}(x)$ of the predictive distribution no longer interpolates the observations $\mathbf{z}^n$ and the variance $s^2(x)$ is not zero at points in the experimental design set. This properties are natural since there is no sense to interpolate the observations if they are tainted by noise. Moreover, at a point $x_i \in \mathbf{D}$, the predictive variance cannot equal zero since it takes into account the observation noise variance. We present in Figure 1.3 an example of kriging model in a noisy framework.

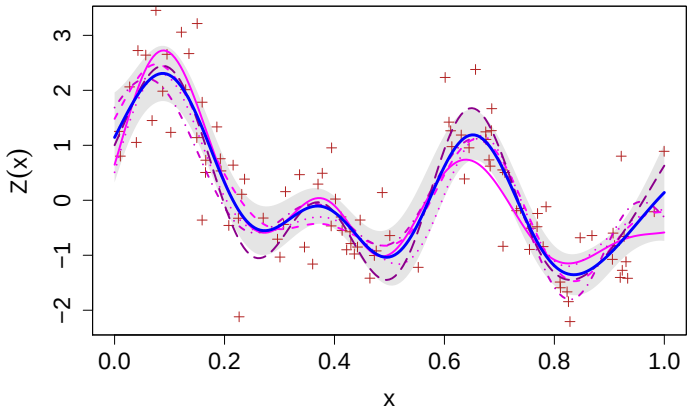


Figure 1.3: Realizations of a conditional Gaussian process distribution with noisy observations and a squared exponential kernel. The variance parameter equals $\sigma^2 = 2$, the hyper-parameter equals $\theta = 0.1$ and the mean $m(x)$ is null. The thin purple lines represent the realizations, the crosses represent the observations, the thick blue line represents the kriging mean $\hat{z}(x)$ and the shade area represents the mean $\hat{z}(x)$ plus and minus twice the standard deviation $s(x)$. Furthermore, the observation noise variance is $\sigma_\varepsilon^2(x) = (2 + \sin(4\pi x))/4$.

1.2.2 Bayesian kriging equations

We discuss in this section about the Bayesian approach in Gaussian process regression. In a Bayesian paradigm the parameters and hyper-parameters of the model are considered as unknown and are modeled by random variables. The first objective is to infer from the observations about the parameters and hyper-parameters. Then the aim is to provide a predictive distribution integrating the posterior distributions of the parameters and hence taking into account their uncertainty.

The Bayesian approach has two important strengths. First, it allows for taking into account all the sources of uncertainty coming from the parameter estimations into the predictive distribution. Second, it allows for taking into account expert knowledges - through a prior distribution - into the parameter estimations. For more detail about the Bayesian methods, the reader is referred to the book of [Robert, 2007].

In counterpart, they are two important flaws in a Bayesian modeling. The first one - perhaps the most important - is that the posterior distributions are sensitive to the prior distributions given by experts. This flaw is even more important that we often restrict the choice of the prior distributions in order to obtain closed form formulas for the posterior predictive distributions. Such prior distributions are called conjugate distributions. The second one is that for general prior distributions, there is no closed form expressions for the predictive distribution. It is then necessary to perform various numerical integrations which are usually done with Monte-Carlo Markov Chain (MCMC). These methods could be computationally expensive and not be suitable for practical applications - this explains the use of conjugate priors. For more detail about MCMC schemes, the reader is referred to the book of [Robert and Casella, 2004].

The Jeffreys law

A question of interest in a Bayesian approach is to describe prior distributions which reflect the fact that there is no prior knowledge about the parameters. These distributions are called non-informative. For the non-informative case, we use the improper distributions corresponding to the "Jeffreys priors" [Jeffreys, 1961]. These laws are based on the Fisher information matrix [Fisher, 1956] which is defined as the expected value of the observed information.

Let us denote by $\mathbf{z}^n$ a sample of a random variable Z and $f(\mathbf{z}^n|\boldsymbol{\psi})$ the likelihood of a parameter $\boldsymbol{\psi} = (\boldsymbol{\psi}_i)_{i=1,\dots,d}$ with respect to $\mathbf{z}^n$. The observed information matrix is defined as:

$$\mathcal{I}(\boldsymbol{\psi}; \mathbf{z}^n) = \left[-\frac{\partial^2}{\partial \boldsymbol{\psi}_i \partial \boldsymbol{\psi}_j} \log(f(\mathbf{z}^n|\boldsymbol{\psi})) \right]_{i,j=1,\dots,d}.$$

Then, the Fisher information matrix is given by:

$$\mathbb{I}(\boldsymbol{\psi}) = \left[-\mathbb{E}\left[\frac{\partial^2}{\partial \boldsymbol{\psi}_i \partial \boldsymbol{\psi}_j} \log(f(\mathbf{z}^n|\boldsymbol{\psi})) \right] \right]_{i,j=1,\dots,d}.$$

where the expectation is taken with respect to the distribution of $\mathbf{z}^n$ with the parameter $\boldsymbol{\psi}$ The "Jeffreys prior" distribution is given by the density function:

$$p(\boldsymbol{\psi}) \propto \left[\det\left(\mathbb{I}(\boldsymbol{\psi})\right)\right]^{1/2}. \tag{1.10}$$

The "Jeffreys prior" distribution is a widely used non-informative prior distribution which is justified because the Fisher information is considered as a measure of the information about $\boldsymbol{\psi}$ contained in the observations. It has the desirable property to be invariant under re-parameterization of the parameter vector $\boldsymbol{\psi}$ [Jeffreys, 1946]. Furthermore, the Cramér-Rao bound states that the inverse of the Fisher information is a lower bound on the variance of any unbiased estimator of $\boldsymbol{\psi}$ ([Cramer, 1999] and [Rao, 1945]). Using a "Jeffreys prior" is equivalent to minimize the impact of the prior distribution.

Let us consider that $\mathbf{z}^n$ is sampled from a multivariate Gaussian distribution with mean $\mathbf{F}\boldsymbol{\beta}$ and covariance matrix $\sigma^2\mathbf{R}$, we have:

$$\mathcal{I}(\sigma^2; \mathbf{z}^n) = -\frac{n}{2\sigma^4} + \frac{(\mathbf{z}^n - \mathbf{F}\boldsymbol{\beta})' \mathbf{R}^{-1} (\mathbf{z}^n - \mathbf{F}\boldsymbol{\beta})}{\sigma^6}.$$

From which we deduce that:

$$\mathbb{I}(\sigma^2) = \frac{n}{2\sigma^4}.$$

The non-informative Jeffreys distribution is then given by:

$$p(\sigma^2) \propto \frac{1}{\sigma^2}. \tag{1.11}$$

Following the same guideline, we find that:

$$p(\boldsymbol{\beta}|\sigma^2) \propto 1. \tag{1.12}$$

We note that an improper prior distribution is not bad if the provided posterior distribution is proper. Indeed, according to the Bayesian version of the likelihood principle, only the posterior

	$[\sigma^2] \sim \mathcal{IG}(\alpha, \gamma)$	$p(\sigma^2) \propto \frac{1}{\sigma^2}$
$[\boldsymbol{\beta}\vert\sigma^2] \sim \mathcal{N}_p(\mathbf{b}_0, \sigma^2\mathbf{V}_0)$	(1)	(2)
$p(\boldsymbol{\beta}\vert\sigma^2) \propto 1$	(3)	(4)

Table 1.1: Four different cases corresponding to four combinations of prior distributions for the model parameters.

distributions are of importance (see [Robert, 2007] Sections 1.3 and 1.5). Furthermore, from a practical point of view, Bayesian methods can be applied as soon as the posterior distributions are proper. We note that some arguments about the advantage of improper prior distributions are given in [Robert, 2007] Section 1.5.

Bayesian parameter estimation

We describe here the Bayesian estimation of the parameters $(\boldsymbol{\beta}, \sigma^2)$ in equations (1.3) and (1.4). We use a hierarchical specification for the model parameters. At the lowest level, we consider the parameter $\boldsymbol{\beta}$. At the second level we have the parameter σ^2 which controls the distribution of $\boldsymbol{\beta}$. At the top level we have the parameter $\boldsymbol{\theta}$ which controls the distribution of σ^2 and $\boldsymbol{\beta}$. In the Bayesian literature, we call hierarchical models those coming from this procedure [Robert, 2007]. Throughout the manuscript, we do not consider the hyper-parameter $\boldsymbol{\theta}$ as a random variable except in Subsection 1.3.1 where we present how to perform a Bayesian estimation of $\boldsymbol{\theta}$. Other estimation methods for $\boldsymbol{\theta}$ are described in Subsection 1.3.

Parameter prior distributions. We consider the following informative prior distributions:

$$[\boldsymbol{\beta}|\sigma^2] := \mathcal{N}\left(\mathbf{b}_0, \sigma^2\mathbf{V}_0\right) \tag{1.13}$$

and

$$[\sigma^2] := \mathcal{IG}(\alpha, \gamma), \tag{1.14}$$

where $\mathcal{IG}(\alpha, \gamma)$ stands for the inverse gamma distribution with density function

$$p(x) = \frac{\gamma^\alpha}{\Gamma(\alpha)} \frac{e^{-\gamma/x}}{x^{\alpha+1}} \mathbf{1}_{x>0}.$$

Those prior distributions are commonly used in Bayesian kriging. They allow for obtaining closed form expression for the predictive distribution. Such priors are called conjugate priors in the Bayesian literature. In the forthcoming developments, we consider the four cases presented in Table 1.1.

Parameter posterior distributions. We gave in Table 1.1 the prior distributions of the parameters. The purpose of this paragraph is to provide their posterior distributions, i.e. the one conditioned by the observed values $\mathbf{z}^n$. The equations derived below can be found in the book of [Santner et al., 2003]. First, let us explain the likelihood of $\boldsymbol{\beta}$ and σ^2:

$$f(\mathbf{z}^n|\boldsymbol{\beta}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}\sqrt{\det \mathbf{R}}} \exp\left(-\frac{1}{2}\frac{(\mathbf{z}^n - \mathbf{F}\boldsymbol{\beta})' \mathbf{R}^{-1} (\mathbf{z}^n - \mathbf{F}\boldsymbol{\beta})}{\sigma^2}\right). \tag{1.15}$$

The Bayes rules[1] give us the following equation

$$p(\boldsymbol{\beta}|\mathbf{z}^n, \sigma^2) \propto f(\mathbf{z}^n|\boldsymbol{\beta}, \sigma^2)p(\boldsymbol{\beta}|\sigma^2), \tag{1.16}$$

from which we can deduce that the posterior distribution $[\boldsymbol{\beta}|\mathbf{z}^n, \sigma^2]$ for parameter $\boldsymbol{\beta}$ is the following one:

$$[\boldsymbol{\beta}|\mathbf{z}^n, \sigma^2] = \mathcal{N}\left(\mathbf{A}\boldsymbol{\nu}, \mathbf{A}\right), \tag{1.17}$$

where:

$$\mathbf{A}^{-1} = \begin{cases} [\mathbf{F}'\mathbf{R}^{-1}\mathbf{F} + \mathbf{V}_0^{-1}]/\sigma^2 & (1)\&(2) \\ [\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}]/\sigma^2 & (3)\&(4) \end{cases}$$

and

$$\boldsymbol{\nu} = \begin{cases} [\mathbf{F}'\mathbf{R}^{-1}\mathbf{z}^n + \mathbf{V}_0^{-1}\mathbf{b}_0]/\sigma^2 & (1)\&(2) \\ [\mathbf{F}'\mathbf{R}^{-1}\mathbf{z}^n]/\sigma^2 & (3)\&(4) \end{cases}$$

Then, the following equality

$$p(\sigma^2|\mathbf{z}^n) = f(\mathbf{z}^n|\boldsymbol{\beta}, \sigma^2)p(\boldsymbol{\beta}|\sigma^2)p(\sigma^2)/p(\boldsymbol{\beta}|\sigma^2, \mathbf{z}^n)/f(\mathbf{z}^n) \tag{1.18}$$

leads to the following posterior distribution $[\sigma^2|\mathbf{z}^n]$ for parameter σ^2:

$$[\sigma^2|\mathbf{z}^n] = \mathcal{IG}(\nu_\sigma, Q_\sigma), \tag{1.19}$$

where

$$Q_\sigma \propto \begin{cases} 2\gamma + (\mathbf{b}_0 - \hat{\boldsymbol{\beta}})(\mathbf{V}_0 + [\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}]^{-1})^{-1}(\mathbf{b}_0 - \hat{\boldsymbol{\beta}}) + \tilde{Q}_\sigma & (1) \\ (\mathbf{b}_0 - \hat{\boldsymbol{\beta}})'(\mathbf{V}_0 + [\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}]^{-1})^{-1}(\mathbf{b}_0 - \hat{\boldsymbol{\beta}}) + \tilde{Q}_\sigma & (2) \\ 2\gamma + \tilde{Q}_\sigma & (3) \\ \tilde{Q}_\sigma & (4) \end{cases},$$

with $\hat{\boldsymbol{\beta}} = (\mathbf{F}'\mathbf{R}^{-1}\mathbf{F})^{-1}(\mathbf{F}'\mathbf{R}^{-1}\mathbf{z}^n)$, $\tilde{Q}_\sigma = (\mathbf{z}^n)'[\mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{F}(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{F}'\mathbf{R}^{-1}]\mathbf{z}^n$ and

$$\nu_\sigma \propto \begin{cases} n/2 + \alpha & (1) \\ n/2 & (2) \\ n - p/2 + \alpha & (3) \\ n - p/2 & (4) \end{cases}.$$

Posterior predictive distribution

We have explained in equations (1.17) and (1.19) the posterior distribution of parameters $(\boldsymbol{\beta}, \sigma^2)$. The purpose of this paragraph is to provide the posterior predictive distribution $[Z(x)|\mathbf{Z}^n = \mathbf{z}^n]$ integrating the parameter posterior distributions.

First, let us integrate the posterior distribution of $\boldsymbol{\beta}$:

$$p(z(x)|\mathbf{z}^n, \sigma^2) = \int p(z(x)|\mathbf{z}^n, \boldsymbol{\beta}, \sigma^2)p(\boldsymbol{\beta}|\mathbf{z}^n, \sigma^2)d\boldsymbol{\beta}.$$

[1] If A and B are events such that $\mathbb{P}(B) \neq 0$, we have $\mathbb{P}(A|B) = \mathbb{P}(B|A)\,\mathbb{P}(A)/\mathbb{P}(B)$. The continuous version of this result is the following one: given two random variables x and y with conditional distribution $f(x|y)$ and marginal distribution $g(y)$, the conditional distribution of y given x is $g(y|x) = f(x|y)g(y)/\int f(x|y)g(y)\,dy$.

Straightforward calculations give us that the predictive distribution $[Z(x)|\mathbf{Z}^n = \mathbf{z}^n, \sigma^2]$ is the following Gaussian one:

$$\mathcal{N}\left(\hat{z}_{\boldsymbol{\beta}}(x), s^2_{\boldsymbol{\beta}}(x)\right),$$

where

$$\hat{z}_{\boldsymbol{\beta}}(x) = \mathbf{f}'(x)\mathbf{A}\boldsymbol{\nu} + \mathbf{k}'(x)\mathbf{K}^{-1}\left(\mathbf{z}^n - \mathbf{F}\mathbf{A}\boldsymbol{\nu}\right), \tag{1.20}$$

$$s^2_{\boldsymbol{\beta}}(x) = \sigma^2\left(1 - \begin{pmatrix}\mathbf{f}'(x) & \mathbf{k}'(x)\end{pmatrix}\begin{pmatrix}V_i & \mathbf{F}' \\ \mathbf{F} & \mathbf{K}\end{pmatrix}^{-1}\begin{pmatrix}\mathbf{f}(x) \\ \mathbf{k}(x)\end{pmatrix}\right) \tag{1.21}$$

and

$$V_i = \begin{cases} -V_0^{-1} & (1)\&(2) \\ 0 & (2)\&(3) \end{cases}.$$

Equations (1.20) and (1.21) are the Universal Kriging equations. It corresponds to the Simple kriging ones after integrating the posterior distribution of the regression parameter $\boldsymbol{\beta}$.

Now, let us consider the predictive distribution $[Z(x)|\mathbf{Z}^n = \mathbf{z}^n]$ after integrating the posterior distribution of the variance parameter σ^2. The corresponding probability density function is:

$$p(z(x)|\mathbf{z}^n) = \int p(z(x)|\mathbf{z}^n, \sigma^2)p(\sigma^2|\mathbf{z}^n)d\sigma^2.$$

The calculations are tractable and we find that $[Z(x)|\mathbf{Z}^n = \mathbf{z}^n]$ is the following Student-t distribution[2]:

$$\mathcal{T}_1\left(\nu_\sigma, \hat{z}_{\boldsymbol{\beta}}(x), Q_{\boldsymbol{\beta},\sigma}(x)\right), \tag{1.22}$$

where $\hat{z}_{\boldsymbol{\beta}}(x)$ is defined in (1.20),

$$Q_{\boldsymbol{\beta},\sigma}(x) = \frac{Q_\sigma}{\nu_\sigma}\left(1 + \begin{pmatrix}\mathbf{f}'(x) & \mathbf{k}'(x)\end{pmatrix}\begin{pmatrix}V_i & \mathbf{F}' \\ \mathbf{F} & \mathbf{K}^{-1}\end{pmatrix}^{-1}\begin{pmatrix}\mathbf{f}(x) \\ \mathbf{k}(x)\end{pmatrix}\right) \tag{1.23}$$

and Q_σ and ν_σ are introduced in Equation (1.19).

The Student-t predictive distribution corresponds to the Universal kriging predictive distribution after integrating the posterior distribution of the parameter σ^2. Despite the fact that we do not have a Gaussian distribution anymore, the surrogate model is still the mean $\hat{z}_{\boldsymbol{\beta}}(x)$ and the variance $\nu_\sigma Q_{\boldsymbol{\beta},\sigma}(x)/(\nu_\sigma - 2)$ of the predictive distribution informs us about the model mean squared error.

[2]Let us consider a random vector $\mathbf{W} = (W_1, \ldots, W_d)$ distributed according to the Student-t distribution $\mathcal{T}_d(\nu, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, its probability density function is $p(\mathbf{w}) = \Gamma((\nu+d)/2)\left(1 + \frac{1}{\nu}(\mathbf{w}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{w}-\boldsymbol{\mu})\right)^{-(\nu+d)/2}/(\det(\boldsymbol{\Sigma}))^{1/2}(\nu\pi)^{d/2}\Gamma(\nu/2)$. The parameter ν represents the degrees of freedom, $\boldsymbol{\mu}$ is the location parameter and $\boldsymbol{\Sigma}$ is the scale matrix.

1.3 Model Selection

We have presented in Subsection 1.2.2 some predictive distributions integrating different parameter posterior distributions. For all cases, we always considered the hyper-parameter $\boldsymbol{\theta}$ as known. We present in this section different methods to estimate it.

1.3.1 Bayesian estimate

Like presented previously (1.15) and according to the methodology in [Rasmussen and Williams, 2006] p.108, the hyper-parameter $\boldsymbol{\theta}$ controls the prior distributions of $\boldsymbol{\beta}$ and σ^2. Therefore, following the same guideline than in Subsection 1.2.2, we can give a prior distribution $p(\boldsymbol{\theta})$ for $\boldsymbol{\theta}$ and estimate its posterior distribution from the observations. We present below the complete Bayesian scheme. We note that we consider the same prior distributions for the parameters $\boldsymbol{\beta}$ and σ^2 than the ones presented in Subsection 1.2.2 (see Table 1.1). First, as presented in Subsection 1.2.2, at the bottom level we have:

$$p(\boldsymbol{\beta}|\mathbf{z}^n, \sigma^2, \boldsymbol{\theta}) = \frac{f(\mathbf{z}^n|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta})p(\boldsymbol{\beta}|\sigma^2, \boldsymbol{\theta})}{p(\mathbf{z}^n|\sigma^2, \boldsymbol{\theta})}, \tag{1.24}$$

where $f(\mathbf{z}^n|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta})$ is the likelihood (1.15) and $p(\boldsymbol{\beta}|\sigma^2, \boldsymbol{\theta})$ is the prior distribution of $\boldsymbol{\beta}$ representing our knowledge about the parameter before having observations (see Table 1.1). The resulting posterior distribution $p(\boldsymbol{\beta}|\mathbf{z}^n, \sigma^2, \boldsymbol{\theta})$ is given by (1.17). Furthermore, $p(\mathbf{z}^n|\sigma^2, \boldsymbol{\theta})$ is given by the following equation:

$$p(\mathbf{z}^n|\sigma^2, \boldsymbol{\theta}) = \frac{f(\mathbf{z}^n|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta})p(\boldsymbol{\beta}|\sigma^2, \boldsymbol{\theta})}{p(\boldsymbol{\beta}|\mathbf{z}^n, \sigma^2, \boldsymbol{\theta})}.$$

Second, we can obtain the posterior distribution of σ^2 with the following equality

$$p(\sigma^2|\mathbf{z}^n, \boldsymbol{\theta}) = \frac{p(\mathbf{z}^n|\sigma^2, \boldsymbol{\theta})p(\sigma^2|\boldsymbol{\theta})}{p(\mathbf{z}^n|\boldsymbol{\theta})}, \tag{1.25}$$

where $p(\sigma^2|\boldsymbol{\theta})$ is the prior distribution about σ^2 (see Table 1.1). The resulting posterior distribution $p(\sigma^2|\mathbf{z}^n, \boldsymbol{\theta})$ is given by (1.19) and $p(\mathbf{z}^n|\boldsymbol{\theta})$ is given by

$$p(\mathbf{z}^n|\boldsymbol{\theta}) = \frac{p(\mathbf{z}^n|\sigma^2, \boldsymbol{\theta})p(\sigma^2|\boldsymbol{\theta})}{p(\sigma^2|\mathbf{z}^n, \boldsymbol{\theta})}.$$

Finally, we can express the posterior distribution of $\boldsymbol{\theta}$ with the following formula

$$p(\boldsymbol{\theta}|\mathbf{z}^n) = \frac{p(\mathbf{z}^n|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{z}^n)}.$$

In practice, Monte-Carlo Markov Chain (MCMC) methods are used to estimate $p(\boldsymbol{\theta}|\mathbf{z}^n)$ [Robert and Casella, 2004]. We highlight that MCMC schemes only require knowledge of $p(\boldsymbol{\theta}|\mathbf{z}^n)$ up to a multiplicative constant and thus it is not necessary to evaluate $p(\mathbf{z}^n)$. Then, we can integrate the posterior distributions into the predictive distribution. First we integrate the posterior distribution of $\boldsymbol{\beta}$ with the following formula

$$p(z(x)|\mathbf{z}^n, \sigma^2, \boldsymbol{\theta}) = \int p(z(x)|\mathbf{z}^n, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta})p(\boldsymbol{\beta}|\mathbf{z}^n, \sigma^2, \boldsymbol{\theta})d\boldsymbol{\beta}.$$

We obtain a Gaussian distribution with mean (1.20) and variance (1.21). Then we can integrate with respect to σ^2

$$p(z(x)|\mathbf{z}^n, \boldsymbol{\theta}) = \int p(z(x)|\mathbf{z}^n, \sigma^2, \boldsymbol{\theta}) p(\sigma^2|\mathbf{z}^n, \boldsymbol{\theta}) d\sigma^2.$$

We obtain the Student-t distribution in Equation (1.22). Finally, we can integrate the posterior distribution of $\boldsymbol{\theta}$:

$$p(z(x)|\mathbf{z}^n) = \int p(z(x)|\mathbf{z}^n, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathbf{z}^n) d\boldsymbol{\theta}.$$

Nevertheless, the calculations are not anymore tractable and the predictive distribution needs to be numerically estimated. In general, MCMC schemes are used. These numerical integrations may be difficult and as noted in [Santner et al., 2003] the choice of the prior distribution is non-trivial. The reader is referred to the article of [Diggle and Ribeiro Jr, 2002] for examples of prior distributions for $\boldsymbol{\theta}$.

As example, let us consider a 2-dimensional Gaussian process $Z(x)$ with zero mean and a Gaussian covariance kernel $k(x, \tilde{x}) = \sigma^2 \exp\left(-||x - \tilde{x}||^2/(2\theta^2)\right)$ where $\sigma^2 = 4$ and $\theta = 0.1$. We sample a realization $Z(x)$ on 40 points. Then, we consider the parameter θ as unknown and we estimate it from the 40 observations with a Bayesian method. We consider the following improper prior distribution for θ:

$$p(\theta) \propto \frac{1}{\theta}.$$

Figure 1.4 illustrates the prior and the posterior distributions of θ. We see that the prior distribution is far from the real value of θ (the real value being 0.1). Then, the mode of the posterior distribution approaches the real value but with a non-negligible uncertainty.

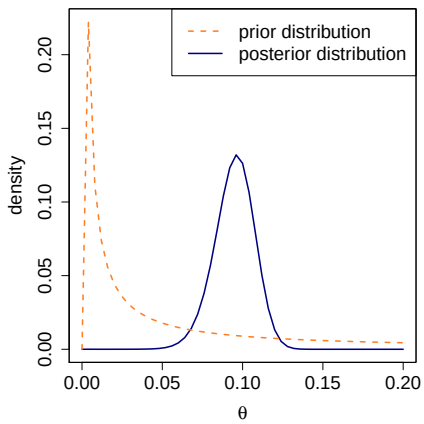


Figure 1.4: Example of prior and posterior distribution for the hyper-parameter θ for an isotropic Gaussian covariance kernel in dimension 2.

Figure 1.5 represents the predictive mean and variance in the Bayesian and non-Bayesian cases. For the non-Bayesian case, we fix $\theta = 0.1$. Since, the mode of the posterior distribution of θ is close to the real value, the means of the predictive distributions are close. Nevertheless, the significant differences between the predictive variances reflect that we take into account

the uncertainty due to the parameter estimation in the Bayesian case. Indeed, we see that in this case the variance is more important.

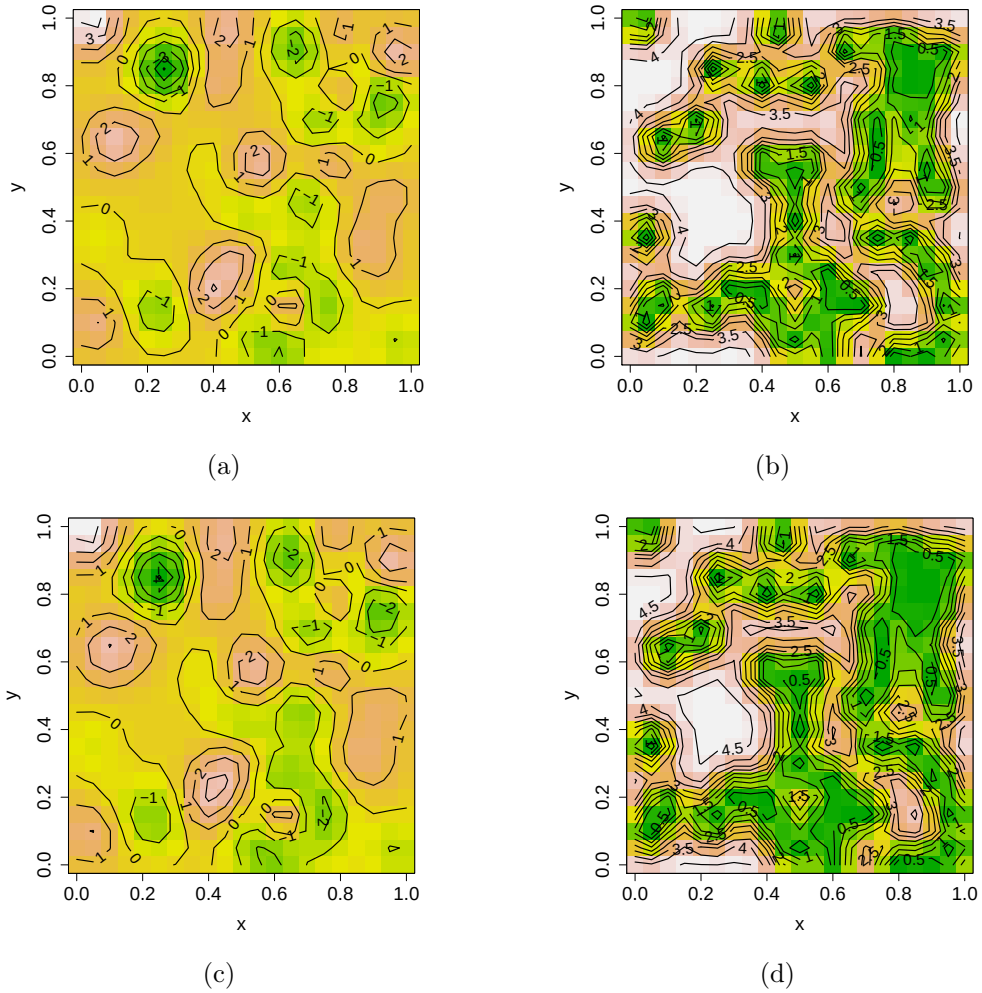


Figure 1.5: Posterior predictive distribution for the Bayesian and the non-Bayesian cases in a 2 dimensional example with a Gaussian kernel. The figures on (a) & (c) represent the posterior means, the figures (b) & (d) represent the predictive variances, the figures (a) & (b) represent the non-Bayesian cases and the figures (c) & (d) represent the Bayesian cases. We see that the predictive means are equivalent. This is due to an efficient estimation of the hyper-parameter θ. Furthermore, the predictive variance is more important in the Bayesian case since we take into account the uncertainty due to the estimation of θ.

1.3.2 Maximum likelihood estimates

The maximum likelihood estimation is a very popular method to estimate parameters. The drawback of the maximum likelihood estimation is that, contrarily to Bayesian estimation, we do not have any information about the variance of the estimator (see [Lehmann and Casella,

1998]). Nevertheless, in a kriging framework, it is significantly less time-consuming than a Bayesian approach. The multivariate normal assumption for $\mathbf{Z}^n$ lead to the following likelihood for parameters $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{\theta}$:

$$f(\mathbf{z}^n|\boldsymbol{\beta},\sigma^2,\boldsymbol{\theta}) = \frac{1}{(2\pi\sigma^2)^{n/2}\sqrt{\det \mathbf{R}_{\boldsymbol{\theta}}}} \exp\left(-\frac{1}{2}\frac{(\mathbf{z}^n - \mathbf{F}\boldsymbol{\beta})' \mathbf{R}_{\boldsymbol{\theta}}^{-1} (\mathbf{z}^n - \mathbf{F}\boldsymbol{\beta})}{\sigma^2}\right). \tag{1.26}$$

The correlation matrix $\mathbf{R}$ is denoted by $\mathbf{R}_{\boldsymbol{\theta}}$ to emphasize its dependence on $\boldsymbol{\theta}$. Conditionally to σ^2 and $\boldsymbol{\theta}$, the maximum likelihood estimate (MLE) of $\boldsymbol{\beta}$ is given by:

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}'\mathbf{R}_{\boldsymbol{\theta}}^{-1}\mathbf{F})^{-1}\mathbf{F}'\mathbf{R}_{\boldsymbol{\theta}}^{-1}\mathbf{z}^n. \tag{1.27}$$

It corresponds to its generalized least squares estimate. Then we can substitute the value of $\hat{\boldsymbol{\beta}}$ in the likelihood (1.26) and maximize it with respect to σ^2. Given $\boldsymbol{\theta}$ we obtain the following MLE for σ^2:

$$\hat{\sigma}^2 = \frac{(\mathbf{z}^n - \mathbf{F}\hat{\boldsymbol{\beta}})'\mathbf{R}_{\boldsymbol{\theta}}^{-1}(\mathbf{z}^n - \mathbf{F}\hat{\boldsymbol{\beta}})}{n}. \tag{1.28}$$

Substituting $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma^2}$ for $\boldsymbol{\beta}$ and σ^2 in Equation (1.26), we obtain that the maximum of the likelihood over $\boldsymbol{\beta}$ and σ^2 is

$$f(\mathbf{z}^n|\boldsymbol{\theta}) = (2\pi\hat{\sigma}^2)^{-n/2}(\det \mathbf{R}_{\boldsymbol{\theta}})^{1/2} \exp\left(-\frac{n}{2}\right),$$

which depends only on $\boldsymbol{\theta}$. Therefore, the MLE of $\boldsymbol{\theta}$ can be found by minimizing the opposite of the log-likelihood given by (up to a constant):

$$\mathcal{L}_{\text{rest}}(\boldsymbol{\theta};\mathbf{z}^n) = n\log(\hat{\sigma}^2) + \log(\det(\mathbf{R}_{\boldsymbol{\theta}})). \tag{1.29}$$

The opposite of this equation is called the concentrated log-likelihood or the marginal likelihood. We illustrate in Figure 1.6 an example of $\mathcal{L}_{\text{rest}}(\boldsymbol{\theta};\mathbf{z}^n)$ (1.29) calculated from the realization of a 2-dimensional Gaussian process of mean zero and covariance $k(x,\tilde{x}) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{i=1}^2 (x^i - \tilde{x}^i)^2/\boldsymbol{\theta}_i^2\right)$ - where $x = (x^1, x^2) \in \mathbb{R}^2$, $\tilde{x} = (\tilde{x}^1, \tilde{x}^2) \in \mathbb{R}^2$, $\boldsymbol{\theta}_1 = 0.1$, $\boldsymbol{\theta}_2 = 0.04$ and $\sigma^2 = 2$ - on 150 design points in $[0,1]^2$. The marginal likelihood has to be numerically minimized with global optimization methods. To have a more effective optimization, one can used the derivative of the marginal likelihood[3]:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\theta}_i}\mathcal{L}_{\text{rest}}(\boldsymbol{\theta};\mathbf{z}^n) &= -n\left((\mathbf{y}^n)'\mathbf{R}_{\boldsymbol{\theta}}^{-1}\mathbf{y}^n\right)^{-1}(\mathbf{y}^n)'\mathbf{R}_{\boldsymbol{\theta}}^{-1}\frac{\partial \mathbf{R}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}_i}\mathbf{R}_{\boldsymbol{\theta}}^{-1}\mathbf{y}^n \\ &\quad + \text{tr}\left(\mathbf{R}_{\boldsymbol{\theta}}^{-1}\frac{\partial \mathbf{R}_{\boldsymbol{\theta}}}{\boldsymbol{\theta}_i}\right),\end{aligned}$$

with $\mathbf{y}^n = \mathbf{z}^n - \mathbf{F}\hat{\boldsymbol{\beta}}$.

Restricted Maximum Likelihood estimate. The restricted maximum likelihood method was introduced by [Patterson and Thompson, 1971] in order to reduce the bias of the maximum likelihood estimator. The restricted maximum likelihood estimates of the parameters σ^2

[3]The proof is straightforward using the derivative of an inverse matrix $\frac{\partial}{\partial\theta}K_\theta^{-1} = -K_\theta^{-1}\frac{\partial K_\theta}{\partial\theta}K_\theta^{-1}$ and the one of the log determinant of a positive definite symmetric matrix $\frac{\partial}{\partial\theta}\log\det K_\theta = \text{tr}\left(K_\theta^{-1}\frac{\partial K_\theta}{\partial\theta}\right)$ where $\frac{\partial K_\theta}{\partial\theta}$ is a matrix of element-wise derivatives (see [Harville, 1997]).

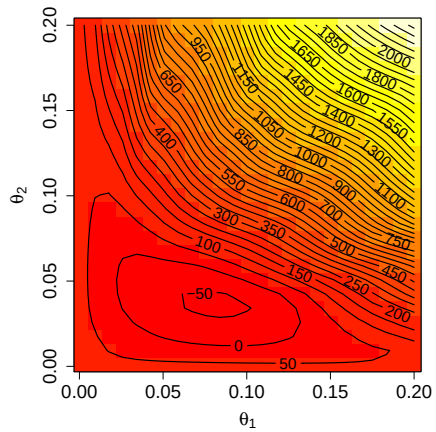


Figure 1.6: An example of the opposite of a log-likelihood calculated with 150 observations sampled from a Gaussian process of zero mean and covariance kernel $k(x, \tilde{x}) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{2}(x^i - \tilde{x}^i)^2/\boldsymbol{\theta}_i^2\right)$ with $\boldsymbol{\theta}_1 = 0.1$, $\boldsymbol{\theta}_2 = 0.04$ and $\sigma^2 = 2$.

and $\boldsymbol{\theta}$ consist in maximizing the likelihood of those parameters for a maximum of independent linear combinations of the observations $\mathbf{z}^n$ and such that all these combinations are orthogonal to $\mathbf{F}\boldsymbol{\beta}$, i.e. the mean of $\mathbf{Z}^n$. For more detail, the reader could refer to the two reference articles [Harville, 1974] and [Harville, 1977].

Now, let us consider a matrix $\mathbf{C}$ of size $(n-p) \times n$ of rank $(n-p)$ such that $\mathbf{CF} = 0$. The restricted maximum likelihood estimate of σ^2 and $\boldsymbol{\theta}$ are given by the classical maximum likelihood estimate but with the transformed data $\tilde{\mathbf{z}}^n = \mathbf{Cz}^n$. We note that the restricted MLE is independent of the choice of $\mathbf{C}$ (see [Harville, 1977]). The likelihood of $\tilde{\mathbf{Z}}^n = \mathbf{CZ}^n$ is given by:

$$f(\tilde{\mathbf{z}}^n|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}) = \frac{1}{(2\pi\sigma^2)^{(n-p)/2}\sqrt{\det(\mathbf{CR}_{\boldsymbol{\theta}}\mathbf{C}')}} \exp\left(-\frac{1}{2}\frac{(\tilde{\mathbf{z}}^n)'\,(\mathbf{CR}_{\boldsymbol{\theta}}\mathbf{C}')^{-1}\tilde{\mathbf{z}}^n}{\sigma^2}\right). \tag{1.30}$$

Maximizing (1.30) with respect to σ^2 and considering that the estimator is independent to the choice of $\mathbf{C}$, we have the following restricted maximum likelihood estimate for the variance parameter:

$$\hat{\sigma}^2_{\text{REML}} = \frac{(\mathbf{z}^n - \mathbf{F}\hat{\boldsymbol{\beta}})'\mathbf{R}_{\boldsymbol{\theta}}^{-1}(\mathbf{z}^n - \mathbf{F}\hat{\boldsymbol{\beta}})}{n-p}. \tag{1.31}$$

Furthermore, substituting σ^2 with $\hat{\sigma}^2_{\text{REML}}$ in the likelihood (1.30), we find that the restricted maximum likelihood of $\boldsymbol{\theta}$ can be found by minimizing:

$$(n-p)\log(\hat{\sigma}^2_{\text{REML}}) + \log(\det(\mathbf{R}_{\boldsymbol{\theta}})). \tag{1.32}$$

Marginal likelihood in a noisy case. In a noisy case, we cannot derive a closed form expression for the estimate of σ^2. Indeed, in that case the likelihood for $\boldsymbol{\beta}$, σ^2, $\boldsymbol{\theta}$ and $\boldsymbol{\Delta}$ - see Equation (1.7) in Subsection 1.2.1 - is given by

$$f(\mathbf{z}^n|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\theta}, \boldsymbol{\Delta}) = \frac{\exp\left(-(\mathbf{z}^n - \mathbf{F}\boldsymbol{\beta})'\left(\mathbf{K}_{\sigma^2,\boldsymbol{\theta}} + \boldsymbol{\Delta}\right)^{-1}(\mathbf{z}^n - \mathbf{F}\boldsymbol{\beta})/2\right)}{(2\pi\sigma^2)^{n/2}\sqrt{\det\left(\mathbf{K}_{\sigma^2,\boldsymbol{\theta}} + \boldsymbol{\Delta}\right)}}. \tag{1.33}$$

We use the notation $\mathbf{K}_{\sigma^2,\boldsymbol{\theta}}$ to emphasize the dependence of $\mathbf{K} = [k(x_i, x_j)]_{i,j=1,\ldots,n}$ to the parameters σ^2 and $\boldsymbol{\theta}$. Thus, we have the following estimate for $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = (\mathbf{F}' \left(\mathbf{K}_{\sigma^2,\boldsymbol{\theta}} + \boldsymbol{\Delta}\right)^{-1} \mathbf{F})^{-1} \mathbf{F}' \left(\mathbf{K}_{\sigma^2,\boldsymbol{\theta}} + \boldsymbol{\Delta}\right)^{-1} \mathbf{z}^n. \tag{1.34}$$

The opposite of the marginal likelihood becomes up to a constant

$$\begin{aligned} \mathcal{L}_{\text{rest}}(\boldsymbol{\theta}, \sigma^2, \boldsymbol{\Delta}; \mathbf{z}^n) &= (\mathbf{z}^n - \mathbf{F}\boldsymbol{\beta})' \left(\mathbf{K}_{\sigma^2,\boldsymbol{\theta}} + \boldsymbol{\Delta}\right)^{-1} (\mathbf{z}^n - \mathbf{F}\boldsymbol{\beta}) \\ &+ \log\det\left(\mathbf{K}_{\sigma^2,\boldsymbol{\theta}} + \boldsymbol{\Delta}\right). \end{aligned}$$

We illustrate in Figure 1.7 an example of $\mathcal{L}_{\text{rest}}(\theta, \sigma^2, \boldsymbol{\Delta} = \sigma_\varepsilon^2 \mathbf{I}; \mathbf{z}^n)$ calculated from the realization of a 1-dimensional Gaussian process of mean zero and covariance $k(x, \tilde{x}) = \sigma^2 \exp\left(-\frac{1}{2}\frac{(x-\tilde{x})^2}{\theta^2}\right) + \sigma_\varepsilon^2 \delta_{x=\tilde{x}}$ - where $x, \tilde{x} \in \mathbb{R}$, $\theta = 0.1$, $\sigma_\varepsilon^2 = 0.25$ and $\sigma^2 = 2$ - on 150 design points in $[0, 1]$. We note that σ^2 is supposed to be known.

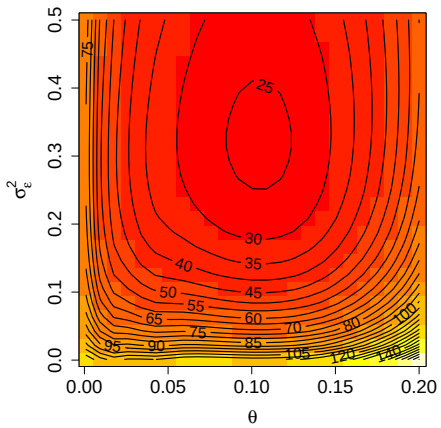


Figure 1.7: An example of the opposite of a log-likelihood calculated with 150 observations sampled from a Gaussian process of zero mean and covariance kernel $k(x, \tilde{x}) = \sigma^2 \exp\left(-\frac{1}{2}\frac{(x-\tilde{x})^2}{\theta^2}\right) + \sigma_\varepsilon^2 \delta_{x=\tilde{x}}$ with $\theta = 0.1$, $\sigma_\varepsilon^2 = 0.25$ and $\sigma^2 = 2$. The variance parameter σ^2 is supposed to be known.

1.3.3 Cross-validation estimate

The principle of a cross-validation (CV) procedure is to split the experimental design set into two disjoint sets, one is used for training and the other one is used to monitor the performance of the surrogate model. The idea of a CV estimation is then to find the parameter $\boldsymbol{\theta}$ leading to the best performance on the test set. A particular case of CV is the Leave-One-Out (LOO) one where n test sets are obtained by removing one observation at-a-time. The CV procedure can be time-consuming for a kriging model - e.g. for the LOO scheme it requires the inversion of n sub-matrices of size $n-1$ - but it is shown by [Rasmussen and Williams, 2006], [Dubrule, 1983] and [Zhang and Wang, 2009] that there are computational shortcuts. We present them in the remainder of this paragraph.

Notations: If ξ is a subset of indices in $\{1,\dots,n\}$, then $\mathbf{A}_{[\xi,\xi]}$ is the sub-matrix of elements $\xi\times\xi$ of $\mathbf{A}$, $\mathbf{a}_{[\xi]}$ is the sub-vector of elements ξ of $\mathbf{a}$, $\mathbf{A}_{[-\xi]}$ represents the matrix $\mathbf{A}$ in which we remove the rows of index ξ, $\mathbf{a}_{[-\xi]}$ represents the vector $\mathbf{a}$ in which we remove the elements of index ξ, $\mathbf{A}_{[-\xi,-\xi]}$ is the sub-matrix of $\mathbf{A}$ in which we remove the rows and columns of index ξ and $\mathbf{A}_{[-\xi,\xi]}$ is the sub-matrix of $\mathbf{A}$ in which we remove the rows of index ξ and keep only the columns of index ξ.

CV for Universal kriging

Let us consider a set of index $\xi\subset\{1,\dots,n\}$ of length k. We denote by $\boldsymbol{\varepsilon}_{\mathrm{CV},\xi}$ the errors (i.e. the real values minus the predicted values) of the cross-validation procedure on the test set $\mathbf{D}_{[\xi]}$ when we learn the kriging model on the training set $\mathbf{D}_{[-\xi]}$. Furthermore, we denote by $\boldsymbol{\sigma}^2_{\mathrm{CV},\xi}$ the predictive CV variances at points in $\mathbf{D}_{[\xi]}$. For the proof, we sort the observations $\mathbf{z}^n$ such that ξ is the index of the k last elements of $\mathbf{z}^n$. Nevertheless, the presented equations remain true whatever the order of the observations. First, we consider the variance parameter σ^2, the hyper-parameter $\boldsymbol{\theta}$ and the regression parameter $\boldsymbol{\beta}$ as known. We are hence in the simple kriging case. Thanks to the block-wise inversion formula[4], we have the following equality:

$$\mathbf{R}^{-1}=\begin{pmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}' & \mathbf{Q}^{-1}\end{pmatrix},$$

with $\mathbf{A}=[\mathbf{R}]^{-1}_{[-\xi,-\xi]}+[\mathbf{R}]^{-1}_{[-\xi,-\xi]}[\mathbf{R}]_{[-\xi,\xi]}\mathbf{Q}^{-1}[\mathbf{R}]_{[\xi,-\xi]}[\mathbf{R}]^{-1}_{[-\xi,-\xi]}$,
$\mathbf{B}'=-\mathbf{Q}^{-1}[\mathbf{R}]_{[\xi,-\xi]}[\mathbf{R}]^{-1}_{[-\xi,\xi]}$ and:

$$\mathbf{Q}=[\mathbf{R}]_{[\xi,\xi]}-[\mathbf{R}]_{[\xi,-\xi]}[\mathbf{R}]^{-1}_{[-\xi,-\xi]}[\mathbf{R}]_{[-\xi,\xi]}.$$

We note that $\mathbf{Q}=\left(\left[\mathbf{R}^{-1}\right]_{[\xi,\xi]}\right)^{-1}$ represents the correlation matrix at points in $\mathbf{D}_{[\xi]}$ with respect to the correlation kernel obtained from the distribution of a Gaussian process of kernel $r(x,x')$ conditioned by $\mathbf{z}^n_{[-\xi]}$ at $\mathbf{D}_{[-\xi]}$. Therefore, we can deduce that in a Simple kriging case, the predictive CV variances $\boldsymbol{\sigma}^2_{\mathrm{CV},\xi,\mathrm{SK}}$ are

$$\boldsymbol{\sigma}^2_{\mathrm{CV},\xi,\mathrm{SK}}=\sigma^2\left(\left[\mathbf{R}^{-1}\right]_{[\xi,\xi]}\right)^{-1}. \tag{1.35}$$

[4]Let us consider $\mathbf{T}$ a $m\times m$ matrix, $\mathbf{U}$ a $m\times n$ matrix, $\mathbf{V}$ a $n\times m$ matrix and $\mathbf{W}$ a $n\times n$ matrix. Let us consider that $\mathbf{T}$ is non-singular, then $\begin{pmatrix}\mathbf{T} & \mathbf{U}\\ \mathbf{V} & \mathbf{W}\end{pmatrix}$, or equivalently, $\begin{pmatrix}\mathbf{W} & \mathbf{V}\\ \mathbf{U} & \mathbf{T}\end{pmatrix}$ is non-singular if and only if the matrix $n\times n$ $\mathbf{Q}=\mathbf{W}-\mathbf{V}\mathbf{T}^{-1}\mathbf{U}$ is non-singular. In this case, we have:

$$\begin{pmatrix}\mathbf{T} & \mathbf{U}\\ \mathbf{V} & \mathbf{W}\end{pmatrix}^{-1}=\begin{pmatrix}\mathbf{T}^{-1}+\mathbf{T}^{-1}\mathbf{U}\mathbf{Q}^{-1}\mathbf{V}\mathbf{T}^{-1} & -\mathbf{T}^{-1}\mathbf{U}\mathbf{Q}^{-1}\\ -\mathbf{Q}^{-1}\mathbf{V}\mathbf{T}^{-1} & \mathbf{Q}^{-1}\end{pmatrix}$$

and

$$\begin{pmatrix}\mathbf{W} & \mathbf{V}\\ \mathbf{U} & \mathbf{T}\end{pmatrix}^{-1}=\begin{pmatrix}\mathbf{Q}^{-1} & -\mathbf{Q}^{-1}\mathbf{V}\mathbf{T}^{-1}\\ -\mathbf{T}^{-1}\mathbf{U}\mathbf{Q}^{-1} & \mathbf{T}^{-1}+\mathbf{T}^{-1}\mathbf{U}\mathbf{Q}^{-1}\mathbf{V}\mathbf{T}^{-1}\end{pmatrix}$$

Furthermore, from the block decomposition of $\mathbf{R}^{-1}$, we have the following equality:

$$\begin{aligned}\left(\left[\mathbf{R}^{-1}\right]_{[\xi,\xi]}\right)^{-1}\left[\mathbf{R}^{-1}\left(\mathbf{z}^n-\mathbf{F}\boldsymbol{\beta}\right)\right]_{[\xi]} &= \mathbf{z}^n_{[\xi]}-\mathbf{F}_{[\xi]}\boldsymbol{\beta} \\ &- \left[\mathbf{R}\right]_{[\xi,-\xi]}\left[\mathbf{R}\right]^{-1}_{[-\xi,-\xi]}\left(\mathbf{z}^n_{[-\xi]}-\mathbf{F}_{[-\xi]}\boldsymbol{\beta}\right)\end{aligned}.$$

We highlight that the term $\mathbf{F}_{[\xi]}\boldsymbol{\beta}+\left[\mathbf{R}\right]_{[\xi,-\xi]}\left[\mathbf{R}\right]^{-1}_{[-\xi,-\xi]}\left(\mathbf{z}^n_{[-\xi]}-\mathbf{F}_{[-\xi]}\boldsymbol{\beta}\right)$ represents the kriging mean predictions on $\mathbf{D}_{[\xi]}$ of a Gaussian process of mean $\mathbf{f}(x)'\boldsymbol{\beta}$ and correlation kernel $r(x,\tilde{x})$ conditioned with the observations $\mathbf{z}^n_{[-\xi]}$. Thus we can deduce that in a Simple kriging case, the CV errors $\boldsymbol{\varepsilon}_{\mathrm{CV},\xi,\mathrm{SK}}$ are

$$\boldsymbol{\varepsilon}_{\mathrm{CV},\xi,\mathrm{SK}}=\left(\left[\mathbf{R}^{-1}\right]_{[\xi,\xi]}\right)^{-1}\left[\mathbf{R}^{-1}\left(\mathbf{z}^n-\mathbf{F}\boldsymbol{\beta}\right)\right]_{[\xi]}. \tag{1.36}$$

Second, we suppose the trend parameter $\boldsymbol{\beta}$ as unknown and we have to re-estimate it when we remove the observations. We emphasize that we are here in a Universal kriging framework. In a Bayesian case, the posterior mean $\bar{\boldsymbol{\beta}}_{-\xi}$ of $\boldsymbol{\beta}$ when we remove the observations of index ξ is given by

$$\bar{\boldsymbol{\beta}}_{-\xi}\left(\left[\mathbf{F}_{[-\xi]}\right]'\left[\mathbf{R}\right]^{-1}_{[-\xi,-\xi]}\mathbf{F}_{[-\xi]}\right)=\left[\mathbf{F}_{[-\xi]}\right]'\left[\mathbf{R}\right]^{-1}_{[-\xi,-\xi]}\mathbf{z}^n_{[-\xi]}. \tag{1.37}$$

From the block-wise inverse of $\mathbf{R}$ we can deduce that $\left[\mathbf{R}\right]^{-1}_{[-\xi,-\xi]}=\mathbf{A}-\mathbf{BQB}'$. To obtain the cross-validation equations in the Universal kriging case, we just have to estimate the following quantity:

$$\boldsymbol{\nu}_\xi=\left(\mathbf{F}_{[\xi]}-\left[\mathbf{R}\right]_{[\xi,-\xi]}\left[\mathbf{R}\right]^{-1}_{[-\xi,-\xi]}\mathbf{F}_{[-\xi]}\right)\boldsymbol{\Sigma}\left(\mathbf{F}_{[\xi]}-\left[\mathbf{R}\right]_{[\xi,-\xi]}\left[\mathbf{R}\right]^{-1}_{[-\xi,-\xi]}\mathbf{F}_{[-\xi]}\right)',$$

with $\boldsymbol{\Sigma}=\left(\left[\mathbf{F}_{[-\xi]}\right]'\left[\mathbf{R}\right]^{-1}_{[-\xi,-\xi]}\mathbf{F}_{[-\xi]}\right)^{-1}$. Indeed, from equations (1.4) and (1.21), we can deduce that $\boldsymbol{\sigma}^2_{\mathrm{CV},\xi}=\boldsymbol{\sigma}^2_{\mathrm{CV},\xi,\mathrm{SK}}+\boldsymbol{\nu}_\xi$. We have the following equality:

$$\left(\mathbf{F}_{[\xi]}-\left[\mathbf{R}\right]_{[\xi,-\xi]}\left[\mathbf{R}\right]^{-1}_{[-\xi,-\xi]}\mathbf{F}_{[-\xi]}\right)=\left(\left[\mathbf{R}^{-1}\right]_{[\xi,\xi]}\right)^{-1}\left[\mathbf{R}^{-1}\mathbf{F}\right]_{[\xi]}.$$

Therefore, the CV predictive errors and variances in a Universal kriging framework are given by

$$\boldsymbol{\varepsilon}_{\mathrm{CV},\xi}=\left(\left[\mathbf{R}^{-1}\right]_{[\xi,\xi]}\right)^{-1}\left[\mathbf{R}^{-1}\left(\mathbf{z}^n-\mathbf{F}\bar{\boldsymbol{\beta}}_{-\xi}\right)\right]_{[\xi]} \tag{1.38}$$

and

$$\begin{aligned}\boldsymbol{\sigma}^2_{\mathrm{CV},\xi} &= \sigma^2\left(\left(\left[\mathbf{R}^{-1}\right]_{[\xi,\xi]}\right)^{-1}+\left(\left[\mathbf{R}^{-1}\right]_{[\xi,\xi]}\right)^{-1}\left[\mathbf{R}^{-1}\mathbf{F}\right]_{[\xi]}\right. \\ &\times \left.\left(\left[\mathbf{F}_{[-\xi]}\right]'\left[\mathbf{R}\right]^{-1}_{[-\xi,-\xi]}\mathbf{F}_{[-\xi]}\right)^{-1}\left(\left(\left[\mathbf{R}^{-1}\right]_{[\xi,\xi]}\right)^{-1}\left[\mathbf{R}^{-1}\mathbf{F}\right]_{[\xi]}\right)'\right)\end{aligned} \tag{1.39}$$

The term $\left[\mathbf{R}\right]^{-1}_{[-\xi,-\xi]}$ is evaluated with the equality:

$$\left[\mathbf{R}\right]^{-1}_{[-\xi,-\xi]}=\left[\mathbf{R}^{-1}\right]_{[-\xi,-\xi]}-\left[\mathbf{R}^{-1}\right]_{[-\xi,\xi]}\left(\left[\mathbf{R}^{-1}\right]_{[\xi,\xi]}\right)^{-1}\left[\mathbf{R}^{-1}\right]_{[\xi,-\xi]}.$$

To obtain the CV predictive errors and variances in a Universal kriging framework, we just have to invert the matrix $\mathbf{R}$ once and then invert the sub-matrix $\left[\mathbf{R}^{-1}\right]_{[\xi,\xi]}$. We note that in a LOO framework, ξ is reduced to an integer and the computational cost for the inversion of $\left[\mathbf{R}^{-1}\right]_{[\xi,\xi]}$ is negligible. In the presented equations, the variance parameter is supposed to be known. We present in Chapter 4 a method to re-estimate it for each removed observations when we consider its maximum likelihood estimate.

Leave-One-Out based estimation

In the previous paragraph, we present the predictive errors and variances resulting from a CV procedure when σ^2 and $\boldsymbol{\theta}$ are fixed. We present here a way to estimate them thanks to a LOO CV technique, i.e. $\xi = i$ with $i = 1, \ldots, n$. The opposite of the predictive log probability at observation $\mathbf{z}^n_{[i]}$ when the model is learned with the observations $\mathbf{z}^n_{[-i]}$ is given by (up to a constant):

$$\mathcal{L}(\sigma^2, \boldsymbol{\theta}; \mathbf{z}^n_{[i]}) = \log \boldsymbol{\sigma}^2_{\mathrm{CV},i} + \frac{\boldsymbol{\varepsilon}^2_{\mathrm{CV},i}}{\boldsymbol{\sigma}^2_{\mathrm{CV},i}}. \tag{1.40}$$

where

$$\boldsymbol{\varepsilon}_{\mathrm{CV},i} = \left(\left[\mathbf{R}^{-1}\right]_{[i,i]}\right)^{-1} \left[\mathbf{R}^{-1}\left(\mathbf{z}^n - \mathbf{F}\bar{\boldsymbol{\beta}}_{-i}\right)\right]_{[i]}$$

and

$$\begin{aligned} \boldsymbol{\sigma}^2_{\mathrm{CV},i} &= \sigma^2 \left(\left(\left[\mathbf{R}^{-1}\right]_{[i,i]}\right)^{-1} + \left(\left[\mathbf{R}^{-1}\right]_{[i,i]}\right)^{-1} \left[\mathbf{R}^{-1}\mathbf{F}\right]_{[i]} \right. \\ &\times \left. \left(\left[\mathbf{F}_{[-i]}\right]' \left[\mathbf{R}\right]^{-1}_{[-i,-i]} \mathbf{F}_{[-i]}\right)^{-1} \left(\left(\left[\mathbf{R}^{-1}\right]_{[i,i]}\right)^{-1} \left[\mathbf{R}^{-1}\mathbf{F}\right]_{[i]}\right)' \right). \end{aligned}$$

From Equation (1.40) we can obtained the opposite of the LOO log-predictive probability

$$\mathcal{L}_{\mathrm{LOO}}(\sigma^2, \boldsymbol{\theta}; \mathbf{z}^n) = \sum_{i=1}^{n} \mathcal{L}(\sigma^2, \boldsymbol{\theta}; \mathbf{z}^n_{[i]}). \tag{1.41}$$

The reader is referred to the books of [Rasmussen and Williams, 2006] p122 for an illustration of this criterion in a robotic application and the article of [Geisser and Eddy, 1979] for a discussion about it. We note that thanks to the equations (1.38) and (1.39), this approach is as computationally expensive as the classical maximum likelihood one.

We illustrate in Figure 1.6 an example of a LOO log predictive probability $\mathcal{L}_{\mathrm{LOO}}(\sigma^2, \boldsymbol{\theta}, \mathbf{z}^n)$ (1.41) calculated from the realization of a 2-dimensional Gaussian process of mean zero and covariance $k(x, \tilde{x}) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{2}(x^i - \tilde{x}^i)^2/\boldsymbol{\theta}_i^2\right)$ - where $x = (x^1, x^2) \in [0,1]^2$, $\tilde{x} = (\tilde{x}^1, \tilde{x}^2) \in [0,1]^2$, $\boldsymbol{\theta}_1 = 0.1$, $\boldsymbol{\theta}_2 = 0.04$ and $\sigma^2 = 2$ - on 150 design points in $[0,1]^2$.

Another approach to estimate the parameters $\boldsymbol{\theta}$ and σ^2 has been suggested by [Bachoc, 2013]. Its principle is the following one. First, noticing that the CV predictive errors (1.38) do not depend on σ^2, we can estimate $\boldsymbol{\theta}$ by minimizing the following sum - also called the squared error loss:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \boldsymbol{\varepsilon}^2_{\mathrm{CV},i,\boldsymbol{\theta}}. \tag{1.42}$$

The LOO CV predictive error (1.38) is denoted by $\boldsymbol{\varepsilon}_{\mathrm{CV},i,\boldsymbol{\theta}}$ to emphasize its dependence on $\boldsymbol{\theta}$. Nonetheless, this procedure does not provide an estimate for σ^2 and can lead to bad predictive variances since it does not take care about the LOO-CV predictive variances. To tackle this issue, [Bachoc, 2013] suggests the following estimator for σ^2:

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} \frac{\boldsymbol{\varepsilon}^2_{\mathrm{CV},i,\hat{\boldsymbol{\theta}}}}{\tilde{\boldsymbol{\sigma}}^2_{\mathrm{CV},i,\hat{\boldsymbol{\theta}}}}, \tag{1.43}$$

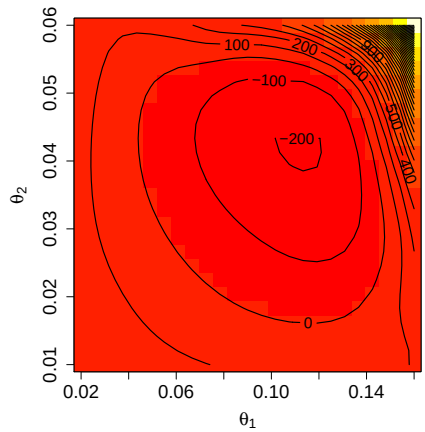


Figure 1.8: An example of LOO log-predictive probability calculated with 150 observations sampled from a Gaussian process of zero mean and covariance kernel $k(x,\tilde{x}) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{2}(x^i-\tilde{x}^i)^2/\boldsymbol{\theta}_i^2\right)$ with $\boldsymbol{\theta}_1 = 0.1$, $\boldsymbol{\theta}_2 = 0.04$ and $\sigma^2 = 2$.

where $\tilde{\boldsymbol{\sigma}}^2_{\mathrm{CV},\xi,\hat{\boldsymbol{\theta}}}$ is obtained from Equation (1.39):

$$\begin{aligned}\tilde{\boldsymbol{\sigma}}^2_{\mathrm{CV},\xi,\hat{\boldsymbol{\theta}}} &= \left(\left[\mathbf{R}_{\hat{\boldsymbol{\theta}}}^{-1}\right]_{[\xi,\xi]}\right)^{-1} \\ &+ \left([\mathbf{R}_{\hat{\boldsymbol{\theta}}}^{-1}]_{[\xi,\xi]}\right)^{-1}\left[\mathbf{R}_{\hat{\boldsymbol{\theta}}}^{-1}\mathbf{F}\right]_{[\xi]}\left([\mathbf{F}_{[-\xi]}]'\left[\mathbf{R}_{\hat{\boldsymbol{\theta}}}\right]^{-1}_{[-\xi,-\xi]}\mathbf{F}_{[-\xi]}\right)^{-1}\left(\left([\mathbf{R}_{\hat{\boldsymbol{\theta}}}^{-1}]_{[\xi,\xi]}\right)^{-1}\left[\mathbf{R}_{\hat{\boldsymbol{\theta}}}^{-1}\mathbf{F}\right]_{[\xi]}\right)'.\end{aligned}$$

This estimator of σ^2 leads to the following desirable property:

$$\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\varepsilon}^2_{\mathrm{CV},i,\hat{\boldsymbol{\theta}}\hat{\sigma}^2}/\boldsymbol{\sigma}^2_{\mathrm{CV},i,\hat{\boldsymbol{\theta}},\hat{\sigma}^2} = 1.$$

An asymptotic normality and efficiency study of this estimator is proposed by [Bachoc, 2013]. For the numerical optimization of equations (1.41) or (1.42), it could be worthwhile to consider their partial derivatives. In a Simple kriging framework (see equations (1.36) and (1.35)), they can be deduced from the two following derivatives:

$$\begin{aligned}\left[\frac{\partial}{\partial\boldsymbol{\theta}}\boldsymbol{\sigma}^2_{\mathrm{CV},i,\mathrm{SK}}\right]_{i=1,\ldots,n} &= \sigma^2\frac{\mathrm{diag}(\mathbf{R}_{\boldsymbol{\theta}}^{-1}\frac{\partial\mathbf{R}_{\boldsymbol{\theta}}}{\partial\boldsymbol{\theta}}\mathbf{R}_{\boldsymbol{\theta}}^{-1})}{\mathrm{diag}(\mathbf{R}_{\boldsymbol{\theta}}^{-1})^2}, \\ \left[\frac{\partial}{\partial\boldsymbol{\theta}}\boldsymbol{\varepsilon}_{\mathrm{CV},i,\mathrm{SK}}\right]_{i=1,\ldots,n} &= \frac{-\mathbf{R}_{\boldsymbol{\theta}}^{-1}\frac{\partial\mathbf{R}_{\boldsymbol{\theta}}}{\partial\boldsymbol{\theta}}\mathbf{R}_{\boldsymbol{\theta}}^{-1}(\mathbf{z}^n-\mathbf{F}\boldsymbol{\beta})}{\mathrm{diag}(\mathbf{R}_{\boldsymbol{\theta}}^{-1})} \\ &\quad+\frac{\mathrm{diag}(\mathbf{R}_{\boldsymbol{\theta}}^{-1}\frac{\partial\mathbf{R}_{\boldsymbol{\theta}}}{\partial\boldsymbol{\theta}}\mathbf{R}_{\boldsymbol{\theta}}^{-1})\mathbf{R}_{\boldsymbol{\theta}}^{-1}(\mathbf{z}^n-\mathbf{F}\boldsymbol{\beta})}{\mathrm{diag}(\mathbf{R}_{\boldsymbol{\theta}}^{-1})^2}.\end{aligned}$$

1.4 Covariance kernels

Certainly one of the most important points of a Gaussian process regression is the choice of the covariance function $k(x,\tilde{x})$, $x,\tilde{x}\in Q\subset\mathbb{R}^d$ of the Gaussian process $Z(x)$ modeling the

objective function $z(x)$. We note that Q is a nonempty open set. We have seen in Section 1.1 that a covariance kernel $k(x, \tilde{x})$ has to be positive definite[5]. This ensures that the covariance matrix $\mathbf{K} = [k(x_i, x_j)]_{i,j=1,\dots,n}$ - also called the Gram matrix - is positive definite for any distinct $(x_i)_{i=1,\dots,n} \in Q$.

Moreover, the covariance kernel can also describe particular relations between $Z(x)$ and $Z(\tilde{x})$. As example, $k(x, \tilde{x})$ is said to be stationary if it is a function of $(x - \tilde{x})$. This means that it is invariant under any translation in the input space and that the relation between $Z(x)$ and $Z(\tilde{x})$ is uniquely determined by the distance between x and $\tilde{x}$. We describe these kernels in Subsection 1.4.2. Then, in Subsection 1.4.3 we present some non-stationary kernels. In particular, we present the fractional Brownian one that we use in Chapter 7. Finally, we deal with the eigenfunction analysis of $k(x, \tilde{x})$ in Subsection 1.4.4.

We highlight that it is easy to build new kernels from other ones thanks to the following properties ([Rasmussen and Williams, 2006]):

1. If $k_1(x, \tilde{x})$ and $k_2(x, \tilde{x})$ are covariance kernels then
$$k(x, \tilde{x}) = k_1(x, \tilde{x}) + k_2(x, \tilde{x})$$
or
$$k(x, \tilde{x}) = k_1(x, \tilde{x})k_2(x, \tilde{x})$$
is a covariance kernel.
2. If $f(x)$ is a deterministic function and $\tilde{k}(x, \tilde{x})$ a covariance kernel, then
$$k(x, \tilde{x}) = f(x)\tilde{k}(x, \tilde{x})f(\tilde{x})$$
is a covariance kernel.
3. If $k_1(x, \tilde{x})$ and $k_2(x, \tilde{x})$ are covariance kernels such that $\int k_1(x, z)k_2(z, \tilde{z})k_1(\tilde{z}, \tilde{x})\, dz\, d\tilde{z} < \infty$, then
$$k(x, \tilde{x}) = \int k_1(x, z)k_2(z, \tilde{z})k_1(\tilde{z}, \tilde{x})\, dz\, d\tilde{z}$$
is a covariance kernel. In particular, if $k_2(z, \tilde{z}) = \delta(z - \tilde{z})$ - $\delta(x)$ stands for the Dirac delta function - and the function $k_x : \tilde{x} \mapsto k(x, \tilde{x})$ is in $L^2(Q)$ for all $x \in Q \subset \mathbb{R}^d$, then we have $k(x, \tilde{x}) = \int k_1(x, u)k_1(u, \tilde{x})\, du$ which is the covariance kernel of the following Gaussian process
$$Z(x) = \int k_1(x, u)\, dW(u),$$
where $W(u)$ is a d-dimensional Wiener process (which is equivalently to say formally that $dW(u)/du$ is a Gaussian white noise).
4. If $k_1(x^1, \tilde{x}^1)$ and $k_2(x^2, \tilde{x}^2)$ are covariance kernels defined on different spaces $\mathcal{X}^1$ and $\mathcal{X}^2$, then
$$k(x, \tilde{x}) = k_1(x^1, \tilde{x}^1) + k_2(x^2, \tilde{x}^2)$$

[5] We recall that a kernel $k(x, \tilde{x})$ is positive definite if and only if for all $(a_i)_{i=1,\dots,N} \in \mathbb{R}$, $N \in \mathbb{N}^*$ and distinct $(x_i)_{i=1,\dots,N} \in Q$, we have $\sum_{i,j=1}^N a_i a_j k(x_i, x_j) \geq 0$ and $\sum_{i,j=1}^N a_i a_j k(x_i, x_j) = 0$ if and only if $a_i = 0$ for all $i = 1, \dots, N$.

or

$$k(x, \tilde{x}) = k_1(x^1, \tilde{x}^1)k_2(x^2, \tilde{x}^2)$$

is a covariance kernel defined on the product space $\mathcal{X}^1 \times \mathcal{X}^2$. We named as a tensorised kernel, a kernel of the form $k(x, \tilde{x}) = k_1(x^1, \tilde{x}^1)k_2(x^2, \tilde{x}^2)$.

1.4.1 Relations between Gaussian process regularities and covariance kernels

To emphasize the importance of the choice of $k(x, \tilde{x})$, let us introduce the concept of mean square differentiability (see [Cramer and Leadbetter, 1967]). First, for a fixed point $x^* \in Q$ a covariance kernel $Z(x)$ is said to be mean square continuous - or continuous in mean square - at x^* if:

$$\mathbb{E}\left[(Z(x^*) - Z(x))^2\right] \overset{x \to x^*}{\longrightarrow} 0.$$

Moreover, we have the following equality $\mathbb{E}\left[(Z(x^*) - Z(x))^2\right] = k(x^*, x^*) - k(x^*, x) + k(x, x) - k(x^*, x)$. Thus, $Z(x)$ is mean square continuous if and only if $k(x, \tilde{x})$ is continuous at $(x, \tilde{x}) = (x^*, x^*)$. Then, we consider at point $x = (x^1, \ldots, x^d)$ the Gaussian process:

$$Z_h^{(i)}(x) = \frac{Z(x + h\mathbf{e}_i) - Z(x)}{h},$$

with $h \in \mathbb{R} \setminus \{0\}$. The mean square derivative of $Z(x)$ in the ith direction is the Gaussian process $\partial Z(x)/\partial x^i$ such that

$$\mathbb{E}\left[\left(\frac{\partial Z(x)}{\partial x^i} - Z_h^{(i)}(x)\right)^2\right] \overset{h \to 0}{\longrightarrow} 0.$$

Furthermore, $\partial Z(x)/\partial x^i$ exists if and only if $k(x, \tilde{x})$ is twice differentiable at point $x = \tilde{x}$ and its covariance kernel is $\partial^2 k(x, \tilde{x})/\partial x^i \partial \tilde{x}^i$. We so have a tight relation between the regularity of the considered Gaussian process and the regularity of the covariance kernel $k(x, \tilde{x})$.

In fact, with more assumptions on $k(x, \tilde{x})$, we can have stronger results about the continuity of $Z(x)$. Let us consider the following definition (see [Cramer and Leadbetter, 1967]).

Definition 1.1 (continuous almost surely random processes)**.** Let us consider a random process $Z(x)$, $x \in Q \subset \mathbb{R}^d$, defined on $(\Omega_Z, \mathcal{F}, \mathbb{P}_Z)$ with values in $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$. Z is continuous almost surely on Q if for almost every $\omega \in \Omega_Z$, $x \mapsto Z_t(x, \omega)$ is continuous on Q.

This definition is of interest since it means that almost all paths of such random processes are continuous. Nonetheless, the definition of continuous almost surely random processes are not easy for general cases. The following theorem provides a useful criterion for establishing the existence of versions of stochastic processes with continuous sample paths (see [Oksendal, 1998]).

Theorem 1.1 (Kolmogorov-Chentsov). *Let $Z(x)$, $x \in Q \subset \mathbb{R}^d$, be a random process defined on $(\Omega_Z, \mathcal{F}, \mathbb{P}_Z)$ with values in $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$. Let us suppose that there are three positive constants $(\gamma, \varepsilon, c) \in (R_+^*)^3$ such that $\forall (x, \tilde{x}) \in Q^2$,*

$$\mathbb{E}\left[|Z(x) - Z(\tilde{x})|^\gamma\right] \leq c||x - \tilde{x}||^{d+\varepsilon}.$$

Then, there is $\tilde{Z}(x)$ a version of $Z(x)$ (i.e. for all $x \in Q$, $\mathbb{P}_Z\left(Z(x) = \tilde{Z}(x)\right) = 1$) such that

$$\forall \alpha \in [0, \varepsilon/\gamma), \mathbb{E}\left[\left(\sup_{\substack{(x,\tilde{x}) \in Q^2 \\ x \neq \tilde{x}}} \left(\frac{|\tilde{Z}(x) - \tilde{Z}(\tilde{x})|}{||x - \tilde{x}||^\alpha}\right)\right)^\gamma\right] < \infty.$$

This means that the sample of $\tilde{Z}(x)$ are almost surely Hölder continuous with Hölder exponent α.

Theorem 1.1 can easily be used in a Gaussian framework. This is highlighted in the following example.

Example 1.1. Let us consider a stationary Gaussian process $Z(x)$ with mean zero and covariance kernel given by $\sigma^2 r(h)$ where $h = x - \tilde{x}$, $x, \tilde{x} \in \mathbb{R}^d$. We have the following equality:

$$\mathbb{E}\left[(Z(x) - Z(\tilde{x}))^2\right] = 2\sigma^2(1 - r(h)).$$

Furthermore, from the following equality

$$\mathbb{E}\left[(Z(x) - Z(\tilde{x}))^{2n}\right] = \frac{(2n)!}{2^n n!}\sigma^{2n}(1 - r(h))^n$$

and the condition $r(h) \in \mathcal{C}^\varepsilon$, we can deduce that $\exists n > d/\varepsilon$ such that

$$\mathbb{E}\left[(Z(x) - Z(\tilde{x}))^{2n}\right] \leq \frac{(2n)!}{2^n n!}\sigma^{2n}\delta_\varepsilon^n ||h||^{n\varepsilon}.$$

Therefore, there is a version $\tilde{Z}(x)$ of $Z(x)$ which is α-Hölder continuous almost surely with $\alpha \in [0, \varepsilon/2)$.

Then, for the unidimensional case $x, \tilde{x} \in Q \subset \mathbb{R}$, a finer result is given by [Fernique, 1964] on $k(x, \tilde{x})$ so that $Z(x)$ is continuous a.s.. As stated in the theorem below, this condition is given in terms of the incremental variance $\mathbb{E}\left[(Z(x) - Z(\tilde{x}))^2\right]$.

Theorem 1.2 (Fernique's theorem). *If for $|x - \tilde{x}| \leq \varepsilon$, $x, \tilde{x} \in Q \subset \mathbb{R}$, there is a function ψ for which $\sqrt{\mathbb{E}\left[(Z(x) - Z(\tilde{x}))^2\right]} \leq \psi(x - \tilde{x})$, where ψ is nondecreasing on $[0, \varepsilon]$ and*

$$\int_0^\varepsilon \frac{\psi(u)}{u\sqrt{\log(1/u)}}\, du < \infty,$$

then $Z(x)$ has an almost sure continuous version.

The first proof of this theorem has been presented by [Dudley, 1967]. Then, several proofs have been suggested (see [Garsia, 1972] and [Marcus and Shepp, 1970]). In particular, [Marcus

and Shepp, 1970] present a proof for stationary covariance kernels $k(x,\tilde{x}) = k(x-\tilde{x})$, $x,\tilde{x} \in \mathbb{R}$. In that case, the condition simply becomes:

$$\int_0^{\varepsilon} \frac{\sqrt{k(0)-k(u)}}{u\sqrt{\log(1/u)}}\, du < \infty.$$

1.4.2 Stationary covariance functions

In this subsection we consider the case $Q = \mathbb{R}^d$ and we are interested in stationary covariance kernels. As presented previously, it corresponds to a covariance kernel $k(x,\tilde{x})$, $x,\tilde{x} \in \mathbb{R}^d$, function of $h = x - \tilde{x}$. We will use the notation $k(x,\tilde{x}) = k(h)$. These kernels are widely used in the framework of computer experiments.

One of their interesting properties is that the regularity of $k(h)$ at $h = 0$ determines the smoothness property of $Z(x)$ in mean square sense. Indeed a Gaussian process $Z(x)$ with covariance $k(h)$ is mean square continuous if k is continuous at $h = 0$. Furthermore, the Gaussian process $\partial^k Z(x)/\partial x^{i_1} \dots \partial x^{i_k}$ corresponding to the kth order partial mean square derivative of $Z(x)$ exists if and only if $\partial^{2k} k(h)/\partial^2 x^{i_1} \dots \partial^2 x^{i_k}$ exists and is finite at $h = 0$.

Another interesting property of stationary covariance kernels is that they can be represented as the Fourier transform of a positive measure as stated in the following theorem (see [Stein, 1999] p.24).

Theorem 1.3 (Bochner's theorem)**.** *For any continuous positive definite function $k(h)$ from $\mathbb{R}^d$ into $\mathbb{R}$, there exists a unique probability measure μ on $\mathbb{R}^d$ such that*

$$k(h) = \int_{\mathbb{R}^d} e^{2\pi i \langle w,h\rangle}\, d\mu(w).$$

We note that $\langle . \rangle$ stands for the scalar product. A proof of this theorem is given by [Gikhman and Skorokhod, 1974]. In the case where $\mu(dw)$ has a density $S(w)$, we call it the spectral density or power spectrum of $k(h)$ and we have

$$k(h) = \int_{\mathbb{R}^d} e^{2\pi i \langle w,h\rangle} S(w)\, dw$$

and

$$S(w) = \int_{\mathbb{R}^d} e^{-2\pi i \langle w,h\rangle} k(h)\, dh.$$

From the spectral density $S(w)$, we can define the following complex representation of the Gaussian process $Z(x)$ (see [Stein, 1999]):

$$Z(x) = \int \sqrt{S(w)} e^{2\pi i \langle w,x\rangle} \hat{n}_w\, dw, \tag{1.44}$$

where $\hat{n}_w$ is the Fourier transform of a Gaussian white noise. Moreover, we can estimate the integral (1.44) with the following sum:

$$Z(x) \approx \sum_{j=1}^{J} \sqrt{S(w_j)} e^{2\pi i \langle w_j,x\rangle} \hat{n}_{w_j} \Delta(j), \tag{1.45}$$

where $(w_j)_{j=1,\dots,J}$, $J \in \mathbb{N}$, is a tensorised grid covering the support of $S(w)$ and $\Delta(j)$ is the volume of the elementary hypercube of the grid associated with w_j. This representation can be used to compute samples of $Z(x)$ at points in $\mathbf{X} = \{x_1, \dots, x_l\}$ using the following equation:

$$(Z(x_l))_{l=1,\dots,n} = \sum_{j=1}^{J} \left[e^{i\langle w_j, x_l \rangle} \right]_{l=1,\dots,n} \left[\sqrt{S(w_j)} \hat{n}_{w_j} \right] \Delta(j). \tag{1.46}$$

The main advantage of this method is that it does not require the Cholesky's decomposition of the covariance matrix $\mathbf{K_X}$ of $Z(x)$ at points in $\mathbf{X}$ with respect to the kernel $k(h)$. Indeed, a commonly used method to sample $Z(x)$ at points in $\mathbf{X}$ is to consider the Cholesky decomposition of the covariance matrix $\mathbf{K_X} = [k(x_i, x_j)]_{i=1,\dots,l}$, $(x_i)_{i=1,\dots,l} \in \mathbf{X}$:

$$\mathbf{K_X} = \mathbf{L_X} \mathbf{L}'_{\mathbf{X}}.$$

Then, a realization of $Z(x)$ at $\mathbf{X}$ can be obtained by sampling a noise $\boldsymbol{\varepsilon}^l = [\varepsilon_i]_{i=1,\dots,l}$ where $(\varepsilon_i)_{i=1,\dots,l}$ are independent and identically distributed with respect to the Gaussian distribution $\mathcal{N}(0,1)$ and by considering the following equation:

$$Z(\mathbf{X}) = \mathbf{L_X} \boldsymbol{\varepsilon}^l.$$

Note that $Z(x)$ is considered to be zero-mean. Otherwise, we just have to add the term $\mathbf{M} = [m(x_i)]_{i=1,\dots,l}$ where $m(x)$ is the mean of $Z(x)$.

We emphasize that we can use a Fast Fourier transform to compute (1.46) and to sample $Z(x)$ by considering a tensorised regular grid. This allows for reducing the complexity of the method.

We present below some examples of stationary covariance kernels. For a more complete list, the reader is referred to [Stein, 1999] and [Rasmussen and Williams, 2006].

The Gaussian or Squared Exponential Covariance Function

The isotropic form of this kernel has already be presented in Section 1.1. It is defined as

$$k(h) = \exp\left(-\frac{1}{2} \frac{||h||^2}{\theta^2} \right), \tag{1.47}$$

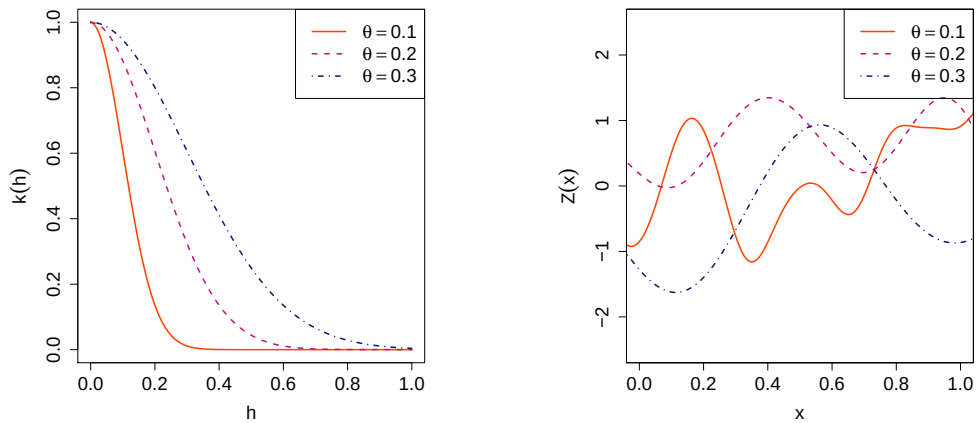
where the parameter θ is the correlation length or characteristic length-scale. Furthermore, it has the following power spectrum:

$$S(w) = \left(2\pi\theta^2\right)^{d/2} \exp\left(-2\pi^2\theta^2 ||w||^2\right).$$

This covariance function is smooth at $h = 0$ and thus corresponds to Gaussian processes which are infinitely mean square differentiable. Moreover, Theorem 1.1 implies that the corresponding Gaussian processes are infinitely differentiable almost surely. Thanks to the point 4. presented in the introduction of Section 1.4, we can easily define the anisotropic Gaussian covariance function as follows with $x = (x^1, \dots, x^d)$ and $\tilde{x} = (x^1, \dots, x^d)$

$$k(h) = \exp\left(-\frac{1}{2} \sum_{i=1}^{d} \frac{(x^i - \tilde{x}^i)^2}{\boldsymbol{\theta}_i^2} \right). \tag{1.48}$$

This kernel is widely used in kriging models but can be unrealistic as mentioned in [Stein, 1999] due to the strong regularity of the underlying Gaussian processes. A covariance function as the ν-Matérn one is in general more appropriate (see below). We illustrate in Figure 1.9 the shape of the 1-dimensional Gaussian kernel with different correlation lengths and examples of resulting Gaussian process realizations.



(a) (b)

Figure 1.9: Figure (a): the Gaussian kernel $k(h)$ in function of $h = x - \tilde{x}$ with different correlation lengths θ. Figure (b): examples of corresponding Gaussian process realizations.

The ν-Matérn covariance function

The isotropic ν-Matérn covariance function is defined as follow (see [Matérn, 1986])

$$k_\nu(h) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}||h||}{\theta} \right)^\nu K_\nu \left(\frac{\sqrt{2\nu}||h||}{\theta} \right), \tag{1.49}$$

where the parameter θ is the correlation length, the parameter ν is the regularity parameter, K_ν is the modified Bessel function ([Abramowitz and Stegun, 1965] sec 9.6), and Γ is the Euler-Gamma function. It has the following power spectrum:

$$S(w) = \frac{2^d \pi^{d/2} \Gamma(\nu + d/2)(2\nu)^\nu}{\Gamma(\nu)\theta^{2\nu}} \left(\frac{2\nu}{\theta^2} + 4\pi^2 ||w||^2 \right)^{-(\nu+d/2)}.$$

A Gaussian process $Z(x)$ with a ν-Matérn covariance kernel is ν-Hölder continuous in mean square and ν'-Hölder continuous almost surely $\forall \nu' < \nu$. Furthermore, for $\nu = p + 1/2$ with $p \in \mathbb{N}$, the ν-Matérn kernel has the following form

$$k_{\nu=p+1/2}(h) = \exp\left(-\frac{\sqrt{2\nu}||h||}{\theta} \right) \frac{\Gamma(p+1)}{\Gamma(2p+1)} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{\sqrt{8\nu}||h||}{\theta} \right)^{p-i}.$$

In a Gaussian process framework, two popular ν-Matérn covariance kernels are the ones for $\nu = 3/2$ and $\nu = 5/2$:

$$k_{\nu=3/2}(h) = \left(1 + \frac{\sqrt{3}||h||}{\theta}\right) \exp\left(-\frac{\sqrt{3}||h||}{\theta}\right),$$

$$k_{\nu=5/2}(h) = \left(1 + \frac{\sqrt{5}||h||}{\theta} + \frac{5}{3}\frac{||h||^2}{\theta^2}\right) \exp\left(-\frac{\sqrt{5}||h||}{\theta}\right).$$

Another special case is the one with $\nu = 1/2$ which leads to the so-called exponential covariance function

$$k_{\nu=1/2}(h) = \exp\left(-\frac{||h||}{\theta}\right).$$

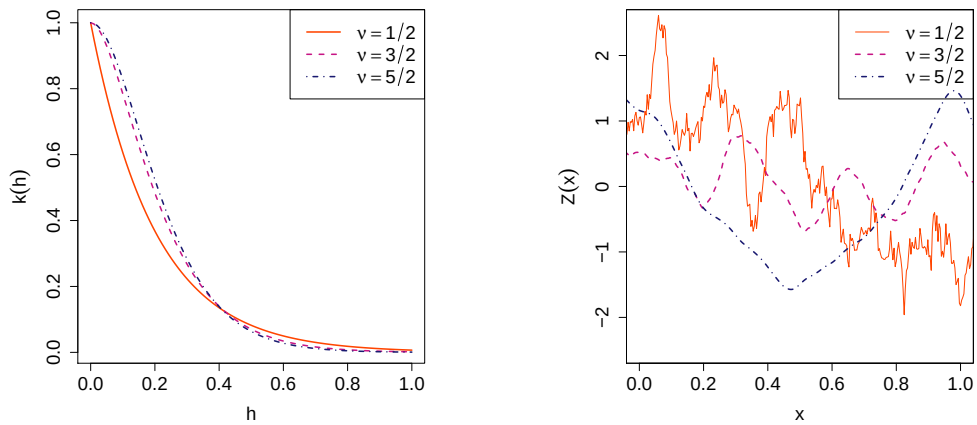
This corresponds to the covariance of an Ornstein-Uhlenbeck process ([Uhlenbeck and Ornstein, 1930]). We can also consider anisotropic Matérn covariance kernels as follows with $x = (x^1, \ldots, x^d)$ and $\tilde{x} = (\tilde{x}^1, \ldots, \tilde{x}^d)$

$$k(x, \tilde{x}) = \prod_{i=1}^{d} k_{\nu^i, \boldsymbol{\theta}^i}(x^i - \tilde{x}^i),$$

where

$$k_{\nu^i, \boldsymbol{\theta}^i}(x^i - \tilde{x}^i) = \frac{2^{1-\nu^i}}{\Gamma(\nu^i)} \left(\frac{\sqrt{2\nu^i}|x^i - \tilde{x}^i|}{\boldsymbol{\theta}^i}\right)^{\nu^i} K_{\nu^i}\left(\frac{\sqrt{2\nu^i}|x^i - \tilde{x}^i|}{\boldsymbol{\theta}^i}\right).$$

We illustrate in Figure 1.10 the shape of the 1-dimensional ν-Matérn kernel with different regularity parameters and a correlation length fixed to $\theta = 0.2$. Examples of resulting Gaussian process realizations are given.



(a) (b)

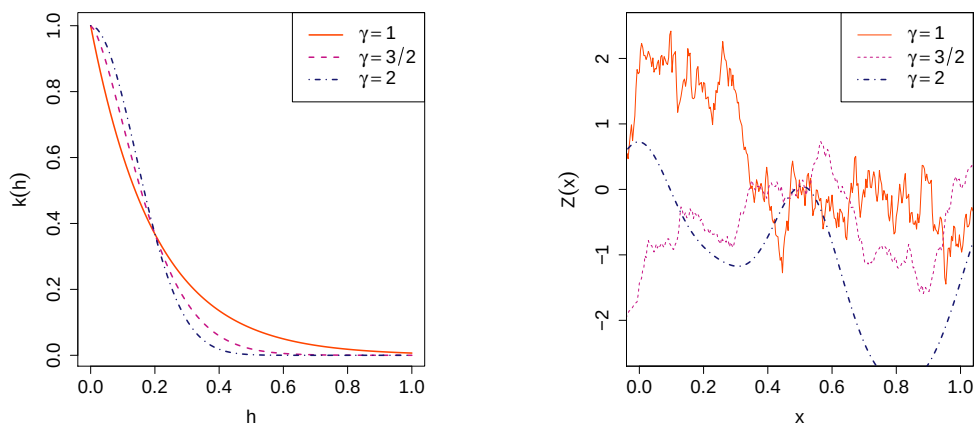
Figure 1.10: Figure (a): the ν-Matérn kernel $k(h)$ in function of $h = x - \tilde{x}$ with a fixed correlation length $\theta = 0.2$ and different regularity parameters ν. Figure (b): examples of corresponding Gaussian process realizations.

The γ-exponential covariance function

The isotropic γ-exponential covariance function is defined as follow

$$k(h) = \exp\left(-\left(\frac{||h||}{\theta}\right)^{\gamma}\right), \quad 0 < \gamma \leq 2.$$

The positive definiteness of this kernel is proved in [Schoeneberg, 1938]. Furthermore, for $\gamma < 2$ the corresponding Gaussian processes are not differentiable in mean square sense whereas for $\gamma = 2$ they are infinitely differentiable. Thus, the use of this kernel for practical applications can be difficult to justify. We illustrate in Figure 1.11 the shape of the 1-dimensional γ-exponential kernel with different parameters γ and a correlation length fixed to $\theta = 0.2$. Examples of resulting Gaussian process realizations are given.



(a) (b)

Figure 1.11: Figure (a): the γ-exponential kernel $k(h)$ in function of $h = x - \tilde{x}$ with a fixed correlation length $\theta = 0.2$ and different parameters ν. Figure (b): examples of corresponding Gaussian process realizations.

1.4.3 Non-stationary covariance kernels

There are many ways to construct non-stationary covariance kernels. As an example, as presented in [Rasmussen and Williams, 2006] p89 Sec.4.4.2 we can cite the dot product covariance functions which are invariant to a rotation on the inputs about the origin. These kernels are commonly used in the field of Geostatistics. Another interesting example is the covariance function presented in [Gibbs, 1997] which allows for varying the length-scale parameter $\boldsymbol{\theta}(x)$ in function of x. It is defined as follows

$$k(x, \tilde{x}) = \prod_{i=1}^{d} \left(\frac{2\boldsymbol{\theta}_i(x)\boldsymbol{\theta}_i(\tilde{x})}{\boldsymbol{\theta}_i^2(x) + \boldsymbol{\theta}_i^2(\tilde{x})}\right)^{1/2} \exp\left(-\sum_{i=1}^{d} \frac{(x^i - \tilde{x}^i)^2}{\boldsymbol{\theta}_i^2(x) + \boldsymbol{\theta}_i^2(\tilde{x})}\right),$$

where $\boldsymbol{\theta}_i(x)$ are positive functions on $x = (x^1, \ldots, x^d)$. In Chapter 7 we use the following kernel:

$$k(x, \tilde{x}) = x^{2H} + \tilde{x}^{2H} - |x - \tilde{x}|^{2H},$$

with $H \in (0,1)$. It corresponds the the kernel of a fractional Brownian motion with Hurst parameter H. This Gaussian process is mean square continuous and nowhere mean square differentiable. Nevertheless, it is Hölder continuous with exponent $H-\varepsilon$, $\forall \varepsilon > 0$. Furthermore, for $H = 1/2$ it corresponds to the Brownian motion. We illustrate in Figure 1.12 some realizations of fractional Brownian motions with different Hurst parameters.

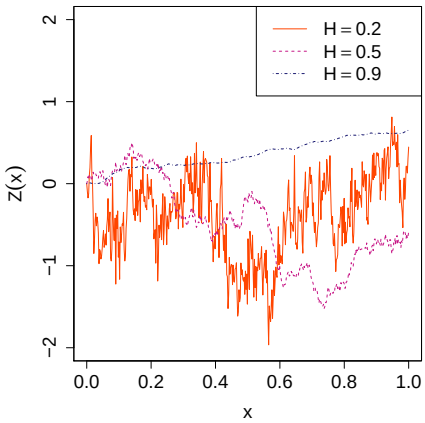


Figure 1.12: Realization of fractional Brownian motions with different Hurst parameters H.

1.4.4 Eigenfunction analysis

We saw in Theorem 1.3 that stationary covariance kernels can have a spectral representation through the Fourier transform of a probability measure. We discuss in this subsection about an interesting theorem which allows for having a spectral decomposition of covariance kernels $k(x, \tilde{x})$ thanks to its eigenvalues and eigenfunctions decomposition. Let us consider this theorem below. It is an extension of the Mercer's theorem [Mercer, 1909] with a probability measure μ and a continuous positive kernel $k(x, \tilde{x})$ satisfying the property $\sup_{x \in Q} k(x,x) < \infty$ with Q an nonempty open subset of $\mathbb{R}^d$ (see [König, 1986] and [Ferreira and Menegatto, 2009]).

Theorem 1.4 (Mercer's theorem)**.** *Let us consider a continuous positive kernel $k(x, \tilde{x})$, $x, \tilde{x} \in Q \subset \mathbb{R}^d$ - such that $\sup_{x \in Q} k(x,x) < \infty$ and Q is an nonempty open set - and a probability measure μ on Q. The kernel $k(x, \tilde{x})$ can be written as follows*

$$k(x, \tilde{x}) = \sum_{p \geq 0} \lambda_p \phi_p(x) \phi_p(\tilde{x}),$$

where $\phi_p(x) \in L^2_\mu(Q)$ are the eigenfunctions of the trace class integral operator

$$(T_k f)(x) = \int k(x,u) f(u)\, d\mu(u),$$

and $(\lambda_p)_{p \geq 0}$ the corresponding nonnegative sequence of eigenvalues sorted in decreasing order. Furthermore, $(\phi_p(x))_{p \geq 0}$ is an orthonormal basis of $L^2_\mu(Q)$ and $\phi_p(x)$ are continuous for all p such that $\lambda_p \neq 0$.

We intensively use this theorem in Chapter 7 and Chapter 8. In particular, we will see that the regularity of a Gaussian process is related to the rate of convergence of its eigenvalues $(\lambda_p)_{p\geq 0}$. Furthermore, we always consider in the manuscript that μ is a probability measure such that $\mu(U) > 0$ for any nonempty open subset U of $Q \subset \mathbb{R}^d$.

We will talk in these chapters about degenerate and non-degenerate kernels. To be clear in the remainder of the manuscript, we define this notion below

Definition 1.2. Let us consider a covariance kernel $k(x, \tilde{x})$ and its Mercer's decomposition

$$k(x, \tilde{x}) = \sum_{p\geq 0} \lambda_p \phi_p(x)\phi_p(\tilde{x}).$$

If $k(x, \tilde{x})$ has a infinite sequence $(\lambda_p)_{p\geq 0}$ of non-zero eigenvalues, then it is called a non-degenerate kernel. Otherwise, if it has a finite number of non-zero eigenvalues, it is called a degenerate kernel.

We see in Chapter 7 that the degenerate or non-degenerate property of a covariance kernel has a strong impact on the rate of convergence of the generalization error of a Gaussian process regression.

Right now, let us present some particular results about this decomposition.

1. By definition, the function $\phi_p(x)$ satisfies the following equality

$$\lambda_p \phi_p(x) = \int k(x, u)\phi_p(u)\, d\mu(u).$$

2. The orthonormal property of $(\phi_p(x))_{p\geq 0}$ implies that

$$\int \phi_q(x)\phi_p(x)\, d\mu(x) = \delta_{p=q},$$

   where δ stands for the Kronecker symbol.

3. We have the following equality:

$$\int k(x, x) d\mu(x) = \sum_{p\geq 0} \lambda_p < +\infty,$$

   This shows that the operator T_k is trace class with

$$\mathrm{tr}(T_k) = \sum_{p\geq 0} \lambda_p.$$

4. For covariance kernels such that $k(x, x) = \sigma^2\ \forall x$, we have $\forall x$:

$$\sigma^2 = \sum_{p\geq 0} \lambda_p \phi_p(x)^2 = \sum_{p\geq 0} \lambda_p,$$

   since $\int \sigma^2\, d\mu(u) = \sigma^2$.

Furthermore, with the Mercer's decomposition, we have the analogous of the complex representation of a Gaussian process as stated below.

Theorem 1.5 (Karhunen-Loeve decomposition)**.** *Let us consider a Gaussian process $Z(x)$ with covariance kernel $k(x, \tilde{x})$ and the following Mercer's decomposition*

$$k(x, \tilde{x}) = \sum_{p \geq 0} \lambda_p \phi_p(x) \phi_p(\tilde{x}).$$

Then, $Z(x)$ can be represented through the following form

$$Z(x) = \sum_{p \geq 0} \sqrt{\lambda_p} \phi_p(x) Z_p,$$

where $(Z_p)_{p \geq 0}$ are independent and identically distributed random variables with distribution $\mathcal{N}(0, 1)$ defined as

$$\sqrt{\lambda_p} Z_p = \int Z(u) \phi_p(u) \, d\mu(u),$$

An important property of the Karhunen-Loeve decomposition is that it provides the best spectral decomposition of a Gaussian process in the sense that it minimizes the total mean squared error resulting of its truncation as stated in the following proposition.

Proposition 1.1. *Let us consider any orthonormal basis $(\psi_p(x))_{p \geq 0}$ of $L^2_\mu(Q)$ and the following decomposition of $Z(x)$*

$$Z(x) = \sum_{p \geq 0} \left(\int Z(u) \psi_p(u) \, d\mu(u) \right) \psi_p(x).$$

Then, for a given $\bar{p} > 0$, the basis minimizing

$$\int \mathbb{E} \left[\left(\sum_{p \geq \bar{p}} \left(\int Z(u) \psi_p(u) \, d\mu(u) \right) \psi_p(x) \right)^2 \right] d\mu(x)$$

is given by $(\phi_p(x))_{p \geq 0}$, i.e. the one of the Karhunen-Loeve decomposition. We note that the functions $\phi_p(x)$ for $p \geq 0$ are unique if and only if the values of ϕ_p for $p \geq 0$ are positive and distinct.

Proof. Let us consider $(\psi_p(x))_{p \geq 0}$ an orthonormal basis of $L^2_\mu(Q)$ and let us denote by

$$\varepsilon^2(x) = \mathbb{E} \left[\left(\sum_{p \geq \bar{p}} \left(\int Z(u) \psi_p(u) \, d\mu(u) \right) \psi_p(x) \right)^2 \right].$$

A direct calculation gives that

$$\varepsilon^2(x) = \sum_{p, q \geq \bar{p}} \psi_p(x) \psi_q(x) \int \int k(u, v) \psi_p(u) \psi_q(v) \, d\mu(u) \, d\mu(v).$$

Then, by integrating we find that $\bar{\varepsilon}^2 = \int \varepsilon^2(x)\, d\mu(x)$ equals:

$$\bar{\varepsilon}^2 = \sum_{p \geq \bar{p}} \int \int k(u,v)\psi_p(u)\psi_p(v)\, d\mu(u)\, d\mu(v).$$

Thus, we want to minimize $\bar{\varepsilon}^2$ with the constraint of normalized $\psi_p(x)$. Let us consider the Lagrangien formulation of this problem

$$\sum_{p \geq \bar{p}} \int \int k(u,v)\psi_p(u)\psi_p(v)\, d\mu(u)\, d\mu(v) - \gamma_p \left(\int \psi_p(u)\psi_p(u)\, d\mu(u) - 1 \right),$$

where γ_p are the Lagrangian multipliers. By differentiation with respect to $\psi_p(u)$ and setting the derivatives equal to 0, we find that for $p \geq \bar{p}$

$$\int k(u,v)\psi_p(v)\, d\mu(v) - \gamma_p \psi_p(u) = 0,$$

i.e. $\psi_p(x) = \phi_p(x)$ and $\gamma_p = \lambda_p$ for all $p \geq \bar{p}$. □

However, contrary to the complex representation, closed form expressions for such a spectral decomposition is rarely available. The Nyström procedure can be used to numerically approximate the Karhunen-Loeve spectral decomposition of a Gaussian process. This procedure being based on a quadrature numerical integration, it could be an issue to perform it in high dimension except for tensorised kernels. Indeed, in that case, the approximation can be performed by considering d 1-dimensional numerical integrations.

First, let us consider the Karhunen-Loeve decomposition of the 1-dimensional Gaussian process $Z(x)$, $x \in [0,1]$:

$$Z(x) = \sum_{p \geq 0} \sqrt{\lambda_p}\phi_p(x) Z_p. \tag{1.50}$$

To evaluate the Karhunen-Loeve spectral decomposition of $Z(x)$ we have to solve the following eigenproblem $\forall p \in \mathbb{N}$:

$$\lambda_p \phi_p(x) = \int_{[0,1]} k(x,u)\phi_p(u)\, d\mu(u). \tag{1.51}$$

Let us consider that the measure μ has a density $f(x)$. We can consider the following numerical integration:

$$\lambda_p \phi_p(x) = \int_{[0,1]} k(x,u)\phi_p(u) f(u)\, du \approx \frac{1}{N} \sum_{i=1}^{N} k(x,x_i)\phi_p(x_i) f(x_i), \tag{1.52}$$

where $(x_i)_{i=1,\ldots,N}$ is a regular grid on $[0,1]$ (the extension to any intervals $[a,b]$ is straightforward). Then, by considering the eigenfunctions $\phi_p(x)$ at points $(x_i)_{i=1,\ldots,N}$, we obtain the following eigenproblem:

$$\lambda_p^R \mathbf{\Phi}_p = \mathbf{K}_N \mathbf{\Phi}_p, \tag{1.53}$$

where $\mathbf{\Phi}_p' = (\phi(x_1), \ldots, \phi(x_N))$, $\lambda_p^R = \lambda_p N$ and $[\mathbf{K}_N]_{i,j} = k(x_j, x_i) f(x_i)$. Therefore, λ_p^R / N is an estimator for λ_p for $i = 1, \ldots, N$. It can be shown that λ_p^R / N converges to λ_p when $N \to \infty$ [Baker, 1977].

Then, the Nyström method for approximating the pth eigenfunction [Baker, 1977] is given by:

$$\phi_p(x) \approx \frac{1}{\lambda_p^R} \mathbf{k}'(x) \mathbf{\Phi}_p, \tag{1.54}$$

where $\mathbf{k}'(x) = (k(x, x_i), \ldots, k(x, x_N))$. Thus, given a point x, we can sample $Z(x)$ by considering the following truncated series:

$$Z(x) \approx \sum_{p \leq N_p} \frac{\mathbf{k}'(x) \mathbf{\Phi}_p}{\sqrt{\lambda_p^R N}} Z_p. \tag{1.55}$$

Second, let us consider the following d-dimensional Gaussian process, $x \in [0, 1]^d$:

$$Z(x) \sim \mathrm{GP}(0, \prod_{i=1}^{d} k_i(x^i, \tilde{x}^i)). \tag{1.56}$$

We note that $Z(x)$ has a d-dimensional tensorised kernel. We have the following Karhunen-Loeve representation of $Z(x)$:

$$Z(x) = \sum_{p_1, \ldots, p_d \geq 0} \prod_{i=1}^{d} \sqrt{\lambda_{p_i}} \phi_{p_i}(x) Z_{p_1, \ldots, p_d}, \tag{1.57}$$

where λ_{p_i} and $\phi_{p_i}(x)$ are respectively the eigenvalues and eigenfunctions of the kernel $k_i(x, \tilde{x})$. Thus, to compute a realization of $Z(x)$ we just have to consider the Nyström approximation of each kernel $k_i(x, \tilde{x})$ for $i = 1, \ldots, d$ (i.e. it corresponds to d 1-dimensional numerical integrations).

1.5 Kriging models: two other approaches

The kriging equations were presented in Section 1.2 through a Bayesian approach. Nonetheless, it was not the original approach suggested by [Krige, 1951]. In Subsection 1.5.1 we present this approach based on a linear formulation as presented in Equation (1.5). In particular, we will see that it leads to the same model as the simple and universal kriging one. We use this result in Chapter 7 to show asymptotic results on the predictive variance in a noisy kriging framework. Then, in Subsection 1.5.2 we present a closely related tool coming from the regularization theory in a reproducing kernel Hilbert space.

1.5.1 The Best Linear Unbiased Predictor

We present in this subsection the concept of the Best Linear Unbiased Predictor (BLUP). We still consider the problem of predicting a random variable $Z(x)$, $x \in Q \subset \mathbb{R}^d$ from a vector of observations $\mathbf{z}^n$ at points $\mathbf{D}$. We recall that $Z(x)$ is a Gaussian process of mean $\mathbf{f}'(x)\boldsymbol{\beta}$ and covariance structure $k(x, \tilde{x})$ modeling the objective function $z(x)$. First of all, we consider the parameter $\boldsymbol{\beta}$ known and equal to zero. Let us consider the linear predictor:

$$\hat{Z}(x) = a_0 + \mathbf{a}' \mathbf{Z}^n. \tag{1.58}$$

We are looking for an unbiased predictor, i.e. $\mathbb{E}\left[\hat{Z}(x)\right] = \mathbb{E}\left[Z(x)\right]$. The unbiased property leads to $a_0 = 0$. Then, we want to determine the best linear unbiased predictor with respect to the mean squared errors loss function. Thus, the problem consists in finding the coefficient $\mathbf{a}$ solving

$$\min_{\mathbf{a}} \mathbb{E}\left[\left(\mathbf{a}'\mathbf{Z}^n - Z(x)\right)^2\right]. \tag{1.59}$$

We have

$$\mathbb{E}\left[\left(\mathbf{a}'\mathbf{Z}^n - Z(x)\right)^2\right] = k(x,x) + \mathbf{a}'\mathbf{K}\mathbf{a} - 2\mathbf{a}'\mathbf{k}(x),$$

which is minimal for $\mathbf{a} = \mathbf{k}'(x)\mathbf{K}^{-1}$. Thus, the BLUP is given by:

$$\hat{Z}(x) = \mathbf{k}'(x)\mathbf{K}^{-1}\mathbf{Z}^n \tag{1.60}$$

and its mean squared error (MSE) is given by

$$\mathrm{MSE}_{\hat{Z}}(x) = k(x,x) - \mathbf{k}'(x)\mathbf{K}^{-1}\mathbf{k}(x). \tag{1.61}$$

Considering the observed values $\mathbf{z}^n$, equations (1.60) and (1.61) with $k(x,\tilde{x}) = \sigma^2 r(x,\tilde{x})$ are identical to the ones of the Simple kriging (1.3) and (1.4). Furthermore, the Gaussian property of the underlying stochastic process $Z(x)$ implies that the predictive distributions of the two approaches are identical.

Now, let us assume that $\boldsymbol{\beta}$ is unknown and consider an unbiased linear predictor of the form

$$\hat{Z}(x) = \mathbf{a}'\mathbf{Z}^n. \tag{1.62}$$

The unbiased property imposes the constraint $\mathbf{a}'\mathbf{F}\boldsymbol{\beta} = \mathbf{f}'(x)\boldsymbol{\beta}$, $\forall\boldsymbol{\beta}$, i.e. $\mathbf{F}'\mathbf{a} = \mathbf{f}(x)$. Thus, the goal is to solve the following constraint optimization problem

$$\begin{cases} \min_{\mathbf{a}} \mathbb{E}\left[(\mathbf{a}'\mathbf{Z}^n - Z(x))^2\right] \\ \mathbf{F}'\mathbf{a} = \mathbf{f}(x) \end{cases}$$

or equivalently

$$\begin{cases} \min_{\mathbf{a}} k(x,x) + \mathbf{a}'\mathbf{K}\mathbf{a} - 2\mathbf{a}'\mathbf{k}(x) \\ \mathbf{F}'\mathbf{a} = \mathbf{f}(x) \end{cases}. \tag{1.63}$$

We can use the method of Lagrange multipliers to minimize the quadratic form in (1.63) subject to $\mathbf{F}'\mathbf{a} = \mathbf{f}(x)$. We aim to find $(\mathbf{a}, \boldsymbol{\lambda}) \in \mathbb{R}^{n+p}$ minimizing the Lagrangian formulation

$$k(x,x) + \mathbf{a}'\mathbf{K}\mathbf{a} - 2\mathbf{a}'\mathbf{k}(x) + 2\boldsymbol{\lambda}'(\mathbf{F}'\mathbf{a} - \mathbf{f}(x)).$$

We can calculate the gradients with respect to $(\mathbf{a}, \boldsymbol{\lambda})$ and set it equal to zero. We find the following system of equations

$$\begin{cases} \mathbf{F}'\mathbf{a} - \mathbf{f}(x) = 0 \\ \mathbf{K}\mathbf{a} - \mathbf{k}(x) + \mathbf{F}\boldsymbol{\lambda} = 0 \end{cases},$$

which leads to

$$\begin{aligned}\begin{pmatrix}\boldsymbol{\lambda}\\ \mathbf{a}\end{pmatrix} &= \begin{pmatrix}0 & \mathbf{F}'\\ \mathbf{F} & \mathbf{K}\end{pmatrix}^{-1}\begin{pmatrix}\mathbf{f}(x)\\ \mathbf{k}(x)\end{pmatrix}\\ &= \begin{pmatrix}-\mathbf{Q} & \mathbf{Q}\mathbf{F}'\mathbf{K}^{-1}\\ \mathbf{K}^{-1}\mathbf{F}\mathbf{Q} & \mathbf{K}^{-1}-\mathbf{K}^{-1}\mathbf{F}\mathbf{Q}\mathbf{F}'\mathbf{K}^{-1}\end{pmatrix}\begin{pmatrix}\mathbf{f}(x)\\ \mathbf{k}(x)\end{pmatrix},\end{aligned}$$

with $\mathbf{Q} = (\mathbf{F}'\mathbf{K}^{-1}\mathbf{F})^{-1}$. Therefore, we find that

$$\mathbf{a} = \mathbf{K}^{-1}\mathbf{F}\mathbf{Q}\mathbf{f}(x) + \left(\mathbf{K}^{-1} - \mathbf{K}^{-1}\mathbf{F}\mathbf{Q}\mathbf{F}'\mathbf{K}^{-1}\right)\mathbf{k}(x)$$

and the resulting predictor is

$$\hat{Z}(x) = \mathbf{f}'(x)\hat{\boldsymbol{\beta}} + \mathbf{k}'(x)\mathbf{K}^{-1}\left(\mathbf{Z}^n - \mathbf{F}\hat{\boldsymbol{\beta}}\right), \tag{1.64}$$

with $\hat{\boldsymbol{\beta}} = (\mathbf{F}'\mathbf{K}^{-1}\mathbf{F})^{-1}\mathbf{F}'\mathbf{K}^{-1}\mathbf{Z}^n$. The MSE of the predictor $\hat{Z}(x)$ in (1.64) is then given by

$$\begin{aligned}\mathrm{MSE}_{\hat{Z}}(x) &= k(x,\tilde{x}) - \mathbf{k}'(x)\mathbf{K}^{-1}\mathbf{k}(x)\\ &\quad +(\mathbf{f}'(x) - \mathbf{k}'(x)\mathbf{K}^{-1}\mathbf{F})(\mathbf{F}'\mathbf{K}^{-1}\mathbf{F})^{-1}(\mathbf{f}'(x) - \mathbf{k}'(x)\mathbf{K}^{-1}\mathbf{F})'\end{aligned}. \tag{1.65}$$

Equations (1.64) and (1.65) with $k(x,\tilde{x}) = \sigma^2 r(x,\tilde{x})$ are identical to the ones of the Universal kriging (1.20) and (1.21). Considering the Gaussian property of the underlying stochastic process $Z(x)$, it gives that the two approaches are equivalent.

1.5.2 Regularization in a Reproducing Kernel Hilbert Space

In this subsection, we present how the Gaussian process regression theory can be related to the regularization problem in a Reproducing Kernel Hilbert Space (RKHS). First of all, we introduce some concepts about RKHS and then we present the famous representer theorem given a general form for the solution of a regularization problem in a RKHS. The forthcoming developments were inspired by the book of [Wahba, 1990] and [Rasmussen and Williams, 2006]. We present here a brief introduction to RKHS, for more detail about them, the reader could refer to the article of [Aronszajn, 1950] or the book of [Wahba, 1990]. Furthermore, for a deep presentation of regularization in a RKHS and the correspondence with Gaussian process regression, we refer to the thesis of [Vazquez, 2005] Chapter 3.

Covariance functions and reproducing kernels in Hilbert spaces

Foremost, we define a general index set $\mathcal{X}$. Examples of $\mathcal{X}$ can be various (e.g. $\mathcal{X} = \{1,\dots,N\}$, $\mathcal{X} = [0,1]$, $\mathcal{X} = \mathcal{S}$ with $\mathcal{S}$ the unit sphere,...). For our purpose, we always consider that $\mathcal{X} \subset \mathbb{R}^d$ but the results presented in this paragraph remain true for more general $\mathcal{X}$. We saw

in Section 1.1 that a kernel $k(x, \tilde{x})$ with $x, \tilde{x} \in \mathcal{X}$ is positive definite if for any $a_1, \ldots, a_n \in \mathbb{R}$, and distinct $x_1, \ldots, x_n \in \mathcal{X}$, $n \in \mathbb{N}^*$, we have

$$\sum_{i,j=1}^{n} a_i a_j k(x_i, x_j) \geq 0.$$

and $\sum_{i,j=1}^{n} a_i a_j k(x_i, x_j) = 0$ if and only if $a_i = 0$ for all $i = 1, \ldots, n$. Furthermore, we can define a Gaussian process $Z(x)$ with covariance structure $k(x, \tilde{x})$ if it fulfills the positive definiteness property. We will see in the forthcoming developments that we can associate the kernel $k(x, \tilde{x})$ to a RKHS. Let us consider the following definition:

Definition 1.3 (Reproducing Kernel Hilbert Space)**.** Let $\mathcal{H}$ be a Hilbert space of real functions f defined on an index set $\mathcal{X}$. Then $\mathcal{H}$ is called a reproducing kernel Hilbert space endowed with an inner product $\langle ., . \rangle_{\mathcal{H}}$ and norm $||f||_{\mathcal{H}} = \sqrt{\langle f, f \rangle_{\mathcal{H}}}$ if there exists a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with the following properties:

1. For every $x \in \mathcal{X}$, the function $k_x : \tilde{x} \mapsto k(x, \tilde{x})$ belongs to $\mathcal{H}$.
2. $k(x, \tilde{x})$ has the reproducing property $\langle k_x, f \rangle_{\mathcal{H}} = f(x)$, $\forall f \in \mathcal{H}$.
3. $\forall x \in \mathcal{X}$ the evaluation functional $k_x(\tilde{x})$ is a bounded linear functional, i.e. $\exists M_x$ such that $\forall f \in \mathcal{H}$, $|f(x)| \leq M_x ||f||_{\mathcal{H}}$.

The form $k_x(.)$ for the evaluation functional comes from the Riesz representation theorem. We note that we have also the property $\langle k_x, k_{\tilde{x}} \rangle_{\mathcal{H}} = k(x, \tilde{x})$. For a given RKHS, the representer $k_x(.)$ of evaluation at x is unique. The converse is true as presented in the following theorem [Aronszajn, 1950]:

Theorem 1.6 (Moore-Aronszajn theorem)**.** *To every RKHS there corresponds a unique positive definite function $k(x, \tilde{x})$ called the reproducing kernel and conversely, given a positive definite function $k(x, \tilde{x})$ we can construct a unique RKHS of real-valued functions on $\mathcal{X}$ with $k(x, \tilde{x})$ as its reproducing kernel.*

Proof. If $\mathcal{H}$ is a RKHS, then the reproducing kernel is $k(x, \tilde{x}) = \langle k_x, k_{\tilde{x}} \rangle_{\mathcal{H}}$, where for each x, $\tilde{x}$, k_x and $k_{\tilde{x}}$ are the representers of evaluation at x and $\tilde{x}$. Furthermore, $k(x, \tilde{x})$ is positive definite since, for any distinct $x_1, \ldots, x_n \in \mathcal{X}$, $a_1, \ldots, a_n \in \mathbb{R}$, $n \in \mathbb{N}^*$, we have:

$$\begin{aligned} \sum_{i,j=1}^{n} a_i a_j k(x_i, x_j) &= \sum_{i,j=1}^{n} a_i a_j \langle k_{x_i}, k_{x_j} \rangle_{\mathcal{H}} \\ &= ||\sum_{i=1}^{n} a_i k_{x_i}||_{\mathcal{H}}^2 \geq 0. \end{aligned}$$

and $||\sum_{i=1}^{n} a_i k_{x_i}||_{\mathcal{H}}^2 = 0$ if and only if $a_i = 0$ for all $i = 1, \ldots, n$. Conversely, given $k(x, \tilde{x})$ we construct $\mathcal{H} \equiv \mathcal{H}_k$ as follows. For each fixed $x \in \mathcal{X}$, denote by k_x the real-valued function such that

$$k_x(.) = k(x, .).$$

Then, construct a manifold by taking all finite linear combinations of the form

$$\sum_{i=1}^{n} a_i k_{x_i},$$

for all choices of n, $a_1, \ldots, a_n \in \mathbb{R}$, $x_1, \ldots, x_n \in \mathcal{X}$ with the inner product

$$\langle \sum_{i=1}^{n} a_i k_{x_i}, \sum_{i=1}^{n} \tilde{a}_i k_{\tilde{x}_i} \rangle_{\mathcal{H}} = \sum_{i,j=1}^{n} a_i \tilde{a}_j \langle k_{x_i}, k_{x_j} \rangle_{\mathcal{H}} = \sum_{i,j=1}^{n} k(x_i, x_j) a_i \tilde{a}_j.$$

The inner-product is well-defined since $k(x, \tilde{x})$ is positive definite. Furthermore, for any f such that $f(x) = \sum_{i=1}^{n} a_i k_{x_i}(x)$ we have $\langle k_x, f \rangle_{\mathcal{H}} = f(x)$. In this linear manifold we have

$$|f_n(x) - f(x)| = |\langle f_n - f, k_x \rangle_{\mathcal{H}}| \leq ||f_n - f||_{\mathcal{H}} ||k_x||_{\mathcal{H}}.$$

Thus, the norm convergence implies the point wise convergence and we can adjoin to this manifold all the limits of Cauchy sequences of functions in the manifold. The resulting Hilbert space is the RKHS $\mathcal{H}$ with the reproducing kernel $k(x, \tilde{x})$. □

In the Hilbert space L^2 with the inner product $\langle f, g \rangle_{L^2} = \int f(x) g(x)\, dx$, the dirac delta function is the representer of evaluation. Indeed, $f(x) = \int f(u) \delta(x - u)\, du$. Nevertheless, the diract delta function does not belong to L^2 and thus L^2 is not a RKHS. As noted in [Rasmussen and Williams, 2006], kernels are the analogues of dirac delta functions within the smoother RKHS.

Now let us consider the eigenfunction decomposition of the kernel $k(x, \tilde{x})$ (see Mercer's Theorem 1.4 in Section 1.4) with μ a probability measure, $\sup_{x \in \mathcal{X}} k(x, x) < \infty$ and $k(x, \tilde{x})$ is continuous on $\mathcal{X} \in \mathbb{R}^d$ - $\mathcal{X}$ is a nonempty open set. There exists an orthonormal sequence of eigenfunctions, $(\phi_p(x))_{p \geq 0} \in L^2_\mu(\mathcal{X})$ with the corresponding eigenvalues $(\lambda_p)_{p \geq 0} \geq 0$ sorting in decreasing order, such that

$$\int_{\mathcal{X}} k(x, \tilde{x}) \phi_p(\tilde{x})\, d\mu(\tilde{x}) = \lambda_p \phi_p(x), \quad p \geq 0,$$

$$k(x, \tilde{x}) = \sum_{p \geq 0} \lambda_p \phi_p(x) \phi_p(\tilde{x}),$$

$$\int_{\mathcal{X}} \int_{\mathcal{X}} k^2(x, \tilde{x})\, d\mu(x)\, d\mu(\tilde{x}) = \sum_{p \geq 0} \lambda_p^2 < \infty.$$

We note that for the case $\mathcal{X} = \{1, \ldots, N\}$ the analogs of the previous equations are $\mathbf{K} \boldsymbol{\phi}_p = \lambda_p \boldsymbol{\phi}_p$, $\mathbf{K} = \boldsymbol{\Gamma} \boldsymbol{\Lambda} \boldsymbol{\Gamma}$ and $\mathrm{tr}(\mathbf{K}^2) = \sum_{i=1}^{N} \lambda_p^2$ where $\mathbf{K} = [k(i,j)]_{i,j=1,\ldots,N}$, $\boldsymbol{\phi}_p = [\phi_p(i)]_{i=1,\ldots,N}$, $\boldsymbol{\Lambda} = \mathrm{diag}\left([\lambda_i]_{i=1,\ldots,N}\right)$ and $\boldsymbol{\Gamma} = [\boldsymbol{\phi}_i]_{i=1,\ldots,N}$ is orthogonal. We have the following proposition:

Proposition 1.2. *Let us consider a covariance kernel $k(x,\tilde{x})$ with an eigenfunction decomposition $k(x,\tilde{x}) = \sum_{p\geq 0} \lambda_p \phi_p(x)\phi_p(\tilde{x})$ with respect to the measure μ. If we consider $f(x) = \sum_{p\geq 0} f_p \phi_p(x)$, $f(x)$ is in the RKHS $\mathcal{H}$ with reproducing kernel $k(x,\tilde{x})$ if and only if*

$$\sum_{p\geq 0} \frac{f_p^2}{\lambda_p} < \infty$$

and $||f||_{\mathcal{H}}^2 = \sum_{p\geq 0} f_p^2/\lambda_p$. If $f(x) \in \mathcal{H}$, then we have the equality

$$f_p = \int_{\mathcal{X}} f(x)\phi_p(x)d\mu(x), \quad \text{for p such that } \lambda_p > 0.$$

Proof. The collection of functions $f(x)$ with $\sum_{p\geq 0} f_p^2/\lambda_p < \infty$ is a Hilbert space $\mathcal{H}$ with $||f||_{\mathcal{H}}^2 = \sum_{p\geq 0} f_p^2/\lambda_p$. We aim to prove that $\mathcal{H}$ is a RKHS with reproducing kernel $k(x,\tilde{x}) = \sum_{p\geq 0} \lambda_p \phi_p(x)\phi_p(\tilde{x})$. We have

$$||k_x||_{\mathcal{H}}^2 = \sum_{p\geq 0} \frac{\lambda_p^2 \phi_p^2(x)}{\lambda_p} = \sum_{p\geq 0} \lambda_p \phi_p^2(x) = k(x,x) < \infty.$$

Thus, k_x belongs to $\mathcal{H}$. Furthermore, we have the equalities

$$\langle f, k_x\rangle_{\mathcal{H}} = \sum_{p\geq 0} \frac{f_p(\lambda_p \phi_p(x))}{\lambda_p} = \sum_{p\geq 0} f_p \phi_p(x) = f(x),$$

which lead that $k(x,\tilde{x})$ has the reproducing property. Finally, we show that the evaluation functional is bounded:

$$\begin{aligned} |f(x)| = \left| \sum_{p\geq 0} \frac{f_p(\sqrt{\lambda_p}\phi_p(x))}{\sqrt{\lambda_p}} \right| &\leq \sqrt{\sum_{p\geq 0} \frac{f_p^2}{\lambda_p} \sum_{p\geq 0} \lambda_p \phi_p^2(x)} \\ &= ||f||_{\mathcal{H}} ||k_x||_{\mathcal{H}}. \end{aligned}$$

□

We can now consider the RKHS constituted by the functions of the form $f(x) = \sum_{p\geq 0} f_p \phi_p(x)$ with the inner product

$$\langle f, g\rangle_{\mathcal{H}} = \sum_{p\geq 0} \frac{f_p g_p}{\lambda_p}, \tag{1.66}$$

with $g(x) = \sum_{p\geq 0} g_p \phi_p(x)$. We note that despite the fact that the eigenvalue decomposition depends on the measure μ, the inner product is invariant under a change of measure [Kailath, 1971]. Another view of the RKHS can be obtained from the reproducing kernel map construction as stated in the following proposition.

Proposition 1.3. *Let us consider a covariance kernel $k(x,\tilde{x})$ $\forall n \in \mathbb{N}$, $x_i \in \mathcal{X}$, $\alpha_i \in \mathbb{R}$, $f(x) = \sum_{i=1}^n \alpha_i k(x, x_i)$ is in the RKHS $\mathcal{H}$ with reproducing kernel $k(x,\tilde{x})$, and $||f||_{\mathcal{H}}^2 = \sum_{i,j=1}^n \alpha_i \alpha_j k(x_i, x_j)$.*

Proof. The collection of functions $f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i)$ is a Hilbert space $\mathcal{H}$ with $||f||_{\mathcal{H}}^2 = \sum_{i,j=1}^{n} \alpha_i \alpha_j k(x_i, x_j)$. Furthermore, k_x belongs to $\mathcal{H}$ and has the reproducing property:

$$\langle f, k_x \rangle_{\mathcal{H}} = \sum_{i=1}^{n} \alpha_i k(x_i, x) = f(x).$$

□

We note that we recognize the form of the predictor given in Equation (1.6) in Subsection 1.2.1.

An example of Reproducing Kernel Hilbert Space in $[0, 1]$

Let us consider a function $f : [0, 1] \to \mathbb{R}$ with $m - 1$ continuous derivatives and such that $f^{(m)} \in L^2([0, 1])$ where $f^{(q)}$ denote the qth derivative of f. The Taylor series expansion gives

$$f(x) = \sum_{q=0}^{m-1} \frac{x^q}{q!} f^{(q)}(0) + \int_0^1 \frac{(x-u)_+^{m-1}}{(m-1)!} f^{(m)}(u)\, du,$$

with $(x - u)_+ = (x - u)\mathbf{1}_{x-u\geq 0}$. Furthermore, let us consider $\mathcal{A}_m$ the class of functions such that $\left(f^{(q)}(0) = 0\right)$, $\forall q = 0, \ldots, m - 1$. Then $f \in \mathcal{A}_m$ implies

$$f(x) = \int_0^1 G_m(x - u) f^{(m)}(u)\, du,$$

where $G_m(x - u) = (x - u)_+^{m-1}/(m - 1)!$. The function G_m is the Green's function for the problem $f^{(m)} = g$. Then, let us denote by $\mathcal{H}_m^0$ the following space

$$\mathcal{H}_m^0 := \left\{ f \in \mathcal{A}_m : [0, 1] \to \mathbb{R}, \left(f^{(q)}(0) = 0\right) \forall q = 0, \ldots, m - 1, f^{(m)} \in L^2([0, 1]) \right\}.$$

The collection of functions $\mathcal{H}_m^0$ is a Hilbert space with norm $||f||_{\mathcal{H}_m^0}^2 = \int_0^1 \left(f^{(m)}(u)\right)^2 du$. Furthermore, let us consider the kernel

$$k(x, \tilde{x}) = \int_0^1 G_m(x - u) G_m(\tilde{x} - u)\, du. \tag{1.67}$$

Denoting $k_x = k(x, .)$ we have

$$k_x^{(m)}(\tilde{x}) = G_m(x - \tilde{x}).$$

Thus, a simple calculation gives that

$$||k_x||_{\mathcal{H}_m^0}^2 = \int_0^1 \left(k_x^{(m)}(u)\right)^2 du = \int_0^1 \left(G_m(x - u)\right)^2 du = k(x, x).$$

Therefore k_x is in $\mathcal{H}_m^0$. Furthermore, we have

$$\langle f, k_x \rangle_{\mathcal{H}_m^0} = \int_0^1 f^{(m)}(u) k_x^{(m)}(u)\, du = \int_0^1 f^{(m)}(u) G_m(x - u)\, du = f(x)$$

and k_x has the reproducing property. Finally, it is easy to check that the evaluation functional if bounded:

$$|f(x)| = \langle f, k_x \rangle_{\mathcal{H}_m^0} \leq ||f||_{\mathcal{H}_m^0} ||k_x||_{\mathcal{H}_m^0} = ||f||_{\mathcal{H}_m^0} \sqrt{k(x, x)}.$$

Connection with Gaussian processes.

Let us consider a Gaussian process $Z(x)$, $x \in \mathcal{X}$ with zero mean and covariance kernel $k(x, \tilde{x}) = \sum_{p\geq 0} \lambda_p \phi_p(x)\phi_p(\tilde{x})$. Then, the Karhunen-Loeve representation of $Z(x)$ is given by

$$Z(x) \sim \sum_{p\geq 0} Z_p \phi_p(x),$$

where $(Z_p)_{p\geq 0}$ are independent Gaussian random variables with mean zero and variance λ_p such that

$$Z_p = \int Z(x)\phi_p(x)\, d\mu(x). \tag{1.68}$$

The integral (1.68) is well defined in quadratic mean [Cramer and Leadbetter, 1967]. Nonetheless, if $k(x, \tilde{x})$ is non-degenerate (i.e., if it has a infinite number of non-zero eigenvalues), then samples of $Z(x)$ do not belong to $\mathcal{H}$. Therefore, the assumption $f \in \mathcal{H}$ and f is a sample of the Gaussian process $Z(x)$ are not equivalent. To illustrate this statement, let us consider the degenerate kernel $k_{\bar{p}}(x, \tilde{x}) = \sum_{p\leq \bar{p}} \lambda_p \phi_p(x)\phi_p(\tilde{x})$ and the corresponding Gaussian process $Z_{\bar{p}}(x) = \sum_{p\leq \bar{p}} Z_p \phi_p(x)$. We have

$$\mathbb{E}\left[|Z_{\bar{p}}(x) - Z(x)|^2\right] = \sum_{p=\bar{p}+1}^{\infty} \lambda_p \phi_p^2(x) \overset{\bar{p}\to\infty}{\longrightarrow} 0.$$

Therefore, $Z_{\bar{p}}(x)$ tends to $Z(x)$ in mean square sense but

$$\mathbb{E}\left[||Z_{\bar{p}}(x)||_{\mathcal{H}}^2\right] = \sum_{p=0}^{\bar{p}} \frac{\mathbb{E}\left[|Z_p|^2\right]}{\lambda_p} = \bar{p} + 1 \overset{\bar{p}\to\infty}{\longrightarrow} \infty.$$

However, as noted in [Rasmussen and Williams, 2006], the posterior mean of the Gaussian process after observing some data will lie in the RKHS due to the averaging.

Now, let us consider the Hilbert space $\mathcal{Z}$ spanned by $Z(x)$, $x \in \mathcal{X}$. It is the collection of random variables of the form $Z = \sum_{i=1}^n \alpha_j Z(x_i)$ with the inner product $\langle Z_1, Z_2\rangle = \mathbb{E}\left[Z_1 Z_2\right]$ and all of their quadratic mean limits. First, the equalities

$$\langle Z(x), Z(\tilde{x})\rangle = \mathbb{E}\left[Z(x)Z(\tilde{x})\right] = k(x, \tilde{x}) = \langle k_x, k_{\tilde{x}}\rangle$$

show that there is a correspondence between the inner product of $\mathcal{Z}$ and the one of $\mathcal{H}$. Now let us consider a bounded linear function in $\mathcal{H}$ with representer η. Thus, η can be written in the form $\eta(x) = \lim_n \eta^{(n)}(x)$ with $\eta^{(n)}(x) = \sum_{i=1}^n \alpha_i k(x_i, x)$. Furthermore, let us define Z_∞ as the L^2-limit of $\sum_{i=1}^n \alpha_i Z(x_i) = Z^{(n)}$, $\eta^{(n)}$ converges in $\mathcal{H}$ if and only if $Z^{(n)}$ converges in L^2 . Therefore, if the limit $\lim_n \mathbb{E}\left[(Z_\infty - \sum_{i=1}^n \alpha_i Z(x_i))^2\right] = 0$ holds, we have

$$\mathbb{E}\left[Z_\infty Z(x)\right] = \lim_n \sum_{i=1}^n \alpha_i \mathbb{E}\left[Z(x_i)Z(x)\right] = \lim_n \sum_{i=1}^n \alpha_i k(x_i, x) = \eta(x).$$

Therefore, the Hilbert space $\mathcal{Z}$ is isomorphic to $\mathcal{H}$ with the correspondences $Z(x) \sim k_x$, $Z_\infty \sim \eta$ and a preserved inner product.

Regularization problem in a RKHS

Let us consider the following functional:

$$J(f) = \frac{\lambda}{2}||f||^2_{\mathcal{H}} + Q(\mathbf{z}^n, \mathbf{f}^n), \tag{1.69}$$

where $\mathbf{z}^n$ is the observed values of the objective function $z(x)$ we are approximating, $\mathbf{f}^n = f(\mathbf{D}) = (f(x_1), \ldots, f(x_n))'$ and λ is a scalar parameter. The term $Q(\mathbf{z}^n, \mathbf{f}^n)$ in (1.69) is a measure of the distance between the observed values $\mathbf{z}^n$ and the predicted ones $\mathbf{f}^n$. Moreover, the norm $||f||_{\mathcal{H}}$ in the Hilbert space $\mathcal{H}$ represents the regularity of the predictor f. The purpose of this section is to determine the function f minimizing (1.69). In a Gaussian process regression framework, we consider that $Q(\mathbf{z}^n, \mathbf{f}^n)$ is a squared loss function, i.e.

$$Q(\mathbf{z}^n, \mathbf{f}^n) = (\mathbf{z}^n - \mathbf{f}^n)'(\mathbf{z}^n - \mathbf{f}^n).$$

More general forms of loss functions can be found in the book of [Wahba, 1990]. Let us consider the following Theorem:

Theorem 1.7 (Representer Theorem)**.** *Let us consider a function f in a RKHS $\mathcal{H}$ with the reproducing kernel $k(x, \tilde{x})$. Each minimizer $f \in \mathcal{H}$ of*

$$J(f) = \frac{\lambda}{2}||f||^2_{\mathcal{H}} + Q(\mathbf{z}^n, \mathbf{f}^n),$$

has the form

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i).$$

Again we recognize the form of the kriging predictor giving in Equation (1.6). Theorem 1.7 was first proved by [Kimeldorf and Wahba, 1971] in the case of squared loss functions.

Now let us consider the following functional

$$J(f) = \frac{1}{2}||f||^2_{\mathcal{H}} + \frac{1}{2\sigma_\varepsilon^2}(\mathbf{z}^n - \mathbf{f}^n)'(\mathbf{z}^n - \mathbf{f}^n). \tag{1.70}$$

Theorem 1.7 gives us that the solution of (1.70) has the form $f(x) = \mathbf{k}'(x)\boldsymbol{\alpha}^n$ with $\boldsymbol{\alpha}^n = (\alpha_i, \ldots, \alpha_n)'$, $n \in \mathbb{N}$ and $\mathbf{k}(x) = [k(x, x_i)]_{i=1,\ldots,n}$. Thus, the functional (1.70) can be written:

$$\begin{aligned} J(\boldsymbol{\alpha}) &= \frac{1}{2}(\boldsymbol{\alpha}^n)'\mathbf{K}\boldsymbol{\alpha}^n + \frac{1}{2\sigma_\varepsilon^2}(\mathbf{z}^n - \mathbf{K}\boldsymbol{\alpha}^n)'(\mathbf{z}^n - \mathbf{K}\boldsymbol{\alpha}^n) \\ &= \frac{1}{2}(\boldsymbol{\alpha}^n)'\left(\mathbf{K} + \frac{1}{\sigma_\varepsilon^2}\mathbf{K}'\mathbf{K}\right)\boldsymbol{\alpha}^n - \frac{1}{\sigma_\varepsilon^2}(\mathbf{z}^n)'\mathbf{K}\boldsymbol{\alpha}^n + \frac{1}{2\sigma_\varepsilon^2}(\mathbf{z}^n)'\mathbf{z}^n, \end{aligned}$$

with $\mathbf{K} = [k(x_i, x_j)]_{i,j=1,\ldots,n}$ and noticing that $||f||^2_{\mathcal{H}} = (\boldsymbol{\alpha}^n)'\mathbf{K}\boldsymbol{\alpha}^n$ as stated in Proposition 1.3. The minimum of $J(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}^n$ is given by

$$\hat{\boldsymbol{\alpha}}^n = \left(\mathbf{K} + \sigma_\varepsilon^2\mathbf{I}\right)^{-1}\mathbf{z}^n.$$

Thus, the solution of the regularization problem is given by:

$$\hat{z}(x) = \mathbf{k}'(x)\left(\mathbf{K} + \sigma_\varepsilon^2\mathbf{I}\right)^{-1}\mathbf{z}^n, \tag{1.71}$$

which is exactly the form of the predictor in a noisy-kriging framework (1.8) with a constant observation noise variance, i.e. $\mathbf{\Delta} = \sigma_\varepsilon^2 \mathbf{I}$. We can consider two extreme cases for the functional $J(f)$ presented in Equation (1.70). First, let us consider the case $\sigma_\varepsilon^2 \to \infty$. Thus, $J(f)$ becomes $J(f) = ||f||_{\mathcal{H}}^2$ which means that we only considered the penalization on the regularity of f. We can derive the same calculations as before and we find that $\boldsymbol{\alpha}^n = 0$. If we refer to the kriging framework, it corresponds to the mean of the Gaussian process $Z(x)$ modeling $z(x)$ in a simple kriging case. In fact, as presented by [Wahba, 1990] Sec 1.3, this case corresponds to the one of the generalized linear regression. Then, let us consider the asymptotic $\sigma_\varepsilon^2 \to 0$ which corresponds to the minimization problem $J(f) = (\mathbf{z}^n - \mathbf{f}^n)'(\mathbf{z}^n - \mathbf{f}^n)$. In that case we find the following solution for the minimization problem

$$\hat{\boldsymbol{\alpha}}^n = \mathbf{K}^{-1}\mathbf{z}^n,$$

which corresponds to the predictor

$$\hat{z}(x) = \mathbf{k}'(x)\mathbf{K}^{-1}\mathbf{z}^n. \tag{1.72}$$

We recognize the form of the predictor obtained in a simple Kriging framework with noisy-free observations (1.3).

A useful property of RKHS

The Riesz representation theorem tells us that any bounded linear function L in $\mathcal{H}$ has a unique representer η in $\mathcal{H}$. The powerful property of the reproducing kernel k_x is that we can deduce η from it. Indeed, we have

$$\eta(\tilde{x}) = \langle \eta, k_{\tilde{x}} \rangle_{\mathcal{H}} = Lk_{\tilde{x}},$$

which means that $\eta(\tilde{x})$ can be obtained by applying L to $k_{\tilde{x}}$. For example, if we consider $\mathcal{X} = \mathbb{R}^d$ and $Lf = \int f(u)\,du$ then $\eta(\tilde{x}) = \int k_{\tilde{x}}(u)\,du$. Moreover, if we consider $\mathcal{X} = \mathbb{R}$, $f(x)$ and $k_{\tilde{x}}(x)$ differentiable and $Lf = \frac{d}{dx}f(x)$ for some $x \in \mathbb{R}$, then $\eta(\tilde{x}) = \frac{d}{dx}k_{\tilde{x}}(x)$.

Then we can consider the space $\mathcal{H}_\eta$ spanned by η and its orthogonal $\mathcal{H}_\eta^\perp$. The spaces $\mathcal{H}_\eta$ and $\mathcal{H}_\eta^\perp$ are two subspaces of $\mathcal{H}$ such that $\mathcal{H} = \mathcal{H}_\eta \oplus \mathcal{H}_\eta^\perp$ and are themselves RKHS. As stated in [Berlinet and Thomas-Agnan, 2004] Theorem 11, the reproducing kernel k_x^η of $\mathcal{H}_\eta$ is given by the orthogonal projection of k_x on $\mathcal{H}_\eta$:

$$k_x^\eta = \langle k_x, \eta \rangle_{\mathcal{H}} \frac{\eta}{||\eta||_{\mathcal{H}}^2} = \eta(x)\frac{\eta}{||\eta||_{\mathcal{H}}^2}. \tag{1.73}$$

Furthermore, the relation $\mathcal{H} = \mathcal{H}_\eta \oplus \mathcal{H}_\eta^\perp$ implies that the kernel of $\mathcal{H}_\eta^\perp$ is given by $k_x - k_x^\eta$. We note that the norm $||\eta||_{\mathcal{H}}^2$ can be deduced from the following equality

$$||\eta||_{\mathcal{H}}^2 = \langle \eta, \eta \rangle_{\mathcal{H}} = \langle Lk_x, Lk_{\tilde{x}} \rangle_{\mathcal{H}}$$

As an application, a very interesting use of this property were suggested by [Durrande et al., 2013] who propose an ANOVA decomposition for the reproducing kernel $k(x, \tilde{x})$. Then,

this decomposition is used to perform sensitivity analysis in an efficient way. Their approach is based on the following proposition (see [Durrande et al., 2013] Proposition 1):

Proposition 1.4. *Let $\mathcal{H}$ be an RKHS with a reproducing kernel $k(x, \tilde{x})$, $x, \tilde{x} \in \mathbb{R}$, then $\mathcal{H}$ can be decomposed as a sum of two orthogonal RKHS*

$$\mathcal{H} = \mathcal{H}_1 \overset{\perp}{\oplus} \mathcal{H}_0,$$

where $\mathcal{H}_0$ is a RKHS of zero-mean functions and $\mathcal{H}_1$ is its orthogonal.

The proof is straightforward according to the previous discussion by considering the bounded linear functional $Lf = \int f(u)\,du$ with its representer $\eta(\tilde{x}) = \int k_{\tilde{x}}(u)\,du$. By applying the presented results, the kernel for $\mathcal{H}_1$ is given by $k_x^1(\tilde{x}) = \eta(x)\eta(\tilde{x})/||\eta||^2_{\mathcal{H}}$, i.e:

$$k_x^1(\tilde{x}) = \int k_x(u)\,du \frac{\int k_{\tilde{x}}(u)\,du}{\int\int k(v,u)\,du\,dv}.$$

Then, the reproducing kernel of the orthogonal space $\mathcal{H}_1^{\perp} = \mathcal{H}_0$ - which corresponds to the collection of functions g such that $\langle \eta, g\rangle_{\mathcal{H}} = Lg = \int g(u)\,du = 0$, i.e. the space of zero-mean functions - is given by

$$k_x^0(\tilde{x}) = k_x(\tilde{x}) - k_x^1(\tilde{x}).$$

Example of a Gaussian process with zero mean function. Let us consider a 1-dimensional Gaussian process $Z(x)$, $x \in [0,1]$ with zero mean and covariance kernel $k(x,\tilde{x}) = \exp\left(-|x-\tilde{x}|/\theta\right)$ with $\theta = 10$. It corresponds to the Ornstein-Uhlenbeck kernel presented in Subsection 1.4.2. The advantage of this kernel is that a closed form expression can be given for Equation (1.73). Indeed, after straightforward calculations, we find that

$$k^1(x,\tilde{x}) = \frac{\left(2\theta - \theta\left(\exp(-\frac{x}{\theta}) + \exp(\frac{x-1}{\theta})\right)\right)\left(2\theta - \theta\left(\exp(-\frac{\tilde{x}}{\theta}) + \exp(\frac{\tilde{x}-1}{\theta})\right)\right)}{2\theta - 2\theta^2 + 2\theta^2\exp\left(-\frac{1}{\theta}\right)}$$

and the reproducing kernel for the sub-RKHS of zero mean functions is given by $k^0(x,\tilde{x}) = k(x,\tilde{x}) - k^1(x,\tilde{x})$. We illustrate in Figure 1.13 one realization of a Gaussian process with covariance kernel $k(x,\tilde{x})$ and the same realization but with covariance kernel $k^0(x,\tilde{x})$.

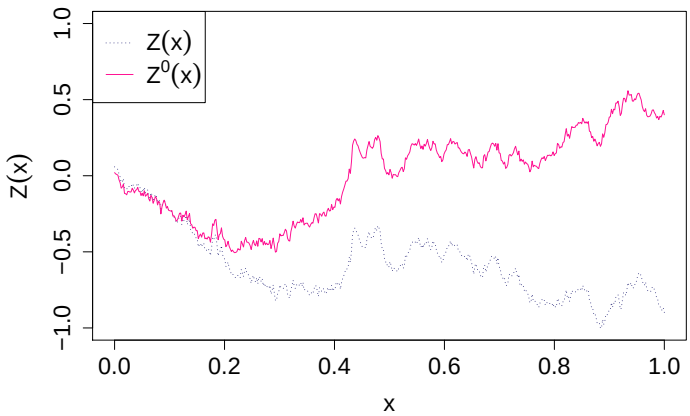


Figure 1.13: Example of realizations for the Gaussian processes $Z(x)$ with covariance kernel $k(x, \tilde{x})$ and $Z^0(x)$ with covariance kernel $k^0(x, \tilde{x})$. $k^0(x, \tilde{x})$ is the reproducing kernel of the sub-RKHS of zero mean functions on $[0, 1]$. The two realizations are computed thanks to the Cholesky's decomposition method (see Subsection 1.4.2) with the same Gaussian white noise. We empirically observe that the mean of the realization of $Z^0(x)$ is close to 0, as expected. Indeed, it equals $-3.5.10^{-5}$ whereas the one of $Z(x)$ is $-3.1.10^{-1}$.

Chapter 2

Co-kriging models

In Chapter 1, we have presented how to surrogate an objective function $z(x)$ with $x \in Q \subseteq \mathbb{R}^d$, Q an nonempty open set, and $z(x) \in \mathbb{R}$. Nevertheless, in practical applications, the objective function can be multivariate, i.e. its output can lie in $\mathbb{R}^s$ with $s \in \mathbb{N}^*$. We denote such functions by $\mathbf{z}(x) = (z_1(x), \ldots, z_s(x)) \in \mathbb{R}^s$ with $x \in Q$. Furthermore, the different components $(z_i(x))_{i=1,\ldots,s}$ of the vector of functions $\mathbf{z}(x)$ can be dependent. Therefore, if we want to approximate a component $z_i(x)$ of $\mathbf{z}(x)$ it could be worthwhile to take into account the other ones $(z_j(x))_{j \neq i}$.

In this chapter, we are interested in that framework. The component of $\mathbf{z}(x)$ that we want to predict is generally called the principal component and the other ones are the secondary components.

In Section 2.1 we present the extension of the kriging model for multivariate functions. This extension is called co-kriging and was first developed in geostatistics (see [Chilès and Delfiner, 1999] and [Wackernagel, 2003]). Then, in Section 2.2 we present the original model of co-kriging suggested in the geostatistical literature. In Section 2.3 we deal with the definition of valid covariance kernels for co-kriging models. Finally, in Section 2.4 we present an approach in computer experiments using co-kriging models to surrogate the output of a code. It corresponds to the case where we want to take into account the code output derivatives into the model.

2.1 Bayesian Kriging models for vectorial functions

Let us suppose that we want to approximate the last component $z_s(x)$ of $\mathbf{z}(x)$ by taking into account the other components $(z_i(x))_{i=1,\ldots,s-1}$. Analogously to the Gaussian process regression, we consider that the output of the objective function is a multivariate Gaussian process $\mathbf{Z}(x) = (Z_1(x), \ldots, Z_s(x))$ with mean $\mathbf{m}(x)$ and matrix-valued covariance function

$\mathbf{V}(x,\tilde{x})$. In a multivariate case, we have

$$\mathbf{m}(x) = \begin{pmatrix} m_1(x) \\ \vdots \\ m_s(x) \end{pmatrix} \tag{2.1}$$

and

$$\mathbf{V}(x,\tilde{x}) = \begin{pmatrix} k_{11}(x,\tilde{x};\boldsymbol{\theta}_{11}) & \dots & k_{1s}(x,\tilde{x};\boldsymbol{\theta}_{1s}) \\ \vdots & \ddots & \vdots \\ k_{s1}(x,\tilde{x};\boldsymbol{\theta}_{s1}) & \dots & k_{ss}(x,\tilde{x};\boldsymbol{\theta}_{ss}) \end{pmatrix}, \tag{2.2}$$

where $k_{ij}(x,\tilde{x}) = \operatorname{cov}(Z_i(x), Z_j(\tilde{x}); \boldsymbol{\theta}_{ij})$, $i,j = 1,\dots,s$ and $m_i(x) = \mathbb{E}[Z_i(x)]$, $i = 1,\dots,s$. We note that the hyper-parameters $\boldsymbol{\theta}_{ij}$, representing the parameters of the covariance kernel $k_{ij}(x,\tilde{x})$, can include the variance parameter. For the moment, we consider that $\mathbf{V}(x,\tilde{x})$ is a valid matrix-valued covariance function. In fact, its choice is non-trivial since assuring the positive definiteness of $\mathbf{V}(x,\tilde{x})$ could be an issue. We present in Section 2.3 how to define admissible covariance structures in a multivariate context. It is though important to note that $\mathbf{V}(x,\tilde{x})$ is not necessarily symmetric, i.e. we can have $k_{ij}(x,\tilde{x}) \neq k_{ji}(x,\tilde{x})$. Moreover, as in a kriging case, we consider that the ith component of $\mathbf{m}(x)$ is of the form $m_i(x) = \mathbf{f}_i'(x)\boldsymbol{\beta}_i$ with $\mathbf{f}_i'(x)$ a vector of functions of size p_i.

2.1.1 Simple co-kriging equations

Let us denote by $\mathbf{Z}^{(s)} = ((\mathbf{Z}_1^{n_1})', \dots, (\mathbf{Z}_s^{n_s})')'$ the values of $(Z_i(x))_{i=1,\dots,s}$ at points in $(\mathbf{D}^i)_{i=1,\dots,s}$ where $\mathbf{D}^i = (x_1^{(i)}, \dots, x_{n_i}^{(i)})$, $x_j^{(n_i)} \in \mathbb{R}^d$, $j = 1,\dots,n_i$, $i = 1,\dots,s$. Furthermore, we denote by $\mathbf{z}^{(s)} = (\mathbf{z}_1^{n_1}, \dots, \mathbf{z}_s^{n_s})$ the values of $(z_i(x))_{i=1,\dots,s}$ at points in $(\mathbf{D}^i)_{i=1,\dots,s}$ and by $\mathbf{M}^{(s)} = (\mathbf{M}_1, \dots, \mathbf{M}_s)$ the values of $(m_i(x))_{i=1,\dots,s}$ at points in $(\mathbf{D}^i)_{i=1,\dots,s}$. Thus, we have $\mathbf{M}_i = \mathbf{f}_i'(\mathbf{D}^i)\boldsymbol{\beta}_i := \mathbf{F}_i\boldsymbol{\beta}_i$ with $\mathbf{F}_i$ a matrix of size $n_i \times p_i$, $i = 1,\dots,s$.

The purpose of the co-kriging model is to predict the value of $Z_s(x)$ by considering the known values $\mathbf{z}^{(s)}$. As in the simple kriging case, the predictive distribution of the simple co-kriging is given by $[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, (\boldsymbol{\beta}_i)_{i=1,\dots,s}, (\boldsymbol{\theta}_{ij})_{i,j=1\dots,s}]$. Let us consider the following Gaussian vector

$$\begin{pmatrix} Z_s(x) \\ \mathbf{Z}_1 \\ \vdots \\ \mathbf{Z}_s \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{f}_s'(x)\boldsymbol{\beta}_s \\ \mathbf{F}_1\boldsymbol{\beta}_1 \\ \vdots \\ \mathbf{F}_s\boldsymbol{\beta}_s \end{pmatrix}, \begin{pmatrix} k_{ss}(x,x) & \mathbf{k}_{s1}'(x) & \dots & \mathbf{k}_{ss}'(x) \\ \mathbf{k}_{1s}(x) & \mathbf{K}_{11} & \dots & \mathbf{K}_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{k}_{ss}(x) & \mathbf{K}_{s1} & \dots & \mathbf{K}_{ss} \end{pmatrix} \right), \tag{2.3}$$

with $\mathbf{k}_{sj}(x) = [k_{sj}(x, x_k^{(j)})]_{k=1,\dots,n_j}$, $\mathbf{k}_{js}(x) = [k_{js}(x_k^{(j)}, x)]_{k=1,\dots,n_j}$ and $\mathbf{K}_{ij} = [k_{ij}(x_k^{(i)}, x_l^{(j)})]_{\substack{k=1,\dots,n_i \\ l=1,\dots,n_j}}$. We note that although in general $k_{ij}(x,\tilde{x}) \neq k_{ji}(x,\tilde{x})$, we have the equality $\mathbf{k}_{sj}(x) = \mathbf{k}_{js}(x)$ and $\mathbf{K}_{ij} = \mathbf{K}_{ji}'$. Indeed, the equality $\operatorname{cov}(Z_i(x), Z_j(\tilde{x})) = \operatorname{cov}(Z_j(\tilde{x}), Z_i(x))$ implies that $k_{sj}(x,\tilde{x}) = k_{js}(\tilde{x},x)$ and thus $\mathbf{k}_{ij}(x) = \mathbf{k}_{ji}(x)$ and $\mathbf{K}_{ij} = \mathbf{K}_{ji}'$. Thus, we obtain that the predictive distribution $[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, (\boldsymbol{\beta}_i)_{i=1,\dots,s}, (\boldsymbol{\theta}_{ij})_{i,j=1\dots,s}]$ is Gaussian with mean $m_{Z_s,SK}(x)$

and variance $s^2_{Z_s,SK}(x)$ given by:

$$m_{Z_s,SK}(x) = \mathbf{f}_s'(x)\boldsymbol{\beta}_s + \mathbf{k}_s'(x)\mathbf{V}_s^{-1}\left(\mathbf{z}^{(s)} - \mathbf{M}^{(s)}\right) \tag{2.4}$$

and

$$s^2_{Z_s,SK}(x) = k_{ss}(x,x) - \mathbf{k}_s'(x)\mathbf{V}_s^{-1}\mathbf{k}_s(x), \tag{2.5}$$

where $\mathbf{k}_s'(x) = \begin{pmatrix}\mathbf{k}_{s1}'(x) & \dots & \mathbf{k}_{ss}'(x)\end{pmatrix}$ and

$$\mathbf{V}_s = \begin{pmatrix}\mathbf{K}_{11} & \dots & \mathbf{K}_{1s} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{s1} & \dots & \mathbf{K}_{ss}\end{pmatrix}. \tag{2.6}$$

Considering the univariate case $s = 1$, the predictive mean (2.4) and variance (2.5) are identical to the ones of the Simple kriging (1.3) and (1.4).

We note that the matrix $\mathbf{V}_s$ must be positive definite. We present in Section 2.3 different covariance structures which ensure this property. Furthermore, the equality $k_{ij}(x,\tilde{x}) = k_{ji}(\tilde{x},x)$ implies that $\mathbf{V}_s$ is symmetric. The predictive mean $m_{Z_s,SK}(x)$ is the surrogate model for the component $z_s(x)$ of $\mathbf{z}(x)$ and the predictive variance $s^2_{Z_s,SK}(x)$ represents the model mean squared error. Like in simple kriging with noisy-free observations, $m_{Z_s,SK}(x)$ interpolates $z_s(x)$ at points of the experimental design set and $s^2_{Z_s,SK}(x)$ equals zero at these points. Furthermore, we can easily integrate a noise variance in the model by considering a nugget effect as presented in Subsection 1.2.1 in the paragraph "The noisy case". In that case, the surrogate model will not interpolate the observed values anymore.

Example of simple co-kriging

Let us consider the bivariate Gaussian process $(Z_1(x), Z_2(x))$, $x \in \mathbb{R}$ such that

$$\begin{cases} Z_1(x) = a_1\delta_1(x) & + & a_2\delta_2(x) \\ Z_2(x) = b_1\delta_1(x) & + & b_2\delta_2(x) \end{cases},$$

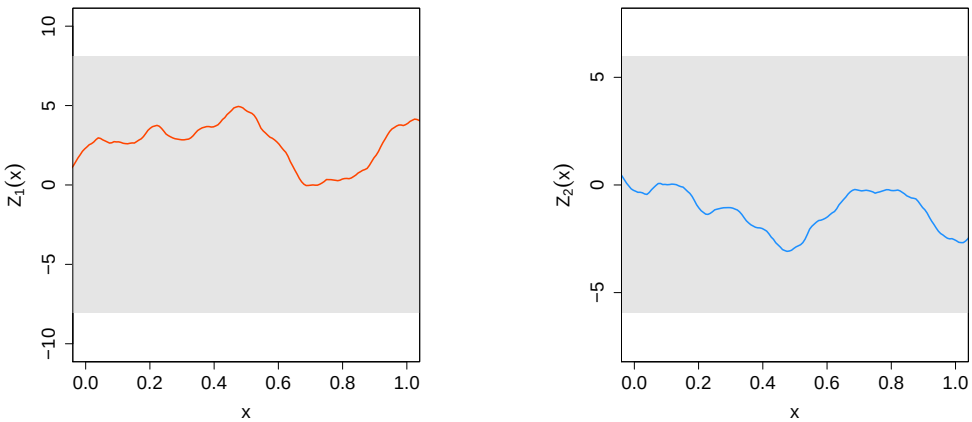
where $\delta_1(x)$ and $\delta_2(x)$ are two independent Gaussian processes with means zero and covariances $k_1(x,\tilde{x})$ and $k_2(x,\tilde{x})$ such that:

- $k_1(x,\tilde{x})$ is a 5/2-Matérn kernel with variance parameter $\sigma^2 = 1$ and characteristic length scale $\theta = 0.2$,
- $k_2(x,\tilde{x})$ is a 3/2-Matérn kernel with variance parameter $\sigma^2 = 1$ and characteristic length scale $\theta = 0.3$.

The bivariate stochastic process $(Z_1(x), Z_2(x))$ is Gaussian since it is a linear combination of the bivariate Gaussian process $(\delta_1(x), \delta_2(x))$. We note that the independence ensures the normality for $(\delta_1(x), \delta_2(x))$. Furthermore, $(Z_1(x), Z_2(x))$ has zero mean and covariance structure

$$\mathbf{V}(x, \tilde{x}) = \begin{pmatrix} a_1^2 k_1(x, \tilde{x}) + a_2^2 k_2(x, \tilde{x}) & a_1 b_1 k_1(x, \tilde{x}) + a_2 b_2 k_2(x, \tilde{x}) \\ a_1 b_1 k_1(x, \tilde{x}) + a_2 b_2 k_2(x, \tilde{x}) & b_1^2 k_1(x, \tilde{x}) + b_2^2 k_2(x, \tilde{x}) \end{pmatrix}. \tag{2.7}$$

Let us consider the sample of $Z_1(x)$ and $Z_2(x)$ showed in Figure 2.1 with $a_1 = 1$, $a_2 = -4$, $b_1 = 0.5$ and $b_2 = 3$.



(a) (b)

Figure 2.1: Example of sample for the bivariate Gaussian process $(Z_1(x), Z_2(x))$ with covariance structure defined in (2.7) with $a_1 = 1$, $a_2 = -4$, $b_1 = 0.5$ and $b_2 = 3$. Figure (a) illustrates the sample of $Z_1(x)$ and Figure (b) illustrates the sample of $Z_2(x)$.

We aim to reconstruct the sample of $Z_1(x)$ from its values at points in $\mathbf{D}^1 = (-0.20, 0.08, 0.36, 0.64, 0.93)$ and the sampled values of $Z_2(x)$ at points in $\mathbf{D}^2 = (-0.20, -0.06, 0.08, 0.22, 0.36, 0.50, 0.64, 0.78, 0.93, 1.07)$. Figure 2.2 illustrates the predictive mean and confidence intervals obtained for the simple co-kriging equations (2.4) and (2.5). Furthermore, we also illustrate the predictive mean (1.3) and variance (1.4) of the simple kriging using only the sampled values of $Z_1(x)$ at points in $\mathbf{D}^1$. We see in Figure 2.2 that the confidence intervals of the co-kriging model are smaller than the ones of the kriging model. Furthermore, they are more relevant in the co-kriging model since they represent more precisely the real model error. Finally, we see that the co-kriging mean is more accurate than the kriging one.

2.1.2 Co-kriging parameter estimation

In a co-kriging framework, the hyper-parameters $(\boldsymbol{\theta}_{ij})_{i,j=1\ldots,s}$ are considered as known - this include the variance parameters. We note that the selection methods presented in Section 1.3 can naturally be extended for the co-kriging model. However, they will be in general extremely computationally expensive. Nevertheless, we will see in Part II that in some particular contexts we can easily infer from some hyper-parameters about the predictive distribution. In this

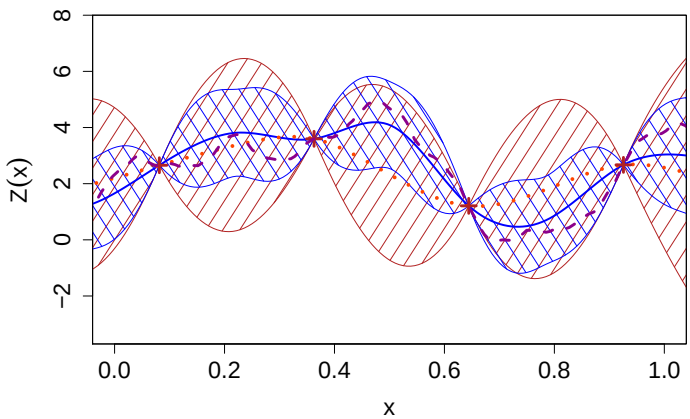


Figure 2.2: Comparison between co-kriging and kriging models. The solid line represents the co-kriging mean, the dotted line represents the kriging mean, the dashed line represents the sample of $Z_1(x)$ that we want to approximate. The shade areas represent the mean plus and minus twice the predictive standard deviation of the co-kriging and kriging models.

subsection, we only deal with the estimation of the vector $\boldsymbol{\beta}^{(s)} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_s)$ of size $(\sum_{i=1}^{s} p_i)$. We consider here a Bayesian estimate for $\boldsymbol{\beta}^{(s)}$ but the maximum likelihood one can be deduced from it without difficulties. First, let us consider the probability density function of the random vector $\mathbf{Z}^{(s)}$

$$p(\mathbf{z}^{(s)}|\boldsymbol{\beta}^{(s)}) = \frac{\exp\left(-\frac{1}{2}\left(\mathbf{z}^{(s)} - \mathbf{F}^{(s)}\boldsymbol{\beta}^{(s)}\right)' \mathbf{V}_s^{-1}\left(\mathbf{z}^{(s)} - \mathbf{F}^{(s)}\boldsymbol{\beta}^{(s)}\right)\right)}{(2\pi)^{n/2}\sqrt{\det \mathbf{V}_s}}, \tag{2.8}$$

where $n = \sum_{i=1}^{s} n_i$ and $\mathbf{F}^{(s)}$ is the following $(\sum_{i=1}^{s} n_i) \times (\sum_{i=1}^{s} p_i)$ matrix

$$\mathbf{F}^{(s)} = \begin{pmatrix} \mathbf{F}_1 & 0 & 0 & \ldots & 0 \\ 0 & \mathbf{F}_2 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & 0 & \mathbf{F}_{s-1} & 0 \\ 0 & \ldots & 0 & 0 & \mathbf{F}_s \end{pmatrix}.$$

We note that $p(\mathbf{z}^{(s)}|\boldsymbol{\beta}^{(s)})$ is the likelihood of parameter $\boldsymbol{\beta}^{(s)}$. Then, from the Bayes rule we have:

$$p(\boldsymbol{\beta}^{(s)}|\mathbf{z}^{(s)}) \propto p(\mathbf{z}^{(s)}|\boldsymbol{\beta}^{(s)})p(\boldsymbol{\beta}^{(s)})$$

and thanks to the improper Jeffrey's prior distribution

$$p(\boldsymbol{\beta}^{(s)}) \propto 1,$$

we find that the distribution $[\boldsymbol{\beta}^{(s)}|\mathbf{z}^{(s)}]$ is

$$\mathcal{N}\left(\bar{\boldsymbol{\beta}}^{(s)}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}^{(s)}}\right), \tag{2.9}$$

where

$$\mathbf{\Sigma}_{\boldsymbol{\beta}^{(s)}} = \left(\left(\mathbf{F}^{(s)} \right)' \mathbf{V}_s^{-1} \mathbf{F}^{(s)} \right)^{-1} \tag{2.10}$$

and

$$\bar{\boldsymbol{\beta}}^{(s)} = \mathbf{\Sigma}_{\boldsymbol{\beta}^{(s)}} \left(\mathbf{F}^{(s)} \right)' \mathbf{V}_s^{-1} \mathbf{z}^{(s)}. \tag{2.11}$$

We emphasize that the posterior distribution of parameter $\boldsymbol{\beta}^{(s)}$ is similar to the one found in Equation (1.17). In particular, for $s = 1$ they are identical. We note that the MLE of $\boldsymbol{\beta}^{(s)}$ is given by $\bar{\boldsymbol{\beta}}^{(s)}$ in (2.11). Furthermore, we can easily extend the result given in Subsection 1.2.2 if we consider a Gaussian prior distribution for $\boldsymbol{\beta}^{(s)}$.

2.1.3 Universal co-kriging equations

As presented in Subsection 1.2.2, we can infer from the posterior distribution of $\boldsymbol{\beta}^{(s)}$ given in Equation (2.9) about the predictive distribution of the simple co-kriging which is a Gaussian with mean given in Equation (2.4) and covariance given in Equation (2.5).

Let us integrate the posterior distribution of $\boldsymbol{\beta}^{(s)}$:

$$p(z_s(x)|\mathbf{z}^{(s)}) = \int p(z_s(x)|\mathbf{z}^{(s)}, \boldsymbol{\beta}^{(s)}) p(\boldsymbol{\beta}^{(s)}|\mathbf{z}^{(s)})\, d\boldsymbol{\beta}^{(s)}.$$

After direct calculations, it can be shown that the predictive distribution $[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, (\boldsymbol{\theta}_{ij})_{i,j=1\ldots,s}]$ is Gaussian with mean

$$m_{Z_s}(x) = \mathbf{f}_s'(x)\hat{\boldsymbol{\beta}}_s + \mathbf{k}_s'(x)\mathbf{V}_s^{-1} \left(\mathbf{z}^{(s)} - \mathbf{F}^{(s)}\hat{\boldsymbol{\beta}}^{(s)} \right) \tag{2.12}$$

and variance

$$s^2_{Z_s}(x) = k_{ss}(x,x) - \left(\left(\mathbf{f}^{(s)}(x)\right)' \quad \mathbf{k}_s'(x) \right) \begin{pmatrix} 0 & \left(\mathbf{F}^{(s)}\right)' \\ \mathbf{F}^{(s)} & \mathbf{V}_s \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{f}^{(s)}(x) \\ \mathbf{k}_s(x) \end{pmatrix}, \tag{2.13}$$

where

$$\mathbf{f}^{(s)}(x) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \mathbf{f}_s(x) \end{pmatrix},$$

$\bar{\boldsymbol{\beta}}^{(s)} = \left(\left(\mathbf{F}^{(s)}\right)' \mathbf{V}_s^{-1} \mathbf{F}^{(s)} \right)^{-1} \left(\mathbf{F}^{(s)}\right)' \mathbf{V}_s^{-1} \mathbf{z}^{(s)}$ and $\hat{\boldsymbol{\beta}}_s$ are the p_s last components of $\bar{\boldsymbol{\beta}}^{(s)}$.

For the univariate case $s = 1$, the predictive mean (2.12) and variance (2.13) are identical to the ones of the Universal kriging (1.20) and (1.21).

We highlight that closed form formulas can also be derived for the predictive distribution when a Gaussian prior distribution is considered for $\boldsymbol{\beta}^{(s)}$. The universal co-kriging equations are then similar to the ones presented in Subsection 1.2.2.

2.2 Co-kriging in geostatistics

We present in this section the geostatistical approach to deal with multivariate objective functions. It is the natural extension to the one presented in Subsection 1.5.1. Similarly to the Bayesian scheme presented in Section 2.1 we want to predict a principal component $z_s(x)$ by taking into account the secondary components $(z_i(x))_{i=1,\dots,s-1}$. As previously, the vector of functions $(z_i)_{i=1,\dots,s}$ is modeled with a multivariate Gaussian process $(Z_i(x))_{i=1,\dots,s}$ with mean $\mathbf{m}(x)$ (2.1) and matrix-valued covariance function $\mathbf{V}(x,\tilde{x})$ (2.2). Nevertheless, in order to simplify the equations, we present the bivariate case $s = 2$. The extension for any s is straightforward.

2.2.1 Simple co-kriging

Let us consider the bivariate Gaussian process $(Z_1(x), Z_2(x))$ and the corresponding Gaussian random vector $(\mathbf{Z}_1^{n_1}, \mathbf{Z}_2^{n_2})$ where $\mathbf{Z}_i^{n_i} := Z_i(\mathbf{D}^i)$, $i = 1, 2$. Furthermore, we consider $\mathbf{M}_i := \mathbf{m}(\mathbf{D}^i) = \mathbf{f}_i'(\mathbf{D}^i)\boldsymbol{\beta}_i := \mathbf{F}_i\boldsymbol{\beta}_i$ where $\mathbf{F}_i$ is a matrix of size $n_i \times p_i$, $i = 1, 2$.

In a simple co-kriging case, the coefficients $(\boldsymbol{\beta}_i)_{i=1,2}$ are considered as known. Therefore, we can suppose them equal to zero without loss of generality. Let us consider that we want to predict the principal component $Z_2(x)$. We consider the following linear unbiased predictor:

$$\hat{Z}_2(x) = \sum_{i=1}^{n_2} \alpha_i Z_2(x_i^{(2)}) + \sum_{i=1}^{n_1} \gamma_i Z_1(x_i^{(1)}) = (\boldsymbol{\alpha}^{n_2})' \mathbf{Z}_2^{n_2} + (\boldsymbol{\gamma}^{n_1})' \mathbf{Z}_1^{n_1}, \tag{2.14}$$

where $\boldsymbol{\alpha}^{n_2} = [\alpha_i]_{i=1,\dots,n_2}$ and $\boldsymbol{\gamma}^{n_1} = [\gamma_i]_{i=1,\dots,n_1}$. Like in Subsection 1.5.1 we want to find the coefficients $\boldsymbol{\alpha}^{n_2}$ and $\boldsymbol{\gamma}^{n_1}$ minimizing

$$\begin{aligned} \mathbb{E}\left[\left(Z_2(x) - \hat{Z}_2(x)\right)^2\right] &= k_{22}(x,x) + \operatorname{var}\left(\hat{Z}_2(x)\right) \\ &\quad -2\left(\mathbf{k}_{22}'(x)\boldsymbol{\alpha}^{n_2} + \mathbf{k}_{21}'(x)\boldsymbol{\gamma}^{n_1}\right), \end{aligned}$$

where

$$\operatorname{var}\left(\hat{Z}_2(x)\right) = (\boldsymbol{\alpha}^{n_2})' \mathbf{K}_{22}\boldsymbol{\alpha}^{n_2} + (\boldsymbol{\gamma}^{n_1})' \mathbf{K}_{11}\boldsymbol{\gamma}^{n_1} + 2\,(\boldsymbol{\alpha}^{n_2})' \mathbf{K}_{21}\boldsymbol{\gamma}^{n_1},$$

$\mathbf{k}_{2j}(x) = [k_{2j}(x, x_k^{(j)})]_{k=1,\dots,n_j}$, $\mathbf{k}_{j2}(x) = [k_{j2}(x_k^{(j)}, x)]_{k=1,\dots,n_j}$ and $\mathbf{K}_{ij} = [k_{ij}(x_k^{(i)}, x_l^{(j)})]_{\substack{k=1,\dots,n_i,\\ l=1,\dots,n_j}}$, $i, j = 1, 2$. We note that $k_{12}(x,\tilde{x}) = k_{21}(\tilde{x}, x)$ implies that $\mathbf{K}_{12} = \mathbf{K}_{21}'$ and $\mathbf{k}_{12}(x) = \mathbf{k}_{21}(x)$. We can derive the mean squared error with respect to $\boldsymbol{\alpha}^{n_2}$ and $\boldsymbol{\gamma}^{n_1}$. Setting the derivatives equal to zero, we obtain that the minimum satisfies the following system of equations:

$$\begin{cases} (\boldsymbol{\alpha}^{n_2})' \mathbf{K}_{22} + (\boldsymbol{\gamma}^{n_1})' \mathbf{K}_{12} = \mathbf{k}_{22}'(x) \\ (\boldsymbol{\gamma}^{n_1})' \mathbf{K}_{11} + (\boldsymbol{\alpha}^{n_2})' \mathbf{K}_{21} = \mathbf{k}_{21}'(x) \end{cases}. \tag{2.15}$$

Therefore, we can deduce $\boldsymbol{\alpha}^{n_2}$ and $\boldsymbol{\gamma}^{n_1}$ from the following linear problem:

$$\begin{pmatrix} \mathbf{K}_{22} & \mathbf{K}_{21} \\ \mathbf{K}_{12} & \mathbf{K}_{11} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}^{n_2} \\ \boldsymbol{\gamma}^{n_1} \end{pmatrix} = \begin{pmatrix} \mathbf{k}_{22}(x) \\ \mathbf{k}_{21}(x) \end{pmatrix}.$$

The estimator is thus given by the equation

$$\hat{Z}_2(x) = \begin{pmatrix} \mathbf{k}'_{22}(x) & \mathbf{k}'_{21}(x) \end{pmatrix} \begin{pmatrix} \mathbf{K}_{22} & \mathbf{K}_{21} \\ \mathbf{K}_{21} & \mathbf{K}_{11} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{Z}_2^{n_2} \\ \mathbf{Z}_1^{n_1} \end{pmatrix}, \tag{2.16}$$

and the predictive variance $s^2_{SK}(x) = \mathbb{E}\left[\left(Z_2(x) - \hat{Z}_2(x)\right)^2\right]$ is

$$s^2_{SK}(x) = k_{22}(x,x) - \begin{pmatrix} \mathbf{k}'_{22}(x) & \mathbf{k}'_{21}(x) \end{pmatrix} \begin{pmatrix} \mathbf{K}_{22} & \mathbf{K}_{21} \\ \mathbf{K}_{12} & \mathbf{K}_{11} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{k}_{22}(x) \\ \mathbf{k}_{21}(x) \end{pmatrix}. \tag{2.17}$$

Conditionally to the observed values, the predictive means (2.16) and (2.4) are identical when we consider $\mathbf{m}(x) = 0$. Furthermore, the predictive variances (2.17) and (2.5) are identical too. Therefore, the predictive distributions of the Bayesian and the best linear unbiased predictor are identical.

We have shown that the Bayesian simple co-kriging and the one introduced in the geostatistical literature give the same predictive distributions in the bivariate case. In fact, the generalization of this result for any multivariate function is straightforward.

2.2.2 Universal co-kriging

We use in this subsection the same notations as in Subsection 2.2.2. In a universal co-kriging context, the coefficients $(\boldsymbol{\beta}_i)_{i=1,2}$ are unknown and have to be taken into account in the constraint of unbiasedness. Let us consider that we want to predict the principal component $Z_2(x)$. We consider the following linear predictor:

$$\hat{Z}_2(x) = \sum_{i=1}^{n_2} \alpha_i Z_2(x_i^{(2)}) + \sum_{i=1}^{n_1} \gamma_i Z_1(x_i^{(1)}) = (\boldsymbol{\alpha}^{n_2})' \mathbf{Z}_2^{n_2} + (\boldsymbol{\gamma}^{n_1})' \mathbf{Z}_1^{n_1}. \tag{2.18}$$

Like in Subsection 2.2.1 we want to find the coefficients $\boldsymbol{\alpha}^{n_2}$ and $\boldsymbol{\gamma}^{n_1}$ minimizing

$$\begin{aligned} \mathbb{E}\left[\left(Z_2(x) - \hat{Z}_2(x)\right)^2\right] &= k_{22}(x,x) + -2 \begin{pmatrix} \mathbf{k}'_{22}(x) & \mathbf{k}'_{21}(x) \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}^{n_2} \\ \boldsymbol{\gamma}^{n_1} \end{pmatrix} \\ &\quad + \begin{pmatrix} (\boldsymbol{\alpha}^{n_2})' & (\boldsymbol{\gamma}^{n_1})' \end{pmatrix} \begin{pmatrix} \mathbf{K}_{22} & \mathbf{K}_{21} \\ \mathbf{K}_{12} & \mathbf{K}_{11} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}^{n_2} \\ \boldsymbol{\gamma}^{n_1} \end{pmatrix}. \end{aligned}$$

Furthermore, the constraint of unbiasedness implies that

$$(\boldsymbol{\alpha}^{n_2})' \mathbf{F}_2 \boldsymbol{\beta}_2 + (\boldsymbol{\gamma}^{n_1})' \mathbf{F}_1 \boldsymbol{\beta}_1 = \mathbf{f}'_2(x)\boldsymbol{\beta},$$

which is generally translated in geostatistic by the following conditions (see [Wackernagel, 2003])

$$\begin{cases} (\boldsymbol{\alpha}^{n_2})' \mathbf{F}_2 = \mathbf{f}'_2(x) \\ (\boldsymbol{\gamma}^{n_1})' \mathbf{F}_1 = 0 \end{cases}. \tag{2.19}$$

We use the Lagrangian formulation of the problem to minimize $\mathbb{E}\left[\left(Z_2(x)-\hat{Z}_2(x)\right)^2\right]$ under the constraints (2.19):

$$\mathbb{E}\left[\left(Z_2(x)-\hat{Z}_2(x)\right)^2\right]+2\boldsymbol{\lambda}_1(\mathbf{F}_2'\boldsymbol{\alpha}^{n_2}-\mathbf{f}_2(x))+2\boldsymbol{\lambda}_2\mathbf{F}_1'\boldsymbol{\gamma}^{n_1},$$

where $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are the Lagrangien multipliers. We obtain the following linear system by calculating the gradients with respect to $(\boldsymbol{\alpha}^{n_2},\boldsymbol{\gamma}^{n_1},\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)$ and setting them equal to zero

$$\begin{pmatrix}\mathbf{K}_{22} & \mathbf{K}_{21} & \mathbf{F}_2' & 0\\ \mathbf{K}_{12} & \mathbf{K}_{11} & 0 & \mathbf{F}_1'\\ \mathbf{F}_2 & 0 & 0 & 0\\ 0 & \mathbf{F}_1 & 0 & 0\end{pmatrix}\begin{pmatrix}\boldsymbol{\alpha}^{n_2}\\ \boldsymbol{\gamma}^{n_1}\\ \lambda_1\\ \lambda_2\end{pmatrix}=\begin{pmatrix}\mathbf{k}_{22}(x)\\ \mathbf{k}_{21}(x)\\ \mathbf{f}_2(x)\\ 0\end{pmatrix}.$$

Let us introduce the following notations:

$$\mathbf{V}_2=\begin{pmatrix}\mathbf{K}_{22} & \mathbf{K}_{21}\\ \mathbf{K}_{12} & \mathbf{K}_{11}\end{pmatrix},\quad \mathbf{F}^{(2)}=\begin{pmatrix}\mathbf{F}_2 & 0\\ 0 & \mathbf{F}_1\end{pmatrix},\quad \mathbf{Z}^{(2)}=\begin{pmatrix}\mathbf{Z}_2^{n_2}\\ \mathbf{Z}_1^{n_1}\end{pmatrix}$$

and $\mathbf{k}_2'(x)=\begin{pmatrix}\mathbf{k}_{22}'(x) & \mathbf{k}_{21}'(x)\end{pmatrix}$. After some algebra, we find that the estimator is given by

$$\hat{Z}_2(x)=\mathbf{f}_2'(x)\hat{\boldsymbol{\beta}}_2+\mathbf{k}_2'(x)\mathbf{V}_2^{-1}\left(\mathbf{Z}^{(2)}-\mathbf{F}^{(2)}\hat{\boldsymbol{\beta}}\right),\tag{2.20}$$

where

$$\hat{\boldsymbol{\beta}}=\left(\left(\mathbf{F}^{(2)}\right)'\mathbf{V}_2^{-1}\mathbf{F}^{(2)}\right)^{-1}\left(\mathbf{F}^{(2)}\right)'\mathbf{V}_2^{-1}\mathbf{Z}^{(2)}\tag{2.21}$$

and $\hat{\boldsymbol{\beta}}_2$ are the p_2 first components of $\hat{\boldsymbol{\beta}}$.

Then, denoting the predictive variance $s_{UK}^2(x)=\mathbb{E}\left[\left(Z_2(x)-\hat{Z}_2(x)\right)^2\right]$ and noticing that $\begin{pmatrix}(\boldsymbol{\alpha}^{n_2})' & (\boldsymbol{\gamma}^{n_1})'\end{pmatrix}\mathbf{F}^{(2)}=\begin{pmatrix}\mathbf{f}_2'(x) & 0\end{pmatrix}$, we have:

$$s_{UK}^2(x)=k_{22}(x,x)-\begin{pmatrix}\mathbf{k}_2'(x) & \mathbf{f}_2'(x) & 0\end{pmatrix}\begin{pmatrix}\mathbf{V}_2 & \mathbf{F}^{(2)}\\ \left(\mathbf{F}^{(2)}\right)' & 0\end{pmatrix}^{-1}\begin{pmatrix}\mathbf{k}_2(x)\\ \mathbf{f}_2(x)\\ 0\end{pmatrix}.\tag{2.22}$$

In the bivariate case, the predictive means (2.20) and (2.12) and the predictive variances (2.22) and (2.13) are identical. Therefore, the predictive distributions of the Bayesian and the best linear unbiased predictor are identical.

For the bivariate case the Bayesian and the geostatistical universal co-kriging provide the same predictive distribution. Furthermore, this result is directly generalizable for any multivariate cases.

2.3 Admissible matrix-valued covariance kernels

In Section 2.1 we have presented the equations of the simple and universal co-kriging which come from the Gaussian assumption for the multivariate stochastic process $\mathbf{Z}(x) = (Z_1(x), \dots, Z_s(x))$, $s \in \mathbb{N}^*$ with mean $\mathbf{m}(x)$ and matrix-valued covariance matrix $\mathbf{V}(x, \tilde{x})$ such that

$$\mathbf{V}(x, \tilde{x}) = \begin{pmatrix} k_{11}(x, \tilde{x}; \boldsymbol{\theta}_{11}) & \dots & k_{1s}(x, \tilde{x}; \boldsymbol{\theta}_{1s}) \\ \vdots & \ddots & \vdots \\ k_{s1}(x, \tilde{x}; \boldsymbol{\theta}_{s1}) & \dots & k_{ss}(x, \tilde{x}; \boldsymbol{\theta}_{ss}) \end{pmatrix}.$$

A valid covariance structure $\mathbf{V}(x, \tilde{x})$ must satisfy the condition of positive definiteness. Namely, for any $(\mathbf{D}^i)_{i=1,\dots,s}$ where $\mathbf{D}^i = (x_1^{(i)}, \dots, x_{n_i}^{(i)})$, $x_j^{(n_i)} \in \mathbb{R}^d$, $j = 1, \dots, n_i$, $i = 1, \dots, s$, the following covariance matrix

$$\mathbf{V}_s = \begin{pmatrix} k_{11}(\mathbf{D}_1, \mathbf{D}_1; \boldsymbol{\theta}_{11}) & \dots & k_{1s}(\mathbf{D}_1, \mathbf{D}_s; \boldsymbol{\theta}_{1s}) \\ \vdots & \ddots & \vdots \\ k_{s1}(\mathbf{D}_s, \mathbf{D}_1; \boldsymbol{\theta}_{s1}) & \dots & k_{ss}(\mathbf{D}_s, \mathbf{D}_s; \boldsymbol{\theta}_{ss}) \end{pmatrix} = \begin{pmatrix} \mathbf{K}_{11} & \dots & \mathbf{K}_{1s} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{s1} & \dots & \mathbf{K}_{ss} \end{pmatrix}$$

has to be positive definite. We note that $\mathbf{V}_s$ is the covariance matrix of $\mathbf{Z}^{(s)} = ((\mathbf{Z}_1^{n_1})', \dots, (\mathbf{Z}_s^{n_s})')'$ the values of $(Z_i(x))_{i=1,\dots,s}$ at points in $(\mathbf{D}^i)_{i=1,\dots,s}$. We present two methods to ensure the positive definiteness of $\mathbf{V}(x, \tilde{x})$. The first one in Subsection 2.3.1 is the approach commonly used in geostatistics. The second one in Subsection 2.3.2 uses an extension of the Bochner's theorem (see Theorem 1.3, Subsection 1.4.2).

2.3.1 Linear transformation of a multivariate Gaussian process

A first method to define admissible matrix-valued covariance kernels $\mathbf{V}(x, \tilde{x})$ is to notice that any linear transformation of a multivariate Gaussian process is a multivariate Gaussian process. We derive in this subsection some examples of valid covariance structures using this property.

Linear model of coregionalization

Let us consider the multivariate Gaussian process $\boldsymbol{\delta}(x) = (\delta_1(x), \dots, \delta_t(x))$ where $(\delta_i(x))_{i=1,\dots,t}$ are univariate Gaussian processes with covariance kernel $k_i(x, \tilde{x})$ and such that $\delta_i(x) \perp \delta_j(x)$ for all $i, j = 1, \dots, t$, $i \neq j$. We note that the independence assumption ensures the normality of $\boldsymbol{\delta}(x)$. Then, any linear combinations of $(\delta_i(x))_{i=1,\dots,t}$ is a multivariate Gaussian process, i.e. if we define for all $i = 1, \dots, s$, with $s \in \mathbb{N}^*$, the following random process

$$Z_i(x) = \sum_{j=1}^{t} \alpha_j^i \delta_j(x),$$

then $\mathbf{Z}(x) = (Z_i(x))_{i=1,\dots,s}$ is a multivariate Gaussian process. Furthermore, we have

$$\operatorname{cov}(Z_i(x), Z_j(\tilde{x})) = \sum_{k=1}^{t} \alpha_k^i \alpha_k^j \operatorname{cov}(\delta_k(x), \delta_k(\tilde{x})) = \sum_{k=1}^{t} \alpha_k^i \alpha_k^j k_k(x, \tilde{x}).$$

Therefore, the covariance structure of $\mathbf{Z}(x)$ is

$$\mathbf{V}(x,\tilde{x}) = \sum_{k=1}^{t} \left[\alpha_k^i \alpha_k^j\right]_{i,j=1,\dots,s} k_k(x,\tilde{x}),$$

where the matrix $\left[\alpha_k^i \alpha_k^j\right]_{i,j=1,\dots,s}$ is nonnegative definite since it can be written with the following form for all $k = 1,\dots,t$

$$\begin{pmatrix} \alpha_k^1\alpha_k^1 & \dots & \alpha_k^1\alpha_k^s \\ \vdots & \ddots & \vdots \\ \alpha_k^s\alpha_k^1 & \dots & \alpha_k^s\alpha_k^s \end{pmatrix} = \begin{pmatrix} \alpha_k^1 \\ \vdots \\ \alpha_k^s \end{pmatrix} \begin{pmatrix} \alpha_k^1 & \vdots & \alpha_k^s \end{pmatrix}.$$

This approach is referred as the linear model of coregionalization and is frequently used in geostatistics (see [Goulard and Voltz, 1992] and [Wackernagel, 2003]). For this model, the smoothness of any Gaussian process $Z_i(x)$, $i = 1,\dots,s$, is the one of the roughest latent process $\delta_j(x)$, $j = 1,\dots,t$ such that α_j^i is not zero.

Convolved Gaussian white noise process

As presented in point 3. in the introduction of Section 1.4, a Gaussian process can be defined with the following form:

$$Z(x) = \int k(x,u)\, dW(u),$$

where $W(x)$ is the Wiener process. Furthermore, $Z(x)$ has the covariance kernel $\int k(x,u)k(u,\tilde{x})\, du$. If we consider t independent Gaussian white noise processes $(W_i(x))_{i=1,\dots,t}$, then by applying the linear operators $(L_jW_i)(x) = \int k_i^j(x,u)W_i(u)\, du = Z_i^j(x)$, $i = 1,\dots,t$, $j = 1,\dots,s$, $s \in \mathbb{N}^*$, the following multivariate stochastic process is still Gaussian:

$$(Z_i^j(x))_{\substack{i=1,\dots,t\,,\\ j=1,\dots,s}}$$

with covariance structure such that

$$\operatorname{cov}\left(Z_i^j(x), Z_k^l(\tilde{x})\right) = \delta_{i=k} \int k_i^j(x,u)k_k^l(u,\tilde{x})\, du.$$

This technique was suggested by [Boyle and Frean, 2005] to deal with multiple output functions. We present below their approach for the bivariate case. Let us consider three independent Gaussian white noise processes $(W_i(x))_{i=1,\dots,3}$ and four covariance kernels $(k_i(x,\tilde{x}))_{i=1,2}$ and $(h_i(x,\tilde{x}))_{i=1,2}$. Then we can define the four following Gaussian processes:

$$\begin{aligned} V_1(x) &= \int h_1(x,u)W_1(u)\, du, \\ Y_1(x) &= \int k_1(x,u)W_2(u)\, du, \\ Y_2(x) &= \int k_2(x,u)W_2(u)\, du, \\ V_2(x) &= \int h_2(x,u)W_3(u)\, du. \end{aligned}$$

We note that the final multivariate random process $(V_1(x), Y_1(x), Y_2(x), V_2(x)$ is Gaussian since it is a linear transformation of a multivariate Gaussian process. Furthermore, its components are all independent except for $Y_1(x)$ and $Y_2(x)$ since they come from the same Gaussian white noise $W_2(x)$. Then, considering two independents Gaussian white noise processes $(\varepsilon_i(x))_{i=1,2}$, one can define the following bivariate Gaussian process:

$$\begin{cases} Z_1(x) &= V_1(x) + Y_1(x) + \sigma_1^2 \varepsilon_1(x) \\ Z_2(x) &= V_2(x) + Y_2(x) + \sigma_2^2 \varepsilon_2(x) \end{cases},$$

where

$$\begin{aligned} \operatorname{cov}(Z_1(x), Z_1(\tilde{x})) &= \int h_1(x,u) h_1(u,\tilde{x})\, du + \int k_1(x,u) k_1(u,\tilde{x})\, du + \sigma_1^2 \delta_{x=\tilde{x}}, \\ \operatorname{cov}(Z_2(x), Z_2(\tilde{x})) &= \int h_2(x,u) h_2(u,\tilde{x})\, du + \int k_2(x,u) k_2(u,\tilde{x})\, du + \sigma_2^2 \delta_{x=\tilde{x}}, \\ \operatorname{cov}(Z_1(x), Z_2(\tilde{x})) &= \int k_1(x,u) k_2(u,\tilde{x})\, du. \end{aligned}$$

For some kernels as the squared exponential one, closed form expressions can be obtained for these integrals (see [Boyle and Frean, 2005]).

Gaussian processes with zero mean

Following the work of [Durrande, 2011], we present here another approach than the one presented in Subsection 1.5.2 to deal with zero-mean Gaussian processes. We consider a Gaussian process $Z(x)$ with mean $\mathbf{f}'(x)\boldsymbol{\beta}$ and covariance kernel $k(x,\tilde{x})$, $x \in Q \subset \mathbb{R}^d$. Furthermore, we consider the following linear transformation of $Z(x)$:

$$LZ(x) = \int_Q Z(u)\, du.$$

Since any linear transformation of a Gaussian process is Gaussian, we have

$$\begin{pmatrix} Z(x) \\ \int Z(u)\, du \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{f}'(x)\boldsymbol{\beta} \\ \int \mathbf{f}'(u)\boldsymbol{\beta}\, du \end{pmatrix}, \begin{pmatrix} k(x,x) & \int k(x,u)\, du \\ \int k(u,x)\, du & \int\int k(u,v)\, du\, dv \end{pmatrix} \right)$$

and thus the distribution of $[Z(x) | \int Z(u)\, du = 0]$ is Gaussian with mean

$$\mathbf{f}'(x)\boldsymbol{\beta} - \int k(x,u)\, du \left(\int\int k(u,v)\, du\, dv \right)^{-1} \int \mathbf{f}'(u)\boldsymbol{\beta}\, du$$

and variance

$$k(x,x) - \int k(x,u)\, du \left(\int\int k(u,v)\, du\, dv \right)^{-1} \int k(u,x)\, du.$$

2.3.2 Spectral analysis of a multivariate covariance structure

Another approach to ensure the positive definiteness for $\mathbf{V}(x, \tilde{x})$ is to consider the stationary case $\mathbf{V}(x, \tilde{x}) = \mathbf{V}(h)$ with $h = x - \tilde{x}$ and the following generalization of the Bochner's Theorem for multivariate Gaussian processes.

Theorem 2.1 (Multivariate Bochner's Theorem)**.** *For any continuous positive definite matrix-valued $\mathbf{V}(h)$ from $\mathbb{R}^d$ into $\mathbb{R}^s \times \mathbb{R}^s$, such that*

$$\mathbf{V}(h) = \begin{pmatrix} k_{11}(h; \boldsymbol{\theta}_{11}) & \dots & k_{1s}(h; \boldsymbol{\theta}_{1s}) \\ \vdots & \ddots & \vdots \\ k_{s1}(h; \boldsymbol{\theta}_{s1}) & \dots & k_{ss}(h; \boldsymbol{\theta}_{ss}) \end{pmatrix},$$

there exists a unique matrix valued positive finite measure μ such that $\mathbf{V}(h) = \int_{\mathbb{R}^d} e^{2\pi i \langle w,h \rangle} \, d\mu(w)$. Furthermore, if $\mu(w)$ has a spectral density $\mathbf{S}(w)$ - $\mathbf{S}(w)$ is non-negative definite - with

$$\mathbf{S}(w) = \begin{pmatrix} S_{11}(w; \boldsymbol{\theta}_{11}) & \dots & S_{1s}(w; \boldsymbol{\theta}_{1s}) \\ \vdots & \ddots & \vdots \\ S_{s1}(w; \boldsymbol{\theta}_{s1}) & \dots & S_{ss}(w; \boldsymbol{\theta}_{ss}) \end{pmatrix},$$

where $S_{ij}(w; \boldsymbol{\theta}_{ij})$ is the power spectrum of $k_{ij}(h; \boldsymbol{\theta}_{ij})$, then $\mathbf{V}(h) = \int_{\mathbb{R}^d} e^{2\pi i \langle w,h \rangle} \mathbf{S}(w) \, dw$.

Therefore, to define a valid covariance structure $\mathbf{V}(h)$, we have to ensure that $\forall w \in \mathbb{R}^d$ $\mathbf{S}(w) \geq 0$ is nonnegative.

An example of valid covariance structure

The example presented below comes from the article of [Gneiting et al., 2010]. Let us consider the covariance $\mathbf{V}(h)$ such that

$$k_{ij}(h) = (c_i * c_j)(h),$$

where $(c_i)_{i=1,\dots,s}$ are square integrable functions. Then, we have

$$k_{ij}(h) = \mathcal{F}^{-1}(\mathcal{F}(c_i)\mathcal{F}(c_j))(h),$$

where $\mathcal{F}$ stands for the Fourier transform. The spectral density of $k_{ij}(h)$ is $S_{ij}(w) = f_i(w) f_j(w)$ where $f_i(w) = \mathcal{F}(c_i)$. Therefore, the matrix of the spectral densities is

$$\mathbf{S}(w) = \begin{pmatrix} f_1(w) f_1(w) & \dots & f_1(w) f_s(w) \\ \vdots & \ddots & \vdots \\ f_s(w) f_j(w) & \dots & f_s(w) f_s(w) \end{pmatrix} = \mathbf{f}(w)\mathbf{f}'(w),$$

with $\mathbf{f}'(w) = (f_1(w), \dots, f_s(w))$. This ensures the property $\mathbf{S}(w)$ is nonnegative.

Valid cross-covariance functions for bivariate random fields

We give here another example inspired by the article of [Gneiting et al., 2010]. Let us suppose a bivariate Gaussian process $\mathbf{Z}(x) = (Z_1(x), Z_2(x))$ with covariance structure :

$$\begin{aligned} k_{11}(h) &= \sigma_1^2 k_1(h;\boldsymbol{\theta}_1), \\ k_{22}(h) &= \sigma_2^2 k_2(h;\boldsymbol{\theta}_2), \\ k_{12}(h) &= \rho_{12}\sigma_1\sigma_2 k_{12}(h;\boldsymbol{\theta}_{12}), \\ k_{21}(h) &= k_{12}(h). \end{aligned}$$

with $h = x - \tilde{x}$, $x, \tilde{x} \in \mathbb{R}^d$. Then, we have :

$$\mathbf{S}(w) = \begin{pmatrix} \sigma_1^2\mathcal{F}(k_1(h;\boldsymbol{\theta}_1))(w) & \rho_{12}\sigma_1\sigma_2\mathcal{F}(k_{12}(h;\boldsymbol{\theta}_{12}))(w) \\ \rho_{12}\sigma_1\sigma_2\mathcal{F}(k_{12}(h;\boldsymbol{\theta}_{12}))(w) & \sigma_2^2\mathcal{F}(k_2(h;\boldsymbol{\theta}_2))(w) \end{pmatrix}.$$

To ensure the nonnegative definiteness, the following inequality must be satisfied for all $w \in \mathbb{R}^d$

$$|\rho_{12}\mathcal{F}(k_{12}(h;\boldsymbol{\theta}_{12}))(w)|^2 \leq \mathcal{F}(k_1(h;\boldsymbol{\theta}_1))(w)\mathcal{F}(k_2(h;\boldsymbol{\theta}_2))(w). \tag{2.23}$$

The isotropic Gaussian kernel class. Let us suppose that $k_1(h;\theta) = k_2(h;\theta) = k_{12}(h;\theta) = k(h;\theta)$ with :

$$k(h;\theta) = \exp\left(-\frac{||h||^2}{2\theta^2}\right).$$

According to Subsection 1.4.2, we have :

$$S(w) = \mathcal{F}(k(h,\theta)) = (2\pi\theta^2)^{d/2}\exp(-2\pi^2\theta^2||w||^2).$$

The condition (2.23) becomes $\forall t \geq 0$:

$$\rho_{12}^2(\theta_{12}^2)^d\exp(-4\pi^2\theta_{12}^2 t) \leq (\theta_1^2)^{d/2}\exp(-2\pi^2\theta_1^2 t)(\theta_2^2)^{d/2}\exp(-2\pi^2\theta_2^2 t).$$

Therefore, we have to satisfy the following condition to respect the nonnegative definiteness property $\forall t \geq 0$:

$$\rho_{12}^2 \leq \frac{(\theta_1^2\theta_2^2)^{d/2}}{(\theta_{12}^2)^d}\inf_{t\geq 0}\exp(-2\pi^2 t(\theta_1^2 - 2\theta_{12}^2 + \theta_2^2)). \tag{2.24}$$

This means that $\theta_1^2 - 2\theta_{12}^2 + \theta_2^2 > 0$ implies $\rho_{12} = 0$ and $\theta_1^2 - 2\theta_{12}^2 + \theta_2^2 \leq 0$ leads to $\rho_{12}^2 \leq (\theta_1^2\theta_2^2)^{d/2}/(\theta_{12}^2)^d$.

The Matérn kernel class. We still consider that $k_1(h;\theta) = k_2(h;\theta) = k_{12}(h;\theta) = k(h;\theta)$. As presented in Subsection 1.4.2, the Matérn kernel class is given by

$$k(h;\theta) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}||h||}{\theta}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}||h||}{\theta}\right),$$

with the power spectrum

$$S(\omega) = \frac{2^d \pi^{d/2} \Gamma(\nu + d/2)(2\nu)^\nu}{\Gamma(\nu)\theta^{2\nu}} \left(\frac{2\nu}{\theta^2} + 4\pi^2 ||w||^2 \right)^{-(\nu+d/2)}.$$

The condition (2.23) gives that:

$$\begin{aligned} \rho_{12}^2 \quad \leq \quad & \frac{\Gamma(\nu_1 + d/2)\Gamma(\nu_2 + d/2)}{\Gamma(\nu_{12} + d/2)^2} \frac{(2\nu_1)^{\nu_1}(2\nu_2)^{\nu_2}}{(2\nu_{12})^{2\nu_{12}}} \frac{\Gamma(\nu_{12})^2}{\Gamma(\nu_1)\Gamma(\nu_2)} \\ & \times \frac{\theta_{12}^{4\nu_{12}}}{\theta_1^{2\nu_1}\theta_2^{2\nu_2}} \inf_{t \geq 0} \left(\frac{2\nu_{12}}{\theta_{12}^2} + 4\pi^2 t \right)^{2\nu_{12}+d} \left(\frac{2\nu_1}{\theta_1^2} + 4\pi^2 t \right)^{-\nu_1 - d/2} \left(\frac{2\nu_2}{\theta_2^2} + 4\pi^2 t \right)^{-\nu_2 - d/2}. \end{aligned}$$

This condition is presented in [Gneiting et al., 2010]. It leads the following cases:

1. $\nu_{12} < \frac{1}{2}(\nu_1 + \nu_2) \Rightarrow \rho_{12} = 0$.
2. $\nu_{12} = \frac{1}{2}(\nu_1 + \nu_2), \quad \frac{\theta_{12}^2}{\nu_1 + \nu_2} > \max\left(\frac{\theta_1^2}{2\nu_1}, \frac{\theta_2^2}{2\nu_2} \right) \Rightarrow$

$$\rho_{12}^2 < \left(\frac{\theta_1 \theta_2}{\theta_{12}^2} \right)^d \frac{\Gamma(\nu_1 + d/2)\Gamma(\nu_2 + d/2)\Gamma(\nu_{12})^2}{\Gamma(\nu_{12} + d/2)^2 \Gamma(\nu_1)\Gamma(\nu_2)} \frac{(\nu_1 + \nu_2)^d}{(4\nu_1\nu_2)^{d/2}}.$$

3. $\nu_{12} = \frac{1}{2}(\nu_1 + \nu_2), \frac{\theta_{12}^2}{\nu_1 + \nu_2} < \min\left(\frac{\theta_1^2}{2\nu_1}, \frac{\theta_2^2}{2\nu_2} \right) \Rightarrow$

$$\rho_{12}^2 < \left(\frac{2\theta_{12}^2 \nu_1}{(\nu_1 + \nu_2)\theta_1^2} \right)^{\nu_1} \left(\frac{2\theta_{12}^2 \nu_2}{(\nu_1 + \nu_2)\theta_2^2} \right)^{\nu_2} \frac{\Gamma(\nu_1 + d/2)\Gamma(\nu_2 + d/2)\Gamma(\nu_{12})^2}{\Gamma(\nu_{12} + d/2)^2 \Gamma(\nu_1)\Gamma(\nu_2)}.$$

4. $\nu_{12} = \frac{1}{2}(\nu_1 + \nu_2), \min\left(\frac{\theta_1^2}{2\nu_1}, \frac{\theta_2^2}{2\nu_2} \right) < \theta_{12}^2 < \max\left(\frac{\theta_1^2}{2\nu_1}, \frac{\theta_2^2}{2\nu_2} \right) \Rightarrow$ the minimum is reached for $t = 0$ (case 3.), or for $t \to \infty$ (case 2.), or for:

$$t = \frac{a_1(2\nu_1 + d) + a_2(2\nu_2 + d) - 2a_{21}(\nu_1 + \nu_2 + d)}{2a_{12}(\nu_1 a_1 + \nu_2 a_2) + a_{12} d(a_1 + a_2) - 2a_1 a_2(\nu_1 + \nu_2 + d)},$$

where:

$$a_1 = \frac{\theta_1^2}{2\nu_1}, \qquad a_2 = \frac{\theta_2^2}{2\nu_2}, \qquad a_{12} = \frac{\theta_{12}^2}{\nu_1 + \nu_2}.$$

2.4 Co-kriging models using function derivatives

We introduce in this section a co-kriging model approach commonly used in the field of computer experiments. We have seen in the introduction of Section 1.4 that the mean square partial derivatives $\partial Z(x)/\partial x^i$, $x = (x^1, \ldots, x^d) \in \mathbb{R}^d$ of a Gaussian process $Z(x)$ exists if and only if its covariance kernel $k(x, \tilde{x})$ is twice differentiable with respect to x^i. As the differential operator is linear, if the covariance kernels are well defined, then the multivariate stochastic process $(Z(x), (\partial Z(x)/\partial x^i)_{i=1,\ldots,d})$ is Gaussian. Furthermore, we have the following cross covariances

$$\operatorname{cov}\left(Z(x), \frac{\partial Z(\tilde{x})}{\partial \tilde{x}^i} \right) = \frac{\partial k(x, \tilde{x})}{\partial \tilde{x}^i}, \tag{2.25}$$

$$\operatorname{cov}\left(\frac{\partial Z(x)}{\partial x^i}, \frac{\partial Z(\tilde{x})}{\partial \tilde{x}^j} \right) = \frac{\partial^2 k(x, \tilde{x})}{\partial x^i \partial \tilde{x}^j}. \tag{2.26}$$

with $i, j = 1, \ldots, d$. Now, let us consider that we want to surrogate an objective function $z(x)$ with a Gaussian process $Z(x)$ of mean $\mathbf{f}'(x)\boldsymbol{\beta}$ and covariance kernel $k(x, \tilde{x})$ and with respect to the partial derivatives of $z(x)$ (see [Morris et al., 1993] and [Mitchell et al., 1994]). We denote by $\mathbf{Z}^n$ the values of $Z(x)$ at points in $\mathbf{D}^n = \{x_1, \ldots, x_n\}$, such that $x_j = (x_j^1, \ldots, x_j^d) \in \mathbb{R}^d$, $j = 1, \ldots, n$ and by $\mathbf{Z}^n_{(i)}$ the values of $\partial Z(x)/\partial x^i$ at points in $\mathbf{D}^n$. Similarly, we denote by $\mathbf{z}^n$ and $\mathbf{z}^n_{(i)}$ the values of $z(x)$ and $\partial z(x)/\partial x^i$ at points in $\mathbf{D}^n$. The joint distribution of $(Z(x), \mathbf{Z}^n, (\mathbf{Z}^n_{(i)})_{i=1,\ldots,d})$ is the following multivariate normal distribution

$$\begin{pmatrix} Z(x) \\ \mathbf{Z}^n \\ \mathbf{Z}^n_{(1)} \\ \vdots \\ \mathbf{Z}^n_{(d)} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{f}'(x) \\ \mathbf{F}^n \\ \mathbf{F}^n_{(1)} \\ \vdots \\ \mathbf{F}^n_{(d)} \end{pmatrix} \boldsymbol{\beta}, \begin{pmatrix} k(x,x) & \mathbf{k}'(x) & \mathbf{k}'_{(1)}(x) & \ldots & \mathbf{k}'_{(d)}(x) \\ \mathbf{k}'(x) & \mathbf{K} & \mathbf{K}_{(01)} & \ldots & \mathbf{K}_{(0d)} \\ \mathbf{k}'_{(1)}(x) & \mathbf{K}_{(10)} & \mathbf{K}_{(11)} & \ldots & \mathbf{K}_{(1d)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{k}'_{(d)}(x) & \mathbf{K}_{(d0)} & \mathbf{K}_{(d1)} & \ldots & \mathbf{K}_{(dd)} \end{pmatrix} \right), \tag{2.27}$$

where

$$\begin{aligned}
&\mathbf{F}^n := \mathbf{f}'(\mathbf{D}^n), \\
&\mathbf{F}^n_{(l)} := [\partial \mathbf{f}'(x_i)/\partial x_i^l]_{i=1,\ldots,n;l=1,\ldots,d}, \\
&\mathbf{K} := [k(x_i, x_j)]_{i,j=1,\ldots,n}, \\
&\mathbf{K}_{(0l)} := [\partial k(x_i, x_j)/\partial x_j^l]_{i,j=1,\ldots,n;l=1,\ldots,d}, \\
&\mathbf{K}_{(kl)} := [\partial^2 k(x_i, x_j)/\partial x_i^k \partial x_j^l]_{i,j=1,\ldots,n;k,l=1,\ldots,d}, \\
&\mathbf{k}'(x) := [k(x, x_i)]_{i=1,\ldots,n}, \\
&\mathbf{k}'_{(l)}(x) := [\partial k(x, x_i)/\partial x_i^l]_{i=1,\ldots,n;l=1,\ldots,d},
\end{aligned}$$

The desired predictive distribution $[Z(x)|\mathbf{Z}^n, (\mathbf{Z}^n_{(i)})_{i=1,\ldots,d}]$ can be obtained following the same technique as the one presented in Subsection 1.2.1. Denoting by

$$\mathbf{h}(x) = \begin{pmatrix} \mathbf{k}(x) \\ \mathbf{k}_{(1)}(x) \\ \ldots \\ \mathbf{k}_{(d)}(x) \end{pmatrix}, \quad \mathbf{z} = \begin{pmatrix} \mathbf{z}^n \\ \mathbf{z}^n_{(1)} \\ \vdots \\ \mathbf{z}^n_{(d)} \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \mathbf{F}^n \\ \mathbf{F}^n_{(1)} \\ \vdots \\ \mathbf{F}^n_{(d)} \end{pmatrix}$$

and

$$\mathbf{V} = \begin{pmatrix} \mathbf{K} & \mathbf{K}_{(01)} & \ldots & \mathbf{K}_{(0d)} \\ \mathbf{K}_{(10)} & \mathbf{K}_{(11)} & \ldots & \mathbf{K}_{(1d)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{K}_{(d0)} & \mathbf{K}_{(d1)} & \ldots & \mathbf{K}_{(dd)} \end{pmatrix},$$

the predictive distribution is normal with mean:

$$\mathbf{f}'(x)\hat{\boldsymbol{\beta}} + \mathbf{h}'(x)\mathbf{V}^{-1}\left(\mathbf{z} - \mathbf{F}\hat{\boldsymbol{\beta}}\right), \tag{2.28}$$

where

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{F}'\mathbf{V}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}'\mathbf{V}^{-1}\mathbf{z},$$

and variance

$$k(x,x) - \begin{pmatrix} \mathbf{f}'(x) & \mathbf{h}'(x) \end{pmatrix} \begin{pmatrix} 0 & \mathbf{F}' \\ \mathbf{F} & \mathbf{V} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{f}(x) \\ \mathbf{h}(x) \end{pmatrix}. \tag{2.29}$$

The predictive mean is the surrogate model for $z(x)$ and the predictive variance represents the model mean squared error. Therefore, we can improve the surrogate model on $z(x)$ by considering its partial derivatives.

Example of Gaussian process regression using derivatives

Let us consider $Z(x)$ a Gaussian process with mean zero and covariance kernel $k(x,\tilde{x}) = \exp\left(-(x-\tilde{x})^2/2\theta^2\right)$ with $\theta = 0.1$ and $x \in [0,1]$. The covariance kernel $k(x,\tilde{x})$ being smooth, the Gaussian process $Z(x)$ is infinitely mean square differentiable. Furthermore, according to the previous developments we have:

$$\mathrm{cov}\left(Z(x), \frac{dZ}{d\tilde{x}}(\tilde{x})\right) = \frac{(x-\tilde{x})}{\theta^2} \exp\left(-\frac{(x-\tilde{x})^2}{2\theta^2}\right)$$

and

$$\mathrm{cov}\left(\frac{dZ}{dx}(x), \frac{dZ}{d\tilde{x}}(\tilde{x})\right) = \left(\frac{1}{\theta^2} - \frac{(x-\tilde{x})^2}{\theta^4}\right) \exp\left(-\frac{(x-\tilde{x})^2}{2\theta^2}\right).$$

Now let us condition $Z(x)$ at points $\mathbf{D} = (0.0, 0.2, 0.4, 0.7, 0.9)$ with $z(\mathbf{D}) = (-1, 2, 6, -2, 6)$ and $(dz/dx)(\mathbf{D}) = (0, -20, 40, 0, 15)$. Figure 2.3 illustrates the predictive means and confidence intervals obtained with a simple kriging and a simple co-kriging using the derivatives. We see in Figure 2.3 that the predictive means are significantly different between the simple

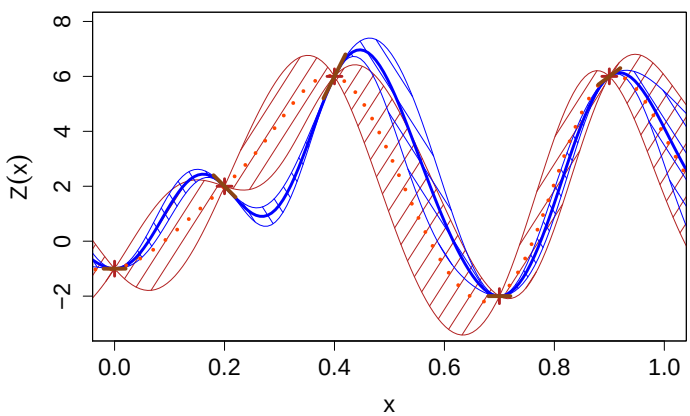


Figure 2.3: Example of Gaussian process regression using derivatives. The dotted line represents the kriging mean, the solid line represents the co-kriging using the derivatives. The shade areas represent the predictive means plus and minus twice the predictive standard deviations.

kriging and the simple co-kriging using the derivatives. Furthermore, the derivatives giving additional information, the confidence intervals for the co-kriging are naturally smaller than the ones of the kriging.

Part II

Contributions in Multi-fidelity Co-kriging models

Chapter 3

The AR(1) multi-fidelity co-kriging model

3.1 Introduction

Large computer codes are widely used in science and engineering to study physical systems since real experiments are often costly and sometimes impossible. Nevertheless, simulations can sometimes be costly and time-consuming as well. In this case, conception based on an exhaustive exploration of the input space of the code is generally impossible under reasonable time constraints. Therefore, a mathematical approximation of the output of the code - also called surrogate or metamodel - is often built with a few simulations to represent the real system.

The Gaussian Process regression presented in Chapter 1 is a particular class of surrogate models which makes the assumption that prior beliefs about the code can be modeled by a Gaussian Process. We focus here on this metamodel and on its extension to multiple response models (see Chapter 2).

Actually, a computer code can often be run at different levels of complexity and a hierarchy of levels of code can hence be obtained. The aim of this chapter is to study the use of several levels of a code to predict the output of a costly computer code (see [Le Gratiet, 2013]).

A first metamodel for multi-level computer codes was built by [Kennedy and O'Hagan, 2000] using a spatially stationary correlation structure. This multi-stage model is a particular case of the co-kriging one presented in Chapter 2. Then, [Forrester et al., 2007] went into more detail about the estimation of the model parameters. Furthermore,they presented the use of co-kriging for multi-fidelity optimization based on the EGO (Efficient Global Optimization) algorithm created by [Jones et al., 1998]. A Bayesian approach was also proposed by [Qian and Wu, 2008] which is computationally expensive and does not provide explicit formulas for the joint distribution of the parameters.

This chapter presents a new approach to estimate the parameters of the model which is effective when many levels of codes are available (see Subsection 3.6.1). In particular, it provides a closed form expression for the posterior distribution of the scale factor which is new and of great practical interest for accuracy and computational cost. Furthermore, this approach allows us to consider prior information in the estimation of the parameters. We also

address the problem of the inversion of the co-kriging covariance matrix when the number of levels is large. A solution to this problem is provided which shows that the inverse can be easily calculated (see Subsection 3.6.2). Finally, it is known that with a non-Bayesian approach, the variance of the predictive distribution may be underestimated [Kennedy and O'Hagan, 2000]. This chapter suggests a Bayesian modeling different from the one presented by [Qian and Wu, 2008] which provides an explicit representation of the joint distribution for the parameters and avoids prohibitive implementations (see Section 3.4.3).

3.2 Building a surrogate model based on a hierarchy of s levels of code

Let us assume that we have s levels of code $z_1(x), \ldots, z_s(x)$, $x \in \mathbb{R}^d, d > 0$. For all $t = 1, \ldots, s$ the tth scalar output $z_t(x)$ is modeled by $z_t(x) = Z_t(x, \omega)$ where $Z_t(x, \omega), \omega \in \Omega$ is a realization of the Gaussian process $Z_t(x)$. We will introduce below a consistent set of hypotheses so that the joint process $(Z_t(x))_{x \in \mathbb{R}^d, t=1,\ldots,s}$ is Gaussian given a certain set of parameters. [Kennedy and O'Hagan, 2000] suggest an autoregressive model to build a metamodel based on a multi-level computer code. Hence, we have a hierarchy of s levels of code - from the less accurate to the most accurate - and for each level, the conditional distribution of the Gaussian process $Z_t(x)$ knowing $Z_1(x), \ldots, Z_{t-1}(x)$ is entirely determined by $Z_{t-1}(x)$. Let us introduce here the mathematical formalism that we will use in this chapter.

$Q \subset \mathbb{R}^d$ is a compact subset of $\mathbb{R}^d$ representing the input space. For $t = 1, \ldots, s$, $\mathbf{D}_t = \{x_1^{(t)}, \ldots, x_{n_t}^{(t)}\}$ is the experimental design set at level t containing n_t points in Q. Let $\mathbf{Z}_t = Z_t(\mathbf{D}_t) = (Z_t(x_1^{(t)}), \ldots, Z_t(x_{n_t}^{(t)}))'$ be the random Gaussian vector containing the values of $Z_t(x)$ for $x \in \mathbf{D}_t$. Let $\mathbf{Z} = (\mathbf{Z}_1', \ldots, \mathbf{Z}_s')'$ be the Gaussian random vector containing the values of the processes $(Z_t(x))_{t=1,\ldots,s}$ at the points of the design sets $(\mathbf{D}_t)_{t=1,\ldots,s}$. We assume here that the code output is observed without measurement error. The column vector of responses is written $\mathbf{z} = (\mathbf{z}_1', \ldots, \mathbf{z}_s')'$, where $\mathbf{z}_t = (z_t(x_1^{(t)}), \ldots, z_t(x_{n_t}^{(t)}))'$ is the output vector for the level t.

If we consider $Z_s(x)$, the Gaussian process modeling the most accurate code, we want to determine the predictive distribution of $Z_s(x_0)$, $x_0 \in Q$ given $\mathbf{Z} = \mathbf{z}$, *i.e.* the following conditional distribution: $[Z_s(x_0)|\mathbf{Z} = \mathbf{z}]$.

We assume the Markov property introduced by [Kennedy and O'Hagan, 2000]:

$$\mathrm{Cov}(Z_t(x), Z_{t-1}(\tilde{x})|Z_{t-1}(x)) = 0 \qquad \forall x \neq \tilde{x}. \tag{3.1}$$

The property $\mathrm{Cov}(Z_t(x), Z_{t-1}(\tilde{x})|Z_{t-1}(x)) = 0$, $\forall x \neq \tilde{x}$ means that if $Z_{t-1}(x)$ is known, then nothing more can be learn about $Z_t(x)$ from any other run of the cheaper code $Z_{t-1}(\tilde{x})$ for $\tilde{x} \neq x$.

This assumption leads to the following autoregressive model (see proof in Appendix A.1):

$$Z_t(x) = \rho_{t-1}(x) Z_{t-1}(x) + \delta_t(x) \quad t = 2, \ldots, s, \tag{3.2}$$

where $\delta_t(x)$ is a Gaussian process independent of $Z_{t-1}(x), \ldots, Z_1(x)$ and $\rho_{t-1}(x)$ represents a scale factor between $Z_t(x)$ and $Z_{t-1}(x)$. It both represents the correlation degree and the scale factor between two successive levels of code:

$$\rho_{t-1}(x) = \frac{\mathrm{Cov}(Z_t(x), Z_{t-1}(x))}{\mathrm{var}(Z_{t-1}(x))}.$$

We assume that $\rho_{t-1}(x) = \mathbf{g}_{t-1}(x)\boldsymbol{\beta}_{\rho_{t-1}}$, $t = 2, \ldots, s$, where $\mathbf{g}_{t-1}(x) = (f^1_{\rho_{t-1}}(x), \ldots, f^{q_{t-1}}_{\rho_{t-1}}(x))'$ is a vector of q_{t-1} regression functions - generally including the constant function : $x \in Q \to 1$ - and $\boldsymbol{\beta}_{\rho_{t-1}} \in \mathbb{R}^{q_{t-1}}$.
Conditioning on parameters σ_t, $\boldsymbol{\beta}_t$ and $\boldsymbol{\theta}_t$, $\delta_t(x)$ is assumed to be a Gaussian process with mean $\mathbf{f}'_t(x)\boldsymbol{\beta}_t$, where $\mathbf{f}_t(x)$ is a p_t-dimensional vector of regression functions, and with a covariance function of the form $k_t(x, \tilde{x}) = \mathrm{cov}(\delta_t(x), \delta_t(\tilde{x})) = \sigma_t^2 r_t(x - \tilde{x}; \boldsymbol{\theta}_t)$, where σ_t^2 is the variance of the Gaussian process and $\boldsymbol{\theta}_t$ are the hyper parameters of the correlation function r_t. Moreover, conditioning on parameters σ_1, $\boldsymbol{\beta}_1$ and $\boldsymbol{\theta}_1$, the simplest code $Z_1(x)$ is modeled as a Gaussian process with mean $\mathbf{f}'_1(x)\boldsymbol{\beta}_1$ and with covariance function $k_1(x, \tilde{x}) = \sigma_1^2 r_1(x - \tilde{x}; \boldsymbol{\theta}_1)$. With this consistent set of hypotheses, the joint process $(Z_1(x), \ldots, Z_t(x))_{x \in Q, t=1,\ldots,s}$ given $\sigma^2 = (\sigma_i^2)_{i=1,\ldots,t}$, $\boldsymbol{\theta} = (\boldsymbol{\theta}_i)_{i=1,\ldots,t}$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_i)_{i=1,\ldots,t}$ and $\boldsymbol{\beta}_\rho = (\boldsymbol{\beta}_{\rho_{i-1}})_{i=2,\ldots,t}$, is Gaussian with mean:

$$\mathbb{E}[Z_t(x)|\sigma^2, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\beta}_\rho] = \mathbf{h}'_t(x)\boldsymbol{\beta}, \tag{3.3}$$

$$\mathbf{h}'_t(x) = \left(\left(\prod_{i=1}^{t-1} \rho_i(x) \right) \mathbf{f}'_1(x), \left(\prod_{i=2}^{t-1} \rho_i(x) \right) \mathbf{f}'_2(x), \ldots, \rho_{t-1}(x)\mathbf{f}'_{t-1}(x), \mathbf{f}'_t(x) \right) \tag{3.4}$$

and covariance:

$$\mathrm{cov}(Z_t(x), Z_t(\tilde{x})|\sigma^2, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\beta}_\rho) = \sum_{j=1}^{t} \sigma_j^2 \left(\prod_{i=j}^{t-1} \rho_i^2(x) \right) r_j(x - \tilde{x}; \boldsymbol{\theta}_j). \tag{3.5}$$

For each level $t = 2, \ldots, s$, the experimental design $\mathbf{D}_t$ is assumed to be such that $\mathbf{D}_t \subseteq \mathbf{D}_{t-1}$. Note that this assumption is not necessary but allows us to have closed form expressions for the parameter estimate formulas. Furthermore, we denote by $R_t(\mathbf{D}_k, \mathbf{D}_l)$ the correlation matrix between observations at points in $\mathbf{D}_k$ and $\mathbf{D}_l$, $1 \leq k, l \leq s$. $R_t(\mathbf{D}_k, \mathbf{D}_l)$ is a $(n_k \times n_l)$ matrix with (i, j) entry given by:

$$[R_t(\mathbf{D}_k, \mathbf{D}_l)]_{i,j} = r_t(x_i^{(k)} - x_j^{(l)}; \boldsymbol{\theta}_t) \qquad 1 \leq i \leq n_k \quad 1 \leq j \leq n_l.$$

We will use the notation: $R_t(\mathbf{D}_k) = R_t(\mathbf{D}_k, \mathbf{D}_k)$.
[Kennedy and O'Hagan, 2000] present the case where $\forall t \in [2, s]$, $\rho_{t-1}(x) = \rho_{t-1}$ is constant. Here, we will consider the general model presented in equations (3.2). We will also propose a new approach to estimate the coefficients $(\boldsymbol{\beta}_t, \boldsymbol{\beta}_{\rho_{t-1}})_{t=2,\ldots,s}$ based on a Bayesian approach, which allows us to get information about their uncertainties. In the following section, we describe the case of 2 levels of code where the scaling coefficient ρ is constant and then we will extend it for s levels in Section 3.6. The general case in which ρ depends on x is addressed in Appendix A.2.

3.3 Building a model with 2 levels of code

Let us assume that we have 2 levels of code $z_2(x)$ and $z_1(x)$. From the previous section we assume that:

$$\begin{cases} Z_2(x) = \rho Z_1(x) + \delta(x), \quad x \in Q \\ (Z_1(x))_{x\in Q} \perp (\delta(x))_{x\in Q} \end{cases}. \tag{3.6}$$

The goal of this section is to build a surrogate model for $Z_2(x)$ given the observations $\mathbf{Z} = \mathbf{z}$ with an uncertainty quantification. The strategy is the following one. In Subsection 3.3.1 we describe the statistical distribution of the output $Z_2(x_0)$ at a new point x_0 given the parameters $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \rho)$, (σ_1^2, σ_2^2) and $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ and the observations $\mathbf{z}$. In Subsection 3.3.2 we describe the Bayesian estimation of the parameters $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \rho)$ and (σ_1^2, σ_2^2) given the observations. As pointed out at the end of Subsection 3.3.2 the hyper-parameters $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ are estimated using a concentrated restricted log-likelihood method.

3.3.1 Conditional distribution of the output

For a point $x_0 \in Q$ we determine in this subsection the distribution of $[Z_2(x_0)|\mathbf{Z} = z, (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \rho), (\sigma_1^2, \sigma_2^2), (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)]$. Standard results for normal distributions (see Chapter 2) give that:

$$[Z_2(x_0)|\mathbf{Z} = \mathbf{z}, (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \rho), (\sigma_1^2, \sigma_2^2), (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)] \sim \mathcal{N}(m_{Z_2}(x_0), s_{Z_2}^2(x_0)), \tag{3.7}$$

with mean function:

$$m_{Z_2}(x) = \mathbf{h}'(x)\boldsymbol{\beta} + \mathbf{k}'(x)\mathbf{V}^{-1}(\mathbf{z} - \mathbf{H}\boldsymbol{\beta}) \tag{3.8}$$

and variance:

$$s_{Z_2}^2(x) = \rho^2\sigma_1^2 + \sigma_2^2 - \mathbf{k}'(x)\mathbf{V}^{-1}\mathbf{k}(x), \tag{3.9}$$

where we have denoted $\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \end{pmatrix}$, $\mathbf{z} = \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{pmatrix}$ and where $\mathbf{H}$ is defined by:

$$\mathbf{H} = \begin{pmatrix} \mathbf{f}_1'(x_1^{(1)}) & 0 \\ \vdots & \vdots \\ \mathbf{f}_1'(x_{n_1}^{(1)}) & 0 \\ \rho\mathbf{f}_1'(x_1^{(2)}) & \mathbf{f}_2'(x_1^{(2)}) \\ \vdots & \vdots \\ \rho\mathbf{f}_1'(x_{n_2}^{(2)}) & \mathbf{f}_2'(x_{n_2}^{(2)}) \end{pmatrix} = \left(\begin{array}{c|c} F_1(\mathbf{D}_1) & 0 \\ \hline \rho F_1(\mathbf{D}_2) & F_2(\mathbf{D}_2) \end{array}\right),$$

with the notation $F_i(\mathbf{D}_j) = \begin{pmatrix} \mathbf{f}_i'(x_{n_1}^{(j)}) \\ \vdots \\ \mathbf{f}_i'(x_{n_j}^{(j)}) \end{pmatrix}$. Furthermore, we have $\mathbf{h}'(x) = (\rho\mathbf{f}_1'(x), \mathbf{f}_2'(x))$ and:

$$\begin{aligned} \mathbf{k}'(x) &= \mathrm{Cov}(Z_2(x), \mathbf{Z}) \\ &= \left(\rho\sigma_1^2 R_1(\{x\}, \mathbf{D}_1), \rho^2\sigma_1^2 R_1(\{x\}, \mathbf{D}_2) + \sigma_2^2 R_2(\{x\}, \mathbf{D}_2)\right) \end{aligned}. \tag{3.10}$$

The covariance matrix $\mathbf{V}$ of the Gaussian vector $\mathbf{Z} = \begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix}$ can be written :

$$\mathbf{V} = \begin{pmatrix} \sigma_1^2 R_1(\mathbf{D}_1) & \rho\sigma_1^2 R_1(\mathbf{D}_1, \mathbf{D}_2) \\ \rho\sigma_1^2 R_1(\mathbf{D}_2, \mathbf{D}_1) & \rho^2\sigma_1^2 R_1(\mathbf{D}_2) + \sigma_2^2 R_2(\mathbf{D}_2) \end{pmatrix}. \tag{3.11}$$

3.3.2 Bayesian estimation of the parameters with 2 levels of code

In this subsection, we describe the Bayesian estimation of the parameters $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \rho, \sigma_1^2, \sigma_2^2, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ for the 2-level model given the observations $\mathbf{Z} = \mathbf{z}$. In particular, we look for the posterior distribution of $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \rho, \sigma_1^2, \sigma_2^2, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ given the observations $\mathbf{Z} = \mathbf{z}$ in the case in which the prior distribution of $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \rho, \sigma_1^2, \sigma_2^2, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ has a special (conjugate) form or a non-informative form. Due to the conditional independence between $Z_1(x)$ and $\delta(x)$, it is possible to estimate separately the parameters $(\boldsymbol{\beta}_1, \sigma_1^2, \boldsymbol{\theta}_1)$ and $(\boldsymbol{\beta}_2, \rho, \sigma_2^2, \boldsymbol{\theta}_2)$. We first describe the posterior distribution of $(\boldsymbol{\beta}_1, \sigma_1^2)$ given $\boldsymbol{\theta}_1$ and $(\boldsymbol{\beta}_2, \sigma_2^2, \rho)$ given $\boldsymbol{\theta}_2$, which can be obtained in closed forms. We then describe how to estimate $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$.

Firstly, we consider the parameters $(\boldsymbol{\beta}_1, \sigma_1^2, \boldsymbol{\theta}_1)$. We choose the following non-informative prior distributions corresponding to the "Jeffreys priors" [Jeffreys, 1961]:

$$p(\boldsymbol{\beta}_1|\sigma_1^2, \boldsymbol{\theta}_1) \propto 1 \qquad p(\sigma_1^2, \boldsymbol{\theta}_1) \propto \frac{1}{\sigma_1^2}. \tag{3.12}$$

Considering the probability density function of $[Z_1|\boldsymbol{\beta}_1, \sigma_1^2, \boldsymbol{\theta}_1]$ and the Bayes formula, the posterior distribution of $[\boldsymbol{\beta}_1|\mathbf{z}_1, \sigma_1^2, \boldsymbol{\theta}_1]$ is :

$$[\boldsymbol{\beta}_1|\mathbf{z}_1, \sigma_1^2, \boldsymbol{\theta}_1] \sim \mathcal{N}_{p_1}\left([\mathbf{F}_1' R_1(\mathbf{D}_1)^{-1}\mathbf{F}_1]^{-1}[\mathbf{F}_1' R_1(\mathbf{D}_1)^{-1}\mathbf{z}_1], [\mathbf{F}_1' \frac{R_1(\mathbf{D}_1)^{-1}}{\sigma_1^2}\mathbf{F}_1]^{-1}\right), \tag{3.13}$$

where $\mathbf{F}_1 := F_1(\mathbf{D}_1)$. Then, using the Bayes formula, we obtain that the posterior distribution of $[\sigma_1^2|\mathbf{z}_1, \boldsymbol{\theta}_1]$ is:

$$[\sigma_1^2|\mathbf{z}_1, \boldsymbol{\theta}_1] \sim \mathcal{IG}(\alpha_{\sigma_1^2|n_1}, \frac{Q_1}{2}), \tag{3.14}$$

where $\mathcal{IG}(\alpha, Q)$ stands for the inverse gamma and the parameters are given by:

$$\alpha_{\sigma_1^2|n_1} = \frac{n_1 - p_1}{2} \qquad Q_1 \quad = (\mathbf{z}_1 - \mathbf{F}_1\tilde{\boldsymbol{\beta}}_1)' R_1(\mathbf{D}_1)^{-1}(\mathbf{z}_1 - \mathbf{F}_1\tilde{\boldsymbol{\beta}}_1) \ , \tag{3.15}$$

with $\tilde{\boldsymbol{\beta}}_1 = \mathbb{E}\left[\boldsymbol{\beta}_1|\mathbf{z}_1, \sigma_1^2, \boldsymbol{\theta}_1\right] = [\mathbf{F}_1' R_1(\mathbf{D}_1)^{-1}\mathbf{F}_1]^{-1}[\mathbf{F}_1' R_1(\mathbf{D}_1)^{-1}\mathbf{z}_1]$.

The posterior mean $\tilde{\boldsymbol{\beta}}_1$ of $\boldsymbol{\beta}_1$ with non-informative "Jeffreys priors" [Jeffreys, 1961] equals the maximum likelihood estimate of $\boldsymbol{\beta}_1$. For the parameter σ_1^2, the estimate given by the posterior harmonic average $\hat{\sigma}_1^2 = \frac{Q_1}{2\alpha_{\sigma_1^2|n_1}}$ is identical to the one obtained with the restricted maximum likelihood method. This method was introduced by Patterson and Thompson [Patterson and Thompson, 1971] in order to reduce the bias of the maximum likelihood estimator.

Secondly, let us consider the set of parameters $(\boldsymbol{\beta}_2, \rho, \sigma_2^2, \boldsymbol{\theta}_2)$. In order to have closed form formulas for the posterior distribution of $(\boldsymbol{\beta}_2, \rho)$, we estimate them together. The idea to carry out a joint Bayesian analysis is proposed for the first time in this chapter and we believe it is

important. Indeed, if the cheaper code is perfectly known, it can be considered as a regression function and so ρ will be a regression parameter. In this case, it is clear that a separated estimation of $\boldsymbol{\beta}_2$ and ρ cannot be optimal.
Using the Jeffrey prior distributions $p((\rho, \boldsymbol{\beta}_2)|\sigma_2^2, \boldsymbol{\theta}_2) \propto 1$ and $p(\sigma_2^2, \boldsymbol{\theta}_2) \propto \frac{1}{\sigma_2^2}$ and the same methodology as for the posterior distribution of $(\boldsymbol{\beta}_1, \sigma_1^2)$, we find that:

$$[(\rho, \boldsymbol{\beta}_2)|\mathbf{z}_1, \mathbf{z}_2, \sigma_2^2, \boldsymbol{\theta}_2] \sim \mathcal{N}_{p_2+1}\left([\mathbf{F}'R_2(\mathbf{D}_2)^{-1}\mathbf{F}]^{-1}[\mathbf{F}'R_2(\mathbf{D}_2)^{-1}\mathbf{F}], [\mathbf{F}'\frac{R_2(\mathbf{D}_2)^{-1}}{\sigma_2^2}\mathbf{F}]^{-1}\right) \tag{3.16}$$

and:

$$[\sigma_2^2|\mathbf{z}_2, \mathbf{z}_1, \boldsymbol{\theta}_2] \sim \mathcal{IG}(\alpha_{\sigma_2^2|n_2}, \frac{Q_2}{2}), \tag{3.17}$$

where:

$$\alpha_{\sigma_2^2|n_2} = \frac{n_2 - p_2 - 1}{2} \qquad Q_2 \ = (\mathbf{z}_2 - \mathbf{F}\tilde{\boldsymbol{\lambda}})'R_2(\mathbf{D}_2)^{-1}(\mathbf{z}_2 - \mathbf{F}\tilde{\boldsymbol{\lambda}}) \ , \tag{3.18}$$

with $\tilde{\boldsymbol{\lambda}} = \mathbb{E}[(\rho, \boldsymbol{\beta}_2)|\mathbf{z}_1, \mathbf{z}_2, \sigma_2^2, \boldsymbol{\theta}_2] = [\mathbf{F}'R_2(\mathbf{D}_2)^{-1}\mathbf{F}]^{-1}[\mathbf{F}'R_2(\mathbf{D}_2)^{-1}\mathbf{z}_2]$. The design matrix $\mathbf{F}$ is such that $\mathbf{F} = [z_1(\mathbf{D}_2) \quad \mathbf{F}_2]$. Furthermore, the estimate of σ_2^2 given by the posterior harmonic average $\hat{\sigma}_2^2 = \frac{Q_2}{2\alpha_{\sigma_2^2|n_2}}$ is the same as the restricted maximum likelihood one.

The hyper-parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are found by minimizing the negative concentrated restricted log-likelihoods:

$$\log\left(|\det\left(R_1(\mathbf{D}_1)\right)|\right) + (n_1 - p_1)\log(\hat{\sigma_1}^2), \tag{3.19}$$

$$\log\left(|\det\left(R_2(\mathbf{D}_2)\right)|\right) + (n_2 - p_2 - 1)\log(\hat{\sigma_2}^2). \tag{3.20}$$

These minimizations problems must be numerically solved with a global optimization method. We use an evolutionary method coupled with a BFGS (Broyden-Fletcher-Goldfarb-Shanno) algorithm [Avriel, 2003]. The drawback of the maximum likelihood estimation is that, contrarily to Bayesian estimation, we do not have any information about the variance of the estimator in non-asymptotic cases (see [Lehmann and Casella, 1998]). Nevertheless, Bayesian estimation of the hyper parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are prohibitive and as noted in [Santner et al., 2003] the choice of the prior distribution is non trivial. Therefore, in this chapter, we will always estimate these parameters with a concentrated restricted likelihood method.

3.4 Bayesian prediction for a code with 2 levels

The aim of a Bayesian prediction is to provide a predictive distribution for $Z_s(x)$ integrating the posterior distributions of the parameters and hence taking into account their uncertainty. The forthcoming developments are the extension of the Bayesian kriging presented in Section 1.2.2 to the multi-fidelity co-kriging model.

A Bayesian prediction for a code with $s = 2$ levels was suggested by [Qian and Wu, 2008]. Nevertheless, we propose here a new Bayesian approach with some significant differences. First, we assume that the adjustment coefficient is a regression function whereas [Qian and Wu, 2008] model it with a Gaussian process. Secondly, we use different prior distributions for the parameter estimation. More specifically, according to the Bayesian estimation of parameters

previously presented, we use a joint prior distribution for $(\boldsymbol{\beta}_2, \rho)$ conditioned by σ_2^2 whereas [Qian and Wu, 2008] use separated prior distributions with ρ not conditioned by σ_2^2. Then, we use a hierarchy between the different parameters. At the lowest level is the regression parameter $\boldsymbol{\beta}$. At the second level is the variance parameter σ^2 which controls the distribution of the parameter $\boldsymbol{\beta}$. At the top level is the parameter $\boldsymbol{\theta}$ which controls the distribution of the parameters at the bottom levels. It is common to use a hierarchical specification of models for Bayesian prediction as presented in [Rasmussen and Williams, 2006]. This strategy will allow us to obtain explicit formulas for the joint distribution of the parameters and above all, to reduce dramatically the cost of the numerical implementation of the complete Bayesian prediction.

We will also present the case in which we do not have any prior information about the parameters. As described in the previous section, the hyper parameter $\boldsymbol{\theta}$ is estimated by minimizing the negative concentrated restricted log-likelihood and it is assumed to be fixed to this estimated value from now on.

3.4.1 Prior distributions and Bayesian estimation of the parameters

Many choices of priors can be made for the Bayesian modeling. Here we study the two following cases:

(I) Priors for each parameter are informative.

(II) Priors for each parameter are non-informative.

For the non-informative case (II), we use the improper distributions corresponding to the "Jeffreys priors" and then the posterior distributions are given in Section 3.3.2. Note that non-informative distributions are used when we do not have prior knowledge. For the informative case (I), we will consider the following prior distributions:

$$[\boldsymbol{\beta}_1|\sigma_1^2] \sim \mathcal{N}_{p_1}(\mathbf{b}_1, \sigma_1^2\mathbf{V}_1), \qquad [(\rho, \boldsymbol{\beta}_2)|\mathbf{z}_1, \sigma_2^2] \sim \mathcal{N}_{1+p_2}\left(\mathbf{b}_\lambda = \begin{pmatrix}\mathbf{b}_\rho \\ \mathbf{b}_2\end{pmatrix}, \sigma_2^2\mathbf{V}_\lambda = \sigma_2^2\begin{pmatrix} v_\rho & 0 \\ 0 & \mathbf{V}_2\end{pmatrix}\right),$$

$$[\sigma_1^2] \sim \mathcal{IG}(\alpha_1, \gamma_1), \qquad [\sigma_2^2|\mathbf{z}_1] \sim \mathcal{IG}(\alpha_2, \gamma_2)$$

where $\mathbf{b}_1 \in \mathbb{R}^{p_1}$, $\mathbf{b}_\lambda \in \mathbb{R}^{1+p_2}$, $\mathbf{V}_1$ is a $(p_1 \times p_1)$ diagonal matrix, $\mathbf{V}_\lambda$ is a $((1+p_2)\times(1+p_2))$ diagonal matrix, v_ρ is a positive scalar and $\alpha_1, \gamma_1, \alpha_2, \gamma_2 > 0$. The forms of the priors are chosen in order to be able to get closed form expressions for the posterior distributions. Note that there are enough free parameters in the prior distributions to allow the user to prescribe their means and variances. From the previous prior definitions, the posterior distributions of the parameters are:

$$[\boldsymbol{\beta}_1|\mathbf{z}_1, \sigma_1^2] \sim \mathcal{N}_{p_1}(\mathbf{A}_i^1\boldsymbol{\nu}_i^1, \mathbf{A}_i^1) \qquad [(\rho, \boldsymbol{\beta}_2)|\mathbf{z}_1, \mathbf{z}_2, \sigma_2^2] \sim \mathcal{N}_{p_2+1}(\mathbf{A}_i^\lambda\boldsymbol{\nu}_i^\lambda, \mathbf{A}_i^\lambda), \tag{3.21}$$

where:

$$\mathbf{A}_i^1 = \begin{cases} \sigma_1^2[\mathbf{F}_1' R_1(\mathbf{D}_1)^{-1}\mathbf{F}_1 + \mathbf{V}_1^{-1}]^{-1} & i = (\mathrm{I}) \\ \sigma_1^2[\mathbf{F}_1' R_1(\mathbf{D}_1)^{-1}\mathbf{F}_1]^{-1} & i = (\mathrm{II}) \end{cases},$$

$$\boldsymbol{\nu}_i^1 = \begin{cases} [\mathbf{F}_1' R_1(\mathbf{D}_1)^{-1}\mathbf{z}_1 + \mathbf{V}_1^{-1}\mathbf{b}_1]/\sigma_1^2 & i = (\mathrm{I}) \\ [\mathbf{F}_1' R_1(\mathbf{D}_1)^{-1}\mathbf{z}_1]/\sigma_1^2 & i = (\mathrm{II}) \end{cases},$$

$$\mathbf{A}_i^\lambda = \begin{cases} \sigma_2^2[\mathbf{F}'R_2(\mathbf{D}_2)^{-1}\mathbf{F} + \mathbf{V}_\lambda^{-1}]^{-1} & i = (\mathrm{I}) \\ \sigma_2^2[\mathbf{F}'R_2(\mathbf{D}_2)^{-1}\mathbf{F}]^{-1} & i = (\mathrm{II}) \end{cases},$$

$$\boldsymbol{\nu}_i^\lambda = \begin{cases} [\mathbf{F}'R_2(\mathbf{D}_2)^{-1}\mathbf{z}_2 + \mathbf{V}_\lambda^{-1}\mathbf{b}_\lambda]/\sigma_2^2 & i = (\mathrm{I}) \\ [\mathbf{F}'R_2(\mathbf{D}_2)^{-1}\mathbf{z}_2]/\sigma_2^2 & i = (\mathrm{II}) \end{cases}$$

and $\mathbf{F} = [z_1(\mathbf{D}_2) \quad \mathbf{F}_2]$. Furthermore, we have:

$$[\sigma_1^2|\mathbf{z}_1] \sim \mathcal{IG}(\alpha_i^{\sigma_1^2|n_1}, \frac{Q_i^1}{2}), \qquad [\sigma_2^2|\mathbf{z}_2, \mathbf{z}_1] \sim \mathcal{IG}(\alpha_i^{\sigma_2^2|n_2}, \frac{Q_i^2}{2}), \tag{3.22}$$

where:

$$Q_i^1 = \begin{cases} 2\gamma_1 + (\mathbf{b}_1 - \tilde{\boldsymbol{\beta}}_1)'(\mathbf{V}_1 + [\mathbf{F}_1'R_1(\mathbf{D}_1)^{-1}\mathbf{F}_1]^{-1})^{-1}(\mathbf{b}_1 - \tilde{\boldsymbol{\beta}}_1) + Q_2^1 & i = (\mathrm{I}) \\ \mathbf{z}_1'[R_1(\mathbf{D}_1)^{-1} - R_1(\mathbf{D}_1)^{-1}\mathbf{F}_1(\mathbf{F}_1'R_1(\mathbf{D}_1)^{-1}\mathbf{F}_1)^{-1}\mathbf{F}_1'R_1(\mathbf{D}_1)^{-1}]\mathbf{z}_1 & i = (\mathrm{II}) \end{cases},$$

$$Q_i^2 = \begin{cases} 2\gamma_2 + (\mathbf{b}_\lambda - \tilde{\lambda})'(\mathbf{V}_\lambda + [\mathbf{F}'R_2(\mathbf{D}_2)^{-1}\mathbf{F}]^{-1})^{-1}(\mathbf{b}_\lambda - \tilde{\lambda}) + Q_2^2 & i = (\mathrm{I}) \\ \mathbf{z}_2'[R_2(\mathbf{D}_2)^{-1} - R_2(\mathbf{D}_2)^{-1}\mathbf{F}(\mathbf{F}'R_2(\mathbf{D}_2)^{-1}\mathbf{F})^{-1}\mathbf{F}'R_2(\mathbf{D}_2)^{-1}]\mathbf{z}_2 & i = (\mathrm{II}) \end{cases},$$

$$\tilde{\boldsymbol{\beta}}_1 = (\mathbf{F}_1'R_1(\mathbf{D}_1)^{-1}\mathbf{F}_1)^{-1}\mathbf{F}_1'R_1(\mathbf{D}_1)^{-1}\mathbf{z}_1, \qquad \tilde{\lambda} = (\mathbf{F}'R_2(\mathbf{D}_2)^{-1}\mathbf{F})^{-1}\mathbf{F}'R_2(\mathbf{D}_2)^{-1}\mathbf{z}_2,$$

$$\alpha_i^{\sigma_1^2|n_1} = \begin{cases} \frac{n_1}{2} + \alpha_1 & i = (\mathrm{I}) \\ \frac{n_1 - p_1}{2} & i = (\mathrm{II}) \end{cases}, \qquad \alpha_i^{\sigma_2^2|n_2} = \begin{cases} \frac{n_2}{2} + \alpha_2 & i = (\mathrm{I}) \\ \frac{n_2 - p_2 - 1}{2} & i = (\mathrm{II}) \end{cases}.$$

Mixing of informative and non-informative priors are of course possible and easy to implement. As we will discuss in Subsection 3.4.4 and see in the examples of Section 3.5, the use of informative priors has minor impact on the mean estimation but may have a strong impact on variance estimation.

3.4.2 Predictive distributions when $\boldsymbol{\beta}_2, \rho, \sigma_1^2$ and σ_2^2 are known

As a preliminary step towards the Bayesian prediction carried out in the next subsection, we give here Bayesian prediction in the form of closed form expressions when the parameters $\boldsymbol{\beta}_2$, ρ, σ_1^2 and σ_2^2 are known. The conditional distribution of $[Z_2(x)|Z = z, \boldsymbol{\beta}_2, \rho, \sigma_1^2, \sigma_2^2]$ is given by:

$$[Z_2(x)|\mathbf{Z} = \mathbf{z}, \boldsymbol{\beta}_2, \rho, \sigma_1^2, \sigma_2^2] \sim \mathcal{N}\left(\mu_i(x), \sigma_i^2(x)\right), \tag{3.23}$$

where:

$$\mu_i(x) = \mathbf{h}'(x)\begin{pmatrix} \mathbf{A}_i^1\boldsymbol{\nu}_i^1 \\ \boldsymbol{\beta}_2 \end{pmatrix} + \mathbf{k}'(x)\mathbf{V}^{-1}\left(\mathbf{z} - \mathbf{H}\begin{pmatrix} \mathbf{A}_i^1\boldsymbol{\nu}_i^1 \\ \boldsymbol{\beta}_2 \end{pmatrix}\right),$$

$$\sigma_i^2(x) = s_{Z_2}^2(x) + \mathbf{g}_1\mathbf{A}_i^1\mathbf{g}_1'$$

and $\mathbf{A}_i^1$ and $\boldsymbol{\nu}_i^1$ are defined by (3.21). Note that the estimated variance is augmented by the term $\mathbf{g}_1\mathbf{A}_i^1\mathbf{g}_1'$ which quantifies the uncertainty due to the estimation of $\boldsymbol{\beta}_1$. $\mathbf{g}_1$ is a $(1 \times p_1)$ vector composed of the p_1 first elements of the $(1 \times p_1, 1 \times p_2)$ vector $\mathbf{g} = (\mathbf{g}_1, \mathbf{g}_2) = \mathbf{h}'(x) - \mathbf{k}'(x)\mathbf{V}^{-1}\mathbf{H}$. $\mathbf{H}$ is given by (3.3.1). The existence of closed form formulas is important as it will allow for a fast numerical implementation.

3.4.3 Bayesian prediction

Before performing the Bayesian prediction we note that - thanks to the explicit joint prior distribution for $\boldsymbol{\beta}_2$ and ρ, the independence hypotheses and the hierarchical specification of the parameters - conditioning on $\boldsymbol{\theta}$, we have an explicit formula for the following joint density (see Section 3.4.1):

$$p(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \rho, \sigma_1^2, \sigma_2^2 | \mathbf{z}_1, \mathbf{z}_2) = p(\boldsymbol{\beta}_1 | \sigma_1^2, \mathbf{z}_1) p(\boldsymbol{\beta}_2, \rho | \sigma_2^2, \mathbf{z}_1, \mathbf{z}_2) p(\sigma_1^2 | \mathbf{z}_1) p(\sigma_2^2 | \mathbf{z}_1, \mathbf{z}_2). \tag{3.24}$$

This explicit joint density is an original result which contrasts with [Qian and Wu, 2008] and which allows us to avoid prohibitive implementation for the Bayesian analysis.

First, we consider the predictive distribution with σ_1^2 and σ_2^2 known. Considering the conditional independence assumption between $(\delta(x))_{x \in Q}$ and $(Z_1(x))_{x \in Q}$, the probability density function of $[Z_2(x) | \mathbf{Z} = \mathbf{z}, \sigma_1^2, \sigma_2^2]$ can be deduced from the following integral:

$$p(z_2(x) | \mathbf{z}_1, \mathbf{z}_2, \sigma_1^2, \sigma_2^2) = \int_{\mathbb{R}^{1+p_2}} p(z_2(x) | \mathbf{z}_1, \mathbf{z}_2, \boldsymbol{\beta}_2, \rho, \sigma_1^2, \sigma_2^2) p(\rho, \boldsymbol{\beta}_2 | \mathbf{z}_1, \mathbf{z}_2, \sigma_2^2) \, d\rho d\boldsymbol{\beta}_2, \tag{3.25}$$

where $p(z_2(x) | \mathbf{z}_1, \mathbf{z}_2, \boldsymbol{\beta}_2, \rho, \sigma_1^2, \sigma_2^2)$ is given by (3.23). This integral has to be numerically evaluated. Since $[\rho, \boldsymbol{\beta}_2 | \mathbf{z}_1, \mathbf{z}_2, \sigma_2^2]$ has a known normal distribution given by (3.21), we here use a Monte-Carlo algorithm when the dimension of $\boldsymbol{\beta}_2$ and ρ is high, or a trapezoidal quadrature method when it is low.

Then, we infer from the parameters σ_1^2 and σ_2^2. Due to the independence between $(\delta(x))_{x \in Q}$ and $(Z_1(x))_{x \in Q}$, the probability density function of $[Z_2(x) | \mathbf{Z} = \mathbf{z}]$ is:

$$p(z_2(x) | \mathbf{z}_1, \mathbf{z}_2) = \int_{\mathbb{R}^2} p(z_2(x) | \mathbf{z}_1, \mathbf{z}_2, \sigma_1^2, \sigma_2^2) p(\sigma_1^2 | \mathbf{z}_1) p(\sigma_2^2 | \mathbf{z}_1, \mathbf{z}_2) \, d\sigma_1^2 d\sigma_2^2, \tag{3.26}$$

where $p(\sigma_1^2 | \mathbf{z}_1)$ and $p(\sigma_2^2 | \mathbf{z}_1, \mathbf{z}_2)$ are given by (3.22). This integral has also to be numerically evaluated. Since we have a double integration, a quadrature method will be efficient. We use here a trapezoidal numerical integration, defining the region of integration $[\sigma_{1_{inf}}^2, \sigma_{1_{sup}}^2] \times [\sigma_{2_{inf}}^2, \sigma_{2_{sup}}^2]$ from Equation (3.22) and such that $p(\sigma_{1_{inf}}^2 | \mathbf{z}_1)$, $p(\sigma_{1_{sup}}^2 | \mathbf{z}_1)$ $p(\sigma_{2_{inf}}^2 | \mathbf{z}_1, \mathbf{z}_2)$ and $p(\sigma_{2_{sup}}^2 | \mathbf{z}_1, \mathbf{z}_2)$ are close to 0. This region essentially contains the support of the function. Furthermore, we create a non-uniform integration grid distributed with a geometric progression.

Finally $p(z_2(x) | \mathbf{z}_1, \mathbf{z}_2)$ is a predictive density function integrating the posterior distribution of parameters $(\boldsymbol{\beta}_2, \rho, \boldsymbol{\beta}_1, \sigma_1^2, \sigma_2^2)$. We hence have a predictive distribution taking into account the uncertainties due to the parameter estimations.

3.4.4 Discussion about the numerical evaluations of the integrals

We saw in the previous section that we can obtain an analytical prediction when $\boldsymbol{\beta}_2$, ρ, σ_1^2 and σ_2^2 are known. From this analytical formula, we can have a Bayesian prediction with only two nested integrations. One of them can be approximated with a quadrature or a Monte Carlo method, which is not too expensive. The other is a double integration approximated with a quadrature method which is efficient and not expensive. Therefore, we do not use any Markov

chain Monte Carlo method and we considerably reduce the time and the complexity of the method. This allows us to easily build an accurate Bayesian metamodel. Practically, we use 441 integration points to approximate (3.26) and 1000 Monte-Carlo particles to approximate (3.25). Therefore, we have 441000 call to the predictive density function (3.23).

To avoid a prohibitive implementation, another approach has also been proposed in [Cumming and Goldstein, 2009]. They adopt a Bayes linear formulation which requires only the specification of the means, variances, and covariances. See [Goldstein and Wooff, 2007] for further details about the Bayes linear approach. The strength of this method is that its computational cost is low. Nonetheless, since it only focuses on posterior means and covariances, it does not provide the full posterior predictive distribution.

Finally, we highlight the fact that our Bayesian procedure can be used to perform multi-fidelity analysis with more than 2 levels of code whereas the cost of the one presented by [Qian and Wu, 2008] is too high to allow for such analysis. We illustrate in Section 3.7 through an industrial case the great practical importance of using more than 2 levels of code.

3.5 Academic examples

We will present in this section some co-kriging metamodels using one-dimensional functions inspired by the example presented by [Forrester et al., 2007]. For the following examples, we will use a non-Bayesian co-kriging model - *i.e.* the one presented by [Kennedy and O'Hagan, 2000] - but with a Bayesian estimation of the parameters (see Section 3.3.2) and for the second example we will use a Bayesian co-kriging.

Furthermore, the correlation kernels are assumed to be:

$$r_t(x_i^{(k)} - x_j^{(l)}; \theta_t) = \exp\left(-\frac{(x_i^{(k)} - x_j^{(l)})^2}{2\theta_t^2}\right),$$

where $t, k, l = 1, 2 \quad 1 \leq i \leq n_1 \quad 1 \leq j \leq n_2$ and the regression functions are $\mathbf{f}_1(x) = 1$ and $\mathbf{f}_2'(x) = (1 \quad x)$.

Example 1. The aim of this example is to emphasize the effectiveness of the presented Bayesian estimation of the parameters (see Section 3.6.1). We assume that the cheap code is given by $z_1(x) = 0.5(6x-2)^2\sin(12x-4) + 10(x-0.5) - 5$ and the expensive code by $z_2(x) = 2z_1(x) - 20x + 20$. The experimental design set of the cheapest code is $D_1 = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$ and the one of the expensive code is $D_2 = \{0, 0.4, 0.6, 1\}$. This example is identical to the one-dimensional demonstration presented in [Forrester et al., 2007]. Figure 3.1 shows the functions $x \mapsto z_2(x)$ and $x \mapsto z_1(x)$, the training data for z_2 and z_1, the ordinary kriging using only the expensive data and the co-kriging using expensive and cheap data.

To validate the model, the Root Mean Squared Error RMSE $= \sum_{x \in T} \left(m_{Z_2}(x) - z_2(x)\right)^2 / n_T$ and the Nash-Sutcliffe model efficiency coefficient (see [Nash and Sutcliffe, 1970]) *Eff* $= 1 - \frac{\sum_{x \in T}\left(m_{Z_2}(x) - z_2(x)\right)^2}{\sum_{x \in T}\left(m_{Z_2}(x) - \bar{z}_2\right)^2}$, $\bar{z}_2 = \sum_{x \in D_2} z_2(x)/n_2$ are computed. The Nash-Sutcliffe efficiency compares the residual variance with the total variance. It is also referenced as Q_2 coefficient. The closer *Eff* is to 1, the more accurate the model is.

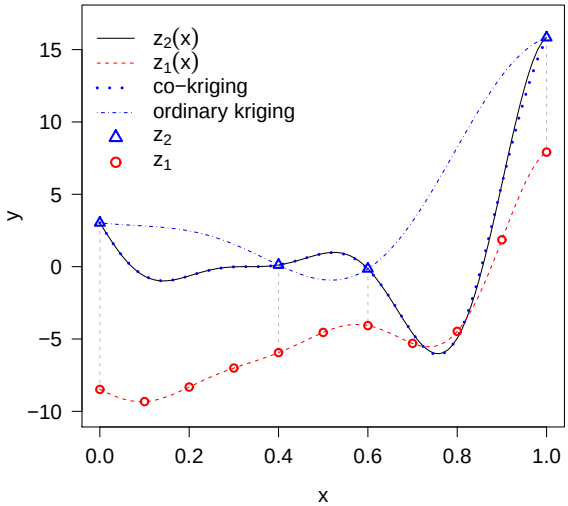


Figure 3.1: Example 1. The co-kriging metamodel is very close to the expensive output $z_2(.)$ and improves significantly the ordinary kriging metamodel using the small design $\mathbf{D}_2$.

The test set T is composed of a regular grid points sampled from 0 to 1 with a grid step equal to 0.01 and $\bar{z}_2$ is the empirical mean evaluated in T. The estimated RMSE is 5.68×10^{-2} and the efficiency *Eff* is 99.98%, so we have a prediction error close to 0. The Bayesian estimates of the parameters of co-kriging are given in Table 3.1. Furthermore, the estimates of the hyper-parameters (θ_1, θ_2), calculated by maximizing the concentrated log-likelihoods (3.19) and (3.20), are $\hat{\theta}_1 = 0.25$ and $\hat{\theta}_2 = 0.80$. $\mathbf{D}_1$ being a regular grid with a grid step equal to 0.1 and $\mathbf{D}_2$ being composed of points sampled from 0 to 1, points of the experimental designs are hence strongly correlated which will imply a smooth surrogate model.

Regression Coefficient	Posterior mean
ρ	2
$\boldsymbol{\beta}_2$	$(20, -20)$
$\boldsymbol{\beta}_1$	-3.49
Variance Coefficient	**Posterior harmonic average**
σ_1^2	32.75
σ_2^2	7.02×10^{-30}

Table 3.1: A co-kriging example with one-variable functions. Bayesian estimation of parameters.

We see that the Bayesian estimation of parameters is very effective since the estimations

of parameters ρ and $\boldsymbol{\beta}_2$ are perfect. Nevertheless this example does not highlight the strength of the method since there is a relation between $z_2(x)_{x\in[0,1]}$ and $z_1(x)_{x\in[0,1]}$ which exactly corresponds to Equation (3.2) with the error δ_2 that can be written in terms of the regression functions $\mathbf{f}_2$ exactly. Therefore, if the cheap code is well modeled, like in this case, the co-kriging is equivalent to a linear regression. Moreover, the very small value of σ_2^2 illustrates this.

Example 2. This example illustrates a case where the non-Bayesian co-kriging underestimates the predictive variance whereas the Bayesian one adjusts it. We assume that the expensive code is given by $z_2(x) = 2z_1(x) - 20x + 20 + \sin(10\cos(5x))$ and the cheaper code is given by $z_1(x) = 0.5((6x-2)^2\sin(12x-4)) + 10(x-0.5) - 5$. Through the term $\sin(10\cos(5x))$, the expensive code has high frequencies which are not captured by the cheap code and the error δ_2 is not a simple linear combination of the regression functions $\mathbf{f}_2$. Therefore, the functions do not exactly match the model presented in Section 3.2 and the high frequency discrepancy makes the problem more challenging. Figure 3.2 shows the results of kriging and co-kriging for these two functions. The estimated RMSE is 1.05 and the efficiency *Eff* is 93.57%, we

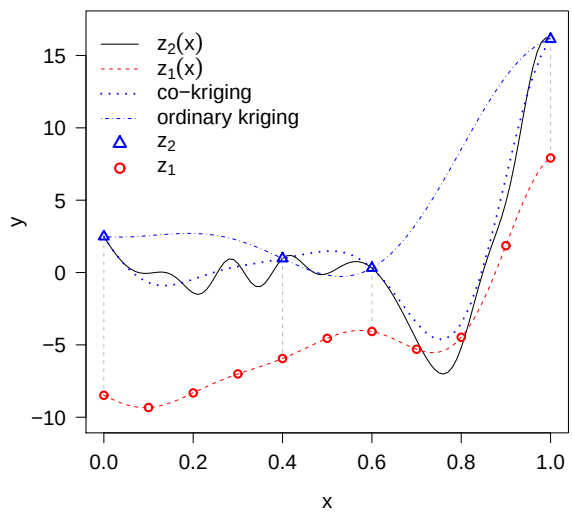


Figure 3.2: Example 2. The high frequency components of the expensive code are not predicted since they are not captured by the cheap code and the coarse grid used for the expensive code cannot detect them either. Nevertheless, the co-kriging improves the ordinary kriging metamodel since the cheap code allows us to predict the low frequencies of the expensive code accurately.

still have a good prediction. The Bayesian estimation of the parameters are given in Table 3.2 and we have $\hat{\theta}_1 = 0.25$ and $\hat{\theta}_2 = 0.07$. The values of θ_1 and θ_2 have been fixed according the following arguments. As the cheap code is the same as the one of the Example 1, we keep the same estimate for θ_1. Then, we consider that there are not enough points to carry out a

significant estimate of θ_2. Therefore, we fix the value of $\hat{\theta}_2$ according to the high frequencies introduced by the term $\sin(10\cos(5x))$.

Regression Coefficient	Posterior mean
ρ	1.86
$\boldsymbol{\beta}_2$	$(18.39, -17.00)$
$\boldsymbol{\beta}_1$	-3.49
Variance Coefficient	**Posterior harmonic average**
σ_1^2	32.75.03
σ_2^2	0.30

Table 3.2: A co-kriging example with one-dimensional functions. Bayesian estimation of parameters.

Due to the additional term $\sin(10\cos(5x))$, the estimate of the parameter ρ is less effective than in the first example. This highlights the dependence between ρ and the mean of $\delta(x)_{x\in[0,1]}$. Furthermore, Figure 3.3 represents the confidence interval at plus or minus twice

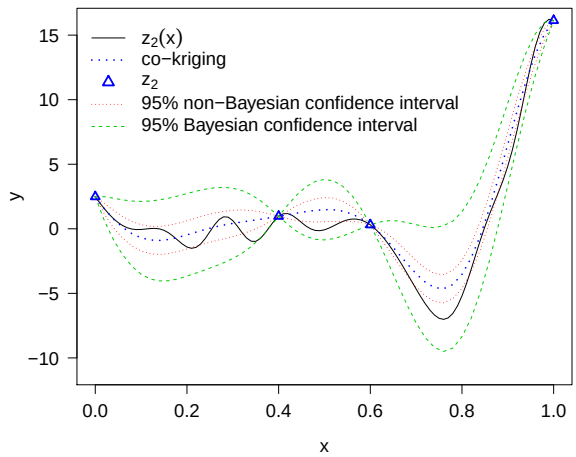


Figure 3.3: Example 2 without any prior information. The thick dotted line represents the prediction mean, the thin dotted lines represent the confidence interval at plus or minus twice the standard deviation in the non-Bayesian case and the dashed lines represent the same confidence interval in the Bayesian case.

the standard deviation of the predictive distribution in the Bayesian and non-Bayesian cases. We see that we underestimate the variance of the predictive distribution in the non-Bayesian case. Its estimate is well adjusted in the Bayesian case.

We finally consider the case in which we have prior information:

$$[(\rho, \beta_2)|z_1, \sigma_2^2] \sim \mathcal{N}\left(\begin{pmatrix} 2 \\ 20 \\ -20 \end{pmatrix}, \sigma_2^2 \begin{pmatrix} 0.05 & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 0.05 \end{pmatrix}\right), \quad [\sigma_2^2|z_1] \sim \mathcal{IG}(3, 1).$$

Figure 3.4 shows the result of the Bayesian co-kriging with the given prior information. The estimated RMSE is 0.79 and the efficiency *Eff* is 96.57%, we hence improve the accuracy of the metamodel. The predictive mean is closer to the true function and the predictive variance is reduced compared to the non-informative Bayesian case, with the confidence intervals that still contain the true function. The posterior distributions of the parameters are given in Table 3.3 and we have $\hat{\theta}_1 = 0.25$ and $\hat{\theta}_2 = 0.07$.

Regression Coefficient	Posterior mean
ρ	2.00
β_2	$(20.12, -19.81)$
β_1	-3.49
Variance Coefficient	**Posteriori harmonic average**
σ_1^2	32.75
σ_2^2	0.29

Table 3.3: A co-kriging example with one-dimensional functions and prior information. Posterior distribution of parameters.

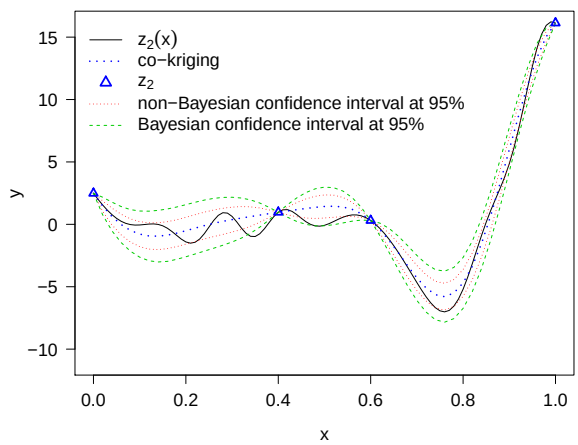


Figure 3.4: Example 2 with prior information. The prior information improves the accuracy of the co-kriging metamodel and the variance of the predictive distribution has decreased.

3.6 The case of s levels of code

The aim of this section is to perform a multi-level co-kriging with any number of codes. Let us consider s levels of code. The generalization of the previous model is straightforward. Actually, if we denote by $\boldsymbol{\beta} = (\boldsymbol{\beta}_1', \dots, \boldsymbol{\beta}_s')'$, $\boldsymbol{\rho} = (\rho_1, \dots, \rho_{s-1})$, $\sigma^2 = (\sigma_1^2, \dots, \sigma_s^2)$ and $\theta = (\theta_1, \dots, \theta_s)$, we have:

$$\forall x \in Q \qquad [Z_s(x)|\mathbf{Z} = \mathbf{z}, \boldsymbol{\beta}, \rho, \sigma^2, \boldsymbol{\theta}] \sim \mathcal{N}\left(m_{Z_s}(x), s_{Z_s}^2(x)\right),$$

where:

$$m_{Z_s}(x) = \mathbf{h}_s'(x)\boldsymbol{\beta} + \mathbf{k}_s'(x)\mathbf{V}_s^{-1}(\mathbf{z} - \mathbf{H}_s\boldsymbol{\beta}) \tag{3.27}$$

and:

$$s_{Z_s}^2(x) = \sigma_{Z_s}^2 - \mathbf{k}_s'(x)\mathbf{V}_s^{-1}\mathbf{k}_s(x). \tag{3.28}$$

Furthermore, the correlation matrix for $\mathbf{D}_t$ and $\rho_s = 0$, $\forall s \leq 0$. The matrix $\mathbf{V}_s$ has the form:

$$\mathbf{V}_s = \begin{pmatrix} \mathbf{V}^{(1,1)} & \dots & \mathbf{V}^{(1,s)} \\ \vdots & \ddots & \vdots \\ \mathbf{V}^{(s,1)} & \dots & \mathbf{V}^{(s,s)} \end{pmatrix}. \tag{3.29}$$

The s diagonal blocks of size $n_t \times n_t$ are defined by:

$$\mathbf{V}^{(t,t)} = \sigma_t^2 R_t(\mathbf{D}_t) + \sigma_{t-1}^2\rho_{t-1}^2 R_{t-1}(\mathbf{D}_t) + \dots + \sigma_1^2 \left(\prod_{i=1}^{t-1} \rho_i^2\right) R_1(\mathbf{D}_t) \tag{3.30}$$

and the off-diagonal blocks of size $n_t \times n_{\tilde{t}}$ are given by:

$$\mathbf{V}^{(t,\tilde{t})} = \left(\prod_{i=t}^{\tilde{t}-1} \rho_i\right) \mathbf{V}^{(t,t)}(\mathbf{D}_t, \mathbf{D}_{\tilde{t}}) \qquad 1 \leq t < \tilde{t} \leq s. \tag{3.31}$$

The vector $\mathbf{k}_s(x)$ is such that $\mathbf{k}_s(x) = (k_1^*(x, \mathbf{D}_1)', \dots, k_s^*(x, \mathbf{D}_s)')'$, where:

$$k_t^*(x, \mathbf{D}_t)' = \rho_{t-1}k_{t-1}^*(x, \mathbf{D}_t)' + \left(\prod_{i=t}^{s-1} \rho_i\right) \sigma_t^2 R_t(x, \mathbf{D}_t) \qquad 1 < t \leq s, \tag{3.32}$$

where $\left(\prod_{i=s}^{s-1} \rho_i\right) = 1$ and $k_1^*(x, \mathbf{D}_1)' = \left(\prod_{i=1}^{s-1} \rho_i\right) \sigma_1^2 R_1(x, \mathbf{D}_1)$. If we define:

$$F_k(\mathbf{D}_l) = \begin{pmatrix} \mathbf{f}_k'(x_1^{(l)}) \\ \vdots \\ \mathbf{f}_k'(x_{n_l}^{(l)}) \end{pmatrix} \qquad 1 \leq k, l \leq s,$$

then the matrix $\mathbf{H}_s$ can be written as:

$$\mathbf{H}_s = \begin{pmatrix} F_1(\mathbf{D}_1) & & & \\ \rho_1 F_1(\mathbf{D}_2) & F_2(\mathbf{D}_2) & 0 & \\ \rho_1\rho_2 F_1(\mathbf{D}_3) & \rho_2 F_2(\mathbf{D}_3) & & \\ \vdots & \vdots & \ddots & \\ \left(\prod_{i=1}^{s-1} \rho_i\right) F_1(\mathbf{D}_s) & \left(\prod_{i=2}^{s-1} \rho_i\right) F_2(\mathbf{D}_s) & \dots & F_s(\mathbf{D}_s) \end{pmatrix}, \tag{3.33}$$

$\mathbf{h}_s'(x)$ and $\mathrm{var}(Z_s(x)) = \sigma_{Z_s}^2$ are given by the equations (3.3) and (3.5).

3.6.1 Bayesian estimation of parameters for s levels of code

From the assumptions of conditional independence between $(\delta_t(x))_{x\in Q}$ and $(Z_{t-1}(x),\dots,Z_1(x))_{x\in Q}$, we can extend the Bayesian estimation of the parameters to the case of s levels. Note that we do not assume the independence of $\boldsymbol{\beta}_t$ and ρ_{t-1}. We can obtain a closed form expression for the posterior distribution of $(\boldsymbol{\beta}_t,\rho_{t-1})$. For all $t=2,\dots,s$, we have:

$$[(\rho_{t-1},\boldsymbol{\beta}_t)|\mathbf{z}_t,\mathbf{z}_{t-1},\boldsymbol{\theta}_t,\sigma_t^2]\sim\mathcal{N}\left(\left(\mathbf{H}_t'R_t(\mathbf{D}_t)^{-1}\mathbf{H}_t\right)^{-1}\mathbf{H}_t'R_t(\mathbf{D}_t)^{-1}\mathbf{z}_t,\sigma_t^2\left(\mathbf{H}_t'R_t(\mathbf{D}_t)^{-1}\mathbf{H}_t\right)^{-1}\right), \tag{3.34}$$

where $\mathbf{H}_t=[\mathbf{z}_{t-1}(\mathbf{D}_t)\quad F_t(\mathbf{D}_t)]$. Furthermore, if we note $\tilde{\boldsymbol{\lambda}}_t=\mathbb{E}[(\rho_{t-1},\boldsymbol{\beta}_t)|\mathbf{z}_t,\mathbf{z}_{t-1},\boldsymbol{\theta}_t,\sigma_t^2]$, then we have:

$$[\sigma_t^2|\mathbf{z}_t,\mathbf{z}_{t-1},\boldsymbol{\theta}_t]\sim\mathcal{IG}(\alpha_t,\frac{Q_t}{2}), \tag{3.35}$$

where $\alpha_t=(n_t-p_t-1)/2$ and $Q_t=(\mathbf{z}_t-\mathbf{H}_t\tilde{\boldsymbol{\lambda}}_t)'R_t(\mathbf{D}_t)^{-1}(\mathbf{z}_t-\mathbf{H}_t\tilde{\boldsymbol{\lambda}}_t)$.

The REML estimator of σ_t^2 is $\hat{\sigma}_t^2=Q_t/2\alpha_t$ and we can estimate $\boldsymbol{\theta}_t$ by minimizing the expression:

$$\log(|\det(R_t(\mathbf{D}_t))|)+(n_t-p_t-q_{t-1})\log(\hat{\sigma}_t^2). \tag{3.36}$$

The generalization of the Bayesian estimation previously presented is important since it shows that the parameter estimation for a s-levels co-kriging is equivalent in terms of numerical complexity to the one for s independent krigings.

3.6.2 Reduction of computational complexity of inverting the covariance matrix $\mathbf{V}_s$

$\mathbf{V}_s$ is an $(\sum_{i=1}^s n_i\times\sum_{i=1}^s n_i)$ matrix, its inverse can hence be difficult to process. We present in this subsection two propositions to reduce the complexity of the processing of $\mathbf{V}_s^{-1}$.

Proposition 3.1. *Let us consider the covariance matrix $\mathbf{V}_s$ presented in Equation (3.29). By sorting the experimental design sets such that $\forall t=2,\dots,s$, $\mathbf{D}_{t-1}=(x_1^{(t-1)},\dots,x_{n_{t-1}-n_t}^{(t-1)},x_1^{(t)},\dots,x_{n_t}^{(t)})=(\mathbf{D}_{t-1}\setminus\mathbf{D}_t,\mathbf{D}_t)$, $\forall t=2,\dots,s$ the inverse of the matrix $\mathbf{V}_s$ has the form:*

$$\mathbf{V}_s^{-1}=\begin{pmatrix}\mathbf{V}_{s-1}^{-1}+\begin{pmatrix}0 & 0\\ 0 & \frac{\rho_{s-1}^2R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2}\end{pmatrix} & -\begin{pmatrix}0\\ \frac{\rho_{s-1}R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2}\end{pmatrix}\\ -\begin{pmatrix}0 & \frac{\rho_{s-1}R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2}\end{pmatrix} & \frac{R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2}\end{pmatrix}\qquad \mathbf{V}_1^{-1}=\frac{R_1(\mathbf{D}_1)^{-1}}{\sigma_1^2}, \tag{3.37}$$

with $\mathbf{V}_{s-1}^{-1}$ an $(\sum_{i=1}^{s-1}n_i\times\sum_{i=1}^{s-1}n_i)$ matrix and $R_s(\mathbf{D}_s)^{-1}$ an $(n_s\times n_s)$ matrix.

Proof. The proof is proposed with the general form $\rho(x)=\mathbf{g}_{t-1}(x)\boldsymbol{\beta}_{\rho_{t-1}}$ for the adjustment coefficient. Throughout the proof, we denote by $\odot$ the matrix element-by-element product (see Appendix A.2). Let us consider the following sorting procedure:

$$\forall t=2,\dots,s\quad \mathbf{D}_{t-1}=(\mathbf{D}_{t-1}\setminus\mathbf{D}_t,\mathbf{D}_t).$$

The proof is based on the block-wise inversion formula of the covariance matrix $\mathbf{V}_s$. The covariance matrix $\mathbf{V}_s$ can be written with the form:

$$\mathbf{V}_s = \begin{pmatrix} \mathbf{V}_{s-1} & \mathbf{U}_{s-1} \\ \mathbf{U}'_{s-1} & \mathbf{V}^{(s,s)} \end{pmatrix} \qquad \mathbf{U}_{s-1} = \begin{pmatrix} \mathbf{V}^{(1,s)} \\ \vdots \\ \mathbf{V}^{(s-1,s)} \end{pmatrix},$$

where $\mathbf{V}_{s-1}$ is the covariance matrix of the random vector $(\mathbf{Z}_1, \ldots, \mathbf{Z}_{s-1})$ and $\mathbf{U}_{s-1}$ is the covariance matrix between $(\mathbf{Z}_1, \ldots, \mathbf{Z}_{s-1})$ and $\mathbf{Z}_s$. Classical block-inversion matrix formula gives that

$$\begin{pmatrix} \mathbf{V}_{s-1} & \mathbf{U}_{s-1} \\ \mathbf{U}'_{s-1} & \mathbf{V}^{(s,s)} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{V}_{s-1}^{-1} + \mathbf{V}_{s-1}^{-1}\mathbf{U}_{s-1}\mathbf{Q}_s^{-1}\mathbf{U}'_{s-1}\mathbf{V}_{s-1}^{-1} & -\mathbf{V}_{s-1}^{-1}\mathbf{U}_{s-1}\mathbf{Q}_s^{-1} \\ -\mathbf{Q}_s^{-1}\mathbf{U}'_{s-1}\mathbf{V}_{s-1}^{-1} & \mathbf{Q}_s^{-1} \end{pmatrix}.$$

where $\mathbf{Q}_s = \mathbf{V}^{(s,s)} - \mathbf{U}'_{s-1}\mathbf{V}_{s-1}^{-1}\mathbf{U}_{s-1}$. For $s > t$ the following equalities stands:

$$\begin{aligned} \mathbf{V}^{(t,s)} &= \operatorname{cov}\left(Z_t(\mathbf{D}_t), Z_s(\mathbf{D}_s)\right) \\ &= \operatorname{cov}\left(Z_t(\mathbf{D}_t), \rho_{s-1}(\mathbf{D}_s) \odot Z_{s-1}(\mathbf{D}_s) + \delta_s(\mathbf{D}_s)\right) \\ &= \operatorname{cov}\left(Z_t(\mathbf{D}_t), \rho_{s-1}(\mathbf{D}_s) \odot Z_{s-1}(\mathbf{D}_s)\right) \\ &= \left(\mathbf{1}_{n_t}\rho_{s-1}(\mathbf{D}_s)'\right) \odot \operatorname{cov}\left(Z_t(\mathbf{D}_t), Z_{s-1}(\mathbf{D}_s)\right) \\ &= \left(\mathbf{1}_{n_t}\rho_{s-1}(\mathbf{D}_s)'\right) \odot \mathbf{V}^{(t,s-1)}(\mathbf{D}_t, \mathbf{D}_s). \end{aligned}$$

Therefore, we have:

$$\mathbf{U}_{s-1} = \begin{pmatrix} \mathbf{V}^{(1,s)} \\ \vdots \\ \mathbf{V}^{(s-1,s)} \end{pmatrix} = \left(\mathbf{1}_{\sum_{i=1}^{s-1} n_i}\rho_{s-1}(\mathbf{D}_s)'\right) \odot \begin{pmatrix} \mathbf{V}^{(1,s-1)}(\mathbf{D}_1, \mathbf{D}_s) \\ \vdots \\ \mathbf{V}^{(s-1,s-1)}(\mathbf{D}_{s-1}, \mathbf{D}_s) \end{pmatrix}.$$

Denoting that

$$\begin{pmatrix} \mathbf{V}^{(1,s-1)}(\mathbf{D}_1, \mathbf{D}_s) \\ \vdots \\ \mathbf{V}^{(s-1,s-1)}(\mathbf{D}_{s-1}, \mathbf{D}_s) \end{pmatrix}$$

are the n_s last columns of $\mathbf{V}_{s-1}$, we obtain that:

$$\begin{aligned} \mathbf{V}_{s-1}^{-1}\mathbf{U}_{s-1} &= \mathbf{V}_{s-1}^{-1}\left(\left(\mathbf{1}_{\sum_{i=1}^{s-1} n_i}\rho_{s-1}(\mathbf{D}_s)'\right) \odot \begin{pmatrix} \mathbf{V}^{(1,s-1)}(\mathbf{D}_1, \mathbf{D}_s) \\ \vdots \\ \mathbf{V}^{(s-1,s-1)}(\mathbf{D}_{s-1}, \mathbf{D}_s) \end{pmatrix}\right) \\ &= \left(\mathbf{1}_{\sum_{i=1}^{s-1} n_i}\rho_{s-1}(\mathbf{D}_s)'\right) \odot \begin{pmatrix} 0_{(\sum_{i=1}^{s-1} n_i - n_s)\times n_s} \\ \mathbf{I}_{n_s} \end{pmatrix}. \end{aligned}$$

Furthermore, we have the equality

$$\begin{aligned} \mathbf{Q}_s &= \operatorname{cov}\left(Z_s(\mathbf{D}_s), Z_s(\mathbf{D}_s)\right) \\ &\quad -\operatorname{cov}\left(\mathbf{Z} \setminus Z_s(\mathbf{D}_s), Z_s(\mathbf{D}_s)\right)' \operatorname{cov}\left(\mathbf{Z} \setminus Z_s(\mathbf{D}_s), \mathbf{Z} \setminus Z_s(\mathbf{D}_s)\right)^{-1} \operatorname{cov}\left(\mathbf{Z} \setminus Z_s(\mathbf{D}_s), Z_s(\mathbf{D}_s)\right), \end{aligned}$$

with $\mathbf{Z} \setminus Z_s(\mathbf{D}_s) = (Z_1(\mathbf{D}_1), \ldots, Z_{s-1}(\mathbf{D}_{s-1}))$. Therefore, $\mathbf{Q}_s$ is the covariance matrix of $Z_s(\mathbf{D}_s)$ conditioned by $(Z_1(\mathbf{D}_1), \ldots, Z_{s-1}(\mathbf{D}_{s-1}))$. Furthermore, the equality:

$$Z_s(\mathbf{D}_s) = \rho_{s-1}(\mathbf{D}_s) \odot Z_{s-1}(\mathbf{D}_s) + \delta_s(\mathbf{D}_s)$$

implies that:

$$\begin{aligned} \operatorname{var}\left(Z_s(\mathbf{D}_s)|\mathbf{Z} \setminus Z_s(\mathbf{D}_s)\right) &= \operatorname{var}\left(\rho_{s-1}(\mathbf{D}_s) \odot Z_{s-1}(\mathbf{D}_s) + \delta_s(\mathbf{D}_s)|\mathbf{Z} \setminus Z_s(\mathbf{D}_s)\right) \\ &= \operatorname{var}\left(\delta_s(\mathbf{D}_s)|\mathbf{Z} \setminus Z_s(\mathbf{D}_s)\right), \end{aligned}$$

since $Z_{s-1}(\mathbf{D}_s)$ is $[\mathbf{Z} \setminus Z_s(\mathbf{D}_s)]$-measurable. Moreover, we have the equality

$$\operatorname{var}\left(Z_s(\mathbf{D}_s)|Z \setminus Z_s(\mathbf{D}_s)\right) = \operatorname{var}\left(\delta_s(\mathbf{D}_s)\right),$$

since $\delta_s(\mathbf{D}_s) \perp \mathbf{Z} \setminus Z_s(\mathbf{D}_s)$. Therefore, we have:

$$\mathbf{Q}_s = \operatorname{var}\left(\delta_s(\mathbf{D}_s)\right) = \sigma_s^2 R_s(\mathbf{D}_s).$$

From the previous equality, we deduce that

$$\mathbf{V}_{s-1}^{-1}\mathbf{U}_{s-1}\mathbf{Q}_s^{-1} = \begin{pmatrix} 0_{(\sum_{i=1}^{s-1} n_i - n_s)\times n_s} \\ \frac{(\rho_{s-1}(\mathbf{D}_s)\mathbf{1}'_{n_s})\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2} \end{pmatrix}$$

and

$$\mathbf{V}_{s-1}^{-1}\mathbf{U}_{s-1}\mathbf{Q}_s^{-1}\mathbf{U}'_{s-1}\mathbf{V}_{s-1}^{-1} = \begin{pmatrix} 0_{(\sum_{i=1}^{s-1} n_i - n_s)\times(\sum_{i=1}^{s-1} n_i - n_s)} & 0_{(\sum_{i=1}^{s-1} n_i - n_s)\times n_s} \\ 0_{n_s\times(\sum_{i=1}^{s-1} n_i - n_s)} & \frac{(\rho_{s-1}(\mathbf{D}_s)\rho_{s-1}(\mathbf{D}_s)')\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2} \end{pmatrix}.$$

Finally, we find that

$$\mathbf{V}_s^{-1} = \begin{pmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \\ \mathbf{W}'_{12} & \mathbf{W}_{22} \end{pmatrix},$$

where

$$\mathbf{W}_{11} = \begin{pmatrix} 0_{(\sum_{i=1}^{s-1} n_i - n_s)\times(\sum_{i=1}^{s-1} n_i - n_s)} & 0_{(\sum_{i=1}^{s-1} n_i - n_s)\times n_s} \\ 0_{n_s\times(\sum_{i=1}^{s-1} n_i - n_s)} & \frac{(\rho_{s-1}(\mathbf{D}_s)\rho_{s-1}(\mathbf{D}_s)')\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2}, \end{pmatrix}$$

$$\mathbf{W}_{12} = -\begin{pmatrix} 0_{(\sum_{i=1}^{s-1} n_i - n_s)\times n_s} \\ \frac{(\rho_{s-1}(\mathbf{D}_s)\mathbf{1}'_{n_s})\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2} \end{pmatrix},$$

$$\mathbf{W}'_{12} = -\begin{pmatrix} 0_{n_s\times(\sum_{i=1}^{s-1} n_i - n_s)} & \frac{(\mathbf{1}_{n_s}\rho_{s-1}(\mathbf{D}_s)')\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2} \end{pmatrix},$$

$$\mathbf{W}_{22} = \frac{R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2}.$$

Furthermore, with the equality $\mathbf{V}_1 = \operatorname{var}\left(Z_1(\mathbf{D}_1)\right) = \sigma_1^2 R_1(\mathbf{D}_1)$ we find the recursive form presented earlier in this subsection. □

This is a very important result since it shows that we can deduce $\mathbf{V}_s^{-1}$ from $R_t(\mathbf{D}_t)^{-1}$, $t = 1, \ldots, s$. Therefore, the complexity of the processing of $\mathbf{V}_s^{-1}$ is $\mathcal{O}(\sum_{i=1}^s n_i^3)$ instead of $\mathcal{O}((\sum_{i=1}^s n_i)^3)$.

From Equation (3.37) and the Bayesian estimation of parameters presented in Section 3.6.1, we have shown here that building a s-level co-kriging is equivalent in terms of numerical complexity to build s independent krigings.

We emphasize that, for practical applications, the form (3.37) for the inverse of $\mathbf{V}_s$ allows us to perform fine matrix regularization in the case of ill-conditioned problems. Indeed, $\mathbf{V}_s$ is invertible if and only if the matrices $R_t(\mathbf{D}_t)$, $t = 1, \ldots, s$ are invertible. Therefore, if the problem is ill-conditioned, we just have to regularize the matrices $R_t(\mathbf{D}_t)$ which are ill-conditioned too. Moreover, we can further simplify the problem by considering the proposition below.

Proposition 3.2. *Let us consider $\mathbf{V}_s$ the covariance matrix presented in Equation (3.29) and $k_s(x)$ the covariance vector presented in Equation (3.32). Then, we have the following equality:*

$$\mathbf{V}_s^{-1}\mathbf{k}_s(x) = \begin{pmatrix} \rho_{s-1}\mathbf{V}_{s-1}^{-1}\mathbf{k}_{s-1}(x) - \begin{pmatrix} 0_{(\sum_{i=1}^{s-1} n_i - n_s)\times 1} \\ \rho_{s-1}R_s(\mathbf{D}_s)^{-1}R_s(\mathbf{D}_s, \{x\}) \end{pmatrix} \\ R_s(\mathbf{D}_s)^{-1}R_s(\mathbf{D}_s, \{x\}) \end{pmatrix}. \tag{3.38}$$

Proof. We know that the vector $\mathbf{k}_s(x)$ is such that $\mathbf{k}_s(x) = (k_1^*(x, \mathbf{D}_1)', \ldots, k_s^*(x, \mathbf{D}_s)')'$, with:

$$k_t^*(x, \mathbf{D}_t)' = \rho_{t-1}'(\mathbf{D}_t) \odot k_{t-1}^*(x, \mathbf{D}_t)' + \left(\prod_{i=t}^{s-1} \rho_i(x)\right) \sigma_t^2 R_t(x, \mathbf{D}_t).$$

Let us denote by

$$\mathbf{A} = \begin{pmatrix} \mathbf{V}_{s-1}^{-1} + \begin{pmatrix} 0 & 0 \\ 0 & \frac{(\rho_{s-1}(\mathbf{D}_s)\rho_{s-1}(\mathbf{D}_s)')\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2} \end{pmatrix} \\ -\begin{pmatrix} 0 & \frac{(\mathbf{1}_{n_1}'\rho_{s-1}(\mathbf{D}_s)')\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2} \end{pmatrix} \end{pmatrix}$$

and

$$\mathbf{B} = \begin{pmatrix} -\begin{pmatrix} 0 \\ \frac{(\rho_{s-1}(\mathbf{D}_s)\mathbf{1}_{n_s})\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2} \end{pmatrix} \\ \frac{R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2} \end{pmatrix}.$$

The following equality stands:

$$\mathbf{k}_s(x)'\mathbf{V}_s^{-1} = \begin{pmatrix} \mathbf{k}_s(x)'\mathbf{A} & \mathbf{k}_s(x)'\mathbf{B} \end{pmatrix}.$$

Let us focus on the term $\mathbf{k}_s(x)'\mathbf{A}$, we have:

$$\begin{aligned} \mathbf{k}_s(x)'\mathbf{A} &= (k_1^*(x, \mathbf{D}_1)', \ldots, k_{s-1}^*(x, \mathbf{D}_{s-1})') \left(\mathbf{V}_{s-1}^{-1} + \begin{pmatrix} 0 & 0 \\ 0 & \frac{(\rho_{s-1}(\mathbf{D}_s)\rho_{s-1}(\mathbf{D}_s)')\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2} \end{pmatrix}\right) \\ &\quad -k_s^*(x, \mathbf{D}_s)' \begin{pmatrix} 0 & \frac{(\mathbf{1}_{n_s}\rho_{s-1}(\mathbf{D}_s)')\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2} \end{pmatrix}. \end{aligned}$$

We note that we have the equality:

$$(k_1^*(x,\mathbf{D}_1)',\ldots,k_{s-1}^*(x,\mathbf{D}_{s-1})') = \rho_{s-1}(x)\mathbf{k}_{s-1}'(x).$$

Indeed, the vector $(k_1^*(x,\mathbf{D}_1)',\ldots,k_{s-1}^*(x,\mathbf{D}_{s-1})')$ represents the covariance between $Z_s(x)$ and $(Z_1(\mathbf{D}_1),\ldots,Z_{s-1}(\mathbf{D}_{s-1})$. Therefore, we have:

$$\begin{aligned}(k_1^*(x,\mathbf{D}_1)',\ldots,k_{s-1}^*(x,\mathbf{D}_{s-1})') &= \mathrm{cov}\left(Z_s(x),(Z_1(\mathbf{D}_1),\ldots,Z_{s-1}(\mathbf{D}_{s-1}))\right)\\ &= \mathrm{cov}\left(\rho_{s-1}(x)Z_{s-1}(x)+\delta_s(x),(Z_1(\mathbf{D}_1),\ldots,Z_{s-1}(\mathbf{D}_{s-1}))\right)\end{aligned}$$

and the independence $\delta_s(x)\perp(Z_1(x),\ldots,Z_{s-1}(x))$, gives that:

$$\begin{aligned}(k_1^*(x,\mathbf{D}_1)',\ldots,k_{s-1}^*(x,\mathbf{D}_{s-1})') &= \mathrm{cov}\left(\rho_{s-1}(x)Z_{s-1}(x),(Z_1(\mathbf{D}_1),\ldots,Z_{s-1}(\mathbf{D}_{s-1}))\right)\\ &= \rho_{s-1}(x)\mathrm{cov}\left(Z_{s-1}(x),(Z_1(\mathbf{D}_1),\ldots,Z_{s-1}(\mathbf{D}_{s-1}))\right)\\ &= \rho_{s-1}(x)\mathbf{k}_{s-1}'(x).\end{aligned}$$

Let us return to the term $\mathbf{k}_s(x)'\mathbf{A}$. Noticing that

$$k_{s-1}^*(x,\mathbf{D}_{s-1})' = (k_{s-1}^*(x,\mathbf{D}_{s-1}\setminus\mathbf{D}_s)' \quad k_{s-1}^*(x,\mathbf{D}_s)'),$$

we obtain the following equality:

$$\begin{aligned}\mathbf{k}_s(x)'\mathbf{A} &= \rho_{s-1}(x)\mathbf{k}_{s-1}'(x)\mathbf{V}_{s-1}^{-1}+\begin{pmatrix}0 & k_{s-1}^*(x,\mathbf{D}_s)'\frac{(\rho_{s-1}(\mathbf{D}_s)\rho_{s-1}(\mathbf{D}_s)')\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2}\end{pmatrix}\\ &\quad -k_s^*(x,\mathbf{D}_s)'\begin{pmatrix}0 & \frac{(\mathbf{1}_{n_s}\rho_{s-1}(\mathbf{D}_s)')\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2}\end{pmatrix}.\end{aligned}$$

We know that $k_s^*(x,\mathbf{D}_s)' = \rho_{s-1}(\mathbf{D}_s)'\odot k_{s-1}^*(x,\mathbf{D}_s)'+\sigma_s^2R_s(x,\mathbf{D}_s)$. Therefore, we can deduce that:

$$\begin{aligned}\mathbf{k}_s(x)'\mathbf{A} &= \rho_{s-1}(x)\mathbf{k}_{s-1}'(x)\mathbf{V}_{s-1}^{-1}-R_s(x,\mathbf{D}_s)\begin{pmatrix}0 & (\mathbf{1}_{n_s}\rho_{s-1}(\mathbf{D}_s)')\odot R_s(\mathbf{D}_s)^{-1}\end{pmatrix}\\ &\quad \rho_{s-1}(x)\mathbf{k}_{s-1}'(x)\mathbf{V}_{s-1}^{-1}-\begin{pmatrix}0_{1\times(\sum_{i=1}^{s-1}n_i-n_s)} & (\rho_{s-1}(\mathbf{D}_s)'\odot R_s(\{x\},\mathbf{D}_s))R_s(\mathbf{D}_s)^{-1}\end{pmatrix}.\end{aligned}$$

Let us focus now on the term $\mathbf{k}_s(x)'\mathbf{B}$:

$$\begin{aligned}\mathbf{k}_s(x)'\mathbf{B} &= -(k_1^*(x,\mathbf{D}_1)',\ldots,k_{s-1}^*(x,\mathbf{D}_{s-1})')\begin{pmatrix}0\\ \frac{(\rho_{s-1}(\mathbf{D}_s)\mathbf{1}_{n_s}')\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2}\end{pmatrix}+k_s^*(x,\mathbf{D}_s)'\frac{R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2}\\ &= -k_{s-1}^*(x,\mathbf{D}_s)'\frac{\left(\rho_{s-1}(\mathbf{D}_s)\mathbf{1}_{n_s}'\right)\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2}+k_s^*(x,\mathbf{D}_s)'\frac{R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2}\\ &= -k_{s-1}^*(x,\mathbf{D}_s)'\frac{\left(\rho_{s-1}(\mathbf{D}_s)\mathbf{1}_{n_s}'\right)\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2}\\ &\quad +\left(\rho_{s-1}(\mathbf{D}_s)'\odot k_{s-1}^*(x,\mathbf{D}_s)'+\sigma_s^2R_s(x,\mathbf{D}_s)\right)\frac{R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2}\\ &= R_s(x,\mathbf{D}_s)R_s^{-1}.\end{aligned}$$

Finally we obtain:

$$\mathbf{V}_s^{-1}\mathbf{k}_s(x) = \begin{pmatrix} \rho_{s-1}\mathbf{V}_{s-1}^{-1}\mathbf{k}_{s-1}(x) - \begin{pmatrix} 0 \\ \rho_{s-1}R_s(\mathbf{D}_s)^{-1}R_s(\mathbf{D}_s, x) \end{pmatrix} \\ R_s(\mathbf{D}_s)^{-1}R_s(\mathbf{D}_s, x) \end{pmatrix}.$$

□

Therefore, $\mathbf{k}_s'(x)\mathbf{V}_s^{-1}$ is independent of σ_s^2. Since $\mathbf{k}_1(x)\mathbf{V}_1^{-1} = R_1(\{x\}, \mathbf{D}_1)R_1(\mathbf{D}_1)^{-1}$ does not depend on σ_1^2, by induction, $\mathbf{k}_s'(x)\mathbf{V}_s^{-1}$ is independent of σ_i^2 for all $1 \leq i \leq s$. We have just shown here that the co-kriging mean does not depend on the variance coefficients.

3.6.3 Numerical test on the reduction of computational complexity

In the previous section, we have presented a reduction of complexity for the co-kriging model by expressing the inverse of the matrix $\mathbf{V}_s$ with the inverses of the matrices $R_t(\mathbf{D}_t)$, $t = 1, \ldots, s$. We present here a numerical test to highlight the gain of CPU time obtained with this method. We focus on the case of 2 levels of code with constant regression functions and the following Gaussian kernel for the 2 levels:

$$r(x - \tilde{x}; \theta) = \exp\left(-\frac{(x - \tilde{x})^2}{2\theta^2}\right).$$

The experimental design set for the cheap code is a regular grid composed of n_1 points between 0 and 1 and the experimental design set for the expensive code are the n_2 first points of this grid. We consider the relation $n_1 = 4n_2$ with $n_2 = 50, 60, \ldots, 500$ and the parameter $\theta = 5/n_2$ (the parameter θ is controlled by n_2 in order to avoid ill-conditioned covariance matrices). The total number of observations is hence $n = n_1 + n_2$. Figure 3.5 compares the CPU time needed to build a co-kriging model with or without reduction complexity.

First, the slope of the two CPU times is close to 3 (the least-squares estimate value is 3.03). The complexity of a matrix inversion being $\mathcal{O}(n^3)$, with n the size of the matrix, the estimate of the slope highlights the fact that it is the matrix inversion which leads the CPU time. Then, Figure 3.5 emphasizes that the reduction of complexity is worthwhile. Indeed, we see that the ratio between the two CPU times is approximately a constant equal to 1.93. We are hence close to the theoretical ratio equal to $(n_1 + n_2)^3/(n_1^3 + n_2^3) \approx 1.92$ which is obtained when we consider that the CPU time is essentially due to the matrix inversion.

3.6.4 Academic example on the complexity reduction

A 3-level co-kriging metamodel is presented in this section to illustrate the gain of CPU which can be obtained with the presented reduction of complexity. We focus on the inversion of the co-kriging matrix $\mathbf{V}_s$ by comparing the CPU time needed with a direct inversion or by using the formula (3.37). We assume that the 3 levels of code are given by the followings three

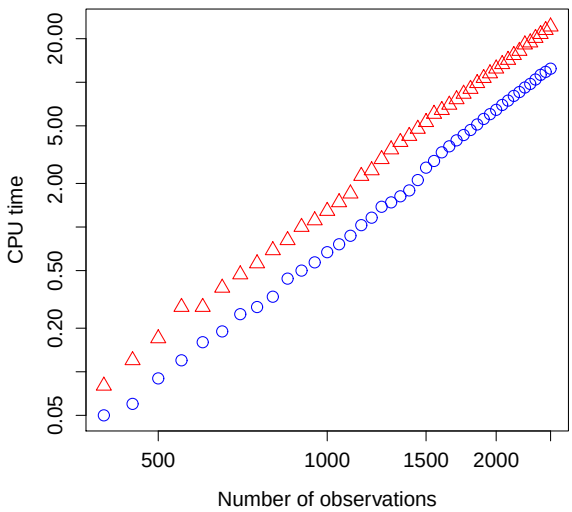


Figure 3.5: CPU time comparison between 2-level co-kriging models. The triangles represent the CPU time for the crude co-kriging model and the circles represent the CPU time for the co-kriging model with the complexity reduction. The gain of CPU time with the reduction complexity is approximately a factor equal to 1.93.

dimensional functions:

$$z_1(x) = \sin(x_1), \tag{3.39}$$
$$z_2(x) = z_1(x) + a\sin(x_2)^2, \tag{3.40}$$
$$z_3(x) = z_2(x) + bx_3^4\sin(x_1), \tag{3.41}$$

with $x = (x_1, x_2, x_3) \in [-\pi, \pi]^3$, $a = 7$ and $b = 1/10$. We note that the complex function $z_3(x)$ corresponds to the Ishigami function which is very popular in the field of sensitivity analysis [Saltelli et al., 2000]. We consider $n_3 = 50$ observations for the most accurate code $z_3(x)$, $n_2 = 200$ for the intermediate code and $n_1 = 400$ for the less accurate code. All experimental design sets are randomly sampled from the uniform distribution. As presented in Section 3.2 we consider nested experimental designs $\forall t = 2, \ldots, s \quad \mathbf{D}_t \subseteq \mathbf{D}_{t-1}$.

We use a tensorised Matérn-5/2 kernel for the three correlation functions:

$$r_t(x, \tilde{x}; \boldsymbol{\theta}_t) = \prod_{i=1}^{d} r_{1D}(x_i, \tilde{x}_i; \boldsymbol{\theta}_{t,i}), \tag{3.42}$$

with $r_{1D}(t, \tilde{t}; \theta) = \left(1 + \sqrt{5}\frac{|t-\tilde{t}|}{\theta} + \frac{5}{3}\frac{(t-\tilde{t})^2}{\theta^2}\right)\exp\left(-\sqrt{5}\frac{|t-\tilde{t}|}{\theta}\right)$, $t, \tilde{t} \in \mathbb{R}$ and constant regression functions $\mathbf{f}_t(x) = 1$.

The estimates of the hyper-parameters $\boldsymbol{\theta}_t$ are presented in Table 3.4.

Parameter	**Estimate**
$\hat{\boldsymbol{\theta}}_1$	$\begin{pmatrix} 0.61 & 1.99 & 2.04 \end{pmatrix}$
$\hat{\boldsymbol{\theta}}_2$	$\begin{pmatrix} 1.98 & 0.26 & 2.48 \end{pmatrix}$
$\hat{\boldsymbol{\theta}}_3$	$\begin{pmatrix} 0.23 & 0.89 & 0.21 \end{pmatrix}$

Table 3.4: Academic example on the complexity reduction. Estimates of the hyper-parameters (correlation lengths) for the 3-level co-kriging.

The hyper-parameter estimates show us that $z_1(x)$ is very smooth in the directions x_2 and x_3 reflecting the fact that it depends only on the first direction x_1. Similarly, the bias between $z_2(x)$ and $z_1(x)$ only depending on the second direction x_2, it is rough in this direction and very smooth in the other ones. Finally, the bias between $z_3(x)$ and $z_2(x)$ is rougher in the direction x_3 than in the directions x_1 and x_2. This is due to the important impact of x_3 on the third level.

The estimates of the variance, scale and regression parameters are given in Table 3.5.

Parameter	Estimate
$\boldsymbol{\beta}_1$	0.00
$\begin{pmatrix} \boldsymbol{\beta}_{\rho_1} \\ \boldsymbol{\beta}_2 \end{pmatrix}$	$\begin{pmatrix} 0.99 \\ 2.44 \end{pmatrix}$
$\begin{pmatrix} \boldsymbol{\beta}_{\rho_2} \\ \boldsymbol{\beta}_3 \end{pmatrix}$	$\begin{pmatrix} 0.95 \\ 0.64 \end{pmatrix}$
σ_1^2	0.09
σ_2^2	1.66
σ_3^2	6.25

Table 3.5: Academic example on the complexity reduction. Estimates of the variance, scale and regression parameters for the 3-level co-kriging.

Table 3.5 shows the efficiency of the suggested method for the parameter estimations since it provides very accurate estimates of ρ_1 and ρ_2.

To evaluate the accuracy of the co-kriging model, we use a test set of 30,000 points uniformly sampled from the uniform distribution. Then, we compute the efficiency *Eff* with the co-kriging predictions and the responses of $z_3(x)$ on this set. We obtain for the co-kriging model *Eff* $= 83.21\%$, we hence have a good accuracy despite the small number of observations used for the high fidelity model. Nonetheless, we have a significant improvement relatively to the kriging model since with the same kernel and the same experimental design set $\mathbf{D}_3$ we obtain *Eff* $= 47.97\%$ which is a very poor accuracy. The hyper-parameter estimate of the kriging model is $\hat{\boldsymbol{\theta}} = (0.79, 0.14, 0.29)$, the variance one is $\hat{\sigma}^2 = 13.66$ and the trend coefficient one is $\hat{\boldsymbol{\beta}} = 3.89$.

Let us now compare the difference of CPU time between the co-kriging building with a crude inversion of the covariance matrix $\mathbf{V}_s$ and the one with an inversion using the formula presented in Subsection 3.6.2. The CPU time necessary without the reduction complexity is $\text{CPU}_{\text{crude}} = 0.47$ whereas the one necessary with the complexity reduction is $\text{CPU}_{\text{light}} = 0.14$. We hence find that the CPU time ratio between the two methods approximately equals 3.36. This is not far from the theoretical ratio which equals $650^3/(400^3 + 200^3 + 50^3) \approx 3.80$. We note that the complexity reduction could be of important practical interest. For example, without it the computational cost of a leave-one-out cross validation procedure will be much more important (the ratio will still be around 3 in our example). The complexity of this procedure being $\mathcal{O}(n^4)$, the gain of CPU time will be substantial.

3.6.5 Comparison with existing methods on an academic example

We proceed here on a numerical comparison between the suggested model and the ones presented by [Kennedy and O'Hagan, 2000] and [Qian and Wu, 2008]. The comparison is made both in terms of RMSE and computational resources. For the comparison, we consider a 2-level co-kriging model with the following functions:

$$\begin{cases} z_1(x) &= \sin(x_1) + a\sin(x_2)^2 \\ z_2(x) &= z_1(x) + bx_3^4\sin(x_1) \end{cases}, \tag{3.43}$$

with $x = (x_1, x_2, x_3) \in [-\pi, \pi]^3$, $a = 7$ and $b = 1/10$. Furthermore, the experimental design set $\mathbf{D}_1$ for the coarse code $z_1(x)$ is composed of 100 points uniformly spread on $[-\pi, \pi]^3$ and the experimental design set for the fine code $z_2(x)$ is composed of 50 points randomly extracted from $\mathbf{D}_1$. Then, we consider a test set X_{test} of 1000 points uniformly spread on $[-\pi, \pi]^3$. In order to propose a fair comparison, we use the R-CRAN package "approximator.1.2-2" on the R.2.15.2 platform to implement the model of Kennedy and O'Hagan. This package has been specially created to compute the equations given by [Kennedy and O'Hagan, 2000]. Then, we use the WinBUGS software version 1.4.3 to implement the model presented by [Qian and Wu, 2008]. It is a software specially dedicated to Bayesian analysis and particularly efficient to develop Metropolis-within-Gibbs algorithms [Liu, 2001]. Finally, we use the R-CRAN package "MuFiCokriging.1.2" to implement our model. This package computes the mean and the variance of the predictive distribution presented in Subsection 3.4.3 and integrates the proposed complexity reductions (see Chapter 4 Section 4.6). For the two correlation functions $r_1(x, x')$ and $r_2(x, x')$ we use Gaussian covariance kernels for the three models

$$r_j(x, x') = \exp\left(-\frac{1}{2}\sum_{i=1}^{d}\frac{(x_i - x_i')^2}{\theta_{i,j}^2}\right),$$

and for the model presented by Qian et al. we assume a Gaussian covariance kernel for the adjustment coefficient. Furthermore, we assume a constant trend for the Gaussian processes modeling the coarse code and the bias between the two codes.

The correlation parameters and the adjustment parameter of the model presented by Kennedy and O'Hagan are estimated with a concentrated likelihood method with a joint estimation of $(\theta_{i,2})_{i=1,\dots,3}$ and ρ as presented in their paper. The other parameters are estimated

with a classical maximum likelihood estimate. Note that in this model the scaling coefficient ρ is constant.

The correlation parameters of the model presented by Qian et al. are estimated with a Bayesian method and the prior for each of them is $\Gamma(2, 0.1)$ where Γ stands for the Gamma distribution. As in [Qian and Wu, 2008] we consider these parameters as known and fixed to the modes of their posterior distributions. Furthermore, for the Bayesian procedure the convergence is achieved after 50,000 burn-in iterations and another 100,000 runs are then generated to compute the posterior distributions as in [Qian and Wu, 2008]. We note that the convergence is assessed both visually and with the method of Geweke [Geweke et al., 1991] as presented by Qian et al.. The other parameters are estimated thanks to the Metropolis-within-Gibbs algorithm with the following parameters for the prior distributions:

- $(\alpha_l, \gamma_l, \alpha_\rho, \gamma_\rho, \alpha_\delta, \gamma_\delta) = (2, 1, 2, 1, 2, 1)$,
- $u_l = 0$,
- $\nu_l = 1$,
- $(u_\rho, \nu_\rho, u_\delta, \nu_\delta) = (1, 1, 0, 1)$,

The reader is referred to [Qian and Wu, 2008] for more detail about these parameters. They reflect that we do not have information about the variance and the regression parameters of the model. Moreover, the prior information on ρ is such that its mean is centered on 1. We note that in this model, ρ depends on x. For the Bayesian procedure, the convergence is reached again after 50,000 burn-in iterations and another 100,00 runs are then generated.

The prediction RMSE of the model presented in Section 3.4 is compared with the ones of the models presented by Kennedy and O'Hagan and Qian et al. on 100 different experimental design sets $\mathbf{D}_1$ and $\mathbf{D}_2$ and test sets $\mathbf{X}_{\text{test}}$. The resulting RMSEs for the three models are given in Figure 3.6.

We see in Figure 3.6 that the RMSEs of the presented model and the one of Qian et al. are significantly better than the one of the model of Kennedy and O'Hagan. Furthermore, our model is slightly better than the one of Qian et al. in terms of RMSE. Indeed, we see that the notches in Figure 3.6 do not overlap. According to [Chambers et al., 1983] p.62, this means that the difference between the two medians are significant. We note that the correlation length for the model of Qian et al. and the one obtained with the restricted maximum likelihood method (see Subsection 3.6.1) are similar, i.e. around $(1.60, 0.45, 1.95)$ for $\boldsymbol{\theta}_1$ and around $(0.30, 1.90, 0.30)$ for $\boldsymbol{\theta}_2$. The difference of RMSE between the proposed model and the one of Qian et al. is essentially explained by a less efficient estimation of the parameter ρ for the model of Qian et al.. Indeed, it varies around 1.13 whereas the real value is 1. Moreover, with the estimation method presented in Subsection 3.6.1 the parameter ρ is estimated to be around 0.99. This highlights the importance to have an efficient estimation of this parameter.

Finally, we compare the three methods in terms of computational costs. Figure 3.7 illustrates the different CPU times obtained from the 100 different experimental and test sets. We see in Figure 3.7 that there is a significant difference between the model CPU times. Indeed, the ratio of CPU time between the model of Kennedy and O'Hagan and the presented one is

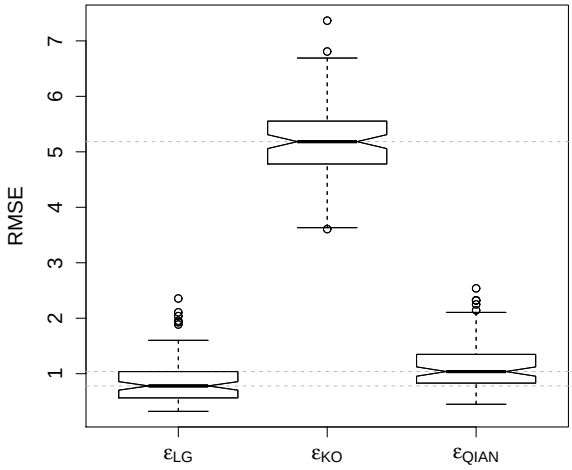


Figure 3.6: RMSEs of the presented model $\varepsilon_{\mathrm{LG}}$, the model of [Kennedy and O'Hagan, 2000] $\varepsilon_{\mathrm{KO}}$ (see [Kennedy and O'Hagan, 2000]) and the model of [Qian and Wu, 2008] $\varepsilon_{\mathrm{QIAN}}$ (see [Qian and Wu, 2008]). The numerical comparisons are performed on the 3-dimensional academic example (3.43) with 100 different experimental and test sets.

around 10 whereas the one between the model of Qian et al. and the presented one is around 1000. The important difference between the model of Qian et al. and the other models is natural since in this model a complex Bayesian scheme is used which is known to be expensive. The one between the suggested model and the one of Kennedy and O'Hagan can be explained by the complexity reduction for the covariance matrix inversion.

3.7 Example : Fluidized-Bed Process

This example illustrates the comparison between 2-level and 3-level co-kriging. A 3-level co-kriging method is applied to a physical experiment modeled by a computer code. The experiment, which is the measurement of the temperature of the steady-state thermodynamic operation point for a fluidized-bed process, was presented by [Dewettinck et al., 1999], who developed a computer model named "Topsim" to calculate the measured temperature. The code, developed for a Glatt GPCG-1, fluidized-bed unit in the top-spray configuration, can be run at 3 levels of complexity. We hence have 4 available responses:

- $\mathbf{T}_{exp}$: the experimental response.
- $\mathbf{T}_3$: the most accurate code modeling the experiment.
- $\mathbf{T}_2$: a simplified version of $\mathbf{T}_3$.
- $\mathbf{T}_1$: the lowest accurate code modeling the experiment.

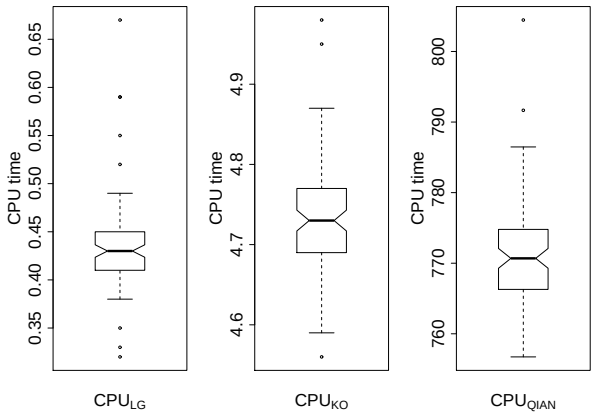


Figure 3.7: CPU times for the presented model $\mathrm{CPU_{LG}}$, the one of Kennedy and O'Hagan $\mathrm{CPU_{KO}}$ and the one of Qian et al. $\mathrm{CPU_{QIAN}}$ (note that the scales are different). The numerical comparisons are performed on the 3-dimensional academic example (3.43) with 100 different experimental and test sets. The ratio between $\mathrm{CPU_{KO}}$ and $\mathrm{CPU_{LG}}$ is around 10 and the ratio between $\mathrm{CPU_{QIAN}}$ and $\mathrm{CPU_{LG}}$ is around 1000.

The differences between $\mathbf{T}_1$, $\mathbf{T}_2$ and $\mathbf{T}_3$ are discussed by Dewettinck et al. (1999). The aim of this study is to predict the experimental response $\mathbf{T}_{exp}$ given the two levels of code $\mathbf{T}_3$ and $\mathbf{T}_2$. We only focus on a 3-level co-kriging using $\mathbf{T}_3$ and $\mathbf{T}_2$ to predict $\mathbf{T}_{exp}$ since 28 responses available for each level is not enough to build a nested experimental design relevant for a 4-level co-kriging. The experimental design set and the responses $\mathbf{T}_1$, $\mathbf{T}_2$, $\mathbf{T}_3$ and $\mathbf{T}_{exp}$ are given by [Qian and Wu, 2008] who have presented a 2-level co-kriging using $\mathbf{T}_{exp}$ and $\mathbf{T}_2$. Furthermore, the responses are parameterized by a 6-dimensional input vector presented by Dewettinck et al. (1999).

3.7.1 Building the 3-level co-kriging

To build the 3-level co-kriging, we use 10 measures of $\mathbf{T}_{exp}$ (measures 1, 3, 8, 10, 12, 14, 18, 19, 20, 27 in Table 4 in [Qian and Wu, 2008]), 20 simulations of $\mathbf{T}_3$ (runs 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 18, 19, 20, 22, 24, 27) and the 28 simulations of $\mathbf{T}_2$ and the input vector is scaled between 0 and 1. The last 18 measures of $\mathbf{T}_{exp}$ are used for validation. The design sets are nested such that $\mathbf{D}_{t-1} = (\mathbf{D}_{t-1} \setminus \mathbf{D}_t, \mathbf{D}_t)$ for $t = 2, 3$ and we use a Matérn-5/2 kernel for the three covariance functions. The estimates of the hyper-parameters which represent correlation lengths of the three covariance kernels are given in Table 3.6.

$\hat{\boldsymbol{\theta}}_1$	1.790	3.988	1.218	1.790	3.595	0.722
$\hat{\boldsymbol{\theta}}_2$	1.810	1.842	2.008	1.036	0.001	0.345
$\hat{\boldsymbol{\theta}}_3$	0.890	0.721	2.008	2.952	1.790	0.241

Table 3.6: Example: fluidized-bed process. Estimates of the hyper-parameters (correlation lengths) for the 3-level co-kriging.

The estimates of hyper-parameters in Table 3.6 show us that the surrogate model is very smooth in the first four directions. For the fifth direction the Gaussian processes modeling the cheap code $\mathbf{T}_2$ and the bias between $\mathbf{T}_{exp}$ and $\mathbf{T}_3$ are very smooth and the one modeling the bias between $\mathbf{T}_3$ and $\mathbf{T}_2$ is close to a regression. Finally, the model is more oscillating in the sixth direction in particular for the two biases where correlation lengths are around 0.3.

Furthermore, Table 3.7 gives the estimates of the variance and regression parameters (see Section 3.6.1).

Regression coefficient	Posterior mean	Posterior Covariance/σ_t^2
$\boldsymbol{\beta}_1$	47.02	0.134
$\begin{pmatrix} \boldsymbol{\beta}_{\rho_1} \\ \boldsymbol{\beta}_2 \end{pmatrix}$	$\begin{pmatrix} 0.97 \\ -0.17 \end{pmatrix}$	$\begin{pmatrix} 0.001 & -0.034 \\ -0.034 & 1.610 \end{pmatrix}$
$\begin{pmatrix} \boldsymbol{\beta}_{\rho_2} \\ \boldsymbol{\beta}_3 \end{pmatrix}$	$\begin{pmatrix} 0.95 \\ 1.93 \end{pmatrix}$	$\begin{pmatrix} 0.003 & -0.121 \\ -0.121 & 5.188 \end{pmatrix}$
Variance coefficient	Q_t	α_t
σ_1^2	1032	13.5
σ_2^2	5.30	9
σ_3^2	8.39	4

Table 3.7: Example: fluidized-bed process. Bayesian estimation of the variance and regression parameters for the 3-level co-kriging.

Table 3.7 shows that the responses have approximately the same scale since the adjustment coefficients are close to 1. Furthermore, we see an important bias between $\mathbf{T}_3$ and $\mathbf{T}_2$ with $\boldsymbol{\beta}_3 = 1.93$. Finally, the variance coefficients for the biases indicate that they are possibly much simpler to model than the cheap code $\mathbf{T}_2$ as their estimates are smaller.

3.7.2 3-level co-kriging prediction: predictions when code output is available

The aim of this section is to show that co-kriging can improve significantly the accuracy of the surrogate model at points where at least one level of responses is available.

The predictions of the 3-level co-kriging are here presented and compared with the predictions obtained with a 2-level co-kriging using only the 10 responses of $\mathbf{T}_{exp}$ and the 20 responses of $\mathbf{T}_3$. The predictions for the 2-level and the 3-level co-krigings vs. the real values

(i.e., the measured temperature $\mathbf{T}_{exp}$) are shown in Figure 3.8. The 3-level co-kriging gives us

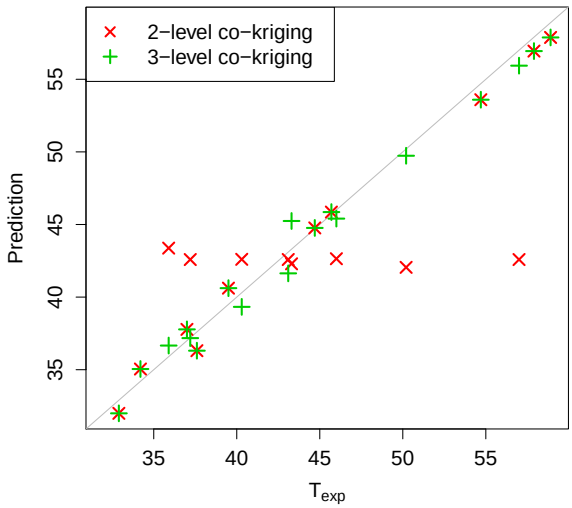


Figure 3.8: Predictions of the 2-level and the 3-level co-krigings for the fluidized-bed process. The 3-level co-kriging improves significantly the predictions of the 2-level one.

the same prediction means as the 2-level co-kriging at the 10 points (points 2, 5, 6, 7, 9, 11, 13, 16, 22, 24) where $\mathbf{T}_3$ is known. These overlapped points mean that $\mathbf{T}_2$ does not influence the surrogate model at these points. This follows from the Markov property introduced in Section 3.2, which implies that the prediction of $\mathbf{T}_{exp}$ is entirely determined by $\mathbf{T}_3$ at these points. We also note that, in general, the 2-level co-kriging predictions - at points where $\mathbf{T}_3$ is unknown - are not accurate and the 3-level co-kriging improves significantly the prediction means compared to the 2-level co-kriging. Table 3.8 compares the 2-level co-kriging with the 3-level co-kriging and summarizes some results about the quality of the predictions on the 18 validation points. Nonetheless, it is important to notice that, in the 3-level case, the output of the cheapest code $\mathbf{T}_2$ is known at the 18 test points. This means that the results of this subsection show that the 3-level co-kriging prediction is more accurate than the 2-level co-kriging prediction at a point where the cheapest response $\mathbf{T}_2$ is available. In the next subsection we will show that the 3-level co-kriging prediction is more accurate than the 2-level one at a point where no response is available.

	Eff	RMSE	MaxAE
2-level co-kriging	61.23 %	4.24	14.04
3-level co-kriging	98.71 %	0.89	1.98
	Average Std. dev.	Median Std. dev.	Maximal Std. dev
2-level co-kriging	2.90	1.02	5.68
3-level co-kriging	0.90	1.02	1.04

Table 3.8: Example: fluidized-bed process. Comparison between 2-level co-kriging and 3-level co-kriging. Predictions are better in the 3-level case and the prediction variance seems well-evaluated since the RMSE and the average standard deviation are close.

Figure 3.9 shows the prediction errors of the 2-level co-kriging and the confidence interval at plus or minus twice the prediction standard deviation. The last 10 prediction errors and their confidence intervals are the same as those of the 3-level case since it corresponds to the points where $\mathbf{T}_3$ is known. We see in Figure 3.9 that the confidence intervals are well

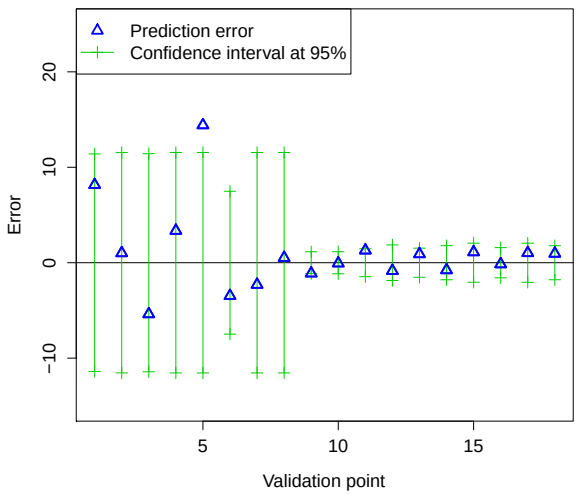


Figure 3.9: Prediction errors of the 2-level co-kriging and confidence intervals at plus or minus twice the standard deviation. We see a significant difference between the accuracy of the predictions means and their confidence intervals for the point where $\mathbf{T}_3$ is unknown (the 8 first validation points) and for the ones where it is known (the last 10 validation points).

predicted. Furthermore, we see a significant difference between the accuracy of the prediction means and their confidence intervals for the point where $\mathbf{T}_3$ is unknown (the 8 first validation points) and for the ones where it is known (the last 10 validation points).

3.7.3 3-level co-kriging prediction: predictions when code output is not available

In this subsection, we show that a multi-level co-kriging can significantly improve the prediction of a surrogate model at points where no response is available.

We have seen in Section 3.7.2 that the 3-level co-kriging improves significantly the 2-level co-kriging at points where $\mathbf{T}_3$ is unknown and $\mathbf{T}_2$ has been sampled. Nevertheless, to have a fair comparison between these two co-kriging models, we compare their accuracy by applying a Leave-One-Out Cross-Validation (LOO-CV) procedure at the 10 points where $\mathbf{T}_{exp}$ is known. This means that we perform for each of these 10 points the following procedure:

1. The experimental and the two code outputs corresponding to the point are removed from the data set.
2. The 2-level co-kriging method and the 3-level co-kriging method are applied using the truncated data set in order to give a confidence interval for the experimental output at the point.

Figure 3.10 shows the result of the LOO-CV procedure for the 2-level and 3-level co-kriging. We see that the 3-level co-kriging is more accurate than the 2-level one. Indeed, the LOO-CV RMSE for the 2-level co-kriging is equal to 1.88 whereas it is equal to 1.09 for the 3-level co-kriging. This shows that the 3-level co-kriging provides better predictions also at points where no response is available. This highlights the strength of the proposed method and shows that a co-kriging method with more than 2 levels of code can be worthwhile.

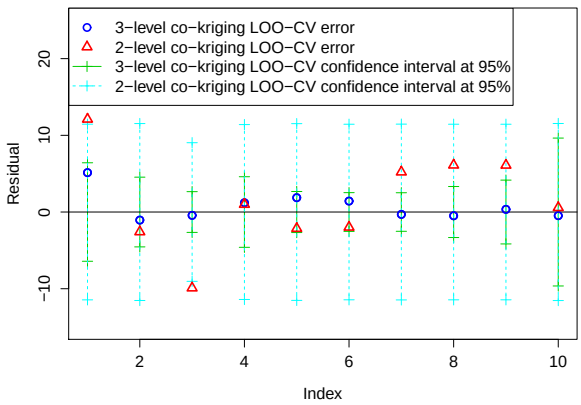


Figure 3.10: Leave-One-Out Cross-Validation predictive errors and variances of the 2-level and 3-level co-kriging. We see that the confidence intervals are accurate and the precision of the 3-level co-kriging is significantly better than the one of the 2-level co-kriging.

3.7.4 Comparison with existing methods

In this subsection we carry out a numerical comparison between the proposed model and the ones of [Kennedy and O'Hagan, 2000] and [Qian and Wu, 2008] on the Fluidized-Bed Process example. The comparison is performed similarly to the one presented in Subsection 3.6.5.

First we consider a 2-level co-kriging with $\mathbf{T}_{exp}$ as fine level and $\mathbf{T}_2$ as coarse level. For the coarse level we randomly extract 20 observations of $\mathbf{T}_2$ and for the fine level we randomly extract 10 observations of $\mathbf{T}_{exp}$ such that the experimental design set of $\mathbf{T}_{exp}$ is nested into the one of $\mathbf{T}_2$. The other 18 observations of $\mathbf{T}_{exp}$ are used as test sets. We have generated 100 different combinations of design and test sets for the numerical comparisons. The comparisons are also performed thanks to the R CRAN package "approximator" for the model of Kennedy and O'Hagan, to the WinBugs software to the one of Qian et al. and to the R CRAN package "MuFiCokriging" for the presented method. Like in Subsection 3.6.5, Gaussian covariance kernels and constant trends are chosen for all the Gaussian processes and constant adjustment coefficients are taken for the suggested model and the one of Kennedy and O'Hagan. Furthermore, for the Bayesian procedure presented by [Qian and Wu, 2008] we choose the following parameters for the prior distributions:

- $(\alpha_l, \gamma_l, \alpha_\rho, \gamma_\rho, \alpha_\delta, \gamma_\delta) = (2, 1, 2, 1, 2, 1)$,
- $u_l = 0$,
- $\nu_l = 1$,
- $(u_\rho, \nu_\rho, u_\delta, \nu_\delta) = (1, 1, 0, 1)$,
- $(a_l, b_l, a_\rho, b_\rho, a_\delta, b_\delta) = (2, 0.1, 2, 0.1, 2, 0.1)$

Like in Subsection 3.6.5 the convergence is reached after 50,000 burn-in iterations and 100,000 additional runs have been generated to compute the posterior distributions.

Figure 3.11 compares the RMSE of the three models evaluated on the 18 test points. We see in Figure 3.11 that the presented model is significantly better than the other ones. Furthermore, contrary to the comparison performed in Subsection 3.6.5, we see that the worst model is the one of Qian et al.. This is explained by the fact that, as mentioned in their article at the end of Section 2.4, the model suggested by Qian et al. supposed that the cheap code is known at a new point x. If it is not the case, they consider it equal to the prediction given by a Bayesian model on the cheap code. Nevertheless, in our example, we only have 20 observations in a 6-dimensional input space and the predictions of the cheap code are not good enough for the method of Qian et al..

Finally, we present in Figure 3.12 the computational costs of the three methods. As pointed out in Subsection 3.6.5, the suggested and the Kennedy and O'Hagan's models are significantly less computationally expensive than the one of Qian et al.. Nevertheless, contrary to the comparison in Subsection 3.6.5, the presented model and the one of Kennedy and O'Hagan are equivalent in terms of CPU times. This is due to the fact that the complexity reduction for the covariance matrix inversion does not bring significant differences when the number of observations is very small as in the Fluidized-Bed Process application.

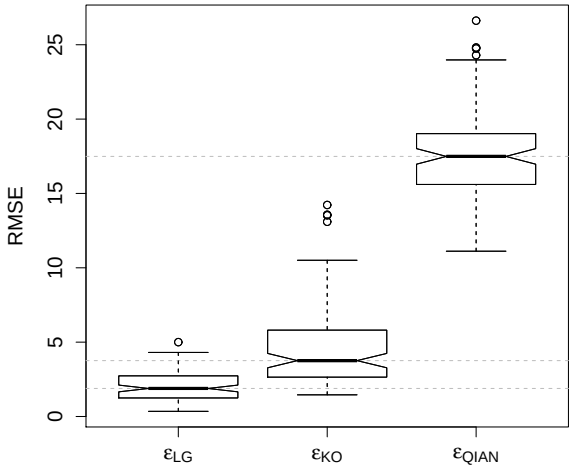


Figure 3.11: RMSEs of the presented model $\varepsilon_{\mathrm{LG}}$, the model of [Kennedy and O'Hagan, 2000] $\varepsilon_{\mathrm{KO}}$ (see [Kennedy and O'Hagan, 2000]) and the model of [Qian and Wu, 2008] $\varepsilon_{\mathrm{QIAN}}$ (see [Qian and Wu, 2008]). The numerical comparisons are performed on the Fluidized-Bed Process application with 100 different experimental and test sets.

3.8 Conclusion

We have presented a method for building kriging models using a hierarchy of codes with different levels of accuracy. This method allows us to improve a surrogate model built on a complex code using information from a cheap one. It is particularly useful when the complex code is very expensive. We see in our literature review that the first multi-level metamodel originally suggested is a first-order auto-regressive model built with Gaussian processes. The AR(1) relation between two levels of code is natural and the building of the model is straightforward. Nevertheless, we have highlighted some key issues which makes it difficult to use this model in practical ways.

First, important parameters of the model, which are the adjustment coefficients between two successive levels of codes, were numerically estimated. We propose here an analytical estimation of these parameters with a Bayesian method. This method allows us to have information about the uncertainties of the estimations and above all, to easily use the AR(1) model and its generalization to the case of non-spatial stationarity. Furthermore, a strength of the proposed method is that it even works for a code with more than 2 levels since its implementation is such that the estimations of the parameters of a s-level co-kriging is equivalent to the ones of s independent krigings in terms of numerical complexity. It is important to highlight that this method is based on a joint Bayesian analysis between the adjustment coefficient and the mean of the Gaussian process modeling the difference between two successive levels of code.

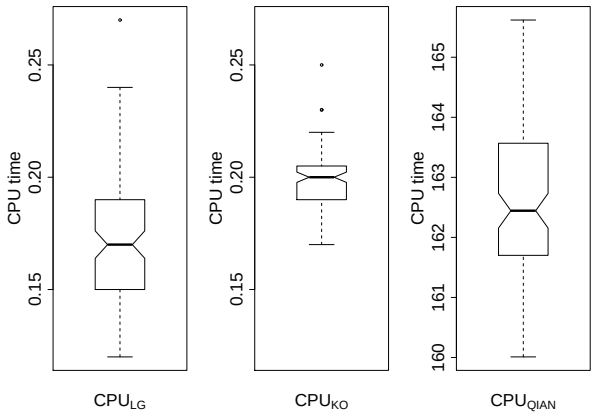


Figure 3.12: CPU times for the presented model $\mathrm{CPU_{LG}}$, the one of Kennedy and O'Hagan $\mathrm{CPU_{KO}}$ and the one of Qian et al. $\mathrm{CPU_{QIAN}}$. The numerical comparisons are performed on the Fluidized-Bed Process application with 100 different experimental and test sets. The ratio between $\mathrm{CPU_{QIAN}}$ and $\mathrm{CPU_{LG}}$ is around 1000 and $\mathrm{CPU_{LG}}$ and $\mathrm{CPU_{KO}}$ have the same order of magnitude.

Second, we have seen that the variance of the predictive distribution of the AR(1) model could be underestimated. A natural approach to improve this estimation is a Bayesian modeling. We propose here a Bayesian co-kriging for 2 levels of code and to avoid computationally expensive implementation, we suggest another model than the one presented. This new model is based on a hierarchical specification of the parameters of the model. This allows us to have a Bayesian model including only two nested integrations without Markov chain Monte Carlo procedure.

Finally, for a non-Bayesian s-level co-kriging, we have proved that building a s-level co-kriging is equivalent to build s independent krigings. This result is very important since it solves one of the most important key issues of the co-kriging which is the inversion of the covariance matrix. A 3-level co-kriging example has been provided to show the efficiency of the presented method.

Chapter 4

Multi-fidelity co-kriging model: recursive formulation

4.1 Introduction

We have developed in Chapter 3 a co-kriging based surrogate model for multi-fidelity computer codes. In fact, the first multi-fidelity model in a computer experiments framework has been proposed by [Craig et al., 1998] and is based on a linear regression formulation. Then this model is improved in [Cumming and Goldstein, 2009] by using a Bayes linear formulation. The reader is referred to [Goldstein and Wooff, 2007] for further detail about the Bayes linear approach. The methods suggested by [Craig et al., 1998] and [Cumming and Goldstein, 2009] have the strength to be relatively computationally cheap but as they are based on a linear regression formulation, they could suffer from a lack of accuracy. Another approach is to use the model of [Kennedy and O'Hagan, 2000] presented in Chapter 3. This method turns out to be very efficient and it has been applied and extended significantly.

The strength of the co-kriging model is that it gives very good predictive models but it is often computationally expensive, especially when the number of simulations is large. Furthermore, large data set can generate problems such as ill-conditioned covariance matrices. These problems are known for kriging but they become even more difficult for co-kriging since the total number of observations is the sum of the observations at all code levels.

In Chapter 3, we solve two mains issues of the model suggested by [Kennedy and O'Hagan, 2000] by proposing a complexity reduction for the inverse of the covariance matrices and by improving the estimation of the model parameters. Despite these improvements, it is hard to use this model to manage some problems such as sequential design (see Chapter 5) or sensitivity analysis (see Chapter 6). Indeed, for sequential design we wish to obtain the part of each code level on the predictive variance. This is not clear with the model of [Kennedy and O'Hagan, 2000]. Moreover, for sensitivity analysis we wish to finely infer from the model uncertainty about the one of the sensitivity indices. This problem is hard to address by using the model of [Kennedy and O'Hagan, 2000] since we are not able to generate samples from the predictive distribution incorporating the posterior distributions of the adjustment and

regression parameters.

To handle these problems, we adopt in this chapter a new approach for multi-fidelity surrogate modeling which uses a co-kriging model but with an original recursive formulation. An important property of this model is that it provides predictive mean and variance identical to the ones presented in [Kennedy and O'Hagan, 2000] and in Chapter 3. Therefore, it has the same efficiency of the model of [Kennedy and O'Hagan, 2000] in terms of prediction accuracy. However, our approach significantly reduces the complexity of the model presented in [Kennedy and O'Hagan, 2000] since it divides the whole set of simulations into groups of simulations corresponding to the ones of each level. Therefore, we will have s sub-matrices to invert which is less expensive and ill-conditioned than a large one. In fact, the computational complexity is equivalent to the one obtained in Chapter 3 Subsection 3.6.2 by using Equation (3.37) for the inverse of the covariance matrix. Therefore, we keep the advantages of the improvement presented in Chapter 3.

We will see in chapters 5 and 6 that the presented formulation allows for dealing effectively with sequential design and sensitivity analysis. Furthermore, a strength of our approach is that it allows to extend classical results of kriging to the considered co-kriging model. The two original results presented in this chapter are the following ones:

1. First, closed form expressions for the universal co-kriging predictive mean and variance are given (Section 4.3).
2. Second, the fast cross-validation method proposed in [Dubrule, 1983] is extended to the multi-fidelity co-kriging model (Section 4.4).

Finally, we illustrate these results in a complex hydrodynamic simulator (Section 4.5).

4.2 Multi-fidelity Gaussian process regression

In Subsection 4.2.1, we detail our recursive approach to build such a model. The recursive formulation of the multi-fidelity model is the first novelty of this chapter. We will see in the next sections that the new formulation allows us to find original results about the co-kriging model and to reduce its computational complexity.

4.2.1 Recursive multi-fidelity model

Let us suppose that we have s levels of code $(z_t(x))_{t=1,\dots,s}$ sorted by increasing order of fidelity and modeled by Gaussian processes $(Z_t(x))_{t=1,\dots,s}$, $x \in Q$. We still consider that $z_s(x)$ is the most accurate and costly code that we want to surrogate and $(z_t(x))_{t=1,\dots,s-1}$ are cheaper versions of it with $z_1(x)$ the less accurate one. Let us consider the following model for $t = 2, \dots, s$:

$$\begin{cases} Z_t(x) = \rho_{t-1}(x)\tilde{Z}_{t-1}(x) + \delta_t(x) \\ \tilde{Z}_{t-1}(x) \perp \delta_t(x) \\ \rho_{t-1}(x) = \mathbf{g}'_{t-1}(x)\boldsymbol{\beta}_{\rho_{t-1}} \end{cases}, \tag{4.1}$$

where $\tilde{Z}_{t-1}(x)$ is a Gaussian process with distribution $[Z_{t-1}(x)|\mathbf{Z}^{(t-1)} = \mathbf{z}^{(t-1)}, \boldsymbol{\beta}_{t-1}, \boldsymbol{\beta}_{\rho_{t-2}}, \sigma^2_{t-1}]$, $\delta_t(x)$ is a Gaussian process with mean $\mathbf{f}'_t(x)\boldsymbol{\beta}_t$ and covariance kernel $\sigma^2_t r_t(x, \tilde{x})$ and $\mathbf{D}_s \subseteq \mathbf{D}_{s-1} \subseteq \cdots \subseteq \mathbf{D}_1$.

Here, $\mathbf{g}_{t-1}(x)$ is a vector of q_{t-1} regression functions, $\mathbf{f}_t(x)$ is a vector of p_t regression functions, $\boldsymbol{\beta}_t$ is a p_t-dimensional vector, $\boldsymbol{\beta}_{\rho_{t-1}}$ is a q_{t-1}-dimensional vector, $\mathbf{Z}^{(s)} = (\mathbf{Z}'_1, \ldots, \mathbf{Z}'_s)'$ is the Gaussian vector containing the values of the random processes $(Z_t(x))_{t=1,\ldots,s}$ at the points in the experimental design sets $(\mathbf{D}_t)_{t=1,\ldots,s}$ and $\mathbf{z}^{(s)} = (\mathbf{z}'_1, \ldots, \mathbf{z}'_s)'$ the vector containing the values of $(z_t(x))_{t=1,\ldots,s}$ at the points in $(\mathbf{D}_t)_{t=1,\ldots,s}$.

The nested property of the experimental design sets is not necessary to build the model but it allows for a simple estimation of the model parameters. Since the codes are sorted in increasing order of fidelity it is not an unreasonable constraint for practical applications. Nonetheless, we present in Appendix B.1 the equations of the multi-fidelity co-kriging model when the experimental design sets are not nested.

The unique difference with the model presented in Chapter 3 is that we express $Z_t(x)$ (the Gaussian process modeling the response at level t) as a function of the Gaussian process $Z_{t-1}(x)$ conditioned by the values $\mathbf{z}^{(t-1)} = (\mathbf{z}_1, \ldots, \mathbf{z}_{t-1})$ at points in the experimental design sets $(\mathbf{D}_i)_{i=1,\ldots,t-1}$. The Gaussian processes $(\delta_t(x))_{t=2,\ldots,s}$ have the same definition as in Chapter 3 and we have for $t = 2, \ldots, s$ and for $x \in Q$:

$$\left[Z_t(x)|\mathbf{Z}^{(t)} = \mathbf{z}^{(t)}, \boldsymbol{\beta}_t, \boldsymbol{\beta}_{\rho_{t-1}}, \sigma^2_t\right] \sim \mathcal{N}\left(\mu_{Z_t}(x), s^2_{Z_t}(x)\right), \tag{4.2}$$

where:

$$\mu_{Z_t}(x) = \rho_{t-1}(x)\mu_{Z_{t-1}}(x) + \mathbf{f}'_t(x)\boldsymbol{\beta}_t + \mathbf{r}'_t(x)\mathbf{R}_t^{-1}\left(\mathbf{z}_t - \rho_{t-1}(\mathbf{D}_t) \odot z_{t-1}(\mathbf{D}_t) - \mathbf{F}_t\boldsymbol{\beta}_t\right) \tag{4.3}$$

and:

$$\sigma^2_{Z_t}(x) = \rho^2_{t-1}(x)\sigma^2_{Z_{t-1}}(x) + \sigma^2_t\left(1 - \mathbf{r}'_t(x)\mathbf{R}_t^{-1}\mathbf{r}_t(x)\right). \tag{4.4}$$

The notation $\odot$ represents the element by element matrix product. $\mathbf{R}_t$ is the correlation matrix $\mathbf{R}_t = (\mathbf{r}_t(x, \tilde{x}))_{x,\tilde{x}\in\mathbf{D}_t}$ and $\mathbf{r}'_t(x)$ is the correlation vector $\mathbf{r}'_t(x) = (\mathbf{r}_t(x, \tilde{x}))_{\tilde{x}\in\mathbf{D}_t}$. We denote by $\rho_t(\mathbf{D}_{t-1})$ the vector containing the values of $\rho_t(x)$ for $x \in \mathbf{D}_{t-1}$, $z_t(\mathbf{D}_{t-1})$ the vector containing the known values of $Z_t(x)$ at points in $\mathbf{D}_{t-1}$ and $\mathbf{F}_t$ is the experience matrix containing the values of $\mathbf{f}_t(x)'$ on $\mathbf{D}_t$.

The mean $\mu_{Z_t}(x)$ is the surrogate model of the response at level t, $1 \leq t \leq s$, taking into account the known values of the t first levels of responses $(\mathbf{z}_i)_{i=1,\ldots,t}$ and the variance $\sigma^2_{Z_t}(x)$ represents the mean squared error of this model. The mean and the variance of the Gaussian process regression at level t being expressed in function of the ones of level $t-1$, we have a recursive multi-fidelity metamodel. Furthermore, in this new formulation, it is clearly emphasized that the mean of the predictive distribution does not depend on the variance parameters $(\sigma^2_t)_{t=1,\ldots,s}$. This is a classical result of kriging which states that for covariance kernels of the form $k(x, \tilde{x}) = \sigma^2 r(x, \tilde{x})$, the mean of the kriging model is independent of σ^2.

An important strength of the recursive formulation is that contrary to the formulation suggested in [Kennedy and O'Hagan, 2000] and in Chapter 3, once the multi-fidelity model is built, it provides the surrogate models of all the responses $(z_t(x))_{t=1\ldots,s}$.

We have the following proposition. We note that we consider here an adjustment coefficient depending on x. The reader is referred to Appendix A.2 for the details about the predictive mean and variance of the model presented in Chapter 3.

Proposition 4.1. *Let us consider s Gaussian processes $(Z_t(x))_{t=1,\ldots,s}$ and $\mathbf{Z}^{(s)} = (\mathbf{Z}_t)_{t=1,\ldots,s}$ the Gaussian vector containing the values of $(Z_t(x))_{t=1,\ldots,s}$ at points in $(\mathbf{D}_t)_{t=1,\ldots,s}$ with $\mathbf{D}_s \subseteq \mathbf{D}_{s-1} \subseteq \cdots \subseteq \mathbf{D}_1$. If we consider the mean $m_{Z_s}(x)$ (3.27) and the variance $s^2_{Z_s}(x)$ (3.28) induced by the model presented in Chapter 3 and the mean $\mu_{Z_s}(x)$ (4.3) and the variance $\sigma^2_{Z_s}(x)$ (4.4) induced by the model (4.1) when we condition the Gaussian process $Z_s(x)$ by the known values $\mathbf{z}^{(s)}$ of $\mathbf{Z}^{(s)}$ and by the parameters $\boldsymbol{\beta}$, $\boldsymbol{\beta}_\rho$ and σ^2, then, we have:*

$$\begin{aligned} \mu_{Z_s}(x) &= m_{Z_s}(x), \\ \sigma^2_{Z_s}(x) &= s^2_{Z_s}(x). \end{aligned}$$

Proof. Let us consider the co-kriging mean of the model presented in Chapter 3 for a t-level co-kriging with $t = 2, \ldots, s$ and $\rho_{t-1}(x) = \mathbf{g}'_{t-1}(x)\boldsymbol{\beta}_{\rho_{t-1}}$:

$$m_{Z_t}(x) = \mathbf{h}'_t(x)\boldsymbol{\beta}^{(t)} + \mathbf{k}'_t(x)\mathbf{V}_t^{-1}(\mathbf{z}^{(t)} - \mathbf{H}_t\boldsymbol{\beta}^{(t)}),$$

where $\boldsymbol{\beta}^{(t)} = (\boldsymbol{\beta}'_1, \ldots, \boldsymbol{\beta}'_t)'$, $\mathbf{z}^{(t)} = (\mathbf{z}'_1, \ldots, \mathbf{z}'_t)'$, $\mathbf{H}_t$ is defined in Equation (3.33) and $\mathbf{h}'_t(x)$ is defined in the following equation:

$$\mathbf{h}'_t(x) = \left(\left(\prod_{i=1}^{t-1} \rho_i(x) \right) \mathbf{f}'_1(x), \left(\prod_{i=2}^{t-1} \rho_i(x) \right) \mathbf{f}'_2(x), \ldots, \rho_{t-1}(x)\mathbf{f}'_{t-1}(x), \mathbf{f}'_t(x) \right). \tag{4.5}$$

We have:

$$\begin{aligned} \mathbf{h}'_t(x)\boldsymbol{\beta}^{(t)} &= \rho_{t-1}(x) \left(\left(\prod_{i=1}^{t-2} \rho_i(x) \right) \mathbf{f}'_1(x), \left(\prod_{i=2}^{t-2} \rho_i(x) \right) \mathbf{f}'_2(x), \ldots, \mathbf{f}'_{t-1}(x) \right) \boldsymbol{\beta}^{(t-1)} + \mathbf{f}'_t(x)\boldsymbol{\beta}_t \\ &= \rho_{t-1}(x)\mathbf{h}'_{t-1}(x)\boldsymbol{\beta}^{(t-1)} + \mathbf{f}'_t(x)\boldsymbol{\beta}_t. \end{aligned}$$

Then, from equations:

$$\mathrm{cov}(Z_t(x), Z_{\tilde{t}}(\tilde{x})|\sigma^2, \boldsymbol{\beta}, \boldsymbol{\beta}_\rho) = \left(\prod_{i=t'}^{t-1} \rho_i(x) \right) \mathrm{cov}(Z_{\tilde{t}}(x), Z_{\tilde{t}}(\tilde{x})|\sigma^2, \boldsymbol{\beta}, \boldsymbol{\beta}_\rho) \tag{4.6}$$

and:

$$\mathrm{cov}(Z_t(x), Z_t(\tilde{x})|\sigma^2, \boldsymbol{\beta}, \boldsymbol{\beta}_\rho) = \sum_{j=1}^{t} \sigma_j^2 \left(\prod_{i=j}^{t-1} \rho_i(x)\rho_i(\tilde{x}) \right) r_j(x, \tilde{x}), \tag{4.7}$$

with $t > \tilde{t}$, we have the following equality:

$$\begin{aligned}\mathbf{k}_t'(x)\mathbf{V}_t^{-1}\mathbf{z}^{(t)} &= \rho_{t-1}(x)\mathbf{k}_{t-1}'(x)\mathbf{V}_{t-1}^{-1}\mathbf{z}^{(t-1)} - \left(\rho_{t-1}'(\mathbf{D}_t)\right) \odot \left(\mathbf{r}_t'(x)\mathbf{R}_t^{-1}z_{t-1}(\mathbf{D}_t)\right) \\ &\quad + \mathbf{r}_t'(x)\mathbf{R}_t^{-1}\mathbf{z}_t\end{aligned}$$

and with Equation (4.5):

$$\mathbf{k}_t'(x)\mathbf{V}_t^{-1}\mathbf{H}_t\boldsymbol{\beta}^{(t)} = \rho_{t-1}(x)\mathbf{k}_{t-1}'(x)\mathbf{V}_{t-1}^{-1}\mathbf{H}_{t-1}\boldsymbol{\beta}^{(t-1)} + \mathbf{r}_t'(x)\mathbf{R}_t^{-1}\mathbf{F}_t(\mathbf{D}_t)\boldsymbol{\beta}_t,$$

where $\odot$ stands for the element by element matrix product. We hence obtain the recursive relation:

$$m_{Z_t}(x) = \rho_{t-1}(x)m_{Z_{t-1}}(x) + \mathbf{f}_t'(x)\boldsymbol{\beta}_t + \mathbf{r}_t'(x)\mathbf{R}_t^{-1}\left[\mathbf{z}_t - \rho_{t-1}(\mathbf{D}_t) \odot z_{t-1}(\mathbf{D}_t) - \mathbf{F}_t(\mathbf{D}_t)\boldsymbol{\beta}_t\right].$$

The co-kriging mean of the model (4.1) satisfies the same recursive relation and we have $m_{Z_1}(x) = \mu_{Z_1}(x)$. This proves the first equality of Proposition 4.1:

$$\mu_{Z_s}(x) = m_{Z_s}(x).$$

We follow the same guideline for the co-kriging covariance:

$$s_{Z_t}^2(x,\tilde{x}) = v_{Z_t}^2(x,\tilde{x}) - \mathbf{k}_t'(x)\mathbf{V}_t^{-1}\mathbf{k}_t(\tilde{x}),$$

where $v_{Z_t}^2(x,\tilde{x})$ is the covariance between $Z_t(x)$ and $Z_t(\tilde{x})$ and $s_{Z_t}^2(x,\tilde{x})$ is the covariance function of the conditioned Gaussian process $[Z_t(x)|\mathbf{Z}^{(t)} = \mathbf{z}^{(t)}, \boldsymbol{\beta}, \boldsymbol{\beta}_\rho, \sigma^2]$ for the model presented in Chapter 3. From Equation (4.7), we can deduce the following equality:

$$\sigma_{Z_t}^2(x,\tilde{x}) = \rho_{t-1}(x)\rho_{t-1}(\tilde{x})v_{Z_{t-1}}^2(x,\tilde{x}) + v_t^2(x,\tilde{x}),$$

where $\sigma_{Z_t}^2(x,\tilde{x})$ is the covariance function of the conditioned Gaussian process $[Z_t(x)|\mathbf{Z}^{(t)} = \mathbf{z}^{(t)}, \boldsymbol{\beta}_t, \boldsymbol{\beta}_{\rho_{t-1}}, \sigma_t^2]$ of the recursive model (4.1). Then, from equations (4.6) and (4.7), we have:

$$\mathbf{k}_t'(x)\mathbf{V}_t^{-1}\mathbf{k}_t(\tilde{x}) = \rho_{t-1}(x)\rho_{t-1}(\tilde{x})\mathbf{k}_{t-1}'(x)\mathbf{V}_{t-1}^{-1}\mathbf{k}_{t-1}(\tilde{x}) + \sigma_t^2\mathbf{r}_t'(x)\mathbf{R}_t^{-1}\mathbf{r}_t(\tilde{x}).$$

Finally we can deduce the following equality:

$$s_{Z_t}^2(x,\tilde{x}) = \rho_{t-1}(x)\rho_{t-1}(\tilde{x})\left(v_{Z_{t-1}}^2(x,\tilde{x}) - \mathbf{k}_{t-1}'(x)\mathbf{V}_{t-1}^{-1}\mathbf{k}_{t-1}(\tilde{x})\right) + \sigma_t^2\left(1 - \mathbf{r}_t'(x)\mathbf{R}_t^{-1}\mathbf{r}_t(\tilde{x})\right),$$

which is equivalent to:

$$s_{Z_t}^2(x,\tilde{x}) = \rho_{t-1}(x)\rho_{t-1}(\tilde{x})s_{Z_{t-1}}^2(x,\tilde{x}) + \sigma_t^2\left(1 - \mathbf{r}_t'(x)\mathbf{R}_t^{-1}\mathbf{r}_t(\tilde{x})\right).$$

This is the same recursive relation as the one satisfies by the co-kriging covariance $\sigma_{Z_t}^2(x,\tilde{x})$ of the model (4.1) (see Equation (4.4)). Since $s_{Z_1}^2(x,\tilde{x}) = \sigma_{Z_1}^2(x,\tilde{x})$, we have :

$$\sigma_{Z_s}^2(x,\tilde{x}) = s_{Z_s}^2(x,\tilde{x}).$$

This equality with $x = \tilde{x}$ proves the second equality of Proposition 4.1. □

Proposition 4.1 shows that the model presented in [Kennedy and O'Hagan, 2000] and the recursive model (4.1) have the same predictive Gaussian distribution. Our objective in the next sections is to show that the new formulation (4.1) has several advantages compared to the one of [Kennedy and O'Hagan, 2000]. First, its computational complexity is lower (Section 4.2.2); second, it provides closed form expressions for the universal co-kriging mean and variance contrarily to [Kennedy and O'Hagan, 2000] (Section 4.3); third, it makes it possible to implement a fast cross-validation procedure (Section 4.4).

4.2.2 Complexity analysis

The computational cost is dominated by the inversion of the covariance matrices. In the original approach proposed in [Kennedy and O'Hagan, 2000] one has to invert the matrix $\mathbf{V}_s$ of size $\sum_{i=1}^{s} n_i \times \sum_{i=1}^{s} n_i$.

Our recursive formulation shows that building a s-level co-kriging is equivalent in terms of numerical complexity to build s independent krigings. This implies a reduction of the model complexity. Indeed, the inversion of s matrices $(\mathbf{R}_t)_{t=1,\dots,s}$ of size $(n_t \times n_t)_{t=1,\dots,s}$ where n_t corresponds to the size of the vector $\mathbf{z}_t$ at level $t = 1, \dots, s$ is less expensive than the inversion of the matrix $\mathbf{V}_s$ of size $\sum_{i=1}^{s} n_i \times \sum_{i=1}^{s} n_i$. We also reduce the memory cost since storing the s matrices $(\mathbf{R}_t)_{t=1,\dots,s}$ requires less memory than storing the matrix $\mathbf{V}_s$. The computational cost is thus equivalent to the one obtained with the results given in Chapter 3 Subsection 3.6.2.

> We note that the model with this formulation is more interpretable since we can deduce the impact of each level of response into the model error through $(\sigma^2_{Z_t}(x))_{t=1,\dots,s}$.

4.2.3 Parameter estimation

We present in this section a Bayesian estimation of the parameter $\boldsymbol{\psi} = (\boldsymbol{\beta}, \boldsymbol{\beta}_\rho, \sigma^2)$ focusing on conjugate and non-informative distributions for the priors. This allows us to obtain closed form expressions for the posterior distributions of the parameters. Furthermore, from the non-informative case, we can obtain the estimates given by a maximum likelihood method. The presented formulas can hence be used in a frequentist approach. We note that the recursive formulation and the nested property of the experimental designs allow for separating the estimations of the parameters $(\boldsymbol{\beta}_t, \boldsymbol{\beta}_{\rho_{t-1}}, \sigma_t^2)_{t=1,\dots,s}$ and $(\boldsymbol{\beta}_1, \sigma_1^2)$.

Like in Chapter 3 Section 3.4, we address two cases in this section

Case (**i**): all the priors are informative

Case (**ii**): all the priors are non-informative

It is of course be possible to address the case of a mixture of informative and non-informative priors. For the non-informative case (ii), we use the "Jeffreys priors" [Jeffreys, 1961]:

$$p(\boldsymbol{\beta}_1|\sigma_1^2) \propto 1, \quad p(\sigma_1^2) \propto \frac{1}{\sigma_1^2}, \quad p(\boldsymbol{\beta}_{\rho_{t-1}}, \boldsymbol{\beta}_t|\mathbf{z}^{(t-1)}, \sigma_t^2) \propto 1, \quad p(\sigma_t^2|\mathbf{z}^{(t-1)}) \propto \frac{1}{\sigma_t^2}, \tag{4.8}$$

where $t = 2, \dots, s$. For the informative case (i), we consider the same conjugate prior distributions as in Chapter 3 Section 3.4:

$$[\boldsymbol{\beta}_1|\sigma_1^2] \sim \mathcal{N}_{p_1}(\mathbf{b}_1, \sigma_1^2\mathbf{V}_1),$$

$$[\boldsymbol{\beta}_{\rho_{t-1}}, \boldsymbol{\beta}_t|\mathbf{z}^{(t-1)}, \sigma_t^2] \sim \mathcal{N}_{q_{t-1}+p_t}\left(\mathbf{b}_t = \begin{pmatrix} \mathbf{b}_{t-1}^{\rho} \\ \mathbf{b}_t^{\boldsymbol{\beta}} \end{pmatrix}, \sigma_t^2\mathbf{V}_t = \sigma_t^2 \begin{pmatrix} \mathbf{V}_{t-1}^{\rho} & 0 \\ 0 & \mathbf{V}_t^{\boldsymbol{\beta}} \end{pmatrix}\right),$$

$$[\sigma_1^2] \sim \mathcal{IG}(\alpha_1, \gamma_1), \qquad [\sigma_t^2|\mathbf{z}^{(t-1)}] \sim \mathcal{IG}(\alpha_t, \gamma_t),$$

with $\mathbf{b}_1$ a vector a size p_1, $\mathbf{b}_{t-1}^{\rho}$ a vector of size q_{t-1}, $\mathbf{b}_t^{\boldsymbol{\beta}}$ a vector of size p_t, $\mathbf{V}_1$ a $p_1 \times p_1$ matrix, $\mathbf{V}_{t-1}^{\rho}$ a $q_{t-1} \times q_{t-1}$ matrix, $\mathbf{V}_t^{\boldsymbol{\beta}}$ a $p_t \times p_t$ matrix, $\alpha_1, \gamma_1, \alpha_t, \gamma_t > 0$ and $\mathcal{IG}$ stands for the inverse Gamma distribution. The posterior distributions are then as follows. We have:

$$[\boldsymbol{\beta}_1|\mathbf{z}_1, \sigma_1^2] \sim \mathcal{N}_{p_1}(\boldsymbol{\Sigma}_1\boldsymbol{\nu}_1, \boldsymbol{\Sigma}_1) \qquad [\boldsymbol{\beta}_{\rho_{t-1}}, \boldsymbol{\beta}_t|\mathbf{z}^{(t)}, \sigma_t^2] \sim \mathcal{N}_{q_{t-1}+q_t}(\boldsymbol{\Sigma}_t\boldsymbol{\nu}_t, \boldsymbol{\Sigma}_t), \tag{4.9}$$

where, for $t \geq 1$:

$$\boldsymbol{\Sigma}_t = \begin{cases} [\mathbf{H}_t' \frac{\mathbf{R}_t^{-1}}{\sigma_2^2}\mathbf{H}_t + \frac{\mathbf{V}_t^{-1}}{\sigma_2^2}]^{-1} & \textbf{(i)} \\ [\mathbf{H}_t' \frac{\mathbf{R}_t^{-1}}{\sigma_2^2}\mathbf{H}_t]^{-1} & \textbf{(ii)} \end{cases} \quad \boldsymbol{\nu}_t = \begin{cases} [\mathbf{H}_t' \frac{\mathbf{R}_t^{-1}}{\sigma_2^2}\mathbf{z}_t + \frac{\mathbf{V}_t^{-1}}{\sigma_2^2}\mathbf{b}_t] & \textbf{(i)} \\ [\mathbf{H}_t' \frac{\mathbf{R}_t^{-1}}{\sigma_2^2}\mathbf{z}_t] & \textbf{(ii)} \end{cases}, \tag{4.10}$$

with $\mathbf{H}_1 = \mathbf{F}_1$ and for $t > 1$, $\mathbf{H}_t = [\mathbf{G}_{t-1} \odot (z_{t-1}(\mathbf{D}_t)\mathbf{1}_{q_{t-1}}') \quad \mathbf{F}_t]$ where

$$\mathbf{G}_{t-1},$$

is the experience matrix containing the values of $\mathbf{g}_{t-1}(x)'$ in $\mathbf{D}_t$ and $\mathbf{1}_{q_{t-1}}'$ is a q_{t-1}-vector of ones. Furthermore, we have for $t \geq 1$:

$$[\sigma_t^2|\mathbf{z}^{(t)}] \sim \mathcal{IG}(a_t, \frac{Q_t}{2}), \tag{4.11}$$

where:

$$Q_t = \begin{cases} 2\gamma_t + (\mathbf{b}_t - \tilde{\boldsymbol{\lambda}}_t)'(\mathbf{V}_t + [\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{H}_t]^{-1})^{-1}(\mathbf{b}_t - \tilde{\boldsymbol{\lambda}}_t) + \tilde{Q}_t & \textbf{(i)} \\ \tilde{Q}_t & \textbf{(ii)} \end{cases},$$

with $\tilde{Q}_t = (\mathbf{z}_t - \mathbf{H}_t\tilde{\boldsymbol{\lambda}}_t)'\mathbf{R}_t^{-1}(\mathbf{z}_t - \mathbf{H}_t\tilde{\boldsymbol{\lambda}}_t)$, $\tilde{\boldsymbol{\lambda}}_t = (\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{H}_t)^{-1}\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{z}_t$ and :

$$a_t = \begin{cases} \frac{n_t}{2} + \alpha_t & \textbf{(i)} \\ \frac{n_t - p_t - q_{t-1}}{2} & \textbf{(ii)} \end{cases},$$

with the convention $q_0 = 0$.

We highlight that the maximum likelihood estimators for the parameters $\boldsymbol{\beta}_1$ and $(\boldsymbol{\beta}_{\rho_{t-1}}, \boldsymbol{\beta}_t)$ are given by the means of their posterior distributions in the non-informative case. Furthermore, the restricted maximum likelihood estimate of the variance parameter σ_t^2 can also be deduced from its posterior distribution in the non-informative case and is given by $\hat{\sigma}_{t,\text{REML}}^2 = \frac{Q_t}{2a_t}$. Finally, we see that the parameter posterior distributions for the recursive model are identical to the ones presented in Chapter 3 Section 3.4. This strengthen the relation between the two models. However, we will see in the remainder of this chapter and in the following chapters that the recursive model bring significant advantages compared to the one presented in Chapter 3.

4.3 Universal co-kriging model

We can see in Equation (4.2) that the predictive distribution of $Z_s(x)$ is conditioned by the observations $\mathbf{z}^{(s)}$ and the parameters $\boldsymbol{\beta}$, $\boldsymbol{\beta}_\rho$ and σ^2. The objective of a Bayesian prediction

is to integrate the parameter posterior distributions into the predictive distribution. Indeed, in the previous subsection, we have expressed the posterior distributions of the variance parameters $(\sigma_t^2)_{t=1,\dots,s}$ conditionally to the observations and the posterior distributions of the trend parameters $\boldsymbol{\beta}_1$ and $(\boldsymbol{\beta}_{\rho_{t-1}}, \boldsymbol{\beta}_t)_{t=2,\dots,s}$ conditionally to the observations and the variance parameters. Thus, using the Bayes formula, we can easily obtain a predictive distribution only conditioned by the observations by integrating into it the posterior distributions of the parameters as presented in Chapter 3 Section 3.4.

As a result of this integration, the predictive distribution is not Gaussian. In particular, we cannot have a closed form expression for the predictive distribution. However, it is possible to obtain closed form expressions for the posterior mean $\mathbb{E}[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}]$ and variance $\mathrm{Var}(Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)})$.

The following proposition giving the closed form expressions of the posterior mean and variance of the predictive distribution only conditioned by the observations is a novelty. The proof of this proposition is based on the recursive formulation which emphasizes the strength of this new approach. Indeed, it does not seem possible to obtain this result by considering directly the model suggested in [Kennedy and O'Hagan, 2000].

Proposition 4.2. *Let us consider s Gaussian processes $(Z_t(x))_{t=1,\dots,s}$ and $\mathbf{Z}^{(s)} = (\mathbf{Z}_t)_{t=1,\dots,s}$ the Gaussian vector containing the values of $(Z_t(x))_{t=1,\dots,s}$ at points in $(\mathbf{D}_t)_{t=1,\dots,s}$ with $\mathbf{D}_s \subseteq \mathbf{D}_{s-1} \subseteq \cdots \subseteq \mathbf{D}_1$. If we consider the conditional predictive distribution in Equation (4.2) and the posterior distributions of the parameters given in equations (4.9) and (4.11), then we have for $t = 1, \dots, s$:*

$$\mathbb{E}[Z_t(x)|\mathbf{Z}^{(t)} = \mathbf{z}^{(t)}] = \mathbf{h}_t'(x)\boldsymbol{\Sigma}_t\boldsymbol{\nu}_t + \mathbf{r}_t'(x)\mathbf{R}_t^{-1}\left(\mathbf{z}_t - \mathbf{H}_t\boldsymbol{\Sigma}_t\boldsymbol{\nu}_t\right), \tag{4.12}$$

with $\mathbf{h}_1' = \mathbf{f}_1'$, $\mathbf{H}_1 = \mathbf{F}_1$ and for $t > 1$, $\mathbf{h}_t'(x) = \left(\mathbf{g}_{t-1}(x)'\mathbb{E}[Z_{t-1}(x)|\mathbf{Z}_{t-1} = \mathbf{z}_{t-1}] \quad \mathbf{f}_t'(x)\right)$ and $\mathbf{H}_t = [\mathbf{G}_{t-1} \odot (z_{t-1}(\mathbf{D}_t)\mathbf{1}_{q_{t-1}}') \quad \mathbf{F}_t]$. Furthermore, we have:

$$\begin{aligned}\mathrm{Var}(Z_t(x)|\mathbf{Z}^{(t)} = \mathbf{z}^{(t)}) \quad = \quad & \hat{\sigma}^2_{\rho_{t-1}}(x)\mathrm{Var}(Z_{t-1}(x)|\mathbf{Z}^{(t-1)} = \mathbf{z}^{(t-1)}) \\ & + \tfrac{Q_t}{2(a_t-1)}\left(1 - \mathbf{r}_t'(x)\mathbf{R}_t^{-1}\mathbf{r}_t'(x)\right) \\ & + \left(\mathbf{h}_t' - \mathbf{r}_t'(x)\mathbf{R}_t^{-1}\mathbf{H}_t\right)\boldsymbol{\Sigma}_t\left(\mathbf{h}_t' - \mathbf{r}_t'(x)\mathbf{R}_t^{-1}\mathbf{H}_t\right)'\end{aligned}, \tag{4.13}$$

with $\hat{\sigma}^2_{\rho_{t-1}}(x) = \mathbf{g}_{t-1}(x)'\left([\boldsymbol{\Sigma}_t]_{[1,\dots,q_{t-1},1,\dots,q_{t-1}]} + [\boldsymbol{\Sigma}_t\boldsymbol{\nu}_t]_{1,\dots,q_{t-1}}[\boldsymbol{\Sigma}_t\boldsymbol{\nu}_t]'_{1,\dots,q_{t-1}}\right)\mathbf{g}_{t-1}(x)$.

Proof. Noting that the mean of the predictive distribution in Equation (4.2) does not depend on σ_t^2 and thanks to the law of total expectation, we have the following equality:

$$\mathbb{E}\left[Z_t(x)|\mathbf{Z}^{(t)} = \mathbf{z}^{(t)}\right] = \mathbb{E}\left[\mathbb{E}\left[Z_t(x)|\mathbf{Z}^{(t)} = \mathbf{z}^{(t)}, \sigma_t^2, \boldsymbol{\beta}_t, \boldsymbol{\beta}_{\rho_{t-1}}\right]\middle|\mathbf{Z}^{(t)} = \mathbf{z}^{(t)}\right].$$

From the equations (4.3) and (4.9), we directly deduce Equation (4.12). Then, we have the following equality:

$$\mathrm{var}\left(\mu_{Z_t}(x)\middle|\mathbf{z}^{(t)}, \sigma_t^2\right) = (\mathbf{h}_t'(x) - \mathbf{r}_t(x)'\mathbf{R}_t^{-1}\mathbf{H}_t)\boldsymbol{\Sigma}_t(\mathbf{h}_t'(x) - \mathbf{r}_t(x)'\mathbf{R}_t^{-1}\mathbf{H}_t)'. \tag{4.14}$$

The law of total variance states that:

$$\begin{aligned}\mathrm{var}(Z_t(x)|\mathbf{z}^{(t)},\sigma_t^2) &= \mathbb{E}\left[\mathrm{var}(Z_t(x)|\mathbf{z}^{(t)},\boldsymbol{\beta}_t,\boldsymbol{\beta}_{\rho_{t-1}},\sigma_t^2)\left|\mathbf{z}^{(t)},\sigma_t^2\right.\right] \\ &+ \mathrm{var}\left(\mathbb{E}\left[Z_t(x)|\mathbf{z}^{(t)},\boldsymbol{\beta}_t,\boldsymbol{\beta}_{\rho_{t-1}},\sigma_t^2\right]\left|\mathbf{z}^{(t)},\sigma_t^2\right.\right).\end{aligned}$$

Thus, from equations (4.3), (4.12) and (4.14), we obtain:

$$\begin{aligned}\mathrm{var}(Z_t(x)|\mathbf{Z}^{(t)}=\mathbf{z}^{(t)},\sigma_t^2) &= \hat{\sigma}^2_{\rho_{t-1}}(x)\mathrm{var}(Z_{t-1}(x)|\mathbf{Z}^{(t-1)}=\mathbf{z}^{(t-1)},\sigma_t^2)+\sigma_t^2\left(1-\mathbf{r}_t'(x)\mathbf{R}_t^{-1}\mathbf{r}_t'(x)\right) \\ &+\left(h_t'-\mathbf{r}_t'(x)\mathbf{R}_t^{-1}\mathbf{H}_t\right)\boldsymbol{\Sigma}_t\left(h_t'-\mathbf{r}_t'(x)\mathbf{R}_t^{-1}\mathbf{H}_t\right)'\end{aligned}. \tag{4.15}$$

Again using the law of total variance and the independence between $\mathbb{E}\left[Z_t(x)|\mathbf{Z}^{(t)}=\mathbf{z}^{(t)},\boldsymbol{\beta}_t,\boldsymbol{\beta}_{\rho_{t-1}}\right]$ and σ_t^2, we have:

$$\mathrm{var}(Z_t(x)|\mathbf{z}^{(t)})=\mathbb{E}\left[\mathrm{var}(Z_t(x))|\mathbf{z}^{(t)},\sigma_t^2\right]. \tag{4.16}$$

We obtain Equation (4.13) from Equation (4.11) by noting that the mean of an inverse Gamma distribution $\mathcal{IG}(a,b)$ is $b/(a-1)$. □

We note that, in the mean of the predictive distribution, the parameters have been replaced by their posterior means. Furthermore, in the variance of the predictive distribution, the variance parameter has been replaced by its posterior mean and the term $\left(\mathbf{h}_t'-\mathbf{r}_t'(x)\mathbf{R}_t^{-1}\mathbf{H}_t\right)\boldsymbol{\Sigma}_t\left(\mathbf{h}_t'-\mathbf{r}_t'(x)\mathbf{R}_t^{-1}\mathbf{H}_t\right)'$ has been added. It represents the uncertainty due to the estimation of the regression parameters (including the adjustment coefficient). We call these formulas the universal co-kriging equations due to their similarities with the universal kriging equations (they are identical for $s=1$).

> An important difference between the universal kriging predictive variance and the universal multi-fidelity co-kriging one is that the latter depends on the observations. Therefore, the classical methods based on the predictive variance (e.g. sequential design strategies) are not easy. We address this question in Chapter 6.

4.4 Fast cross-validation for co-kriging surrogate models

The idea of a cross-validation procedure is to split the experimental design set into two disjoint sets, one is used for training and the other one is used to monitor the performance of the surrogate model. The idea is that the performance on the test set can be used as a proxy for the generalization error. A particular case of this method is the Leave-One-Out Cross-Validation (noted LOO-CV) where n test sets are obtained by removing one observation at a time. This procedure can be time-consuming for a kriging model but it is shown in [Dubrule, 1983], [Rasmussen and Williams, 2006], [Zhang and Wang, 2009] and Chapter 1 Subsection 1.3.3 that there are computational shortcuts. Our recursive formulation allows us to extend these ideas to co-kriging models (which is not possible with the original formulation in [Kennedy and O'Hagan, 2000]). Furthermore, the cross-validation equations proposed in

this section extend the previous ones even for $s = 1$ (i.e. the classical kriging model) since they do not suppose that the regression and the variance coefficients are known. Therefore, those parameters are re-estimated for each training set. We note that the re-estimation of the variance coefficient is a novelty which is important since fixing this parameter can lead to huge errors for the estimate of the cross-validation predictive variance when the number of observations is small or when the number of points in the test set is important.

If we denote by ξ_s the set of indices of the n_{test} points in $\mathbf{D}_s$ constituting the test set $\mathbf{D}_{test}$ and ξ_t, $1 \leq t < s$, the corresponding set of indices in $\mathbf{D}_t$ - indeed, we have $\mathbf{D}_s \subset \mathbf{D}_{s-1} \subset \cdots \subset \mathbf{D}_1$, therefore $\mathbf{D}_{test} \subset \mathbf{D}_t$. The nested experimental design assumption implies that, in the cross-validation procedure, if we remove a set of points from $\mathbf{D}_s$ we can also remove it from $\mathbf{D}_t$, $1 \leq t \leq s$.

The following proposition gives the vectors of the cross-validation predictive errors and variances at points in the test set $\mathbf{D}_{test}$ when we remove them from the t highest levels of code. In the proposition, we consider that we are in the non-informative case for the parameter posterior distributions (see Section 4.2.3) but it can be easily extended to the informative case presented in Section 4.2.3. We note that this result presented for the first time to a multi-fidelity co-kriging model can be obtained thanks to the recursive formulation.

Notations: If $\boldsymbol{\xi}$ is a set of indices, then $\mathbf{A}_{[\boldsymbol{\xi},\boldsymbol{\xi}]}$ is the sub-matrix of elements $\boldsymbol{\xi} \times \boldsymbol{\xi}$ of $\mathbf{A}$, $\mathbf{a}_{[\boldsymbol{\xi}]}$ is the sub-vector of elements $\boldsymbol{\xi}$ of $\mathbf{a}$, $\mathbf{B}_{[-\boldsymbol{\xi}]}$ represents the matrix $\mathbf{B}$ in which we remove the rows of index $\boldsymbol{\xi}$, $\mathbf{C}_{[-\boldsymbol{\xi},-\boldsymbol{\xi}]}$ is the sub-matrix of $\mathbf{C}$ in which we remove the rows and columns of index $\boldsymbol{\xi}$ and $\mathbf{C}_{[-\boldsymbol{\xi},\boldsymbol{\xi}]}$ is the sub-matrix of $\mathbf{C}$ in which we remove the rows of index $\boldsymbol{\xi}$ and keep only the columns of index $\boldsymbol{\xi}$.

Proposition 4.3. *Let us consider s Gaussian processes $(Z_t(x))_{t=1,\ldots,s}$ and $\mathbf{Z}^{(s)} = (\mathbf{Z}_t)_{t=1,\ldots,s}$ the Gaussian vector containing the values of $(Z_t(x))_{t=1,\ldots,s}$ at points in $(\mathbf{D}_t)_{t=1,\ldots,s}$ with $\mathbf{D}_s \subseteq \mathbf{D}_{s-1} \subseteq \cdots \subseteq \mathbf{D}_1$. We denote by $\mathbf{D}_{test}$ a set made with the points of index $\boldsymbol{\xi}_s$ of $\mathbf{D}_s$ and $\boldsymbol{\xi}_t$ the corresponding points in $\mathbf{D}_t$ with $1 \leq t \leq s$. Then, if we denote by $\varepsilon_{Z_s,\boldsymbol{\xi}_s}$ the errors (i.e. real values minus predicted values) of the cross-validation procedure when we remove the points of $\mathbf{D}_{test}$ from the t highest levels of code, we have:*

$$\left(\varepsilon_{Z_s,\boldsymbol{\xi}_s} - \hat{\rho}_{s-1}(\mathbf{D}_{test}) \odot \varepsilon_{Z_{s-1},\boldsymbol{\xi}_{s-1}}\right) \left[\mathbf{R}_s^{-1}\right]_{[\boldsymbol{\xi}_s,\boldsymbol{\xi}_s]} = \left[\mathbf{R}_s^{-1}\left(\mathbf{z}_s - \mathbf{H}_s\boldsymbol{\lambda}_{s,-\boldsymbol{\xi}_s}\right)\right]_{[\boldsymbol{\xi}_s]}, \quad (4.17)$$

with $\varepsilon_{Z_u,\boldsymbol{\xi}_u} = 0$ when $u < t$, $\boldsymbol{\lambda}_{s,-\boldsymbol{\xi}_s} = \left([\mathbf{H}_s]'_{[-\boldsymbol{\xi}_s]}\mathbf{K}_s[\mathbf{H}_s]_{[-\boldsymbol{\xi}_s]}\right)^{-1}[\mathbf{H}_s]'_{[-\boldsymbol{\xi}_s]}\mathbf{K}_s z_s(\mathbf{D}_s \setminus \mathbf{D}_{test})$, $\hat{\rho}_{s-1}(\mathbf{D}_{test}) = \mathbf{g}'_{s-1}(\mathbf{D}_{test})[\boldsymbol{\lambda}_{s,-\boldsymbol{\xi}_s}]_{1,\ldots,q_{s-1}}$ and:

$$\mathbf{K}_s = \left[\mathbf{R}_s^{-1}\right]_{[-\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]} - \left[\mathbf{R}_s^{-1}\right]_{[-\boldsymbol{\xi}_s,\boldsymbol{\xi}_s]}\left(\left[\mathbf{R}_s^{-1}\right]_{[\boldsymbol{\xi}_s,\boldsymbol{\xi}_s]}\right)^{-1}\left[\mathbf{R}_s^{-1}\right]_{[\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]}. \quad (4.18)$$

Furthermore, if we note $\sigma^2_{Z_s,\boldsymbol{\xi}_s}$ the variances of the corresponding cross-validation procedure, we have:

$$\sigma^2_{Z_s,\boldsymbol{\xi}_s} = \hat{\sigma}^2_{\rho_{s-1},-\boldsymbol{\xi}_s}(\mathbf{D}_{\text{test}}) \odot \sigma^2_{Z_{s-1},\boldsymbol{\xi}_{s-1}} + \sigma^2_{s,-\boldsymbol{\xi}_s} \text{diag}\left(\left(\left[\mathbf{R}_s^{-1}\right]_{[\boldsymbol{\xi}_s,\boldsymbol{\xi}_s]}\right)^{-1}\right) + \mathcal{V}_s, \quad (4.19)$$

with $\boldsymbol{\Sigma}_{\rho,s-1,-\boldsymbol{\xi}_s} = \left[\left([\mathbf{H}_s]'_{[-\boldsymbol{\xi}_s]}\mathbf{K}_s[\mathbf{H}_s]_{[-\boldsymbol{\xi}_s]}\right)^{-1}\right]_{[1,\ldots,q_{s-1},1,\ldots,q_{s-1}]}$,

$$\hat{\sigma}^2_{\rho_{s-1},-\boldsymbol{\xi}_s}(\mathbf{D}_{\text{test}}) = \mathbf{g}'_{s-1}(\mathbf{D}_{test})\left(\boldsymbol{\Sigma}_{\rho,s-1,-\boldsymbol{\xi}_s} + [\boldsymbol{\lambda}_{s,-\boldsymbol{\xi}_s}]_{1,\ldots,q_{s-1}}[\boldsymbol{\lambda}_{s,-\boldsymbol{\xi}_s}]'_{1,\ldots,q_{s-1}}\right)\mathbf{g}_{s-1}(\mathbf{D}_{test}),$$

and

$$\sigma^2_{s,-\boldsymbol{\xi}_s} = \frac{\left(z_s(\mathbf{D}_s \setminus \mathbf{D}_{\text{test}}) - [\mathbf{H}_s]_{[-\boldsymbol{\xi}_s]}\boldsymbol{\lambda}_{s,-\boldsymbol{\xi}_s}\right)'\mathbf{K}_s\left(z_s(\mathbf{D}_s \setminus \mathbf{D}_{\text{test}}) - [\mathbf{H}_s]_{[-\boldsymbol{\xi}_s]}\boldsymbol{\lambda}_{s,-\boldsymbol{\xi}_s}\right)}{n_s - p_s - q_{s-1} - n_{train}}.$$

where $\sigma^2_{u,-\boldsymbol{\xi}_u} = 0$ *when* $u < t$, n_{train} *is the length of the index vector* $\boldsymbol{\xi}_s$, $\mathbf{H}_s = [\mathbf{G}_{s-1} \odot (z_{s-1}(\mathbf{D}_s)\mathbf{1}'_{q_{s-1}}) \quad \mathbf{F}_s]$ *and:*

$$\mathcal{V}_s = \mathcal{U}'_s\left([\mathbf{H}_s]'_{[-\boldsymbol{\xi}_s]}\mathbf{K}_s[\mathbf{H}_s]_{[-\boldsymbol{\xi}_s]}\right)^{-1}\mathcal{U}_s, \quad (4.20)$$

with $\mathcal{U}_s = \left([\mathbf{R}_s^{-1}]_{[\boldsymbol{\xi}_s,\boldsymbol{\xi}_s]}\right)^{-1}\left[\mathbf{R}_s^{-1}\mathbf{H}_s\right]_{[\boldsymbol{\xi}_s]}$.

Proof. Let us consider that $\boldsymbol{\xi}_s$ is the index of the k last points of $\mathbf{D}_s$. We denote by $\mathbf{D}_{\text{test}}$ these points. First we consider the variance and the trend parameters as fixed, i.e. $\sigma^2_{t,-\boldsymbol{\xi}_t} = \frac{Q_t}{2(a_t-1)}$ and $\boldsymbol{\lambda}_{t,-\boldsymbol{\xi}_t} = \boldsymbol{\Sigma}_t\boldsymbol{\nu}_t$, and $\mathcal{V}_s = 0$, i.e. we are in the simple co-kriging case. Thanks to the block-wise inversion formula, we have the following equality:

$$\mathbf{R}_s^{-1} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}' & \mathcal{Q}^{-1} \end{pmatrix}, \quad (4.21)$$

with $\mathbf{A} = [\mathbf{R}_s]^{-1}_{[-\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]} + [\mathbf{R}_s]^{-1}_{[-\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]}[\mathbf{R}_s]_{[-\boldsymbol{\xi}_s,\boldsymbol{\xi}_s]}\mathcal{Q}^{-1}[\mathbf{R}_s]_{[\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]}[\mathbf{R}_s]^{-1}_{[-\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]}$,
$\mathbf{B}' = -\mathcal{Q}^{-1}[\mathbf{R}_s]_{[\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]}[\mathbf{R}_s]^{-1}_{[-\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]}$ and:

$$\mathcal{Q} = [\mathbf{R}_s]_{[\boldsymbol{\xi}_s,\boldsymbol{\xi}_s]} - [\mathbf{R}_s]_{[\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]}[\mathbf{R}_s]^{-1}_{[-\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]}[\mathbf{R}_s]_{[-\boldsymbol{\xi}_s,\boldsymbol{\xi}_s]}. \quad (4.22)$$

We note that $\frac{Q_s}{2(a_s-1)}\mathcal{Q} = \frac{Q_t}{2(a_t-1)}\left(\left[\mathbf{R}_s^{-1}\right]_{[\boldsymbol{\xi}_s,\boldsymbol{\xi}_s]}\right)^{-1}$ represents the covariance matrix of the points in $\mathbf{D}_{\text{test}}$ with respect to the covariance kernel of a Gaussian process of kernel $\frac{Q_s}{2(a_s-1)}r_s(x,\tilde{x})$ (which is the one of $\delta_s(x)$) conditioned by the points $\mathbf{D}_s \setminus \mathbf{D}_{\text{test}}$. Therefore, from the previous remark and Equation (4.4), we can deduce Equation (4.19).

Furthermore, we have the following equality:

$$\begin{aligned}\left(\left[\mathbf{R}_s^{-1}\right]_{[\boldsymbol{\xi}_s,\boldsymbol{\xi}_s]}\right)^{-1}\left[\mathbf{R}_s^{-1}\left(\mathbf{z}_s - \mathbf{H}_s\boldsymbol{\lambda}_{s,-\boldsymbol{\xi}_s}\right)\right]_{[\boldsymbol{\xi}_s]} &= z_s(\mathbf{D}_{\text{test}}) - h'_s(\mathbf{D}_{\text{test}})\boldsymbol{\Sigma}_s\boldsymbol{\nu}_s \\ &- [\mathbf{R}_s]_{[\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]}[\mathbf{R}_s]^{-1}_{[-\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]} \\ &\left(z_s(\mathbf{D}_s \setminus \mathbf{D}_{\text{test}}) - [\mathbf{H}_s]'_{[-\boldsymbol{\xi}_s]}\boldsymbol{\Sigma}_s\boldsymbol{\nu}_s\right)\end{aligned}. \quad (4.23)$$

where $h'_s(x) = [\rho_{s-1}(x)\mu_{Z_{s-1}}(x) \quad \mathbf{f}'_t(x)]$. From this equation and Equation (4.3), we can directly deduce Equation (4.17) with $\varepsilon_{Z_s,\boldsymbol{\xi}_s} = z_s(\mathbf{D}_{\text{test}}) - \mu_{Z_s}(\mathbf{D}_{\text{test}})$.

Then, we suppose the trend and the variance parameters as unknown and we have to re-estimate them when we remove the observations. Thanks to the parameter posterior distribution presented in Section 4.2.3, we can deduce that the estimates of $\sigma^2_{t,-\boldsymbol{\xi}_t}$ and $\boldsymbol{\lambda}_{t,-\boldsymbol{\xi}_t}$ when we remove observations of index $\boldsymbol{\xi}_t$ are given by the following equations:

$$\boldsymbol{\lambda}_{s,-\boldsymbol{\xi}_s} = \left([\mathbf{H}_s]'_{[-\boldsymbol{\xi}_s]}\mathbf{K}_s[\mathbf{H}_s]_{[-\boldsymbol{\xi}_s]}\right)^{-1}[\mathbf{H}_s]'_{[-\boldsymbol{\xi}_s]}\mathbf{K}_s z_s(\mathbf{D}_s \setminus \mathbf{D}_{test}) \tag{4.24}$$

and:

$$\sigma^2_{s,-\boldsymbol{\xi}_s} = \frac{\left(z_s(\mathbf{D}_s \setminus \mathbf{D}_{\text{test}}) - [\mathbf{H}_s]_{-\boldsymbol{\xi}_s}\boldsymbol{\lambda}_{s,-\boldsymbol{\xi}_s}\right)'\mathbf{K}_s\left(z_s(\mathbf{D}_s \setminus \mathbf{D}_{\text{test}}) - [\mathbf{H}_s]_{-\boldsymbol{\xi}_s}\boldsymbol{\lambda}_{s,-\boldsymbol{\xi}_s}\right)}{n_s - p_s - q_{s-1} - n_{train}}, \tag{4.25}$$

with $\mathbf{K}_s = [\mathbf{R}_s]^{-1}_{[-\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]}$.

From the equality (4.21), we can deduce that $\mathbf{K}_s = \mathbf{A} - \mathbf{B}\mathcal{Q}\mathbf{B}'$ from which we obtain Equation (4.18). Finally, to obtain the cross-validation equations for the universal co-kriging, we just have to estimate the following quantity (see Equation (4.13)):

$$\left(h'_s(\mathbf{D}_{\text{test}}) - [\mathbf{R}_s]_{[\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]}\mathbf{K}_s[\mathbf{H}_s]_{-\boldsymbol{\xi}_s}\right)\boldsymbol{\Sigma}_s\left(h'_s(\mathbf{D}_{\text{test}}) - [\mathbf{R}_s]_{[\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]}\mathbf{K}_s[\mathbf{H}_s]_{-\boldsymbol{\xi}_s}\right)', \tag{4.26}$$

with $\boldsymbol{\Sigma}_s = \left([\mathbf{H}'_s]_{-\boldsymbol{\xi}_s}\mathbf{K}_s[\mathbf{H}_s]_{-\boldsymbol{\xi}_s}\right)^{-1}$. The following equality:

$$\left(h'_s(\mathbf{D}_{\text{test}}) - [\mathbf{R}_s]_{[\boldsymbol{\xi}_s,-\boldsymbol{\xi}_s]}\mathbf{K}_s[\mathbf{H}_s]_{-\boldsymbol{\xi}_s}\right) = \left(\left([\mathbf{R}_s^{-1}]_{[\boldsymbol{\xi}_s,\boldsymbol{\xi}_s]}\right)^{-1}\left[\mathbf{R}_s^{-1}\mathbf{H}_s\right]_{[\boldsymbol{\xi}_s]}\right), \tag{4.27}$$

allows us to obtain Equation (4.20) and completes the proof. □

We note that these equations are also valid when $s = 1$, i.e. for kriging model. We hence have closed form expressions for the equations of a k-fold cross-validation with a re-estimation of the regression and variance parameters. These expressions can be deduced from the universal co-kriging equations. The complexity of this procedure is essentially determined by the inversion of the matrices $\left(\left[\mathbf{R}_u^{-1}\right]_{[\boldsymbol{\xi}_u,\boldsymbol{\xi}_u]}\right)_{u=t,\dots,s}$ of size $n_{test} \times n_{test}$. Furthermore, if we suppose the parameters of variance and/or trend as known, we do not have to compute $\sigma^2_{t,-\boldsymbol{\xi}_t}$ and/or $\boldsymbol{\lambda}_{t,-\boldsymbol{\xi}_t}$ (they are fixed to their estimated value, i.e. $\sigma^2_{t,-\boldsymbol{\xi}_t} = \frac{Q_t}{2(a_t-1)}$ and $\boldsymbol{\lambda}_{t,-\boldsymbol{\xi}_t} = \boldsymbol{\Sigma}_t\boldsymbol{\nu}_t$, see Section 4.2.3) which reduces substantially the complexity of the method. These equations generalize those of [Dubrule, 1983] and [Zhang and Wang, 2009] where the variance $\sigma^2_{t,-\boldsymbol{\xi}_t}$ is supposed to be known. Finally, the term $\mathcal{V}_s$ is the additive term due to the parameter posterior distributions in the universal co-kriging. Therefore, if the trend parameters are supposed to be known, this term is equal to 0.

Remark: We must recognize that our closed form cross-validation formulas do not allow for the re-estimation of the hyper-parameters of the correlation functions. However, as discussed in Subsection 4.5.1, Proposition 4.3 is useful even in that case to reduce the computational complexity of the cross-validation procedure.

4.5 Illustration: hydrodynamic simulator

In this section we apply our co-kriging method to the hydrodynamic code "MELTEM". The aim of the study is to build a prediction as accurate as possible using only a few runs of the complex code and to assess the uncertainty of this prediction. In particular, we show the efficiency of the co-kriging model compared to the kriging one. We also illustrate the difference between simple and universal co-kriging and the results of the LOO-CV procedure. These illustrations are made possible and easy by the closed form formulas for the predictive mean and variance for universal co-kriging and by the fast cross-validation procedure described in Section 4.4 and 4.3 respectively. Finally, we show that considering an adjustment coefficient depending on x can be worthwhile.

The code MELTEM simulates a second-order turbulence model for gaseous mixtures induced by Richtmyer-Meshkov instability [Grégoire et al., 2005]. Two input parameters x_1 and x_2 are considered. They are phenomenological coefficients used in the equations of the energy of dissipation of the turbulent flow. These two coefficients vary in the region $[0.5, 1.5] \times [1.5, 2.3]$. The considered code outputs, called *eps* and L_c, are respectively the dissipation factor and the mixture characteristic length. The simulator is a finite-elements code which can be run at $s = 2$ levels of accuracy by altering the finite-elements mesh. The simple code $z_1(.)$, using a coarse mesh, takes 15 seconds to produce an output whereas the complex code $z_2(.)$, using a fine mesh, takes 8 minutes. We use 5 runs for the complex code $z_2(x)$ and 25 runs for the cheap code $z_1(x)$. This represents 8 minutes on a hexa-core processor, which is our constraint for an operational use. Then, we build an additional set of 175 points to test the accuracy of the models. We note that no prior information is available: we are hence in the non-informative case.

4.5.1 Estimation of the hyper-parameters

In the previous sections, we considered the correlation kernels $(r_t(x, \tilde{x}))_{t=1,\dots,s}$ as known. In practical applications, we choose these kernels in a parameterized family of correlation kernels. Therefore, we consider kernels such that $r_t(x, \tilde{x}) = r_t(x, \tilde{x}; \phi_t)$. For $t = 1, \dots, s$ the hyper-parameter ϕ_t can be estimated by maximizing the concentrated restricted log-likelihood (see [Santner et al., 2003] and Chapter 1 Section 1.3) with respect to ϕ_t:

$$\log\left(|\det\left(\mathbf{R}_t\right)|\right) + (n_t - p_t - q_{t-1}) \log\left(\sigma^2_{t,\mathrm{REML}}\right), \tag{4.28}$$

with the convention $q_0 = 0$ and $\sigma^2_{t,\mathrm{REML}}$ is the restricted likelihood estimate of the variance σ^2_t (see Section 4.2.3). This minimization problem has to be solved numerically.

It is a common choice to estimate the hyper-parameters by maximum likelihood [Santner et al., 2003]. It is also possible to estimate the hyper-parameters $(\phi_t)_{t=1,\dots,s}$ by minimizing a loss function of a Leave-One-Out Cross-Validation procedure (see Section 1.3). Usually, the complexity of this procedure is $\mathcal{O}\left(\left(\sum_{i=1}^s n_i\right)^4\right)$. Nonetheless, thanks to Proposition 4.3, it is reduced to $\mathcal{O}\left(\sum_{i=1}^s n_i^3\right)$ since it is essentially determined by the inversions of the s matrices $(\mathbf{R}_t)_{t=1,\dots,s}$.Therefore, the complexity for the estimation of $(\phi_t)_{t=1,\dots,s}$ is substantially reduced.

Furthermore, the recursive formulation of the problem allows us to estimate the parameters $(\phi_t)_{t=1,\dots,s}$ one at a time by starting with ϕ_1 and estimating ϕ_t, $t = 2, \dots, s$ recursively.

4.5.2 Comparison between kriging and multi-fidelity co-kriging

Before considering the real case study, we propose in this section a comparison between the kriging and co-kriging models when the number of runs n_2 for the complex code varies such that $n_2 = 5, 10, 15, 20, 25$. For the co-kriging model, we consider $n_1 = 25$ runs for the cheap code. In this section, we focus on the output *eps*.

To perform the comparison, we generate randomly 500 experimental design sets $(\mathbf{D}_{2,i}, \mathbf{D}_{1,i})_{i=1,\dots,500}$ such that $\mathbf{D}_{2,i} \subset \mathbf{D}_{1,i}$, $i = 1, \dots, 500$, $\mathbf{D}_{1,i}$ has n_1 points and $\mathbf{D}_{2,i}$ has n_2 points.

We use for both kriging and co-kriging models a Matérn-5/2 covariance kernel and we consider ρ, $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ as constant. The accuracies of the two models are evaluated on the test set composed of 175 observations. From them, the Root Mean Squared Error (RMSE) is computed: $\text{RMSE} = \left(\frac{1}{175}\sum_{i=1}^{175}(\mu_{Z_2}(x_i^{\text{test}}) - z_2(x_i^{\text{test}}))^2\right)^{1/2}$.

Figure 4.1 gives the mean and the quantiles of probability 5% and 95% of the RMSE computed from the 500 sets $(\mathbf{D}_{2,i}, \mathbf{D}_{1,i})_{i=1,\dots,500}$ when the number of runs for the expensive code n_2 varies. In Figure 4.1, we can see that the errors converge to the same value when n_2

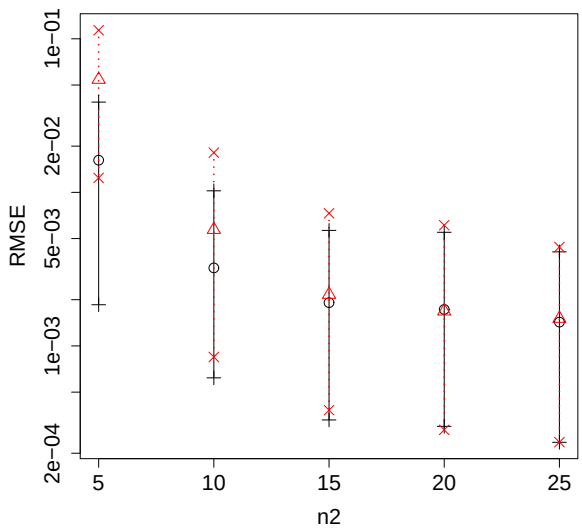


Figure 4.1: Comparison between kriging and co-kriging with $n_1 = 25$ runs for the cheap code (500 nested design sets have been randomly generated for each n_2). The circles represent the averaged RMSE of the co-kriging, the triangles represent the averaged RMSE of the kriging, the crosses represent the quantiles of probability 5% and 95% for the co-kriging RMSE and the times signs represent the quantiles of probability 5% and 95% of the kriging RMSE. Co-kriging predictions are better than the ordinary kriging ones for small n_2 and they converge to the same accuracy when n_2 tends to $n_1 = 25$.

tends to n_1. Indeed, due to the Markov property given in Section 3.2, when $\mathbf{D}_2 = \mathbf{D}_1$, only the observations $\mathbf{z}_2$ are taken into account. Furthermore, we can see that for small values of n_2, it is worth considering the co-kriging model since its accuracy is significantly better than the one of the kriging model.

4.5.3 Nested space filling design

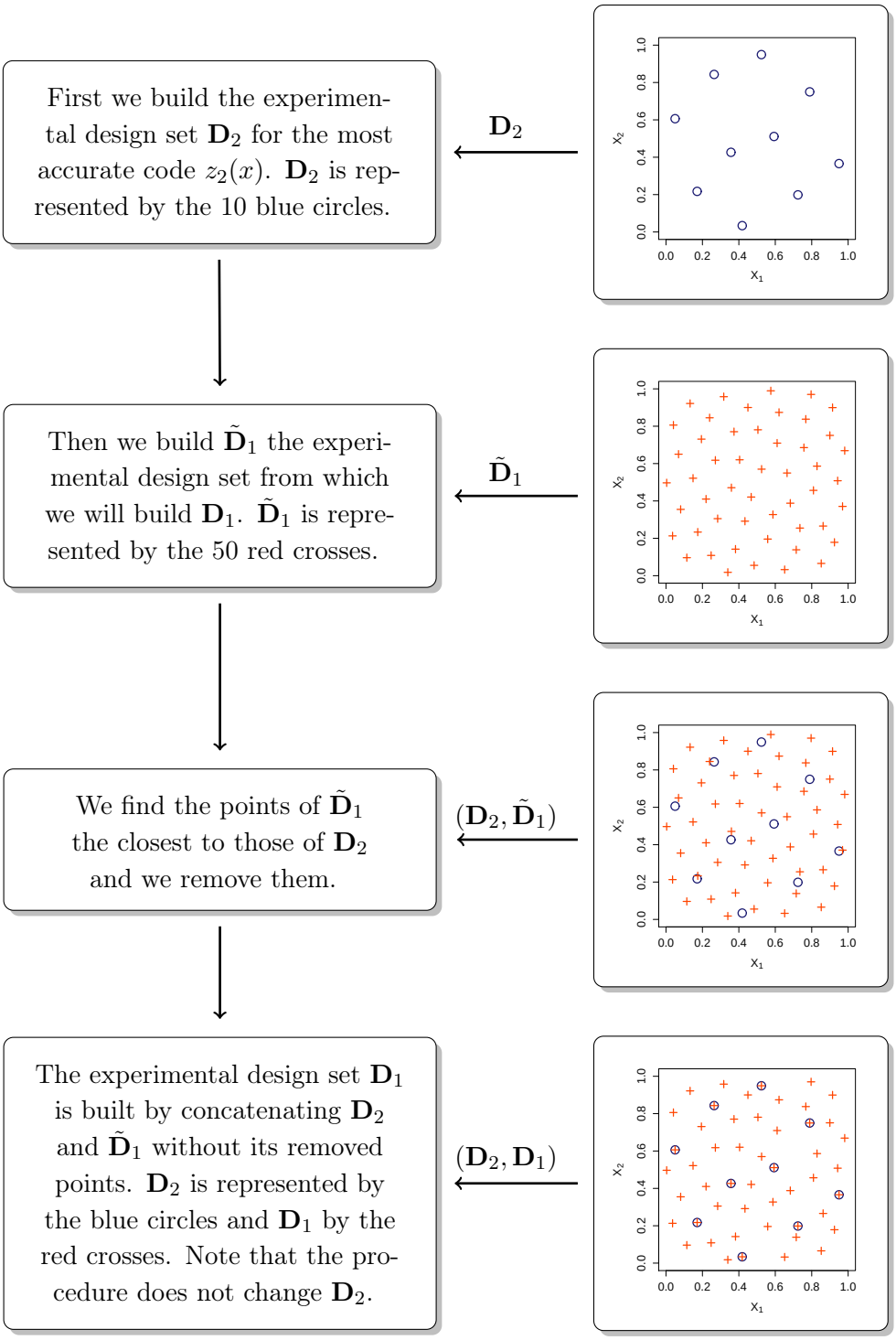
As presented in Section 4.2 we consider nested experimental design sets: $\forall t = 2, \ldots, s \quad \mathbf{D}_t \subseteq \mathbf{D}_{t-1}$. Therefore, we have to adopt particular design strategies to uniformly spread the inputs for all $\mathbf{D}_t$. A strategy based on Orthogonal array-based Latin hypercube for nested space-filling designs is proposed by [Qian et al., 2009].
We consider here another strategy for space-filling design, described in the following algorithm, which is very simple and not time-consuming. The number of points n_t for each design $\mathbf{D}_t$ is prescribed by the user, as well as the experimental design method applied to determine the coarsest grid $\mathbf{D}_s$ used for the most expensive code z_s (see [Fang et al., 2006] for a review of different methods).

Algorithm 1 Nested space filling design

```
build D_s = {x_j^(s)}_{j=1,...,n_s} with the experimental design method prescribed by the user.
for t = s to 2 do
    build design D̃_{t-1} with the experimental design method prescribed by the user.
    for i = 1 to n_t do
        find x̃_j^(t-1) ∈ D̃_{t-1} the closest point from x_i^(t) ∈ D_t where j ∈ [1, n_{t-1}].
        remove x̃_j^(t-1) from D̃_{t-1}.
    end for
    D_{t-1} = D̃_{t-1} ∪ D_t.
end for
```

This strategy allows us to use any space-filling design method and it conserves the initial structure of the experimental design $\mathbf{D}_s$ of the most accurate code, contrarily to a strategy based on selection of subsets of an experimental design for the less accurate code as presented by [Kennedy and O'Hagan, 2000] and [Forrester et al., 2007]. We hence can ensure that $\mathbf{D}_s$ has excellent space-filling properties. Moreover, the experimental design $\mathbf{D}_{t-1}$ being equal to $\tilde{\mathbf{D}}_{t-1} \cup \mathbf{D}_t$, this method ensures the nested property.

We illustrate in the next page the different stage of the nested design procedure for $s = 2$.



In the presented application, we consider $n_2 = 5$ points for the expensive code $z_2(x)$ and $n_1 = 25$ points for the cheap one $z_1(x)$. We apply the previous algorithm to build $\mathbf{D}_2$ and $\mathbf{D}_1$ such that $\mathbf{D}_2 \subset \mathbf{D}_1$. For the experimental design set $\mathbf{D}_2$, we use a Latin-Hypercube-Sampling [Stein, 1987] optimized with respect to the S-optimality criterion which maximizes the mean distance from each design point to all the other points [Stocki, 2005]. Furthermore, the set

$\mathbf{D}_1$ is built using a maximum entropy design [Shewry and Wynn, 1987] optimized with the Fedorov-Mitchell exchange algorithm [Currin et al., 1991]. These algorithms are implemented in the library R lhs. The obtained nested designs are shown in Figure 4.2.

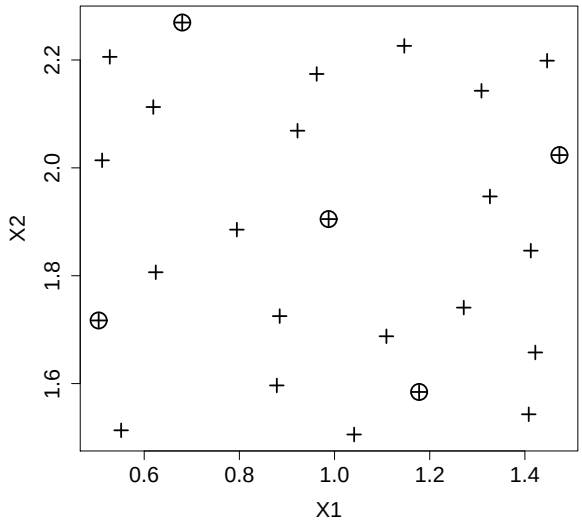


Figure 4.2: Nested experimental design sets for the hydrodynamic application. The crosses represent the $n_1 = 25$ points of the experimental design set $\mathbf{D}_1$ of the cheap code and the circles represent the $n_2 = 5$ points of the experimental design set $\mathbf{D}_2$ of the expensive code.

4.5.4 Multi-fidelity surrogate model for the dissipation factor *eps*

We build here a co-kriging model for the dissipation factor *eps*. The obtained model is compared to a kriging one. This first example is used to illustrate the efficiency of the co-kriging method compared to the kriging. It will also allow us to highlight the difference between the simple and the universal co-kriging.

We use the experimental design sets presented in Section 4.5.3. To validate and compare our models, the 175 simulations of the complex code uniformly spread on $[0.5, 1.5] \times [1.5, 2.3]$ are used. To build the different correlation matrices, we consider a tensorised Matérn-5/2 kernel (see [Rasmussen and Williams, 2006] and Chapter 1 Section 1.4):

$$r(x, \tilde{x}; \theta_t) = r_{1d}(x_1, \tilde{x}_1; \theta_{t,1}) r_{1d}(x_2, \tilde{x}_2; \theta_{t,2}), \tag{4.29}$$

with $x = (x_1, x_2) \in [0.5, 1.5] \times [1.5, 2.3]$, $\theta_{t,1}, \theta_{t,2} \in (0, +\infty)$ and:

$$r_{1d}(x_i, \tilde{x}_i; \theta_{t,i}) = \left(1 + \sqrt{5}\frac{|x_i - \tilde{x}_i|}{\theta_{t,i}} + \frac{5}{3}\frac{(x_i - \tilde{x}_i)^2}{\theta_{t,i}^2}\right) \exp\left(-\sqrt{5}\frac{|x_i - \tilde{x}_i|}{\theta_{t,i}}\right). \tag{4.30}$$

Then, we consider $\mathbf{g}_1(x) = 1$, $\mathbf{f}_2(x) = 1$, $\mathbf{f}_1(x) = 1$ (see Section 4.2.1) and, using the concentrated maximum likelihood (see subsections 4.5.1 and 1.3.2), we have the following estimates for the correlation hyper-parameters: $\hat{\boldsymbol{\theta}}_1 = (0.69, 1.20)$ and $\hat{\boldsymbol{\theta}}_2 = (0.27, 1.37)$.

According to the values of the hyper-parameter estimates, the co-kriging model is smooth since the correlation lengths are of the same order as the size of the input parameter space. Furthermore, the estimated Pearson correlation between the two codes is 82.64%, which shows that the amount of information contained in the cheap code is substantial.

Table 4.1 presents the results of the parameter Bayesian estimation (see Section 4.2.3).

Trend coefficient	$\boldsymbol{\Sigma}_t \boldsymbol{\nu}_t$	$\boldsymbol{\Sigma}_t / \sigma_t^2$
$\boldsymbol{\beta}_1$	8.84	0.48
$\begin{pmatrix} \boldsymbol{\beta}_{\rho_1} \\ \boldsymbol{\beta}_2 \end{pmatrix}$	$\begin{pmatrix} 0.92 \\ 0.74 \end{pmatrix}$	$\begin{pmatrix} 1.98 & -18.13 \\ -18.13 & 165.82 \end{pmatrix}$
Variance coefficient	Q_t	$2\alpha_t$
σ_1^2	6.98	24
σ_2^2	0.06	3

Table 4.1: Application: hydrodynamic simulator. Parameter Bayesian estimation results for the response *eps* (see equations (4.9) and (4.11)).

We see in Table 4.1 that the correlation between $\boldsymbol{\beta}_{\rho_1}$ and $\boldsymbol{\beta}_2$ is important which highlights the importance of taking into account the correlation between these two coefficients for the parameter estimation. We also see that the adjustment parameter $\boldsymbol{\beta}_{\rho_1}$ is close to 1, which means that the two codes are highly correlated.

Figure 4.3 illustrates the contour plot of the kriging and co-kriging means, we can see significant differences between the two surrogate models.

Table 4.2 compares the prediction accuracy of the co-kriging and the kriging models. The different coefficients are estimated with the 175 responses of the complex code on the test set:

MaxAE: Maximal absolute value of the observed error.

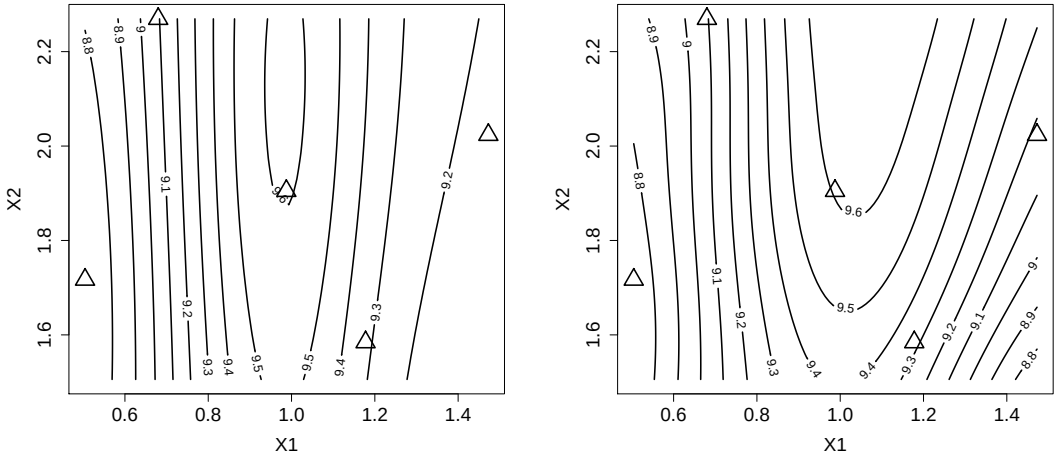
RMSE : Root mean squared value of the observed error.

Eff $= 1 - ||\mu_{Z_2}(\mathbf{D}_{\text{test}}) - z_2(\mathbf{D}_{\text{test}})||^2 / ||\mu_{Z_2}(\mathbf{D}_{\text{test}}) - \bar{z}_2||^2$, with $\bar{z}_2 = (\sum_{i=1}^{n_2} z_2(x_i^{\text{test}}))/n_2$.

RIMSE : Root of the average value of the kriging or co-kriging variance.

	Eff	RMSE	MaxAE	RIMSE.
kriging	75.83%	0.133	0.49	0.110
co-kriging	98.01%	0.038	0.14	0.046

Table 4.2: Application: hydrodynamic simulator. Comparison between kriging and co-kriging. The co-kriging model provides predictions significantly better than the ones of the kriging model.



(a) (b)

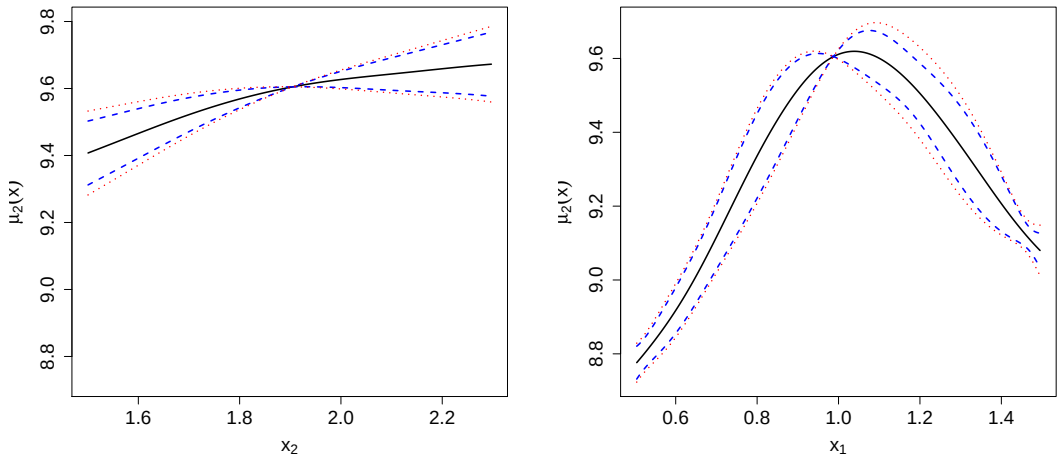
Figure 4.3: Contour plot of the kriging mean (Figure (a)) and the co-kriging mean (Figure (b)). The triangles represent the $n_2 = 5$ points of the experimental design set of the expensive code.

We can see that the difference of accuracy between the two models is important. Indeed, the one of the co-kriging model is significantly better. Furthermore, comparing the RMSE and the RIMSE estimates in Table 4.2, we see that we have good estimates of the predictive distribution variances for the two models. We note that the predictive variance for the co-kriging is obtained with a simple co-kriging model. Therefore, it will be slightly larger in the universal co-kriging case. Indeed, by computing the universal co-kriging equations, we find RIMSE $= 0.058$.

We can compare the RMSE obtained with the test set with the RMSE obtained with a Leave-One-Out cross validation procedure (see Section 4.4). For this procedure, we test our model on $n_2 = 5$ validation sets obtained by removing one observation at a time. As presented in Section 4.4, we can either choose to remove the observations from $\mathbf{z}_2$ or from $\mathbf{z}_2$ and $\mathbf{z}_1$. The root mean squared error of the Leave-One-Out cross validation procedure obtained by removing observations from $\mathbf{z}_2$ is $\text{RMSE}_{z_2,LOO} = 4.80.10^{-3}$ whereas the one obtained by removing observations from $\mathbf{z}_2$ and $\mathbf{z}_1$ is $\text{RMSE}_{z_1,z_2,LOO} = 0.10$. Comparing $\text{RMSE}_{z_2,LOO}$ and $\text{RMSE}_{z_1,z_2,LOO}$ to the RMSE obtained with the external test set, we see that the procedure which consists in removing points from $\mathbf{z}_2$ and $\mathbf{z}_1$ provides a better proxy for the generalization error. Indeed, $\text{RMSE}_{z_2,LOO}$ is a relevant proxy for the generalization error only at points where $\mathbf{z}_1$ is available. Therefore, it underestimates the error at locations where $\mathbf{z}_1$ is unknown.

Figure 4.4 represents the mean and confidence intervals at plus or minus twice the standard deviation of the simple and universal co-krigings for points along the vertical line $x_1 = 0.99$ and the horizontal line $x_2 = 1.91$ ($x = (0.99, 1.91)$ corresponds to the coordinates of the point of $\mathbf{D}_2$ in the center of the domain $[0.5, 1.5] \times [1.5, 2.3]$ in Figure 4.2). In Figure 4.4 on the right hand side, we see a necked point around the coordinates $x_1 = 1.5$ since, in the direction

of x_2, the correlation hyper-parameters length for $Z_1(x)$ and $\delta_2(x)$ are large ($\theta_{1,2} = 1.20$ and $\theta_{2,2} = 1.37$) and a point of $\mathbf{D}_2$ has almost the same coordinate.



(a) (b)

Figure 4.4: Mean and confidence intervals for the simple and the universal co-kriging. Figure (a) represents the predictions along the vertical line $x_1 = 0.99$ and (b) represents the predictions along the horizontal line $x_2 = 1.91$. The solid black lines represent the mean of the two co-kriging models, the dashed lines represent the confidence interval at plus or minus twice the standard deviation of the simple co-kriging and the dotted lines represent the same confidence intervals for the universal co-kriging.

4.5.5 Multi-fidelity surrogate model for the mixture characteristic length L_c

In this section, we build a co-kriging model for the mixture characteristic length L_c. The aim of this example is to highlight that it can be worth having an adjustment coefficient ρ_1 depending on x. We use the same training and test sets as in the previous section and we consider a tensorised Matérn-5/2 kernel (4.29). Let us consider the two following cases:

Case 1: $\mathbf{g}_1(x) = 1$, $\mathbf{f}_2(x) = 1$ and $\mathbf{f}_1(x) = 1$

Case 2: $\mathbf{g}_1'(x) = \begin{pmatrix} 1 & x_1 \end{pmatrix}$, $\mathbf{f}_2(x) = 1$ and $\mathbf{f}_1(x) = 1$

We have the following hyper-parameter maximum likelihood estimates for the two cases

Case 1: $\hat{\boldsymbol{\theta}}_1 = (0.52, 1.09)$ and $\hat{\boldsymbol{\theta}}_2 = (0.03, 0.02)$

Case 2: $\hat{\boldsymbol{\theta}}_1 = (0.52, 1.09)$ and $\hat{\boldsymbol{\theta}}_2 = (0.14, 1.37)$

The estimate of $\hat{\boldsymbol{\theta}}_1$ is identical in the two cases since it does not depend on ρ_1 and it is estimated with the same observations. Furthermore, we can see an important difference between the

estimates of $\hat{\boldsymbol{\theta}}_2$. Indeed, they are larger in the Case 2 than in the Case 1 which indicates that the model is smoother in the Case 2. Table 4.3 presents the posterior distributions of $\boldsymbol{\beta}_1$ and σ_1^2 for the two cases (see Section 4.2.3).

Trend coefficient	$\boldsymbol{\Sigma}_1\boldsymbol{\nu}_1$	$\boldsymbol{\Sigma}_1/\sigma_1^2$
$\boldsymbol{\beta}_1$	1.26	0.97
Variance coefficient	Q_1	$2\alpha_1$
σ_1^2	15.62	24

Table 4.3: Application: hydrodynamic simulator. Posterior distributions of $\boldsymbol{\beta}_1$ and σ_1^2 for the response L_c (see equations (4.9) and (4.11)).

Then, Table 4.4 presents the posterior distributions of $\boldsymbol{\beta}_2$, $\boldsymbol{\beta}_{\rho_1}$ and σ_2^2 for the Case 1, i.e. when ρ_1 is constant (see Section 4.2.3).

Trend coefficient	$\boldsymbol{\Sigma}_2\boldsymbol{\nu}_2$	$\boldsymbol{\Sigma}_2/\sigma_2^2$
$\begin{pmatrix}\boldsymbol{\beta}_{\rho_1}\\ \boldsymbol{\beta}_2\end{pmatrix}$	$\begin{pmatrix}1.49\\ -0.26\end{pmatrix}$	$\begin{pmatrix}0.83 & -0.79\\ -0.79 & 0.95\end{pmatrix}$
Variance coefficient	Q_2	$2\alpha_2$
σ_2^2	0.01	3

Table 4.4: Application: hydrodynamic simulator. Posterior distributions of $\boldsymbol{\beta}_2$, $\boldsymbol{\beta}_{\rho_1}$ and σ_2^2 for the Case 1, i.e. when ρ_1 is constant, for the response L_c (see equations (4.9) and (4.11)).

Finally, Table 4.5 presents the posterior distributions of $\boldsymbol{\beta}_2$, $\boldsymbol{\beta}_{\rho_1}$ and σ_2^2 for the Case 2, i.e. when ρ_1 depends on x (see Section 4.2.3).

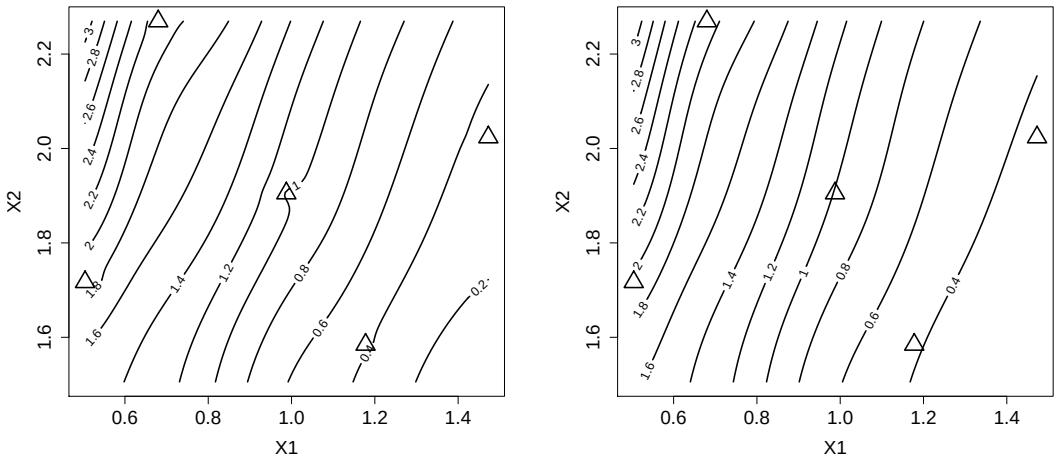
Trend coefficient	$\boldsymbol{\Sigma}_2\boldsymbol{\nu}_2$	$\boldsymbol{\Sigma}_2/\sigma_2^2$
$\begin{pmatrix}\boldsymbol{\beta}_{\rho_1}\\ \boldsymbol{\beta}_2\end{pmatrix}$	$\begin{pmatrix}1.66\\ -0.48\\ -0.04\end{pmatrix}$	$\begin{pmatrix}2.34 & -3.50 & 0.44\\ -3.50 & 9.18 & -3.67\\ 0.44 & -3.67 & 2.60\end{pmatrix}$
Variance coefficient	Q_2	$2\alpha_2$
σ_2^2	$3.24.10^{-4}$	2

Table 4.5: Application: hydrodynamic simulator. Posterior distributions of $\boldsymbol{\beta}_2$, $\boldsymbol{\beta}_{\rho_1}$ and σ_2^2 for the Case 2, i.e. when ρ_1 depends on x, for the response L_c (see equations (4.9) and (4.11)).

We see in Table 4.4 that the adjustment coefficient is around 1.5 which indicates that the magnitude of the expensive code is slightly more important than the one of the cheap code. Furthermore, we see in Table 4.5 that if we consider an adjustment coefficient which linearly depends on x_1 (i.e. with $\mathbf{g}_1'(x) = \begin{pmatrix}1 & x_1\end{pmatrix}$), the constant part of $\boldsymbol{\beta}_{\rho_1}$ is more important (it

is around 1.66) and there is a negative slope in the direction x_1 (it is around -0.48). Since $x \in [0.5, 1.5]$, the averaged value of ρ_1 is 1.18 and goes from 1.42 at $x_1 = 0.5$ to 0.94 at $x_1 = 1.5$. We see also a significant difference between the two case for the variance estimate. Indeed, the variance estimate in the Case 1 (see Table 4.4) is much more important than the one in the Case 2 (see Table 4.5). This could mean that we learn better in the Case 2 than in the Case 1.

Figure 4.5 illustrates the contour plot of the two co-kriging models, i.e. when ρ_1 is constant and when ρ_1 depends on x.



(a) (b)

Figure 4.5: Contour plot of the co-kriging mean when ρ_1 is constant (Figure (a)) and when ρ_1 is depends on x (Figure (b)). The triangles represent the $n_2 = 5$ points of the experimental design set of the expensive code.

Furthermore, Table 4.6 compares the prediction accuracy of the co-kriging in the two cases. The precision is computed on the test set of 175 observations.

	RMSE	MaxAE
Case 1	$7.26.10^{-3}$	0.23
Case 2	$1.53.10^{-3}$	0.16

Table 4.6: Application: hydrodynamic simulator. Comparison between co-kriging when ρ_1 is constant (Case 1) and co-kriging when ρ_1 depends on x (Case 2). The Case 2 provides predictions better than the Case 1, it is hence worthwhile to consider an adjustment coefficient that is not constant.

We see that the co-kriging model in Case 2 is clearly better than the one in Case 1. Therefore, we illustrate in this application that it can be worth considering an adjustment

coefficient not constant contrarily to the model presented in [Kennedy and O'Hagan, 2000] and [Forrester et al., 2007].

4.6 The R CRAN package MuFiCokriging

We have implemented a R CRAN package named "MuFiCokriging" which allows for computing the recursive multi-fidelity co-kriging model presented in this chapter. This package can be used with the software R available on the following website: http://cran.r-project.org. The package includes the major part of the previous developments, i.e.:

- The model definition and building with non-informative Bayesian parameter estimation,
- The model predictive mean and variance for the Simple and Universal co-kriging,
- The fast cross-validation procedures,
- The algorithm for designing nested experimental design sets.

We present in this section the different procedures implemented into the package "MuFiCokriging" by following an academic example with $s = 3$ levels of code and with the input dimension set to $d = 2$. Note that any s and d can be used. The package is available on the following url:

http://cran.r-project.org/web/packages/MuFiCokriging

We emphasize that our package depends on the "DiceKriging" R CRAN package (see [Roustant et al., 2012]). This allows us to benefit from the advances and the computational efficiency proposed by this package.

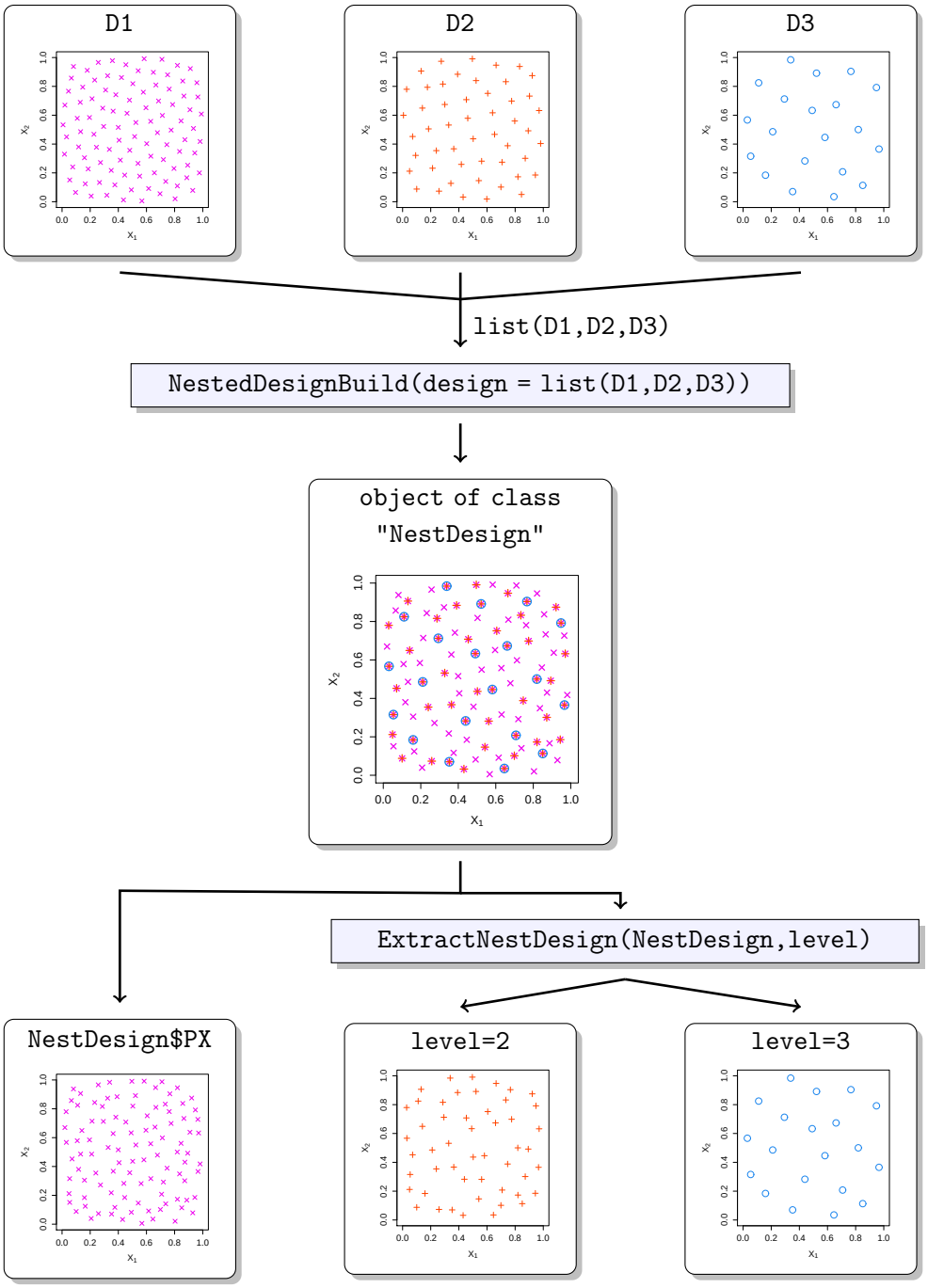
First of all, the package installation is made thanks to the following command:

```
library(MuFiCokriging)
```

We note that the text with the `verbatim font` is used to represent R codes. Furthermore, to have more detail about a function of the package, the user may use the command `help()`.

4.6.1 Nested Experimental design sets

First, let us present the function allowing for building nested experimental design sets. This function named `NestedDesignBuild` computes Algorithm 1. It takes as arguments a list of s non-nested matrices `list(D1,D2,D3)` representing the experimental design sets for all code levels. The order of the list is important, `D1` represents the experimental design set of the less accurate code and `D3` the one of the most accurate. The procedure nests the design sets such that `D3` $\subset$ `D2` $\subset$`D1` with respect to Algorithm 1 and such that `D3` will be unchanged.



As we see in the next script, the experimental design sets for the levels 1 and 2 are changed and the one for the level 3 is unchanged.

```
> identical(D1,NestDesign$PX)
[1] FALSE
> identical(D2,ExtractNestDesign(NestDesign,2))
[1] FALSE
> identical(D3,ExtractNestDesign(NestDesign,3))
```

```
[1] TRUE
```

The object class "NestDesign" is built thanks to the following procedure

```
NestedDesign(x, nlevel , indices = NULL, n = NULL)
```

where:

- `x` represents the experimental design $\mathbf{D}_1$ at level 1,
- `nlevel` represents the number s of code levels,
- `indices` is a list of index. The tth element of the list is the index of $\mathbf{D}_{t-1}$ corresponding to the points in $\mathbf{D}_t$.
- `n` is a list of integers representing the number of points for each level. It is necessary to set `n` only if `indices=NULL`. In that case, the experimental design sets $(\mathbf{D}_t)_{t=2,\dots,s}$ are randomly generated from $\mathbf{D}_1$.

The procedure `ExtractNestDesign` allows for extracting the design sets $(\mathbf{D}_t)_{t=2,\dots,s}$ from an object of class "NestDesign". We note that the experimental design set $\mathbf{D}_1$ can be obtained with the command `NestDesign$PX` where `NestDesign` is an object of class "NestDesign". Therefore, we have the following correspondence:

$\mathbf{D}_1$: `NestDesign$PX`

$\mathbf{D}_2$: `ExtractNestDesign(NestDesign,2)`

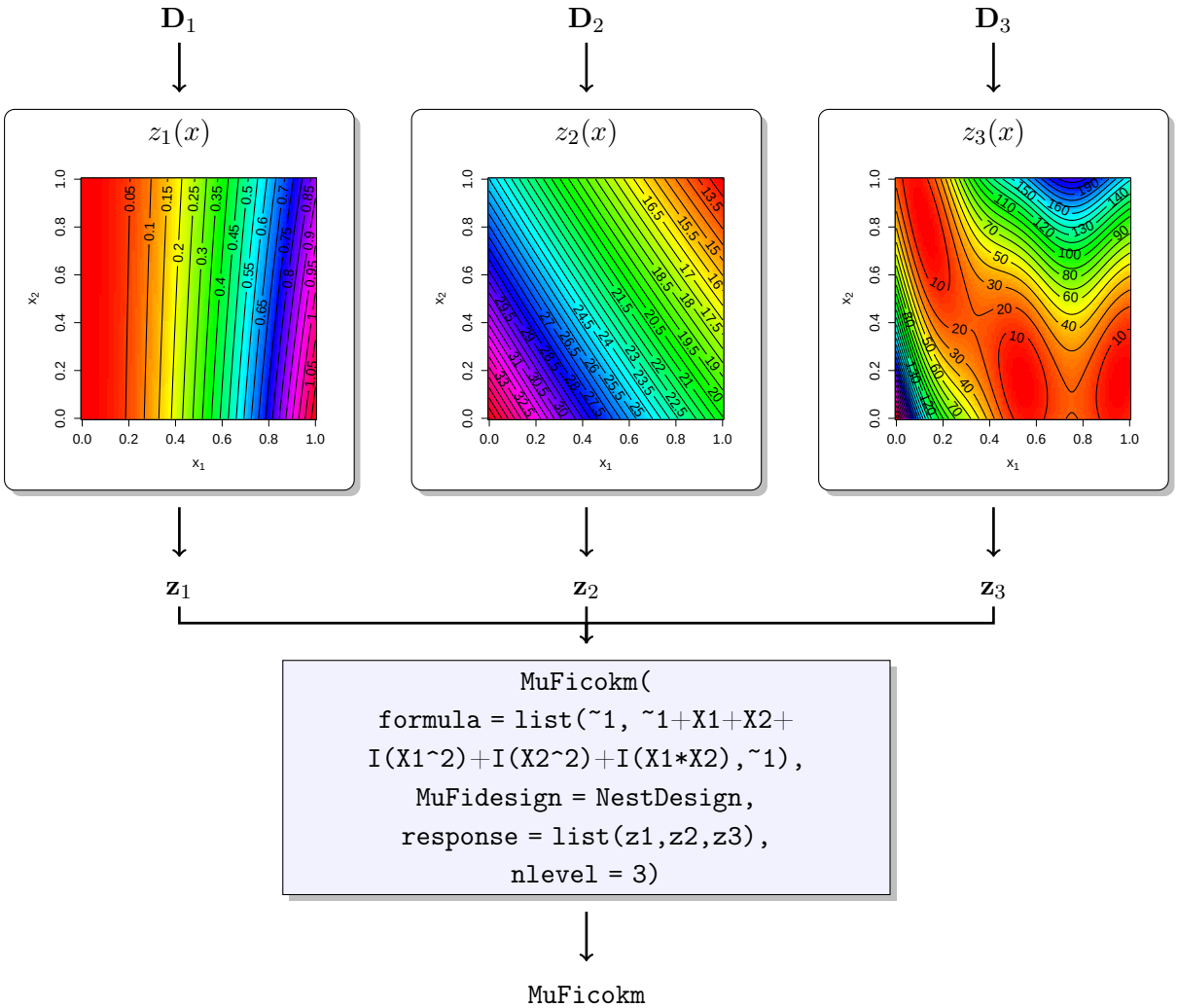
$\mathbf{D}_3$: `ExtractNestDesign(NestDesign,3)`

4.6.2 Building a multi-fidelity co-kriging models with MuFiCokriging R package

Let us consider the three following functions:

$$\begin{cases} z_1(x) = \left(\frac{5(15x_1-5)^2}{4\pi^2}\right)^2 - 2\left(15x_2 + \frac{5(15x_1-5)}{\pi} - 6\right)\frac{5(15x_1-5)^2}{4\pi^2} \\ z_2(x) = z_1(x) + \left(15x_2 + \frac{5(15x_1-5)}{\pi} - 6\right)^2 \\ z_3(x) = z_2(x) + 10\left(1 - \frac{1}{8\pi}\right)\cos(15x_1 - 5) + 10 \end{cases} . \tag{4.31}$$

The function $z_3(x)$ corresponds to the Branin's function where the inputs $x = (x_1, x_2) \in [0,1]^2$ are normalized (see [Jones et al., 1998]). We consider the nested experimental design sets building in the previous section and representing by the object `NestDesign` of class "NestDesign". First, we have to obtain the observations of $z_1(x)$, $z_2(x)$, $z_3(x)$ at points in $\mathbf{D}_1$, $\mathbf{D}_2$, $\mathbf{D}_3$. The contour plot of the three functions are illustrated in the following sketch.



The procedure `MuFicokm` is used to build a multi-fidelity co-kriging model. It returns an object of class `MuFicokm` representing the model definition including the parameter estimations. Its main arguments are the following ones:

`formula`: an object of class `formula` allowing to define the regression functions $\mathbf{f}_t(x)$. Example of scripts corresponding to a regression function $\mathbf{f}(x) = (1,\, x_1,\, x_2,\, x_1x_2)$:

```
> names(data.frame(NestDesign$PX))
[1] "X1" "X2"
> formula = ~1 + X1 + X2 + I(X1*X2)
```

`MuFidesign`: an object of class `NestDesign` representing the nested experimental design sets.

`response`: a list of vector representing the observations $(\mathbf{z}_t)_{t=1,\ldots,s}$.

`nlevel`: an integer representing the number of levels s.

`formula.rho`: an object of class `formula` allowing to define the regression functions $\mathbf{g}_t(x)$ for the adjustement coefficients $(\rho_{t-1}(x))_{t=2,\dots,s}$.

`covtype`: the type of covariance matrix for $Z_1(x)$ and $(\delta_t(x))_{t=2,\dots,s}$. The available kernels are (see Subsection 1.4.2):

`"gauss"`: Squared Exponential covariance function

`"matern5_2"`: 5/2-Matérn covariance function

`"matern3_2"`: 3/2-Matérn covariance function

`"exp"`: exponential covariance function

`"powexp"`: γ-exponential covariance function

In a simple co-kriging case, the user can fix the values of the parameters and hyper-parameters with the following arguments:

`coef.trend`: a list of vectors containing the values of $(\boldsymbol{\beta}_t)_{t=1,\dots,s}$.

`coef.rho`: a list of vectors containing the values of $(\boldsymbol{\beta}_{\rho_{t-1}})_{t=2,\dots,s}$.

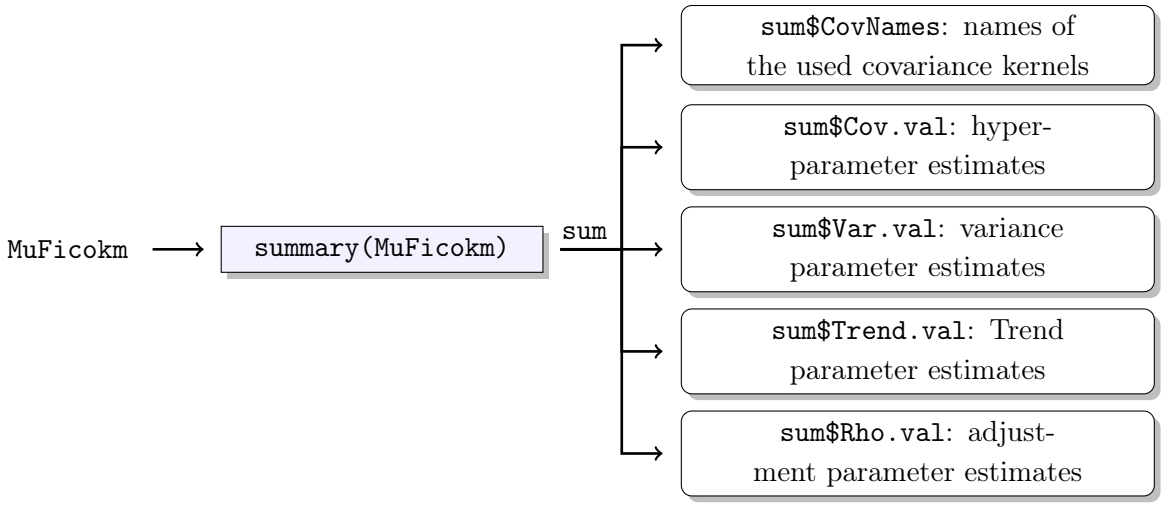
`coef.var`: a list of positive reals containing the values of $(\sigma_t^2)_{t=1,\dots,}$,

`coef.cov`: a list of vectors with strictly positive components representing the values of $(\theta)_{t=1,\dots,s}$.

`nugget`: a list of reals representing the "nugget effect" for each level of code.

`estim.method`: an optional argument allowing to indicate which method is used for the estimation of $(\theta)_{t=1,\dots,s}$. Two choice are possible: `"EML"` corresponds to the maximum restricted likelihood estimation; `"LOO"` corresponds to the Leave-One-Out cross validation estimation with the squared error loss function.

If they are set to `NULL` the parameters are estimated thanks to the method presented in Subsection 4.2.3 with non-informative prior distributions. The values of the estimates correspond to the posterior means of the regression, adjustment and variance parameters. Furthermore, the hyper-parameters are estimated by minimizing the negative restricted log-likelihood or the Leave-One-Out cross validation squared error (see Subsection 1.3.3). The remaining arguments are essentially used to control the optimization procedure for the hyper-parameter estimations. After obtaining the multi-fidelity co-kriging model `MuFicokm`, the user can have a summary of the model thanks to the `summary` procedure:



4.6.3 Predictive means and variances at new points

At this stage, we have built a multi-fidelity co-kriging model from $(\mathbf{D}_t)_{t=1,\ldots,s}$ and $(\mathbf{z}_t)_{t=1,\ldots}$. We are know interested in predicting $z_3(x)$ at new points $\mathbf{X} = \{x^1, \ldots, x^n\}$. The predictive mean and variance are implemented in the `predict` procedure which has three arguments:

`object`: an object of class `MuFicokm`.

`newdata`: a matrix representing the points $\mathbf{X}$ where to perform the predictions.

`type`: a character string indicating the type of used multi-fidelity co-kriging.

- `"SK"`: simple co-kriging, i.e. when trend and adjustment parameters are known.
- `"UK"`: universal co-kriging, i.e. when trend and adjustment parameters are estimated.

As stated in Subsection 4.2.1, once the multi-fidelity predictive means and variances are built for $z_s(x)$, the ones for $(z_t(x))_{t=1,\ldots,s-1}$ are also available. The outputs of the `predict` procedure are the following ones:

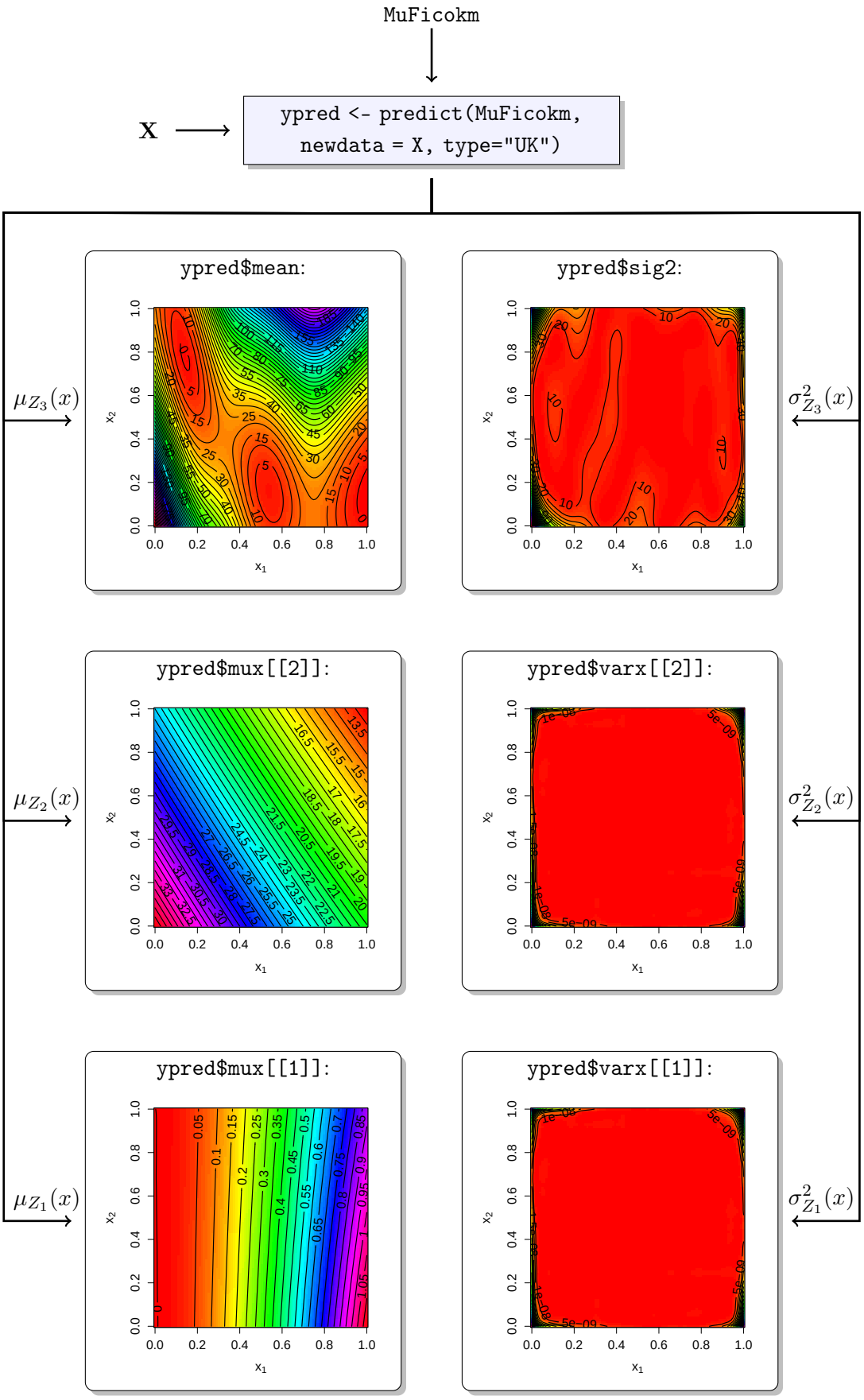
`mean`: the predictive mean for $z_s(x)$.

`sig2`: the predictive variance for $z_s(x)$.

`mux`:a list of predictive means. the ith element of the list corresponds to the predictive mean of $z_i(x)$, $i = 1, \ldots, s$.

`varx`:a list of predictive variances. the ith element of the list corresponds to the predictive variance of $z_i(x)$, $i = 1, \ldots, s$.

The procedure `predict` can also provide the predictive covariance matrix at points in $\mathbf{X}$ with the optional arguments `cov.compute = TRUE`. The resulting covariance at level s is obtained with the output `C` and the ones for levels $t = 1, \ldots, s$ are obtained with the output `CovMat`.


MuFicokm
X
ypred <- predict(MuFicokm,
newdata = X, type="UK")
ypred$mean:
ypred$sig2:
ypred$mux[[2]]:
ypred$varx[[2]]:
ypred$mux[[1]]:
ypred$varx[[1]]:
$\mu_{Z_3}(x)$
$\sigma^2_{Z_3}(x)$
$\mu_{Z_2}(x)$
$\sigma^2_{Z_2}(x)$
$\mu_{Z_1}(x)$
$\sigma^2_{Z_1}(x)$
x1
x2

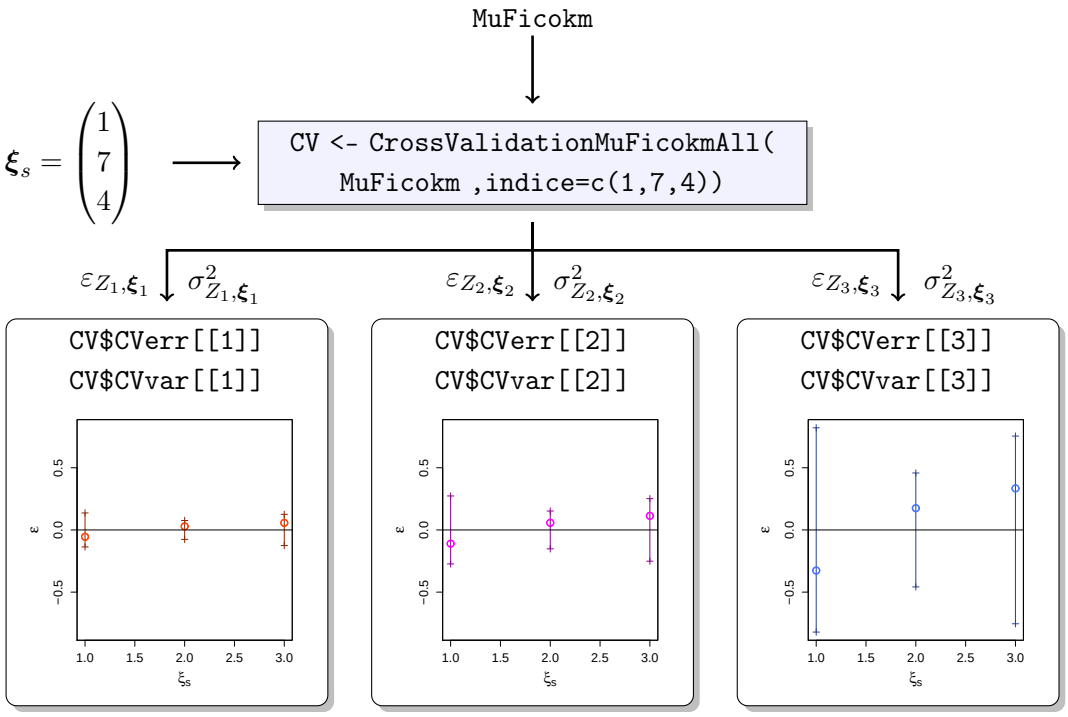


Figure 4.6: Example of CV prediction errors when we remove the three points of $\mathbf{D}_s$ indexed by $\boldsymbol{\xi}_s = (1, 7, 4)$. The confidence intervals equal twice the CV predictive standard deviations.

4.6.4 Cross validation procedures

The fast cross-validation method presented in Section 4.4 is implemented in the procedure `CrossValidationMuFicokmAll`. As stated in the application 4.5, the practitioner can either decide during the CV procedure to remove points from all code levels or from levels $s, \ldots, t$ with $0 < t \leq s$. The `CrossValidationMuFicokmAll` procedure computes all these cases. Its arguments are an object of class `MuFicokm` representing the multi-fidelity co-kriging model and a vector of integer `indice` indicating the index of the points that we remove from $\mathbf{D}_s$ for the CV procedure. Then, the procedure outputs `CVerrall`, `CVvarall` and `CVCovall` provide the CV predictive errors, variances and covariances when we remove the points from all code levels. Furthermore, the outputs `CVerr`, `CVvar` and `CVCov` are lists where the tth elements correspond to the cross validation predictive means, variances and covariances at level t.

4.7 Conclusion

We have presented in this chapter a recursive formulation for a multi-fidelity co-kriging model. This model allows us to build surrogate models using data from simulations of different levels of fidelity.

The strength of the suggested approach is that it considerably reduces the complexity of the co-kriging model while it preserves its predictive efficiency. Furthermore, one of the most

important consequences of the recursive formulation is that the construction of the surrogate model is equivalent to build s independent kriging models. Consequently, we can naturally adapt results of kriging to the proposed co-kriging model.

First, we present a Bayesian estimation of the model parameters which provides closed form expressions for the parameters of the posterior distributions. We note that, from these posterior distributions, we can deduce the maximum likelihood estimates of the parameters. Second, thanks to the joint distributions of the parameters and the recursive formulation, we can deduce closed form formulas for the mean and covariance of the posterior predictive distribution. Due to their similarities with the universal kriging equations, we call these formulas the universal co-kriging equations. Third, we present closed form expressions for the cross-validation equations of the co-kriging surrogate model. These expressions reduce considerably the complexity of the cross-validation procedure and are derived from the ones of kriging model that we have extended.

The suggested model has been successfully applied to a hydrodynamic code. We also present in this application a practical way to design the experiments of the multi-fidelity model.

Chapter 5

Sequential design for kriging and Multi-fidelity co-kriging models

Usually, in real applications, two stages are performed to surrogate a computer code with a kriging model. The first one consists in building a kriging model from simulations coming from an initial experimental design set. Many methods exist to build the initial design set, in order to ensure appropriate space filling properties, the reader is referred to [Fang et al., 2006] for a non-exhaustive review of them. The second stage consists in adding simulations sequentially at new design points which complete the initial set. The selection of the new points are usually based on criteria to improve the global accuracy of the kriging model and this will be our goal in this chapter. To be complete, we mention that sequential kriging has also been widely used in optimization (see [Jones et al., 1998], [Picheny et al., 2012]) and to estimate probabilities of failure [Bect et al., 2012]

Kriging models are a powerful tool to enrich an experimental design set since it provides through the kriging variance - also called predictor Mean Squared Error (MSE) or variance of prediction - an estimator of the model MSE. Kriging literature provides lot of criteria usually based on the kriging variance for sequentially design the experiments [Sacks et al., 1989b]. Furthermore, [Bates et al., 1996] and [Picheny et al., 2010] propose more efficient criteria by considering the Integrated MSE (IMSE). It consists in integrating the mean value of the MSE integrated over the input parameter space. We note though that the IMSE can be computationally expensive to assess, especially when the dimension increases. Although these criteria are efficient for many cases, they can suffer from an important flaw when the accuracy of the kriging model is not homogeneous over the input parameter space. Indeed, the kriging variance is determined by the distances between prediction and design points but not by the real model errors. To fix this important flaw, we can use the Empirical IMSE suggested in [Sacks et al., 1989b] which evaluates the model errors through a test set. Nevertheless, in a complex computer code framework, it could be too expensive to consider an external test set and cross-validation (CV) based criteria are more significant. As an illustration [Kleijnen and van Beers, 2004] and [van Beers and Kleijnen, 2008] combine a bootstrapping and a CV procedure to evaluate the predictor MSE. Although this method improves the classical

approach, it still does not take into account the real model errors. We note that a strength of the method proposed by [Kleijnen and van Beers, 2004] is that it can be applied to other types of surrogate models than the kriging one.

The first focus of this chapter is on sequential design to improve the accuracy of a kriging model. In particular, we propose new criteria combining the kriging variance and the Leave-One-Out CV (LOO-CV) errors. The CV errors allow for focusing the new observations on regions where the real model errors are large. Furthermore, thanks to the equations presented in [Dubrule, 1983] and in Subsection 1.3.3, the LOO-CV equations are fast to compute and thus the suggested approach is not expensive.

Defining sequential design strategies in a multi-fidelity framework is also of interest and is still an open problem. A method based on nested Latin hypercube designs is suggested in [Xiong and Qian, 2012]. However, it does not allow for adding a small number of additional simulations (e.g. it cannot perform an one step at-a-time sequential design) and it does not take into account the accuracies of the coarse code versions and the time ratios between two code levels.

The second focus of this chapter is on sequential design for co-kriging model. We adapt the new strategies suggested for the kriging model to the multi-fidelity co-kriging one. The strength of the proposed extensions is that they not only provide the new points where to perform new simulations but they also determine which version of code is worth being simulated. These new criteria take into account the computational time ratios between code versions. They are based on a proxy of the IMSE reduction and on the recursive formulation presented in Chapter 4 giving the contribution of each code on the total variance of the model. We note that sequential design in a multi-fidelity framework has also been applied for optimization purposes [Forrester et al., 2007] and [Huang et al., 2006].

The chapter is organized as follows. First, we present our CV-based sequential design strategies. We illustrate these strategies in tabulated functions. Secondly, we present the extensions of the previous strategies for the multi-fidelity co-kriging model. Finally, we apply the sequential co-kriging approach to a mechanical example.

5.1 Kriging models and sequential designs

In this section, we briefly introduce the kriging equations presented in Chapter 1 and some of its classical sequential design criteria. Then, we will present our sequential strategies to enhance kriging models considering the region with large LOO-CV errors.

5.1.1 The Kriging model

Let us denote by $z(x)$ the output of the code that we want to surrogate at point $x \in Q \subset \mathbb{R}^d$. In our framework, we set that the prior knowledges about the code is modeled by a Gaussian process $Z_0(x)$ with mean of the form $m_0(x) = \mathbf{f}'(x)\boldsymbol{\beta}$ and with covariance function $k_0(x, \tilde{x}) = \sigma^2 r\left(x, \tilde{x}; \boldsymbol{\theta}\right)$. We use the subscript 0 to emphasize that at this stage no observations are considered. Using the same notation as in Chapter 1 Subsection 1.2.2, the kriging equations are given by the distribution of the Gaussian process $Z_0(x)$ conditioned by its known values

$\mathbf{z}^n$ at points in $\mathbf{D}$:

$$Z_n(x) \sim [Z_0(x)|Z_0(\mathbf{D}) = \mathbf{z}^n] = \mathrm{GP}\left(m_n(x), k_n(x, \tilde{x})\right), \tag{5.1}$$

where:

$$m_n(x) = \mathbf{f}'(x)\hat{\boldsymbol{\beta}} + \mathbf{r}'(x)\mathbf{R}^{-1}(\mathbf{z}^n - \mathbf{F}\hat{\boldsymbol{\beta}}) \tag{5.2}$$

and:

$$k_n(x, \tilde{x}) = \sigma^2 \left(r(x, \tilde{x}) - \begin{pmatrix} \mathbf{f}'(x) & \mathbf{r}'(x) \end{pmatrix} \begin{pmatrix} 0 & \mathbf{F}' \\ \mathbf{F} & \mathbf{R} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{f}(\tilde{x}) \\ \mathbf{r}(\tilde{x}) \end{pmatrix} \right), \tag{5.3}$$

where $\hat{\boldsymbol{\beta}} = (\mathbf{F}'\mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{F}'\mathbf{R}^{-1}\mathbf{z}^n$ is the usual least-squares estimate of $\boldsymbol{\beta}$ (see Section 1.3). The model parameters σ^2 and $\boldsymbol{\theta}$ can be estimated by maximizing their Likelihood (see [Santner et al., 2003] and Subsection 1.3.2) or with a cross-validation procedure (see [Rasmussen and Williams, 2006], [Bachoc, 2013] and Subsection 1.3.3). Furthermore, the Maximum restricted Likelihood Estimate (MLE) of σ^2 is given by $\hat{\sigma}^2 = (\mathbf{z}^n - \mathbf{F}\hat{\boldsymbol{\beta}})'\mathbf{R}^{-1}(\mathbf{z}^n - \mathbf{F}\hat{\boldsymbol{\beta}})/(n-p)$. We note that the kriging predictive mean and covariance are denoted by $m_n(x)$ and $k_n(x, \tilde{x})$ to emphasize their dependence on the number of observations n.

1 point at-a-time Sequential design

Now, let us suppose that we want to add a new point x_{n+1} in $\mathbf{D}$ in order to enhance the accuracy of the kriging model. From the kriging variance $k_n(x, x)$ - representing the model MSE - some sequential design methods have been derived [Sacks et al., 1989b], [Bates et al., 1996] and [Picheny et al., 2010]. A first one consists in adding x_{n+1} where the kriging variance is the largest (see [Sacks et al., 1989b]):

$$x_{n+1} = \arg\max_x k_n(x, x). \tag{5.4}$$

However, as presented in [Kleijnen and van Beers, 2004], its performance is poor. Then, it has been improved with a criterion which consists in adding the new point which leads the most important IMSE reduction (see [Bates et al., 1996] and [Picheny et al., 2010]):

$$x_{n+1} = \arg\max_x \int_{u \in Q} k_n(u, u) - k_{n+1}(u, u)\, du, \tag{5.5}$$

where

$$k_{n+1}(u, \tilde{u}) = \sigma^2 \left(r(u, \tilde{u}) - \begin{pmatrix} \mathbf{f}(u) \\ \mathbf{r}(u) \\ r(u, x) \end{pmatrix}' \begin{pmatrix} 0 & \mathbf{F}' & \mathbf{f}(x) \\ \mathbf{F} & \mathbf{R} & \mathbf{r}(x) \\ \mathbf{f}'(x) & \mathbf{r}'(x) & 1 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{f}(\tilde{u}) \\ \mathbf{r}(\tilde{u}) \\ r(\tilde{u}, x) \end{pmatrix} \right).$$

Here, the covariance kernels $k_{n+1}(u, \tilde{u})$ corresponds to the one of the distribution of the Gaussian process $Z_n(u)$ (5.1) conditioned by a new observation at x. Furthermore, Equation (5.3) shows that the kriging variance does not depend on the observations if we consider known the parameters σ^2 and $\boldsymbol{\theta}$. Therefore, in that case, $k_{n+1}(u, u)$ can be computed without new simulations. We denote by MinIMSE this criterion. Finally, we also consider the criterion

presented by [Kleijnen and van Beers, 2004] using a Jackknife estimator for the predictor variance. Its principle is the following one. Let us consider $m_{n,-i}(x)$ the kriging mean built without the ith observation, the Jackknife variance is given by:

$$s^2_{jack}(x) = \frac{1}{n(n-1)} \sum_{i=1}^{n} (\tilde{z}_i - \bar{\tilde{z}})^2, \tag{5.6}$$

where $\tilde{z}_i = nm_n(x) - (n-1)m_{n,-i}(x)$ and $\bar{\tilde{z}} = \sum_{i=1}^{n} \tilde{z}_i/n$. Then, we consider candidate points coming from a maximin LHS Design [Fang et al., 2006] and we add the one which maximizes the Jackknife variance. We denote by KleiCrit this criterion.

q points at-a-time Sequential design

There is a natural way to extend these algorithms when the simulations can be performed simultaneously. Indeed, the covariance kernel $k_{n+1}(x, \tilde{x})$ of the Gaussian process $Z_n(x)$ conditioned by the new observation at point x_{n+1} can be computed without knowing $z(x_{n+1})$ when we consider the model parameters σ^2 and $\boldsymbol{\theta}$ as known. Then, from $k_{n+1}(x, \tilde{x})$, we can find a new point x_{n+2} where to perform a new simulation using the same criterion as in Equation (5.5) and the kernel $k_{n+2}(x, \tilde{x})$. Thus, considering the parameters σ^2 and $\boldsymbol{\theta}$ as known (they are fixed to their estimated values), we can determine with this procedure q good locations where to perform simulations. We call this method the liar sequential kriging. This idea is also extended in the framework of kriging-based optimization in [Ginsbourger et al., 2010].

5.1.2 LOO-CV based strategies for kriging sequential design

We present in this subsection new sequential-kriging strategies. The main difference between these new strategies and the previous ones is that they take into account the real model errors through the LOO-CV equations.

The proposed sequential methods is based on Proposition 4.3 for the univariate case $s = 1$. This proposition provides a powerful tool to compute the LOO-CV predictive means and variances. Indeed, several elements of the equations presented in Proposition 4.3 have been already computed during the model construction (e.g. the inverse of the matrix $\mathbf{R}$). Consequently, the LOO-CV equations are fast to compute and can be easily recomputed at each step of the sequential strategy. We note that the original result which is the estimation of $\sigma^2_{1,-i}$ is of great importance. Indeed, as we use the value of $k_{n,-i}(x_i, x_i)$, $x_i \in \mathbf{D}$, strongly depending on $\sigma^2_{1,-i}$ in our forthcoming developments, it is important to well estimate it. We note that $k_{n,-i}(x_i, x_i)$ corresponds to the covariance kernel of the distribution of $Z_0(x)$ conditioned by the known value $\mathbf{z}^n$ minus the ith one and $\sigma^2_{1,-i}$ is the restricted maximum likelihood estimate of σ^2 performed without the ith observation of $\mathbf{z}^n$.

Now, let us denote by $\mathbf{e}^2_{\text{LOO-CV}} = \left[((z(x_i) - m_{n,-i}(x_i))^2\right]_{i=1,\dots,n}$ the vector of the LOO-CV squared errors and $\mathbf{s}^2_{\text{LOO-CV}} = [k_{n,-i}(x_i, x_i)]_{i=1,\dots,n}$ the vector of the LOO-CV variances with $m_{n,-i}$ the kriging predictive mean building without the ith observation of $\mathbf{z}^n$ and $(x_i)_{i=1,\dots,n} \in \mathbf{D}$. Furthermore, let us consider the Voronoi cells $(V_i)_{i=1,\dots,n}$ associated with the points $(x_i)_{i=1,\dots,n}$:

$$V_i = \{x \in Q,\ ||x - x_i|| \leq ||x - x_j||,\ \forall j \neq i\},\ i, j = 1, \dots, n. \tag{5.7}$$

In the remainder of this section, we present two strategies to sequentially add simulations which use $\mathbf{e}^2_{\text{LOO-CV}}$, $\mathbf{s}^2_{\text{LOO-CV}}$ and V_i. The intuitive idea of the suggested criteria is to enhance the predictive variance in the locations where the LOO-CV errors are important.

LOO-CV-based 1 point at-a-time Sequential design

Let us denote by x_{n+1} the new point that we want to add to $\mathbf{D}$. We consider the point solving the following problem:

$$x_{n+1} = \arg\max_x \left\{ k_n(x,x) \left(1 + \sum_{i=1}^{n} \frac{[\mathbf{e}^2_{\text{LOO-CV}}]_i}{[\mathbf{s}^2_{\text{LOO-CV}}]_i} \mathbf{1}_{x \in V_i} \right) \right\}, \tag{5.8}$$

where $\mathbf{1}$ stands for the indicator function.

This criterion considers the predictor MSE $k_n(x,x)$ adjusted with the LOO-CV errors and variances. For equivalent $k_n(x,x)$, the criterion favors the points close to an experimental design point with large LOO-CV errors. Furthermore, if two points are in the same Voronoi cell, the one with the largest predictor MSE is considered. Therefore, a sequential strategy with this criterion focus on the regions of Q where the LOO-CV errors are the largest. We note that the standardization with $\mathbf{s}^2_{\text{LOO-CV}}$ is important since it is not necessary to enlarge the predictor MSE in the regions where it is well or over estimated. As example, $[\mathbf{e}^2_{\text{LOO-CV}}]_i \ll [\mathbf{s}^2_{\text{LOO-CV}}]_i$ means that the kriging variance is over-estimated around the point x_i, i.e. $k_n(x,x)$ is too large for $x \in V_i$. In that case, the standardization with $[\mathbf{s}^2_{\text{LOO-CV}}]_i$ implies that $\sum_{i=1}^{n} \frac{[\mathbf{e}^2_{\text{LOO-CV}}]_i}{[\mathbf{s}^2_{\text{LOO-CV}}]_i} \mathbf{1}_{x \in V_i} \approx 0$ for $x \in V_i$ and thus the term in Equation (5.8) is approximately equal to $k_n(x,x)$.

We illustrate in Figure 5.1 the adjusted variance presented in Equation (5.8) and the classical kriging variance (5.3) in a 1-dimensional example. The considered function is $f(x) = (\sin(7x) + \cos(14x))x^2 \exp(-4x)$, $x \in [0,4]$. We use a kriging model with a 5/2-Matérn kernel with $\sigma^2 = 1.10^{-3}$ and $\theta = 1$ and the experimental design set is a regular grid of 8 points between 0 and 4. We see in Figure 5.1 that the kriging model is not accurate in the domain $[0,2]$ where the function variations are important and the adjusted kriging variance (5.8) focuses on that region.

As illustrated in Figure 5.1, the adjusted kriging variance allows for taking into account the LOO-CV error in a sequential procedure focusing on the large error domain. Nevertheless, it does not entirely fix the issue of the relevance of $k_n(x,x)$ to represent the model error. Indeed, our criterion enlarges the kriging variance around points where $k_n(x,x)$ is under-estimated but it does not reduce it at locations where it is over-estimated. However, it gives more information about the relevance of $m_n(x)$ since it highlights the regions where it is not accurate. Furthermore, it also aids in the interpretation of $k_n(x,x)$ since it emphasizes whether it is under-estimated or not.

An efficient method to solve the problem in Equation (5.8) is to use an evolutionary algorithm coupled with a descent algorithm. Indeed, when $x \in V_i$ we have to solve the problem $\arg\max_{x \in V_i} k_n(x,x)$. This can be performed with classical optimization methods (e.g. Conjugate gradient, Newton,...). Then, we can use an evolutionary algorithm to explore

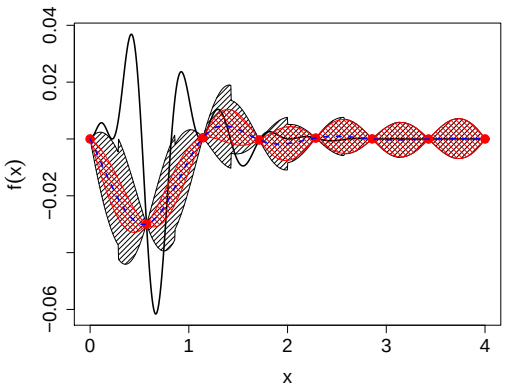


Figure 5.1: Illustration of the adjusted kriging variance in a 1-dimensional example. The solid thick line represents the true function, the dashed thick line represents the kriging mean, the bullets represent the observations and the dashed areas represent the kriging mean plus or minus twice the kriging standard deviation and adjusted standard deviation. We see that the kriging variance is enlarged at the domain where the function variations are important.

different cells $(V_i)_{i=1,\dots,n}$. Furthermore, for low-dimensional problems (i.e. $d < 10$), a Monte-Carlo method can be efficiently used as exploratory algorithm. We note that it is not necessary to compute the Voronoi tessellation since the criterion only requires to determine in which Voronoi cells lies a given point $x \in Q \subset \mathbb{R}^d$. This is computationally simple and cheap even for high dimension d.

LOO-CV-based q points at-a-time Sequential design

We extend here the previous criterion for a q points at-a-time sequential design. First, we emphasize that the liar sequential kriging is not relevant for this new criterion. Indeed, conditioning on model parameters, with a liar method we can compute the kriging variances $(k_{n+i}(x,x))_{i=1,\dots,q}$ but not the LOO-CV equations. Therefore, we use another strategy to propose q new locations where to perform the simulations. This approach is proposed in [Dubourg et al., 2011] in a different framework. The idea of the suggested method is to select the q best points with respect to the criterion (5.8) from N candidate points. These N candidate points are chosen with the following algorithm.

1. Generate N_{MCMC} samples with respect to the probability density function proportional to $k_n(x,x)$ with a suitable Markov Chain Monte Carlo (MCMC) technique [Robert and Casella, 2004].
2. Extract from these samples N representative points with a N-means clustering technique [MacQueen, 1967].

As presented in [Dubourg et al., 2011] the use of this algorithm to select N candidate points in a kriging framework is efficient. Indeed, it allows us to concentrate the points at the modes of the kriging variance. In the proposed strategy, we always take $N \geq q$

and we choose from the N cluster centers $(C_i)_{i=1\ldots,N}$ the q points where $k_{n,\text{adj}}(x,x) = k_n(x,x)\left(1+\sum_{i=1}^{n}\frac{[\mathbf{e}^2_{\text{LOO-CV}}]_i}{[\mathbf{s}^2_{\text{LOO-CV}}]_i}\mathbf{1}_{x\in V_i}\right)$ is the largest. For the MCMC procedure, we use a Metropolis-Hastings (M-H) algorithm with a Gaussian jumping distribution. It is centered on the last sample point and has a standard deviation such that the acceptance rate is around 30% (see [Robert and Casella, 2004]). Furthermore, we set N_{MCMC} such that $N_{\text{MCMC}} \gg N$. For the N-means procedure, we choose the value of N with respect to the following criterion:

$$\max_{N\geq q}\ \min_{x\in(C_i)_{i=1,\ldots,N}} k_n(x,x), \tag{5.9}$$

where $(C_i)_{i=1\ldots,N}$ are the cluster centers. This criterion prevents from having a cluster center in a region where the kriging variance is close to zero. Furthermore, if the number of clusters is too high, the cluster centers get away from the modes and consequently the value of $\min_{x\in(C_i)_{i=1\ldots,N}} k_n(x,x)$ decreases. Therefore, this criterion also prevents from having a number of clusters too large. In practice, we choose N on a finite sequence from q to $2n$ where n is the number of observations and we run the N-means procedure several times for each N. Then, we select the cluster centers minimizing (5.9). We note that the MCMC plus N-means procedure requires careful implementation and appropriate diagnostics. For the N-means procedure, we use the algorithm suggested by [Hartigan and Wong, 1979] with complexity $\mathcal{O}(NN_{\text{MCMC}})$. For the M-H procedure we use the R CRAN Package mcmc. To avoid computational issues, one can extract the q-points from candidates generated with space-filling design techniques [Fang et al., 2006]. However, with this technique, the candidate points will not anymore be concentrated in the regions of high mean squared error and the method will be less efficient.

5.2 Sequential design in a multi-fidelity framework

In this section, we consider the multi-fidelity co-kriging model presented in Chapter 4 with constant scale factors $(\rho_{t-1})_{t=2,\ldots,s}$ and we extend the previous sequential design strategies in this framework. We note that, in a multi-fidelity framework, the search for the best locations where to run the code is not the only point of interest. Indeed, once the best locations are determined, we also have to decide which code level is worth being run. This will not only depend on the time-ratios between the code levels but also on the contribution of each code level to the total predictor MSE.

5.2.1 Multi-fidelity co-kriging models

Let us suppose that we want to surrogate a computer code output $z_s(x)$ and that coarse versions of this code $(z_t(x))_{t=1,\ldots,s-1}$ are available. These codes are sorted by order of fidelity from the less accurate $z_1(x)$ to the most accurate $z_{s-1}(x)$. We consider the universal multi-fidelity co-kriging equations presented in Section 4.3 with constant scale factors $(\rho_{t-1})_{t=2,\ldots,s}$.

Thus, using the same notation as in Chapter 4 Section 4.2, the predictive mean $\mu^t_{n_t}(x)$ and variance $k^t_{n_t}(x,\tilde{x})$ at level $t=2,\ldots,s$ is given by the following equations:

$$\mu^t_{n_t}(x) = \hat{\rho}_{t-1}\mu^{t-1}_{n_{t-1}}(x) + \mathbf{f}'_t(x)\hat{\boldsymbol{\beta}}_t + \mathbf{r}'_t(x)\mathbf{R}_t^{-1}(\mathbf{z}_t - \mathbf{F}_t\hat{\boldsymbol{\beta}}_t - \hat{\rho}_{t-1}z_{t-1}(\mathbf{D}^t)) \tag{5.10}$$

and:

$$k^t_{n_t}(x,\tilde{x}) = \hat{\sigma}^2_{\rho_{t-1}} k^{t-1}_{n_{t-1}}(x,\tilde{x}) + \sigma^2_t \left(r_t(x,\tilde{x}) - \begin{pmatrix} \mathbf{h}'_t(x) & \mathbf{r}'_t(x) \end{pmatrix} \begin{pmatrix} 0 & \mathbf{H}'_t \\ \mathbf{H}_t & \mathbf{R}_t \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{h}_t(\tilde{x}) \\ \mathbf{r}_t(\tilde{x}) \end{pmatrix} \right), \tag{5.11}$$

where $\begin{pmatrix} \hat{\rho}_{t-1} \\ \hat{\boldsymbol{\beta}}_t \end{pmatrix} = (\mathbf{H}'_t\mathbf{R}_t^{-1}\mathbf{H}_t)^{-1}\mathbf{H}'_t\mathbf{R}_t^{-1}\mathbf{z}_t$ is the least-squares estimates of $\begin{pmatrix} \rho_{t-1} \\ \boldsymbol{\beta}_t \end{pmatrix}$, $\hat{\sigma}^2_{\rho_{t-1}} = \hat{\rho}^2_{t-1} + \left[(\mathbf{H}'_t\mathbf{R}_t^{-1}\mathbf{H}_t)^{-1}\right]_{[1,1]}$ and $\mathbf{H}_t = [z_{t-1}(\mathbf{D}_t) \quad \mathbf{F}_t]$. Furthermore, the restricted maximum likelihood estimate of σ^2_t is given by

$$\hat{\sigma}^2_t = \frac{\left(\mathbf{z}_t - \mathbf{H}_t \begin{pmatrix} \hat{\rho}_{t-1} \\ \hat{\boldsymbol{\beta}}_t \end{pmatrix}\right)' \mathbf{R}_t^{-1} \left(\mathbf{z}_t - \mathbf{H}_t \begin{pmatrix} \hat{\rho}_{t-1} \\ \hat{\boldsymbol{\beta}}_t \end{pmatrix}\right)}{(n_t - p_t - 1)}.$$

We note that the predictive mean and variance at level t are denoted by $\mu^t_{n_t}(x)$ and $k^t_{n_t}(x,\tilde{x})$ to higlight their dependence of the number of observations n_t at level t.

The important property of this co-kriging model is that its MSE (5.11) provides through the term $\hat{\sigma}^2_{\rho_{t-1}} k^{t-1}_{n_{t-1}}$ the contribution of the code level $t-1$ to the total predictor MSE at level t, $t = 2, \ldots, s$. Therefore, it can allow us to determine which code level is worth being simulated at a new location x.

5.2.2 Sequential design for multi-fidelity co-kriging models

The aim of this subsection is to extend the sequential kriging strategies proposed in Subsection 5.1.2 to the suggested multi-fidelity co-kriging model. These extensions are based on the variance decomposition property presented in Subsection 5.2.1 in Equation (5.11) and on the cross-validation equations presented in Proposition 4.3. From them, the LOO-CV equations are fast to compute and consequently they can be used in a sequential procedure with a low computational cost. Furthermore, since the experimental design sets are nested, we state that during the LOO-CV procedure at level t, the points are removed from all code levels. Finally, from these equations, we can adjust the co-kriging variances $\left(k^t_{n_t}(x,\tilde{x})\right)_{t=1,\ldots,s}$ at each level using the same method as presented in Equation (5.8).

1 point at-a-time sequential co-kriging. First, let us consider x_{new} the point solving the problem:

$$x_{\text{new}} = \arg\max_x k^s_{n_s}(x,x). \tag{5.12}$$

Therefore, we want to compute a new simulation at point where the predictor MSE is maximal. Now, let us consider two successive code levels $t-1$ and t. The question of interest is to estimate which of these two code levels is worth being simulated.

First, thanks to Equation (5.11), we can deduce the contribution of each code levels to the predictor MSE. Let us define the following notation for $t = 2, \dots, s$:

$$\sigma^2_{\delta^t}(x) = \sigma^2_t \left(1 - \begin{pmatrix} \mathbf{h}'_t(x) & \mathbf{r}'_t(x) \end{pmatrix} \begin{pmatrix} 0 & \mathbf{H}'_t \\ \mathbf{H}_t & \mathbf{R}_t \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{h}_t(x) \\ \mathbf{r}_t(x) \end{pmatrix} \right) \tag{5.13}$$

and $\sigma^2_{\delta^1}(x) = k^1_{n_1}(x, x)$. Then, we have:

$$k^t_{n_t}(x, x) = \sum_{i=1}^{t} \sigma^2_{\delta^i}(x) \prod_{j=i}^{t-1} \hat{\sigma}^2_{\rho_j}. \tag{5.14}$$

Let us consider that the parameters $(\boldsymbol{\theta}_t)_{t=1,\dots,s}$ define the characteristic length-scales of the kernels $((r_t(x, \tilde{x}; \boldsymbol{\theta}_t))_{i=1,\dots,s}$ (see [Rasmussen and Williams, 2006] p.83 and Chapter 1 Section 1.4). Then, we can approximate the reduction of the IMSE after adding a new point x_{new} at level t with the following formula:

$$\text{IMSE}^t_{\text{red}}(x_{\text{new}}) = \sum_{i=1}^{t} \sigma^2_{\delta^i}(x_{\text{new}}) \prod_{j=i}^{t-1} \hat{\sigma}^2_{\rho_j} \prod_{m=1}^{d} \boldsymbol{\theta}^m_i, \tag{5.15}$$

with $\boldsymbol{\theta}_t = (\boldsymbol{\theta}^1_t, \dots, \boldsymbol{\theta}^d_t)$. Indeed, at each stage, $\sigma^2_{\delta^i}(x_{\text{new}}) \prod_{j=i}^{t-1} \hat{\sigma}^2_{\rho_j}$ represents the contribution of the bias $\delta^i(x)$ to the co-kriging variance and $\prod_{m=1}^{d} \boldsymbol{\theta}^m_i$ represents the volume of influence of x_{new} at level j. This criterion is justify by the fact that the reduction of IMSE^t defined by $\text{IMSE}^t = \int_Q \sigma^2_{\delta^t}(x)\, dx$ after adding a new point x_{new} has the same order of magnitude than $\sigma^2_{\delta^i}(x_{\text{new}})$ times the volume of influence $\prod_{m=1}^{d} \boldsymbol{\theta}^m_i$ of x_{new}.

We illustrate below the criterion (5.15) for a kriging model in dimension 2. Let us consider that we want to approximate the Branin-Hoo function (see [Jones et al., 1998]) from 12 observations. The considered experimental design set and the Branin-Hoo function are illustrated in Figure 5.2.

Figure 5.3 represents the kriging predictive mean and variance. The estimated characteristic length scales are $\theta_1 = 0.22$ and $\theta_2 = 0.65$ and the empirical IMSE is 1648. Let us consider that we want to simulate a new observation at point $x_{\text{new}} = (0.25, 0.5)$ (see Figure 5.3b), the approximation of the IMSE reduction given by the criterion in Equation (5.15) is 468.

Figure 5.4 represents the kriging predictive mean and variance after adding a new simulation at point $x_{\text{new}} = (0.25, 0.5)$. The obtained empirical IMSE is 1130. Therefore, the empirical uncertainty reduction equals $1648 - 1130 = 518$ which is close to the approximation given by Equation (5.15) which is 468.

Now, let us consider that the ratio of computational times between the codes $z_t(x)$ and $z_{t-1}(x)$ equals $B_{t/t-1}$. It means that the computational cost for running one simulation on $z_t(x)$ and one simulation on $z_{t-1}(x)$ (the experimental design sets must be nested) is the same as the one for running $1 + B_{t/t-1}$ simulations on $z_{t-1}(x)$ – i.e. for running $z_{t-1}(x)$ on $1 + B_{t/t-1}$ different points x_{new}. Therefore, it is worth running the code $z_{t-1}(x)$ if $(1 + B_{t/t-1})\text{IMSE}^{t-1}_{\text{red}}(x_{\text{new}}) > \text{IMSE}^t_{\text{red}}(x_{\text{new}})$, i.e. if the potential uncertainty reduction by running $1 + B_{t/t-1}$ times $z_{t-1}(x)$ is greater than the one when we run one simulation on $z_t(x)$ and

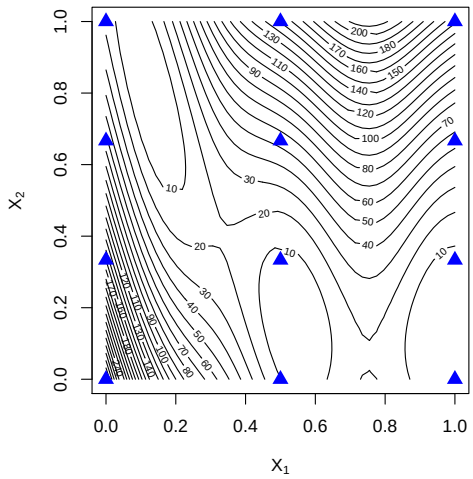


Figure 5.2: Contour plot of the Branin-Hoo function. The blue triangles represent the considered experimental design set.

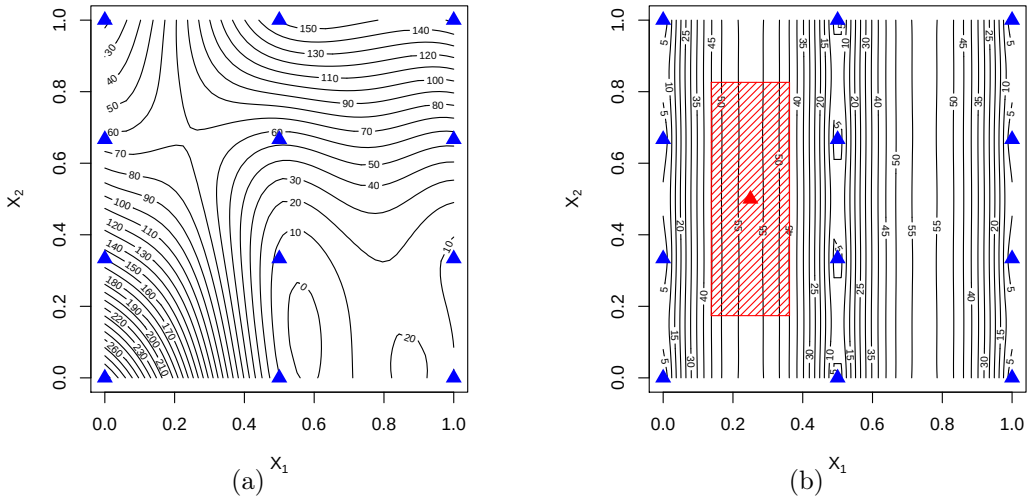


Figure 5.3: Figure (a) illustrates the kriging predictive mean and Figure (b) illustrates the kriging predictive variance. The blue triangles represent the experimental design set and the red triangle is the point $x_{\text{new}} = (0.25, 0.5)$ where to perform a new simulation. The filled rectangle is the volume of influence of x_{new} evaluated from $\theta_1 = 0.22$ and $\theta_2 = 0.65$.

one simulation on $z_{t-1}(x)$. From this criterion, we can deduce the following algorithm for an one at-a-time sequential co-kriging model taking into account both the computational ratios between the different code levels and the contribution of each level to the total co-kriging variance.

Remarks: Algorithm 2 evaluates for two successive code levels $t-1$ and t, which one is worth being simulated. It starts with the levels one and two, then two and three and so on.

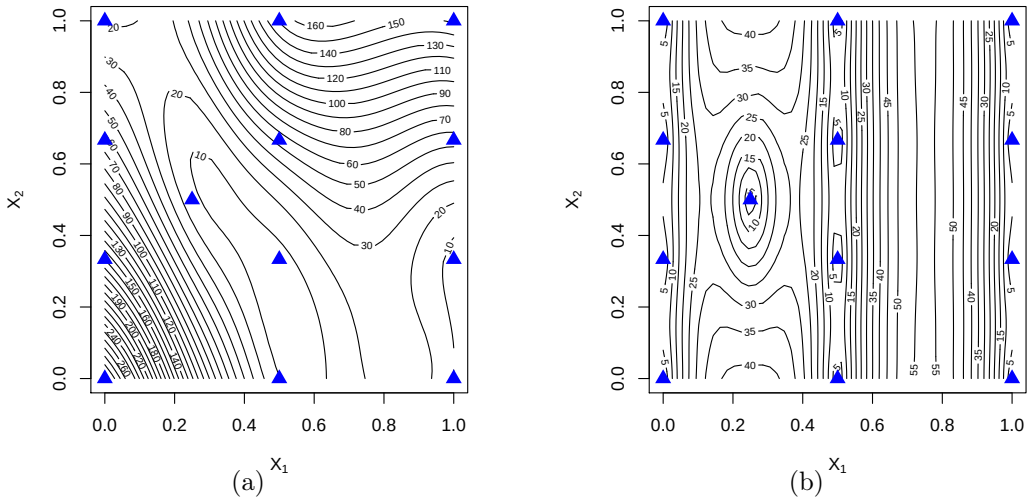


Figure 5.4: Figure (a) illustrates the kriging predictive mean and Figure (b) illustrates the kriging predictive variance. The blue triangles represent the experimental design set.

When it finds that the level $t-1$ is more promising than the level t, it stops the loop and simulate x_{new} at code levels $z_1(x), \ldots, z_{t-1}(x)$. Since the loop is defined from level 1 to level s, it favors simulations at low code levels. Therefore, it will tend to learn the coarse code versions before learning the accurate ones. We note that during the loop of Algorithm 2, the parameters are not re-estimated. In fact, they are re-estimated after adding the new point x_{new}. Moreover, the first test $\sigma^2_{\delta^t}(x_{\text{new}}) < \text{IMSE}^t$ checks if the code level t at point x_{new} is worth being run. Then, the test $\text{IMSE}^{t-1}_{\text{red}}(x_{\text{new}})/\text{IMSE}^t_{\text{red}}(x_{\text{new}}) > 1/(1+B_{t/t-1})$ evaluates which code levels between t and $t-1$ is the most promising. Finally, if we consider that the code level t is more promising than the code level $t-1$, we confront it to the following code level $t+1$. We note that Algorithm 2 is reiterated until a prescribed accuracy is reached or the computational time budget is spent.

1 point at-a-time sequential co-kriging with adjusted predictor MSE. From Proposition 4.3, Algorithm 2 and Equation (5.15), we can extend the criterion (5.8) to the multi-fidelity co-kriging model. Let us consider the following quantity:

$$\begin{aligned}\text{IMSE}^t_{\text{red,adj}}(x_{\text{new}}) &= \textstyle\sum_{i=1}^t \sigma^2_{\delta^i}(x_{\text{new}}) \prod_{j=i}^{t-1} \hat{\sigma}^2_{\rho_j} \prod_{m=1}^d \boldsymbol{\theta}_i^m \\ &\times \left(1 + \textstyle\sum_{j=1}^{n_i} \frac{\left(\varepsilon_{\text{LOO-CV},i}(x^i_j) - \hat{\rho}_{-j,i-1}\varepsilon_{\text{LOO-CV},i-1}(x^i_j)\right)^2}{\sigma^2_{\text{LOO-CV},i}(x^i_j) - \hat{\sigma}^2_{\rho_{i-1},-j}\sigma^2_{\text{LOO-CV},i-1}(x^i_j)}\right),\end{aligned} \tag{5.16}$$

where $\hat{\rho}_{-j,0} = 0$, $\hat{\rho}_{-j,i}$ corresponds to the first element of $\boldsymbol{\lambda}_{i,-j}$ in Proposition 4.3, $\hat{\sigma}^2_{\rho_0,-j} = 0$, $\hat{\sigma}^2_{\rho_{i-1},-j}$ corresponds to the element $[1,1]$ of the matrix $\boldsymbol{\Sigma}_{\rho,i,-j}$ in Proposition 4.3, x^i_j is the jth point of $\mathbf{D}_i$,

$$\varepsilon_{\text{LOO-CV},i}(x^i_j) = z_i(x^i_j) - \mu^i_{n_i,-j}(x^i_j),$$
$$\sigma^2_{\text{LOO-CV},i}(x^i_j) = k^i_{n_i,-j}(x^i_j, x^i_j),$$

$k^i_{n_i,-j}(x,\tilde{x})$ is the covariance kernel $k^i_{n_i}(x,\tilde{x})$ at level i built without the jth observation of $\mathbf{z}_i$, $\mu^i_{n_i,-j}$ is the predictive mean $\mu^i_{n_i,}$ at level i built without the jth observation of $\mathbf{z}_i$ and

Algorithm 2 One point at-a-time sequential co-kriging

```
Find x_new such that x_new = arg max_x k^s_{n_s}(x, x)
for t = 2, ..., s do
    if (σ²_{δ^t}(x_new) < IMSE^t) then
        Run z_{t-1}(x_new)
        end for
    else
        if (IMSE^{t-1}_red(x_new)/IMSE^t_red(x_new) > 1/(1 + B_{t/t-1})) then
            Run z_{t-1}(x_new)
            end for
        end if
    end if
end for
if (t = s) then
    Run z_t(x_new)
end if
```

$j = 1, \ldots, n_i$, $i = 1, \ldots, t$. In Equation (5.16), the kriging variance $\sigma^2_{\delta^i}(x)$, $i = 1, \ldots, t$, in Equation (5.14) is replaced with the adjusted kriging variance presented in Subsection 5.1.2. We note that $\left(\varepsilon_{\text{LOO-CV},i}(x^i_j) - \hat{\rho}_{-j,i-1}\varepsilon_{\text{LOO-CV},i-1}(x^i_j)\right)^2$ is the part of the LOO-CV squared error explained by the bias $\delta^i(x)$ and $\sigma^2_{\text{LOO-CV},i}(x^i_j) - \hat{\sigma}^2_{\rho_{i-1},-j}\sigma^2_{\text{LOO-CV},i-1}(x^i_j)$ is the corresponding LOO-CV predictive variance. To adapt the adjusted co-kriging variance in a multi-fidelity framework, we just have to replace $\text{IMSE}^t_{\text{red}}(x)$ with $\text{IMSE}^t_{\text{red,adj}}(x)$ in Algorithm 2 and $k^s_{n_s}(x,x)$ with:

$$\begin{aligned} k^s_{n_s,\text{adj}}(x,x) &= \textstyle\sum_{i=1}^s \sigma^2_{\delta^i}(x) \prod_{k=i}^{s-1} \hat{\sigma}^2_{\rho_k} \\ &\times \left(1 + \textstyle\sum_{j=1}^{n_i} \frac{\left(\varepsilon_{\text{LOO-CV},i}(x^i_j) - \hat{\rho}_{-j,i-1}\varepsilon_{\text{LOO-CV},i-1}(x^i_j)\right)^2}{\sigma^2_{\text{LOO-CV},i}(x^i_j) - \hat{\sigma}^2_{\rho_{i-1},-j}\sigma^2_{\text{LOO-CV},i-1}(x^i_j)}\right) . \end{aligned} \tag{5.17}$$

$k^s_{n_s,\text{adj}}(x,x)$ corresponds to $k^s_{n_s}(x,x)$ in Equation (5.14) where the kriging variance $\sigma^2_{\delta^j}(x)$ is replaced with its adjusted version. We highlight that thanks to Proposition 4.3, the elements $\varepsilon_{\text{LOO-CV},i}(x^i_j)$, $\sigma^2_{\text{LOO-CV},i}(x^i_j)$, $\hat{\sigma}^2_{\rho_{i-1},-j}$ and $\hat{\rho}_{-j,i-1}$ are fast to compute.

$(\mathbf{q^i})_{\mathbf{i=1,\ldots,s}}$ points at-a-time sequential co-kriging. In this paragraph, we propose an extension for the multi-fidelity model of the q points at-a-time sequential design presented in Subsection 5.1.2. Its principle is the following one. First, we select q^t new points for the code $z_t(x)$ with the method presented in Subsection 5.1.2 "LOO-CV based q points at-a-time Sequential design". Then, we consider these points as known for the code $z_{t-1}(x)$ and we select q^{t-1} new points for this code with the same method. We note that, as presented in Subsection 5.1.1, we can use a liar method to compute the new co-kriging variance without simulating $z_{t-1}(x)$ at the q^t new points. Finally, we repeat this procedure for all code levels from $z_{t-2}(x)$ to $z_1(x)$. At the end of the procedure, we have $\sum_{i=j}^t q^i$ new points at level j and we want to find the allocation $\{q^1, \ldots, q^t\}$ leading to the largest potential uncertainty reduction and under the constraint of a constant CPU time budget. We note the CPU time budget

$T = \sum_{j=1}^{t} \sum_{i=j}^{t} q^i \mathcal{T}^j$ where $(\mathcal{T}^i)_{i=1,\dots,s}$ represents the CPU times of codes $(z_i(x))_{i=1,\dots,s}$. Algorithm 3 presents the suggested q points at-a-time sequential co-kriging.

Algorithm 3 $(q^i)_{i=1,\dots,s}$ points at-a-time sequential co-kriging

1: Set the budget $T > 0$ and the allocation $\{q^1, \dots, q^t\}$ such that $\sum_{j=1}^{t} \sum_{i=j}^{t} q^i \mathcal{T}^j = T$
2: Set $(N^i_{\text{MCMC}})_{i=1,\dots,t}$ for the M-H procedures.
3: Generate N^t_{MCMC} samples distributed with respect to $k^t_{n_t}(x,x)$.
4: Find the N^t cluster centers $(C^t_i)_{i=1,\dots,N^t}$ such that $N^t = \max_{N \geq q^t} \min_{x \in (C^t_i)_i} k^t_{n_t}(x,x)$
5: Select from $(C^t_i)_{i=1,\dots,N^t}$ the q^t points $(x^t_{\text{new,i}})_{i=1,\dots,q_t}$ where $k^t_{n_t,\text{adj}}(x,x)$ is the largest.
6: **for** $m = t-1, \dots, 1$ **do**
7: Compute $k^m_{n_m + \sum_{i=m+1}^{t} q^i}(x,x)$ with the new points $\left((x^j_{\text{new,i}})_{i=1,\dots,q_t}\right)_{j=m+1,\dots,t}$
8: Generate N^m_{MCMC} samples with respect to $k^m_{n_m + \sum_{i=m+1}^{t} q^i}(x,x)$.
9: Find the N^m cluster centers $(C^m_i)_{i=1,\dots,N^m}$ such that $N^m = \max_{N \geq q^m} \min_{x \in (C^m_i)_i} k^m_{n_m + \sum_{i=m+1}^{t} q^i}(x,x)$
10: Select from $(C^m_i)_{i=1,\dots,N^m}$ the q^m points $(x^m_{\text{new,i}})_{i=1,\dots,q_m}$ where $k^m_{n_m + \sum_{i=m+1}^{t} q^i,\text{adj}}(x,x)$ is the largest.
11: **end for**

Algorithm 3 details. In line 3, $k^l_{n_l}(x,x)$ comes from Equation (5.11). In line 4, the N^l-clustering is performed from the N^l_{MCMC} samples generated in line 3. The N^l cluster centers are the candidate points from which we extract the q^l new points having the maximum adjusted variance $k^l_{n_l,\text{adj}}(x,x)$ (line 5):

$$\begin{aligned} k^l_{n_l,\text{adj}}(x,x) &= \sum_{i=1}^{l} \sigma^2_{\delta^i}(x) \prod_{k=i}^{l-1} \hat{\sigma}^2_{\rho_k} \\ &\times \left(1 + \sum_{j=1}^{n_i} \frac{\left(\varepsilon_{\text{LOO-CV},i}(x^i_j) - \hat{\rho}_{-j,i-1}\varepsilon_{\text{LOO-CV},i-1}(x^i_j)\right)^2}{\sigma^2_{\text{LOO-CV},i}(x^i_j) - \hat{\sigma}^2_{\rho_{i-1},-j}\sigma^2_{\text{LOO-CV},i-1}(x^i_j)}\right) . \end{aligned}$$

In the 'For' loop, the same procedure is repeated for all code levels $m = l-1, \dots, 1$ except that we update the kriging variances $k^m_{n_m}(x,x)$ with the points added in level $m+1, \dots, l$ (since the experimental design sets must be nested). Therefore, in Algorithm 3, $k^t_{n_t + \sum_{i=l+1}^{s} q^i}(x,x)$ corresponds to the kernel distribution of a random process $Z^t_{n_t}(x) \sim [Z_t(x) | \mathbf{Z}^{(t)} = \mathbf{z}^{(t)}]$ conditioned by the observations at points $\left((x^j_{\text{new,i}})_{i=1,\dots,q^s}\right)_{j=l+1,\dots,s}$ when the parameters $(\sigma^2_i)_{i=1,\dots,t}$ and $(\boldsymbol{\theta}_i)_{i=1,\dots,t}$ are considered as known (i.e. this corresponds to a liar method). Furthermore, $k^t_{n_t + \sum_{i=l+1}^{s} q^i,\text{adj}}(x,x)$ corresponds to the predictor variance $k^t_{n_t + \sum_{i=l+1}^{s} q^i}(x,x)$ adjusted with the LOO-CV errors and variances:

$$\begin{aligned} k^t_{n_t + \sum_{i=l+1}^{s} q^i,\text{adj}}(x,x) &= \sum_{i=1}^{t} \sigma^2_{\delta^i + \sum_{i=l+1}^{s} q^i}(x) \prod_{j=i}^{t-1} \hat{\sigma}^2_{\rho_j} \prod_{m=1}^{d} \boldsymbol{\theta}^m_i \\ &\times \left(1 + \sum_{j=1}^{n_i} \frac{\left(\varepsilon_{\text{LOO-CV},i}(x^i_j) - \hat{\rho}_{-j,i-1}\varepsilon_{\text{LOO-CV},i-1}(x^i_j)\right)^2}{\sigma^2_{\text{LOO-CV},i}(x^i_j) - \hat{\sigma}^2_{\rho_{i-1},-j}\sigma^2_{\text{LOO-CV},i-1}(x^i_j)}\right) , \end{aligned} \tag{5.18}$$

where $k^1_{n_1 + \sum_{i=l+1}^{s} q^i}$ and $\sigma^2_{\delta^i + \sum_{i=l+1}^{s} q^i}(x_{\text{new}})$ are deduced from Equation (5.11). We note that for the M-H procedures, we use a Gaussian jumping distribution with a standard deviation such that acceptance rate is around 30%.

Furthermore, let us consider the following quantity

$$\mathrm{IMSE}_{\mathrm{red,q}} = \sum_{i=1}^{t} \sum_{r=1,\dots,q^i} \sigma^2_{\delta^i}(x^i_{\mathrm{new},r}) \prod_{j=i}^{t-1} \hat{\sigma}^2_{\rho_j} \prod_{m=1}^{d} \boldsymbol{\theta}^m_i. \tag{5.19}$$

We consider the allocation $\{q^1, \dots, q^t\}$ which solves the following optimization problem:

$$\{q^1, \dots, q^t\} = \arg \max_{\{q^1,\dots,q^t\}} \mathrm{IMSE}_{\mathrm{red,q}} \text{ such that } \sum_{j=1}^{t} \sum_{i=j}^{t} q^i \mathcal{T}^j = T, \tag{5.20}$$

i.e. we look for the allocation leading the maximal uncertainty reduction. This optimization problem is very complex to solve. Nevertheless, when the number of code levels and the budget T are low (e.g. $s = 2$ in our application) an exhaustive exploration of the allocation $\{q^1, \dots, q^t\}$ can be performed. We are in that case in the presented application . Furthermore, we note that $\mathrm{IMSE}_{\mathrm{red,q}}$ is a proxy on the IMSE reduction when we add $\left((x^m_{\mathrm{new,i}})_{i=1,\dots,q_m}\right)_{m=1,\dots,t}$ at code levels $(y^m(x))_{m=1,\dots,t}$.

In practical application, Algorithm 3 is reiterated until we reach a prescribed precision or the computational time budget is exhausted.

5.3 Applications

We compare in this section the MinIMSE, KleiCrit and AdjMMSE criteria on toy examples and on an application concerning a spherical tank under pressure. We present both the cases of 1 point at-a-time and q points at-a-time sequential kriging. Then, we compare on the tank application, the suggested sequential kriging and co-kriging methods with $s = 2$ levels. The purpose of this section is to emphasize the efficiency of the LOO-CV-based criteria and to highlight that a multi-fidelity analysis can be worthwhile. Finally, for the multi-fidelity sequential co-kriging, we present the allocation of the simulations between the coarse code and the accurate one. We note that for the different examples, we compare the different methods given a prescribed computational time budget.

5.3.1 Comparison between sequential kriging criteria

In this subsection, the 1 point at-a-time sequential kriging criteria (MinIMSE, KleiCrit, AdjMMSE) are compared on three tabulated functions:

- Ackley's function on $[-2, 2]^2$ [Ackley, 1987]:

$$f(x, y) = -20\exp\left(-0.2\sqrt{\frac{x^2 + y^2}{2}}\right) - \exp\left(\frac{\cos(2\pi x) + \cos(2\pi y)}{2}\right) + 20 + \exp(1).$$

- Shubert's function on $[-2, 2]^2$ [Xian, 2001]:

$$f(x, y) = \left(\sum_{k=1}^{5} k\cos\left((k+1)x + k\right)\right)\left(\sum_{k=1}^{5} k\cos\left((k+1)y + k\right)\right).$$

- Michalewicz's function on $[0, \pi]^2$ [Michalewicz, 1992]:

$$f(x, y) = -\sin(x)\left(\sin\left(\frac{x^2}{\pi}\right)\right)^{20} - \sin(y)\left(\sin\left(\frac{y^2}{\pi}\right)\right)^{20}.$$

The comparison is performed on a test set $\mathbf{D}_{\text{test}}$ composed of $n_{\text{test}} = 1000$ points uniformly spread on the input parameter space and from 50 different initial experimental design sets. We compare the different methods with respect to the Normalized RMSE:

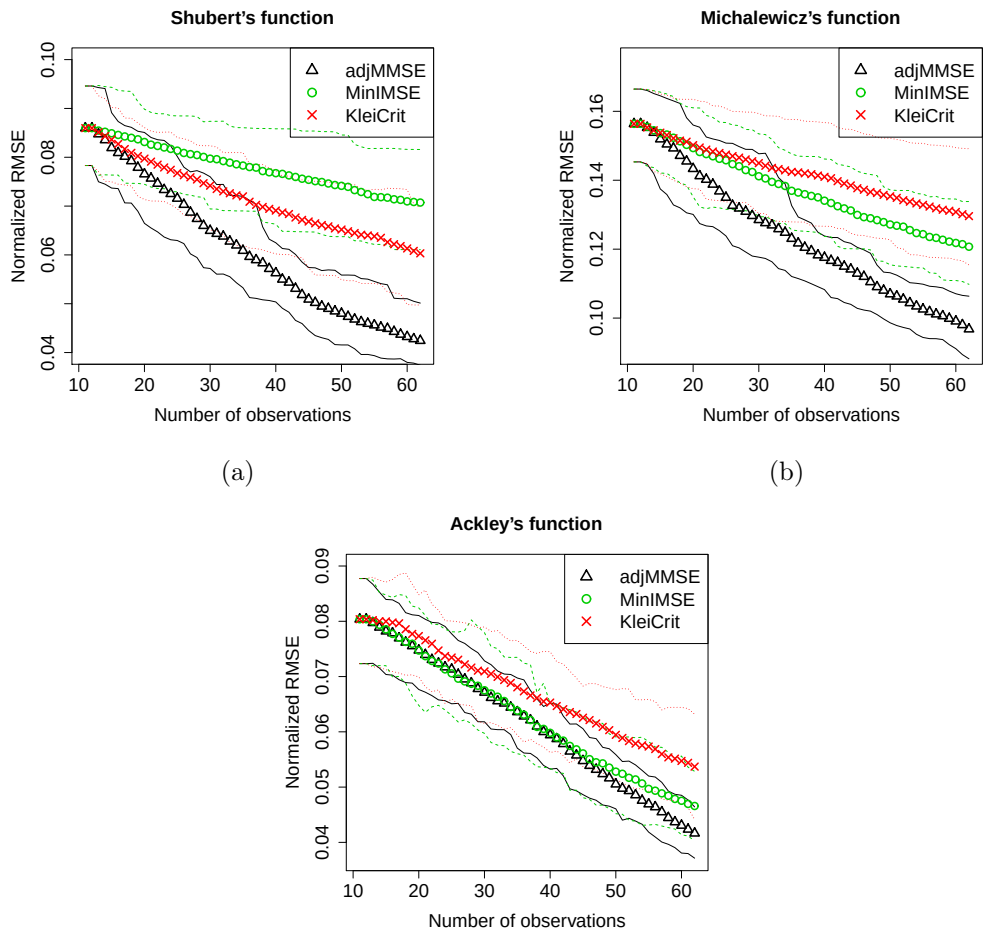
$$\text{Norm RMSE} = \frac{\sqrt{\sum_{i=1}^{n_{\text{test}}} \left(z_{\text{real}}(x^i_{\text{test}}) - z_{\text{pred}}(x)\right)^2 / n_{\text{test}}}}{\max_{x \in \mathbf{D}_{\text{test}}} z_{\text{real}}(x) - \min_{x \in \mathbf{D}_{\text{test}}} z_{\text{real}}(x)}, \tag{5.21}$$

where $z_{\text{real}}(x)$ is the real value of the output and $z_{\text{pred}}(x)$ the predicted one. The 50 initial experimental design sets are LHS designs of 10 points optimized with respect to the S-optimality [Stocki, 2005]. From these designs, 50 sequential krigings are performed and the convergence of the mean and the quantiles of the Normalized RMSE are computed for the three criteria. The mean and confidence intervals of the Normalized RMSE with respect to these 50 initial design sets are presented in Figure 5.5. We use for each kriging a tensorised 5/2-Matérn covariance function and a constant trend. Furthermore, after each added point, the parameters β, σ^2 and $\boldsymbol{\theta}$ (see equations (5.1), (5.2) and (5.3)) of the kriging models are re-estimated with a maximum likelihood method. These estimations are performed thanks to the R library 'DiceKriging' [Roustant et al., 2012].

Figure 5.5 illustrates the efficiency of the criterion AdjMMSE. Indeed, for the Shubert's and Michalewicz's functions, we see that the accuracy of the 1 point at-a-time kriging with this criterion is significantly better than the one of the others criteria (both in terms of mean and quantiles of the Normalized RMSE). In fact, these functions have the particularity to have important variations in some areas of the input parameter space. Thus, the errors are more important in these locations and the suggested criterion focuses the new points on it. Furthermore, the contrast of variations are particularly important for the Shubert's function. For this reason, the IMSE criterion performed very poorly in that case. Indeed, this criterion is efficient for functions with homogeneous variations (i.e. when the predictor MSE well predicts the model errors). In contrast, the Jackknife predictor MSE provided by the criterion KleiCrit manages to catch this heterogeneity and it performs better than the IMSE criterion. Moreover, we see that the performance of the AdjMMSE and IMSE criteria are equivalent for the Ackley's function. We note that the variations of the Ackley's function have the same order of magnitude over the input parameter space.

These examples illustrate the fact that our criterion is more efficient than the other criteria when the functions have important contrast variations and it remains efficient even in the cases where the functions have homogeneous variations (its efficiency is equivalent to the one of the IMSE criterion).

Another point of interest is to compare the gain of CPU time by using the short cuts of Leave-One-Out Cross Validation presented in equations (4.17) and (4.19). For the three academic examples, the CPU time of the sequential design using the criterion AdjMMSE with equations (4.17) and (4.19) is around 14 whereas the one without them is around 19. Therefore, the gain is substancial (it is approximately 25%).



(a) (b)

(c)

Figure 5.5: Comparison between 1 point at-a-time sequential kriging criteria on toy examples. The bold triangles represent the mean of the Normalized RMSE for the AdjMMSE criterion, the bold circles represent it for the MinIMSE criterion and the bold Crosses represent it for the KleiCrit criterion. Furthermore, the solid lines represent the quantiles of probabilities 10% and 90% of the Normalized RMSE, the dotted lines represent them for the MinIMSE criterion and the dotted lines represents them for the KleiCrit criterion. The means and confidence intervals are computed from 50 different sequential design procedures.

5.3.2 Spherical tank under internal pressure example

In this section, we deal with an example about a spherical tank under internal pressure. We are interested in the von Mises stresses on the three points labeled in Figure 5.6. Indeed, we want to prevent from material yielding which occurs when the von Mises stress reaches the critical yield strength.

The system illustrated in Figure 5.6 depends on the following parameters:

- $P\,(MPa) \in [30, 50]$: the value of the internal pressure.

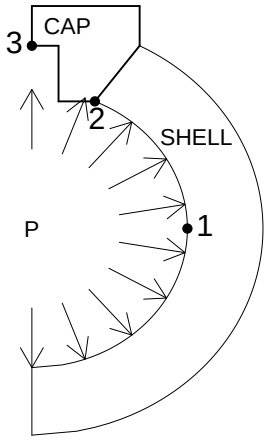


Figure 5.6: Scheme of the spherical tank under pressure.

- $R_{int}\,(mm) \in [1500, 2500]$: the length of the internal radius of the shell.
- $T_{shell}\,(mm) \in [300, 500]$: the thickness of the shell.
- $T_{cap}\,(mm) \in [100, 300]$: the thickness of the cap.
- $E_{shell}\,(GPa) \in [63, 77]$: the Young's modulus of the shell material.
- $E_{cap}\,(GPa) \in [189, 231]$: the Young's modulus of the cap material.
- $\sigma_{y,shell}\,(MPa) \in [200, 300]$: the yield stress of the cap material.
- $\sigma_{y,cap}\,(MPa) \in [400, 800]$: the yield stress of the cap material.

The accurate code output $y^2(x)$ is the value of the von Mises stress provided by an Aster finite elements code (`http://www.code-aster.org`) modeling the system presented in Figure 5.6. We use the notation $x = (P, R_{int}, T_{shell}, T_{cap}, E_{shell}, E_{cap}, \sigma_{y,shell}, \sigma_{y,cap})$. We note that the material properties of the shell correspond to high quality aluminum and the ones of the cap corresponds to steel from classical to high quality. Then, the coarse code output $z_1(x)$ is the value of the von Mises stress given by the 1D simplification of the tank (5.22) (it corresponds to a perfect spherical tank under pressure, i.e. without cap):

$$z_1(x) = \frac{3}{2}\frac{(R_{int} + T_{shell})^3}{(R_{int} + T_{shell})^3 - R_{int}^3}P. \tag{5.22}$$

According to Equation (5.22), the actual input dimension of $z_1(x)$ is three (it depends only on P, R_{int} and T_{shell}) while a sensitivity analysis performed with a Sobol decomposition gives that the accurate code depends essentially on four parameters (P, R_{int}, T_{shell} and T_{cap}). Furthermore, the response is highly stationary. Therefore, only few points are necessary to well predict the output of the code. For these reasons, we can start the sequential strategies from an initial experimental design set with only 10 points.

Thus, for the different comparisons, we use a S-optimal LHS design $\mathbf{D}^2$ of 10 points for the code $z_2(x)$. For the coarse code $z_1(x)$, we start with a design $\mathbf{D}^1$ of 20 points. It is created with the following procedure. First, we create a S-Optimal design $\tilde{\mathbf{D}}^1$ of 20 points.

Second, we remove from $\tilde{\mathbf{D}}^1$ the 10 points that are the closest to those of $\mathbf{D}^2$. Finally, $\mathbf{D}^1$ is the concatenation of $\mathbf{D}^2$ and $\tilde{\mathbf{D}}^1$ (this procedure ensures the nested property $\mathbf{D}^2 \subset \mathbf{D}^1$, see Chapter 4 Section 4.5.3). We note that the CPU time is around 1 minute for the accurate code and 10^{-8} seconds for the coarse code. Nevertheless, to be in a more realistic case, we consider that the CPU time ratio between $z_2(x)$ and $z_1(x)$ equals $B_{2/1} = 10$. Furthermore, each sequential procedure is performed with 40 different initial design sets. Then, the mean and the quantiles of probabilities 90% and 10% of the empirical Normalized MSE are computed from a test set composed of 1000 points uniformly spread on the input parameter space. Finally, for the M-H procedure, we use a Gaussian jumping distribution such that the acceptance rate is around 30% and we set $N_{\text{MCMC}} = 50000$ (we use 5 000 samples for the the burn-in procedure of the M-H method, see [Robert and Casella, 2004]). For the M-H procedure, we use the package R CRAN mcmc. We note that after each added points, the parameters of the kriging or co-kriging models are re-estimated with a maximum likelihood method and that 5/2-Matérn kernels are used for all models.

The remainder of this section is organized as follows. First we compare the MSE of the 1 point at-a-time sequential kriging with the one of the $q = 5$ points at-a-time one. Second, we compare for a given CPU time budget the sequential kriging and cokriging strategies. In the forthcoming developments, the response $i = 1, 2, 3$ refers to the value of the von Mises stress at point i on Figure 5.6.

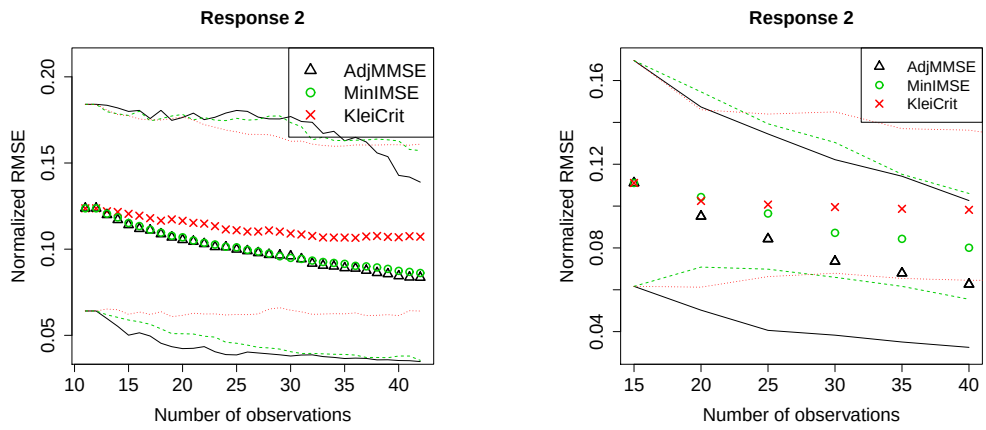
Comparison between sequential kriging criteria

Figure 5.7 compares the different criteria of the 1 point at-a-time and the $q = 5$ points at-a-time sequential kriging. We see that the criteria MinIMSE and AdjMMSE give equivalent values for the MSE for the 1 point at-a-time procedure and they perform better than the KleiCrit criterion. They are equivalent since the output $z_2(x)$ is perfectly stationary. Nevertheless, the criterion AdjMMSE is the most efficient for the $q = 5$ points at-a-time procedure. Indeed, the 5 points provided by a liar method with the MinIMSE criterion are not necessarily those which maximize the reduction of the IMSE. The method suggested in Section 5.2.2 seems to give a better solution.

Comparison between kriging and co-kriging sequential analysis

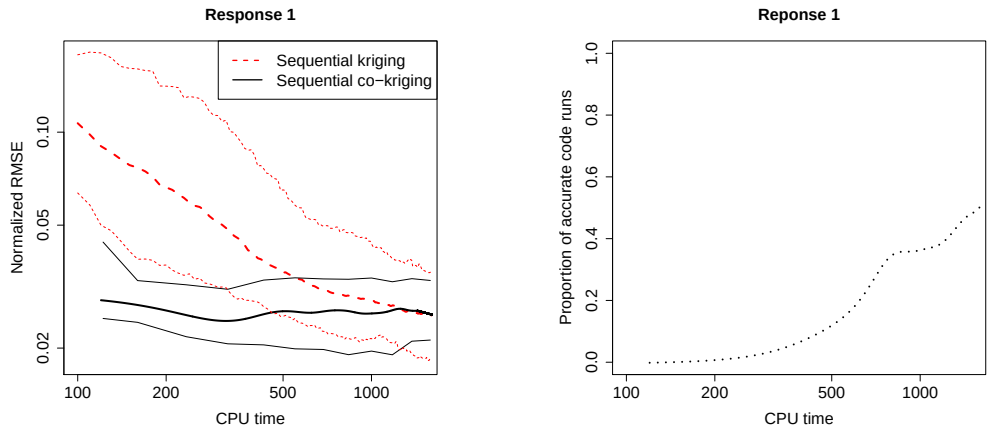
In this section, we compare the sequential kriging strategy with the sequential co-kriging with respect to the AdjMMSE criterion. Figure 5.8 gives the convergence of the empirical normalized MSE for the response 1. We see that the sequential co-kriging performs better than the kriging one. Furthermore, at the beginning of the method, the proportion of runs for the accurate code is very low. Indeed, the coarse code and the accurate code are extremely correlated for this response (around 99%) and thus, during the sequential strategy, the bias between the two codes is well estimated. Then, when the coarse code is well approximated, the sequential strategy starts to run the accurate one (for a CPU time around 500).

Figure 5.9 gives the convergence of the errors for the response 2. For this response, the correlation between the coarse and the accurate code is around 80%. Therefore, the proportion



(a) (b)

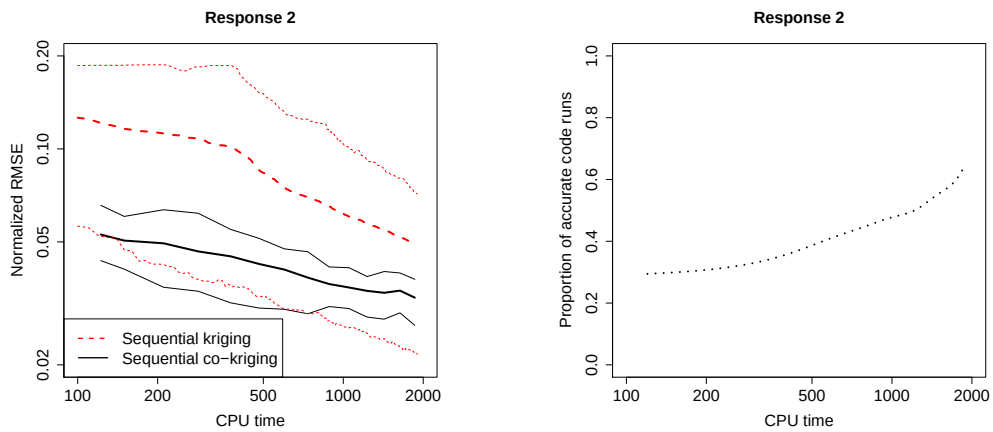
Figure 5.7: Comparison between 1 point at-a-time sequential kriging criteria (a) and batch sequential kriging criteria with $q = 5$ (b) on the spherical tank example. The bold triangles represent the mean of the Normalized RMSE for the AdjMMSE criterion, the bold circles represent it for the MinIMSE criterion and the bold Crosses represent it for the KleiCrit criterion. Furthermore, the solid lines represent the quantiles of probabilities 10% and 90% of the Normalized RMSE, the dashed lines represent them for the MinIMSE criterion and the dotted lines represent them for the KleiCrit criterion.



(a) (b)

Figure 5.8: Comparison between 1 point at-a-time sequential kriging and co-kriging on the response 1 of the spherical tank example with respect to the AdjMMSE criterion (a). The thick dashed line represents the mean of the Normalized RMSE for the sequential kriging and the thick solid line represents it for the sequential co-kriging. The thin lines represent the quantiles of probabilities 10% and 90% of the Normalized RMSE. Figure (b) represents the proportion of runs allocated to the accurate code.

of runs for the accurate code determined by the sequential strategy is more important than in Figure 5.8. Furthermore, we see that this proportion increases with the CPU time. It means that the sequential co-kriging improves the approximation of the coarse code at the beginning of the procedure and then focuses on the accurate code. As a result, we see that the sequential co-kriging strategy is substantially better than the kriging one.



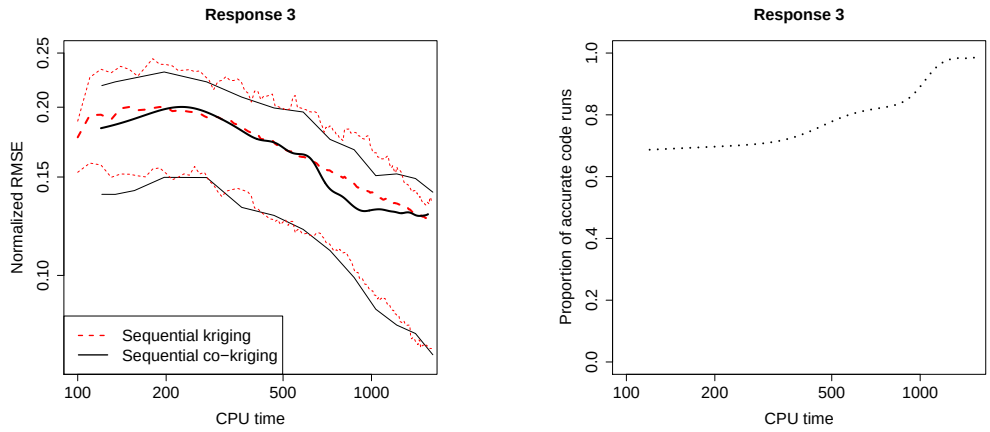
(a) (b)

Figure 5.9: Comparison between 1 point at-a-time sequential kriging and co-kriging on the response 2 of the spherical tank example with respect to the AdjMMSE criterion (a). The thick dashed line represents the mean of the Normalized RMSE for the sequential kriging and the thick solid line represents it for the sequential co-kriging. The thin lines represent the quantiles of probabilities 10% and 90% of the Normalized RMSE. Figure (b) represents the proportion of runs allocated to the accurate code.

Figures 5.8 and 5.9 illustrate the efficiency of the sequential co-kriging when the coarse code bring information on the accurate code. For the response 3, the coarse code is weakly correlated with the accurate code (around 45%). This is due to the fact that the coarse code models the von Mises stress in a perfect spherical tank whereas the response 3 corresponds to the one in the cap. Figure 5.10 shows that in this case, the sequential co-kriging model manages to determine that the coarse code is not worth being simulated. Indeed, the proportion of runs for the accurate code is very high. Furthermore, it shows that the co-kriging sequential design performs as well as the kriging one when the coarse code is non-informative.

Finally, Figure 5.11 shows the efficiency of the (q^1, q^2) at-a-time sequential co-kriging. We set in Algorithm 3 that $T = q^1 + q^2 + 10q^2 = 120$ where the CPU time of the coarse code is 1 and the one of the accurate code is 10. For the the sequential kriging, we use a $q = 10$ at-a-time sequential procedure. Furthermore, Figure 5.11 shows that at the beginning of the procedure, the sequential co-kriging focuses on the approximation of the coarse code whereas at the end it focuses on the accurate code. We note that the allocation of runs for the accurate code in Figure 5.11 agrees with the proportion of runs given in Figure 5.9.

The results of the sequential co-kriging on the different responses show that the criterion suggested in Section 5.2.1 performs very well. Indeed, it is always better than the sequential



(a) (b)

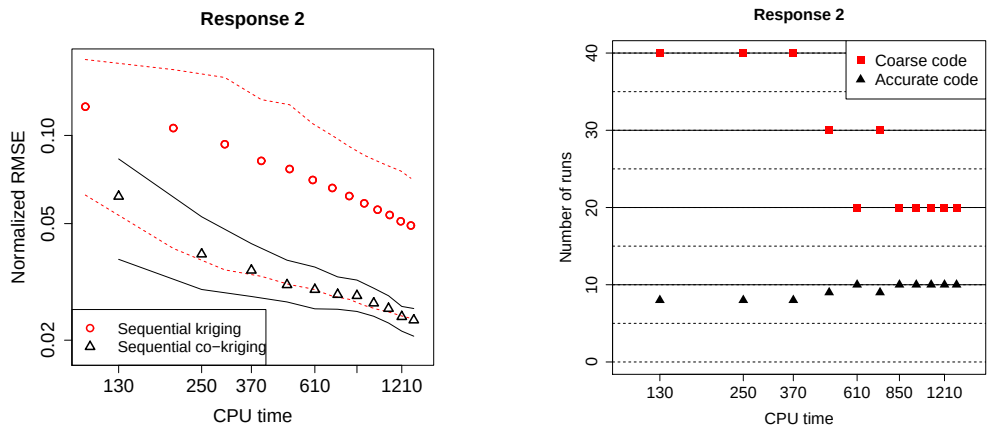
Figure 5.10: Comparison between 1 point at-a-time sequential kriging and co-kriging on the response 3 of the spherical tank example with respect to the AdjMMSE criterion (a). The thick dashed line represents the mean of the Normalized RMSE for the sequential kriging and the thick solid line represents it for the sequential co-kriging. The thin lines represent the quantiles of probabilities 10% and 90% of the Normalized RMSE. Figure (b) represents the proportion of runs allocated to the accurate code.

kriging when the coarse code is informative and its performance is equivalent to it when the coarse code is not useful. Furthermore, the different proportions of runs for the accurate code emphasizes that the criterion accurately determines the contribution of each code to the total model error and the optimal run allocation between the accurate and the coarse codes.

5.4 Conclusion

This chapter deals with sequential strategies for kriging and co-kriging models. First, we have presented classical sequential criteria for the kriging model and we have suggested another criterion based on the Leave-One-Out cross validation errors. This criterion has allowed us to set the new observations at locations where the model error is important. The examples presented in the last section have highlighted the efficiency of the suggested criterion. Indeed, for non-stationary functions, it provides results significantly better than classical criteria and for stationary ones its performance is equivalent to them. We have also emphasized the performance of the suggested criterion on a real application. Furthermore, we show in the application that when the simulations can be performed in parallel, our method has performed better.

Second, we have presented the extension of our criterion to multi-fidelity co-kriging models. We have shown in the application that performing a multi-fidelity sequential co-kriging is worthwhile when the coarse code versions are informative (i.e. highly correlated with the accurate code). Furthermore, a strength of the proposed approach is that it performs as well as a sequential kriging when the coarse code versions are not informative. In fact, the proposed



(a) (b)

Figure 5.11: Comparison between $q = 10$ points at-a-time sequential kriging and (q^1, q^2) points at-a-time sequential co-kriging. On Figure (a) the bold circles represents the mean of the Normalized RMSE for the sequential kriging and the bold triangles represent the one of the sequential co-kriging. Furthermore, the solid lines represent the quantiles of probabilities 10% and 90% of the Normalized RMSE for the sequential co-kriging and the dashed ones represent it for the sequential kriging. On Figure (b) the squares represent the median number of runs for the coarse code during the sequential co-kriging and the triangles represent it for the accurate code.

extension takes into account the contribution of each code to the total predictor mean squared errors and it determines the best run allocation between accurate and coarse code versions given a CPU time budget.

Chapter 6

Multi-fidelity sensitivity analysis

6.1 Introduction

Complex computer codes usually have a large number d of input parameters. The determination of the important input parameters can be carried out by a global sensitivity analysis. We focus on Sobol indices [Sobol, 1993] which are a variance-based importance measure of the model input parameters on the model response. They are based on the Hoeffding-Sobol decomposition suggested by [Hoeffding, 1948] which is valid when the input parameters are independent random variables. We consider the independent case in our framework. For an extension of the Hoeffding-Sobol decomposition in a non-independent case, the reader is referred to [Chastaing et al., 2012]. Furthermore, other strategies for sensitivity analysis with dependent inputs are suggested by [Borgonovo, 2007], [Da Veiga et al., 2009], [Li et al., 2010], [Kucherenko et al., 2012] and [Mara and Tarantola, 2012]. Nevertheless, the estimation of the Sobol indices by sampling methods requires a large number of simulations, that are sometimes too costly and time-consuming. A popular method to overcome this difficulty is to build a mathematical approximation of the code output [Marseguerra et al., 2003] and [Iooss et al., 2006].

We deal in this chapter with the use of kriging and multi-fidelity co-kriging models to estimate Sobol indices. A pioneering article dealing with the kriging approach to perform global sensitivity analysis is the one of [Oakley and O'Hagan, 2004]. They suppose that our prior knowledge about the code can be modeled by a Gaussian process and they estimate the Sobol indices thanks to numerical integrations. The strength of the suggested approach is that it allows for inferring from the surrogate model uncertainty about the Sobol indices. This method is also investigated in [Marrel et al., 2009]. However, the implementation of the method is complex and it is computationally expensive for general covariance kernels. Furthermore, it does not take into account the numerical errors related to the integral evaluations. Another flaw of the method presented in [Oakley and O'Hagan, 2004] and [Marrel et al., 2009] is that it is not able to handle the real Sobol indices but only an approximation of them.

On the other hand, a method giving confidence intervals for the Sobol index estimates and taking into account both the meta-model uncertainty and the numerical errors on the Sobol

index estimations is suggested in [Janon et al., 2011]. They consider a sampling strategy to estimate the Sobol indices instead of numerical integrations and they infer from the sampling errors thanks to a bootstrap procedure. Furthermore, to deal with the meta-model error, they consider an upper bound on it. In the kriging case they use the kriging variance up to a multiplicative constant as upper bound. Nevertheless, this is a rough upper bound which considers the worst error on a test sample. Furthermore, this method does not allow for inferring from the meta-model uncertainty about the Sobol indices.

We propose in this chapter a method combining the approaches of [Oakley and O'Hagan, 2004] and [Janon et al., 2011]. As in [Oakley and O'Hagan, 2004] we consider the code as a realization of a Gaussian process. Nevertheless, we use the estimator suggested in [Janon et al., 2011] to estimate the Sobol indices instead of numerical integrations. As a consequence, we can use the bootstrap method presented in [Archer et al., 1997] to infer from the sampling error on the Sobol indices estimation. Furthermore, contrary to [Oakley and O'Hagan, 2004] and [Marrel et al., 2009] we deal with the real Sobol indices. As a consequence, we introduce non-asymptotics certified Sobol indices estimations, i.e. with confidence intervals which take into account the surrogate model error and the numerical integration error.

Finally, we extend the suggested approach to multi-fidelity co-kriging models. A definition of Sobol indices for multi-fidelity computer codes is presented in [Jacques et al., 2006]. However, their approach is based on tabulated biases between fine and coarse codes and does not allow for inferring from the meta-model uncertainty. The co-kriging model fixes these weaknesses since it allows for considering general forms for the biases and for inferring from the surrogate model error.

This chapter is organized as follows. First we introduce in Section 6.2 the so-called Sobol indices. Then, we present in Section 6.3 the kriging-based sensitivity analysis suggested by [Oakley and O'Hagan, 2004]. Our approach is developed in Section 6.4. In particular, we give an important result allowing for effectively sampling with respect to the kriging predictive distribution in Subsection 6.4.3. Finally, we extend in Section 6.5 the presented approaches to multi-fidelity co-kriging models. We highlight that we present in Subsection 6.5.2 a method to sampling with respect to the multi-fidelity predictive distribution in a Universal co-kriging case. Indeed, as presented in Section 4.3, in this case the predictive distribution is not anymore Gaussian. We propose a method to tackle this issue.

6.2 Global sensitivity analysis: the method of Sobol

We present in this section the method of Sobol for global sensitivity analysis [Sobol, 1993]. It is inspired by the book of [Saltelli et al., 2000] giving an overview of classical sensitivity analysis methods.

6.2.1 Sobol variance-based sensitivity analysis

Let us consider the input parameter space $Q \subseteq \mathbb{R}^d$ such that $(Q, \mathcal{B}(Q))$ is a measurable product space of the form:

$$(Q, \mathcal{B}(Q)) = (Q_1 \times \cdots \times Q_d, \mathcal{B}(Q_1 \times \cdots \times Q_d)),$$

where $\mathcal{B}$ is the Borelian σ-algebra and $Q_i \subset \mathbb{R}$ is a nonempty open set, for $1, \ldots, d$. Furthermore, we consider a probability measure μ on $(Q, \mathcal{B}(Q))$, values in $\mathbb{R}$ and of the form

$$\mu(x) = \mu_1(x^1) \otimes \cdots \otimes \mu_d(x^d).$$

The Hoeffding-Sobol decomposition (see [Hoeffding, 1948]) states that any function $z(x) \in L^2_\mu(Q)$ can be decomposed into summands of increasing dimensionality in such way:

$$z(x) = z_0 + \sum_{i=1} z_i(x^i) + \sum_{1 \le i < j \le k} z_{ij}(x^i, x^j) + \cdots + z_{1,2,\ldots,d}(x^1, \ldots, x^d) = \sum_{u \in \mathcal{P}} z_u(x^u), \quad (6.1)$$

where $\mathcal{P}$ is the collection of all subsets of $\{1, \ldots, d\}$ and x^u is a group of variables such that $x^u = (x^i)_{i \in u}$. Furthermore, the decomposition is unique if we consider the following property for every summand $u = (u_1, \ldots, u_k)_{1 \le k \le d}$, $1 \le u_i \le d$:

$$\int z_u(x^u)\, d\mu_{u_i}(x^{u_i}) = 0, \quad \forall i = 1, \ldots, k. \quad (6.2)$$

A consequence of this property is that all the summands are orthogonal, i.e. for every $z_u(x^u)$ and $z_v(x^v)$ such that $u, v \in \mathcal{P}$ and $u \neq v$, we have:

$$\int z_u(x^u) z_v(x^v)\, d\mu(x) = 0. \quad (6.3)$$

Another consequence is that z_0 represents the mean of $z(x)$ with respect to the measure $\mu(x)$

$$z_0 = \int z(x)\, d\mu(x). \quad (6.4)$$

Sobol [Sobol, 1993] showed that the decomposition (6.1) can be evaluated via multi-dimensional integrals through the following procedure

$$\begin{aligned} z_i(x^i) &= \int z(x)\, d\mu_{-i}(x) - z_0, \\ z_{ij}(x^i, x^j) &= \int z(x)\, d\mu_{-\{i,j\}}(x) - z_i(x^i) - z_j(x^j) - z_0, \\ &\vdots \\ z_u(x^u) &= \int z(x)\, d\mu_{-u}(x) - \sum_{v \subset u} z_v(x^v), \end{aligned}$$

where $\mu_{-u}(x^{-u}) = \bigotimes_{\substack{i=1 \\ i \notin u}}^{d} \mu_i(x^i)$ and $u \in \mathcal{P}$. From this scheme, we can naturally develop the variance-based sensitivity indices of Sobol. First, let us consider the total variance D of $z(x)$:

$$D = \int z^2(x)\, d\mu(x) - z_0^2. \quad (6.5)$$

From the orthogonal property (6.3) and by squaring and integrating the decomposition (6.1), we obtain

$$D = \sum_{i=1}^{d} D_i + \sum_{1 \le i < j \le d} D_{ij} + \cdots + D_{1,2,\ldots,d} = \sum_{u \in \mathcal{P}} D_u, \quad (6.6)$$

with

$$D_u = \int z_u^2(x^u)\, d\mu_{-u}(x). \tag{6.7}$$

Finally, the Sobol sensitivity indices are given by

$$S_u = \frac{D_u}{D}, \tag{6.8}$$

where $u \in \mathcal{P}$. We note that we have the following useful equality which allows for easily interpreting S_u as the part of variance of $z(x)$ due to x^u and not explained by x^v with $v \subset u$.

$$1 = \sum_{i=1}^{d} S_i + \sum_{1 \le i < j \le d} S_{ij} + \cdots + S_{1,2,\ldots,d} = \sum_{u \in \mathcal{P}} S_u. \tag{6.9}$$

In particular, S_i is called the first-order sensitivity index for variable x^i. It measures the main effect of x^i on the output, i.e. the part of variance of $z(x)$ explained by the factor x^i. Furthermore, S_{ij} for $i \neq j$ is the second-order sensitivity index. It measures the part of variance of $z(x)$ due to x^i and x^j and not explained by the individual effects of x^i and x^j.

6.2.2 Monte-Carlo Based estimations of Sobol indices

Now, let us suppose that the inputs are a random vector $X = (X^1, \ldots, X^d)$ defined on the probability space $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$ and with measure μ. Using the previous formalism, the summands of the Hoeffding-Sobol decomposition (6.1) can be interpreted as conditional expectations on the probability space $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$:

$$\begin{aligned}
z_0 &= \mathbb{E}_X\left[z(X)\right], \\
z_i(X^i) &= \mathbb{E}_X\left[z(X)|X^i\right] - z_0, \\
z_{ij}(X^i, X^j) &= \mathbb{E}_X\left[z(X)|X^i, X^j\right] - z_i(X^i) - z_j(X^j) - z_0, \\
&\vdots \\
z_u(X^u) &= \mathbb{E}_X\left[z(X)|X^u\right] - \sum_{v \subset u} z_v(X^v),
\end{aligned}$$

with $u \in \mathcal{P}$. Furthermore, the total variance in (6.5) becomes:

$$D = \mathrm{var}_X\left(z(X)\right) \tag{6.10}$$

and the partial variances presented in (6.7) can be written with the following form

$$D_u = \mathrm{var}_X\left(\mathbb{E}_X\left[z(X)|X^u\right]\right) - \sum_{v \subset u} \mathrm{var}_X\left(\mathbb{E}_X\left[z(X)|X^v\right]\right). \tag{6.11}$$

Now, let us denote by $Q^{d_1} = Q_{i_1} \times \cdots \times Q_{i_{d_1}}$, $d_1 \le d$, $\{i_1, \ldots, i_{d_1}\} \in \mathcal{P}$ and $Q^{d_2} = Q_{j_1} \times \cdots \times Q_{j_{d_2}}$ such that $\{j_1, \ldots, j_{d_2}\} = \{1, \ldots, d\} \setminus \{i_1, \ldots, i_{d_1}\}$. Analogously, we use the notation $X^{d_1} = (X^i)_{i \in \{i_1, \ldots, i_{d_1}\}}$, $X^{d_2} = (X^j)_{j \in \{j_1, \ldots, j_{d_2}\}}$, $\mu^{d_1} = \left(\bigotimes_{i \in \{i_1, \ldots, i_{d_1}\}} \mu_i\right)$ and $\mu^{d_2} = \left(\bigotimes_{j \in \{j_1, \ldots, j_{d_2}\}} \mu_j\right)$ where μ^{d_1} and μ^{d_2} are probability measures on $(Q^{d_1}, \mathcal{B}(Q^{d_1}))$ and

$(Q^{d_2}, \mathcal{B}(Q^{d_2}))$. Consequently, we have the equalities $\mu = \mu^{d_1} \otimes \mu^{d_2}$, $Q = Q^{d_1} \times Q^{d_2}$ and $X = (X^{d_1}, X^{d_2})$ with $d = d_1 + d_2$.

We are interested in evaluating the closed sensitivity index:

$$\mathcal{S}^{X^{d_1}} = \frac{V^{X^{d_1}}}{V} = \frac{\mathrm{var}_X\left(\mathbb{E}_X\left[z(X)|X^{d_1}\right]\right)}{\mathrm{var}_X\left(z(X)\right)}. \tag{6.12}$$

A first method would be to use d-dimensional numerical integrations to approximate the numerator and denominator of (6.12) as presented in [Oakley and O'Hagan, 2004] and [Marrel et al., 2009]. Nonetheless, since d is large in general, this method leads to numerical issues and is computationally expensive. A second approach is to take advantage of the probabilistic interpretation of the Sobol indices and to use a Monte-Carlo procedure to evaluate the different integrals as presented in the forthcoming developments (see [Sobol, 1993]).

Proposition 6.1. *Let us consider the random vectors $(X, \tilde{X})$ with $X = (X^{d_1}, X^{d_2})$ and $\tilde{X} = (X^{d_1}, \tilde{X}^{d_2})$ where X^{d_1} is a random vector on Q^{d_1} with the measure μ^{d_1}, X^{d_2} and $\tilde{X}^{d_2}$ are random vectors on Q^{d_2} with the measure μ^{d_2} and $X^{d_2} \perp \tilde{X}^{d_2}$. We have the following equality:*

$$\mathrm{var}_X\left(\mathbb{E}_X\left[z(X)|X^{d_1}\right]\right) = \mathrm{cov}_X\left(z(X), z(\tilde{X})\right). \tag{6.13}$$

Proof. First, the equality $z(X) \overset{\mathcal{L}}{=} z(\tilde{X})$ implies that

$$\begin{aligned}
\mathrm{cov}_X\left(z(X), z(\tilde{X})\right) &= \mathbb{E}_X\left[z(X)z(\tilde{X})\right] - \mathbb{E}_X\left[z(\tilde{X})\right]\mathbb{E}_X\left[z\left(X\right)\right] \\
&= \mathbb{E}_X\left[z(X)z(\tilde{X})\right] - \mathbb{E}_X\left[z\left(X\right)\right]^2.
\end{aligned}$$

Then, the following equalities hold since $X^{d_2} \perp \tilde{X}^{d_2}$ and $z(X) \overset{\mathcal{L}}{=} z(\tilde{X})$

$$\begin{aligned}
\mathbb{E}_X\left[z(X)z(\tilde{X})\right] &= \mathbb{E}_X\left[\mathbb{E}_X\left[z(X)z(\tilde{X})|X^{d_1}\right]\right] \\
&= \mathbb{E}_X\left[\mathbb{E}_X\left[z(\tilde{X})|X^{d_1}\right]\mathbb{E}_X\left[z(X)|X^{d_1}\right]\right] \\
&= \mathbb{E}_X\left[\mathbb{E}_X\left[z(X)|X^{d_1}\right]^2\right].
\end{aligned}$$

Finally, denoting that $\mathbb{E}_X\left[z\left(X\right)\right] = \mathbb{E}_X\left[\mathbb{E}_X\left[z\left(X\right)|X^{d_1}\right]\right]$ we obtain the equalities

$$\begin{aligned}
\mathrm{cov}_X\left(z(X), z(\tilde{X})\right) &= \mathbb{E}_X\left[\mathbb{E}_X\left[z(X)|X^{d_1}\right]^2\right] - \mathbb{E}_X\left[\mathbb{E}_X\left[z\left(X\right)|X^{d_1}\right]\right]^2 \\
&= \mathrm{var}_X\left(\mathbb{E}_X\left[z(X)|X^{d_1}\right]\right).
\end{aligned}$$

□

$\mathcal{S}^{X^{d_1}}$ in Equation (6.12) can thus be estimated by considering two random vectors $(X_i)_{i=1,\dots,m}$ and $(\tilde{X}_i)_{i=1,\dots,m}$, $m \in \mathbb{N}^*$ lying in $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$ such that $X_i \overset{\mathcal{L}}{=} X$ and $\tilde{X}_i \overset{\mathcal{L}}{=} \tilde{X}$ ($\overset{\mathcal{L}}{=}$ stands for an equality in distribution) and by using an estimator for the covariance $\mathrm{cov}_X\left(z(X), z(\tilde{X})\right)$ and the variance $\mathrm{var}_X\left(z(X)\right)$.

Following this principle, Sobol [Sobol, 1993] suggests the following estimator for the ratio in Equation (6.12):

$$\frac{V_m^{X^{d_1}}}{V_m} = \frac{\frac{1}{m}\sum_{i=1}^m z(X_i)z(\tilde{X}_i) - \frac{1}{m}\sum_{i=1}^m z(X_i)\frac{1}{m}\sum_{i=1}^m z(\tilde{X}_i)}{\frac{1}{m}\sum_{i=1}^m z(X_i)^2 - \left(\frac{1}{m}\sum_{i=1}^m z(X_i)\right)^2}. \tag{6.14}$$

This estimation is improved by [Janon et al., 2012] who propose the following estimator:

$$\frac{V_m^{X^{d_1}}}{V_m} = \frac{\frac{1}{m}\sum_{i=1}^m z(X_i)z(\tilde{X}_i) - \left(\frac{1}{2m}\sum_{i=1}^m z(X_i) + z(\tilde{X}_i)\right)^2}{\frac{1}{m}\sum_{i=1}^m z(X_i)^2 - \left(\frac{1}{2m}\sum_{i=1}^m z(X_i) + z(\tilde{X}_i)\right)^2}. \tag{6.15}$$

In particular they demonstrate that the asymptotic variance in (6.15) is better than the one in (6.14) and they show that the estimator (6.15) is asymptotically efficient for the first order indices. The main weakness of the estimators (6.14) and (6.15) is that they are sometimes not accurate for small values of $V^{X^{d_1}}/V$ in (6.12). To tackle this issue, [Sobol et al., 2007] propose the following estimator

$$\frac{V_m^{X^{d_1}}}{V_m} = \frac{\frac{1}{m}\sum_{i=1}^m z(X_i)z(\tilde{X}_i) - \frac{1}{m}\sum_{i=1}^m z(X_i)z(\tilde{\tilde{X}}_i)}{\frac{1}{m}\sum_{i=1}^m z(X_i)^2 - \left(\frac{1}{m}\sum_{i=1}^m z(X_i)\right)^2}, \tag{6.16}$$

where $\tilde{\tilde{X}} = (\tilde{X}^{d_1}, \tilde{X}^{d_2})$, $\tilde{X}^{d_1} \overset{\mathcal{L}}{=} X^{d_1}$, $\tilde{X}^{d_1} \perp X^{d_1}$ and $(\tilde{\tilde{X}}_i)_{i=1,\dots,m}$ is such that $\tilde{\tilde{X}}_i \overset{\mathcal{L}}{=} \tilde{\tilde{X}}$ for all $i = 1, \dots, m$.

6.3 Kriging-based sensitivity analysis: a first approach

We present in this section the approach suggested by [Oakley and O'Hagan, 2004] and [Marrel et al., 2009] to perform global sensitivity analysis using kriging surrogate models. Then, we present an alternative method that allows us to avoid complex numerical integrations. Nevertheless, we will see that the two proposed approaches do not provide a correct representation of the Sobol indices. We handle this problem in the next section.

6.3.1 Kriging-based sensitivity indices

Let us introduce the kriging-based global sensitivity analysis presented in [Oakley and O'Hagan, 2004] and [Marrel et al., 2009]. The idea is to consider that our prior knowledge about the code $z(x)$ can be modeled by a Gaussian process $Z(x)$ with mean $\mathbf{f}'(x)\boldsymbol{\beta}$ and covariance kernel $\sigma^2 r(x, \tilde{x})$. Then, we surrogate the code $z(x)$ by a Gaussian process $Z_n(x)$ having the predictive distribution of $Z(x)$ conditioning by the known value $\mathbf{z}^n$ of $z(x)$ at points in the experimental design set $\mathbf{D} = \{x^1, \dots, x^n\}$, $x^i \in Q$:

$$Z_n(x) \sim \mathrm{GP}\left(m_n(x), s_n^2(x, \tilde{x})\right), \tag{6.17}$$

where the mean $m_n(x)$ and the variance $s_n^2(x, \tilde{x})$ corresponds to the kriging equations presented in Subsection 1.2.1:

$$m_n(x) = \mathbf{f}'(x)\hat{\boldsymbol{\beta}} + \mathbf{r}'(x)\mathbf{R}^{-1}\left(\mathbf{z}^n - \mathbf{F}\hat{\boldsymbol{\beta}}\right),$$

$$s_n^2(x,\tilde{x}) = \sigma^2 \left(1 - \begin{pmatrix} \mathbf{f}'(x) & \mathbf{r}'(x) \end{pmatrix} \begin{pmatrix} 0 & \mathbf{F}' \\ \mathbf{F} & \mathbf{R} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{f}(\tilde{x}) \\ \mathbf{r}(\tilde{x}) \end{pmatrix} \right),$$

where $\hat{\boldsymbol{\beta}} = \left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}'\mathbf{R}^{-1}\mathbf{z}^n$ and σ^2 is estimated with a restricted maximum likelihood method, i.e. $\hat{\sigma}^2 = (\mathbf{z}^n - \hat{\boldsymbol{\beta}}\mathbf{F})'\mathbf{R}^{-1}(\mathbf{z}^n - \hat{\boldsymbol{\beta}}\mathbf{F})/(n-p)$ where p is the size of $\boldsymbol{\beta}$.

The idea suggested in [Oakley and O'Hagan, 2004] and [Marrel et al., 2009] is to substitute $z(x)$ with $Z_n(x)$ in Equation (6.12):

$$\mathcal{S}_n^{X^{d_1}} = \frac{V_n^{X^{d_1}}}{V_n} = \frac{\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)}{\mathrm{var}_X\left(Z_n(X)\right)}. \tag{6.18}$$

Therefore, if we denote by $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$ the probability space where the Gaussian process $Z(x)$ lies, then the estimator $\mathcal{S}_n^{X^{d_1}}$ lies in $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$ (it is hence random). We note that $Z_n(X)$ is defined on the product probability space $(\Omega_X \times \Omega_Z, \sigma(\mathcal{F}_X \times \mathcal{F}_Z), \mathbb{P}_X \otimes \mathbb{P}_Z)$.

Nevertheless, the distribution of $\mathcal{S}_n^{X^{d_1}}$ is intractable and [Oakley and O'Hagan, 2004] and [Marrel et al., 2009] focus on its mean and variance. More precisely, in order to derive analytically the Sobol index estimates they consider the following quantity:

$$\hat{\mathcal{S}}_n^{X^{d_1}} = \frac{\mathbb{E}_Z\left[\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)\right]}{\mathbb{E}_Z\left[\mathrm{var}_X\left(Z_n(X)\right)\right]}, \tag{6.19}$$

where $\mathbb{E}_Z[.]$ stands for the expectation in the probability space $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$. Furthermore, the uncertainty on $\hat{\mathcal{S}}_n^{X^{d_1}}$ is evaluated with the following quantity:

$$\sigma^2(\tilde{\mathcal{S}}_n^{X^{d_1}}) = \frac{\mathrm{var}_Z\left(\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)\right)}{\mathbb{E}_Z\left[\mathrm{var}_X\left(Z_n(X)\right)\right]^2}. \tag{6.20}$$

As shown in [Oakley and O'Hagan, 2004] and [Marrel et al., 2009], the equations (6.19) and (6.20) can be derived analytically through multi-dimensional integrals for the cases $d_1 = i$, $i = 1, \ldots, d$, i.e. for the first-order indices. Furthermore, with some particular formulations of $\mathbf{f}(x)$, $\mu(x)$ and $r(x,\tilde{x})$, these multi-dimensional integrals can be written as product of one-dimensional ones.

Discussions: The method suggested in [Oakley and O'Hagan, 2004] and [Marrel et al., 2009] provides an interesting tool to perform sensitivity analysis of complex models. Nevertheless, in our opinion it suffers from the following flaws:

1. For general choice of $\mathbf{f}(x)$, $\mu(x)$ and $r(x,\tilde{x})$, the numerical evaluations of (6.19) and (6.20) can be very complex since it requires multi-dimensional integrals.
2. The method is derived for first-order sensitivity indices and cannot easily be extended to higher order indices.
3. The method allows for inferring from the surrogate model uncertainty about the sensitivity indices but does not allow for taking into account the numerical errors related to the multi-dimensional integral estimations.

4. The considered index expectation and deviation do not correspond to the real Sobol index ones since we obviously have

$$\frac{\mathbb{E}_Z\left[\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)\right]}{\mathbb{E}_Z\left[\mathrm{var}_X\left(Z_n(X)\right)\right]} \neq \mathbb{E}_Z\left[\frac{\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)}{\mathrm{var}_X\left(Z_n(X)\right)}\right]$$

and

$$\frac{\mathrm{var}_Z\left(\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)\right)}{\mathbb{E}_Z\left[\mathrm{var}_X\left(Z_n(X)\right)\right]^2} \neq \mathrm{var}_Z\left(\frac{\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)}{\mathrm{var}_X\left(Z_n(X)\right)}\right).$$

In the next subsection, we deal with the points 1, 2 and 3 by suggesting a Monte-Carlo sampling method to evaluate (6.19) and (6.20) instead of quadrature integrations. Nonetheless, we do not tackle the issue of point 4. To handle it, we suggest another method in Section 6.4.

6.3.2 Monte-Carlo estimations for the first approach

We present in this subsection, another approach to deal with the evaluation of $\tilde{\mathcal{S}}_n^{X^{d_1}}$ in (6.19). Its principle simply consists in using the estimation methods suggested in Subsection 6.2.2 instead of quadrature integrations to compute $\mathbb{E}_Z\left[\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)\right]$ and $\mathbb{E}_Z\left[\mathrm{var}_X\left(Z_n(X)\right)\right]$. We present the method with the estimator presented in [Sobol, 1993]. The extension to those presented in [Janon et al., 2011] and [Sobol et al., 2007] is straightforward. Let us substitute in the estimator presented in Equation (6.14) the code $z(x)$ by the Gaussian process $Z_n(x)$:

$$\frac{V_{m,n}^{X^{d_1}}}{V_{m,n}} = \frac{\frac{1}{m}\sum_{i=1}^m Z_n(X_i)Z_n(\tilde{X}_i) - \frac{1}{m}\sum_{i=1}^m Z_n(X_i)\frac{1}{m}\sum_{i=1}^m Z_n(\tilde{X}_i)}{\frac{1}{m}\sum_{i=1}^m Z_n(X_i)^2 - \left(\frac{1}{m}\sum_{i=1}^m Z_n(X_i)\right)^2}, \tag{6.21}$$

where the samples $(X_i)_{i=1,\dots,m}$ and $(\tilde{X}_i)_{i=1,\dots,m}$ are those introduced in Subsection 6.2.2. Therefore, $V_{m,n}^{X^{d_1}}/V_{m,n}$ is an estimator of $V^{X^{d_1}}/V$ (6.12) when we replace the true function $z(x)$ by its approximation $Z_n(x)$ built from n observations $\mathbf{z}^n$ of $z(x)$ and when we estimate the variances and the expectation involved in (6.12) by a Monte-Carlo method with m particles. To be clear in the remainder of this chapter, we name as Monte-Carlo error the one related to the Monte-Carlo estimation and we name as meta-model error the one related to the substitution of $z(x)$ by a surrogate model. Furthermore, m will always denote the number of Monte-Carlo particles and n the number of observations used to build the surrogate model.

The strength of this formulation is that it gives closed form formulas for the evaluation of (6.19) for any choice of $\mathbf{f}(x)$, $\mu(x)$ and $r(x,\tilde{x})$ contrary to [Oakley and O'Hagan, 2004] and [Marrel et al., 2009]. Furthermore, this method can directly be used for any order of Sobol indices which contrasts with the one presented in Subsection (6.3.1). Finally, unlike quadrature integrations, Monte-Carlo integrations allow for taking into account the numerical errors related to the integral evaluations. In particular, as presented in [Archer et al., 1997], the bootstrap method can be directly used to obtain confidence intervals on the Sobol indices.

We give in the following equation the Monte-Carlo estimation of $\tilde{\mathcal{S}}_n^{X^{d_1}}$ (6.19) corresponding to the kriging-based sensitivity indices presented in [Oakley and O'Hagan, 2004] and [Marrel et al., 2009].

$$\begin{aligned}\tilde{\mathcal{S}}_{m,n}^{X^{d_1}} &= \frac{\mathbb{E}_Z\left[V_{m,n}^{X^{d_1}}\right]}{\mathbb{E}_Z[V_{m,n}]} \\ &= \frac{\frac{1}{m}\sum_{i=1}^m s_n^2(X_i,\tilde{X}_i)+m_n(X_i)m_n(\tilde{X}_i)-\frac{1}{m^2}\sum_{i,j=1}^m s_n^2(X_i,\tilde{X}_j)+m_n(X_i)m_n(\tilde{X}_j)}{\frac{1}{m}\sum_{i=1}^m s_n^2(X_i,X_i)+m_n(X_i)m_n(X_i)-\frac{1}{m^2}\sum_{i,j=1}^m s_n^2(X_i,X_j)+m_n(X_i)m_n(X_j)}.\end{aligned} \tag{6.22}$$

We note that the expression of $\tilde{\mathcal{S}}_{m,n}^{X^{d_1}}$ is different from the one obtained by estimating $V_m^{X^{d_1}}/V_m$ in (6.14) by replacing $z(x)$ by the predictive mean $m_n(x)$. In $\tilde{\mathcal{S}}_{m,n}^{X^{d_1}}$ we take into account the kriging predictive covariance through the terms $s_n^2(X_i,\tilde{X}_j)$ and $s_n^2(X_i,X_j)$.

6.4 Kriging-based sensitivity analysis: a second approach

We have highlighted at the end of Subsection 6.3.1 that one of the main flaws of the method presented by [Oakley and O'Hagan, 2004] is that it does not care about the real Sobol indices. We present in Subsection 6.4.1 another approach which deals with this issue. Then, in Subsection 6.4.3 we present an efficient method to compute it.

6.4.1 Kriging-based Sobol index estimation

First of all, in the previous section we have considered the variance of the main effects $V^{X^{d_1}}$ and the total variance V separately in Equation (6.12). That is why the ratio of the expectations is considered as a sensitivity index in Equation (6.19). In fact, in a Sobol index framework, we are interested in the ratio between $V^{X^{d_1}}$ and V. Therefore, we suggest to deal directly with the following estimator (see Equation (6.21)):

$$\mathcal{S}_{m,n}^{X^{d_1}} = \frac{V_{m,n}^{X^{d_1}}}{V_{m,n}}, \tag{6.23}$$

which corresponds to the ratio $V^{X^{d_1}}/V$ after substituting the code $z(x)$ by the Gaussian process $Z_n(x)$ and estimating the terms $\mathrm{var}_X\left(\mathbb{E}_X\left[Z_n(X)|X^{d_1}\right]\right)$ and $\mathrm{var}_X\left(Z_n(X)\right)$ with a Monte-Carlo procedure as presented in [Sobol, 1993]. We note that we can naturally adapt the presented estimator with the ones suggested by [Sobol et al., 2007] and [Janon et al., 2012]. Nevertheless, we cannot obtain closed form expressions for the mean or the variance of this estimator. We thus have to numerically estimate them. We present in Algorithm 4 the suggested method to compute the distribution of $\mathcal{S}_{m,n}^{X^{d_1}}$.

The output $\left(\hat{\mathcal{S}}_{m,n,k,l}^{X^{d_1}}\right)_{\substack{k=1,\dots,N_Z\\ l=1,\dots,B}}$ of Algorithm 4 is a sample of size $N_Z \times B$ of $\mathcal{S}_{m,n}^{X^{d_1}}$ defined on $(\Omega_X \times \Omega_Z, \sigma(\mathcal{F}_X \times \mathcal{F}_Z), \mathbb{P}_X \times \mathbb{P}_Z)$ (i.e. $\mathcal{S}_{m,n}^{X^{d_1}}$ takes both into account the uncertainty of the meta-model and the one of the Monte-Carlo integrations). Then, we can deduce the following estimate $\bar{\mathcal{S}}_{m,n}^{X^{d_1}}$ for $\mathcal{S}^{X^{d_1}}$:

$$\bar{\mathcal{S}}_{m,n}^{X^{d_1}} = \frac{1}{N_Z B}\sum_{\substack{k=1,\dots,N_Z\\ l=1,\dots,B}} \hat{\mathcal{S}}_{m,n,k,l}^{X^{d_1}}. \tag{6.24}$$

Algorithm 4 Evaluation of the distribution of $\mathcal{S}_{m,n}^{X^{d_1}}$.

1: Build $Z_n(x)$ from the n observations $\mathbf{z}^n$ of $z(x)$ at points in $\mathbf{D}$ (see Equation (6.17)).
2: Generate two samples $(x_i)_{i=1,\dots,m}$ and $(\tilde{x}_i)_{i=1,\dots,m}$ of the random vectors $(X_i)_{i=1,\dots,m}$ and $(\tilde{X}_i)_{i=1,\dots,m}$ with respect to the probability measure μ (see Proposition 6.1).
3: Set N_Z the number of samples for $Z_n(x)$ and B the number of bootstrap samples for evaluating the uncertainty related to Monte-Carlo integrations.
4: **for** $k = 1, \dots, N_Z$ **do**
5: Sample a realization $z_n(\mathbf{x})$ of $Z_n(\mathbf{x})$ with $\mathbf{x} = \{(x_i)_{i=1,\dots,m}, (\tilde{x}_i)_{i=1,\dots,m}\}$
6: Compute $\hat{\mathcal{S}}_{m,n,k,1}^{X^{d_1}}$ thanks to Equation (6.21) from $z_n(\mathbf{x})$.
7: **for** l=2,...,B **do**
8: Sample with replacements two samples $\mathbf{u}$ and $\tilde{\mathbf{u}}$ from $\{(x_i)_{i=1,\dots,m}\}$ and $\{(\tilde{x}_i)_{i=1,\dots,m}\}$.
9: Compute $\hat{\mathcal{S}}_{m,n,k,l}^{X^{d_1}}$ from $z_n(\mathbf{x}^B)$ with $\mathbf{x}^B = \{\mathbf{u}, \tilde{\mathbf{u}}\}$.
10: **end for**
11: **end for**
return $\left(\hat{\mathcal{S}}_{m,n,k,l}^{X^{d_1}}\right)_{\substack{k=1,\dots,N_Z \\ l=1,\dots,B}}$

Furthermore, we can estimate the variance of $\mathcal{S}_{m,n}^{X^{d_1}}$ with:

$$\hat{\sigma}^2(\mathcal{S}_{m,n}^{X^{d_1}}) = \frac{1}{N_Z B - 1} \sum_{\substack{k=1,\dots,N_Z \\ l=1,\dots,B}} \left(\hat{\mathcal{S}}_{m,n,k,l}^{X^{d_1}} - \bar{\mathcal{S}}_{m,n}^{X^{d_1}}\right)^2. \tag{6.25}$$

We note that the computational limitation of the algorithm is the sampling of the Gaussian process $Z_n(x)$ on $\mathbf{x} = \{(x_i)_{i=1,\dots,m}, (\tilde{x}_i)_{i=1,\dots,m}\}$. For that reason, we use a bootstrap procedure to evaluate the uncertainty of the Monte-Carlo integrations instead of sampling different realizations of the random vectors $(X_i)_{i=1,\dots,m}$ and $(\tilde{X}_i)_{i=1,\dots,m}$. Furthermore, the same bootstrap samples are used for the N_Z realizations of $Z_n(x)$.

Nevertheless, the number of Monte-Carlo particles m is very large in general - it is often around $m = 5000d$ - and it thus can be an issue to compute realizations of $Z_n(x)$ on $\mathbf{x}$. We present in the Subsection 6.4.3 an efficient method to deal with this point for any choice of $\mu(x)$, $\mathbf{f}(x)$ and $r(x, \tilde{x})$ and any index order.

6.4.2 Determining the minimal number of Monte-Carlo particles m

We are interested here in quantifying the uncertainty of the considered estimator $\mathcal{S}_{m,n}^{X^{d_1}}$ (6.23). This estimator integrates two sources of uncertainty, the first one is related to the meta-model approximation and the second one is related to the Monte-Carlo integration. Therefore, we can decompose the variance of $\mathcal{S}_{m,n}^{X^{d_1}}$ as follows:

$$\mathrm{var}\left(\mathcal{S}_{m,n}^{X^{d_1}}\right) = \mathrm{var}_Z\left(\mathbb{E}_X\left[\mathcal{S}_{m,n}^{X^{d_1}} \middle| Z_n(x)\right]\right) + \mathrm{var}_X\left(\mathbb{E}_Z\left[\mathcal{S}_{m,n}^{X^{d_1}} \middle| (X_i, \tilde{X}_i)_{i=1,\dots,m}\right]\right)$$

where $\mathrm{var}_Z\left(\mathbb{E}_X\left[\mathcal{S}_{m,n}^{X^{d_1}} \middle| Z_n(x)\right]\right)$ is the contribution of the meta-model on the variability of $\mathcal{S}_{m,n}^{X^{d_1}}$ and $\mathrm{var}_X\left(\mathbb{E}_Z\left[\mathcal{S}_{m,n}^{X^{d_1}} \middle| (X_i, \tilde{X}_i)_{i=1,\dots,m}\right]\right)$ is the one of the Monte-Carlo integration. Fur-

thermore, we have the following equalities:

$$\begin{cases} \mathrm{var}_Z\left(\mathbb{E}_X\left[\mathcal{S}_{m,n}^{X^{d_1}}\middle|Z_n(x)\right]\right) & = & \mathbb{E}_X\left[\mathrm{var}_Z\left(\mathcal{S}_{m,n}^{X^{d_1}}\middle|(X_i,\tilde{X}_i)_{i=1,\dots,m}\right)\right] \\ \mathrm{var}_X\left(\mathbb{E}_Z\left[\mathcal{S}_{m,n}^{X^{d_1}}\middle|(X_i,\tilde{X}_i)_{i=1,\dots,m}\right]\right) & = & \mathbb{E}_Z\left[\mathrm{var}_X\left(\mathcal{S}_{m,n}^{X^{d_1}}\middle|Z_n(x)\right)\right] \end{cases}$$

Therefore, from the sample $\left(\hat{\mathcal{S}}_{m,n,k,l}^{X^{d_1}}\right)_{\substack{k=1,\dots,N_Z\\ l=1,\dots,B}}$ we can estimate the part of variance of the estimator $\mathcal{S}_{m,n}^{X^{d_1}}$ related to the meta-modelling as follows:

$$\hat{\sigma}_{Z_n}^2(\mathcal{S}_{m,n}^{X^{d_1}}) = \frac{1}{B}\sum_{l=1}^{B}\frac{1}{N_Z-1}\sum_{k=1}^{N_Z}\left(\hat{\mathcal{S}}_{m,n,k,l}^{X^{d_1}} - \bar{\hat{\mathcal{S}}}_{m,n,l}^{X^{d_1}}\right)^2 \tag{6.26}$$

where $\bar{\hat{\mathcal{S}}}_{m,n,l}^{X^{d_1}} = \left(\sum_{i=1}^{N_Z}\mathcal{S}_{m,n,i,l}^{X^{d_1}}\right)/N_Z$. Furthermore, we can evaluate the part of variance of $\mathcal{S}_{m,n}^{X^{d_1}}$ related to the Monte-Carlo integrations as follows:

$$\hat{\sigma}_{MC}^2(\mathcal{S}_{m,n}^{X^{d_1}}) = \frac{1}{N_Z}\sum_{i=1}^{N_Z}\frac{1}{B-1}\sum_{i=1}^{B}\left(\hat{\mathcal{S}}_{m,n,k,i}^{X^{d_1}} - \bar{\bar{\hat{\mathcal{S}}}}_{m,n,k}^{X^{d_1}}\right)^2 \tag{6.27}$$

where $\bar{\bar{\hat{\mathcal{S}}}}_{m,n,k}^{X^{d_1}} = \left(\sum_{i=1}^{B}\mathcal{S}_{m,n,k,i}^{X^{d_1}}\right)/B$.

Therefore, we have three different cases:

1. $\hat{\sigma}_{Z_n}^2(\mathcal{S}_{m,n}^{X^{d_1}}) \gg \hat{\sigma}_{MC}^2(\mathcal{S}_{m,n}^{X^{d_1}})$: the estimation error of $\mathcal{S}_{m,n}^{X^{d_1}}$ is essentially due to the meta-model error.
2. $\hat{\sigma}_{Z_n}^2(\mathcal{S}_{m,n}^{X^{d_1}}) \ll \hat{\sigma}_{MC}^2(\mathcal{S}_{m,n}^{X^{d_1}})$: the estimation error of $\mathcal{S}_{m,n}^{X^{d_1}}$ is essentially due to the Monte-Carlo error.
3. $\hat{\sigma}_{Z_n}^2(\mathcal{S}_{m,n}^{X^{d_1}}) \approx \hat{\sigma}_{MC}^2(\mathcal{S}_{m,n}^{X^{d_1}})$: the metamodel and the Monte-Carlo errors have the same contribution on the estimation error of $\mathcal{S}_{m,n}^{X^{d_1}}$.

Considering that the number of observations n is fixed, the minimal number of Monte-Carlo particles m is the one such that $\hat{\sigma}_{Z_n}^2(\mathcal{S}_{m,n}^{X^{d_1}}) \approx \hat{\sigma}_{MC}^2(\mathcal{S}_{m,n}^{X^{d_1}})$. We call it "minimal" since it is the one from which the Monte-Carlo error no longer dominates. Therefore, it should be the minimum number of required particles in practical applications. In practice, to determine it, we start with a small value of m and we increase it while the inequality $\hat{\sigma}_{Z_n}^2(\mathcal{S}_{m,n}^{X^{d_1}}) > \hat{\sigma}_{MC}^2(\mathcal{S}_{m,n}^{X^{d_1}})$ is true.

6.4.3 Sampling with respect to the kriging predictive distribution on large data sets

We saw in the previous subsection in Algorithm 4 that in a kriging framework, we can assess the distribution of the Sobol index estimators from realizations of the conditional Gaussian process $Z_n(x)$ at points in $\mathbf{x}$. Nevertheless, the size of the corresponding random vector could be important since it equals twice the number of Monte-Carlo particles m. Therefore, computing such realizations could lead numerical issues such as ill-conditioned matrix or huge

computational cost. Especially if we use a Cholesky decomposition since its complexity is $\mathcal{O}((2m)^3)$ and it often leads ill-conditioned matrix since the predictive variance of $Z_n(x)$ is close to zero around the experimental design points.

Let us introduce the following unconditioned Gaussian process:

$$\tilde{Z}(x) \sim \mathrm{GP}(0, \sigma^2 r(x, \tilde{x})). \tag{6.28}$$

We have the following proposition [Chilès and Delfiner, 1999]:

Proposition 6.2 (Sampling $Z_n(x)$ by kriging conditioning)**.** *Let us consider the following Gaussian process:*

$$\tilde{Z}_n(x) = m_n(x) - \tilde{m}_n(x) + \tilde{Z}(x), \tag{6.29}$$

where $m_n(x)$ is the predictive mean of $Z_n(x)$ (6.17),

$$\tilde{m}_n(x) = \mathbf{f}'(x)\tilde{\boldsymbol{\beta}} + \mathbf{r}'(x)\mathbf{R}^{-1}\left(\tilde{Z}(\mathbf{D}) - \mathbf{F}\tilde{\boldsymbol{\beta}}\right) \tag{6.30}$$

and $\tilde{\boldsymbol{\beta}} = \left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}'\mathbf{R}^{-1}\tilde{Z}(\mathbf{D})$. Then, we have

$$\tilde{Z}_n(x) \overset{\mathcal{L}}{=} Z_n(x),$$

where $Z_n(x)$ has the distribution of the Gaussian process $Z(x)$ of mean $\mathbf{f}'(x)\boldsymbol{\beta}$ and covariance kernel $\sigma^2 r(x, \tilde{x})$ conditioned by $\mathbf{z}^n$ at points in $\mathbf{D}$ (6.17). We note that we are in a Universal kriging case, i.e. we infer from the parameter $\boldsymbol{\beta}$. In a simple kriging case, the proposition remains true by setting $\tilde{\boldsymbol{\beta}} = 0$.

Proof. Let us introduce the following random process:

$$\tilde{Z}_n(x) = m_n(x) - \tilde{m}_n(x) + \tilde{Z}(x), \tag{6.31}$$

where:

$$\tilde{Z}(x) \sim \mathrm{GP}(0, \sigma^2 r(x, \tilde{x})).$$

The random process $\tilde{Z}_n(x)$ is Gaussian since it is a linear transformation of the Gaussian process $\tilde{Z}(x)$. As a Gaussian process is entirely determined by its mean and covariance kernel, we just have to prove the following equalities:

$$\mathbb{E}[\tilde{Z}_n(x)] = m_n(x) \tag{6.32}$$

and:

$$\mathrm{cov}(\tilde{Z}_n(x), \tilde{Z}_n(\tilde{x})) = s_n^2(x, \tilde{x}). \tag{6.33}$$

First, from the equalities $\mathbb{E}[\tilde{Z}(x)] = 0$ and:

$$\mathbb{E}[\tilde{m}_n(x)] \quad = \quad \mathbf{f}'(x)\mathbb{E}[\tilde{\boldsymbol{\beta}}] + \mathbf{r}'(x)\mathbf{R}^{-1}\left(\mathbb{E}[\tilde{Z}(\mathbf{D})] - \mathbf{F}\mathbb{E}[\tilde{\boldsymbol{\beta}}]\right) = 0,$$

the equality (6.32) holds. Then, we have to verify the equality (6.33).

$$\mathrm{cov}(\tilde{Z}_n(x), \tilde{Z}_n(\tilde{x})) \quad = \quad \mathrm{cov}(\tilde{m}_n(x), \tilde{m}_n(\tilde{x})) - 2\mathrm{cov}(\tilde{Z}(x), \tilde{m}_n(\tilde{x})) + \mathrm{cov}(\tilde{Z}(x), \tilde{Z}(\tilde{x})).$$

First, we have:

$$\mathrm{cov}(\tilde{Z}(x), \tilde{Z}(\tilde{x})) = \sigma^2 r(x - \tilde{x}). \tag{6.34}$$

Second, we have the following equality:

$$\begin{aligned}
\mathrm{cov}(\tilde{m}_n(x), \tilde{m}_n(\tilde{x})) &= (\mathbf{f}'(x) - \mathbf{r}'(x)\mathbf{R}^{-1}\mathbf{F})\mathrm{cov}(\tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}})(\mathbf{f}(\tilde{x}) - \mathbf{F}'\mathbf{R}^{-1}\mathbf{r}(\tilde{x})) \\
&+ \mathbf{r}'(x)\mathbf{R}^{-1}\mathrm{cov}(\tilde{Z}(\mathbf{D}), \tilde{Z}(\mathbf{D}))\mathbf{R}^{-1}\mathbf{r}(\tilde{x}) + 2\mathbf{f}'(x)\mathrm{cov}(\tilde{\boldsymbol{\beta}}, \tilde{Z}(\mathbf{D}))\mathbf{R}^{-1}\mathbf{r}(\tilde{x}) \\
&- 2\mathbf{r}'(x)\mathbf{R}^{-1}\mathrm{cov}(\tilde{Z}(\mathbf{D}), \tilde{\boldsymbol{\beta}})\mathbf{F}'\mathbf{R}^{-1}\mathbf{r}(\tilde{x}),
\end{aligned}$$

where:

$$\begin{aligned}
\mathrm{cov}(\tilde{\boldsymbol{\beta}}, \tilde{Z}(\mathbf{D})) &= \left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}'\mathbf{R}^{-1}\mathrm{cov}(\tilde{Z}(\mathbf{D}), \tilde{Z}(\mathbf{D})) \\
&= \sigma^2\left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}'
\end{aligned}$$

and:

$$\begin{aligned}
\mathrm{cov}(\tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\beta}}) &= \left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}'\mathbf{R}^{-1}\mathrm{cov}(\tilde{Z}(\mathbf{D}), \tilde{Z}(\mathbf{D}))\mathbf{R}^{-1}\mathbf{F}\left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}\right)^{-1} \\
&= \sigma^2\left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}\right)^{-1}.
\end{aligned}$$

Therefore, the following equality stands:

$$\begin{aligned}
\mathrm{cov}(\tilde{m}_n(x), \tilde{m}_n(\tilde{x}))/\sigma^2 &= (\mathbf{f}'(x) - \mathbf{r}'(x)\mathbf{R}^{-1}\mathbf{F})\left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}\right)^{-1}(\mathbf{f}(\tilde{x}) - \mathbf{F}'\mathbf{R}^{-1}\mathbf{r}(\tilde{x})) \\
&+ \mathbf{r}'(x)\mathbf{R}^{-1}\mathbf{r}(\tilde{x}) - 2\mathbf{r}'(x)\mathbf{R}^{-1}\mathbf{F}\left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}'\mathbf{R}^{-1}\mathbf{r}(\tilde{x}) \\
&+ 2\mathbf{f}'(x)\left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}'\mathbf{R}^{-1}\mathbf{r}(\tilde{x}).
\end{aligned}$$

Third, the following equality stands:

$$\begin{aligned}
\mathrm{cov}(\tilde{Z}(x), \tilde{m}_n(\tilde{x}))/\sigma^2 &= \mathbf{r}'(x)\mathbf{R}^{-1}\mathbf{F}\left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}\right)^{-1}\mathbf{f}(\tilde{x}) \\
&+ \mathbf{r}'(x)\left(\mathbf{R}^{-1}\mathbf{r}(\tilde{x}) - \mathbf{R}^{-1}\mathbf{F}\left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}\right)^{-1}\mathbf{F}'\mathbf{R}^{-1}\mathbf{r}(\tilde{x})\right).
\end{aligned}$$

Finally, we obtain:

$$\begin{aligned}
\mathrm{cov}(\tilde{Z}_n(x), \tilde{Z}_n(\tilde{x}))/\sigma^2 &= r(x - \tilde{x}) - \mathbf{r}'(x)\mathbf{R}^{-1}\mathbf{r}(\tilde{x}) \\
&+ (\mathbf{f}'(x) - \mathbf{r}'(x)\mathbf{R}^{-1}\mathbf{F})\left(\mathbf{F}'\mathbf{R}^{-1}\mathbf{F}\right)^{-1}(\mathbf{f}(\tilde{x}) - \mathbf{F}'\mathbf{R}^{-1}\mathbf{r}(\tilde{x})).
\end{aligned}$$

Therefore, we have the following equality:

$$\mathrm{cov}(\tilde{Z}_n(x), \tilde{Z}_n(\tilde{x})) = s_n^2(x, \tilde{x}) \tag{6.35}$$

and $\tilde{Z}_n(x)$ has the same distribution as $Z_n(x)$. □

The strength of Proposition 6.2 is that it allows for sampling with respect to the distribution of $Z_n(x)$ by sampling an unconditioned Gaussian process $\tilde{Z}(x)$. The first consequence is that the conditioning of the covariance matrix is better since the variance of $\tilde{Z}(x)$ is not close to zero around points in $\mathbf{D}$. The second important consequence is that it allows for using efficient algorithms to compute realizations of $\tilde{Z}(x)$. For example, if $r(x, \tilde{x})$ is a stationary kernel,

one can use the Bochner's Theorem 1.3 and the Fourier representation of $\tilde{Z}(x)$ to compute realizations of $\tilde{Z}(x)$ as presented in Subsection 1.4.2 and in [Stein, 1999]. Furthermore, for tensorised covariance kernel (see Introduction of Section 1.4), an even more efficient method is to use the Mercer's Theorem 1.4.4 and the Nyström procedure to approximate the Karhunen-Loeve decomposition of $\tilde{Z}(x)$ as presented in Subsection 1.4.4. One of the main advantage of the Karhunen-Loeve decomposition of $Z(x)$ is that it allows for sequentially adding new points to $\mathbf{x}$ without re-estimating the decomposition. Therefore, we can easily obtain the values of a given realization $z_n(x)$ of $Z_n(x)$ at new points not in $\mathbf{x}$. This interesting property allows us to efficiently estimate the number m of Monte-Carlo particles such that the metamodel error and the Monte-Carlo estimation one are equivalent (see Subsection 6.4.2).

6.5 Multi-fidelity co-kriging based sensitivity analysis

Now let us suppose that we have s levels of code $(z_t(x))_{t=1,\dots,s}$ from the less accurate one $z_1(x)$ to the most accurate one $z_s(x)$ and that we want to perform a Global sensitivity analysis for $z_s(x)$. To surrogate $z_s(x)$, we consider the multi-fidelity co-kriging model presented in Chapter 4 Subsection 4.2.1 after integrating the posterior distribution of the regression parameters $\boldsymbol{\beta} = (\boldsymbol{\beta}_t)_{t=1,\dots,s}$ and adjustment parameters $\rho = (\rho_{t-1})_{t=2,\dots,s}$, i.e. the following predictive distribution:

$$[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, \boldsymbol{\sigma}^2], \tag{6.36}$$

where $\boldsymbol{\sigma}^2 = (\sigma_t^2)_{t=1,\dots,s}$ (see Subsection 4.2.1). We note that we consider constant adjustment coefficients $(\rho_{t-1})_{t=2,\dots,s}$. The extension to the case $\rho_{t-1}(x) = \mathbf{g}'(x)\boldsymbol{\beta}_{\rho_{t-1}}$ is straightforward (see Chapter 4). As presented in Chapter 4 Section 4.3, the predictive distribution (6.36) is not anymore Gaussian. Nevertheless, we can have closed form expressions for its mean $\mu_{n_s}^s(x)$ and covariance $k_{n_s}^s(x,\tilde{x})$:

$$\mu_{n_s}^s(x) = \hat{\rho}_{s-1}\mu_{n_{s-1}}^{s-1}(x) + \mu_{\delta_s}(x) \tag{6.37}$$

and:

$$k_{n_s}^s(x,\tilde{x}) = \hat{\sigma}_{\rho_{s-1}}^2 k_{n_{s-1}}^{s-1}(x,\tilde{x}) + k_{\delta_s}(x,\tilde{x}), \tag{6.38}$$

where for $t = 1,\dots,s$, $\begin{pmatrix}\hat{\rho}_{t-1}\\ \hat{\boldsymbol{\beta}}_t\end{pmatrix} = (\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{H}_t)^{-1}\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{z}_t$, $\mathbf{H}_t = [z_{t-1}(\mathbf{D}_t) \quad \mathbf{F}_t]$, $\mathbf{F}_t = \mathbf{f}_t'(\mathbf{D}_t)$, $\hat{\rho}_0 = 0$, $\mathbf{H}_1 = \mathbf{F}_1$, $\hat{\sigma}_{\rho_{s-1}}^2 = \hat{\rho}_{t-1}^2 + \left[(\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{H}_t)^{-1}\right]_{[1,1]}$,

$$\mu_{\delta_t}(x) = \mathbf{f}_t'(x)\hat{\boldsymbol{\beta}}_t + \mathbf{r}_t'(x)\mathbf{R}_t^{-1}(\mathbf{z}_t - \mathbf{F}_t\hat{\boldsymbol{\beta}}_t - \hat{\rho}_{t-1}z_{t-1}(\mathbf{D}^t)) \tag{6.39}$$

and

$$k_{\delta_t}(x,\tilde{x}) = \sigma_t^2\left(r_t(x,\tilde{x}) - \begin{pmatrix}\mathbf{h}_t'(x) & \mathbf{r}_t'(x)\end{pmatrix}\begin{pmatrix}0 & \mathbf{H}_t'\\ \mathbf{H}_t & \mathbf{R}_t\end{pmatrix}^{-1}\begin{pmatrix}\mathbf{h}_t(\tilde{x})\\ \mathbf{r}_t(\tilde{x})\end{pmatrix}\right), \tag{6.40}$$

with $\mathbf{h}_t'(x) = [\mu_{n_{t-1}}^{t-1}(x) \quad \mathbf{f}_t'(x)]$ and $\mathbf{h}_1'(x) = \mathbf{f}_1'(x)$.

The other notations are presented in Chapter 4 Subsection 4.2.1. We note that the variance parameter σ_t^2 is estimated with a restricted maximum likelihood method. Thus, its estimation is given by $\hat{\sigma}_t^2 = (\mathbf{z}_t - \mathbf{H}_t\hat{\boldsymbol{\beta}}_t)'\mathbf{R}_t^{-1}(\mathbf{z}_t - \mathbf{H}_t\hat{\boldsymbol{\beta}}_t)/(n_t - p_t - 1)$ where p_t is the size of $\boldsymbol{\beta}_t$.

We present in Subsection 6.5.1 the extension in a multi-fidelity framework of the Monte-Carlo estimations for the method of [Oakley and O'Hagan, 2004]. Then, we present in Subsection 6.5.2 the extension of our approach to perform co-kriging-based multi-fidelity sensitivity analysis.

6.5.1 Extension of the method of Oakley and O'Hagan for multi-fidelity co-kriging

Let us denote by $\tilde{\mathcal{S}}_{m,s}^{X^{d_1}}$ the estimation of $V^{X^{d_1}}/V$ (6.12) when we substitute $z_s(x)$ by $Z_{n,s}(x) \sim [Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, \boldsymbol{\sigma}^2]$ and when we use the Sobol procedure to perform Monte-Carlo integrations (see [Sobol, 1993] and Subsection 6.2.2). Then, the estimator suggested in [Oakley and O'Hagan, 2004] and [Marrel et al., 2009] becomes in a multi-fidelity framework:

$$\begin{aligned}\tilde{\mathcal{S}}_{m,s}^{X^{d_1}} &= \frac{\frac{1}{m}\sum_{i=1}^m k_{n_s}^s(X_i,\tilde{X}_i) + \mu_{n_s}^s(X_i)\mu_{n_s}^s(\tilde{X}_i) - \frac{1}{m^2}\sum_{i,j=1}^m k_{n_s}^s(X_i,\tilde{X}_j) + \mu_{n_s}^s(X_i)\mu_{n_s}^s(\tilde{X}_j)}{\frac{1}{m}\sum_{i=1}^m k_{n_s}^s(X_i,X_i) + \mu_{n_s}^s(X_i)\mu_{n_s}^s(X_i) - \frac{1}{m^2}\sum_{i,j=1}^m k_{n_s}^s(X_i,X_j) + \mu_{n_s}^s(X_i)\mu_{n_s}^s(X_j)} \\ &= \frac{U}{D},\end{aligned}$$

where

$$\begin{aligned}U &= \frac{1}{m}\sum_{i=1}^m \left(\sum_{t=1}^s \left(\prod_{j=t}^{s-1}\hat{\sigma}^2\rho_j\right) k_{\delta_t}(X_i,\tilde{X}_i) + \sum_{t,\tilde{t}=1}^s \left(\prod_{j=t}^{s-1}\hat{\rho}_j\right)\left(\prod_{j=\tilde{t}}^{s-1}\hat{\rho}_j\right)\mu_{\delta_t}(X_i)\mu_{\delta_{\tilde{t}}}(\tilde{X}_i)\right) \\ &\quad -\frac{1}{m^2}\sum_{i,j=1}^m \left(\sum_{t=1}^s \left(\prod_{j=t}^{s-1}\hat{\sigma}^2_{\rho_j}\right) k_{\delta_t}(X_i,\tilde{X}_j) + \sum_{t,\tilde{t}=1}^s \left(\prod_{j=t}^{s-1}\hat{\rho}_j\right)\left(\prod_{j=\tilde{t}}^{s-1}\hat{\rho}_j\right)\mu_{\delta_t}(X_i)\mu_{\delta_{\tilde{t}}}(\tilde{X}_j)\right),\end{aligned}$$

$$\begin{aligned}D &= \frac{1}{m}\sum_{i=1}^m \left(\sum_{t=1}^s \left(\prod_{j=t}^{s-1}\hat{\sigma}^2_{\rho_j}\right) k_{\delta_t}(X_i,X_i) + \sum_{t,\tilde{t}=1}^s \left(\prod_{j=t}^{s-1}\hat{\rho}_j\right)\left(\prod_{j=\tilde{t}}^{s-1}\hat{\rho}_j\right)\mu_{\delta_t}(X_i)\mu_{\delta_{\tilde{t}}}(X_i)\right) \\ &\quad -\frac{1}{m^2}\sum_{i,j=1}^m \left(\sum_{t=1}^s \left(\prod_{j=t}^{s-1}\hat{\sigma}^2_{\rho_j}\right) k_{\delta_t}(X_i,X_j) + \sum_{t,\tilde{t}=1}^s \left(\prod_{j=t}^{s-1}\hat{\rho}_j\right)\left(\prod_{j=\tilde{t}}^{s-1}\hat{\rho}_j\right)\mu_{\delta_t}(X_i)\mu_{\delta_{\tilde{t}}}(X_j)\right)\end{aligned}$$

and with the conventions $\hat{\rho}_0 = 0$, $\prod_{i=s}^{s-1}\hat{\rho}_i = 1$, $\mu_{\delta_1}(x) = \mu_{n_1}^1(x)$ and $k_{\delta_1}(x,\tilde{x}) = k_{n_1}^1(x,\tilde{x})$.

We note that $\tilde{\mathcal{S}}_{m,s}^{X^{d_1}}$ is the analogous of $\tilde{\mathcal{S}}_{m,n}^{X^{d_1}}$ presented in Subsection 6.3.2. Furthermore, the developed expression of $\tilde{\mathcal{S}}_{m,s}^{X^{d_1}}$ allows for identifying the contribution of each code level t to the sensitivity index and the one of the covariance between the bias and the code at level t. We note that the covariance here is with respect to the distribution of the input parameters X. Nevertheless, as pointed out in previous sections, this estimator is based on a ratio of expectations and thus does not correspond to the true Sobol indices.

6.5.2 Extension of the second approach for multi-fidelity co-kriging models

We present here the extension of the approach presented in Section 6.4 to the multi-fidelity co-kriging model. Therefore, we aim to sample with respect to the distribution of

$$\mathcal{S}_{m,s}^{X^{d_1}} = \frac{\frac{1}{m}\sum_{i=1}^m Z_{n,s}(X_i)Z_{n,s}(\tilde{X}_i) - \frac{1}{m}\sum_{i=1}^m Z_{n,s}(X_i)\frac{1}{m}\sum_{i=1}^m Z_{n,s}(\tilde{X}_i)}{\frac{1}{m}\sum_{i=1}^m Z_{n,s}(X_i)^2 - \left(\frac{1}{m}\sum_{i=1}^m Z_{n,s}(X_i)\right)^2}, \tag{6.41}$$

which is the analog of $\mathcal{S}_{m,n}^{X^{d_1}}$ (6.23) in an univariate case when we substitute $z(x)$ with $Z_{n,s}(x) \sim [Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, \boldsymbol{\sigma}^2]$. In fact, we can directly use Algorithm 4 by sampling realizations of $Z_{n,s}(x)$ instead of $Z_n(x)$. Moreover, the procedure presented in Subsection 6.4.2 to determine the optimal number of Monte-Carlo particles m is straightforward.

However, the distribution of $Z_{n,s}(x)$ is not Gaussian and thus the methods presented in Subsection 6.4.3 cannot be used directly. In order to handle this problem, we consider the conditional distribution $[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, \boldsymbol{\sigma}^2, \boldsymbol{\rho}, \boldsymbol{\beta}]$, with $\boldsymbol{\sigma}^2 = (\sigma_t^2)_{t=1,\dots,s}$, $\boldsymbol{\beta} = (\boldsymbol{\beta}_t)_{t=2,\dots,s}$ and $\boldsymbol{\rho} = (\rho_{t-1})_{t=2,\dots,s}$ which is Gaussian (see Chapter 4 Section 4.2). Note that we infer from $\boldsymbol{\beta}_1$. Furthermore, the Bayesian estimation of $(\rho_{t-1}, \boldsymbol{\beta}_t)$ gives us for all $t = 2, \dots, s$ (see Subsection 4.2.3):

$$\begin{pmatrix} \rho_{t-1} \\ \boldsymbol{\beta}_t \end{pmatrix} \sim \mathcal{N}\left((\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{H}_t)^{-1}\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{z}_t, \sigma_t^2(\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{H}_t)^{-1}\right). \tag{6.42}$$

Finally, thanks to the recursive formulation given in Chapter 4, we know that the following Gaussian process has the distribution $[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, \sigma^2, \rho, \boldsymbol{\beta}]$:

$$Z_{n,s,\rho,\boldsymbol{\beta}}(x) = \left(\prod_{j=1}^{s-1} \rho_j\right) Z_{n,1}(x) + \sum_{t=2}^{s-1} \left(\prod_{j=t}^{s-1} \rho_j\right) \delta_{t,\rho_{t-1},\boldsymbol{\beta}_t}(x) + \delta_{t,\rho_{t-1},\boldsymbol{\beta}_s}(x), \tag{6.43}$$

where (see equations (6.39) and (6.40)):

$$Z_{n,1}(x) \sim \mathrm{GP}\left(\mu_{\delta_1}(x), k_{\delta_1}(x, \tilde{x})\right) \tag{6.44}$$

and for $t = 2, \dots, s$:

$$\delta_{t,\rho_{t-1},\boldsymbol{\beta}_t}(x) \sim \mathrm{GP}\left(\mu_{t,\rho_{t-1},\boldsymbol{\beta}_t}(x), k_{t,\rho_{t-1},\boldsymbol{\beta}_t}(x, \tilde{x})\right), \tag{6.45}$$

with $\mu_{t,\rho_{t-1},\boldsymbol{\beta}_t}(x) = \mathbf{r}_t'(x)\mathbf{R}_t^{-1}(\mathbf{z}_t - \mathbf{F}_t\boldsymbol{\beta}_t - \rho_{t-1}z_{t-1}(\mathbf{D}^t))$, $\left((\delta_{t,\rho_{t-1},\boldsymbol{\beta}_t}(x))_{t=2,\dots,s}, Z_{n,1}(x)\right)$ independent and

$$k_{t,\rho_{t-1},\boldsymbol{\beta}_t}(x, \tilde{x}) = \sigma_t^2\left(r_t(x, \tilde{x}) - \mathbf{r}_t'(x)\mathbf{R}_t^{-1}\mathbf{r}_t(\tilde{x})\right).$$

As a consequence, we can deduce the following algorithm to compute realizations of $Z_{n,s}(x) \sim [Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, \sigma^2]$.

Algorithm 5 provides an efficient tool to sample with respect to the distribution $[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, \boldsymbol{\sigma}^2]$. Then, from each sample we can estimate the Sobol indices with a Monte-Carlo procedure. Naturally, we can easily use a bootstrap procedure to take into account the uncertainty related to the Monte-Carlo scheme. Furthermore, we see in Algorithm 5 that once a sample of $[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, \boldsymbol{\sigma}^2]$ is available, a sample for each distribution $[Z_t(x)|\mathbf{Z}^{(t)} = \mathbf{z}^{(t)}, \boldsymbol{\sigma}^2]$, $t = 1, \dots, s-1$ is also available. Therefore, we can directly in the analyze quantify the difference between the Sobol indices at a level t and the ones at another level $\tilde{t}$.

6.6 Numerical illustrations on an academic example

We illustrate here the kriging-based sensitivity analysis suggested in Section 6.4. We remind that the aim of this approach is to perform a sensitivity index taking into account both

Algorithm 5 Sampling with respect to the non-Gaussian distribution $[Z_s(x)|\mathbf{Z}^{(s)} = \mathbf{z}^{(s)}, \sigma^2]$.

1: Generate a sample $z_{n,1}(x)$ with respect to (6.44) thanks to the method presented in Proposition 6.2 in the universal kriging case.
2: Set $z_{n,s}(x) = z_{n,1}(x)$.
3: **for** t=2,. . . ,s **do**
4: Generate a sample $\begin{pmatrix} \rho^*_{t-1} \\ \boldsymbol{\beta}^*_t \end{pmatrix}$ with respect to (6.42).
5: Conditionally to $\begin{pmatrix} \rho^*_{t-1} \\ \boldsymbol{\beta}^*_t \end{pmatrix}$, generate a sample $\delta^*_{t,\rho^*_{t-1},\boldsymbol{\beta}^*_t}(x)$ with respect to (6.45) thanks to the method presented in Proposition 6.2 in the simple kriging case.
6: Set $z_{n,s}(x) = \rho^*_{t-1} z_{n,s}(x) + \delta^*_{t,\rho^*_{t-1},\boldsymbol{\beta}^*_t}(x)$.
7: **end for**
return $z_{n,s}(x)$.

the uncertainty related to the surrogate modeling and the one related to the Monte-Carlo integrations. Let us consider the Ishigami function:

$$z(x_1, x_2, x_3) = \sin(x_1) + 7\sin(\mathrm{x}_2)^2 + 0.1x_3^4\sin(x_1),$$

where μ_i is uniform on $[-\pi, \pi]$, $i = 1, 2, 3$. We are interested in the first order sensitivity indices given by

$$(S_1, S_2, S_3) = (0.314, 0.442, 0).$$

This section is organized as follows. First, in Subsection 6.6.1 we compare the Sobol index estimator $\hat{\mathcal{S}}^{X^{d_1}}_{m,n}$ (6.22) proposed by [Oakley and O'Hagan, 2004], the suggested one given by the mean of $\mathcal{S}^{X^{d_1}}_{m,n}$ (6.23) and the usual one which consists in substituting $z(x)$ by the predictive mean $m_n(x)$ (6.17) in (6.15). Then, in sections 6.6.3, 6.6.4 and 6.6.5 we deal with the approach presented in Section 6.4. In particular, we show that this approach is relevant to perform an uncertainty quantification taking into account both the uncertainty of the meta-modeling and the one of the Monte-Carlo integrations. We note that the construction of the surrogate models used in sections 6.6.3, 6.6.4 and 6.6.5 is presented in Section 6.6.2.

6.6.1 Comparison between the different methods

The aim of this subsection is to perform a numerical comparison between $\tilde{\mathcal{S}}^{X^{d_1}}_{m,n}$ (6.22), the empirical mean of $\mathcal{S}^{X^{d_1}}_{m,n}$ (6.23) given in Equation (6.24) and the following estimator (see (6.15)):

$$\check{\mathcal{S}}^{X^{d_1}}_{m,n} = \frac{\frac{1}{m}\sum_{i=1}^m m_n(X_i)m_n(\tilde{X}_i) - \left(\frac{1}{2m}\sum_{i=1}^m m_n(X_i) + m_n(\tilde{X}_i)\right)^2}{\frac{1}{m}\sum_{i=1}^m m_n(X_i)^2 - \left(\frac{1}{2m}\sum_{i=1}^m m_n(X_i) + m_n(\tilde{X}_i)\right)^2}. \tag{6.46}$$

We note that the mean $\bar{\mathcal{S}}^{X^{d_1}}_{m,n}$ of $\mathcal{S}^{X^{d_1}}_{m,n}$ is evaluated thanks to Algorithm 4, with $N_Z = 500$ and $B = 150$:

$$\bar{\mathcal{S}}^{X^{d_1}}_{m,n} = \frac{1}{N_Z B} \sum_{\substack{k=1,\dots,N_Z \\ l=1,\dots,B}} \hat{\mathcal{S}}^{X^{d_1}}_{m,n,k,l},$$

and for $\tilde{\mathcal{S}}_{m,n}^{X^{d_1}}$ and $\mathcal{S}_{m,n}^{X^{d_1}}$ we use the Monte-Carlo estimator (6.15) suggested in [Janon et al., 2012] (it is the one used in (6.46). Then, for the comparison we randomly build 100 LHS experimental design sets with $n = 40, 50, 60, 70, 90, 120, 150, 200$ observations. From these experimental design sets, we build kriging models with a constant trend β and a tensorised 5/2-Matérn kernel. Furthermore, the characteristic length scales $(\theta_i)_{i=1,2,3}$ are estimated with a maximum likelihood procedure for each design set. The Nash-Sutcliffe model efficiencies,

$$\mathit{Eff}_n = 1 - \frac{\sum_{x\in T}(m_n(x) - z(x))^2}{\sum_{x\in T}(m_n(x) - \bar{z}(x))^2}, \quad \bar{z}(x) = \frac{1}{\#T}\sum_{x\in T} z(x),$$

of the different kriging models are evaluated on a test set T composed of 10,000 points uniformly spread on the input parameter space $[-\pi, \pi]^3$. The values of Eff_n are presented in Figure 6.1.

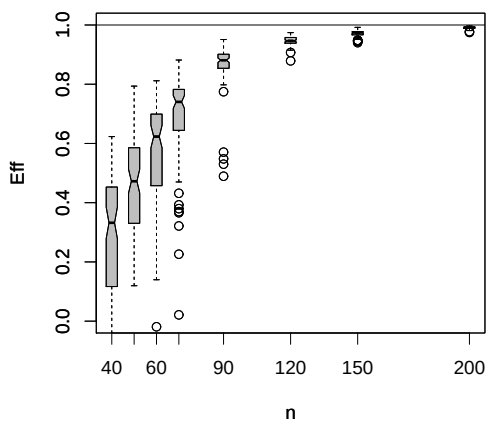
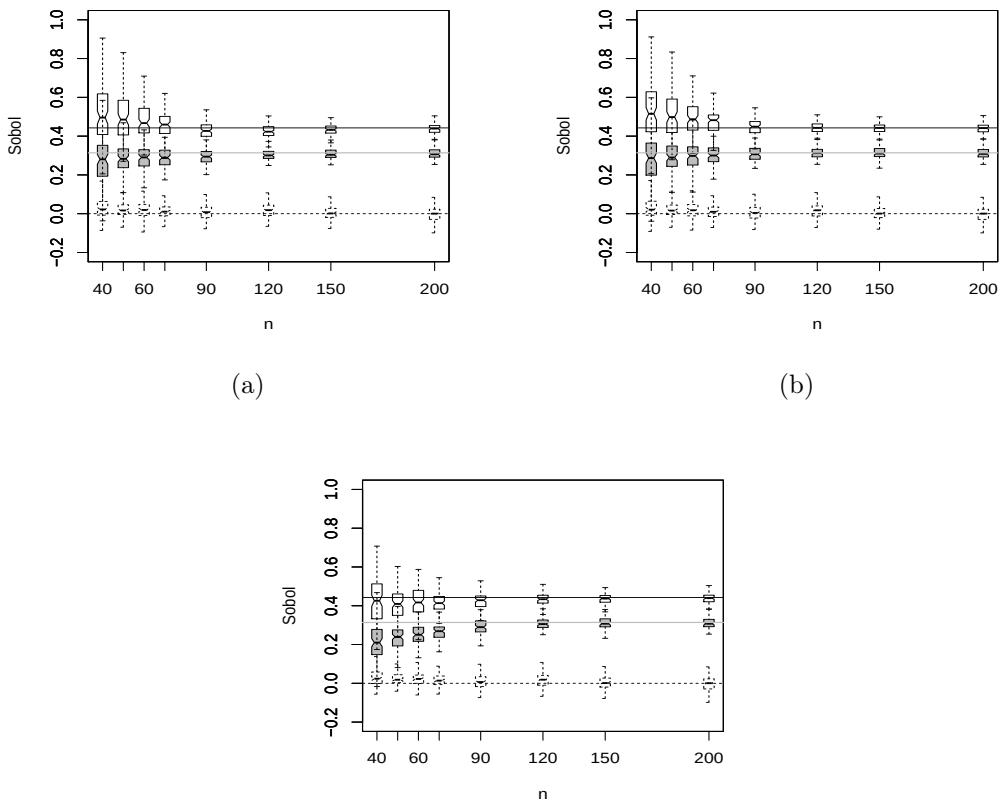


Figure 6.1: Convergence of the model efficiency when the number n of observations increases. 100 LHS experimental design sets are randomly sampled for each number of observations n. The closer Eff is to 1, the more accurate is the model $m_n(x)$.

Figure 6.2 illustrates the Sobol index estimates obtained with the three methods. We see in Figure 6.2 that the suggested estimator $\bar{\mathcal{S}}_{m,n}^{X^{d_1}}$ performs as well as the usual estimator $\check{\mathcal{S}}_{m,n}^{X^{d_1}}$ (6.46). In fact, as we will see in the next subsections, the strength of the suggested estimator is to provide more relevant uncertainty quantification. Finally, we see in Figure 6.2c that the estimator $\tilde{\mathcal{S}}_{m,n}^{X^{d_1}}$ (6.22) suggested in [Oakley and O'Hagan, 2004] seems to underestimate the true value of the Sobol index.

6.6.2 Model building and Monte-Carlo based estimator

For the numerical illustrations in sections 6.6.3, 6.6.4 and 6.6.5, we use different kriging models built from different experimental design sets of size $n = 30, \ldots, 200$. They are LHS optimized with respect to the centered L_2-discrepancy criterion. The design sets are built thanks to



(a) (b)

(c)

Figure 6.2: Comparison between three Sobol index estimators. The comparison are performed from 100 random LHS experimental design sets for each number of observations n. Figure (a) corresponds to the suggested Sobol estimator (see Section 6.4), Figure (b) corresponds to the usual estimator (see Equation (6.46)) and Figure (c) corresponds to the estimator suggested in [Oakley and O'Hagan, 2004]. The horizontal lines represent the true values of the Sobol indices (solid gray line: S_1; solid black line: S_2 and dashed black line: S_3)

R CRAN package "DiceDesign" Furthermore, for all kriging models, we consider a constant trend β and a tensorised 5/2-Matérn kernel (see Section 1.4).

The characteristic length scales $(\theta_i)_{i=1,2,3}$ are estimated for each experimental design set by maximizing the marginal likelihood. Furthermore, the variance parameter σ^2 and the trend parameter β are estimated with a maximum likelihood method for each experimental design set too. Then for each n, the Nash-Sutcliffe model efficiency is evaluated on a test set composed of 10,000 points uniformly spread on the input parameter space $[-\pi, \pi]^3$. Figure 6.3 illustrates the estimated values of Eff_n with respect to the number of observations n.

Then, for estimating the Sobol indices, we use the Monte-Carlo based estimator given by (6.15). It has the strength to be asymptotically efficient for the first order indices (see [Janon et al., 2012]).

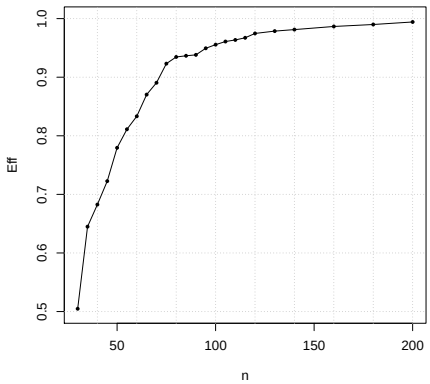


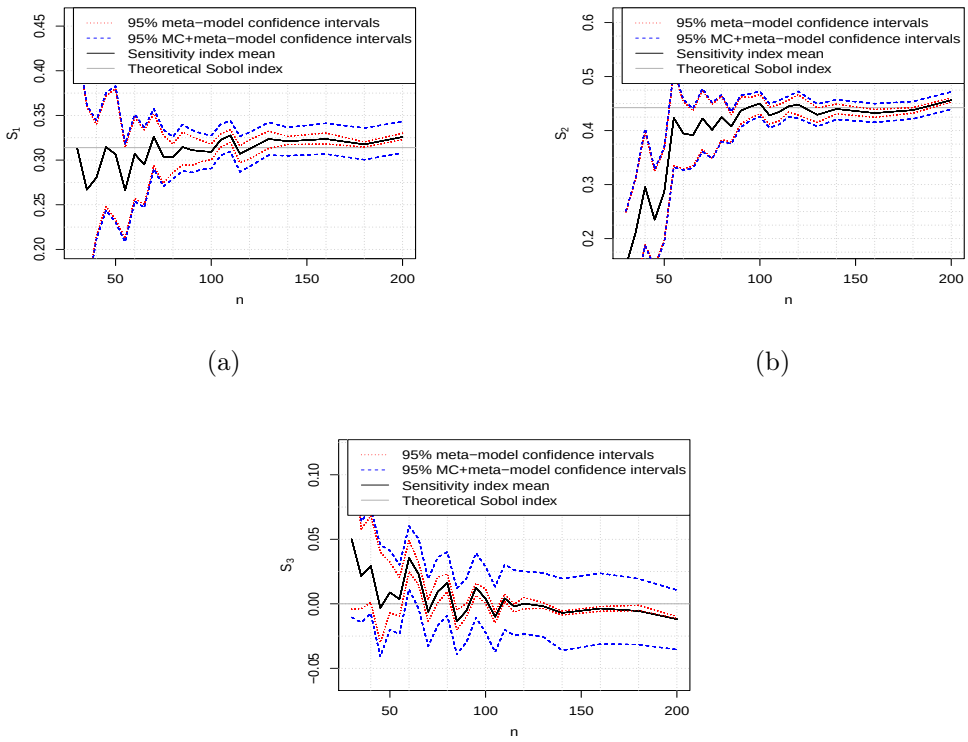
Figure 6.3: Convergence of the model efficiency when the number n of observations increases. For each number of observations n, the experimental design set is a LHS optimized with respect to the centered L_2-discrepancy. The closer *Eff* is to 1, the more accurate is the model $m_n(x)$.

6.6.3 Sensitivity index estimates when n increases

Let us consider a fixed number of Monte-Carlo particles $m = 10,000$. The aim of this subsection is to quantify the part of the index estimator uncertainty related to the Monte-Carlo integrations and the one related to the surrogate modeling.

To perform such analysis we use the procedure presented in Algorithm 4 with $B = 300$ bootstrap samples and $N_Z = 500$ realizations of $Z_n(x)$ (6.17). It results for each $i = 1, 2, 3$ a sample $\left(\hat{S}^i_{m,n,k,l}\right)$, $k = 1, \ldots, N_Z$, $l = 1, \ldots, B$, with respect to the distribution of the estimator obtained by substituting $z(x)$ with $Z_n(x)$ in (6.15).

Then, we estimate the 0.05 and 0.95 quantiles of $\left(\hat{S}^i_{m,n,k,1}\right)$, $k = 1, \ldots, N_Z$ for each $i = 1, 2, 3$ with a bootstrap procedure. The resulting quantiles represent the uncertainty related to the surrogate modeling. Furthermore, we estimate the 2.50% and 97.50% quantiles of $\left(\hat{S}^i_{m,n,k,l}\right)$, $k = 1, \ldots, N_Z$, $l = 1, \ldots, B$ with a bootstrap procedure too. These quantiles represent the total uncertainty of the index estimator. Figure 6.4 illustrates the result of this procedure for different numbers of observations n. We see in Figure 6.4 that for small values of n, the error related to the surrogate modeling dominates. Then, when n increases, this error decreases and it is the one related to the Monte-Carlo integrations which is the largest. This emphasizes that it is worth to adapt the number of Monte-Carlo particles m to the number of observations n. Finally, we highlight that the equilibrium between the two types of uncertainty does not occur for the same n for the three indices. Indeed, it is around $n = 100$ for S_1, $n = 150$ for S_2 and around $n = 75$ for S_3. We observe that the smaller the index is, the larger its Monte-Carlo estimation error is.



(a) (b)

(c)

Figure 6.4: Sensitivity index estimates when n increases. The solid black lines represent the means of the sensitivity index estimators. The dotted red lines represent the 2.50% and 97.50% confidence intervals taking into account only the uncertainty related to the surrogate modeling. The dashed blue lines represent the 2.50% and 97.50% confidence intervals taking into account both the uncertainty related to the surrogate modeling and the one related to the Monte-Carlo integrations. The horizontal gray lines represent the true values of S_1 (a), S_2 (b) and S_3 (c).

6.6.4 Optimal Monte-Carlo resource when n increases

We saw in the previous subsection that the equilibrium between the error related to the Monte-Carlo integrations and the one related to the surrogate modeling depends on the considered sensitivity index. The purpose of this subsection is to determine this equilibrium for each index. To perform such analysis, we use the method presented in Subsection 6.4.2.

Let us consider a sample $\left(\hat{S}^i_{m,n,k,l}\right)$, $m = 30, \ldots, 200$, $k = 1, \ldots, N_Z$, $l = 1, \ldots, B$, $i = 1, 2, 3$, generated with Algorithm 4 and using the Monte-Carlo estimator presented in (6.15). For each pair (m, n) we can evaluate the variance $\hat{\sigma}^2_{Z_n}\left(S^i_{m,n}\right)$, $i = 1, 2, 3$, related to the meta-modeling with Equation (6.26) and the variance $\hat{\sigma}^2_{MC}\left(S^i_{m,n}\right)$, $i = 1, 2, 3$, related to the Monte-Carlo integrations with Equation (6.27). We state that the equilibrium between the

two types of uncertainty corresponds to the case

$$\hat{\sigma}^2_{Z_n}\left(S^i_{m,n}\right) = \hat{\sigma}^2_{MC}\left(S^i_{m,n}\right). \tag{6.47}$$

We present in Figure 6.5 the pairs (m, n) such that the equality (6.47) is satisfied. We see that the smaller is the sensitivity index, the more important is the number of particles m required to have the equilibrium. Furthermore, we note that the curve increases extremely quickly for the index $S_3 = 0$. Therefore, it could be unrealistic to consider the equilibrium for this case, especially when n is important (i.e. $n > 100$).

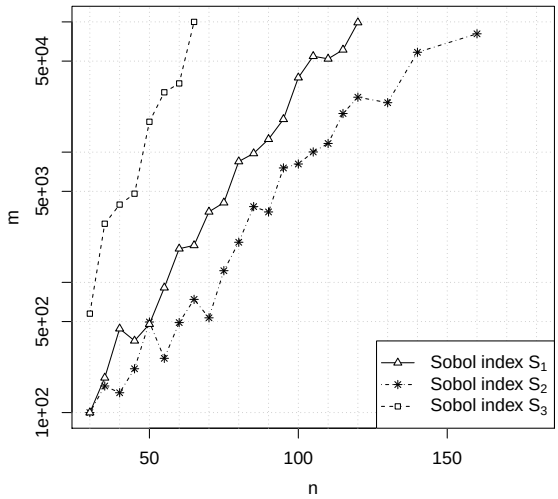


Figure 6.5: Relation between the number of observations n and the number of Monte-Carlo particles m such that the error related to the meta-modeling and the one related to the Monte-Carlo scheme have the same order of magnitude.

The presented analysis is of practical interest since it provides the appropriate number of Monte-Carlo particles m for the sensitivity index estimation in function of the number of observations n. Furthermore, in the framework of computer experiments, the observations are often time-consuming and n cannot be large. Therefore, we look for a number of particles m such that the variance $\hat{\sigma}^2_{Z_n}\left(S^i_{m,n}\right)$ related to the meta-modeling is smaller than the one of the Monte-Carlo integration $\hat{\sigma}^2_{MC}\left(S^i_{m,n}\right)$. However, we saw that it could be unfeasible for some values of sensitivity index. In this case a compromise must necessarily be done.

6.6.5 Coverage rate of the suggested Sobol index estimator

Algorithm 4 in Subsection 6.4.1 allows for obtaining a sample $\left(\hat{S}^i_{m,n,k,l}\right)$, $k = 1, \ldots, N_Z$, $l = 1, \ldots, B$ of the estimator of S_i for each $i = 1, 2, 3$. The purpose of this subsection is to verify the relevance of the confidence intervals provided by $\left(\hat{S}^i_{m,n,k,l}\right)$. To perform such

analysis, we generate 200 random LHS $(\mathbf{D}_{n,j})_{j=1,\dots,200}$ for different numbers of observations n. For each $\mathbf{D}_{n,j}$, we build a kriging model with the procedure presented in Subsection 6.6.2 and we generate a sample $\left(\hat{S}^i_{m,n,k,l}\right)$, $k = 1, \dots, N_Z$, $l = 1, \dots, B$, with $B = 200$ and $N_Z = 300$. The efficiency of the different kriging models with respect to the number of observation n is presented in Figure 6.6. From this sample, we evaluate the 2.50% and 97.50% quantiles with a bootstrap procedure and we check if the true value of S_i is covered by these two quantiles. At the end of the procedure, the ratio between the number of confidence intervals covering the true value of S_i and the total number of confidence intervals (i.e. 200) has to be close to 95% for each n.

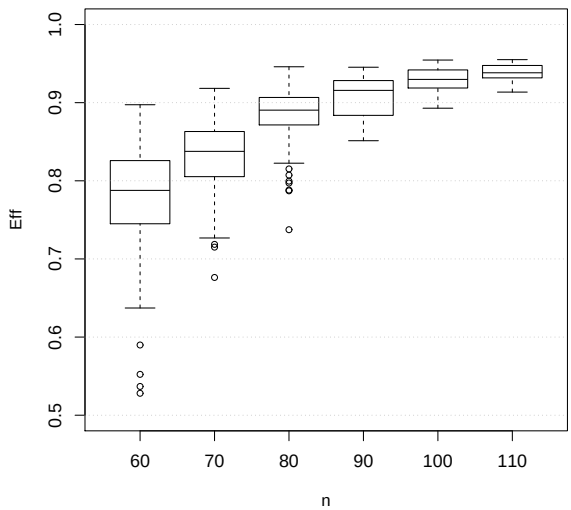


Figure 6.6: Convergence of the model efficiency when the number n of observations increases. For each number of observations n, 200 LHS experimental design sets are randomly sampled. The closer *Eff* is to 1, the more accurate is the model $m_n(x)$.

Furthermore, to perform the analysis we use different values of m according to the procedure presented in Subsection 6.4.2 for S_1 and S_2 (i.e. such that the variance related to the meta-modeling has the same order of magnitude than the one related to the Monte-Carlo integrations). For S_3, the number of Monte-Carlo particles m increases too quickly with respect to n to use the method presented in Subsection 6.4.2. Therefore we fix m to the values presented in Table 6.1. We note that the values of m for S_3 are larger than the ones for S_1 and S_2.

n	60	70	80	90	100	110
m	1,000	3,000	5,000	10,000	40,000	60,000

Table 6.1: Numbers of Monte-Carlo particles m for different values of the number of observations n for the estimation of S_3.

The empirical 95%-confidence intervals as a function of the number of observations n are presented in Figure 6.7. We study three cases:

1. The confidence intervals are built from $\left(\hat{S}^i_{m,n,k,l}\right)$, $k = 1, \ldots, N_Z$, $l = 1, \ldots, B$. Therefore, it takes into account both the uncertainty related to the meta-model and the one related to the Monte-Carlo estimations.
2. The confidence intervals are built from $\left(\hat{S}^i_{m,n,k,1}\right)$, $k = 1, \ldots, N_Z$. In this case, we do not use the bootstrap procedure to evaluate the uncertainty due to the Monte-Carlo procedure. Therefore, we only take into account the one due to the meta-model.
3. The confidence intervals are built from the estimator $\tilde{\mathcal{S}}^{X^{d_1}}_{m,n}$ (6.46) with a bootstrap procedure. Here, we estimate the Sobol indices with the kriging mean and we do not infer from the uncertainty of the meta-model. Therefore, we only take into account the uncertainty related to the Monte-Carlo estimations.

We see in Figure 6.7 that the confidence intervals provided by the approach presented in Section 6.4 are well evaluated for indices S_1 and S_3. Furthermore, they are underestimated when we take into account only the meta-model or the Monte-Carlo uncertainty. This highlights the relevance of the suggested approach to perform uncertainty quantification on the Sobol index estimates. However, the coverage rate is underestimated for index S_2. This is even worst if we only consider the meta-model error. This may be due to a poor learning in the direction x_2 for the the surrogate model. This emphasizes that the suggested method is valid only if the kriging variance well represents the modeling error.

6.7 Application of multi-fidelity sensitivity analysis

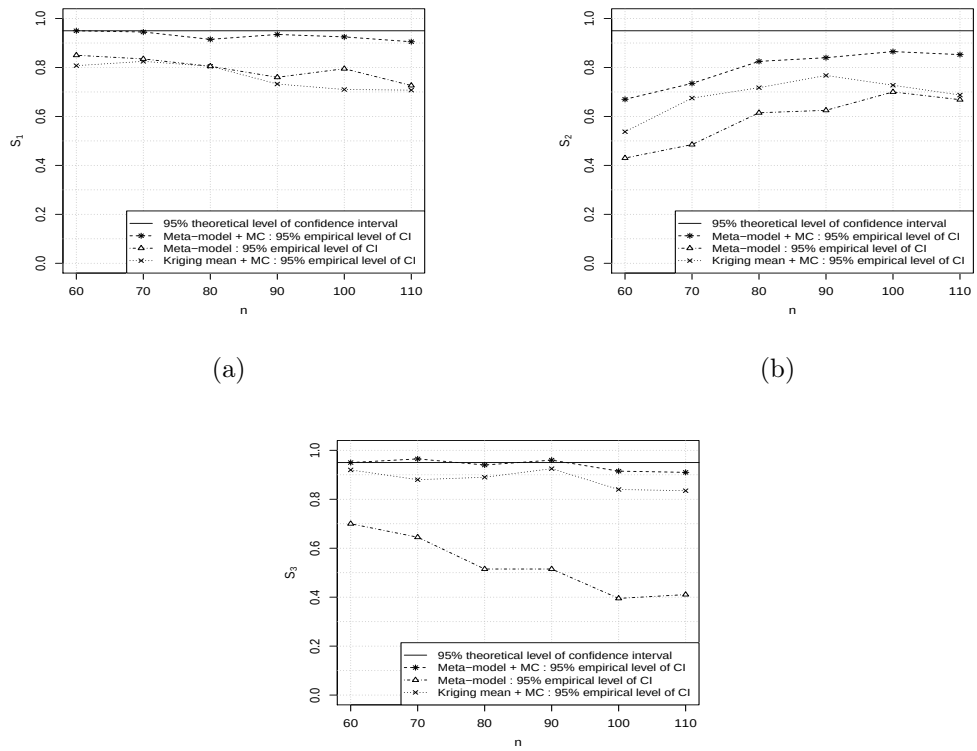
In this section, we illustrate the multi-fidelity co-kriging based sensitivity analysis presented in Section 6.5 on the example about a spherical tank under internal pressure presented in Chapter 5 Section 5.3.

The scheme of the considered tank is presented in Figure 5.6. We are interested in the von Mises stress at the point labeled 2 in Figure 5.6.

The physical system depends on 8 parameters and the von Mises stress $z_2(x)$ at point $x = (P, R_{int}, T_{shell}, T_{cap}, E_{shell}, E_{cap}, \sigma_{y,shell}, \sigma_{y,cap})$ is provided by an Aster finite elements code.

The cheaper version $z_1(x)$ of $z_2(x)$ is obtained by the 1D simplification of the tank corresponding to a perfect spherical tank, i.e. without the cap:

$$z_1(x) = \frac{3}{2}\frac{(R_{int} + T_{shell})^3}{(R_{int} + T_{shell})^3 - R_{int}^3}P.$$



(a) (b)

(c)

Figure 6.7: Empirical 95% confidence intervals with respect to the number of observations n for S_1 (a), S_2 (b) and S_3 (c). The empirical coverage rates are evaluated from 200 kriging models build from different random LHS design sets.

6.7.1 Multi-fidelity model building

We present here the construction of the model presented in Section 6.5. For the implementation, we use the R CRAN package "MuFiCokriging" presented in Chapter 4 Section 4.6.

First, we build two LHS design sets $\tilde{\mathbf{D}}_1$ and $\mathbf{D}_2$ of size $n_1 \times 8$ and $n_2 \times 8$ optimized with respect to the centered L_2-discrepancy criterion, with $n_1 = 100$ and $n_2 = 20$. We note that the input parameter x is normalized so that the measure $\mu(x)$ of the input parameters is uniform on $[0, 1]^8$. In order to respect the nested property for the experimental design sets, we remove from $\tilde{\mathbf{D}}_1$ the n_2 points that are the closest to those of $\mathbf{D}_2$ and we set that $\mathbf{D}_1$ is the concatenation of $\mathbf{D}_2$ and $\tilde{\mathbf{D}}_1$. This procedure ensures that $\mathbf{D}_2 \subset \mathbf{D}_1$ without operates any transformation on $\mathbf{D}_2$ (see Algorithm 1 in Chapter 4 Section 4.5).

Second, we run the expensive code $z_2(x)$ on points in $\mathbf{D}_2$ and the coarse code $z_1(x)$ on points in $\mathbf{D}_1$. The CPU time of the expensive code is around 1 minute. Furthermore, in order to have a fair illustration, we consider that the CPU time of the coarse code $z_1(x)$ is not negligible and we restrict its runs to $n_1 = 100$.

Third, we use tensorised 5/2-Matérn covariance kernels for $\sigma_1^2 r_1(x, \tilde{x})$ and $\sigma_2^2 r_2(x, \tilde{x})$ with

characteristic length scales $(\theta_1^i)_{i=1,\dots,8}$ and $(\theta_2^i)_{i=1,\dots,8}$. Furthermore, we set that the regression functions are constants, i.e. $\mathbf{f}_1(x) = 1$ and $\mathbf{f}_2(x) = 1$.

The estimates of the characteristic length scales are given in Table 6.2.

$\hat{\boldsymbol{\theta}}_1$	1.71	1.38	1.97	1.98	1.98	1.99	1.95	1.41
$\hat{\boldsymbol{\theta}}_2$	1.83	1.89	0.5	1.93	1.93	0.64	1.89	0.79

Table 6.2: Maximum likelihood estimates of the characteristic length scales of the tensorised 5/2-Matérn covariance kernels use in the multi-fidelity co-kriging model. $\hat{\boldsymbol{\theta}}_1$ represents the estimates for the code level 1 and $\hat{\boldsymbol{\theta}}_2$ represents the ones for the bias between the code levels 1 and 2.

The estimates of the characteristic length scales given in Table 6.3 show that the model is very smooth. Then, Table gives the posterior mean of the parameters $(\rho_1, \boldsymbol{\beta}_2)$ and $\boldsymbol{\beta}_1$ and the restricted maximum likelihood estimates of σ_1^2 and σ_2^2.

$\hat{\boldsymbol{\beta}}_1$	148.67	$\hat{\sigma}_1^2$	495.63
$(\hat{\rho}_1, \hat{\boldsymbol{\beta}}_2)$	(0.92, 57.61)	$\hat{\sigma}_2^2$	551.07

Table 6.3: Posterior means of the trend parameters $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ and the adjustment parameter ρ_1 and maximum likelihood estimates of the variance parameters σ_1^2 and σ_2^2.

The parameter estimates presented in Table 6.3 show that there is an important bias between the cheap code and the expensive code since $\hat{\boldsymbol{\beta}}_2 \approx 58$ whereas the trend of the cheap code is $\hat{\boldsymbol{\beta}}_1 \approx 150$. In particular, it is greater than the standard deviation of the bias which is $\hat{\sigma}_2 \approx 23$. Then, the posterior mean of the adjustment parameter $\hat{\rho}_1 = 0.92$ does not indicate a perfect correlation between the two levels of code. Indeed, the estimated correlation between $z_2(x)$ and $z_1(x)$ is 0.77. Furthermore their estimated variance equals 1514 for $z_2(x)$ and 810 for $z_1(x)$. In fact, we remind that the adjustment parameter:

$$\rho_1 = \frac{\mathrm{cov}(Z_2(x), Z_1(x))}{\mathrm{var}(Z_1(x))}$$

represents both the correlation degree and the scale factor between the codes $z_2(x)$ and $z_1(x)$.

Finally, we can estimate the accuracy of the suggested model with a Leave-One-Out cross validation procedure. From the Leave-One-Out errors, we estimate the Nash-Sutcliffe model efficiency $\mathit{Eff}_{LOO} = 83\%$. This means that the suggested multi-fidelity co-kriging model explains 83% of the variability of the model. We note that the closer Eff_{LOO} is to 1, the more accurate is the model. Therefore, we have an excellent model despite the small number of observations $n_2 = 20$ used for the expensive code $z_2(x)$. In order to strengthen this result, we test the multi-fidelity model on an external test set of $7,000$ points and the estimated efficiency is 86% which is even better.

6.7.2 Multi-fidelity sensitivity analysis

Now let us perform a multi-fidelity sensitivity analysis using the approach presented in Subsection 6.5.2. We are interested in the first-order sensitivity indices.

The principle of the method is to sample from the distribution (6.41) using Algorithm 5. We note that we use the Monte-Carlo estimator (6.15) instead of (6.14) since it is asymptotically efficient for the first-order indices. We repeat Algorithm 5 to have $N_Z = 200$ realizations of the predictive distribution $[Z_2(x)|\mathbf{Z}^{(2)} = \mathbf{z}^{(2)}, \boldsymbol{\sigma}^2]$ and for each realization we generate $B = 150$ bootstrap samples. Furthermore, we choose $m = 20,000$ for the Monte-Carlo sampling size so that the error related to the Monte-Carlo integrations is negligible compared to the one related to the surrogate modeling (see subsections 6.4.2 and 6.6.4).

Sensitivity analysis for the cheap code.

First, let us present the result of the sensitivity analysis for the cheap code. As emphasized in Subsection 6.5.2, once samples with respect to the distribution $[Z_2(x)|\mathbf{Z}^{(2)} = \mathbf{z}^{(2)}, \boldsymbol{\sigma}^2]$ are available, samples for $[Z_1(x)|\mathbf{Z}^{(1)} = \mathbf{z}^{(1)}, \sigma_1^2]$ are also available. Therefore, from them we can perform a sensitivity analysis as presented in Section 6.4. Moreover, from the explicit formula of $z_1(x)$ we expect that only the three variables P, R_{int} and T_{shell} have an impact on the output.

The result of the sensitivity analysis for the cheap code $z_1(x)$ is given in Figure 6.8. We see in Figure 6.8 that only the three parameters P, R_{int} and T_{shell} are influent as expected. Furthermore, the internal pressure is the most important parameter whereas the geometrical parameter R_{int} and T_{shell} have equivalent impact on the output. The sum of the first-order sensitivity index means informs us that 97% of the variability of the output is explained by the first-order indices. The interactions between the parameters are thus negligible. Further, we see that the confidence intervals are tight and that the uncertainty on the Sobol index estimator is essentially related to the Monte-Carlo integrations. This means that the model's error on the cheap code is very low.

Sensitivity analysis for the expensive code.

Second, we perform a sensitivity analysis for the expensive code $z_2(x)$ using the predictive distribution $[Z_2(x)|\mathbf{Z}^{(2)} = \mathbf{z}^{(2)}, \boldsymbol{\sigma}^2]$. The result of the analysis is presented in Figure 6.9.

We see in Figure 6.9 that the result of the sensitivity analysis for the expensive code is substantially different than the one for the cheap code. First, the importance measure of the parameters P, R_{int} and T_{shell} decreases although the internal pressure P remains the most influent parameter. Second, the material parameters E_{shell}, E_{cap}, $\sigma_{y,shell}$ and $\sigma_{y,cap}$ have still a negligible influence except for the rigidity of the cap E_{cap}. Then, the most noticeable difference is for the thickness of the cap T_{cap} which is now the second most important parameter. Finally, the sum of the index estimator means equals 96.7%. This means that the first order indices still explain the main part of the model variability.

The hierarchy between the parameters can be easily interpreted. Indeed, the coarse code corresponds to the approximation of the tank without the cap. Therefore, it is natural that the parameters related to the cap have no influence. On the contrary, for the expensive

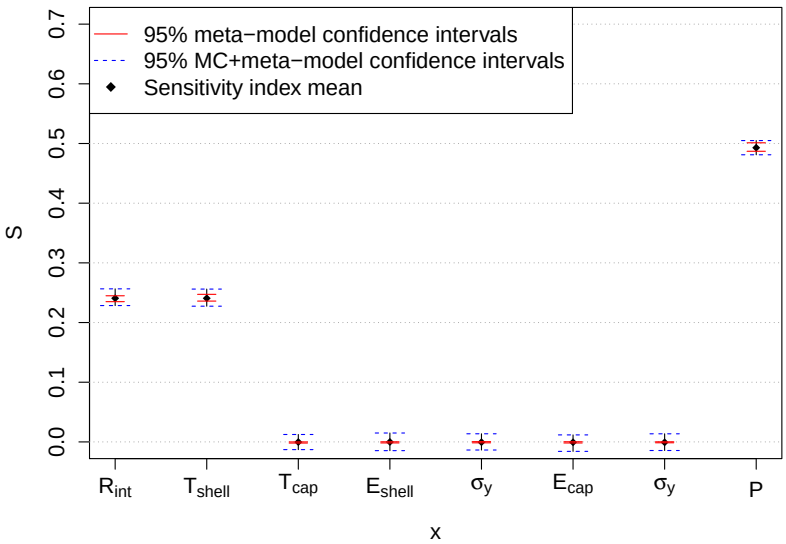


Figure 6.8: Kriging based sensitivity analysis for the cheap code. The diamonds represent the means of the first-order sensitivity index estimators, the solid red lines represent the 95% confidence intervals taking into account only the meta-modeling uncertainty and the dashed blue lines represent the 95% confidence intervals taking into account the uncertainty related to both the Monte-Carlo integrations and the meta-modeling. The means and the confidence intervals are obtained with Algorithm 4.

code, we are interested in the von Mises stress at the junction between the cap and the shell. Consequently, the parameters related to the cap have now an influence. However, it was difficult to have a prior on the impact of the cap onto the response variability. We deduce from this analysis that it is in fact very important.

For the material parameters, their influences are negligible because we are in the regime of elastic deformations. It is thus physically coherent. In fact, they would be more influent in a plastic deformation regime which can occur for more important internal pressure P.

The other important differences between the two sensitivity analysis is the magnitude of the confidence intervals. Indeed, we see in Figure 6.9 that contrary to the cheap code, the confidence intervals for the sensitivity index estimators of the expensive code are very large. Therefore, despite the good multi-fidelity approximation of the expensive code, we have an important uncertainty on it. This is natural since we only use 20 runs of $z_2(x)$ to learn it. Finally, we note that the most important uncertainty is for T_{cap}. This is explained by the fact that this parameter is not considered by the cheap code. Therefore, $z_1(x)$ brings no information about T_{cap} contrary to R_{int}, T_{shell} and P.

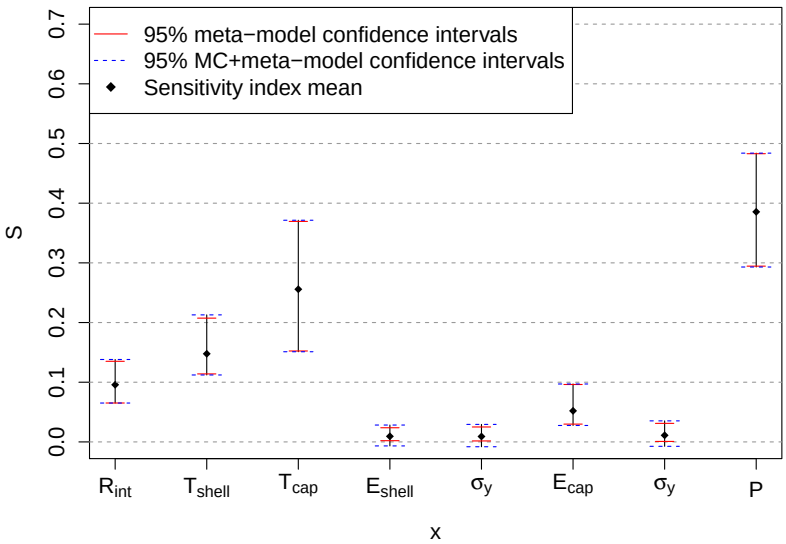


Figure 6.9: Co-kriging based sensitivity analysis for the expensive code. The diamonds represent the means of the first order sensitivity index estimators, the solid red lines represent the 95% confidence intervals taking into account only the meta-modeling uncertainty and the dashed blue lines represent the 95% confidence intervals taking into account the uncertainty related to both the Monte-Carlo integrations and the meta-modeling.

6.8 Conclusion

This chapter deals with the sensitivity analysis of complex computer codes using Gaussian process regression. The purpose of the chapter is to build Sobol index estimators taking into account both the uncertainty related to the surrogate modeling and the one related to the numerical evaluations of the variances and covariances involved in the Sobol index definition. The aim is to provide relevant confidence intervals for the index estimator.

To provide such estimators, we suggest a method which mixes a Gaussian process regression model with Monte-Carlo based integrations. From it, we can quantify the impact of both the Gaussian process regression and the Monte-Carlo procedure on the index estimator variability. In particular, we present a procedure to equilibrate these two sources of uncertainty. Furthermore, we suggest numerical methods to avoid ill-conditioned problems and to easily handle the suggested index estimator.

Then, we propose an extension of the suggested approach for multi-fidelity computer codes. They are of practical interest since they allow for dealing with the problem of very expensive simulations. To deal with these codes, we use the multi-fidelity co-kriging model presented in Chapter 4.

Finally, we illustrate the suggested strategy on an academic example for the univariate case and with a real application on a tank under internal pressure for the multi-fidelity analysis.

Part III

Contributions in noisy-kriging

Introduction to Monte-Carlo simulators

Context

For many realistic cases, we do not have direct access to the function $f(x)$ to be approximated but only to noisy versions of it (as presented in Chapter 1 Subsection 1.2.1 Paragraph "The noisy case", we use the notation $f(x)$ to design a function for which we have noisy observations). For example, if the objective function is the result of an experiment, the available responses can be tainted by measurement noise. In that case, we can reduce the noise of the observations by repeating the experiments at the same locations. Another example is the Monte-Carlo based simulators - also called stochastic simulators - which use Monte-Carlo or Monte-Carlo Markov Chain methods to solve a system of partial differential equations through its probabilistic interpretation. For such simulators, the noise level can be tuned by the number of Monte-Carlo particles used in the procedure.

As presented in Subsection 1.2.1, Gaussian process regression can easily be adapted to the case of noisy observations. Recently, many authors were interested in kriging models in a stochastic simulator framework ([Kleijnen and Van Beers, 2005], [Picheny, 2009], [Boukouvalas and Cornford, 2009], [Marrel et al., 2010], [Yin et al., 2011] and [Kleijnen, 2012]). In particular, [Kleijnen and Van Beers, 2005], [Boukouvalas and Cornford, 2009] and [Yin et al., 2011] deal with heteroscedastic noises, [Marrel et al., 2010] deal with noisy-kriging-based global sensitivity analysis and [Picheny, 2009] addresses the problem of optimal resource allocation. The aim of this chapter is to introduce the framework of stochastic simulators. We note that the presented result can also be used in the framework of experiments with repetitions.

As an introductory example, let us consider $f_{s_1}(x)$ the output of a stochastic simulator obtained with s_1 Monte-Carlo particles $((Y_i(x))_{i=1,\dots,s_1}$. Furthermore, let us consider $f_{s_2}(x)$, $s_2 > s_1$ the output of the same simulator obtained from the particles $((Y_i(x))_{i=1,\dots,s_1}$ and $((Y_i(x))_{i=s_1+1,\dots,s_2}$. In that example, $f_{s_2}(x)$ is more accurate and time-consuming than $f_{s_1}(x)$. Now, let us suppose that we want to surrogate $f_{s_2}(x)$ using both the information of the observations of $f_{s_2}(x)$ and $f_{s_1}(x)$ at points in $\mathbf{D}_2$ and $\mathbf{D}_1$ such that $\mathbf{D}_2 \subset \mathbf{D}_1$. Considering the models presented in Part II, we are tempted to use a multi-fidelity co-kriging approach. We show in this chapter that it is equivalent to use a noisy-kriging approach with heterogeneous observation noise variances.

A multi-fidelity approach being equivalent to a noisy-kriging one and the number of Monte-Carlo particles monitoring the observation noise level, for a fixed number of M-C particles a question of interest is to find the best allocation of the Monte-Carlo particles into the points of the experimental design set. This point was originally addressed in the linear regression theory. A pioneering work is the one of [Elfving, 1952] which deals with the optimal resource allocation with respect to criteria such as G-optimality or D-optimality (see [Fedorov, 1972]). The G-optimality aims to minimize the maximum of the predictive variance, i.e. $\max_{x\in Q} s^2(x)$ in a kriging framework (see Subsection 1.2.1) and the D-optimality addresses the problem of minimizing the determinant of the information matrix $\mathbf{F}'\mathbf{F}$. We note that the D-optimality cannot be used in a kriging framework since it works only for linear models. Then, many authors deal with the problem of optimal design in a linear regression framework by suggesting other optimality criteria and algorithms of construction ([Kiefer and Wolfowitz, 1959], [Kiefer, 1961], [Fedorov, 1972], [Wu, 1978], [Cook and Nachtrheim, 1980], [Fedorov and Hackl, 1997] and [Molchanov and Zuyev, 2002]). Furthermore, [Picheny, 2009] presents an exploratory work on optimal design for noisy kriging.

We give in Chapter 7 a proposition providing an optimal resource allocation under certain restricted conditions for heteroscedastic noisy kriging models and with respect to the I-optimality. The I-optimality corresponds to the minimization of the averaged predictive variance, i.e. $\int s^2(x)\, d\mu(x)$ in a kriging framework (see Subsection 1.2.1). Furthermore, we numerically observe in Appendix D that this allocation remains efficient in more general cases although it is not anymore optimal.

Stochastic simulators and noisy-kriging models

Let us consider that we want to approximate the function

$$\begin{aligned} f : Q \subset \quad \mathbb{R}^d \quad &\to \mathbb{R} \\ x \quad &\mapsto f(x), \end{aligned}$$

from noisy observations at points $\mathbf{D} = (x_i)_{i=1,\dots,n}$ sampled from the design measure μ and with s_i replications at each point x_i, $i = 1, \dots, n$. We hence have $(\sum_{i=1}^n s_i)$ data of the form:

$$z_{i,j} = f(x_i) + \sigma_\varepsilon(x_i)\varepsilon_{i,j}$$

and we consider that $(\varepsilon_{i,j})_{\substack{i=1,\dots,n \\ j=1,\dots,s_i}}$ are independently distributed from a Gaussian distribution with mean zero and variance one. Such a function can represent the output of a stochastic simulator or the observation of an experiment. We present below the framework of stochastic simulators and the use of co-kriging models to surrogate such computer codes.

Stochastic simulators

In a framework of stochastic simulators or experiments with repetitions, we consider outputs of the form:

$$z_{s_i} = \frac{1}{s_i} \sum_{j=1}^{s_i} z_{i,j}. \tag{6.48}$$

We note that for stochastic simulators s_i represents the number of Monte-Carlo particles and for experiments s_i represents the number of repetitions. Therefore, denoting the vector of the observed values by $\mathbf{z}^n = (z_{s_i})_{i=1,\dots,n} = (\sum_{j=1}^{s_i} z_{i,j}/s_i)_{i=1,\dots,n}$, the variance of an observation z_{s_i} is

$$\mathrm{var}\,(z_{s_i}) = \frac{\sigma_\varepsilon^2(x_i)}{s_i}.$$

The accuracy of an observation is hence inversely proportional to the number of Monte-Carlo particles s_i. Furthermore, we define the budget T as follows:

$$T = \sum_{i=1}^{n} s_i. \tag{6.49}$$

Let us consider the outputs of two code levels $z_{s_i^1}$ and $z_{s_i^2}$, $i = 1, \dots, n$, such that $s_i^1 < s_i^2$

$$z_{s_i^1} = \frac{1}{s_i^1} \sum_{j=1}^{s_i^1} z_{i,j}$$

and

$$z_{s_i^2} = \frac{1}{s_i^2} \sum_{j=1}^{s_i^2} z_{i,j}.$$

We note that the particles $(z_{i,j})_{i=1,\dots,s_i^1}$ of $z_{s_i^1}$ are also used to compute $z_{s_i^2}$. Since $s_i^1 < s_i^2$, the code output $z_{s_i^2}$ is more accurate and time-consuming than the code output $z_{s_i^1}$. Furthermore, since the two outputs have common Monte-Carlo particles, they are correlated:

$$\begin{aligned}
\mathrm{cov}\left(z_{s_i^1}, z_{s_i^2}\right) &= \mathrm{cov}\left(\frac{1}{s_i^1}\sum_{j=1}^{s_i^1} z_{i,j}, \frac{1}{s_i^2}\sum_{j=1}^{s_i^2} z_{i,j}\right) \\
&= \frac{1}{s_i^1 s_i^2}\mathrm{cov}\left(\sum_{j=1}^{s_i^1} z_{i,j}, \sum_{j=1}^{s_i^1} z_{i,j} + \sum_{j=s_i^1+1}^{s_i^2} z_{i,j}\right) \\
&= \frac{\sigma_\varepsilon^2(x_i)}{s_i^2}.
\end{aligned}$$

We note that in practice the output $z_{s_i^2}(x_i)$ corresponds to the one of $z_{s_i^1}(x_i)$ for which we continue the Monte-Carlo convergence. This is relevant for practical applications since for obtaining accurate simulations it is less time consuming to start from former simulations partially converged.

Stochastic simulators and co-kriging models

Now let us consider that we want to surrogate $f(x)$ from the observations $\mathbf{z}_{s^1}^{n_1} = (z_{s_i^1})_{i=1,\dots,n_1}$ and $\mathbf{z}_{s^2}^{n_2} = (z_{s_i^2})_{i=1,\dots,n_2}$ such that $n_1 > n_2$ and $s_i^2 > s_i^1$ for all $i = 1, \dots, n_2$. We denote by $\mathbf{D} = \{x_1, \dots, x_{n_1}\}$ the experimental design set corresponding to the observations $\mathbf{z}_{s^1}^{n_1}$ and $\tilde{\mathbf{D}} = \{x_1, \dots, x_{n_2}\}$ the one corresponding to the observations $\mathbf{z}_{s^2}^{n_2}$. We note that we have the

nested property $\mathbf{D} = \tilde{\mathbf{D}} \cup \{x_{n_2+1}, \dots, x_{n_1}\}$. Furthermore, we suppose that $f(x)$ is a realization of a Gaussian process $Z(x)$ with mean $\mathbf{f}'(x)\boldsymbol{\beta}$ and covariance kernel $\sigma^2 r(x, \tilde{x})$. Therefore, the observations $\left(z_{s_i^j}\right)_{\substack{j=1,2 \\ i=1,\dots,n_j}}$ are realizations of the following random variables:

$$Z_{s_i^j}(x_i) = Z(x_i) + \frac{1}{s_i^j} \sum_{k=1}^{s_i^j} \sigma_\varepsilon(x_i)\varepsilon_{k,i}, \quad j = 1, 2,\ i = 1, \dots, n_j,$$

where $\varepsilon_{k,i} \sim \mathcal{N}(0, 1)$ and $(\varepsilon_{k,i})_{k=1,\dots,s_i^j}$ are independent. To predict $f(x)$ at a new location, we consider the following joint distribution where $\mathbf{Z}_{s^j}^{n_j}$, $j = 1, 2$ is the random vector $\left(Z_{s_i^j}(x_i)\right)_{i=1,\dots,n_j}$:

$$\begin{pmatrix} Z(x) \\ \mathbf{Z}_{s^1}^{n_1} \\ \mathbf{Z}_{s^2}^{n_2} \end{pmatrix}) \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{f}'(x) \\ \mathbf{f}'(\mathbf{D}) \\ \mathbf{f}'(\tilde{\mathbf{D}}) \end{pmatrix} \boldsymbol{\beta}, \sigma^2 \begin{pmatrix} 1 & \mathbf{r}'(x) & \tilde{\mathbf{r}}'(x) \\ \mathbf{r}(x) & \mathbf{K} & \mathbf{U} \\ \tilde{\mathbf{r}}(x) & \mathbf{U}' & \tilde{\mathbf{K}} \end{pmatrix} \right),$$

with $\mathbf{K} = [r(x_i, x_j) + (\sigma_\varepsilon^2(x_i)/s_i^1)\delta_{x_i=x_j}]_{i,j=1,\dots,n_1}$, $\tilde{\mathbf{K}} = [r(x_i, x_j) + (\sigma_\varepsilon^2(x_i)/s_i^2)\delta_{x_i=x_j}]_{i,j=1,\dots,n_2}$, $\mathbf{U} = [r(x_i, x_j) + (\sigma_\varepsilon^2(x_i)/s_i^2)\delta_{x_i=x_j}]_{\substack{i=1,\dots,n_1 \\ j=1,\dots,n_2}}$, $\mathbf{k}'(x) = [r(x, x_i)]_{i=1,\dots,n_1}$ and $\tilde{\mathbf{k}}'(x) = [r(x, x_i)]_{i=1,\dots,n_2}$. The surrogate model for $f(x)$ is given by the conditional distribution $[Z(x)|\mathbf{Z}_{s^1}^{n_1} = \mathbf{z}_{s^1}^{n_1}, \mathbf{Z}_{s^2}^{n_2} = \mathbf{z}_{s^2}^{n_2}, \sigma^2]$. Let us consider the proposition below.

Proposition 6.3. *Let us denote by* $\mathbf{Z}_{s^1}^{n_1-n_2} = \left(Z_{s_i^1}(x_i)\right)_{i=n_2+1,\dots,n_1}$ *and* $\mathbf{z}_{s^1}^{n_1-n_2} = \left(z_{s_i^1}\right)_{i=n_2+1,\dots,n_1}$. *Then* $[Z(x)|\mathbf{Z}_{s^1}^{n_1} = \mathbf{z}_{s^1}^{n_1}, \mathbf{Z}_{s^2}^{n_2} = \mathbf{z}_{s^2}^{n_2}, \sigma^2]$ *and* $[Z(x)|\mathbf{Z}_{s^1}^{n_1-n_2} = \mathbf{z}_{s^1}^{n_1-n_2}, \mathbf{Z}_{s^2}^{n_2} = \mathbf{z}_{s^2}^{n_2}, \sigma^2]$ *has the same distribution.*

Proposition 6.3 is of interest since it shows that using a co-kriging model with the observations $\mathbf{z}_{s^1}^{n_1}$ and $\mathbf{z}_{s^2}^{n_2}$ is equivalent to use a kriging model considering only the most accurate observations at points in $\mathbf{D}$.

Conclusion

We show in this introduction that in a framework of Monte-Carlo simulators - or experiments with repetitions - using a multi-fidelity co-kriging model is equivalent to use a noisy-kriging model with heteroscedastic observation noise variance. We note that the equivalence stands if we consider that fine code outputs correspond to coarse ones after continuing the Monte-Carlo convergence or repeating the experiments. Since we will always consider this case in the remainder of Part III, we will only use noisy-kriging models throughout it.

Chapter 7

Asymptotic analysis of the learning curve

7.1 Introduction

The purpose of this chapter is to describe the asymptotic behavior of the generalization error - defined as the averaged mean squared error - when the number of observations is large. As seen in the previous introduction, in many cases the noise variance is inversely proportional to the number of repetitions, and thus proportional to the number of observations, see Example 7.1 below. We consider this framework in this chapter.

Many authors were interested in obtaining learning curves describing the generalization error as a function of the training set size [Rasmussen and Williams, 2006]. The problem has been addressed in the statistical and numerical analysis areas. For an overview, the reader is referred to [Ritter, 2000b] for a numerical analysis point of view and to [Rasmussen and Williams, 2006] for a statistical one. In particular, in the numerical analysis literature, the authors are interested in numerical differentiation of functions from noisy data (see [Ritter, 2000a] and [Bozzini and Rossini, 2003]). They have found very interesting results for kernels satisfying the Sacks-Ylvisaker conditions of order r [Sacks and Ylvisaker, 1981] but only valid for 1-D or 2-D functions.

In the statistical literature [Sollich and Halees, 2002] give accurate approximations to the learning curve and [Opper and Vivarelli, 1999] and [Williams and Vivarelli, 2000] give upper and lower bounds on it. Their approximations give the asymptotic value of the learning curve (for a very large number of observations). They are based on the Woodbury-Sherman-Morrison matrix inversion lemma [Harville, 1997] which holds in finite-dimensional cases which correspond to degenerate covariance kernels in our context. Nonetheless, classical kernels used in Gaussian process regression are non-degenerate and we hence are in an infinite-dimensional case and the Woodbury-Sherman-Morrison formula cannot be used directly. Another proof for degenerate kernels can be found in [Picheny, 2009].

The main result of this chapter is a theorem giving the value of the Gaussian process regression mean squared error for a large training set size when the observation noise variance is proportional to the number of observations. This value is given as a function of the eigenvalues and eigenfunctions of the covariance kernel. From this theorem, we can deduce an approx-

imation of the learning curve for non-degenerate and degenerate kernels (which generalizes results in [Opper and Vivarelli, 1999], [Sollich and Halees, 2002] and [Picheny, 2009]) and for any dimension (which generalizes results in [Ritter, 2000b], [Ritter, 2000a] and [Bozzini and Rossini, 2003]). Finally, from this approximation we can deduce the rate of convergence of the Best Linear Unbiased Predictor (BLUP) in a Gaussian process regression framework.

The rate of convergence of the BLUP is of practical interest since it provides a powerful tool for decision support. Indeed, from an initial experimental design set, it can predict the additional computational budget necessary to reach a given desired accuracy when the observation noise variance is homogeneous in space.

The chapter is organized as follows. First we present the considered Gaussian process regression model with noisy observations. Second, we present the main result of the chapter which is the theorem giving the mean squared error of the considered model for a large training size. Third, we study the rate of convergence of the generalization error when the noise variance decreases. Academic examples are presented to compare the theoretical convergences given by the theorem and numerically observed convergences. Finally, an industrial application to the safety assessment of a nuclear system containing fissile materials is considered. This real case emphasizes the effectiveness of the theoretical rate of convergence of the BLUP since it predicts a very good approximation of the budget needed to reach a prescribed precision.

7.2 Gaussian process regression

Let us suppose that we want to approximate an objective function $x \in Q \subseteq \mathbb{R}^d \to f(x) \in \mathbb{R}$, Q a nonempty open set, from noisy observations of it at points $(x_i)_{i=1,\dots,n}$ with $x_i \in Q$. The points of the experimental design set $(x_i)_{i=1,\dots,n}$ are supposed to be sampled independently from the probability measure μ over Q. μ is called the design measure. We hence have n observations of the form $z_i = f(x_i) + \sqrt{\tau(x_i)}\varepsilon_i$ and we consider that $(\varepsilon_i)_{i=1,\dots,n}$ are independently sampled from the Gaussian distribution with mean zero and variance n:

$$\varepsilon \sim \mathcal{N}(0, n). \tag{7.1}$$

Note that the number of observations and the observation noise variance are both controlled by n. It means that if we increase the number n of observations, we automatically increase the uncertainty on the observations. An observation noise variance proportional to n is natural in the framework of experiments with repetitions or stochastic simulators. Indeed, for a fixed number of experiments (or simulations), the user can decide to perform them in few points with many repetitions (in that case the noise variance will be low) or to perform them in many points with few repetitions (in that case the noise variance will be large). We introduce in Example 7.1 the framework of repeated experiments. We note that the framework is the same as the one of stochastic simulators and it is the one considered in sections 7.5 and 7.6.

Example 7.1 (Gaussian process regression with repeated experiments)**.** Let us consider that we want to approximate the function $x \in Q \subseteq \mathbb{R}^d \to f(x) \in \mathbb{R}$ from noisy observations at points $(x_i)_{i=1,\dots,n}$ sampled from the design measure μ and with s replications at each point.

We hence have ns data of the form $z_{i,j} = f(x_i) + \sigma_\varepsilon(x_i)\varepsilon_{i,j}$ and we consider that $(\varepsilon_{i,j})_{\substack{i=1,\dots,n\\ j=1,\dots,s}}$ are independently distributed from a Gaussian distribution with mean zero and variance one. Then, denoting the vector of observed values by $\mathbf{z}^n = (z_i^n)_{i=1,\dots,n} = (\sum_{j=1}^s z_{i,j}/s)_{i=1,\dots,n}$, the variance of an observation z_i^n is $\sigma_\varepsilon^2(x_i)/s$. Thus, if we consider a fixed budget $T = ns$, we have $\sigma_\varepsilon^2(x_i)/s = n\tau(x_i)$ with $\tau(x_i) = \sigma_\varepsilon^2(x_i)/T$ and the observation noise variance is proportional to n.

In Section 7.3 we give the value of the generalization error for n large. Then, in Section 7.4 we are interested in its convergence for n large and when $\tau(x)$ tends to zero. Finally, in Section 7.5 we consider the non-uniform allocation $(s_i)_{i=1,\dots,n}$ with $T = \sum_{i=1}^n s_i$ and we address the question of optimal allocation of the repetitions $(s_i)_{i=1,\dots,n}$ as a function of the noise level $\sigma_\varepsilon^2(x_i)$ so as to minimize the generalization error.

The main idea of the Gaussian process regression is to suppose that the objective function $f(x)$ is a realization of a Gaussian process $Z(x)$ with a known mean and a known covariance kernel $k(x, \tilde{x})$ (note that we are here in a simple kriging case). The mean can be considered equal to zero without loss of generality. Then, denoting by $\mathbf{z}^n = [f(x_i) + \sqrt{\tau(x_i)}\varepsilon_i]_{1\le i\le n}$ the vector of length n containing the noisy observations, we choose as predictor the Best Linear Unbiased Predictor (BLUP) given by the equation (see Subsection 1.5.1 Equation (1.60)):

$$\hat{f}(x) = \mathbf{k}'(x)(\mathbf{K} + n\mathbf{\Delta})^{-1}\mathbf{z}^n, \qquad \mathbf{\Delta} = \mathrm{diag}[(\tau(x_i))_{i=1,\dots,n}], \tag{7.2}$$

where $\mathbf{k}(x) = [k(x, x_i)]_{1\le i\le n}$ is the n-vector containing the covariances between $Z(x)$ and $Z(x_i)$, $\quad 1 \le i \le n$ and $\mathbf{K} = [k(x_i, x_j)]_{1\le i,j\le n}$ is the $n \times n$-matrix containing the covariances between $Z(x_i)$ and $Z(x_j)$, $\quad 1 \le i, j \le n$. When $\tau(x)$ is independent of x, we have $\mathbf{\Delta} = \tau\mathbf{I}$ with $\mathbf{I}$ the $n\times n$ identity matrix. The BLUP minimizes the Mean Squared Error (MSE) which equals (see Subsection 1.5.1 Equation (1.61)):

$$\sigma^2(x) = k(x, x) - \mathbf{k}'(x)(\mathbf{K} + n\mathbf{\Delta})^{-1}\mathbf{k}(x). \tag{7.3}$$

Indeed, if we consider a Linear Unbiased Predictor (LUP) of the form $\mathbf{a}'(x)\mathbf{z}^n$, its MSE is given by:

$$\mathbb{E}[(Z(x) - \mathbf{a}'(x)\mathbf{Z}^n)^2] = k(x, x) - 2\mathbf{a}'(x)\mathbf{k}(x) + \mathbf{a}'(x)(\mathbf{K} + n\mathbf{\Delta})\mathbf{a}(x), \tag{7.4}$$

where $\mathbf{Z}^n = [Z(x_i) + \sqrt{\tau(x_i)}\varepsilon_i]_{1\le i\le n}$ and $\mathbb{E}$ stands for the expectation with respect to the distribution of the Gaussian process $Z(x)$. The value of $\mathbf{a}(x)$ minimizing (7.4) is $\mathbf{a}'_{\mathrm{opt}}(x) = \mathbf{k}'(x)(\mathbf{K}+n\mathbf{\Delta})^{-1}$. Therefore, the BLUP given by $\mathbf{a}'_{\mathrm{opt}}(x)\mathbf{z}^n$ is equal to (7.2) and by substituting $\mathbf{a}(x)$ with $\mathbf{a}_{\mathrm{opt}}(x)$ in Equation (7.4) we obtain the MSE of the BLUP given by Equation (7.3).

The main result of this chapter is the proof of a theorem that gives the asymptotic value of $\sigma^2(x)$ when $n \to +\infty$ and $\mathbf{\Delta} = \tau\mathbf{I}$. Thanks to this theorem, we can deduce the asymptotic value of the Integrating Mean Squared Error (IMSE) - also called learning curve or generalization error - when $n \to +\infty$. The IMSE is defined by:

$$\mathrm{IMSE} = \int_{\mathbb{R}^d} \sigma^2(x)\, d\mu(x), \tag{7.5}$$

where μ is the design measure of the input space parameters. The asymptotic value of the IMSE that we obtain can be viewed as a generalization of previous results (see [Rasmussen and Williams, 2006], [Ritter, 2000b], [Ritter, 2000a], [Bozzini and Rossini, 2003], [Opper and Vivarelli, 1999], [Sollich and Halees, 2002] and [Picheny, 2009]). It can be used to determine the budget required to reach a prescribed accuracy (see Section 7.5). Note that the proof of the theorem holds for a constant observation noise variance τ. Nevertheless, to provide optimal resource allocation, it can be important to take into account the heterogeneity of the observation noise variance. We give in Proposition 7.3 under certain restricted conditions (i.e., when $\mathbf{K}$ is diagonal) the optimal allocation taking into account the noise heterogeneity. Moreover, we numerically observe in Appendix D that this allocation remains efficient in more general cases although it is not anymore optimal (it remains more efficient than the uniform one).

7.3 Convergence of the learning curve for Gaussian process regression

This section deals with the convergence of the BLUP when the number of observations is large and the reduced noise variance does not depend on x, i.e. $\tau(x) = \tau$ and $\mathbf{\Delta} = \tau\mathbf{I}$. The speed of convergence of the BLUP is evaluated through the generalization error - i.e. the IMSE - defined in (7.5). The main theorem of this chapter follows:

Theorem 7.1. *Let us consider $Z(x)$ a Gaussian process with zero mean and covariance kernel $k(x, \tilde{x}) \in \mathcal{C}^0(Q \times Q)$ and $(x_i)_{i=1,\dots,n}$ an experimental design set of n independent random points sampled with the probability measure μ on $Q \subseteq \mathbb{R}^d$. We assume that $\sup_{x\in\mathbb{R}^d} k(x, x) < \infty$. According to Mercer's theorem (see Subsection 1.4.4 Theorem 1.4), we have the following representation of $k(x, \tilde{x})$:*

$$k(x, \tilde{x}) = \sum_{p\geq 0} \lambda_p \phi_p(x)\phi_p(\tilde{x}),$$

where $(\phi_p(x))_p$ is an orthonormal basis of $L^2_\mu(Q)$ (denoting the set of square integrable functions) consisting of eigenfunctions of $(T_{\mu,k}f)(x) = \int_{\mathbb{R}^d} k(x, \tilde{x})f(\tilde{x})d\mu(\tilde{x})$ and λ_p is the nonnegative sequence of corresponding eigenvalues sorted in decreasing order. Then, for a non-degenerate kernel - i.e. when $\lambda_p > 0$, $\forall p > 0$ - we have the following convergence in probability for the MSE (7.3) of the BLUP:

$$\sigma^2(x) \overset{n\to\infty}{\longrightarrow} \sum_{p\geq 0} \frac{\tau\lambda_p}{\tau + \lambda_p}\phi_p(x)^2. \tag{7.6}$$

For degenerate kernels - i.e. when only a finite number of λ_p are not zero - the convergence is almost sure. We note that the convergences hold with respect to the distribution of the points $(x_i)_{i=1,\dots,n}$ of the experimental design set.

The sketch of the proof of Theorem 7.1 is given below. The full proof is given in Section 7.7.

Sketch of Proof. We first prove the theorem for degenerate kernels (see Section 7.7.1) which was already known in that case. Next we find a lower bound for $\sigma^2(x)$ for non-degenerate kernels. Let us consider the Karhunen-Loève decomposition of $Z(x) = \sum_{p\geq 0} Z_p\sqrt{\lambda_p}\phi_p(x)$ where $(Z_p)_p$ is a sequence of independent Gaussian random variables with mean zero and variance one. If we denote by $a_{\text{opt},i}(x)$, $i = 1,\dots,n$, the coefficients of the BLUP associated to $Z(x)$, the Gaussian process regression mean squared error can be written $\sigma^2(x) = \sum_{p\geq 0}\lambda_p\left(\phi_p(x) - \sum_{i=1}^n a_{\text{opt},i}(x)\phi_p(x_i)\right)^2 + n\tau\sum_{i=1}^n a_{\text{opt},i}(x)^2$. Then, for a fixed $\bar{p}$, the following inequality holds:

$$\sigma^2(x) \geq \sum_{p\leq\bar{p}}\lambda_p\left(\phi_p(x) - \sum_{i=1}^n a_{\text{opt},i}(x)\phi_p(x_i)\right)^2 + n\tau\sum_{i=1}^n a_{\text{opt},i}(x)^2 = \sigma^2_{LUP,\bar{p}}(x), \tag{7.7}$$

where, $\sigma^2_{LUP,\bar{p}}(x)$ is the MSE of the Linear Unbiased Predictor (LUP) of coefficients $a_{\text{opt},i}(x)$ associated to the Gaussian process $Z_{\bar{p}}(x) = \sum_{p\leq\bar{p}} Z_p\sqrt{\lambda_p}\phi_p(x)$. Let us consider $\sigma^2_{\bar{p}}(x)$ the MSE of the BLUP of $Z_{\bar{p}}(x)$, we have the following inequality:

$$\sigma^2_{LUP,\bar{p}}(x) \geq \sigma^2_{\bar{p}}(x). \tag{7.8}$$

Since $Z_{\bar{p}}(x)$ has a degenerate kernel, $\forall\bar{p} > 0$, the almost sure convergence (7.6) holds for $\sigma^2_{\bar{p}}(x)$. Then, considering inequalities (7.7), the convergence (7.6) for $\sigma^2_{\bar{p}}(x)$ and the limit $\bar{p}\to\infty$, we obtain:

$$\liminf_{n\to\infty}\sigma^2(x) \geq \sum_{p\geq 0}\frac{\tau\lambda_p}{\tau+\lambda_p}\phi_p(x)^2. \tag{7.9}$$

It remains to find an upper bound for $\sigma^2(x)$. Since $\sigma^2(x)$ is the MSE of the BLUP associated to $Z(x)$, if we consider any other LUP associated to $Z(x)$, then the corresponding MSE denoted by $\sigma^2_{LUP}(x)$ satisfies the following inequality:

$$\sigma^2(x) \leq \sigma^2_{LUP}(x).$$

The idea is to find a LUP so that its MSE is a tight upper bound of $\sigma^2(x)$. Let us consider the LUP:

$$\hat{f}_{LUP}(x) = \mathbf{k}'(x)\mathbf{A}\mathbf{z}^n, \tag{7.10}$$

with $\mathbf{A}$ the $n\times n$ matrix defined by $\mathbf{A} = \mathbf{L}^{-1} + \sum_{k=1}^q(-1)^k(\mathbf{L}^{-1}\mathbf{M})^k\mathbf{L}^{-1}$ with $\mathbf{L} = n\tau\mathbf{I} + \sum_{p<p^*}\lambda_p[\phi_p(x_i)\phi_p(x_j)]_{1\leq i,j\leq n}$, $\mathbf{M} = \sum_{p\geq p^*}\lambda_p[\phi_p(x_i)\phi_p(x_j)]_{1\leq i,j\leq n}$, q a finite integer and p^* such that $\lambda_{p^*} < \tau$. The choice of this LUP is motivated by the fact that the matrix $\mathbf{A}$ is an approximation of the inverse of the matrix $(n\tau\mathbf{I}+\mathbf{K}) = \mathbf{L}+\mathbf{M}$ that is tractable in the following calculations. Remember that the BLUP is $\hat{f}_{\text{BLUP}}(x) = \mathbf{k}'(x)(\mathbf{K}+n\tau\mathbf{I})^{-1}\mathbf{z}^n$. Then, the MSE of the LUP (7.10) is given by:

$$\sigma^2_{LUP}(x) = k(x,x) - \mathbf{k}'(x)\mathbf{L}^{-1}\mathbf{k}(x) - \sum_{i=1}^{2q+1}(-1)^i\mathbf{k}'(x)(\mathbf{L}^{-1}\mathbf{M})^i\mathbf{L}^{-1}\mathbf{k}(x).$$

Thanks to the Woodbury-Sherman-Morrison formula[1], the strong law of large numbers and the continuity of the inverse operator in the space of p-dimensional invertible matrices, we

[1] If $\mathbf{B}$ is a non-singular $p\times p$ matrix, C a non-singular $m\times m$ matrix and $\mathbf{A}$ a $m\times p$ matrix with $m,p<\infty$, then $(\mathbf{B}+\mathbf{A}\mathbf{C}^{-1}\mathbf{A})^{-1} = \mathbf{B}^{-1} - \mathbf{B}^{-1}\mathbf{A}(\mathbf{A}'\mathbf{B}^{-1}\mathbf{A}+\mathbf{C})^{-1}\mathbf{A}'\mathbf{B}^{-1}$.

have the following almost sure convergence:

$$\mathbf{k}'(x)\mathbf{L}^{-1}\mathbf{k}(x) \overset{n\to\infty}{\longrightarrow} \sum_{p<p^*} \frac{\lambda_p^2}{\lambda_p+\tau}\phi_p(x)^2 + \frac{1}{\tau}\sum_{p\geq p^*} \lambda_p^2\phi_p(x)^2.$$

We note that we can use the Woodbury-Sherman-Morrison formula and the strong law of large numbers since p^* is finite and independent of n. Then, using the Markov inequality and the equality $\sum_{p\geq 0}\lambda_p\phi_p(x)^2 = k(x,x) < \infty$, we have the following convergence in probability:

$$\mathbf{k}'(x)(\mathbf{L}^{-1}\mathbf{M})^i\mathbf{L}^{-1}\mathbf{k}(x) \overset{n\to\infty}{\longrightarrow} \left(\frac{1}{\tau}\right)^{i+1} \sum_{p\geq p^*} \lambda_p^{i+2}\phi_p(x)^2.$$

We highlight that we cannot use the strong law of large numbers here due to the infinite sum in p in the definition of $\mathbf{M}$. Finally, we obtain the following convergence in probability:

$$\limsup_{n\to\infty} \sigma^2(x) \leq \lim_{n\to\infty} \sigma^2_{LUP}(x) = \sum_{p\geq 0}\left(\lambda_p - \frac{\lambda_p^2}{\tau+\lambda_p}\right)\phi_p(x)^2 - \sum_{p\geq p^*}\lambda_p^2 \frac{\left(\frac{\lambda_p}{\tau}\right)^{2q+1}}{\tau+\lambda_p}\phi_p(x)^2.$$

By taking the limit $q\to\infty$ in the right hand side and using the inequality $\lambda_{p^*} < \tau$, we obtain the following upper bound for $\sigma^2(x)$:

$$\limsup_{n\to\infty} \sigma^2(x) \leq \sum_{p\geq 0} \frac{\tau\lambda_p}{\tau+\lambda_p}\phi_p(x)^2. \tag{7.11}$$

The result announced in Theorem 7.1 is deduced from the lower and upper bounds (7.9) and (7.11). □

Remark 1 For non-degenerate kernels such that $||\phi_p(x)||_{L^\infty} < \infty$ uniformly in p, the convergence is almost sure. Some kernels such as the one of the Brownian motion satisfy this property.

The following theorem gives the asymptotic value of the learning curve when n is large.

Theorem 7.2. *Let us consider $Z(x)$ a Gaussian process with known mean and covariance kernel $k(x,\tilde{x}) \in \mathcal{C}^0(Q\times Q)$ such that $\sup_{x\in\mathbb{R}^d} k(x,x) < \infty$ and $(x_i)_{i=1,\dots,n}$ an experimental design set of n independent random points sampled with the probability measure μ on $Q \subseteq \mathbb{R}^d$. Then, for a non-degenerate kernel, we have the following convergence in probability:*

$$\text{IMSE} \overset{n\to\infty}{\longrightarrow} \sum_{p\geq 0} \frac{\tau\lambda_p}{\tau+\lambda_p}. \tag{7.12}$$

For degenerate kernels, the convergence is almost sure.

Proof. From Theorem 7.1 and the orthonormal property of the basis $(\phi_p(x))_p$ in $L^2_\mu(Q)$, the proof of the theorem is straightforward by integration. We note that we can permute the integral and the limit thanks to the dominated convergence theorem since $\sigma^2(x) \leq k(x,x)$. □

The obtained limit is identical to the one established in [Opper and Vivarelli, 1999], [Rasmussen and Williams, 2006] and [Picheny, 2009]. The originality of the presented result is the proof giving the asymptotic value of the learning curve for a non-degenerate kernel. This result is of practical interest since the usual kernels for Gaussian process regression are non-degenerate and we will exhibit dramatic differences between the learning curves of degenerate and non-degenerate kernels. We note that intuitive arguments are given in [Opper and Vivarelli, 1999] and [Picheny, 2009] to justify the relevance of the result for non-degenerate kernels.

Proposition 7.1. *Let us denote* $\mathrm{IMSE}_\infty = \lim_{n\to\infty} \mathrm{IMSE}$. *The following inequality holds:*

$$\frac{1}{2} B_\tau^2 \leq \mathrm{IMSE}_\infty \leq B_\tau^2, \tag{7.13}$$

with

$$B_\tau^2 = \sum_{p \text{ s.t. } \lambda_p \leq \tau} \lambda_p + \tau \# \left\{ p \text{ s.t. } \lambda_p > \tau \right\}. \tag{7.14}$$

Proof. The proof is directly deduced from Theorem 7.2 and the following inequality:

$$\frac{1}{2} h_\tau(x) \leq \frac{x}{x+\tau} \leq h_\tau(x),$$

with:

$$h_\tau(x) = \begin{cases} x/\tau & x \leq \tau \\ 1 & x > \tau \end{cases}.$$

□

7.4 Examples of rates of convergence for the learning curve

Proposition 7.1 shows that the rate of convergence of the generalization error IMSE_∞ in function of τ is equivalent to the one of B_τ^2. In this section, we analyze the rate of convergence of IMSE_∞ (or equivalently B_τ^2) when τ is small. We note that the presented results can be interpreted as a rate of convergence in function of the number of observations since τ is the ratio between the noise variance $n\tau$ and the number of observations n.

In this section, we consider that the design measure μ is uniform on $[0,1]^d$.

Example 2 (Degenerate kernels) For degenerate kernels we have $\# \left\{ p \text{ s.t. } \lambda_p > 0 \right\} < \infty$. Thus, when $\tau \to 0$, we have:

$$\sum_{p \text{ s.t. } \lambda_p < \tau} \lambda_p = 0,$$

from which we deduce:

$$B_\tau^2 \propto \tau. \tag{7.15}$$

Therefore, the IMSE decreases as τ. We find here a classical result about Monte-Carlo convergence which gives that the variance decay is proportional to the observation noise variance ($n\tau$) divided by the number of observations n whatever the dimension. Nevertheless, for non-degenerate kernels, the number of non-zero eigenvalues is infinite and we are hence in an infinite-dimensional case (contrarily to the degenerate one). We see in the following examples that we do not conserve the usual Monte-Carlo convergence rate in this case which emphasizes the importance of Theorem 7.1 dealing with non-degenerate kernels.

Example 3 (The fractional Brownian motion) Let us consider the fractional Brownian kernel with Hurst parameter $H \in (0,1)$:

$$k(x,y) = x^{2H} + y^{2H} - |x-y|^{2H}. \tag{7.16}$$

The associated Gaussian process - called fractional Brownian motion - is Hölder continuous with exponent $H-\varepsilon$, $\forall \varepsilon > 0$. According to [Bronski, 2003], we have the following result:

Proposition 7.2. *The eigenvalues of the fractional Brownian motion with Hurst exponent $H \in (0,1)$ satisfy the behavior*

$$\lambda_p \quad = \quad \frac{\nu_H}{p^{2H+1}} + o\left(p^{-\frac{(2H+2)(4H+3)}{4H+5}+\delta}\right), \qquad p \gg 1,$$

where $\delta > 0$ is arbitrary, $\nu_H = \frac{\sin(\pi H)\Gamma(2H+1)}{\pi^{2H+1}}$, and Γ is the Euler Gamma function.

Therefore, when $\tau \ll 1$, we have:

$$\lambda_p < \tau \quad \text{if} \quad p > \left(\frac{\nu_H}{\tau}\right)^{\frac{1}{2H+1}}.$$

We hence have the following approximation for B_τ^2:

$$B_\tau^2 \approx \sum_{p > \left(\frac{\nu_H}{\tau}\right)^{\frac{1}{2H+1}}} \frac{\nu_H}{p^{2H+1}} + \tau \left(\frac{\nu_H}{\tau}\right)^{\frac{1}{2H+1}}.$$

Furthermore, we have:

$$\sum_{p > \left(\frac{\nu_H}{\tau}\right)^{\frac{1}{2H+1}}} \frac{\nu_H}{p^{2H+1}} \approx \int_{\left(\frac{\nu_H}{\tau}\right)^{\frac{1}{2H+1}}}^{+\infty} \frac{\nu_H}{x^{2H+1}}\, dx = \frac{\nu_H}{2H\left(\frac{\nu_H}{\tau}\right)^{1-\frac{1}{2H+1}}},$$

from which we deduce:

$$B_\tau^2 \approx C_H \tau^{1-\frac{1}{2H+1}}, \qquad \tau \ll 1, \tag{7.17}$$

where C_H is a constant independent of τ.

The rate of convergence for a fractional Brownian motion with Hurst parameter H is $\tau^{1-\frac{1}{2H+1}}$. We note that the case $H = 1/2$ corresponds to the classical Brownian motion. We observe that the larger the Hurst parameter is (i.e. the more regular the Gaussian process is), the faster the convergence is. Furthermore, for $H \to 1$ the convergence rate gets close to $\tau^{2/3}$. Therefore, even for the most regular fractional Brownian motion, we are still far from the classical Monte-Carlo convergence rate.

Example 4 (The 1-D Matérn covariance kernel) In this example we deal with the Matérn kernel with regularity parameter $\nu > 0$ in dimension 1:

$$k_{1D}(x, \tilde{x}; \nu, l) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}|x - \tilde{x}|}{l} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}|x - \tilde{x}|}{l} \right), \tag{7.18}$$

where K_ν is the modified Bessel function [Abramowitz and Stegun, 1965]. The associated Gaussian process is Hölder continuous with exponent $\nu - \varepsilon$, $\forall \varepsilon > 0$. The eigenvalues of this kernel satisfy the following asymptotic behavior [Nazarov and Nikitin, 2004]:

$$\lambda_p \approx \frac{1}{p^{2(\nu+1/2)}}, \qquad p \gg 1.$$

Following the guideline of the Example 3 we deduce the following asymptotic behavior for B_τ^2:

$$B_\tau^2 \approx C_\nu \tau^{1-\frac{1}{2(\nu+1/2)}}, \qquad \tau \ll 1, \tag{7.19}$$

where C_ν is a constant independent of τ.

This result is in agreement with the one of [Ritter, 2000a] who proved that for 1-dimensional kernels satisfying the Sacks-Ylvisaker of order r conditions (where r is an integer), the generalization error for the best linear estimator and experimental design set strategy decays as $\tau^{1-\frac{1}{2r+2}}$. Indeed, for such kernels, the eigenvalues satisfy the large-p behavior $\lambda_p \propto 1/p^{2r+2}$ [Rasmussen and Williams, 2006] and by following the guideline of the previous examples we find the same convergence rate. We note that the Matérn kernel with parameter $\nu = r + 1/2$ satisfies the Sacks-Ylvisaker of order r conditions. Furthermore, our result generalizes the one of [Ritter, 2000a] since it provides convergence rates for more general kernels and for any dimension (see below). Finally, our result shows that the random sampling gives the same decay rate as the optimal experimental design.

Example 5 (The d-D tensorised Matérn covariance kernel) We focus here on the d-dimensional tensorised Matérn kernel with isotropic regularity parameter $\nu > \frac{1}{2}$. According to [Pusev, 2011] the eigenvalues of this kernel satisfy the asymptotics:

$$\lambda_p \approx \phi(p), \qquad p \gg 1,$$

where the function ϕ is defined by:

$$\phi(p) = \frac{\log(1+p)^{2(d-1)(\nu+1/2)}}{p^{2(\nu+1/2)}}.$$

Its inverse ϕ^{-1} satisfies:

$$\phi^{-1}(\varepsilon) = \varepsilon^{-\frac{1}{2(\nu+1/2)}} \left(\log \left(\varepsilon^{-\frac{1}{2(\nu+1/2)}} \right) \right)^{d-1} (1 + o(1)), \qquad \varepsilon \ll 1.$$

We hence have the approximation:

$$B_\tau^2 \approx \frac{2(\nu+1/2) - 1}{\phi^{-1}(\tau)^{2(\nu+1/2)-1}} \log \left(1 + \phi^{-1}(\tau)\right)^{2(d-1)(\nu+1/2)} + \tau \phi^{-1}(\tau).$$

We can deduce the following rate of convergence for B_τ^2:

$$B_\tau^2 \approx C_{\nu,d}\tau^{1-\frac{1}{2(\nu+1/2)}}\log(1/\tau)^{d-1}, \qquad \tau \ll 1, \tag{7.20}$$

with $C_{\nu,d}$ a constant independent of τ.

Example 6 (The d-D Gaussian covariance kernel) According to [Todor, 2006] the asymptotic behavior of the eigenvalues for a Gaussian kernel is:

$$\lambda_p \leq c' \exp\left(-cp^{\frac{1}{d}}\right),$$

where c and c' are constants that depend on the correlation length and the diameter of the domain Q. Applying the procedure presented in the previous examples, it can be shown than the rate of convergence of the IMSE is bounded by:

$$C_d\tau\log(1/\tau)^d, \qquad \tau \ll 1, \tag{7.21}$$

with C_d a constant independent of τ.

We can see from the previous examples that for smooth kernels, the convergence rate is close to τ, i.e. the classical Monte-Carlo rate.

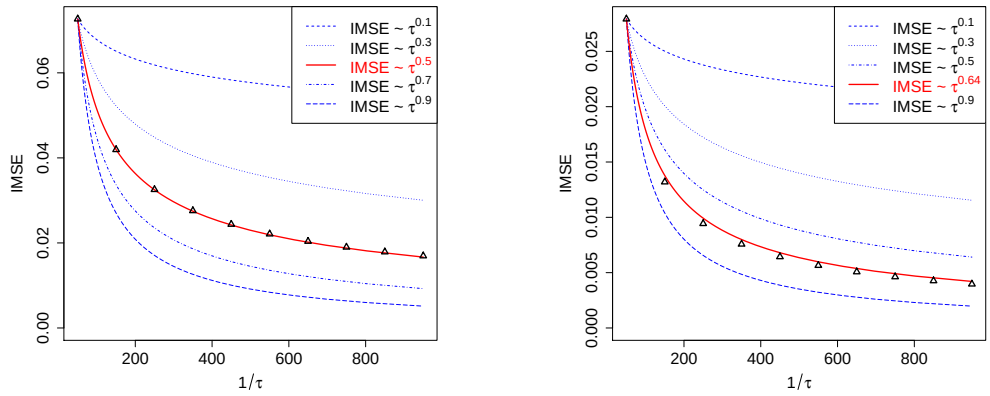
We compare the previous theoretical results on the rate of convergence of the generalization error with full numerical simulations. In order to observe the asymptotic convergence, we fix $n = 200$ and we consider $1/\tau$ varying from 50 to 1000. The experimental design sets are sampled from a uniform measure on $[0, 1]$ and the observation noise is $n\tau$. To estimate the IMSE (7.5) we use a trapezoidal numerical integration with 4000 quadrature points over $[0, 1]$. Furthermore, to build the convergence curves (i.e to estimate the multiplicative coefficients) in figures 7.1 and 7.2 we use a linear regression with the first value of the IMSE, an intercept fixed to zero (since the IMSE tends to 0 when τ tends to 0) and a unique explanatory variable corresponding to the tested convergence (e.g. $\tau^{0.1}$, $\tau\log(1/\tau)$,...).

First, we deal with the 1-D fractional Brownian kernel (7.16) with Hurst parameter H. We have proved that for large n, the IMSE decays as $\tau^{1-\frac{1}{2H+1}}$. Figure 7.1 compares the numerically estimated convergences to the theoretical ones.

We see in Figure 7.1 that the observed rate of convergence is perfectly fitted by the theoretical one. We note that we are far from the classical Monte-Carlo rate since we are in a non-degenerate case.

Finally, we deal with the 2-D tensorised Matérn-5/2 kernel and the 1-D Gaussian kernel. The 1-dimensional Matérn-ν class of covariance functions $k_{1D}(t, t'; \nu, \theta)$ is given by (7.18) and the 2-D tensorised Matérn-ν covariance function is given by:

$$k(x, \tilde{x}; \nu, \theta) = k_{1D}(x_1, x_1'; \nu, \theta_1)k_{1D}(x_2, x_2'; \nu, \theta_2). \tag{7.22}$$



(a) (b)

Figure 7.1: Rate of convergence of the IMSE when the level of observation noise decreases for a fractional Brownian motion with Hurst parameter $H = 0.5$ (a) and $H = 0.9$ (b). The number of observations is $n = 200$ and the observation noise variance is $n\tau$ with $1/\tau$ varying from 50 to 1000. The triangles represent the numerically estimated IMSE, the solid line represents the theoretical convergence, and the other non-solid lines represent various convergence rates.

Furthermore, the 1-D Gaussian kernel is defined by:

$$k(x, \tilde{x}; \theta) = \exp\left(-\frac{1}{2}\frac{(x-\tilde{x})^2}{\theta^2}\right).$$

Figure 7.2 compares the numerically observed convergence of the IMSE to the theoretical one when $\theta_1 = \theta_2 = 0.2$ for the Matérn-5/2 kernel and when $\theta = 0.2$ for the Gaussian kernel. We see in Figure 7.2 that the theoretical rate of convergence is a sharp approximation of the observed one.

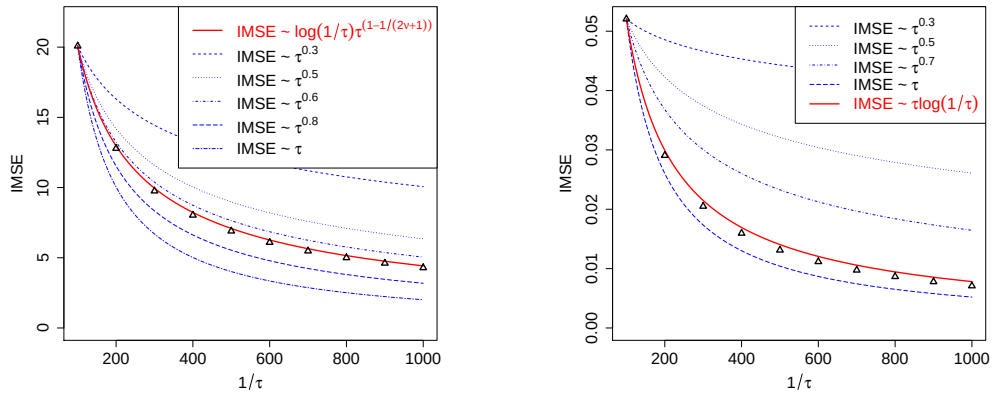
7.5 Applications of the learning curve

Let us consider that we want to approximate the function $x \in Q \subseteq \mathbb{R}^d \to f(x)$ from noisy observations at fixed points $(x_i)_{i=1,\dots,n}$, with $n \gg 1$, sampled from the design measure μ and with s_i replications at each point x_i.

In this section, we consider the situation described in Example 7.1:

- The budget T is defined as the sum of repetitions on all points of the experimental design set - i.e. $T = \sum_{i=1}^n s_i$.
- An observation z_i^n at point x_i has a noise variance equal to $\sigma_\varepsilon^2(x_i)/s_i$ with $i = 1, \dots, n$.

In Subsection 7.5.1 we present how to determine the needed budget T to achieve a prescribed precision. Then, in Subsection 7.5.2, we address the problem of the optimal allocation $\{s_1, s_2, \dots, s_n\}$ for a given budget T with Proposition 7.3.



(a) (b)

Figure 7.2: Rate of convergence of the IMSE when the level of observation noise decreases for a 2-D tensorised Matérn-5/2 kernel (a) and for a 1-D Gaussian kernel (b). The number of observations is $n = 200$ and the observation noise variance is $n\tau$ with $1/\tau$ varying from 100 to 1000. The triangles represent the numerically estimated IMSE, the solid line represents the theoretical convergence, and the other non-solid lines represent various convergences.

7.5.1 Estimation of the budget required to reach a prescribed precision

Let us consider a prescribed generalization error denoted by $\bar{\varepsilon}$. The purpose of this subsection is to determine from an initial budget T_0 the budget T for which the generalization error reaches the value $\bar{\varepsilon}$. We handle this issue by considering a uniform allocation $s_i = s$ with $i = 1, \ldots, n$ and a constant reduced noise variance σ_ε^2.

First, we build an initial experimental design set $(x_i^{\text{train}})_{i=1,\ldots,n}$ sampled with respect to the design measure μ and with s^* replications at each point such that $T_0 = ns^*$. From the s^* replications $(z_{i,j})_{j=1,\ldots,s^*}$, we can estimate the observation noise variances $\sigma_\varepsilon^2(x_i^{\text{train}})$ with a classical empirical estimator: $\sum_{j=1}^{s^*}(z_{i,j} - z_i^n)^2/(s^* - 1)$, $z_i^n = \sum_{j=1}^{s^*} z_{i,j}/s^*$. Then, we consider a constant reduced noise variance σ_ε^2 equal to the mean $\int_{\mathbb{R}^d} \sigma_\varepsilon^2(x)\, d\mu(x)$ estimated with $\sum_{i=1}^n \sigma_\varepsilon^2(x_i^{\text{train}})/n$.

Second, we use the observations $z_i^n = (\sum_{j=1}^{s^*} z_{i,j})/s^*$ to estimate the covariance kernel $k(x, \tilde{x})$. In practice, we consider a parametrized family of covariance kernels and we select the parameters which maximize the likelihood (see [Stein, 1999] and Chapter 1 Section 1.3).

Third, from Proposition 7.1 we can get the expression of the generalization error decay with respect to T (denoted by IMSE_T). Therefore, we just have to determine the budget T such that $\text{IMSE}_T = \bar{\varepsilon}$. In practice, we will not use Proposition 7.1 but the asymptotic results described in Section 7.4.

This strategy will be applied to an industrial case in Section 7.6. We note that in the application presented in Section 7.6, we have $s^* = 1$. In fact, in this example the observations are themselves obtained by an empirical mean of a Monte-Carlo sample and thus the noise variance can be estimated without processing replications.

7.5.2 Optimal resource allocation for a given budget

Let us consider a fixed budget T. As presented in Subsection 7.5.1, to determine this budget we make the approximation of a reduced noise variance $\sigma_\varepsilon^2(x)$ independent of x and we consider the uniform allocation $s_i = s$.

Despite the fact that the uniform allocation $s_i = s$ is needed to determine T, in order to provide the optimal resource allocation - i.e. the sequence of integers $\{s_1, s_2, \ldots, s_n\}$ minimizing the generalization error - it is worth taking into account the heterogeneity of the noise. For a Monte-Carlo based simulator, the number of repetitions s could represent the number of Monte-Carlo particles and the procedure presented below can be applied.

Determining the optimal allocation of the budget T whatever the Gaussian process for a heterogeneous noise is an open and non-trivial problem. To solve this problem, we first consider the continuum approximation in which we look for an optimal sequence of real numbers $(s_i)_{i=1,\ldots,n}$ and then we round the optimal solution to obtain a quasi-optimal integer-valued allocation $(s_{i,\text{int}})_{i=1,\ldots,n}$. The following proposition gives the optimal resource allocation under certain restricted conditions for the continuous case. The reader is referred to [Munoz Zuniga et al., 2011] for a proof of this proposition in a different framework (the proof uses the Karush-Kuhn-Tucker approach to solve the minimization problem with equality and inequality constraints [Kuhn and Tucker, 1951] and [Karush, 1939]). We note that the optimal allocation given in Proposition 7.3 for a fixed budget T can also be used for any $n > 0$ and for any experimental design set.

Proposition 7.3. *Let us consider $Z(x)$ a Gaussian process with a known mean and covariance kernel $k(x, x') \in \mathcal{C}^0(Q \times Q)$ with $\sup_x k(x, x) < \infty$. Let $(x_i)_{i=1,\ldots,n}$ be a given experimental design set of n points sorted such that the sequence $\left(\frac{k(x_j,x_j)+\sigma_\varepsilon^2(x_j)}{\sqrt{c(x_j)\sigma_\varepsilon^2(x_j)}}\right)_{j=1,\ldots,n}$ is non-increasing, where $\sigma_\varepsilon^2(x_i)$ is the reduced noise variance of an observation at point x_i, $c(x) = \int_{\mathbb{R}^d} k(x', x)^2 \, d\eta(x')$ and $\eta(x)$ is a positive measure used to calculate the Integrated Mean Squared Error (IMSE). When the covariance matrix $\mathbf{K}$ is diagonal, the real-valued allocation $(s_i)_{i=1,\ldots,n}$ minimizing the generalization error:*

$$\text{IMSE} = \int_{\mathbb{R}^d} \left(k(x, x) - \mathbf{k}'(x)(\mathbf{K} + \mathbf{\Delta})^{-1}\mathbf{k}(x)\right) \, d\eta(x), \tag{7.23}$$

under the constraints $\sum_{i=1}^n s_i = T$ and $s_i \geq 1$, $\forall i = 1, \ldots, n$ is given by:

$$s_i^{\text{opt}} = \begin{cases} 1 & i \leq i^* \\ \frac{1}{k(x_i,x_i)} \left(\frac{\sqrt{c(x_i)\sigma_\varepsilon^2(x_i)}}{\sum_{j=i^*+1}^n \frac{\sqrt{c(x_j)\sigma_\varepsilon^2(x_j)}}{k(x_j,x_j)}} \left(T - i^* + \sum_{j=i^*+1}^n \frac{\sigma_\varepsilon^2(x_j)}{k(x_j,x_j)}\right) - \sigma_\varepsilon^2(x_i) \right) & i > i^* \end{cases}, \tag{7.24}$$

where $\mathbf{\Delta} = \text{diag}\left[\left(\frac{\sigma_\varepsilon^2(x_i)}{s_i}\right)_{i=1,\ldots,n}\right]$ and:

$$i^* = \max\left\{ i = 1, \dots, n \quad \text{such that} \quad \frac{k(x_i, x_i) + \sigma_\varepsilon^2(x_i)}{\sqrt{c(x_i)\sigma_\varepsilon^2(x_i)}} \geq \frac{T - i + \sum_{j=i+1}^n \frac{\sigma_\varepsilon^2(x_j)}{k(x_j,x_j)}}{\sum_{j=i+1}^n \frac{\sqrt{c(x_j)\sigma_\varepsilon^2(x_j)}}{k(x_j,x_j)}} \right\}. \tag{7.25}$$

By convention, if:

$$\frac{k(x_i, x_i) + \sigma_\varepsilon^2(x_i)}{\sqrt{c(x_i)\sigma_\varepsilon^2(x_i)}} < \frac{T - i + \sum_{j=i+1}^n \frac{\sigma_\varepsilon^2(x_j)}{k(x_j,x_j)}}{\sum_{j=i+1}^n \frac{\sqrt{c(x_j)\sigma_\varepsilon^2(x_j)}}{k(x_j,x_j)}}, \qquad \forall i = 1, \dots, n, \tag{7.26}$$

then $i^* = 0$.

The proof of Proposition 7.3 is given in Appendix D. We note that the proof holds because the problem is separable due to the diagonal property of the covariance matrix. The optimization problem in Proposition 7.3 admits a solution if and only if $T \geq n$ which reflects the fact that n simulations are already available. Furthermore, when T is large enough, we have $i^* = 0$ and the solution has the following form:

$$s_i^{\text{opt}} = \frac{1}{k(x_i, x_i)} \left(\frac{\sqrt{c(x_i)\sigma_\varepsilon^2(x_i)}}{\sum_{j=1}^n \frac{\sqrt{c(x_j)\sigma_\varepsilon^2(x_j)}}{k(x_j,x_j)}} \left(T + \sum_{j=1}^n \frac{\sigma_\varepsilon^2(x_j)}{k(x_j, x_j)} \right) - \sigma_\varepsilon^2(x_i) \right). \tag{7.27}$$

While Proposition 7.3 gives a continuous optimal allocation, an admissible allocation must be an integer-valued sequence. Therefore we solve the optimization problem with the continuous approximation and then we round the continuous solution to obtain a quasi-optimal integer-valued solution $s_{i,\text{int}}^{\text{opt}}$. The rounding is performed by solving the following problem:

Find J such that $\sum_{i=1}^n s_{i,\text{int}}^{\text{opt}} = T$ with:

$$s_{i,\text{int}}^{\text{opt}} = \begin{cases} [s_i^{\text{opt}}] + 1 & i \leq J \\ [s_i^{\text{opt}}] & i > J \end{cases},$$

where $[x]$ denotes the integer part of a real number x.

We note that this allocation is not optimal in general (i.e. when $\mathbf{K}$ is not diagonal). Nevertheless we have numerically observed that it remains efficient in general cases and is better than the uniform allocation strategy. We perform numerical comparisons in Appendix D.

Proposition 7.3 shows that it is worth allocating more resources at locations where the reduced noise variance $\sigma_\varepsilon^2(x)$ and the quantity $c(x_i) = \int_{\mathbb{R}^d} k(x, x_i)^2 \, d\eta(x)$ (representing the local concentration of the IMSE) are more important.

7.6 Industrial Case: code MORET

We illustrate in this section an industrial application of our results about the rate of convergence of the IMSE. The case is about the safety assessment of a nuclear system containing

fissile materials. The system is modeled by a neutron transport code called MORET [Fernex et al., 2005]. In particular, we study a benchmark system of dry PuO_2 storage. We note that we are in the framework presented in Example 7.1.

This section is divided into 3 parts. First, we present the Gaussian process regression model built on an initial experimental design set. Then we apply the strategy described in Section 7.5.1 to determine the computational budget T needed to achieve a prescribed precision. Finally, we allocate the resource T on the experimental design set.

7.6.1 Data presentation

The benchmark system safety is evaluated through the neutron multiplication factor k_{eff}. This is our output of interest that we want to surrogate. This factor models the criticality of a chain nuclear reaction:

- $k_{\text{eff}} > 1$ leads to an uncontrolled chain reaction due to an increasing neutron population.
- $k_{\text{eff}} = 1$ leads to a self-sustained chain reaction with a stable neutron population.
- $k_{\text{eff}} < 1$ leads to a faded chain reaction due to an decreasing neutron population.

The neutron multiplication factor depends on many parameters and it is evaluated using the stochastic simulator called MORET. We focus here on two parameters:

- $d_{\text{PuO}_2} \in [0.5, 4]\text{g.cm}^{-3}$, the density of the fissile powder. It is scaled in this section to $[0, 1]$.
- $d_{\text{water}} \in [0, 1]\text{g.cm}^{-3}$, the density of water between storage tubes.

The other parameters are fixed to a nominal value given by an expert and we use the notation $x = (d_{\text{PuO}_2}, d_{\text{water}})$ for the input parameters.

The MORET code provides outputs of the following form:

$$k_{\text{eff},s}(x) = \frac{1}{s}\sum_{j=1}^{s} Y_j(x),$$

where $(Y_j(x))_{j=1,\dots,s}$ are realizations of independent and identically distributed random variables which are themselves obtained by an empirical mean of a Monte-Carlo sample of 4000 particles. From these particles, we can also estimate the variance $\sigma_\varepsilon^2(x)$ of the observation $Y_j(x)$ by a classical empirical estimator. The simulator gives noisy observations and the variance of an observation $k_{\text{eff},s}(x)$ equals $\sigma_\varepsilon^2(x)/s$.

A large data base $(Y_j(x_i))_{i=1,\dots,5625,j=1,\dots,200}$ is available to us. We divide it into a training set and a test set. Let us denote by $Y_j(x_i)$ the jth observation at point x_i - the 5625 points x_i of the data base come from a 75×75 grid over $[0, 1]^2$. The training set consists of $n = 100$ points $(x_i^{\text{train}})_{i=1,\dots,n}$ extracted from the complete data base using a maximin LHS and of the first observations $(Y_1(x_i^{\text{train}}))_{i=1,\dots,100}$. We will use the other 5525 points as a test set.

The aim of the study is - given the training set - to predict the budget needed to achieve a prescribed precision for the surrogate model and to allocate optimally these resources. More

precisely, let us denote by s_i the resource allocated to the point x_i^{train} of the experimental design set. First, we want to determine the budget $T = \sum_{i=1}^n s_i$ which allows us to achieve the target precision (see Subsection 7.5.1). Second, we want to determine the best resource allocation $(s_i)_{i=1,\dots,n}$ (see Subsection 7.5.2).

To evaluate the needed computational budget T the observation noise variance $\sigma_\varepsilon^2(x)$ is approximated by a constant $\bar{\sigma}_\varepsilon^2$. The constant variance equals the mean $\int_{\mathbb{R}^2} \sigma_\varepsilon^2(x)\, d\mu(x)$ of the noise variance which is here estimated by $\bar{\sigma}_\varepsilon^2 = \frac{1}{100}\sum_{i=1}^{100} \sigma_\varepsilon^2(x_i^{\text{train}}) = 3.3.10^{-3}$. Furthermore, we look for a uniform budget allocation, i.e. $s_i = s\ \forall i = 1, \dots, n$. In this case, the total computational budget is $T = ns$.

7.6.2 Model selection

To build the model, we consider the training set plotted in Figure 7.4. It is composed of the $n = 100$ points $(x_i^{\text{train}})_{i=1,\dots,n}$ which are uniformly spread on $Q = [0,1]^2$.

Let us suppose that the response is a realization of a Gaussian process with a tensorised Matérn-ν covariance function. The 2-D tensorised Matérn-ν covariance function $k(x, \tilde{x}; \nu, \boldsymbol{\theta})$ is given in (7.22). The hyper-parameters are estimated by maximizing the concentrated Likelihood:

$$-\frac{1}{2}(\mathbf{z}^n - m)'(\sigma^2\mathbf{K} + \sigma_\varepsilon^2\mathbf{I})^{-1}(\mathbf{z}^n - m) - \frac{1}{2}\det(\sigma^2\mathbf{K} + \bar{\sigma}_\varepsilon^2\mathbf{I}),$$

where $\mathbf{K} = [k(x_i^{\text{train}}, x_j^{\text{train}}; \nu, \boldsymbol{\theta})]_{i,j=1,\dots,n}$, $\mathbf{I}$ is the identity matrix, σ^2 the variance parameter, m the mean of $k_{\text{eff},s}(x)$ and $\mathbf{z}^n = (Y_1(x_1^{\text{train}}), \dots, Y_1(x_n^{\text{train}}))$ the observations at points in the training set. The mean of $k_{\text{eff},s}(x)$ is estimated by $m = \frac{1}{100}\sum_{i=1}^{100} Y_1(x_i^{\text{train}}) = 0.65$.

Due to the fact that the convergence rate is strongly dependent of the regularity parameter ν, we have to perform a good estimation of this hyper-parameter to evaluate the model error decay accurately. Note that we cannot have a closed form expression for the estimator of σ^2, it hence has to be estimated jointly with $\boldsymbol{\theta}$ and ν.

Let us consider the vector of parameters $\boldsymbol{\phi} = (\nu, \theta_1, \theta_2, \sigma^2)$. In order to perform the maximization, we have first randomly generated a set of 10,000 parameters $(\boldsymbol{\phi}_k)_{k=1,\dots,10^4}$ on the domain $[0.5, 3]\times[0.01, 2]\times[0.01, 2]\times[0.01, 1]$. We have then selected the 150 best parameters (i.e. the ones maximizing the concentrated Maximum Likelihood) and we have started a quasi-Newton based maximization from these parameters. More specifically, we have used the BFGS method [Shanno, 1970]. Finally, from the results of the 150 maximization procedures, we have selected the best parameter. We note that the quasi-Newton based maximizations have all converged to two parameter values, around 30% to the actual maximum and 70% to another local maximum.

The estimates of the hyper-parameters are $\nu = 1.31$, $\theta_1 = 0.67$, $\theta_2 = 0.45$ and $\sigma^2 = 0.24$. This means that we have a rough surrogate model which is not differentiable and α-Hölder continuous with exponent $\alpha = 0.81$. The variance of the observations is $\bar{\sigma}_\varepsilon^2 = 3.3.10^{-3}$, using the same notations as Example 7.1, we have $\tau = \bar{\sigma}_\varepsilon^2/T_0$ with $T_0 = n$ (it corresponds to $s = 1$).

The IMSE of the Gaussian process regression is $\text{IMSE}_{T_0} = 1.0.10^{-3}$ and its empirical mean squared error is $\text{EMSE}_{T_0} = 1.2.10^{-3}$. To compute the empirical mean squared error (EMSE), we use the observations $(Y_j(x_i))_{i=1,\dots,5525,\, j=1\dots,200}$ with $x_i \neq x_k^{\text{train}}\ \forall k = 1, \dots, 100, i =$

$1, \ldots, 5525$ and to compute the IMSE (7.5) (that depends only on the positions of the training set and on the selected hyper-parameters) we use a trapezoidal numerical integration into a 75×75 grid over $[0,1]^2$. For $s = 200$, the observation variance of the output $k_{\text{eff},s}(x)$ equals $\frac{\bar{\sigma}_\varepsilon^2}{200} = 1.64.10^{-5}$ and is neglected for the estimation of the empirical error. We can see that the IMSE is close to the empirical mean squared error which means that our model describes the observations accurately.

7.6.3 Convergence of the IMSE

According to (7.20), we have the following convergence rate for the IMSE:

$$\text{IMSE} \sim \log(1/\tau)\tau^{1-\frac{1}{2(\nu+1/2)}} = \frac{\log(T/\bar{\sigma}_\varepsilon^2)}{(T/\bar{\sigma}_\varepsilon^2)^{1-\frac{1}{2(\nu+1/2)}}}, \tag{7.28}$$

where the model parameter ν plays a crucial role. We can therefore expect that the IMSE decays as (see Subsection 7.5.1):

$$\text{IMSE}_T = \text{IMSE}_{T_0} \frac{\log(T/\bar{\sigma}_\varepsilon^2)}{(T/\bar{\sigma}_\varepsilon^2)^{1-\frac{1}{2(\nu+1/2)}}} / \frac{\log(T_0/\bar{\sigma}_\varepsilon^2)}{(T_0/\bar{\sigma}_\varepsilon^2)^{1-\frac{1}{2(\nu+1/2)}}}. \tag{7.29}$$

Let us assume that we want to reach an IMSE of $\bar{\varepsilon} = 2.10^{-4}$. According to the IMSE decay and the fact that the IMSE for the budget T_0 has been estimated to be equal to $1.0.10^{-3}$, the total budget required is $T = ns = 2000$, *i.e.* $s = 20$. Figure 7.3 compares the empirical mean squared error convergence and the predicted convergence (7.29) of the IMSE.

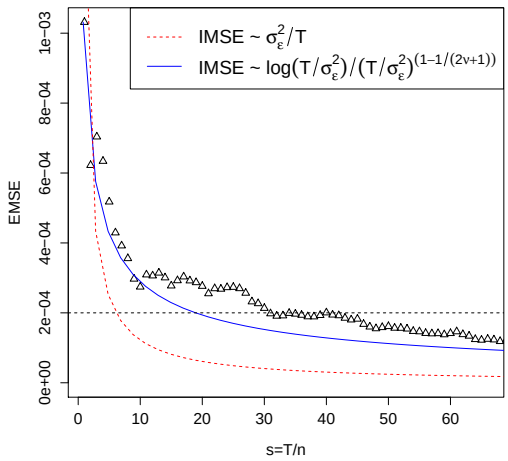


Figure 7.3: Comparison between Empirical mean squared error (EMSE) decay and theoretical IMSE decay for $n = 100$ when the total budget $T = ns$ increases. The triangles represent the Empirical MSE, the solid line represents the theoretical decay, the horizontal dashed line represents the desired accuracy and the dashed line the classical Monte-Carlo convergence. We see that Monte-Carlo decay does not match the empirical MSE and it is too fast.

We see empirically that the EMSE of $\bar{\varepsilon} = 2.10^{-4}$ is achieved for $s = 31$. This shows that

the predicted IMSE and the empirical MSE are close and that the selected kernel captures the regularity of the response accurately.

Let us consider the classical Monte-Carlo convergence rate $\bar{\sigma}_\varepsilon^2/T$, which corresponds to the convergence rate of degenerate kernels, *i.e.* in the finite -dimensional case. Figure 7.3 compares the theoretical rate of convergence of the IMSE with the classical Monte-Carlo one. We see that the Monte-Carlo decay is too fast and does not represent correctly the empirical MSE decay. If we had considered the rate of convergence IMSE $\sim \bar{\sigma}_\varepsilon^2/T$, we would have reached an IMSE of $\bar{\varepsilon} = 2.10^{-4}$ for $s = 6$ (which is far from the observed value $s = 31$).

7.6.4 Resource allocation

We have determined in the previous section the computational budget required to reach an IMSE of 2.10^{-4}. We observe that the predicted allocation is accurate since it gives an empirical MSE close to 2.10^{-4}. To calculate the observed MSE, we uniformly allocate the computational budget on the points of the training set. We know that this allocation is optimal when the variance of the observation noise is homogeneous. Nevertheless, we are not in this case and to build the final model we allocate the budget taking into account the heterogeneous noise level $\sigma_\varepsilon^2(x)$. We note that the total budget is $T = \sum_{i=1}^n s_i$ where $n = 100$ is the number observations and s_i the budget allocated to the point x_i^{train}.

From (7.27) in Proposition 7.3, when the input parameter distribution μ is uniform on $[0, 1]$ and for a diagonal covariance matrix, the optimal allocation is given by:

$$s_i = \frac{1}{\sigma^2}\left(\frac{\sqrt{\sigma_\varepsilon^2(x_i)}}{\sum_{j=1}^n \sqrt{\sigma_\varepsilon^2(x_j)}}\left(\sigma^2 T + \sum_{j=1}^n \sigma_\varepsilon^2(x_j)\right) - \sigma_\varepsilon^2(x_i)\right). \tag{7.30}$$

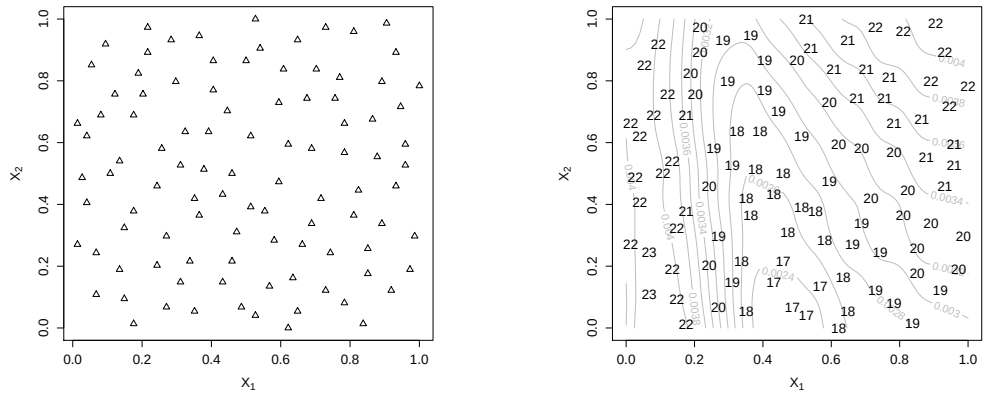
Here we use this allocation to build the model. Let us consider that we do not have observed the empirical MSE decay, we hence consider the budget given by the theoretical decay $T = 2400$. The allocation given by Equation (7.30) after the rounding procedure is illustrated in Figure 7.4 with the contour of the noise level.

We see in Figure 7.4 that the resources allocation is more important at points where the noise variance is higher. Table 7.1 compares the performances of the two models build with the two allocations on the test set.

	Uniform Allocation	Optimal Allocation
MSE	$2.71.10^{-4}$	$2.62.10^{-4}$
MaxSE	$5.66.10^{-2}$	$5.35.10^{-2}$

Table 7.1: Comparison between uniform and optimal (under the condition $\mathbf{K}$ diagonal) allocation of resources.

We see in Table 7.1 that the budget allocation given by Equation (7.30) gives predictions slightly more accurate than the uniform one.



(a) (b)

Figure 7.4: Figure (a): initial experimental design set with $n = 100$. Figure (b): noise level dependence of the resources allocation. The solid lines represent the reduced noise variance $\sigma^2_\varepsilon(x)$ contour plot and the numbers represent the resources $(s_i)_{i=1,\dots,n}$ allocated to the points of the experimental design set.

7.7 Proof of Theorem 7.1

7.7.1 Proof of Theorem 7.1: the degenerate case

The proof in the degenerate case follows the lines of the ones given by [Opper and Vivarelli, 1999], [Rasmussen and Williams, 2006] and [Picheny, 2009]. For a degenerate kernel, the number $\bar{p}$ of non-zero eigenvalues is finite. Let us denote $\mathbf{\Lambda} = \mathrm{diag}(\lambda_i)_{1\leq i\leq \bar{p}}$, $\phi(x) = (\phi_1(x), \dots, \phi_{\bar{p}}(x))$ and $\Phi = \begin{pmatrix}\phi(x_1)' & \dots & \phi(x_n)'\end{pmatrix}'$. The MSE of the Gaussian process regression is given by:

$$\sigma^2(x) = \phi(x)\mathbf{\Lambda}\phi(x)' - \phi(x)\mathbf{\Lambda}\Phi'\left(\Phi\mathbf{\Lambda}\Phi' + n\tau\mathbf{I}\right)^{-1}\Phi\mathbf{\Lambda}\phi(x)'.$$

Thanks to the Woodbury-Sherman-Morrison formula and according to [Opper and Vivarelli, 1999] and [Picheny, 2009] the Gaussian process regression error can be written:

$$\sigma^2(x) = \phi(x)\left(\frac{\Phi'\Phi}{n\tau} + \mathbf{\Lambda}^{-1}\right)^{-1}\phi(x)'.$$

Since $\bar{p}$ is finite, by the strong law of large numbers, the $\bar{p}\times\bar{p}$ matrix $\frac{1}{n}\Phi'\Phi$ converges almost surely as $n\to\infty$. We so have the following almost sure convergence:

$$\sigma^2(x) \overset{n\to\infty}{\longrightarrow} \sum_{p\leq\bar{p}} \frac{\tau\lambda_p}{\tau+\lambda_p}\phi_p(x)^2. \tag{7.31}$$

7.7.2 Proof of Theorem 7.1: the lower bound for $\sigma^2(x)$

The objective is to find a lower bound for $\sigma^2(x)$ for non-degenerate kernels. Let us consider the Karhunen-Loève decomposition of $Z(x) = \sum_{p\geq 0} Z_p\sqrt{\lambda_p}\phi_p(x)$ where $(Z_p)_p$ is a sequence

of independent Gaussian random variables with mean zero and variance 1. If we denote by $a_i(x)$ the coefficients of the BLUP associated to $Z(x)$, the mean squared error can be written

$$\begin{aligned}
\sigma^2(x) &= \mathbb{E}\left[\left(Z(x) - \sum_{i=1}^{n} a_i(x) Z(x_i)\right)^2\right] \\
&= \mathbb{E}\left[\left(\sum_{p\geq 0} \sqrt{\lambda_p}\left(\phi_p(x) - \sum_{i=1}^{n} a_i(x)\phi_p(x_i)\right) Z_p\right)^2\right] \\
&= \sum_{p\geq 0} \lambda_p \left(\phi_p(x) - \sum_{i=1}^{n} a_i(x)\phi_p(x_i)\right)^2 + n\tau \sum_{i=1}^{n} a_i(x)^2.
\end{aligned}$$

Then, for a fixed $\bar{p}$, the following inequality holds:

$$\sigma^2(x) \geq \sum_{p\leq \bar{p}} \lambda_p \left(\phi_p(x) - \sum_{i=1}^{n} a_i(x)\phi_p(x_i)\right)^2 + n\tau \sum_{i=1}^{n} a_i(x)^2 = \sigma^2_{LUP,\bar{p}}(x). \tag{7.32}$$

$\sigma^2_{LUP,\bar{p}}(x)$ is the MSE of the LUP of coefficients $a_i(x)$ associated to the Gaussian process $Z_{\bar{p}}(x) = \sum_{p\leq \bar{p}} Z_p \sqrt{\lambda_p}\phi_p(x)$. Let us consider $\sigma^2_{\bar{p}}(x)$ the MSE of the BLUP of $Z_{\bar{p}}(x)$, we have the following inequality:

$$\sigma^2_{LUP,\bar{p}}(x) \geq \sigma^2_{\bar{p}}(x). \tag{7.33}$$

Since $Z_{\bar{p}}(x)$ has a degenerate kernel, the almost sure convergence given in Equation (7.31) holds for $\sigma^2_{\bar{p}}(x)$. Then, considering inequalities (7.32) and (7.33) and the convergence (7.31), we obtain:

$$\liminf_{n\to\infty} \sigma^2(x) \geq \sum_{p\leq \bar{p}} \left(\frac{\tau\lambda_p}{\tau + \lambda_p}\right) \phi_p(x)^2. \tag{7.34}$$

Taking the limit $\bar{p} \to \infty$ in the right hand side gives the desired result.

7.7.3 Proof of Theorem 7.1: the upper bound for $\sigma^2(x)$

The objective is to find an upper bound for $\sigma^2(x)$. Since $\sigma^2(x)$ is the MSE of the BLUP associated to $Z(x)$, if we consider any other LUP associated to $Z(x)$ its MSE denoted by $\sigma^2_{LUP}(x)$ satisfies the following inequality:

$$\sigma^2(x) \leq \sigma^2_{LUP}(x). \tag{7.35}$$

The idea is to find a LUP so that its MSE is a tight upper bound of $\sigma^2(x)$. Let us consider the LUP:

$$\hat{f}_{LUP}(x) = \mathbf{k}'(x)\mathbf{A}\mathbf{z}^n, \tag{7.36}$$

with $\mathbf{A}$ the $n \times n$ matrix defined by $\mathbf{A} = \mathbf{L}^{-1} + \sum_{k=1}^{q}(-1)^k(\mathbf{L}^{-1}\mathbf{M})^k\mathbf{L}^{-1}$ with $\mathbf{L} = n\tau\mathbf{I} + \sum_{p\leq p^*} \lambda_p[\phi_p(x_i)\phi_p(x_j)]_{1\leq i,j\leq n}$, $\mathbf{M} = \sum_{p>p^*} \lambda_p[\phi_p(x_i)\phi_p(x_j)]_{1\leq i,j\leq n}$, q a finite integer and

p^* such that $\lambda_{p^*} < \tau$. The matrix $\mathbf{A}$ is an approximation of the inverse of the matrix $\mathbf{L}+\mathbf{M} = n\tau\mathbf{I}+\mathbf{K}$. Then, the MSE of the LUP (7.36) is given by:

$$\sigma^2_{LUP}(x) = k(x,x) - \mathbf{k}'(x)\left(2\mathbf{A} - \mathbf{A}(n\tau\mathbf{I}+\mathbf{K})\mathbf{A}\right)\mathbf{k}(x)$$

and by substituting the expression of $\mathbf{A}$ into the previous equation we obtain:

$$\sigma^2_{LUP}(x) = k(x,x) - \mathbf{k}'(x)\mathbf{L}^{-1}\mathbf{k}(x) - \sum_{i=1}^{2q+1}(-1)^i\mathbf{k}'(x)(\mathbf{L}^{-1}\mathbf{M})^i\mathbf{L}^{-1}\mathbf{k}(x). \tag{7.37}$$

First, let us consider the term $\mathbf{k}'(x)\mathbf{L}^{-1}\mathbf{k}(x)$. Since $p^* < \infty$, the matrix $\mathbf{L}$ can be written:

$$\mathbf{L} = n\tau\mathbf{I} + \Phi_{p^*}\mathbf{\Lambda}\Phi'_{p^*}, \tag{7.38}$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_i)_{1\le i\le p^*}$, $\Phi_{p^*} = \begin{pmatrix}\phi(x_1)' & \dots & \phi(x_n)'\end{pmatrix}'$ and $\phi(x) = (\phi_1(x),\dots,\phi_{p^*}(x))$.

Thanks to the Woodbury-Sherman-Morrison formula, the matrix $\mathbf{L}^{-1}$ is given by:

$$\mathbf{L}^{-1} = \frac{\mathbf{I}}{n\tau} - \frac{\Phi_{p^*}}{n\tau}\left(\frac{\Phi'_{p^*}\Phi_{p^*}}{n\tau} + \mathbf{\Lambda}^{-1}\right)^{-1}\frac{\Phi'_{p^*}}{n\tau}. \tag{7.39}$$

From the continuity of the inverse operator for invertible $p^*\times p^*$ matrices and by applying the strong law of large numbers, we obtain the following almost sure convergence :

$$\begin{aligned}\mathbf{k}'(x)\mathbf{L}^{-1}\mathbf{k}(x) &= \frac{1}{n\tau}\sum_{i=1}^n k(x,x_i)^2 - \frac{1}{\tau^2}\sum_{p,q=0}^{p^*}\left[\left(\frac{\Phi'_{p^*}\Phi_{p^*}}{n\tau}+\mathbf{\Lambda}^{-1}\right)^{-1}\right]_{p,q}\\ &\times \left[\frac{1}{n}\sum_{i=1}^n k(x,x_i)\phi_p(x_i)\right]\left[\frac{1}{n}\sum_{j=1}^n k(x,x_j)\phi_q(x_j)\right],\end{aligned}$$

$$\overset{n\to\infty}{\longrightarrow} \frac{1}{\tau}\mathbb{E}_\mu[k(x,X)^2] - \frac{1}{\tau^2}\sum_{p,q=0}^{p^*}\left[\left(\frac{\mathbf{I}}{\tau}+\mathbf{\Lambda}^{-1}\right)^{-1}\right]_{p,q}\mathbb{E}_\mu[k(x,X)\phi_p(X)]\mathbb{E}_\mu[k(x,X)\phi_q(X)],$$

where $\mathbb{E}_\mu$ is the expectation with respect to the design measure μ. We note that we can use the Woodbury-Sherman-Morrison formula and the strong law of large numbers since p^* is finite and independent of n. Then, the orthonormal property of the basis $(\phi_p(x))_{p\ge 0}$ implies:

$$\mathbb{E}_\mu[k(x,X)^2] = \sum_{p\ge 0}\lambda_p^2\phi_p(x)^2, \qquad \mathbb{E}_\mu[k(x,X)\phi_p(X)] = \lambda_p\phi_p(x).$$

Therefore, we have the following almost sure convergence:

$$\mathbf{k}'(x)\mathbf{L}^{-1}\mathbf{k}(x) \overset{n\to\infty}{\longrightarrow} \sum_{p\le p^*}\frac{\lambda_p^2}{\lambda_p+\tau}\phi_p(x)^2 + \frac{1}{\tau}\sum_{p>p^*}\lambda_p^2\phi_p(x)^2. \tag{7.40}$$

Second, let us consider the term $\sum_{i=1}^{2q+1}(-1)^i\mathbf{k}'(x)(\mathbf{L}^{-1}\mathbf{M})^i\mathbf{L}^{-1}\mathbf{k}(x)$. We have the following equality:

$$\begin{aligned}\mathbf{k}'(x)(\mathbf{L}^{-1}\mathbf{M})^i\mathbf{L}^{-1}\mathbf{k}(x) &= \sum_{l=0}^i\binom{i}{l}\frac{1}{n\tau}\mathbf{k}'(x)\left(\frac{\mathbf{M}}{n\tau}\right)^l\left(-\frac{\mathbf{L}'\mathbf{M}}{(n\tau)^2}\right)^{i-l}\mathbf{k}(x)\\ &\quad -\mathbf{k}'(x)\left(\frac{\mathbf{M}}{n\tau}\right)^l\left(-\frac{\mathbf{L}'\mathbf{M}}{(n\tau)^2}\right)^{i-l}\frac{\mathbf{L}'}{(n\tau)^2}\mathbf{k}(x),\end{aligned}$$

where:

$$\mathbf{L}' = \Phi_{p^*}\left(\frac{\Phi'_{p^*}\Phi_{p^*}}{n\tau} + \mathbf{\Lambda}^{-1}\right)^{-1}\Phi'_{p^*} = \sum_{p,p'\leq p^*} d^{(n)}_{p,p'}[\phi_p(x_i)\phi_p(x_j)]_{1\leq i,j\leq n}, \tag{7.41}$$

with $d^{(n)}_{p,p'} = \left[\left(\frac{\Phi'_{p^*}\Phi_{p^*}}{n\tau} + \mathbf{\Lambda}^{-1}\right)^{-1}\right]_{p,p'}$. Since $q < \infty$, we can obtain the convergence in probability of $\sum_{i=1}^{2q+1}(-1)^i\mathbf{k}'(x)(\mathbf{L}^{-1}\mathbf{M})^i\mathbf{L}^{-1}\mathbf{k}(x)$ from the ones of:

$$\mathbf{k}'(x)\frac{1}{n}\left(\frac{\mathbf{M}}{n}\right)^j\left(\frac{\mathbf{L}'\mathbf{M}}{n^2}\right)^{i-j}\mathbf{k}(x) \tag{7.42}$$

and:

$$\mathbf{k}'(x)\left(\frac{\mathbf{M}}{n}\right)^j\left(\frac{\mathbf{L}'\mathbf{M}}{n^2}\right)^{i-j}\frac{\mathbf{L}'}{n^2}\mathbf{k}(x), \tag{7.43}$$

with $i \leq 2q+1$ and $j \leq i$. Let us consider $\mathbf{k}'(x)\frac{1}{n}\left(\frac{\mathbf{M}}{n}\right)^j\left(\frac{\mathbf{L}'\mathbf{M}}{n^2}\right)^{i-j}\mathbf{k}(x)$ and $i > j$, we have:

$$\mathbf{k}'(x)\frac{1}{n}\left(\frac{\mathbf{M}}{n}\right)^j\left(\frac{\mathbf{L}'\mathbf{M}}{n^2}\right)^{i-j}\mathbf{k}(x) = \sum_{\substack{p_1,\ldots,p_{i-j}\leq p^*\\ p'_1,\ldots,p'_{i-j}\leq p^*}} d^{(n)}_{p_1,p'_1}\ldots d^{(n)}_{p_{i-j},p'_{i-j}} \sum_{\substack{q_1,\ldots,q_{i-j}>p^*\\ m_1,\ldots,m_j>p^*}} S^{(n)}_{q,m}, \tag{7.44}$$

with:

$$\begin{aligned} S^{(n)}_{q,m} &= \left(\frac{\sqrt{\lambda_{m_1}}}{n}\sum_{r=1}^n k(x,x_r)\phi_{m_1}(x_r)\right)\left(\frac{\sqrt{\lambda_{m_j}}}{n}\sum_{r=1}^n \phi_{m_j}(x_r)\phi_{p'_1}(x_r)\right)\\ &\times\left(\frac{\lambda_{q_{i-j}}}{n}\sum_{r=1}^n k(x,x_r)\phi_{q_{i-j}}(x_r)\sum_{r=1}^n \phi_{p_{i-j}}(x_r)\phi_{q_{i-j}}(x_r)\right)\\ &\times\prod_{l=1}^{j-1}\frac{\sqrt{\lambda_{m_l}\lambda_{m_{l+1}}}}{n}\sum_{r=1}^n \phi_{m_l}(x_r)\phi_{m_{l+1}}(x_r)\prod_{l=1}^{i-j-1}\frac{\lambda_{q_l}}{n}\sum_{r=1}^n \phi_{q_l}(x_r)\phi_{p_{l+1}}(x_r)\sum_{r=1}^n \phi_{q_l}(x_r)\phi_{p'_l}(x_r). \end{aligned}$$

We consider now the term:

$$a^{(n)}_{q,p,p'} = \frac{\lambda_q}{n}\sum_{r=1}^n \phi_q(x_r)\phi_p(x_r)\frac{1}{n}\sum_{r=1}^n \phi_{p'}(x_r)\phi_q(x_r), \tag{7.45}$$

with $p, p' \leq p^*$. From Cauchy Schwarz inequality and thanks to the following inequality:

$$|\phi_p(x)|^2 \leq \frac{1}{\lambda_p}\sum_{p'\geq 0}\lambda_{p'}|\phi_{p'}(x)|^2 = \lambda_p^{-1}k(x,x),$$

we obtain (using $\lambda_p \geq \lambda_{p^*}$, $\forall p \leq p^*$ and $[\sum_{r=1}^n |\phi_q(x_r)|]^2 \leq n\sum_{r=1}^n \phi_q(x_r)^2$):

$$\left|a^{(n)}_{q,p,p'}\right| \leq \sigma^2\lambda_{p^*}^{-1}\frac{\lambda_q}{n}\sum_{r=1}^n \phi_q(x_r)^2 \qquad \forall p, p' \leq p^*,$$

with $\sigma^2 = \sup_x k(x,x)$. Considering the expectation with respect to the distribution of points x_r, we obtain $\forall \bar{p} < \infty$:

$$\mathbb{E}_\mu \left[\sum_{q>\bar{p}} \left| a^{(n)}_{q,p,p'} \right| \right] \leq \sigma^2 \lambda_{p^*}^{-1} \sum_{q>\bar{p}} \lambda_q.$$

From Markov inequality, $\forall \delta > 0$, we have:

$$\mathbb{P}_\mu \left(\left| \sum_{q>\bar{p}} a^{(n)}_{q,p,p'} \right| > \delta \right) \leq \frac{\mathbb{E}_\mu \left[\left| \sum_{q>\bar{p}} a^{(n)}_{q,p,p'} \right| \right]}{\delta} \leq \frac{\sigma^2 \lambda_{p^*}^{-1} \sum_{q>\bar{p}} \lambda_q}{\delta}. \tag{7.46}$$

Furthermore, $\forall \delta > 0$, $\forall \bar{p} > p^*$:

$$\mathbb{P}_\mu \left(\left| \sum_{q>p^*} a^{(n)}_{q,p,p'} \right| > 2\delta \right) \leq \mathbb{P}_\mu \left(\left| \sum_{p^*<q\leq\bar{p}} a^{(n)}_{q,p,p'} \right| > \delta \right) + \mathbb{P}_\mu \left(\left| \sum_{q>\bar{p}} a^{(n)}_{q,p,p'} \right| > \delta \right).$$

We have for all $q \in (p^*, \bar{p}] : a^{(n)}_{q,p,p'} \to a_{q,p,p'} = \lambda_q \delta_{q=p} \delta_{q=p'} = 0$ (with δ the Kronecker product), as $n \to \infty$, therefore:

$$\limsup_{n\to\infty} \mathbb{P}_\mu \left(\left| \sum_{q>p*} a^{(n)}_{q,p,p'} \right| > 2\delta \right) \leq \frac{\sigma^2 \lambda_{p^*}^{-1} \sum_{q>\bar{p}} \lambda_q}{\delta}.$$

Taking the limit $\bar{p} \to \infty$ in the right hand side, we obtain the convergence in probability of $\sum_{q>p^*} a^{(n)}_{q,p,p'}$ when $n \to \infty$:

$$\sum_{q>p^*} \frac{\lambda_q}{n} \sum_{r=1}^{n} \phi_q(x_r)\phi_p(x_r) \frac{1}{n} \sum_{r=1}^{n} \phi_{p'}(x_r)\phi_q(x_r) \xrightarrow{\mathbb{P}_\mu} 0 \qquad \forall p, p' \leq p^*. \tag{7.47}$$

Following the same method, we obtain the convergence:

$$\sum_{q>p^*} \frac{\lambda_q}{n} \sum_{r=1}^{n} k(x,x_r)\phi_q(x_r) \sum_{r=1}^{n} \phi_p(x_r)\phi_q(x_r) \xrightarrow{\mathbb{P}_\mu} 0 \qquad \forall p \leq p^*. \tag{7.48}$$

Let us return to $S^{(n)}_{q,m}$. By using Cauchy Schwarz inequality and bounding by the constant K_M all the terms independent of q_i and m_i, we obtain:

$$\begin{aligned} \left| \sum_{q_1,\ldots,q_{i-j}>p^*} S^{(n)}_{q,m} \right| \leq\ & K_M \prod_{l=1}^{j} \lambda_{m_l} \frac{1}{n} \sum_{r=1}^{n} \phi_{m_l}(x_r)^2 \\ & \times \left| \sum_{q_{i-j}>p^*} \left(\frac{\lambda_{q_{i-j}}}{n} \sum_{r=1}^{n} k(x,x_r)\phi_{q_{i-j}}(x_r) \sum_{r=1}^{n} \phi_{p_{i-j}}(x_r)\phi_{q_{i-j}}(x_r) \right) \right| \\ & \times \left| \sum_{q_1,\ldots,q_{i-j-1}>p^*} \prod_{l=1}^{i-j-1} \frac{\lambda_{q_l}}{n} \sum_{r=1}^{n} \phi_{q_l}(x_r)\phi_{p_{l+1}}(x_r) \sum_{r=1}^{n} \phi_{q_l}(x_r)\phi_{p'_l}(x_r) \right|. \end{aligned}$$

Since $\sum_{p\geq 0}\lambda_p\phi_p(x)^2 = k(x,x) \leq \sigma^2$, we have the inequality $0 \leq \sum_{m_1,\ldots,m_j}\prod_{l=1}^{j}\lambda_{m_l}\frac{1}{n}\sum_{r=1}^{n}\phi_{m_l}(x_r)^2 \leq (\sigma^2)^j$. Thus, for $i > j$ and from (7.47) and (7.48) we obtain the following convergence in probability when $n \to \infty$:

$$\sum_{\substack{q_1,\ldots,q_{i-j}>p^* \\ m_1,\ldots,m_j>p^*}} S_{q,m}^{(n)} \xrightarrow{\mathbb{P}_\mu} 0.$$

Therefore, from (7.44) we obtain the following convergence when $n \to \infty$:

$$\mathbf{k}'(x)\frac{1}{n}\left(\frac{\mathbf{M}}{n}\right)^j\left(\frac{\mathbf{L}'\mathbf{M}}{n^2}\right)^{i-j}\mathbf{k}(x) \xrightarrow{\mathbb{P}_\mu} 0 \qquad \forall i < j. \tag{7.49}$$

Following the same guideline as previously, it can be shown that when $n \to \infty$:

$$\mathbf{k}'(x)\frac{1}{n}\left(\frac{\mathbf{M}}{n}\right)^j\left(\frac{\mathbf{L}'\mathbf{M}}{n^2}\right)^{i-j}\frac{\mathbf{L}'}{n^2}\mathbf{k}(x) \xrightarrow{\mathbb{P}_\mu} 0 \qquad \forall i \leq j. \tag{7.50}$$

From the convergences (7.49) and (7.50), we deduce the following one when $n \to \infty$:

$$\mathbf{k}'(x)\left(\mathbf{L}^{-1}\mathbf{M}\right)^q\mathbf{L}^{-1}\mathbf{k}(x) - \frac{1}{n}\mathbf{k}'(x)\left(\frac{\mathbf{M}}{n}\right)^q\mathbf{k}(x) \xrightarrow{\mathbb{P}_\mu} 0. \tag{7.51}$$

Therefore, to complete the proof we have to show that:

$$\frac{1}{n}\mathbf{k}'(x)\left(\frac{\mathbf{M}}{n}\right)^q\mathbf{k}(x) \xrightarrow{\mathbb{P}_\mu} \sum_{p>p^*}\lambda_p^{q+2}\phi_p(x)^2.$$

Let us consider for a fixed $j \geq 1$:

$$\frac{1}{n}\mathbf{k}'(x)\left(\frac{\mathbf{M}}{n}\right)^j\mathbf{k}(x) = \sum_{m_1,\ldots,m_j>p^*} a_m^{(n)}(x),$$

with $m = (m_1,\ldots,m_j)$ and:

$$\begin{aligned} a_m^{(n)}(x) &= \left(\frac{1}{n}\sum_{r=1}^{n}k(x,x_r)\phi_{m_1}(x_r)\right)\left(\frac{1}{n}\sum_{r=1}^{n}k(x,x_r)\phi_{m_j}(x_r)\right) \\ &\quad \times\prod_{l=1}^{j-1}\frac{1}{n}\sum_{r=1}^{n}\phi_{m_l}(x_r)\phi_{m_{l+1}}(x_r)\prod_{i=1}^{j}\lambda_{m_i}. \end{aligned}$$

From Cauchy-Schwarz inequality, we have:

$$\left|a_m^{(n)}(x)\right| \leq \left(\frac{1}{n}\sum_{r=1}^{n}k(x,x_r)^2\right)\prod_{i=1}^{j}\frac{1}{n}\sum_{r=1}^{n}\lambda_{m_i}\phi_{m_i}(x_r)^2 \tag{7.52}$$

$$\leq \sigma^4\prod_{i=1}^{j}\frac{1}{n}\sum_{r=1}^{n}\lambda_{m_i}\phi_{m_i}(x_r)^2. \tag{7.53}$$

Therefore, considering the expectation with respect to the distribution of the points $(x_r)_{r=1,\dots,n}$, we have:

$$\mathbb{E}_\mu\left[\left|a_m^{(n)}(x)\right|\right] \quad \leq \quad \sigma^4\left(\prod_{i=1}^{j}\lambda_{m_i}\right)\frac{1}{n^j}\sum_{t_1,\dots,t_j=1}^{n}\mathbb{E}_\mu\left[\phi_{m_1}(X_{t_1})^2\dots\phi_{m_j}(X_{t_j})^2\right] \qquad \forall x\in\mathbb{R}^d.$$

The following inequality holds uniformly in $t_1,\dots,t_j=1,\dots,n$:

$$\mathbb{E}_\mu\left[\prod_{i=1}^{j}\phi_{m_i}(X_{t_i})^2\right] \quad \leq \quad b_m,$$

where $b_m=\sum_{\substack{\mathcal{P}\in\Pi(\{1,\dots,j\})\\ \mathcal{P}=\cup_{r=1}^{l}I_r}}\prod_{r=1}^{l}\mathbb{E}_\mu\left[\prod_{i\in I_r}\phi_{m_i}(X)^2\right]$ because the term of left hand side of the inequality is equal to one of the terms in the sum of the right hand side. Here $\Pi(\{1,\dots,j\})$ is the collection of all partitions of $\{1,\dots,j\}$ and $I_r\cap I_{r'}=\emptyset$, $\forall r\neq r'$. We hence have:

$$\mathbb{E}_\mu\left[\left|a_m^{(n)}(x)\right|\right] \quad \leq \quad \sigma^4\prod_{i=1}^{j}\lambda_{m_i}b_m.$$

Since $\sum_{p\geq 0}\lambda_p\phi_p(x)^2\leq\sigma^2$, we have:

$$\begin{aligned}
\sum_{m_1,\dots,m_j>p^*}\prod_{i=1}^{j}\lambda_{m_i}b_m \quad &= \quad \sum_{m_1,\dots,m_j>p^*}\prod_{l=1}^{j}\lambda_{m_l}\sum_{\substack{\mathcal{P}\in\Pi(\{1,\dots,j\})\\ \mathcal{P}=\cup_{r=1}^{l}I_r}}\prod_{r=1}^{l}\mathbb{E}_\mu\left[\prod_{i\in I_r}\phi_{m_i}(X)^2\right]\\
&= \quad \sum_{\substack{\mathcal{P}\in\Pi(\{1,\dots,j\})\\ \mathcal{P}=\cup_{r=1}^{l}I_r}}\prod_{r=1}^{l}\mathbb{E}_\mu\left[\prod_{i\in I_r}\sum_{m_i>p^*}\lambda_{m_i}\phi_{m_i}(X)^2\right]\\
&\leq \quad \sigma^{2j}\#\{\Pi(\{1,\dots,j\})\}.
\end{aligned}$$

Since the cardinality of the collection $\Pi(\{1,\dots,j\})$ of partitions of $\{1,\dots,j\}$ is finite, the series $\sum_{m_1,\dots,m_j>p^*}\prod_{i=1}^{j}\lambda_{m_i}b_m$ converges. Furthermore, as it is a series with non-negative terms, $\forall\varepsilon>0$, $\exists\bar{p}>p^*$ such that :

$$\sigma^4\sum_{m\in M_{\bar{p}}^C}\prod_{i=1}^{j}\lambda_{m_i}b_m\leq\varepsilon,$$

where $M_{\bar{p}}^C$ designs the complement of $M_{\bar{p}}$ defined by the collection of $m=(m_1,\dots,m_j)$ such that:

$$M=\{m=(m_1,\dots,m_j)\quad\text{such that}\quad m_i>p^*,\quad i=1,\dots,j\},$$

$$M_{\bar{p}}=\{m=(m_1,\dots,m_j)\quad\text{such that}\quad p^*<m_i\leq\bar{p},\quad i=1,\dots,j\},$$

$$M_{\bar{p}}^C=M\setminus M_{\bar{p}}.$$

Therefore, we have $\forall \delta > 0$, $\forall \varepsilon > 0$ $\exists \bar{p} > 0$ such that uniformly in n:

$$\sum_{m \in M_{\bar{p}}^C} \mathbb{E}_\mu \left[\left| a_m^{(n)}(x) \right| \right] \leq \frac{\varepsilon \delta}{2}.$$

Applying the Markov inequality, we obtain:

$$\mathbb{P} \left(\sum_{m \in M_{\bar{p}}^C} \left| a_m^{(n)}(x) \right| > \frac{\delta}{2} \right) \leq \varepsilon. \tag{7.54}$$

Furthermore, by denoting $a_m(x) = \lim_{n \to \infty} a_m^{(n)}(x)$, we have:

$$a_m(x) = \lambda_{m_1} \lambda_{m_j} \phi_{m_1}(x) \phi_{m_j}(x) \prod_{i=1}^{j} \lambda_{m_i} \prod_{i=1}^{j-1} \delta_{m_i = m_{i+1}} \tag{7.55}$$

and from Cauchy-Schwarz inequality (see Equation (7.53)), we have:

$$|a_m(x)| \leq \sigma^4 \prod_{i=1}^{j} \lambda_{m_i}.$$

We hence can deduce the inequality:

$$\sum_{m \in M_{\bar{p}}^C} |a_m(x)| \leq \sigma^4 \sum_{m \in M_{\bar{p}}^C} \prod_{i=1}^{j} \lambda_{m_i}. \tag{7.56}$$

Thus, $\exists \bar{p}$ such that $\sum_{m \in M_{\bar{p}}^C} |a_m(x)| \leq \frac{\delta}{2}$ for all $x \in \mathbb{R}^d$. From the inequalities (7.54) and (7.56), we find that $\exists \bar{p}$ such that:

$$\mathbb{P}_\mu \left(\left| \sum_{m \in M} a_m^{(n)}(x) - \sum_{m \in M} a_m(x) \right| > 2\delta \right) \leq \varepsilon + \mathbb{P}_\mu \left(\left| \sum_{m \in M_{\bar{p}}} a_m^{(n)}(x) - \sum_{m \in M_{\bar{p}}} a_m(x) \right| > \delta \right).$$

Since $M_{\bar{p}}$ is a finite set:

$$\limsup_{n \to \infty} \mathbb{P}_\mu \left(\left| \sum_{m \in M_{\bar{p}}} a_m^{(n)}(x) - \sum_{m \in M_{\bar{p}}} a_m(x) \right| > \delta \right) = 0,$$

therefore:

$$\limsup_{n \to \infty} \mathbb{P}_\mu \left(\left| \sum_{m \in M} a_m^{(n)}(x) - \sum_{m \in M} a_m(x) \right| > 2\delta \right) \leq \varepsilon.$$

The previous inequality holds $\forall \varepsilon > 0$, thus we have the convergence in probability of $\sum_{m \in M} a_m^{(n)}(x)$ to $\sum_{m \in M} a_m(x)$ with (by using the limit in Equation (7.55)):

$$\sum_{m \in M} a_m(x) = \sum_{p > p^*} \lambda_p^{j+2} \phi_p(x)^2.$$

Finally, we have the following convergence in probability when $n \to \infty$:

$$\mathbf{k}'(x)(\mathbf{L}^{-1}\mathbf{M})^i\mathbf{L}^{-1}\mathbf{k}(x) \overset{n\to\infty}{\longrightarrow} \left(\frac{1}{\tau}\right)^{i+1} \sum_{p>p^*} \lambda_p^{i+2}\phi_p(x)^2. \tag{7.57}$$

We highlight that we cannot use the strong law of large numbers here due to the infinite sum in $\mathbf{M}$.

From Equation (7.37) and the convergences (7.40) and (7.51), we obtain the following convergence in probability:

$$\sigma^2_{LUP}(x) \overset{n\to\infty}{\longrightarrow} \sum_{p\geq 0}\left(\lambda_p - \frac{\lambda_p^2}{\tau+\lambda_p}\right)\phi_p(x)^2 - \sum_{p>p^*}\lambda_p^2\frac{\left(\frac{\lambda_p}{\tau}\right)^{2q+1}}{\tau+\lambda_p}\phi_p(x)^2. \tag{7.58}$$

By considering the limit $q \to \infty$ and the inequality $\lambda_{p^*} < \tau$, we obtain the following upper bound for $\sigma^2(x)$:

$$\limsup_{n\to\infty} \sigma^2(x) \leq \sum_{p\geq 0}\frac{\tau\lambda_p}{\tau+\lambda_p}\phi_p(x)^2. \tag{7.59}$$

7.8 Conclusion

The main result of this chapter is a theorem giving the Gaussian process regression mean squared error when the number of observations is large and the observation noise variance is proportional to the number of observations. The asymptotic value of the mean squared error is derived in terms of the eigenvalues and eigenfunctions of the covariance function and holds for degenerate and non-degenerate kernels and for any dimension. We emphasize that a noise variance proportional to the number of observations is natural in the framework of experiments with replications or Monte-Carlo simulators.

From this theorem, we can deduce the asymptotic behavior of the generalization error - defined in this chapter as the integrated mean squared error - as a function of the reduced observation noise variance (it corresponds to the noise variance when the number of observations equals one). This result generalizes previous ones which give this behavior in dimension one or two or for a restricted class of covariance kernels (for degenerate ones). The significant differences between the rate of convergence of degenerate and non-degenerate kernels highlight the relevance of our theorem which holds for non-degenerate kernels. This is especially important as usual kernels for Gaussian process regression are non-degenerate.

Our work deals with Gaussian process regression when the variance of the noise can be reduced by increasing the budget (i.e. the number of replications at each point). Our results are of practical interest in this case since it gives the total budget needed to reach a precision prescribed by the user. We efficiency of the presented result is emphasize on an industrial application to the safety assessment of a nuclear system containing fissile materials.

Chapter 8

Asymptotic normality of a Sobol index estimator in noisy kriging framework

8.1 Introduction

As in the noisy-free case presented in Chapter 6, stochastic simulators commonly have a large number d of input parameters for which we want to measure their importance on the model output. Like in Chapter 6, we focus on the variance-based Sobol indices [Sobol, 1993] coming from the Hoeffding-Sobol decomposition [Hoeffding, 1948]. We recall that we consider independent input random variables.

Monte-Carlo methods are widely used to estimate the Sobol indices (see [Sobol, 1993], [Sobol et al., 2007] and [Janon et al., 2012]). Their main advantages are that they allow for quantifying the uncertainty related to the estimation errors. In particular, for non-asymptotic cases, this can be easily carried out with a bootstrap procedure as presented in [Archer et al., 1997] and [Janon et al., 2011]. Furthermore, in asymptotic cases, useful properties can be shown as the asymptotic normality [Janon et al., 2012]. The reader is referred to [van der Vaart, 1998] for an extensive presentation of asymptotic statistics.

Nevertheless, Monte-Carlo methods require a large number of simulations and are often unachievable under reasonable time constraints. Therefore, in order to avoid prohibitive computational costs, we surrogate the simulator with a meta-model and we perform the estimations on it. In this chapter, we consider a special surrogate model corresponding to a Gaussian process regression with a large number of observations. Indeed, we have seen in Chapter 7 that in a stochastic simulator framework with a fixed budget, the noise variance of the observations is proportional to their number. Therefore, in principle we have to make a trade-off between the number of simulations and the output accuracy. Actually, we consider the asymptotic case where the number of observations tends to infinity.

More precisely we consider an idealized regression problem for which we can deduce a posterior predictive mean and variance tractable for our purpose. Furthermore, thanks to the results presented in Chapter 7, we can explicitly derive the rate of convergence of this meta-model approximation error with respect to the computational budget. Therefore, the

Sobol index estimates - which are evaluated with a Monte-Carlo procedure by replacing the true code with the posterior predictive mean - have two sources of uncertainty: the one related to the Monte-Carlo scheme and the one related to the meta-model approximation. The error due to the Monte-Carlo procedure tends to zero when the number of particles (calls of the meta-model) tends to infinity and as presented in Chapter 7 the error due to the meta-model tends to zero when the budget (calls of the complex simulator used to build the meta-model) tends to infinity. A question of interest is whether the asymptotic normality presented in [Janon et al., 2011] is maintained.

The aim of this chapter is thus to provide conditions on the budget and the number of Monte-Carlo particles which ensure the asymptotic normality of a Sobol index estimator. The principal difficulty of the study is that the estimator lies in a product probability space which takes into account the uncertainty of the Gaussian process and the one of the Monte-Carlo sample.

We emphasize that [Janon et al., 2011] present such a result for noise-free Gaussian process regression using a squared exponential covariance kernel (see Subsection 1.4.2). They give conditions on the number of simulations and the number of Monte-Carlo particles which ensure the asymptotic normality for the Sobol index estimators. A part of our developments is inspired by their work nevertheless they are different with some important respects. Indeed, the particular case of noise-free Gaussian process regression with squared exponential covariance kernel allows for not considering the probability space in which lies the Gaussian process. This significantly simplifies the mathematical developments. Unfortunately this simplification does not hold in our general framework.

The main result of this chapter is a theorem giving sufficient conditions to ensure the asymptotic normality of Sobol indices estimators based on the Monte-Carlo procedure of [Sobol, 1993] through the presented Gaussian process regression and for a large class of covariance kernels. The asymptotic normality is of interest since it allows for giving asymptotic confidence intervals on the Sobol index estimators. This result is illustrated with an academic example dealing with a partial differential equations problem.

8.2 Gaussian process regression for stochastic simulators

We present in Subsection 8.2.1 the practical problem that we want to deal with. In order to handle the asymptotic framework of a large number of observations, we replace the true problem by an idealized version of it in Subsection 8.2.2. This idealization allows us to study the asymptotic normality of the Sobol index estimator in Section 8.3.

8.2.1 Gaussian process regression with a large number of observations

Let us suppose that we want to surrogate a function $f(x)$, $x \in Q \subset \mathbb{R}^d$, from noisy observations of it at points $(x_i)_{i=1,\dots,n}$ sampled from the probability measure μ - μ is called the design measure and Q is an nonempty open set. Furthermore, we consider that we have r replications at each point. We hence have ns experiments of the form $z_{i,j} = f(x_i) + \varepsilon_{i,j}$, $i = 1, \dots, n$,

$j = 1, \ldots, s$ and we consider that $(\varepsilon_{i,j})_{\substack{i=1,\ldots,n \\ j=1,\ldots,s}}$ are independently sampled from a Gaussian distribution with mean zero and variance σ_ε^2. A stochastic simulator provides outputs of the following form

$$z_i = \frac{1}{s}\sum_{j=1}^{s} z_{i,j} = f(x_i) + \varepsilon_i, \quad \forall i = 1, \ldots, n,$$

where $(\varepsilon_i)_{i=1,\ldots,n}$ are the observation noises sampled from a zero-mean Gaussian distribution with variance σ_ε^2/s. Therefore, if we consider a fixed number of experiments $T = ns$, we have an observation noise variance equal to $n\sigma_\varepsilon^2/T$.

Note that an observation noise variance proportional to n is natural in the framework of stochastic simulators as presented in Chapter 7. Indeed, for a fixed total number of experiments $T = ns$, we can either decide to perform them in few points (i.e. n small) but with lot of replications (i.e. s large) or decide to perform them in lot of points (i.e. n large) but with few replications (i.e. s small).

In a Gaussian process regression framework, we model $f(x)$ as a Gaussian process with a known mean (that we take equal to zero without loss of generality) and a covariance kernel $k(x, \tilde{x})$. Therefore, in the remainder of this chapter, the function $f(x)$ is random. The predictive Mean Squared Error (MSE) of the Best Linear Unbiased Predictor (BLUP) given by

$$\hat{z}_{T,n}(x) = \mathbf{k}'(x)\left(\mathbf{K} + \frac{n\sigma_\varepsilon^2}{T}\mathbf{I}\right)^{-1}\mathbf{z}^n, \tag{8.1}$$

is

$$\sigma_{T,n}^2(x) = k(x,x) - \mathbf{k}'(x)\left(\mathbf{K} + \frac{n\sigma_\varepsilon^2}{T}\mathbf{I}\right)^{-1}\mathbf{k}(x), \tag{8.2}$$

where $\mathbf{z}^n = (z_i)_{i=1,\ldots,n}$ denotes the vector of the observed values, $\mathbf{k}(x) = [k(x, x_i)]_{1\le i\le n}$ is the n-vector containing the covariances between $f(x)$ and $f(x_i)$, $1 \le i \le n$, $\mathbf{K} = [k(x_i, x_j)]_{1\le i,j\le n}$ is the $n\times n$-matrix containing the covariances between $f(x_i)$ and $f(x_j)$, $1 \le i, j \le n$ and $\mathbf{I}$ is the $n \times n$ identity matrix.

In this chapter, we consider the case $n \gg 1$. It corresponds to a massive experimental design set but with observations with a large noise variance. This case is realistic for stochastic simulators where the computational cost resulting from one Monte-Carlo particle is very low and thus can be run in lot of points $(x_i)_{i=1,\ldots,n}$.

8.2.2 Idealized Gaussian process regression

We assume from now on that the positive kernel $k(x, \tilde{x})$ is continuous and that $\sup_{x\in Q} k(x, x) < \infty$ where Q is a nonempty open subset of $\mathbb{R}^d$. We introduce the Mercer's decomposition of $k(x, \tilde{x})$ (see Chapter 1 Section 1.4):

$$k(x, \tilde{x}) = \sum_{p\ge 0} \lambda_p \phi_p(x)\phi_p(\tilde{x}), \tag{8.3}$$

where $(\phi_p(x))_p$ is an orthonormal basis of $L^2_\mu(Q)$ consisting of eigenfunctions of the integral operator $(T_{\mu,k}g)(x) = \int_{\mathbb{R}^d} k(x,u)g(u)d\mu(u)$ and λ_p is the nonnegative sequence of corresponding eigenvalues sorted in decreasing order.

Let us consider the following predictor:

$$\hat{z}_T(x) = \sum_{p\geq 0} \frac{\lambda_p}{\lambda_p + \sigma_\varepsilon^2/T} z_p \phi_p(x), \tag{8.4}$$

where $z_p = f_p + \varepsilon_p^*$, $f_p = \int f(x)\phi_p(x)\, d\mu(x)$, $\varepsilon_p^* \sim \mathcal{N}\left(0, \sigma_\varepsilon^2/T\right)$, ε_p^* independent of ε_q^* for $p \neq q$ and $(\varepsilon_p^*)_{p\geq 0}$ independent of $(f_p)_{p\geq 0}$. Note that we have $f_p \sim \mathcal{N}(0, \lambda_p)$, f_p independent of f_q for $p \neq q$ and $f(x) = \sum_{p\geq 0} f_p \phi_p(x)$.

Let us introduce the probability space $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z) = (\Omega_f \times \Omega_\varepsilon, \sigma(\mathcal{F}_f \times \mathcal{F}_\varepsilon), \mathbb{P}_f \times \mathbb{P}_\varepsilon)$ where $(\Omega_f, \mathcal{F}_f, \mathbb{P}_f)$ corresponds to the probability space where $f(x)$ and the sequence $(f_p)_{p\geq 0}$ are defined and $(\Omega_\varepsilon, \mathcal{F}_\varepsilon, \mathbb{P}_\varepsilon)$ is the probability space where the observation noises $(\varepsilon_i)_{i\in\mathbb{N}}$ and the sequence $(\varepsilon_p^*)_{p\geq 0}$ are defined. Further, let us consider the sequence of independent random variables $(X_i)_{i\in\mathbb{N}}$ with probability measure μ on $Q \subseteq \mathbb{R}^d$ and defined on the probability space $(\Omega_D, \mathcal{F}_D, \mathbb{P}_D)$. The sequence $(X_i)_{i=1,\dots,n}$ represents the experimental design set considered as a random variable. Therefore, the predictors $\hat{z}_{T,n}(x)$ in (8.1) and $\hat{z}_T(x)$ in (8.4) are associated to the random experimental design set $(X_i)_{i\in\mathbb{N}}$. We have the following convergence in probability when $n \to \infty$ (see Chapter 7 Theorem 7.1):

$$\sigma_{T,n}^2(x) \xrightarrow[n\to\infty]{\mathbb{P}_D} \sigma_T^2(x), \tag{8.5}$$

where $\sigma_{T,n}^2(x) = \mathbb{E}_Z\left[(\hat{z}_{T,n}(x) - f(x))^2\right]$ (8.2) and $\sigma_T^2(x) = \mathbb{E}_Z\left[(\hat{z}_T(x) - f(x))^2\right]$. We recall:

$$\sigma_T^2(x) = \sum_{p\geq 0} \frac{\sigma_\varepsilon^2 \lambda_p/T}{\sigma_\varepsilon^2/T + \lambda_p} \phi_p(x)^2. \tag{8.6}$$

Therefore $\hat{z}_T(x)$ in (8.4) is a relevant candidate for an idealized version of $\hat{z}_{T,n}(x)$ in (8.1) for the considered asymptotics $n \to \infty$. The following proposition allows for completing the justification of the relevance of $\hat{z}_{T,n}(x)$.

Proposition 8.1. *Let us consider $f(x)$ a Gaussian process of zero mean and covariance kernel $k(x, \tilde{x})$, $\hat{z}_{T,n}(x)$ in (8.1) and $\hat{z}_T(x)$ in (8.4) both associated to the random experimental design set $(X_i)_{i\in\mathbb{N}}$. Consequently $f(x) = \sum_{p\geq 0} f_p \phi_p(x)$ where $f_p \sim \mathcal{N}(0, \lambda_p)$, $(f_p)_{p\geq 0}$ independent and $(\phi_p(x))_{p\geq 0}$ defined in (8.3). The following convergence holds $\forall \delta > 0$ and for any Borel set $A \subset \mathbb{R}^2$ such that the Lebesgue measure its boundary is zero:*

$$\mathbb{P}_D\left(\left|\mathbb{P}_Z\left((\hat{z}_{T,n}(x), f(x)) \in A\right) - \mathbb{P}_Z\left((\hat{z}_T(x), f(x)) \in A\right)\right| > \delta\right) \xrightarrow{n\to\infty} 0. \tag{8.7}$$

Proof of Proposition 8.1. First of all, we note that for a fixed $\omega_D \in \Omega_D$ the random variables $(\hat{z}_{T,n}(x), f(x))$ and $(\hat{z}_T(x), f(x))$ are Gaussian since they are linear transformations of $((\varepsilon_i)_{i\in\mathbb{N}}, (f_p)_{p\geq 0})$ and $((\varepsilon_p^*)_{p\geq 0}, (f_p)_{p\geq 0})$ which are both independently distributed from Gaussian distributions.

Thanks to the equality $\mathbb{E}_Z\left[(\hat{z}_{T,n}(x))^2\right] = k(x,x) - \sigma_{T,n}^2(x)$ with $k(x,x) = \sum_{p\geq 0} \lambda_p \phi_p(x)^2$, to the definition of $\hat{z}_T(x)$ in (8.4) and to the convergence (8.5), the following convergence holds in probability:

$$\mathbb{E}_Z\left[(\hat{z}_{T,n}(x))^2\right] \xrightarrow[n\to\infty]{\mathbb{P}_D} \mathbb{E}_Z\left[(\hat{z}_T(x))^2\right]. \tag{8.8}$$

Furthermore, we also have the equality $\mathbb{E}_Z\left[\hat{z}_{T,n}(x)f(x)\right] = k(x,x) - \sigma^2_{T,n}(x)$ that leads the convergence:

$$\mathbb{E}_Z\left[\hat{z}_{T,n}(x)f(x)\right] \xrightarrow[n\to\infty]{\mathbb{P}_D} \mathbb{E}_Z\left[\hat{z}_T(x)f(x)\right]. \tag{8.9}$$

We can deduce the following convergence of the covariance of the two-dimensional Gaussian vector $(\hat{z}_{T,n}(x), f(x))$ to the one of the two-dimensional Gaussian vector $(\hat{z}_T(x), f(x))$:

$$\mathrm{cov}_Z\left((\hat{z}_{T,n}(x), f(x))\right) \xrightarrow[n\to\infty]{\mathbb{P}_D} \mathrm{cov}_Z\left((\hat{z}_T(x), f(x))\right). \tag{8.10}$$

Furthermore, the following equality holds:

$$\mathbb{E}_Z\left[(\hat{z}_{T,n}(x), f(x))\right] = \mathbb{E}_Z\left[(\hat{z}_T(x), f(x))\right] = (0,0). \tag{8.11}$$

Let us denote by $\mathbf{C}_n = \mathrm{cov}_Z\left((\hat{z}_{T,n}(x), f(x))\right)$, for all Borel sets $A \subset \mathbb{R}^2$ such that $\nu(\partial A) = 0$ (ν denotes the Lebesgue measure and ∂A the boundary of A), we have the following equality almost surely with respect to $(\Omega_D, \mathcal{F}_D, \mathbb{P}_D)$:

$$\mathbb{P}_Z\left((\hat{z}_{T,n}(x), f(x)) \in A\right) = \phi_2\left(\mathbf{C}_n^{-1/2} A\right),$$

where ϕ_2 stands for the bivariate normal distribution $\mathcal{N}(0, \mathbf{I}_2)$. We note that $\mathbf{C}_n$ is a random variable defined on the probability space $(\Omega_D, \mathcal{F}_D, \mathbb{P}_D)$. Let us denote by $\mathbf{C} = \mathrm{cov}_Z\left((\hat{z}_T(x), f(x))\right)$. The matrix $\mathbf{C}$ being nonsingular, the convergence (8.10) implies the following one:

$$\mathbf{C}_n^{-1/2} \xrightarrow[n\to\infty]{\mathbb{P}_D} \mathbf{C}^{-1/2}.$$

Therefore, for all Borel sets $A \subset \mathbb{R}^2$ such that $\nu(\partial A) = 0$, we have:

$$\phi_2(\mathbf{C}_n^{-1/2} A) \xrightarrow[n\to\infty]{\mathbb{P}_D} \phi_2(\mathbf{C}^{-1/2} A).$$

Finally, we can deduce that $\forall \delta > 0$ and for all Borel sets $A \subset \mathbb{R}^2$ such that $\nu(\partial A) = 0$, the convergence in (8.7) holds. □

The function $\hat{z}_T(x)$ is the surrogate model that we consider in this chapter. We note that $\hat{z}_T(x)$ is not equal to the objective function $f(x)$ since $\sigma^2_\varepsilon/T \neq 0$. In practical applications, we expect that the idealized model (8.4) is close enough to the actual surrogate model (8.1) so that it provides relevant confidence intervals.

Note that with this formalism $f(x)$ is a random process defined on the probability space $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$. The random series $(z_p)_{p\geq 0}$ is defined on $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$ as well. In order to study the convergence of $\hat{z}_T(x)$ to the real function $f(x)$, let us define the Integrated Mean Squared Error (IMSE):

$$\mathrm{IMSE}_T = \int_{\mathbb{R}^d} \sigma^2_T(x)\, d\mu(x) = \mathbb{E}_Z\left[||\hat{z}_T(x) - f(x)||^2_{L^2_\mu}\right]. \tag{8.12}$$

The following equality holds:

$$\mathrm{IMSE}_T = \sum_{p\geq 0} \frac{\sigma^2_\varepsilon \lambda_p/T}{\sigma^2_\varepsilon/T + \lambda_p}. \tag{8.13}$$

We can link the asymptotic rate of convergence of the IMSE (8.13) with the asymptotic decay of the eigenvalues $(\lambda_p)_{p\geq 0}$ thanks to the following inequalities (see Chapter 7 Section 7.3):

$$B_T^2/2 \leq \mathrm{IMSE}_T \leq B_T^2, \tag{8.14}$$

with:

$$B_T^2 = \sum_{p \text{ s.t. } \lambda_p \leq \sigma_\varepsilon^2/T} \lambda_p + \frac{\sigma_\varepsilon^2}{T} \#\{p \text{ s.t. } \lambda_p > \sigma_\varepsilon^2/T\}. \tag{8.15}$$

8.3 Asymptotic normality of a Sobol index estimator

We present in this section the main theorem of this chapter about the asymptotic normality of a Sobol index estimators using Monte-Carlo integrations and the meta-model $\hat{z}_T(x)$ presented in Subsection 8.2.2. In the forthcoming development, we suppose that T is an increasing sequence indexed by the number m of Monte-Carlo particles used to estimate the variance and covariance terms involved in the Sobol index. We use the notation T_m to emphasize that T depends on m. First of all, let us define in Subsection 8.3.1 the considered Monte-Carlo estimator.

8.3.1 A Sobol index estimator

Let us suppose that the input parameter is a random vector X with probability measure $\mu = \mu_1 \otimes \mu_2$ on $(\mathbb{R}^{d_1} \times \mathbb{R}^{d_2}, \mathcal{B}(\mathbb{R}^{d_1} \times \mathbb{R}^{d_2}))$ with $d = d_1 + d_2$. We consider the random vector $(X, \tilde{X})$ defined in the probability space $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$ with $X = (X^1, X^2)$ and $\tilde{X} = (X^1, \tilde{X}^2)$ where X^1 is a random vector with values in $\mathbb{R}^{d_1}$ and with distribution μ_1, X^2 and $\tilde{X}^2$ are random vectors with values in $\mathbb{R}^{d_2}$ with distribution μ_2, and X^1, X^2 and $\tilde{X}^2$ are independent.

As presented in Chapter 6, the Sobol index of parameter X^1 can be deduced from:

$$S^{X^1} = \frac{V^{X^1}}{V} = \frac{\mathrm{var}_X\left(\mathbb{E}_X\left[f(X)|X^1\right]\right)}{\mathrm{var}_X\left(f(X)\right)} = \frac{\mathrm{cov}_X\left(f(X), f(\tilde{X})\right)}{\mathrm{var}_X\left(f(X)\right)}, \tag{8.16}$$

where the random variables $f(X)$ and $f(\tilde{X})$ are defined on the product probability space $(\Omega_Z \times \Omega_X, \sigma\left(\mathcal{F}_Z \times \mathcal{F}_X\right), \mathbb{P}_Z \times \mathbb{P}_X)$ and S^{X^1}, V^{X^1} and V are defined on the probability space $(\Omega_Z, \mathcal{F}_Z, \mathcal{F}_Z)$.

Furthermore, let us consider the sequence $(X_i, \tilde{X}_i)_{i=1}^{\infty}$ of random variables defined in $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$ independent and identically distributed such that $(X_i, \tilde{X}_i) \overset{\mathcal{L}}{=} (X, \tilde{X})$ for all $i \in \mathbb{N}^*$. We use the following estimator for (8.16) (see [Sobol, 1993]):

$$S_m^{X^1} = \frac{V_m^{X^1}}{V_m} = \frac{m^{-1}\sum_{i=1}^m f(X_i)f(\tilde{X}_i) - m^{-2}\sum_{i,j=1}^m f(X_i)f(\tilde{X}_j)}{m^{-1}\sum_{i=1}^m f^2(X_i) - m^{-2}(\sum_{i=1}^m f(X_i))^2}, \tag{8.17}$$

where the random variable $S_m^{X^1}$, $V_m^{X^1}$ and V_m are defined on the probability space $(\Omega_Z \times \Omega_X, \sigma\left(\mathcal{F}_Z \times \mathcal{F}_X\right), \mathbb{P}_Z \times \mathbb{P}_X)$. Furthermore, after substituting $f(x)$ with the meta-model $\hat{z}_{T_m}(x)$, we obtain the following estimator:

$$S_{T_m,m}^{X^1} = \frac{V_{T_m,m}^{X^1}}{V_{T_m,m}} = \frac{m^{-1}\sum_{i=1}^m \hat{z}_{T_m}(X_i)\hat{z}_{T_m}(\tilde{X}_i) - m^{-2}\sum_{i,j=1}^m \hat{z}_{T_m}(X_i)\hat{z}_{T_m}(\tilde{X}_j)}{m^{-1}\sum_{i=1}^m \hat{z}_{T_m}^2(X_i) - m^{-2}(\sum_{i=1}^m \hat{z}_{T_m}(X_i))^2}, \tag{8.18}$$

where the random variables $S^{X^1}_{T_m,m}$, $V^{X^1}_{T_m,m}$, $V_{T_m,m}$, $\hat{z}_{T_m}(X_i)$ and $\hat{z}_{T_m}(\tilde{X}_j)$ are defined on the product probability space $(\Omega_Z \times \Omega_X, \sigma(\mathcal{F}_Z \times \mathcal{F}_X), \mathbb{P}_Z \times \mathbb{P}_X)$.

8.3.2 Theorem on the asymptotic normality of the Sobol index estimator

The theorem below gives the relation between T_m and m which ensures the asymptotic normality of the estimator $V^{X^1}_{T_m,m}/V_{T_m,m}$ when $m \to \infty$. We note that $V^{X^1}_{T_m,m}/V_{T_m,m}$ is the estimator of the Sobol index $V^{X^1}/V = \mathrm{cov}_X\left(f(X), f(\tilde{X})\right)/\mathrm{var}_X(f(X))$ when we replace the true function by the surrogate model (8.4) and when we use a Monte-Carlo estimator (8.16) for the variance and covariance involved in the Sobol index.

Theorem 8.1. *Let us consider the estimator $S^{X^1}_{T_m,m}$ (8.18) of S^{X^1} (8.16) with T_m an increasing function of $m \in \mathbb{N}^*$. We have the following convergences:*

If $mB^2_{T_m} \overset{m\to\infty}{\longrightarrow} 0$, then for all interval $I \in \mathbb{R}$ and $\forall \delta > 0$, we have the convergence:

$$\mathbb{P}_Z\left(\left|\mathbb{P}_X\left(\sqrt{m}\left(S^{X^1}_{T_m,m} - S^{X^1}\right) \in I\right) - \int_I g(x)dx\right| > \delta\right) \overset{m\to\infty}{\longrightarrow} 0. \tag{8.19}$$

If $mB^2_{T_m} \overset{m\to\infty}{\longrightarrow} \infty$, then $\forall \delta > 0$, $\exists C > 0$ such that :

$$\mathbb{P}_Z\left(\left|\mathbb{P}_X\left(B^{-1}_{T_m}\left(S^{X^1}_{T_m,m} - S^{X^1}\right) \geq C\right) - 1\right| > \delta\right) \overset{m\to\infty}{\longrightarrow} 0. \tag{8.20}$$

Here $g(x)$ is the probability density function of a zero-mean Gaussian random variable with variance:

$$\frac{\mathrm{var}_X\left((f(X) - \mathbb{E}_X[f(X)])\left(f(\tilde{X}) - \mathbb{E}_X[f(X)] - S^{X^1}f(X) + S^{X^1}\mathbb{E}_X[f(X)]\right)\right)}{\left(\mathrm{var}_X(f(X))\right)^2}, \tag{8.21}$$

and $B^2_{T_m}$ is given by (8.15).

Theorem 8.1 is of interest since it gives how fast T_m has to increase with respect to m so that the error of the surrogate modeling and the one of the Monte-Carlo sampling have the same order of magnitude. Indeed, for a given size m of the Monte-Carlo sample, it is not necessary to choose a too large T_m otherwise the Monte-Carlo estimation error will dominate (it corresponds the case $mB^2_{T_m} \overset{m\to\infty}{\longrightarrow} 0$). On the other hand, if T_m is taken too large (it corresponds to the case $mB^2_{T_m} \overset{m\to\infty}{\longrightarrow} \infty$), the estimation error is dominated by the meta-model approximation.

Furthermore, we see that when $mB^2_{T_m} \overset{m\to\infty}{\longrightarrow} 0$, the asymptotic normality is assessed for the estimator $S^{X^1}_{T_m,m}$ with a variance $\mathbf{\Sigma}$. By studying the case $S^{X^1} = 0$ and $S^{X^1} = 1$ we see that the estimator is more precise for large values of Sobol indices than for small ones. A more efficient estimator for small index values is given in [Sobol et al., 2007].

We show in Section 8.5 that the product $mB^2_{T_m}$ can easily be handled when we have explicit formula for the eigenvalues of the Mercer's decomposition of $k(x, \tilde{x})$. The proof of Theorem 8.1 is given in the next subsection.

8.4 Proof of Theorem 8.1

Let us denote by $S^{X^1}_{T_m} = \mathrm{cov}_X\left(\hat{z}_{T_m}(X), \hat{z}_{T_m}(\tilde{X})\right)/\mathrm{var}_X\left(\hat{z}_{T_m}(X)\right)$ the variance of the main effect of X^1 for the surrogate model $\hat{z}_{T_m}(x)$ (8.4). The random variables S^{X^1} and $S^{X^1}_{T_m}$ are defined on the probability space $(\Omega_Z, \mathcal{F}_Z\ , \mathbb{P}_Z)$ and the random variables $S^{X^1}_{T_m,m}$, $\hat{z}_{T_m}(X)$ and $f(X)$ are defined on the product probability space $(\Omega_Z \times \Omega_X, \sigma(\mathcal{F}_Z \times \mathcal{F}_X), \mathbb{P}_Z \otimes \mathbb{P}_X)$.

Let us consider the following decomposition:

$$S^{X^1}_{T_m,m} - S^{X^1} = S^{X^1}_{T_m,m} - S^{X^1}_{T_m} + S^{X^1}_{T_m} - S^{X^1}. \tag{8.22}$$

In a first hand we deal with the convergence of $\sqrt{m}\left(S^{X^1}_{T_m,m} - S^{X^1}_{T_m}\right)$. We handle this problem thanks to the Skorokhod representation theorem, the Lindeberg-Feller theorem and the Delta method. In a second hand, we study the convergence of $\sqrt{m}\left(S^{X^1}_{T_m} - S^{X^1}\right)$ through the Skorokhod representation theorem.

In the forthcoming developments, we consider that $mB^2_{T_m} \overset{m\to\infty}{\longrightarrow} 0$. Therefore, there exists $g(T_m)$ such that $g(T_m) \overset{m\to\infty}{\longrightarrow} 0$ and $mB^2_{T_m} g^{-2}(T_m) \overset{m\to\infty}{\longrightarrow} 0$. The function $g(T_m)$ considered in the remainder of this section satisfies this property.

8.4.1 The Skorokhod representation theorem

Let us consider the following random variables defined on the probability space $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$:

$$a_{T_m}(x) = (\hat{z}_{T_m}(x) - f(x))B^{-1}_{T_m} g(T_m), \tag{8.23}$$

$$b_{T_m}(x) = (\hat{z}_{T_m}(x) - f(x))g(T_m)^{1/3} B^{-1/3}_{T_m}. \tag{8.24}$$

Markov's inequality and (8.14) give us $\forall \delta > 0$:

$$\mathbb{P}_Z(||a_{T_m}(x)||^2_{L^2_\mu} > \delta) \leq \mathbb{E}_Z(||a_{T_m}(x)||^2_{L^2_\mu})/\delta \leq g(T_m)^2/\delta.$$

Therefore, we have the following convergence in probability in $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$:

$$\lim_{m\to\infty} ||a_{T_m}(x)||^2_{L^2_\mu} = 0$$

and the inequalities in (8.14) ensure the following one:

$$||a_{T_m}(x)||^2_{L^2_\mu} \geq g(T_m)^2/2. \tag{8.25}$$

Furthermore, the following equality stands since $f(x)$ is a Gaussian process:

$$\mathbb{E}_Z[(\hat{z}_{T_m}(x) - f(x))^6] = 15\sigma^6_{T_m}(x).$$

Cauchy-Schwarz inequality leads to:

$$\mathbb{E}_Z[||\hat{z}_{T_m}(x) - f(x)||^6_{L^6_\mu}] \leq 15 \int \sigma^6_{T_m}(x)\, d\mu(x) \leq 15 B^2_{T_m} \sup_x k^2(x,x).$$

Therefore, thanks to Markov's inequality we have:

$$\mathbb{P}_Z(||b_{T_m}(x)||^6_{L^6_\mu} > \delta) \leq 15 g(T_m)^2 \sup_x k^2(x,x)/\delta$$

and the following convergence stands in probability in $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$:

$$\lim_{m\to\infty} ||b_{T_m}(x)||^6_{L^6_\mu} = 0.$$

Therefore, we have the following convergences in probability in $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$ when $m \to \infty$:

$$\begin{cases} f(x) \\ a_{T_m}(x) = (\hat{z}_{T_m}(x) - f(x))g(T_m)B^{-1}_{T_m} \\ b_{T_m}(x) = (\hat{z}_{T_m}(x) - f(x))g(T_m)^{1/3}B^{-1/3}_{T_m} \end{cases} \overset{L^6_\mu \times L^2_\mu \times L^6_\mu}{\underset{m\to\infty}{\longrightarrow}} \begin{pmatrix} f(x) \\ 0 \\ 0 \end{pmatrix}.$$

As $L^6_\mu \times L^2_\mu \times L^6_\mu$ is separable we can use the Skorokhod's representation theorem [Billingsley, 1999] presented below.

Theorem 8.2 (Skorokhod's representation theorem)**.** *Let μ_n, $n \in \mathbb{N}$ be a sequence of probability measures on a topological space S; suppose that μ_n converges weakly to some probability measure μ on S as $n \to \infty$. Suppose also that the support of μ is separable. Then there exist random variables X_n and X defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$ such that:*

(i) μ_n is the distribution of X_n

(ii) μ is the distribution of X

(iii) $X_n(\omega) \to X(\omega)$ as $n \to \infty$ for every $\omega \in \Omega$.

Therefore, there is a probability space denoted by $(\tilde{\Omega}_Z, \tilde{\mathcal{F}}_Z, \tilde{\mathbb{P}}_Z)$ such that

$$(\tilde{f}_{T_m}(x), \tilde{a}_{T_m}(x), \tilde{b}_{T_m}(x)) \overset{\mathcal{L}}{=} (f(x), a_{T_m}(x), b_{T_m}(x)), \tag{8.26}$$

with $(\tilde{f}_{T_m}(x), \tilde{a}_{T_m}(x), \tilde{b}_{T_m}(x))$, $\tilde{f}(x)$ defined on $(\tilde{\Omega}_Z, \tilde{\mathcal{F}}_Z, \tilde{\mathbb{P}}_Z)$ and $(f(x), a_{T_m}(x), b_{T_m}(x))$ defined on $(\Omega_Z, \mathcal{F}_Z, \mathbb{P}_Z)$ - and $\forall \tilde{\omega}_Z \in \tilde{\Omega}_Z$ the following convergence holds for $m \to \infty$:

$$(\tilde{f}_{T_m}(x), \tilde{a}_{T_m}(x), \tilde{b}_{T_m}(x)) \overset{L^6_\mu \times L^2_\mu \times L^6_\mu}{\underset{m\to\infty}{\longrightarrow}} (\tilde{f}(x), 0, 0). \tag{8.27}$$

First, let us build below the analogous of $z_{T_m}(x)$ in $(\tilde{\Omega}_Z, \tilde{\mathcal{F}}_Z, \tilde{\mathbb{P}}_Z)$. For a fixed $T_m > 0$, we have the equality $a_{T_m}(x)g(T_m)^{-1}B_{T_m} = b_{T_m}(x)g(T_m)^{-1/3}B^{1/3}_{T_m}$. Therefore, we have

$$||a_{T_m}(x)g(T_m)^{-1}B_{T_m} - b_{T_m}(x)g(T_m)^{-1/3}B^{1/3}_{T_m}||_{L^2_\mu} = 0$$

and

$$\mathbb{P}_Z\left(||a_{T_m}(x)g(T_m)^{-1}B_{T_m} - b_{T_m}(x)g(T_m)^{-1/3}B^{1/3}_{T_m}||_{L^2_\mu} = 0\right) = 1.$$

The equality $(\tilde{a}_{T_m}(x), \tilde{b}_{T_m}(x)) \overset{\mathcal{L}}{=} (a_{T_m}(x), b_{T_m}(x))$ leads to the following one

$$\tilde{\mathbb{P}}_Z\left(||\tilde{a}_{T_m}(x)g(T_m)^{-1}B_{T_m} - \tilde{b}_{T_m}(x)g(T_m)^{-1/3}B^{1/3}_{T_m}||_{L^2_\mu} = 0\right) = 1.$$

Thus, for μ-almost every $\tilde{\omega_Z}$ in $\tilde{\Omega}_Z$, we have

$$||\tilde{a}_{T_m}(x)g(T_m)^{-1}B_{T_m} - \tilde{b}_{T_m}(x)g(T_m)^{-1/3}B_{T_m}^{1/3}||_{L^2_\mu} = 0. \tag{8.28}$$

If we consider such a $\tilde{\omega}_Z$, we have the equality $\tilde{a}_{T_m}(x)g(T_m)^{-1}B_{T_m} = \tilde{b}_{T_m}(x)g(T_m)^{-1/3}B_{T_m}^{1/3}$ for μ-almost every x

Let us denote by

$$\tilde{z}_{T_m}(x) = \tilde{f}_{T_m}(x) + g(T_m)^{-1}B_{T_m}\tilde{a}_{T_m}(x),$$

$\tilde{z}_{T_m}(x)$ is defined on $(\tilde{\Omega}_Z, \tilde{\mathcal{F}}_Z, \tilde{\mathbb{P}}_Z)$. For a fixed $\tilde{\omega}_Z \in \tilde{\Omega}_Z$ such that (8.28) holds, we have the equality $\tilde{z}_{T_m}(x) = \tilde{f}_{T_m}(x) + g(T_m)^{-1/3}B_{T_m}^{1/3}\tilde{b}_{T_m}(x)$ for μ-almost every x.

8.4.2 Convergences with a fixed $\tilde{\omega}_Z \in \tilde{\Omega}_Z$

Let us consider a fixed $\tilde{\omega}_Z \in \tilde{\Omega}_Z$ such that (8.28) holds. We aim to study the convergence of $\sqrt{m}\left(\tilde{S}^{X^1}_{T_m,m} - \tilde{S}^{X^1}_{T_m}\right)$ and $\sqrt{m}\left(\tilde{S}^{X^1}_{T_m} - \tilde{S}^{X^1}\right)$ in $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$ with:

$$\tilde{S}^{X^1} = \mathrm{cov}_X(\tilde{f}(X), \tilde{f}(\tilde{X}))/\mathrm{var}_X(\tilde{f}(X)), \tag{8.29}$$

$$\tilde{S}^{X^1}_{T_m} = \mathrm{cov}_X(\tilde{z}_{T_m}(X), \tilde{z}_{T_m}(\tilde{X}))/\mathrm{var}_X(\tilde{z}_{T_m}(X)) \tag{8.30}$$

and

$$\tilde{S}^{X^1}_{T_m,m} = \frac{m^{-1}\sum_{i=1}^n \tilde{z}_{T_m}(X_i)\tilde{z}_{T_m}(\tilde{X}_i) - m^{-2}\sum_{i,j=1}^n \tilde{z}_{T_m}(X_i)\tilde{z}_{T_m}(\tilde{X}_j)}{m^{-1}\sum_{i=1}^n \tilde{z}^2_{T_m}(X_i) - m^{-2}(\sum_{i=1}^n \tilde{z}_{T_m}(X_i))^2}. \tag{8.31}$$

Convergence of $\sqrt{m}\left(\tilde{S}^{X^1}_{T_m,m} - \tilde{S}^{X^1}_{T_m}\right)$ in $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$

Let us denote by $Y_{T_m,i} = \tilde{z}_{T_m}(X_i)$, $Y^{X^1}_{T_m,i} = \tilde{z}_{T_m}(\tilde{X}_i)$ and

$$\begin{aligned} U_{T_m,i} = \quad & \Big((Y_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}])(Y^{X^1}_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}]), \\ & Y_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}], Y^{X^1}_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}], (Y_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}])^2\Big) \end{aligned}. \tag{8.32}$$

Since $\tilde{\omega}_Z \in \tilde{\Omega}_Z$ is fixed, $Y_{T_m,i}$, $Y^{X^1}_{T_m,i}$ and $U_{T_m,i}$ are defined on the probability space $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$. For each m, $(U_{T_m,i}/\sqrt{m})_{i=1,\dots,m}$ is a sequence of independent random vectors such that for any $\varepsilon > 0$:

$$\begin{aligned} \sum_{i=1}^m \mathbb{E}_X\left[||U_{T_m,i}||^2/m\mathbf{1}_{\{||U_{T_m,i}||>\varepsilon\sqrt{m}\}}\right] &= \mathbb{E}_X\left[||U_{T_m,1}||^2\mathbf{1}_{\{||U_{T_m,1}||>\varepsilon\sqrt{m}\}}\right] \\ &\leq \mathbb{E}_X\left[||U_{T_m,1}||^3\right]/(\varepsilon\sqrt{m}), \end{aligned}$$

since $||U_{T_m,1}|| > \varepsilon\sqrt{m}$.

We aim below to find an upper bound for $\sup_{T_m}\mathbb{E}_X\left[||U_{T_m,i}||^3\right]$. First, for any m let us consider the component $(Y_{T_m,i} - \mathbb{E}_X[Y_{T_m}])(Y^{X^1}_{T_m,i} - \mathbb{E}_X[Y_{T_m}])$. We have the following inequality:

$$\mathbb{E}_X\left[|(Y_{T_m,i} - \mathbb{E}[Y_{T_m,i}])(Y^{X^1}_{T_m,i} - \mathbb{E}[Y_{T_m,i}])|^3\right] \leq C\mathbb{E}_X\left[|Y_{T_m,i}|^6\right],$$

with $C > 0$ a constant. Minkowski inequality and the equality $\tilde{z}_{T_m}(x) = \tilde{f}_{T_m}(x) + g(T_m)^{-1/3} B_{T_m}^{1/3} \tilde{b}_{T_m}(x)$ for μ-almost every x give that there exists $C, C' > 0$ such that:

$$\mathbb{E}_X\left[|Y_{T_m,i}|^6\right] \leq C||\tilde{f}_{T_m}(x)||_{L^6_\mu}^6 + C' B_{T_m}^2 g(T_m)^{-2} ||\tilde{b}_{T_m}(x)||_{L^6_\mu}^6.$$

The convergence $(\tilde{f}_{T_m}(x), \tilde{b}_{T_m}(x)) \overset{L^6_\mu \times L^6_\mu}{\underset{m\to\infty}{\longrightarrow}} (\tilde{f}(x), 0)$ implies that there exists $C > 0$ such that for any m:

$$\mathbb{E}_X\left[|(Y_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}])(Y_{T_m,i}^{X^1} - \mathbb{E}_X[Y_{T_m,i}])|^3\right] \leq C. \tag{8.33}$$

Second, following the same guideline, we find that there exists $C, C', C'' > 0$ such that:

$$\mathbb{E}_X\left[|(Y_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}])^2|^3\right] \leq C, \tag{8.34}$$

$$\mathbb{E}_X\left[|Y_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}]|^3\right] \leq C', \tag{8.35}$$

$$\mathbb{E}_X\left[|Y_{T_m,i}^{X^1} - \mathbb{E}_X[Y_{T_m,i}]|^3\right] \leq C'. \tag{8.36}$$

Third, the inequalities (8.33), (8.35), (8.35) and (8.36) give that $\sup_{T_m} \mathbb{E}_X\left[||U_{T_m}||^3\right] < \infty$.

The inequality $\sum_{i=1}^m \mathbb{E}_X\left[||U_{T_m,i}||^2/m\mathbf{1}_{\{||U_{T_m,i}||>\varepsilon\sqrt{m}\}}\right] \leq \mathbb{E}_X\left[||U_{T_m,1}||^3\right]/(\varepsilon\sqrt{m})$ and the uniform boundedness of $\mathbb{E}_X\left[||U_{T_m}||^3\right]$ lead to the following convergence $\forall\varepsilon > 0$ when $m \to \infty$:

$$\sum_{i=1}^m \mathbb{E}_X\left[||U_{T_m,i}||^2/m\mathbf{1}_{\{||U_{T_m,i}||>\varepsilon\sqrt{m}\}}\right] = \mathbb{E}_X\left[||U_{T_m,i}||^2\mathbf{1}_{\{||U_{T_m,i}||>\varepsilon\sqrt{m}\}}\right] \overset{m\to\infty}{\longrightarrow} 0 \tag{8.37}$$

and thus $||U_{T_m,i}||^2$ is uniformly integrable.

Now, we aim to show the convergence in probability of $U_{T_m,i} \overset{m\to\infty}{\longrightarrow} U_i$ in $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$. Let us denote by

$$U_i = \left((Y_i - \mathbb{E}_X[Y_i])(Y_i^{X^1} - \mathbb{E}_X[Y_i]), Y_i - \mathbb{E}_X[Y_i], Y_i^{X^1} - \mathbb{E}_X[Y_i], (Y_i - \mathbb{E}_X[Y_i])^2\right),$$

with $Y_i = \tilde{f}(X_i)$ and $Y_i^{X^1} = \tilde{f}(\tilde{X}_i)$. The random variables U_i, Y_i and $Y_i^{X^1}$ are defined on $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$ since $\tilde{\omega}_Z \in \tilde{\Omega}_Z$ is fixed.

First, we study the term $\mathbb{E}_X\left[\left|U_{T_m,i}^{(1)} - U_i^{(1)}\right|\right]$ where $U_i^{(1)} = (Y_i - \mathbb{E}_X[Y_i])(Y_i^{X^1} - \mathbb{E}_X[Y_i])$ and $U_{T_m,i}^{(1)} = (Y_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}])(Y_{T_m,i}^{X^1} - \mathbb{E}_X[Y_{T_m,i}])$. We have the following equality:

$$\begin{aligned}\mathbb{E}_X\left[\left|U_{T_m,i}^{(1)} - U_i^{(1)}\right|\right] &= \mathbb{E}_X\Big[\Big|(Y_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}])\left((Y_{T_m,i}^{X^1} - \mathbb{E}_X[Y_{T_m,i}]) - (Y_i^{X^1} - \mathbb{E}_X[Y_i])\right) \\ &+ (Y_i^{X^1} - \mathbb{E}_X[Y_i])\left((Y_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}]) - (Y_i - \mathbb{E}_X[Y_i])\right)\Big|\Big],\end{aligned}$$

from which we deduce the inequality:

$$\begin{aligned}\mathbb{E}_X\left[\left|U_{T_m,i}^{(1)} - U_i^{(1)}\right|\right] &\leq \mathbb{E}_X\left[\left|(Y_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}])\left((Y_{T_m,i}^{X^1} - \mathbb{E}_X[Y_{T_m,i}]) - (Y_i^{X^1} - \mathbb{E}_X[Y_i])\right)\right|\right] \\ &+ \mathbb{E}_X\left[\left|(Y_i^{X^1} - \mathbb{E}_X[Y_i])\left((Y_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}]) - (Y_i - \mathbb{E}_X[Y_i])\right)\right|\right]\end{aligned}$$

and from Cauchy-Schwarz inequality there exists $C, C', C'' > 0$ such that:

$$\begin{aligned}\mathbb{E}_X\left[\left|U^{(1)}_{T_m,i} - U^{(1)}_i\right|\right] &\leq C\mathbb{E}_X\left[(Y_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}])^2\right]^{1/2}\mathbb{E}_X\left[(Y^{X^1}_{T_m,i} - Y^{X^1}_i)^2\right]^{1/2} \\ &+ C'\mathbb{E}_X\left[(Y^{X^1}_i - \mathbb{E}_X[Y_i])^2\right]^{1/2}\mathbb{E}_X\left[(Y_{T_m,i} - Y_i)^2\right]^{1/2} \\ &\leq C''\mathbb{E}_X\left[(Y_{T_m,i} - Y_i)^2\right]^{1/2}\left(\mathbb{E}_X\left[(Y^{X^1}_i)^2\right]^{1/2} + \mathbb{E}_X\left[(Y_{T_m,i})^2\right]^{1/2}\right).\end{aligned}$$

The equality $Y_{T_m,i} - Y_i = g(T_m)^{-1}B_{T_m}\tilde{a}_{T_m}(X_i)$ for $\mathbb{P}_X$-almost every $\omega_X \in \Omega_X$ implies that $\mathbb{E}_X\left[(Y_{T_m,i} - Y_i)^2\right]^{1/2} = g(T_m)^{-1}B_{T_m}\mathbb{E}_X\left[(\tilde{a}_{T_m}(X_i))^2\right]^{1/2}$. Since $\tilde{a}_{T_m}(x) \overset{m\to\infty}{\longrightarrow} 0$ in L^2_μ, we have the convergence $\mathbb{E}_X\left[(Y_{T_m,i} - Y_i)^2\right]^{1/2} \overset{m\to\infty}{\longrightarrow} 0$.

Furthermore, there exists $C, C' > 0$ such that $\mathbb{E}_X\left[(Y^{X^1}_i)^2\right]^{1/2} < C$ and $\mathbb{E}_X\left[(Y_{T_m,i})^2\right]^{1/2} < C'$ since $\tilde{z}_{T_m}(x) = \tilde{f}_{T_m}(x) + g(T_m)^{-1}B_{T_m}\tilde{a}_{T_m}(x)$, $\tilde{f}_{T_m}(x) \overset{m\to\infty}{\longrightarrow} \tilde{f}(x)$ in L^6_μ and $\tilde{a}_{T_m}(x) \overset{m\to\infty}{\longrightarrow} 0$ in L^2_μ. Therefore, we have the following convergence:

$$\mathbb{E}_X\left[\left|U^{(1)}_{T_m,i} - U^{(1)}_i\right|\right] \overset{m\to\infty}{\longrightarrow} 0. \tag{8.38}$$

Then, if we consider the terms $U^{(4)}_i = (Y_i - \mathbb{E}_X[Y_i])^2$ and $U^{(4)}_{T_m,i} = (Y_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}])^2$. Following the same guideline we find the convergence:

$$\mathbb{E}_X\left[\left|U^{(4)}_{T_m,i} - U^{(4)}_i\right|\right] \overset{m\to\infty}{\longrightarrow} 0. \tag{8.39}$$

Furthermore, denoting by $U^{(2)}_i = (Y_i - \mathbb{E}_X[Y_i])$, $U^{(2)}_{T_m,i} = (Y_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}])$, $U^{(3)}_i = (Y^{X^1}_i - \mathbb{E}_X[Y_i])$ and $U^{(3)}_{T_m,i} = (Y^{X^1}_{T_m,i} - \mathbb{E}_X[Y_{T_m,i}])$, we have the following inequalities:

$$\mathbb{E}_X\left[\left|U^{(2)}_{T_m,i} - U^{(2)}_i\right|\right] \leq C\mathbb{E}_X\left[(Y_{T_m,i} - Y_i)^2\right]^{1/2},$$

$$\mathbb{E}_X\left[\left|U^{(3)}_{T_m,i} - U^{(3)}_i\right|\right] \leq C'\mathbb{E}_X\left[(Y^{X^1}_{T_m,i} - Y^{X^1}_i)^2\right]^{1/2},$$

with C, C' positive constants. The convergences $\tilde{f}_{T_m}(x) \overset{L^6_\mu}{\to} \tilde{f}$ and $\tilde{a}_{T_m}(x) \overset{L^6_\mu}{\to} 0$ when $m \to \infty$ ensure that:

$$\mathbb{E}_X\left[\left|U^{(2)}_{T_m,i} - U^{(2)}_i\right|\right] \overset{m\to\infty}{\longrightarrow} 0 \tag{8.40}$$

and

$$\mathbb{E}_X\left[\left|U^{(3)}_{T_m,i} - U^{(3)}_i\right|\right] \overset{m\to\infty}{\longrightarrow} 0. \tag{8.41}$$

Finally, the convergences presented in (8.38), (8.39), (8.40) and (8.41) imply the desired one:

$$\mathbb{E}_X\left[||U_{T_m,i} - U_i||\right] \overset{m\to\infty}{\longrightarrow} 0. \tag{8.42}$$

Markov's inequality gives $\forall\delta > 0$:

$$\mathbb{P}_X\left(||U_{T_m,i} - U_i|| \geq \delta\right) \leq \mathbb{E}_X\left[||U_{T_m,i} - U_i||\right]/\delta. \tag{8.43}$$

The equations (8.42) and (8.43) imply the convergence $U_{T_m,i} \overset{m\to\infty}{\longrightarrow} U_i$ in probability in $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$.

This convergence in probability and the uniform integrability of $||U_{T_m,i}||^2$ implies that $U_{T_m,i} \overset{m\to\infty}{\longrightarrow} U_i$ in $L^2(\Omega_X)$ and thus $\mathrm{cov}_X(U_{T_m,i}) \overset{m\to\infty}{\longrightarrow} \mathrm{cov}_X(U_i) = \mathbf{\Sigma}$. We note that we have also the convergence $\mathbb{E}_X[U_{T_m,i}] \to \mathbb{E}_X[U_i] = \boldsymbol{\mu}$ since the convergence in $L^2(\Omega_X)$ implies the one in $L^1(\Omega_X)$.

The condition (8.37) and the convergence $\sum_{i=1}^m \mathrm{cov}_X(U_{T_m,i})/m = \mathrm{cov}_X(U_{T_m,i}) \overset{m\to\infty}{\longrightarrow} \mathbf{\Sigma}$ allow for using the Lindeberg-Feller Theorem (see [van der Vaart, 1998]) which ensures the following convergence in $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$:

$$\begin{aligned}\sum_{i=1}^m (U_{T_m,i}/\sqrt{m} - \mathbb{E}_X[U_{T_m,i}/\sqrt{m}]) &= \sqrt{m}\left(\sum_{i=1}^m (U_{T_m,i})/m - \mathbb{E}_X[U_{T_m,i}]\right) \\ &\underset{m\to\infty}{\overset{\mathcal{L}}{\longrightarrow}} \mathcal{N}(0, \mathbf{\Sigma}).\end{aligned}$$

Furthermore, we have the following equality:

$$\tilde{S}^{X^1}_{T_m,m} = \Phi(\bar{U}_{T_m}),$$

where $\bar{U}_{T_m} = \sum_{i=1}^m U_{T_m,i}/m$ and $\Phi(x,y,z,t) = (x - yz)/(t - y^2)$. Therefore, the Delta method gives that in $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$:

$$\sqrt{m}\left(\tilde{S}^{X^1}_{T_m,m} - \tilde{S}^{X^1}_{T_m}\right) \underset{m\to\infty}{\overset{\mathcal{L}}{\longrightarrow}} \mathcal{N}\left(0, \nabla\Phi^T(\boldsymbol{\mu})\mathbf{\Sigma}\nabla\Phi(\boldsymbol{\mu})\right), \tag{8.44}$$

where $\boldsymbol{\mu} = \mathbb{E}_X[U_i] = \left(\mathrm{cov}_X(Y_i, Y_i^{X^1}), 0, 0, \mathrm{var}_X(Y_i)\right)$. We note that the assumption $\mathrm{var}_X(Y_i) \neq 0$ justifies the use of the Delta method. A simple calculation gives that:

$$\nabla\Phi^T(\boldsymbol{\mu})\mathbf{\Sigma}\nabla\Phi(\boldsymbol{\mu}) = \frac{\mathrm{var}_X\left((Y_i - \mathbb{E}_X[Y_i])\left(Y_i^{X^1} - \mathbb{E}_X[Y_i] - S^{X^1}Y_i + S^{X^1}\mathbb{E}_X[Y_i]\right)\right)}{(\mathrm{var}_X(Y_i))^2}, \tag{8.45}$$

with $S^{X^1} = \mathrm{cov}_X(Y_i, Y_i^{X^1})/\mathrm{var}_X(Y_i) = \mathrm{var}_X(\mathbb{E}_X[Y_i|X^1])/\mathrm{var}_X(Y_i)$.

Convergence of $\sqrt{m}\left(\tilde{S}^{X^1}_{T_m} - \tilde{S}^{X^1}\right)$ in $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$

Analogously to [Janon et al., 2012], we have the equality:

$$\tilde{S}^{X^1}_{T_m} - \tilde{S}^{X^1} = \frac{\mathrm{var}_X(\tilde{\delta}_{T_m,i})^{1/2} C_{\tilde{\delta}_{T_m,i}}}{\mathrm{var}_X(Y_i) + 2\mathrm{cov}_X(Y_i, \tilde{\delta}_{T_m,i}) + \mathrm{var}_X(\tilde{\delta}_{T_m,i})},$$

where $\tilde{\delta}_{T_m}(x) = g(T_m)^{-1} B_{T_m} \tilde{a}_{T_m}(x)$,

$$\begin{aligned} C_{\tilde{\delta}_{T_m,i}} = \; & 2\mathrm{var}_X(Y_i)^{1/2}(\mathrm{cor}_X(Y_i, \tilde{\delta}_{T_m,i}) - \mathrm{cor}_X(Y_i, Y_i^{X^1})\mathrm{cor}_X(Y_i, \tilde{\delta}_{T_m,i})) \\ & + \mathrm{var}_X(\tilde{\delta}_{T_m,i})^{1/2}(\mathrm{cor}_X(\tilde{\delta}_{T_m,i}, \tilde{\delta}^{X^1}_{T_m,i}) - \mathrm{cor}_X(Y_i, Y_i^{X^1})) \end{aligned}, \tag{8.46}$$

$\tilde{\delta}_{T_m,i} = \tilde{\delta}_{T_m,i}(X_i)$ and $\tilde{\delta}^{X^1}_{T_m,i} = \tilde{\delta}_{T_m,i}(\tilde{X}_i)$. The random variables $\tilde{\delta}_{T_m,i}$ and $\tilde{\delta}^{X^1}_{T_m,i}$ are defined on the product space $(\tilde{\Omega}_Z \times \Omega_X, \sigma(\tilde{\mathcal{F}}_Z \times \mathcal{F}_X), \tilde{\mathbb{P}}_Z \otimes \mathbb{P}_X)$ and $\tilde{S}^{X^1}$, $\tilde{\delta}_{T_m}(x)$ and $C_{\tilde{\delta}_{T_m,i}}$ are

defined on $(\tilde{\Omega}_Z, \tilde{\mathcal{F}}_Z, \tilde{\mathbb{P}}_Z)$. We still consider a fixed $\tilde{\omega}_Z \in \tilde{\Omega}_Z$. The assumption $\mathrm{var}_X(Y_i) \neq 0$ ensures that the denominator is not equal to zero and the convergences $\tilde{f}_{T_m}(x) \xrightarrow[m\to\infty]{L^6_\mu} \tilde{f}(x)$ and $\tilde{a}_{T_m}(x) \xrightarrow[m\to\infty]{L^2_\mu} 0$ give that $\sup_m C_{\tilde{\delta}_{T_m,i}} < \infty$. Furthermore, since $\tilde{a}_{T_m}(x) \xrightarrow[m\to\infty]{L^2_\mu} 0$ we have the following inequalities:

$$\mathrm{var}_X(\tilde{\delta}_{T_m,i}) \leq C\mathbb{E}_X[(B_{T_m} g(T_m)^{-1}\tilde{a}_{T_m}(X_i))^2] \leq C' g(T_m)^{-2} B^2_{T_m},$$

with C, C' positive constants.

Thanks to Slutsky's theorem, the convergence $mg(T_m)^{-2}B^2_{T_m} \overset{m\to\infty}{\longrightarrow} 0$ ensures the following asymptotic normality when $m \to \infty$ in $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$:

$$\sqrt{m}\left(\tilde{S}^{X^1}_{T_m,m} - \tilde{S}^{X^1}\right) \xrightarrow[m\to\infty]{\mathcal{L}} \mathcal{N}\left(0, \nabla\Phi^T(\boldsymbol{\mu})\boldsymbol{\Sigma}\nabla\Phi(\boldsymbol{\mu})\right). \tag{8.47}$$

The case $mB^2_{T_m} \overset{m\to\infty}{\longrightarrow} \infty$.

Let us suppose that $mB^2_{T_m} \overset{m\to\infty}{\longrightarrow} \infty$. We consider the convergences of

$$B^{-1}_{T_m}\left(\tilde{S}^{X^1}_{T_m,m} - \tilde{S}^{X^1}_{T_m}\right) \tag{8.48}$$

and

$$B^{-1}_{T_m}\left(\tilde{S}^{X^1}_{T_m} - \tilde{S}^{X^1}\right),$$

in $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$ with a fixed $\tilde{\omega}_Z \in \tilde{\Omega}_Z$ such that (8.28) holds. We have the following equality:

$$B^{-1}_{T_m}\left(\tilde{S}^{X^1}_{T_m,m} - \tilde{S}^{X^1}_{T_m}\right) = (\sqrt{m}B_{T_m})^{-1}\sqrt{m}\left(\tilde{S}^{X^1}_{T_m,m} - \tilde{S}^{X^1}_{T_m}\right).$$

The convergence $(\sqrt{m}B_{T_m})^{-1} \overset{m\to\infty}{\longrightarrow} 0$ and the convergence in (8.44) (which does not depend on the convergence of the ratio between $B^{-2}_{T_m}$ and $\sqrt{m}$) imply the following one:

$$B^{-1}_{T_m}\left(\tilde{S}^{X^1}_{T_m,m} - \tilde{S}^{X^1}_{T_m}\right) \overset{m\to\infty}{\longrightarrow} 0.$$

Finally, thanks to the inequality (8.25), there exists $C, C' > 0$ such that

$$\begin{aligned} B^{-1}_{T_m}\left(\tilde{S}^{X^1}_{T_m} - \tilde{S}^{X^1}\right) &= B^{-1}_{T_m}\frac{g(T_m)^{-1}B_{T_m}\mathrm{var}_X(\tilde{a}_{T_m}(X_i))^{1/2}C_{\tilde{\delta}_{T_m,i}}}{\mathrm{var}_X(Y_i) + 2\mathrm{cov}_X(Y_i, \tilde{\delta}_{T_m,i}) + \mathrm{var}_X(\tilde{\delta}_{T_m,i})} \\ &\geq Cg(T_m)^{-1}\frac{g(T_m)C_{\tilde{\delta}_{T_m,i}}}{\mathrm{var}_X(Y_i) + 2\mathrm{cov}_X(Y_i, \tilde{\delta}_{T_m,i}) + \mathrm{var}_X(\tilde{\delta}_{T_m,i})} \\ &\geq C'C_{\tilde{\delta}_{T_m,i}}. \end{aligned}$$

Therefore, if we have $C_{\tilde{\delta}_{T_m,i}} > 0$, the asymptotic normality is not reached and the estimator is biased. Regarding the expression of $C_{\tilde{\delta}_{T_m,i}}$ in (8.46) and assuming that $\mathrm{var}_X(Y_i) \neq 0$, $C_{\tilde{\delta}_{T_m,i}} = 0$ could happen if:

- $\mathrm{cor}_X(Y_i, Y_i^{X^1}) = 1$, i.e. all the variability of $\tilde{f}(x)$ is explained by the variable X^1.
- $\mathrm{var}_X(\tilde{\delta}_{T_m,i}) = 0$, i.e. the surrogate model error is null.

8.4.3 Convergence in the probability space $(\Omega_Z \times \Omega_X, \sigma(\mathcal{F}_Z \times \mathcal{F}_X), \mathbb{P}_Z \otimes \mathbb{P}_X)$.

We have proved that for almost every $\tilde{\omega}_Z \in \tilde{\Omega}_Z$:

If $mB_{T_m}^2 \overset{m\to\infty}{\longrightarrow} 0$, then

$$\forall I \in \mathbb{R},\ \mathbb{P}_X\left(\sqrt{m}\left(\tilde{S}_{T_m,m}^{X^1} - \tilde{S}^{X^1}\right) \in I\right) \overset{m\to\infty}{\longrightarrow} \int_I \tilde{g}(x)dx.$$

If $mB_{T_m}^2 \overset{m\to\infty}{\longrightarrow} \infty$, then

$$\exists C > 0 \,\text{s.t.}\ \mathbb{P}_X\left(B_{T_m}^{-1}\left(\tilde{S}_{T_m,m}^{X^1} - \tilde{S}^{X^1}\right) \geq C\right) \overset{m\to\infty}{\longrightarrow} 1,$$

where $\tilde{g}(x)$ is the probability density function of a random Gaussian vector of zero mean and covariance $\nabla\Phi^T(\boldsymbol{\mu})\boldsymbol{\Sigma}\nabla\Phi(\boldsymbol{\mu})$ (8.45). Therefore, in the probability space $(\tilde{\Omega}_Z \times \Omega_X, \sigma(\tilde{\mathcal{F}}_Z \times \mathcal{F}_X), \tilde{\mathbb{P}}_Z \otimes \mathbb{P}_X)$ we have

If $mB_{T_m}^2 \overset{m\to\infty}{\longrightarrow} 0$, then

$$\forall I \in \mathbb{R}, \forall \delta > 0,\ \tilde{\mathbb{P}}_Z\left(\left|\mathbb{P}_X\left(\sqrt{m}\left(\tilde{S}_{T_m,m}^{X^1} - \tilde{S}^{X^1}\right) \in I\right) - \int_I \tilde{g}(x)dx\right| > \delta\right) \overset{m\to\infty}{\longrightarrow} 0.$$

If $mB_{T_m}^2 \overset{m\to\infty}{\longrightarrow} \infty$, then

$$\forall \delta > 0, \exists C > 0 \,\text{s.t.}\ \tilde{\mathbb{P}}_Z\left(\left|\mathbb{P}_X\left(B_{T_m}^{-1}\left(\tilde{S}_{T_m,m}^{X^1} - \tilde{S}^{X^1}\right) \geq C\right) - 1\right| > \delta\right) \overset{m\to\infty}{\longrightarrow} 0.$$

and the equalities $(\tilde{f}_{T_m}(x), \tilde{a}_{T_m}(x), \tilde{b}_{T_m}(x)) \overset{\mathcal{L}}{=} (f(x), a_{T_m}(x), b_{T_m}(x))\ \forall T_m$ and $\tilde{f}(x) \overset{\mathcal{L}}{=} f(x)$ for all m give us in the probability space $(\Omega_Z \times \Omega_X, \sigma(\mathcal{F}_Z \times \mathcal{F}_X), \mathbb{P}_Z \otimes \mathbb{P}_X)$:

If $mB_{T_m}^2 \overset{m\to\infty}{\longrightarrow} 0$, then

$$\forall I \in \Omega_X, \forall \delta > 0,\ \mathbb{P}_Z\left(\left|\mathbb{P}_X\left(\sqrt{m}\left(S_{T_m,m}^{X^1} - S^{X^1}\right) \in I\right) - \int_I g(x)dx\right| > \delta\right) \overset{m\to\infty}{\longrightarrow} 0.$$

If $mB_{T_m}^2 \overset{m\to\infty}{\longrightarrow} \infty$, then

$$\forall \delta > 0, \exists C > 0 \,\text{s.t.}\ \mathbb{P}_Z\left(\left|\mathbb{P}_X\left(B_{T_m}^{-1}\left(S_{T_m,m}^{X^1} - S^{X^1}\right) \geq C\right) - 1\right| > \delta\right) \overset{m\to\infty}{\longrightarrow} 0,$$

where $g(x)$ is the probability density function of a random Gaussian vector of zero mean and variance

$$\frac{\mathrm{var}_X\left((f(X) - \mathbb{E}_X[f(X)])\left(f(\tilde{X}) - \mathbb{E}_X[f(X)] - S^{X^1}f(X) + S^{X^1}\mathbb{E}_X[f(X)]\right)\right)}{\left(\mathrm{var}_X(f(X))\right)^2}.$$

This completes the proof.

8.5 Examples of asymptotic normality for Sobol's index

According to the previous developments, the desired asymptotic normality is assessed under the assumption $mB^2_{T_m} \overset{m\to\infty}{\longrightarrow} 0$. In the remainder of this section, we present relations between T_m and m which lead the convergence $mB^2_{T_m} \overset{m\to\infty}{\longrightarrow} 0$ for some usual kernels.

8.5.1 Asymptotic normality with d-tensorised Matérn-ν kernels

We focus here on the d-tensorised Matérn-ν kernel with regularity parameter $\nu > 1/2$:

$$k(x,\tilde{x}) = \prod_{i=1}^{d} \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}|x^i - \tilde{x}^i|}{\theta_i} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}|x^i - \tilde{x}^i|}{\theta_i} \right),$$

where K_ν is the modified Bessel function [Abramowitz and Stegun, 1965]. The eigenvalues of this kernel satisfy the following asymptotic behavior [Pusev, 2011]:

$$\lambda_p = \phi(p), \quad p \gg 1,$$

where $\phi(p) = \left(\log(1+p)^{2(d-1)(\nu+1/2)}\right) p^{-2(\nu+1/2)} \left(1 + \mathrm{O}(1/p)\right)$. Therefore, for $T_m \gg 1$:

$$B^2_{T_m} \approx \log(T_m/\sigma^2_\varepsilon)^{d-1} \left(\frac{\sigma^2_\varepsilon}{T_m} \right)^{1-1/2(\nu+1/2)}.$$

Section 8.3 suggests that the asymptotic normality of the Sobol's index estimator is assessed when:

$$mB^2_{T_m} \overset{m\to\infty}{\longrightarrow} 0.$$

Let us consider the following that T_m is such that:

$$\log(T_m/\sigma^2_\varepsilon)^{d-1} \left(\frac{\sigma^2_\varepsilon}{T_m} \right)^{1-1/2(\nu+1/2)} = 1/m. \tag{8.49}$$

It corresponds to the critical point $mB^2_{T_m} \approx 1$. In this case, the error originates both from the meta-model approximation error and the Monte-Carlo estimation error. Equation (8.49) leads to the following critical budget:

$$\frac{T_m}{\sigma^2_\varepsilon} = \sigma^2_\varepsilon m^{1/(1-1/2(\nu+1/2))} \log(m)^{(d-1)}, \tag{8.50}$$

and, the asymptotic normality is assessed for:

$$\frac{T_m}{\sigma^2_\varepsilon} = \sigma^2_\varepsilon m^{1/(1-1/2(\nu+1/2))+\alpha} \log(m)^{(d-1)}, \forall \alpha > 0. \tag{8.51}$$

In practice, we want to minimize the budget allocated to the simulator and thus consider the case α tends to zero. As a consequence, for applications we will consider the allocation of the critical point (8.50).

8.5.2 Asymptotic normality for d-dimensional Gaussian kernels

Let us consider the d-dimensional Gaussian kernel:

$$k(x,\tilde{x}) = \exp\left(-\frac{1}{2}\sum_{i=1}^{d}\frac{(x^i-\tilde{x}^i)^2}{\theta_i^2}\right). \tag{8.52}$$

Thanks to [Todor, 2006], we have the following upper bound for the eigenvalues:

$$\lambda_p \leq c'\exp\left(-cp^{1/d}\right), \tag{8.53}$$

with c and c' constants. From this inequality, we can deduce that there exists $C > 0$ such that:

$$B_{T_m}^2 \approx C\log(T_m/\sigma_\varepsilon^2)^d\left(\frac{\sigma_\varepsilon^2}{T_m}\right).$$

Therefore, the critical budget corresponding to the critical point $mB_{T_m}^2 \approx 1$ is given by

$$T_m/\sigma_\varepsilon^2 = m\log(m)^d \tag{8.54}$$

and the asymptotic normality for the Sobol index estimator is assessed with:

$$T_m/\sigma_\varepsilon^2 = m^{1+\alpha}\log(m)^d, \ \forall\alpha > 0. \tag{8.55}$$

We note that the condition is only sufficient since we have an inequality in (8.53).

8.5.3 Asymptotic normality for d-dimensional Gaussian kernels with a Gaussian measure $\mu(x)$

Let us consider a Gaussian measure $\mu \sim \mathcal{N}(0,\sigma_\mu^2\mathbf{I})$ in dimension d and the Gaussian kernel (8.52). As presented in [Zhu et al., 1998], we have analytical expressions for the eigenvalues and eigenfunctions of $k(x,\tilde{x})$:

$$\lambda_p = \prod_{i=1}^{d}\sqrt{\frac{2a}{A_i}}B_i^p,$$

$$\phi_p(x) = \exp\left(-\sum_{i=1}^{d}(c_i-a)(x^i)^2\right)\prod_{i=1}^{d}H_p(\sqrt{2c_i}x^i),$$

where $H_p(x) = (-1)^p\exp(x^2)\frac{d^p}{dx^p}\exp(-x^2)$ is the pth order Hermite polynomial (see [Gradshteyn et al., 2007]), $a = 1/(2\sigma_\mu)^2$, $b_i = 1/(2\theta_i^2)$ and

$$c_i = \sqrt{a^2+2ab_i}, \quad A_i = a+b_i+c_i, \quad B_i = b_i/A_i.$$

Therefore, the eigenvalues satisfy the following asymptotic behavior

$$\lambda_p \propto \exp(-p\xi_d), \tag{8.56}$$

where $\xi_d = \sum_{i=1}^{d}\log(1/B_i)$. For $T_m \gg 1$, we have:

$$B_{T_m}^2 \approx \left(\sigma_\varepsilon^2/T_m\right)\log\left(T_m/\sigma_\varepsilon^2\right)/\xi_d. \tag{8.57}$$

Let us consider the critical point $B_{T_m}^2 = 1/m$. Then, the critical budget is given by

$$\frac{T_m}{\sigma_\varepsilon^2} = \xi_d m \log(m)$$

and the asymptotic normality is assessed for:

$$\frac{T_m}{\sigma_\varepsilon^2} = \xi_d m^{1+\alpha} \log(m), \ \forall \alpha > 0. \tag{8.58}$$

8.6 Numerical illustration

The purpose of this section is to perform a global sensitivity analysis of a stochastic code solving the following heat equation:

$$\frac{\partial u}{\partial t}(x,t) - \frac{1}{2}\Delta u(x,t) = 0, \tag{8.59}$$

with $x \in \mathbb{R}^d$ and $u(x,0) = g(x) = \exp(-\sum_{i=1}^d x_i^2/(2\sigma_{g,i}^2))$. The function $u(x,t)$ has the following probabilistic representation:

$$u(x,t) = \mathbb{E}_{W_t}[g(x+W_t)], \tag{8.60}$$

where W_t is the 1-dimensional Brownian motion. We evaluate the function $u(x,t)$ through the following stochastic code:

$$u_r^{\text{code}}(x,t) = \frac{1}{r}\sum_{i=1}^r \left(\frac{1}{s}\sum_{j=1}^s g(x+W_{t,i,j})\right), \tag{8.61}$$

where the number of replications r tunes the precision of the output, $s = 30$ and $(W_{t,i,j})_{\substack{i=1,\dots,r\\ j=1,\dots,s}}$ are sampled from a Gaussian random variable of mean zero and variance t.

We note that there is a closed form expression for the solution of the considered heat equation, that will allow us to compute exactly the Sobol indices and to assess the quality of our estimate:

$$u(x,t) = \prod_{i=1}^d \left(\frac{\sigma_{g,i}^2}{\sigma_{g,i}^2+t}\right)^{1/2} \exp\left(-\frac{x_i^2}{2(\sigma_{g,i}^2+t)}\right). \tag{8.62}$$

8.6.1 Exact Sobol indices

Let us consider that x is a random variable X defined on $(\Omega_X, \mathcal{F}_X, \mathbb{P}_X)$ such that $X \sim \mathcal{N}\left(0, \sigma_\mu^2 \mathbf{I}\right)$. We are interested for the application in the first order Sobol indices, i.e. the contribution of $(X^j)_{j=1,\dots,d}$. By straightforward calculations it can be shown that:

$$S^{X^j} = \frac{V^{X^j}}{V} = \frac{\text{var}_X(\mathbb{E}_X[u(X,t)|X^j])}{\text{var}_X(u(X,t))} = \frac{B_j - 1}{\left(\prod_{i=1}^d B_i\right) - 1}, \tag{8.63}$$

where X^j is the jth component of the random vector X with $j = 1, \ldots, d$ and

$$B_j = \sigma_\mu \left(\frac{2}{t} - \frac{2}{t^2} \left(\frac{1}{t} + \frac{1}{\sigma_{g,i}^2} \right)^{-1} + \frac{1}{\sigma_\mu^2} \right)^{-\frac{1}{2}} \left(\frac{1}{t} + \frac{1}{\sigma_\mu^2} - \frac{1}{t^2} \left(\frac{1}{t} + \frac{1}{\sigma_{g,i}^2} \right)^{-1} \right).$$

Therefore, the importance measure of the jth input is directly linked with the dispersion parameter $\sigma_{g,i}^2$ of the function $g(x)$. Furthermore, when t tends to the infinity, the response $u(x,t)$ tends to zero as the variance of the main effect. In this section, we consider the response at $t = 1$.

8.6.2 Model selection

Let us consider a Gaussian process of covariance $k_u(x, \tilde{x})$ and mean m_u to surrogate $u(x,t)$ at $t = 1$. We consider the predictive mean and variance presented in equations (8.1) and (8.2). As the response $u(x,t)$ is smooth, we choose a squared exponential covariance kernel:

$$k_u(x, \tilde{x}) = \sigma^2 \exp\left(-\frac{1}{2} \sum_{i=1}^{d} \frac{(x^i - \tilde{x}^i)^2}{\theta_i^2} \right).$$

Furthermore, as $u(x,t)$ tends to zero when x tends to the infinity, we consider that $m_u = 0$. Indeed, we want that the model tends to zero when we move away from the design points.

The experimental design set $\mathbf{D}$ is composed of $n = 3000$ training points x_i^{train} sampled from the multivariate normal distribution $\mathcal{N}\left(0, \sigma_\mu^2 \mathbf{I}\right)$ with $\sigma_\mu = 2$ and $d = 5$. Furthermore, the initial budget is $T_0 = 3000$. It corresponds to a unique repetition $r_0 = 1$ at each point of $\mathbf{D}$. The n observations of $u_{r_0}^{\text{code}}(x, 1)$ at points in $\mathbf{D}$ are denoted by $\mathbf{u}^n$.

The hyper-parameters σ^2, θ and σ_ε^2 are estimated by maximizing the marginal Likelihood:

$$-\frac{1}{2} \left(\mathbf{u}^n\right)' \left(\sigma^2 \mathbf{K} + \sigma_\varepsilon \mathbf{I}\right)^{-1} \mathbf{u}^n - \frac{1}{2} \det\left(\sigma^2 \mathbf{K} + \sigma_\varepsilon \mathbf{I}\right),$$

where $\mathbf{K} = [k_u(x_i, x_j)]_{i,j=1,\ldots,n}$. To solve the maximization problem, we have first randomly generated a set of 1,000 parameters $(\sigma^2, \theta, \sigma_\varepsilon)$ on the domain $(0,10) \times (0,2)^d \times (0,1)$ and we have started a quasi-Newton based maximization from the 10 best parameters using the BFGS method. We obtain the following parameter estimations.

- $\hat{\theta} = \begin{pmatrix} 1.01 & 1.02 & 1.03 & 1.00 & 1.07 \end{pmatrix}$
- $\hat{\sigma}^2 = 1.46$
- $\hat{\sigma}_\varepsilon^2 = 6.74.10^{-2}$

Furthermore, the dispersion term of $g(x)$ are set to:

- $(\sigma_{g,i}^2)_{i=1,\ldots,d} = (5, 3, 2, 1, 1)$

8.6.3 Convergence of IMSE_T

As presented in Subsection 8.2.2 and Section 8.3, the asymptotic normality of the Sobol index estimator is closely related to the convergence of the generalization error IMSE_T (8.12). Therefore, in order to effectively estimate the confidence intervals of the estimators, we have to characterize this convergence. Especially, we have to take into account the initial budget used to select the model. The value of IMSE_{T_0} where T_0 corresponds to the initial budget allocated to $\mathbf{D}$ is estimated to $\text{IMSE}_{T_0} = 6.06.10^{-1}$. According to (8.57), we have the following convergence rate for IMSE_T with respect to T:

$$\text{IMSE}_T \sim \left(\sigma_\varepsilon^2/T\right) \log\left(T/\sigma_\varepsilon^2\right)/\xi_d.$$

Therefore, from an initial budget T_0 we expect that IMSE_T as a function of T decays as:

$$\text{IMSE}_T = \text{IMSE}_{T_0} \frac{T_0 \log\left(T/\sigma_\varepsilon^2\right)}{T \log\left(T_0/\sigma_\varepsilon^2\right)}.$$

The critical ratio $mB_T^2 = 1$ presented in Section 8.5 leads to the following budget:

$$T = \frac{m}{C} \log\left(\frac{m}{C\sigma_\varepsilon^2}\right), \tag{8.64}$$

with $C = \log\left(T_0/\sigma_\varepsilon^2\right)/(T_0 \text{IMSE}_{T_0})$. We consider this ratio since there is numerically no difference between $T = \frac{m}{C}\log\left(\frac{m}{C\sigma_\varepsilon^2}\right)$ and $T = \frac{m^{1+\varepsilon}}{C}\log\left(\frac{m}{C\sigma_\varepsilon^2}\right)$ for a very small value of ε (e.g. 10^{10}).

8.6.4 Confidence intervals for the Sobol index estimations

According to Theorem 8.1, if T follows the relation in (8.64), the Sobol index estimator presented in Subsection 8.3.1 is asymptotically distributed with respect to a Gaussian random variable centered on the true index and with variance given in (8.21). We use this property to build 90% confidence intervals on the estimations of $(S^j)_{j=1,\dots,d}$ (8.63). The exact values of the Sobol indices (8.63) are given by:

$$(S^j)_{j=1,\dots,d} = (0.052, 0.088, 0.124, 0.194, 0.194).$$

Remember that m represents the number of particles for the Monte-Carlo integrations and T is the budget used to construct the surrogate model $\hat{z}_T(x)$. In order to illustrate the relevance of (8.64), we consider the following equation:

$$T = \sigma_\varepsilon^2 \frac{m^\alpha}{C} \log\left(\frac{m}{C}\right),$$

with different values of α - the right value being $\alpha = 1$ - and different values of m. For each combination (α, m), we estimate the Sobol indices with the estimator (8.18) and from 500 different Monte-Carlo samples $(x_i^{\text{MC}})_{i=1,\dots,m}$. For each sample we evaluate the 90% confidence intervals thanks to (8.21) and we check if the estimations are covered or not. The result of the procedure is presented in Table 8.1.

m	α	S^1	S^2	S^3	S^4	S^5
1,000	0.8	88.00	86.20	87.60	88.20	86.40
1,000	0.9	89.00	91.80	89.60	86.20	86.00
1,000	1.0	88.40	87.00	89.40	87.60	90.80
1,000	1.1	88.00	89.40	88.80	87.00	88.60
1,000	1.2	90.00	91.00	86.60	88.80	89.00
3,000	0.8	88.00	87.60	86.60	87.80	87.20
3,000	0.9	89.80	87.80	87.40	88.60	88.00
3,000	1.0	89.40	90.40	89.20	89.40	89.60
3,000	1.1	90.40	90.60	91.00	91.60	90.80
3,000	1.2	92.00	91.80	92.00	91.40	91.40
5,000	0.8	87.60	86.20	87.40	88.20	86.40
5,000	1.0	89.20	89.40	90.80	89.80	89.60
5,000	1.2	92.00	91.40	92.80	90.60	92.20

Table 8.1: Coverage rates for $(S^j)_{j=1,\dots,d}$ in percentage. The confidence intervals are built from the variance presented in (8.21) in Theorem 8.1. The theoretical rates is 90% and the estimations is performed from 500 different Monte-Carlo samples.

We see in Table 8.1 that the asymptotic behavior is not reached for $m = 1,000$ Monte-Carlo particles since the coverage is globally too low in this case for every α. Furthermore, for $m = 3,000$ and $m = 5,000$, we see that the coverage is globally better for $\alpha = 1$ than for the other values. Indeed, the covering rate is underestimated for $\alpha < 1$ and often overestimated for $\alpha > 1$ whereas it is always around 90% for $\alpha = 1$.

Furthermore, the confidence intervals seem to be well evaluated either for large values of S^j with S^4 and S^5, for intermediate values of S^j with S^3 or for small values of S^j with S^2 and S^1. Therefore, this example emphasize the relevance of the asymptotic normality for the Sobol index estimators presented in Theorem 8.1.

8.7 Conclusion

This chapter focuses on the estimation of the Sobol indices to perform global sensitivity analysis for stochastic simulators. We suggest an index estimator which combines a Monte-Carlo scheme to estimate the integrals involved in the index definition and a Gaussian process regression to surrogate the stochastic simulator. The surrogate model is necessary since the Monte-Carlo integrations require an important number of simulations.

In a stochastic simulator framework, for a fixed computational budget the observation noise variance is inversely proportional to the number of simulations. In this chapter, we consider the special case of a large number of observations with an important uncertainty on the output. This choice allows us to consider an idealized version of the regression problem from which we can define a surrogate model which is tractable for our purpose.

In particular we aim to build confidence intervals for the index estimator taking into account both the uncertainty due to the Monte-Carlo integrations and the one due to the surrogate modeling. To handle this point, we present a theorem providing sufficient conditions to ensure the asymptotic normality of the suggested estimator. The proof of the theorem is the main point of this chapter. It gives a closed form expression for the variance of the asymptotic distribution of the estimator. From it we can easily estimate the desired confidence intervals. Furthermore, a strength of the suggested theorem is that it gives the relation between the number of particles for the Monte-Carlo integrations and the computational budget allocated to the surrogate model so that they have the same contribution on the error of the Sobol index estimations.

Conclusion and perspectives

The general framework of the thesis is the Gaussian process regression for computer experiments. The objective is to build a surrogate model - also called meta-model - of a computer code in order to have a fast approximation of its input/output relation. From this approximation, one can perform uncertainty quantification, optimization, sensitivity analysis, quantile estimation... For practical applications, using a surrogate model is often necessary since the complex computer codes are generally time-consuming and the cited analyses require a large number of simulations.

However, surrogate models require careful implementations and appropriate validation diagnostics. Furthermore, the construction of a meta-model often depends on the conception objective. As an example, for an optimization purpose, we will concentrate the observations at locations where the improvement expectation is important. On the contrary, for a prediction purpose, the observations are generally spread over all the input parameter space. Another important point is that a meta-model is valid only over the space covered by the experimental design set. In particular, it is not appropriate to perform extrapolations.

In this manuscript, we are interested in a first part in simulators which have coarser and computationally cheaper versions. The aim is to improve the approximation of a computer code output using these coarse versions. To surrogate such simulators we make the choice to use an extension of Gaussian process regression for multivariate outputs. Furthermore, we also focus on a particular structure defining the relation between the different code levels. This choice has two main strengths. First, the Gaussian process assumption allows for having an information about the model accuracy and provides a basis for statistical inference. Second, the suggested structure allows for easily handling with the surrogate model and thus for deriving interesting tools for practical applications. The numerous applications addressed in the manuscript highlight the performance of the suggested approach.

Nevertheless, there is no reason that the Gaussian process assumption is relevant. Therefore, it is worth exploring other meta-models such as polynomial models, neural networks, support vector machine... Especially since the Gaussian assumption is not appropriate for some types of computer outputs. It is not well-suited for highly non-linear responses and it is hard to use for non-stationary outputs. These two examples are of importance since they are common for simulators dealing with complex physics.

Furthermore, the suggested structure between the code levels (i.e. the autoregressive relation) is simple and cannot be relevant for all applications. A first case which can cause a failure is the one when the bias between two codes is as difficult to learn as the complex code. In that case, a multi-fidelity analysis as presented in this manuscript is not significant. A second case which can cause a failure is when there is a transformation on the input parameters between two code levels. Of course, more complex relations between two code levels can be imagined. However, their estimation could require a large number of observations from the complex computer code. We must keep in mind that a multi-fidelity analysis is worth only if it reduces this number of observations. Examples of more complicated relations can be found in the field of fluid dynamics with turbulent flow where the coarse codes can be linear simplifications of the complex one. In this case, log-transformations are sometimes used to predict the output of the complex code from the ones of the simplified codes.

Another class of problems for which the suggested approach is not relevant is the one when we do not know which code level is the most accurate. Indeed, it is common that for a given physical system, several simulators can be used to model it and no classification can be made between them. A promising approach to deal with these problems is the multi-armed bandit method. The multi-armed bandit problem was originally introduced by [Robbins, 1952]. A multi-armed bandit is a bandit machine with more than one lever. Moreover, each lever has its own expected profit. The purpose of the problem is to find the most rewarding levers through repeated trials. Several strategies has been suggested to solve this problem and it has been intensively investigated in the last decades (e.g the ε-greedy strategy [Watkins, 1989], [Auer et al., 2002a], [Mannor and Tsitsiklis, 2004], the SoftMax strategy [Wyatt, 1998], [Auer et al., 2002b], the interval estimation strategy [Kaelbling, 1993], [Meuleau and Bourgine, 1999] and the POKER strategy [Vermorel and Mohri, 2005]). These strategies try to minimize the so-called "regret" defined as the difference between the reward sum associated to an optimal strategy and the sum of the collected rewards. Furthermore, simulators can also be effective at different locations of the input parameter space. In that case, mixture of experts methods can be used (see [Jordan and Jacobs, 1994], [Waterhouse et al., 1996], [Ueda and Ghahramani, 2002] and [Bishop et al., 2006]).

Naturally, from the model we have suggested, many investigations can be led. We propose in this manuscript how to use it for performing sensitivity analysis and for improving the prediction capability of the model. Of course, we could have studied the use of this model for global optimization ([Jones et al., 1998], [Mockus, 1994], [Williams et al., 2000], [Mockus, 2002], [Huang et al., 2006], [Villemonteix et al., 2009], [Vazquez and Bect, 2010], [Marzat et al., 2012], [Picheny et al., 2012] and [Janusevskis and Le Riche, 2013]), reliability-based design optimization ([Bichon et al., 2008], [Valdebenito and Schuëller, 2010] and [Huang and Chan, 2010]), estimation of probabilities of failure ([Oakley, 2004], [Picheny et al., 2010], [Dubourg et al., 2011], [Bect et al., 2012], [Li et al., 2012] and [Picard and Williams, 2013]) or model calibration ([Kennedy and O'Hagan, 2001], [Higdon et al., 2004], [Van Oijen et al., 2005], [McFarland et al., 2008], [Higdon et al., 2008] and [Wilkinson, 2010]) which are commonly performed with Gaussian process models.

In the second part of the manuscript, we address more theoretical questions. In particular,

we deal with the rate of convergence of the Gaussian process regression model when the code output are tainted by measurement noise. We study the case where the observation noise variance is proportional to the number of observations and we focus on the asymptotics corresponding to a large number of observations. Despite the fact that this assumption is relevant for stochastic simulators it is not the case for all noisy responses. Furthermore, it is not obvious that the asymptotics of a large number of observations is the more relevant in practice for stochastic simulators. Indeed, we can easily imagine that the best choice is an intermediate between few accurate observations and lot of inaccurate observations. Moreover, an interesting result would be to obtain the rate of convergence of the Gaussian process regression for a fixed noise variance or when the noise variance equals zero, especially since these cases most often occur in practical applications.

From this discussion, we see that many researches and improvements can be conducted both for theoretical and practical perspectives. For theoretical ones, it would be interesting to extend the asymptotic results on the Gaussian process regression to more general cases. For practical ones, it would be nice to develop methods minimizing the importance of the Gaussian assumption. Finally, for multi-fidelity codes, many investigations can be led to deal with the cases where the autoregressive assumption fail.

Part IV

Appendix

Appendix A

Chapter 3 supplementary materials

A.1 A Markovian property for covariance structure

An AR(1) autoregressive model

We present in this section the proof that the Markovian covariance structure presented in Chapter 3 is equivalent to the AR(1) autoregressive model. The proof comes from the technical report [O'Hagan, 1998]. Let us suppose that we want to predict $f(x,t)$, $(x,t) \in \mathbb{R}^d \times \mathbb{R}^+$ and that we have already observed $f(x,t')$ with $t' \neq t$. It is natural to assume that no more information about $f(x,t)$ can be learn from $f(x',t')$ for $x \neq x'$. It is a kind of Markov assumption which states that $f(x,t)$ depend on $\{f(x',t'), \quad x' \in \mathbb{R}^d\}$, for given x, t and t' only through the nearest observation $f(x,t')$.

We denote by $\mathcal{M}(\mathbb{R}^d; t, t')$ this property which can formally be written with the following form:

$$\operatorname{cov}\big(f(x,t), f(x',t')\big) = \frac{\operatorname{cov}\left(f(x,t), f(x,t')\right) \operatorname{cov}\left(f(x,t'), f(x',t')\right)}{\operatorname{cov}\left(f(x,t'), f(x,t')\right)}. \tag{A.1}$$

We obtain $\operatorname{cov}\left(f(x,t), f(x',t') | f(x,t')\right) = 0$. We note that (A.1) implies a linear independence, therefore there is no equivalence between (A.1) and $\mathcal{M}(\mathbb{R}^d; t, t')$. Nevertheless in a Gaussian framework there is equivalence between independence and linear independence. The AR(1) formula is obtained thanks to the following theorem.

Theorem A.1. *The Markovian property $\mathcal{M}(\mathbb{R}^d; t, t')$ is satisfied for given t and t' if and only if there exists $r(x)$, such that $\forall x \in \mathbb{R}^d$, we have $\{f(x,t) - r(x)f(x,t'), x \in \mathbb{R}^d\}$ linearly independent of $\{f(x,t'), x \in \mathbb{R}^d\}$.*

Proof. Let us consider t and t' fixed, $e(x) = f(x,t) - r(x)f(x,t')$ and $g(x) = f(x,t')$.

$\Leftarrow$ **Sufficiency**.

Let us consider $e(x)$ and $g(x')$ uncorrelated for all x and x' in $\mathbb{R}^d$. We denote by

$\text{cov}\,(e(x), e(x')) = c_e(x, x')$ and $\text{cov}\,(g(x), g(x')) = c_g(x, x')$. For $t \neq t'$, we have:

$$\begin{aligned}\text{cov}\,(f(x,t), f(x',t')) &= \text{cov}\,(e(x) + r(x)g(x), g(x')) \\ &= r(x)c_g(x, x')\end{aligned}.$$

Furthermore, we have:

$$\text{cov}\left(f(x,t'), f(x',t')\right) = c_g(x, x'),$$

$$\text{cov}\left(f(x,t'), f(x,t')\right) = c_g(x, x)$$

and:

$$\text{cov}\left(f(x,t), f(x,t')\right) = r(x)c_g(x, x),$$

where:

$$\begin{aligned}\text{cov}\,(f(x,t), f(x',t'))\,\text{cov}\,(f(x,t'), f(x,t')) &= r(x)c_g(x,x')c_g(x,x) \\ &= \text{cov}\,(f(x,t), f(x,t'))\,\text{cov}\,(f(x,t'), f(x',t'))\end{aligned}.$$

We obtain Equation (A.1). The Markovian property $\mathcal{M}(\mathbb{R}^d; t, t')$ is thus satisfied for $t \neq t'$. For $t = t'$ the property is obvious.

$\Rightarrow$ **Necessity**.

Let us suppose that we have:

$$\text{cov}\left(f(x,t), f(x',t')\right) = \frac{\text{cov}\,(f(x,t), f(x,t'))\,\text{cov}\,(f(x,t'), f(x',t'))}{\text{cov}\,(f(x,t'), f(x,t'))}.$$

We denote by:

$$r(x) = \frac{\text{cov}\,(f(x,t), f(x,t'))}{\text{cov}\,(f(x,t'), f(x,t'))}.$$

We have:

$$\begin{aligned}\text{cov}\left(e(x), g(x')\right) &= \text{cov}\left(f(x,t) - r(x)f(x,t'), f(x',t')\right) \\ &= \text{cov}\left(f(x,t), f(x',t')\right) - r(x)\text{cov}\left(f(x,t'), f(x',t')\right) \\ &= \text{cov}\left(f(x,t), f(x',t')\right) - \frac{\text{cov}\,(f(x,t), f(x,t'))}{\text{cov}\,(f(x,t'), f(x,t'))}\text{cov}\left(f(x,t'), f(x',t')\right) \\ &= 0.\end{aligned}$$

□

A.2 The case of ρ depending on x

A.2.1 Building a model with s levels of code

Let us consider s levels of code, if we note $\boldsymbol{\beta} = (\boldsymbol{\beta}_1', \ldots, \boldsymbol{\beta}_s')'$, $\boldsymbol{\beta}_\rho = (\boldsymbol{\beta}_{\rho_1}', \ldots, \boldsymbol{\beta}_{\rho_{s-1}}')'$, $\sigma^2 = (\sigma_1^2, \ldots, \sigma_s^2)$ and $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_s)$, we have $[Z_s(x)|\mathbf{Z} = \mathbf{z}, \boldsymbol{\beta}, \boldsymbol{\beta}_\rho, \sigma^2, \boldsymbol{\theta}] \sim \mathcal{N}\left(m_{Z_s}(x), s^2_{Z_s}(x)\right)$ where $m_{Z_s}(x)$ and $s^2_{Z_s}(x)$ are defined in equations (3.27) and (3.28). Let us define the notation $\bigodot_{i=k}^{l} \mathbf{A}_i = \mathbf{A}_k \odot \cdots \odot \mathbf{A}_l$ where $\odot$ represents the matrix element-by-element product.

Furthermore, let us denote by $\rho_t = \rho_t(\mathbf{D}_t)$ the vector containing the values of $\rho_t(x)$, $x \in \mathbf{D}_t$. The s diagonal blocks of $\mathbf{V}_s$ (3.29) of size $n_t \times n_t$ are defined by:

$$\mathbf{V}^{(t,t)} = \sigma_t^2 R_t(\mathbf{D}_t) + \sigma_{t-1}^2 \left(\rho_{t-1}(\mathbf{D}_t)\rho'_{t-1}(\mathbf{D}_t)\right) \odot R_{t-1}(\mathbf{D}_t) + \cdots + \sigma_1^2 \left(\bigodot_{i=1}^{t-1} \rho_i(\mathbf{D}_t)\rho'_i(\mathbf{D}_t)\right) \odot R_1(\mathbf{D}_t),$$

and the off-diagonal blocks of size $n_t \times n_{t'}$ are given by:

$$\mathbf{V}^{(t,t')} = \left(\mathbf{1}_{n_t} \left(\bigodot_{i=t}^{t'-1} \rho_i(\mathbf{D}_{t'})\right)'\right) \odot \mathbf{V}^{(t,t)}(\mathbf{D}_t, \mathbf{D}_{t'}) \qquad 1 \leq t < t' \leq s,$$

where $\mathbf{1}_n$ denotes a vector of size n where all components are 1. The vector $\mathbf{k}_s(x)$ in equations (3.27) and (3.28) is such that $\mathbf{k}_s(x) = (k_1^*(x, \mathbf{D}_1)', \ldots, k_s^*(x, \mathbf{D}_s)')'$, where:

$$k_t^*(x, \mathbf{D}_t)' = \rho'_{t-1}(\mathbf{D}_t) \odot k_{t-1}^*(x, \mathbf{D}_t)' + \left(\prod_{i=t}^{s-1} \rho_i(x)\right) \sigma_t^2 R_t(x, \mathbf{D}_t),$$

where $1 < t \leq s$, $\left(\prod_{i=s}^{s-1} \rho_i(x)\right) = 1$ and $k_1^*(x, \mathbf{D}_1)' = \left(\prod_{i=1}^{s-1} \rho_i(x)\right) \sigma_1^2 R_1(x, \mathbf{D}_1)$. Furthermore, the matrix $\mathbf{H}_s$ in equations 3.33 can be written as:

$$\mathbf{H}_s = \begin{pmatrix} & & \vdots & & \ddots & \\ \left(\left(\bigodot_{i=1}^{j-1} \rho_i(\mathbf{D}_j)\right) \mathbf{1}'_{p_1}\right) \odot F_1(\mathbf{D}_j) & \left(\left(\bigodot_{i=2}^{j-1} \rho_i(\mathbf{D}_j) \mathbf{1}'_{p_2}\right)\right) \odot F_2(\mathbf{D}_j) & \ldots & F_j(\mathbf{D}_j) & 0 \\ & & \vdots & & & \ddots \end{pmatrix}.$$

A.2.2 Bayesian estimation of parameters for s levels of code

We can extend the Bayesian estimation of the parameters to the case of ρ depending on x. Note that we do not assume the independence of $\boldsymbol{\beta}_t$ and $\boldsymbol{\beta}_{\rho_{t-1}}$. We have:

$$[(\boldsymbol{\beta}_{\rho_{t-1}}, \boldsymbol{\beta}_t) | \mathbf{z}_t, \mathbf{z}_{t-1}, \boldsymbol{\theta}_t, \sigma_t^2] \sim \mathcal{N}\left(\left(\mathbf{H}'_t R_t(\mathbf{D}_t)^{-1} \mathbf{H}_t\right)^{-1} \mathbf{H}'_t R_t(\mathbf{D}_t)^{-1} \mathbf{z}_t, \sigma_t^2 \left(\mathbf{H}'_t R_t(\mathbf{D}_t)^{-1} \mathbf{H}_t\right)^{-1}\right),$$

where $\mathbf{H}_t = [F_{\rho_{t-1}}(\mathbf{D}_t) \odot (z_{t-1}(\mathbf{D}_t)\mathbf{1}'_{q_{t-1}}) \quad F_t(\mathbf{D}_t)]$. Furthermore, we have:

$$[\sigma_t^2 | z_t, z_{t-1}, \boldsymbol{\theta}_t] \sim \mathcal{IG}(\alpha_t, \frac{Q_t}{2}),$$

where

$$\alpha_t = \frac{n_t - p_t - q_{t-1}}{2},$$

$$Q_t = (\mathbf{z}_t - \mathbf{H}_t \hat{\boldsymbol{\lambda}}_t)' R_t(\mathbf{D}_t)^{-1} (\mathbf{z}_t - \mathbf{H}_t \hat{\boldsymbol{\lambda}}_t),$$

$$\hat{\boldsymbol{\lambda}}_t = \mathbb{E}\left[(\boldsymbol{\beta}_{\rho_{t-1}}, \boldsymbol{\beta}_t) | z_t, z_{t-1}, \boldsymbol{\theta}_t, \sigma_t^2\right].$$

The REML estimator of σ_t^2 is $\hat{\sigma}_t^2 = Q_t / 2\alpha_t$ and we can estimate $\boldsymbol{\theta}_t$ by minimizing the expression:

$$\log(|\det(R_t(\mathbf{D}_t))|) + (n_t - p_t - q_{t-1}) \log(\hat{\sigma}_t^2).$$

A.2.3 Some important results about the covariance matrix $\mathbf{V}_s$

By sorting the experimental design sets as in Subsection 3.6.2, it can be shown that $\forall t = 2, \dots, s$ the inverse of the matrix $\mathbf{V}_s$ has the form:

$$\mathbf{V}_s^{-1} = \begin{pmatrix} \mathbf{V}_{s-1}^{-1} + \begin{pmatrix} 0 & 0 \\ 0 & \frac{(\rho_{s-1}(\mathbf{D}_s)\rho'_{s-1}(\mathbf{D}_s))\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2} \end{pmatrix} & -\begin{pmatrix} 0 \\ \frac{(\rho_{s-1}(\mathbf{D}_s)\mathbf{1}'_{n_s})\odot R_s(\mathbf{D}_s)^{-1}}{\sigma_s^2} \end{pmatrix} \\ -\begin{pmatrix} 0 & \frac{(\mathbf{1}_{n_s}\rho'_{s-1}(\mathbf{D}_s))\odot R_s^{-1}(\mathbf{D}_s)}{\sigma_s^2} \end{pmatrix} & \frac{R_s^{-1}(\mathbf{D}_s)}{\sigma_s^2} \end{pmatrix},$$

with $\mathbf{V}_1^{-1} = \frac{R_1^{-1}(\mathbf{D}_1)}{\sigma_1^2}$, $\mathbf{V}_{s-1}^{-1}$ an $(\sum_{i=1}^{s-1} n_i \times \sum_{i=1}^{s-1} n_i)$ matrix and $R_s(\mathbf{D}_1)^{-1}$ an $(n_s \times n_s)$ matrix. It can also be shown that:

$$\mathbf{V}_s^{-1}\mathbf{k}_s(x) = \begin{pmatrix} \rho_{s-1}(x)\mathbf{V}_{s-1}^{-1}\mathbf{k}_{s-1}(x) - \begin{pmatrix} 0 \\ \rho_{s-1}(\mathbf{D}_s) \odot (R_s(\mathbf{D}_s)^{-1}R_s(\mathbf{D}_s, \{x\})) \end{pmatrix} \\ R_s(\mathbf{D}_s)^{-1}R_s(\mathbf{D}_s, \{x\}) \end{pmatrix}.$$

A.2.4 Bayesian prediction for a code with 2 levels

The equations for the Bayesian prediction when ρ depends on x can be directly derived from the Section 3.4 by replacing ρ with $\boldsymbol{\beta}_\rho$ and noting that the design matrix $\mathbf{F}$ is such that:

$$\mathbf{F} = [F_\rho(\mathbf{D}_2) \odot (z_1(\mathbf{D}_2)\mathbf{1}'_{p_\rho}) \quad \mathbf{F}_2].$$

Finally, for the Bayesian prediction, we just have to adapt the integral (3.25) :

$$p(z_2(x)|\mathbf{z}_1, \mathbf{z}_2, \sigma_1^2, \sigma_2^2) = \int_{\mathbb{R}^{p_\rho + p_2}} p(z_2(x)|\mathbf{z}_1, \mathbf{z}_2, \boldsymbol{\beta}_2, \boldsymbol{\beta}_\rho, \sigma_1^2, \sigma_2^2) p(\boldsymbol{\beta}_\rho, \boldsymbol{\beta}_2|\mathbf{z}_1, \mathbf{z}_2, \sigma_2^2)\, d\boldsymbol{\beta}_\rho d\boldsymbol{\beta}_2.$$

Appendix B

Extension of the recursive formulation without nested experimental design sets (Chapter 4)

B.1 Multi-fidelity co-kriging models without nested experimental design sets

Thanks to the recursive formulation of the multi-fidelity co-kriging model presented in Chapter 4, we can easily adapt the method when the experimental design sets are not nested.

B.1.1 Building multi-fidelity co-kriging models when the design sets are not nested

Let us consider the recursive formulation of the multi-fidelity model $t = 2, \dots, s$:

$$\begin{cases} Z_t(x) = \rho_{t-1}(x) + \delta_t(x) \\ Z_{t-1}(x) \perp \delta_t(x) \\ \rho_{t-1}(x) = \mathbf{g}'_{t-1}(x)\boldsymbol{\beta}_{\rho_{t-1}} \end{cases},$$

where $\tilde{Z}_{t-1}(x)$ is a Gaussian process with distribution $[Z_{t-1}(x)|\mathbf{Z}^{(t-1)} = \mathbf{z}^{(t-1)}, \boldsymbol{\beta}_{t-1}, \boldsymbol{\beta}_{\rho_{t-2}}, \sigma^2_{t-1}]$. Without the nested property for the experimental design sets $(\mathbf{D}_t)_{t=1,\dots,s}$, we have for $t = 2, \dots, s$:

$$\begin{pmatrix} Z_t(x) \\ Z_t(\tilde{x}) \\ Z_t(\mathbf{D}_t) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \rho_{t-1}(x)\mu_{Z_{t-1}}(x) + \mathbf{f}'_t(x)\boldsymbol{\beta}_t \\ \rho_{t-1}(\tilde{x})\mu_{Z_{t-1}}(\tilde{x}) + \mathbf{f}'_t(\tilde{x})\boldsymbol{\beta}_t \\ \rho_{t-1}(\mathbf{D}_t) \odot \mu_{Z_{t-1}}(\mathbf{D}_t) + \mathbf{F}_t\boldsymbol{\beta}_t \end{pmatrix}, \boldsymbol{\Sigma} \right),$$

where:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} & \boldsymbol{\Sigma}_{13} \\ \boldsymbol{\Sigma}'_{12} & \boldsymbol{\Sigma}_{22} & \boldsymbol{\Sigma}_{23} \\ \boldsymbol{\Sigma}'_{13} & \boldsymbol{\Sigma}'_{23} & \boldsymbol{\Sigma}_{33} \end{pmatrix}$$

and:

$$\begin{aligned}
\mathbf{\Sigma}_{11} &= \operatorname{cov}\left(Z_t(x), Z_t(x)\right) = \rho_{t-1}^2(x) s_{Z_{t-1}}^2(x) + \sigma_t^2, \\
\mathbf{\Sigma}_{12} &= \mathbf{\Sigma}_{12}' = \operatorname{cov}\left(Z_t(x), Z_t(\tilde{x})\right) = \rho_{t-1}(x)\rho_{t-1}(\tilde{x}) s_{Z_{t-1}}^2(x,\tilde{x}) + \sigma_t^2 r_t(x,\tilde{x}), \\
\mathbf{\Sigma}_{13} &= \mathbf{\Sigma}_{13}' = \operatorname{cov}\left(Z_t(x), Z_t(\mathbf{D}_t)\right) = \rho_{t-1}(x)(\rho_{t-1}(\mathbf{D}_t)' \odot s_{Z_{t-1}}^2(x,\mathbf{D}_t)) + \sigma_t^2 \mathbf{r}_t'(x), \\
\mathbf{\Sigma}_{22} &= \operatorname{cov}\left(Z_t(\tilde{x}), Z_t(\tilde{x})\right) = \rho_{t-1}^2(\tilde{x}) s_{Z_{t-1}}^2(\tilde{x}) + \sigma_t^2, \\
\mathbf{\Sigma}_{23} &= \mathbf{\Sigma}_{23}' = \operatorname{cov}\left(Z_t(\tilde{x}), Z_t(\mathbf{D}_t)\right) = \rho_{t-1}(\tilde{x})(\rho_{t-1}(\mathbf{D}_t)' \odot s_{Z_{t-1}}^2(\tilde{x},\mathbf{D}_t)) + \sigma_t^2 \mathbf{r}_t'(\tilde{x}), \\
\mathbf{\Sigma}_{33} &= \operatorname{cov}\left(Z_t(\mathbf{D}_t), Z_t(\mathbf{D}_t)\right) = \left(\rho_{t-1}(\mathbf{D}_t)\rho_{t-1}(\mathbf{D}_t)'\right) \odot s_{Z_{t-1}}^2(\mathbf{D}_t,\mathbf{D}_t) + \sigma_t^2 \mathbf{R}_t.
\end{aligned}$$

We note that if $\tilde{x} \in \mathbf{D}_{t-1}$, then $\operatorname{cov}\left(Z_{t-1}(x), Z_{t-1}(\tilde{x}) | Z_{t-1}(\mathbf{D}_{t-1}) = z^{t-1}\right) = 0$. From the previous normal distribution, we deduce that for all $t = 2, \ldots, s$:

$$\left[Z_t(x) | Z_t(\mathbf{D}_t) = z^t\right] \sim \mathcal{PG}\left(\mu_{Z_t}(x), s_{Z_t}^2(x,\tilde{x})\right),$$

where the predictive mean is:

$$\begin{aligned}
\mu_{Z_t}(x) &= \rho_{t-1}(x)\mu_{Z_{t-1}}(x) + \mathbf{f}_t'(x)\boldsymbol{\beta}_t \\
&+ \begin{bmatrix} \rho_{t-1}(x)(\rho_{t-1}(\mathbf{D}_t)' \odot s_{Z_{t-1}}^2(x,\mathbf{D}_t)) \\ +\sigma_t^2 \mathbf{r}_t'(x) \end{bmatrix} \begin{bmatrix} [\rho_{t-1}(\mathbf{D}_t)\rho_{t-1}(\mathbf{D}_t)'] \odot s_{Z_{t-1}}^2(\mathbf{D}_t,\mathbf{D}_t) \\ +\sigma_t^2 \mathbf{R}_t \end{bmatrix}^{-1} \\
&\times \left(\mathbf{z}_t - \rho_{t-1}(\mathbf{D}_t) \odot \mu_{Z_{t-1}}(\mathbf{D}_t) - \mathbf{F}_t\boldsymbol{\beta}_t\right)
\end{aligned}$$

and the predictive variance is given by:

$$\begin{aligned}
s_{Z_t}^2(x,\tilde{x}) &= \rho_{t-1}(x)\rho_{t-1}(\tilde{x}) s_{Z_{t-1}}^2(x,\tilde{x}) + \sigma_t^2 r_t(x,\tilde{x}) \\
&- \begin{bmatrix} \rho_{t-1}(x)(\rho_{t-1}(\mathbf{D}_t)' \odot s_{Z_{t-1}}^2(x,\mathbf{D}_t)) \\ +\sigma_t^2 \mathbf{r}_t'(x) \end{bmatrix} \begin{bmatrix} [\rho_{t-1}(\mathbf{D}_t)\rho_{t-1}(\mathbf{D}_t)'] \odot s_{Z_{t-1}}^2(\mathbf{D}_t,\mathbf{D}_t) \\ +\sigma_t^2 \mathbf{R}_t \end{bmatrix}^{-1} \\
&\times \begin{bmatrix} \rho_{t-1}(\tilde{x})(\rho_{t-1}(\mathbf{D}_t) \odot s_{Z_{t-1}}^2(\mathbf{D}_t,\tilde{x})) \\ +\sigma_t^2 \mathbf{r}_t(\tilde{x}) \end{bmatrix}.
\end{aligned}$$

Furthermore, for the first level, we have

$$\begin{cases} \mu_{Z_1}(x) = \mathbf{f}_1'(x)\boldsymbol{\beta}_1 + \mathbf{r}_1'(x)\mathbf{R}_1^{-1}\left(\mathbf{z}_1 - \mathbf{F}_1\boldsymbol{\beta}_1\right) \\ s_{Z_1}^2(x,\tilde{x}) = \sigma_1^2\left(1 - \mathbf{r}_1'(x)\mathbf{R}_1^{-1}\mathbf{r}_1(\tilde{x})\right) \end{cases}.$$

B.1.2 Parameter estimation for the multi-fidelity co-kriging model when the design sets are not nested

We present two methods to estimate the parameters when the design are not nested. The first one assumes that the intersection between two successive design sets is not empty. The second one consider this intersection as empty.

The case where $\mathbf{D}_t \cap \mathbf{D}_{t-1} \neq \emptyset$: Let us denote by $\mathbf{D}_{t\cap t-1} = \mathbf{D}_t \cap \mathbf{D}_{t-1}$, we have:

$$Z_t(\mathbf{D}_{t\cap t-1}) \sim \mathcal{N}\left(\rho_{t-1}(\mathbf{D}_{t\cap t-1}) \odot z_{t-1}(\mathbf{D}_{t\cap t-1}) + \mathbf{F}_t(\mathbf{D}_{t\cap t-1})\boldsymbol{\beta}_t, \sigma_t^2 \mathbf{R}_t(\mathbf{D}_{t\cap t-1})\right),$$

where

$$\rho_{t-1}(\mathbf{D}_{t\cap t-1}) = \mathbf{G}_{t-1}(\mathbf{D}_{t\cap t-1})\boldsymbol{\beta}_{\rho_{t-1}}.$$

Denoting by $\mathbf{H}_t = [\mathbf{G}_{t-1}(\mathbf{D}_{t\cap t-1}) \odot z_{t-1}(\mathbf{D}_{t\cap t-1}) \quad \mathbf{F}_t(\mathbf{D}_{t\cap t-1})]$ and $\boldsymbol{\xi}_t = (\boldsymbol{\beta}_{\rho_{t-1}}, \boldsymbol{\beta}_t)$, we find exactly the same estimation as for the case of nested design sets for $\boldsymbol{\beta}$, $\boldsymbol{\beta}_\rho$, σ^2 and θ (see Subsection 4.2.3).

By using the law of total covariance, we can infer from $\boldsymbol{\beta}$ and $\boldsymbol{\beta}_{\rho_{t-1}}$ about the predictive covariance:

$$\begin{aligned}
s^2_{Z_t}(x,\tilde{x}) &= \rho_{t-1}(x)\rho_{t-1}(\tilde{x})s^2_{Z_{t-1}}(x,\tilde{x}) + \sigma_t^2 r_t(x,\tilde{x}) \\
&- \begin{bmatrix} \rho_{t-1}(x)\rho_{t-1}(\mathbf{D}_t)' \odot s^2_{Z_{t-1}}(x,\mathbf{D}_t) \\ +\sigma_t^2 \mathbf{r}_t'(x) \end{bmatrix} \begin{bmatrix} [\rho_{t-1}(\mathbf{D}_t)\rho_{t-1}(\mathbf{D}_t)'] \odot s^2_{Z_{t-1}}(\mathbf{D}_t,\mathbf{D}_t) \\ +\sigma_t^2 \mathbf{R}_t \end{bmatrix}^{-1} \\
&\times \begin{bmatrix} \rho_{t-1}(\tilde{x})\rho_{t-1}(\mathbf{D}_t) s^2_{Z_{t-1}}(\mathbf{D}_t,\tilde{x}) \\ +\sigma_t^2 \mathbf{r}_t \tilde{x}) \end{bmatrix} \\
&+ \left(\mathbf{f}_t'(x) - \begin{bmatrix} \rho_{t-1}(x)\rho_{t-1}(\mathbf{D}_t)' \odot s^2_{Z_{t-1}}(x,\mathbf{D}_t) \\ +\sigma_t^2 \mathbf{r}_t'(x) \end{bmatrix} \begin{bmatrix} [\rho_{t-1}(\mathbf{D}_t)\rho_{t-1}(\mathbf{D}_t)'] \odot s^2_{Z_{t-1}}(\mathbf{D}_t,\mathbf{D}_t) \\ +\sigma_t^2 \mathbf{R}_t \end{bmatrix}^{-1} \mathbf{F}_t\right) \\
&\times \sigma_t^2(\mathbf{H}_t\mathbf{R}_t^{-1}\mathbf{H}_t)^{-1} \\
&\times \left(\mathbf{f}_t'(\tilde{x}) - \begin{bmatrix} \rho_{t-1}(\tilde{x})\rho_{t-1}(\mathbf{D}_t)' \odot s^2_{Z_{t-1}}(\tilde{x},\mathbf{D}_t) \\ +\sigma_t^2 \mathbf{r}_t'(\tilde{x}) \end{bmatrix} \begin{bmatrix} [\rho_{t-1}(\mathbf{D}_t)\rho_{t-1}(\mathbf{D}_t)'] \odot s^2_{Z_{t-1}}(\mathbf{D}_t,\mathbf{D}_t) \\ +\sigma_t^2 \mathbf{R}_t \end{bmatrix}^{-1} \mathbf{F}_t\right)'.
\end{aligned}$$

The inference from $\boldsymbol{\beta}$, $\boldsymbol{\beta}_\rho$ and σ^2 about the predictive mean is straightforward by using law of total expectation. Nevertheless, the inference from σ_t^2 about the predictive covariance is not explicit since the predictive mean depends on it.

The case where $\mathbf{D}_t \cap \mathbf{D}_{t-1} = \emptyset$: When the intersection between two successive design sets is empty, the parameter estimations become complex. Nevertheless, we can estimate them sequentially starting with $(\boldsymbol{\beta}_1, \sigma_1, \theta_1)$ and continuing with $(\boldsymbol{\beta}_t, \boldsymbol{\beta}_{\rho_t}, \sigma_t, \theta_t)_{t=2,\dots,s}$. The estimations of $(\boldsymbol{\beta}_1, \sigma_1, \theta_1)$ is not a problem and can be performed with classical parameter estimation methods for kriging (see Section 1.3).

For the estimation of $\boldsymbol{\beta}_t, \boldsymbol{\beta}_{\rho_t}, \sigma_t, \theta_t$, we can use a maximum likelihood method. Indeed, we have:

$$Z_t(\mathbf{D}_t) \sim \mathcal{N}\left(\left(\rho_{t-1}(\mathbf{D}_t) \odot \mu_{Z_{t-1}}(\mathbf{D}_t) + \mathbf{F}_t\boldsymbol{\beta}_t\right), \begin{bmatrix} [\rho_{t-1}(\mathbf{D}_t)\rho_{t-1}(\mathbf{D}_t)'] \odot s^2_{Z_{t-1}}(\mathbf{D}_t,\mathbf{D}_t) \\ +\sigma_t^2 \mathbf{R}_t \end{bmatrix}\right),$$

$$Z_t(\mathbf{D}_t) \sim \mathcal{N}(\boldsymbol{\upsilon}, \boldsymbol{\Upsilon}),$$

therefore, the negative log-likelihood equals (up to a constant):

$$-\log\left(\mathcal{L}(z_t(\mathbf{D}_t), \boldsymbol{\beta}_t, \boldsymbol{\beta}_{\rho_t}, \sigma_t, \theta_t)\right) \propto \log\left(|\det(\boldsymbol{\Upsilon})|\right) + \boldsymbol{\upsilon}^T \boldsymbol{\Upsilon}^{-1} \boldsymbol{\upsilon}.$$

We estimate $\boldsymbol{\beta}_t, \boldsymbol{\beta}_{\rho_t}, \sigma_t, \theta_t$ by minimizing $-\log\left(\mathcal{L}(z_t(\mathbf{D}_t), \boldsymbol{\beta}_t, \boldsymbol{\beta}_{\rho_t}, \sigma_t, \theta_t)\right)$.

B.2 Fast cross validation for co-kriging multi-fidelity models without nested experimental design sets

We deal in this section with the fast-cross validation equations presented in Subsection 4.4 when the experimental design sets are not nested. Let us consider the model presented in Section B.1 and let us introduce the following notation:

$$\boldsymbol{\Sigma}_t = \begin{bmatrix} (\rho_{t-1}(\mathbf{D}_t)\rho_{t-1}(\mathbf{D}_t)') \odot s^2_{Z_{t-1}}(\mathbf{D}_t, \mathbf{D}_t) \\ +\sigma_t^2 \mathbf{R}_t \end{bmatrix}.$$

With the block matrix inverse formula, we obtain that:

$$\begin{aligned}
\left[\left[\boldsymbol{\Sigma}_t^{-1}\right]_{[\boldsymbol{\zeta}_t,\boldsymbol{\zeta}_t]}\right]^{-1} &= (\rho_{t-1}(\mathbf{D}_{t,\boldsymbol{\zeta}_t})\rho_{t-1}(\mathbf{D}_{t,\boldsymbol{\zeta}_t})') \odot s^2_{Z_{t-1}}(\mathbf{D}_{t,\boldsymbol{\zeta}_t}, \mathbf{D}_{t,\boldsymbol{\zeta}_t}) + \sigma_t^2 r_t(\mathbf{D}_{t,\boldsymbol{\zeta}_t}, \mathbf{D}_{t,\boldsymbol{\zeta}_t}) \\
&- \begin{bmatrix} (\rho_{t-1}(\mathbf{D}_{t,\boldsymbol{\zeta}_t})\rho_{t-1}(\mathbf{D}_{t,-\boldsymbol{\zeta}_t})') \odot s^2_{Z_{t-1}}(\mathbf{D}_{t,\boldsymbol{\zeta}_t}, \mathbf{D}_{t,-\boldsymbol{\zeta}_t}) \\ +\sigma_t^2 r_t(\mathbf{D}_{t,\boldsymbol{\zeta}_t}, \mathbf{D}_{t,-\boldsymbol{\zeta}_t}) \end{bmatrix} \\
&\times \begin{bmatrix} \left(\rho_{t-1}(\mathbf{D}_{t,-\boldsymbol{\zeta}_t})\rho_{t-1}(\mathbf{D}_{t,-\boldsymbol{\zeta}_t})'\right) \odot s^2_{Z_{t-1}}(\mathbf{D}_{t,-\boldsymbol{\zeta}_t}, \mathbf{D}_{t,-\boldsymbol{\zeta}_t}) \\ +\sigma_t^2 r_t(\mathbf{D}_{t,-\boldsymbol{\zeta}_t}, \mathbf{D}_{t,-\boldsymbol{\zeta}_t}) \end{bmatrix}^{-1} \\
&\times \begin{bmatrix} (\rho_{t-1}(\mathbf{D}_{t,\boldsymbol{\zeta}_t})'\rho_{t-1}(\mathbf{D}_{t,-\boldsymbol{\zeta}_t})) \odot s^2_{Z_{t-1}}(\mathbf{D}_{t,-\boldsymbol{\zeta}_t}, \mathbf{D}_{t,\boldsymbol{\zeta}_t}) \\ +\sigma_t^2 r_t(, \mathbf{D}_{t,-\boldsymbol{\zeta}_t}\mathbf{D}_{t,\boldsymbol{\zeta}_t}) \end{bmatrix},
\end{aligned}$$

where $\boldsymbol{\zeta}_t$ are the index that we remove from the design set $\mathbf{D}_t$. $\left[\left[\boldsymbol{\Sigma}_t^{-1}\right]_{[\boldsymbol{\zeta}_t,\boldsymbol{\zeta}_t]}\right]^{-1}$ is the predictive covariance matrix of the cross validation procedure. The predictive variance corresponds to the diagonal of this matrix:

$$\varsigma^2_{Z_t,\boldsymbol{\zeta}_t}(\mathbf{D}_{t,\boldsymbol{\zeta}_t}) = \operatorname{diag}\left(\left[\left[\boldsymbol{\Sigma}_t^{-1}\right]_{[\boldsymbol{\zeta}_t,\boldsymbol{\zeta}_t]}\right]^{-1}\right).$$

Furthermore, denoting by $\boldsymbol{\xi}_t = (\boldsymbol{\beta}_{\rho_{t-1}}, \boldsymbol{\beta}_t)$, we have:

$$\begin{aligned}
\left[\left[\boldsymbol{\Sigma}_t^{-1}\right]_{[\boldsymbol{\zeta}_t,\boldsymbol{\zeta}_t]}\right]^{-1} \left[\boldsymbol{\Sigma}_t^{-1}\left[\mathbf{z}_t - \mathbf{H}_t\boldsymbol{\xi}_t\right]\right]_{[\boldsymbol{\zeta}_t,\boldsymbol{\zeta}_t]} &= \mathbf{z}_t(\mathbf{D}_{t,\boldsymbol{\zeta}_t}) - \mathbf{H}_{t,-\boldsymbol{\zeta}_t}\boldsymbol{\xi}_t \\
&- \begin{bmatrix} (\rho_{t-1}(\mathbf{D}_{t,\boldsymbol{\zeta}_t})\rho_{t-1}(\mathbf{D}_{t,-\boldsymbol{\zeta}_t})') \odot s^2_{Z_{t-1}}(\mathbf{D}_{t,\boldsymbol{\zeta}_t}, \mathbf{D}_{t,-\boldsymbol{\zeta}_t}) \\ +\sigma_t^2 r_t(\mathbf{D}_{t,\boldsymbol{\zeta}_t}, \mathbf{D}_{t,-\boldsymbol{\zeta}_t}) \end{bmatrix} \\
&\times \begin{bmatrix} \left[\rho_{t-1}(\mathbf{D}_{t,-\boldsymbol{\zeta}_t})\rho_{t-1}(\mathbf{D}_{t,-\boldsymbol{\zeta}_t})'\right] \odot s^2_{Z_{t-1}}(\mathbf{D}_{t,-\boldsymbol{\zeta}_t}, \mathbf{D}_{t,-\boldsymbol{\zeta}_t}) \\ +\sigma_t^2 r_t(\mathbf{D}_{t,-\boldsymbol{\zeta}_t}, \mathbf{D}_{t,-\boldsymbol{\zeta}_t}) \end{bmatrix}^{-1} \\
&\times \left(\mathbf{z}_t(\mathbf{D}_{t,-\boldsymbol{\zeta}_t}) - \rho_{t-1}(\mathbf{D}_{t,-\boldsymbol{\zeta}_t}) \odot \mu_{Z_{t-1}}(\mathbf{D}_{t,-\boldsymbol{\zeta}_t}) - \mathbf{F}_{t,-\boldsymbol{\zeta}_t}\boldsymbol{\beta}_t\right),
\end{aligned}$$

where $\mathbf{H}_t = [\mathbf{G}_{t-1}(\mathbf{D}_t) \odot \mu_{Z_{t-1}}(\mathbf{D}_t) \quad \mathbf{F}_t]$. Finally, we obtain that:

$$\varepsilon_{Z_t,\boldsymbol{\zeta}_t}(\mathbf{D}_{t,\boldsymbol{\zeta}_t}) = \left[\left[\boldsymbol{\Sigma}_t^{-1}\right]_{[\boldsymbol{\zeta}_t,\boldsymbol{\zeta}_t]}\right]^{-1} \left[\boldsymbol{\Sigma}_t^{-1}\left[\mathbf{z}_t - \mathbf{H}_t\boldsymbol{\xi}_t\right]\right]_{[\boldsymbol{\zeta}_t,\boldsymbol{\zeta}_t]}.$$

Appendix C

Equivalence between multi-fidelity co-kriging models and noisy-kriging (Introduction of Part III)

Proof of Proposition 6.3

Proof. The normality of $\begin{pmatrix} Z(x) \\ \mathbf{Z}_{s^1}^{n_1} \\ \mathbf{Z}_{s^2}^{n_2} \end{pmatrix}$ and $\begin{pmatrix} Z(x) \\ \mathbf{Z}_{s^1}^{n_1-n_2} \\ \mathbf{Z}_{s^2}^{n_2} \end{pmatrix}$ implies that the distributions $[Z(x)|\mathbf{Z}_{s^1}^{n_1} = \mathbf{z}_{s^1}^{n_1}, \mathbf{Z}_{s^2}^{n_2} = \mathbf{z}_{s^2}^{n_2}, \sigma^2]$ and $[Z(x)|\mathbf{Z}_{s^1}^{n_1-n_2} = \mathbf{z}_{s^1}^{n_1-n_2}, \mathbf{Z}_{s^2}^{n_2} = \mathbf{z}_{s^2}^{n_2}, \sigma^2]$ are Gaussian. We just have to prove that they have the same mean and variance.

First, let us denote by $\mathbf{R}$ the correlation matrix of $\begin{pmatrix} \mathbf{Z}_{s^1}^{n_1} \\ \mathbf{Z}_{s^2}^{n_2} \end{pmatrix}$. We sort $\mathbf{D} = \{x_1, \ldots, x_{n_1}\}$ and $\tilde{\mathbf{D}} = \{x_1, \ldots, x_{n_2}\}$ such that for a fixed $l = 1, \ldots, n_2$, $Z_{s_l^1}(x_l)$ and $Z_{s_l^2}(x_l)$ are the last components of $\begin{pmatrix} \mathbf{Z}_{s^1}^{n_1} \\ \mathbf{Z}_{s^2}^{n_2} \end{pmatrix}$. After the sorting procedure, $\mathbf{R}$ can be written with the following form:

$$\mathbf{R} = \begin{pmatrix} \mathbf{T} & \mathbf{V}_l \\ \mathbf{V}_l' & \mathbf{W} \end{pmatrix},$$

where $\mathbf{V}_l = (\mathbf{v}_l \quad \mathbf{v}_l)$, $\mathbf{v}_l' = ([r(x_l, x_i)]_{i=1,\ldots,n_1-1} \quad [r(x_l, x_i)]_{i=1,\ldots,n_2-1})$,

$$\mathbf{W} = \begin{pmatrix} 1 + \sigma_\varepsilon^2(x_l)/s_l^1 & 1 + \sigma_\varepsilon^2(x_l)/s_l^2 \\ 1 + \sigma_\varepsilon^2(x_l)/s_l^2 & 1 + \sigma_\varepsilon^2(x_l)/s_l^2 \end{pmatrix}$$

and

$$\mathbf{T} = \begin{pmatrix} \mathrm{cor}\left(\mathbf{Z}_{s^1,-l}^{n_1}, \mathbf{Z}_{s^1,-l}^{n_1}\right) & \mathrm{cor}\left(\mathbf{Z}_{s^1,-l}^{n_1}, \mathbf{Z}_{s^2,-l}^{n_2}\right) \\ \mathrm{cor}\left(\mathbf{Z}_{s^2,-l}^{n_2}, \mathbf{Z}_{s^1,-l}^{n_1}\right) & \mathrm{cor}\left(\mathbf{Z}_{s^2,-l}^{n_2}, \mathbf{Z}_{s^2,-l}^{n_2}\right) \end{pmatrix},$$

where $\mathbf{Z}^{n_j}_{s^j,-l}$ denotes the vector $\mathbf{Z}^{n_j}_{s^j}$ without the lth components $j = 1, 2$. Let us consider the following matrix:

$$\mathbf{Q} = \begin{pmatrix} 1 - a_l + \tau_l^1 & 1 - a_l + \tau_l^2 \\ 1 - a_l + \tau_l^2 & 1 - a_l + \tau_l^2 \end{pmatrix},$$

with $a_l = \mathbf{v}_l'\mathbf{T}\mathbf{v}_l$, $\tau_l^1 = \sigma_\varepsilon^2(x_l)/s_l^1$ and $\tau_l^2 = \sigma_\varepsilon^2(x_l)/s_l^2$. Denoting by $b_l = (\tau_l^1 - \tau_l^2)^{-1}$, the inverse of $\mathbf{Q}$ is given by:

$$\mathbf{Q}^{-1} = \begin{pmatrix} b_l & -b_l \\ -b_l & (1 - a_l + \tau_l^2)^{-1} + b_l \end{pmatrix}.$$

The block matrix inversion formula gives

$$\mathbf{R}^{-1} = \begin{pmatrix} \mathbf{T}^{-1} + \mathbf{T}^{-1}\mathbf{v}_l(1 - a_l + \tau_l^2)^{-1}\mathbf{v}_l'\mathbf{T}^{-1} & -\mathbf{T}^{-1}\begin{pmatrix} 0_{(n_1+n_2-2)\times 1} & \mathbf{v}_l(1 - a_l + \tau_l^2)^{-1} \end{pmatrix} \\ \begin{pmatrix} 0_{1\times(n_1+n_2-2)} \\ \mathbf{v}_l'(1 - a_l + \tau_l^2)^{-1} \end{pmatrix} & \begin{pmatrix} b_l & -b_l \\ -b_l & (1 - a_l + \tau_l^2)^{-1} + b_l \end{pmatrix} \end{pmatrix}.$$

Then, if we denote by $\mathbf{k}'(x)$ the correlation vector between $Z(x)$ and $\begin{pmatrix} \mathbf{Z}^{n_1}_{s^1} \\ \mathbf{Z}^{n_2}_{s^2} \end{pmatrix}$, after the sorting procedure, it has the following form

$$\mathbf{k}'(x) = \begin{pmatrix} \operatorname{cor}\left(Z(x), \begin{pmatrix} \mathbf{Z}^{n_1}_{s^1,-l} \\ \mathbf{Z}^{n_2}_{s^2,-l} \end{pmatrix}\right) & c_l & c_l \end{pmatrix},$$

where $c_l = \operatorname{cor}(Z(x), Z(x_l))$. Furthermore, the vector of observed values can be written:

$$\mathbf{z}^{n_1+n_2} = \begin{pmatrix} \mathbf{z}^{n_1}_{s^1,-l} & \mathbf{z}^{n_2}_{s^2,-l} & z_{s_l^1}(x_l) & z_{s_l^2}(x_l) \end{pmatrix},$$

where $\mathbf{z}^{n_j}_{s^j,-l}$ stands for the vector $\mathbf{z}^{n_j}_{s^j}$ without the lth components for $j = 1, 2$. After straightforward calculations we obtain the equalities:

$$\mathbf{k}'(x)\mathbf{R}^{-1}\mathbf{k}(x) = \tilde{\mathbf{k}}'(x)\tilde{\mathbf{R}}^{-1}\tilde{\mathbf{k}}(x),$$

$$\mathbf{k}'(x)\mathbf{R}^{-1}\mathbf{z}^{n_1+n_2} = \tilde{\mathbf{k}}'(x)\tilde{\mathbf{R}}^{-1}\tilde{\mathbf{z}}^{n_1+n_2}_{-l},$$

$$\mathbf{k}'(x)\mathbf{R}^{-1}\begin{pmatrix} \mathbf{f}'(\mathbf{D}) \\ \mathbf{f}'(\tilde{\mathbf{D}}) \end{pmatrix} = \tilde{\mathbf{k}}'(x)\tilde{\mathbf{R}}^{-1}\begin{pmatrix} \mathbf{f}'(\mathbf{D}_{-l}) \\ \mathbf{f}'(\tilde{\mathbf{D}}) \end{pmatrix},$$

$$\begin{pmatrix} \mathbf{f}(\mathbf{D}) & \mathbf{f}(\tilde{\mathbf{D}}) \end{pmatrix}\mathbf{R}^{-1}\begin{pmatrix} \mathbf{f}'(\mathbf{D}) \\ \mathbf{f}'(\tilde{\mathbf{D}}) \end{pmatrix} = \begin{pmatrix} \mathbf{f}(\mathbf{D}_{-l}) & \mathbf{f}(\tilde{\mathbf{D}}) \end{pmatrix}\tilde{\mathbf{R}}^{-1}\begin{pmatrix} \mathbf{f}'(\mathbf{D}_{-l}) \\ \mathbf{f}'(\tilde{\mathbf{D}}) \end{pmatrix}$$

and

$$\begin{pmatrix} \mathbf{f}(\mathbf{D}) & \mathbf{f}(\tilde{\mathbf{D}}) \end{pmatrix}\mathbf{R}^{-1}\mathbf{z}^{n_1+n_2} = \begin{pmatrix} \mathbf{f}(\mathbf{D}_{-l}) & \mathbf{f}(\tilde{\mathbf{D}}) \end{pmatrix}\tilde{\mathbf{R}}^{-1}\tilde{\mathbf{z}}^{n_1+n_2}_{-l}.$$

where $\mathbf{D}_{-l}$ is the experimental design set $\mathbf{D}$ without the lth row,

$$\tilde{\mathbf{k}}'(x) = \begin{pmatrix} \operatorname{cor}\left(Z(x), \begin{pmatrix} \mathbf{Z}^{n_1}_{s^1,-l} \\ \mathbf{Z}^{n_2}_{s^2,-l} \end{pmatrix}\right) & c_l \end{pmatrix},$$

$$\tilde{\mathbf{z}}_{-l}^{n_1+n_2} = \begin{pmatrix} \mathbf{z}_{s^1,-l}^{n_1} & \mathbf{z}_{s^2,-l}^{n_2} & z_{s_l^2}(x_l) \end{pmatrix}$$

and

$$\tilde{\mathbf{R}} = \begin{pmatrix} \text{cor}\left(\mathbf{Z}_{s^1,-l}^{n_1}, \mathbf{Z}_{s^1,-l}^{n_1}\right) & \text{cor}\left(\mathbf{Z}_{s^1,-l}^{n_1}, \mathbf{Z}_{s^2}^{n_2}\right) \\ \text{cor}\left(\mathbf{Z}_{s^2}^{n_2}, \mathbf{Z}_{s^1,-l}^{n_1}\right) & \text{cor}\left(\mathbf{Z}_{s^2}^{n_2}, \mathbf{Z}_{s^2}^{n_2}\right) \end{pmatrix}.$$

Using the same result as presented in Subsection 1.2.2, the predictive mean $\mu_{n_1,n_2}(x)$ and variance $s^2_{n_1,n_2}(x)$ of $[Z(x)|\mathbf{Z}_{s^1}^{n_1} = \mathbf{z}_{s^1}^{n_1}, \mathbf{Z}_{s^2}^{n_2} = \mathbf{z}_{s^2}^{n_2}, \sigma^2]$ are given by:

$$\mu_{n_1,n_2}(x) = \mathbf{f}'(x)\hat{\boldsymbol{\beta}} + \mathbf{k}'(x)\mathbf{R}^{-1}\left(\mathbf{z}^{n_1+n_2} - \mathbf{H}\hat{\boldsymbol{\beta}}\right),$$

where: $\hat{\boldsymbol{\beta}} = \left(\mathbf{H}'\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}'\mathbf{R}^{-1}\mathbf{z}^{n_1+n_2}$ and

$$\begin{aligned} s^2_{n_1,n_2}(x) &= \sigma^2\Big(1 - \mathbf{k}'(x)\mathbf{R}^{-1}\mathbf{k}(x) \\ &\quad + (\mathbf{f}'(x) - \mathbf{k}'(x)\mathbf{R}^{-1}\mathbf{H})\Big)\left[\mathbf{H}'\mathbf{R}^{-1}\mathbf{H}\right]^{-1}(\mathbf{f}'(x) - \mathbf{k}'(x)\mathbf{R}^{-1}\mathbf{H})', \end{aligned}$$

with $\mathbf{H}' = \begin{pmatrix} \mathbf{f}(\mathbf{D}) & \mathbf{f}(\tilde{\mathbf{D}}) \end{pmatrix}$.

Furthermore, the mean $\mu_{n_1,n_2,-l}(x)$ and the variance $s^2_{n_1,n_2,-l}(x)$ of the distribution $[Z(x)|\mathbf{Z}_{s^1,-l}^{n_1} = \mathbf{z}_{s^1,-l}^{n_1}, \mathbf{Z}_{s^2}^{n_2} = \mathbf{z}_{s^2}^{n_2}, \sigma^2]$ equal:

$$\mu_{n_1,n_2,-l}(x) = \mathbf{f}'(x)\hat{\tilde{\boldsymbol{\beta}}} + \tilde{\mathbf{k}}'(x)\tilde{\mathbf{R}}^{-1}\left(\tilde{\mathbf{z}}_{-l}^{n_1+n_2} - \mathbf{H}_{-l}\hat{\tilde{\boldsymbol{\beta}}}\right),$$

where: $\hat{\tilde{\boldsymbol{\beta}}} = \left(\mathbf{H}'_{-l}\tilde{\mathbf{R}}^{-1}\mathbf{H}_{-l}\right)^{-1}\mathbf{H}'_{-l}\tilde{\mathbf{R}}^{-1}\tilde{\mathbf{z}}_{-l}^{n_1+n_2}$ and

$$\begin{aligned} s^2_{n_1,n_2,-l}(x) &= \sigma^2\Big(-\tilde{\mathbf{k}}'(x)\tilde{\mathbf{R}}^{-1}\tilde{\mathbf{k}}(x) \\ &\quad + (\mathbf{f}'(x) - \tilde{\mathbf{k}}'(x)\tilde{\mathbf{R}}^{-1}\mathbf{H}_{-l})\Big)\left[\mathbf{H}'_{-l}\tilde{\mathbf{R}}^{-1}\mathbf{H}_{-l}\right]^{-1}(\mathbf{f}'(x) - \tilde{\mathbf{k}}'(x)\tilde{\mathbf{R}}^{-1}\mathbf{H}_{-l})', \end{aligned}$$

with $\mathbf{H}'_{-l} = \begin{pmatrix} \mathbf{f}(\mathbf{D}_{-l}) & \mathbf{f}(\tilde{\mathbf{D}}) \end{pmatrix}$. Therefore, we obtain with the previous equalities:

$$\begin{aligned} \mu_{n_1,n_2,-l}(x) &= \mu_{n_1,n_2}(x) \\ s^2_{n_1,n_2,-l}(x) &= s^2_{n_1,n_2}(x) \end{aligned}.$$

Finally, proceeding in the same way for all $l = 1, \ldots, n_2$ we obtain that $[Z(x)|\mathbf{Z}_{s^1}^{n_1} = \mathbf{z}_{s^1}^{n_1}, \mathbf{Z}_{s^2}^{n_2} = \mathbf{z}_{s^2}^{n_2}, \sigma^2]$ and $[Z(x)|\mathbf{Z}_{s^1}^{n_1-n_2} = \mathbf{z}_{s^1}^{n_1-n_2}, \mathbf{Z}_{s^2}^{n_2} = \mathbf{z}_{s^2}^{n_2}, \sigma^2]$, have the same mean and variance. □

Appendix D

Optimal resource allocation (Chapter 7)

D.1 Proof of Proposition 7.3

The minimization of the IMSE in Equation (7.23) with respect to $(s_i)_{i=1,\dots,n}$ and under the constraints $\sum_{i=1}^n s_i = T$ and $s_i \geq 1$, $\forall i = 1, \dots, n$ is equivalent to the following minimization problem when $\mathbf{K}$ is diagonal:

$$\arg\min_{(u_i)_i} \sum_{i=1}^n -\frac{c(x_i)}{k(x_i,x_i) + \frac{\sigma_\varepsilon^2(x_i)}{u_i+1}}, \tag{D.1}$$

$$\text{u.c.} \qquad \sum_{i=1}^n u_i = T - n, \qquad u_i \geq 0, \forall i = 1, \dots, n, \tag{D.2}$$

with $s_i = u_i + 1$ and $T \geq n$. The Lagrangian formulation of this problem with $(u, \lambda) \in \mathbb{R}_+^n \times \mathbb{R}$ is given by:

$$\begin{aligned}
\mathcal{L}(r\lambda) &= \sum_{i=1}^n -\frac{c(x_i)}{k(x_i,x_i) + \frac{\sigma_\varepsilon^2(x_i)}{1+u_i}} + \lambda\left(\sum_{i=1}^n u_i - T + n\right) \\
&= \sum_{i=1}^n \left(-\frac{c(x_i)}{k(x_i,x_i) + \frac{\sigma_\varepsilon^2(x_i)}{1+u_i}} + \lambda u_i\right) - \lambda(T-n).
\end{aligned}$$

We solve the dual problem which consists on finding $(u(\lambda^*), \lambda^*)$ such that:

$$\mathcal{L}(u(\lambda^*), \lambda^*) = \max_{\lambda \geq 0} \min_{u \in \mathbb{R}_+^n} \mathcal{L}(u, \lambda). \tag{D.3}$$

We note that $u(\lambda^*)$ will be the solution of the problem (D.1). Minimizing $\mathcal{L}(u, \lambda)$ with respect to u for a fixed λ is equivalent to minimizing each element $-\frac{c(x_i)}{k(x_i,x_i)+\frac{\sigma_\varepsilon^2(x_i)}{1+u_i}} + \lambda u_i$ with respect to $u_i \geq 0$.

Let us consider the function $h_i(x, \lambda) = -\frac{c(x_i)}{k(x_i,x_i)+\frac{\sigma_\varepsilon^2(x_i)}{1+x}} + \lambda x$ with $x \in \mathbb{R}_+$ and let us denote $u_i(\lambda) = \arg\min_{x \geq 0} h_i(x, \lambda)$. The sign of the derivative of $h_i(x, \lambda)$ is the same as the one of $\lambda\left(k(x_i,x_i)(1+u_i) + \sigma_\varepsilon^2(x_i)\right)^2 - c(x_i)\sigma_\varepsilon^2(x_i)$. Therefore, we have the three following cases:

1. $\lambda \leq 0 \Rightarrow \forall i,\ h_i(x,\lambda)$ is decreasing with respect to x and $u_i(\lambda) = +\infty$.
2. $0 < \lambda \leq \frac{c(x_i)\sigma_\varepsilon^2(x_i)}{(k(x_i,x_i)+\sigma_\varepsilon^2(x_i))^2} \Rightarrow h_i(x,\lambda)$ reaches is unique minimum at $u_i(\lambda) = \frac{1}{k(x_i,x_i)}\left(\sqrt{\frac{c(x_i)\sigma_\varepsilon^2(x_i)}{\lambda}} - k(x_i,x_i) - \sigma_\varepsilon^2(x_i)\right)$.
3. $\lambda > \frac{c(x_i)\sigma_\varepsilon^2(x_i)}{(k(x_i,x_i)+\sigma_\varepsilon^2(x_i))^2} \Rightarrow h_i(x,\lambda)$ is increasing with respect to x and $u_i(\lambda) = 0$.

For the rest of the proof, we use the notation:

$$\alpha(x) = \frac{c(x)\sigma_\varepsilon^2(x)}{(k(x,x)+\sigma_\varepsilon^2(x))^2}$$

Let us look for the $\lambda \geq 0$ which maximizes $\min_{u\in\mathbb{R}_+^n} \mathcal{L}(u,\lambda) = \mathcal{L}(u(\lambda),\lambda)$. According to the three previous cases, the maximum will obtained for $\lambda > 0$. We hence have to maximize with respect to $\lambda > 0$ the following quantity:

$$\begin{aligned}\mathcal{L}(u(\lambda),\lambda) &= \sum_{i=1}^n 1_{0<\lambda\leq\alpha(x_i)} \frac{c(x_i)}{k(x_i,x_i)}\left(2\sqrt{\frac{\lambda\sigma_\varepsilon^2(x_i)}{c(x_i)}} - 1 - \frac{\lambda\left(\sigma_\varepsilon^2(x_i)+k(x_i,x_i)\right)}{c(x_i)}\right)\\ &+ \sum_{i=1}^n 1_{\lambda>\alpha(x_i)} \frac{-c(x_i)}{k(x_i,x_i)+\sigma_\varepsilon^2(x_i)} - \lambda(T-n).\end{aligned}$$

Then:

$$\partial_\lambda\mathcal{L}(r(\lambda),\lambda) = \sum_{i=1}^n 1_{0<\lambda\leq\alpha(x_i)} \frac{c(x_i)}{k(x_i,x_i)}\left(\sqrt{\frac{\sigma_\varepsilon^2(x_i)}{\lambda c(x_i)}} - \frac{k(x_i,x_i)+\sigma_\varepsilon^2(x_i)}{c(x_i)}\right) - (T-n). \quad \text{(D.4)}$$

The function $\partial_\lambda\mathcal{L}(u(\lambda),\lambda)$ is continuous with respect to λ, equals $-T+n$ for $\lambda > \max_{i=1,\dots,n}\alpha(x_i)$ and is strictly decreasing on $(0, \max_{i=1,\dots,n}\alpha(x_i))$. Furthermore, $\partial_\lambda\mathcal{L}(u(\lambda),\lambda) \to \infty$ when $\lambda \to 0$. Therefore, $\mathcal{L}(u(\lambda),\lambda)$ admits a unique maximum at λ^* verifying the equation $\partial_\lambda\mathcal{L}(u(\lambda^*),\lambda^*) = 0$. We now re-index the experimental design set $\{1,\dots,n\}$ such that the quantities $\alpha(x_i)$ form a non-decreasing sequence. This sequence gives a partition of $(0, \max_{i=1,\dots,n}\alpha(x_i))$ and we will look for the sub-interval containing λ^*.

If $\partial_\lambda\mathcal{L}\left(u\left(\alpha(x_i)\right),\alpha(x_i)\right) < 0 \quad \forall i$, we set $i^* = 0$ and we have $\lambda^* \in (0,\alpha(x_1))$. Otherwise, i^* is the index such that:

$$\partial_\lambda\mathcal{L}\left(u\left(\alpha(x_{i^*})\right),\alpha(x_{i^*})\right) \geq 0 \quad \text{(D.5)}$$

and:

$$\partial_\lambda\mathcal{L}\left(u\left(\alpha(x_{i^*+1})\right),\alpha(x_{i^*+1})\right) < 0, \quad \text{(D.6)}$$

and then:

$$\lambda^* \in \left[\alpha(x_{i^*}),\alpha(x_{i^*}+1)\right). \quad \text{(D.7)}$$

Therefore, for $\lambda \in (0, \max_{i=1,\dots,n}\alpha(x_i))$, we have:

$$\partial_\lambda\mathcal{L}(u(\lambda),\lambda) = \sum_{i=i^*+1}^n \frac{c(x_i)}{k(x_i,x_i)}\left(\sqrt{\frac{\sigma_\varepsilon^2(x_i)}{\lambda c(x_i)}} - \frac{k(x_i,x_i)+\sigma_\varepsilon^2(x_i)}{c(x_i)}\right) - (T-n). \quad \text{(D.8)}$$

Furthermore, we have $\partial_\lambda \mathcal{L}(u(\lambda^*), \lambda^*) = 0$ which is equivalent to:

$$\frac{1}{\sqrt{\lambda^*}} = \frac{T - i^* + \sum_{i=i^*+1}^n \frac{\sigma_\varepsilon^2(x_i)}{k(x_i,x_i)}}{\sum_{i=i^*+1}^n \frac{\sqrt{c(x_i)\sigma_\varepsilon^2(x_i)}}{k(x_i,x_i)}}. \tag{D.9}$$

From which we deduce that:

$$u_i(\lambda^*) = \begin{cases} 0 & i \leq i^* \\ \frac{1}{k(x_i,x_i)} \left(\frac{\sqrt{c(x_i)\sigma_\varepsilon^2(x_i)}}{\sum_{j>i^*}^n \frac{\sqrt{c(x_j)\sigma_\varepsilon^2(x_j)}}{k(x_j,x_j)}} \left(T - i^* + \sum_{j>i^*}^n \frac{\sigma_\varepsilon^2(x_j)}{k(x_j,x_j)} \right) - \sigma_\varepsilon^2(x_i) \right) - 1 & i > i^* \end{cases}. \tag{D.10}$$

Finally, we have $\mathcal{L}(u(\lambda^*), \lambda^*) = \max_{\lambda \geq 0} \min_{u \in \mathbb{R}_+^n} \mathcal{L}(u, \lambda)$. As the function to minimize is a convex differentiable function, the function $\mathcal{L}(u(\lambda), \lambda)$ is concave and the constraints are affine, the saddle point found verifies the Karush-Kuhn-Tucker (KKT) conditions and consequently is the unique solution of the problem.

Furthermore, since $\partial_\lambda \mathcal{L}(u(\lambda), \lambda)$ is strictly decreasing with respect to λ on the interval $(0, \max_{i=1,\dots,n} \alpha(x_i))$, we have have the following equivalences:

$$i \leq i^* \quad \Leftrightarrow \quad \partial_\lambda \mathcal{L}\left(u\left(\alpha(x_i)\right), \alpha(x_i)\right) \geq 0 \tag{D.11}$$

$$\Leftrightarrow \quad \frac{k(x_j,x_j) + \sigma_\varepsilon^2(x_j)}{\sqrt{c(x_j)\sigma_\varepsilon^2(x_j)}} \geq \frac{T - i + \sum_{j=i+1}^n \frac{\sigma_\varepsilon^2(x_j)}{k(x_j,x_j)}}{\sum_{j=i+1}^n \frac{\sqrt{c_j\sigma_\varepsilon^2(x_j)}}{k(x_j,x_j)}}, \tag{D.12}$$

$$i^* = 0 \Leftrightarrow \frac{k(x_j,x_j) + \sigma_\varepsilon^2(x_j)}{\sqrt{c(x_j)\sigma_\varepsilon^2(x_j)}} < \frac{T - i + \sum_{j=i+1}^n \frac{\sigma_\varepsilon^2(x_j)}{k(x_j,x_j)}}{\sum_{j=i+1}^n \frac{\sqrt{c_j\sigma_\varepsilon^2(x_j)}}{k(x_j,x_j)}}, \qquad \forall i = 1, \dots, n. \tag{D.13}$$

The result announced in the proposition is obtained by replacing $s_i = u_i + 1$. □.

D.2 Numerical illustrations

We present in this section numerical illustrations on the optimal allocation suggested in Proposition 7.3. We compute them for various covariance kernels (Matérn, Gaussian, γ-exponential) with variance parameter $\sigma^2 = 1$, different dimensions d and number of observations n and two type of measure μ for the experimental design set (uniform and Gaussian). First, we present the case of large T - i.e. $i^* = 0$ in Proposition 7.3 - and small characteristic length scales to fit with the assumptions of the proposition. Then, we present the general case of non-diagonal covariance matrix $\mathbf{K}$. Finally, we illustrate the allocation for small budget T.

Let us summary below the protocol used for the comparison:

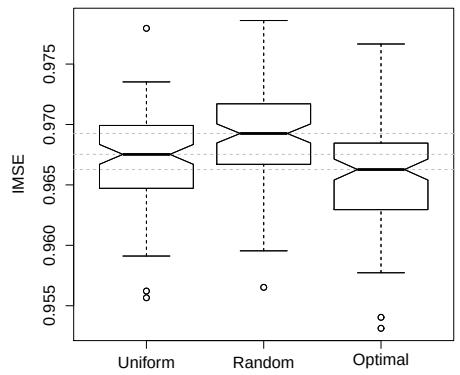
1. Two measures $\mu(x)$ are considered for the experimental design sets: $\mu_{\mathrm{G}}(x) \sim \mathcal{N}(0, \mathbf{I}_d)$, $\mu_{\mathrm{U}}(x) = \prod_{i=1}^d \mathbf{1}_{x^i \in [0,1]}$

2. The measure for the evaluation of the IMSE is $\eta(x) = \mu(x)$ and it is performed thanks to a Monte-Carlo integration with $10000d$ points when $d = 6$ and with a trapezoidal numerical integration with 2000 points when $d = 1$.

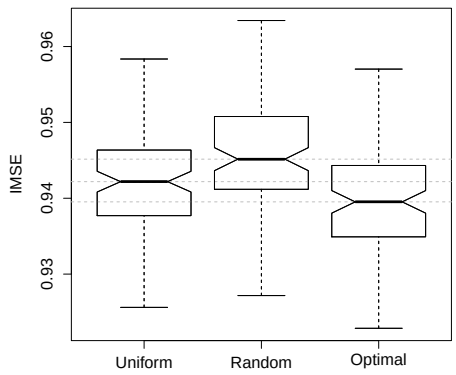
3. The comparisons are performed from 100 different experimental design sets generated with respect to $\mu(x)$.

4. The noise variance for the n observations are randomly sampled from a uniform distribution between 0 and 5.

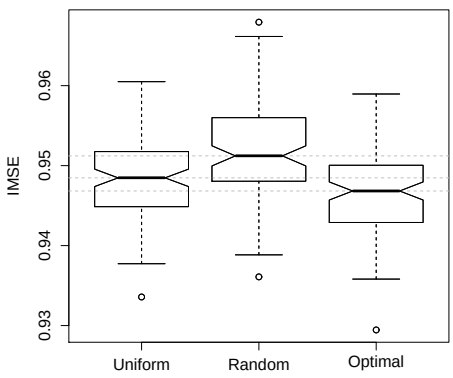
Comparison in dimension 1 with a uniform measure $\mu_{\mathrm{U}}(x)$ with large T.



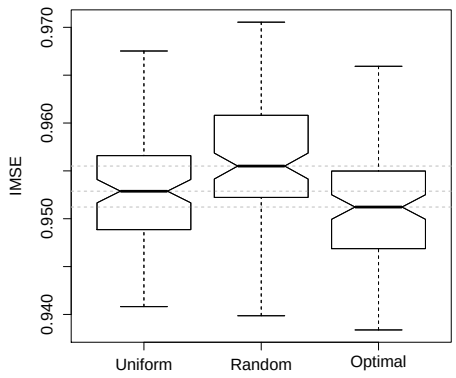
(a) $d = 1$, $T = 50$, $n = 10$, $\theta = 5.10^{-3}$
$k(.,.)$: exponential, $\eta(x) = \mu_{\mathrm{U}}(x)$



(b) $d = 1$, $T = 50$, $n = 10$, $\theta = 5.10^{-3}$
$k(.,.)$: Gaussian, $\eta(x) = \mu_{\mathrm{U}}(x)$

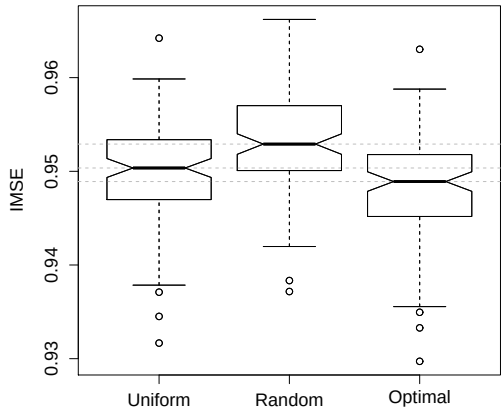


(c) $d = 1$, $T = 50$, $n = 10$, $\theta = 5.10^{-3}$
$k(.,.)$: 5/2-Matérn, $\eta(x) = \mu_{\mathrm{U}}(x)$

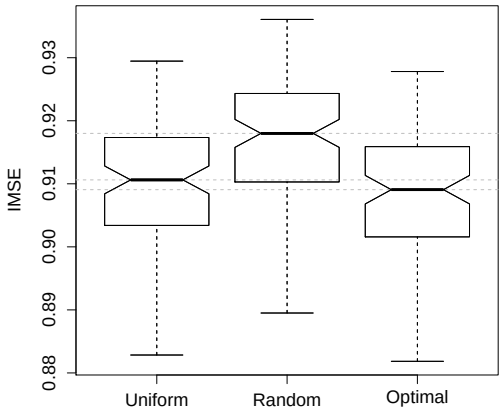


(d) $d = 1$, $T = 50$, $n = 10$, $\theta = 5.10^{-3}$
$k(.,.)$: 3/2-Matérn, $\eta(x) = \mu_{\mathrm{U}}(x)$

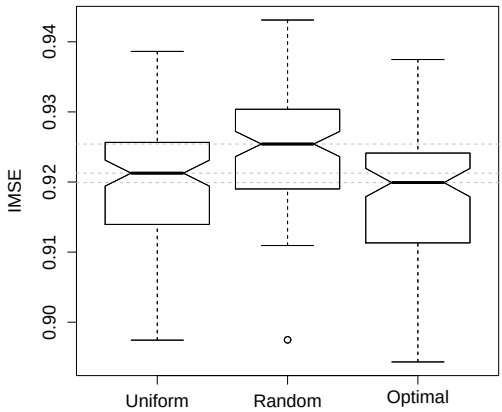
Figure D.1: Comparison between uniform, random and optimal allocations on an example in dimension $d = 1$ with heteroscedastic observation noise variance. 100 experimental design sets of $n = 10$ points are randomly sampled from the measure $\mu_{\mathrm{U}}(x)$. The budget is $T = 50$ and the correlation length $\theta = 5.10^{-3}$ is small in order to be close to the assumption $\mathbf{K}$ diagonal. We see that the optimal allocation is significantly better than the two other ones. This is natural since we fit to the assumptions of Proposition 7.3 (i.e. $\mathbf{K}$ diagonal).



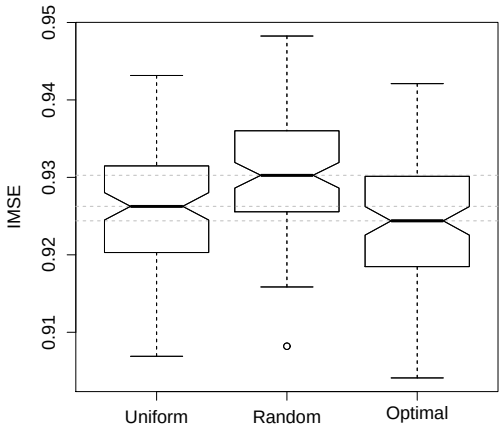
(a) $d = 1$, $T = 50$, $n = 20$, $\theta = 5.10^{-3}$
$k(.,.)$: exponential, $\eta(x) = \mu_{\mathrm{U}}(x)$



(b) $d = 1$, $T = 50$, $n = 20$, $\theta = 5.10^{-3}$
$k(.,.)$: Gaussian, $\eta(x) = \mu_{\mathrm{U}}(x)$

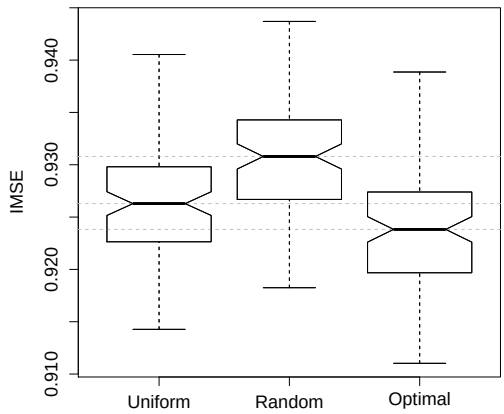


(c) $d = 1$, $T = 50$, $n = 20$, $\theta = 5.10^{-3}$
$k(.,.)$: 5/2-Matérn, $\eta(x) = \mu_{\mathrm{U}}(x)$

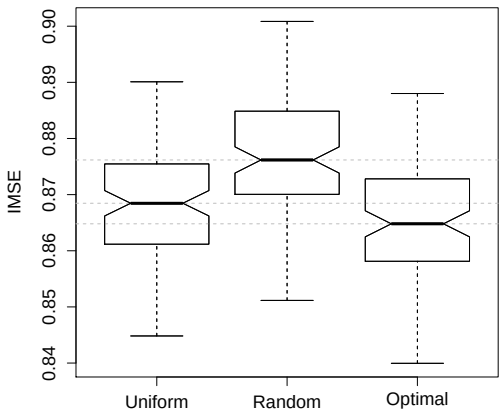


(d) $d = 1$, $T = 50$, $n = 20$, $\theta = 5.10^{-3}$
$k(.,.)$: 3/2-Matérn, $\eta(x) = \mu_{\mathrm{U}}(x)$

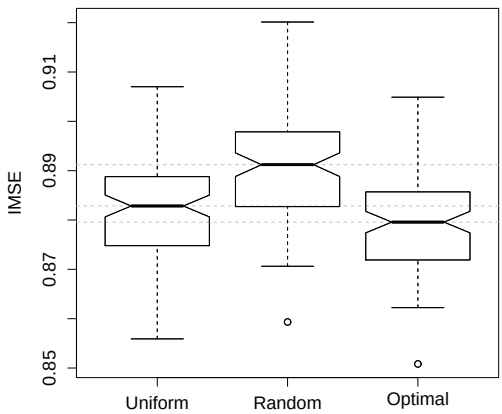
Figure D.2: Comparison between uniform, random and optimal allocations on an example in dimension $d = 1$ with heteroscedastic observation noise variance. 100 experimental design sets of $n = 20$ points are randomly sampled from the measure $\mu_{\mathrm{U}}(x)$. The budget is $T = 50$ and the correlation length $\theta = 5.10^{-3}$ is small in order to be close to the assumption $\mathbf{K}$ diagonal. We see that the optimal allocation is better than the two other ones. Nevertheless, the difference between the uniform and the optimal allocation is smaller than in the case illustrated in Figure D.1. This is due to the fact that since n increases, we stray from the assumptions of Proposition 7.3 (i.e. $\mathbf{K}$ diagonal). Furthermore, the difference between the uniform and the optimal allocation is smaller for the Gaussian and the 5/2-Matèrn covariance kernels (Figures (b) and (c)) than for the exponential and the 3/2-Matèrn covariance kernels (Figures (a) and (d)). This is natural since for irregular kernels, we are closer to the assumption $\mathbf{K}$ diagonal.



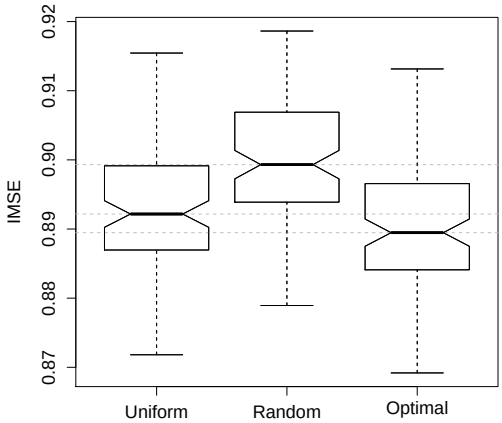
(a) $d = 1$, $T = 200$, $n = 20$, $\theta = 5.10^{-3}$
$k(.,.)$: exponential, $\eta(x) = \mu_{\mathrm{U}}(x)$



(b) $d = 1$, $T = 200$, $n = 20$, $\theta = 5.10^{-3}$
$k(.,.)$: Gaussian, $\eta(x) = \mu_{\mathrm{U}}(x)$



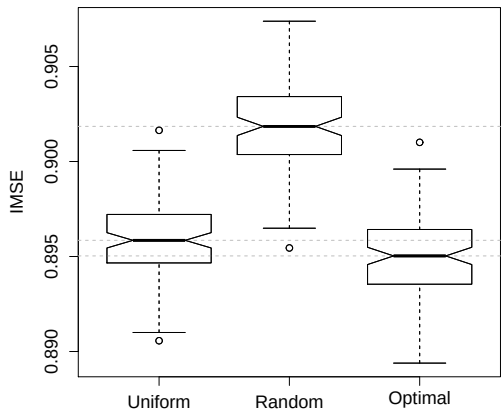
(c) $d = 1$, $T = 200$, $n = 20$, $\theta = 5.10^{-3}$
$k(.,.)$: 5/2-Matérn, $\eta(x) = \mu_{\mathrm{U}}(x)$



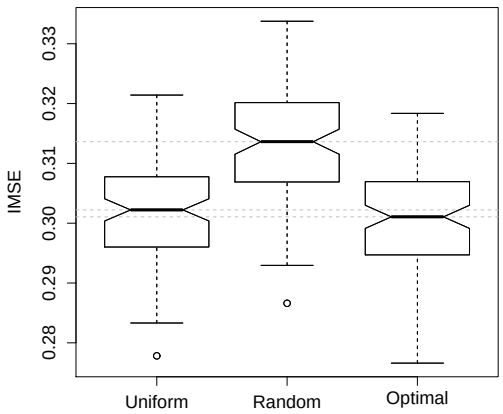
(d) $d = 1$, $T = 200$, $n = 20$, $\theta = 5.10^{-3}$
$k(.,.)$: 3/2-Matérn, $\eta(x) = \mu_{\mathrm{U}}(x)$

Figure D.3: Comparison between uniform, random and optimal allocations on an example in dimension $d = 1$ with heteroscedastic observation noise variance. 100 experimental design sets of $n = 20$ points are randomly sampled from the measure $\mu_{\mathrm{U}}(x)$. The budget is $T = 200$ and the correlation length $\theta = 5.10^{-3}$ is small in order to be close to the assumption $\mathbf{K}$ diagonal. We see that the optimal allocation is significantly better than the two other ones. Furthermore, the difference between the uniform and the optimal allocation is larger than in the case illustrated in Figure D.2.

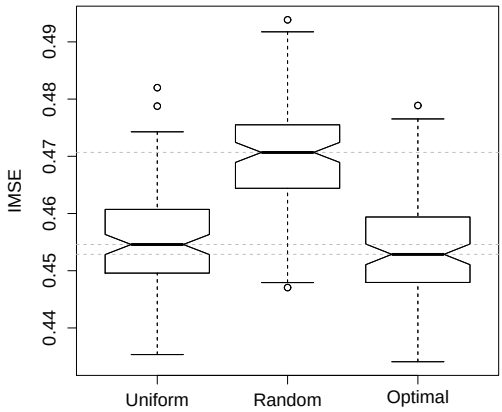
Comparison in dimension 6 with a uniform measure $\mu_{\mathrm{U}}(x)$ with large T.



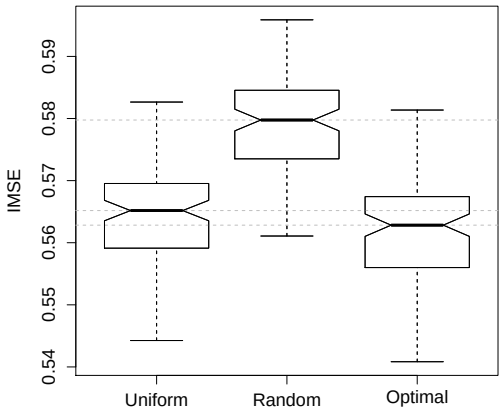
(a) $d = 6$, $T = 500$, $n = 100$, $\theta = 5.10^{-1}$
$k(.,.)$: exponential, $\eta(x) = \mu_{\mathrm{U}}(x)$



(b) $d = 6$, $T = 500$, $n = 100$, $\theta = 5.10^{-1}$
$k(.,.)$: Gaussian, $\eta(x) = \mu_{\mathrm{U}}(x)$



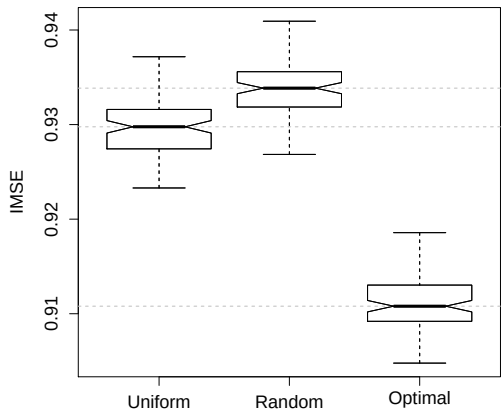
(c) $d = 6$, $T = 500$, $n = 100$, $\theta = 5.10^{-1}$
$k(.,.)$: 5/2-Matérn, $\eta(x) = \mu_{\mathrm{U}}(x)$



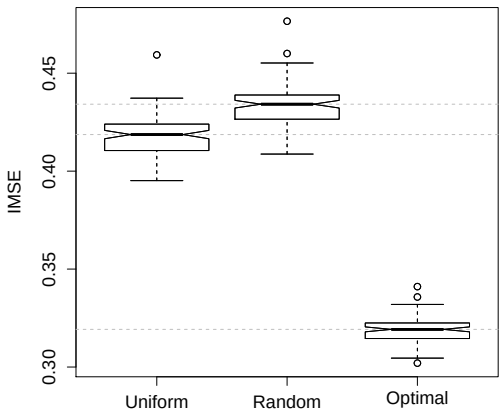
(d) $d = 6$, $T = 500$, $n = 100$, $\theta = 5.10^{-1}$
$k(.,.)$: 3/2-Matérn, $\eta(x) = \mu_{\mathrm{U}}(x)$

Figure D.4: Comparison between uniform, random and optimal allocations on an example in dimension $d = 6$ with heteroscedastic observation noise variance. 100 experimental design sets of $n = 100$ points are randomly sampled from the measure $\mu_{\mathrm{U}}(x)$. The budget is $T = 500$ and the correlation length is $\theta = 5.10^{-1}$. We note that the covariance matrix $\mathbf{K}$ is not diagonal. Though we do not respect the assumption of Proposition 7.3, we see that the suggested optimal allocation is better than the two other ones. Furthermore, the difference between the uniform and the optimal allocation decreases with the regularity of the covariance kernel. Indeed, the smallest is for the Gaussian kernel in Figure (b) and the largest is for the exponential one in Figure (a). This is due to the fact that less regular is the kernel closer we are to a diagonal matrix $\mathbf{K}$.

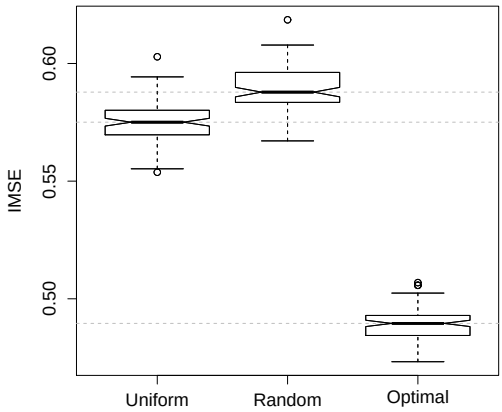
Comparison in dimension 6 with a Gaussian measure $\mu_{\mathrm{G}}(x)$ with large T.



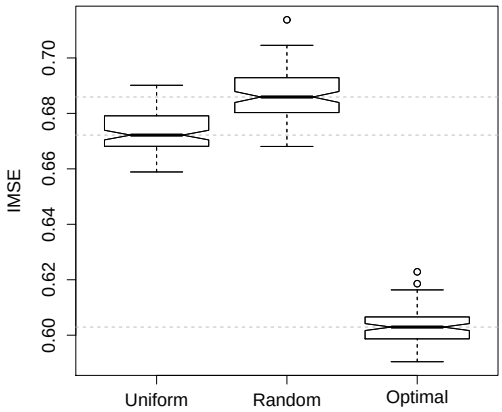
(a) $d = 6$, $T = 500$, $n = 100$, $\theta = 5.10^{-1}$
$k(.,.)$: exponential, $\eta(x) = \mu_{\mathrm{G}}(x)$



(b) $d = 6$, $T = 500$, $n = 100$, $\theta = 5.10^{-1}$
$k(.,.)$: Gaussian, $\eta(x) = \mu_{\mathrm{G}}(x)$



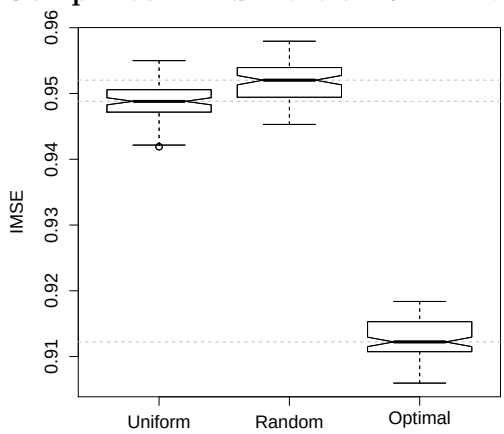
(c) $d = 6$, $T = 500$, $n = 100$, $\theta = 5.10^{-1}$
$k(.,.)$: 5/2-Matérn, $\eta(x) = \mu_{\mathrm{G}}(x)$



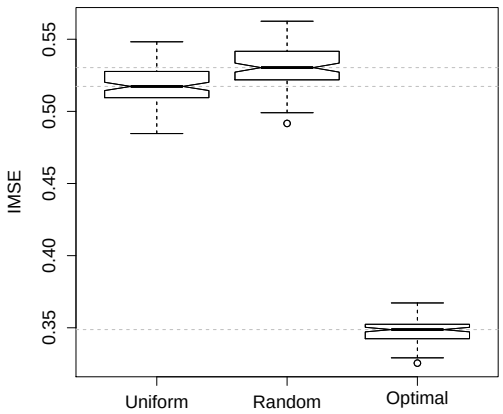
(d) $d = 6$, $T = 500$, $n = 100$, $\theta = 5.10^{-1}$
$k(.,.)$: 3/2-Matérn, $\eta(x) = \mu_{\mathrm{G}}(x)$

Figure D.5: Comparison between uniform, random and optimal allocations on an example in dimension $d = 6$ with heteroscedastic observation noise variance. 100 experimental design sets of $n = 100$ points are randomly sampled from the Gaussian measure $\mu_{\mathrm{G}}(x)$. The budget is $T = 500$ and the correlation length is $\theta = 5.10^{-1}$. We note that the covariance matrix $\mathbf{K}$ is not diagonal. Though we do not respect the assumption of Proposition 7.3, we see that the suggested optimal allocation is significantly better than the two other ones. Furthermore, the difference between the uniform and the optimal allocations is particularly important compared to the illustrations in Figures D.1, D.2, D.3 and D.4. This is explained by the fact that we use a Gaussian measure for processing the IMSE whereas in Figures D.1, D.2, D.3 and D.4 we use an uniform measure which fits with the uniform allocation. Therefore, this comparison shows that it is worth taking into account the measure of averaging to allocate the resource.

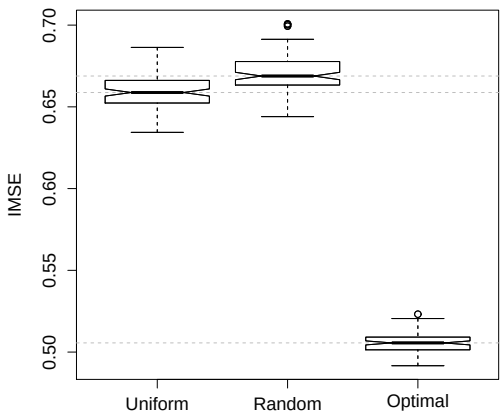
Comparison in dimension 6 with a Gaussian measure $\mu_{\mathrm{G}}(x)$ with small T.



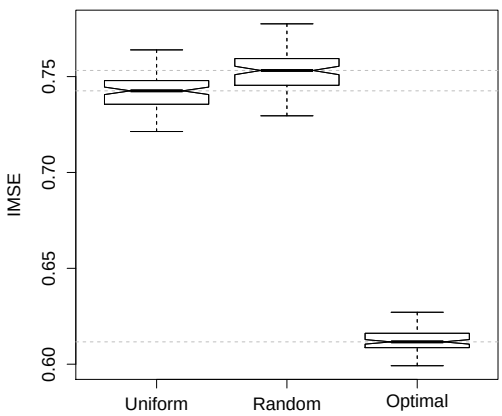
(a) $d = 6$, $T = 150$, $n = 100$, $\theta = 5.10^{-1}$
$k(.,.)$: exponential, $\eta(x) = \mu_{\mathrm{G}}(x)$



(b) $d = 6$, $T = 150$, $n = 100$, $\theta = 5.10^{-1}$
$k(.,.)$: Gaussian, $\eta(x) = \mu_{\mathrm{G}}(x)$



(c) $d = 6$, $T = 150$, $n = 100$, $\theta = 5.10^{-1}$
$k(.,.)$: 5/2-Matérn, $\eta(x) = \mu_{\mathrm{G}}(x)$



(d) $d = 6$, $T = 150$, $n = 100$, $\theta = 5.10^{-1}$
$k(.,.)$: 3/2-Matérn, $\eta(x) = \mu_{\mathrm{G}}(x)$

Figure D.6: Comparison between uniform, random and optimal allocations on an example in dimension $d = 6$ with heteroscedastic observation noise variance. 100 experimental design sets of $n = 100$ points are randomly sampled from the Gaussian measure $\mu_{\mathrm{G}}(x)$. The budget is $T = 150$ and the correlation length is $\theta = 5.10^{-1}$. We note that the covariance matrix $\mathbf{K}$ is not diagonal. Though we do not respect the assumption of Proposition 7.3, we see that the suggested optimal allocation is significantly better than the two other ones. Furthermore, the difference between the uniform and the optimal allocations is particularly important compared to the one given in the illustrations in Figures D.1, D.2, D.3 and D.4. This is explained by the fact that contrary to the uniform allocation we take into account the averaging measure into the optimal allocation. Furthermore, we see that the constraint of having one Monte-Carlo particle for each point of the design set is well handled by the suggested allocation. We highlight that its performance compared to the uniform one is even better than the one illustrated in Figure D.5.

Bibliography

[Abramowitz and Stegun, 1965] Abramowitz, M. and Stegun, I. A. (1965). *Handbook of Mathematical Functions*. Dover, New York.

[Ackley, 1987] Ackley, D. H. (1987). *A connectionist machine for genetic hillclimbing*. Kluwer Academic Publishers, Boston.

[Archer et al., 1997] Archer, G., Saltelli, A., and Sobol, I. (1997). Sensitivity measures, anova-like techniques and the use of bootstrap. *Journal of Statistical Computation and Simulation*, 58(2):99–120.

[Aronszajn, 1950] Aronszajn, N. (1950). Theory of reproducing kernels. *Trans. Amer. Math. Soc*, 68(3):337–404.

[Auer et al., 2002a] Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256.

[Auer et al., 2002b] Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002b). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77.

[Avriel, 2003] Avriel, M. (2003). *Nonlinear programming: analysis and methods*. Dover Publications.

[Bachoc, 2013] Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification.

[Baker, 1977] Baker, C. T. H. (1977). *The Numerical Treatment of Integral Equations*. Clarendon Press, Oxford.

[Bates et al., 1996] Bates, R. A., Buck, R., Riccomagno, E., and Wynn, H. (1996). Experimental design and observation for large systems. *Journal of the Royal Statistical Society, Series B*, 58 (1):77–94.

[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*, 22:773–793.

[Berger et al., 2001] Berger, J. O., De Oliveira, V., and Sans, B. (2001). Objective bayesian analysis of spatially correlated data objective bayesian analysis of spatially correlated data. *Journal of the American Statistical Association*, 96:1361–1374.

[Berlinet and Thomas-Agnan, 2004] Berlinet, A. and Thomas-Agnan, C. (2004). *Reproducing kernel Hilbert spaces in probability and statistics*. Kluwer Academic Publishers.

[Bichon et al., 2008] Bichon, B., Eldred, M., Swiler, L., Mahadevan, S., and McFarland, J. (2008). Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA journal*, 46(10):2459–2468.

[Billingsley, 1999] Billingsley, P. (1999). *Convergence of Probability Measures*. Wiley Series in Probability and Statistics, New York.

[Bishop et al., 2006] Bishop, C. M. et al. (2006). *Pattern recognition and machine learning*, volume 1. springer New York.

[Borgonovo, 2007] Borgonovo, E. (2007). A new uncertainty importance measure. *Reliability Engineering & System Safety*, 92:771–784.

[Boukouvalas and Cornford, 2009] Boukouvalas, A. and Cornford, D. (2009). Learning heteroscedastic gaussian processes for complex datasets. Technical report, Technical report, Neural Computing Research Group, Aston University, Birmingham, UK.

[Boyle and Frean, 2005] Boyle, P. and Frean, M. (2005). Dependent gaussian processes. *Advances in Neural Information Processing Systems*, 17:217–224.

[Bozzini and Rossini, 2003] Bozzini, M. and Rossini, M. (2003). Numerical differentiation of 2d functions from noisy data. *Computer and Mathematics with Applications*, 45:309–327.

[Bronski, 2003] Bronski, J. C. (2003). Asymptotics of Karhunen-Loève eigenvalues and tight constants for probability distributions of passive scalar transport. *Communications in Mathematical Physics*, 238:563–582.

[Chambers et al., 1983] Chambers, J., Cleveland, J. M., Kleiner, W. S., and Tukey, P. A. (1983). *Graphical Methods for Data Analysis*. Wadsworth & Brooks/Cole.

[Chastaing et al., 2012] Chastaing, G., Gamboa, F., and Prieur, C. (2012). Generalized hoeffding-Sobol decomposition for dependent variables -application to sensitivity analysis. *Electronic Journal of Statistics*, 6:2420–2448.

[Chilès and Delfiner, 1999] Chilès, J. and Delfiner, P. (1999). Geostatistics: modeling spatial uncertainty. *Wiley series in probability and statistics (Applied probability and statistics section)*.

[Conti and O'Hagan, 2010] Conti, S. and O'Hagan, A. (2010). Bayesian emulation of complex multi-output and dynamic computer models. *Journal of statistical planning and inference*, 140(3):640–651.

[Cook and Nachtrheim, 1980] Cook, R. D. and Nachtrheim, C. J. (1980). A comparison of algorithms for constructing exact d-optimal designs. *Technometrics*, 22(3):315–324.

[Craig et al., 1998] Craig, P. S., Goldstein, M., Seheult, A. H., and Smith, J. A. (1998). Constructing partial prior specifications for models of complex physical systems. *Applied Statistics*, 47:37–53.

[Cramer, 1999] Cramer, H. (1999). *Mathematical Methods of Statistics (PMS-9)*, volume 9. Princeton university press.

[Cramer and Leadbetter, 1967] Cramer, H. and Leadbetter, M. (1967). *Stationary And Related Stochastic Processes*. John Wiley, New York.

[Cumming and Goldstein, 2009] Cumming, J. A. and Goldstein, M. (2009). Small sample bayesian designs for complex high-dimensional models based on information gained using fast approximations. *Technometrics*, 51:377–388.

[Currin et al., 1991] Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86(416):953–963.

[Da Veiga et al., 2009] Da Veiga, S., Wahl, F., and Gamboa, F. (2009). Local polynomial estimation for sensitivity analysis on models with correlated inputs. *Technometrics*, 51(4):452–463.

[Dewettinck et al., 1999] Dewettinck, K., De Visscher, A., Deroo, L., and Huyghebaert, A. (1999). Modeling the steady-state thermodynamic operation point of top-spray fluidized bed processing. *Journal of Food Engineering*, 39:131–143.

[Diggle and Ribeiro Jr, 2002] Diggle, P. and Ribeiro Jr, P. (2002). Bayesian inference in gaussian model-based geostatistics. *Geographical and Environmental Modelling*, 6(2):129–146.

[Dubourg et al., 2011] Dubourg, V., Sudret, B., and Bourinet, J.-M. (2011). Reliability-based design optimization using kriging surrogates and subset simulation. *Structural and Multidisciplinary Optimization*, 44 - 5:673–690.

[Dubrule, 1983] Dubrule, O. (1983). Cross validation of kriging in a unique neighborhood. *Mathematical Geology*, 15:687–699.

[Dudley, 1967] Dudley, R. M. (1967). The sizes of compact subsets of hilbert space and continuity of gaussian processes. *J. Funct. Anal*, 1(3):290–330.

[Durrande, 2011] Durrande, N. (2011). *Etude de classes de noyaux adaptées à la simplification et à l'interprétation des modèles d'approximation*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint Etienne.

[Durrande et al., 2013] Durrande, N., Ginsbourger, D., Roustant, O., and Laurent, C. (2013). Anova kernels and rkhs of zero mean functions for model-based sensitivity analysis. *Journal of Multivariate Analysis*, 115:57–67.

[Elfving, 1952] Elfving, G. (1952). Optimum allocation in linear regression theory. *The Annals of Mathematical Statistics*, 23(2):255–262.

[Fang et al., 2006] Fang, K.-T., Li, R., and Sudjianto, A. (2006). *Design and Modeling for Computer Experiments*. Chapman & Hall - Computer Science and Data Analysis Series, London.

[Fedorov, 1972] Fedorov, V. V. (1972). *Theory of optimal experiments*, volume 12. Academic Press.

[Fedorov and Hackl, 1997] Fedorov, V. V. and Hackl, P. (1997). *Model-oriented design of experiments*, volume 125. Springer Verlag.

[Fernex et al., 2005] Fernex, F., Heulers, L., Jacquet, O., Miss, J., and Richet, Y. (2005). The Moret 4b monte carlo code new features to treat complex criticality systems. In *MandC International Conference on Mathematics and Computation Supercomputing, Reactor and Nuclear and Biological Application*, Avignon, France.

[Fernique, 1964] Fernique, X. (1964). Continuité des processus gaussiens. *CR Acad. Sci. Paris*, 258:6058–6060.

[Ferreira and Menegatto, 2009] Ferreira, J. and Menegatto, V. (2009). Eigenvalues of integral operators defined by smooth positive definite kernels. *Integral Equations and Operator Theory*, 64(1):61–81.

[Fisher, 1956] Fisher, R. (1956). Statistical methods and scientific inference.

[Forrester et al., 2007] Forrester, A. I. J., Sobester, A., and Keane, A. J. (2007). Multi-fidelity optimization via surrogate modelling. *Proc. R. Soc. A*, 463:3251–3269.

[Garsia, 1972] Garsia, A. (1972). Continuity properties of gaussian processes with multidimensional time parameter. In *Proc. Sixth Berkeley Symp. Math. Statist. Probab*, volume 2, pages 369–374.

[Geisser and Eddy, 1979] Geisser, S. and Eddy, W. (1979). A predictive approach to model selection. *Journal of the American Statistical Association*, 74(365):153–160.

[Geweke et al., 1991] Geweke, J. et al. (1991). *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*. Federal Reserve Bank of Minneapolis, Research Department.

[Gibbs, 1997] Gibbs, M. N. (1997). *Bayesian Gaussian Processes for Regression and Classification*. PhD thesis, Department of Physics, University of Cambridge.

[Gikhman and Skorokhod, 1974] Gikhman, I. and Skorokhod, A. (1974). The theory of stochastic processes.

[Ginsbourger et al., 2010] Ginsbourger, D., Le Riche, R., and Carraro, L. (2010). *Kriging is well-suited to parallelize optimization. In Computational Intelligence in Expensive Optimization Problems*, pages 131–162. Adaptation Learning and Optimization. Springer, Berlin.

[Gneiting et al., 2010] Gneiting, T., Kleiber, W., and Schlater, M. (2010). Matérn cross-covariance functions for multivariate random fields. *Journal of the American Statistical Association*, 105:1167–1177.

[Goldstein and Wooff, 2007] Goldstein, M. and Wooff, D. A. (2007). *Bayes Linear Statistics: Theory and Methods.* Chichester, England: Wiley.

[Goulard and Voltz, 1992] Goulard, M. and Voltz, M. (1992). Linear coregionalization model: tools for estimation and choice of cross-variogram matrix. *Mathematical Geology*, 24(3):269–286.

[Gradshteyn et al., 2007] Gradshteyn, I. S., Ryzhik, I. M., Jeffrey, A., and Zwillinger, D. (2007). *Table of integrals, series, and products.* Academic press.

[Gramacy and Lian, 2012] Gramacy, R. B. and Lian, H. (2012). Gaussian process single-index models as emulators for computer experiments. *Technometrics*, 54(1):30–41.

[Grégoire et al., 2005] Grégoire, O., Souffland, D., and Serge, G. (2005). A second order turbulence model for gaseous mixtures induced by Richtmyer-Meshkov instability. *Journal of Turbulence*, 6:1–20.

[Hartigan and Wong, 1979] Hartigan, J. A. and Wong, M. A. (1979). A k-means clustering algorithm. *Applied Statistics*, 28:100–108.

[Harville, 1974] Harville, D. (1974). Bayesian inference for variance components using only error contrasts. *Biometrika*, 61(2):383–385.

[Harville, 1977] Harville, D. (1977). Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*, 72(358):320–338.

[Harville, 1997] Harville, D. A. (1997). *Matrix Algebra from Statistician's Perspective.* Springer-Verlag, New York.

[Higdon et al., 2004] Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., and Ryne, R. D. (2004). Combining field data and computer simulation for calibration and prediction. *SIAM Journal on Scientific Computing*, 26:448–466.

[Higdon et al., 2008] Higdon, D., Nakhleh, C., Gattiker, J., and Williams, B. (2008). A bayesian calibration approach to the thermal problem. *Computer Methods in Applied Mechanics and Engineering*, 197(29):2431–2441.

[Hoeffding, 1948] Hoeffding, W. (1948). A class of statistics with asymptotically normal distribution. *The annals of Mathematical Statistics*, 19(3):293–325.

[Huang et al., 2006] Huang, D., Allen, T. T., Notz, W. I., and Miller, R. A. (2006). Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32:369–382.

[Huang and Chan, 2010] Huang, Y.-C. and Chan, K.-Y. (2010). A modified efficient global optimization algorithm for maximal reliability in a probabilistic constrained space. *Journal of Mechanical Design*, 132:061002.

[Iooss et al., 2006] Iooss, B., Van Dorpe, F., and Devictor, N. (2006). Response surfaces and sensitivity analyses for an environmental model of dose calculations. *Reliability Engineering & System Safety*, 91(10):1241–1251.

[Jacques et al., 2006] Jacques, J., Lavergne, C., and Devictor, N. (2006). Sensitivity analysis in presence of model uncertainty and correlated inputs. *Reliability Engineering and System Safety*, 91:1126–1134.

[Janon et al., 2012] Janon, A., Klein, T., Lagnoux, A., Nodet, M., and Prieur, C. (2012). Asymptotic normality and efficiency of two Sobol index estimators.

[Janon et al., 2011] Janon, A., Nodet, M., Prieur, C., et al. (2011). Uncertainties assessment in global sensitivity indices estimation from metamodels.

[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.

[Jeffreys, 1946] Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461.

[Jeffreys, 1961] Jeffreys, H. (1961). *Theory of Probability.* Oxford University Press, London.

[Jones et al., 1998] Jones, D. R., Schonlau, M., and Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Biometrika*, 13:455–492.

[Jordan and Jacobs, 1994] Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214.

[Kaelbling, 1993] Kaelbling, L. P. (1993). *Learning in embedded systems.* The MIT Press.

[Kailath, 1971] Kailath, T. (1971). Rkhs approach to detection and estimation problems–i: Deterministic signals in gaussian noise. *Information Theory, IEEE Transactions on*, 17(5):530–549.

[Karush, 1939] Karush, W. (1939). *Minima of functions of several variables with inequalities as side constraints.* PhD thesis, Master's thesis, Dept. of Mathematics, Univ. of Chicago.

[Kennedy and O'Hagan, 2000] Kennedy, M. C. and O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87:1–13.

[Kennedy and O'Hagan, 2001] Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society, Series B*, 63:425–464.

[Kiefer, 1961] Kiefer, J. (1961). Optimum designs in regression problems, ii. *The Annals of Mathematical Statistics*, pages 298–325.

[Kiefer and Wolfowitz, 1959] Kiefer, J. and Wolfowitz, J. (1959). Optimum designs in regression problems. *The Annals of Mathematical Statistics*, 30(2):271–294.

[Kimeldorf and Wahba, 1971] Kimeldorf, G. and Wahba, G. (1971). Some results on tchebycheffian spline functions. *J. Mathematical Analysis and Applications*, 33(1):82–95.

[Kleijnen and van Beers, 2004] Kleijnen, J. and van Beers, W. (2004). Application-driven sequential designs for simulation experiments: Kriging metamodelling. *Journal of the Operational Research Society*, 55:876–883.

[Kleijnen, 2012] Kleijnen, J. P. (2012). Design and analysis of monte carlo experiments. *Handbook of Computational Statistics*, pages 529–547.

[Kleijnen and Van Beers, 2005] Kleijnen, J. P. and Van Beers, W. (2005). Robustness of kriging when interpolating in random simulation with heterogeneous variances: some experiments. *European Journal of Operational Research*, 165(3):826–834.

[Koehler and Owen, 1996] Koehler, J. and Owen, A. (1996). Computer experiments. *Handbook of statistics*, 13(13):261–308.

[König, 1986] König, H. (1986). *Eigenvalue distribution of compact operators*. Birkhäuser Basel.

[Krige, 1951] Krige, D. G. (1951). A statistical approach to some basic mine valuation problems on the witwatersrand. *Technometrics*, 52:119–139.

[Kucherenko et al., 2012] Kucherenko, S., Tarantola, S., and Annoni, P. (2012). Estimation of global sensitivity indices for models with dependent variables. *Computer Physics Communications*, 183:937–946.

[Kuhn and Tucker, 1951] Kuhn, H. W. and Tucker, A. W. (1951). Nonlinear programming. In *Second Berkeley symposium on mathematical statistics and probability*, volume 1, pages 481–492.

[Laslett, 1994] Laslett, G. M. (1994). Kriging and splines: An empirical comparison of their predictive performance in some applications kriging and splines: An empirical comparison of their predictive performance in some applications. *Journal of the American Statistical Association*, 89:391–400.

[Le Gratiet, 2013] Le Gratiet, L. (2013). Bayesian analysis of hierarchical multifidelity codes. *SIAM/ASA J. Uncertainty Quantification*, 1(1):244–269.

[Lehmann and Casella, 1998] Lehmann, E. L. and Casella, G. (1998). *Theory of Point Estimation*. Springer-Verlag, New York.

[Li et al., 2010] Li, G., Rabitz, H., Yelvington, P. E., Oluwole, O., Bacon, F., E, K. C., and Schoendorf, J. (2010). Global sensitivity analysis with independent and/or correlated inputs. *Journal of Physical Chemistry A*, 114:6022–6032.

[Li et al., 2012] Li, L., Bect, J., and Vazquez, E. (2012). Bayesian subset simulation: a kriging-based subset simulation algorithm for the estimation of small probabilities of failure. *arXiv preprint arXiv:1207.1963*.

[Liu, 2001] Liu, J. S. (2001). *Monte-Carlo Strategies in Scientific Computing*. Springer-Verlag, New York.

[MacQueen, 1967] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *In: Le Cam J LM Neyman (ed) Proc. 5th Berkeley Symp. on Math. Stat. & Prob., University of California Press, Berkeley, CA*, 1:281–297.

[Mannor and Tsitsiklis, 2004] Mannor, S. and Tsitsiklis, J. N. (2004). The sample complexity of exploration in the multi-armed bandit problem. *The Journal of Machine Learning Research*, 5:623–648.

[Mara and Tarantola, 2012] Mara, T. and Tarantola, S. (2012). Variance-based sensitivity analysis of computer models with dependent inputs. *Reliability Engineering & System Safety*, 107:115–121.

[Marcus and Shepp, 1970] Marcus, M. and Shepp, L. (1970). Continuity of gaussian processes. *Trans. Amer. Math. Soc*, 151:377–391.

[Marrel et al., 2010] Marrel, A., Iooss, B., Da Veiga, S., and Ribatet, M. (2010). Global sensitivity analysis of stochastic computer models with joint metamodels. *Statistics and Computing*, pages 1–15.

[Marrel et al., 2009] Marrel, A., Iooss, B., Laurent, B., and Roustant, O. (2009). Calculations of Sobol indices for the Gaussian process metamodel. *Reliability Engineering and System Safety*, 94:742–751.

[Marseguerra et al., 2003] Marseguerra, M., Masini, R., Zio, E., and Cojazzi, G. (2003). Variance decomposition-based sensitivity analysis via neural networks. *Reliability Engineering & System Safety*, 79(2):229–238.

[Marzat et al., 2012] Marzat, J., Walter, E., and Piet-Lahanier, H. (2012). Worst-case global optimization of black-box functions through kriging and relaxation. *Journal of Global Optimization*, pages 1–21.

[Matérn, 1986] Matérn, B. (1986). *Spatial Variation*. Springer-Verlag, Berlin.

[Matheron, 1963] Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, 58:1246–1266.

[Matheron, 1969] Matheron, G. (1969). *Le krigeage Universel*. Ecole des Mines de Paris, Paris.

[McFarland et al., 2008] McFarland, J., Mahadevan, S., Romero, V., and Swiler, L. (2008). Calibration and uncertainty analysis for computer simulations with multivariate output. *AIAA journal*, 46(5):1253–1265.

[Mercer, 1909] Mercer, J. (1909). Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society A*, 209:441–458.

[Meuleau and Bourgine, 1999] Meuleau, N. and Bourgine, P. (1999). Exploration of multi-state environments: Local measures and back-propagation of uncertainty. *Machine Learning*, 35(2):117–154.

[Michalewicz, 1992] Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, New York.

[Mitchell et al., 1994] Mitchell, T., Morris, M., and Ylvisaker, D. (1994). Asymptotically optimum experimental designs for prediction of deterministic functions given derivative information. *Journal of statistical planning and inference*, 41(3):377–389.

[Mockus, 1994] Mockus, J. (1994). Application of bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4(4):347–365.

[Mockus, 2002] Mockus, J. (2002). Bayesian heuristic approach to global optimization and examples. *Journal of Global Optimization*, 22(1-4):191–203.

[Molchanov and Zuyev, 2002] Molchanov, I. and Zuyev, S. (2002). Steepest descent algorithms in a space of measures. *Statistics and Computing*, 12(2):115–123.

[Morris et al., 1993] Morris, M. D., Mitchell, T. J., and Ylvisaker, D. (1993). Bayesian design and analysis of computer experiments: use of derivatives in surface prediction. *Technometrics*, 35(3):243–255.

[Munoz Zuniga et al., 2011] Munoz Zuniga, M., Garnier, J., Remy, E., and de Rocquigny, E. (2011). Adaptative directional stratification for controlled estimation of the probability of a rare event. *Reliability Engineering and System Safety*, 96:1691–1712.

[Nash and Sutcliffe, 1970] Nash, J. and Sutcliffe, J. (1970). River flow forecasting through conceptual models part i-a discussion of principles. *Journal of hydrology*, 10(3):282–290.

[Nazarov and Nikitin, 2004] Nazarov, A. I. and Nikitin, Y. Y. (2004). Exact l_2-small ball behaviour of integrated Gaussian processes and spectral asymptotics of boundary value problems. *Probab. Theory Relat. Fields*, 129:469–494.

[Oakley, 2004] Oakley, J. (2004). Estimating percentiles of uncertain computer code outputs. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 53(1):83–93.

[Oakley and O'Hagan, 2004] Oakley, J. E. and O'Hagan, A. (2004). Probabilistic sensitivity analysis of complex models a bayesian approach. *Journal of the Royal Statitistical Society series B*, 66:part 3, 751–769.

[O'Hagan, 1998] O'Hagan, A. (1998). A Markov property for covariance structures.

[O'Hagan, 2006] O'Hagan, A. (2006). Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering & System Safety*, 91(10):1290–1300.

[Oksendal, 1998] Oksendal, B. (1998). *Stochastic differential equations*. Springer, Berlin.

[Opper and Vivarelli, 1999] Opper, M. and Vivarelli, F. (1999). General bounds on Bayes errors for regression with Gaussian processes. *Advances in Neural Information Processing Systems 11*, pages 302–308.

[Patterson and Thompson, 1971] Patterson, H. and Thompson (1971). Recovery of interblock information when block sizes are unequal. *Biometrika*, 58:545–554.

[Picard and Williams, 2013] Picard, R. and Williams, B. (2013). Rare event estimation for computer models. *The American Statistician*, 67(1):22–32.

[Picheny, 2009] Picheny, V. (2009). *Improving Accuracy and Compensating for Uncertainty in Surrogate Modeling*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint Etienne.

[Picheny et al., 2012] Picheny, V., Ginsbourger, D., Richet, Y., and Caplin, G. (2012). Quantile-based optimization of noisy computer experiments with tunable precision. *Technometrics*.

[Picheny et al., 2010] Picheny, V., Ginsbourger, D., Roustant, O., Haftka, R. T., and Nam-Ho, K. (2010). Adaptative designs of experiments for accurate approximation of a target region. *Journal of Mechanical Design*, 132:071008–1 – 071008–9.

[Pusev, 2011] Pusev, R. S. (2011). Small deviation asymptotics for Matèrn processes and fields under weighted quadratic norm. *Theory Probab. Appl.*, 55:164–172.

[Qian et al., 2009] Qian, P. Z. G., Ai, M., and Wu, C. F. J. (2009). Construction of nested space-filling designs. *The Annals of Statistics*, 37:3616–3643.

[Qian and Wu, 2008] Qian, P. Z. G. and Wu, C. F. J. (2008). Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. *Technometrics*, 50:192–204.

[Rao, 1945] Rao, R. C. (1945). Information and accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37(3):81–91.

[Rasmussen and Williams, 2006] Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge.

[Ritter, 2000a] Ritter, K. (2000a). Almost optimal differentiation using noisy data. *Journal of approximation theory*, 86:293–309.

[Ritter, 2000b] Ritter, K. (2000b). *Average-Case Analysis of Numerical Problems*. Springer Verlag, Berlin.

[Robbins, 1952] Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535.

[Robert, 2007] Robert, C. (2007). *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer, New York.

[Robert and Casella, 2004] Robert, C. and Casella, G. (2004). *Monte Carlo statistical methods (2nd edition)*. Springer Series in Statistics, Springer Verlag, New York.

[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. *Journal of Statistical Software*, 51(1):1–55.

[Sacks et al., 1989a] Sacks, J., Schiller, S. B., and Welch, W. J. (1989a). Designs for computer experiments. *Technometrics*, 31(1):41–47.

[Sacks et al., 1989b] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989b). Design and analysis of computer experiments. *Statistical Science*, 4:409–423.

[Sacks and Ylvisaker, 1981] Sacks, J. and Ylvisaker, D. (1981). Variance estimation for approximately linear models. *Series Statistics*, 12:147–162.

[Saltelli et al., 2000] Saltelli, A., Chan, K., and M., S. E. (2000). *Sensitivity Analysis*. Wiley Series in Probability and Statistics, England.

[Santner et al., 2003] Santner, T. J., Williams, B. J., and Notz, W. I. (2003). *The Design and Analysis of Computer Experiments*. Springer, New York.

[Schoeneberg, 1938] Schoeneberg, I. J. (1938). metric space and positive definite function. *Trans. American Mathematical Society*, 44(3):522–536.

[Schonlau, 1998] Schonlau, M. (1998). *Computer experiments and global optimization*. University of Waterloo.

[Shanno, 1970] Shanno, D. F. (1970). Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, 24:647–656.

[Shewry and Wynn, 1987] Shewry, M. C. and Wynn, H. P. (1987). Maximum entropy sampling. *Journal of Applied Statistics*, 14:165–170.

[Sobol et al., 2007] Sobol, I., Tarantola, S., Gatelli, D., Kucherenko, S., and Mauntz, W. (2007). Estimating the approximation error when fixing unessential factors in global sensitivity analysis. *Reliability Engineering & System Safety*, 92(7):957–960.

[Sobol, 1993] Sobol, I. M. (1993). Sensitivity estimates for non linear mathematical models. *Mathematical Modelling and Computational Experiments*, 1:407–414.

[Sollich and Halees, 2002] Sollich, P. and Halees, A. (2002). Learning curves for Gaussian process regression: approximations and bounds. *Neural computation*, 14:1393–1428.

[Stein, 1987] Stein, M. L. (1987). Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29:143–151.

[Stein, 1999] Stein, M. L. (1999). *Interpolation of Spatial Data*. Springer Series in Statistics, New York.

[Stocki, 2005] Stocki, R. (2005). A method to improve design reliability using optimal latin hypercube sampling. *Computer Assisted Mechanics and Engineering Sciences*, 12:87–105.

[Todor, 2006] Todor, R. A. (2006). Robust eigenvalue computation for smoothing operators. *SIAM J. Numer. Anal.*, 44:865–878.

[Ueda and Ghahramani, 2002] Ueda, N. and Ghahramani, Z. (2002). Bayesian model search for mixture models based on optimizing variational bounds. *Neural Networks*, 15(10):1223–1242.

[Uhlenbeck and Ornstein, 1930] Uhlenbeck, G. E. and Ornstein, L. S. (1930). On the theory of the brownian motion. *Physical Review*, 36(5):823.

[Valdebenito and Schuëller, 2010] Valdebenito, M. A. and Schuëller, G. I. (2010). A survey on approaches for reliability-based optimization. *Structural and Multidisciplinary Optimization*, 42(5):645–663.

[van Beers and Kleijnen, 2008] van Beers, W. and Kleijnen, J. (2008). Customized sequential designs for random simulation experiments: Kriging metamodelling and bootstrapping. *European journal of operational research*, 186:1099–1113.

[van der Vaart, 1998] van der Vaart, A. W. (1998). *Asymptotic Statistics*. Cambridge University Press, New-York.

[Van Oijen et al., 2005] Van Oijen, M., Rougier, J., and Smith, R. (2005). Bayesian calibration of process-based forest models: bridging the gap between models and data. *Tree Physiology*, 25(7):915–927.

[Vazquez, 2005] Vazquez, E. (2005). *Modélisation comportementale de systèmes non-linéaires multivariables par méthodes à noyaux et applications.* PhD thesis, Université Paris Sud - Paris XI.

[Vazquez and Bect, 2010] Vazquez, E. and Bect, J. (2010). Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and Inference*, 140(11):3088–3095.

[Vermorel and Mohri, 2005] Vermorel, J. and Mohri, M. (2005). Multi-armed bandit algorithms and empirical evaluation. In *Machine Learning: ECML 2005*, pages 437–448. Springer.

[Villemonteix et al., 2009] Villemonteix, J., Vazquez, E., and Walter, E. (2009). An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509–534.

[Wackernagel, 2003] Wackernagel, H. (2003). *Multivariate Geostatistics.* Springer-Verlag, Berlin.

[Wahba, 1990] Wahba, G. (1990). *Spline models for observational data*, volume 59. Society for Industrial and Applied Mathematics.

[Waterhouse et al., 1996] Waterhouse, S., MacKay, D., and Robinson, T. (1996). Bayesian methods for mixtures of experts. *Advances in neural information processing systems*, pages 351–357.

[Watkins, 1989] Watkins, C. J. C. H. (1989). *Learning from delayed rewards.* PhD thesis, University of Cambridge.

[Wilkinson, 2010] Wilkinson, R. D. (2010). Bayesian calibration of expensive multivariate computer experiments. *Large-Scale Inverse Problems and Quantification of Uncertainty, Ser. Comput. Stat., edited by LT Biegler et al*, pages 195–216.

[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*, 10(4):1133–1152.

[Williams and Vivarelli, 2000] Williams, C. K. I. and Vivarelli, F. (2000). Upper and lower bounds on the learning curve for Gaussian processes. *Machine Learning*, 40:77–102.

[Wu, 1978] Wu, C.-F. (1978). Some algorithmic aspects of the theory of optimal designs. *The Annals of Statistics*, pages 1286–1301.

[Wyatt, 1998] Wyatt, J. (1998). Exploration and inference in learning from reinforcement.

[Xian, 2001] Xian, L. (2001). Finding global minima with a computable filled function. *Journal of Global Optimization*, 19:191–204.

[Xiong and Qian, 2012] Xiong, S. and Qian, Peter Z. G.and Jeff Wu, C. F. (2012). Sequential design and analysis of high-accuracy and low-accuracy computer codes. *Technometrics*, DOI:10.1080/00401706.2012.723572.

[Yin et al., 2011] Yin, J., Ng, S., and Ng, K. (2011). Kriging metamodel with modified nugget-effect: The heteroscedastic variance case. *Computers & Industrial Engineering*, 61(3):760–777.

[Zhang and Wang, 2009] Zhang, H. and Wang, Y. (2009). Kriging and cross-validation for massive spatial data. *Environmetrics*, 21:290–304.

[Zhu et al., 1998] Zhu, H., Williams, C. K., Rohwer, R., and Morciniec, M. (1998). *Gaussian regression and optimal finite dimensional linear models.* Springer-Verlag, Berlin.

List of contributions

Articles

1 Le Gratiet, L. (2013), Bayesian analysis of hierarchical multifidelity codes, *SIAM/ASA J. Uncertainty Quantification* **1-1**, pp. 244-269.

2 Le Gratiet, L. and Garnier, J. (2012), Recursive cokriging model for Design of Computer experiments with multiple levels of fidelity, *submitted to International Journal of Uncertainty Quantification*.

3 Le Gratiet, L. & Cannamela C. (2012), Cokriging-based sequential design strategies using fast cross-validation techniques for multi-fidelity computer codes. *submitted to TECHNOMETRICS.*

4 Le Gratiet, L., Cannamela, C. and Iooss, B. (2013). A Bayesian approach for global sensitivity analysis of (multi-fidelity) computer codes. *submitted to SIAM/ASA J. Uncertainty Quantification.*

5 Le Gratiet, L. and Garnier, J. (2012) Regularity dependence of the rate of convergence of the learning curve for Gaussian process regression. *Submitted to Journal of Machine Learning - Springer.*

6 Le Gratiet, L. (2013). Asymptotic normality of a Sobol index estimator in Gaussian process regression framework. *Submitted to ESAIM probability and statistics.*

Package

Le Gratiet, L. (2012), MuFiCokriging : Multi-Fidelity Cokriging models, *CRAN - Package MuFiCokriging*

Résumé

Cette thèse porte sur l'approximation par processus gaussiens d'un code de calcul qui peut être exécuté à différents niveaux de précision. L'objectif est d'améliorer les prédictions d'un méta-modèle d'un code complexe en utilisant des approximations rapides de celui-ci. Une nouvelle formulation d'une méthode basée sur un modèle de co-krigeage est proposée. En particulier, cette formulation permet de simplifier numériquement la méthode et d'obtenir des expressions analytiques des moyenne et variance de co-krigeage universel. Ceci est une avancée importante qui permet d'utiliser ces modèles aisément en pratique. Des méthodes de validation croisée rapides, de planification d'expériences séquentielle et d'analyse de sensibilité ont également été étendues au cadre du co-krigeage multi-fidélité.

Ensuite, la thèse étudie une conjecture sur la dépendance de la courbe d'apprentissage (c'est à dire le taux de décroissance de l'erreur quadratique moyenne) par rapport à la régularité de la fonction à approcher. Une preuve dans un cadre général (qui comprend les modèles classiques de régression par processus gaussiens avec noyaux stationnaires) a été obtenue, tandis que les preuves précédentes ne sont valides que pour des noyaux dégénérés (c'est à dire quand le processus est de dimension finie). Ce résultat permet d'aborder des questions pratiques telles que l'allocation optimale du budget de temps de calcul entre les différents niveaux de codes dans le cadre multi-fidélité.

Mots clés : Codes de calcul multi-fidélité, Régression par processus gaussien, Co-krigeage, Planification séquentielle, Analyse de sensibilité, Courbe d'apprentissage.

Abstract

This work is on Gaussian-process based approximation of a code which can be run at different levels of accuracy. The goal is to improve the predictions of a surrogate model of a complex computer code using fast approximations of it. A new formulation of a co-kriging based method has been proposed. In particular this formulation allows for fast implementation and for closed-form expressions for the predictive mean and variance for universal co-kriging in the multi-fidelity framework, which is a breakthrough as it really allows for the practical application of such a method in real cases. Furthermore, fast cross validation, sequential experimental design and sensitivity analysis methods have been extended to the multi-fidelity co-kriging framework.

This thesis also deals with a conjecture about the dependence of the learning curve (ie the decay rate of the mean square error) with respect to the smoothness of the underlying function. A proof in a fairly general situation (which includes the classical models of Gaussian-process based metamodels with stationary covariance functions) has been obtained while the previous proofs hold only for degenerate kernels (ie when the process is in fact finite-dimensional). This result allows for addressing rigorously practical questions such as the optimal allocation of the budget between different levels of codes in the multi-fidelity framework.

Keywords : Multi-fidelity computer codes, Gaussian process regression, Co-kriging, Sequential design, Sensitivity analysis, Learning curve.