Fast uncertainty reduction strategies relying on Gaussian process models

Clément Chevalier

Fast uncertainty reduction strategies relying on Gaussian process models

A thesis submitted by

Clément Chevalier

for the degree of doctor of philosophy

Affiliations:

Institute of mathematical statistics and actuarial science,
University of Bern

Institut de Radioprotection et de Sûreté Nucléaire (IRSN)

The ReDICE Consortium

Ph.D. advisors: David Ginsbourger (UNIBE), Yann Richet (IRSN)

defended publicly on September 18^{th}, 2013 with a jury composed of:

Prof. Ilya Molchanov	University of Bern	President
Dr. David Ginsbourger	University of Bern	Examiner
Prof. Lutz Dümbgen	University of Bern	Examiner
Prof. Hans Rudolf Künsch	ETH Zurich	Referee and Examiner

Fast uncertainty reduction strategies relying on Gaussian process models

Inauguraldissertation

der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

Clément Chevalier

von Frankreich

Leiter der Arbeit:

Dr. David Ginsbourger

und

Prof. Dr. Lutz Dümbgen

Institut für mathematische Statistik und Versicherungslehre
der Universität Bern

Von der Philosophisch-naturwissenschaftlichen Fakultät angenommen.

Bern, 18. September 2013

Der Dekan

Prof. Dr. Silvio Decurtins

To my daughter, Julie ...

Acknowledgements

A bit less than three years after having left Paris to start new challenges, this work is coming to an end...

The achievement of this Ph.D. thesis was made possible thanks to the constant support of my Ph.D. supervisors, David Ginsbourger and Prof. Lutz Dümbgen at the University of Bern, and Yann Richet at the French institute of nuclear safety (IRSN).
I would also like to warmly thank Prof. Hans Rudolf Künsch for accepting to serve both as referee of this manuscript and examiner in my Ph.D. committee, and Prof. Ilya Molchanov for serving as president of this committee.

This work could not have been realized alone, and my first thoughts go to my two Ph.D. advisors, David Ginsbourger and Yann Richet. David gave me the chance to come to Bern and start working under his kind guidance. He even helped me to handle my installation in Switzerland with my wife. Yann was the very first person to contact me to discuss about this Ph.D. offer and granted me his confidence, despite my "unusual" profile (see, my CV at the end of this manuscript !). He also warmly welcomed me at the IRSN in Fontenay-aux-Roses during the first months of this Ph.D., and started the advisory with David in a clever and efficient way, enabling me to understand the concrete scientific and operational challenges of this Ph.D. subject. I must say that both David and Yann transmitted me a contagious motivation and that working with them has been a chance, a pleasure, and an honor.

My genuine gratitude goes to the professors at the Ecole des Mines de Saint-Etienne (EMSE) whom I first contacted to apply for a Ph.D.

Profs. Olivier Roustant, Rodolphe Le Riche and Xavier Bay where all extremely helpful. They advised me, and actually enabled me to get in contact with Yann and David. My application was also kindly supported by the Laboratoire de Statistique Théorique et Appliquée (LSTA) of the University Paris VI. In particular, although I had obtained my Master Degree five long years ago, Prof. Paul Deheuvels agreed to write a kind recommendation letter for my application. I also received help, encouragements and advices from Profs. Denis Bosq and Annick Valibouze, together with the caring assistance of the secretary of the LSTA, Louise Lamart.

Throughout these three years, I had the opportunity to meet and have fruitful discussions with many people. I first would like to show my gratitude to my co-authors, David Ginsbourger and Ilya Molchanov from Bern, Yann Richet (IRSN), Julien Bect and Emmanuel Vazquez (SUPELEC), Victor Picheny (INRA), Xavier Emery (University of Chile). In particular, successful collaborations were made with Julien Bect when he visited David and myself in Bern. These collaborations and the work which followed are the source of an important part of the contribution of this Ph.D.
A fruitful collaboration with the Oeschger Centre for Climate Change Research (in Bern) also started during these three years. I would like to warmly thank Prof. Olivia Romppainen-Martius and Irina Mahlstein — whose work with David and myself led to my actual first publication — and Christoph Raible who coordinated the first interactions.

During this Ph.D., I handled, together with David, the coordination of the ReDice consortium which gathers academic and industrial partners around advanced mathematical methods for the design and analysis of computer experiments. This has been a great opportunity to chat, learn and work with researchers like Bertrand Iooss and Geraud Blatman (EDF), Jean Baccou, Frédéric Péralès, Fabrice Fouet, Grégory Caplin and Yann Richet (IRSN), Mickaële Le Ravalec-Dupin, Mathieu Feraille and Sébastien Da Veiga (IFPEN), Jean Giorla and Jean-Marc

Martinez (CEA), Frédéric Mercier and Yves Tourbier (Renault). There is still a lot of ongoing work with these people and I hope that these collaborations will continue in the future.
A precious help and advices on my work have been given by the academic partners of the consortium, namely: Olivier Roustant and Xavier Bay (EMSE), Céline Helbert (INRIA - Ecole Centrale Lyon); and by Profs. Fabrice Gamboa (Institute of Mathematics, Toulouse) and Luc Pronzato (I3S laboratory, University of Nice Sophia Antipolis). My thanks also go to Yves Deville for the past (and future) help on the rigorous and user-friendly programming of R packages, and to the Ph.D. students and Postdocs met in the context of ReDice (and beyond): Nicolas Durande, Miguel Munoz Zuniga, Federico Zertuche, Mickaël Binois, François Bachoc, Loic Le-Gratiet, Jana Fruth, Momchil Ivanov... and those who I might forget.

In addition to the great scientific framework provided by the professors and researcher in the Redice consortium, I need to point out that I received a fantastic welcome at the IRSN during the first months of my Ph.D., and kind feedbacks regarding my work or presentations. I would like to thank Eric Letang, Joachim Miss, Isabelle Duhamel, Gregory Caplin, and the whole team of engineers and researchers at the IRSN SEC and BERAC who offered me a good work environment, where I could feel at home.
In Bern, I had the opportunity to meet Professors, and other Ph.D. students. Prolific scientific discussions, which contributed to this manuscript, were offered to me by Profs. Lutz Dümbgen and Ilya Molchanov. I could also enjoy animated games of "töggele" (table football) with the members of the Institute of Mathematical Statistics and Actuarial Science, like Kaspar "Kasparov" Stucki, Niki Zumbrunnen, Christoph Kopp, Jasmin Wandel, Benjamin "Benny" Baumgartner, Petro Kolesnyk, Bernhard Freiermuth, Martin "Mario" Bigler, Dominic "Schuhmissette" Schuhmacher, Michael Mayer, Bastien "f-null" Rospopov, and many others. I also would like to thank Qiyu Li who shared her office

with me during these three years and had occasionally to bear my long and loud discussions in French with David. In addition, I would like to point out that the welcome in Bern has also been very nice. This would not have been possible without the kindness of the secretaries of the institute, Andrea Fraefel, Barbla Fraefel, Andrea Hitz and Pia Weingart, who were always smiling and helpful whenever I needed something.

During these three years, I had the opportunity to do teaching and assistantship at both the University of Bern and the Ecole des Mines de Saint-Etienne. I would like to thank Prof. Lutz Dümbgen for entrusting me the assistantship in the lecture of Multivariate Statistics, in Bern. I also thank Michel Piot and David Ginsbourger who handled respectively the lecture of Statistics in Climate Sciences and Optimization methods. The assistantship for all these lectures allowed me to strengthen my scientific background and to have a nice first experience in teaching. In Saint-Etienne, I warmly thank Xavier Bay and Olivier Roustant for allowing me to participate in the lecture of (quantitative) portfolio management.

The scientific interactions in this Ph.D. have not only been performed with Bern or with French academics or industrials. I would like to thank the team of the Stochastic Hydrogeogy Group of the University of Neuchâtel (in particular, Prof. Philippe Renard, Julien Straubhaar, Guillaume Pirot, Emanuel Huber, and many others) for the coorganisation of our workshop on Monte-Carlo methods. Though I don't claim to bring new contributions in this topic, this work, together with the kind help of Julien Cornebise (mathematical consultant, former Ph.D. student at Paris VI), helped to understand and use some useful recent methods. In this workshop, I also appreciated the scientific and informal chats with Laureline Josset, from the University of Lausanne.

Finally I must point out that my change of life was constantly and widely supported by family. My parents, Murielle and Patrick, my sister Claire, my brother Raphaël to whom I owe a lot. I hope we

will continue enjoying beautiful moments, as a family, in Schiltigheim, Kertoubiba or anywhere else. My last though are obviously for my delightful wife, Elodie, to whom I have been married for almost seven years now. Though it is difficult to describe my feelings with the right words, Elodie has always been the most supportive and gentle wife that a husband can possibly dream of. She left her job, colleagues and friends in Paris to come with me in Switzerland, and gave me two month ago a lovely daughter, Julie.

Clément Chevalier, August 20^{th} 2013.

Abstract

This work deals with sequential and batch-sequential evaluation strategies of real-valued functions under limited evaluation budget, using Gaussian process models. Optimal Stepwise Uncertainty Reduction (SUR) strategies are investigated for two different problems, motivated by real test cases in nuclear safety. First we consider the problem of identifying the excursion set above a given threshold T of a real-valued function f. Then we study the question of finding the set of "safe controlled configurations", i.e. the set of controlled inputs where the function remains below T, whatever the value of some others non-controlled inputs. New SUR strategies are presented, together with efficient procedures and formulas to compute and use them in real-world applications. The use of fast formulas to recalculate quickly the posterior mean or covariance function of a Gaussian process (referred to as the "kriging update formulas") does not only provide substantial computational savings. It is also one of the key tools to derive closed-form formulas enabling a practical use of computationally-intensive sampling strategies. A contribution in batch-sequential optimization (with the multi-points Expected Improvement) is also presented.

Contents

Contents viii

List of Figures xi

Introduction 1

I General context 4

1 Motivations, Gaussian processes (GPs) 5

1.1 Context 5

1.2 Gaussian process, standard definition and properties 8

1.2.1 Gaussian random variable 8

1.2.2 Gaussian vector 9

1.2.3 Gaussian process 11

1.3 Advanced properties on Gaussian vectors 12

1.3.1 Tallis formulas for the expectation of the maximum 12

1.3.2 Updating the mean and covariance function of a Gaussian process 14

2 Kriging 16

2.1 Kriging basics 16

2.1.1 Simple kriging 18

2.1.2 Universal kriging 19

2.1.3 Kriging with the Gaussian assumption 21

2.2 The Kriging update equations 22

2.3 Global Optimization using kriging, the Multi-points Expected Improvement . . . 24

II The Stepwise Uncertainty Reduction paradigm 29

3 Stepwise Uncertainty Reduction (SUR) 30
3.1 Introduction . . . 30
3.2 k-steps lookahead optimal SUR strategies . . . 32
3.3 The example of the Expected Improvement . . . 34
3.4 SUR strategies in inversion . . . 35

4 Contribution in SUR strategies for inversion 38
4.1 Fast computation of batch-sequential existing SUR strategies . . . 38
4.2 New SUR strategies using notions from random set theory . . . 45
4.3 R programming: KrigInv package and auxiliary problems . . . 51
4.3.1 KrigInv . . . 51
4.3.2 Numerical integration, toward the Sequential Monte-Carlo sampler . . . 52

III Contribution in robust inversion 57

5 Motivations in Nuclear Safety 58
5.1 Introduction to Nuclear Criticality Safety . . . 58
5.2 Nuclear Safety and inversion . . . 62
5.3 Nuclear Safety and robust inversion . . . 65

6 SUR criterion for robust inversion 68
6.1 Optimal SUR criteria . . . 68
6.2 A first approximation of the optimal criteria . . . 70
6.2.1 An approximation of the uncertainty leading to a criterion . . . 71
6.2.2 Computing the criterion . . . 72
6.2.3 Application to the test-case . . . 75

Appendix A: Kriging update formulas 81

Appendix B: Explicit formulas for the Multi-points Expected Improvement 86

Appendix C: Fast computation of batch-sequential SUR strategies for inversion 98

Appendix D: New SUR strategies for inversion using random set theory 123

Appendix E: A tutorial for the KrigInv R package 132

Appendix F: Ongoing work in robust inversion 158

F.1 A new SUR criterion based on fast Gaussian process conditional simulation 158

F.1.1 Updating Gaussian process realizations 159

F.1.2 Uncertainty quantification and SUR criteria 160

F.1.3 Application to the test-case 164

F.2 Bonferroni bounds for the exceedance probability 167

References 169

Curriculum Vitae 178

List of Figures

2.1 Kriging mean, standard-deviation and covariance functions for two types of kriging: Simple kriging and Universal kriging. 21

2.2 2 iterations of a batch-sequential optimization strategy using the q-EI, for $q = 2$, on a $1d$-function. Left: kriging mean (blue dotted line) and confidence intervals. Right: function q-EI($\mathbf{x}^{(2)}$) where $\mathbf{x}^{(2)}$ is a batch of two points. At each iteration, the batch evaluated is the black triangle on the right plots. 28

4.1 Left: excursion probability after 10 evaluation of a 2d test function. Middle: $\widetilde{J}_n(\mathbf{x}^{(q)})$ function, with $q = 1$ and its optimum (black triangle). Right: $J_n(\mathbf{x}^{(q)})$ function and its optimum. 42

4.2 Nine conditional realizations of the random set $\mathbf{\Gamma}$. 43

4.3 Left: excursion set of the Branin function (multiplied by a factor -1) when $D = [-10, \infty)$. Right: excursion probability function $p_n(\cdot)$ obtained from $n = 10$ observations and Vorob'ev expectation. . 47

4.4 Symmetrical difference $\mathbf{\Gamma} \Delta Q_{n,T_n}$ for nine conditional realizations of $\mathbf{\Gamma}$. 48

4.5 Left: Excursion probability and Vorob'ev Expectation after five iterations (with $q = 4$) of the batch-sequential criterion which aims at reducing the Vorob'ev deviation. Right: Five iterations of the $\widetilde{J}_n$ criterion presented in Section 4.1. In the KrigInv package, the latter criterion is called "sur". Grey circles are the newly evaluated locations. Numbers correspond to iteration number where these locations were chosen. 51

4.6 Function $p_n(\mathbf{x})(1 - p_n(\mathbf{x}))$ at different iterations of the inversion. . . 54

4.7 $p_n(\mathbf{x})(1-p_n(\mathbf{x}))$ function at different iterations of the inversion. The function is plotted together with the sample of integration points used to calculate the SUR criterion (Equation (4.14)) at each iteration. This sample is obtained using a Sequential Monte-Carlo sampler. 56

5.1 Operational/functional views of criticality parametric calculation. . 61
5.2 Contour lines of the function $\mathrm{k_{eff}} = f(\mathrm{MassePu}, \mathrm{logConcPu})$ with the excursion set (in white) of $\mathrm{k_{eff}}$ corresponding to a threshold $T = 0.95$. This approximations is obtained by computing the kriging mean function from 300 evaluations on a space-filling design of experiments. 63
5.3 Plot of the function $p_n(\mathbf{x}) = P(\mathbf{x} \in \mathbf{\Gamma}|\mathcal{A}_n)$ after n evaluations of the simulator. The triangles are the ten points of the initial design. The squares are the points sampled using the $\widetilde{J}_n$ criterion. Areas in black correspond to $p_n(\mathbf{x}) \approx 0$ and areas in white correspond to $p_n(\mathbf{x}) \approx 1$. The dotted line indicates the true excursion set. 64
5.4 Evolution of $\widetilde{H}_n$ during the sequential sampling strategy. 65
5.5 Mass-Geometry criticality system, depending on three parameters: radius, mass and concentration of fissile mass (in orange), inside a cylindrical container. 66
5.6 Set $\Gamma_c^\star$ (in white) obtained from evaluations of the $\mathrm{k_{eff}}$ on a grid. . . 67

6.1 Plot of the function $\widehat{p_n}(\mathbf{x}_c)$ after n evaluations of the simulator. The triangles are the (projected) 30 points of the initial design. The squares are the points sampled using the $\widehat{J}_n$ criterion. Areas in black correspond to $\widehat{p_n} \approx 0$ and areas in white correspond to $\widehat{p_n} \approx 1$. 77
6.2 Evolution of the uncertainty $\widehat{H}_n$ during the sequential sampling strategy. 78

F.3 Plot of the function $\widehat{p_n}(\mathbf{x}_c)$ of Equation (F.4) after n evaluations of the simulator. The triangles are the (projected) 30 points of the initial design. Areas in black correspond to $\widehat{p_n} \approx 0$ and areas in white correspond to $\widehat{p_n} \approx 1$. 166

Introduction

Computer codes with prohibitive evaluation times and costs play an important role in many engineering applications as they are more and more used to analyse and design complete systems, and often eventually influence decision making processes. The high complexity of these simulators motivated the use of Design and Analysis of Computer Experiments [Santner et al., 2003], through the introduction of simplified "surrogate" models, used to predict the output of the computer codes, relying on already available results, and also to efficiently guide further evaluations so as to answer various potential motivating problems.

Our work started in this context, motivated by real tests cases in nuclear safety, on an expensive computer code. The present thesis deals with sequential and batch-sequential evaluation strategies of function under limited evaluation budget using Gaussian process models (a.k.a *kriging*). While the main contributions of the Ph.D. work are summarized in Chapters 1 to 4 and 6, the Chapters are complemented by 5 articles appended to the thesis. As this work has already been properly done in the past (see, e.g. Ginsbourger [2009]; Rasmussen and Williams [2006]), this manuscript does not aim at giving a broad picture of different metamodeling techniques. In the first Chapter, a focus is put on the presentation of the main notions coming into play in the forthcoming contributions. Important notions where we did not contribute (like the choice of the covariance kernel in a kriging metamodel) are voluntarily omitted. As each Chapter is constantly referring to the results seen in the previous Chapters, we suggest to read this manuscript in the order where it is presented.

The manuscript is organized as follows:

- Chapter 1 introduces briefly the motivation of Gaussian process (GP) modelling of expensive-to-evaluate functions. After some quick reminders on definitions (Gaussian random variable, Gaussian vector, Gaussian process) some advanced properties, used in the contributions, are highlighted.

- Chapter 2 recalls important notions on kriging and presents two contributions (constituting Appendix A and B, respectively). The first contribution on kriging update formulas is of particular importance in this manuscript, as it is used to obtain many closed-form formulas. The second contribution, in kriging-based optimization, focuses on the derivation of a closed-form formula for the multi-points Expected Improvement, which can be seen as a generalization to batch-sequential settings of the Expected Improvement (EI) criterion [Mockus et al., 1978], later popularized with the EGO algorithm of Jones et al. [1998].

- Chapter 3 details the concept of Stepwise Uncertainty Reduction [Fleuret and Geman, 1999], used to derive optimal k-steps lookahead sampling strategies for various problems such as probability of failure estimation in reliability analysis. The ideas presented in this Chapter are recent, but not new. They are detailed for two reasons. First it is important from a theoretical point of view to define the notion of optimal function evaluation strategy in the Gaussian process modelling framework. Such strategies are proposed in the next Chapters for variants of the problems usually addressed in SUR strategies. Second, we wish to stress the computational challenges involved in the computation of SUR criteria and strategies, justifying at once the restriction to 1-step lookahead strategies and the energy spent on obtaining closed-form expressions for SUR criteria.

- Chapter 4 summarizes the contribution of this thesis for inversion problems. Three articles (see, Appendix C, D and E) are summarized and explained. Although the reader is invited to read these articles, the Chapter is meant to give a synthetic and relatively quick overview of the main contributions presented in these articles.

- Chapter 5 presents motivations in nuclear safety. Test-cases are presented

and SUR strategies are applied.

- Finally, Chapter 6 details the so-called problem of "robust inversion" which can be seen as an inversion problem in a context where some parameters are non-controlled environment parameters. New SUR strategies are presented to deal with this problem. As in inversion, the practical implementation of these strategies mainly rely on the Kriging update formulas. As a complement to Chapter 6, some promising ongoing results on an alternative robust inversion approach are presented in Appendix F.

This thesis contains 5 articles, which are sent in Appendix A to E:

- C. Chevalier, D. Ginsbourger, and X. Emery. Corrected kriging update formulae for batch-sequential data assimilation. *Proceedings of the IAMG2013 conference, Madrid*, 2013. URL http://arxiv.org/abs/1203.6452

- C. Chevalier and D. Ginsbourger. Fast computation of the multi-points Expected Improvement with applications in batch selection. *Proceedings of the LION7 conference, Lecture Notes in Computer Science*, 2013.

- C. Chevalier, J. Bect, D. Ginsbourger, E. Vazquez, V. Picheny, and Y. Richet. Fast parallel kriging-based Stepwise Uncertainty Reduction with application to the identification of an excursion set. *Accepted with minor revision to Technometrics*, 2013. URL http://hal.inria.fr/hal-00641108/en

- C. Chevalier, D. Ginsbouger, J. Bect, and I. Molchanov. Estimating and quantifying uncertainties on level sets using the Vorobev expectation and deviation with gaussian process models. *mODa 10, Advances in Model-Oriented Design and Analysis, Physica-Verlag HD*, 2013.

- C. Chevalier, V. Picheny, and D. Ginsbourger. Kriginv: An efficient and user-friendly implementation of batch-sequential inversion strategies based on kriging. *Computational Statistics & Data Analysis*, 2013.
doi: http://dx.doi.org/10.1016/j.csda.2013.03.008

Part I

General context

Chapter 1

Motivations, Gaussian processes (GPs)

1.1 Context

Engineering applications in nuclear safety, meteorology, finance, military domain, telecommunications, oil exploration, crashworthiness and many other domains have been increasingly relying on numerical simulations. The fast improvement of computer capacities, especially over the last three decades, gave birth to the so called "simulators", "black-box functions" or "computer codes", which aim at describing a physical phenomenon. Simulators can be powerful tools to avoid performing long, expensive, and/or dangerous (if not impossible) real physical experiments. For instance, in car crashworthiness, one may be interested by the outputs of a black-box simulating the physics of the crash instead of having to crash a real car. In nuclear safety, needless to say that it might be preferable to have a black-box predicting whether fissile material in a burning truck may explode or not, rather than performing the true experiment.

It is common nowadays to base important decisions on the informations (the outputs) provided by these simulators, which naturally raises the crucial question of the correctness of such outputs. When the simulator is a finite-element or a Monte-Carlo code its accuracy can often be improved by running more calculations (i.e. by diminishing the scale of the finite element code, or by increasing the number

of Monte-Carlo simulations) which explains that, despite the phenomenal recent progress, the appetite for more computational capacity is not – and may never be – satisfied. Thus, a major problem at stake is the calculation time and the necessity to base the decisions on *few* evaluations of the simulator.

A wide range of problems can be formulated on black-boxes. Some of them will be dealt with in this thesis. For instance, one may be interested in the input values of the simulator for which one (scalar) output takes the highest or lowest value (optimization problem). Alternatively we could seek for the set of inputs where some scalar, or vector-valued output is in a set of interest (inversion problem). As explained before, the key common point shared by these problems is the need to bring a reasonable solution using few evaluations of the simulator. This implies to select carefully the different locations where the simulator is evaluated, according to a well-defined strategy.

A very common approach to build evaluation strategies of the simulator is the rely on *metamodels*, or *surrogate models* (see, e.g., Wang and Shan [2006], Ginsbourger [2009], Jin [2011], for a review of different metamodeling techniques). A surrogate model is a deterministic or probabilistic representation of the simulator's input/output relationship, built from a restricted number of evaluations. In this thesis, the surrogate model will usually be constructed from an initial set of evaluations (the initial Design Of Experiments, DOE) and will be used as a tool for both representing the function of interest and selecting the next evaluation points. Although very little will be said on the question of the choice of the initial DOE (see, e.g, Stein [1999]), a major effort will be put on the construction of sequential strategies for selecting the next point, or – sometimes – batch of points.

In this work, our metamodel will be a *kriging metamodel* (see, e.g., Matheron [1973], Chilès and Delfiner [2012], Rasmussen and Williams [2006]). In short and without showing any equations, in kriging we assume that the real valued function of interest is a specific realization of a random process, which is often, but not always, assumed Gaussian. The latter assumption is not mandatory in kriging (see, e.g, Palacios and Steel [2006]) as, in general, the "kriging predictor" (also called "kriging mean") at a given non-evaluated point is simply the Best Linear

Unbiased Predictor (BLUP) at this point, from the observations.

The Gaussian assumption widely eases the tractability of the random process when new observations are assimilated. Indeed, an important mathematical result is that a Gaussian process conditioned on some new observations is still a Gaussian process. This result alone mainly explain why the Gaussian assumption is often used in kriging: as new observations are assimilated, the whole conditional distribution of the random process can be calculated easily, through closed form formulas. Today, Gaussian processes play a major role in the computer experiments and machine learning literature (see, e.g., Sacks et al. [1989b], Welch et al. [1992], Santner et al. [2003], Rasmussen and Williams [2006], and the references therein). In the present thesis, Gaussian processes will be used for two crucial purposes:

- Computing pointwise predictions (kriging means) or prediction uncertainties (kriging variances) at non-evaluated points. This may give an approximate representation of the function of interest.
- Constructing a **sampling criterion** which will be a well-chosen function related to the kriging mean, variance and covariance and which will be used to select sequentially the next point (or batch of points) to evaluate. Once a sampling criterion is defined, the considered strategy always consists in evaluating sequentially the point (or batch) where the criterion is maximized or minimized.

The latter idea has been first used for global optimization problems [Mockus, 1989; Mockus et al., 1978; Schonlau, 1997], with the now famous Expected Improvement (EI) criterion. It was popularized by Jones et al. [1998], Jones [2001] with the so called Efficient Global Optimization (EGO) algorithm. The global algorithm is summarized below (see, Algorithm 1). The exact same algorithm will be used in our applications, for a wider range a problems. The only exception is that the sampling criterion will vary depending on the problem we want to solve (optimization, inversion, etc...).
Note that, in Algorithm 1, finding a new point to evaluate requires an optimization of the sampling criterion, $J_n(\cdot)$ over the input space. Thus, the calculation time of $J_n(\cdot)$ itself can be a crucial issue, as we will see in this thesis (mainly in Parts II

Algorithm 1 EGO algorithm in global optimization

Require: n evaluations of the function of interest, f: the initial DOE,

Require: a sampling criterion, J_n. This is a real valued function, defined on the parameter space of f, which depends on some outputs of a kriging metamodel. Here, the sampling criterion is the Expected Improvement (EI).

Do: Fit a kriging metamodel from the initial DOE.

while evaluation budget is not consumed **do**
- Find the input location where the value of the sampling criterion is optimized (here: maximized).
- Evaluate f at this point.
- Update the kriging metamodel, using this new observation.

end while

and III). It is indeed not reasonable to spend days for deciding what the next evaluation point will be if the simulator takes only a few hours to provide its response. In general, one wants the calculation time of the decision algorithm to be small compared of the calculation time of the simulator. Before going further on these issues we give some useful definitions and properties on Gaussian processes in the next Section.

1.2 Gaussian process, standard definition and properties

This Section aims at giving standard definitions and properties on Gaussian random variables, vectors and processes.

1.2.1 Gaussian random variable

A real-valued random variable Y is said to be Gaussian, or normally distributed if it has the following characteristic function:

$$\phi_Y(t) := \mathbb{E}\left(\exp(itY)\right) = \exp(itm - \sigma^2 t^2/2) \tag{1.1}$$

where $m \in \mathbb{R}$ and $\sigma \geq 0$ are two parameters of the distribution.
We write $Y \sim \mathcal{N}(m, \sigma^2)$ to say that Y has a normal distribution with parameters $m = \mathbb{E}(Y)$ and $\sigma^2 = Var(Y)$. We also say than Y has a standard normal distribution when Y has a normal distribution with parameters $m = 0$ and $\sigma^2 = 1$. When $\sigma \neq 0$ the density function of Y is of the form:

$$\varphi_{m,\sigma^2}(x) := \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2\right), \ x \in \mathbb{R}, \tag{1.2}$$

The density of the standard normal distribution is usually denoted by $\varphi(\cdot)$ and we will use the notation $\varphi_{m,\sigma^2}(\cdot)$ when $(m, \sigma^2) \neq (0, 1)$.

The cumulative distribution function (c.d.f) of the standard normal distribution is the function:

$$\Phi(x) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-u^2/2} \mathrm{du}, \ \mathrm{x} \in \mathbb{R}, \tag{1.3}$$

and, while no closed form expression is available to compute this integral, there exist fast algorithms to compute Expression 1.3 with high precision [Cody, 1969; Hart et al., 1968]. We sometimes also use the tail probability function of the standard normal distribution:

$$\Psi(x) := 1 - \Phi(x), \ x \in \mathbb{R}. \tag{1.4}$$

1.2.2 Gaussian vector

A random vector $\mathbf{Y} = (Y_1, \ldots, Y_p)^\top \in \mathbb{R}^p$, $p \geq 1$ is said to have a multivariate normal distribution in dimension p if every linear combination of its coordinates has a normal distribution. Mathematically:

$$\forall (\alpha_1, \ldots, \alpha_p) \in \mathbb{R}^p, \exists m \in \mathbb{R}, \sigma \geq 0 : \sum_{i=1}^{p} \alpha_i Y_i \sim \mathcal{N}(m, \sigma^2)$$

If $\mathbf{Y}$ has a multivariate normal distribution in dimension p, there exist a vector $\mathbf{m} \in \mathbb{R}^p$ and a positive semi-definite matrix $\mathbf{\Sigma} \in \mathbb{R}^{p \times p}$ such that the characteristic

function of $\mathbf{Y}$ is:

$$\phi_{\mathbf{Y}}(\mathbf{t}) := \mathbb{E}\left(\exp(i\langle \mathbf{t}, \mathbf{Y}\rangle)\right) = \exp\left(i\langle \mathbf{t}, \mathbf{m}\rangle - \frac{1}{2}\mathbf{t}^\top \mathbf{\Sigma} \mathbf{t}\right),\ \mathbf{t} \in \mathbb{R}^p. \tag{1.5}$$

We write that $\mathbf{Y} \sim \mathcal{N}_p(\mathbf{m}, \mathbf{\Sigma})$ as $\mathbf{m} = \mathbb{E}(\mathbf{Y})$ and $\mathbf{\Sigma} = Var(\mathbf{Y})$.
If in addition $\mathbf{\Sigma}$ is non-singular, the random vector $\mathbf{Y}$ has density:

$$\varphi_{\mathbf{m},\mathbf{\Sigma}}(\mathbf{x}) = \frac{1}{|\mathbf{\Sigma}|^{1/2}(2\pi)^{p/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{m})^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\mathbf{m})\right), \tag{1.6}$$

where $|\mathbf{\Sigma}|$ denotes the determinant of $\mathbf{\Sigma}$. The c.d.f. of the multivariate normal distribution, which will be extensively used in this manuscript is the function:

$$\Phi_{\mathbf{m},\mathbf{\Sigma}}(\mathbf{x}) := \mathbb{P}(Y_1 \leq x_1, \ldots, Y_p \leq x_p) \tag{1.7}$$

$$= \int_{[-\infty,x_1]\times\ldots\times[-\infty,x_p]} \varphi_{\mathbf{m},\mathbf{\Sigma}}(\mathbf{u})\mathrm{d}\mathbf{u},\ \mathbf{x} \in \mathbb{R}^{\mathrm{p}}, \tag{1.8}$$

where $\mathbf{Y} = (Y_1, \ldots, Y_p)^\top \sim \mathcal{N}_p(\mathbf{m}, \mathbf{\Sigma})$.
As, trivially, $\Phi_{\mathbf{m},\mathbf{\Sigma}}(\mathbf{x}) = \Phi_{\mathbf{0},\mathbf{\Sigma}}(\mathbf{x}-\mathbf{m})$, we will also adopt the notation $\Phi(\mathbf{x};\mathbf{\Sigma})$, or $\Phi_p(\mathbf{x};\mathbf{\Sigma})$ for the c.d.f. of the centered multivariate Gaussian distribution, in dimension p, with covariance matrix $\mathbf{\Sigma}$.

Note that, from Equation (1.5), we can see that uncorrelated Gaussian variables are independent[1]. We also have that the multivariate Gaussian distribution is entirely determined by the first two moments, and that if $\mathbf{Y} \sim \mathcal{N}_p(\mathbf{m}, \mathbf{\Sigma})$ then $\mathbf{a} + \mathbf{B}^\top \mathbf{Y} \sim \mathcal{N}_q(\mathbf{a} + \mathbf{B}^\top \mathbf{m}, \mathbf{B}^\top \mathbf{\Sigma} \mathbf{B})$ for all $q \geq 1, \mathbf{a} \in \mathbb{R}^q, \mathbf{B} \in \mathbb{R}^{p\times q}$. From the equivalence between independence and non-correlation we can prove the following results:

Proposition 1 (Gaussian vector conditioning) *Let* $(\mathbf{Y}_1^\top, \mathbf{Y}_2^\top)^\top$ *be a Gaussian*

[1] as the characteristic function can be expressed as a product of p characteristic functions depending only on $x_1, \ldots, x_p$ respectively

random vector $(\mathbf{Y}_1 \in \mathbb{R}^p, \mathbf{Y}_2 \in \mathbb{R}^q)$ *with distribution:*

$$\mathcal{N}_{p+q}\left(\begin{pmatrix}\mathbf{m}_1\\ \mathbf{m}_2\end{pmatrix}\begin{pmatrix}\mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12}\\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22}\end{pmatrix}\right),$$

$\mathbf{m}_1 \in \mathbb{R}^p, \mathbf{m}_2 \in \mathbb{R}^q, \mathbf{\Sigma}_{11} \in \mathbb{R}^{p\times p}, \mathbf{\Sigma}_{12} = \mathbf{\Sigma}_{21}^\top \in \mathbb{R}^{p\times q}, \mathbf{\Sigma}_{22} \in \mathbb{R}^{q\times q}$,

with $\mathbf{\Sigma}_{11}, \mathbf{\Sigma}_{22}$ *positive semi-definite and* $\mathbf{\Sigma}_{22}$ *non singular.*

- *The conditional expectation of* $\mathbf{Y}_1$ *knowing* $\mathbf{Y}_2$ *coincides with the linear expectation, i.e.* $\exists \mathbf{a} \in \mathbb{R}^p, \mathbf{B} \in \mathbb{R}^{p\times q} :\ \mathbb{E}(\mathbf{Y}_1|\mathbf{Y}_2) = \mathbf{a} + \mathbf{B}\mathbf{Y}_2$
- $\mathcal{L}(\mathbf{Y}_1|\mathbf{Y}_2 = \mathbf{y}_2) = \mathcal{N}_p\left(\mathbf{m}_1 + \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}(\mathbf{y}_2 - \mathbf{m}_2), \mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}\right)$

These results are proven in many books or articles. Here is a sketch of the proof, in the particular case $\mathbf{m} = 0$:
proof: Let $\varepsilon := Y_1 - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}Y_2$. ε is a Gaussian vector and one can verify that $Cov(Y_2, \varepsilon) = 0$, which means (Gaussian case) that ε and Y_2 are independent. Now, ε is orthogonal to any random variable $g(Y_2)$, with g a Borel function which shows (uniqueness of the orthogonal projection) that $\mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}Y_2$ is indeed the conditional expectation $\mathbb{E}(Y_1|Y_2)$. Moreover, a decomposition of Y_1 shows that $Var(Y_1|Y_2) = Var(\varepsilon|Y_2) = Var(\varepsilon) = \mathbf{\Sigma}_{11} - \mathbf{\Sigma}_{12}\mathbf{\Sigma}_{22}^{-1}\mathbf{\Sigma}_{21}$. Finally, from $Y_1 = \mathbb{E}(Y_1|Y_2) + \varepsilon$ we see that the random variable Y_1 conditioned on $Y_2 = y_2$ has a normal distribution, the mean and variance of which have been exhibited.□

1.2.3 Gaussian process

Let $\mathbb{X}$ be a parameter space. A random (or stochastic) process ξ over $\mathbb{X}$ is a collection of random variables

$$\{\xi(\mathbf{x}) : \mathbf{x} \in \mathbb{X}\}$$

defined over the same probability space. Now a Gaussian process is a random process for which all finite dimensional distributions are Gaussian. Mathematically: $\forall q \geq 1, \forall(\mathbf{x}_1, \ldots, \mathbf{x}_q) \in \mathbb{X}^q$, the random vector: $(\xi(\mathbf{x}_1), \ldots, \xi(\mathbf{x}_q))^\top$ has a multivariate Gaussian distribution in dimension q.

The distribution of a Gaussian process is[1] fully determined by its mean function $m(\cdot)$ and its covariance function (or kernel) $k(\cdot,\cdot)$:

$$m(\mathbf{x}) := \mathbb{E}(\xi(\mathbf{x})),\ \mathbf{x} \in \mathbb{X} \tag{1.9}$$

$$k(\mathbf{x},\mathbf{x}') := Cov(\xi(\mathbf{x}),\xi(\mathbf{x}')),\ (\mathbf{x},\mathbf{x}') \in \mathbb{X}^2 \tag{1.10}$$

We write $\xi \sim \mathrm{GP}(m(\cdot),k(\cdot,\cdot))$ or even sometimes $\xi \sim \mathrm{GP}(m,k)$ as there is no ambiguity that, here, m and k are functions. We can use Proposition 1 to show that a Gaussian process $\xi \sim \mathrm{GP}(m,k)$ conditioned on some observations $\mathbf{Y} := (\xi(\mathbf{x}_1),\dots,\xi(\mathbf{x}_n))$ is still a Gaussian process with a different mean function and covariance kernel. More precisely:

$$\begin{aligned} \mathcal{L}\left(\xi|\mathbf{Y}\right) &= \mathrm{GP}(m_n,k_n) \text{ with:} \\ m_n(\mathbf{x}) &= m(\mathbf{x}) + \mathbf{k}(\mathbf{x})^\top K^{-1}(\mathbf{Y} - m(\underline{\mathbf{x}_n})) \\ k_n(\mathbf{x},\mathbf{x}') &= k(\mathbf{x},\mathbf{x}') - \mathbf{k}(\mathbf{x})^\top K^{-1}\mathbf{k}(\mathbf{x}') \end{aligned} \tag{1.11}$$

where $\mathbf{k}(\mathbf{x}) := (k(\mathbf{x},\mathbf{x}_1),\dots,k(\mathbf{x},\mathbf{x}_n))^\top$ is the covariance between $\xi(\mathbf{x})$ and the observations, $m(\underline{\mathbf{x}_n}) := (m(\mathbf{x}_1),\dots,m(\mathbf{x}_n))^\top$, and K is the covariance matrix between the observations (i.e. $K \in \mathbb{R}^{n\times n}$ with $K_{ij} := k(\mathbf{x}_i,\mathbf{x}_j)$).

1.3 Advanced properties on Gaussian vectors

We now give some properties on Gaussian vectors that are going to be useful in the sequel of this work. Two subsections are presented here. In addition, one subsection which opens interesting perspectives for the work presented in Chapter 6 is sent in the Appendix F, Section F.2.

1.3.1 Tallis formulas for the expectation of the maximum

Let $\mathbf{Y} = (Y_1,\dots,Y_q)^\top$ be a Gaussian vector, taking values in $\mathbb{R}^q$, with mean vector $\mathbf{m} := (m_1,\dots,m_q)^\top \in \mathbb{R}^q$ and covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{q\times q}$. Let $\mathbf{b} :=$

[1] as, using Kolmogorov extension theorem, a collection of finite dimensional distribution defines the stochastic process

$(b_1, \ldots, b_q)^\top \in (\mathbb{R} \cup \{\infty\})^q$ be a vector of size q that may contain, real numbers, or ∞. Tallis' formula [Da Veiga and Marrel, 2012; Tallis, 1961] enables to calculate the expectation of any coordinate Y_k of $\mathbf{Y}$ under the linear constraint: $\forall j \in \{1, \ldots, q\}, Y_j \leq b_j$, which is abbreviated with the vectorized notation $\mathbf{Y} \leq \mathbf{b}$. The formula is given below:

Proposition 2 (Tallis formula)

$$\mathbb{E}(Y_k | \mathbf{Y} \leq \mathbf{b}) = m_k - \frac{1}{p} \sum_{i=1}^{q} \boldsymbol{\Sigma}_{ik} \; \varphi_{m_i, \boldsymbol{\Sigma}_{ii}}(b_i) \; \Phi_{q-1}\left(\mathbf{c}_{.i}; \boldsymbol{\Sigma}_{.i}\right) \tag{1.12}$$

where:

- $p := \mathbb{P}(\mathbf{Y} \leq \mathbf{b}) = \Phi_q(\mathbf{b} - \mathbf{m}; \boldsymbol{\Sigma})$
- $\Phi_q(\cdot; \boldsymbol{\Sigma})$ *is the c.d.f. of the centered multivariate Gaussian distribution in dimension* q*, with covariance matrix* $\boldsymbol{\Sigma}$*.*
- $\mathbf{c}_{.i}$ *is the vector of* $\mathbb{R}^{q-1}$ *with general term* $(b_j - m_j) - (b_i - m_i)\frac{\boldsymbol{\Sigma}_{ij}}{\boldsymbol{\Sigma}_{ii}}$*,* $j \neq i$
- $\boldsymbol{\Sigma}_{.i}$ *is a* $(q-1) \times (q-1)$ *matrix obtained by computing* $\boldsymbol{\Sigma}_{uv} - \frac{\boldsymbol{\Sigma}_{iu}\boldsymbol{\Sigma}_{iv}}{\boldsymbol{\Sigma}_{ii}}$ *for* $u \neq i$ *and* $v \neq i$*.*

One may note that the computation of Expression (1.12) requires one call to the c.d.f. of the multivariate normal distribution in dimension q, through the calculation of $\mathbb{P}(\mathbf{Y} \leq \mathbf{b})$, and q calls to the c.d.f. of this distribution in dimension $q - 1$. Moreover, the vector $\mathbf{c}_{.i}$ and the matrix $\boldsymbol{\Sigma}_{.i}$ can be interpreted using the Gaussian vectors conditioning formula seen before (see, Equation (1.11)). Indeed, $\mathbf{c}_{.i}$ is the mean vector of the centered Gaussian vector $\mathbf{Y}_{-i} := (Y_1 - m_1, \ldots, Y_{i-1} - m_{i-1}, Y_{i+1} - m_{i+1}, \ldots, Y_q - m_q)^\top$ knowing that $Y_i = b_i$, and $\boldsymbol{\Sigma}_{.i}$ is its covariance matrix:

$$\mathcal{L}(\mathbf{Y}_{-i} | Y_i = b_i) = \mathcal{N}_{q-1}(\mathbf{c}_{.i}, \boldsymbol{\Sigma}_{.i}) \tag{1.13}$$

A proof for Equation (1.12) is available in Tallis [1961] and also in an article [Chevalier and Ginsbourger, 2013] in the Appendix of this thesis (see, Appendix B).

The Tallis' formula happens to be a powerful tool to compute analytically expressions of the type $\mathbb{E}(\max_{i\in\{1,\dots,q\}} Y_i)$. Indeed, as detailed in Chevalier and Ginsbourger [2013] and following the idea of Ginsbourger et al. [2010], the random variable $\max_{i\in\{1,\dots,q\}} Y_i$ can be decomposed into a sum of q random variables $Y_i \mathbb{1}_{(\forall k\neq i, Y_i>Y_k)}, i = 1,\dots,q$, simply because the maximum of the Y_i's has to be one of the Y_i's. The expectation of each of these q random variable can be calculated with a straightforward application of Tallis' formula. Thus, one can construct analytical expressions for $\mathbb{E}(\max_{i\in\{1,\dots,q\}} Y_i)$ and even $\mathbb{E}(\max_{i\in\{1,\dots,q\}} g(Y_i))$ where $g(\cdot)$ is an affine function. An adapted choice of function $g(\cdot)$ leads to a new analytical formula for the so called *multi-points Expected Improvement* [Ginsbourger, 2009; Ginsbourger et al., 2010; Schonlau, 1997] defined as:

$$\mathbb{E}(\max_{i\in\{1,\dots,q\}} \max(0, Y_i - T)), \tag{1.14}$$

where $T \in \mathbb{R}$ is a threshold. This is further detailed in Chapter 2, Section 2.3. Also, the idea that the maximum of a Gaussian vector has to be one of its coordinate is further used in the Appendix F, Section F.2 for constructing upper bounds for an exceedance probability.

1.3.2 Updating the mean and covariance function of a Gaussian process

Let $\xi \sim \mathrm{GP}(m, k)$ be a Gaussian process on a domain $\mathbb{X}$. We assume that we observed the values of ξ at n points $\mathbf{x}_{\text{old}} := (\mathbf{x}_1,\dots,\mathbf{x}_n) \in \mathbb{X}^n$ and then at r additional points $\mathbf{x}_{\text{new}} := (\mathbf{x}_{n+1},\dots,\mathbf{x}_{n+r})$. Let $\mathbf{x}, \mathbf{x}'$ be two points in $\mathbb{X}$, we are interested in the quantities:

$$m_{n+r}(\mathbf{x}) := \mathbb{E}(\xi(\mathbf{x})|\mathcal{A}_{n+r}) \tag{1.15}$$

$$k_{n+r}(\mathbf{x}, \mathbf{x}') := Cov(\xi(\mathbf{x}), \xi(\mathbf{x}')|\mathcal{A}_{n+r}), \tag{1.16}$$

where $\mathcal{A}_{n+r}$ is the $\sigma-$algebra generated by $(\xi(\mathbf{x}_i))_{1\leq i\leq n+r}$. A direct computation can be performed using Gaussian process conditioning formulas (see, Equation (1.11)) at the cost of an $(n+r)\times(n+r)$ matrix inversion. However, if the r

last observation come after the n first ones, one can take advantage of a previous computation of $m_n(\mathbf{x})$ and $k_n(\mathbf{x}, \mathbf{x}')$ to reduce this calculation cost. We indeed have that:

$$m_{n+r}(\mathbf{x}) = m_n(\mathbf{x}) + k_n(\mathbf{x}, \mathbf{x}_{\text{new}})^\top K_{\text{new}}^{-1}(\mathbf{Y}_{\text{new}} - m_n(\mathbf{x}_{\text{new}})) \tag{1.17}$$

$$k_{n+r}(\mathbf{x}, \mathbf{x}') = k_n(\mathbf{x}, \mathbf{x}') - k_n(\mathbf{x}, \mathbf{x}_{\text{new}})^\top K_{\text{new}}^{-1} k_n(\mathbf{x}', \mathbf{x}_{\text{new}}) \tag{1.18}$$

where $K_{\text{new}} := k_n(\mathbf{x}_{n+i}, \mathbf{x}_{n+j})_{1 \leq i,j \leq r}$, $k_n(\mathbf{x}, \mathbf{x}_{\text{new}}) := (k_n(\mathbf{x}, \mathbf{x}_{n+1}), \ldots, k_n(\mathbf{x}, \mathbf{x}_{n+r}))^\top$, $\mathbf{Y}_{\text{new}} = (\xi(\mathbf{x}_{n+1}), \ldots, \xi(\mathbf{x}_{n+r}))^\top$ and $m_n(\mathbf{x}_{\text{new}}) = (m_n(\mathbf{x}_{n+1}), \ldots, m_n(\mathbf{x}_{n+r}))^\top$. The latter formulas are applications of Equation (1.11) to the Gaussian process ξ conditioned on $\mathcal{A}_n$ as $\mathcal{L}(\xi|\mathcal{A}_n) = \text{GP}(m_n, k_n)$. In the present setting, their use avoid a $(n+r) \times (n+r)$ matrix inversion. We will see in the next chapter that similar formulas apply in the more general setting of *kriging*.

Chapter 2

Kriging

2.1 Kriging basics

Let ξ be a random process, defined over an input space $\mathbb{X}$. The term *kriging* is generally used when one aims at performing linear prediction of a random variable $\xi(\mathbf{x})$, $\mathbf{x} \in \mathbb{X}$, using n available observations of the random process ξ, denoted by $\mathbf{Y} := (\xi(\mathbf{x}_1), \ldots, \xi(\mathbf{x}_n))^\top$.

Before going further, we need to introduce some definitions

Definition 1 *A random process ξ is said to be stationary, or strongly stationary, if any of its finite dimensional distribution is invariant by translation, i.e.:*

$$\forall \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{h} \in \mathbb{X}, (\xi(\mathbf{x}_1), \ldots, \xi(\mathbf{x}_n)) \stackrel{\mathcal{L}}{=} (\xi(\mathbf{x}_1 + \mathbf{h}), \ldots, \xi(\mathbf{x}_n + \mathbf{h})), \tag{2.1}$$

where $\stackrel{\mathcal{L}}{=}$ denotes the equality in law.
A random process ξ is said to be weakly stationary, or stationary of order 2 if $\forall \mathbf{x} \in \mathbb{X}, \mathbb{E}(\xi(\mathbf{x})^2) < \infty$ and:

- *the mean function of ξ, $m(\mathbf{x}) := \mathbb{E}(\xi(\mathbf{x}))$ exists and is constant,*
- *the covariance function of ξ, $k(\mathbf{x}, \mathbf{x}') := Cov(\xi(\mathbf{x}), \xi(\mathbf{x}'))$ exists and only depends on $\mathbf{h} := \mathbf{x} - \mathbf{x}'$.*

A random process ξ is said to be intrinsic-stationary if:

- $\forall \mathbf{x}, \mathbf{h} \in \mathbb{X}, \mathbb{E}(\xi(\mathbf{x}+\mathbf{h}) - \xi(\mathbf{x})) = 0$
- *The function* $(\mathbf{x}, \mathbf{h}) \mapsto Var(\xi(\mathbf{x}+\mathbf{h}) - \xi(\mathbf{x}))$ *exists and only depends on* $\mathbf{h}$.

Historically, in kriging, depending on the assumptions on the mean function of the random process ξ, ξ is assumed to be weakly stationary or intrinsic-stationary. In all this chapter we will instead assume that ξ is a L^2 random process, i.e. that the covariance $k(\mathbf{x}, \mathbf{x}') := Cov(\xi(\mathbf{x}), \xi(\mathbf{x}'))$ always exists and is finite. Let us compare this assumption with the historical settings:

- In the so-called *simple kriging* (as defined later), ξ is often assumed weakly stationary. Thus our assumption of a L^2 random process is more general than the historical assumptions. Indeed, the covariance $k(\mathbf{x}, \mathbf{x}+\mathbf{h})$ only needs to exist but does no longer need to be a function of $\mathbf{h}$.
- In the so-called *universal kriging* ξ is often assumed intrinsic-stationary. As some intrinsic-stationary processes are not L^2, the equations are not written in terms of covariance, but instead use the notion of semi-variogram, $2\gamma(\mathbf{h}) := Var(\xi(\mathbf{x}+\mathbf{h}) - \xi(\mathbf{x}))$ (see, e.g., Chilès and Delfiner [2012]). Thus our assumption of a L^2 random process is not more general than the historical assumption. However our assumption is neither more restrictive as a L^2 random process is not necessarily intrinsic-stationary.

These definitions and clarifications being done, let us move back to the motivations. Let's assume that ξ is a L^2 random process. In kriging we are interested in a linear predictor of $\xi(\mathbf{x})$ written as:

$$m_n(\mathbf{x}) := \sum_{i=1}^{n} \lambda_{i,n}(\mathbf{x})\xi(\mathbf{x}_i) := \boldsymbol{\lambda}(\mathbf{x})^\top \mathbf{Y}, \tag{2.2}$$

where the so-called *kriging weights* $\boldsymbol{\lambda}(\mathbf{x}) := (\lambda_{1,n}(\mathbf{x}), \ldots, \lambda_{n,n}(\mathbf{x}))^\top$ are chosen in order to minimize, over all possible $\boldsymbol{\lambda} \in \mathbb{R}^n$ satisfying some constraints, the variance of the residual, $Var(\xi(\mathbf{x}) - \boldsymbol{\lambda}^\top \mathbf{Y})$, with the unbiasedness constraint $\mathbb{E}(\xi(\mathbf{x}) - m_n(\mathbf{x})) = 0$. The residual (*or kriging error*) obtained with these kriging

weights has a variance,

$$s_n^2(\mathbf{x}) := Var(\xi(\mathbf{x}) - \boldsymbol{\lambda}(\mathbf{x})^\top \mathbf{Y}), \tag{2.3}$$

which is called *kriging variance*. The covariance between two kriging errors at different locations,

$$k_n(\mathbf{x}, \mathbf{x}') := Cov(\xi(\mathbf{x}) - \boldsymbol{\lambda}(\mathbf{x})^\top \mathbf{Y}, \xi(\mathbf{x}') - \boldsymbol{\lambda}(\mathbf{x}')^\top \mathbf{Y}), \tag{2.4}$$

is called *kriging covariance*. Note that in Equations (2.3) and (2.4), the variances and covariances refers to the randomness of $\xi(\mathbf{x})$ **and also** $\mathbf{Y} = (\xi(\mathbf{x}_1), \ldots, \xi(\mathbf{x}_n))$.

We can now define, in the next subsections, two different settings that are often used with kriging: Simple Kriging (SK) and Universal Kriging (UK).

2.1.1 Simple kriging

Let ξ be a L^2 random process with known covariance function $k(\cdot, \cdot)$. In Simple kriging, the mean function $m(\mathbf{x})$ of ξ is assumed to be known. In that case, minimizing, over all possible weights $\boldsymbol{\lambda}$ the function $Var(\xi(\mathbf{x}) - \boldsymbol{\lambda}^\top \mathbf{Y})$ is not difficult because of the convexity of the function in $\boldsymbol{\lambda}$ (see, e.g., Baillargeon [2005]). When the covariance matrix $K := (k(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i,j \leq n}$ between the observations is non-singular we obtain the so-called Simple kriging equations written here in the case where $m(\mathbf{x}) = 0$:

$$\boldsymbol{\lambda}(\mathbf{x}) = K^{-1}\mathbf{k}(\mathbf{x}) \tag{2.5}$$

$$m_n(\mathbf{x}) = \boldsymbol{\lambda}(\mathbf{x})^\top \mathbf{Y} = \mathbf{k}(\mathbf{x})^\top K^{-1} \mathbf{Y} \tag{2.6}$$

$$s_n^2(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^\top K^{-1} \mathbf{k}(\mathbf{x}) \tag{2.7}$$

$$k_n(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}(\mathbf{x})^\top K^{-1} \mathbf{k}(\mathbf{x}') \tag{2.8}$$

where $\mathbf{k}(\mathbf{x}) := (k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_n))^\top$.
Note that, in the case where $m(\mathbf{x})$ is not equal to zero, it suffices to consider the

centered process $\xi - m$ to obtain:

$$m_n(\mathbf{x}) = m(\mathbf{x}) + \mathbf{k}(\mathbf{x})^\top K^{-1}(\mathbf{Y} - m(\underline{\mathbf{x}_n})) \tag{2.9}$$

where $m(\underline{\mathbf{x}_n}) := (m(\mathbf{x}_1), \ldots, m(\mathbf{x}_n))^\top$. The kriging variances and covariances when $m(\mathbf{x})$ is not zero do not change.

Remark 1 *Historically, as mentioned in the beginning of this Chapter, one assumes in simple kriging that ξ is weakly stationary. This supplementary assumption is not required to obtain simple kriging equations (2.5), (2.6), (2.7), (2.8), and is only added to simplify the equations (see, e.g. Roustant et al. [2012]).*

2.1.2 Universal kriging

The hypothesis of a zero (or known) mean function for the random process ξ is often not realistic for practitioners. In this section, we still consider a L^2 random process ξ with known covariance function $k(\cdot, \cdot)$; but we also assume that the mean function of ξ is unknown and can be written:

$$m(\mathbf{x}) = \sum_{i=1}^{l} \beta_i f_i(\mathbf{x}), \tag{2.10}$$

where $\beta_1, \ldots, \beta_l \in \mathbb{R}$ are unknown coefficients and $f_1(\cdot), \ldots, f_l(\cdot)$ are l known basis functions. Let us denote by $\mathbf{f}(\mathbf{x}) := (f_1(\mathbf{x}), \ldots, f_l(\mathbf{x}))^\top$ and let $\mathbb{F} \in \mathbb{R}^{n \times l}$ be the matrix with row i equal to $\mathbf{f}(\mathbf{x}_i)^\top$. The goal is again to minimize, over $\boldsymbol{\lambda} \in \mathbb{R}^n$, the variance of the kriging residual $Var(\xi(\mathbf{x}) - \boldsymbol{\lambda}^\top \mathbf{Y})$ with the unbiasedness constraint.
Calculations (detailed in, e.g., Cressie [1993]) lead to the Universal kriging equa-

tions:

$$\boldsymbol{\lambda}(\mathbf{x}) = K^{-1}\left(\mathbf{k}(\mathbf{x}) + \mathbb{F}(\mathbb{F}^\top K^{-1}\mathbb{F})^{-1}(\mathbf{f}(\mathbf{x}) - \mathbb{F}^\top K^{-1}\mathbf{k}(\mathbf{x}))\right) \tag{2.11}$$

$$m_n(\mathbf{x}) = \boldsymbol{\lambda}(\mathbf{x})^\top \mathbf{Y} = \mathbf{f}(\mathbf{x})^\top \widehat{\boldsymbol{\beta}} + \mathbf{k}(\mathbf{x})^\top K^{-1}\left(\mathbf{Y} - \mathbb{F}\widehat{\boldsymbol{\beta}}\right) \tag{2.12}$$

$$\begin{aligned} s_n^2(\mathbf{x}) = {} & k(\mathbf{x},\mathbf{x}) - \mathbf{k}(\mathbf{x})^\top K^{-1}\mathbf{k}(\mathbf{x}) + \\ & (\mathbf{f}(\mathbf{x})^\top - \mathbf{k}(\mathbf{x})^\top K^{-1}\mathbb{F})(\mathbb{F}^\top K^{-1}\mathbb{F})^{-1}(\mathbf{f}(\mathbf{x})^\top - \mathbf{k}(\mathbf{x})^\top K^{-1}\mathbb{F})^\top \end{aligned} \tag{2.13}$$

$$\begin{aligned} k_n(\mathbf{x},\mathbf{x}') = {} & k(\mathbf{x},\mathbf{x}') - \mathbf{k}(\mathbf{x})^\top K^{-1}\mathbf{k}(\mathbf{x}') + \\ & (\mathbf{f}(\mathbf{x})^\top - \mathbf{k}(\mathbf{x})^\top K^{-1}\mathbb{F})(\mathbb{F}^\top K^{-1}\mathbb{F})^{-1}(\mathbf{f}(\mathbf{x}')^\top - \mathbf{k}(\mathbf{x}')^\top K^{-1}\mathbb{F})^\top \end{aligned} \tag{2.14}$$

where $\widehat{\boldsymbol{\beta}} := (\mathbb{F}^\top K^{-1}\mathbb{F})^{-1}\mathbb{F}^\top K^{-1}\mathbf{Y}$.

An equivalent formulation of the Universal kriging equations (see, e.g. Bect et al. [2012]) is to say that the kriging weights are solution of the linear system:

$$\begin{pmatrix} K & \mathbb{F} \\ \mathbb{F}^\top & \mathbf{0} \end{pmatrix} \underbrace{\begin{pmatrix} \boldsymbol{\lambda}(\mathbf{x}) \\ \boldsymbol{\mu}(\mathbf{x}) \end{pmatrix}}_{:=\widetilde{\boldsymbol{\lambda}}(\mathbf{x})} = \underbrace{\begin{pmatrix} \mathbf{k}(\mathbf{x}) \\ \mathbf{f}(\mathbf{x}) \end{pmatrix}}_{:=\widetilde{\mathbf{k}}(\mathbf{x})} \tag{2.15}$$

where $\boldsymbol{\mu}(\mathbf{x})$ are Lagrange multipliers associated with the unbiasedness constraint. Using these notations, we have:

$$m_n(\mathbf{x}) = \boldsymbol{\lambda}(\mathbf{x})^\top \mathbf{Y} \tag{2.16}$$

$$s_n^2(\mathbf{x}) = k(\mathbf{x},\mathbf{x}) - \widetilde{\mathbf{k}}(\mathbf{x})^\top \widetilde{\boldsymbol{\lambda}}(\mathbf{x}) \tag{2.17}$$

$$k_n(\mathbf{x},\mathbf{x}') = k(\mathbf{x},\mathbf{x}') - \widetilde{\mathbf{k}}(\mathbf{x})^\top \widetilde{\boldsymbol{\lambda}}(\mathbf{x}'). \tag{2.18}$$

Note that, when $l = 1$ and $f_1(\cdot)$ is the constant function equal to 1 we are in the particular settings of *Ordinary kriging*. In that case, the random process ξ has an unknown, constant, mean $\boldsymbol{\beta} = \beta_1$. Usually, $\boldsymbol{\beta}$ is estimated together with the kriging weights and is a linear combination of the n observations.

Figure 2.1 presents an example of Simple and Universal kriging with four observations of a 1d-function. The covariance function is known and is assumed to be a Matérn covariance with parameter $\nu = 3/2$ (see, Stein [1999]; Yaglom [1986]

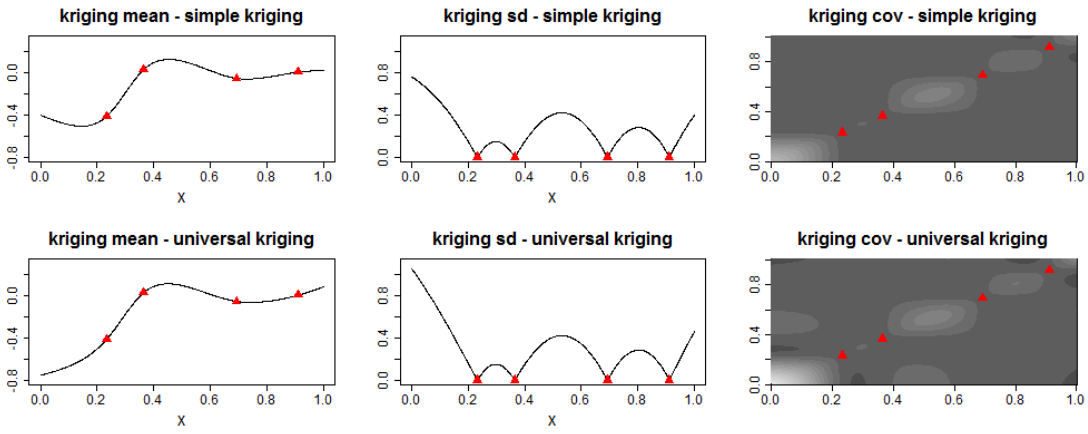

Figure 2.1: Kriging mean, standard-deviation and covariance functions for two types of kriging: Simple kriging and Universal kriging.

for more details on classical covariance functions used in kriging). Also, for the Universal kriging, the basis functions are here $f_1(\mathbf{x}) = 1, f_2(\mathbf{x}) = \mathbf{x}$ (linear trend). We can note that, in Simple kriging, the kriging mean tends to be attracted by the mean of the process (here: zero), which is not the case in Universal kriging, where there is a linear trend. Moreover, the uncertainty on the trend parameters leads to higher kriging standard-deviations in Universal kriging than in Simple kriging, specially in sparse regions, with no observations.

2.1.3 Kriging with the Gaussian assumption

In applications, kriging is often used with the additional assumption that ξ is a **Gaussian** process, although the previous equations hold without this assumption. In this setting, the Simple kriging equations (2.6), (2.7), (2.8) coincide with the Gaussian process conditioning formulas (1.11). The kriging mean and covariance are the conditional expectations and covariance functions of the Gaussian process ξ given the observations. Also, As mentioned in Chapter 1, ξ conditioned on the observations is still a Gaussian process, which will be particularly convenient in the next Chapters.

In the Universal kriging setting, a convenient way to ensure that the process ξ

conditioned on the observations is still Gaussian is to adopt a Bayesian approach. More precisely, let us consider the following prior for ξ:

$$\xi \sim \mathrm{GP}\left(\sum_{i=1}^{l} \beta_i f_i(\cdot), k(\cdot,\cdot)\right) \tag{2.19}$$

where the covariance kernel k is known and where the vector $\boldsymbol{\beta}$ has an improper uniform distribution over $\mathbb{R}^l$. With such a prior, the posterior distribution of ξ, knowing the observations $\mathbf{Y}$ is Gaussian with mean function $m_n(\cdot)$ given by Equation (2.12) and covariance function given by Equation (2.14) (see, e.g., O'Hagan [1978] for a proof). Mathematically, in this Bayesian setting, $\xi|\mathbf{Y} \sim \mathrm{GP}(m_n, k_n)$. The latter result is the main reason why we used the same notations as in Chapter 1 for the kriging mean, variance and covariance.

The formulas presented in this Chapter require the knowledge of the covariance function k, which is often unrealistic in application. If it is only assumed that k belongs to a parametric family k_θ of covariance functions, a common approach consists in plugging the maximum likelihood (or maximum a posteriori) estimate of the parameter vector θ (see, e.g., Stein [1999]). This plug-in approach will be used in Parts II and III of this work. However, as mentioned in the Appendix C, the results obtained in these Parts are applicable (at the cost of additional computational efforts) if a fully Bayesian approach is chosen, i.e. if a prior distribution is chosen for θ.

2.2 The Kriging update equations

Equations (2.12), (2.13), (2.14) are rather complicated and computer intensive if they are computed at a large number of points and/or if the number of observations is high. We will see in Part II on Stepwise Uncertainty Reduction strategies that one is often interested in calculating *updated* kriging means, variances or covariances. The problem at hand is rather simple: in an Universal kriging setting we consider n observations $\xi(\mathbf{x}_1), \ldots, \xi(\mathbf{x}_n)$. We assume that, from these n observations, we already computed the kriging mean $m_n(\mathbf{x})$, and variance $s_n^2(\mathbf{x})$ at some point $\mathbf{x}$ and/or the kriging covariance $k_n(\mathbf{x}, \mathbf{x}')$ with some other point $\mathbf{x}'$. Let

$\mathbf{x}_{\text{new}} := (\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+r}) \in \mathbb{X}^r$ be a batch of r points. It is possible to take advantage of previous calculations to compute quickly updated kriging means, variances and covariances once $\xi(\mathbf{x}_{n+1}), \ldots, \xi(\mathbf{x}_{n+r})$ have been observed. We indeed have that:

$$\boldsymbol{\lambda}_{\text{new}}(\mathbf{x}) = K_{\text{new}}^{-1} k_n(\mathbf{x}, \mathbf{x}_{\text{new}}) \tag{2.20}$$

$$m_{n+r}(\mathbf{x}) = m_n(\mathbf{x}) + k_n(\mathbf{x}, \mathbf{x}_{\text{new}})^\top K_{\text{new}}^{-1}(\mathbf{Y}_{\text{new}} - m_n(\mathbf{x}_{\text{new}})) \tag{2.21}$$

$$s_{n+r}^2(\mathbf{x}) = s_n^2(\mathbf{x}) - k_n(\mathbf{x}, \mathbf{x}_{\text{new}})^\top K_{\text{new}}^{-1} k_n(\mathbf{x}, \mathbf{x}_{\text{new}}) \tag{2.22}$$

$$k_{n+r}(\mathbf{x}, \mathbf{x}') = k_n(\mathbf{x}, \mathbf{x}') - k_n(\mathbf{x}, \mathbf{x}_{\text{new}})^\top K_{\text{new}}^{-1} k_n(\mathbf{x}', \mathbf{x}_{\text{new}}) \tag{2.23}$$

where $K_{\text{new}} := k_n(\mathbf{x}_{n+i}, \mathbf{x}_{n+j})_{1 \leq i,j \leq r}$, $k_n(\mathbf{x}, \mathbf{x}_{\text{new}}) := (k_n(\mathbf{x}, \mathbf{x}_{n+1}), \ldots, k_n(\mathbf{x}, \mathbf{x}_{n+r}))^\top$, $\mathbf{Y}_{\text{new}} = (\xi(\mathbf{x}_{n+1}), \ldots, \xi(\mathbf{x}_{n+r}))^\top$, $m_n(\mathbf{x}_{\text{new}}) := (m_n(\mathbf{x}_{n+1}), \ldots, m_n(\mathbf{x}_{n+r}))^\top$, and $\boldsymbol{\lambda}_{\text{new}}(\mathbf{x})$ are the r kriging weights of $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+r}$ for the prediction at point $\mathbf{x}$. The formulas above, referred to as the *kriging update formulas*, happen to be similar to Equations (2.6), (2.7), (2.8). They are Simple kriging equations applied on a random process with mean function $m_n(\cdot)$ and covariance function $k_n(\cdot, \cdot)$.

Note that, in the case where ξ is Gaussian with known mean and covariance functions, the Equations above also exactly correspond to Gaussian process conditioning formulas. However, in the more general setting of Universal kriging (non-Gaussian processes, mean function unknown) they have been proven only quite recently in the particular case $r = 1$ by Barnes and Watson [1992]; Gao et al. [1996]. For $r > 1$, a recent article by Emery [2009] proved Equation (2.21) but failed to prove Equations (2.22) and (2.23). To the best of our knowledge, Equations (2.22) and (2.23), in the general settings of Universal kriging, are new. A proof for these Equations is given in the paper of Chevalier et al. [2013c] in Appendix A of this manuscript.

Let us give a setting where the kriging update equations enable substantial computational savings. Let $\mathbf{x}, \mathbf{x}' \in \mathbb{X}$ and assume that $m_n(\mathbf{x})$, $s_n^2(\mathbf{x})$ and/or $k_n(\mathbf{x}, \mathbf{x}')$ have been precomputed from n observations. One clearly sees from Equations (2.21), (2.22), (2.23) that, to calculate the updated kriging mean, variance or covariance from one batch of r additional new observations, it remains to compute $k_n(\mathbf{x}, \mathbf{x}_{\text{new}})$. In order words, one needs to compute r kriging covariances between a fixed point

$\mathbf{x}$ and r new points. This can be done at a cost of $O(rn)$ using Equation (2.14) if – in addition – one pre-computes the terms of Equation (2.14) that do not depend on $\mathbf{x}_{\text{new}}$. The complexity of these precomputation, of $O(n^2)$, is significant but becomes small if we consider a large number p of different batches $\mathbf{x}^1_{\text{new}}, \ldots, \mathbf{x}^p_{\text{new}}$. Indeed the precomputation needs to be performed only one time, and not p times.

In addition, the complexity to compute K^{-1}_{new} for a given batch is also high as it is of $O(rn^2 + r^2 n)$, which is equal to $O(rn^2)$ if we assume that $n \gg r$. However, this price has to be paid only one time per batch. It may also become small if, instead of computing updated kriging means, variances or covariances at a single point $\mathbf{x}$, we compute them at M different $\mathbf{x}$'s, with M a large number.

To summarize, if we want to compute p kriging updates (one per new batch) at M different locations, with $p, M \gg n \gg r$, the cost of the precomputations is $O(prn^2 + Mn^2)$ and this cost is dominated by the cost to compute the updates of $O(pMrn)$. In a classic algorithm and still in the case where $p, M \gg n \gg r$ the dominating term would be of $O(pMn^2)$. In this setting, the kriging update formulas yield a gain of a factor n/r, which can be interesting as n is often equal to a few hundreds while r is typically small (i.e. lower than 10). More details on the exact algorithms and complexity calculations are provided in Appendix C, Supplementary Material. The kriging update equations are a key milestone to reduce the computation cost of so-called Stepwise Uncertainty Reduction strategies (see, Part II).

2.3 Global Optimization using kriging, the Multi-points Expected Improvement

The Efficient Global Optimization (EGO) algorithm is a global optimization algorithm, introduced by Jones et al. [1998], which relies on kriging and on an Expected-Improvement (EI) criterion [Mockus et al., 1978]. The algorithm is briefly detailed in Section 1.1 (see, Algorithm 1).

EGO consists in using a kriging metamodel to represent the target function $f : \mathbb{X} \mapsto \mathbb{R}$ of interest. f is supposed to be a particular realization of a random process

ξ which is assumed Gaussian for tractability. The Universal kriging setting is used, with a covariance function $k(\cdot,\cdot)$ estimated, from the observation, among a parametric family of functions.

The use of kriging allows to have a probabilistic framework to study our function f. When n observations of f at points $\mathbf{x}_1,\ldots,\mathbf{x}_n$, $\mathcal{A}_n$, are available, the Expected Improvement is a criterion of interest quantifying the potential gain that would be obtained from the evaluation of an additional point $\mathbf{x}_{n+1}$. In a maximization problem, the criterion is:

$$EI(\mathbf{x}_{n+1}) = \mathbb{E}\left((\xi(\mathbf{x}_{n+1}) - T)_+ | \mathcal{A}_n\right), \tag{2.24}$$

where $(\cdot)_+ := \max(0,\cdot)$ and T is a threshold which is generally equal to the maximum of the observation, $T := \max(\xi(\mathbf{x}_i))$. At each iteration of the EGO algorithm, the newly evaluated point is the point with maximum EI.

Note that Equation (2.24) is rather intuitive in the sense that we want to evaluate the point which hopefully improves as much as possible the current maximum found. The EI criterion is particularly convenient when a Gaussian prior is used for ξ. Indeed, in this case, the full conditional distribution of $\xi(\mathbf{x})$ is known for each $\mathbf{x} \in \mathbb{X}$: $\xi(\mathbf{x})|\mathcal{A}_n \sim \mathcal{N}(m_n(\mathbf{x}), s_n^2(\mathbf{x}))$. As the EI is simply the expectation of the exceedance of a random variable (with known distribution) over a threshold T, it can be calculated with a closed-form expression:

$$EI(\mathbf{x}_{n+1}) = (m_n(\mathbf{x}_{n+1}) - T)\Phi\left(\frac{m_n(\mathbf{x}_{n+1}) - T}{s_n(\mathbf{x}_{n+1})}\right) + s_n(\mathbf{x}_{n+1})\varphi\left(\frac{m_n(\mathbf{x}_{n+1}) - T}{s_n(\mathbf{x}_{n+1})}\right). \tag{2.25}$$

The latter equation can be obtained with standard integral manipulations and allows an easy use of the criterion in an optimization algorithm. If the reader is keen on financial mathematics, he may have noted that a calculation of the Expected Improvement in the case where $\xi(\mathbf{x})$ is LogNormal (and not Gaussian) would lead to the famous Black-Scholes formula for a European call option !

In the Gaussian setting, a lot of effort has been put recently on efficient optimization algorithms using the multi-points Expected Improvement (q-EI) (see,

Schonlau [1997] as well as the recent work of Frazier [2012]; Ginsbourger et al. [2010]; Girdziusas et al. [2012]; Janusevskis et al. [2011]). This criterion allows the use of a parallel (or *batch-sequential*) EGO algorithm, i.e. an algorithm where a batch of $q > 1$ points is evaluated at each iteration, through the maximization of the q-EI. The q-EI is useful when many CPUs are available to evaluate the target function f at q points simultaneously and is a natural generalization of the one-point EI. Let $\mathbf{x}^{(q)} := (\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+q}) \in \mathbb{X}^q$ be a batch of q points. The q-EI is defined as follows:

$$qEI(\mathbf{x}^{(q)}) := \mathbb{E}\left((\max_{i=1,\ldots,q} \xi(\mathbf{x}_{n+i}) - T)_+ | \mathcal{A}_n \right). \tag{2.26}$$

If we denote $\mathbf{Y} = (\xi(\mathbf{x}_{n+1}), \ldots, \xi(\mathbf{x}_{n+q}))$ and remember that $\mathbf{Y}$ conditioned on $\mathcal{A}_n$ has a normal distribution in dimension q with known mean and covariance matrix, calculating the q-EI amounts to finding closed-form expressions for:

$$\mathbb{E}\left((\max_{i=1,\ldots,q} Y_i - T)_+ \right), \tag{2.27}$$

where $\mathbf{Y} \sim \mathcal{N}_q(\mathbf{m}, \mathbf{\Sigma}), \mathbf{m} \in \mathbb{R}^q, \mathbf{\Sigma} \in \mathbb{R}^{q \times q}$. So far, only Monte-Carlo approximation methods were proposed to compute Expression (2.27) [Janusevskis et al., 2011]. It appears that, as detailed in an article of Chevalier and Ginsbourger [2013] in the Appendix B of this manuscript, Expression (2.27) can be computed with a closed-form formula obtained by applying q times the Tallis' formula (1.12). The formula can be seen in Appendix B and its proof stems from the idea that the maximum of the Y_i's has to be one of the Y_i's. Mathematically:

$$\max_{i=1,\ldots,q} Y_i = \sum_{i=1}^{q} Y_i \mathbb{1}_{(Y_i > Y_j \forall j \neq i)}. \tag{2.28}$$

A straightforward application of this idea gives:

$$\begin{aligned} qEI(\mathbf{x}^{(q)}) &= \mathbb{E}\left((\max_{i=1,\dots,q} Y_i - T)_+ \right) \\ &= \sum_{i=1}^{q} \mathbb{E}(Y_i - T)\mathbb{1}_{(Y_i>T, Y_i>Y_j \forall j\neq i)} \\ &= \sum_{i=1}^{q} \mathbb{E}(Y_i - T | Y_i > T, Y_i > Y_j \ \forall j \neq i)\mathbb{P}(Y_i > T, Y_i > Y_j \ \forall j \neq i), \end{aligned}$$

and we see now that each term of the sum above can be computed with Tallis' formula (1.12) on q different well-chosen Gaussian vectors. Indeed:

Proposition 3 *Let $\mathbf{Y} := (Y_1, \dots, Y_q)$ be a Gaussian Vector with mean $\mathbf{m} \in \mathbb{R}^q$ and covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{q\times q}$. For $k \in \{1, \dots, q\}$ consider the Gaussian vectors $\mathbf{Z}^{(k)} := (Z_1^{(k)}, \dots, Z_q^{(k)})$ defined as follows:*

$$\begin{aligned} Z_j^{(k)} &:= Y_j - Y_k \ , \ j \neq k \\ Z_k^{(k)} &:= \ -Y_k \end{aligned}$$

Denoting by $\mathbf{m}^{(k)}$ and $\mathbf{\Sigma}^{(k)}$ the mean and covariance matrix of $\mathbf{Z}^{(k)}$, and defining the vector $\mathbf{b}^{(k)} \in \mathbb{R}^q$ by $b_k^{(k)} = -T$ and $b_j^{(k)} = 0$ if $j \neq k$, the q-EI of $\mathbf{x}^{(q)}$ is:

$$qEI(\mathbf{x}^{(q)}) = \sum_{k=1}^{q} \left((m_k - T)p_k + \sum_{i=1}^{q} \mathbf{\Sigma}_{ik}^{(k)} \varphi_{m_i^{(k)}, \mathbf{\Sigma}_{ii}^{(k)}}(b_i^{(k)}) \Phi_{q-1}\left(\mathbf{c}_{.i}^{(k)}, \mathbf{\Sigma}_{.i}^{(k)} \right) \right) \quad (2.29)$$

where:

- $p_k := \mathbb{P}(\mathbf{Z}^{(k)} \leq \mathbf{b}^{(k)}) = \Phi_q(\mathbf{b}^{(k)} - \mathbf{m}^{(k)}, \mathbf{\Sigma}^{(k)})$.
  p_k is actually the probability that Y_k exceeds T and $Y_k = \max_{j=1,\dots,q} Y_j$.
- $\mathbf{c}_{.i}^{(k)} \in \mathbb{R}^{q-1}$ *has general term* $(b_j^{(k)} - m_j^{(k)}) - (b_i^{(k)} - m_i^{(k)})\frac{\mathbf{\Sigma}_{ij}^{(k)}}{\Sigma_{ii}^{(k)}}$, *with* $j \neq i$
- $\mathbf{\Sigma}_{.i}^{(k)} \in \mathbb{R}^{(q-1)\times(q-1)}$ *is the conditional covariance matrix of the random vector* $\mathbf{Z}_{-i}^{(k)} := (Z_1^{(k)}, \dots, Z_{i-1}^{(k)}, Z_{i+1}^{(k)}, \dots, Z_q^{(k)})$ *knowing* $Z_i^{(k)}$.

Examples of use of the q-EI are proposed in Appendix B, as well as comparisons of parallel optimization strategies using the qEI. Figure 2.2 gives a simple example

of use of the q-EI in a batch-sequential EGO algorithm. For this example we chose a function in dimension 1 and $q = 2$ in order to be able to represent (right plots) the value of the q-EI. Though, in an EGO algorithm, the covariance parameters are often estimated (e.g. using Maximum Likelihood, Roustant et al. [2012]) and plugged-in, they are assumed to be known for this example.

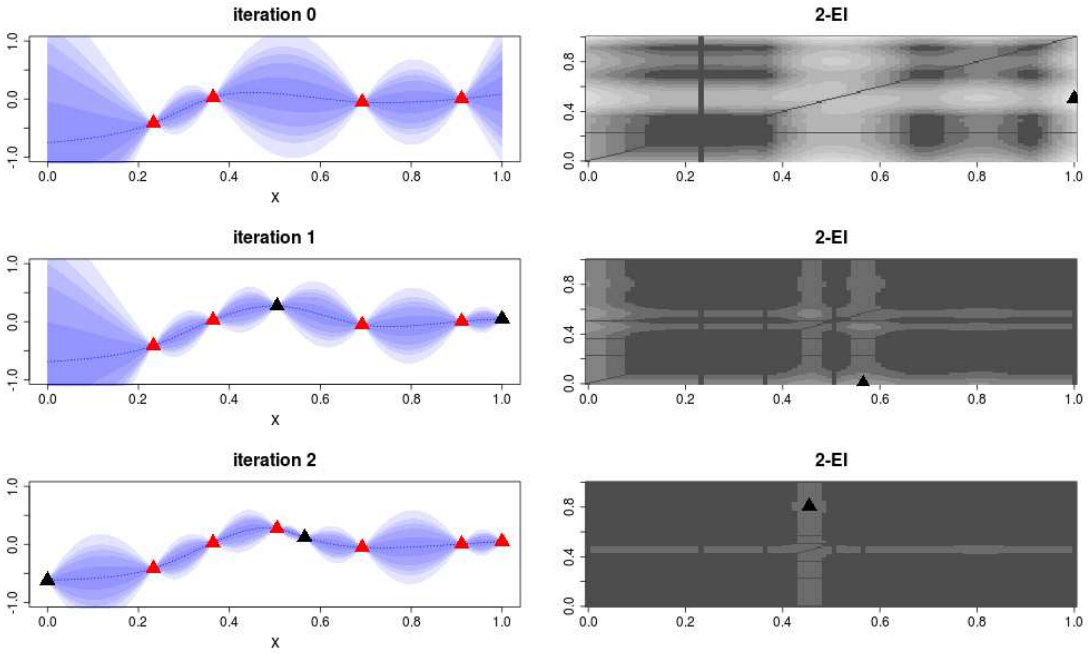

Figure 2.2: 2 iterations of a batch-sequential optimization strategy using the q-EI, for $q = 2$, on a $1d$-function. Left: kriging mean (blue dotted line) and confidence intervals. Right: function q-EI($\mathbf{x}^{(2)}$) where $\mathbf{x}^{(2)}$ is a batch of two points. At each iteration, the batch evaluated is the black triangle on the right plots.

Part II

The Stepwise Uncertainty Reduction paradigm

Chapter 3

Stepwise Uncertainty Reduction (SUR)

3.1 Introduction

Let $f : \mathbb{X} \mapsto \mathbb{R}$ be a real-valued function defined over a compact domain $\mathbb{X} \subset \mathbb{R}^d$, $d \geq 1$. A Stepwise Uncertainty Reduction (SUR) strategy is a sequential evaluation strategy of f which aims at constructing a sequence of evaluation locations, $\mathbf{X}_1, \mathbf{X}_2, \ldots$ in order to reduce the *uncertainty* (as discussed later) on a given quantity of interest. A suitable definition for the term "strategy" can be found in Ginsbourger and Le Riche [2010][1]. The term SUR refers to the work of Geman and Jedynak [1996] in satellite image analysis and has been used to solve different types of problems [Bettinger, 2009; Villemonteix, 2008] on expensive-to-evaluate functions.

Here, we work in the particular setting where f is a sample path of a random process ξ. The uncertainty is defined as a function, H which maps any finite sequence of observations (i.e. a sequence of couples $(\mathbf{X}, \xi(\mathbf{X}))$) to $\mathbb{R}_+$. The function H restricted to sequences of length n is denoted H_n, so that, when n observations $\mathcal{A}_n := \{(\mathbf{X}_i, \xi(\mathbf{X}_i))_{1 \leq i \leq n}\}$ are available we denote $H_n(\mathcal{A}_n) := H(\mathcal{A}_n)$ the uncer-

[1] In short, a strategy will be a sequence of functions where each function $f_i(\cdot)$ depends on the past evaluations at locations $\mathbf{x}_1, \ldots, \mathbf{x}_i$ and returns the next evaluation point $\mathbf{x}_{i+1}$. See Ginsbourger and Le Riche [2010] for more details

tainty at time n. We also work in the particular setting of a *limited evaluation budget* for f. At time n, we assume that we have p evaluations left. When $\mathcal{A}_n$ is known the goal of a SUR strategy is to find p **optimally chosen** locations $\mathbf{X}_{n+1}, \ldots, \mathbf{X}_{n+p}$ such that the residual uncertainty $H(\mathcal{A}_{n+p}) = H_{n+p}(\mathcal{A}_{n+p})$ is as small as possible.

At this stage, we need to make some important remarks. First the locations $\mathbf{X}_{n+1}, \ldots, \mathbf{X}_{n+p}$ are upper-cased to emphasize that they are random variables. Indeed in a SUR strategy, at step n, the location $\mathbf{X}_{n+2}$ may depend on the (unknown) value of ξ at location $\mathbf{X}_{n+1}$. More precisely, and for all $n > 0$, $\mathbf{X}_{n+1}$ is going to be a $\sigma(\mathcal{A}_n)$-measurable random variable where $\sigma(\mathcal{A}_n)$ is the σ-algebra generated by $\mathcal{A}_n$. Second, the first remark implies that, given n observations $\mathcal{A}_n$, the quantity $H_{n+p}(\mathcal{A}_{n+p})$ is also a random variable. Thus, instead of trying to construct the sequence $\mathbf{X}_{n+1}, \ldots, \mathbf{X}_{n+p}$ in order to have a low future uncertainty $H_{n+p}(\mathcal{A}_{n+p})$, the goal of the SUR strategy is to have a low future uncertainty *in expectation*. When n observations are available, we aim at finding a sequence of p random variables $\mathbf{X}^\star_{n+1}, \ldots, \mathbf{X}^\star_{n+p}$ minimizing, over all random variable sequences of length p, $\mathbf{X}_{n+1}, \ldots, \mathbf{X}_{n+p}$ the expected future uncertainty:

$$\mathbb{E}\left(H_{n+p}(\mathcal{A}_{n+p})|\mathcal{A}_n\right) \tag{3.1}$$

The SUR strategy finding the optimal locations given above will be called *optimal strategy*.
We will often use the notation $\mathbb{E}_n(\cdot) := \mathbb{E}(\cdot|\mathcal{A}_n)$. Moreover, when $\mathcal{A}_n$ is known we will also use the simplified notations:

$$\begin{aligned}\mathbb{E}_n(H_{n+p}(\mathcal{A}_{n+p})) :=& \mathbb{E}_n(H_{n+p}((\mathbf{X}_{n+1}, \xi(\mathbf{X}_{n+1})), \ldots, (\mathbf{X}_{n+p}, \xi(\mathbf{X}_{n+p})))) &(3.2)\\ :=& \mathbb{E}_n(H_{n+p}(\mathbf{X}_{n+1}, \ldots, \mathbf{X}_{n+p})). &(3.3)\end{aligned}$$

The notation in Equation (3.3) is abusive as the randomness of the future uncertainty at time $n+p$ comes from the randomness of $\mathbf{X}_{n+1}, \ldots, \mathbf{X}_{n+p}$ but also of $\xi(\mathbf{X}_{n+1}), \ldots, \xi(\mathbf{X}_{n+p})$. However, we will use this shorter notation as there is no ambiguity that, when n observations are known, the responses $\xi(\mathbf{X}_{n+i}), 1 \leq i \leq p$

are random. At time n, the function:

$$J_n(\mathbf{X}_{n+1}, \ldots, \mathbf{X}_{n+p}) := \mathbb{E}_n(H_{n+p}(\mathbf{X}_{n+1}, \ldots, \mathbf{X}_{n+p})) \tag{3.4}$$

will be called *sampling criterion*. Finding the optimal locations $(\mathbf{X}^\star_{n+1}, \ldots, \mathbf{X}^\star_{n+p})$ amounts to minimizing the sampling criterion. Consequently, once an uncertainty function H is defined, there exist a sampling criterion associated to it and an optimal SUR strategy which consists in sequentially choosing the future evaluation point to minimize the expectation of the future uncertainty. The sampling criteria corresponding to given SUR strategies will sometimes be (abusively) called *SUR criteria*.

3.2 k-steps lookahead optimal SUR strategies

To introduce this Section, we would like to use the example of the Expected Improvement (EI) criterion (see, Equation (2.24)) for the problem of maximizing a function $f : \mathbb{X} \mapsto \mathbb{R}$ considered as a realization of a Gaussian process ξ. It is proven in the next Section that the maximization of the EI is an optimal (1-step lookahead, as defined below) SUR strategy for a specific definition of the uncertainty function which is exhibited. Let us assume in this example that the remaining evaluation budget is $p = 2$. At time n we would like to find the two optimal locations $\mathbf{X}_{n+1}, \mathbf{X}_{n+2}$, maximizing the 2-steps EI defined as follows:

$$EI_2(\mathbf{X}_{n+1}, \mathbf{X}_{n+2}) := \mathbb{E}_n(\max(\xi(\mathbf{X}_{n+1}), \xi(\mathbf{X}_{n+2})) - T_n)_+ \tag{3.5}$$

where $T_n := \max_{1 \leq i \leq n} \xi(\mathbf{x}_i)$. This criterion is different than the multi-points Expected Improvement of Equation (2.26) as here $\mathbf{X}_{n+2}$ is a $\sigma(\mathcal{A}_{n+1})$-measurable random variable (i.e. it depends on $\xi(\mathbf{X}_{n+1})$, which is unknown at time n). It is shown in Ginsbourger and Le Riche [2010] that, though the optimal evaluation location $\mathbf{X}_{n+2}$ is a maximizer of the classical EI at time $n+1$, the optimal $\mathbf{X}_{n+1}$ is not necessarily a maximizer of the EI at time n. In other words, when $p > 1$ evaluations are left, it is not optimal to optimize sequentially p times the classical EI. In this example, finding the location $\mathbf{X}_{n+1}$ amounts to optimize the function,

called EI_2 here, given in Algorithm 2 below. A single evaluation[1] of $EI_2(\cdot)$ requires to run many optimizations over $\mathbb{X}$. Thus the optimization of $EI_2(\cdot)$ over $\mathbb{X}$ is expected to be cumbersome.

Algorithm 2 2-steps EI function: $EI_2(\mathbf{x}_{n+1})$

Require: n evaluations of f, $\mathcal{A}_n$, and a current maximum $y := \max f(\mathbf{x}_i)$.
Require: a location $\mathbf{x}_{n+1}$
Require: a discretization of the possible response $\xi(\mathbf{x}_{n+1})$ to approximate the expectation with respect to this random response, i.e. responses $y_1, \ldots, y_r$ and weights $w_1, \ldots, w_r$.

for i = 1 to r **do**
- assume that $\xi(\mathbf{x}_{n+1}) = y_i$
- find $\mathbf{X}_{n+2}$ maximizing the classical EI with respect to the new maximum, $\max(y, y_i)$.
- record a total Expected Improvement: $EI_i := (y_i - y)_+ + EI(\mathbf{X}_{n+2})$

end for
Return the result $\sum_{i=1}^{r} w_i EI_i$

In general, a SUR strategy which aims at finding the optimal k locations $\mathbf{X}_{n+1}, \ldots, \mathbf{X}_{n+k}$ to reduce the uncertainty[2] is called optimal *k-steps lookahead* SUR strategy. In the literature, a general construction for deriving optimal k-steps lookahead strategies is presented in Bect et al. [2012]; Ginsbourger and Le Riche [2010]. These strategies can be defined mathematically through dynamic programming [Bect et al., 2012]. However, their practical use is difficult. For these reasons, practitioners often use 1-step lookahead strategies even if the remaining evaluation budget p, is greater than 1. In the case where $p > 1$, the sequential use of 1-step lookahead strategies is suboptimal and is sometimes called "myopic" to emphasize that we select the next evaluation point as if it is the last one.

[1] one may remark that it would be useful in this function to also return $\mathbf{X}_{n+2}(y_i, \mathbf{x}_{n+1})$ as, once the location $\mathbf{x}^\star_{n+1}$ is found by maximizing EI_2, one could immediately obtain the location $\mathbf{x}^\star_{n+2}$ at time $n+1$, i.e. after having evaluated the function at location $\mathbf{x}^\star_{n+1}$. However this approach is limited by the size, r, of the discretization of the response (i.e. the number of y_i's).

[2] the term "location" might be a bit confusing for the reader. Indeed, at time n, $\mathbf{X}_{n+1}$ is a measurable random variable. However, $\mathbf{X}_{n+2}$ is function of the unknown $\xi(\mathbf{X}_{n+1})$; $\mathbf{X}_{n+3}$ is a function of $\xi(\mathbf{X}_{n+1}), \mathbf{X}_{n+2}, \xi(\mathbf{X}_{n+2})$ and so on. Thus the locations $\mathbf{X}_{n+2}, \mathbf{X}_{n+3}, \cdots$ are **random** locations.

The use of k-steps lookahead optimal SUR strategies can be generalized in the case where we aim at finding k optimal **batches** of points, $(\mathbf{X}_{n+1}, \ldots, \mathbf{X}_{n+q}), \ldots,$ $(\mathbf{X}_{n+1+q(k-1)}, \ldots, \mathbf{X}_{n+qk})$, to reduce the uncertainty. In these settings, a batch of points is evaluated at the same time, so that the random vector $(\mathbf{X}_{n+q+1}, \ldots, \mathbf{X}_{n+2q})$ is a $\sigma(\mathcal{A}_{n+q})$-measurable random vector, and so on. In this thesis we will use this generalization. However, we will focus only on 1-step lookahead SUR strategies. The latter still offer significant challenges in application and can also be difficult to implement.

3.3 The example of the Expected Improvement

Let us consider the settings of the EGO algorithm presented in Section 2.3. We consider the Expected Improvement criterion (or its multi-points version, the q-EI) to maximize $f : \mathbb{X} \mapsto \mathbb{R}$, considered as a realization of a Gaussian process ξ.

Let us introduce an uncertainty function, H_n , defined as follows:

$$H_n(\mathcal{A}_n) := \mathbb{E}_n(\max_{\mathbf{x}\in\mathbb{X}} \xi(\mathbf{x}) - T_n) \tag{3.6}$$

where $T_n = \max_{i=1,\ldots,n} \xi(\mathbf{x}_i)$. In a maximization problem, such definition for the uncertainty is adapted in the sense that the uncertainty is non-negative, and low if $\xi(\mathbf{x})$ is not likely to get much greater than T_n. Now let us assume that we seek for the batch $(\mathbf{x}^\star_{n+1}, \ldots, \mathbf{x}^\star_{n+q})$[1] minimizing - in expectation - the uncertainty at time $n+q$. We have, for all $(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+q}) \in \mathbb{X}^q$:

$$\begin{aligned}
J_n(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+q}) :=& \mathbb{E}_n(H_{n+q}(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+q})) \\
=& \mathbb{E}_n(\mathbb{E}_{n+q}(\max_{\mathbf{x}\in\mathbb{X}} \xi(\mathbf{x}) - T_{n+q})) \\
=& \mathbb{E}_n(\mathbb{E}_{n+q}(\max_{\mathbf{x}\in\mathbb{X}} \xi(\mathbf{x}) - T_n)) - \mathbb{E}_n((\max_{i=1,\ldots,q} \xi(\mathbf{x}_{n+i}) - T_n)_+) \\
=& \mathbb{E}_n(\max_{\mathbf{x}\in\mathbb{X}} \xi(\mathbf{x}) - T_n) - qEI(\mathbf{x}^{(q)}) \\
=& H_n(\mathcal{A}_n) - qEI(\mathbf{x}^{(q)}).
\end{aligned}$$

[1] written with lower-case letters as we are at time n and deal with a $\sigma(\mathcal{A}_n)-$measurable random vector, i.e. we are in a 1-step lookahead setting

The latter calculation proves that the expectation of the future uncertainty when a batch of q points is added is equal to the current uncertainty minus the q-EI at this batch. Consequently, the maximizer of the q-EI is also the minimizer of J_n[1]. Thus, the maximization of the q-EI is the optimal 1-step lookahead SUR strategy if the uncertainty is defined using Equation (3.6). In this example, it is noticeable that the uncertainty itself is difficult to compute numerically, as it involves the maximum of a Gaussian process over a non-countable set. However the expected uncertainty reduction when a batch is added (i.e., the q-EI) can be computed with a closed-form expression (see, Section 2.3). We here have an example where a SUR strategy can be efficiently applied without having to compute the current uncertainty. A similar example will come in the next Chapter.

3.4 SUR strategies in inversion

A significant part of the contribution of this thesis relates to the efficient use of SUR strategies for inversion problems (as defined later). The contributions in inversion are detailed in the next Chapter. Before that, we detail in this Section the problems at hand and motivate our contributions.

We are in the settings of the Universal kriging (see, Chapter 2) and deal with a function $f : \mathbb{X} \mapsto \mathbb{R}$[2] , where $\mathbb{X}$ is a compact subset of $\mathbb{R}^d, d \geq 1$ (often, a hyper-rectangle). Our function of interest f is assumed to be a realization of a random process ξ, which will usually be considered as Gaussian with known (or estimated and plugged-in) covariance function $k(\cdot, \cdot)$. We also assume, like in the previous Chapters, that f has already been evaluated at n locations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and denote $\mathcal{A}_n$ the set of the n couples $(\mathbf{x}_i, f(\mathbf{x}_i))_{1 \leq i \leq n}$. Now, let us consider a set $D \subset \mathbb{R}$. We are interested in the set:

$$\mathbf{\Gamma}^\star := \{\mathbf{x} \in \mathbb{X} : f(\mathbf{x}) \in D\}, \tag{3.7}$$

More precisely, many problems involving the set $\mathbf{\Gamma}^\star$ can be formulated:

[1] Note that the assumption that the random process ξ is Gaussian is not used to obtain this result.

[2] in some cases, our work will be valid for functions valued in $\mathbb{R}^p$ with $p > 1$.

- Determine the volume of $\mathbf{\Gamma}^\star$, $\alpha^\star := \mathbb{P}_{\mathbb{X}}(\Gamma^\star)$, where $\mathbb{P}_{\mathbb{X}}$ is a fixed measure. Though it is not necessary, $\mathbb{P}_{\mathbb{X}}$ will often be a probability measure quantifying the uncertainties on the inputs. The problem described is often referred to as a *probability of failure* estimation problem [Bect et al., 2012].

- Determine the set $\mathbf{\Gamma}^\star$ itself, i.e.: construct a classifier which tells us, for every arbitrary point $\mathbf{x} \in \mathbb{X}$, if $\mathbf{x}$ is in $\mathbf{\Gamma}^\star$ or not.

- In the particular case where $D = [T, \infty)$, with T a given threshold: find the contour line $\mathcal{C}^\star := \{\mathbf{x} \in \mathbb{X} : f(\mathbf{x}) = T\}$. This problem is often referred to as a contour line (or level set) estimation problem [Picheny, 2009; Picheny et al., 2010].

The problems described above are rather close and in this thesis we group them using the term *inversion*.

Dealing with these problems using a kriging metamodel (and a Gaussian prior) has already been done in the literature (see, Bichon et al. [2008]; Echard et al. [2011]; Ranjan et al. [2008] as well as Appendix E and Bect et al. [2012] for a review of some sampling strategies for these problems). The methods proposed so far have strengths and weaknesses. For example, the strategies proposed in Bichon et al. [2008]; Echard et al. [2011]; Ranjan et al. [2008] are simple, fast, and easy to implement (see, Appendix E for a speed comparison between some sampling criteria). However, none of these criteria corresponds to an optimal SUR strategy. Recently, Bect et al. [2012] showed that the 1-step lookahead SUR strategies tend to outperform in application the "simpler" strategies proposed by Bichon et al. [2008]; Echard et al. [2011]; Ranjan et al. [2008].

Our contribution on SUR strategies for inversion can be split into four categories:

1. Generalization of SUR strategies in the case where we evaluate $q > 1$ points at a time. Batch-sequential 1-step lookahead SUR strategies are presented. [Chevalier et al., 2013a].

2. Closed-form expression to efficiently compute SUR criteria: in many cases the use of 1-step lookahead SUR strategies is complicated because of computation time. Thus, similarly to the k-steps lookahead SUR strategies, even

if the strategy "exists", it might be difficult to use in application. In the Appedix C we present a contribution [Chevalier et al., 2013a] where the use of two SUR strategy introduced in Bect et al. [2012] becomes possible in a reasonable time, through closed-form formulas. The formulas apply for both sequential and batch-sequential strategies.

3. Introduction of new SUR strategies for inversion: notions from random set theory are used to define the conditional "variance" of a random set. We use these notions to define our uncertainty function and derive the corresponding optimal SUR strategy. The 1-step lookahead SUR strategy is implemented. This contribution is detailed in the next Chapter and in a paper given in Appendix D [Chevalier et al., 2013b].

4. Implementation in R language of the KrigInv package, which allows the use of the studied strategies. Some auxiliary problems relative to numerical computations are discussed. A paper dealing with KrigInv (Chevalier et al. [2013d]) is given in Appendix E.

Chapter 4

Contribution in SUR strategies for inversion

In this Chapter, we use the setting and notations given in Section 3.4.

4.1 Fast computation of batch-sequential existing SUR strategies

Let us consider a function of interest $f : \mathbb{X} \subset \mathbb{R}^d \mapsto \mathbb{R}$, and n evaluations $\mathcal{A}_n := (\mathbf{x}_i, f(\mathbf{x}_i))_{1 \leq i \leq n}$. We are in the Universal kriging setting and f is a sample path of a Gaussian process ξ. Let $D \subset \mathbb{R}$ be a closed set. We recall that we are interested in the set:

$$\mathbf{\Gamma}^\star := \{\mathbf{x} \in \mathbb{X} : f(\mathbf{x}) \in D\}. \tag{4.1}$$

Let us define the set:

$$\mathbf{\Gamma} := \{\mathbf{x} \in \mathbb{X} : \xi(\mathbf{x}) \in D\}. \tag{4.2}$$

The latter set is random as ξ is a random process. Some realizations of $\mathbf{\Gamma}$ conditioned on $\mathcal{A}_n$ may be obtained using conditional simulations of ξ (as an example, see Figure 4.2).

Let us define $\alpha := \mathbb{P}_{\mathbb{X}}(\mathbf{\Gamma})$ where $\mathbb{P}_{\mathbb{X}}$ is a given σ-finite measure on $\mathbb{X}$. As mentioned in Bect et al. [2012] the conditional variance of α is a natural way to quantify uncer-

tainties on the real excursion volume $\alpha^\star$. We write the corresponding uncertainty function as follows:

$$H_n(\mathcal{A}_n) := Var_n(\alpha), \tag{4.3}$$

where $Var_n(\cdot) := Var(\cdot|\mathcal{A}_n)$. The computation of this uncertainty may require Monte-Carlo approximations using conditional simulations of Gaussian random fields. For this reason, the 1-step lookahead SUR strategy associated to this uncertainty was judged impractical in Bect et al. [2012], so that the authors proposed a different definition for the uncertainty function:

$$\widetilde{H}_n(\mathcal{A}_n) := \int_{\mathbb{X}} p_n(\mathbf{x})(1 - p_n(\mathbf{x}))\mathrm{d}\mathbb{P}_{\mathbb{X}}(\mathbf{x}) \tag{4.4}$$

where $p_n(\mathbf{x}) := \mathbb{P}(\mathbf{x} \in \mathbf{\Gamma}|\mathcal{A}_n)$ is called excursion probability function or, sometimes, coverage probability function.
The definition of the uncertainty given in Equation (4.4) is obtained through an application of Cauchy-Schwarz inequality, using the fact that

$$\alpha = \int_{\mathbb{X}} \mathbb{1}_{(\xi(\mathbf{x})\in D)}\mathrm{d}\mathbb{P}_{\mathbb{X}}(\mathbf{x}) \tag{4.5}$$

$$\mathbb{E}_n(\alpha) = \int_{\mathbb{X}} p_n(\mathbf{x})\mathrm{d}\mathbb{P}_{\mathbb{X}}(\mathbf{x}). \tag{4.6}$$

The calculations are the following:

$$\begin{aligned}
Var_n(\alpha) &= \mathbb{E}_n((\alpha - \mathbb{E}_n(\alpha))^2) \\
&= \mathbb{E}_n\left(\left(\int_{\mathbb{X}} (\mathbb{1}_{(\xi(\mathbf{x})\in D)} - p_n(\mathbf{x}))\mathrm{d}\mathbb{P}_{\mathbb{X}}(\mathbf{x})\right)^2\right) \\
&\leq \mathbb{P}_{\mathbb{X}}(\mathbb{X})\ \mathbb{E}_n\left(\int_{\mathbb{X}} (\mathbb{1}_{(\xi(\mathbf{x})\in D)} - p_n(\mathbf{x}))^2\mathrm{d}\mathbb{P}_{\mathbb{X}}(\mathbf{x})\right) \\
&= \mathbb{P}_{\mathbb{X}}(\mathbb{X})\ \widetilde{H}_n(\mathcal{A}_n)
\end{aligned}$$

The function $p_n(\cdot)$ can be interpreted as the probability that the random set $\mathbf{\Gamma}$ contains $\mathbf{x}$, or, equivalently, that $\xi(\mathbf{x}) \in D$. As $\xi(\mathbf{x})|\mathcal{A}_n \sim \mathcal{N}(m_n(\mathbf{x}), s_n^2(\mathbf{x}))$, $p_n(\mathbf{x})$

has a simple expression in the particular case where $D = [T, \infty)$:

$$p_n(\mathbf{x}) = \Phi\left(\frac{m_n(\mathbf{x}) - T}{s_n(\mathbf{x})}\right). \tag{4.7}$$

More generally, if D is a (closed) finite union of disjoint intervals written as:

$$D = \bigcup_{i=1}^{l} [a_i, b_i],\ a_i \in \mathbb{R} \cup \{-\infty\},\ b_i \in \mathbb{R} \cup \{+\infty\},\ a_1 < b_1 < \ldots < a_l < b_l \tag{4.8}$$

then $p_n(\mathbf{x})$ can still be computed as:

$$p_n(\mathbf{x}) = \sum_{i=1}^{l} \Phi\left(\frac{m_n(\mathbf{x}) - a_i}{s_n(\mathbf{x})}\right) - \Phi\left(\frac{m_n(\mathbf{x}) - b_i}{s_n(\mathbf{x})}\right). \tag{4.9}$$

with the convention $\Phi(\infty) = 1$ and $\Phi(-\infty) = 0$.

Equation (4.4) introduces a definition of the uncertainty function, $\widetilde{H}_n$, which is suitable for many reasons. First, the inequality in Equation (4.4) ensures that, if $\widetilde{H}_n$ goes to zero when new observations are sequentially added then H_n goes to zero as well. Second, $\widetilde{H}_n$ can be interpreted easily as it is small if, for all $\mathbf{x} \in \mathbb{X}$, $p_n(\mathbf{x})$ is close to either 0 or 1. In other words, if for every $\mathbf{x}$ we are able to decide if $\mathbf{x}$ is in the excursion set (i.e., $p_n(\mathbf{x})$ close to 1) or not ($p_n(\mathbf{x})$ close to 0) then $\widetilde{H}_n$ is low. On the contrary if there exist large (in the sense of their volume) zones where $p_n(\mathbf{x}) \approx 1/2$ then $\widetilde{H}_n$ is high, as $p(1-p)$ takes its maximum value for $p = 1/2$.

The optimal 1-step lookahead SUR criteria associated to the uncertainties H_n and $\widetilde{H}_n$ are respectively:

$$J_n(\mathbf{x}_{n+1}) = \mathbb{E}_n(Var_{n+1}(\alpha)) \tag{4.10}$$

$$\widetilde{J}_n(\mathbf{x}_{n+1}) = \mathbb{E}_n\left(\int_{\mathbb{X}} p_{n+1}(\mathbf{x})(1 - p_{n+1}(\mathbf{x}))\mathrm{d}\mathbb{P}_{\mathbb{X}}(\mathbf{x})\right), \tag{4.11}$$

where the expectation is taken with respect to the random response $\xi(\mathbf{x}_{n+1})$. As explained previously, in Bect et al. [2012], the criterion J_n is considered intractable. The criterion $\widetilde{J}_n$ is implemented and the expectation is handled through a Monte-

Carlo approximation which considers a finite number of possible responses $\xi(\mathbf{x}_{n+1})$.

A first contribution in this thesis is to propose a natural batch-sequential generalization of these two SUR criteria (i.e. criteria which depend on $q > 1$ points), and to provide closed form expressions allowing their computation for both $q = 1$ and $q > 1$. This contribution, detailed in Chevalier et al. [2013a], can be found in the Appendix C and is summarized in this Section.
The 1-step lookahead SUR criteria which sample a batch of $q > 1$ points at each iteration are given below:

$$J_n(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+q}) = \mathbb{E}_n(Var_{n+q}(\alpha)) \tag{4.12}$$

$$\widetilde{J}_n(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+q}) = \mathbb{E}_n\left(\int_{\mathbb{X}} p_{n+q}(\mathbf{x})(1 - p_{n+q}(\mathbf{x}))\mathrm{d}\mathbb{P}_{\mathbb{X}}(\mathbf{x})\right). \tag{4.13}$$

Using the notation $\mathbf{x}^{(q)} := (\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+q})$, and assuming that $D = [T, \infty)$ for some $T \in \mathbb{R}$, our "fast" closed-form expressions are:

$$J_n(\mathbf{x}^{(q)}) = \gamma_n - \int_{\mathbb{X}\times\mathbb{X}} \Phi_2\left(\begin{pmatrix} a(\mathbf{z}_1) \\ a(\mathbf{z}_2) \end{pmatrix}, \begin{pmatrix} c(\mathbf{z}_1) & d(\mathbf{z}_1, \mathbf{z}_2) \\ d(\mathbf{z}_1, \mathbf{z}_2) & c(\mathbf{z}_2) \end{pmatrix}\right) \mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{z}_1)\mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{z}_2) \tag{4.14}$$

$$\widetilde{J}_n(\mathbf{x}^{(q)}) = \int_{\mathbb{X}} \Phi_2\left(\begin{pmatrix} a(\mathbf{x}) \\ -a(\mathbf{x}) \end{pmatrix}, \begin{pmatrix} c(\mathbf{x}) & 1 - c(\mathbf{x}) \\ 1 - c(\mathbf{x}) & c(\mathbf{x}) \end{pmatrix}\right) \mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{x}) \tag{4.15}$$

where:

- $\Phi_2(\cdot, M)$ is the c.d.f. of the centered bivariate Gaussian distribution with covariance matrix M,
- $a(\mathbf{x}) := (m_n(\mathbf{x}) - T)/s_{n+q}(\mathbf{x})$,
- $\mathbf{b}(\mathbf{x}) := \frac{1}{s_{n+q}(\mathbf{x})}\boldsymbol{\Sigma}^{-1}(k_n(\mathbf{x}, \mathbf{x}_{n+1}), \ldots, k_n(\mathbf{x}, \mathbf{x}_{n+q}))^\top$,
- $c(\mathbf{x}) := 1 + \mathbf{b}(\mathbf{x})^\top\boldsymbol{\Sigma}\mathbf{b}(\mathbf{x}) = s_n^2(\mathbf{x})/s_{n+q}^2(\mathbf{x})$
- $d(\mathbf{z}_1, \mathbf{z}_2) := \mathbf{b}(\mathbf{z}_1)^\top\boldsymbol{\Sigma}\mathbf{b}(\mathbf{z}_2)$,
- $\boldsymbol{\Sigma}$ is the covariance matrix of $(\xi(\mathbf{x}_{n+1}), \ldots, \xi(\mathbf{x}_{n+q}))^\top$ conditional on $\mathcal{A}_n$,

- γ_n is a constant in the sense that it does not depend on $\mathbf{x}^{(q)}$.

Note that the expressions above are only proven when $D = [T, \infty)$. However, they can be extended to the more general case where D is a finite union of intervals (see, Equation (4.8)).

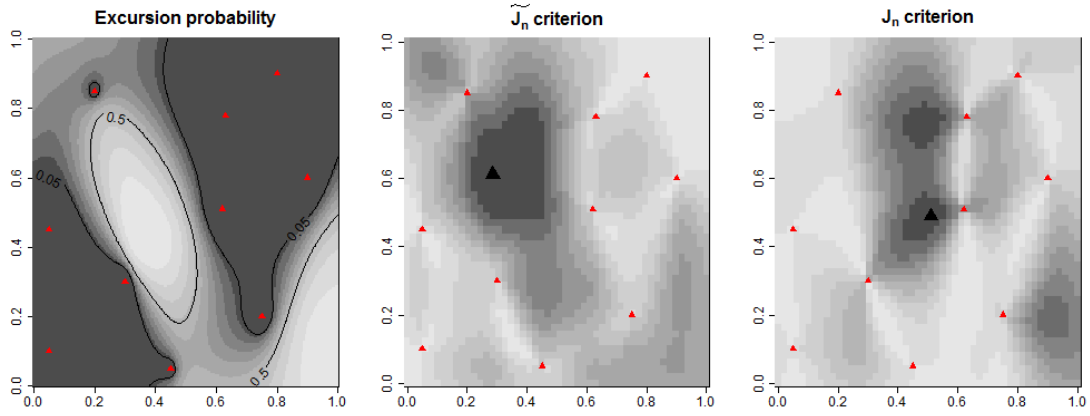

Figure 4.1: Left: excursion probability after 10 evaluation of a 2d test function. Middle: $\widetilde{J}_n(\mathbf{x}^{(q)})$ function, with $q = 1$ and its optimum (black triangle). Right: $J_n(\mathbf{x}^{(q)})$ function and its optimum.

Figure 4.1 gives an example of use of these SUR strategies on a 2d test function which is the Branin-Hoo function multiplied by a factor -1. We explain in Appendix E that SUR strategies which use $\widetilde{J}_n$ tend to spread points in sparse regions with $p_n(\mathbf{x}) \approx 1/2$ while strategies using J_n tend to surround the excursion set in order to control its volume.

Though quick proofs for Equations (4.14) and (4.15) can be found in Appendix C, we would like to spend some time to detail how these expressions were obtained. The explanations are here only given for the $\widetilde{J}_n$ criterion but are similar for J_n. Our starting point when attempting to compute the expectation of a random variable (here: $p_{n+q}(\mathbf{x})(1 - p_{n+q}(\mathbf{x}))$, for some $\mathbf{x} \in \mathbb{X}$) with respect to the response $\xi(\mathbf{x}^{(q)}) := (\xi(\mathbf{x}_{n+1}), \ldots, \xi(\mathbf{x}_{n+q}))$ is to study how the random variable depends on these (Gaussian) responses. With our settings, the use of the kriging update formulas is natural because the random variable $p_{n+q}(\mathbf{x})(1 - p_{n+q}(\mathbf{x}))$ is a functional

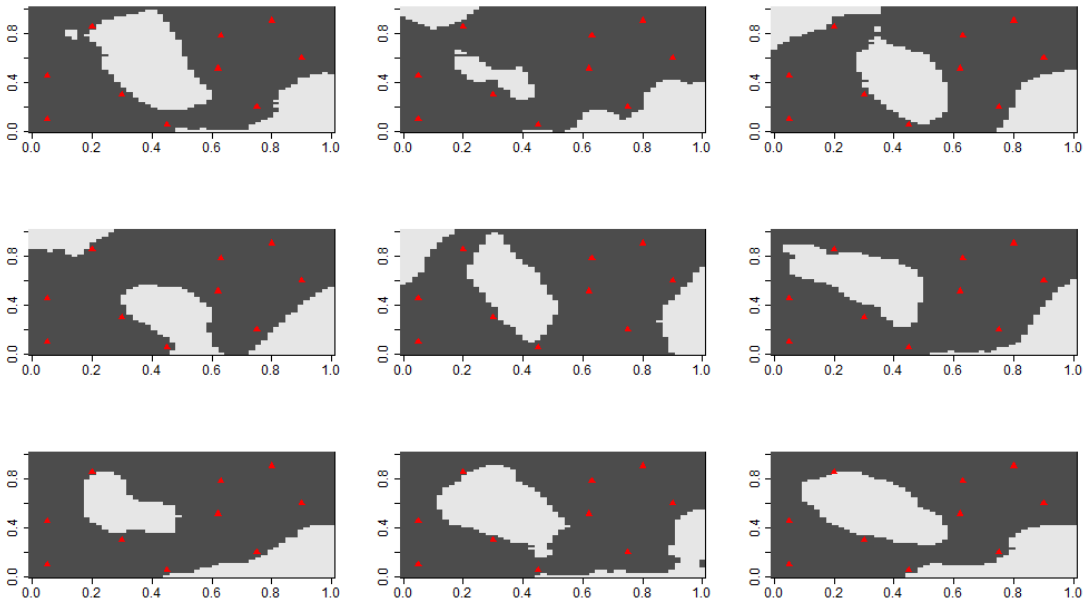

Figure 4.2: Nine conditional realizations of the random set $\mathbf{\Gamma}$.

of $\xi(\mathbf{x}^{(q)})$. At time $n+q$, $\xi(\mathbf{x})$ has a Gaussian distribution with parameters obtained from the kriging update formulas. In particular, the updated kriging mean, $m_{n+q}(\mathbf{x})$ can be written in a form which explicitly shows its dependence on $\xi(\mathbf{x}^{(q)})$:

$$m_{n+q}(\mathbf{x}) = m_n(\mathbf{x}) + \boldsymbol{\lambda}_{\text{new}}(\mathbf{x})^\top \xi_{\text{c}}(\mathbf{x}^{(q)}) \tag{4.16}$$

where $\boldsymbol{\lambda}_{\text{new}}(\mathbf{x})$ are kriging weights given in Equation (2.20) and $\xi_{\text{c}}(\mathbf{x}^{(q)})$ is the random vector of the centered responses $(\xi(\mathbf{x}_{n+1}) - m_n(\mathbf{x}_{n+1}), \ldots, \xi(\mathbf{x}_{n+q}) - m_n(\mathbf{x}_{n+q}))^\top$ with conditional distribution $\mathcal{N}_q(\mathbf{0}, \boldsymbol{\Sigma})$. Thus, from the simple expression (4.7) of the excursion probability, our random variable at hand can be written simply in function of $\mathbf{U} := \xi_{\text{c}}(\mathbf{x}^{(q)})$:

$$p_{n+q}(\mathbf{x})(1 - p_{n+q}(\mathbf{x})) = \Phi\left(a(\mathbf{x}) + \mathbf{b}(\mathbf{x})^\top \mathbf{U}\right) \Phi\left(-a(\mathbf{x}) - \mathbf{b}(\mathbf{x})^\top \mathbf{U}\right), \tag{4.17}$$

where $a(\mathbf{x})$ and $\mathbf{b}(\mathbf{x})$ are defined in Equations (4.14), (4.15). Once this expression is established, it is possible to compute the expectation of $p_{n+q}(\mathbf{x})(1 - p_{n+q}(\mathbf{x}))$

in the case where $\mathbf{U}$ has a centered multivariate normal distribution. Indeed, let $(N_1, N_2)^\top \sim \mathcal{N}_2(\mathbf{0}, I_2)$ be a Gaussian random vector independent from $\mathbf{U}$. A starting point is to remark that, trivially :

$$\mathbb{E}(\mathbb{P}(N_1 \leq a(\mathbf{x}) + \mathbf{b}(\mathbf{x})^\top \mathbf{U}, N_2 \leq -a(\mathbf{x}) - \mathbf{b}(\mathbf{x})^\top \mathbf{U})) \tag{4.18}$$

$$= \mathbb{P}(N_1 \leq a(\mathbf{x}) + \mathbf{b}(\mathbf{x})^\top \mathbf{U}, N_2 \leq -a(\mathbf{x}) - \mathbf{b}(\mathbf{x})^\top \mathbf{U}) \tag{4.19}$$

$$= \Phi_2\left(\begin{pmatrix} a(\mathbf{x}) \\ -a(\mathbf{x}) \end{pmatrix}, \begin{pmatrix} c(\mathbf{x}) & 1 - c(\mathbf{x}) \\ 1 - c(\mathbf{x}) & c(\mathbf{x}) \end{pmatrix}\right) \tag{4.20}$$

However, Expression (4.18) can also be written differently, in an integral which considers all the possibles responses of the random vector $\mathbf{U}$:

$$\mathbb{E}(\mathbb{P}(N_1 \leq a(\mathbf{x}) + \mathbf{b}(\mathbf{x})^\top \mathbf{U}, N_2 \leq -a(\mathbf{x}) - \mathbf{b}(\mathbf{x})^\top \mathbf{U})) \tag{4.21}$$

$$= \int_{\mathbb{R}^q} \mathbb{P}(N_1 \leq a(\mathbf{x}) + \mathbf{b}(\mathbf{x})^\top \mathbf{u}, N_2 \leq -a(\mathbf{x}) - \mathbf{b}(\mathbf{x})^\top \mathbf{u}) \Psi(\mathrm{d}\mathbf{u}) \tag{4.22}$$

$$= \int_{\mathbb{R}^q} \mathbb{P}(N_1 \leq a(\mathbf{x}) + \mathbf{b}(\mathbf{x})^\top \mathbf{u}) \mathbb{P}(N_2 \leq -a(\mathbf{x}) - \mathbf{b}(\mathbf{x})^\top \mathbf{u}) \Psi(\mathrm{d}\mathbf{u}) \tag{4.23}$$

$$= \int_{\mathbb{R}^q} \Phi(a(\mathbf{x}) + \mathbf{b}(\mathbf{x})^\top \mathbf{u}) \Phi(-a(\mathbf{x}) - \mathbf{b}(\mathbf{x})^\top \mathbf{u}) \Psi(\mathrm{d}\mathbf{u}) \tag{4.24}$$

$$= \mathbb{E}\left(\Phi\left(a(\mathbf{x}) + \mathbf{b}(\mathbf{x})^\top \mathbf{U}\right) \Phi\left(-a(\mathbf{x}) - \mathbf{b}(\mathbf{x})^\top \mathbf{U}\right)\right) \tag{4.25}$$

where $\Psi(\cdot)$ is the p.d.f. of the random vector $\mathbf{U}$. It is thus possible to start from Expression (4.25) to obtain the result (4.20). The "trick" here is that we get rid of the expectation when we go from (4.18) to (4.19). In this thesis, a similar calculation scheme (basing on the kriging update formulas and managing to get rid of the expectation with respect to $\mathbf{U}$) will be applied for other sampling criteria (see, Sections 4.2, 6.2, F.1).

Note that practical uses of the J_n and $\widetilde{J}_n$ criteria are presented in Appendix C and E. SUR criteria are indeed implemented in the KrigInv package [Chevalier et al., 2013d]. Though it is not done in Appendix C, nor implemented in KrigInv, the presented "trick" is applicable if D takes the form (4.8), i.e. is a union of l disjoint intervals. In that case, Equation (4.17) becomes a sum of at most $(2l)^2$ terms. The expectation of each of these $(2l)^2$ terms at time $n+q$ (i.e. when a

batch is added) requires one call to the Φ_2 function. For example, if $D = [T_1, T_2]$, we need four calls to Φ_2 per integration point $\mathbf{x}$, which is still reasonably fast in applications as fast algorithms exist to compute Φ_2 [Genz, 1992; Genz and Bretz, 2009].

The SUR criteria J_n which decreases the variance of $\alpha = \mathbb{P}_{\mathbb{X}}(\mathbf{\Gamma})$ is a (1-step lookahead) optimal for the problem of evaluating the true excursion's volume $\alpha^\star$ but has some drawbacks if one aims at finding the excursion set $\mathbf{\Gamma}^\star$ itself. There are indeed no guarantees that, if $Var_n(\alpha)$ goes to zero, then all the points are getting perfectly classified, i.e. $p_n(\mathbf{x}) \to 0$ or 1 a.e.[1]. For this reason, the use of the $\widetilde{J}_n$ criterion might be preferable if one aims at finding $\mathbf{\Gamma}^\star$[2]. However, this criterion is derived from an uncertainty function, $\widetilde{H}_n$, obtained by bounding the initial uncertainty $Var_n(\alpha)$. This uncertainty function does not really measure any "variance" of the random set $\mathbf{\Gamma}$ and is thus not completely satisfying. This problem motivates the next Section, where we aim at constructing a SUR criterion from a new uncertainty measure, which will be a "variance" of the random set $\mathbf{\Gamma}$ itself.

4.2 New SUR strategies using notions from random set theory

We investigate in this thesis new SUR strategies for inversion, based on original definitions for the uncertainty. Let us consider the random set

$$\mathbf{\Gamma} := \{\mathbf{x} \in \mathbb{X} : \xi(\mathbf{x}) \in D\}. \tag{4.26}$$

As said before, conditional realizations of this random set can be simulated from (conditional) simulations of ξ (see, Figure 4.2). When our target function f is a sample realization of ξ, the real unknown excursion set can be seen as one of these

[1] Indeed, a counter example can be constructed. For example we can think about a Gaussian field on $[0,1]^d$ which is constant on a first half of the domain, $[0, 0.5) \times [0,1]^{d-1}$ and constant - with opposite value - on the second half $(0.5, 1] \times [0,1]^{d-1}$. If we take $D = [0, \infty)$ and use the Lebesgue measure, the volume of the random set $\mathbf{\Gamma}$ is constant, equal to $1/2$, and thus has zero variance. However, $p_n(\mathbf{x}) = 1/2$ for all $\mathbf{x} \in [0,1]^d$.

[2] For this criterion we have that if $\widetilde{H}_n$ goes to zero then $p_n(\mathbf{x}) \to 0$ or 1 a.e.

conditional realizations of $\mathbf{\Gamma}$. Therefore, instead of focusing on the volume of $\mathbf{\Gamma}$, it is natural to aim at defining a conditional "variance" for the random set $\mathbf{\Gamma}$ itself.

The book of Molchanov [2005] gives many possible definitions for the variance of a random closed set. In an article [Chevalier et al., 2013b] given in Appendix D, we use the work of Vorob'ev [Vorobyev, 1984; Vorobyev and Lukyanova, 2013] to build a new SUR strategy which aims at decreasing the *Vorob'ev deviation* of the random set $\mathbf{\Gamma}$. Even if precise definitions are given in Appendix D, let us quickly summarize how this "variance" is defined and how the SUR strategy based on it can be used in applications.

Let us consider the excursion probability function, $p_n(\cdot)$, introduced in the previous section (see, Equations (4.7) or (4.9), depending on the form of D). In Molchanov [2005], such function is called *coverage probability* function. The Vorob'ev conditional expectation of the random set $\mathbf{\Gamma}$ is a set obtained by thresholding $p_n(\cdot)$ with a "well-chosen" threshold denoted by T_n. This expectation, called Vorob'ev Expectation, is the set:

$$Q_{n,T_n} := \{\mathbf{x} \in \mathbb{X} : p_n(\mathbf{x}) \geq T_n\}, \tag{4.27}$$

where the threshold T_n is chosen so that the volume of the Vorob'ev Expectation is equal to the expected volume of the random set $\mathbf{\Gamma}$:

$$\mathbb{P}_{\mathbb{X}}(Q_{n,T_n}) = \mathbb{E}_n(\mathbb{P}_{\mathbb{X}}(\mathbf{\Gamma})) \tag{4.28}$$

$$= \int_{\mathbb{X}} p_n \mathrm{d}\mathbb{P}_{\mathbb{X}} := \alpha_{\mathrm{n}}. \tag{4.29}$$

A remarkable fact is that that the set Q_{n,T_n} defined above is the minimizer, among all closed set Q with volume α_n, of the following conditional "variance", called Vorob'ev deviation:

$$Var_n(\mathbf{\Gamma}; Q) := \mathbb{E}\left(\mathbb{P}_{\mathbb{X}}(\mathbf{\Gamma} \Delta Q) | \mathcal{A}_n\right), \tag{4.30}$$

where $A \Delta B$ denotes the symmetric difference between two sets A and B: $A \Delta B :=$

$(A \cup B) \setminus (A \cap B)$. In other words, for all closed set Q with volume α_n we have

$$Var_n(\mathbf{\Gamma}; Q_{n,T_n}) \leq Var_n(\mathbf{\Gamma}; Q). \tag{4.31}$$

A proof for the latter statement can be found in Molchanov [2005], p. 193. From now on, $Var_n(\mathbf{\Gamma}; Q_{n,T_n})$ will be simply denoted by $Var_n(\mathbf{\Gamma})$: the Vorob'ev deviation of $\mathbf{\Gamma}$ at time n. The quantity $Var_n(\mathbf{\Gamma})$ is a natural candidate for our uncertainty function H_n:

$$H_n(\mathcal{A}_n) := Var_n(\mathbf{\Gamma}). \tag{4.32}$$

Moreover, computing $Var_n(\mathbf{\Gamma})$ is not difficult if we get back to its initial definition (4.30). We show in Appendix D that:

$$Var_n(\mathbf{\Gamma}) = \int_{Q_{n,T_n}} (1 - p_n(\mathbf{x}))\mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{x}) + \int_{\mathrm{Q}^{\mathrm{c}}_{\mathrm{n,T_n}}} \mathrm{p_n}(\mathbf{x})\mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{x}). \tag{4.33}$$

where A^c denotes the complementary set of A in $\mathbb{X}$. The Vorob'ev deviation can thus be seen as the sum of "small" (i.e. lower than T_n) excursion probabilities computed over Q^c_{n,T_n} plus the small "non-excursion" probabilities computed over Q_{n,T_n}.

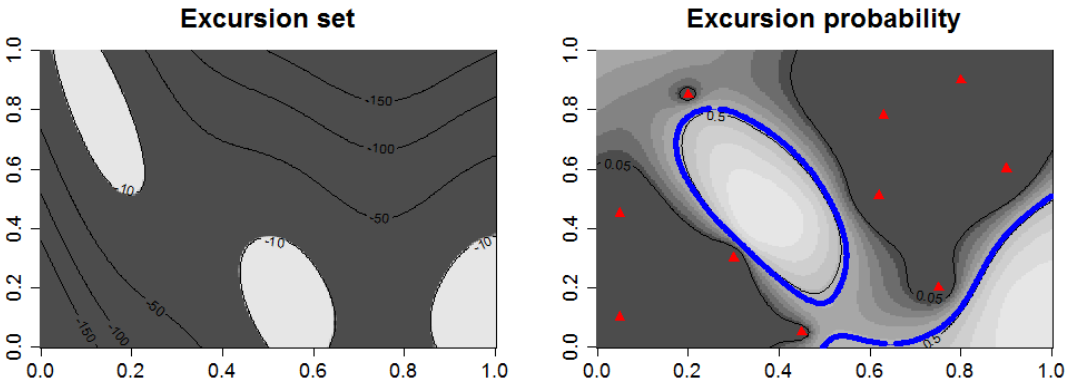

Figure 4.3: Left: excursion set of the Branin function (multiplied by a factor -1) when $D = [-10, \infty)$. Right: excursion probability function $p_n(\cdot)$ obtained from $n = 10$ observations and Vorob'ev expectation.

Figure 4.3 shows an example of computation of the Vorob'ev expectation on the Branin-Hoo function. The Vorob'ev (conditional) expectation is the set of points delimited by the blue line, which excursion probability is greater or equal than the Vorob'ev threshold (here, $T_n \approx 0.45$). Figure 4.4 represents the symmetric difference between the Vorob'ev expectation and the nine conditional realizations of $\mathbf{\Gamma}$ shown on Figure 4.2.

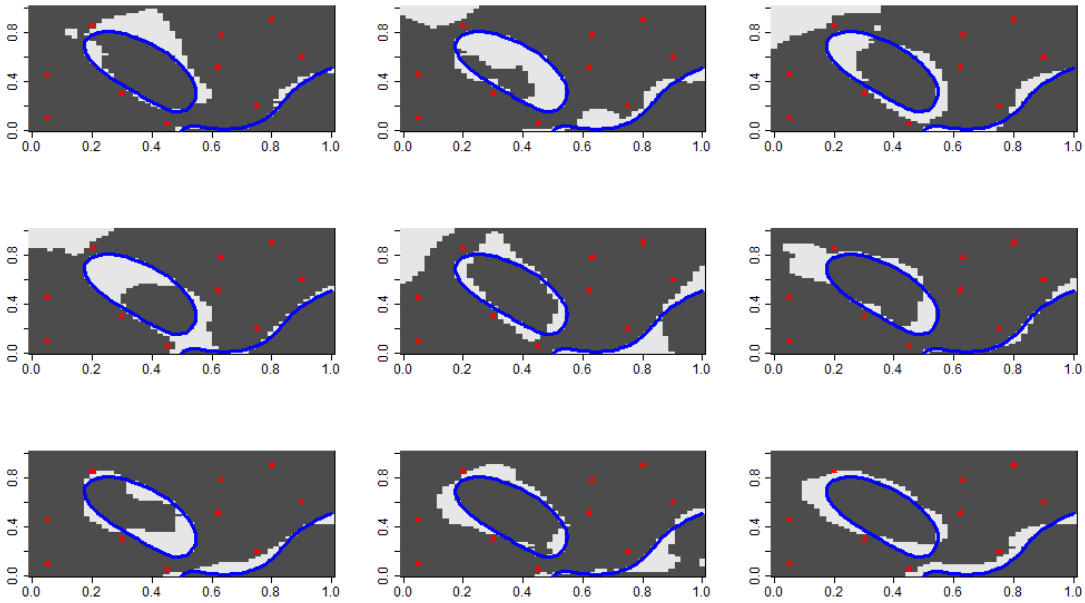

Figure 4.4: Symmetrical difference $\mathbf{\Gamma} \Delta Q_{n,T_n}$ for nine conditional realizations of $\mathbf{\Gamma}$.

In order to use in applications a SUR strategy based on the uncertainty $Var_n(\mathbf{\Gamma})$, one needs to be able to efficiently compute the expectation of the future uncertainty if a batch of q points, $\mathbf{x}^{(q)}$ is added:

$$J_n(\mathbf{x}^{(q)}) := \mathbb{E}_n \left(Var_{n+q}(\mathbf{\Gamma}) \right). \tag{4.34}$$

Now, Equation (4.33) can be rewritten:

$$Var_n(\mathbf{\Gamma}) = \int_{\mathbb{X}} \left(p_n(\mathbf{x}) \mathbb{1}_{(p_n(\mathbf{x}) < T_n)} + (1 - p_n(\mathbf{x})) \mathbb{1}_{(p_n(\mathbf{x}) \geq T_n)} \right) \mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{x}) \tag{4.35}$$

so that:

$$J_n(\mathbf{x}^{(q)}) = \mathbb{E}_n\left(\int_{\mathbb{X}} \left(p_{n+q}(\mathbf{x})\mathbb{1}_{(p_{n+q}(\mathbf{x})<T_{n+q})} + (1-p_{n+q}(\mathbf{x}))\mathbb{1}_{(p_{n+q}(\mathbf{x})\geq T_{n+q})}\right)\mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{x})\right) \tag{4.36}$$

Given an integration point $\mathbf{x} \in \mathbb{X}$, computing the expectation of the integrand may seem difficult, mainly because we do not control well how the Vorob'ev threshold "reacts" when new observations are assimilated. Empirically (see, Appendix D) we verified that, from one iteration to another, the Vorob'ev threshold has only small variations. Consequently, instead of trying to find closed form expressions for Equation (4.36) we will instead investigate the simpler criterion:

$$\widetilde{J}_n(\mathbf{x}^{(q)}) = \mathbb{E}_n\left(\int_{\mathbb{X}} \left(p_{n+q}(\mathbf{x})\mathbb{1}_{(p_{n+q}(\mathbf{x})<T_n)} + (1-p_{n+q}(\mathbf{x}))\mathbb{1}_{(p_{n+q}(\mathbf{x})\geq T_n)}\right)\mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{x})\right), \tag{4.37}$$

where the (random) Vorob'ev threshold T_{n+q} is fixed to T_n. Finding analytical expressions for Equation (4.37) involves the computation, for any integration points $\mathbf{x} \in \mathbb{X}$, of the following expressions:

$$\mathbb{E}_n(p_{n+q}(\mathbf{x})\mathbb{1}_{(p_{n+q}(\mathbf{x})<T_n)}), \tag{4.38}$$

$$\mathbb{E}_n(p_{n+q}(\mathbf{x})\mathbb{1}_{(p_{n+q}(\mathbf{x})\geq T_n)}), \tag{4.39}$$

$$\mathbb{E}_n(\mathbb{1}_{(p_{n+q}(\mathbf{x})\geq T_n)}). \tag{4.40}$$

Equation (4.37) is indeed the integral of (4.38) − (4.39) + (4.40).

Note that Expression (4.39) can be easily calculated from (4.38), as, using the law of total expectation we have that: (4.38) + (4.39) = $p_n(\mathbf{x})$. We will thus focus in the first place on the calculation of Expression (4.38). The calculations detailed below apply in the particular case where $D = [T, \infty)$ for some $T \in \mathbb{R}$. However, as in the previous section, they may be adapted to the case where D is a finite union of intervals. From the kriging update formulas, we have:

$$\mathbb{E}_n(p_{n+q}(\mathbf{x})\mathbb{1}_{(p_{n+q}(\mathbf{x})<T_n)}) = \int_{\mathbb{R}^q} \Phi(a(\mathbf{x}) + \mathbf{b}(\mathbf{x})^\top \mathbf{u})\mathbb{1}_{(a(\mathbf{x})+\mathbf{b}(\mathbf{x})^\top\mathbf{u}<\Phi^{-1}(T_n))}\Psi(\mathrm{d}\mathbf{u})$$

where $a(\mathbf{x})$, $\mathbf{b}(\mathbf{x})$, $\Psi(\cdot)$ and $\boldsymbol{\Sigma}$ are defined as in Section 4.1, Equations (4.14) and (4.15).

Then, by defining again $N_1 \sim \mathcal{N}(0,1)$ independent from the random centered response at the batch $\mathbf{x}^{(q)}$, $\mathbf{U} \sim \mathcal{N}_q(\mathbf{0}, \boldsymbol{\Sigma})$, we have:

$$\begin{aligned}\mathbb{E}_n(p_{n+q}(\mathbf{x})\mathbb{1}_{(p_{n+q}(\mathbf{x})<T_n)}) &= \int_{\mathbb{R}^q} P(N_1 < a(\mathbf{x}) + \mathbf{b}(\mathbf{x})^\top \mathbf{u})\mathbb{1}_{(a(\mathbf{x})+\mathbf{b}(\mathbf{x})^\top \mathbf{u}<\Phi^{-1}(T_n))}\Psi(\mathrm{d}\mathbf{u}) \\ &= \mathbb{E}\left(P(N_1 < a(\mathbf{x}) + \mathbf{b}(\mathbf{x})^\top \mathbf{U}, a(\mathbf{x}) + \mathbf{b}(\mathbf{x})^\top \mathbf{U} < \Phi^{-1}(T_n)\right)\end{aligned}$$

where the last expectation refers to the randomness of $\mathbf{U}$. At this stage, we see that we can get rid of the expectation to obtain:

$$\mathbb{E}_n(p_{n+q}(\mathbf{x})\mathbb{1}_{(p_{n+q}(\mathbf{x})<T_n)}) = \Phi_2\left(\begin{pmatrix} a(\mathbf{x}) \\ \Phi^{-1}(T_n) - a(\mathbf{x}) \end{pmatrix}, \begin{pmatrix} c(\mathbf{x}) & 1-c(\mathbf{x}) \\ 1-c(\mathbf{x}) & c(\mathbf{x})-1 \end{pmatrix}\right) \tag{4.41}$$

The last term, (4.40), can be calculated more easily. We obtain:

$$\mathbb{E}_n(\mathbb{1}_{(p_{n+q}(\mathbf{x})\geq T_n)}) = \Phi\left(\frac{a(\mathbf{x}) - \Phi^{-1}(T_n)}{\sqrt{c(\mathbf{x})-1}}\right) \tag{4.42}$$

so that our final result, written in one formula is:

$$\begin{aligned}\widetilde{J}_n(\mathbf{x}^{(q)}) = \int_{\mathbb{X}} \Bigg(2\Phi_2\left(\begin{pmatrix} a(\mathbf{x}) \\ \Phi^{-1}(T_n) - a(\mathbf{x}) \end{pmatrix}, \begin{pmatrix} c(\mathbf{x}) & 1-c(\mathbf{x}) \\ 1-c(\mathbf{x}) & c(\mathbf{x})-1 \end{pmatrix}\right) \\ - p_n(\mathbf{x}) + \Phi\left(\frac{a(\mathbf{x}) - \Phi^{-1}(T_n)}{\sqrt{c(\mathbf{x})-1}}\right)\Bigg) \mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{x}).\end{aligned} \tag{4.43}$$

The sampling criterion defined by Equation (4.43) has been implemented in the KrigInv R package (see, Chevalier et al. [2013d], Section 4.3.1 and Appendix E). Like the other criteria presented in Section 4.1, its computation simply requires computations of updated kriging variances and covariances as well as efficient algorithms for the function Φ_2 [Azzalini, 2012; Kenkel, 2011]. The integral over $\mathbb{X}$ is computed numerically, through a Monte Carlo sampling, and a relevant instrumental distribution may be selected in order to reduce the Monte-Carlo error (see, Appendix E, and next Section for more details).

Keeping our example with the Branin function, an example of use of this $\widetilde{J}_n$ crite-

rion, dedicated to the reduction of the Vorob'ev deviation $Var_n(\mathbf{\Gamma})$, is presented on Figure 4.5. To compare the behaviour of the criterion with the criteria presented in the last section, we also run the batch-sequential inversion on the same function, but using the criterion called "sur" in KrigInv, which is the $\widetilde{J}_n$ criterion of Section 4.1. On this example, the batches chosen by the two criteria seem to be rather similar.

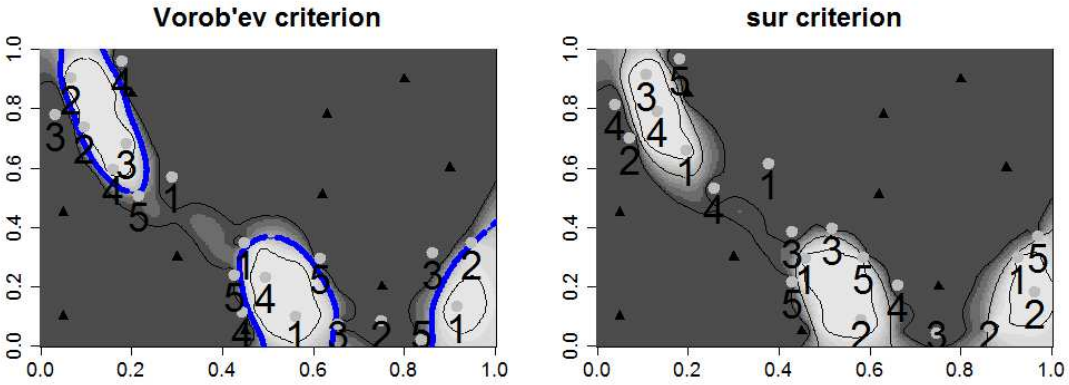

Figure 4.5: Left: Excursion probability and Vorob'ev Expectation after five iterations (with $q = 4$) of the batch-sequential criterion which aims at reducing the Vorob'ev deviation. Right: Five iterations of the $\widetilde{J}_n$ criterion presented in Section 4.1. In the KrigInv package, the latter criterion is called "sur". Grey circles are the newly evaluated locations. Numbers correspond to iteration number where these locations were chosen.

4.3 R programming: KrigInv package and auxiliary problems

4.3.1 KrigInv

The sampling criteria presented in this Chapter have all been implemented in the KrigInv R package [Chevalier et al., 2012, 2013d]. The package enables the use of sequential and batch-sequential sampling strategies for inversion. A tutorial is available in Appendix E.

Without repeating the content of the article given in Appendix E we would like to briefly summarize what can be done with the criteria available in KrigInv. First, three simple sampling criteria available in the literature (see, Bichon et al. [2008]; Picheny et al. [2010]; Ranjan et al. [2008]) are referred to as *pointwise* criteria. These criteria are dedicated to inversion problems in the particular case where D is of the form $D = [T, \infty)$. Pointwise criteria are easy and fast to compute. However, the criteria are not associated with an uncertainty measure and are not optimal. In addition, these three criteria cannot easily - to be best of our knowledge - be adapted to batch-sequential settings.

In addition to pointwise criteria, a major contribution in KrigInv is the coding of SUR criteria, including the criteria presented in this Chapter. An additional SUR criterion called *target Integrated Mean Square Error* (tIMSE) [Picheny, 2009; Picheny et al., 2010] is implemented. This criterion is dedicated to the estimation of a contour line, but may as well be used to find the set $\mathbf{\Gamma}^\star := \{\mathbf{x} : f(\mathbf{x}) \in D\}$. The tIMSE criterion can be used only in the particular case where $D = [T, \infty)$, while the three criteria studied in this Chapter may be used when D has a more general form. However, as of today, only the case $D = [T, \infty)$ is implemented. Extending the implementation when D is a finite union of interval is currently a work in progress. The J_n and $\widetilde{J}_n$ criterion of Section 4.1 are respectively called "jn" and "sur". The $\widetilde{J}_n$ criterion of Section 4.2 is called "vorob".

In the Appendix E, more details and examples are given on the sampling criteria. Moreover, auxiliary problems are discussed, like the choice of the optimization method to find the point (or batch) that maximizes or minimizes a given criterion. In the next subsection, we provide some supplementary work (which is not in the article) for dealing with the integrals (over $\mathbb{X}$ or $\mathbb{X} \times \mathbb{X}$) present in the SUR criteria.

4.3.2 Numerical integration, toward the Sequential Monte-Carlo sampler

The computation of the "timse", "sur", "jn" or "vorob" criteria involves numerical integration (see, Equations (4.14), (4.15), (4.43)). The integration domain is $\mathbb{X}$ for

"timse", "sur", "vorob" and $\mathbb{X} \times \mathbb{X}$ for "jn". In applications, these integrals are usually computed using Monte-Carlo integration.

A standard method to compute these integrals is to use and i.i.d. sample of integration points, generated uniformly on the integration domain. It is however known (see, Robert and Casella [2004] for details) that *importance sampling* techniques may reduce the Monte-Carlo error, through the choice of an adapted *instrumental distribution*. The optimal instrumental distribution (leading to a Monte-Carlo error of zero) is exhibited in Robert and Casella [2004] but its use requires knowledge of the integral at hand, and is thus pointless.

We propose in the Appendix E some instrumental densities adapted to the integrals at hand (i.e. hopefully "close" to the optimal distribution), and give a procedure to generate a sample from these densities. The idea is the same for each SUR criteria, and consists in saying that the integrand present in a SUR criterion may not depart much from the integrand of the corresponding uncertainty measure. This suggests the use of the following instrumental densities $h(\cdot)$:

- "sur" criterion (see, Equation (4.13)): $h(\mathbf{x}) \propto p_n(\mathbf{x})(1 - p_n(\mathbf{x}))\mathrm{d}\mathbb{P}_{\mathbb{X}}(\mathbf{x})$
- "jn" criterion (see, Equation (4.12)): $h(\mathbf{z}_1, \mathbf{z}_2) \propto p_n(\mathbf{z}_1)p_n(\mathbf{z}_2)\mathrm{d}\mathbb{P}_{\mathbb{X}}(\mathbf{z}_1)\mathrm{d}\mathbb{P}_{\mathbb{X}}(\mathbf{z}_2)$
- "vorob" criterion (see, Equation (4.37)): $h(\mathbf{x}) \propto (p_n(\mathbf{x})\mathbb{1}_{(p_n(\mathbf{x})<T_n)} + (1 - p_n(\mathbf{x}))\mathbb{1}_{(p_n(\mathbf{x})\geq T_n)})\mathrm{d}\mathbb{P}_{\mathbb{X}}(\mathbf{x})$

In KrigInv, samples from these instrumental distribution can be generated to compute the SUR criterion. The samples are renewed at each iteration. Generating these Monte-Carlo samples is not an easy task. Indeed, let us consider the example of the "sur" criterion, and, for simplicity, let us assume that $\mathbb{P}_{\mathbb{X}}$ is the uniform measure on $\mathbb{X} = [0, 1]^2$. One can see (Figure 4.6) that, as the inversion progresses, the region where $p_n(\mathbf{x})(1 - p_n(\mathbf{x}))$ is not zero does not have a simple shape and has the tendency to become more and more narrow. The issues described here exclude a straightforward use of standard MCMC methods, like the ones based on the Metropolis-Hastings algorithm (see, Robert and Casella [2004] for a review of the methods used in Markov Chain Monte-Carlo).

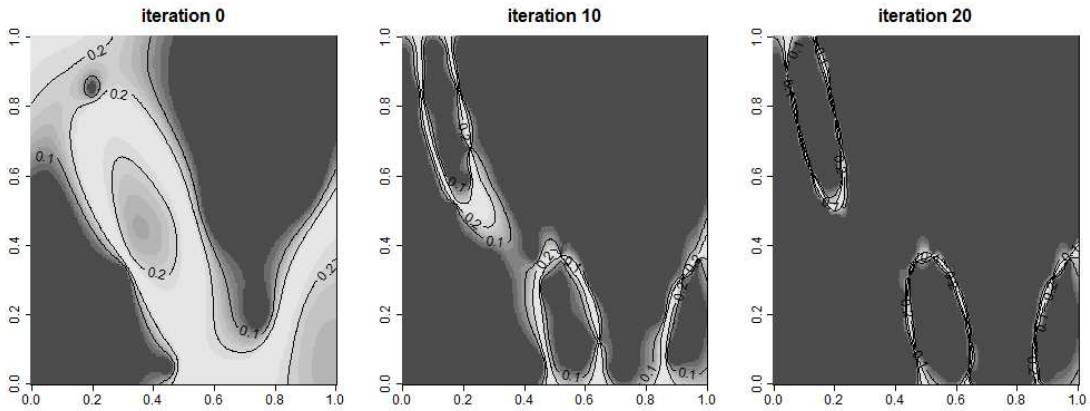

Figure 4.6: Function $p_n(\mathbf{x})(1-p_n(\mathbf{x}))$ at different iterations of the inversion.

The current method used in KrigInv to generate an (i.i.d) sample from the instrumental distribution is detailed in Appendix E. Keeping the example of the "sur" criterion, the method consists in sampling from a simpler *discrete* distribution proportional to:

$$\sum_{j=1}^{N} p_n(\mathbf{u}_j)(1-p_n(\mathbf{u}_j))\delta_{\mathbf{u}_j}, \tag{4.44}$$

where N is a large number and $\mathbf{u}_1, \ldots, \mathbf{u}_N$ is an i.i.d. sample of points with distribution $\mathbb{P}_{\mathbb{X}}$. Sampling from the distribution above requires to calculate $p_n(\mathbf{u}_j)$ for all $\mathbf{u}_j$'s and the normalizing constant $c := \sum_j p_n(\mathbf{u}_j)(1-p_n(\mathbf{u}_j))$. Then the location $\mathbf{u}_j$ is selected with probability $c^{-1}p_n(\mathbf{u}_j)(1-p_n(\mathbf{u}_j))$. This method has the advantage to be simple and easy to implement. However, it has many drawbacks. First, both N and the number of points sampled from this distribution need to tend to infinity to ensure the convergence of the Monte-Carlo estimator to the real integral. Second, the method is rather computer-intensive as N is large, and does not re-use the sample at time n to build a sample at time $n+q$. Despite these important drawbacks, it is shown that, in application, the use of these instrumental distributions significantly reduces the Monte-Carlo error.

We now would like to detail a work in progress in this field, which is not detailed in Appendix E. In our settings, we would like to take advantage of the sequentiality of

the problem. When a Monte-Carlo sample is built at time n, it would be interesting to be able re-use this sample to construct a new one at time $n+q$. The argument here is that the distributions to sample are "close" from one iteration to another. Indeed, with the example of the "sur" criterion, the quantity $p_{n+q}(\mathbf{x})(1-p_{n+q}(\mathbf{x}))$ may not depart much from $p_n(\mathbf{x})(1-p_n(\mathbf{x}))$.

A recent article of Del Moral et al. [2006] gives methods and algorithms adapted to our problem. We recently implemented algorithms derived from this article. The Sequential Monte-Carlo samplers aim at sampling sequentially from a sequence of probability measures $\pi_1, \pi_2, \ldots, \pi_k$, known up to a multiplicative constant and defined over the same measurable space. Here, $\pi_1 \propto p_{n_0}(1-p_{n_0})$ for some $n_0 \in \mathbb{N}$, $\pi_2 \propto p_{n_0+q}(1-p_{n_0+q})$ and so on. If one wants to construct, at each iteration i, a sample of size s with distribution π_i, an application of the algorithms of Del Moral et al. [2006] to our particular settings leads to the following algorithm:

1. First Monte-Carlo sample: sample from a distribution π_1 proportional to $p_{n_0}(1-p_{n_0})$. The obtained sample is denoted by $\mathbf{x}^1 \in \mathbb{X}^s$. Set, $\mathbf{x}^i = \mathbf{x}^1$ and $n = n_0$.

2. Evaluate the SUR criterion with this sample. Find the batch of q points optimizing the criterion and evaluate the target function f at this batch.

3. For i from 2 to k, perform the following steps:

   - When the q new evaluations are available, re-sample in the previous Monte-Carlo sample $\mathbf{x}^{i-1}$ using the weights: $p_{n+q}(\mathbf{x}_j)(1-p_{n+q}(\mathbf{x}_j))/(p_n(\mathbf{x}_j)(1-p_n(\mathbf{x}_j)))$, $\mathbf{x}_j \in \mathbf{x}^{i-1}$. The obtained sample is denoted by $\mathbf{x}^i_{\text{resample}}$
   - Perform a single Metropolis-Hastings jump for each point of the sample $\mathbf{x}^i_{\text{resample}}$. This gives a new sample $\mathbf{x}^i$ from π_i.
   - Use this sample to compute the SUR criterion at iteration i. Find the batch of q points optimizing the criterion and evaluate the target function f at this batch. Set $n = n+q$.

The algorithm above bases on a Sequential Monte-Carlo Sampler, using at step i, a MCMC kernel with invariant distribution π_i (see, Del Moral et al. [2006],

Section 3.3.2.3). It is relevant in our settings as $\pi_{i+1} \approx \pi_i$. A sample of size 5000, distributed from π_i for different values of i, is shown on Figure 4.7. In this example, the MCMC transition kernel is a Metropolis-Hastings kernel with Gaussian jumps.

In our settings, the algorithm has important advantages. First, it enables to sample from a distribution even if its support becomes more and more narrow (see, Figure 4.6). Second, the algorithm has a lower complexity of $O(s)$, where s is the number of integration points. This algorithm is not released yet in KrigInv.

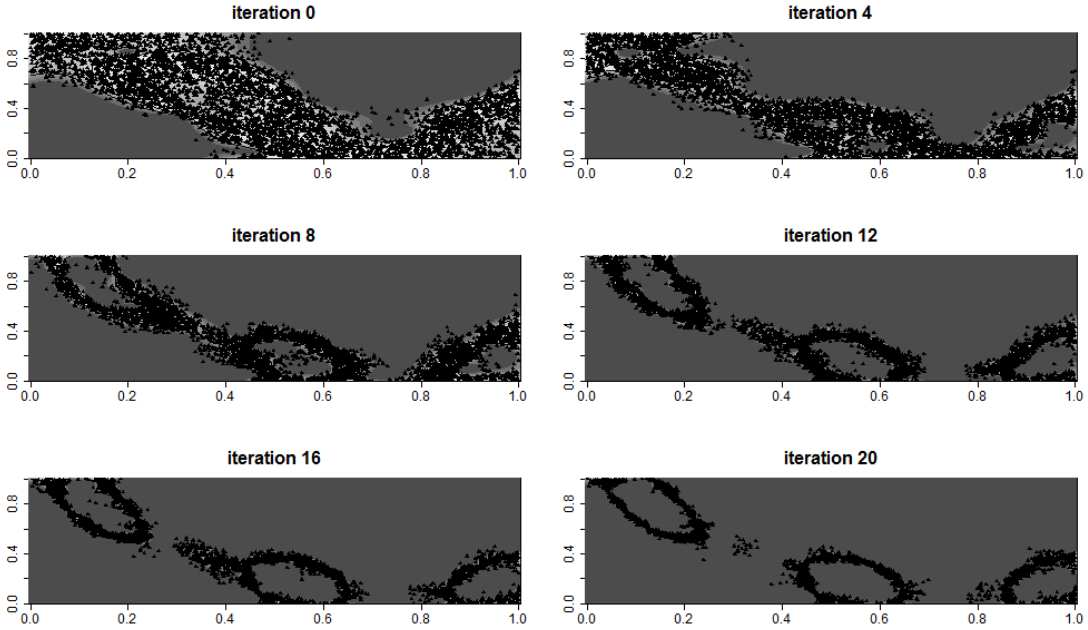

Figure 4.7: $p_n(\mathbf{x})(1 - p_n(\mathbf{x}))$ function at different iterations of the inversion. The function is plotted together with the sample of integration points used to calculate the SUR criterion (Equation (4.14)) at each iteration. This sample is obtained using a Sequential Monte-Carlo sampler.

Part III

Contribution in robust inversion

Chapter 5

Motivations in Nuclear Safety

This part of the thesis details our contribution in "robust inversion" of an expensive-to-evaluate function. The problem will be properly defined mathematically in Section 5.3. In this Chapter, we first present real-life test cases motivating the use of SUR strategies for inversion and robust inversion. The test cases are provided by the laboratory of criticality research of the French Institute of Nuclear Safety (IRSN, Institut de Radioprotection et de Sûreté Nucléaire).

5.1 Introduction to Nuclear Criticality Safety

Fission reactions in nuclear plants are based on neutrons which are both a product and an initiator of the fission reaction. The kernel of an heavy atom (of, e.g., Uranium, Plutonium) hit by neutrons may indeed fission into lighter kernels and free other neutrons which may, again, hit other heavy kernels. The fission produces a high quantity of energy and thus needs to be controlled. In particular, one wants to avoid the overproduction of neutrons.

The criticality safety of a system is evaluated through the neutron multiplication factor (called k-effective or k_{eff}), which models the nuclear chain reaction trend. While a $k_{\text{eff}} > 1$ implies an increasing neutron production leading to an uncontrolled chain reaction, a $k_{\text{eff}} < 1$ is the safety state required for fuel storage. The neutron multiplication factor depends on many parameters such as the composition of fissile materials, operation conditions, geometry, etc. For a given set of

physical parameters, the value of k_{eff} can be evaluated using a simulator, often based on Markov Chain Monte-Carlo (MCMC) simulation techniques, to solve the underlying Boltzman equation which describes neutronic behaviour of the system. An example of code to evaluate the k_{eff} is the MORET Monte Carlo code [Fernex et al., 2005].

The k_{eff} can thus be seen as a multivariate function of the parameters of the system, $\mathbf{x} \in \mathbb{X}$, (mass of fissile material, geometry, ...). Each call to MORET delivers a noisy evaluation of k_{eff} and a variance obtained from the Monte-Carlo estimation. A typical task to be performed by safety assessors is hence to find the worst combination of input parameters of the criticality code (i.e. leading to maximum reactivity) over the whole operating range. For instance, checking sub-criticality can be done by solving a maximization problem where the objective function (possibly, observed with a noise) is the k_{eff} returned by the simulator. This straightforward view of criticality parametric calculations complies with the framework of Design of Computer Experiments. It may provide a support to enhance and consolidate good practices in safety assessment. Indeed, supplementing the standard "expert driven" assessment by a suitable algorithm may be helpful to increase the reliability of the whole process, and the robustness of its conclusions.

In application, safety is assessed by checking if k_{eff} does not reach an upper safety limit (USL) of 0.95[1]. To accept this demonstration, the regulation requires safety assessors to define some parameters (such as the fissile mass) to be strictly controlled. In return, all the other parameters (say "non-controlled", for example the amount of water in the designed equipment) are assumed to take any values within a given credible range [IRSN, 2010].

Evaluating $k_{eff}(\mathbf{x})$ for one given configuration $\mathbf{x}$ is an expensive operation which takes 5 to 30 minutes, depending on the numerical conditioning of the model. Given the time constraints in the release of safety reports, it is thus important to have a sound allocation strategy. The IRSN has at its disposal a computing infrastructure allowing to evaluate the $k_{eff}(\cdot)$ function in parallel, at different "locations", i.e. for different sets of input parameters. This feature encourages the

[1] In fact, true criticality is reached when $k_{eff} > 1$, but, considering the consequences of a critical excursion, a margin from true criticality threshold is deducted to define the USL.

use of batch-sequential strategies for solving problems on the k_{eff} function (optimization, inversion, etc...). An example of use of the Efficient Global Optimization (EGO) algorithm of Jones et al. [1998] to find the most critical parameters is given in Richet et al. [2013].

At IRSN, the MORET code can be called sequentially or in batches using a software called Prométhée[1], which has the advantage of allowing the coupling of sequential evaluation strategies coded in R. The new algorithms of, e.g., the KrigInv package can thus be used directly on nuclear safety real cases, which allows smooth interactions with the physicists. The work presented in this thesis may be used to propose different model-based solutions to different general problems on the $k_{\text{eff}}(\cdot)$ function. Three problems studied in the thesis are listed below.

- Find the configurations $\mathbf{x}$ leading to the highest (here, assumed noise-free) response $k_{\text{eff}}(\mathbf{x})$: optimization problem. This can be done for instance using a batch-sequential EGO algorithm with the multi-points Expected Improvement criterion presented in Section 2.3 and Appendix B.

- Find the set $\mathbf{\Gamma}^\star$ of "unwanted" configurations, where $k_{\text{eff}}(\mathbf{x})$ is greater than $T = 0.95$: inversion problem. This can be achieved using batch-sequential SUR strategies for inversion presented in Chapter 4, and is also illustrated in next Section.

- Find the set of safe-controlled configurations, as defined in Section 5.3 using a sequential or batch-sequential SUR strategy for robust inversion. This part is detailed in Chapter 6 and Appendix F.

It is important to emphasize that, in nuclear safety, kriging-based evaluation strategies are used only as a complement to the expertise of the physicists, and can by no means replace it. In nuclear safety studies, the experts calculate - from the physics - the possible solutions to their problem at hand and may, as a complement, use the evaluation strategies presented in this thesis to validate their results.

[1] see, Prométhée project: A grid computing environment dedicated to design of computer experiments, http://promethee.irsn.org

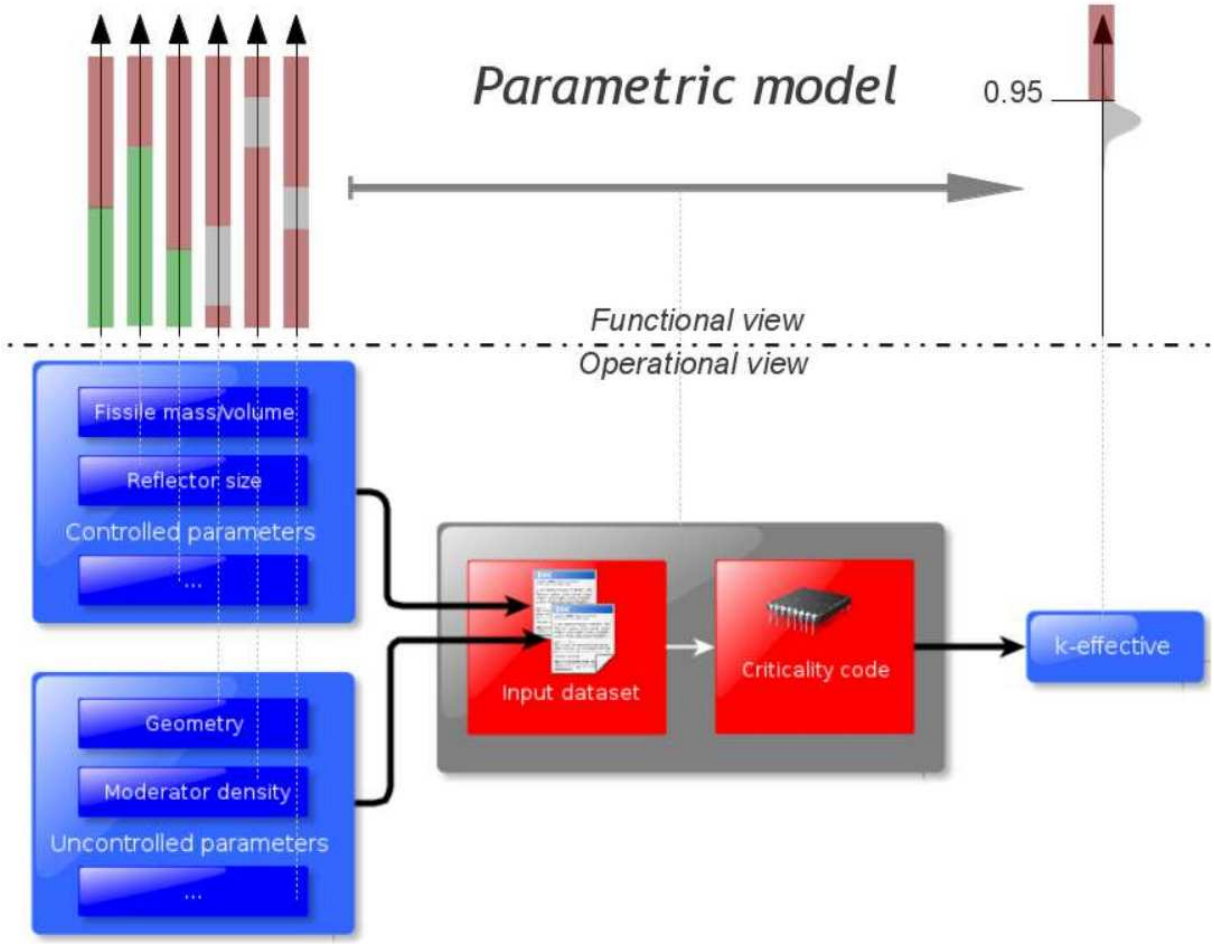

Figure 5.1: Operational/functional views of criticality parametric calculation.

5.2 Nuclear Safety and inversion

In the present application case, defined by the IRSN, the system is a storage facility of plutonium powder, whose criticality (k_{eff}) is controlled by two real-valued input parameters:

- the mass of plutonium (MassePu) in the system
- the logarithm of the concentration of plutonium (logConcPu).

The input domain $\mathbb{X}$ is (MassePu,logConcPu)$\in \mathbb{X} = [0.1, 2] \times [2, 9.6]$ and our aim is to locate the excursion set: $\mathbf{\Gamma}^\star = \{\mathbf{x} \in \mathbb{X} : f(\mathbf{x}) \geq T\}$, where $T = 0.95$ and $f(\mathbf{x}) := k_{eff}(\text{MassePu}, \text{logConcPu})$.

As explained previously we do not directly observe $f(\mathbf{x})$ itself but rather $f(\mathbf{x}) + \varepsilon$, where ε is a noise with a standard deviation (here: 0.001) estimated by the MCMC simulator. In addition the noises ε at different locations are assumed to be independents. The noise is taken into account in our sampling strategy as the SUR strategies for inversion presented in this thesis are applicable in these settings.

The unknown excursion set is represented in Figure 5.2. Such estimate is constructed for validation purposes, relying on 300 evaluations of the code on a space-filling design of experiment, and by thresholding the kriging mean at $T = 0.95$. Our aim is to determine whether SUR strategies can identify accurately the excursion set with a small fraction of this budget. Note that the present test case is also presented in Appendix C.
We start with an initial design of 10 evaluations, obtained with the "maximiLHS" function of the "LHS" R package. Then, 5 iterations of the $\widetilde{J}_n$ criterion of Section 4.1 (i.e. the "sur" criterion in KrigInv) are run, with batches of 4 points at each iterations. The parameters of the algorithm are summarized below:

- Trend functions for the kriging model: only a constant (ordinary kriging).
- Covariance kernel: Matérn covariance with parameter $\nu = 3/2$.
- Estimation of the covariance parameters: the covariance parameters (variance and ranges) are re-estimated at each iteration by Maximum Likelihood, using the DiceKriging package.

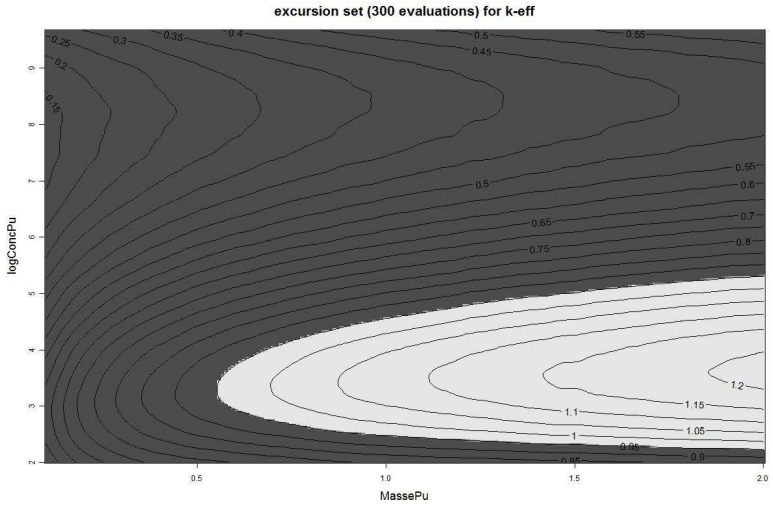

Figure 5.2: Contour lines of the function $k_{eff} = f(\text{MassePu}, \text{logConcPu})$ with the excursion set (in white) of k_{eff} corresponding to a threshold $T = 0.95$. This approximations is obtained by computing the kriging mean function from 300 evaluations on a space-filling design of experiments.

- Initial design of experiments: 10 points (maximin LHS in dimension $d = 2$).
- Number of points evaluated per iteration: $q = 4$.
- Total number of iterations: 5.
- SUR criterion minimized at each iteration: $\widetilde{J}_n$ function of Equations (4.13),(4.15).
- Criterion optimization: with the genoud R package [Mebane and Sekhon, 2011] and with the heuristic strategy detailed in Appendix E, Section 4.1.2.
- Number of integration points for the integral over $\mathbb{X}$: 600 points renewed at each iteration.
- Choice of the integration points: sampling from the "sur" instrumental density described in Appendix E, Sections 4.2.1 and 4.2.2.
- Parameters of the genoud algorithm: pop.size = 200, max.generation = 20.

Figures 5.3 and 5.4 show the evolution of the algorithm. The newly evaluated points are represented together with the decrease of the uncertainty $\widetilde{H}_n$ defined in Equation (4.4).

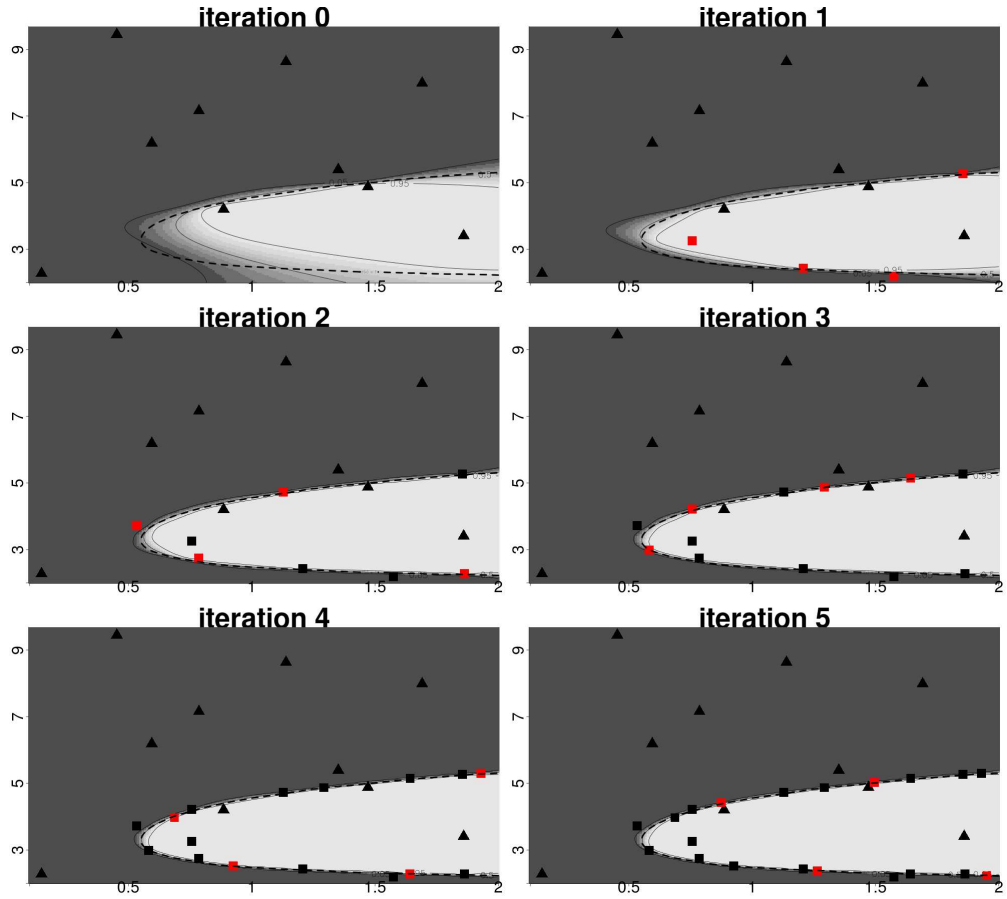

Figure 5.3: Plot of the function $p_n(\mathbf{x}) = P(\mathbf{x} \in \mathbf{\Gamma} | \mathcal{A}_n)$ after n evaluations of the simulator. The triangles are the ten points of the initial design. The squares are the points sampled using the $\widetilde{J}_n$ criterion. Areas in black correspond to $p_n(\mathbf{x}) \approx 0$ and areas in white correspond to $p_n(\mathbf{x}) \approx 1$. The dotted line indicates the true excursion set.

The SUR strategy on this rather simple smooth function shows good performances as the true excursion set is quickly identified. More numerical tests are performed in Appendix C on more difficult functions (one function in dimension 6 and one function with a non-connected excursion set).

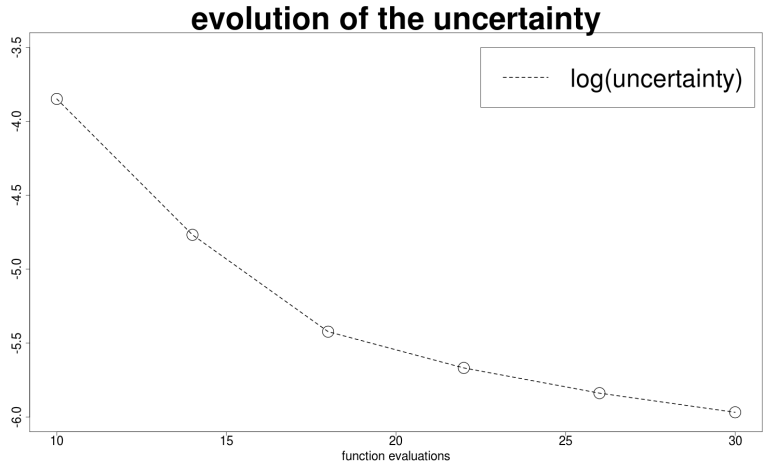

Figure 5.4: Evolution of $\widetilde{H}_n$ during the sequential sampling strategy.

5.3 Nuclear Safety and robust inversion

The so-called problem of "robust inversion" is defined in this Section, using an example in nuclear-safety. We consider again a deterministic simulator $f : \mathbb{X} \mapsto \mathbb{R}$, which response is possibly observed with some noise.

The practical motivation in nuclear safety is that, often, some parameters of the simulator can be controlled by practitioners (e.g. the mass of fissile material) while other parameters are environment variables and cannot be controlled (e.g. the quantity of water in the storage system). In nuclear safety, engineers are interested by the "safe" configurations of a system, which are the configurations where $f(\mathbf{x}) := \mathrm{k}_{\mathrm{eff}}(\mathbf{x}) \leq T$, where $T = 0.95$. However, if one assumes that some of the parameters of f are non-controlled, it is more relevant to seek the configurations of controlled parameters where the system remains safe for all the possible values of the non-controlled (environment) parameters. This problem is given the name of "robust inversion".

Mathematically, we consider that the input domain $\mathbb{X}$ for the d parameters can be written $\mathbb{X} = \mathbb{X}_c \times \mathbb{X}_{nc}$, where $\mathbb{X}_c$ and $\mathbb{X}_{nc}$ are respectively the input domains for the controlled and non-controlled parameters. A set of parameters $\mathbf{x} \in \mathbb{X}$ is also written $\mathbf{x} = (\mathbf{x}_c, \mathbf{x}_{nc}) \in \mathbb{X}_c \times \mathbb{X}_{nc}$. The goal in robust inversion is to identify the

set:

$$\mathbf{\Gamma}_c^\star := \{\mathbf{x}_c \in \mathbb{X}_c : \forall \mathbf{x}_{nc} \in \mathbb{X}_{nc}, f(\mathbf{x}_c, \mathbf{x}_{nc}) \leq T\} \tag{5.1}$$

Note that $\mathbf{\Gamma}_c^\star$ is a subset of $\mathbb{X}_c$ (and not of $\mathbb{X}$). Also, the total number d of parameters is written as the sum of the numbers of controlled on non-controlled parameters: $d = d_c + d_{nc}$.

The test case studied in the next Chapter is fuel storage which depends of three parameters. Two parameters are controlled: the mass of fissile material in the storage and the geometry (i.e., the radius of the container). One parameter is not controlled: the concentration of fissile material. Both mass and geometry are parameters used to limit the k_{eff}, while the fissile concentration is not known nor measured. It is just guaranteed to stay inside some physical bounds.

Figure 5.5: Mass-Geometry criticality system, depending on three parameters: radius, mass and concentration of fissile mass (in orange), inside a cylindrical container.

The domain $\mathbb{X}$ is an hyper-rectangle in dimension 3, and, instead of using directly the MORET code to evaluate the function, a good fast approximation of it is provided in R. Figure 5.6 gives the set $\Gamma_c^\star$ obtained from the calculation of the k_{eff} on a $100 \times 100 \times 100$ grid. Obviously, in our application, we do not want to perform millions of evaluations to identify $\Gamma_c^\star$. A realistic budget is more of the order of a few hundreds of evaluations.

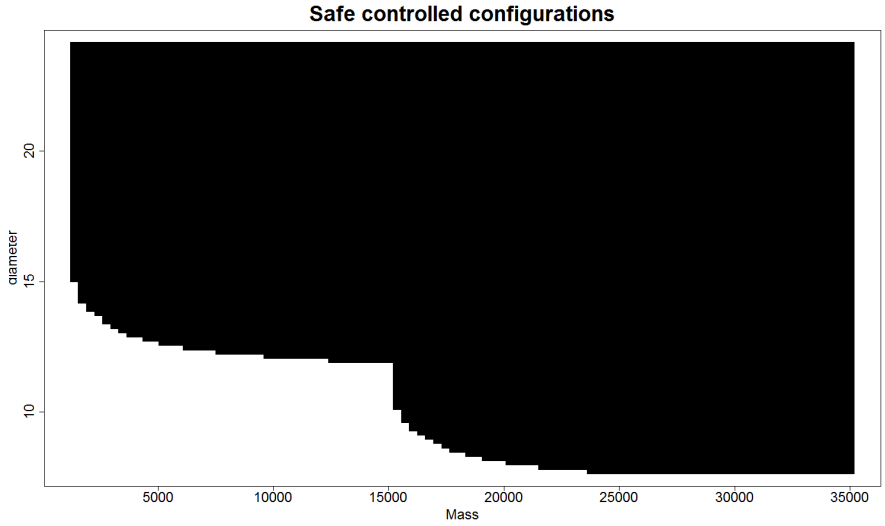

Figure 5.6: Set $\Gamma_c^\star$ (in white) obtained from evaluations of the k_{eff} on a grid.

The next chapter presents new SUR strategy which aim at solving such robust-inversion problem. As we will see, the work performed in inversion will be useful (and serve as a baseline) to work on this problem.

Chapter 6

SUR criterion for robust inversion

In this Chapter we deal with SUR strategies aiming to solve the problem of robust inversion defined in Section 5.3. The settings and notations introduced in Section 5.3 are used throughout all the Chapter.

6.1 Optimal SUR criteria

In this section, we introduce optimal one-step-lookahead SUR sampling criteria for the problem of robust inversion, i.e. the problem of identifying the set:

$$\mathbf{\Gamma}_c^\star := \{\mathbf{x}_c \in \mathbb{X}_c : \forall \mathbf{x}_{nc} \in \mathbb{X}_{nc}, f(\mathbf{x}_c, \mathbf{x}_{nc}) \leq T\}, \tag{6.1}$$

where $f : \mathbb{X} \mapsto \mathbb{R}$ is a function which is possibly observed with some noise (the variance of which is assumed to be known), and T is a fixed threshold[1]. We recall that the d scalar inputs of f can be separated into d_c controlled parameters and d_{nc} non-controlled parameters, so that the input space $\mathbb{X}$ can be written $\mathbb{X} = \mathbb{X}_c \times \mathbb{X}_{nc}$. Following the methodology developed in Chapter 4, we introduce the random set:

$$\mathbf{\Gamma}_c := \{\mathbf{x}_c \in \mathbb{X}_c : \forall \mathbf{x}_{nc} \in \mathbb{X}_{nc}, \xi(\mathbf{x}_c, \mathbf{x}_{nc}) \leq T\}, \tag{6.2}$$

where ξ is a random process which is assumed to be Gaussian.

[1] though not detailed in this Chapter, all the methods presented in this Chapter can be extended to the case where $f(\mathbf{x}_c, \mathbf{x}_{nc}) \in D$ where D is a closed finite union of intervals

When n observations of f, $\mathcal{A}_n$, are available, conditional realizations of $\mathbf{\Gamma}_c$ can be simulated from conditional realizations of ξ. We introduce the random variable $\alpha_c := \mathbb{P}_{\mathbb{X}_c}(\mathbf{\Gamma}_c)$ where $\mathbb{P}_{\mathbb{X}_c}$ is a given σ-finite measure on $\mathbb{X}_c$. Following the work of Bect et al. [2012] in inversion, a natural choice of uncertainty measure is:

$$H_n(\mathcal{A}_n) := Var_n(\alpha_c), \tag{6.3}$$

and, through the Cauchy-Schwartz inequality, the uncertainty defined in Equation (6.3) can be bounded as follows:

$$Var_n(\alpha_c) \leq \mathbb{P}_{\mathbb{X}_c}(\mathbb{X}_c)\widetilde{H}_n(\mathcal{A}_n), \tag{6.4}$$

where:

$$\widetilde{H}_n(\mathcal{A}_n) := \int_{\mathbb{X}_c} \widetilde{p_n}(1-\widetilde{p_n})\mathrm{d}\mathbb{P}_{\mathbb{X}_c} \tag{6.5}$$

and $\widetilde{p_n}(\mathbf{x}_c) := P(\mathbf{x}_c \in \mathbf{\Gamma}_c|\mathcal{A}_n)$ is the coverage probability function. The use of the uncertainty measure defined in Equation (6.5) is convenient as decreasing $\widetilde{H}_n(\mathcal{A}_n)$ ensures that all the domain $\mathbb{X}_c$ can be classified correctly. More precisely, if $\widetilde{H}_n(\mathcal{A}_n)$ decreases to zero as n tends to infinity, it means that, for ε arbitrary small, we have:

$$\mathbb{P}_{\mathbb{X}_c}\left(\{\mathbf{x}_c \in \mathbb{X}_c : \widetilde{p_n}(\mathbf{x}_c) > \varepsilon \text{ and } \widetilde{p_n}(\mathbf{x}_c) < 1-\varepsilon\}\right) \xrightarrow[n\to\infty]{} 0 \tag{6.6}$$

The condition given by Equation (6.6) does not necessarily hold if $H_n(\mathcal{A}_n)$ (and not $\widetilde{H}_n(\mathcal{A}_n)$) goes to zero, which justifies the use of $\widetilde{H}_n(\mathcal{A}_n)$ to quantify uncertainties on the true excursion set $\mathbf{\Gamma}_c^*$.

The coverage probability function, $\widetilde{p_n}(\cdot)$, can be rewritten as follows:

$$\widetilde{p_n}(\mathbf{x}_c) = P\left(\forall \mathbf{x}_{nc} \in \mathbb{X}_{nc},\ \xi(\mathbf{x}_c, \mathbf{x}_{nc}) \leq T|\mathcal{A}_n\right) \tag{6.7}$$

$$= P\left(\max_{\mathbf{x}_{nc}\in\mathbb{X}_{nc}} \xi(\mathbf{x}_c, \mathbf{x}_{nc}) \leq T|\mathcal{A}_n\right), \tag{6.8}$$

and the latter expression emphasizes that the coverage probability function at a given point $\mathbf{x}_c \in \mathbb{X}_c$ is equal to the *non-exceedance* probability of the *sectional Gaussian process* $\xi_{\mathbf{x}_c}(\cdot) := \xi(\mathbf{x}_c, \cdot)$.

The uncertainty measures being defined with Equations (6.3) and (6.5), the optimal one-step-lookahead SUR criteria associated with these uncertainties are:

$$J_n(\mathbf{x}^{(q)}) := \mathbb{E}_n(Var_{n+q}(\alpha_c)) \tag{6.9}$$

$$\widetilde{J}_n(\mathbf{x}^{(q)}) := \mathbb{E}_n\left(\int_{\mathbb{X}_c} \widetilde{p_{n+q}}(1-\widetilde{p_{n+q}})\mathrm{d}\mathbb{P}_{\mathbb{X}_c}\right), \tag{6.10}$$

where $\mathbf{x}^{(q)} \in \mathbb{X}^q$ is a batch of q points and where the expectations are taken with respect to the random values of ξ at this batch.

In Chapter 4, closed-form expression allowing efficient computations of these criteria were found (see, Equations (4.14) and (4.15)). However, for this new problem of robust inversion, finding closed-form expressions appears to be more challenging. In fact, the computation of the coverage probability, $\widetilde{p_n}(\mathbf{x}_c)$, itself is a difficult question as it amounts to calculate the exceedance probability of a non-stationary Gaussian process. This can be done through Gaussian process simulations but is expected to be computationally expensive. Our work in the next sections will be to propose approximations of the function $\widetilde{p_n}(\cdot)$ which allow a practical use of the SUR criterion defined by Equation (6.10).

6.2 A first approximation of the optimal criteria

In this section, we propose a multi-points SUR sampling criterion which is constructed from the optimal one-step-lookahead criterion defined by Equation (6.10). This new criterion has the advantage to be simpler to compute and has been implemented in the Prométhée platform, at the IRSN. A work in progress on a second criterion is presented in Appendix F.

6.2.1 An approximation of the uncertainty leading to a criterion

The computation of non-exceedance probabilities of a (non-stationary) Gaussian process conditioned on some observations, like

$$\widetilde{p_n}(\mathbf{x}_c) := P\left(\max_{\mathbf{x}_{nc}\in\mathbb{X}_{nc}} \xi_{\mathbf{x}_c}(\mathbf{x}_{nc}) \leq T|\mathcal{A}_n\right), \tag{6.11}$$

for some $\mathbf{x}_c \in \mathbb{X}_c$, is considered to be difficult and numerically expensive if Monte-Carlo simulations are used.
Let $\mathbf{x}_c \in \mathbb{X}_c$. A quite natural idea to obtain an approximation of $\widetilde{p_n}(\mathbf{x}_c)$ is to observe the value of the sectional Gaussian process $\xi_{\mathbf{x}_c}(\cdot)$ on a finite number of locations, rather than on the whole set $\mathbb{X}_{nc}$.

Mathematically, if $\ell > 0$ and if $\mathbf{x}_{nc}^1(\mathbf{x}_c), \ldots, \mathbf{x}_{nc}^\ell(\mathbf{x}_c)$ are ℓ points in $\mathbb{X}_{nc}$ that depend on $\mathbf{x}_c$, we propose to approximate the coverage probability $\widetilde{p_n}(\mathbf{x}_c)$ with the following expression:

$$\widehat{p_n}(\mathbf{x}_c) := P\left(\max_{\mathbf{x}_{nc}\in\{\mathbf{x}_{nc}^1,\ldots,\mathbf{x}_{nc}^\ell\}} \xi_{\mathbf{x}_c}(\mathbf{x}_{nc}) \leq T|\mathcal{A}_n\right), \tag{6.12}$$

where the locations $\mathbf{x}_{nc}^1(\mathbf{x}_c), \ldots, \mathbf{x}_{nc}^\ell(\mathbf{x}_c)$ are denoted by $\mathbf{x}_{nc}^1, \ldots, \mathbf{x}_{nc}^\ell$ to alleviate notations. One may remark approximating the non-exceedance probability $\widetilde{p_n}$ with $\widehat{p_n}$ introduces a bias. Indeed, we have that $\widetilde{p_n}(\mathbf{x}_c) \leq \widehat{p_n}(\mathbf{x}_c)$, as, trivially,

$$\max_{\mathbf{x}_{nc}\in\mathbb{X}_{nc}} \xi_{\mathbf{x}_c}(\mathbf{x}_{nc}) \geq \max_{\mathbf{x}_{nc}\in\{\mathbf{x}_{nc}^1,\ldots,\mathbf{x}_{nc}^\ell\}} \xi_{\mathbf{x}_c}(\mathbf{x}_{nc}).$$

However, the (positive) difference between $\widehat{p_n}(\mathbf{x}_c)$ and $\widetilde{p_n}(\mathbf{x}_c)$ can be mitigated by choosing the locations $\mathbf{x}_{nc}^1, \ldots, \mathbf{x}_{nc}^\ell$ in order to minimize $\widehat{p_n}(\mathbf{x}_c)$ (i.e. to maximize the exceedance probability). For this reason, although $\widehat{p_n}(\mathbf{x}_c)$ depends on the choice of $\mathbf{x}_{nc}^1, \ldots, \mathbf{x}_{nc}^\ell$, we will stick to that notation as we will assume that $\mathbf{x}_{nc}^1, \ldots, \mathbf{x}_{nc}^\ell$ are chosen in order to minimize $\widehat{p_n}(\mathbf{x}_c)$.

When the location $\mathbf{x}_{nc}^1, \ldots, \mathbf{x}_{nc}^\ell$ are fixed, the computation of $\widehat{p_n}(\mathbf{x}_c)$ requires one call to the c.d.f. of the multivariate normal distribution in dimension ℓ, Φ_ℓ, which

can be done in R with packages like "mnormt" and "mvtnorm" [Azzalini, 2012; Genz et al., 2012]. In the sequel, ℓ will be referred to as the *discretization parameter*.
Now, writing the uncertainty as follows:

$$\widehat{H}_n(\mathcal{A}_n) := \int_{\mathbb{X}_c} \widehat{p_n}(1-\widehat{p_n})\mathrm{d}\mathbb{P}_{\mathbb{X}_c}, \tag{6.13}$$

the 1-step lookahead SUR criterion becomes:

$$\widehat{J}_n(\mathbf{x}^{(q)}) := \mathbb{E}_n\left(\int_{\mathbb{X}_c} \widehat{p_{n+q}}(1-\widehat{p_{n+q}})\mathrm{d}\mathbb{P}_{\mathbb{X}_c}\right). \tag{6.14}$$

As we will see, a closed form expression can be found for the integrand of the latter expression. As in the Chapter 4, this expression will be established using the kriging update formulas and manipulations with the c.d.f. of the multivariate normal distribution.

6.2.2 Computing the criterion

Let $\mathbf{x}_c \in \mathbb{X}_c$ and $(\mathbf{x}_{nc}^1, \ldots, \mathbf{x}_{nc}^\ell) \in \mathbb{X}_{nc}^\ell$. We recall that the locations $\mathbf{x}_{nc}^1, \ldots, \mathbf{x}_{nc}^\ell$ are chosen such that:

$$\widehat{p_n}(\mathbf{x}_c) := P\left(\max_{\mathbf{x}_{nc}\in\{\mathbf{x}_{nc}^1,\ldots,\mathbf{x}_{nc}^\ell\}} \xi_{\mathbf{x}_c}(\mathbf{x}_{nc}) \leq T | \mathcal{A}_n\right)$$

is minimized, so that the positive difference between $\widehat{p_n}(\mathbf{x}_c)$ and the true non-exceedance probability $\widetilde{p_n}(\mathbf{x}_c)$ is as small as possible.
Let $\mathbf{x}^{(q)} \in \mathbb{X}^q$ be a batch of q points. Our goal is to find closed-form expressions for:

$$(*) := \mathbb{E}_n\left(\widehat{p_{n+q}}(\mathbf{x}_c)(1-\widehat{p_{n+q}}(\mathbf{x}_c))\right), \tag{6.15}$$

in order to allow efficient computations of the SUR criterion given by Equation (6.14).

Let us fix the notations. The vector of ℓ locations $((\mathbf{x}_c, \mathbf{x}_{nc}^1), \ldots, (\mathbf{x}_c, \mathbf{x}_{nc}^\ell))^\top$ has, at time n a kriging mean denoted by $\mathbf{m}_n^{(\ell)}$ and a kriging covariance matrix denoted

by $\mathbf{\Sigma}_n^{(\ell)}$. The kriging covariance of the candidate batch $\mathbf{x}^{(q)}$ at time n is denoted by $\mathbf{\Sigma}_n^{(q)}$. For $1 \leq i \leq \ell$ we also denote by $\mathbf{B}_i$ the $q \times 1$ column vector of kriging weights of $\mathbf{x}^{(q)}$ for the prediction at point $(\mathbf{x}_c, \mathbf{x}_{nc}^i)$ and by $\mathbf{B} := (\mathbf{B}_1, \ldots, \mathbf{B}_\ell)$. The matrix $\mathbf{B}$ has q rows and ℓ columns. Finally, we denote by $\mathbf{T}_\ell$ the column vector of size ℓ with each component equal to T. Our closed-form expression is given below:

$$(*) = \widehat{p_n}(\mathbf{x}_c) - \Phi_{2\ell}\left(\begin{pmatrix} \mathbf{T}_\ell - \mathbf{m}_n^{(\ell)} \\ \mathbf{T}_\ell - \mathbf{m}_n^{(\ell)} \end{pmatrix}, \begin{pmatrix} \mathbf{\Sigma}_n^{(\ell)} & \mathbf{B}^\top \mathbf{\Sigma}_n^{(q)} \mathbf{B} \\ \mathbf{B}^\top \mathbf{\Sigma}_n^{(q)} \mathbf{B} & \mathbf{\Sigma}_n^{(\ell)} \end{pmatrix}\right), \tag{6.16}$$

where $\Phi_{2\ell}(\cdot, \mathbf{\Sigma})$ is the c.d.f. of the centered multivariate normal distribution in dimension 2ℓ, with covariance matrix $\mathbf{\Sigma}$.

proof: First, from the law of total expectation we have that:

$$\mathbb{E}_n\left(\widehat{p_{n+q}}(\mathbf{x}_c)(1 - \widehat{p_{n+q}}(\mathbf{x}_c))\right) = \widehat{p_n}(\mathbf{x}_c) - \mathbb{E}_n\left((\widehat{p_{n+q}}(\mathbf{x}_c))^2\right), \tag{6.17}$$

so that we now seek a closed-form expression for $\mathbb{E}_n\left((\widehat{p_{n+q}}(\mathbf{x}_c))^2\right)$. This will be done by exhibiting the dependence of $\widehat{p_{n+q}}(\mathbf{x}_c)$ to the random response at location $\mathbf{x}^{(q)}$. From the kriging update formulas, we have that:

$$\widehat{p_{n+q}}(\mathbf{x}_c) = P\left(\max_{\mathbf{x}_{nc} \in \{\mathbf{x}_{nc}^1, \ldots, \mathbf{x}_{nc}^\ell\}} \xi_{\mathbf{x}_c}(\mathbf{x}_{nc}) \leq T | \mathcal{A}_{n+q}\right) \tag{6.18}$$

$$= P(\mathbf{N} + \mathbf{m}_n^{(\ell)} + \mathbf{B}^\top \xi_c(\mathbf{x}^{(q)}) \leq \mathbf{T}_\ell) \tag{6.19}$$

where $\mathbf{N} \sim \mathcal{N}_\ell(\mathbf{0}, \mathbf{\Sigma}_{n+q}^{(\ell)})$, $\xi_c(\mathbf{x}^{(q)})$ is the centered response at location $\mathbf{x}^{(q)}$: $\xi_c(\mathbf{x}^{(q)}) = (\xi(\mathbf{x}_{n+1}) - m_n(\mathbf{x}_{n+1}), \ldots, \xi(\mathbf{x}_{n+q}) - m_n(\mathbf{x}_{n+q}))^\top$, and where the vector inequalities of Equation (6.19) mean component-wise inequalities.

We now proceed as in Chapter 4, but with multivariate calculations. We use the notation $\mathbf{U} := \xi_c(\mathbf{x}^{(q)})$. Let $\mathbf{N}_1, \mathbf{N}_2$ be two random vectors independent from each other, and independent from $\mathbf{U}$, with distribution $\mathcal{N}_\ell(\mathbf{0}, \mathbf{\Sigma}_{n+q}^{(\ell)})$. Using similar calculations than in Section 4.1 (see, all the calculations from Equation (4.18) to Equation (4.25)), we have that:

$$\mathbb{E}_n\left((\widehat{p_{n+q}}(\mathbf{x}_c))^2\right) = P(\mathbf{N}_1 + \mathbf{B}^\top \mathbf{U} \leq \mathbf{T}_\ell - \mathbf{m}_n^{(\ell)}, \mathbf{N}_2 + \mathbf{B}^\top \mathbf{U} \leq \mathbf{T}_\ell - \mathbf{m}_n^{(\ell)}), \tag{6.20}$$

which shows that $\mathbb{E}_n\left((\widehat{p_{n+q}}(\mathbf{x}_c))^2\right)$ can be computed with a call to the $\Phi_{2\ell}$ function. To conclude our proof, it remains to note that the vectors $\mathbf{N}_1+\mathbf{B}^\top\mathbf{U}$ and $\mathbf{N}_2+\mathbf{B}^\top\mathbf{U}$ are both centered, but not independent. To compute the covariance matrix of the vector of size 2ℓ we can perform block calculations:

$$
\begin{aligned}
Cov(\mathbf{N}_1+\mathbf{B}^\top\mathbf{U}, \mathbf{N}_1+\mathbf{B}^\top\mathbf{U}) =& Var(\mathbf{N}_1) + Var(\mathbf{B}^\top\mathbf{U}) && (6.21)\\
=& \boldsymbol{\Sigma}_{n+q}^{(\ell)} + \mathbf{B}^\top\boldsymbol{\Sigma}_n^{(q)}\mathbf{B} && (6.22)\\
=& \boldsymbol{\Sigma}_n^{(\ell)} && (6.23)\\
Cov(\mathbf{N}_1+\mathbf{B}^\top\mathbf{U}, \mathbf{N}_2+\mathbf{B}^\top\mathbf{U}) =& Var(\mathbf{B}^\top\mathbf{U}) && (6.24)\\
=& \mathbf{B}^\top\boldsymbol{\Sigma}_n^{(q)}\mathbf{B} && (6.25)\\
& && (6.26)
\end{aligned}
$$

which completes the proof. □

Note that, for a given $\mathbf{x}_c \in \mathbb{X}_c$, the term $\widehat{p_n}(\mathbf{x}_c)$ of Equation (6.16) does not depend on the candidate batch $\mathbf{x}^{(q)}$ and can thus be precomputed, or even ignored if we aim at finding an optimal batch $\mathbf{x}^{(q)}$. Following our notations we have that

$$
\widehat{p_n}(\mathbf{x}_c) = \Phi_\ell(\mathbf{T}_\ell - \mathbf{m}_n^{(\ell)}, \boldsymbol{\Sigma}_n^{(\ell)}). \tag{6.27}
$$

As in Chapter 4, the obtained closed-form expression involves the c.d.f. of the multivariate distribution, in a dimension which does not depend on q (the size of the batch). The obtained criterion is thus suitable for large batch sizes. On the other hand, the computation cost of the criterion widely depends on the choice of the discretization parameter, ℓ, which should ideally be as high as possible. Indeed, when ℓ is larger, one can expect the function $\widehat{p_n}$ to be a better approximation of $\widetilde{p_n}$. Unfortunately, in our case, computation times generally prevents the use of a discretization parameter higher than 10 as the computation cost of $\Phi_{2\ell}$ gets larger when $\ell > 10$. Section F.1 introduces a SUR criterion which mitigates this problem. But before that, we would like to test the presented SUR criterion on our robust inversion test-case in nuclear safety.

6.2.3 Application to the test-case

The test-case introduced in Section 5.3 is now studied using the SUR criterion $\widehat{J}_n$ given by Equation (6.14). We recall that the objective functions has three scalar inputs, two of which being controlled. The set of the safe controlled configuration (calculated from 1 million evaluations of k_{eff}) is represented on Figure 5.6.

Many parameters are involved in our batch-sequential sampling algorithm. As in Chapter 5, we summarize them below:

- Trend functions for the kriging model: only a constant (ordinary kriging).
- Covariance kernel: Matérn covariance with parameter $\nu = 3/2$.
- Estimation of the covariance parameters: By Maximum Likelihood. The estimation is renewed at each iteration.
- Initial design of experiments: 30 points (maximin LHS in dimension 3).
- Number of points evaluated per iteration: $q = 4$.
- Total number of iterations: 40.
- SUR criterion minimized at each iteration: $\widehat{J}_n$ (see, Equations (6.14),(6.16)).
- Optimizer for the criterion: rgenoud function of the genoud package, with the heuristic strategy detailed in Appendix E, Section 4.1.2.
- Number of integration point for the integral over $\mathbb{X}_c$: 200 points renewed at each iteration.
- Choice of the integration points: sampling from an instrumental density derived with the same method than in Appendix E, Section 4.2.1 and 4.2.2.
- Parameters of the genoud algorithm: pop.size = 100, max.generation = 5.
- Discretization parameter: $\ell = 5$.
- Optimization method to find the points $\mathbf{x}_{nc}^1(\mathbf{x}_c), \ldots, \mathbf{x}_{nc}^\ell(\mathbf{x}_c)$ for each integration point $\mathbf{x}_c \in \mathbb{X}_c$: Monte-Carlo optimization which generates uniformly, over $\mathbb{X}_c^\ell$, 100 candidate batches of ℓ points and selects the best one.

In comparison to Chapter 5, a lower budget is spent to optimize the criterion: the parameter "pop.size" of the genetic algorithm is set to 100 instead of 200. In addition, the number of integration points to compute the main integrand (over $\mathbb{X}_c$) is reduced from 600 to 200, mainly to reduce computation time. In our settings, evaluating the criterion requires 200 calls to $\Phi_{2\ell}$ (one per integration point) and, here the discretization parameter, ℓ, is chosen equal to 5. As these calls are much more expensive than the calls to Φ_2 in Chapter 5, the number of integration points and the budget for optimization is reduced.
With these settings, one iteration of the algorithm (i.e., finding the best batch of $q = 4$ points where the function is evaluated) takes approximately 20 minutes with a workstation with a 2.53 GHz CPU. This cost also tends to increase with number of observations.

Figures 6.1, and 6.2 show the evolution of the coverage probability function $\widehat{p_n}(\cdot)$ and the uncertainty $\widehat{H}_n$ during the algorithm. Note that the value of the discretization parameter, ℓ, used to do these plots does not need to be the same than the one in the SUR strategy. For these plots, we used $\ell = 20$.

As in Chapter 5, the main output of these strategies is not really an excursion set, but instead a function which gives an (hopefully, usually close to 0 or 1) excursion probability. This function can be seen as a classifier. In the present case, the excursion domain $\mathbf{\Gamma}_c^\star$ is well recovered by the algorithm. Indeed, after 40 iterations (160 evaluations) the whole domain $\mathbb{X}_c$ has an excursion probability $\widehat{p_n}$ close to either 0 or 1.

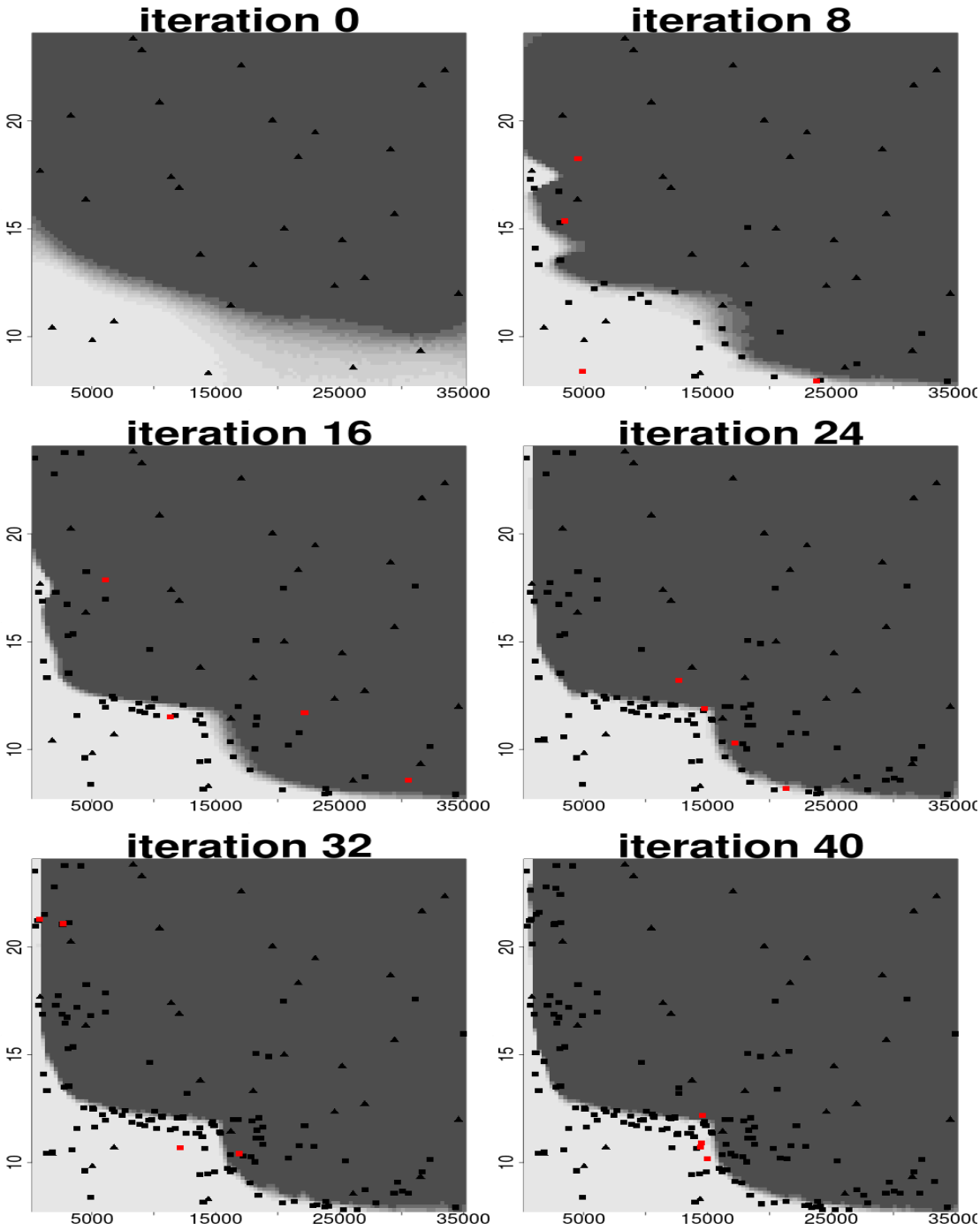

Figure 6.1: Plot of the function $\widehat{p_n}(\mathbf{x}_c)$ after n evaluations of the simulator. The triangles are the (projected) 30 points of the initial design. The squares are the points sampled using the $\widehat{J_n}$ criterion. Areas in black correspond to $\widehat{p_n} \approx 0$ and areas in white correspond to $\widehat{p_n} \approx 1$.

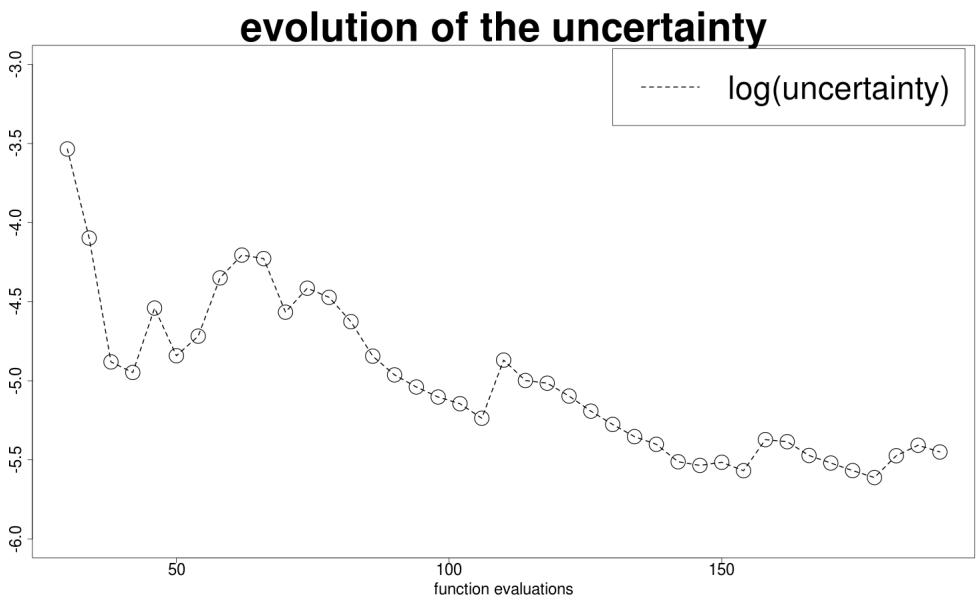

Figure 6.2: Evolution of the uncertainty $\widehat{H}_n$ during the sequential sampling strategy.

Conclusions and future work

In this thesis we study sequential and batch-sequential evaluation strategies of real-valued functions under limited evaluation budget, with a kriging metamodel.

Motivated by real test cases in nuclear safety, we investigated sequential and batch-sequential strategies for inversion problems. The Stepwise Uncertainty Reduction (SUR) paradigm and the related (1-step lookahead) optimal strategies were presented, followed by some outputs of this Ph.D. work in terms of efficient calculation and implementation of SUR strategies using original closed-form criteria, and relying on update formulas. At the IRSN, the use of batch-sequential SUR strategies - which was previously too computer-intensive - is now possible, as a complement to the expertise of the physicists. The kriging update formulas [Chevalier et al., 2013c] are not only useful for the computational savings that they may provide. They can also be used as a tool to investigate the relation between the unknown response at a given point or batch and any quantity depending on the future kriging mean and variance functions, as illustrated for instance in the newest results presented in Appendix F.

Besides, another contribution was proposed in global optimization, with closed-form formulas to compute the multi-points Expected Improvement criterion [Ginsbourger et al., 2010; Schonlau, 1997]. These formulas, derived from Tallis' formulas [Tallis, 1961] allow the use of a batch-sequential EGO algorithm which selects as each iteration a batch of q points, possibly using a gradient algorithm in the space of dimension $d \times q$. The implementation of such an algorithm constitutes an interesting perspective.

In the field of inversion, new SUR strategies based on random set theory were proposed and implemented in the KrigInv R package. These methods have the

advantage of being able to estimate a set and to deliver, together with the estimation, a quantification of the estimation uncertainty. Important work remains to be done on difficult questions of convergence for these SUR strategies, though a first contribution under a deterministic space-filling design assumption was already obtained in Chevalier et al. [2013b]. Also, important auxiliary problems could be further investigated, such as the global optimization of SUR sampling criteria, or the numerical integrations appearing in the calculation of most SUR criteria. For this latter problem, we believe that the use of Sequential Monte-Carlo algorithms may be very well adapted.

Finally, a contribution was proposed for the so-called problem of "robust inversion". We investigated two SUR sequential sampling strategies obtained, again, with more work on the kriging update formulas. The work in progress with updates of GP simulations (see, Appendix F) could open interesting perspectives when one deals with sequential data assimilation and wants to quantify uncertainties using GP conditional simulations. In the domain of robust inversion, another quite ambitious perspective would be to investigate on potential approximations of exceedance probabilities of GPs using the work of Adler and Taylor [2007]. Bridges might be constructed between the (difficult) theory of exceedance probability of stationary GPs and SUR strategies for robust inversion. In particular, a recent article of Taylor et al. [2007] could provide useful solutions for this problem.

Appendix A: Kriging update formulas

Corrected kriging update formulae for batch-sequential data assimilation

Clément Chevalier, David Ginsbourger, Xavier Emery

Abstract Recently, a lot of effort has been paid to the efficient computation of kriging predictors when observations are assimilated sequentially. In particular, kriging update formulae enabling significant computational savings were derived. Taking advantage of the previous kriging mean and variance computations helps avoiding a costly matrix inversion when adding one observation to the n already available ones. In addition to traditional update formulae taking into account a single new observation, Emery (2009) also proposed formulae for the batch-sequential case, i.e. when $k > 1$ new observations are simultaneously assimilated. However, the kriging variance and covariance formulae given in Emery (2009) for the batch-sequential case are not correct. In this work, we fix this issue and establish correct expressions for updated kriging variances and covariances when assimilating observations in parallel. An application in sequential conditional simulation finally shows that coupling update and residual substitution approaches may enable significant speed-ups.

Key words: Gaussian process, kriging weights, sequential conditional simulation

1 Kriging update formulae for batch-sequential data assimilation

Let us consider a real-valued second-order random field Z indexed by $D \subset \mathbb{R}^d$. The term kriging is often used when one aims at calculating a linear predictor $\hat{Z}_n(\mathbf{x})$ and the associated prediction variance $\sigma_n^2(\mathbf{x})$ (often called kriging mean and variance), of the field Z at a point $\mathbf{x} \in D$, from a set of n observations, at locations $\mathbf{x}_1, \ldots, \mathbf{x}_n \in$

Clément Chevalier and David Ginsbourger
IMSV, University of Bern, Alpeneggstrasse 22, 3012 Bern, Switzerland, e-mail: clement.chevalier@stat.unibe.ch,david.ginsbourger@stat.unibe.ch
Xavier Emery
Department of Mining Engineering and Advanced Mining Technology Center, University of Chile, Avenida Tupper 2069, Santiago, Chile, e-mail: xemery@ing.uchile.cl

D. Also, denote by $\sigma_n : (\mathbf{x},\mathbf{y}) \in D^2 \to \sigma_n(\mathbf{x},\mathbf{y}) := \mathbb{E}[(Z(\mathbf{x}) - \hat{Z}_n(\mathbf{x}))(Z(\mathbf{y}) - \hat{Z}_n(\mathbf{y}))]$ the *kriging covariance* function, giving covariances between kriging errors. Kriging means, variances and covariances can be computed using the so-called kriging equations, given in, e.g., [2]. Recently, a lot of effort has been put in reducing the cost for computing kriging means, variances and covariances when the observations are assimilated sequentially. In particular, when n observations are available, one may take advantage of previous computations to reduce the calculation cost of the kriging predictors when $k > 1$ additional observations are available, at locations $\mathbf{x}_{n+1},\ldots,\mathbf{x}_{n+k}$. In that setup, [3] recently proposed the following kriging update formulae:

$$\hat{Z}_{n+k}(\mathbf{x}) = \hat{Z}_n(\mathbf{x}) + \sum_{i=1}^{k} \lambda_{n+i|n+k}(\mathbf{x}) \left(Z(\mathbf{x}_{n+i}) - \hat{Z}_n(\mathbf{x}_{n+i})\right), \tag{1}$$

$$\sigma^2_{n+k}(\mathbf{x}) = \sigma^2_n(\mathbf{x}) - \sum_{i=1}^{k} \lambda^2_{n+i|n+k}(\mathbf{x}) \sigma^2_n(\mathbf{x}_{n+i}), \tag{2}$$

$$\sigma_{n+k}(\mathbf{x},\mathbf{y}) = \sigma_n(\mathbf{x},\mathbf{y}) - \sum_{i=1}^{k} \lambda_{n+i|n+k}(\mathbf{x}) \lambda_{n+i|n+k}(\mathbf{y}) \sigma^2_n(\mathbf{x}_{n+i}) \tag{3}$$

where $\lambda_{n+i|n+k}(\mathbf{x})$ denotes the kriging weight of $Z(\mathbf{x}_{n+i})$ when predicting $Z(\mathbf{x})$ relying on $Z(\mathbf{x}_1),\ldots,Z(\mathbf{x}_{n+k})$. In [3], Eqs. (2), (3) are proven only for $k = 1$. In fact, for $k > 1$, a counter example for Eq. (2) can be obtained rather easily with the Brownian motion, as shown in a draft version of the present paper [1]. The next sections provide corrected formulae and an application in Gaussian field simulation.

2 Corrected kriging update formulae

We now propose corrected expressions that replace Eqs. (2), (3). To improve the readability, we adopt the following simplified notations:

- $\mathbf{X}_{\text{old}} := \{\mathbf{x}_1,\ldots,\mathbf{x}_n\}$, and $\mathbf{X}_{\text{new}} := \{\mathbf{x}_{n+1},\ldots,\mathbf{x}_{n+k}\}$,
- $\mathbf{Z}_{\text{old}} := (Z(\mathbf{x}_1),\ldots,Z(\mathbf{x}_n))$, and $\mathbf{Z}_{\text{new}} := (Z(\mathbf{x}_{n+1}),\ldots,Z(\mathbf{x}_{n+k}))$,
- $\lambda_{\text{new,old}}(\mathbf{x}) := (\lambda_{1|n+k}(\mathbf{x}),\ldots,\lambda_{n|n+k}(\mathbf{x}))^\top$,
- $\lambda_{\text{new,new}}(\mathbf{x}) := (\lambda_{n+1|n+k}(\mathbf{x}),\ldots,\lambda_{n+k|n+k}(\mathbf{x}))^\top$,
- $\sigma^2_{\text{old}}(\mathbf{x}) := \sigma^2_n(\mathbf{x})$, $\sigma^2_{\text{new}}(\mathbf{x}) := \sigma^2_{n+k}(\mathbf{x})$, and similarly for the covariances.

For conciseness and coherence, $\hat{Z}_n(\mathbf{x})$ and $\hat{Z}_{n+k}(\mathbf{x})$ are also denoted by $\hat{Z}_{\text{old}}(\mathbf{x})$ and $\hat{Z}_{\text{new}}(\mathbf{x})$, respectively. The corrected update formulae are given below:

Proposition 1 *(Corrected kriging update equations for the batch-sequential case)*

$$\hat{Z}_{new}(\mathbf{x}) = \hat{Z}_{old}(\mathbf{x}) + \lambda_{new,new}(\mathbf{x})^\top (\mathbf{Z}_{new} - \hat{Z}_{old}(\mathbf{X}_{new})) \tag{4}$$

$$\sigma_{new}(\mathbf{x},\mathbf{y}) = \sigma_{old}(\mathbf{x},\mathbf{y}) - \lambda_{new,new}(\mathbf{x})^\top \Sigma_{new} \lambda_{new,new}(\mathbf{y}) \tag{5}$$

where $\Sigma_{new} := \mathrm{Cov}[\mathbf{Z}_{new} - \hat{Z}_{old}(\mathbf{X}_{new})]$ *is the covariance matrix of kriging errors.*

Note that Eq. 4 is exactly the same result as Eq. 1 (which original proof is correct).

Proof. Subtracting $Z(\mathbf{x})$ to both sides of Eq. 4, simple manipulations give

$$\hat{Z}_{\mathrm{old}}(\mathbf{x}) - Z(\mathbf{x}) = (\hat{Z}_{\mathrm{new}}(\mathbf{x}) - Z(\mathbf{x})) - \lambda_{\mathrm{new,new}}(\mathbf{x})^\top (\mathbf{Z}_{\mathrm{new}} - \hat{Z}_{\mathrm{old}}(\mathbf{X}_{\mathrm{new}})).$$

Using the uncorrelatedness between kriging errors and observations, we then obtain

$$\begin{aligned} Var[\hat{Z}_{\mathrm{old}}(\mathbf{x}) - Z(\mathbf{x})] =& Var[\hat{Z}_{\mathrm{new}}(\mathbf{x}) - Z(\mathbf{x})] + Var[\lambda_{\mathrm{new,new}}(\mathbf{x})^\top (\mathbf{Z}_{\mathrm{new}} - \hat{Z}_{\mathrm{old}}(\mathbf{X}_{\mathrm{new}}))] \\ \sigma^2_{\mathrm{old}}(\mathbf{x}) =& \sigma^2_{\mathrm{new}}(\mathbf{x}) + \lambda_{\mathrm{new,new}}(\mathbf{x})^\top \Sigma_{\mathrm{new}} \lambda_{\mathrm{new,new}}(\mathbf{x}) \end{aligned}$$

which proves Eq. (5) for $\mathbf{x} = \mathbf{y}$. A proof with $\mathbf{x} \neq \mathbf{y}$ can be obtained similarly. □

Proposition 2 *(Kriging update equations in terms of kriging covariance)*

$$\Sigma_{new} \lambda_{new,new}(\mathbf{x}) = \sigma_{old}(\mathbf{X}_{new}, \mathbf{x}) \tag{6}$$

$$\hat{Z}_{new}(\mathbf{x}) = \hat{Z}_{old}(\mathbf{x}) + \sigma_{old}(\mathbf{X}_{new}, \mathbf{x})^T \Sigma_{new}^{-1} (\mathbf{Z}_{new} - \hat{Z}_{old}(\mathbf{X}_{new})) \tag{7}$$

$$\sigma_{new}(\mathbf{x}, \mathbf{y}) = \sigma_{old}(\mathbf{x}, \mathbf{y}) - \sigma_{old}(\mathbf{X}_{new}, \mathbf{x})^T \Sigma_{new}^{-1} \sigma_{old}(\mathbf{X}_{new}, \mathbf{y}) \tag{8}$$

Proof. We prove Eq. (6) using a Gaussian assumption on the field Z. The formula remains valid in non-Gaussian cases as the best linear prediction and the conditional expectation coincide in the Gaussian case. Using the orthogonal projection interpretation of the conditional expectation,

$$\begin{aligned} Z(\mathbf{x}) &= \mathbb{E}(Z(\mathbf{x})|\mathbf{Z}_{\mathrm{old}}, \mathbf{Z}_{\mathrm{new}}) + \overbrace{Z(\mathbf{x}) - \mathbb{E}(Z(\mathbf{x})|\mathbf{Z}_{\mathrm{old}}, \mathbf{Z}_{\mathrm{new}})}^{=:\varepsilon} \\ &= \lambda_{\mathrm{new,old}}(\mathbf{x})^\top \mathbf{Z}_{\mathrm{old}} + \lambda_{\mathrm{new,new}}(\mathbf{x})^\top \mathbf{Z}_{\mathrm{new}} + \varepsilon \ , \end{aligned}$$

with ε centered, and independent of $\mathbf{Z}_{\mathrm{old}}$ and $\mathbf{Z}_{\mathrm{new}}$. Let us now calculate the conditional covariance between $Z(\mathbf{x})$ and $\mathbf{Z}_{\mathrm{new}}$ knowing the observations $\mathbf{Z}_{\mathrm{old}}$:

$$\begin{aligned} \sigma_{\mathrm{old}}(\mathbf{X}_{\mathrm{new}}, \mathbf{x}) :=& \, Cov(\mathbf{Z}_{\mathrm{new}}, Z(\mathbf{x})|\mathbf{Z}_{\mathrm{old}}) \\ =& \, 0 + Cov\left(\mathbf{Z}_{\mathrm{new}}, \lambda_{\mathrm{new,new}}(\mathbf{x})^\top \mathbf{Z}_{\mathrm{new}} \,\middle|\, \mathbf{Z}_{\mathrm{old}}\right) + Cov(\mathbf{Z}_{\mathrm{new}}, \varepsilon|\mathbf{Z}_{\mathrm{old}}) \\ =& \, \Sigma_{\mathrm{new}} \lambda_{\mathrm{new,new}}(\mathbf{x}) + Cov(\mathbf{Z}_{\mathrm{new}}, \varepsilon|\mathbf{Z}_{\mathrm{old}}) \end{aligned}$$

Noting that $Cov(\mathbf{Z}_{\mathrm{new}}, \varepsilon|\mathbf{Z}_{\mathrm{old}}) = \mathbf{0}$, the latter equation proves Eq. (6). Eqs. (7), and (8) follow by plugging in Eq. (6) into Eqs. (4), (5). □

3 GP simulation, with batch-sequential data assimilation

A well known algorithm for simulating M Gaussian process (GP) realizations in p points conditionally on n observations consists in adding to the kriging mean ob-

tained with the n real observations M kriging residual functions artificially obtained based on non-conditional realizations [4, 2]. The kriging update formulae can be used in this algorithm to reduce computation costs in the case where one aims at smoothly "converting" GP realizations conditioned on n observations to realizations conditioned on $n+k$ observations (see, Fig.1, with $n=6$ and $k=3$). Computing kriging means knowing n observations and k real or simulated new observations (denoted by $\mathbf{Z}_{\text{new}}$ and $\mathbf{Z}_{\text{sim}}$ respectively) requires to use Eq. (4). It appears that the difference between these two updated kriging mean functions (i.e. with k observations equal to $\mathbf{Z}_{\text{new}}$ and $\mathbf{Z}_{\text{sim}}$) only depends on $\mathbf{Z}_{\text{new}} - \mathbf{Z}_{\text{sim}}$ and on $\lambda_{\text{new,new}}$, which can be obtained from Eq. 6. Then, the calculation of $\lambda_{\text{new,new}}^{\top}(\mathbf{Z}_{\text{new}} - \mathbf{Z}_{\text{sim}})$ for p points has $O(pk)$ complexity and $O(Mpk)$ for M simulations. This is faster than standard algorithms based on a decomposition (e.g., LU or Cholesky) of the $p \times p$ covariance matrix which require M matrix-vector product for a cost of $O(Mp^2)$. The gain of $O(p/k)$ can be substantial: in an application set up, with $M = 100000$, $p = 200$, $n = 6$, $k = 3$, the computation time is divided by more than 10.

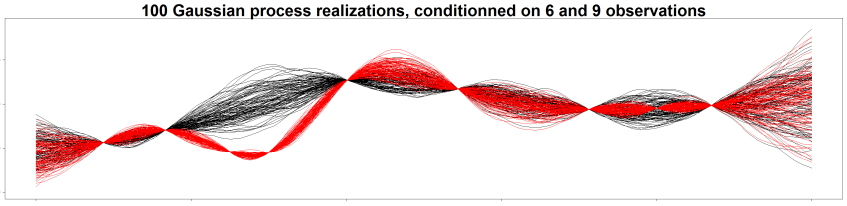

Fig. 1 100 GP realizations conditioned on 6 data (black lines) and 9 data (red lines)

Acknowledgements Part of this work has been conducted within the frame of the ReDice Consortium, gathering industrial (CEA, EDF, IFPEN, IRSN, Renault) and academic (Ecole des Mines de Saint-Etienne, INRIA, and the University of Bern) partners around advanced methods for Computer Experiments. Clément Chevalier also gratefully acknowledges support from the French Nuclear Safety Institute (IRSN) and warmly thanks Prof. Julien Bect for fruitful discussions on Gaussian process simulations. David acknowledges support from the IMSV.

References

1. Chevalier, C., Ginsbourger, D.: Corrected kriging update formulae for batch-sequential data assimilation (2012). URL http://arxiv.org/abs/1203.6452
2. Chilès, J.P., Delfiner, P.: Geostatistics: Modeling Spatial Uncertainty, second edition edn. Wiley, New York (2012)
3. Emery, X.: The kriging update equations and their application to the selection of neighboring data. Computational Geosciences **13**(1), 211–219 (2009)
4. Hoshiya, M.: Kriging and conditional simulation of gaussian field. Journal of engineering mechanics **121**(2), 181–186 (1995)

Appendix B: Explicit formulas for the Multi-points Expected Improvement

Fast Computation of the Multi-points Expected Improvement with Applications in Batch Selection

Clément Chevalier[1,2] and David Ginsbourger[2]

[1] Institut de Radioprotection et de Sûreté Nucléaire (IRSN)
31, avenue de la Division Leclerc, 92260 Fontenay-aux-Roses, France
[2] IMSV, University of Bern,
Alpeneggstrasse 22, 3012 Bern, Switzerland
{clement.chevalier,ginsbourger}@stat.unibe.ch

Abstract. The Multi-points Expected Improvement criterion (or q-EI) has recently been studied in batch-sequential Bayesian Optimization. This paper deals with a new way of computing q-EI, without using Monte-Carlo simulations, through a closed-form formula. The latter allows a very fast computation of q-EI for reasonably low values of q (typically, less than 10). New parallel kriging-based optimization strategies, tested on different toy examples, show promising results.

Keywords: Computer Experiments, Kriging, Parallel Optimization, Expected Improvement

1 Introduction

In the last decades, *metamodeling* (or *surrogate modeling*) has been increasingly used for problems involving costly computer codes (or "black-box simulators"). Practitioners typically dispose of a very limited evaluation budget and aim at selecting evaluation points cautiously when attempting to solve a given problem.

In global optimization, the focus is usually put on a real-valued function f with d-dimensional source space. In this settings, [1] proposed the now famous *Efficient Global Optimization* (EGO) algorithm, relying on a kriging metamodel [2] and on the Expected Improvement (EI) criterion [3]. In EGO, the optimization is done by sequentially evaluating f at points maximizing EI. A crucial advantage of this criterion is its fast computation (besides, the analytical gradient of EI is implemented in [4]), so that the hard optimization problem is replaced by series of much simpler ones.

Coming back to the decision-theoretic roots of EI [5], a Multi-points Expected Improvement (also called "q-EI") criterion for batch-sequential optimization was defined in [6] and further developed in [7, 8]. Maximizing this criterion enables choosing batches of $q > 1$ points at which to evaluate f in parallel, and is of particular interest in the frequent case where several CPUs are simultaneously available. Even though an analytical formula was derived for the 2-EI in [7], the

Monte Carlo (MC) approach of [8] for computing q-EI when $q \geq 3$ makes the criterion itself expensive-to-evaluate, and particularly hard to optimize.

A lot of effort has recently been paid to address this problem. The pragmatic approach proposed by [8] consists in circumventing a direct q-EI maximization, and replacing it by simpler strategies where batches are obtained using an offline q-points EGO. In such strategies, the model updates are done using dummy response values such as the kriging mean prediction (Kriging Believer) or a constant (Constant Liar), and the covariance parameters are re-estimated only when real data is assimilated. In [9] and [10], q-EI optimization strategies were proposed relying on the MC approach, where the number of MC samples is tuned online to discriminate between candidate designs. Finally, [11] proposed a q-EI optimization strategy involving stochastic gradient, with the crucial advantage of *not* requiring to evaluate q-EI itself.

In this article we derive a formula allowing a fast and accurate approximate evaluation of q-EI. This formula may contribute to significantly speed up strategies relying on q-EI. The main result, relying on Tallis' formula, is given in Section 2. The usability of the proposed formula is then illustrated in Section 3 through benchmark experiments, where a brute force maximization of q-EI is compared to three variants of the Constant Liar strategy. In particular, a new variant (CL-mix) is introduced, and is shown to offer very good performances at a competitive computational cost. For self-containedness, a slightly revisited proof of Tallis' formula is given in appendix.

2 Multi-points Expected Improvement explicit formulas

In this section we give an explicit formula allowing a fast and accurate deterministic approximation of q-EI. Let us first give a few precisions on the mathematical settings. Along the paper, f is assumed to be one realisation of a Gaussian Process (GP) with known covariance kernel and mean known up to some linear trend coefficients, so that the conditional distribution of a vector of values of the GP conditional on past observations is still Gaussian (an improper uniform prior is put on the trend coefficients when applicable). This being said, most forthcoming derivations boil down to calculations on Gaussian vectors. Let $\mathbf{Y} := (Y_1, \ldots, Y_q)$ be a Gaussian Vector with mean $\mathbf{m} \in \mathbb{R}^q$ and covariance matrix Σ. Our aim in this paper is to explicitly calculate expressions of the following kind:

$$\mathbb{E}\left[\left(\max_{i\in\{1,\ldots,q\}} Y_i - T\right)_+\right] \tag{1}$$

where $(.)_+ := \max(., 0)$. In Bayesian optimization (say maximization), expectations and probabilities are taken conditional on response values at a given set of n points $(\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{X}^n$ where $\mathbb{X}$ is the input set of f (often, a compact subset of $\mathbb{R}^d$, $d \geq 1$), the threshold $T \in \mathbb{R}$ is usually the maximum of those n available response values, and $\mathbf{Y}$ is the vector of unknown responses at a given batch of q points, $\mathbf{X}^q := (\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+q}) \in \mathbb{X}^q$. In such framework, the vector $\mathbf{m}$ and the

matrix Σ are the so-called "Kriging mean" and "Kriging covariance" at $\mathbf{X}^q$ and can be calculated relying on classical Kriging equations (see, e.g., [12]).

In order to obtain a tractable analytical formula for Expression (1), not requiring any Monte-Carlo simulation, let us first give a useful formula obtained by [13], and recently used in [14] for GP modeling with inequality constraints:

Proposition 1 (Tallis' formulas) *Let $\mathbf{Z} := (Z_1, \ldots, Z_q)$ be a Gaussian Vector with mean $\mathbf{m} \in \mathbb{R}^q$ and covariance matrix $\Sigma \in \mathbb{R}^{q \times q}$. Let $\mathbf{b} = (b_1, \ldots, b_q) \in \mathbb{R}^q$. The expectation of any coordinate Z_k under the linear constraint $(\forall j \in \{1, \ldots, q\},\ Z_j \leq b_j)$ denoted by $\mathbf{Z} \leq \mathbf{b}$ can be expanded as follows:*

$$\mathbb{E}(Z_k | \mathbf{Z} \leq \mathbf{b}) = m_k - \frac{1}{p} \sum_{i=1}^{q} \Sigma_{ik}\ \varphi_{m_i, \Sigma_{ii}}(b_i)\ \Phi_{q-1}\left(\mathbf{c}_{.i}, \Sigma_{.i}\right) \tag{2}$$

where:

- $p := \mathbb{P}(\mathbf{Z} \leq \mathbf{b}) = \Phi_q(\mathbf{b} - \mathbf{m}, \Sigma)$
- $\Phi_q(\mathbf{u}, \Sigma)$ $(\mathbf{u} \in \mathbb{R}^q, \Sigma \in \mathbb{R}^{q \times q}, q \geq 1)$ *is the c.d.f. of the centered multivariate Gaussian distribution with covariance matrix Σ.*
- $\varphi_{m,\sigma^2}(.)$ *is the p.d.f. of the univariate Gaussian distribution with mean m and variance σ^2*
- $\mathbf{c}_{.i}$ *is the vector of $\mathbb{R}^{q-1}$ with general term $(b_j - m_j) - (b_i - m_i)\frac{\Sigma_{ij}}{\Sigma_{ii}}$, $j \neq i$*
- $\Sigma_{.i}$ *is a $(q-1) \times (q-1)$ matrix obtained by computing $\Sigma_{uv} - \frac{\Sigma_{iu}\Sigma_{iv}}{\Sigma_{ii}}$ for $u \neq i$ and $v \neq i$. This matrix corresponds to the conditional covariance matrix of the random vector $\mathbf{Z}_{-i} := (Z_1, \ldots, Z_{i-1}, Z_{i+1}, \ldots, Z_q)$ knowing Z_i.*

For the sake of brevity, the proof of this Proposition is sent in the Appendix. A crucial point for the practical use of this result is that there exist very fast procedures to compute the c.d.f. of the multivariate Gaussian distribution. For example, the work of [15], [16] have been used in many R packages (see, e.g., [17], [18]). The Formula (2) above is an important tool to efficiently compute Expression (1) as shown with the following Property:

Proposition 2 *Let $\mathbf{Y} := (Y_1, \ldots, Y_q)$ be a Gaussian Vector with mean $\mathbf{m} \in \mathbb{R}^q$ and covariance matrix Σ. For $k \in \{1, \ldots, q\}$ consider the Gaussian vectors $\mathbf{Z}^{(k)} := (Z_1^{(k)}, \ldots, Z_q^{(k)})$ defined as follows:*

$$\begin{aligned} Z_j^{(k)} &:= Y_j - Y_k\ ,\ j \neq k \\ Z_k^{(k)} &:= -Y_k \end{aligned}$$

Denoting by $\mathbf{m}^{(k)}$ and $\Sigma^{(k)}$ the mean and covariance matrix of $\mathbf{Z}^{(k)}$, and defining the vector $\mathbf{b}^{(k)} \in \mathbb{R}^q$ by $b_k^{(k)} = -T$ and $b_j^{(k)} = 0$ if $j \neq k$, the EI of $\mathbf{X}^q$ writes:

$$\boxed{EI(\mathbf{X}^q) = \sum_{k=1}^{q} \left((m_k - T)p_k + \sum_{i=1}^{q} \Sigma_{ik}^{(k)} \varphi_{m_i^{(k)}, \Sigma_{ii}^{(k)}}(b_i^{(k)}) \Phi_{q-1}\left(\mathbf{c}_{.i}^{(k)}, \Sigma_{.i}^{(k)}\right) \right)} \tag{3}$$

where:

- $p_k := \mathbb{P}(\mathbf{Z}^{(k)} \leq \mathbf{b}^{(k)}) = \Phi_q(\mathbf{b}^{(k)} - \mathbf{m}^{(k)}, \Sigma^{(k)})$.
  p_k *is actually the probability that* Y_k *exceeds* T *and* $Y_k = \max_{j=1,\ldots,q} Y_j$.
- $\Phi_q(.,\Sigma)$ *and* $\varphi_{m,\sigma^2}(.)$ *are defined in Proposition 1*
- $\mathbf{c}_{.i}^{(k)}$ *is the vector of* $\mathbb{R}^{q-1}$ *constructed like in Proposition 1, by computing* $(b_j^{(k)} - m_j^{(k)}) - (b_i^{(k)} - m_i^{(k)})\frac{\Sigma_{ij}^{(k)}}{\Sigma_{ii}^{(k)}}$, *with* $j \neq i$
- $\Sigma_{.i}^{(k)}$ *is a* $(q-1)\times(q-1)$ *matrix constructed from* $\Sigma^{(k)}$ *like in Proposition 1. It corresponds to the conditional covariance matrix of the random vector* $\mathbf{Z}_{-i}^{(k)} := (Z_1^{(k)}, \ldots, Z_{i-1}^{(k)}, Z_{i+1}^{(k)}, \ldots, Z_q^{(k)})$ *knowing* $Z_i^{(k)}$.

Proof. Using that $\mathbb{1}_{\{\max_{i\in\{1,\ldots,q\}} Y_i \geq T\}} = \sum_{k=1}^{q} \mathbb{1}_{\{Y_k \geq T,\ Y_j \leq Y_k\ \forall j\neq k\}}$, we get

$$\begin{aligned}
EI(\mathbf{X}^q) &= \mathbb{E}\left[\left(\max_{i\in\{1,\ldots,q\}} Y_i - T\right)\sum_{k=1}^{q} \mathbb{1}_{\{Y_k \geq T,\ Y_j \leq Y_k\ \forall j\neq k\}}\right] \\
&= \sum_{k=1}^{q} \mathbb{E}\left((Y_k - T)\mathbb{1}_{\{Y_k \geq T,\ Y_j \leq Y_k\ \forall j\neq k\}}\right) \\
&= \sum_{k=1}^{q} \mathbb{E}\left(Y_k - T \middle| Y_k \geq T,\ Y_j \leq Y_k\ \forall j \neq k\right) \mathbb{P}\left(Y_k \geq T, Y_j \leq Y_k\ \forall j \neq k\right) \\
&= \sum_{k=1}^{q} \left(-T - \mathbb{E}\left(Z_k^{(k)} \middle| \mathbf{Z}^{(k)} \leq \mathbf{b}^{(k)}\right)\right) \mathbb{P}\left(\mathbf{Z}^{(k)} \leq \mathbf{b}^{(k)}\right)
\end{aligned}$$

Now the computation of $p_k := \mathbb{P}\left(\mathbf{Z}^{(k)} \leq \mathbf{b}^{(k)}\right)$ simply requires one call to the Φ_q function and the proof can be completed by applying Tallis' formula (2) to the random vectors $\mathbf{Z}^{(k)}$ ($1 \leq k \leq q$).

Remark 1. From Properties (1) and (2), it appears that computing q-EI requires a total of q calls to Φ_q and q^2 calls to Φ_{q-1}. The proposed approach performs thus well when q is moderate (typically lower than 10). For higher values of q, estimating q-EI by Monte-Carlo might remain competitive. Note that, when q is larger (say, $q = 50$) and when q CPUs are available, one can always split the calculations of the q^2 calls to Φ_{q-1} on these q CPUs.

Remark 2. In the particular case $q = 1$ and with the convention $\Phi_0(.,\Sigma) = 1$, Equation (3) corresponds to the classical EI formula proven in [5, 1].

Remark 3. The Multi-points EI can be used in a batch-sequential strategy to optimize a given expensive-to-evaluate function f, as detailed in the next Section. Moreover, the same criterion can also be used for *constrained* optimization provided that the constraints are linear. For example an expression like: $\mathbb{E}\left[\left(\max_{i\in\{1,\ldots,q\}} Y_i - T\right)_+ | \mathbf{Y} \leq \mathbf{a}\right], \mathbf{a} \in \mathbb{R}^q$, can also be computed using Tallis' formula and the same proof.

3 Batch sequential optimization using Multi-points EI

Let us first illustrate Proposition 2 and show that the proposed q-EI calculation based on Tallis' formula is actually consistent with a Monte Carlo estimation. From a kriging model based on 12 observations of the Branin-Hoo function [1], we generated a 4-point batch (Figure 1, left plot) and calculated its q-EI value (middle plot, dotted line). The MC estimates converge to a value close to the latter, and the relative error after $5 * 10^9$ runs is less than 10^{-5}. 4-point batches generated from the three strategies detailed below are drawn on the right plot.

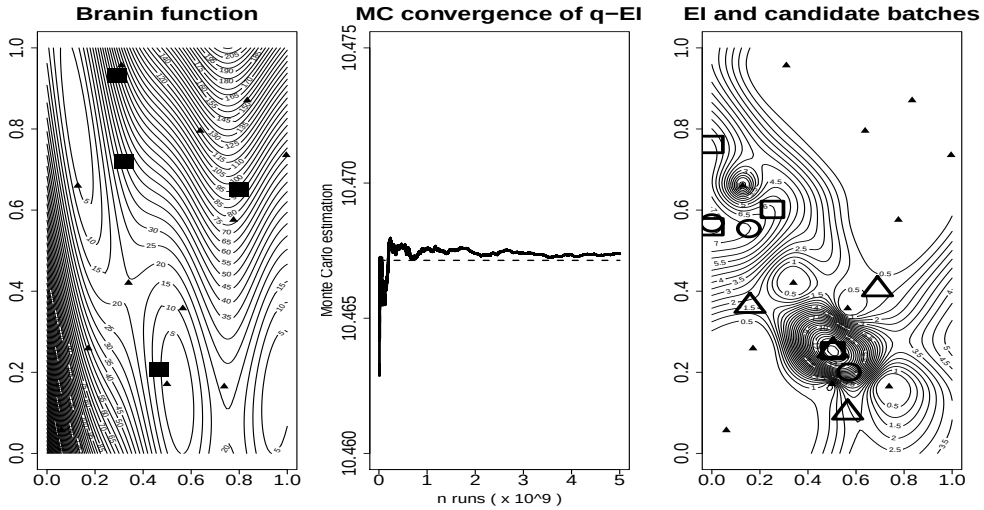

Fig. 1. Convergence (middle) of MC estimates to the q-EI value calculated with Proposition 2 in the case of a batch of four points (shown on the left plot). Right: candidate batches obtained by q-EI stepwise maximisation (squares), and the CL-min (circles) and CL-max (triangles) strategies.

We now compare a few kriging-based batch-sequential optimization methods on two different functions: the function $x \mapsto -\log(-\text{Hartman6}(x))$ (see, e.g., [1]), defined on $[0,1]^6$ and the Rastrigin function ([19, 20]) in dimension two restricted to the domain $[0, 2.5]^2$. The first function in dimension 6 is unimodal, while the second one has a lot of local optima (see: Figure 2). The Rastrigin function is one of the 24 noiseless test function of the Black-Box Optimization Benchmark (BBOB) [19].

For each runs, we start with a random initial Latin hypercube design (LHS) of $n_0 = 10$ (Rastrigin) or 50 (Hartman6) points and estimate the covariance parameters by Maximum Likelihood (here a Matérn kernel with $\nu = 3/2$ is chosen). For both functions and all strategies, batches of $q = 6$ points are added

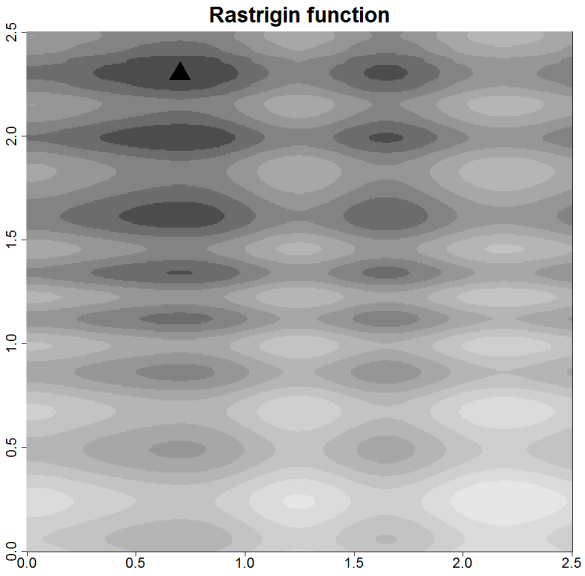

Fig. 2. Contour lines of the Rastrigin function (grayscale) and location of the global optimizer (black triangle)

at each iteration, and the covariance parameters are re-estimated after each batch assimilation. Since the tests are done for several designs of experiments, we chose to represent, along the runs, the relative mean squared error:

$$\mathrm{rMSE} = \frac{1}{M}\sum_{i=1}^{\mathrm{M}} \left(\frac{y_{\min}^{(i)} - y_{\mathrm{opt}}}{y_{\mathrm{opt}}} \right)^2 \tag{4}$$

where $y_{\min}^{(i)}$ in the current observed minimum in run number i and y_{opt} is the real unknown optimum. The total number M of different initial designs of experiments is fixed to 50. The tested strategies are:

- (1) q-EI stepwise maximization: q sequential d-dimensional optimizations are performed. We start with the maximization of the 1-point EI and add this point to the new batch. We then maximize the 2-point EI (keeping the first point obtained as first argument), add the maximizer to the batch, and iterate until q points are selected.
- (2) Constant Liar min (CL-min): We start with the maximization of the 1-point EI and add this point to the new batch. We then assume a dummy response (a"lie") at this point, and update the Kriging metamodel with this point and the lie. We then maximize the 1-point EI obtained with the updated kriging metamodel, get a second point, and iterate the same process

until a batch of q points is selected. The dummy response has the same value over the $q-1$ lies, and is here fixed to the minimum of the current observations.
- (3) Constant Liar max (CL-max): The lie in this Constant Liar strategy is fixed to the maximum of the current observations.
- (4) Constant Liar mix (CL-mix): At each iteration, two batches are generated with the CL-min and CL-max strategies. From these two "candidate" batches, we choose the batch with the best actual q-EI value, calculated based on Proposition 2.
- (5) Random sampling.

Note that CL-min tends to explore the function near the current minimizer (as the lie is a low value and we are minimizing f) while CL-max is more exploratory. Thus, CL-min is expected to perform well on unimodal functions. On the contrary, CL-max may perform better on multimodal functions. For all the tests we use the DiceKriging and DiceOptim packages [4]. The optimizations of the different criteria rely on a genetic algorithm using derivatives, available in the rgenoud package [21]. Figure 3 represents the compared performances of these strategies.

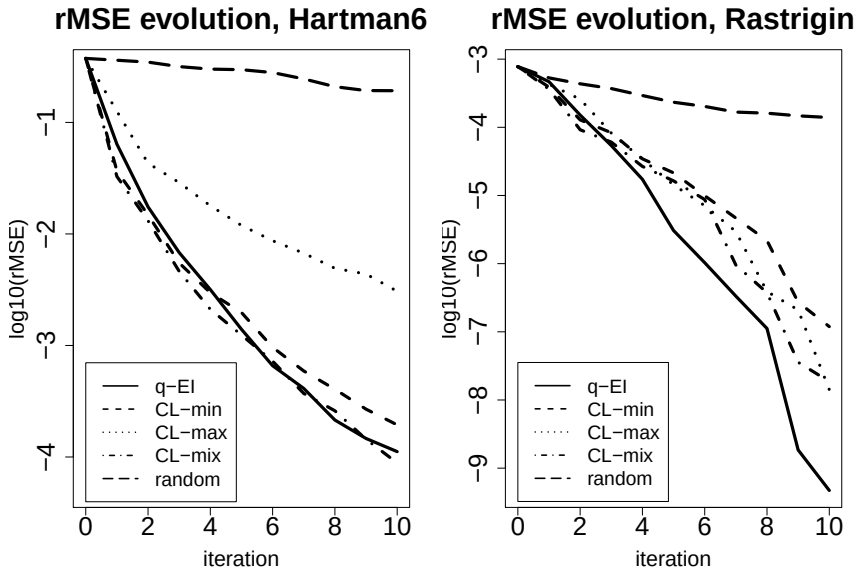

Fig. 3. Compared performances of the five considered batch-sequential optimization strategies, on two test functions.

From these plots we draw the following conclusions: first, the q-EI stepwise maximization strategy outperforms the strategies based on constant lies, CL-

min and CL-max. However, the left graph of Figure 3 points out that the CL-min strategy seems particularly well-adapted to the Hartman6 function. Since running a CL is computationally much cheaper than a brute fore optimization of q-EI, it is tempting to recommend the CL-min strategy for Hartman6. However, it is not straightforward to know in advance which of CL-min or CL-max will perform better on a given test case. Indeed, for example, CL-max outperforms CL-min on the Rastrigin function.

Now, we observe that using q-EI in the CL-mix heuristic enables very good performances in both cases without having to select one of the two lie values in advance. For the Hartman6 function, CL-mix even outperforms both CL-min and CL-max and has roughly the same performance as a brute force q-EI maximization. This suggests that a good heuristic might be to generate, at each iteration, candidate batches obtained with different strategies (e.g. CL with different lies) and to discriminate those batches using q-EI.

Conclusion

In this article we give a closed-form expression enabling a fast computation of the Multi-points Expected Improvement criterion for batch sequential Bayesian global optimization. This formula is consistent with the classical Expected Improvement formula and its computation does not require Monte Carlo simulations. Optimization strategies based on this criterion are now ready to be used on real test cases, and a brute maximization of this criterion shows promising results. In addition, we show that good performances can be achieved by using a cheap-to-compute criterion and by discriminating the candidate batches generated by such criterion with the q-EI. Such heuristics might be particularly interesting when the time needed to generate batches becomes a computational bottleneck, e.g. when $q \geq 10$ and calls to the Gaussian c.d.f. become expensive.

A perspective, currently under study, is to improve the maximization of q-EI itself, e.g. through a more adapted choice of the algorithm and/or an analytical calculation of q-EI's gradient.

Acknowledgments This work has been conducted within the frame of the ReDice Consortium, gathering industrial (CEA, EDF, IFPEN, IRSN, Renault) and academic (Ecole des Mines de Saint-Etienne, INRIA, and the University of Bern) partners around advanced methods for Computer Experiments. Clément Chevalier gratefully acknowledges support from the French Nuclear Safety Institute (IRSN). The authors also would like to thank Dr. Sébastien Da Veiga for raising our attention to Tallis' formula.

References

1. Jones, D.R., Schonlau, M., William, J.: Efficient global optimization of expensive black-box functions. Journal of Global Optimization **13**(4) (1998) 455–492

2. Santner, T.J., Williams, B.J., Notz, W.: The Design and Analysis of Computer Experiments. Springer Verlag (2003)
3. Mockus, J.: Bayesian Approach to Global Optimization. Theory and Applications. Kluwer Academic Publisher, Dordrecht (1989)
4. Roustant, O., Ginsbourger, D., Deville, Y.: DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by Kriging-Based Metamodelling and Optimization. Journal of Statistical Software **51 (1)** (2012) 1–55
5. Mockus, J., Tiesis, V., Zilinskas, A.: The application of Bayesian methods for seeking the extremum. In Dixon, L., Szego, E.G., eds.: Towards Global Optimization. Volume 2. Elsevier (1978) 117–129
6. Schonlau, M.: Computer Experiments and global optimization. PhD thesis, University of Waterloo (1997)
7. Ginsbourger, D.: Métamodèles multiples pour l'approximation et l'optimisation de fonctions numériques multivariables. PhD thesis, Ecole nationale supérieure des Mines de Saint-Etienne (2009)
8. Ginsbourger, D., Le Riche, R., L., C.: Kriging is well-suited to parallelize optimization. In: Computational Intelligence in Expensive Optimization Problems. Volume 2 of Adaptation Learning and Optimization. Springer (2010) 131–162
9. Janusevskis, J., Le Riche, R., Ginsbourger, D.: Parallel expected improvements for global optimization: summary, bounds and speed-up. (August 2011)
10. Janusevskis, J., Le Riche, R., Ginsbourger, D., Girdziusas, R.: Expected improvements for the asynchronous parallel global optimization of expensive functions : Potentials and challenges. In: LION 6 Conference (Learning and Intelligent OptimizatioN), Paris : France. (2012)
11. Frazier, P.I.: Parallel global optimization using an improved multi-points expected improvement criterion. In: INFORMS Optimization Society Conference, Miami FL. (2012)
12. Chilès, J.P., Delfiner, P.: Geostatistics: Modeling Spatial Uncertainty. Wiley, New York (1999)
13. Tallis, G.: The moment generating function of the truncated multi-normal distribution. J. Roy. Statist. Soc. Ser. B **23**(1) (1961) 223–229
14. Da Veiga, S., Marrel, A.: Gaussian process modeling with inequality constraints. Annales de la Faculté des Sciences de Toulouse **21 (3)** (2012) 529–555
15. Genz, A.: Numerical computation of multivariate normal probabilities. Journal of Computational and Graphical Statistics **1** (1992) 141–149
16. Genz, A., Bretz, F.: Computation of Multivariate Normal and t Probabilities. Springer-Verlag (2009)
17. Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., Bornkamp, B., Hothorn, T.: mvtnorm: Multivariate Normal and t Distributions. (2012) R package version 0.9-9992.
18. Azzalini, A.: mnormt: The multivariate normal and t distributions. (2012) R package version 1.4-5.
19. Finck, S., Hansen, N., Ros, R., Auger, A.: Real-parameter black-box optimization bencharking 2009: Presentation of the noiseless functions. Technical report, Research Center PPE, 2009 (2010)
20. Hansen, N., Finck, S., Ros, R., Auger, A.: Real-parameter black-box optimization bencharking 2009: Noiseless functions definitions. Technical report, INRIA 2009 (2010)
21. Mebane, W., Sekhon, J.: Genetic optimization using derivatives: The rgenoud package for r. Journal of Statistical Software **Vol. 42, Issue 11** (2011) 1–26

22. Cressie, N., Davis, A., Leroy Folks, J.: The moment-generating function and negative integer moments. The American Statistician **35 (3)** (1981) 148–150

Appendix: proof for Tallis' formula (2)

The proof proposed here follows exactly the method given in [13] in the particular case of a centered Gaussian Vector with normalized covariance matrix (i.e. a covariance matrix equal to the *correlation* matrix). Here, the proof is slightly more detailed and applies in a more general case.

Let $\mathbf{Z} := (Z_1, \ldots, Z_q) \sim \mathcal{N}(\mathbf{m}, \Sigma)$ with $\mathbf{m} \in \mathbb{R}^q$ and $\Sigma \in \mathbb{R}^{q \times q}$. Let $\mathbf{b} = (b_1, \ldots, b_q) \in \mathbb{R}^q$. Our goal is to calculate: $\mathbb{E}(Z_k | \mathbf{Z} \leq \mathbf{b})$. The method proposed by Tallis consists in calculating the conditional joint moment generating function (MGF) of $\mathbf{Z}$ defined as follows:

$$M_{\mathbf{Z}}(\mathbf{t}) := \mathbb{E}(\exp(\mathbf{t}^\top \mathbf{Z}) | \mathbf{Z} \leq \mathbf{b}) \tag{5}$$

It is known (see, e.g., [22]) that the conditional expectation of Z_k can be obtained by deriving such MGF with respect to t_k, in $\mathbf{t} = \mathbf{0}$. Mathematically this writes:

$$\mathbb{E}(Z_k | \mathbf{Z} \leq \mathbf{b}) = \left. \frac{\partial M_{\mathbf{Z}}(\mathbf{t})}{\partial t_k} \right|_{\mathbf{t}=\mathbf{0}} \tag{6}$$

The main steps of this proof are then to calculate such MGF and its derivative with respect to any coordinate t_k.

Let us consider the **centered** random variable $\mathbf{Z}^c := \mathbf{Z} - \mathbf{m}$. Denoting $\mathbf{h} = \mathbf{b} - \mathbf{m}$, conditioning on $\mathbf{Z} \leq \mathbf{b}$ or on $\mathbf{Z}^c \leq \mathbf{h}$ are equivalent. The MGF of $\mathbf{Z}^c$ can be calculated as follows:

$$\begin{aligned} M_{\mathbf{Z}^c}(\mathbf{t}) :=& \mathbb{E}(\exp(\mathbf{t}^\top \mathbf{Z}^c) | \mathbf{Z}^c \leq \mathbf{h}) \\ =& \frac{1}{p} \int_{-\infty}^{h_1} \ldots \int_{-\infty}^{h_q} \exp(\mathbf{t}^\top \mathbf{u}) \varphi_{\mathbf{0},\Sigma}(\mathbf{u}) \mathrm{d}\mathbf{u} \\ =& \frac{1}{p} (2\pi)^{-\frac{q}{2}} |\Sigma|^{-\frac{1}{2}} \int_{-\infty}^{h_1} \ldots \int_{-\infty}^{h_q} \exp\left(-\frac{1}{2} \left(\mathbf{u}^\top \Sigma^{-1} \mathbf{u} - 2\mathbf{t}^\top \mathbf{u} \right) \right) \mathrm{d}\mathbf{u} \end{aligned}$$

where $p := \mathbb{P}(\mathbf{Z} \leq \mathbf{b})$ and $\varphi_{\mathbf{v},\Sigma}(.)$ denotes the p.d.f. of the multivariate normal distribution with mean $\mathbf{v}$ and covariance matrix Σ. The calculation can be continued by noting that:

$$\begin{aligned} M_{\mathbf{Z}^c}(\mathbf{t}) =& \frac{1}{p} (2\pi)^{-\frac{q}{2}} |\Sigma|^{-\frac{1}{2}} \exp\left(\frac{1}{2} \mathbf{t}^\top \Sigma \mathbf{t} \right) \int_{-\infty}^{h_1} \ldots \int_{-\infty}^{h_q} \exp\left(-\frac{1}{2} \left(\mathbf{u} - \Sigma \mathbf{t} \right)^\top \Sigma^{-1} \left(\mathbf{u} - \Sigma \mathbf{t} \right) \right) \mathrm{d}\mathbf{u} \\ =& \frac{1}{p} \exp\left(\frac{1}{2} \mathbf{t}^\top \Sigma \mathbf{t} \right) \Phi_q(\mathbf{h} - \Sigma \mathbf{t}, \Sigma) \end{aligned}$$

where $\Phi_q(., \Sigma)$ is the c.d.f. of the centered multivariate normal distribution with covariance matrix Σ.

Now, let us calculate for some $k \in \{1, \ldots, q\}$ the partial derivative $\frac{\partial M_{\mathbf{Z}^c}(\mathbf{t})}{\partial t_k}$ in $\mathbf{t} = \mathbf{0}$, which is equal by definition to $\mathbb{E}(Z_k^c | \mathbf{Z}^c \leq \mathbf{h})$.

$$\begin{aligned} p\ \mathbb{E}(Z_k^c | \mathbf{Z}^c \leq \mathbf{h}) &= p\ \left.\frac{\partial M_{\mathbf{Z}^c}(\mathbf{t})}{\partial t_k}\right|_{\mathbf{t}=\mathbf{0}} \\ &= 0 + 1.\frac{\partial}{\partial t_k}\left(\Phi_q\left(\mathbf{h} - t_k\begin{pmatrix}\Sigma_{1k}\\ \vdots \\ \Sigma_{qk}\end{pmatrix}, \Sigma\right)\right)\Bigg|_{t_k=0} \\ &= -\sum_{i=1}^{q}\Sigma_{ik}\int_{-\infty}^{h_1}\!\!\ldots\int_{-\infty}^{h_{i-1}}\int_{-\infty}^{h_{i+1}}\!\!\ldots\int_{-\infty}^{h_q}\varphi_{\mathbf{0},\Sigma}(\mathbf{u}_{-i}, u_i = h_i)\mathrm{d}\mathbf{u}_{-\mathrm{i}} \end{aligned}$$

The last step is obtained applying the chain rule to $\mathbf{x} \mapsto \Phi_q(\mathbf{x}, \Sigma)$ at the point $\mathbf{x} = \mathbf{h}$. Here, $\varphi_{\mathbf{0},\Sigma}(\mathbf{u}_{-i}, u_i = h_i)$ denotes the c.d.f. of the centered multivariate normal distribution at given points $(\mathbf{u}_{-i}, u_i = h_i) := (u_1, \ldots, u_{i-1}, h_i, u_{i+1}, \ldots, u_q)$. Note that the integrals in the latter Expression are in dimension $q-1$ and not q. In the i^{th} term of the sum above, we integrate with respect to all the q components except the component i. To continue the calculation we can use the identity:

$$\forall \mathbf{u} \in \mathbb{R}^q, \varphi_{\mathbf{0},\Sigma}(\mathbf{u}) = \varphi_{0,\Sigma_{ii}}(u_i)\varphi_{\Sigma_{ii}^{-1}\Sigma_i u_i, \Sigma_{-i,-i} - \Sigma_i\Sigma_{ii}^{-1}\Sigma_i^\top}(\mathbf{u}_{-i}) \tag{7}$$

where $\Sigma_i = (\Sigma_{1i}, \ldots, \Sigma_{i-1i}, \Sigma_{i+1i}, \ldots, \Sigma_{qi})^\top$ ($\Sigma_i \in \mathbb{R}^{q-1}$) and $\Sigma_{-i,-i}$ is the $(q-1) \times (q-1)$ matrix obtained by removing the line and column i from Σ. This identity can be proven using Bayes formula and Gaussian vectors conditioning formulas. Its use gives:

$$\begin{aligned} p\ \mathbb{E}(Z_k^c | \mathbf{Z}^c \leq \mathbf{h}) &= -\sum_{i=1}^{q}\Sigma_{ik}\varphi_{0,\Sigma_{ii}}(h_i)\Phi_{q-1}(\mathbf{h}_{-i} - \Sigma_{ii}^{-1}\Sigma_i h_i, \Sigma_{-i,-i} - \Sigma_i\Sigma_{ii}^{-1}\Sigma_i^\top) \\ &= -\sum_{i=1}^{q}\Sigma_{ik}\varphi_{m_i,\Sigma_{ii}}(b_i)\Phi_{q-1}(\mathbf{h}_{-i} - \Sigma_{ii}^{-1}\Sigma_i h_i, \Sigma_{-i,-i} - \Sigma_i\Sigma_{ii}^{-1}\Sigma_i^\top) \end{aligned}$$

which finally delivers Tallis' formula, see Equation (2).

Appendix C: Fast computation of batch-sequential SUR strategies for inversion

Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set

Clément Chevalier, IMSV, University of Bern
Julien Bect, Supelec
David Ginsbourger, IMSV, University of Bern
Emmanuel Vazquez, Supelec
Victor Picheny, CERFACS
Yann Richet, IRSN

April 2, 2012

Abstract

Stepwise Uncertainty Reduction (SUR) strategies aim at constructing a sequence of sampling points for a function $f : \mathbb{R}^d \to \mathbb{R}$, in such a way that the residual uncertainty about a quantity of interest becomes small. In the context of Gaussian Process-based approximation of computer experiments, these strategies have been shown to be particularly efficient for the problem of estimating the volume of excursion of a function f above a threshold. However, these strategies remain difficult to use in practice because of their high computational complexity, and they only deliver at each iteration a single point to evaluate. In this paper we introduce parallel sampling criteria, which allow selecting several sampling points simultaneously. Such criteria are of particular interest when the function f is expensive to evaluate and many CPUs are available. We also manage to drastically reduce the computational cost of these strategies using closed form expressions. We illustrate their performances in various numerical experiments, including a nuclear safety test case.

Keywords: Computer experiments, Gaussian processes, Sequential design, Probability of failure, Active learning, Inversion

1 Introduction

Whether in natural sciences, engineering, or economics, the study of complex phenomena is increasingly relying on numerical simulations. From an end user's perspective, a numerical simulator can often be considered as a black box taking a number of real-valued parameters as inputs and returning one or several quantities of interest after a post-processing stage. Formally, the space of inputs is a set $\mathbb{X} \subset \mathbb{R}^d$ and the simulator can be viewed as a function $f : \mathbb{X} \to \mathbb{R}$ that maps the inputs to a cost or a performance indicator. In many practical applications, the objective is to obtain information about the simulator from a number of runs, or, in other words, to infer a quantity of interest from a number of evaluations of f. A problem that is often at stake is the estimation of the probability that a cost exceeds a given threshold. This problem corresponds to the estimation of the volume $\alpha^\star$ of the excursion set $\Gamma^\star = \{x \in \mathbb{X} : f(x) \geq T\}$, with T a given threshold, under a measure $\mathsf{P}_\mathbb{X}$ on $\mathbb{X}$. In safety analysis, $\mathsf{P}_\mathbb{X}$ typically models the uncertainty on input parameters. If f is expensive to evaluate, the estimation of $\alpha^\star$ must be performed with a limited number of evaluations of f, which naturally excludes brute-force approaches like Monte Carlo sampling.

A popular approach consists in constructing a response surface (also known as surrogate or meta-model) based on available evaluations of f, together with an uncertainty measure about this surface. Using this uncertainty measure is one of the key concepts in the design and analysis of computer experiments [see, e.g., Santner et al., 2003, Fang et al., 2006, Bayarri et al., 2007, Forrester et al., 2008, and references therein]. It has been found to be a convenient and powerful tool, providing efficient answers to the issues of designing experiments (Sacks et al. [1989]) or global optimization (Jones et al. [1998]) for instance.

For the problem of estimating a probability of failure, several sampling strategies based on a kriging metamodel have already been proposed [see Bect et al., 2011, for a review]. Note that some of these strategies were initially designed to estimate the boundary of the excursion set (and not its volume) but, as these problems are quite close, we expect these criteria to have fairly good performances for the problem of estimating a probability of failure. The sampling criteria proposed by Ranjan et al. [2008], Bichon et al. [2008] and Echard et al. [2010]consist of heuristic modifications of the famous Expected Improvement criterion of Jones et al. [1998]. They compute a *pointwise* trade-off between predicted closeness to the threshold T, and

high prediction uncertainty. In contrast, Stepwise Uncertainty Reduction (SUR) strategies [Vazquez and Bect, 2009, Bect et al., 2011] rely on global measures of uncertainty about the excursion set $\Gamma^\star$ and take into account the important fact that sampling at a point x also brings useful information on the neighbourhood of x. Numerical experiments [reported by Bect et al., 2011] showed that SUR criteria widely outperform pointwise criteria in terms of quickly estimating the true volume of excursion $\alpha^\star$.

Perhaps the most natural SUR sampling criterion, for the problem of estimating a probability of failure, is the expected posterior variance of the volume of the *random* excursion set $\Gamma = \{x \in \mathbb{X} : \xi(x) \geq T\}$, where ξ is a Gaussian process modeling our current (prior) knowledge about f. This criterion has been considered impractical in previous publications [Vazquez and Bect, 2009, Bect et al., 2011], since its computation *seems* to require conditional simulations of the Gaussian process ξ, which are very expensive. Alternative SUR strategies were proposed instead: in short, they consist in defining a measure of uncertainty dedicated to the problem at hand, and then sampling sequentially at the location that will reduce the most, in expectation, this uncertainty.

An example of application of a SUR strategy is shown on Figure 1, on a real test case. Here a simulator f calculates whether a storage facility of plutonium powder presents risks of nuclear chain reactions or not, as a function of two variables, the mass and the concentration of Plutonium. A sequential sampling of this 2-dimensional "expensive" function, using a SUR strategy, manages to identify with very few evaluations the set of "dangerous" configurations.

Despite their very good performances in applications, SUR strategies still have important drawbacks. Computing the value of a SUR criterion at a single point $x_{n+1} \in \mathbb{X}$ is indeed very computer demanding since it relies on numerical integration. Besides, these strategies where designed to sample one point at a time while practionners often the have the capacity to run $r > 1$ simulations in parallel. This very high numerical complexity to simply compute the value of a sampling criterion at one point mainly explains why, despite their very good performances on numerical experiments, SUR strategies based on kriging are not yet widely used by practitioners for the problem of estimating a probability of failure.

In this paper, we bring new solutions to the issues mentioned above. We first introduce new *parallel* SUR sampling criteria and provide methods and algorithms allowing to run

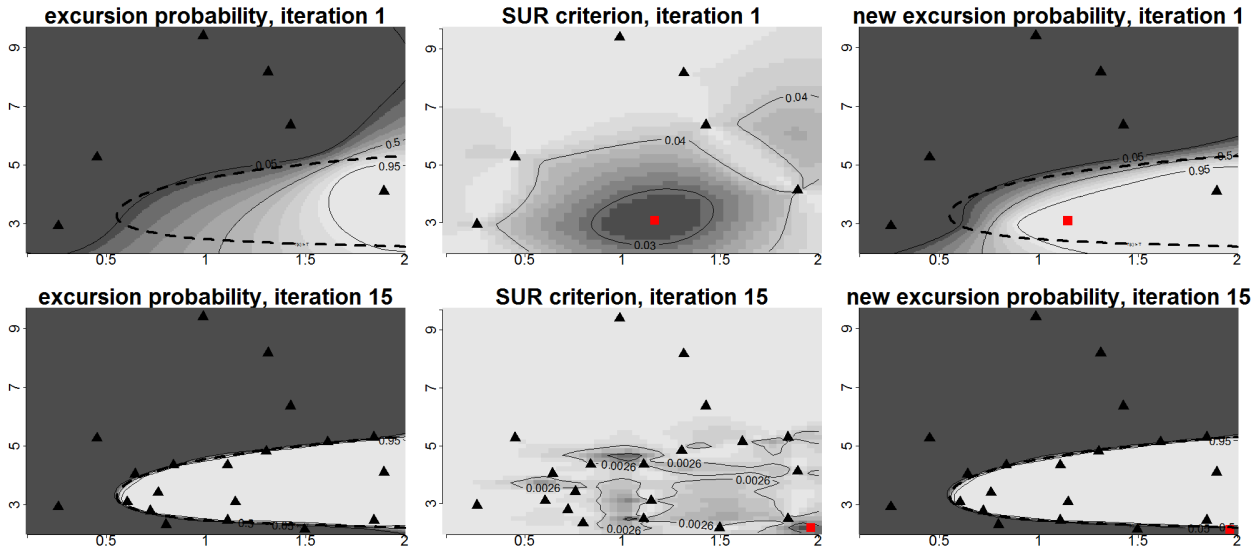

Figure 1: SUR strategy (first and last iteration) applied to a nuclear criticality safety simulator. The black triangles stand for the design points at the current iteration. The red square is the point sampled using the SUR criterion. Areas in black (resp. white) correspond to excursion probabilities near 0 (resp. 1). The dotted line indicates a fine approximation of the true but unknown excursion set's boundary.

them in a very reasonable time. In particular, we show that the unaffordable (one step look-ahead) optimal criterion presented in Bect et al. [2011] can be computed quickly, without simulating any Gaussian Process realization. Furthermore, we illustrate the use of parallel criteria in real-life applications, and investigate their performances on several test cases.

The paper is organised as follows: Section 2 introduces notations and gives two examples of SUR criteria (including the optimal onse-step-lookahead criterion) with their new parallel versions. The theoretical basis of our methods to quickly compute the criteria are detailed in Section 3 and our new algorithms are tested in Section 4 on different test cases, including a nuclear safety application. For the sake of brevity, basic notions about kriging and details about the choice of the integrations points are presented in appendix. In addition, detailed computational complexity calculations are provided as Supplementary Material.

2 Kriging-based Stepwise Uncertainty Reduction

A Stepwise Uncertainty Reduction (SUR) strategy aims at constructing a sequence $X_1, X_2, \ldots$ of evaluation points of f in such a way that the residual uncertainty about a quantity of interest given the information provided by the evaluation results becomes small. More precisely, SUR strategies are based on three main ideas. The first (Bayesian) idea is to consider f as a sample path of a random process ξ, which is assumed Gaussian for the sake of tractability. The second idea is to introduce a measure of the uncertainty about the quantity of interest conditioned on the σ-algebra $\mathcal{A}_n$ generated by $\{(X_i, \xi(X_i)),\ 1 \leq i \leq n\}$. We will denote by H_n such a measure of uncertainty, which is an $\mathcal{A}_n$-measurable random variable. The third idea is to choose evaluation points sequentially in order to minimize, at each step n, the *expected* value of the future uncertainty measure H_{n+1} with respect to the random outcomes of the new evaluation of ξ:

$$X_{n+1} \ = \ \underset{x_{n+1} \in \mathbb{X}}{\operatorname{argmin}} \, J_n(x_{n+1}) \tag{1}$$

where

$$J_n(x_{n+1}) := \mathsf{E}_n \left(H_{n+1} \,\middle|\, X_{n+1} = x_{n+1} \right) , \tag{2}$$

and $\mathsf{E}_n(\,\cdot\,)$ stands for the conditional expectation $\mathsf{E}(\,\cdot \mid \mathcal{A}_n)$.

Depending of the definition given to the measure of uncertainty, many sequential SUR strategies can be designed in order to infer any quantity of interest. For the question of estimating a probability of failure, two SUR strategies are presented in this section.

Example 1: criterion $J_n^{(\alpha)}$. Recall that we denote by Γ the random excursion set $\{x \in \mathbb{X} : \xi(x) \geq T\}$ and α its volume, $\alpha = \mathsf{P}_{\mathbb{X}}(\Gamma)$. The conditional variance $\mathrm{Var}_n\left(\alpha\right)$ of α is a natural choice for H_n to quantify the (residual) uncertainty about $\alpha^\star$ given $\mathcal{A}_n$. In the rest of the paper, we denote this uncertainty by $H_n^{(\alpha)}$. A possible SUR strategy to estimate $\alpha^\star$ would consist, at step n, in choosing as next evaluation point an optimizer of the criterion:

$$J_n^{(\alpha)}(x_{n+1}) := \mathsf{E}_n \left(Var_{n+1}(\alpha) \,\middle|\, X_{n+1} = x_{n+1} \right) \tag{3}$$

A quite natural parallel extension of this criterion is now introduced. The following criterion depends indeed on $r > 0$ points $(x_{n+1}, \ldots, x_{n+r}) \in \mathbb{X}^r$:

$$J_n^{(\alpha)}(x_{n+1}, \ldots, x_{n+r}) := \mathsf{E}_n \left(Var_{n+r}(\alpha) \,\middle|\, X_{n+1} = x_{n+1}, \ldots, X_{n+r} = x_{n+r} \right) \tag{4}$$

Note that the latter criterion, is considered intractable in Bect et al. [2011] for $r = 1$ because its computation has a very high numerical complexity (it requires the simulation of a large number of Gaussian Process realizations). We will see in the next sections that both parallel and non parallel versions of this criterion can be computed quickly and used in applications.

Example 2: criterion $J_n^{(\Gamma)}$. The excursion volume can be characterized by the random variable $\mathbb{1}_{\{\xi(x)>T\}}$. This random variable has conditional expectation:

$$p_n(x) := \mathsf{E}_n \mathbb{1}_{\{\xi(x)>T\}} = \mathsf{P}(\xi(x) > T|\mathcal{A}_n) = \Phi\left(\frac{m_n(x) - T}{s_n(x)}\right),$$

where $m_n(x)$ and $s_n(x)$ are the kriging mean and variance at point x at time n (see the Appendix for a brief reminder about kriging and the notations used throughout the paper), and Φ denotes the cumulative distribution function (c.d.f.) of the standard Gaussian distribution. The random variable $\mathbb{1}_{\{\xi(x)>T\}}$ has conditional variance $p_n(x)(1 - p_n(x))$, so that $\int_{\mathbb{X}} p_n(1 - p_n)\mathrm{d}\mathsf{P}_{\mathbb{X}}$ can serve as a measure of global uncertainty about $\alpha^\star$. We denote this uncertainty measure by $H_n^{(\Gamma)}$, and the corresponding SUR sampling criterion is

$$J_n^{(\Gamma)}(x_{n+1}) := \mathsf{E}_n\left(\int_{\mathbb{X}} p_{n+1}(1 - p_{n+1})\mathrm{d}\mathsf{P}_{\mathbb{X}} \;\middle|\; X_{n+1} = x_{n+1}\right). \tag{5}$$

This criterion was first introduced by Bect et al. [2011]. Again, a natural extension is the following new parallel criterion:

$$J_n^{(\Gamma)}(x_{n+1}, \ldots, x_{n+r}) = \mathsf{E}_n\left(\int_{\mathbb{X}} p_{n+r}(1 - p_{n+r})\mathrm{d}\mathsf{P}_{\mathbb{X}} \;\middle|\; X_{n+1} = x_{n+1}, \ldots, X_{n+r} = x_{n+r}\right). \tag{6}$$

In Bect et al. [2011], the numerical computation of $J_n^{(\Gamma)}$ in (6) is considered only for $r = 1$ and is based on quadrature formulas written as

$$J_n^{(\Gamma)}(x_{n+1}) \;\approx\; \frac{1}{M}\sum_{q=1}^{Q}\sum_{m=1}^{M} w^{(q)} v_{n+1}(x^{(m)}; x_{n+1}, y_{n+1}^{(q)}). \tag{7}$$

Q is the number of points used to approximate the conditional expectation with respect to the random outcome of the evaluation at x_{n+1}, which has a $\mathcal{N}(m_n(x_{n+1}), s_n^2(x_{n+1}))$ distribution. M is the number of points used to obtain a Monte-Carlo approximation of $H_{n+1}^{(\Gamma)}$. The $x^{(m)}$'s are i.i.d. according to $\mathsf{P}_{\mathbb{X}}$; $(y_{n+1}^{(1)}, \ldots, y_{n+1}^{(Q)})$ and $(w^{(1)}, \ldots, w^{(Q)})$ stand for the quadrature points and quadrature weights of the Gauss-Hermite quadrature. Here the computation of $v_{n+1}(x^{(m)}; x_{n+1}, y_{n+1}^{(q)})$ in (7) involves the calculation of the kriging mean and the kriging

variance at $x^{(m)}$ from the evaluations of ξ at $X_1, \ldots, X_n$ and x_{n+1}. It follows (See supplementary materialfor more detail about algorithmic complexities) that the computation of $J_n^{(\Gamma)}$ at one point has a $O(n^3 + Mn^2 + MQ)$ complexity. Since we need to evaluate $J_n^{(\Gamma)}$ several times to carry out the minimization in (1), the computational cost of this SUR sampling strategy implemented using (7) can be very large.

The problem becomes even more difficult for $r > 1$, which requires a higher value for Q. Indeed, when $r > 1$, we have to approximate a conditional expectation with respect to the random outcome of the Gaussian vector $(\xi(x_{n+1}), \ldots, \xi(x_{n+r}))^\top$, which requires a discretization of an integral over $\mathbb{R}^r$. As a consequence, the complexity to compute the parallel SUR criterion presented above is expected to rise quickly with r, which makes it impractical even for small r.

The next section brings useful properties allowing to circumvent these issues. In particular, new analytical formulas allow us to get rid of the cumbersome integral over $\mathbb{R}^r$ and make it possible to compute efficiently both parallel and non-parallel criteria.

3 Efficient calculation of parallel SUR criteria

In this section, we provide new expressions allowing to efficiently compute the two parallel SUR strategies introduced in the previous section.

3.1 Criterion $J_n^{(\Gamma)}$

As explained in the previous sections, the proposed parallel criterion $J_n^{(\Gamma)}$ is the conditional expectation given $\mathcal{A}_n$ of the future uncertainty $H_{n+r}^{(\Gamma)}$, assuming that r new points will be evaluated. Such a future uncertainty is an $\mathcal{A}_{n+r}$-measurable random variable, meaning that the computation of its conditional expectation given $\mathcal{A}_n$ requires to discretize an integral over $\mathbb{R}^r$. It turns out that the complexity for computing $J_n^{(\Gamma)}$ can be drastically reduced, using the new analytical expressions given below.

Proposition 1.

$$J_n^{(\Gamma)}(x_{n+1}, \ldots, x_{n+r}) = \int_{\mathbb{X}} \Phi_2\left(\begin{pmatrix} a(x) \\ -a(x) \end{pmatrix}, \begin{pmatrix} c(x) & 1-c(x) \\ 1-c(x) & c(x) \end{pmatrix}\right) \mathsf{P}_{\mathbb{X}}(\mathrm{d}x), \qquad (8)$$

where:

- $\Phi_2(.,M)$ *is the c.d.f. of the centered bivariate Gaussian with covariance matrix* M
- $a(x) := (m_n(x) - T)/s_{n+r}(x)$,
- $\mathbf{b}(x) := \frac{1}{s_{n+r}(x)}\boldsymbol{\Sigma}^{-1}(k_n(x, x_{n+1}), \ldots, k_n(x, x_{n+r})^\top$
- $c(x) := 1 + \mathbf{b}(x)^\top \boldsymbol{\Sigma}\mathbf{b}(x) = s_n^2(x)/s_{n+r}^2(x)$
- $\boldsymbol{\Sigma}$ *is the* $r \times r$ *covariance matrix of* $(\xi(x_{n+1}), \ldots, \xi(x_{n+r}))^\top$ *conditional on* $\mathcal{A}_n$.

Proof. First, an interchange of integral and expectation (Fubini-Tonelli theorem) delivers

$$J_n^{(\Gamma)}(x_{n+1}, \ldots, x_{n+r}) = \int_{\mathbb{X}} \mathsf{E}_n\left(p_{n+r}(x)(1 - p_{n+r}(x))\right) \mathsf{P}_{\mathbb{X}}(\mathrm{d}x), \tag{9}$$

where the conditioning on $X_{n+i} = x_{n+i}$'s is not explicitely reproduced, to alleviate notations. Now, using the kriging update formula (see, e.g., Barnes and Watson [1992], Gao et al. [1996], Emery [2009], as well as Chevalier and Ginsbourger [2012]), we obtain:

$$m_{n+r}(x) = m_n(x) + (k_n(x, x_{n+1}), \ldots, k_n(x, x_{n+r}))\boldsymbol{\Sigma}^{-1}\mathbf{y}_{\text{centered}}, \tag{10}$$

where $\mathbf{y}_{\text{centered}} := (\xi(x_{n+1}) - m_n(x_{n+1}), \ldots, \xi(x_{n+r}) - m_n(x_{n+r}))^\top$, so that

$$p_{n+r}(x) = \Phi\left(a(x) + \mathbf{b}(x)^\top \mathbf{y}_{\text{centered}}\right) \tag{11}$$

A plug-in of expression (10) in the integrand of expression (6) gives:

$$\mathsf{E}_n\left(p_{n+r}(x)(1 - p_{n+r}(x))\right) \;=\; \int_{\mathbb{R}^r} \Phi(a(x) \,+\, \mathbf{b}(x)^\top \mathbf{u})\Phi(-a(x) \,-\, \mathbf{b}(x)^\top \mathbf{u})\Psi(\mathbf{u})\mathrm{d}\mathbf{u} \tag{12}$$

where Ψ is the $\mathcal{N}(\mathbf{0}, \Sigma)$ density of $\mathbf{y}_{\text{centered}}$ knowing $\mathcal{A}_n$. By definition of Φ, we then get

$$\begin{aligned}\mathsf{E}_n\left(p_{n+r}(x)(1 - p_{n+r}(x))\right) &= P_n(N_1 < a(x) + \mathbf{b}(x)^\top \mathbf{y}_{\text{centered}}, N_2 < -a(x) - \mathbf{b}(x)^\top \mathbf{y}_{\text{centered}}) \\ &= P_n(N_1 - \mathbf{b}(x)^\top \mathbf{y}_{\text{centered}} < a(x), N_2 + \mathbf{b}(x)^\top \mathbf{y}_{\text{centered}} < -a(x)),\end{aligned}$$

where $(N_1, N_2)^T \sim \mathcal{N}(\mathbf{0}, I_2)$ independently of $\mathbf{y}_{\text{centered}}$. Finally, $N_1 - \mathbf{b}(x)^\top \mathbf{y}_{\text{centered}}$ and $N_2 + \mathbf{b}(x)^\top \mathbf{y}_{\text{centered}}$ form a Gaussian couple with componentwise variances equal to $c(x)$ and covariance $1 - c(x)$, so that the anounced result directly follows by integration over $\mathbb{X}$. □

Remark 1. *For the latter Proposition 1, we managed to get rid of an integral over* $\mathbb{R}^r$*. Moreover, the given formula is "exact" in the sense that we no longer have to compute an estimate (relying on quadrature points) of such integral over* $\mathbb{R}^r$*. Besides, the computation of* $J_n^{(\Gamma)}$ *is now available for* $r > 1$ *at a cost that is not quickly increasing with* r*. For* n *observations and* M *discretization points for the integral over* $\mathbb{X}$ *, the complexity to compute* $J_n^{(\Gamma)}$ *for one batch of* r *points is mainly of* $O(rMn)$ *if we assume that* $r << n << M$ *(which is often the case in practice) and that some quantities have been pre-computed (see algorithms in the Supplementary Material for more details). This means that the complexity is roughly linear in* r*, which ensures that batches with large values for* r *can be used in applications.*

Remark 2. *When the integral over* $\mathbb{X}$ *is discretized based on* M *integration points, the computation of the* $J_n^{(\Gamma)}$ *criterion requires to calculate the updated kriging variance* $s^2_{n+r}(x)$ *for each of the* M *points. The updated kriging variance can be efficiently calculated using a kriging variance update formula given and proven in Chevalier and Ginsbourger [2012].*

Remark 3. *By reducing equation* (6) *to equation* (8)*, we achieved to reduce the integral over* $\mathbb{R}^r$ *to an integral over* $\mathbb{R}^2$ *(*Φ_2*). Moreover, although calculating* Φ_2 *is not trivial, this bivariate integral is standard, and there exist very efficient numerical procedures to compute it. For instance, Genz [1992] wrote routines in Fortran77 which have been wrapped in many R Packages (e.g., mnormt, pbivnorm, mvtnorm, available on CRAN).*

3.2 Criterion $J_n^{(\alpha)}$

In the kriging framework and conditionally on $\mathcal{A}_n$, the conditional expectation of the volume of excursion α is given by $\widehat{\alpha} := \int_{\mathbb{X}} p_n d\mathsf{P}_{\mathbb{X}}$. As explained before, the conditional variance $\mathrm{Var}_n\left(\alpha\right)$ of α given $\mathcal{A}_n$ is a very natural choice to quantify the uncertainty about α but, even for $r = 1$, it was considered intractable so far. In fact, with the help of the kriging update formulas (See Eq. 10) and the calculation schemes introduced in the proof of Proposition 1, we will now show that this criterion can be expressed in a numerically tractable form, for both parallel and non-parallel versions.

Proposition 2.

$$J_n^{(\alpha)}(x_{n+1},\ldots,x_{n+r}) = \gamma_n - \int_{\mathbb{X}\times\mathbb{X}} \Phi_2\left(\begin{pmatrix} a(z_1) \\ a(z_2) \end{pmatrix}, \begin{pmatrix} c(z_1) & d(z_1,z_2) \\ d(z_1,z_2) & c(z_2) \end{pmatrix}\right) \mathsf{P}_{\mathbb{X}}(\mathrm{d}z_1)\mathsf{P}_{\mathbb{X}}(\mathrm{d}z_2), \tag{13}$$

where

- $\Phi_2, a, \mathbf{b}$, *and* $\boldsymbol{\Sigma}$ *are defined as in Proposition 1,*
- $d(z_1,z_2) := \mathbf{b}(z_1)^\top \boldsymbol{\Sigma} \mathbf{b}(z_2)$
- γ_n *is a constant, in the sense that it does not depend on* $(x_{n+1},\ldots,x_{n+r})$.

Proof. Neglecting again the conditioning on the $X_{n+i} = x_{n+i}$'s in the notations, we have:

$$\begin{aligned}
J_n^{(\alpha)}(x_{n+1},\ldots,x_{n+r}) &:= \mathsf{E}_n\left(Var_{n+r}(\alpha) \;\middle|\; X_{n+1} = x_{n+1},\ldots,X_{n+r} = x_{n+r}\right) \\
&= \mathsf{E}_n\left(\mathsf{E}_{n+r}\left(\int_{\mathbb{X}} (\mathbb{1}_{\{\xi(x)>T\}} - p_{n+r}(z))\mathsf{P}_{\mathbb{X}}(\mathrm{d}z)\right)^2\right) \\
&= \mathsf{E}_n\left(\mathsf{E}_{n+r}\iint_{\mathbb{X}\times\mathbb{X}} (\mathbb{1}_{\{\xi(z_1)>T\}} - p_{n+r}(z_1))(\mathbb{1}_{\{\xi(z_2)>T\}} - p_{n+r}(z_2))\mathsf{P}_{\mathbb{X}}(\mathrm{d}z_1)\mathsf{P}_{\mathbb{X}}(\mathrm{d}z_2)\right) \\
&= \mathsf{E}_n\left(\iint_{\mathbb{X}\times\mathbb{X}} \left(\mathsf{E}_{n+r}(\mathbb{1}_{\{\xi(z_1)>T\}}\mathbb{1}_{\{\xi(z_2)>T\}}) - p_{n+r}(z_1)p_{n+r}(z_2)\right)\mathsf{P}_{\mathbb{X}}(\mathrm{d}z_1)\mathsf{P}_{\mathbb{X}}(\mathrm{d}z_2)\right)
\end{aligned}$$

By applying the law of total expectation, we see that, for any $(z_1,z_2) \in \mathbb{X}^2$:

$$\mathsf{E}_n(\mathsf{E}_{n+r}(\mathbb{1}_{\{\xi(z_1)>T\}}\mathbb{1}_{\{\xi(z_2)>T\}})) = \mathsf{E}_n(\mathbb{1}_{\{\xi(z_1)>T\}}\mathbb{1}_{\{\xi(z_2)>T\}}) = P(\xi(z_1)>T,\xi(z_2)>T|\mathcal{A}_n)$$

Thus, this quantity does not depend on the choice of the r points $(x_{n+1},\ldots,x_{n+r})$. Writing $\gamma_n := \iint_{\mathbb{X}\times\mathbb{X}} P(\xi(z_1)>T,\xi(z_2)>T|\mathcal{A}_n)\mathsf{P}_{\mathbb{X}}(\mathrm{d}z_1)\mathsf{P}_{\mathbb{X}}(\mathrm{d}z_2)$, $J_n^{(\alpha)}$ simplifies to

$$J_n^{(\alpha)}(x_{n+1},\ldots,x_{n+r}) = \gamma_n - \iint_{\mathbb{X}\times\mathbb{X}} \mathsf{E}_n(p_{n+r}(z_1)p_{n+r}(z_2))\mathsf{P}_{\mathbb{X}}(\mathrm{d}z_1)\mathsf{P}_{\mathbb{X}}(\mathrm{d}z_2).$$

The end result is obtained using similar calculations as in the proof of Property 1.

□

Remark 4. *This new expression is very similar to the expression found in Proposition 1 and can be computed with the same complexity. However, in practice, the number of integrations points M has to be higher because the domain to be discretized is $\mathbb{X} \times \mathbb{X}$. In the examples of Section 4, we use importance sampling techniques to choose these M integration points. In Section 4.1, we empirically demonstrate that using M^2 integration points to compute the $J_n^{(\alpha)}$ criterion and using M points to compute $J_n^{(\Gamma)}$ yields comparable performances for estimating the true volume of excursion in the case where the unknown function is actually a Gaussian Process realization.*

4 Applications

In this section, we illustrate our sequential sampling strategies on several test cases. The examples include simulated realizations of two-dimensional Gaussian Processes, a two-dimensional nuclear safety case study and a six-dimensional test function.

4.1 Benchmark on simulated Gaussian Process realizations

The first objective of this section is to compare the *non parallel* versions of the $J_n^{(\Gamma)}$ and $J_n^{(\alpha)}$ criteria. The test functions are 200 independent realizations of a two-dimensional Gaussian Process (GP) indexed by $[0,1]^2$. The covariance parameters for the kriging models are fixed equal to the actual ones of the GP. Besides comparing the two criteria, we want to estimate the effect of numerical integration erros on the global performance of the SUR strategies. The criterion $J_n^{(\alpha)}$ requires to compute an integral over $\mathbb{X} \times \mathbb{X}$, so is it expected that the error will be higher than for the criterion $J_n^{(\Gamma)}$, which requires an integration over $\mathbb{X}$ only. Therefore, as a rule of thumb, we use M integration points for $J_n^{(\Gamma)}$ and M^2 for $J_n^{(\alpha)}$.

For each GP realization, we fix the threshold T in order to have a constant volume of excursion $\alpha^\star = 0.2$. The volumes are calculated using 1 000 reference points, so for each Gaussian Process realization, exactly 200 points are in the excursion set. The initial design consists of $n_0 = 12$ points using maximin Latin Hypercube Sampling (LHS), and a total of $n_1 = 40$ points are added to the design using either the $J_n^{(\Gamma)}$ criterion or the $J_n^{(\alpha)}$ criterion. For all realizations, the performance of both criteria are measured in term of the relative squared volume error $\mathrm{SE} := (\widehat{\alpha} - \alpha^\star)^2/\alpha^{\star 2}$, where $\widehat{\alpha}$ is the estimated volume (equal to the

average probability of excursion of the reference points).

Two strategies are considered for numerical integration: first, we use $M = 50$ and 100 integration points to compute $J_n^{(\Gamma)}$ obtained using a Sobol sequence; *note that the integration points are not bound to lie among the* 1 000 *reference points*. In that case, the M^2 points used to compute $J_n^{(\alpha)}$ correspond to a $M \times M$ grid. We also test the use of $M = 50$ points chosen using a specific instrumental distribution (and renewed at each iteration) versus $M^2 = 2500$ points over $\mathbb{X} \times \mathbb{X}$ chosen using some other distribution on $\mathbb{X} \times \mathbb{X}$. In this last case, the M^2 points are not on a grid. Further details about the choice of the integration points are given in the Appendix, section B.

Figure 2 draws the evolution of $\overline{SE}$, the average of the SE values over the 200 realizations, as a function of the number of observations. First, one can see that the number of integration points has a direct impact on the performance of both SUR strategy, since the experiments corresponding to $M = 50$ with quasi-Monte Carlo sampling –based here on a Sobol sequence– provide the worst results (bold curves with the MC legend on Figure 2).

Besides, the $J_n^{(\Gamma)}$ criterion with M integration points has roughly the same performance as the $J_n^{(\alpha)}$ criterion with M^2 integration points. This suggests that, in high dimension, the criterion $J_n^{(\Gamma)}$ should be chosen since it requires a significantly lower computational effort.

A third conclusion is that the use of importance sampling (with a well chosen instrumental distribution) has a significant impact on the performance of these strategies, especially after a high number of iterations. Indeed, as the algorithm progresses, the criterion becomes more difficult to calculate with a good accuracy as explained in Appendix B. In that case, a clever choice of the integration points has a crucial impact on the global performance of the strategy.

From this application we can conclude that the criterion $J_n^{(\Gamma)}$ roughly achieves the same performances as $J_n^{(\alpha)}$ at a lower computational cost. This is why, in the next applications, we will mostly focus our attention on the $J_n^{(\Gamma)}$ criterion and its parallel extension.

4.2 Nuclear safety test case

In this section, we illustrate a batch-sequential SUR strategy on an engineering problem, and provide an efficient strategy for optimizing $J_n^{(\Gamma)}$ when the batch size r is large.

A system involving fissile materials may produce a chain reaction based on neutrons,

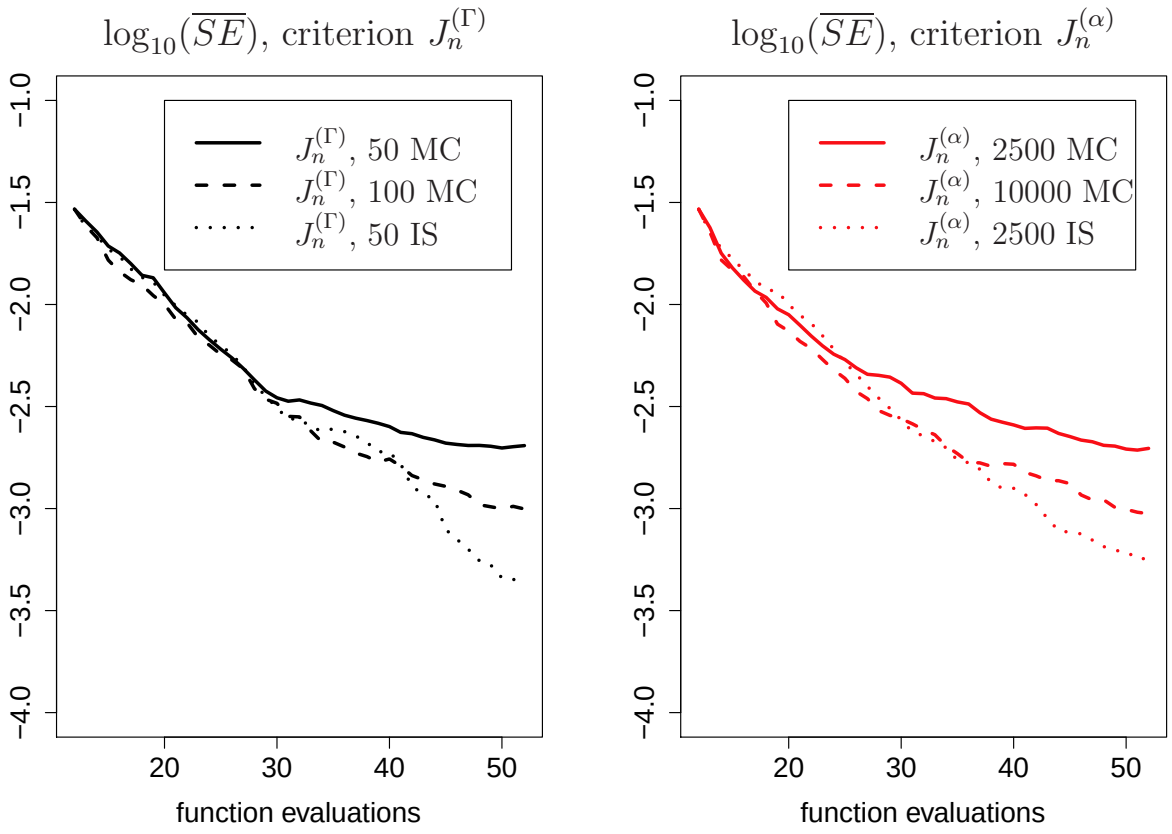

Figure 2: Performance (measured in term of mean squared relative error) of two SUR strategies based on the $J_n^{(\Gamma)}$ and $J_n^{(\alpha)}$ criteria, in function of the number of integration points and the method for choosing them (importance sampling or quasi Monte Carlo).

which are both a product and an initiator of fission reactions. Nuclear criticality safety assessment aims at avoiding "criticality accidents" (overproduction of neutrons) within the range of operational conditions. In order to check subcriticality of a system, the neutrons multiplication factor, k_{eff}, is estimated using a costly simulator. In our case, the system is a storage facility of plutonium powder, whose k_{eff} depends on two input parameters: the mass of plutonium (MassePu) and the concentration of plutonium (logConcPu). We aim at finding the set of "dangerous" configurations $\{(\text{MassePu}, \text{logConcPu}) : k_{\text{eff}}(\text{MassePu}, \text{logConcPu}) > T\}$, where T is threshold fixed at 0.95. The main issue lies in the high cost to evaluate k_{eff} at one single configuration. Many CPU are available to evaluate points in parallel, which means that our sampling strategy has to provide us, at each iteration, a number of points $r > 1$ at which to evaluate the simulator simultaneously.

In this section, we run our stepwise algorithms on this two-dimensional problem. The $J_n^{(\Gamma)}$ sampling criterion is used with an initial design of experiment of $n = 6$ points. The criterion is computed using $M = 600$ integration points renewed at each iteration, sampled from a specific instrumental distribution (See Appendix B for more detail). At each iteration, batches of $r = 4$ points are evaluated in parallel. Instead of performing the optimization of $J_n^{(\Gamma)}$ directly on the $\mathbb{X}^r$ space, we propose the following heuristic:

- find the point x_{n+1} optimizing the criterion for $r = 1$;
- while $k < r$, consider the points $x_{n+1}, \ldots, x_{n+k}$ as fixed, find x_{n+k+1} such that the set of points $(x_{n+1}, \ldots, x_{n+k}, x_{n+k+1})$ optimizes the criterion for $r = k+1$, and set $k \leftarrow k+1$.

This heuristic is of course sub-optimal but it allows us to replace the difficult optimization in $r \times d$ dimensions into r consecutive optimizations in dimension d. Note that this option allows using high values of r.

The evolution of the algorithm is shown on Figure 3. One can see that the excursion set is accurately identified in few (three) iterations of the parallel SUR strategy. After 18 evaluations (i.e. six initial evaluations plus three iterations, each providing a batch of $r = 4$ points), the excursion probability $p_n(x)$ does not depart much from the true function $\mathbb{1}_{x \in \Gamma^\star}$.

A key question here is to compare performances between the parallel criterion and the non-parallel one ($r = 1$). If the total number of evaluation of f is strictly identical, we generally expect the parallel criterion to have a worse performance than the non parallel one, in term of reducing the uncertainty $H_n^{(\Gamma)}$, because in the non parallel case the n^{th} evaluation point is chosen based on $n-1$ past evaluations, while in the parallel case it is chosen based on $n-r$ evaluations. In an ideal case, the uncertainty would decrease at the same rate, meaning that n/r iterations of the parallel criterion gives the same remaining uncertainty as n iterations of the non-parallel one (for n a multiple of r, say). Thus, if f is very expensive to evaluate, the time saving for the practitioner might be considerable.

Figure 4 gives the evolution of the uncertainty $H_n^{(\Gamma)}$ obtained during the uncertainty reduction with the parallel and the non-parallel criteria. It also shows $J_n^{(\Gamma)}(\mathbf{x_n^\star})$, which is the values of the $J_n^{(\Gamma)}$ criterion (with $r = 4$) at its current minimizer $\mathbf{x_n^\star}$. Note that, here, $\mathbf{x_n^\star}$ is a batch of r points. One can see on Figure 4 that at each iteration, $J_n^{(\Gamma)}(\mathbf{x_n^\star})$ is lower

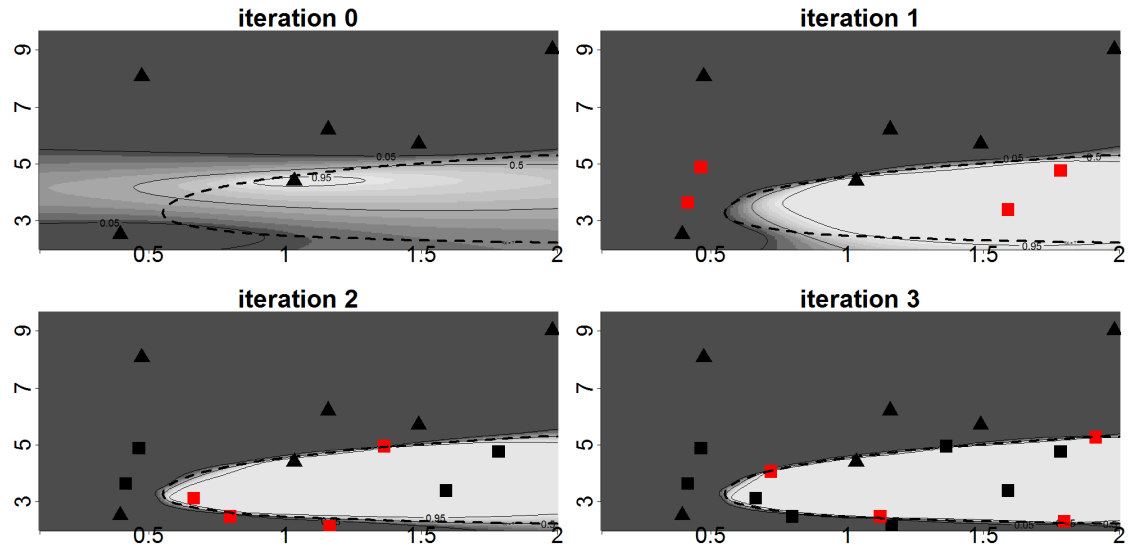

Figure 3: Plot of the function $p_n(x) = P_n(x \in \Gamma) = \Phi\left(\frac{m_n(x)-T}{s_n(x)}\right)$ after n evaluations of the simulator. The triangles are the six points of the initial DOE. The squares are the points sampled using the $J_n^{(\Gamma)}$ criterion. Areas in black correspond to $p_n(x) \approx 0$ and areas in white correspond to $p_n(x) \approx 1$. The dotted line indicates the true excursion set. The contour lines indicate the three level sets $p_n(x) = 0.05, 0.5$ and 0.95.

than $H_n^{(\Gamma)}$. This was to be expected, since $J_n^{(\Gamma)}(\mathbf{x_n^\star})$ is precisely the expectation of the future uncertainty $H_{n+r}^{(\Gamma)}$ if the r points $\mathbf{x_n^\star}$ are added to the design of experiments.

A striking conclusion to this section is that, here, the parallel criterion has the *same performance* as the non-parallel one, in term of reducing the uncertainty $H_n^{(\Gamma)}$, which corresponds to the ideal case mentioned before.

4.3 Six dimensional example

The Hartman6 function is a well known 6-dimensional function used in unconstrained global optimisation (Torn and Zilinskas [1989]). We test our SUR strategies on this function for two reasons. First we want to prove that the sampling strategy works (i.e., is able to recover the true volume of excursion) on higher dimensional functions, and provides better performances than a "basic" random sampling strategy. Second, we want to confirm the compared performances of the parallel $J_n^{(\Gamma)}$ criterion with the non parallel one observed

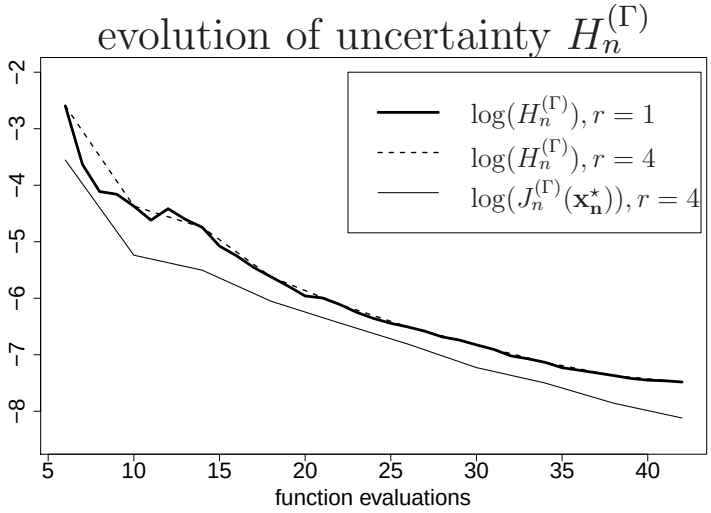

Figure 4: Evolution of $H_n^{(\Gamma)}$ during the sequential sampling strategy on the nuclear safety case study. The optimum $J_n^{(\Gamma)}(\mathbf{x_n^\star})$ of the $J_n^{(\Gamma)}$ criterion is usually lower than $H_n^{(\Gamma)}$ and corresponds to the expectation of $H_n^{(\Gamma)}$ at the next iteration.

earlier. We follow Jones et al. [1998] and perform the following change of variables:

$$y_H : \mathbf{x} \in \mathbb{R}^6 \mapsto -\log(-\text{Hartman6}(\mathbf{x})) \ .$$

We work with a threshold $T = 4$ and use two measures of performance:

- the uncertainty $H_n^{(\Gamma)}$,
- the relative squared volume error SE, defined in section 4.1.

All the performance calculations are done using 10 000 reference points (with a sobol sequence). In this example, $\alpha^\star = 0.2127$ which means that exactly 2127 of the 10 000 reference points are in the excursion set.

The results are averaged over 100 random initial design of experiments of 36 points (all of them being maximin Latin Hypercube Designs, generated with the R package *lhs* Carnell [2009]). The average uncertainty and the squared error are denoted by $\overline{H_n^{(\Gamma)}}$ and $\overline{SE}$ respectively.

In Figure 5, the parallel $J_n^{(\Gamma)}$ criterion, with $r = 4$, and the non parallel one are tested and compared. The criteria are calculated based on 250 integration points renewed at each iteration, using an instrumental distribution. A total of 80 new points are evaluated for

each instance of any of the two considered strategies. Two main conclusions can be obtained from Figure 5. First, as anticipated, the SUR strategies are sequentially reducing the relative real volume error faster than the basic random sampling strategy. From a relative error, in absolute value, of approximately 15% (with the initial design), we end up with a relative error (after having added 80 new observations) of approximately 3.3% on average. Second, the parallel strategy has again almost the same performance as the non parallel one. This means that we are, again, very close to the "ideal case" mentioned in the previous application.

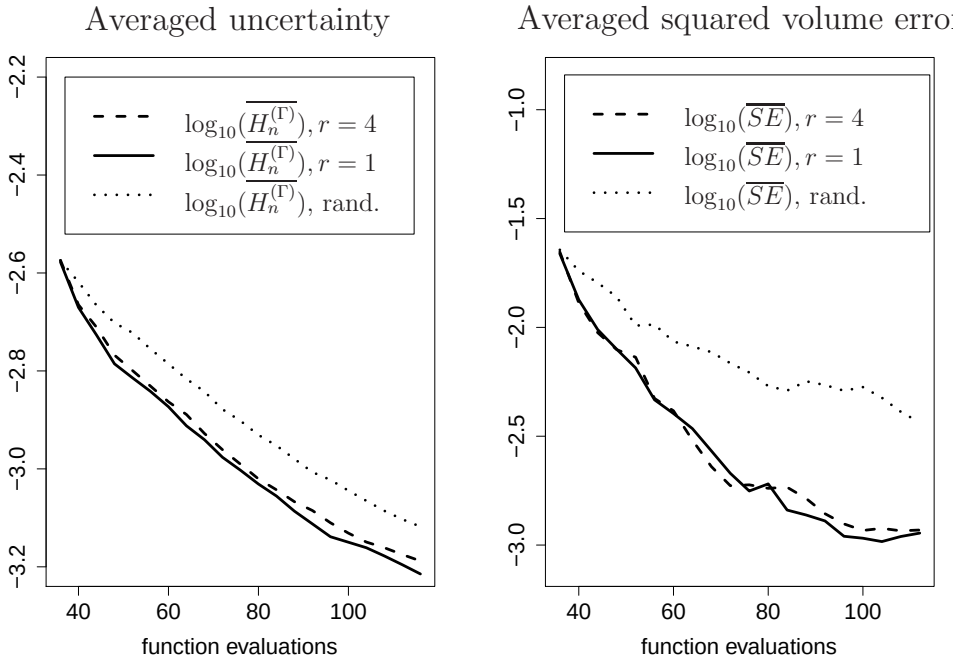

Figure 5: Evolution of $\overline{H_n^{(\Gamma)}}$ and of the averaged squared relative volume error during both the considered sequential and batch sequential algorithms, on the y_H function.

5 Conclusion and future work

In this paper, we presented algorithms for the computation of parallel and non-parallel Kriging-based infill sampling criteria. We showed that the use of the formulas introduced in this paper enables a practically sound implementation of the Stepwise Uncertainty Reduction (SUR) criteria proposed in Bect et al. [2011] and of batch-sequential versions of them. In particular, the complexity for computing a SUR criterion giving r points to evaluate si-

multaneously is "only" linear in r. Sampling criteria that were previously unaffordable in practical applications can now be used for parallel or non-parallel inversion. In addition, we showed that the proposed parallel SUR criteria do perform extremely well, in terms of quickly estimating a probability of failure. For low values of r, computing one iteration of the parallel criterion improves the accuracy of the estimation at almost the same pace than r sequential iterations of the non-parallel criterion. In applications on expensive-to-evaluate simulators, this allows a considerable time saving for the practitioners. Finally, a new version of the R package KrigInv (Chevalier et al. [2012]) is now available online, and allows using the presented sequential and batch-sequential strategies.

Further improvement are possible in this work and were mentioned in this paper. Sequential Monte Carlo methods might be an interesting alternative to compute a set of integration points that "evolves" from one iteration to another (See, e.g, Li et al.). Finally, from a more theoretical perspective, approaches directly based on random set notions (considering a "variance" of the excursion set itself, rather than the variance of the excursion volume) may provide elegant alternative sampling criteria for inversion and related problems.

References

R. J. Barnes and A. Watson. Efficient updating of kriging estimates and variances. *Mathematical geology*, 24(1):129–133, 1992.

M. J. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu. A framework for validation of computer models. *Technometrics*, 49(2):138–154, 2007.

J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vazquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, pages 1–21, 2011. DOI: 10.1007/s11222-011-9241-4.

B. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, and J. M. McFarland. Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA Journal*, 46 (10):2459–2468, 2008.

R. Carnell. *lhs: Latin Hypercube Samples*, 2009. R package version 0.5.

C. Chevalier and D. Ginsbourger. Corrected kriging update formulae for batch-sequential data assimilation. March 2012. URL `http://arxiv.org/abs/1203.6452`.

C. Chevalier, V. Picheny, and D. Ginsbourger. *KrigInv: Kriging-based Inversion for Deterministic and Noisy Computer Experiments*, 2012. URL `http://CRAN.R-project.org/package=KrigInv`. R package version 1.2.

V. Dubourg. *Métamodèles adaptatifs pour l'analyse de fiabilité et l'optimisation sous contrainte fiabiliste*. PhD thesis, IFMA, 2011.

B. Echard, N. Gayton, and M. Lemaire. Kriging-based Monte Carlo simulation to compute the probability of failure efficiently: AK-MCS method. In *6èmes Journées Nationales de Fiabilité, 24–26 mars, Toulouse, France*, 2010.

X. Emery. The kriging update equations and their application to the selection of neighboring data. *Computational Geosciences*, 13(1):211–219, 2009.

K.-T. Fang, R. Li, and A. Sudjianto. *Design and modeling for computer experiments*. Chapman & Hall / CRC Press, 2006.

A. I. J. Forrester, A. Sóbester, and A. J. Keane. *Engineering design via surrogate modelling: a practical guide*. Wiley, 2008.

H. Gao, J. Wang, and P. Zhao. The updated kriging variance and optimal sample design. *Mathematical Geology*, 28:295–313, 1996.

A. Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1:141–149, 1992.

D. R. Jones, M. Schonlau, and J. William. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

L. Li, J. Bect, and E. Vazquez. Bayesian Subset Simulation : a kriging-based subset simulation algorithm for the estimation of small probabilities of failure. Submitted to PSAM 11 and ESREL 2012. URL `http://publicationslist.org/jbect`.

P. Ranjan, D. Bingham, and G. Michailidis. Sequential experiment design for contour estimation from complex computer codes. *Technometrics*, 50(4):527–541, 2008.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

R. Rubinstein and D. Kroese. *Simulation and the Monte Carlo method*. Wiley, 2008.

J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.

T. J. Santner, B. J. Williams, and W. Notz. *The Design and Analysis of Computer Experiments*. Springer Verlag, 2003.

A. Torn and A. Zilinskas. *Global Optimization*. Springer Verlag, New York, 1989.

E. Vazquez and J. Bect. A sequential Bayesian algorithm to estimate a probability of failure. In *Proceedings of the 15th IFAC Symposium on System Identification, SYSID 2009 15th IFAC Symposium on System Identification, SYSID 2009*, Saint-Malo France, 2009.

APPENDIX

A Kriging mean and variance

In this section, we shall recall how to obtain the kriging mean and variance. Let $\xi \sim \mathrm{GP}(m, k)$ be a Gaussian random process with mean function $m(\cdot) = \mathsf{E}(\xi(\cdot))$ and covariance function $k(\cdot,\cdot) = \mathrm{cov}(\xi(\cdot), \xi(\cdot))$. We assume that the mean can be written as a linear combination

$$m(\cdot) \;=\; \sum_{i=1}^{l} \beta_i \, p_i(\cdot) \tag{14}$$

of basis functions $p_1, \ldots, p_l$ (very often, these are monomials), where $\beta_1, \ldots, \beta_l$ are *unknown* parameters. The covariance k is assumed to be a given symmetric *strictly* positive function.

The kriging predictor of ξ at a point $x \in \mathbb{X}$ from n observations $\xi(x_1), \ldots, \xi(x_n)$ is the best linear unbiased predictor (BLUP) of $\xi(x)$ from the observations, that we shall denote by

$$m_n(x) \;=\; \lambda(x; \underline{x}_n)^{\mathsf{T}} \begin{pmatrix} \xi(x_1) \\ \vdots \\ \xi(x_n) \end{pmatrix}. \tag{15}$$

The vector of *kriging weights* $\lambda(x; \underline{x}_n) \in \mathbb{R}^n$ can be obtained by solving the linear system

$$\underbrace{\begin{pmatrix} K(\underline{x}_n) & p(\underline{x}_n)^{\mathsf{T}} \\ p(\underline{x}_n) & \mathbf{0} \end{pmatrix}}_{:=\widetilde{K}(\underline{x}_n)} \cdot \underbrace{\begin{pmatrix} \lambda(x; \underline{x}_n) \\ \mu(x; \underline{x}_n) \end{pmatrix}}_{:=\widetilde{\lambda}(x;\underline{x}_n)} \;=\; \underbrace{\begin{pmatrix} k(x, \underline{x}_n) \\ p(x) \end{pmatrix}}_{:=\widetilde{k}(x,\underline{x}_n)} \tag{16}$$

where $K(\underline{x}_n)$ is the $n \times n$ covariance matrix of the random vector $(\xi(x_1), \ldots, \xi(x_n))^{\mathsf{T}}$, $p(\underline{x}_n)$ is the $l \times n$ matrix with general term $p_i(x_j)$, $k(x, \underline{x}_n)$ is the column vector with general term $k(x, x_i)$ and $\mu(x; \underline{x}_n)$ is a vector of l Lagrange multipliers associated to the unbiasedness constraint.

The covariance function of the prediction error

$$k_n(x, y) \;:=\; \mathsf{E}_n\left(\left(\xi(x) - m_n(x)\right) \left(\xi(y) - m_n(y)\right) \right), \tag{17}$$

also called *kriging covariance*, can be written using the notations of equation (16) as

$$k_n(x, y) \;=\; k(x, y) - \widetilde{k}(x, \underline{x}_n)^{\mathsf{T}} \, \widetilde{\lambda}(y; \underline{x}_n). \tag{18}$$

The conditional variance of the prediction error at a point $x \in \mathbb{X}$, also called *kriging variance*, will be denoted by $s_n^2(x) := k_n(x,x)$.

Remark 5. *To ensure that the conditional process $f|\mathcal{A}_n$ is still Gaussian when the mean function is of the form (14), with an unknown vector of parameters β, it is necessary to adopt a Bayesian approach and to use an (improper) uniform distribution over $\mathbb{R}^l$ as a prior distribution for β (see Bect et al. [2011], Section 2.3, Proposition 2, and the references therein for more detail).*

B Modelling choices

B.1 Choice of the integration points

The criteria studied in this paper involve the numerical computation of integrals over the domains $\mathbb{X}$ or $\mathbb{X} \times \mathbb{X}$. We showed in section 4 that the number of integration points has an important impact on the performance of the corresponding strategies. In this section we deal with the question of how these integration points are chosen.

The $J_n^{(\Gamma)}$ and $J_n^{(\alpha)}$ criteria may be written under the following general form:

$$J_n(x_{n+1},\ldots,x_{n+r}) = \text{Const} \pm \int_D \mathsf{E}_n\left(v_{n+r}(\mathbf{u}) \;\middle|\; X_{n+1}=x_{n+1},\ldots,X_{n+r}=x_{n+r}\right) Q(d\mathbf{u}) \quad (19)$$

More specifically, for the $J_n^{(\Gamma)}$ criterion, we have Const $= 0$, $D := \mathbb{X}$, $v_{n+r}(x) = p_{n+r}(x)(1 - p_{n+r}(x))$ and $Q := \mathsf{P}_{\mathbb{X}}$. For the $J_n^{(\alpha)}$ criterion, Const $= \gamma_n$, $D := \mathbb{X} \times \mathbb{X}$, $v_{n+r}(z_1,z_2) = p_{n+r}(z_1)p_{n+r}(z_2)$ and $Q := \mathsf{P}_{\mathbb{X}} \otimes \mathsf{P}_{\mathbb{X}}$. Note that in the remainder of this section, we omit the conditioning on $(X_{n+1} = x_{n+1},\ldots,X_{n+r} = x_{n+r})$ in order to simplify the notations.

For both criteria, a straightforward option to for calculating the integral over D would be to use Monte Carlo sampling with distribution Q. However, importance sampling techniques (see, e.g., Rubinstein and Kroese [2008], Robert and Casella [2004]) are a good choice for reducing the variance of the Monte Carlo error in case a suitable instrumental density is chosen. For the integral in equation 19, a natural choice for such an instrumental density is:

$$h(\mathbf{u}) \propto v_n(\mathbf{u})Q(\mathbf{u}) \; . \quad (20)$$

Indeed, if a point $\mathbf{u}$ has, in expectation, a high "future uncertainty" $v_{n+r}(\mathbf{u})$ it generally means that the current uncertainty at point $\mathbf{u}$, $v_n(\mathbf{u})$ is already high. For the $J_n^{(\Gamma)}$ criterion, we thus propose the following instrumental density:

$$h(\mathbf{u}) \propto p_n(\mathbf{u})(1 - p_n(\mathbf{u}))\mathsf{P}_{\mathbb{X}}(\mathbf{u}) \tag{21}$$

Similarly, for the $J_n^{(\alpha)}$ criterion, the proposed instrumental density is:

$$h(z_1, z_2) \propto p_n(z_1)p_n(z_2)\mathsf{P}_{\mathbb{X}} \otimes \mathsf{P}_{\mathbb{X}}(z_1, z_2) \tag{22}$$

Remark 6. *Using importance sampling techniques for the problem of estimating a probability of failure with a kriging metamodel has already been proposed and applied in Dubourg [2011].*

B.2 Sampling from our instrumental distribution

Sampling from the densities h defined above in Equations 21 and 22 is a difficult task. Figure 6 shows the value of the density $p_n(x)(1-p_n(x))\mathsf{P}_{\mathbb{X}}(x)$, which is (up to a multiplicative factor) the instrumental density proposed to compute the criterion $J_n^{(\Gamma)}$. When $n = 20$ evaluations of the simulator are available, one may remark (right graph) that the support of the instrumental density becomes very narrow. This issue complicates the use of standard MCMC algorithms to obtain a sample distributed according to the instrumental density. Sequential Monte Carlo methods may provide a nice solution to this issue but they have not been investigated and implemented yet in the KrigInv package (Chevalier et al. [2012]). Instead, we decided to use a simpler approximation. The idea consists in replacing the integral in 19 by the following estimator:

$$\int_D \mathsf{E}_n\left(v_{n+r}(\mathbf{u})\right) Q(d\mathbf{u}) \approx \frac{1}{N}\sum_{j=1}^{N} \mathsf{E}_n(v_{n+r}(\mathbf{u}_j)) \ , \tag{23}$$

where N is a large number and $\mathbf{u}_1, \ldots, \mathbf{u}_N$ is a i.i.d sample of N points with distribution Q. Rather than aiming at computing the integral of origin, we try to approximate this finite sum using a discrete instrumental density proportional to:

$$\sum_{j=1}^{N} v_n(\mathbf{u}_j)\delta_{\mathbf{u}_j}$$

Such new discrete instrumental density is easy to sample from.

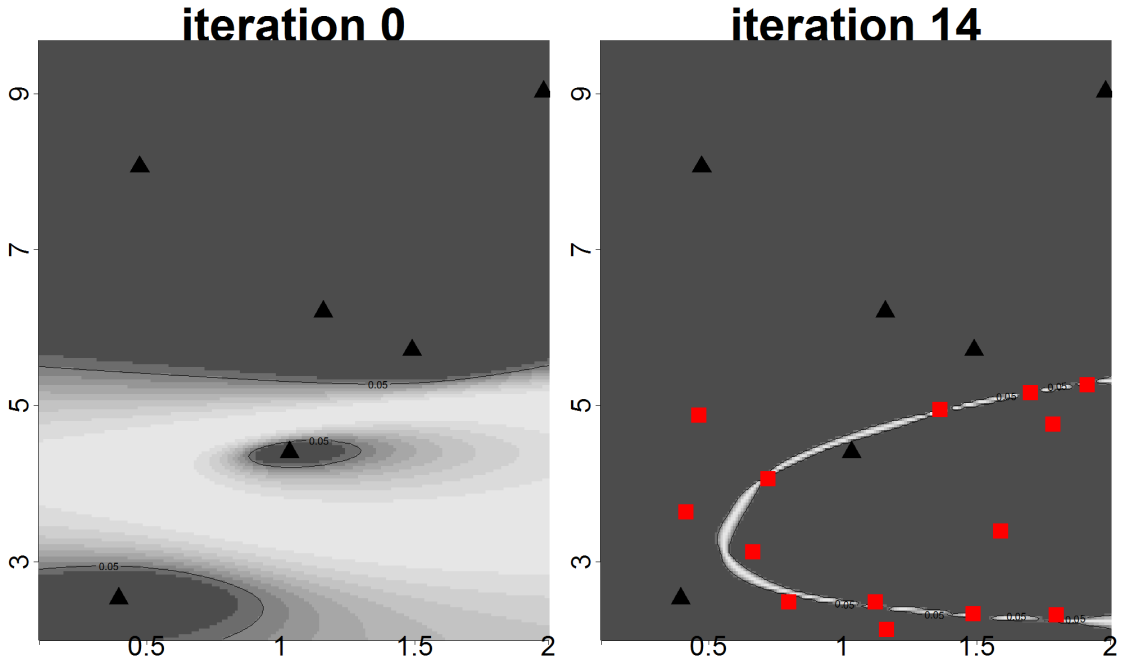

Figure 6: Plot of the function $p_n(x)(1-p_n(x))$ after n evaluations of the MORET code (left). The triangles are the six points of the initial DOE. Right: the squares are the points sampled using the $J_n^{(\Gamma)}$ criterion. Areas in black correspond to low uncertainty zones.

Of course, this method has important limitations. In particular the new "objective" quantity $\frac{1}{N}\sum_{j=1}^{N} \mathsf{E}_n(v_{n+r}(\mathbf{u}_j))$ can be completely different from $\int_D \mathsf{E}_n\left(v_{n+r}(\mathbf{u})\right) Q(d\mathbf{u})$ if all the N points from the initial large sample have an uncertainty v_n close to zero, or if N is not large enough. In essence, both N and the number of draws should tend to infinity in order for the estimator to converge to the true value of the integral. However, even if adapted MCMC approaches are likely to perform better in future implementations, this simple and easy option proposed here already provided a significantly improved calculation of the proposed SUR criteria compared to a standard quasi-Monte Carlo approach, as presented in section 4.

Appendix D: New SUR strategies for inversion using random set theory

Estimating and quantifying uncertainties on level sets using the Vorob'ev expectation and deviation with Gaussian process models

Clément Chevalier, David Ginsbourger, Julien Bect and Ilya Molchanov

Abstract Several methods based on Kriging have been recently proposed for calculating a probability of failure involving costly-to-evaluate functions. A closely related problem is to estimate the set of inputs leading to a response exceeding a given threshold. Now, estimating such level set – and not solely its volume – and quantifying uncertainties on it are not straightforward. Here we use notions from random set theory to obtain an estimate of the level set, together with a quantification of estimation uncertainty. We give explicit formulae in the Gaussian process set-up and provide a consistency result. We then illustrate how space-filling versus adaptive design strategies may sequentially reduce level set estimation uncertainty.

1 Introduction

Reliability studies increasingly rely on complex deterministic simulations. A problem that is often at stake is to identify, from a limited number of evaluations of f : $D \subset \mathbb{R}^d \mapsto \mathbb{R}$, the level set of "dangerous" configurations $\Gamma_f = \{x \in D : f(x) \geq T\}$, where T is a given threshold. In such context, it is commonplace to predict quantities of interest relying on a surrogate model for f. This approach was popularized in the design and analysis of computer experiments [12, 11, 7] . In the Kriging framework, several works have already been proposed for reliability problems (see, e.g., [2, 9, 10, 6] and the references therein). However, the quantity of interest is usually the *volume* of Γ_f, and none of the methods explicitly reconstruct Γ_f itself.

C. Chevalier, D. Ginsbourger, I. Molchanov
Institute of Mathematical Statistics and Actuarial Science, University of Bern, Sidlerstrasse 5 and Alpeneggstrasse 22, CH-3012 Bern, Switzerland e-mail: clement.chevalier@stat.unibe.ch; david.ginsbourger@stat.unibe.ch; ilya.molchanov@stat.unibe.ch

Julien Bect
Supélec Sciences des Systèmes - EA4454 (E3S), SUPELEC, Plateau de Moulon, 3 rue Joliot-Curie, 91192 Gif-sur-Yvette cedex, France, e-mail: julien.bect@supelec.fr

An illustrative example for this issue is given on Figure 1. A Kriging model is built from five evaluations of a $1d$ function (left plot). Three level set realisations (with $T = 0.8$) are obtained from Gaussian process (GP) conditional simulations. The focus here is on summarizing the conditional distribution of excursion sets using ad hoc notions of expectation and deviation from the theory of random sets. We address this issue using an approach based on the Vorob'ev expectation [1, 8].

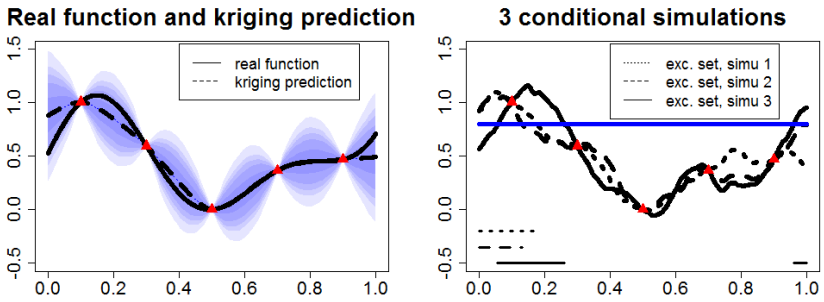

Fig. 1: Conditional simulations of level sets. Left: Kriging model obtained from five evaluations of a $1d$ function. Right: Three GP conditional simulations, leading to three different level sets. Here the threshold is fixed to $T = 0.8$.

In Section 2 we present the Vorob'ev expectation and deviation for a closed random set. In Section 3 we then give analytical expressions for these quantities in the GP framework. In addition we give consistency result regarding the convergence of the Vorob'ev expectation to the actual level set. To the best of our knowledge, this is the first Kriging-based approach focusing on the level set itself, and not solely its volume. Our results are illustrated on a test case in Section 4.

2 The Vorob'ev expectation and deviation in Random Set theory

Random variables are usually defined as measurable maps from a probability space $(\Omega, \mathscr{G}, \mathbb{P})$ to some measurable space, such as $(\mathbb{R}, \mathscr{B}(\mathbb{R}))$ or $(\mathbb{R}^d, \mathscr{B}(\mathbb{R}^d))$. However there has been a growing interest during the last decades for set-valued random elements, and in particular for *random closed sets* [8].

Definition 1. Let $\mathscr{F}$ be the family of all closed subsets of D. A map $X : \Omega \mapsto \mathscr{F}$ is called a *random closed set* if, for every compact set K in D,

$$\{\omega : X(\omega) \cap K \neq \emptyset\} \in \mathscr{G}. \tag{1}$$

As mentioned in [8], this definition basically means that for any compact K, one can always say when observing X if it hits K or not. Defining the expectation of a random set is far from being straightforward. Different candidate notions of expectation from the random set literature are documented in [8] (Chapter 2), with a major development on the *selection expectation*. Some alternative expectations mentioned in [8] include the *linearisation approach*, the *Vorob'ev expectation*, the *distance average*, the *Fréchet expectation*, and the *Doss and Herer expectations*.

In the present work we focus on the Vorob'ev expectation, which is based on the intuitive notion of coverage probability function. Given a random closed set X over a space D with σ-finite measure μ (say $D \subset \mathbb{R}^d$ and $\mu = \mathrm{Leb}_d$), then X is associated with a random field $(\mathbb{1}_X(x))_{x\in D}$. The coverage function is defined as the expectation of this binary random field:

Definition 2 (coverage function and α-quantiles of a random set). The function

$$p_X : x \in D \mapsto \mathbb{P}(x \in X) = \mathbb{E}(\mathbb{1}_X(x)) \tag{2}$$

is called the *coverage function* of X. The *α-quantiles* of X are the level sets of p_X,

$$Q_\alpha := \{x \in D : p_X(x) \geq \alpha\},\ \alpha \in (0,1]. \tag{3}$$

Note that in Equation 2, the expectation is taken with respect to the set X and not to the point x. In Figure 1 (right) we plotted three conditional realizations of the random set $X := \{x \in [0,1], \xi(x) \geq T\}$, where ξ is a GP. The α-quantile defined in Definition 2 can be seen as the set of points having a (conditional, in Figure 1) probability of belonging to X greater or equal than α. This definition is particularly useful here as, now, the so-called Vorob'ev expectation of the random set X will be defined as a "well-chosen" α-quantile of X.

Definition 3 (Vorob'ev expectation). Assuming $\mathbb{E}(\mu(X)) < \infty$, the *Vorob'ev expectation* of X is defined as the α^*-quantile of X, where α^* is determined from

$$\mathbb{E}(\mu(X)) = \mu(Q_{\alpha^*}) \tag{4}$$

if this equation has a solution, or in general, from the condition

$$\mu(Q_\beta) \leq \mathbb{E}(\mu(X)) \leq \mu(Q_{\alpha^*}) \text{ for all } \beta > \alpha^*. \tag{5}$$

Throughout this paper, an α^* satisfying the condition of Definition 3 will be referred to as a *Vorob'ev threshold*.

Property 1. For any measurable set M with $\mu(M) = \mathbb{E}(\mu(X))$, we have:

$$\mathbb{E}(\mu(Q_{\alpha^*} \Delta X)) \leq \mathbb{E}(\mu(M \Delta X)), \tag{6}$$

where $A \Delta B$ denotes the symmetric difference between two sets A and B. The quantity $\mathbb{E}(\mu(Q_{\alpha^*} \Delta X))$ is called *Vorob'ev deviation*.

The Vorob'ev expectation thus appears as a global minimizer of the deviation, among all closed sets with volume equal to the average volume of X. A proof can be found in [8], p. 193. In the next section, we will use these definitions and properties for our concrete problem, where the considered random set is a level set of a GP.

3 Conditional Vorob'ev expectation for level sets of a GP

In this section, we focus on the particular case where the random set (denoted by X in the previous section) is a level set

$$\Gamma := \{x \in D : \xi(x) \geq T\} \tag{7}$$

of a GP ξ above a fixed threshold $T \in \mathbb{R}$. Once n evaluation results $\mathscr{A}_n := ((x_1, \xi(x_1)), \ldots, (x_n, \xi(x_n)))$ are known, the main object of interest is then the conditional distribution of the level set Γ given $\mathscr{A}_n$. We propose to use the Vorob'ev expectation and deviation to capture and quantify the variability of the level set Γ conditionally on the available observations $\mathscr{A}_n$.

3.1 Conditional Vorob'ev expectation and deviation

In the simple Kriging GP set-up (see, e.g., [5]), we know the marginal conditional distributions of $\xi(x)|\mathscr{A}_n$:

$$\mathscr{L}(\xi(x)|\mathscr{A}_n) = \mathscr{N}(m_n(x), s_n^2(x)), \tag{8}$$

where $m_n(x) = \mathbb{E}(\xi(x)|\mathscr{A}_n)$ and $s_n^2(x) = Var(\xi(x)|\mathscr{A}_n)$ are respectively the *simple Kriging* mean and variance functions. The coverage probability function and any α-quantile of Γ can be straightforwardly calculated (given $\mathscr{A}_n$) as follows.

Property 2. **(i)** The coverage probability function of Γ has the following expression:

$$p_n(x) := \mathbb{P}(x \in \Gamma|\mathscr{A}_n) = \mathbb{P}(\xi(x) \geq T|\mathscr{A}_n) = \Phi\left(\frac{m_n(x) - T}{s_n(x)}\right), \tag{9}$$

where $\Phi(.)$ denotes the c.d.f. of the standard Gaussian distribution.
(ii) For any $\alpha \in (0,1]$, the α-quantile of Γ (conditional on $\mathscr{A}_n$) is

$$Q_{n,\alpha} = \{x \in D :\ m_n(x) - \Phi^{-1}(\alpha)s_n(x) \geq T\}. \tag{10}$$

(iii) For any $\alpha \in (0,1]$, the α-quantile of Γ can also be seen as the excursion set above T of the Kriging quantile with level $1-\alpha$.

From Property 2, one can see hat the Vorob'ev expectation is in fact the excursion set above T of a certain Kriging quantile. In applications, an adequate Vorob'ev

threshold value can be determined by tuning α to a level α_n^* such that $\mu(Q_{n,\alpha_n^*}) = \mathbb{E}(\mu(\Gamma)|\mathscr{A}_n) = \int_D p_n(x)\mu(\mathrm{dx})$. This can be done through a simple dichotomy.

Once the Vorob'ev expectation is calculated, the computation of the Vorob'ev deviation $\mathbb{E}(\mu(Q_{n,\alpha_n^*}\Delta\Gamma)|\mathscr{A}_n)$ does not require to simulate Γ. Indeed,

$$\begin{aligned}\mathbb{E}(\mu(Q_{n,\alpha_n^*}\Delta\Gamma)|\mathscr{A}_n) &= \mathbb{E}\left(\int_D (\mathbb{1}_{x\in Q_{n,\alpha_n^*},x\notin\Gamma} + \mathbb{1}_{x\notin Q_{n,\alpha_n^*},x\in\Gamma})\mu(\mathrm{dx})\Big|\mathscr{A}_\mathrm{n}\right)\\ &= \int_{Q_{n,\alpha_n^*}} \mathbb{E}(\mathbb{1}_{x\notin\Gamma}|\mathscr{A}_n)\mu(\mathrm{dx}) + \int_{\mathrm{Q^c_{n,\alpha_n^*}}} \mathbb{E}(\mathbb{1}_{\mathrm{x}\in\Gamma}|\mathscr{A}_\mathrm{n})\mu(\mathrm{dx})\\ &= \int_{Q_{n,\alpha_n^*}} (1-p_n(x))\mu(\mathrm{dx}) + \int_{\mathrm{Q^c_{n,\alpha_n^*}}} \mathrm{p_n(x)}\mu(\mathrm{dx}). \qquad (11)\end{aligned}$$

We will present in Section 4 an example of computation of Vorob'ev expectation and deviation. Before that, we give in the next subsection a consistency result for the case where observations of ξ progressively fill the space D.

3.2 Consistency result

Let us consider a (zero-mean, stationary) GP Z and a deterministic sequence of sampling points $x_1, x_2, \ldots$, such that $s_n^{\max} \triangleq \sup_{x\in D} s_n \to 0$ (this holds, e.g., for any space-filling sequence, assuming that the covariance function is merely continuous). We denote by α_n^* the Vorob'ev threshold selected for the first n sampling points, and by $\kappa_n = \Phi^{-1}(\alpha_n^*)$ and $Q_{n,\alpha_n^*} \subset D$ the corresponding quantile and Vorob'ev expectation. Our goal here is to prove that the Vorob'ev expectation is a consistent estimator of the true excursion set Γ, in the sense that $\mu\left(Q_{n,\alpha_n^*}\,\Delta\,\Gamma\right) \to 0$ for some appropriate convergence mode. To do so, we shall consider a slightly modified estimator Q_{n,α_n^*}, where the choice of the Vorob'ev threshold α_n^* is constrained in such a way that $|\kappa_n| \leq \kappa_n^{\max}$, for some deterministic sequence of positive constants $\kappa_n^{\max}$.

Proposition 1 *Assume that $\mu(D) < +\infty$ and $\kappa_n^{max} = O\left(\sqrt{|\log s_n^{max}|}\right)$. Then*

$$\mathbb{E}\left(\mu\left(Q_{n,\alpha_n^*}\,\Delta\,\Gamma\right)\right) \;=\; O\left(s_n^{max}\sqrt{|\log s_n^{max}|}\right).$$

As a consequence, $\mu\left(Q_{n,\alpha_n^*}\,\Delta\,\Gamma\right) \to 0$ for the convergence in mean.

Proof. The result has been proven in [13, 14] in the special case $\kappa_n^{\max} = 0$ (i.e., with $\alpha_n^* = 1/2$). We follow their proof very closely.

Let us first rewrite the probability of misclassification at $x \in D$ as

$$\mathbb{E}\left(\mathbb{1}_{Q_{n,\alpha_n^*}\,\Delta\,\Gamma}(x)\right) \;=\; \mathbb{E}\Big(\mathbb{1}_{p_n(x)\geq\alpha_n^*}\,(1-p_n(x)) + \mathbb{1}_{p_n(x)<\alpha_n^*}\,p_n(x)\Big), \qquad (12)$$

and consider the events

$$E_n^+ = \{m_n(x) \geq T + w_n(x)\}, \quad E_n^- = \{m_n(x) \geq T - w_n(x)\},$$

where $w_n(x)$ is a deterministic sequence that will be specified later. Let us assume that $\kappa_n^{\max} s_n(x) = O(w_n(x))$, uniformly in x. Then we have

$$|\kappa_n|\, s_n(x) \;\leq\; \kappa_n^{\max} s_n(x) \;\leq\; C w_n(x)$$

for some $C > 1$ (without loss of generality), and thus

$$\mathbb{1}_{p_n(x) \geq \alpha_n^*} \;=\; \mathbb{1}_{m_n(x) \geq T + \kappa_n s_n(x)} \;\leq\; \mathbb{1}_{|m_n(x) - T| \leq C w_n(x)} + \mathbb{1}_{E_n^+}.$$

As a consequence, noting that $\frac{m_n(x)-T}{s_n(x)} \geq \frac{w_n(x)}{s_n(x)}$ on E_n^+, we obtain:

$$\begin{aligned}
\mathbb{1}_{p_n(x) \geq \alpha_n^*} \,(1 - p_n(x)) \;&\leq\; \mathbb{1}_{|m_n(x) - T| \leq C w_n(x)} + \mathbb{1}_{E_n^+} \,(1 - p_n(x)) \\
&\leq\; \mathbb{1}_{|m_n(x) - T| \leq C w_n(x)} + \Psi\left(\frac{w_n(x)}{s_n(x)}\right),
\end{aligned}$$

where Ψ denotes the standard normal complementary cdf. Proceeding similarly with the second term in (12), it follows that

$$\mathbb{E}\left(\mathbb{1}_{Q_{n,\alpha_n^*}\Delta\Gamma}(x)\right) \;\leq\; 2\left[\Psi\left(\frac{w_n(x)}{s_n(x)}\right) + \mathbb{P}\left(|m_n(x) - T| \leq C w_n(x)\right)\right].$$

Using the tail inequality $\Psi(u) \leq \frac{1}{u\sqrt{2\pi}} \exp(-\frac{1}{2}u^2)$, and observing that $Var\,(m_n(x)) \geq s_0^2 - (s_n^{\max})^2 \geq s_0^2/4$ for n larger than some n_0 that does not depend on x, we have:

$$\mathbb{E}\left(\mathbb{1}_{Q_{n,\alpha_n^*}\Delta\Gamma}(x)\right) \;\leq\; \sqrt{\frac{2}{\pi}}\left[\frac{s_n(x)}{w_n(x)} \exp\left(-\frac{1}{2}\frac{w_n^2(x)}{s_n^2(x)}\right) + 4C\,\frac{w_n(x)}{s_0}\right]. \tag{13}$$

Finally, taking $w_n(x) = \sqrt{2}\, s_n(x) \sqrt{|\log s_n(x)|}$ as in [13], we have indeed $\kappa_n^{\max} s_n(x) = O(w_n(x))$ uniformly in x, and from (13) we deduce that

$$\mathbb{E}\left(\mathbb{1}_{Q_{n,\alpha_n^*}\Delta\Gamma}(x)\right) \;=\; O\left(s_n^{\max}\sqrt{|\log s_n^{\max}|}\right)$$

uniformly in x. The result follows by integrating with respect to μ over D.

4 Application to adaptive design for level set estimation

Here we present a 2-dimensional example on the notions and results previously detailed. We consider the Branin-Hoo function, with variables normalised so that the domain D is $[0,1]^2$. We multiply the function by a factor -1 and we are interested in the set $\{x \in D : f(x) \geq -10\}$. Figure 2 (top) gives the real level set and the coverage probability function obtained from $n = 10$ observations. The covariance

parameters of the Gaussian process used for Kriging are assumed to be known. The measure μ is the uniform measure on $D = [0,1]^2$ and the current Vorob'ev deviation is $\mathbb{E}(\mu(Q_{n,\alpha_n^*}\Delta\Gamma)|\mathscr{A}_n) \approx 0.148$. All the integrals are calculated using the KrigInv R package [4] with a Sobol' Quasi Monte Carlo sequence of 10000 points.

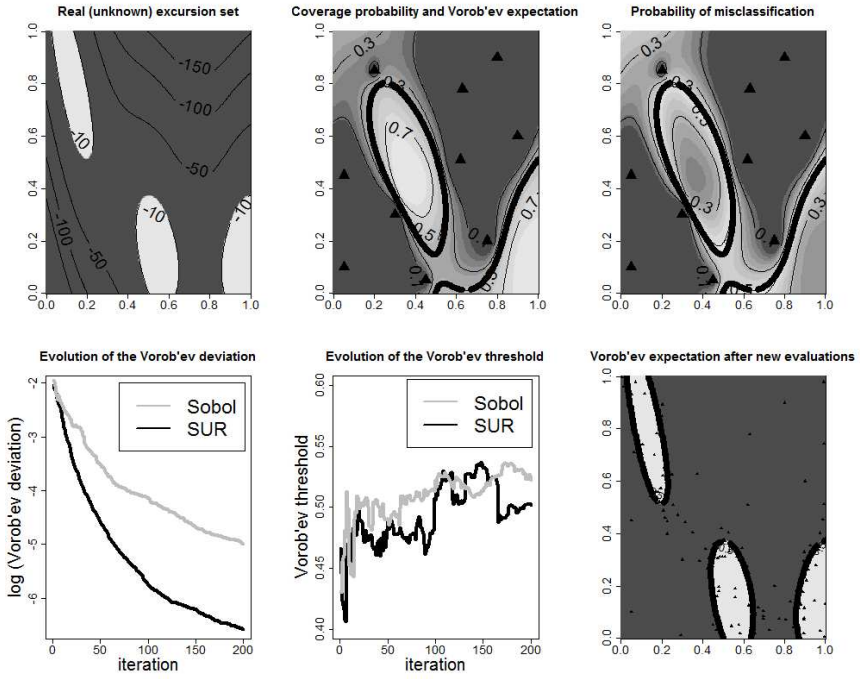

Fig. 2: Top left: Level set of a $2d$ function. Middle: Coverage probability function after 10 evaluations of f. Top right: $\mathbb{E}\left(\mathbb{1}_{Q_{n,\alpha_n^*}\Delta\Gamma}(\cdot)\right)$ function. Bottom left: Decrease of the Vorob'ev deviation when new points are added (2 strategies). Middle: Evolution of α^*. Bottom right: New Vorob'ev expectation (SUR strategy).

On Figure 2 (bottom plots) one can see the evolution of the Vorob'ev deviation and threshold when new points are added. Two different strategies are tested: a simple space filling strategy (with, again, the Sobol' sequence) and a so-called Stepwise Uncertainty Reduction (SUR) strategy, aiming at reducing the variance of $\mu(\Gamma)$ (see, [2], criterion $J_{4,n}^{\mathrm{SUR}}$, or [3] for more details). We observe that the SUR strategy manages to quickly reduce the Vorob'ev deviation (bottom right plot) and that the Vorob'ev expectation obtained after the new evaluations matches with the true level set. However, note that the consistency of the adaptive approach is not guaranteed by Proposition 1 as the latter only holds for a deterministic space filling sequence. Further research is needed to establish an extension of Proposition 1 to adaptive settings.

5 Conclusion

In this paper we proposed to use random set theory notions, the Vorob'ev expectation and deviation, to estimate and quantify uncertainties on a level set of a real-valued function. This approach has the originality of focusing on the set itself rather than solely on its volume. When the function is actually a GP realization, we proved that the Vorob'ev deviation converges to zero in infill asymptotics, under some mild conditions. However, the final example illustrates that a space-filling approach based on a Sobol' sequence may not be optimal for level set estimation, as it clearly was outperformed by an adaptive strategy dedicated to volume of excursion estimation. In future works, we may investigate sampling criteria and adaptive strategies dedicated to uncertainty reduction in the particular context of set estimation.

References

1. Baddeley, A., Molchanov, I.: Averaging of random sets based on their distance functions. Journal of Mathematical Imaging and Vision **8**(1), 79–92 (1998)
2. Bect, J., Ginsbourger, D., Li, L., Picheny, V., Vazquez, E.: Sequential design of computer experiments for the estimation of a probability of failure. Statistics and Computing **22(3)**, 773–793 (2012)
3. Chevalier, C., Bect, J., Ginsbourger, D., Vazquez, E., Picheny, V., Richet, Y.: Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set (2012). URL http://hal.inria.fr/hal-00641108/en
4. Chevalier, C., Picheny, V., Ginsbourger, D.: KrigInv: Kriging-based Inversion for Deterministic and Noisy Computer Experiments (2012). URL http://CRAN.R-project.org/package=KrigInv. R package version 1.3
5. Chilès, J.P., Delfiner, P.: Geostatistics: Modeling Spatial Uncertainty. Wiley, New York (1999)
6. Dubourg, V.: Métamodèles adaptatifs pour l'analyse de fiabilité et l'optimisation sous contrainte fiabiliste. Ph.D. thesis, IFMA (2011)
7. Forrester, A.I.J., Sóbester, A., Keane, A.J.: Engineering design via surrogate modelling: a practical guide. Wiley (2008)
8. Molchanov, I.: Theory of Random Sets. Springer (2005)
9. Picheny, V., Ginsbourger, D., Roustant, O., Haftka, R.T., Kim, N.H.: Adaptive designs of experiments for accurate approximation of target regions. Journal of Mechanical Design **132**(7) (2010)
10. Ranjan, P., Bingham, D., Michailidis, G.: Sequential experiment design for contour estimation from complex computer codes. Technometrics **50**(4), 527–541 (2008)
11. Rasmussen, C.R., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press (2006)
12. Santner, T.J., Williams, B.J., Notz, W.: The Design and Analysis of Computer Experiments. Springer Verlag (2003)
13. Vazquez, E., Piera-Martinez, M.: Estimation of the volume of an excursion set of a Gaussian process using intrinsic Kriging (2006). URL http://arxiv.org/abs/math/0611273
14. Vazquez, E., Piera-Martinez, M.: Estimation du volume des ensembles d'excursion d'un processus gaussien par krigeage intrinsèque. In: 39ème Journées de Statistiques. Angers France (2007)

Appendix E: A tutorial for the KrigInv R package

KrigInv: An efficient and user-friendly implementation of batch-sequential inversion strategies based on Kriging

Clément Chevalier, IMSV, University of Bern[1]
Victor Picheny, INRA Toulouse[2]
David Ginsbourger, IMSV, University of Bern[1]

Abstract: Several strategies relying on kriging have recently been proposed for adaptively estimating contour lines and excursion sets of functions under severely limited evaluation budget. The recently released R package **KrigInv**[3] is presented and offers a sound implementation of various sampling criteria for those kinds of inverse problems. **KrigInv** is based on the DiceKriging package, and thus benefits from a number of options concerning the underlying kriging models. Six implemented sampling criteria are detailed in a tutorial and illustrated with graphical examples. Different functionalities of **KrigInv** are gradually explained. Additionally, two recently proposed criteria for batch-sequential inversion are presented, enabling advanced users to distribute function evaluations in parallel on clusters or clouds of machines. Finally, auxiliary problems are discussed. These include the fine tuning of numerical integration and optimization procedures used within the computation and the optimization of the considered criteria.

Keywords: Computer experiments, Gaussian process modeling, Sequential design, Probability of failure, Contour line estimation, Excursion set, Active learning

[1]Clément Chevalier and David Ginsbourger, Institute of Mathematical Statistics and Actuarial Science, Alpeneggstrasse 22, CH-3012 Bern, Switzerland.
E-mail: {clement.chevalier}{ginsbourger}@stat.unibe.ch, Tel: +41 31 631 {57 83}{54 62}.
[2]Victor Picheny, Unité de Biométrie et Intelligence Artificielle, INRA, Auzeville, BP 52627 31326 Castanet Tolosan Cedex, France.
E-mail: Victor.Picheny@toulouse.inra.fr, Tel: +33 5 61 28 54 39.
[3]URL: http://cran.r-project.org/web/packages/KrigInv/index.html

1. Introduction

In many engineering fields, the use of *metamodeling* or *surrogate modeling* techniques has become commonplace for dealing efficiently with time-consuming high-fidelity simulations. These techniques consist of replacing the expensive model by a simpler one, based on a limited number of evaluations, in order to compute predictions and/or to guide an evaluation strategy of the simulator. The **KrigInv** R package, available on CRAN, was developed in this context. Its main goal is to propose evaluation strategies dedicated to inversion (as defined later), based on a *kriging* metamodel.

Mathematically, the expensive simulator (or typically an objective function built upon it) is considered here as a real-valued function f defined on a compact domain $\mathbb{X} \subset \mathbb{R}^d$, often assumed to be a hyper-rectangle. We assume further that:

- **No closed-form expression is available for** f. The objective function is seen as a "black-box" taking $\mathbf{x} \in \mathbb{X}$ as input and returning $f(\mathbf{x})$ without any other information, such as gradients.
- **The dimension of the input domain** $\mathbb{X}$ **is moderate**. $\mathbb{X}$ is typically a compact subset of $\mathbb{R}^d$ with d of the order of 10.
- **We have a small evaluation budget**. Evaluating f at any point $\mathbf{x}$ is assumed to be slow or expensive, so our problem needs to be solved in only a few evaluations of f: at most a few hundred, but, very often, much less.
- f **can be evaluated sequentially**. We usually dedicate a fraction of the budget for the initial design of experiments and then evaluate sequentially f at well-chosen points. The next point (or batch) to evaluate f at is chosen by optimizing a given *sampling criterion*.
- **Noisy simulators are handled**. Our methods work in the setting where we do not directly observe $f(\mathbf{x})$ but rather $f(\mathbf{x}) + \varepsilon$, where ε is a centered noise with **known** (or previously estimated and plugged-in) variance.

In the setting described above, metamodeling techniques have already proven to be efficient (see, e.g., Santner et al. (2003); Fang et al. (2006); Rasmussen and Williams (2006); Forrester et al. (2008); Gramacy and Lee (2008); Marrel et al. (2008)). From a set of n evaluation results $\{f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)\}$, an approximated response surface can be constructed, jointly with a measure of uncertainty at non-evaluated points, in order to guide a sequential sampling strategy of f. This idea has led to the famous Efficient Global Optimization (EGO) algorithm (Jones et al., 1998), where a kriging metamodel (Sacks et al., 1989; Stein, 1999; Cressie, 1993) and the *Expected Improvement* (EI) criterion are used to optimize an expensive function f. In the methods presented here, similar concepts are used, except that our final aim is not to find the optimum of f. **KrigInv** provides sequential sampling strategies aiming at solving the following inverse problems:

- Estimating the excursion set $\Gamma^* = \{\mathbf{x} \in \mathbb{X} : f(\mathbf{x}) \geq T\}$, where T is a fixed threshold.
- Estimating the volume of excursion: $\alpha^* := \mathbb{P}_{\mathbb{X}}(\Gamma^*)$, where $\mathbb{P}_{\mathbb{X}}$ is a given measure.
- Estimating the contour line $\mathcal{C}^* := \{\mathbf{x} \in \mathbb{X} : f(\mathbf{x}) = T\}$.

Note that the second problem is often encountered as a *probability of failure estimation* problem in the reliability literature (Bect et al., 2012), assuming that the input variables are random, with known distribution. The three problems described above are quite similar (a criterion dedicated to anyone of them is expected to perform fairly well on the others) and in this paper we group them under the term *inversion*.

Estimating a probability of failure is classically done through classical Monte Carlo sampling, or even refinements of Monte Carlo methods like subset sampling (Au and Beck, 2001) or cross-entropy methods (Rubinstein and Kroese, 2004). These methods are not adapted to our setting as they require too many evaluations of f. Some response surface methods make parametric approximations of f (see, e.g. Kim and Na (1997), Gayton et al. (2003)) or of the boundary of the excursion set $\{\mathbf{x} : f(\mathbf{x}) \geq T\}$ with the so-called First and Second Order Reliability Methods (FORM and SORM, see e.g. Zhao and Ono (1999)). Though they may provide an interesting alternative they are not considered in **KrigInv**. Our non-parametric approach relies on a *kriging metamodel* and on different sampling criteria available in the literature and described in Section 3.

An example of sequential inversion using kriging, widely developed in this paper, is provided in Figure 1. In this example, f is the Branin-Hoo function, i.e. a two dimensional function defined on $[0, 1]^2$, available in the DiceKriging package (Roustant et al., 2012). We fix a threshold $T = 80$. The real excursion set (assumed unknown) is represented together with the excursion probability function (as defined in Section 2) based on 12 function evaluations. Such excursion probability function is also represented once 10 additional well-chosen points have been evaluated.

The paper is organised as follows. Section 2 introduces the excursion probability function, which will be crucial for understanding the evaluation strategies. Section 3 presents the sampling criteria available in **KrigInv**, and Section 4 finally provides the user with advanced settings like the choice of the integration points for criteria involving numerical integration. An introduction to kriging and further details on the outputs of an inversion are described in appendix, for the sake of brevity.

2. Kriging and excursion probability

The goal of this section is to recall a few necessary basics and notations in Gaussian process modeling, and to illustrate the *excursion probability function*, onto which most kriging-based inversion methods are built. In Gaussian process modeling, we assume that f is a realization of a Gaussian random field ξ indexed by $\mathbb{X}$. Considering the distribution

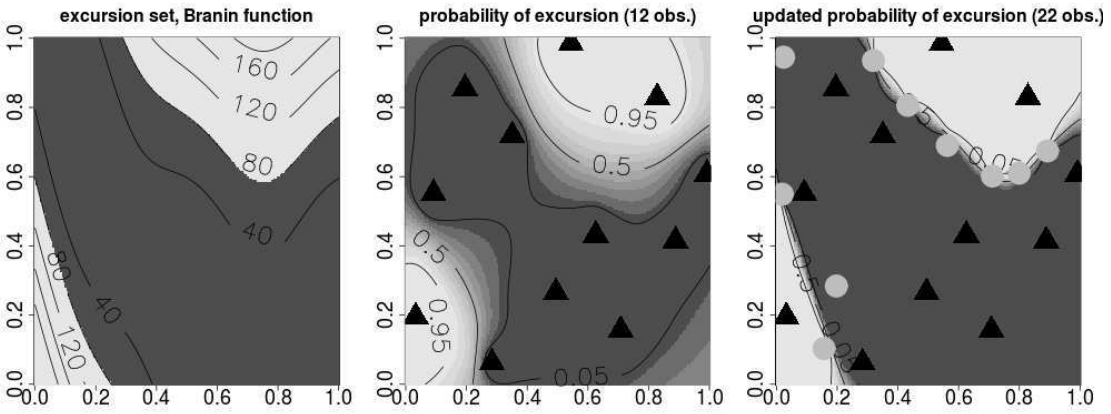

Figure 1: Left: excursion set of the Branin-Hoo function above $T = 80$. Middle and right: excursion probability function based on 12 and 22 evaluations of f.

of ξ knowing the event $\mathcal{A}_n := \{\xi(\mathbf{x}_1) = f(\mathbf{x}_1), \ldots, \xi(\mathbf{x}_n) = f(\mathbf{x}_n)\}$, the corresponding conditional expectation yields an approximated response surface. Such approximation is called *kriging mean* and is denoted by $m_n(\mathbf{x})$. At a non-evaluated point $\mathbf{x}$, the uncertainty on $\xi(\mathbf{x})$ is handled through the *kriging variance*, $s_n^2(\mathbf{x})$ and, as the conditional random field $\xi|\mathcal{A}_n$ is still Gaussian, we have that $\mathcal{L}(\xi(\mathbf{x})|\mathcal{A}_n) = \mathcal{N}(m_n(\mathbf{x}), s_n^2(\mathbf{x}))$. Kriging mean and variance can be calculated using the closed-form formulas (see, e.g. Chilès and Delfiner (1999)) implemented in the DiceKriging package (Roustant et al., 2012).

In the Gaussian process framework, finding $\Gamma^* = \{\mathbf{x} \in \mathbb{X} : f(\mathbf{x}) \geq T\}$ or $\alpha^* = \mathbb{P}_{\mathbb{X}}(\Gamma^*)$ becomes an estimation problem. Since $\forall \mathbf{x} \in \mathbb{X}$, $\xi(\mathbf{x}) \sim \mathcal{N}(m_n(\mathbf{x}), s_n^2(\mathbf{x}))$, the excursion probability,

$$p_n(\mathbf{x}) := P(\xi(\mathbf{x}) \geq T|\mathcal{A}_n),$$

can be calculated in closed form:

$$\begin{aligned} P(\xi(\mathbf{x}) \geq T|\mathcal{A}_n) &= P\left(\frac{\xi(\mathbf{x}) - m_n(\mathbf{x})}{s_n(\mathbf{x})} \geq \frac{T - m_n(\mathbf{x})}{s_n(\mathbf{x})} \,\middle|\, \mathcal{A}_n\right) \\ &= \Phi\left(\frac{m_n(\mathbf{x}) - T}{s_n(\mathbf{x})}\right), \end{aligned}$$

where $\Phi(\cdot)$ is the c.d.f. of the standard Gaussian distribution. The function $p_n(\cdot)$ plays a crucial role in solving the inversion problems described above (respectively, estimation of Γ^*, α^* and $\mathcal{C}^*$). Indeed the three following estimators can be used (Bect et al. (2012)):

$$\begin{aligned} \widehat{\Gamma} =& \{\mathbf{x} \in \mathbb{X} : p_n(\mathbf{x}) \geq 1/2\}, \\ \widehat{\alpha} =& \int_{\mathbb{X}} p_n(\mathbf{x}) \mathrm{d}\mathbf{x}, \\ \widehat{\mathcal{C}} =& \{\mathbf{x} \in \mathbb{X} : p_n(\mathbf{x}) = 1/2\} = \{\mathbf{x} \in \mathbb{X} : m_n(\mathbf{x}) = T\}. \end{aligned}$$

It hence appears that the function $p_n(\cdot)$ can be used as a classifier. In some sense, an "ideal" kriging model for inversion would be able to perfectly discriminate the excursion region, i.e. would give either $p_n(\mathbf{x}) = 0$ or 1 for all $\mathbf{x} \in \mathbb{X}$. We will see in the next section that this idea is extensively used to build the sequential sampling strategies available in **KrigInv**. Appendix B provides additional details on using a kriging model to obtain relevant informations for inversion.

3. Package structure and sampling criteria

This section gives an exhaustive description of the sampling criteria available in **KrigInv**. A sampling criterion aims at giving, at each iteration, a point or a batch of points for evaluation. More precisely, all **KrigInv** algorithms share the following general scheme:

1. Evaluate f at an initial set of design points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$.
2. Build a first metamodel based on $\{f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)\}$.
3. While the evaluation budget is not exhausted:
   - choose the next design point $\mathbf{x}_{n+1}$, or batch of design points $(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+r})$, by maximizing a given sampling criterion over $\mathbb{X}^r$,
   - evaluate f at the chosen design point(s),
   - update the metamodel.

In **KrigInv**, the available sampling criteria are called using the functions `EGI` (standing for *Efficient Global Inversion*) and `EGIparallel`. The criteria are separated into two categories: *pointwise criteria*, which involve only the conditional distribution of $\xi(\mathbf{x}_{n+1})|\mathcal{A}_n$, and *integral criteria*, which involve a numerical integration over the whole domain $\mathbb{X}$.

Note that the integral criteria covered here are well suited for delivering batches of $r > 1$ points, which is very useful in practice when several CPUs are available in parallel.

3.1. Pointwise sampling criteria

Three pointwise sampling criteria are available in **KrigInv**. These are criteria which depend on a point $\mathbf{x}_{n+1} \in \mathbb{X}$ and which evaluation mainly involves the computation of $m_n(\mathbf{x}_{n+1})$ and $s_n^2(\mathbf{x}_{n+1})$. For these three criteria, the sampled point is the point where the value of the criterion is **maximized**. Criteria proposed by Ranjan et al. (2008), Bichon et al. (2008) and Picheny et al. (2010) are reviewed in this section. The main idea with these three criteria (respectively *ranjan*, *bichon* and *tmse*) is that the interesting points $\mathbf{x}_{n+1} \in \mathbb{X}$ to evaluate f at are the points having both a high kriging variance and an excursion probability close to 1/2.

tmse criterion: The Targeted Mean Square Error criterion has been proposed by Picheny et al. (2010). The idea is to decrease the Mean Square Error (i.e. the kriging variance) at points where m_n is close to T. The criterion consists in the following quantity:

$$\text{tmse}(\mathbf{x}_{n+1}) = s_n^2(\mathbf{x}_{n+1}) \frac{1}{\sqrt{2\pi(s_n^2(\mathbf{x}_{n+1}) + \varepsilon^2)}} \exp\left(-\frac{1}{2} \left(\frac{m_n(\mathbf{x}_{n+1}) - T}{\sqrt{s_n^2(\mathbf{x}_{n+1}) + \varepsilon^2}} \right)^2 \right), \quad (1)$$

where $\varepsilon \geq 0$ is a parameter that tunes the bandwidth of a window of interest around the threshold T. In **KrigInv**, ε is equal to zero by default and can be modified using the argument *method.param* of the `EGI` function, detailed in Section 3.3. High values of ε make the criterion more exploratory, while low values concentrate the evaluations near the contour line of the kriging mean, $\{\mathbf{x} : m_n(\mathbf{x}) = T\}$. Unless the user wants to force exploration of sparse regions, we recommend to use the default value $\varepsilon = 0$.

ranjan and *bichon* criteria: These two criteria (see, Ranjan et al. (2008), Bichon et al. (2008)) depend on a parameter α which can also be set with the *method.param* argument. The default value for α is 1. Bect et al. (2012) provide the following common general expression for these two criteria:

$$\text{expr}(\mathbf{x}) = \mathbb{E}_n \left[\left((\alpha s_n(\mathbf{x}))^\delta - |T - \xi(\mathbf{x})|^\delta \right)_+ \right], \quad (2)$$

where $\mathbb{E}_n(\cdot) := \mathbb{E}(\cdot|\mathcal{A}_n)$, $(\cdot)_+ := \max(\cdot, 0)$ and δ is an additional parameter equal to 1 for the *bichon* criterion and 2 for the *ranjan* criterion. The goal is to sample a point $\mathbf{x}_{n+1}$ with a kriging mean close to T and a high kriging variance, so that the positive difference between $(\alpha s_n(\mathbf{x}_{n+1}))^\delta$ and $|T - \xi(\mathbf{x}_{n+1})|^\delta$ is maximal in expectation. The choice $\delta = 2$ (*ranjan* criterion) should favour sparse regions, of high kriging variance. However, in practice, these two criteria have very similar behaviours.
Calculations detailed in Bect et al. (2012) with $\delta = 1$ and 2 respectively lead to the following expressions, which, unlike Expression 2, can be computed easily from the kriging mean and variance:

$$\begin{aligned}
\text{bichon}(\mathbf{x}_{n+1}) &= s_n(\mathbf{x}_{n+1}) \left[\alpha(\Phi(t^+) - \Phi(t^-)) - t(2\Phi(t) - \Phi(t^+) - \Phi(t^-)) \right. \\
&\quad \left. -(2\phi(t) - \phi(t^+) - \phi(t^-)) \right], \\
\text{ranjan}(\mathbf{x}_{n+1}) &= s_n^2(\mathbf{x}_{n+1}) \left[(\alpha^2 - 1 - t^2)(\Phi(t^+) - \Phi(t^-)) - 2t(\phi(t^+) - \phi(t^-)) \right. \\
&\quad \left. +t^+\phi(t^+) - t^-\phi(t^-) \right],
\end{aligned}$$

where ϕ is the p.d.f. of the standard Gaussian distribution, $t := (m_n(\mathbf{x}_{n+1}) - T)/s_n(\mathbf{x}_{n+1})$, $t^+ := t + \alpha$ and $t^- := t - \alpha$.

Illustration: Figure 2 shows, on the $2d$ example introduced in Section 1, the excursion probability function $p_n(\cdot)$ after ten iterations based on these criteria. An example of code generating these plots is given in Section 3.3. The sets of points evaluated with

these criteria (circles) are rather similar and the criteria tend to evaluate points at the boundary of the domain $\mathbb{X}$. We recall that these criteria only depend on the marginal distribution at a point $\mathbf{x}_{n+1}$. Consequently, they do not take into account the fact that sampling at a point $\mathbf{x}_{n+1}$ may also bring useful information on the neighbourhood of $\mathbf{x}_{n+1}$. Recently, Bect et al. (2012) showed that these pointwise criteria are outperformed in applications by the integral sampling criteria presented in the next section. The price to pay for such more efficient criteria will be a higher computation time.

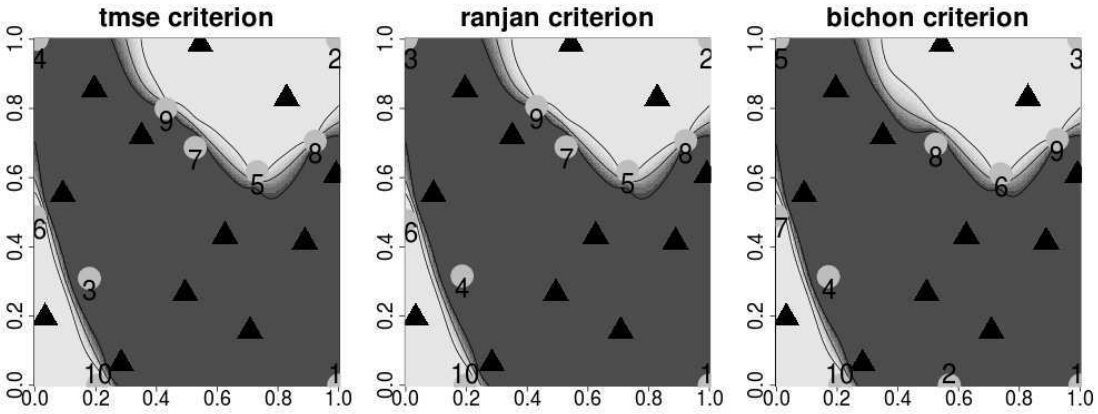

Figure 2: Excursion probability after ten iterations of the *tmse*, *ranjan* and *bichon* criterion. New evaluated points are represented by circles. The number associated with each point corresponds to the iteration at which the point is evaluated.

3.2. Integral sampling criteria

The term "integral criteria" refers to sampling criteria involving numerical integration over the design space $\mathbb{X}$. We give here details on the three integral criteria available in **KrigInv**. For the moment, two out of these three criteria can yield, at each iteration, a batch of r observations in lieu of a unique point. Note that the corresponding multi-point criteria have been shown to perform very well in applications (Chevalier et al., 2012).

All integral criteria presented here rely on the concept of *Stepwise Uncertainty Reduction* (SUR, see, e.g., Bect et al. (2012)). In short, the idea of SUR consists in defining an arbitrary measure of uncertainty given n observations $\mathcal{A}_n$, and seeking the point $\mathbf{x}_{n+1}$ (or batch $(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+r})$) such that evaluating $\xi(x_{n+1})$ (or $(\xi(\mathbf{x}_{n+1}), \ldots, \xi(\mathbf{x}_{n+r}))$) reduces the most (in expectation) this uncertainty. Consequently, different definitions for the term "uncertainty" will lead to different sampling criteria.

timse criterion: The Targeted Integrated Mean Square Error criterion (*timse*) was originally dedicated to contour line estimation (Picheny et al., 2010). It may easily be used as well for the problem of estimating the excursion set or its volume. The *timse* criterion

can be seen as the integral version of the *tmse* criterion. From the SUR point of view, the uncertainty measure underlying *timse* is the following:

$$\begin{aligned}\text{Uncertainty}^{\text{timse}} &:= \int_{\mathbb{X}} \text{tmse}(\mathbf{x})\mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{x}) \\ &= \int_{\mathbb{X}} s_n^2(\mathbf{x}) \frac{1}{\sqrt{2\pi(s_n^2(\mathbf{x})+\varepsilon^2)}} \exp\left(-\frac{1}{2}\left(\frac{m_n(\mathbf{x})-T}{\sqrt{s_n^2(\mathbf{x})+\varepsilon^2}}\right)^2\right)\mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{x}) \\ &= \int_{\mathbb{X}} s_n^2(\mathbf{x}) W_n(\mathbf{x})\mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{x}),\end{aligned}$$

where $W_n(\mathbf{x})$ is a weight function and ε is a parameter with the same role as in the *tmse* criterion. More details and interpretations of the weight function $W_n(\mathbf{x})$ are available in Picheny et al. (2010), Section 3.

The goal of the criterion is to sample a new point (or batch), in order to reduce $\text{Uncertainty}^{\text{timse}}$. It can be shown that the expectation of the future uncertainty when adding a batch of r points $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+r}$ has a simple closed form expression:

$$\text{timse}(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+r}) := \int_{\mathbb{X}} s_{n+r}^2(\mathbf{x}) W_n(\mathbf{x})\mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{x}), \tag{3}$$

where $s_{n+r}^2(\mathbf{x})$ is the kriging variance at point $\mathbf{x}$ once the batch $(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+r})$ has been added to the design of experiments. This variance (referred to as *updated kriging variance* here) does not depend on the unknown $\xi(\mathbf{x}_{n+1}), \ldots, \xi(\mathbf{x}_{n+r})$. Efficient formulas to compute $s_{n+1}^2(\mathbf{x})$ are given in Emery (2009). Also, Chevalier and Ginsbourger (2012) give formulas to quickly compute $s_{n+r}^2(\mathbf{x})$ when $r > 1$.

From Equation 3, we see that this criterion aims at reducing the kriging variance in "interesting" regions. These regions are selected using the weight function W_n. This weight is high when both m_n is close to T and s_n^2 is high. In Equation 3, the integral over $\mathbb{X}$ is discretized in M integration points. The choice of these integration points is an open option for the user of **KrigInv**. See Section 4.2 for more detail.

sur criterion: The *sur* criterion is introduced in Bect et al. (2012) and uses the following definition of the uncertainty:

$$\text{Uncertainty}^{\text{sur}} := \int_{\mathbb{X}} p_n(\mathbf{x})(1-p_n(\mathbf{x}))\mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{x}). \tag{4}$$

This definition can be obtained with non-heuristic considerations (see, Bect et al. (2012)) but, intuitively, the uncertainty is low when $p_n(\mathbf{x}) = 0$ or 1 over the whole domain, meaning that we are able to classify each point $x \in \mathbb{X}$. The sampling criterion associated with this definition of the uncertainty is:

$$\begin{aligned}\text{sur}(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+r}) := \mathbb{E}_n\Big(\int_{\mathbb{X}} p_{n+r}(\mathbf{x})(1-p_{n+r}(\mathbf{x}))\mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{x}) \\ \Big| \mathbf{X}_{n+1} = \mathbf{x}_{n+1}, \ldots, \mathbf{X}_{n+r} = \mathbf{x}_{n+r}\Big),\end{aligned} \tag{5}$$

where the condition $\mathbf{X}_{n+1} = \mathbf{x}_{n+1}, \ldots, \mathbf{X}_{n+r} = \mathbf{x}_{n+r}$ means that the next evaluation points are $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+r}$. Computing Equation 5 for a batch of points $(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+r})$ requires integrating both over $\mathbb{X}$ and over all possible responses $(\xi(\mathbf{x}_{n+1}), \ldots, \xi(\mathbf{x}_{n+r}))$, which may be quite impractical. In fact, Equation 5 can be simplified through the following closed-form expression (see: Chevalier et al. (2012) for a complete proof):

$$\mathrm{sur}(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+r}) = \int_{\mathbb{X}} \Phi_2 \left(\begin{pmatrix} a(\mathbf{x}) \\ -a(\mathbf{x}) \end{pmatrix}, \begin{pmatrix} c(\mathbf{x}) & 1 - c(\mathbf{x}) \\ 1 - c(\mathbf{x}) & c(\mathbf{x}) \end{pmatrix} \right) \mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{x}), \qquad (6)$$

where $\Phi_2(\cdot, \Sigma)$ is the c.d.f. of the centered bivariate Gaussian with covariance matrix Σ, $a(\mathbf{x}) := (m_n(\mathbf{x}) - T)/s_{n+r}(\mathbf{x})$ and $c(\mathbf{x}) := s_n^2(\mathbf{x})/s_{n+r}^2(\mathbf{x})$. With Equation 6, computing efficiently the multi-point criterion involves an update formula for kriging variances, and efficient numerical procedures to compute the bivariate Gaussian c.d.f. Φ_2 (Genz, 1992).

jn criterion: The *jn* criterion, introduced in Bect et al. (2012), is an optimal sampling criterion to estimate the excursion volume. The *jn* criterion can be naturally obtained by considering the volume α of the random excursion set $\Gamma = \{\mathbf{x} \in \mathbb{X} : \xi(\mathbf{x}) > T\}$. When n observations are available, the uncertainty is defined as follows:

$$\mathrm{Uncertainty}^{\mathrm{jn}} := \mathrm{Var}_n(\alpha), \qquad (7)$$

where $\mathrm{Var}_n(\cdot) := \mathrm{Var}(\cdot|\mathcal{A}_n)$. The associated sampling criterion is:

$$\mathrm{jn}(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+r}) := \mathbb{E}_n \left(\mathrm{Var}_{n+r}(\alpha) \middle| \mathbf{X}_{n+1} = \mathbf{x}_{n+1}, \ldots, \mathbf{X}_{n+r} = \mathbf{x}_{n+r} \right). \qquad (8)$$

The criterion samples a batch of points $(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+r})$ in order to decrease as much as possible (in expectation) the future variance of the excursion volume. An analytical expression allowing to compute efficiently Equation 8 is available in Chevalier et al. (2012) and is not reproduced here. Note that in the current version of **KrigInv** this criterion can be used with $r = 1$ only.

Evaluating *jn* involves an integral over $\mathbb{X} \times \mathbb{X}$, which is more difficult to compute than the integrals over $\mathbb{X}$ for the *timse* and *sur* criteria. Remarkably, *jn* often tends to be more space-filling than *sur*. Compared to cheaper criteria, *jn* performs especially well in cases where the excursion set has a complicated shape or is not connected. Indeed, as the criterion focuses on the excursion volume, it tends to evaluate points which are not too close to the boundary of the excursion set.

Illustration: Figure 3 shows the same plots as Figure 2 with the pointwise criteria replaced by the integral criteria *timse*, *sur* and *jn*, with $r = 1$. The plots are realized with the default parameters for the integration and optimization methods (see: Section 4). In this example, *sur* and *timse* show similar behaviours (the first three evaluations are almost the same), while *jn* tends to be slightly more exploratory. The *jn* criterion focuses on the excursion volume. So when a point which is "far" from the current estimated excursion region (the zone in white) has a non zero (say 0.01) excursion probability,

it may be picked by the *jn* criterion because the event $\{\xi(\mathbf{x}) > T\}$, even if it has a low probability, would change considerably the volume of excursion. The converse is also true: when a point with excursion probability of say 0.99 is far from the estimated boundary, it may be picked for the same reasons. Comparing the results to Figure 2, a

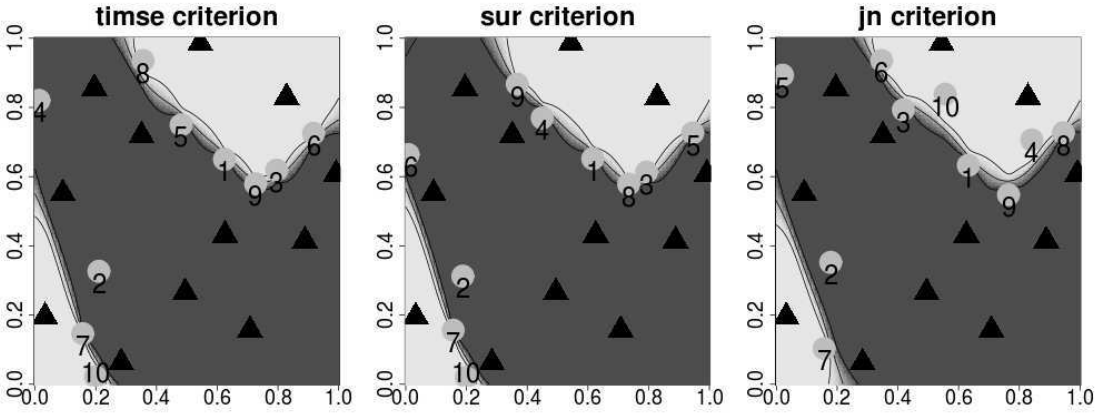

Figure 3: Excursion probability after ten iterations of the *timse*, *sur* and *jn* criterion. New evaluated points are represented by circles.

major difference is that no point is placed on the corners of the domain, while, for all the pointwise criteria, three corners were explored. The boundaries of $\mathbb{X}$ are often sampled by the pointwise criteria since those are regions with high kriging variance. However, sampling at the boundary of $\mathbb{X}$ does not contribute as efficiently to the reduction of uncertainty as sampling inside the design region. The unfortunate tendency of pointwise criteria to sample on the boundaries of $\mathbb{X}$ partially explains the better performances of integral criteria in general.

Figure 4 shows three iterations of the *timse* and *sur* parallel criteria, with $r = 4$ points evaluated at each iteration. As in the non-parallel case, *sur* and *timse* show rather similar behaviours. The parallel *sur* and *timse* criteria tend to spread points on the estimated boundary of the excursion set.

3.3. Using the criteria: the `EGI` and `EGIparallel` functions

`EGI` and `EGIparallel` are the two main functions of the **KrigInv** package. Users may choose to rely on these two functions only, as they are interfacing with all the other **KrigInv** functions. However, we export and provide a help file for all the coded functions, including the low level ones that are normally only called via other functions. `EGI` allows using criteria yielding one point per iteration while `EGIparallel` is dedicated to batch-sequential strategies. A general example of using `EGI` follows:

```
n <- 12 ; fun <- branin
design <- data.frame(optimumLHS(n,k=2)) #initial design (a LHS)
```

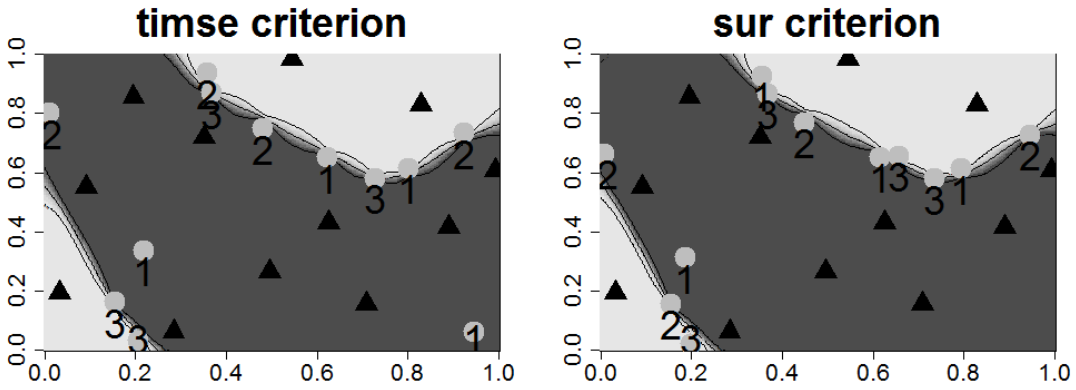

Figure 4: Excursion probability after three iterations based on the multi-point *timse* and *sur* criteria. Explored design points are represented by circles.

```
response <- fun(design)
model <- km(formula=~1, design = design, response = response,
            covtype="matern3_2")
T <- 80 ; iter <- 10
obj <- EGI(T=T,model=model,method="ranjan",fun=fun,
          iter=iter,lower=c(0,0),upper=c(1,1))
print_uncertainty_2d(model=obj$lastmodel,T=T,new.points=iter,
           main="10 iterations of the ranjan criterion")
```

`EGI` and `EGIparallel` take as argument a *km* object generated with the `km` function of the DiceKriging package. This choice ensures that the user has a basic knowledge of the DiceKriging package before using **KrigInv**. The other arguments relate to the problem at hand, that is: the target function *fun*, the lower and upper bounds of the hyper-rectangle $\mathbb{X}$ (*lower* and *upper*), the threshold T, the number of iterations *iter* (each of them bringing 1 or $r > 1$ observations), the sampling criterion *method*, and the number of points per batch *batchsize* (`EGIparallel` only). More advanced options, related to the evaluation and optimization of the criteria, are described in Section 4.

`EGI` returns a list with several fields. One of them (*lastmodel*) is the last kriging model obtained after all iterations. In our example we used this *km* object in a `print_uncertainty` call, displaying the excursion probability once the ten new points are evaluated. Other important outputs include the newly sampled points, *par*, and the value of f at these points, *value*.

The following example is a basic use of the `EGIparallel` function with three iterations. In this example, each iteration gives a batch of $r = 4$ points, for parallel evaluations of f. The output of this code is provided in Figure 4 (right).

```
#a km object called "model" is built as before (code not reproduced)
T <- 80 ; iter <- 3 ; r <- 4
obj <- EGIparallel(T=T,model=model,method="sur",fun=fun,
           iter=iter,batchsize=r,lower=c(0,0),upper=c(1,1))
print_uncertainty_2d(model=obj$lastmodel,T=T,new.points=iter*r,
           main="3 iterations of a parallel sampling criterion")
```

The two previous examples can be used with different sampling criteria by simply changing the argument *method* of `EGI` or `EGIparallel`.

Remark 1. *Other examples of use of* `EGIparallel` *(one in dimension* 6 *and one* $2d-$*test case in nuclear safety) are presented in Chevalier et al. (2012).*

3.4. Elements of computational effort required by the strategies

In this section, we provide some elements related to the computational time required to run the different sampling strategies. In general, the computational effort grows rapidly with the dimension d of the input space $\mathbb{X}$, because of three main effects.

- With kriging, the number of observations n often grows with the dimension, in order to learn the covariance function and ensure a reasonable space filling. The cost therefore grows accordingly because of the inversion of a $n \times n$ matrix required to compute kriging means and variances.
- The optimization in dimension d of any sampling criterion is more difficult and requires evaluating the criterion at more locations.
- When an integral criterion is used, the number of integration points required to compute the criterion with a good accuracy is higher.

Note that the computation time of all criteria described above is also marginally impacted by d through the higher computing cost of the covariance function, e.g., when a separable covariance function is chosen.

In Table 1, we show indicative computational time for evaluating the *ranjan* and *sur* criteria (with $r = 1$), and the time required to maximize them over $\mathbb{X}$. Three problems, respectively in dimension two, six and twenty are considered. Default parameters are used for optimization and integration. All models are based on standard Latin Hypercube Sampling (LHS) designs. The number of design points for each problem is arbitrary and quickly increases with the dimension. The 6D function is the classical benchmark function *hartman* (Dixon and Szegö (1978)); the 20D function is the spherical function $-\sum_{i=1}^{20}(x_i - 1/2)^2$. Note that the function themselves do not have any impact on the computation time of the criteria. Only the dimension does. Times are given for a workstation with a 2.53GHz CPU and 3GB of RAM.

Table 1: Computation time (in seconds) required to evaluate and optimize pointwise and integral criteria, on different test problems.

	ranjan	ranjan maximization	sur	sur maximization
Branin (2D), $n = 12$ pts	< 0.001	0.65	0.0024	1.4
6D function, $n = 60$ pts	< 0.001	4.1	0.005	10.4
20D function, $n = 400$ pts	0.008	82	0.025	133

On these examples, the computational effort increases quickly with dimension for both criteria. This is due to the higher number of observations and also to the higher number of points tested by the *genoud* optimizer (approximately 600, 3000 and 6000, respectively, for the three problems). We see here that the practical interest of the methods implemented in **KrigInv** depends on the time required to obtain design points. Since it takes approximately two minutes in 20D to choose such a point, the use of the *sur* criterion only makes sense if the evaluation time of f is several times slower.

Note that we recommend to use these algorithms only for dimensions $d \leq 20$, even though this limit is of course indicative. Indeed, while problem-dependent, the number of observations n needed to accurately identify the excursion set may increase rapidly with the dimension. This makes the use of kriging impractical because of the $n \times n$ matrix inversion used in kriging which limits n to a few hundreds, or thousands at most.

4. Optimizing the performances of the sampling strategies with advanced options

This section describes the options available to the user for two major sub-problems of the inversion problems tackled here. First, our sampling strategies usually require to optimize a sampling criterion at each iteration. The question of choosing the optimization method arises naturally. Second, for the criteria involving numerical integration, the questions of the number and the choice of integration points are detailed.

4.1. Optimization of the sampling criteria

For a one-point criterion, finding $\mathbf{x}^* \in \mathbb{X}$ maximizing or minimizing the criterion amounts to performing an optimization in dimension d. The options for such optimization are detailed in Section 4.1.1. For a multi-point criterion, finding the optimal batch of r points requires an optimization in dimension rd and can be impractical for high r or high d. In that case, a heuristic optimization strategy consisting in r sequential optimizations in dimension d is proposed and explained in Section 4.1.2.

4.1.1. One-point criteria: discrete or continuous optimization

The *optimcontrol* argument of the `EGI` or `EGIparallel` functions allows to tune the optimization of the selected sampling criterion. *optimcontrol* is a list with several fields. The field *method* has two possible values: "discrete" for an optimization over a discrete set of points or "genoud" (default) for a continuous optimization with a genetic algorithm (Mebane and Sekhon, 2011).

When *method* is set to "discrete", the user can manually set the field *optim.points* to indicate which points will be evaluated. The new observation is chosen as the best point over the discrete set. This may be useful if the user wants to optimize the criterion over a discrete grid in dimension d (for small d) or if the user has a guess on the location of the optimum. If *optim.points* is not set, $100d$ points are independently chosen at random, with a uniform distribution. Alternatively, the `genoud` algorithm (recommended option) optimizes a function by building generations of points spread on the domain $\mathbb{X}$, selecting the ones with the best f values and mutating them in order to have a new generation of points to evaluate. The user has the possibility to tune the parameters of the `genoud` algorithm, including the fields *pop.size* (default: $50d$) or *max.generation* (default: $10d$) which are respectively the number of points in each generation and the maximum number of generations. The maximum number criterion evaluations is $pop.size \times max.generation$.

4.1.2. Parallel criteria: standard or heuristic optimization

The optimization of the multi-point sampling criteria is not trivial. Instead of searching for an optimal point $\mathbf{x}_{n+1} \in \mathbb{X}$, multi-point sampling criteria are looking for an optimal **batch** of r points $(\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+r}) \in \mathbb{X}^r$. This optimization problem of dimension rd and can be very challenging.

In **KrigInv**, the user can choose between two optimization scenarios using the field *optim.option* of the *optimcontrol* list:

- Standard optimization in dimension rd, $optim.option = 1$: the optimizer works directly in dimension rd to find the optimal batch of r points.
- Heuristic optimization strategy (default), $optim.option = 2$: this option applies the following heuristic optimization strategy. First, find the point $\mathbf{x}_{n+1}$ optimizing the criterion for $r = 1$. Then, consider $\mathbf{x}_{n+1}$ as fixed and find a point $\mathbf{x}_{n+2}$ optimizing the criterion for $r = 2$. Iterate this procedure r times to finally obtain the batch of points $\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+r}$. Though this heuristic is clearly sub-optimal in theory, it often outperforms the standard optimization when the dimension rd becomes difficult to handle for the optimizer.

Similarly to the one-point case, the user can choose between continous and discrete optimization. When continuous optimization (with `genoud`) is selected, the user can choose between the standard and heuristic scenarios. For discrete optimizations, only

the heuristic strategy is applied, as combinatorial explosion prevents from optimizing over all the combinations of r points among the discrete set of points.

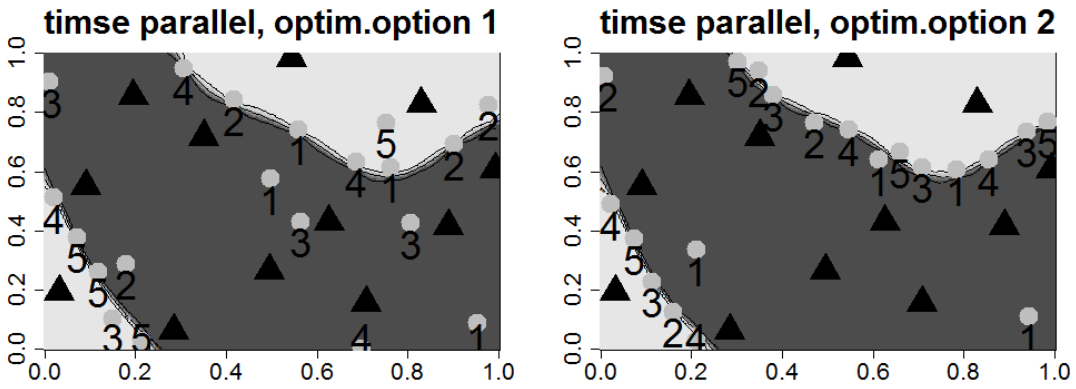

Figure 5: Excursion probability after five iterations of the multi-point *timse* criterion, with two scenarios for the optimization. New evaluated points are represented by circles.

In Figure 5, we perform five iterations of the multi-point *timse* criterion with $r = 4$. Here, we observe that brute force optimization (left) yields less satisfying results than the heuristic strategy (right). Indeed, eight points are located far from the boundary, which correspond to either poor local optima or premature termination of the optimization. On the contrary (right), all the points but three are well spread on the estimated boundary when the heuristic strategy is chosen.

To conclude this section, we would recommend to use either the heuristic optimization strategy (default) or, if the user wants an optimization in dimension rd, to increase the value of *pop.size* to at least $100rd$ (if affordable numerically).

4.2. Importance sampling for numerical integration

4.2.1. Integration options

The computation of the integral criteria presented in Equations 3, 5 and 8 involves numerical integration. The integration domain is $\mathbb{X}$ for *sur* and *timse*, and $\mathbb{X} \times \mathbb{X}$ for *jn*. Computing such integrals is not trivial as the evaluation cost of the integrand is significant. Consequently, the integration points should be chosen carefully, especially when the dimension is high, in order to accurately evaluate the criteria at reasonable cost.

In **KrigInv**, three options are available for choosing integration points: fixed integration points (provided by the user), random integration points with uniform distribution, or random integration points with an instrumental distribution. The *integcontrol* argument of `EGI` or `EGIparallel` is a list specifying how to build these integration points. The most important fields in this list are *integration.points* and *integration.weights* in

case the user decides to specify manually his own integration points and weights, obtained from another procedure, or *n.points* and *distrib* to specify instead the number M of integration points and the distribution to sample from. Possible values for *distrib* are *"sobol"* (default) or *"MC"*, to use the Sobol sequence or a sample from a uniform distribution. Other (recommended) values are the names of the integral criteria: *"timse"*, *"sur"* and *"jn"*. If the argument *integcontrol* is not set, the default setting is to take $100d$ integration points with the Sobol sequence.

Three instrumental distributions are available to compute the integrals present in the *timse*, *sur* and *jn* criteria. We use the expressions "*timse* (respectively, *sur* or *jn*) instrumental distribution" as each distribution is adapted to the corresponding sampling criterion. These distributions were obtained by noting that the integrand of any criterion (Equations 3, 5 and 8) may not depart much from the integrand of the corresponding uncertainty measure. This suggests the use of the following instrumental densities $h(\cdot)$:

- *timse* criterion (one-point and multi-point): $h(\mathbf{x}) \propto s_n^2(\mathbf{x})W_n(\mathbf{x})\mathrm{d}\mathbb{P}_{\mathbb{X}}(\mathbf{x}), \mathbf{x} \in \mathbb{X}$,
- *sur* criterion (one-point and multi-point): $h(\mathbf{x}) \propto p_n(\mathbf{x})(1-p_n(\mathbf{x}))\mathrm{d}\mathbb{P}_{\mathbb{X}}(\mathbf{x}), \mathbf{x} \in \mathbb{X}$,
- *jn* criterion: $h(z_1, z_2) \propto p_n(z_1)p_n(z_2)\mathrm{d}\mathbb{P}_{\mathbb{X}}(z_1)\mathrm{d}\mathbb{P}_{\mathbb{X}}(z_2),\ (z_1, z_2) \in \mathbb{X} \times \mathbb{X}$.

When *integcontrol$distrib*="timse","sur" or "jn", **KrigInv** automatically builds integration samples from the corresponding distributions. The integration sample is renewed at each iteration of the sequential inversion. We strongly recommend users to use these instrumental distributions as they have been shown to considerably enhance the computation and the optimization of integral criteria in practice (Chevalier et al., 2012).

4.2.2. Sampling from the instrumental density

Sampling from the instrumental densities $h(\cdot)$ is not an easy task. Indeed, as the inversion progresses, the region where the instrumental densities are strictly positive can become very narrow and non-connected, which excludes a basic Markov Chain Monte-Carlo approach.

For the moment, a simple procedure has been implemented to tackle this problem. We explain it shortly in the particular case of the *sur* instrumental distribution. The idea remains valid for the two other distributions. For *sur*, the idea consists in sampling from a simpler **discrete** instrumental distribution: $\sum_{j=1}^{N} p_n(\mathbf{u}_j)(1-p_n(\mathbf{u}_j))\delta_{\mathbf{u}_j}$, where N is a large number and $\mathbf{u}_1, \ldots, \mathbf{u}_N$ is an i.i.d sample of points with distribution $\mathbb{P}_{\mathbb{X}}$. Obtaining a weighted sample of M integration points from this discrete distribution is not hard. A major drawback of this method, mentioned in Chevalier et al. (2012), is that both M and N must tend to infinity to ensure the convergence to the integral.

In **KrigInv**, the user can modify the value of N (default: $10M$) in the *integcontrol* list through the *n.candidates* field. A higher value of N entails a more precise instrumental density and improves the quality of the integration sample (and thus the accuracy of the

criterion computation). However, it also increases computation time. The distribution $\mathbb{P}_{\mathbb{X}}$ of these N points (field *init.distrib*) is uniform by default. The user has the possibility to specify manually the position of these N points if he wants a sample from a non-uniform distribution $\mathbb{P}_{\mathbb{X}}$. Note that when the *jn* criterion is used with the corresponding *jn* instrumental distribution, points are sampled in $\mathbb{X} \times \mathbb{X}$ (and not $\mathbb{X}$). If M integration points in $\mathbb{X}$ (and not $\mathbb{X} \times \mathbb{X}$) are imposed, **KrigInv** automatically creates a grid of M^2 integration points in $\mathbb{X} \times \mathbb{X}$. Users choosing the *jn* criterion are strongly encouraged not to choose this expensive option and to rely on the *jn* instrumental distribution instead.

4.2.3. Illustration

We now illustrate the advantages of sampling from an instrumental density instead of uniform sampling with our R example. The code below generates a sample of 1000 integration points from the *sur* distribution (i.e. a distribution with density $h(\mathbf{x}) \propto p_n(\mathbf{x})(1 - p_n(\mathbf{x}))$), and a sample of 1000 integration points from a uniform distribution. The random samples are plotted on Figure 6.

```
#a km object is built as before (code not reproduced) from 12 obs.
#Sample of integration points from the "sur" distribution.
integcontrol <- list(n.points=1000,distrib="sur")
integ.outputs <- integration_design(integcontrol=integcontrol,d=2,
                 lower=c(0,0),upper=c(1,1),model=model,T=T)
print_uncertainty_2d(model=model,T=T,type="pn",show.points=FALSE,
             main="one sample from the instrumental density")
points(integ.outputs$integration.points,pch=17,cex=2)

#Sample of integration points from the uniform distribution.
integcontrol <- list(n.points=1000,distrib="MC")
integ.outputs <- integration_design(integcontrol=integcontrol,d=2,
                 lower=c(0,0),upper=c(1,1))
print_uncertainty_2d(model=model,T=T,type="pn",show.points=FALSE,
             main="one sample from uniform distribution")
points(integ.outputs$integration.points,pch=17,cex=2)
```

Table 2 compares the values of the *sur* criterion at point $\mathbf{x} = (0.2, 0.2)$ obtained using the two sampling scheme (averaged over 1000 repetitions) to an accurate estimation of the criterion based on 100000 points (using a Sobol sequence). We see that both methods have no bias as, in average, the value of the *sur* criterion at point $\mathbf{x}$ is the actual exact value. However, for a comparable computational cost, the evaluation of the criterion is a lot more accurate if the *sur* distribution is chosen. Indeed, for different random samples of 1000 integration points, the standard deviation of the value of the *sur* criterion at $\mathbf{x}$ with the instrumental distribution is one order of magnitude smaller than with a uniform distribution.

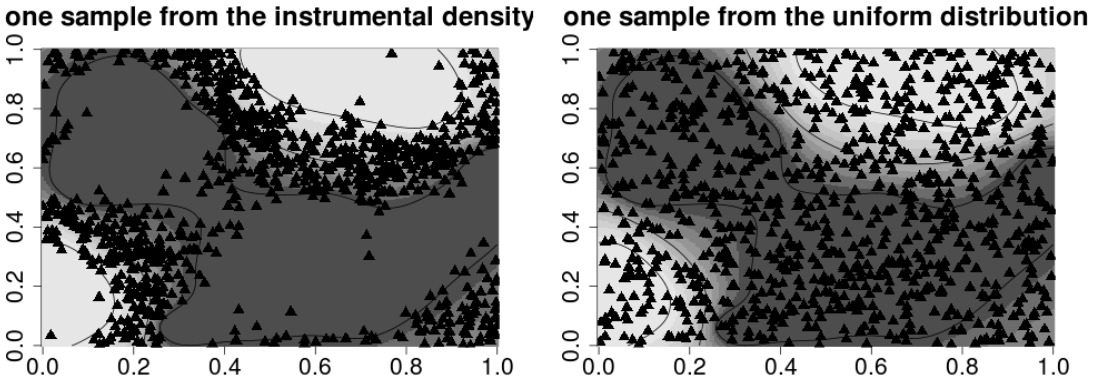

Figure 6: Excursion probability after 12 initial evaluations of the Branin function with a threshold $T = 80$. The set of triangles correspond to the sample used for the numerical integration. Such a sample is distributed with an instrumental density proportional to $p_n(\mathbf{x})(1 - p_n(\mathbf{x}))$ (left) and uniformly (right).

Table 2: *sur* values based on different sampling schemes. Numbers in parenthesis are standard deviations over 1000 repetitions.

MC (1000 pts)	Instrumental distribution sur (1000 pts)	Sobol (100000 pts)
6.04e-2 (2.4e-3)	6.04e-2 (3.3e-4)	6.04e-2

5. Conclusion and perspectives

The R package **KrigInv** offers sequential sampling strategies to estimate excursion sets, probabilities of failure and contour lines of a real-valued expensive-to-evaluate function by means of a kriging model. The goal of the present tutorial is to make the package accessible to people who are not familiar with kriging, and to clearly emphasize the strengths and limitations of such metamodel-based inversion methods.

From an end-user perspective, we would recommend to use a sampling criterion and parameters that do an adapted trade-off between performance of the criterion and computation time. If a single evaluation of f takes only a few seconds, pointwise criteria can quickly provide interesting results. On the other hand, if evaluating f takes many hours, it may be worth spending a few minutes to choose carefully the design points with an integral criterion and a generous budget for both efficient integration and optimization. Finally, if several CPUs are available to evaluate f simultaneously at different points, the user can take advantage of the parallel sampling criteria. In that case, the computational savings are likely to be very significant.

The current version of the **KrigInv** package can be further improved in different ways. Allowing the users to use their own optimizer for selecting the best points according to the proposed criteria may provide more flexibility to advanced users. Sequential

Monte Carlo methods might improve the performances of criteria involving integrals. Some of the sampling criteria implemented in the package (*sur* and *jn*) might be used in the case where the objective function returns a multivariate output (Conti and O'Hagan, 2010; Paulo et al., 2012) or even a function (Hung et al., 2012; Rougier, 2008), provided that $p_n(\cdot)$ can be computed easily. Finally, new criteria involving random set considerations are currently being studied and will be implemented in **KrigInv** in the longer term.

Acknowledgements: Part of this work has been conducted within the frame of the ReDice Consortium, gathering industrial (CEA, EDF, IFPEN, IRSN, Renault) and academic (Ecole des Mines de Saint-Etienne, INRIA, and the University of Bern) partners around advanced methods for Computer Experiments. Clément Chevalier also gratefully acknowledges support from the French Nuclear Safety Institute (IRSN). The authors would like to thank Dr Y. Richet (IRSN) for his help and feedback on the package.

Bibliography

Au, S. K., Beck, J., 2001. Estimation of small failure probabilities in high dimensions by subset simulation. Probab. Engrg. Mechan. 16 (4), 263–277.

Bect, J., Ginsbourger, D., Li, L., Picheny, V., Vazquez, E., 2012. Sequential design of computer experiments for the estimation of a probability of failure. Statistics and Computing 22(3), 773–793.

Bichon, B. J., Eldred, M. S., Swiler, L. P., Mahadevan, S., McFarland, J. M., 2008. Efficient global reliability analysis for nonlinear implicit performance functions. AIAA Journal 46 (10), 2459–2468.

Chevalier, C., Bect, J., Ginsbourger, D., Vazquez, E., Picheny, V., Richet, Y., April 2012. Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set.
URL `http://hal.inria.fr/hal-00641108/en`

Chevalier, C., Ginsbourger, D., March 2012. Corrected kriging update formulae for batch-sequential data assimilation.
URL `http://arxiv.org/abs/1203.6452`

Chilès, J.-P., Delfiner, P., 1999. Geostatistics: Modeling Spatial Uncertainty. Wiley, New York.

Conti, S., O'Hagan, A., 2010. Bayesian emulation of complex multi-output and dynamic computer models. Journal of Statistical Planning and Inference 140, 640–651.

Cressie, N., 1993. Statistics for spatial data.

Dixon, L., Szegö, G., 1978. Towards global optimisation 2. Vol. 2. North Holland.

Emery, X., 2009. The kriging update equations and their application to the selection of neighboring data. Computational Geosciences 13 (1), 211–219.

Fang, K.-T., Li, R., Sudjianto, A., 2006. Design and modeling for computer experiments. Chapman & Hall / CRC Press.

Forrester, A. I. J., Sóbester, A., Keane, A. J., 2008. Engineering design via surrogate modelling: a practical guide. Wiley.

Gayton, N., Bourinet, J. M., Lemaire, M., 2003. CQ2RS: a new statistical approach to the response surface method for reliability analysis. Structural Safety 25, 99–121.

Genz, A., 1992. Numerical computation of multivariate normal probabilities. Journal of Computational and Graphical Statistics 1, 141–149.

Gramacy, R. B., Lee, H. K., 2008. Gaussian processes and limiting linear models. Computational Statistics & Data Analysis 53 (1), 123–136.

Hung, Y., Roshan Joseph, V., Melkote, S., 2012. Analysis of computer experiments with functional response. Tentatively accepted by Technometrics.

Jones, D. R., Schonlau, M., William, J., 1998. Efficient global optimization of expensive black-box functions. Journal of Global Optimization 13 (4), 455–492.

Kim, S.-H., Na, S.-W., 1997. Response surface method using vector projected sampling points. Structural Safety 19, 3–19.

Marrel, A., Iooss, B., Van Dorpe, F., Volkova, E., 2008. An efficient methodology for modeling complex computer codes with gaussian processes. Computational Statistics & Data Analysis 52 (10), 4731–4744.

Mebane, W., Sekhon, J., 2011. Genetic optimization using derivatives: The rgenoud package for r. Journal of Statistical Software Vol. 42, Issue 11, 1–26.

Paulo, R., García-Donato, G., Palomo, J., 2012. Calibration of computer models with multivariate output. Computational Statistics & Data Analysis.

Picheny, V., Ginsbourger, D., Roustant, O., Haftka, R. T., Kim, N.-H., 2010. Adaptive designs of experiments for accurate approximation of target regions. Journal of Mechanical Design 132 (7).

Ranjan, P., Bingham, D., Michailidis, G., 2008. Sequential experiment design for contour estimation from complex computer codes. Technometrics 50 (4), 527–541.

Rasmussen, C. R., Williams, C. K. I., 2006. Gaussian Processes for Machine Learning. MIT Press.

Richet, Y., Deville, Y., Chevalier, C., 2012. DiceView : Plot methods for computer experiments design and surrogate.
URL http://cran.r-project.org/web/packages/DiceView

Rougier, J., 2008. Efficient emulators for multivariate deterministic functions. Journal of Computational and Graphical Statistics 17:4, 827–843.

Roustant, O., Ginsbourger, D., Deville, Y., 2012. Dicekriging, Diceoptim: Two R packages for the analysis of computer experiments by kriging-based metamodelling and optimization. Journal of Statistical Software 51.
URL http://www.jstatsoft.org/v51/i01

Rubinstein, R., Kroese, D., 2004. The Cross-Entropy Method. Springer.

Sacks, J., Welch, W. J., Mitchell, T. J., Wynn, H. P., 1989. Design and analysis of computer experiments. Statistical Science 4 (4), 409–435.

Santner, T. J., Williams, B. J., Notz, W., 2003. The Design and Analysis of Computer Experiments. Springer Verlag.

Stein, M. L., 1999. Interpolation of Spatial Data: Some Theory for Kriging. Springer, New York.

Zhao, Y.-G., Ono, T., 1999. A general procedure for first/second-order reliability method (form/sorm). Structural Safety 21, 95.

Appendix A. Kriging basics

The goal of this section is to provide a basic understanding of the kriging metamodel. Well known references on kriging include Stein (1999); Cressie (1993). In kriging we consider that f is a realization of a Gaussian process ξ. A key property in this setting is that, when n observations $\mathcal{A}_n$ of ξ are available, the conditional process $\xi|\mathcal{A}_n$ is still Gaussian. The unconditional covariance function $k(\cdot,\cdot)$ of ξ is assumed to be a known symmetric positive definite function or *kernel*. The *kriging mean* at a point $\mathbf{x} \in \mathbb{X}$, denoted by $m_n(\mathbf{x})$, is the best linear unbiased predictor of $\xi(\mathbf{x})$ from the observations. The conditional covariance from the n observations between two points $\mathbf{x}$ and $\mathbf{x}'$, known as *kriging covariance*, is denoted by $k_n(\mathbf{x},\mathbf{x}')$, so that, finally, $\xi|\mathcal{A}_n \sim GP(m_n, k_n)$. In particular, for all $\mathbf{x} \in \mathbb{X}$, $\xi(\mathbf{x})$ has a Gaussian distribution with mean $m_n(\mathbf{x})$ and variance $s_n^2(\mathbf{x}) := k_n(\mathbf{x},\mathbf{x})$. In *simple kriging* the unconditional mean function $m(\cdot)$ of ξ is assumed to be zero. In the *ordinary kriging* setting, the mean function is assumed to be an unknown constant μ. In that case, the kriging mean and covariance are given by:

$$m_n(\mathbf{x}) = \widehat{\mu} + k(\mathbf{x})^\top K^{-1}(\mathbf{y} - \widehat{\mu}\mathbb{1}) \tag{A.1}$$

$$k_n(\mathbf{x},\mathbf{x}') = k(\mathbf{x},\mathbf{x}') - k(\mathbf{x})^\top K^{-1} k(\mathbf{x}') + \frac{(1 - \mathbb{1}^\top K^{-1} k(\mathbf{x}))(1 - \mathbb{1}^\top K^{-1} k(\mathbf{x}'))}{\mathbb{1}^\top K^{-1}\mathbb{1}}, \tag{A.2}$$

where K is the $n \times n$ covariance matrix at the observations: $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $k(\mathbf{x})$ is the vector of size n with i^{th} entry equal to $k(\mathbf{x}, \mathbf{x}_i)$. $\mathbb{1}$ is the vector of size n with components equal to one and $\widehat{\mu}$ is the estimator of the trend from the n observations $\mathbf{y} = (\xi(\mathbf{x}_1), \ldots, \xi(\mathbf{x}_n))^\top$:

$$\widehat{\mu} = \frac{\mathbb{1}^\top K^{-1} \mathbf{y}}{\mathbb{1}^\top K^{-1} \mathbb{1}}. \tag{A.3}$$

The reader is referred to Roustant et al. (2012), Section 2, for the exact expressions of the kriging mean and variance in the more general *universal kriging* setting. The knowledge of these formulas is not required as all the computations are transparently performed in the DiceKriging package (Roustant et al., 2012). Such package allows to compute easily kriging means and variances from the observations at any points $\mathbf{x}$ through the construction of a *km* (kriging model) object.

Appendix B. Some outputs of an inversion

In this section, we describe what the actual outputs of an inversion can be. Indeed, at the end of an inversion, it is obviously not enough to indicate only what the newly evaluated points are. The function `print_uncertainty` has been coded to settle this issue.

This function is a wrapper around three functions, `print_uncertainty_1d`, `_2d` and `_nd`, which are called depending on the dimension d of the domain $\mathbb{X}$. The main feature of this function is to plot the function $p_n(\cdot)$ over the whole domain $\mathbb{X}$. Such a task is not difficult when $d \leq 2$, but becomes more challenging when $d > 2$. A first example in dimension one follows.

```
f <- function(x) return(x^2)
design <- matrix(c(0.1,0.3,0.4,0.9),ncol=1)
response <- f(design)
model1d <- km(formula=~1, design = design, response = response,
              covtype="matern3_2")
print_uncertainty_1d(model=model1d,T=0.5,type="pn",
                     xlab="x",ylab="pn(x)",cex.lab=1.5,cex.points=3,
                     main="excursion probability",cex.main=1.5)

design.updated <- matrix(c(design,0.6,0.7,0.8),ncol=1)
response.updated <- f(design.updated)
model1d.updated <- km(formula=~1, design = design.updated,
                      response = response.updated,covtype="matern3_2")
print_uncertainty_1d(model=model1d.updated,T=0.5,type="pn",
                     xlab="x",ylab="pn(x)",cex.lab=1.5,cex.points=3,
```

```
                    main="updated excursion probability",cex.main=1.5,
                    new.points=3,pch.points.end=19)
```

Figure B.7 gives the output of such code. In this example in dimension $d = 1$, the unknown function is $f(x) = x^2$. The threshold T is fixed to 0.5. As no domain $\mathbb{X}$ is specified in the arguments of `print_uncertainty_1d`, the default value $[0, 1]^d$ is used. The plots on the right are generated using the DiceView package (Richet et al., 2012) and represent the kriging mean and confidence intervals on the whole domain $\mathbb{X}$. These plots are useful to visualize our knowledge of the function f and the regions where f may exceed the threshold T. The plots on the left give the value of the uncertainty function. By default this function is defined as the function $p_n(\cdot)$, but other definitions can be set with the argument *type*. In the R example above we additionally evaluate f at points $0.6, 0.7, 0.8$. The result is a better knowledge of the excursion set $\{x \in [0, 1] : f(x) > 0.5\}$ as one can see that, with these three new evaluations, $p_n(x)$ is equal to 0 or 1 on almost all the domain.

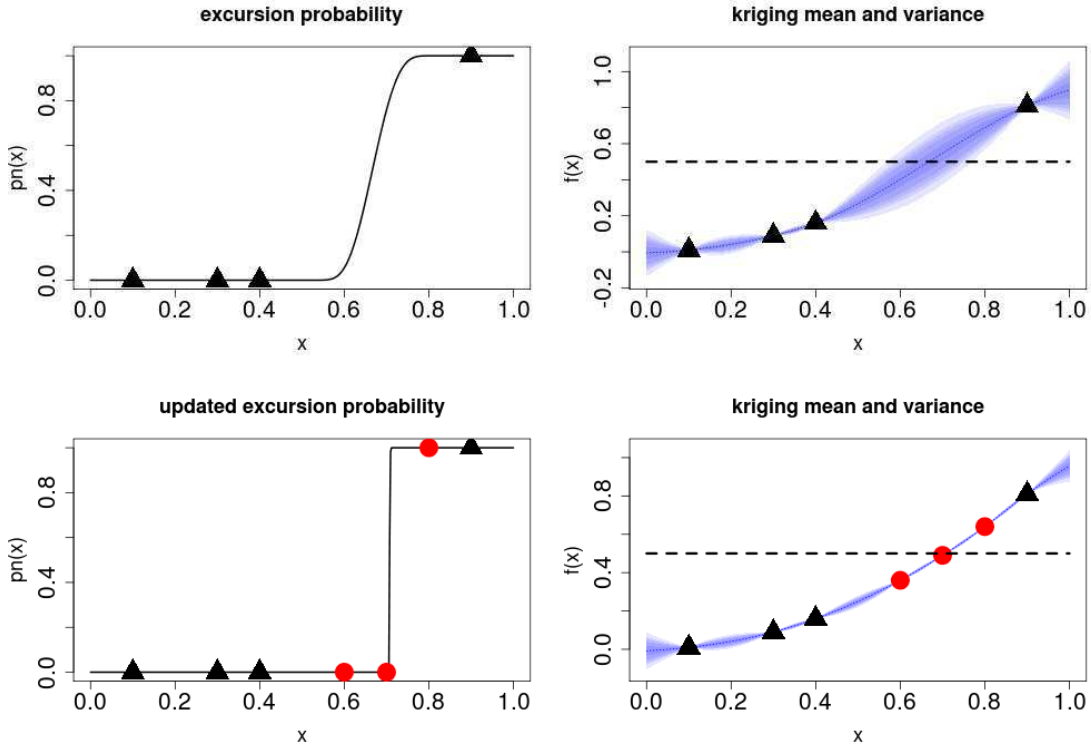

Figure B.7: Two calls of the `print_uncertainty_1d` function with two different *km* objects

An example in dimension $d = 2$ on the Branin-Hoo function is already widely developed in this paper and is not reproduced here. In dimension $d > 2$ it is not trivial to

represent the value of $p_n(\mathbf{x})$ on the whole domain $\mathbb{X}$. Let $\mathbf{x} = (x^{(1)}, \ldots, x^{(d)}) \in \mathbb{X}$. The `print_uncertainty_nd` function offers a pair plot with two possible options:

- option="mean": For all possible pairs of components $1 \leq i < j \leq d$ we plot the two-dimensional function:

$$g_{ij}(u, v) = \int_{\{\mathbf{x} \in \mathbb{X}:\ x^{(i)}=u, x^{(j)}=v\}} p_n(\mathbf{x}) \mathbb{P}_{\mathbb{X}}(\mathrm{d}\mathbf{x}),$$

- option="max": For all possible pairs of components $1 \leq i < j \leq d$ we plot the two-dimensional function:

$$h_{ij}(u, v) = \max_{\{\mathbf{x} \in \mathbb{X}:\ x^{(i)}=u, x^{(j)}=v\}} p_n(\mathbf{x}).$$

Figure B.8 represents the output of the following R example, with a three-dimensional function f.

```
f <- function(x) return( branin(c(x[1],x[2]) )*x[3] )
n <- 50 #high number of evaluations
T <- 80 #threshold
design <- data.frame(maximinLHS(n,k=3)) #initial design (a LHS)
response <- apply(X=design,MARGIN=1,FUN=f)
model3d <- km(formula=~1, design = design, response = response,
              covtype="matern3_2")

print_uncertainty_nd(model=model3d,T=80,type="pn",option="max",
                     main="max excursion probability",cex.main=2,
                     levels=c(0.05,0.5,0.95),
                     nintegpoints=100,resolution=30)
print_uncertainty_nd(model=model3d,T=80,type="pn",option="mean",
                     main="average excursion probability",cex.main=2,
                     levels=c(0.05,0.5,0.95),
                     nintegpoints=100,resolution=30)
```

In Figure B.8 "max" (or "average") excursion probability refers to a maximum (resp. average) excursion probability with respect to $d-2$ variables. Note that the computation performed in the `print_uncertainty_nd` function are very intensive. One can change the arguments *nintegpoints* to control the number of integration points in the integral of g_{ij} (or the maximum in h_{ij}) and *resolution* to tune the resolution of each image. Each pixel corresponds to one evaluation of the function g_{ij} or h_{ij}.

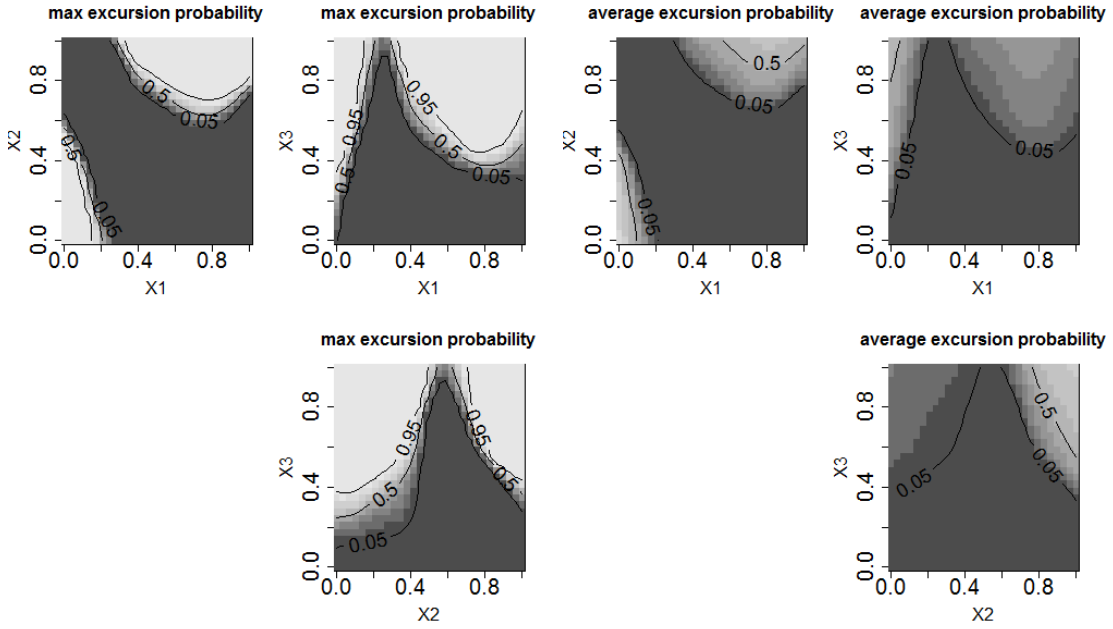

Figure B.8: Two calls of the `print_uncertainty_nd` function with two different options

Appendix F: Ongoing work in robust inversion

F.1 A new SUR criterion based on fast Gaussian process conditional simulation

This Section presents an ongoing work on an alternative SUR strategy for robust inverson which mitigates one of the drawbacks of the SUR strategy of Section 6.2 (see, remarks on the discretization parameter, ℓ, in the last paragraph of Section 6.2.2). The construction of this new SUR strategy relies, again, on update formulas which allow significant computational savings and – sometimes – enable to find closed-form expressions when conditional expectations (like the ones of Equations (6.14),(4.12),(4.13)) are at stake.

An important difference is that the update formulas involved are not meant to update a kriging mean or variance but rather to update a **conditional realization of a Gaussian process**. In short, a Gaussian process realization conditioned on n observations can be "transformed" efficiently in order to be conditioned on $n+q$ observations. Moreover, the dependence of the Gaussian process realization on the response at these q new points can be exhibited explicitly, which may allow computation of expectations with respect to these random responses. The next subsection details these arguments. We then propose an uncertainty measure, provide closed-form expressions for the associated SUR criteria and test it on the

nuclear safety case.

F.1.1 Updating Gaussian process realizations

Let $\xi \sim \mathrm{GP}(m, k)$ be a Gaussian process on a domain $\mathbb{X}$ and $(\mathbf{x}^1, \ldots, \mathbf{x}^p) := \mathbf{x}^{(p)} \in \mathbb{X}^p$ be a batch of p locations. Computing simulations of ξ at locations $\mathbf{x}^{(p)}$ amounts to simulate realizations of a Gaussian random vector with mean vector $m(\mathbf{x}^{(p)}) = (m(\mathbf{x}^1), \ldots, m(\mathbf{x}^p))^\top$ and covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{p \times p}$ such that $\mathbf{\Sigma}_{ij} = k(\mathbf{x}^i, \mathbf{x}^j)$. Standard algorithms are based on a decomposition (e.g., LU or Cholesky) of $\mathbf{\Sigma}$ which comes with at a cost of $O(p^3)$. Then, each simulation consists of a matrix-vector product so that, when a large number M of simulations is performed, the complexity is of $O(Mp^2)$ (see, e.g. Ripley [2009]).

As explained in Appendix A, a well known algorithm for simulating M Gaussian process realizations in p locations conditionally on n observations consists in adding to the kriging mean obtained with the n observations M kriging residual functions artificially obtained from non-conditional realizations. The algorithm is detailed in, e.g., Chilès and Delfiner [2012]; Hoshiya [1995]; Vazquez [2005].
Another possible interpretation of the algorithm is to say that the conditional simulation is obtained by adding the difference between two kriging mean functions to the non-conditional simulation. The two kriging mean functions are obtained from the real n observations or from n simulated observations.

As the method described here does not depend on the choice of the mean function m and covariance kernel k, one may apply this algorithm to realizations of a Gaussian process already conditioned on n observations (with mean function $m_n(\cdot)$ and covariance function $k_n(\cdot, \cdot)$) in order to condition the realizations to $q \geq 1$ additional observations at a batch $\mathbf{x}_{\text{new}} = \mathbf{x}^{(q)} \in \mathbb{X}^q$. The observations locations are denoted by $\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+q}$. Mathematically if $\mathbf{x}$ is a point of the batch $\mathbf{x}^{(p)}$ we have:

$$z_{n+q}(\mathbf{x}) = z_n(\mathbf{x}) + (m_{n+q}(\mathbf{x})_{\text{real}} - m_{n+q}(\mathbf{x})_{\text{simulated}}), \tag{F.1}$$

where $z_n(\mathbf{x})$ is the old simulated value at point $\mathbf{x}$, $z_{n+q}(\mathbf{x})$ is the newly simulated

value, $m_{n+q}(\cdot)_{\text{real}}$ is the updated kriging mean function obtained from the real q new observations (at $\mathbf{x}_{\text{new}}$) and $m_{n+q}(\cdot)_{\text{simulated}}$ is the updated kriging mean function obtained from q simulated observations at $\mathbf{x}_{\text{new}}$.
Note that if the q points of the batch $\mathbf{x}_{\text{new}}$ are also points of the batch $\mathbf{x}^{(p)}$ then the simulations at these points are already done. However, in the opposite case the conditional simulation of $\xi(\mathbf{x}_{\text{new}})$ need to be performed before computing the kriging mean. These simulations are performed conditionally on $n+p$ observations, which involves a potentially important cost of $O((n+p)^3 + Mq^2)$.

Equation (F.1) written is this form emphasizes that we need to compute a difference between two updated kriging mean function. This motivates the use of kriging update formulas (see, Equation (2.21)). Equation (F.1) can be rewritten:

$$z_{n+q}(\mathbf{x}) = z_n(\mathbf{x}) + k_n(\mathbf{x}, \mathbf{x}_{\text{new}})^\top K_{\text{new}}^{-1}(\xi(\mathbf{x}_{\text{new}})_{\text{real}} - \xi(\mathbf{x}_{\text{new}})_{\text{simulated}}) \tag{F.2}$$

where $K_{\text{new}} := k_n(\mathbf{x}_{n+i}, \mathbf{x}_{n+j})_{1\leq i,j\leq q}$ and $k_n(\mathbf{x}, \mathbf{x}_{\text{new}}) := (k_n(\mathbf{x}, \mathbf{x}_{n+1}), \ldots, k_n(\mathbf{x}, \mathbf{x}_{n+q}))^\top$. In conclusion, once the q kriging weights $\boldsymbol{\lambda}_{\text{new}}(\mathbf{x}) := K_{\text{new}}^{-1} k_n(\mathbf{x}, \mathbf{x}_{\text{new}})$ of $\mathbf{x}_{\text{new}}$ for the prediction at point $\mathbf{x}$ are precomputed for each point $\mathbf{x}$ of the batch $\mathbf{x}^{(p)}$, the update of one conditional simulation comes at a cost of only $O(pq)$ (i.e. p vector-vector products of size q). This cost is lower than the cost of $O(p^2)$ of classical algorithms. An example of update of a conditional Gaussian process realization is illustrated in Appendix A.

In addition to a reduced cost for updating conditional realizations of a Gaussian process, Equation (F.2) has the advantage to exhibit the relation between the updated Gaussian process realization and the true response $\xi(\mathbf{x}_{\text{new}}) = \xi(\mathbf{x}_{\text{new}})_{\text{real}}$. The latter will be further used in the next subsection.

F.1.2 Uncertainty quantification and SUR criteria

Let us consider again the problem of using SUR strategies for robust inversion. We are using the notations of Sections 6.1 and 6.2. Let $\mathbf{x}_c \in \mathbb{X}_c$. In this section

we consider a new approximation of the non-excursion probability,

$$\widetilde{p_n}(\mathbf{x}_c) := P\left(\max_{\mathbf{x}_{nc}\in\mathbb{X}_{nc}} \xi_{\mathbf{x}_c}(\mathbf{x}_{nc}) \leq T | \mathcal{A}_n\right), \tag{F.3}$$

of the sectional Gaussian process $\xi_{\mathbf{x}_c}$. Let $M > 1$ and let us consider M realizations of $\xi_{\mathbf{x}_c}$ conditioned on n real observations $\mathcal{A}_n$. Note that the observation locations, $\mathbf{x}_1, \ldots, \mathbf{x}_n$ lie in $\mathbb{X}$ and not in $\mathbb{X}_c$.

The M conditional simulations of $\xi_{\mathbf{x}_c}$ are supposed to be performed in ℓ points of $\mathbb{X}_{nc}$, $\mathbf{x}_{nc}^1, \ldots, \mathbf{x}_{nc}^\ell$ and are denoted by $(z_{1,n}^1, \ldots, z_{1,n}^\ell), \ldots, (z_{M,n}^1, \ldots, z_{M,n}^\ell)$. A natural Monte-Carlo approximation of $\widetilde{p_n}(\mathbf{x}_c)$ consists in counting the proportion of realizations that do not exceed T:

$$\widehat{p_n}(\mathbf{x}_c) := \frac{1}{M}\#\{1 \leq i \leq M : \max_{1\leq j\leq \ell} z_{i,n}^j \leq T\}, \tag{F.4}$$

where # denotes the cardinal of a set. Now, as in Section 6.2, the uncertainty at time n and the associated SUR criterion can be written:

$$\widehat{H}_n(\mathcal{A}_n) := \int_{\mathbb{X}_c} \widehat{p_n}(1-\widehat{p_n})\mathrm{d}\mathbb{P}_{\mathbb{X}_c} \tag{F.5}$$

$$\widehat{J}_n(\mathbf{x}^{(q)}) := \mathbb{E}_n\left(\int_{\mathbb{X}_c} \widehat{p_{n+q}}(1-\widehat{p_{n+q}})\mathrm{d}\mathbb{P}_{\mathbb{X}_c}\right). \tag{F.6}$$

We will now use results from Section F.1.1 to obtain closed-form expression for Equation (F.6) in the particular case where the size of the batch is $q = 1$.
Let us consider a batch of $q = 1$ point, $\mathbf{x}^{(q)} := \mathbf{x}_{n+1} \in \mathbb{X}$. If $\mathbf{x}_{n+1}$ is equal to one of the points $(\mathbf{x}_c, \mathbf{x}_{nc}^j)$ for some $1 \leq j \leq \ell$, then let $z_{i,n}^{\ell+1} := z_{i,n}^j, 1 \leq i \leq M$. Otherwise for each $(1 \leq i \leq M)$, the number $z_{i,n}^{\ell+1}$ is obtained by simulating $\xi(\mathbf{x}_{n+1})$, conditionally on the $n+\ell$ observations $\xi(\mathbf{x}_1), \ldots, \xi(\mathbf{x}_n), z_{i,n}^1, \ldots, z_{i,n}^\ell$. Now, Equation (F.2) can be rewritten as follows:

$$z_{i,n+1}^k = z_{i,n}^k + \lambda(\mathbf{x}_{nc}^k)\left(\xi(\mathbf{x}_{n+1}) - z_{i,n}^{\ell+1}\right), \ 1 \leq i \leq M, \ 1 \leq k \leq \ell, \tag{F.7}$$

where $\lambda(\mathbf{x}_{nc}^k)$ is the kriging weight of $\mathbf{x}_{n+1}$ for the prediction at point $(\mathbf{x}_c, \mathbf{x}_{nc}^k)$, at time n. This weight is equal to $k_n(\mathbf{x}_{n+1}, (\mathbf{x}_c, \mathbf{x}_{nc}^k))/s_n^2(\mathbf{x}_{n+1})$.
Equation (F.7) will be useful to calculate the coverage probability at time $n+1$:

$$\widehat{p_{n+1}}(\mathbf{x}_c) := \frac{1}{M}\#\{1 \leq i \leq M : \max_{1\leq j\leq \ell} z_{i,n+1}^j \leq T\}. \tag{F.8}$$

as, for each simulation i we will manage to know, in function of $\xi(\mathbf{x}_{n+1})$, whether $\max_{1\leq j\leq \ell} z_{i,n+1}^j \leq T$ or not.

Let $k \in \{1, \ldots, \ell\}$ and let us the notations $a_i^k := z_{i,n}^k - \lambda(\mathbf{x}_{nc}^k) z_{i,n}^{\ell+1}$ and $b^k := \lambda(\mathbf{x}_{nc}^k)$. If $b^k > 0$, we have,

$$\begin{aligned} z_{i,n+1}^k \leq T \Leftrightarrow & a_i^k + b^k \xi(\mathbf{x}_{n+1}) \leq T \\ \Leftrightarrow & \xi(\mathbf{x}_{n+1}) \leq \frac{T - a_i^k}{b^k}, \end{aligned}$$

and, then,

$$\max_{1\leq j\leq \ell,\ b^j>0} z_{i,n+1}^j \leq T \Leftrightarrow \xi(\mathbf{x}_{n+1}) \leq \min_{1\leq j\leq \ell,\ b^j>0} \frac{T - a_i^j}{b^j} \tag{F.9}$$

Similarly, we can show that:

$$\max_{1\leq j\leq \ell,\ b^j\leq 0} z_{i,n+1}^j \leq T \Leftrightarrow \xi(\mathbf{x}_{n+1}) \geq \max_{1\leq j\leq \ell,\ b^j<0} \frac{T - a_i^j}{b^j}. \tag{F.10}$$

Also, the case of zero kriging weights needs to be taken into account:

$$\max_{1\leq j\leq \ell,\ b^j=0} z_{i,n+1}^j \leq T \Leftrightarrow \max_{1\leq j\leq \ell,\ b^j=0} z_{i,n}^j \leq T. \tag{F.11}$$

In conclusion, the updated simulation number i does not exceed T if and only if

the condition (F.11) is satisfied and if:

$$\xi(\mathbf{x}_{n+1}) \in [u_i, v_i] := \left[\max_{1\leq j\leq \ell,\ b^j<0} \frac{T-a_i^j}{b^j}, \min_{1\leq j\leq \ell,\ b^j>0} \frac{T-a_i^j}{b^j} \right] \tag{F.12}$$

if the latter interval exists (i.e. if the lower bound is lower than the upper bound); and with the convention : $u_i := -\infty$ if no b_j is negative and $v_i := +\infty$ if no b_j is positive.

As, at time n, $\xi(\mathbf{x}_{n+1}) \sim \mathcal{N}(m_n(\mathbf{x}_{n+1}), s_n^2(\mathbf{x}_{n+1}))$, let us write $\xi(\mathbf{x}_{n+1}) =^d m_n(\mathbf{x}_{n+1}) + s_n(\mathbf{x}_{n+1})N$, where $N \sim \mathcal{N}(0,1)$. Equation (F.12) can be rewritten:

$$N \in [u_i', v_i'] := \left[\frac{u_i - m_n(\mathbf{x}_{n+1})}{s_n(\mathbf{x}_{n+1})}, \frac{v_i - m_n(\mathbf{x}_{n+1})}{s_n(\mathbf{x}_{n+1})} \right]. \tag{F.13}$$

Note that, $u_i' \geq v_i'$ means that the updated simulation i **always** exceeds or equals T, simply because conditions (F.9) and (F.10) cannot be both satisfied. . Thus, to count the number of (updated) simulations that do not exceed T we only consider the set of indices i, where the simulation i might not exceed T, written as follows:

$$E := \{1 \leq i \leq M : \max_{1\leq j\leq \ell,\ b^j=0} z_{i,n}^j \leq T \text{ and } u_i' < v_i'\} \tag{F.14}$$

A simple way to count this number of simulations, in function of the (centered, normalized) response N is to count the number of u_i''s lower than N and to subtract the number of v_i''s lower than N:

$$\widehat{p_{n+1}}(\mathbf{x}_c) := \frac{1}{M} \left(\#\{i \in E : u_i' < N\} - \#\{i \in E : v_i' < N\} \right). \tag{F.15}$$

It thus appears that $\widehat{p_{n+1}}(\mathbf{x}_c)$, seen as a function of the normalized response N is piecewise constant on $k-1$ intervals $(w_1 := -\infty, w_2), (w_2, w_3), \ldots, (w_{k-1}, w_k := \infty)$, for some $k > 1$, where the w_i's can be calculated by sorting the u_i''s and the v_i''s. With these notations, $\widehat{p_{n+1}}(\mathbf{x}_c)(1 - \widehat{p_{n+1}}(\mathbf{x}_c))$ seen as a function of N is also piecewise constant on these $k-1$ intervals, with values denoted by $\gamma_1, \ldots, \gamma_{k-1}$. To obtain the expectation of $\widehat{p_{n+1}}(\mathbf{x}_c)(1 - \widehat{p_{n+1}}(\mathbf{x}_c))$ with respect to the random

response N, it remains to integrate this function with respect to the standard Gaussian distribution:

$$\mathbb{E}_n\left(\widehat{p_{n+1}}(\mathbf{x}_c)(1-\widehat{p_{n+1}}(\mathbf{x}_c))\right) = \sum_{j=1}^{k-1} \gamma_j(\Phi(w_{j+1}) - \Phi(w_j)), \tag{F.16}$$

which ends our calculations as the SUR criterion given by Equation (F.6) can be computed by integrating the latter expression over $\mathbb{X}_c$.

The obtained closed-form expression in Equation (F.16) has advantages and drawbacks compared to the SUR criterion studied in Section 6.2. Its use allows the choice of a much larger discretization parameter ℓ as the computation cost is not expected to rise exponentially with ℓ like before, with the call to $\Phi_{2\ell}$. Moreover, the computation cost of the criterion is expected to be lower as the computation of kriging weights (to obtain the u_i''s and v_i''s), or of the c.d.f., Φ of the standard normal distribution, are not expensive operations. On the other hand, the computation of this SUR criterion is expected to require an important memory as, for each integration point $\mathbf{x}_c \in \mathbb{X}_c$, a large number, M, of conditional simulations in ℓ points needs to be saved. Moreover, the explicit expressions for the expectation of the future uncertainties were obtained when only $q = 1$ point is added. Further work is required to establish similar expressions for $q > 1$ and, one may expect that these expressions will involve computations of Φ_q, which will be more expensive.

F.1.3 Application to the test-case

The test-case introduced in Section 5.3 is now studied using this new SUR criterion computed using Equation (F.16). Most of the parameters of the algorithm are similar to those of Section 6.2.3. However, we list below some changes. The values between parenthesis are the values used in Section 6.2.3.

- Number of points evaluated per iteration: $q = 1$ (4).
- Total number of iterations: 160 (40).
- SUR criterion minimized at each iteration: $\widehat{J}_n$ (see, Equation (F.6)).

- Number of integration point for the integral over $\mathbb{X}_c$: 100 points renewed at each iteration (200).
- Parameters of the genoud algorithm: pop.size = 50, max.generation = 4 (100,5).
- Discretization parameter: $\ell = 50$ (5).
- Number of GP realization per integration point: $M = 500$.

In comparison to Section 5.3, the budget for evaluation of the criterion (number of integration points) and for its optimization (parameters of the genetic algorithm) is slightly reduced because the current R implementation of the SUR strategy is not well optimized yet, and quite computer intensive at the moment.
Figure F.3 shows that one part of the safe region is not properly identified (at the top left on $\mathbb{X}_c$). Some newly evaluated locations (in red) do not seem to be relevant as there are in a region where $\widehat{p_n} \approx 0$ or 1. Further work is necessary to determine whether this is due to the low optimization budget (and the small number of integration points) or not. The SUR criteria presented in this Section is indeed expected to be - in average - more reliable than the previous one as it enables the use or a larger discretization parameter ℓ which is convenient, specially if the number of non-controlled parameters, d_{nc} is greater than 1.

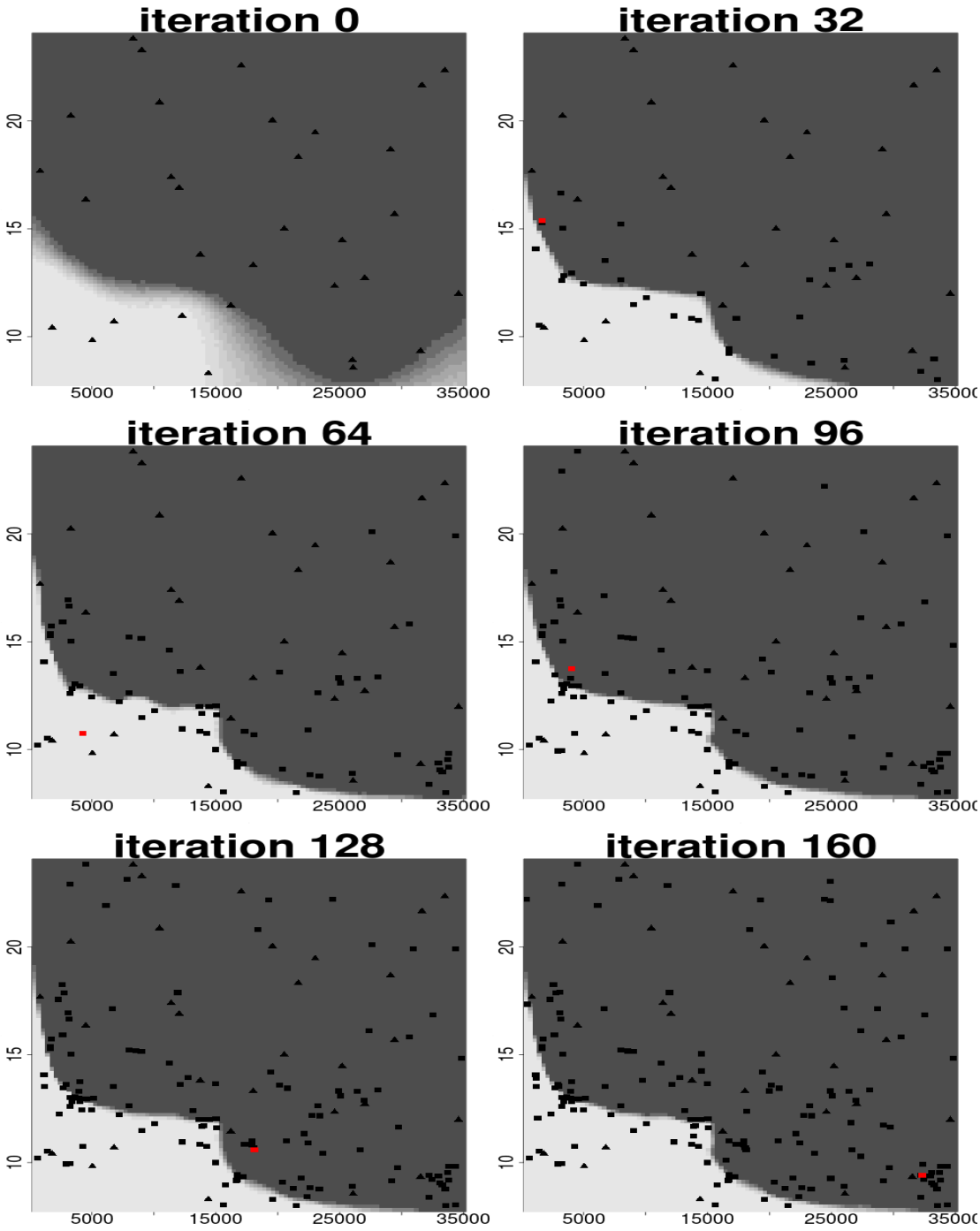

Figure F.3: Plot of the function $\widehat{p_n}(\mathbf{x}_c)$ of Equation (F.4) after n evaluations of the simulator. The triangles are the (projected) 30 points of the initial design. Areas in black correspond to $\widehat{p_n} \approx 0$ and areas in white correspond to $\widehat{p_n} \approx 1$.

F.2 Bonferroni bounds for the exceedance probability

In this Section, we are interested in bounds for the quantity:

$$p := \mathbb{P}(\max_{i\in\{1,\dots,q\}} Y_i \geq T), \tag{F.17}$$

where $T \in \mathbb{R}$ is an arbitrary threshold and $\mathbf{Y} = (Y_1, \dots, Y_q)^\top$ is a Gaussian vector of large size q (typically, $q \gg 1000$), with mean $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$. A direct calculation of p is possible through the c.d.f. of the multivariate normal distribution in dimension q (see, e.g., the algorithms of Genz [1992]) but such calculation is expected to be computationally expensive; so that we are here interested in a fast-to-compute upper bound of p.

We follow the article of Taylor et al. [2007] to mention that a trivial upper bound for p is the so called "Bonferroni" bound:

$$p \leq \sum_{i=1}^{q} \mathbb{P}(Y_i \geq T) = \sum_{i=1}^{q} \Phi\left(\frac{m_i - T}{\sqrt{\mathbf{\Sigma}_{ii}}}\right). \tag{F.18}$$

However, the Bonferroni bound might not be sharp if the correlation between the Y_i's is strong. Thus, following the idea that the maximum of the Y_i's has to be one of them we have:

$$p = \sum_{i=1}^{q} \mathbb{P}(Y_i \geq T \text{ and for all } j \neq i, Y_i > Y_j) \tag{F.19}$$

$$\leq \sum_{i=1}^{q} \mathbb{P}(Y_i \geq T \text{ and for some } j(i) \neq i, Y_i > Y_{j(i)}). \tag{F.20}$$

Equation (F.20) raises the choice of the indices $j(i)$ leading to a good bound (i.e. as low as possible), without needing expensive calls to the c.d.f. of the normal distribution in high dimension.

In Taylor et al. [2007], $Y_1, \ldots, Y_q$ are considered as unknown responses of a non-stationary Gaussian process ξ which is discretely sampled at q points $\mathbf{x}_1, \ldots, \mathbf{x}_q$ of a $d-$dimensional grid in an input space $\mathbb{X}$, which is a compact subset of $\mathbb{R}^d$. Thus, p is seen as an approximation of the probability of exceedance of ξ above T. When ξ is stationary and centered, Taylor et al. [2007] suggest to select, in each term of the sum (F.20) the $2d$ "neighbours" of the point $\mathbf{x}_i$ on the grid. This writes:

$$p \leq \sum_{i=1}^{q} \mathbb{P}(\xi(\mathbf{x}_i) \geq T \text{ and } \xi(\mathbf{x}_i) > \xi(\mathbf{x}_j) \text{ for all } \mathbf{x}_j \text{ neighbour of } \mathbf{x}_i) \qquad \text{(F.21)}$$

Each term of the sum (F.21) requires one call to the c.d.f. of the multivariate normal distribution in dimension $2d+1$. However, when ξ is stationary and centered, Taylor et al. [2007] manage to significantly reduce the cost for computing these terms.

In a more general setting (arbitrary Gaussian vector $\mathbf{Y}$) we may choose the indices $j(i)$ using a different strategy. More precisely, we believe that this work opens interesting perspectives for using SUR strategies in robust inversion (see, Chapter 6).

References

R.J Adler and J.E. Taylor. *Random fields and geometry*, volume 115. Springer, 2007. 80

S. K. Au and J. Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probab. Engrg. Mechan.*, 16(4):263–277, 2001.

A. Azzalini. *mnormt: The multivariate normal and t distributions*, 2012. URL `http://azzalini.stat.unipd.it/SW/Pkg-mnormt/`. R package version 1.4-5. 50, 72

S. Baillargeon. Le krigeage, revue de la théorie et application à l'interpolation spatiale de données de pr'ecipitations. Master's thesis, Université de Laval, 2005. 18

R. J. Barnes and A.G. Watson. Efficient updating of kriging estimates and variances. *Mathematical geology*, 24(1):129–133, 1992. 23

M. J. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu. A framework for validation of computer models. *Technometrics*, 49(2):138–154, 2007.

J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vazquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3):773–793, 2012. 20, 33, 36, 37, 38, 39, 40, 69

R. Benassi, J. Bect, and E. Vazquez. Robust gaussian process-based global optimization using a fully bayesian expected improvement criterion. In Springer,

editor, *Learning and Intelligent Optimization, Lecture Notes in Computer Science*, volume 6683, pages 176–190, 2011.

R. Bettinger. *Inversion dun systeme par krigeage. Application à la synthese de catalyseurs à haut débit*. PhD thesis, PhD thesis, université de Nice-Sophia Antipolis, 2009. 30

B. J. Bichon, M. S. Eldred, L. P. Swiler, S. Mahadevan, and J. M. McFarland. Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA Journal*, 46(10):2459–2468, 2008. 36, 52

C. Chevalier and D. Ginsbourger. Fast computation of the multi-points expected improvement with applications in batch selection. *Proceedings of the LION7 conference, Lecture Notes in Computer Science*, 2013. 13, 14, 26

C. Chevalier, V. Picheny, and D. Ginsbourger. *KrigInv: Kriging-based Inversion for Deterministic and Noisy Computer Experiments*, 2012. URL http://CRAN.R-project.org/package=KrigInv. R package version 1.3. 51

C. Chevalier, J. Bect, D. Ginsbourger, E. Vazquez, V. Picheny, and Y. Richet. Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Accepted with minor revision to Technometrics*, 2013a. URL http://hal.inria.fr/hal-00641108/en. 36, 37, 41

C. Chevalier, D. Ginsbourger, J. Bect, and I. Molchanov. Estimating and quantifying uncertainties on level sets using the vorob'ev expectation and deviation with gaussian process models. *mODa 10, Advances in Model-Oriented Design and Analysis, Physica-Verlag HD*, 2013b. 37, 46, 80

C. Chevalier, D. Ginsbourger, and X. Emery. Corrected kriging update formulae for batch-sequential data assimilation. *Proceedings of the IAMG2013 conference, Madrid*, 2013c. URL http://arxiv.org/abs/1203.6452. 23, 79

C. Chevalier, V. Picheny, and D. Ginsbourger. Kriginv: An efficient and user-friendly implementation of batch-sequential inversion strategies based on kriging. *Computational Statistics & Data Analysis*, 2013d. doi: http://dx.doi.org/10.1016/j.csda.2013.03.008. 37, 44, 50, 51

J.-P. Chilès and P. Delfiner. *Geostatistics: Modeling Spatial Uncertainty*. Wiley, New York, second edition edition, 2012. 6, 17, 159

W.J. Cody. Rational Chebyshev approximations for the error function. *Math. Comp.*, 23:631–637, 1969. 9

S. Conti and A. O'Hagan. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of Statistical Planning and Inference*, 140: 640–651, 2010.

N. Cressie. *Statistics for spatial data*. 1993. 19

N. Cressie, A.S. Davis, and J. Leroy Folks. The moment-generating function and negative integer moments. *The American Statistician*, 35 (3):148–150, 1981.

S. Da Veiga and A. Marrel. Gaussian process modeling with inequality constraints. *Annales de la Faculté des Sciences de Toulouse*, 21 (3):529–555, 2012. 13

P. Del Moral, A. Doucet, and A. Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006. 55

L.C.W. Dixon and G.P. Szegö. *Towards global optimisation 2*, volume 2. North Holland, 1978.

V. Dubourg. *Métamodèles adaptatifs pour l'analyse de fiabilité et l'optimisation sous contrainte fiabiliste*. PhD thesis, IFMA, 2012.

B. Echard, N. Gayton, and M. Lemaire. Kriging-based Monte Carlo simulation to compute the probability of failure efficiently: AK-MCS method. In *6èmes Journées Nationales de Fiabilité, 24–26 mars, Toulouse, France*, 2010.

B. Echard, N. Gayton, and M. Lemaire. AK-MCS: An active learning reliability method combining kriging and monte carlo simulation. *Structural Safety*, 33: 145–154, 2011. 36

X. Emery. The kriging update equations and their application to the selection of neighboring data. *Computational Geosciences*, 13(1):211–219, 2009. 23

K.-T. Fang, R. Li, and A. Sudjianto. *Design and modeling for computer experiments*. Chapman & Hall / CRC Press, 2006.

F. Fernex, L. Heulers, O. Jacquet, J. Miss, and Y. Richet. The moret 4b monte carlo code - new features to treat complex criticality systems. *M&C International Conference on Mathematics and Computation Supercomputing, Reactor Physics and Nuclear and Biological Application, Avignon, France.*, 2005. 59

S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization bencharking 2009: Presentation of the noiseless functions. Technical report, Research Center PPE, 2009, 2010.

F. Fleuret and D. Geman. Graded learning for object detection. In *Proceedings of the workshop on Statistical and Computational Theories of Vision of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR/SCTV)*, 1999. 2

A. I. J. Forrester, A. Sóbester, and A. J. Keane. *Engineering design via surrogate modelling: a practical guide*. Wiley, 2008.

P. I. Frazier. Parallel global optimization using an improved multi-points expected improvement criterion. In *INFORMS Optimization Society Conference, Miami FL*, 2012. 26

H. Gao, J. Wang, and P. Zhao. The updated kriging variance and optimal sample design. *Mathematical Geology*, 28:295–313, 1996. 23

N. Gayton, J. M. Bourinet, and M. Lemaire. CQ2RS: a new statistical approach to the response surface method for reliability analysis. *Structural Safety*, 25: 99–121, 2003.

D. Geman and B. Jedynak. An active testing model for tracking roads in satellite images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(1):1–14, 1996. 30

A. Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1:141–149, 1992. 45, 167

A. Genz and F. Bretz. *Computation of multivariate normal and t probabilities*, volume 195. Springer, 2009. 45

A. Genz, F. Bretz, T. Miwa, X. Mi, F. Leisch, F. Scheipl, B. Bornkamp, and T. Hothorn. *mvtnorm: Multivariate Normal and t Distributions*, 2012. URL http://cran.r-project.org/web/packages/mvtnorm/vignettes/MVT_Rnews.pdf. R package version 0.9-9992. 72

D. Ginsbourger. *Métamodèles multiples pour l'approximation et l'optimisation de fonctions numériques multivariables*. PhD thesis, Ecole nationale supérieure des Mines de Saint-Etienne, 2009. 1, 6, 14

D. Ginsbourger and R. Le Riche. Towards gaussian process-based optimization with finite time horizon. In *mODa 9–Advances in Model-Oriented Design and Analysis*, pages 89–96. Springer, 2010. 30, 32, 33

D. Ginsbourger, R. Le Riche, and Carraro L. Kriging is well-suited to parallelize optimization. In *Computational Intelligence in Expensive Optimization Problems*. Springer, 2010. 14, 26, 79

R. Girdziusas, J. Janusevskis, and R Le Riche. On integration of multi-point improvements. *Neural Information Processing Systems (NIPS) workshop*, 2012. 26

R. Gramacy and H. Lee. Gaussian processes and limiting linear models. *Computational Statistics & Data Analysis*, 53(1):123–136, 2008.

R. Gramacy and N. Polson. Particle learning of gaussian process models for sequential design and optimization. *Journal of Computational and Graphical Statistics*, 20(1):pp. 102–118, 2011.

R.B. Gramacy and H.K.H. Lee. Adaptive design and analysis of supercomputer experiments. *Technometrics*, 51(2):130–145, 2009.

M. Hamada, HF Martz, CS Reese, and AG Wilson. Finding near-optimal bayesian experimental designs via genetic algorithms. *The American Statistician*, 55(3): 175–181, 2001.

N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization bencharking 2009: Noiseless functions definitions. Technical report, INRIA 2009, 2010.

J.F. Hart, E.W. Cheney, and C.L. Lawson. *Computer Approximations*. Wiley, New York, 1968. 9

M. Hoshiya. Kriging and conditional simulation of gaussian fields. *Journal of engineering mechanics*, 121(2):181–186, 1995. 159

Y. Hung, V. R. Joseph, and S.N. Melkote. Analysis of computer experiments with functional response. 2012. tentatively accepted by Technometrics.

IRSN. Analysis guide - nuclear criticality risks and their prevention in plants and laboratories. Technical report, IRSN DSU/SEC/T/2013-334, 2010. URL http://www.irsn.fr/EN/publications/technical-publications/Documents/IRSN_report_nuclear_criticality_risks.pdf. 59

J. Janusevskis, R. Le Riche, and D. Ginsbourger. Parallel expected improvements for global optimization: summary, bounds and speed-up. August 2011. URL http://hal-emse.ccsd.cnrs.fr/hal-00613971. 26

J. Janusevskis, R. Le Riche, D. Ginsbourger, and R. Girdziusas. Expected improvements for the asynchronous parallel global optimization of expensive functions : Potentials and challenges. In *LION 6 Conference (Learning and Intelligent OptimizatioN), Paris : France*, 2012.

Y. Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1 (2):61–70, 2011. 6

D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21:345–383, 2001. 7

D. R. Jones, M. Schonlau, and J. William. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998. 2, 7, 24, 60

B. Kenkel. pbivnorm : Vectorized bivariate normal cdf, 2011. URL http://cran.r-project.org/web/packages/pbivnorm. 50

S-H. Kim and S-W Na. Response surface method using vector projected sampling points. *Structural Safety*, 19:3–19, 1997.

Ling Li, Julien Bect, and Emmanuel Vazquez. Bayesian Subset Simulation : a kriging-based subset simulation algorithm for the estimation of small probabilities of failure. Technical report, 2012. URL http://publicationslist.org/jbect. Submitted to PSAM 11 and ESREL 2012.

Amandine Marrel, Bertrand Iooss, François Van Dorpe, and Elena Volkova. An efficient methodology for modeling complex computer codes with gaussian processes. *Computational Statistics & Data Analysis*, 52(10):4731–4744, 2008.

G. Matheron. The intrinsic random functions, and their applications. *Advances in Applied Probability*, 5:439–468, 1973. 6

W. Mebane and J. Sekhon. Genetic optimization using derivatives: The rgenoud package for r. *Journal of Statistical Software*, Vol. 42, Issue 11:1–26, 2011. 63

J. Mockus. *Bayesian Approach to Global Optimization. Theory and Applications.* Kluwer Academic Publisher, Dordrecht, 1989. 7

J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. In L. Dixon and Eds G. Szego, editors, *Towards Global Optimization*, volume 2, pages 117–129. Elsevier, 1978. 2, 7, 24

I. Molchanov. *Theory of Random Sets.* Springer, 2005. 46, 47

M.D. Morris and T.J. Mitchell. Exploratory designs for computational experiments. *Journal of statistical planning and inference*, 43(3):381–402, 1995.

A. O'Hagan. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(1):1–42, 1978. 22

M.B. Palacios and M.F.J. Steel. Non-gaussian bayesian geostatistical modeling. *Journal of the American Statistical Association*, 101:604–618, 2006. 6

Rui Paulo, Gonzalo García-Donato, and Jesús Palomo. Calibration of computer models with multivariate output. *Computational Statistics & Data Analysis*, 2012.

V. Picheny. *Improving accuracy and compensating for uncertainty in surrogate modeling*. PhD thesis, École Nationale Supérieure des Mines de Saint-Étienne, 2009. 36, 52

V. Picheny, D. Ginsbourger, O. Roustant, R. T. Haftka, and N.-H. Kim. Adaptive designs of experiments for accurate approximation of target regions. *Journal of Mechanical Design*, 132(7), 2010. 36, 52

P. Ranjan, D. Bingham, and G. Michailidis. Sequential experiment design for contour estimation from complex computer codes. *Technometrics*, 50(4):527–541, 2008. 36, 52

C. R. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. 1, 6, 7

Y. Richet, Y. Deville, and C. Chevalier. DiceView : Plot methods for computer experiments design and surrogate, 2012. URL http://cran.r-project.org/web/packages/DiceView.

Y. Richet, G. Caplin, J. Crevel, D. Ginsbouger, and V. Picheny. Using the efficient global optimization algorithm to assist nuclear criticality safety assessment. *to appear in Nuclear Science and Engineering*, 2013. 60

Brian D Ripley. *Stochastic simulation*, volume 316. Wiley, 2009. 159

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004. 53

J. Rougier. Efficient emulators for multivariate deterministic functions. *Journal of Computational and Graphical Statistics*, 17:4:827–843, 2008.

O. Roustant, D. Ginsbourger, and Y. Deville. Dicekriging, Diceoptim: Two R packages for the analysis of computer experiments by kriging-based metamodelling and optimization. *Journal of Statistical Software*, 51, 2012. URL http://www.jstatsoft.org/v51/i01. 19, 28

R.Y. Rubinstein and D.P. Kroese. *The Cross-Entropy Method*. Springer, 2004.

R.Y. Rubinstein and D.P. Kroese. *Simulation and the Monte Carlo method*. Wiley, 2008.

J. Sacks and S. Schiller. Spatial designs. *Statistical decision theory and related topics IV*, 2(S S):385–399, 1988.

J. Sacks, S.B. Schiller, and W.J. Welch. Designs for computer experiments. *Technometrics*, 31(1):41–47, 1989a.

J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989b. 7

T. J. Santner, B. J. Williams, and W. Notz. *The Design and Analysis of Computer Experiments*. Springer Verlag, 2003. 1, 7

M. Schonlau. *Computer Experiments and global optimization*. PhD thesis, University of Waterloo, 1997. 7, 14, 26, 79

M. L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, New York, 1999. 6, 20, 22

G.M. Tallis. The moment generating function of the truncated multi-normal distribution. *J. Roy. Statist. Soc. Ser. B*, 23(1):223–229, 1961. 13, 79

J. E. Taylor, K.J. Worsley, and F. Gosselin. Maxima of discretely sampled random fields, with an application to bubbles. *Biometrika*, 94(1):1–18, 2007. 80, 167, 168

A. Torn and A. Zilinskas. *Global Optimization*. Springer Verlag, New York, 1989.

E. Vazquez. *Modélisation comportementale de systèmes non-linéaires multivariables par méthodes à noyaux et applications*. PhD thesis, Univ Paris XI, Orsay, France, 2005. 159

E. Vazquez and J. Bect. A sequential Bayesian algorithm to estimate a probability of failure. In *Proceedings of the 15th IFAC Symposium on System Identification, SYSID 2009 15th IFAC Symposium on System Identification, SYSID 2009*, Saint-Malo France, 2009.

E. Vazquez and M. Piera-Martinez. Estimation of the volume of an excursion set of a Gaussian process using intrinsic Kriging, 2006. URL http://arxiv.org/abs/math/0611273.

E. Vazquez and M. Piera-Martinez. Estimation du volume des ensembles d'excursion d'un processus gaussien par krigeage intrinsèque. In *39ème Journées de Statistiques Conférence Journée de Statistiques*, Angers France, 2007.

J. Villemonteix. *Optimisation de fonctions coûteuses*. PhD thesis, Université Paris-Sud XI, Faculté des Sciences d'Orsay, 2008. 30

O. Y. Vorobyev. Mean measure modeling. *Nauka*, 1984. (in Russian). 46

O. Y. Vorobyev and N.A. Lukyanova. A mean probability event for a set of events. *Journal of Siberian Federal University. Mathematics and Physics.*, 6(1):128–136, 2013. 46

G. G. Wang and S. Shan. Review of metamodeling techniques in support of engineering design optimization. *ASME Transactions, Journal of Mechanical Design Optimization*, 129 (4):370–380, 2006. 6

W. J. Welch, R. J. Buck, J. Sacks, H. P. Wynn, T. J. Mitchell, and M. D. Morris. Screening, predicting and computer experiments. *Technometrics*, 34:15–25, 1992. 7

A. M. Yaglom. *Correlation Theory of Stationary and Related Random Functions I: Basic Results*. Springer Series in Statistics. Springer-Verlag, New york, 1986. 20

Y-G Zhao and T. Ono. A general procedure for first/second-order reliability method (form/sorm). *Structural Safety*, 21:95, 1999.

Clément Chevalier

Adress: rue des Pianos 31, 2503 Bienne, Suisse
Tel. : +41 (0)77.431.54.04
E-mail : clement.chevalier1@gmail.com
Age: 31
Nationality: French
Civil status: Married, with one child

WORK EXPERIENCE

Since Nov.2010 — **University of Bern, Switzerland**
Ph.D. student - Lecturer

- Ph.D. in applied mathematics / statistics. Particular focus on Bayesian sequential evaluation strategies of multivariate functions, relying on Gaussian process models.
- Coordination of the ReDice consortium, http://www.redice-project.org/ gathering partners from industry and academia around innovative mathematical methods for the design and analysis of computer experiments.

Nov.2005- Oct.2010 — **BNP Paribas, Paris, France**
Equity Derivative, Product development
Pricing, Structuring & Innovation on Equity Derivative exotic products

- Quick pricing of exotic equity derivative products.
- Animation of internal training seminars on derivative market and market sentiment.

2004 — **Centre de Géostatistique des Mines de Paris, Fontainebleau, France**
Internship- academic research

- Research work on a geostatistical approach of the behavior of a numerical simulator.

EDUCATION

2004-2005 — **Master degree in Statistics, Université Pierre et Marie Curie - Paris VI**
One of the leading French Universities, Master completed with the highest honours

- Fundamental modules: stochastic process, empirical process.
- Applied modules: statistics for insurance, data-mining, design of experiments.

2002-2005 — **Ecole des Mines de Saint Etienne,**
One of the top French Engineering Schools

- Advanced mathematics: numerical methods, optimization, statistics
- Computer science: C, Matlab, object-oriented languages (C++,Java)

COMPETENCES

French	Mother tongue
English	fluent
Spanish	fluent
German	notions
Computer Skills	Proficient in using Excel and coding in R, VBA, C++, Matlab, PL/SQL
R packages	Author and maintainer of the KrigInv package (Kriging-based inversion). Co-author of the DiceView package. Contributor to the DiceKriging and DiceOptim packages.

OTHER ACTIVITIES

Travel, hikes, ski

TEACHING

- Since January 2012: Lecture in Portfolio Management (quantitative finance) at the Ecole des Mines de Saint-Etienne (France).
- University of Bern, fall Semester 2012 and spring semester 2013: assistantship in Statistics in Climate Sciences, with Dr. Michel Piot.
- University of Bern, spring semester 2012: assistantship in Optimization methods, with Dr. David Ginsbourger.
- University of Bern , fall Semester 2011: assistantship in Multivariate Statistics, with Prof. Lutz Dümbgen.

PUBLICATIONS

- C. Chevalier, J. Bect, D. Ginsbourger, E. Vazquez, V. Picheny, and Y. Richet. Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set.
  Status: Accepted with minor revision to Technometrics, (2013).
  URL: http://hal.inria.fr/hal-00641108/en
- C. Chevalier, D. Ginsbourger, and X. Emery. Corrected kriging update formulae for batch-sequential data assimilation. Proceedings of the IAMG2013 conference, Madrid, (2013).
  URL: http://arxiv.org/abs/1203.6452
- C. Chevalier, D. Ginsbouger, J. Bect, and I. Molchanov. Estimating and quantifying uncertainties on level sets using *the vorob'ev expectation and deviation with gaussian process models.* mODa 10, Advances in Model-Oriented Design and Analysis, Physica-Verlag HD, (2013).
- C. Chevalier and D. Ginsbouger. Fast computation of the multi-points expected improvement with applications in batch selection. Proceedings of the LION7 conference, Lecture Notes in Computer Science, (2013).
- C. Chevalier, V. Picheny, and D. Ginsbourger. Kriginv: An efficient and user-friendly implementation of batch sequential inversion strategies based on kriging. Computational Statistics & Data Analysis, (2013).
  doi: http://dx.doi.org/10.1016/j.csda.2013.03.008
- I. Mahlstein, O. Martius, C. Chevalier, and D. Ginsbourger. Changes in the odds of extreme events in the Atlantic basin depending on the position of the extratropical jet. Geophysical research letters, vol. 39, L22805, doi:10.1029/2012GL053993 (2012).

CONFERENCES AND OTHER PRESENTATIONS

- Jul. 2013, SAMO 2013 conference, oral presentation on my Ph.D. for the GDR Mascot Num. Nice, France.
- Jun. 2013, Fourth Ph.D. presentation for the ReDice consortium. Saint-Etienne France.
- Jan. 2013, LION7 conference: Fast computation of the Multi-points Expected Improvement with application in Batch selection. Catania, Italy.
- Jan. 2013, Third Ph.D. presentation for the ReDice consortium. EDF research center, Chatou, France.
- Nov. 2012, presentation for the French nuclear safety institute (IRSN): Optimisation-Inversion sur base de Krigeage. Fontenay-aux-Roses, France.
- Oct. 2012, Ph.D. presentation for the IRSN. Ph.D. student's day. Le Croisic, France.
- Jul. 2012, UCM 2012 conference: Multipoint sampling criteria for the identification of excursion sets. Sheffield, UK.
- Jun. 2012, Second Ph.D. presentation for the ReDice consortium. Bern, Switzerland.

- May 2012, IMSV institute seminar: Quantifying and Reducing Uncertainties on an Excursion Set using Random Set theory and Kriging. Bern, Switzerland.

- Apr. 2012, presentation for the IRSN: Sequential and batch-sequential kriging-based sampling strategies for the identification of an excursion set. Cadarache, France.

- Apr. 2012, SIAM conference on Uncertainty Quantification: Quantifying and Reducing Uncertainties on a set of failure using Random Set theory and Kriging. Railey, USA.

- Mar. 2012, poster presentation on my Ph.D. for the GDR Mascot Num. Bruyères-le-Châtel, France.

- Dec. 2011, ERCIM'11 conference: KrigInv, an R package for sequential inversion of expensive-to-evaluate black-box simulators. London, UK.

- Dec. 2011, First Ph.D. presentation for the ReDice consortium. Fontenay-aux-Roses, France.

- Jul. 2011, World Congress of Global Optimization 2011: Sampling criteria for the sequential inversion of black-box simulators. Chania, Greece.

- Jun. 2011, Presentation for the IRSN: Kriging-based inversion for nuclear safety. Fontenay-aux-Roses, France.

- Jun. 2011: IMSV institute seminar: Kriging-based inversion. Bern, Switzerland

- May 2011, Stochastic seminar: Kriging for large dataset, description of approximation methods. Bern, Switzerland.

- Jan. 2011, Presentation for the IRSN: Optimisation, Inversion et Optimisation-Inversion à l'aide d'un méta-modèle de Krigeage. Fontenay-aux-Roses, France.