Zuse Institute Berlin

Takustrasse 7
D-14195 Berlin-Dahlem
Germany

GUILLAUME SAGNOL, HANS-CHRISTIAN HEGE, MARTIN WEISER

Using sparse kernels to design computer experiments with tunable precision

ZIB Report 16-33 (June 2016)

Zuse Institute Berlin
Takustrasse 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

e-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Using sparse kernels to design computer experiments with tunable precision

Guillaume Sagnol, Hans-Christian Hege, Martin Weiser

June 3, 2016

Abstract

Statistical methods to design computer experiments usually rely on a Gaussian process (GP) surrogate model, and typically aim at selecting design points (combinations of algorithmic and model parameters) that minimize the average prediction variance, or maximize the prediction accuracy for the hyperparameters of the GP surrogate. In many applications, experiments have a *tunable precision*, in the sense that one software parameter controls the tradeoff between accuracy and computing time (e.g., mesh size in FEM simulations or number of Monte-Carlo samples). We formulate the problem of allocating a budget of computing time over a finite set of candidate points for the goals mentioned above. This is a continuous optimization problem, which is moreover convex whenever the tradeoff function accuracy vs. computing time is concave. On the other hand, using non-concave weight functions can help to identify sparse designs. In addition, using sparse kernel approximations drastically reduce the cost per iteration of the multiplicative weights updates that can be used to solve this problem.

Keywords: Optimal design of computer experiments, Gaussian process, Sparse kernels

1 Introduction

We consider a computer code taking an input $\boldsymbol{x} \in \mathcal{X}$ (called a *design point*) in a compact set $\mathcal{X} \subset \mathbb{R}^d$ and a parameter τ specifying the time allowed for the computation, and returning an output of the form

$$Y(\boldsymbol{x}, \tau) = \eta(\boldsymbol{x}) + \epsilon(\boldsymbol{x}, \tau), \tag{1}$$

where $\eta(\cdot)$ is an unknown function and $\epsilon(\boldsymbol{x}, \tau)$ represents uncorrelated errors: $\mathbb{E}[\epsilon(\boldsymbol{x}, \tau)] = 0$, $\boldsymbol{u} \neq \boldsymbol{v} \Rightarrow \mathbb{E}[\epsilon(\boldsymbol{u}, \tau_u)\epsilon(\boldsymbol{v}, \tau_v)] = 0$ for all $\tau_u, \tau_v > 0$, where $\mathbb{E}[X]$ stands for the expectation of X. We assume that the experiments have a *tunable precision*, in the sense that the variance of the error $\mathbb{V}[\epsilon(\boldsymbol{x}, \tau)]$ is a decreasing function of the time τ spent to compute an approximation of $\eta(\boldsymbol{x})$. Specifically, we assume that there is a *known* parameter σ_N^2 (where the subscript N stands for *noise*) and a *known* differentiable, nondecreasing function $w : \mathbb{R}_+ \mapsto \mathbb{R}_+$ satisfying $w(0) = 0$, such that $\mathbb{V}[\epsilon(\boldsymbol{x}, \tau)] = \sigma_N^2 w(\tau)^{-1}$. Let the experimental design (or simply *the design*) be represented by $\xi = \{\boldsymbol{x}_i, \tau_i\}_{i \in \{1,\dots,n\}}$, where the *given candidate points* $\boldsymbol{x}_i \in \mathcal{X}$ are distinct, and $\tau_i \geq 0$ is the computing time spent on design point $\boldsymbol{x}_i$. Note that $\tau_i = 0$ means that no computation is carried out at $\boldsymbol{x}_i$, and formally we have $\mathbb{V}[\epsilon(\boldsymbol{x}_i, \tau_i)] = +\infty$. Candidate points $\boldsymbol{x}_i$ with $\tau_i > 0$ are called *support points* and are those which are actually selected for the design.

We assume that a two-stage approach is used, and observations have already been collected during the initial stage from a design $\xi_{\text{init}} = \{\boldsymbol{x}_i^0, \tau_i^0\}_{i=1,\dots,n_0}$, with $\boldsymbol{x}_i^0 \neq \boldsymbol{x}_j$ for all i, j. The purpose of this article is to develop efficient algorithms for the computation of near-optimal computing times τ_i, subject to a constraint on the total computing time allowed for the second stage: $\sum_{i=1}^n \tau_i = T$. In other words, we search for a (near-)optimal design within the class of all designs that assign a total computing time of T, and whose support is a subset of the $\boldsymbol{x}_i$'s. In practice, this two-stage approach can be turned into a sequential one, as follows. Given an optimized design $\boldsymbol{\tau}^*$, select one support point $\boldsymbol{x}_i$ and compute $Y(\boldsymbol{x}_i, \tau_i^*)$. Then, append $\{\boldsymbol{x}_i, \tau_i^*\}$ to ξ_{init}, decrement T by τ_i^*, remove $\boldsymbol{x}_i$ from the list of candidate points, update the surrogate model for $\eta(\cdot)$, compute the next design $\boldsymbol{\tau}^*$, and iterate. This procedure can also be generalized to work on a parallel architecture, where several design points can be processed simultaneously.

Our assumption about the existence of an information function $w()$ is not common. Most authors focus on the search for *exact designs*, i.e. $w_i = w(\tau_i) \in \{0, 1\}$, and $w_i = 1$ indicates that the design point $\boldsymbol{x}_i$ belongs to ξ. We refer the reader to [PM12] for a comprehensive review on exact designs for computer experiments. For the standard linear model, a popular technique is to relax the integer constraint on w_i, which led to the success story of the *theory of approximate designs* [Páz86, Puk93]. Approximate designs are used most often as a heuristic to find good exact designs, typically by rounding. For computer experiments, however, the total computing time is of much more importance than the number of design points, which motivates to study the tradeoff between tunable accuracy and computing time in more detail. We give two examples:

- In the case of Monte-Carlo simulations, the variance is inversely proportional to the number of samples, and hence $w(\tau) = \tau$. In fact, τ can take integer values only, but we expect the approximation to be good enough for a large number of simulation runs.
- The standard a priori error estimate for finite element solutions of ansatz order p and mesh width h for sufficiently regular stationary elliptic problems in $\Omega \subset \mathbb{R}^d$ is $\|u - u_h\|_{H^1(\Omega)} = \mathcal{O}(h^p)$. The optimal computational complexity is $\tau = \mathcal{O}(h^{-d})$, which means $\tau = \mathcal{O}(\|u - u_h\|_{H^1(\Omega)}^{-d/p})$, see, e.g., [DW12]. We could thus model the error by a noise of variance $\mathbb{V}[\epsilon(\boldsymbol{x}, \tau)] = \mathcal{O}(\tau^{-2p/d})$, i.e. $w(\tau) = \mathcal{O}(\tau^{2p/d})$.

Note, however, that errors at two design points might be correlated, as is usually the case in finite element models. An alternative could be to use a Gaussian process to model the error process. We leave this for future research, and simply assume as an approximation that the $\epsilon(\boldsymbol{x}_i, \tau_i)$ are uncorrelated, which might be acceptable if the $\boldsymbol{x}_i$'s are far enough from each other.

Following the kriging methodology, we assume that $\eta(\boldsymbol{x}) = f(\boldsymbol{x})^T\boldsymbol{\beta} + Z(\boldsymbol{x})$, where $f : \mathbb{R}^d \mapsto \mathbb{R}^m$ is a spatial regression function (generally a polynomial), $\boldsymbol{\beta} \in \mathbb{R}^m$ is an unknown vector of parameters, and Z is a Gaussian random field with zero mean and *known correlation structure*,

$$\mathbb{E}[Z(\boldsymbol{x})] = 0, \qquad \mathbb{E}[Z(\boldsymbol{u})Z(\boldsymbol{v})] = \sigma_Z^2\, C(\boldsymbol{u}, \boldsymbol{v}),$$

where $C : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ is a positive semidefinite kernel satisfying $C(\boldsymbol{u}, \boldsymbol{u}) = 1$ for all $\boldsymbol{u} \in \mathcal{X}$.

Denote by $Y(\xi) = [Y(\boldsymbol{x}_1), \dots, Y(\boldsymbol{x}_n)]^T$ the vector of observations associated with the design ξ. We have

$$Y(\xi) \sim \mathcal{N}\big(\boldsymbol{F}^T\boldsymbol{\beta},\, \sigma_Z^2\boldsymbol{C} + \boldsymbol{\Delta}\big), \tag{2}$$

where $\boldsymbol{F} = [f(\boldsymbol{x}_1), \dots, f(\boldsymbol{x}_n)] \in \mathbb{R}^{m\times n}$, $\{\boldsymbol{C}\}_{ij} := C(\boldsymbol{x_i}, \boldsymbol{x_j})$. and $\boldsymbol{\Delta} = \sigma_N^2\,\mathrm{Diag}(w(\boldsymbol{\tau}))^{-1}$. We also define the *signal-to-noise ratio* $\gamma = (\sigma_Z/\sigma_N)^2$ and $\boldsymbol{\Sigma} = \boldsymbol{C} + \mathrm{Diag}(\gamma w(\boldsymbol{\tau}))^{-1}$, so that $\mathbb{V}[Y(\xi)] = \sigma_Z^2\boldsymbol{\Sigma}$. The best linear unbiased estimator (BLUE) of $\boldsymbol{\beta}$ and its variance are given by

$$\hat{\boldsymbol{\beta}} = \big(\boldsymbol{F}\boldsymbol{\Sigma}^{-1}\boldsymbol{F}^T\big)^{-1}\boldsymbol{F}\boldsymbol{\Sigma}^{-1}Y(\xi), \qquad \mathbb{V}[\hat{\boldsymbol{\beta}}] = \sigma_Z^2\big(\boldsymbol{F}\boldsymbol{\Sigma}^{-1}\boldsymbol{F}^T\big)^{-1}.$$

Then, the best linear unbiased predictor (BLUP) of the unknown function η at $\boldsymbol{x} \in \mathcal{X}$ based on the observations $Y(\xi)$ is given by

$$\hat{\eta}(\boldsymbol{x}|\xi) = f(\boldsymbol{x})^T\hat{\boldsymbol{\beta}} + c(\boldsymbol{x})^T\Sigma^{-1}\big(Y(\xi) - \boldsymbol{F}^T\hat{\boldsymbol{\beta}}\big),$$

where $\{c(\boldsymbol{x})\}_i := C(\boldsymbol{x}, \boldsymbol{x}_i)$ is the vector of cross-covariances between $\boldsymbol{x}$ and the design points, and the mean-squared prediction error (MSPE) is

$$\begin{aligned}\rho(\boldsymbol{x}) &:= \mathbb{E}[\big(\hat{\eta}(\boldsymbol{x}|\xi) - \eta(\boldsymbol{x})\big)^2] \\ &= \sigma_Z^2\left\{1 - c(\boldsymbol{x})^T\boldsymbol{\Sigma}^{-1}c(\boldsymbol{x}) + (f(\boldsymbol{x}) - \boldsymbol{F}\boldsymbol{\Sigma}^{-1}c(\boldsymbol{x}))^T\big(\boldsymbol{F}\boldsymbol{\Sigma}^{-1}\boldsymbol{F}^T\big)^{-1}(f(\boldsymbol{x}) - \boldsymbol{F}\boldsymbol{\Sigma}^{-1}c(\boldsymbol{x}))\right\}.\end{aligned}$$

The above expression reduces to $\rho(\boldsymbol{x}) = \sigma_Z^2\big(1 - c(\boldsymbol{x})^T\boldsymbol{\Sigma}^{-1}c(\boldsymbol{x})\big)$ when the trend parameter $\boldsymbol{\beta}$ is known. Note that $\rho(\boldsymbol{x})$ depend on the design ξ through $\boldsymbol{\Sigma}$. A standard approach is to choose ξ so as to minimize the integrated mean squared error (IMSE):

$$\mathrm{IMSE}(\xi) := \int_{\mathcal{X}} \rho(\boldsymbol{x})\, d\mu(\boldsymbol{x}).$$

The IMSE criterion depends on a measure μ on $\mathcal{X}$, which can be used to weigh the interest of the experimenter for knowing the value of η at $\boldsymbol{x}$. E.g., if the goal is to minimize $\eta(\boldsymbol{x})$ over $\mathcal{X}$, or to estimate the probability that $\eta(\boldsymbol{x})$ lies below some threshold, μ should weigh regions of $\mathcal{X}$ such as to balance the exploration/exploitation tradeoff; see, e.g., [JSW98, BGL+12].

It was shown (for the standard case where the variance of ϵ is not a function of τ) in [Fed96, FF97] that model (1) can be approximated arbitrarily well by a Bayesian linear model of the form

$$Y(\boldsymbol{x}) \simeq [f(\boldsymbol{x})^T, g(\boldsymbol{x})^T]\begin{bmatrix}\boldsymbol{\beta}\\ \boldsymbol{\alpha}\end{bmatrix} + \epsilon(\boldsymbol{x}), \tag{3}$$

where $\boldsymbol{\alpha}$ is a random regression parameter with prior $\boldsymbol{\alpha} \sim \mathcal{N}(0, \sigma_Z^2\boldsymbol{I}_s)$, $\boldsymbol{I}_s$ is the $s \times s-$identity matrix, and the function $g : \mathbb{R}^d \mapsto \mathbb{R}^s$ can be obtained by truncating the Mercer's expansion of the kernel $C(\cdot,\cdot)$. In our case, recall that observations have already been collected during an initial stage at $\xi_{\text{init}} = \{\boldsymbol{x}_i^0, \tau_i^0\}_{i=1,\dots,n_0}$, $\mathbb{V}[\epsilon(\boldsymbol{x},\tau)] = \sigma_N^2 w(\tau)^{-1}$ and the noise is uncorrelated. Then, by using standard results from the literature on Bayesian designs, see, e.g., [Pil91], one obtains the following approximation of the Kriging variance for the design $\xi = \{\boldsymbol{x}_i, \tau_i\}_{i=1,\dots,n}$:

$$\tilde{\rho}(\boldsymbol{x}) \simeq \sigma_Z^2\, h(\boldsymbol{x})^T\boldsymbol{M}(\xi)^{-1}h(\boldsymbol{x}),$$

where $h(\boldsymbol{x}) = [f(\boldsymbol{x})^T, g(\boldsymbol{x})^T]^T$ and $\boldsymbol{M}(\xi)$ is the (scaled) Fisher information matrix for $(\boldsymbol{\beta}, \boldsymbol{\alpha})$:

$$\boldsymbol{M}(\xi) := \sum_{i=1}^{n}\gamma\, w(\tau_i)\begin{bmatrix} f(\boldsymbol{x}_i)\\ g(\boldsymbol{x}_i)\end{bmatrix}\begin{bmatrix} f(\boldsymbol{x}_i)\\ g(\boldsymbol{x}_i)\end{bmatrix}^T + \underbrace{\sum_{i=1}^{n_0}\gamma\, w(\tau_i)\begin{bmatrix} f(\boldsymbol{x}_i^0)\\ g(\boldsymbol{x}_i^0)\end{bmatrix}\begin{bmatrix} f(\boldsymbol{x}_i^0)\\ g(\boldsymbol{x}_i^0)\end{bmatrix}^T + \begin{bmatrix}\boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}_s\end{bmatrix}}_{\boldsymbol{\Gamma}}. \tag{4}$$

Further, the IMSE criterion can be approximated by a criterion of Bayesian A-optimality:

$$\tilde{\mathrm{IMSE}}(\xi) = \int_{\mathcal{X}} \tilde{\rho}(\boldsymbol{x})\, d\mu(\boldsymbol{x}) = \mathrm{trace}\, \boldsymbol{M}(\xi)^{-1}\boldsymbol{L}$$

with a coefficient matrix $\boldsymbol{L} = \sigma_Z^2\int_{\mathcal{X}} h(\boldsymbol{x})h(\boldsymbol{x})^T d\mu(\boldsymbol{x})$. We restrict our attention to the situation in which $\boldsymbol{\beta}$ is estimable from the observations collected with the initial design ξ_{init}, which ensures that $\boldsymbol{\Gamma}$ is positive definite, and $\boldsymbol{M}(\xi)$ is invertible for all designs.

This technique was recently used by [SP10, GP16], who compute approximate designs by using standard algorithms of Bayesian A-optimality, and use rounding heuristics to find exact designs. One

disadvantage is that it requires the knowledge of a Mercer's expansion of the kernel. To tackle this problem, a polar spectral approximation of the kernel has been used [SP10, SP15], but it is not clear whether this can be generalized for parameter spaces of dimension $d > 2$. In [GP16] it is assumed that μ has a finite support containing the candidate points $\boldsymbol{x}_i$, so the computation of the $g(\boldsymbol{x}_i)$ reduce to a standard matrix eigenproblem. In Section 3 we establish a link between this approach and the class of SOR kernels commonly used in machine learning.

In this article, we focus on two classes of sparse kernels, which are commonly used in Machine Learning, and for which there is a simple, finite Mercer's expansion. We will show in Section 3 that using sparse kernels with s inducing points reduce the cost per iteration of the multiplicative weights update algorithm from $\mathcal{O}(n^3)$ to $\mathcal{O}(ns^2)$, when the goal is to compute the weights of a design minimizing the IMSE over n *predefined candidate points*. This reduction is crucial for computer experiments with parameter dimension $d \geq 4$, where a very large number n of candidate points is required to fill the space of parameters. We show further in Section 4 that the same complexity reduction can be achieved for the search of optimal designs for the prediction of hyperparameters of the kernel. However, we point out that the optimization problems we consider are, in general, not convex. Hence, global optimality cannot be guaranteed, but one can implement standard strategies to try and escape local optima. One exception are the sparse IMSE-optimal design problems studied in Section 3, which are convex when the information function $w()$ is concave. Finally, some numerical experiments illustrate our method in Section 5.

The second goal of this article, covered in the next section, is to extend the theory of approximate optimal designs to the situation in which the weights w_i depend on the true design parameters τ_i via an information function $w()$.

2 Approximate designs in presence of an information function

For a design $\xi = \{\boldsymbol{x}_i, \tau_i\}_i$, define $\boldsymbol{M}(\xi) := \sum_{i=1}^n w(\boldsymbol{x}_i, \tau_i) h(\boldsymbol{x}_i) h(\boldsymbol{x}_i)^T + \boldsymbol{\Gamma}$. Throughout this section, we assume that we are given a set $X = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\} \subseteq \mathcal{X}$ of candidate points, and we consider a design problem of the form

$$\min_{\boldsymbol{\tau} \in \Delta_T} \Phi(\xi_{\boldsymbol{\tau}}) := \operatorname{trace} \boldsymbol{M}(\xi_{\boldsymbol{\tau}})^{-1} \boldsymbol{L}, \tag{5}$$

where $\xi_{\boldsymbol{\tau}}$ represents the design $\{\boldsymbol{x}_i, \tau_i\}_{i=1,\dots,n}$. The matrix $\boldsymbol{L}$ is positive semidefinite, and the computing times are constrained in the set $\Delta_T := \{\boldsymbol{\tau} \in \mathbb{R}_+^n : \sum_{i=1}^n \tau_i = T\}$. For the sake of generality, the information function w is allowed to depend on the design point $\boldsymbol{x}_i$. For all $\boldsymbol{x} \in \mathcal{X}$, the function $w(\boldsymbol{x}, \cdot)$ is assumed to be continuously differentiable and nondecreasing on $\mathbb{R}_+$. We restrict our attention to the case of a positive definite $\boldsymbol{\Gamma}$ for the sake of simplicity (so that $\boldsymbol{M}(\xi_{\boldsymbol{\tau}})$ is invertible for all $\boldsymbol{\tau} \in \Delta_T$), but we stress that the results presented here can be extended to the case of a positive semidefinite $\boldsymbol{\Gamma}$.

It is well known that $\Phi(\xi_{\boldsymbol{\tau}})$ is a convex function of the vector of design weights $\boldsymbol{w} = [w(\boldsymbol{x}_1, \tau_1), \dots, w(\boldsymbol{x}_n, \tau_n)]^T$. Using the fact that $\Phi(\xi_{\boldsymbol{\tau}})$ is a nonincreasing function of $w(\boldsymbol{x}_i, \tau_i)$, standard composition theorems yield the following:

Proposition 2.1. *If the information function $\tau \mapsto w(\boldsymbol{x}, \tau)$ is nondecreasing and concave for all $\boldsymbol{x} \in \mathcal{X}$, then the function $\boldsymbol{\tau} \mapsto \Phi(\xi_{\boldsymbol{\tau}})$ is convex over Δ_T.*

The success of the theory of approximate designs is largely due to equivalence theorems such as the Kiefer-Wolfowitz theorem [KW60], that give simple means to check the optimality of a design.

First we state the Karush-Kuhn-Tucker (KKT) necessary optimality conditions for a vector of design weights $\boldsymbol{\tau}$ over Δ_T, which are valid even if the $w(\boldsymbol{x}_i, \cdot)$ are not concave:

Proposition 2.2. *Let $\boldsymbol{L} = \boldsymbol{K}\boldsymbol{K}^T$ and $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ be given candidate points in $\mathcal{X}$. For all $i \in \{1, \dots, n\}$, define*

$$d_i(\boldsymbol{\tau}) = \frac{\partial w(\boldsymbol{x}_i, \tau_i)}{\partial \tau_i} \| h(\boldsymbol{x}_i)^T \boldsymbol{M}(\xi_{\boldsymbol{\tau}})^{-1} \boldsymbol{K} \|^2.$$

If $\boldsymbol{\tau}$ is a local minimizer of $\Phi(\xi_{\boldsymbol{\tau}})$ over Δ_T, then we have: $\forall i \in \{1,\dots,n\},\ d_i(\boldsymbol{\tau}) \le \frac{1}{T}\sum_{k=1}^n \tau_k d_k(\boldsymbol{\tau})$. Moreover, the inequality becomes an equality for support points of $\xi_{\boldsymbol{\tau}}$, i.e. for all i such that $\tau_i > 0$.

Proof. Observe that $\frac{\partial \Phi(\xi_{\boldsymbol{\tau}})}{\partial \tau_i} = -\frac{\partial w(\boldsymbol{x}_i,\tau_i)}{\partial \tau_i} \operatorname{trace} \boldsymbol{M}(\xi_{\boldsymbol{\tau}})^{-1} h(\boldsymbol{x}) h(\boldsymbol{x})^T \boldsymbol{M}(\xi_{\boldsymbol{\tau}})^{-1} \boldsymbol{L} = -d_i(\boldsymbol{\tau})$. Then, the dual feasibility and complementary slackness KKT-conditions of the optimization problem $\min\{\Phi(\xi_{\boldsymbol{\tau}}) : \forall i \in \{1,\dots,n\}, \tau_i \ge 0, \sum_i \tau_i = T\}$ can be expressed as follows:

$$\exists \lambda \ge 0 : \forall i \in \{1,\dots,n\}, \quad \Big((\tau_i = 0 \ \text{ and } \ d_i(\boldsymbol{\tau}) \le \lambda) \quad \text{or} \quad (\tau_i \ge 0 \ \text{ and } \ d_i(\boldsymbol{\tau}) = \lambda) \Big).$$

Moreover, the Lagrange multiplier λ must satisfy $T\lambda = \sum_i \tau_i \lambda = \sum_i \tau_i d_i(\boldsymbol{\tau})$. Substituting the value of λ in the KKT conditions yields the proposition. □

If the information functions are concave, we obtain a much stronger result. For the next theorem we temporarily drop the assumption that the $\boldsymbol{x}_i$'s are given. We characterize optimal designs over the set $\Xi = \Big\{\xi = \{\boldsymbol{x}_i, \tau_i\}_{i=1,\dots,n} : n \in \mathbb{N},\ \forall i \in \{1,\dots,n\},\ \boldsymbol{x}_i \in \mathcal{X}, \tau_i \ge 0,\ \sum_i \tau_i = T\Big\}$ of all designs with support points in $\mathcal{X}$:

Theorem 2.3. *Let $\boldsymbol{L} = \boldsymbol{K}\boldsymbol{K}^T$, and assume that the condition of Proposition 2.1 is satisfied. For all $\boldsymbol{x} \in \mathcal{X}$ define*

$$d(\xi; \boldsymbol{x}) = \frac{\partial w(\boldsymbol{x}, \tau(\boldsymbol{x}))}{\partial \tau} \| h(\boldsymbol{x})^T \boldsymbol{M}(\xi)^{-1} \boldsymbol{K} \|^2,$$

where $\tau(\boldsymbol{x})$ is the computing time spent on the design point $\boldsymbol{x}$ (i.e., $\tau(\boldsymbol{x}) = 0$ if $\boldsymbol{x} \notin \operatorname{supp}(\xi)$ and $\tau(\boldsymbol{x}) = \tau_i$ if $\boldsymbol{x} = \boldsymbol{x}_i \in \operatorname{supp}(\xi)$). Then, $\xi^* = \{\boldsymbol{x}_i, \tau_i\}_{i=1,\dots,n}$ minimizes Φ over Ξ if and only if

$$\forall \boldsymbol{x} \in \mathcal{X},\ d(\xi^*; \boldsymbol{x}) \le \frac{1}{T} \sum_{i=1}^n \tau_i\, d(\xi^*, \boldsymbol{x}_i).$$

Moreover the above inequality becomes an equality for all support points of ξ^*.

Proof. First note that the *only if* part of the theorem is a simple consequence of Proposition 2.2. For the *if* part, assume that the condition of the theorem holds. It implies

$$\forall \boldsymbol{\tau'} \in \mathbb{R}_+^n \quad \text{such that} \quad \sum_i \tau_i' = T, \qquad \sum_{i=1}^n \tau_i'\, d(\xi^*, \boldsymbol{x}_i) \le \sum_{i=1}^n \tau_i\, d(\xi^*, \boldsymbol{x}_i).$$

Using the fact that $\frac{\partial \Phi(\xi^*)}{\partial \tau_i} = -d(\xi^*, \boldsymbol{x}_i)$, this can be rewritten as:

$$\forall \xi' = \{\boldsymbol{x}_i, \tau_i'\} \in \Xi, \quad \left. \frac{\partial \Phi((1-\alpha)\xi^* + \alpha\xi')}{\partial \alpha} \right|_{\alpha=0} \ge 0.$$

Finally, consider an arbitrary design $\xi' \in \Xi$, and define the function $\psi : \alpha \mapsto \Phi((1-\alpha)\xi^* + \alpha\xi')$. By proposition 2.1, ψ is convex on $[0,1]$, and we know that $\psi'(0) \ge 0$. So we have $\Phi(\xi') = \psi(1) \ge \psi(0) + \psi'(0)(1-0) \ge \psi(0) = \Phi(\xi^*)$, which proves the optimality of ξ^*. □

We next adapt the multiplicative weights update algorithm of Titterington [STT78] for Problem (5). The multiplicative algorithm was originally presented in the general setting in which a function f must be minimized over a unit simplex $\{\boldsymbol{w} \in \mathbb{R}_n^+ : \sum_i w_i = 1\}$, so it can be adapted in a straightforward manner to the case of a design problem with information functions, i.e., $w_i = w(\boldsymbol{x}_i, \tau_i)$. Given an exponent $q > 0$ and an initial vector $\boldsymbol{\tau}^{(0)} > \boldsymbol{0}$, the iterations are:

$$\forall i \in \{1,\dots,n\}, \quad \tau_i^{(k+1)} \leftarrow T \frac{\tau_i^{(k)}\ d_i(\boldsymbol{\tau}^{(k)})^q}{\sum_{j=1}^n \tau_j^{(k)}\ d_j(\boldsymbol{\tau}^{(k)})^q}. \tag{6}$$

In its standard version, that is, when $w(\boldsymbol{x}_i, \tau_i) = \tau_i$, this algorithm converges monotonically towards an $A-$optimal design when $q = \frac{1}{2}$. The process can be accelerated by pruning candidate points with a sufficiently low weight, which ensures that they do not belong to the support of any optimal design [Pro13]. Convergence for a variety of optimality criteria was shown in [Yu10].

Consider now the general setting of a function f to be minimized over Δ_T, with $d_i(\boldsymbol{\tau}) = \frac{\partial f(\boldsymbol{\tau})}{\partial \tau_i}$. If the iterations (6) converge, then the limit point $\boldsymbol{\tau}^*$ must satisfy the necessary condition of Proposition 2.2, under some mild conditions [GM92]. In practice, we experienced numerical convergence of the above algorithm towards *local minima* of $f : \boldsymbol{\tau} \mapsto \Phi(\xi_{\boldsymbol{\tau}})$ when q is well chosen, even in the cases where the information functions $w(\boldsymbol{x}_i, \cdot)$ are not concave.

3 Sparse covariance kernels

Here we consider two classes of sparse kernel functions commonly used in machine learning for Gaussian process regression with a large number n of samples. These approximations rely on a small set of *inducing points*, $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_s\} \subset \mathcal{X}$, and assume that the covariance $\mathrm{cov}(Z(\boldsymbol{x}), Z(\boldsymbol{y}))$ of the process between the points $\boldsymbol{x}$ and $\boldsymbol{y} \in \mathcal{X}$ only depends on the covariances between the $\boldsymbol{u}_i$, $\boldsymbol{x}$ and the $\boldsymbol{u}_i$, and $\boldsymbol{y}$ and the $\boldsymbol{u}_i$. This reduces the complexity of training a Gaussian process on a dataset with n samples from $\mathcal{O}(n^3)$ to $\mathcal{O}(ns^2)$. We refer to [QCR05] for a comprehensive review.

SOR-kernels. The *Subset of Regressors* (SOR) approximation consists in replacing the correlation function $C(\boldsymbol{x}, \boldsymbol{y})$ by a low-rank kernel,

$$C_{\mathrm{SOR}}(\boldsymbol{x}, \boldsymbol{y}) = c_u(\boldsymbol{x})^T \boldsymbol{K}_{u,u}^{-1} c_u(\boldsymbol{y}),$$

where $\{\boldsymbol{K}_{u,u}\}_{i,j} = C(\boldsymbol{u}_i, \boldsymbol{u}_j)$ is the $s \times s$ matrix of correlations between the $Z(\boldsymbol{u}_i)$, and $\{c_u(\boldsymbol{x})\}_i = C(\boldsymbol{u}_i, \boldsymbol{x})$ is the s-dimensional vector of correlations between $Z(\boldsymbol{x})$ and the $Z(\boldsymbol{u}_i)$. Hence, if we let $\boldsymbol{J}_u$ be any matrix satisfying $\boldsymbol{J}_u \boldsymbol{J}_u^T = \boldsymbol{K}_{u,u}^{-1}$, then the function $g : \boldsymbol{x} \mapsto \boldsymbol{J}_u^T c_u(\boldsymbol{x})$ satisfies

$$\forall (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{X}, \quad C_{\mathrm{SOR}}(\boldsymbol{x}, \boldsymbol{y}) = g(\boldsymbol{x})^T g(\boldsymbol{y}).$$

Hence, for a SOR-kernel the observation model (1) is equivalent to model (3). Indeed, $Z(\xi) = [Z(\boldsymbol{x}_1), \dots, Z(\boldsymbol{x}_n)]^T \sim \mathcal{N}(0, \sigma_Z^2 \boldsymbol{C})$ has the same distribution as $[g(\boldsymbol{x}_1), \dots, g(\boldsymbol{x}_n)]^T \boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \sim \mathcal{N}(0, \sigma_Z^2 \boldsymbol{I}_s)$. To put it in other words, if we assume that the true kernel is C_{SOR}, then model (3) is exact, so that $\mathrm{IMSE}(\xi_{\boldsymbol{\tau}}) = \widetilde{\mathrm{IMSE}}(\xi_{\boldsymbol{\tau}}) = \mathrm{trace}\, \boldsymbol{M}(\xi_{\boldsymbol{\tau}})^{-1} \boldsymbol{L}$, and we can use the multiplicative weights update (6) to compute an optimal design. The complexity of computing $\mathrm{IMSE}(\xi_{\boldsymbol{\tau}})$ and its gradient $[d_1(\boldsymbol{\tau}), \dots, d_n(\boldsymbol{\tau})]^T$ is $\mathcal{O}(n(s+m)^2)$, which is $\mathcal{O}(ns^2)$ because the dimension m of the regression parameter $\boldsymbol{\beta}$ is a small constant. So the cost of one iteration (6) is $\mathcal{O}(ns^2)$. In contrast, for a full kernel the computation involves $\boldsymbol{\Sigma}^{-1}$ and takes $\mathcal{O}(n^3)$ operations.

There is a vast literature on the selection of inducing points $\boldsymbol{u}_i$ to approximate a kernel $C(\cdot, \cdot)$ by a SOR kernel [Tit09, CBFH15, WN15]. For example, [WN15] uses a regular grid to exploit the Kronecker structure of $\boldsymbol{K}_{u,u}$ when C is a product of one-dimensional kernels, and to speed-up the computations by using fast Fourier transforms. Note that if the points $\boldsymbol{u}_1, \dots, \boldsymbol{u}_s$ are sampled randomly and independently from the probability measure μ, the approximation $C(\boldsymbol{x}, \boldsymbol{y}) \simeq C_{\mathrm{SOR}}(\boldsymbol{x}, \boldsymbol{y})$ can be regarded as an expansion of the form $C(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{s} \lambda_i \phi_i(\boldsymbol{x}) \phi_i(\boldsymbol{y})$, where the λ_i and $\phi_i(\cdot)$ are the solutions of the Nyström approximation of the eigenproblem

$$\int_{\mathcal{X}} C(\boldsymbol{x}, \boldsymbol{y}) \phi(\boldsymbol{y}) d\mu(\boldsymbol{y}) = \lambda \phi(\boldsymbol{x}).$$

This approximation consists in replacing the integral by $\frac{1}{s}\sum_{i=1}^{s} C(\boldsymbol{x},\boldsymbol{u}_i)\phi(\boldsymbol{u}_i)$, and reduces the infinite-dimensional eigenproblem to a standard $s\times s-$matrix eigenproblem [WS01]. If we choose $\boldsymbol{J}_u = \boldsymbol{U}\boldsymbol{\Lambda}^{-\frac{1}{2}}$, where $\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$ is a spectral decomposition of $\boldsymbol{K}_{u,u}$, this is equivalent to the approach of [GP14, GP16], where μ is approximated by a discrete measure $\hat{\mu}$ supported by the $\boldsymbol{u}_i$'s (i.e., the IMSE is approximated by a quadrature). In [GP14], the authors further suggest to choose the candidates $\boldsymbol{x}_i$ in the support of $\hat{\mu}$, i.e. $n \leq s$. Then, g must only be evaluated at the $\boldsymbol{u}_i$'s, and the vectors $g(\boldsymbol{u}_i)$ $(i = 1,\dots,s)$ are the columns of $\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{U}^T$, hence they are orthogonal. This contrasts with our approach, where we generate a large number of candidate points in order to fill the design space, but use a small number of inducing points for the sake of computation ($n \gg s$).

FITC-kernels. The FITC approximation (*Fully Independent Training Conditional*) is very similar to SOR, but a diagonal noise is added to the kernel, to ensure that $C_{\mathrm{FITC}}(\boldsymbol{x},\boldsymbol{x}) = C(\boldsymbol{x},\boldsymbol{x}) = 1$:

$$C_{\mathrm{FITC}}(\boldsymbol{x}_i,\boldsymbol{x}_j) = C_{\mathrm{SOR}}(\boldsymbol{x}_i,\boldsymbol{x}_j) + \big(1 - C_{\mathrm{SOR}}(\boldsymbol{x}_i,\boldsymbol{x}_i)\big)\delta_{ij},$$

If we define as before the function $g : \boldsymbol{x} \mapsto \boldsymbol{J}_u^T c_u(\boldsymbol{x})$, where $\boldsymbol{J}_u\boldsymbol{J}_u^T = \boldsymbol{K}_{u,u}^{-1}$, we obtain:

$$C_{\mathrm{FITC}}(\boldsymbol{x}_i,\boldsymbol{x}_j) = g(\boldsymbol{x}_i)^T g(\boldsymbol{x}_j) + (1 - \|g(\boldsymbol{x}_i)\|^2)\delta_{ij}.$$

It follows that for a FITC kernel, the observation model is equivalent to

$$Y(\boldsymbol{x},\tau) = [f(\boldsymbol{x})^T, g(\boldsymbol{x})^T]\begin{bmatrix}\boldsymbol{\beta}\\ \boldsymbol{\alpha}\end{bmatrix} + \nu(\boldsymbol{x}) + \epsilon(\boldsymbol{x},\tau), \tag{7}$$

where $\boldsymbol{\alpha}$ is a random regression parameter with prior $\boldsymbol{\alpha} \sim \mathcal{N}(0,\sigma_Z^2\boldsymbol{I}_k)$, and $\nu(\boldsymbol{x})$ is an unbiased and uncorrelated noise, which is heteroschedastic with $\mathbb{V}[\nu(\boldsymbol{x})] = \sigma_Z^2(1 - \|g(\boldsymbol{x})\|^2)$. Under this model, the Fisher information matrix for $(\boldsymbol{\beta},\boldsymbol{\alpha})$ becomes (up to a scaling factor σ_Z^2):

$$\boldsymbol{M}(\xi) := \sum_{i=1}^{n} w(\boldsymbol{x}_i,\tau_i)\, h(\boldsymbol{x}_i)h(\boldsymbol{x}_i)^T + \boldsymbol{\Gamma}, \qquad \text{where } \; w(\boldsymbol{x},\tau) := \frac{\gamma w(\tau)}{1 + (1 - \|g(\boldsymbol{x})\|^2)\gamma w(\tau)}. \tag{8}$$

Note that $\boldsymbol{M}(\xi)$ has the form of the Fisher information matrix of the problem studied in Section 2. Moreover, elementary calculus shows that $\tau \mapsto w(\boldsymbol{x},\tau)$ is concave if $\tau \mapsto w(\tau)$ is concave. In this situation, Proposition 2.1 shows that (5) is a convex optimization problem for FITC kernels.

4 Optimal designs for the estimation of kernel hyperparameters

Until now, we have assumed that the kernel function $C(\cdot,\cdot)$ was known. In practice however, the kernel depends on a set of hyperparameters $\boldsymbol{\theta} \in \mathbb{R}^p$, which must be estimated by maximum likelihood from the set of observations $Y(\xi)$. Recall that $Y(\xi) \sim \mathcal{N}\big(\boldsymbol{F}^T\boldsymbol{\beta},\, \sigma_Z^2\boldsymbol{\Sigma}_\theta\big)$, where $\boldsymbol{\Sigma}_\theta = \boldsymbol{C}_\theta + \boldsymbol{D}^{-1}$, $\boldsymbol{D} = \mathrm{Diag}(\gamma w(\boldsymbol{\tau}))$, and we have inserted the symbol θ as subscript to stress the dependency on the hyperparameters. Then, the $p \times p$ Fisher information matrix for the vector of parameters $\boldsymbol{\theta}$ can be derived from standard formulas:

$$\{\boldsymbol{M}_\theta(\xi)\}_{ij} = \frac{1}{2}\,\mathrm{trace}\,\boldsymbol{\Sigma}_\theta^{-1}\,\frac{\partial \boldsymbol{C}_\theta}{\partial\theta_i}\,\boldsymbol{\Sigma}_\theta^{-1}\,\frac{\partial \boldsymbol{C}_\theta}{\partial\theta_j}. \tag{9}$$

Given a current estimate of $\boldsymbol{\theta}$, we propose to search a design ξ maximizing the criterion of $D-$optimality, $\log\det \boldsymbol{M}_\theta(\xi)$. Here, note that we assume that $\boldsymbol{\beta}$ and σ_Z^2 are known. We refer to [PM12] for a review on approaches to deal with a total Fisher information matrix for the set of parameters $(\boldsymbol{\beta},\sigma_Z^2,\boldsymbol{\theta})$. In particular, Müller and Stehlík proposed a compound criterion with a weighing factor that balances the goals of estimating $\boldsymbol{\beta}$ and estimating $\boldsymbol{\theta}$ [MS10].

We want to optimize the computing times τ_i associated with a large number of candidate points $\boldsymbol{x}_i$. This is a hard optimization problem, since here the $D-$criterion is not convex with respect to $\boldsymbol{\tau}$. Nevertheless we propose to use the multiplicative update iterations (6), where $d_i(\boldsymbol{\tau}) := \frac{\partial \log\det \boldsymbol{M}_\theta(\xi_{\boldsymbol{\tau}})}{\partial \tau_i}$, in order to identify good local optima. However, if n is very large, the computation of $\boldsymbol{M}_\theta(\xi_{\boldsymbol{\tau}})$ and $d_i(\boldsymbol{\tau})$ is extremely time-consuming. It involves the inversion of the $n \times n$ matrix $\boldsymbol{\Sigma}_\theta$, and many $n \times n$ matrix-matrix multiplications.

In this section we show that this computational burden can be reduced if sparse kernels are used. From now on, we assume that $\boldsymbol{C}_\theta = \boldsymbol{G}_\theta \boldsymbol{G}_\theta^T$, where $\boldsymbol{G}_\theta = [g_\theta(\boldsymbol{x}_1), \dots, g_\theta(\boldsymbol{x}_n)]^T \in \mathbb{R}^{n\times s}$. As is previous section, the function g_θ is defined by $g_\theta(\boldsymbol{x}) = \boldsymbol{J}_u^T c_u(\boldsymbol{x})$, where $\boldsymbol{J}_u \boldsymbol{J}_u^T = \boldsymbol{K}_{u,u}^{-1}$. From now on we set $\boldsymbol{J}_u$ to the Cholesky factor of $\boldsymbol{K}_{u,u}^{-1}$, because this choice yields compact formulas.

First note that the low-rank decomposition makes it possible to use the Woodbury matrix identity:

$$\boldsymbol{\Sigma}_\theta^{-1} = (\boldsymbol{C}_\theta + \boldsymbol{D}^{-1})^{-1} = \boldsymbol{D} - \boldsymbol{D}\boldsymbol{G}_\theta(\boldsymbol{I}_s + \boldsymbol{G}_\theta^T \boldsymbol{D}\boldsymbol{G}_\theta)^{-1}\boldsymbol{G}_\theta^T \boldsymbol{D}. \tag{10}$$

Then, we also need to compute derivatives of g_θ with respect to θ. This is possible thanks to the following lemma:

Lemma 4.1. *Define the function Φ which returns the lower triangle and half the diagonal of a square matrix:*

$$\forall \boldsymbol{M} \in \mathbb{R}^{n\times n}, \quad \{\Phi(\boldsymbol{M})\}_{ij} = \begin{cases} M_{ij} & \text{if } i > j \\ \frac{1}{2} M_{ij} & \text{if } i = j \\ 0 & \text{if } i < j. \end{cases}$$

Then, we have: $\forall \boldsymbol{x} \in \mathcal{X}$,

$$\frac{\partial g_\theta(\boldsymbol{x})}{\partial \theta_i} = \boldsymbol{J}_u^T \frac{\partial c_u(\boldsymbol{x})}{\partial \theta_i} - \Phi\left(\boldsymbol{J}_u^T \frac{\partial \boldsymbol{K}_{u,u}}{\partial \theta_i} \boldsymbol{J}_u\right)^T \boldsymbol{J}_u^T c_u(\boldsymbol{x}).$$

Proof. A formula for the derivative of the Cholesky decomposition $\boldsymbol{X} = \boldsymbol{J}\boldsymbol{J}^T$ can be found in [Sär13, Theorem A.1], and can be proved by implicit differentiation:

$$\frac{\partial \boldsymbol{J}}{\partial \theta} = \boldsymbol{J}\Phi(\boldsymbol{J}^{-1}\frac{\partial \boldsymbol{X}}{\partial \theta}\boldsymbol{J}^{-T}).$$

The formula of the lemma can now be obtained, by applying standard formulas for the differentiation of products and matrix inverse. □

We can use this lemma to compute the matrices $\boldsymbol{G}_i := \frac{\partial \boldsymbol{G}_\theta}{\partial \theta_i}$. Now, we also define $\boldsymbol{G}_0 := \boldsymbol{G}_\theta$ to simplify the notation. Substituting $\frac{\partial \boldsymbol{C}_\theta}{\partial \theta_i} = \boldsymbol{G}_i \boldsymbol{G}_0^T + \boldsymbol{G}_0 \boldsymbol{G}_i^T$ and (10) into (9) yields an expression for $\{\boldsymbol{M}_\theta(\xi)\}_{ij}$ that depends on $\boldsymbol{G}_0, \boldsymbol{G}_1, \dots, \boldsymbol{G}_p$. After some simplifications, we obtain

$$\{\boldsymbol{M}_\theta(\xi)\}_{ij} = \operatorname{trace} \boldsymbol{A}_{0i}\boldsymbol{A}_{0j} + \boldsymbol{A}_{00}\boldsymbol{A}_{ij},$$

where for all $k, l \in \{0, \dots, p\}$, $\boldsymbol{A}_{kl} := \boldsymbol{B}_{kl} - \boldsymbol{B}_{k0}(\boldsymbol{I}_s + \boldsymbol{B}_{00})^{-1}\boldsymbol{B}_{0j}$ and $\boldsymbol{B}_{kl} := \boldsymbol{G}_k^T \boldsymbol{D}\boldsymbol{G}_l$. From these expressions, it is easy to see that $\boldsymbol{M}_\theta(\xi)$ can be computed in $\mathcal{O}(ns^2)$, which is a great improvement compared to $\mathcal{O}(n^3)$ for a full kernel.

Similarly, we can compute $\boldsymbol{\nabla}_\tau \{\boldsymbol{M}_\theta(\xi)\}_{ij} = \left[\frac{\partial \{\boldsymbol{M}_\theta(\xi)\}_{ij}}{\tau_1}, \dots, \frac{\partial \{\boldsymbol{M}_\theta(\xi)\}_{ij}}{\tau_n}\right]^T$ in $\mathcal{O}(ns^2)$. For all $k \in \{0, \dots, p\}$, define $\boldsymbol{P}_k := \boldsymbol{G}_k - \boldsymbol{G}_0(\boldsymbol{I}_s + \boldsymbol{B}_{00})^{-1}\boldsymbol{B}_{0k}$. Then, we can show that (details omitted):

$$\boldsymbol{\nabla}_\tau \{\boldsymbol{M}_\theta(\xi)\}_{ij} = \gamma \operatorname{Diag}(w'(\boldsymbol{\tau})) \operatorname{Diag}\left(\boldsymbol{P}_i \boldsymbol{A}_{0j} \boldsymbol{P}_0^T + \boldsymbol{P}_j \boldsymbol{A}_{0i} \boldsymbol{P}_0^T + \boldsymbol{P}_0 \boldsymbol{A}_{ij} \boldsymbol{P}_0^T + \boldsymbol{P}_j \boldsymbol{A}_{00} \boldsymbol{P}_i^T\right).$$

Finally, the gradient of the criterion is obtained by $d_i(\xi) = \operatorname{trace} \boldsymbol{M}_\theta(\xi)^{-1} \frac{\partial \boldsymbol{M}_\theta(\xi)}{\partial \tau_i}$. Hence, we have shown that the gradient of the criterion can be computed in $\mathcal{O}(ns^2)$ for a sparse kernel with s inducing points. In contrast, for a full kernel one requires $\mathcal{O}(n^3)$ operations.

5 Numerical Experiments

We consider the Ishigami-like function η to illustrate the effect of the information functions:

$$\forall \boldsymbol{x} \in \mathcal{X} = [0,1] \times [0,1], \qquad \eta(\boldsymbol{x}) = 1.1 \sin(\pi(2x_1 - 1)) + 7 \sin^2(\pi(2x_2 - 1)).$$

First, 8 noisy observations of the function $\eta(\boldsymbol{x})$ are taken, with $\sigma_N^2 = 0.05$, and $\tau_{init} = \frac{1}{8}$ at each of the 8 locations indicated by yellow dots in Figure 1. These initial values are used to estimate, by maximum likelihood, σ_Z^2 and the hyperparameters (ℓ_1, ℓ_2) of the Gaussian kernel

$$C(\boldsymbol{x}, \boldsymbol{y}) = e^{-\frac{1}{2}\left[\left(\frac{x_1 - y_1}{\ell_1}\right)^2 + \left(\frac{x_2 - y_2}{\ell_2}\right)^2\right]}$$

The plots of Figure 1 show some designs for the distribution of $T = 1$ additional hour of computing time over a regular grid of $n = 31^2 = 961$ candidate points. The size of the red dots indicate the time to spend on a design point, and the color in the background indicates the prior Kriging variance (after the initial 8 observations; blue: small variance, red: high variance), according to the considered covariance model: In Plot (a) and (b), we respectively used the SOR and FITC Kernel associated with $C(\cdot, \cdot)$, for $s = 12$ inducing points (marked with black squares), that were generated in a space-filling fashion with a Sobol sequence; the number $s = 12$ is rather small, on purpose, to illustrate the effect of sparsity. Plots (c)-(g) rely on a SOR kernel with $s = 60$ inducing points (not marked for the sake of visibility); with that many inducing points, the relative errors between IMSE(ξ) and $\widetilde{\text{IMSE}}(\xi)$ were in the order of $0,1\%$ for the designs we computed. Also, different information functions were used, cf. Plot (h).

The plots (a),(b),(c),(e),(g) show (near-)optimal weights τ_i for the IMSE criterion at the specified n locations of the $\boldsymbol{x}_i$'s, while (d),(f) are nearly $D-$optimal weights for the estimation of $\boldsymbol{\theta} = (\ell_1, \ell_2)$. For all computations, the matrix $\boldsymbol{L} = \sigma_Z^2 \int_{\mathcal{X}} h(\boldsymbol{x}) h(\boldsymbol{x})^T d\mu(\boldsymbol{x})$ was computed with a Monte-Carlo method with $N = 10^5$ samples, with μ the Lebesgue measure over $[0,1]^2$. The stopping criterion for the multiplicative update iterations was

$$\max_{i=1,\ldots,n} d_i(\boldsymbol{\tau}) \leq \frac{1}{T} \sum_{k=1}^{n} \tau_k d_k(\boldsymbol{\tau}) + \varepsilon, \tag{11}$$

where $\varepsilon = 10^{-9}$. Note that the design weights plotted in (a)-(c) and (g) are provably optimal (up to the tolerance ε), because the considered optimization problems are convex. This is not the case for the designs shown in Plots (d), (e) and (f). Here, the multiplicative update algorithm is likely to fall in local optima, so we performed several restarts and kept the best design.

Plots (a) and (b) show the effect of using a sparse kernel, and are to be compared with Plot (c), which can be considered as the optimal design for the full kernel C when $w(\tau) = \tau$. Observe that the kriging variance tends to be underestimated with the SOR kernel (a), while it is overestimated with the FITC kernel (b). As a consequence of the (strict) concavity of $w(\boldsymbol{x}, \tau)$, the FITC design is more spread out than the SOR design. Also, the FITC design has a slightly better efficiency than the SOR design (73% vs. 69%, cf. Formula (12)).

The effect of the information function can be seen by comparing the second column (standard case $w(\tau) = \tau$) to the third and fourth columns. In the second column, the information function is convex near 0, so that we need some minimal amount of computing time to get some information; see blue curve in plot (f). As a consequence, the IMSE-optimal design for this information function are very sparse, a feature that can be very valuable for the experimenter. In contrast, a concave information function is used in the third column (green curve in plot (f)). This incentivizes designs with many design points spread out over regions with a high variance.

Next, we show some results in higher dimensional spaces to illustrate the importance of using sparse approximations of the kernel. We report results for tests in dimension $d = 4$ and $d = 7$. In each case,

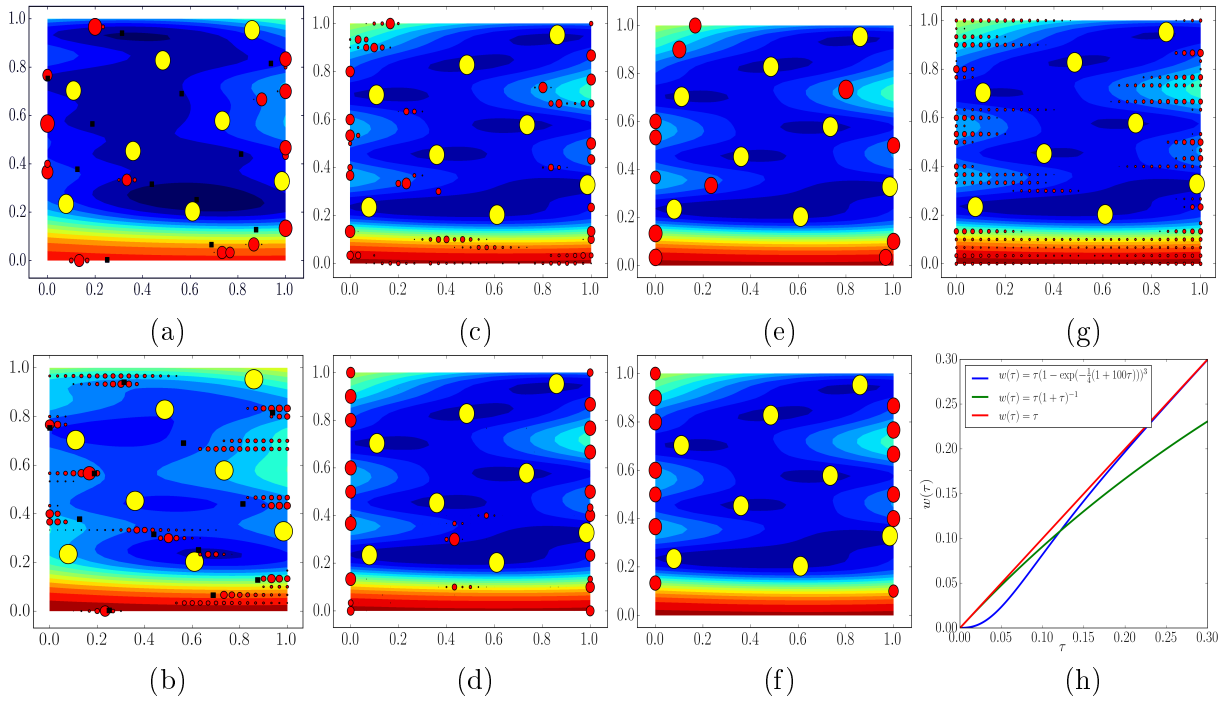

Figure 1: Near-optimal design weights for the test function. Kernel: C_{SOR} with $s = 12$ inducing points in (a); C_{FITC} with $s = 12$ inducing points in (b), and C_{SOR} with $s = 60$ inducing points in (c)-(g). Optimality criterion: IMSE in (a),(b),(c),(e),(g) and D-criterion for $\boldsymbol{\theta} = (\ell_1, \ell_2)$ in (d),(f). Information function: $w(\tau) = \tau$ in (a)-(d); $w(\tau) = \tau(1 - \exp(-\frac{1}{4}(1 + 100\tau)))^3$ in (e),(f); $w(\tau) = \tau(1 + \tau)^{-1}$ in (g). These information functions are plotted in (h).

we considered 10 instances, corresponding to different functions $\eta(\cdot)$; These functions were dummy rational functions, in which we have selected the coefficients at random.

Our experiments used $T_{\text{init}} = 1$ hour of computing time distributed uniformly over $n_{\text{init}} = 50$ initial observations, and aimed at distributing $T = 1$ additional hour of computing time over $n = 1500$ randomly generated candidate points for the problems in dimension 4, and $n = 5300$ points for the problem in dimension 7. The function $w()$ was set to the identity: $w(\tau) = \tau$.

For each problem, we have computed the *true* optimal design ξ^* (within the subset of designs supported over the given candidate points), by using the multiplicative update iterations (6) with a formula for the derivative of the true criterion: $d_i(\xi) = \frac{\partial \text{IMSE}(\xi)}{\partial \tau_i}$. The efficiency of a design was evaluated by the following formula:

$$\text{efficiency}(\xi) = \frac{\text{IMSE}(\xi)^{-1} - \text{IMSE}(\xi_{\text{init}})^{-1}}{\text{IMSE}(\xi^*)^{-1} - \text{IMSE}(\xi_{\text{init}})^{-1}}. \tag{12}$$

Here, ξ_{init} denotes the initial design supported by the n_{init} initial observation points, so the numerator expresses the gain of information provided by ξ, compared to the situation where no additional measurement is performed. Figure 2 shows the efficiency of designs computed by using a SOR approximation of the kernel, for 10 instances with $d = 4$ (left) and $d = 7$ (right). In both cases, we observe an excellent efficiency when $s \geq 100$, and even for $s \geq 70$ for the instances in a 4-dimensional space. In terms of computing time, the speed-up was on the order of x200 on average for $s = 70$ and $d = 4$,

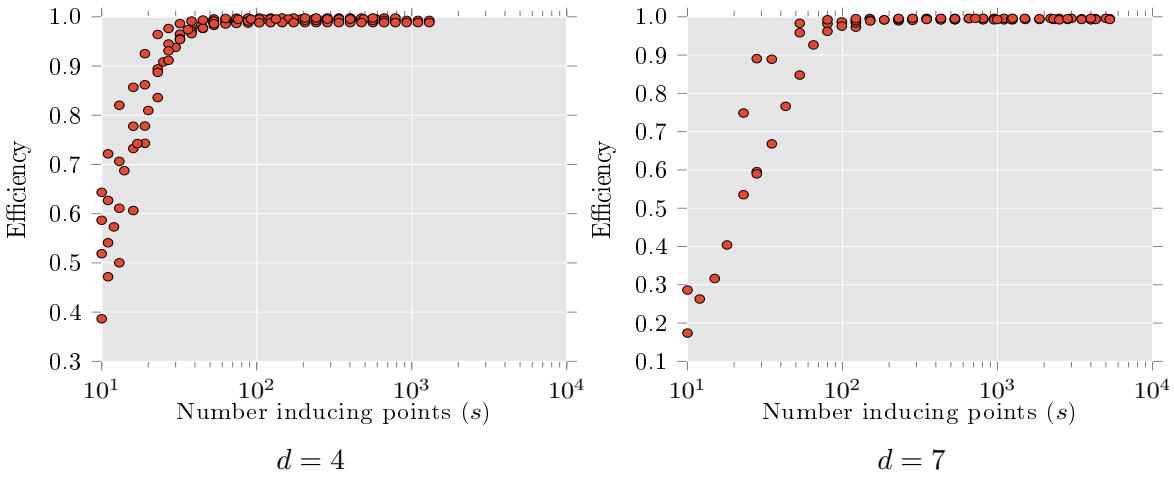

Figure 2: Efficiency of IMSE-optimal design, for an approximation of the kernel based on a SOR kernel with s inducing points.

and even of x350 for $s = 100$ and $d = 7$. For the latter instances, the computations took more than 36 hours on a Linux PC with 8 cores at 3.60 GHz, while with the SOR kernel a solution was found within 6 minutes (with a tolerance parameter ε set to 10^{-4} in the stopping criterion (11)).

Acknowledgments. The authors want to thank the two anonymous reviewers for their valuable comments and suggestions that significantly improved the presentation.

References

[BGL+12] 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.

[CBFH15] Y. Cao, M.A. Brubaker, D.J. Fleet, and A. Hertzmann. Efficient optimization for sparse gaussian process regression. *IEEE transations on Pattern Analysis and Machine Intelligence*, 37(12):2415–2427, 2015.

[DW12] P. Deuflhard and M. Weiser. *Adaptive numerical solution of PDEs*. Walter de Gruyter, 2012.

[Fed96] V. Fedorov. Design of spatial experiments: Model fitting and prediction. *Handbook of Statistics*, 13:515–553, 1996.

[FF97] V.V. Fedorov and D. Flanagan. Optimal monitoring network design based on Mercer's expansion of covariance kernel. *Journal of Combinatorics, Information and System Sciences*, 23:237–250, 1997.

[GM92] N. Gaffke and R. Mathar. On a class of algorithms from experimental design theory. *Optimization*, 24(1-2):91–126, 1992.

[GP14] B. Gauthier and L. Pronzato. Spectral approximation of the IMSE criterion for optimal designs in kernel-based interpolation models. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):805–825, 2014.

[GP16] B. Gauthier and L. Pronzato. Optimal design for prediction in random field models via covariance kernel expansions. In *Proceedings of the 11th Workshop on Model-Oriented Data Analysis and Optimum Designs (mODa'11)*, 2016.

[JSW98] D.R. Jones, M. Schonlau, and W.J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[KW60] J. Kiefer and J. Wolfowitz. The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12:363–366, 1960.

[MS10] W.G. Müller and M. Stehlík. Compound optimal spatial designs. *Environmetrics*, 21(3-4):354–364, 2010.

[Páz86] A. Pázman. *Foundations of optimum experimental design*. D. Reidel Dordrecht, 1986.

[Pil91] J. Pilz. *Bayesian estimation and experimental design in linear regression models*, volume 212. John Wiley & Sons Inc, 1991.

[PM12] L. Pronzato and W.G. Müller. Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22(3):681–701, 2012.

[Pro13] L. Pronzato. A delimitation of the support of optimal designs for Kiefer's ϕ_p-class of criteria. *Statistics & Probability Letters*, 83(12):2721–2728, 2013.

[Puk93] F. Pukelsheim. *Optimal Design of Experiments*. Wiley, 1993.

[QCR05] J. Quiñonero-Candela and C.E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

[Sär13] S. Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.

[SP10] G. Spöck and J. Pilz. Spatial sampling design and covariance-robust minimax prediction based on convex design ideas. *Stochastic Environmental Research and Risk Assessment*, 24(3):463–482, 2010.

[SP15] G. Spöck and J. Pilz. Incorporating covariance estimation uncertainty in spatial sampling design for prediction with trans-gaussian random fields. *Frontiers in Environmental Science*, 3(39), 2015.

[STT78] S.D. Silvey, D.M. Titterington, and B. Torsney. An algorithm for optimal designs on a finite design space. *Communications in Statistics - Theory and Methods*, 7(14):1379–1389, 1978.

[Tit09] M.K. Titsias. Variational learning of inducing variables in sparse gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 567–574, 2009.

[WN15] A.G. Wilson and H. Nickisch. Kernel interpolation for scalable structured gaussian processes (KISS-GP). In *Proceedings of The 32nd International Conference on Machine Learning*, volume 37 of *JMLR: W&CP*, pages 1775–1784, Lille, France, 2015.

[WS01] C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Proceedings of the 14th Annual Conference on Neural Information Processing Systems*, number EPFL-CONF-161322, pages 682–688, 2001.

[Yu10] Y. Yu. Monotonic convergence of a general algorithm for computing optimal designs. *The Annals of Statistics*, 38(3):1593–1606, 2010.