Replication or exploration? Sequential design for stochastic simulation experiments

Mickaël Binois* Jiangeng Huang† Robert B. Gramacy†
Mike Ludkovski‡

January 28, 2019

Abstract

We investigate the merits of replication, and provide methods for optimal design (including replicates), with the goal of obtaining globally accurate emulation of *noisy* computer simulation experiments. We first show that replication can be beneficial from both design and computational perspectives, in the context of Gaussian process surrogate modeling. We then develop a lookahead based sequential design scheme that can determine if a new run should be at an existing input location (i.e., replicate) or at a new one (explore). When paired with a newly developed heteroskedastic Gaussian process model, our dynamic design scheme facilitates learning of signal and noise relationships which can vary throughout the input space. We show that it does so efficiently, on both computational and statistical grounds. In addition to illustrative synthetic examples, we demonstrate performance on two challenging real-data simulation experiments, from inventory management and epidemiology.

Keywords: computer experiment, Gaussian process, surrogate model, input-dependent noise, replicated observations, lookahead

1 Introduction

Historically, design and analysis of computer experiments focused on deterministic solvers from the physical sciences via Gaussian process (GP) interpolation (Sacks et al., 1989). But nowadays computer modeling is common in the social (Cioffi-Revilla, 2014, Chapter 8), management (Law, 2015) and biological (Johnson, 2008) sciences, where stochastic simulations abound. Noisier simulations demand bigger experiments to isolate signal from noise, and more sophisticated GP models—not just adding nuggets to smooth the noise, but variance

*Corresponding author: The University of Chicago Booth School of Business, Chicago IL, 60637; mbinois@mcs.anl.gov
†Department of Statistics, Virginia Tech
‡Department of Statistics and Applied Probability, University of California Santa Barbara

processes to track changes in noise throughout the input space in the face of heteroskedasticity (Binois et al., 2016). In this context there are not many simple tools: most add to, rather than reduce, modeling and computational complexity.

Replication in the experiment design is an important exception, offering a pure look at noise, not obfuscated by signal. Since usually the signal is of primary interest, a natural question is: How much replication should be performed in a simulation experiment? The answer to that question depends on a number of factors. In this paper the focus is on global surrogate model prediction accuracy and computational efficiency, and we show that replication can be a great benefit to both, especially for heteroskedastic systems.

There is evidence to support this in the literature. Ankenman et al. (2010) demonstrated how replicates could facilitate signal isolation, via stochastic kriging (SK), and that accuracy could be improved without much extra computation by augmenting existing degrees of replication in stages (also see Liu and Staum, 2010; Quan et al., 2013; Mehdad and Kleijnen, 2018). Wang and Haaland (2017) showed that replicates have an important role in characterizing sources of inaccuracy in SK. Boukouvalas et al. (2014) demonstrated the value of replication in (Fisher) information metrics, and Plumlee and Tuo (2014) provided asymptotic results favoring replication in quantile regression. Finally, replication has proved helpful in the surrogate-assisted (i.e., Bayesian) optimization of noisy blackbox functions (Horn et al., 2017; Jalali et al., 2017).

However, none of these studies address what we see as the main decision problem for design of GP surrogates in the face of noisy simulations. That is: how to allocate a set of unique locations, and the degree of replication thereon, to obtain the best overall fit to the data. That sentiment has been echoed independently in several recent publications (Kleijnen, 2015; Weaver et al., 2016; Jalali et al., 2017; Horn et al., 2017). The standard approach of allocating a uniform number of replicates leaves plenty of room for improvement. One exception is Chen and Zhou (2014, 2017) who proposed several criteria to explore the replication/exploration trade-off, but only for a finite set of candidate designs.

Here we tackle the issue sequentially, one new design element at a time. We study the conditions under which the new element should be a replicate, or rather explore a new location, under an integrated mean-square prediction error (IMSPE) criterion. We also highlight how replicates offer computational savings in surrogate model fitting and prediction with GPs, augmenting results of Binois et al. (2016) with fast updates as new data arrives. Inspired by those findings, we develop a new IMSPE-based criterion that offers lookahead over future replicates. This criterion is the first to acknowledge that exploring now offers a new site for replication later, and conversely that replicating first offers the potential to learn a little more (cheaply, in terms of surrogate modeling computation) before committing to a new design location. A key component in solving this sequential decision problem in an efficient manner is a closed form expression for IMSPE, and its derivatives, allowing for fast numerical optimization.

While our IMSPE criterion corrects for myopia in replication, it is important to note that it is not a full lookahead scheme. Rather, we illustrate that it is biased toward replication: longer lookahead horizons tend to tilt toward more replication in the design. In our experi-

ence, full lookahead, even when approximated, is impractical for all but the most expensive simulations. Even the cleverest dynamic programming-like schemes (e.g., Ginsbourger and Le Riche, 2010; Gonzalez et al., 2016; Lam et al., 2016; Huan and Marzouk, 2016) require approximation to remain tractable or otherwise only manage to glimpse a few steps into the future despite enormous computational cost. Our more thrifty scheme can search dozens of iterations ahead. That flexibility allows us to treat the horizon as a tuning parameter that can be adjusted, online, to meet design and/or surrogate modeling goals. When simulations are cheap and noisy, we provide an adaptive horizon scheme that favors replicates to keep surrogate modeling costs down; when surrogate modeling costs are less of a concern, we provide a scheme that optimizes out-of-sample RMSE, which might or might not favor longer horizons (i.e., higher replication).

The structure of the remainder of the paper is as follows. First we summarize relevant elements of GPs, sequential design and the computational savings enjoyed through replication in Section 2. Then in Section 3 we detail IMSPE, with emphasis on sequential applications and computational enhancements (e.g., fast GP updating) essential for the tractability of our framework. Section 3.3 discusses our lookahead scheme, while Section 4 provides practical elements for the implementation, including tuning the horizon of the lookahead scheme. Finally, in Section 5 results are presented from several simulation experiments, including illustrative test problems, and real simulations from epidemiology and inventory management, which benefit from disparate design strategies.

2 Background and proof of concept

Here we introduce relevant surrogate modeling and design elements while at the same time illustrating proof-of-concept for our main methodological contributions. Namely that replication can be valuable computationally, as well as for accuracy in surrogate modeling.

2.1 Gaussian process regression with replication

We consider Gaussian process (GP) surrogate models for an unknown function over a fixed domain $f : D \subset \mathbb{R}^d \to \mathbb{R}$ based on noisy observations $\mathbf{Y} = (y_1, \ldots, y_N)^\top$ at design locations $\mathbf{X} = (\mathbf{x}_1^\top, \ldots, \mathbf{x}_N^\top)$. For simplicity, we assume a zero-mean GP prior, completely specified by covariance kernel $k(\cdot, \cdot)$, a positive definite function. Many different choices of kernel are possible, while in the computer experiments literature the power exponential and Matérn families are the most common. Often the families are parametererized by unknown quantities such as lengthscales, scales, etc., which are inferred from data (see, e.g., Rasmussen and Williams, 2006; Santner et al., 2013). The noise is presumed to be zero-mean i.i.d. Gaussian, with variance $r(\mathbf{x}) = \mathbb{V}\mathrm{ar}[Y(\mathbf{x})|f(\mathbf{x})]$. While we discuss our preferred modeling and inference apparatus in Section 4.1, for now we make the (unrealistic) assumption that kernel hyperparameters, along with the potentially non-constant $r(\mathbf{x})$, are known. Altogether, the data-generating mechanism follows a multivariate normal distribution, $\mathbf{Y} \sim \mathcal{N}_N(0, \mathbf{K}_N)$, where $\mathbf{K}_N$ is an $N \times N$ matrix comprised of $k(\mathbf{x}_i, \mathbf{x}_j) + \delta_{ij} r(\mathbf{x}_i)$, for $1 \le i, j \le N$ and with

δ_{ij} being the Kronecker delta function.

Conditional properties of multivariate normal (MVN) distributions yield that the predictive distribution $Y(\mathbf{x})|\mathbf{Y}$ is Gaussian with

$$\begin{aligned}\mu_N(\mathbf{x}) &= \mathbb{E}(Y(\mathbf{x})|\mathbf{Y}) = \mathbf{k}_N(\mathbf{x})^\top \mathbf{K}_N^{-1}\mathbf{Y}, \quad \text{with } \mathbf{k}_N(\mathbf{x}) = (k(\mathbf{x},\mathbf{x}_1),\ldots,k(\mathbf{x},\mathbf{x}_N))^\top; \\ \sigma_N^2(\mathbf{x}) &= \mathbb{V}\text{ar}(Y(\mathbf{x})|\mathbf{Y}) = k(\mathbf{x},\mathbf{x}) + r(\mathbf{x}) - \mathbf{k}_N^\top(\mathbf{x})\mathbf{K}_N^{-1}\mathbf{k}_N(\mathbf{x}). \end{aligned} \tag{1}$$

It can be shown that $\mu(\mathbf{x})$ is a best linear unbiased predictor (BLUP) for $Y(\mathbf{x})$ (and $f(\mathbf{x})$). Although testaments to the high accuracy and attractive uncertainty quantification features abound in the literature, one notable drawback is that when N is large the computational expense of $\mathcal{O}(N^3)$ due to decomposing $\mathbf{K}_N$ (e.g., to solve for $\mathbf{K}_N^{-1}$) can be prohibitive.

When the observations $y(\mathbf{x})$ are deterministic (i.e., $r(\mathbf{x}) = 0$), often N can be kept to a manageable size. When data are noisy, with potentially varying noise level, many samples may be needed to separate signal from noise. Indeed in our motivating applications, the signal-to-noise ratios can be very low, so even for a relatively small input space, thousands of training observations are necessary. In that context replication can offer significant computational gains. To illustrate, let $\bar{\mathbf{x}}_i$, $i = 1,\ldots,n$ denote the $n \le N$ unique input locations, and $y_i^{(j)}$ be the j^{th} out of $a_i \ge 1$ replicates observed at $\bar{\mathbf{x}}_i$, i.e., $j = 1,\ldots,a_i$, where $\sum_{i=1}^n a_i = N$. Also, let $\bar{\mathbf{Y}}_{(N,n)} = (\bar{y}_1,\ldots,\bar{y}_n)^\top$ store averages over replicates, $\bar{y}_i = \frac{1}{a_i}\sum_{j=1}^{a_i} y_i^{(j)}$. Then Binois et al. (2016) show that predictive equations based on this "unique-n" formulation, i.e., following Eq. (1) except with $\bar{\mathbf{Y}}_{(N,n)}$ and $\mathbf{K}_{(N,n)} = \left(k(\bar{\mathbf{x}}_i,\bar{\mathbf{x}}_j) + \delta_{ij}\frac{r(\bar{\mathbf{x}}_i)}{a_i}\right)_{1\le i,j\le n}$, are identical. Compared to the "full-N" formulation, the respective costs are reduced from $\mathcal{O}(N^3)$ to just $\mathcal{O}(n^3)$, without any approximations.

2.2 Sequential design for GPs

Although there are many criteria dedicated to design for GP regression (see, e.g., Pronzato and Müller, 2012), our focus here is on global predictive accuracy defined via integrated mean-squared prediction error (IMSPE). Fixing $\mathbf{X}$, the IMSPE integrates the "de-noised" posterior variance $\check{\sigma}_N^2(\mathbf{x}) = \sigma_N^2(\mathbf{x}) - r(\mathbf{x})$ over D,

$$\text{IMSPE}(\mathbf{x}_1,\ldots,\mathbf{x}_N) = \int_{\mathbf{x}\in D} \check{\sigma}_N^2(\mathbf{x})\, d\mathbf{x} =: I_N. \tag{2}$$

Note that although this definition removes $r(\mathbf{x})$, it is still present in $\mathbf{K}_N$ and therefore affects $\check{\sigma}_N^2(\mathbf{x})$. Removing $r(\mathbf{x})$ is not required, but since $\int r(\mathbf{x})\, d\mathbf{x}$ is constant over $\mathbf{x}_1,\ldots,\mathbf{x}_N$, it simplifies future expressions.

Even in the highly idealized case were all covariance $k(\cdot,\cdot)$ and noise $r(\cdot)$ relationships are presumed known, one-shot design—i.e., choosing all N locations $\mathbf{X}$ at once to minimize (2)—is an extraordinarily difficult task owing to the $(N \times d)$-dimensional search space. Only in very specific cases, such as $d = 1$ and a exponential kernel (Antognini and Zagoraiou, 2010), or with the simpler task of allocating N replicates to a fixed set of n unique sites $\bar{\mathbf{x}}_1,\ldots,\bar{\mathbf{x}}_n$ (Ankenman et al., 2010), is a computationally tractable solution known.

Therefore, we consider here the simpler case of a purely sequential design, building up a big design greedily, one simulation at a time. Note that this means that N grows by 1 after each iteration. While n is also evolving, the precise change is dependent on whether a replicate or a new location is selected. In the generic step, we condition on existing $\mathbf{x}_1, \ldots, \mathbf{x}_N$ locations and optimize IMSPE$(\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{x}_{N+1})$ over $\mathbf{x}_{N+1}$. Recall that the posterior variance $\check{\sigma}^2_N$ only depends on the geometry of $\mathbf{X}$, i.e., it is independent of the outputs $\mathbf{Y}$ and hence we can view the above as minimizing $I_{N+1}(\mathbf{x}_{N+1}) := \text{IMSPE}(\mathbf{x}_{N+1}|\mathbf{x}_1, \ldots, \mathbf{x}_N)$. Later we establish specific closed-form expressions both for I_{N+1} and its gradient which enable fast optimization via library-based numerical schemes. Foreshadowing these developments, and utilizing the calculations detailed therein, we illustrate here the possibility that $\mathbf{x}_{N+1} = \text{argmin}_{\mathbf{x}} I_{N+1}(\mathbf{x})$ is a replicate. The conditions under which replication is advantageous, which we describe shortly in Section 3.1, have to our knowledge only been illustrated empirically (Boukouvalas, 2010), or conceptually (e.g., Wang and Haaland (2017) highlight that replication is more beneficial as the signal-to-noise ratio decreases, via upper bounds on the MSPE), or to bolster technical results (e.g., Plumlee and Tuo (2014) demand a sufficient degree of replication to ensure asymptotic efficiency).

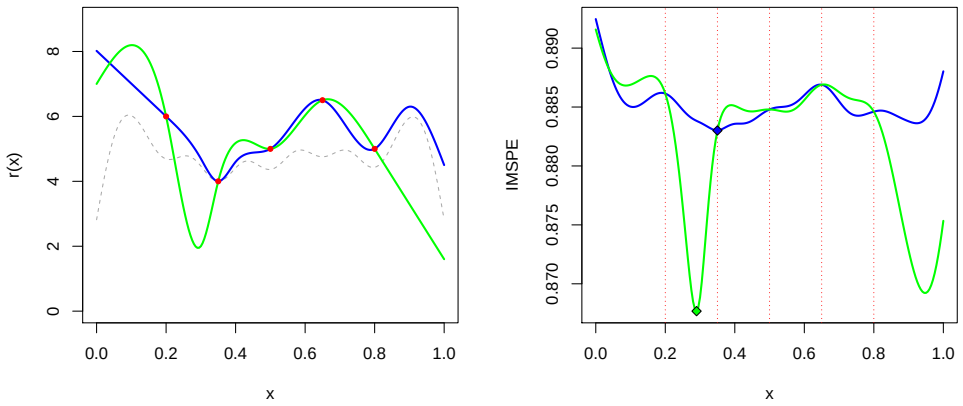


Figure 1: Illustration of the effect of noise variance on IMSPE optimization. Left: two examples of noise variance functions $r(\cdot)$ (blue solid and green dashed lines), with observations at $\mathbf{X}$ (five red points). The grey dotted line represents the minimum $r(\mathbf{x})$ that guarantees that replicating is optimal. Right: $I_{N+1}(\mathbf{x})$ for the two respective $r(\cdot)$. Diamonds highlight minimum values, and red dotted lines the existing designs $\mathbf{x}_1, \ldots, \mathbf{x}_5$.

The *left* panel of Figure 1 shows two different noise levels, $r(\mathbf{x})$, for a stylized heteroskedastic GP predictor trained at $N = 5$ locations whose $\mathbf{x}_1, \ldots, \mathbf{x}_5$ values are shown as red dots. The fact that the two $r(\mathbf{x})$ curves coincide at these locations is not material to this discussion. Later in Section 3.1 this feature and a description of the gray-dotted curve will be provided. The right panel in the figure shows the predicted IMSPE, $I_{N+1}(\mathbf{x}) = \text{IMSPE}(\mathbf{x}_1, \ldots, \mathbf{x}_5, \mathbf{x})$ derived from $\check{\sigma}^2_N$ calculations using those $\mathbf{x}_1, \ldots, \mathbf{x}_5$ values combined with the two $r(\mathbf{x})$ settings. With smaller IMSPE being better, we see that the solid blue regime calls for $\mathbf{x}_6$ being a replicate ($\text{argmin}\, I_{N+1}(\mathbf{x}) = \mathbf{x}_2$), whereas the dashed green regime wants to explore at a new unique location ($\text{argmin}\, I_{N+1}(\mathbf{x}) \simeq 0.32$). Also note that the IMSPE surfaces are multi-

modal, which may pose a challenge to numerical optimizers, and that even for the dashed green curve there are replicates (e.g., at $\mathbf{x}_2$) with lower IMSPE than some local minima, meaning that augmenting with a cheap discrete search over replicates may be more effective than deploying a multi-start optimization scheme.

Ultimately, we entertain the far more realistic setup of unknown kernel hyperparameters and noise processes. In this context, sequential design to "learn-as-you-go" is essential. We take this approach not simply to avoid pathologies in hyperparameter mis-specification, as discussed in homoskedastic setups (e.g., Seo et al., 2000; Krause and Guestrin, 2007), but explicitly to gain the flexibility to sample non-uniformly in a manner that can only be adapted after a degree of initial sampling allows a fit of the noise process $\hat{r}(\mathbf{x})$ to be obtained, and further refined. Our empirical results illustrate that reasonable, yet inaccurate, *a priori* simplifications such as constant $r(\mathbf{x})$ may—even if just for the purposes of design, not subsequent fitting—lead to inferior prediction. Previously such adaptive behavior and non-uniform sampling was only available via more cumbersome fully nonstationary methods, say involving treed partitioning (Gramacy and Lee, 2009).

3 IMSPE through the lens of replication

Over the years several authors (e.g., Ankenman et al., 2010; Anagnostopoulos and Gramacy, 2013; Burnaev and Panov, 2015; Leatherman et al., 2017) have provided closed form expressions for IMSPE (i.e., for the integral in (2)) via variations on the criterion's definition (i.e., versions somewhat different than our preferred version in (2)), or via simplifications to the GP specification or to the argument $\mathbf{x}_1, \ldots, \mathbf{x}_N$, obtained by constraining the search set. Others have argued in more general contexts that d-dimensional numerical integration, usually via sums over a (potentially random) reference set, is the only viable option in their setting (Seo et al., 2000; Gramacy and Lee, 2009; Gauthier and Pronzato, 2014; Gorodetsky and Marzouk, 2016; Pratola et al., 2017).

Here we provide a new closed-form expression for the IMSPE which, despite being intimately connected to earlier versions, is quite general and, we think, could replace many of the prevailing numerical schemes. This development uses the "unique-n" representation for efficient calculation under replication, however the analogue "full-N" version is immediate. We then consider an "add one" variation, $I_{N+1}(\tilde{\mathbf{x}}) = \text{IMSPE}(\tilde{\mathbf{x}}|\mathbf{x}_1, \ldots, \mathbf{x}_N)$, for efficient calculation in the sequential design setting and derive a condition under which replication is preferred for the next sample. Here we use $\tilde{\mathbf{x}}$ for a potential new location, while $\mathbf{x}_{N+1}$ will ultimately be chosen as the best candidate (i.e., minimizing IMSPE over $\tilde{\mathbf{x}}$). Note that if $\mathbf{x}_{N+1}$ turns out to be a replicate, n would not increase.

One important reason to have a closed-form IMSPE is the calculation of gradients, also in closed form, to aid in optimization. We provide the first such derivative expressions of which we are aware. Finally, acknowledging the dual role of replication (to speed calculations and separate signal from noise) we describe two new lookahead IMSPE heuristics for tuning the lookahead horizon in an online fashion, depending on whether speed or accuracy is more important.

3.1 IMSPE closed-formed expressions

We start by writing the IMSPE, shorthanded as I_N in Eq. (2), as an expectation:

$$I_N = \int_{\mathbf{x}\in D} \check{\sigma}_n^2(\mathbf{x})\, d\mathbf{x} = \mathbb{E}[\check{\sigma}_n^2(X)] = \mathbb{E}[k(X,X)] - \mathbb{E}[\mathbf{k}_n(X)^\top \mathbf{K}_n^{-1}\mathbf{k}_n(X)]$$

with X uniformly sampled in D, and using the linearity of the expectation. Notice that $\mathbf{K}_n$ depends on the number of replicates per unique design, so this representation includes a tacit dependence on the noise and replication counts $a_1, \ldots, a_n$. Then, as shown in Lemma 3.1, the integration of $\check{\sigma}_n^2$ over D may be reduced to integrations of the covariance function.

Lemma 3.1. *Let $\mathbf{W}_n$ be an $n \times n$ matrix with entries comprising integrals of kernel products $w(\mathbf{x}_i, \mathbf{x}_j) = \int_{\mathbf{x}\in D} k(\mathbf{x}_i, \mathbf{x}) k(\mathbf{x}_j, \mathbf{x})\, d\mathbf{x}$ for $1 \le i, j \le n$, and let $E = \int_{\mathbf{x}\in D} k(\mathbf{x}, \mathbf{x})\, d\mathbf{x}$. Then*

$$I_N = E - \mathrm{tr}(\mathbf{K}_n^{-1}\mathbf{W}_n). \tag{3}$$

Proof. The first part, involving E, follows simply by definition. For the second part, let $\mathbf{z}$ be a random vector of size n with mean $\mathbf{m}$ and covariance $\mathbf{M}$. Petersen et al. (2008) provides that $\mathbb{E}\left[\mathbf{z}^\top \mathbf{K}_n^{-1}\mathbf{z}\right] = \mathrm{tr}\left(\mathbf{K}_n^{-1}\mathbf{M}\right) + \mathbf{m}^\top \mathbf{K}_n^{-1}\mathbf{m}$. Therefore using $\mathbf{m}^\top \mathbf{K}_n^{-1}\mathbf{m} = \mathrm{tr}(\mathbf{K}_n^{-1}\mathbf{m}\mathbf{m}^\top)$, we have $\mathbb{E}[\mathbf{k}_n(X)^\top \mathbf{K}_n^{-1}\mathbf{k}_n(X)] = \mathrm{tr}(\mathbf{K}_n^{-1}(\mathbf{M} + \mathbf{m}\mathbf{m}^\top))$ where $\mathbf{m} = \mathbb{E}[\mathbf{k}_n(X)]$ and $\mathbf{M} = \mathbb{C}ov(\mathbf{k}_n(X)^\top, \mathbf{k}_n(X))$. Observing that $\mathbf{W}_n = \mathbf{M} + \mathbf{m}\mathbf{m}^\top$ gives the desired result. □

Our interest in the re-characterization in (3) is three-fold. First and foremost, some of the most commonly used kernels enjoy closed form expressions of E and $w(\cdot,\cdot)$. In Appendix B we provide $w(\cdot,\cdot)$ for (i) Gaussian, (ii) Matérn-5/2, (iii) Matérn-3/2, and (iv) Matérn-1/2 families. For those families, E further reduces to their scale hyperparameter. Section 4.1 offers specific forms for the generic expression (3) under our `hetGP` model. Second, note that even when closed forms are not available, as may arise when the kernel $k(X,X)$ cannot be analytically integrated over D, this formulation may still be advantageous. Numerically integrating $k(\mathbf{x}, \cdot)$ inside $\mathbf{W}_n$ will likely be far easier than the alternative of integrating $\check{\sigma}_n^2$, which can be highly multi-modal. Third, we remark that $\mathrm{tr}(\mathbf{K}_n^{-1}\mathbf{W}_n) = \mathbf{1}^\top(\mathbf{K}_n^{-1} \circ \mathbf{W}_n)\mathbf{1}$ where $\circ$ stands for the Hadamard (i.e., element-wise) product. Once $\mathbf{K}_n^{-1}$ and $\mathbf{W}_n$ are computed, the cost is in $\mathcal{O}(n^2)$, whereas the naïve alternative is $\mathcal{O}(n^3)$.

Now, in sequential application the goal is to choose a new $\mathbf{x}_{N+1}$ by optimizing $I_{N+1}(\tilde{\mathbf{x}})$ over candidates $\tilde{\mathbf{x}}$. Fixing the first n unique design elements simplifies calculations substantially if we assume that $\mathbf{K}_n^{-1}$ and $\mathbf{W}_n$ are previously available. In that case, write

$$\mathbf{K}_{n+1} = \begin{bmatrix} \mathbf{K}_n & \mathbf{k}_n(\tilde{\mathbf{x}}) \\ \mathbf{k}_n(\tilde{\mathbf{x}})^\top & k(\tilde{\mathbf{x}},\tilde{\mathbf{x}}) + r(\tilde{\mathbf{x}}) \end{bmatrix}, \quad \mathbf{W}_{n+1} = \begin{bmatrix} \mathbf{W}_n & \mathbf{w}(\tilde{\mathbf{x}}) \\ \mathbf{w}(\tilde{\mathbf{x}})^\top & w(\tilde{\mathbf{x}},\tilde{\mathbf{x}}) \end{bmatrix}$$

with $\mathbf{w}(\tilde{\mathbf{x}}) = (w(\tilde{\mathbf{x}}, \bar{\mathbf{x}}_i))_{1\le i\le n}$. The partition inverse equations (Barnett, 1979) give

$$\mathbf{K}_{n+1}^{-1} = \begin{bmatrix} \mathbf{K}_n^{-1} + \mathbf{g}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top \sigma_n^2(\tilde{\mathbf{x}}) & \mathbf{g}(\tilde{\mathbf{x}}) \\ \mathbf{g}(\tilde{\mathbf{x}})^\top & \sigma_n^2(\tilde{\mathbf{x}})^{-1} \end{bmatrix}, \tag{4}$$

where $\mathbf{g}(\tilde{\mathbf{x}}) = -\sigma_n^2(\tilde{\mathbf{x}})^{-1}\mathbf{K}_n^{-1}\mathbf{k}_n(\tilde{\mathbf{x}})$ and $\sigma_n^2(\tilde{\mathbf{x}}) = \check{\sigma}_n^2(\tilde{\mathbf{x}}) + r(\tilde{\mathbf{x}})$ as in (1). Combining those two results together leads to

$$\begin{aligned} I_{N+1}(\tilde{\mathbf{x}}) &= E - \mathbf{1}^\top[\mathbf{K}_{n+1}^{-1} \circ \mathbf{W}_{n+1}]\mathbf{1} \\ &= E - \left(\mathbf{1}^\top[\mathbf{K}_n^{-1} \circ \mathbf{W}_n]\mathbf{1} + \sigma_n^2(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top\mathbf{W}_n\mathbf{g}(\tilde{\mathbf{x}}) + 2\mathbf{w}(\tilde{\mathbf{x}})^\top\mathbf{g}(\tilde{\mathbf{x}}) + \sigma_n^2(\tilde{\mathbf{x}})^{-1}w(\tilde{\mathbf{x}},\tilde{\mathbf{x}})\right) \\ &= I_N - \left(\sigma_n^2(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top\mathbf{W}_n\mathbf{g}(\tilde{\mathbf{x}}) + 2\mathbf{w}(\tilde{\mathbf{x}})^\top\mathbf{g}(\tilde{\mathbf{x}}) + \sigma_n^2(\tilde{\mathbf{x}})^{-1}w(\tilde{\mathbf{x}},\tilde{\mathbf{x}})\right). \end{aligned} \tag{5}$$

Both (4–5) only require $\mathcal{O}(n^2)$ computation.

After optimizing the latter part of (5) over $\tilde{\mathbf{x}}$ and choosing the best new design $\mathbf{x}_{N+1}$, one may utilize those inverse equations again to update the GP fit. Although similar identities have been provided in the literature (e.g., Gramacy and Polson, 2011; Chevalier et al., 2014), the ones we provide here are the first to exploit the thrifty "unique-n" representation, and to tailor to the setting where $\mathbf{x}_{N+1}$ is a replicate, i.e., an $\bar{\mathbf{x}}_k$, for $k \in \{1, \ldots, n\}$, versus a new distinct $\bar{\mathbf{x}}_{n+1}$ location.

Lemma 3.2. *Suppose $\mathbf{x}_{N+1} = \bar{\mathbf{x}}_k$. Then the updated predictive mean and variance (increasing N but not n) are given by*

$$\begin{aligned} \mu_{(N+1,n)}(\mathbf{x}) &:= \mu_{(N,n)}(\mathbf{x}) + \mathbf{k}_n(\mathbf{x})^\top(\mathbf{K}_{(N,n)}^{-1}(\bar{\mathbf{Y}}_{(N+1,n)} - \bar{\mathbf{Y}}_{(N,n)}) - \mathbf{B}_k\bar{\mathbf{Y}}_{(N+1,n)}), \\ \sigma^2_{(N+1,n)}(\mathbf{x}) &= \sigma^2_{(N,n)}(\mathbf{x}) - \mathbf{k}_n(\mathbf{x})^\top\mathbf{B}_k\mathbf{k}_n(\mathbf{x}), \end{aligned}$$

with $\mathbf{B}_k = \frac{\left(\mathbf{K}_{(N,n)}^{-1}\right)_{\cdot,k}\left(\mathbf{K}_{(N,n)}^{-1}\right)_{k,\cdot}}{a_k(a_k+1)/r(\bar{\mathbf{x}}_k) - \left(\mathbf{K}_{(N,n)}\right)^{-1}_{k,k}}$, *a rank-one matrix.*

Proof. By adding a replicate at $\bar{\mathbf{x}}_k$, the only change is to augment a_k by one in $\mathbf{K}_{(N+1,n)}$, namely $\mathbf{K}_{(N+1,n)} - \mathbf{K}_{(N,n)} = -\operatorname{Diag}\left(0, \ldots, 0, \frac{r(\bar{\mathbf{x}}_k)}{a_k(a_k+1)}, 0, \ldots, 0\right) =: -r(\bar{\mathbf{x}}_k)\mathbf{u}\mathbf{u}^\top = r(\bar{\mathbf{x}}_k)\mathbf{u}'\mathbf{u}^\top$ with $\mathbf{u}' = -\mathbf{u}$. Similarly, $\bar{\mathbf{Y}}_{(N+1,n)} - \bar{\mathbf{Y}}_{(N,n)} = \left(0, \ldots, 0, \frac{1}{a_k+1}(y_k^{(a_{k+1})} - \bar{y}_k^{(N)}), \ldots, 0\right)$ has only one non-zero element, residing in position k.

The Sherman-Morrison (i.e., rank-one Woodbury) formula gives

$$\mathbf{K}_{(N+1,n)}^{-1} = (\mathbf{K}_{(N,n)} + r(\bar{\mathbf{x}}_k)\mathbf{u}'\mathbf{u}'^\top)^{-1} = \mathbf{K}_{(N,n)}^{-1} + \frac{\left(\mathbf{K}_{(N,n)}^{-1}\right)_{\cdot,k}\left(\mathbf{K}_{(N,n)}^{-1}\right)_{k,\cdot}}{(r(\bar{\mathbf{x}}_k)u_k^2)^{-1} - \left(\mathbf{K}_{(N,n)}^{-1}\right)_{k,k}} = \mathbf{K}_{(N,n)}^{-1} + \mathbf{B}_k. \tag{6}$$

This enables us to write $\mu_{(N+1,n)}(\mathbf{x}) - \mu_{(N,n)}(\mathbf{x}) = \mathbf{k}_n(\mathbf{x})^\top\left(\mathbf{K}_{(N+1,n)}^{-1}\bar{\mathbf{Y}}_{n+1} - \mathbf{K}_{(N,n)}^{-1}\bar{\mathbf{Y}}_{(N,n)}\right)$ and $\sigma^2_{(N+1,n)}(\mathbf{x}) - \sigma^2_{(N,n)}(\mathbf{x}) = \mathbf{k}_n(\mathbf{x})^\top\left(\mathbf{K}_{(N+1,n)}^{-1} - \mathbf{K}_{(N,n)}^{-1}\right)\mathbf{k}_n(\mathbf{x})$ and substitute $\mathbf{K}_{(N+1,n)}^{-1} - \mathbf{K}_{(N,n)}^{-1} = \mathbf{B}_k$ from (6). From the proof we also see that adding a replicate $\mathbf{x}_{N+1}$ incurs $\mathcal{O}(n)$ rather than the usual $\mathcal{O}(n^2)$ cost. □

As a corollary we obtain the following formula for one-step-ahead IMSPE at existing designs $I_{N+1}(\bar{\mathbf{x}}_k)$ (relying on the fact that $\mathbf{W}_n$ is unchanged when replicating):

$$I_{N+1}(\bar{\mathbf{x}}_k) = E - \operatorname{tr}(\mathbf{K}_{(N+1,n)}^{-1}\mathbf{W}_n) = E - \operatorname{tr}((\mathbf{K}_{(N,n)}^{-1} + \mathbf{B}_k)\mathbf{W}_n) = I_N - \operatorname{tr}(\mathbf{B}_k\mathbf{W}_n). \tag{7}$$

Besides enabling a "quick check" (with cost $\mathcal{O}(n^2)$) for finding the best replicate, perhaps a more important application of this result is that (7) yields an explicit condition under which replication is optimal.

Proposition 3.1. *Given n unique design locations $\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_n$, replicating is optimal (with respect to I_{N+1}) if*

$$r(\tilde{\mathbf{x}}) \geq \frac{\mathbf{k}(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1} \mathbf{W}_n \mathbf{K}_n^{-1} \mathbf{k}(\tilde{\mathbf{x}}) - 2\mathbf{w}(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1} \mathbf{k}(\tilde{\mathbf{x}}) + w(\tilde{\mathbf{x}}, \tilde{\mathbf{x}})}{\operatorname{tr}(\mathbf{B}_{k^*} \mathbf{W}_n)} - \check{\sigma}_n^2(\tilde{\mathbf{x}}), \quad \forall \tilde{\mathbf{x}} \in D, \tag{8}$$

where $k^* \in \operatorname{argmin}_{1 \leq k \leq n} I_{N+1}(\bar{\mathbf{x}}_k)$.

Proof. We proceed by comparing $I_{N+1}(\tilde{\mathbf{x}})$ values when $\tilde{\mathbf{x}}$ is a replicate vis-à-vis a new design. Summarizing our results from above, we have $I_{N+1}(\bar{\mathbf{x}}_k^*) = I_N - \operatorname{tr}(\mathbf{B}_{k^*} \mathbf{W}_n)$ for the (best) replicate and $I_{N+1}(\tilde{\mathbf{x}}) = I_N - \left(\sigma_n^2(\tilde{\mathbf{x}}) \mathbf{g}(\tilde{\mathbf{x}})^\top \mathbf{W}_n \mathbf{g}(\tilde{\mathbf{x}}) + 2\mathbf{w}(\tilde{\mathbf{x}})^\top \mathbf{g}(\tilde{\mathbf{x}}) + \sigma_n^2(\tilde{\mathbf{x}})^{-1} w(\tilde{\mathbf{x}}, \tilde{\mathbf{x}})\right)$ for a new design. Replicating is better if $I_{N+1}(\bar{\mathbf{x}}_k^*) \leq I_{N+1}(\tilde{\mathbf{x}})$ for all $\tilde{\mathbf{x}}$, or when

$$\operatorname{tr}(\mathbf{B}_{k^*} \mathbf{W}_n) \geq \sigma_n^2(\tilde{\mathbf{x}})^{-1} \left(\mathbf{k}(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1} \mathbf{W}_n \mathbf{K}_n^{-1} \mathbf{k}(\tilde{\mathbf{x}}) - 2\mathbf{w}(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1} \mathbf{k}(\tilde{\mathbf{x}}) + w(\tilde{\mathbf{x}}, \tilde{\mathbf{x}})\right).$$

Using the fact that $\sigma_n^2(\tilde{\mathbf{x}}) = \check{\sigma}_n^2(\tilde{\mathbf{x}}) + r(\tilde{\mathbf{x}})$ establishes the desired result. □

Referring back to Figure 1, the gray-dotted line in the *left* panel represents the right hand side of Eq. (8). Thus, any noise surfaces with $r(\mathbf{x})$ above this line will lead to the I_{N+1} minimizer being a replicate, cf. the solid blue $r(\mathbf{x})$ case in the figure. Although this illustration involves a heteroskedastic example, the inequality in (8) can also hold in the homoskedastic case. In practice, replication in homoskedastic processes is most often at the edges of the input space, however particular behavior is highly sensitive to the settings of the n design locations, and their degrees of replication, a_i.

3.2 Gradient expressions

To facilitate the optimization of $I_{N+1}(\tilde{\mathbf{x}})$ with respect to $\tilde{\mathbf{x}}$, we provide closed-form expressions for its gradient, via partial derivatives. Below the subscript (p) denotes the p-th coordinate of the d-dimensional design $\tilde{\mathbf{x}} \in D$. As a starting point, the chain rule gives

$$\frac{\partial I_{N+1}(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}_{(p)}} = -\frac{\partial \operatorname{tr}(\mathbf{K}_{n+1}^{-1} \mathbf{W}_{n+1})}{\partial \tilde{\mathbf{x}}_{(p)}} = -\operatorname{tr}\left(\mathbf{K}_{n+1}^{-1} \frac{\partial \mathbf{W}_{n+1}}{\partial \tilde{\mathbf{x}}_{(p)}}\right) - \operatorname{tr}\left(\frac{\partial \mathbf{K}_{n+1}^{-1}}{\partial \tilde{\mathbf{x}}_{(p)}} \mathbf{W}_{n+1}\right). \tag{9}$$

To manage the computational costs, we notate below how the partial derivatives are distributed in another application of the partition inverse equations:

$$\frac{\partial \mathbf{K}_{n+1}^{-1}}{\partial \tilde{\mathbf{x}}} = \begin{bmatrix} \mathbf{H}(\tilde{\mathbf{x}}) & \mathbf{h}(\tilde{\mathbf{x}}) \\ \mathbf{h}(\tilde{\mathbf{x}})^\top & v_1(\tilde{\mathbf{x}}) \end{bmatrix} \tag{10}$$

$$\frac{\partial \mathbf{W}_{n+1}}{\partial \tilde{\mathbf{x}}_{(p)}} = \begin{bmatrix} \mathbf{0}_{n \times n} & \mathbf{c}_1(\tilde{\mathbf{x}}) \\ \mathbf{c}_1(\tilde{\mathbf{x}})^\top & c_2(\tilde{\mathbf{x}}) \end{bmatrix} \tag{11}$$

where the detailed expressions and derivations are given in Appendix A.

The expressions above are collected into the following lemma.

Lemma 3.3. *The p^{th} component of the gradient for sequential ISMPE is*

$$-\frac{\partial I_{N+1}}{\partial \tilde{\mathbf{x}}_{(p)}} = 2\mathbf{c}_1(\tilde{\mathbf{x}})^\top \mathbf{g}(\tilde{\mathbf{x}}) + \mathbf{c}_2 \sigma_n^2(\tilde{\mathbf{x}})^{-1} + \mathbf{1}_n^\top [\mathbf{H}(\tilde{\mathbf{x}})\sigma_n^2(\tilde{\mathbf{x}}) \circ \mathbf{W}_n]\mathbf{1}_n \\ + 2\mathbf{w}(\tilde{\mathbf{x}})^\top \mathbf{h}(\tilde{\mathbf{x}}) + v_1(\tilde{\mathbf{x}}) w(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}). \tag{12}$$

Proof. Beginning with Eq. (9), substitute (10) for the partial derivative of $\mathbf{K}_{n+1}^{-1}$, and (11) for that of $\mathbf{W}_{n+1}$. Then, note that $\mathbf{1}_n^\top [\mathbf{H}(\tilde{\mathbf{x}})\sigma_n^2(\tilde{\mathbf{x}}) \circ \mathbf{W}_n]\mathbf{1}_n$ can be rewritten as $v_2(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top \mathbf{W}_n \mathbf{g}(\tilde{\mathbf{x}})$ $+2\sigma_n^2(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top \mathbf{W}_n \mathbf{h}(\tilde{\mathbf{x}})$. □

Since no further matrix decompositions are required, note that calculating the gradient of $I_{N+1}(\tilde{\mathbf{x}})$ in this way incurs computational costs in $\mathcal{O}(n^2)$.

3.3 Looking ahead over replication

Under certain conditions, sequential design via IMSPE, i.e., greedily minimizing I_{N+1} to choose $\mathbf{x}_{N+1}$, can well-approximate a one-shot batch design of size $N_{\max}$ because the criterion is monotone supermodular (Das and Kempe, 2008; Krause et al., 2008). However, these results assume a known kernel hyperparameterization $k(\cdot, \cdot)$ and constant noise level $r(\cdot)$. In the more realistic case where those quantities must be estimated from data, and potentially with non-constant variance, there is ample evidence in the literature suggesting that sequential design can be *much better* than a batch design, e.g., based on a poorly-chosen parameterization, and no worse than an idealistic one (Seo et al., 2000; Gramacy and Lee, 2009). However, that does not mean that greedy, myopic, selection is optimal. By accounting for potential future selections in choosing the very next one, it is possible to obtain substantially improved final designs. However, the calculations involved, especially to "look ahead" from early sequential decisions to a far-away horizon $N_{\max}$, require expensive dynamic programming techniques to search an enormous decision space.

Approximating that full search, by limiting the lookahead horizon or otherwise reducing the scope of the decision space, has become an active area in Bayesian optimization via expected improvement (Ginsbourger and Le Riche, 2010; Gonzalez et al., 2016; Lam et al., 2016). Targeting overall accuracy has seen rather less development, the work by Huan and Marzouk (2016) being an important exception. Here we aim to port many of these ideas to our setting of IMSPE optimization, where the nature of our approximation involves a weak bias towards replication which we have shown can be doubly beneficial in design.

The essential decision boils down to either choosing an $\mathbf{x}_{N+1}$ to explore, i.e., a new design element $\bar{\mathbf{x}}_{n+1}$, or choosing to replicate with $\mathbf{x}_{N+1}$ taken to be some $\bar{\mathbf{x}}_k$, for $k \in \{1, \ldots, n\}$. However, rather than directly minimizing (5) or (7), respectively, we perform a "rollout" lookahead procedure similar to Lam et al. (2016) in order to explore the impact of those choices on a limited space of future design decisions. The updating equations in the previous subsections make this tractable.

In particular we consider a horizon $h \in \{0, 1, 2, \ldots\}$ determining the number of design iterations to look ahead, with $h = 0$ representing ordinary (myopic) IMSPE search. Although larger values of h entertain future sequential design decisions, the goal (for any h) is

to determine what to do *now*. Toward that end, we evaluate $h+1$ "decision paths" spanning alternatives between exploring sooner and replicating later, or vice versa. During each iteration along a given path, either (5) or (7) (but not simultaneously) is taken up as the hypothetical action. On the first iteration, if a new $\bar{\mathbf{x}}_{n+1}$ is chosen by optimizing Eq. (5), that location (along with the existing $\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_n$) are considered as candidates for future replication over the remaining h lookahead iterations (when $h \geq 1$). If instead a replicate is chosen in the first iteration, the lookahead recursively searches over the choice of which of the remaining h iterations will pick a new $\bar{\mathbf{x}}_{n+1}$, with the others optimizing over replicates. This recursion is resolved by moving to the second iteration and again splitting the decision path into the choice between replicate-explore-replicate-... and replicate-replicate-..., etc. After recursively optimizing up to horizon h along the $h+1$ paths, the ultimate IMSPE for the respective hypothetical design with size $N+1+h$ is computed, and the decision path yielding the smallest IMSPE is noted. Finally, the next $\bar{\mathbf{x}}_{N+1}$ is a new location if the explore-first path was optimal, and is a replicate otherwise.

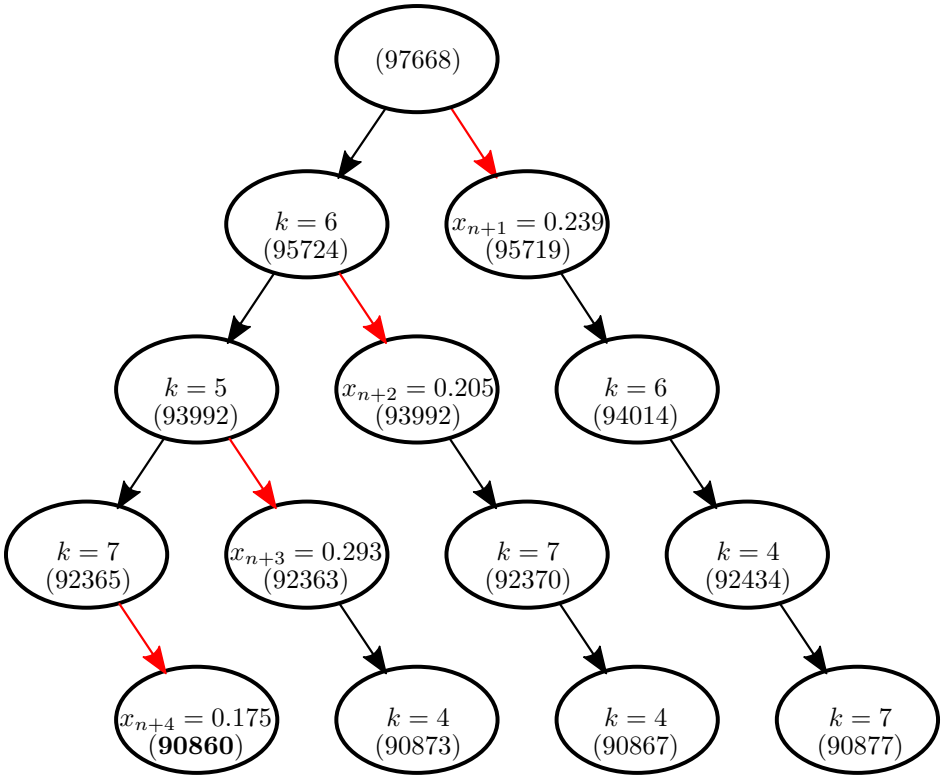


Figure 2: Lookahead strategy for $h = 3$, starting with n unique designs. Each ellipse is a state, with a specific training set. Black dashed arrows represent the action of adding the best replicate, red solid arrows represent adding the best new design. This example considers augmenting the design for the example shown in Figure 3. Numbers in parenthesis indicates the IMSPE at each stage (values have been multiplied by 10^8).

A diagram depicting the decision space is shown in Figure 2. In this example we attempt

to augment the design from Figure 3 using $h = 3$. The path yielding the lowest IMSPE at the horizon involves here replicating three times (adding copies of design elements 6, 5, and 7 respectively), with exploration at $x_{n+4} = 0.175$ in the final stage. Consequently, replication is preferred over exploration and the next design element will be a replicate, duplicating $\bar{\mathbf{x}}_6$. The figure also illustrates that the cost of searching for the best replicate over adding a new design element involves at most $h + 1$ global optimizations of Eq. (5), using the gradient. Although $(h+1)(h+2)/2 - 1$ discrete searches over (7) are required, the diagram indicates that $(h + 1)$ searches of mixed continuous and discrete type may be performed in parallel. In practice, global optimization with a budget of at least the same order as n is an order of magnitude more expensive than looking for the best replicate.

In this scheme the horizon, h, determines the extent to which replicates are entertained in the lookahead, and therefore larger h somewhat inflates the likelihood of replication. Indeed, as h grows, there are more and more decision paths that delay exploration to a later iteration; if any of them yield a smaller IMSPE than the explore-first path, the immediate action is to replicate. However, note that although larger h allows more replication before committing to a new, unique $\bar{\mathbf{x}}_{n+1}$, it also magnifies the value of an $\bar{\mathbf{x}}_{n+1}$ chosen in the first iteration, as it could potentially accrue its own replicates in subsequent rollout iterations. Therefore, although we do find in practice that larger h leads to more replication in the final design, this association is weak. Indeed, we frequently encounter situations where exploration is (temporarily) preferred for arbitrarily large horizons.

4 Modeling, inference and implementation

Here we consider inference and implementation details, in particular for learning the noise process $r(\cdot)$. Our presumption is that little is known about the noise, however it is worth noting that this assumption may not be well aligned to some data-generating mechanisms, e.g., as arising from Monte Carlo simulations with known convergence rates (Picheny and Ginsbourger, 2013). After reviewing a promising new framework called `hetGP`, for heteroskedastic GP surrogate modeling (Binois et al., 2016), we provide extensions facilitating fast sequential updating of that predictor along with its (hyper-) parameterization. We conclude with schemes for adjusting the lookahead horizon introduced in Section 3.3.

4.1 Heteroskedastic modeling

One way of dealing with heteroskedasticity in GP regression is to use empirical estimates of the variance as in SK (Ankenman et al., 2010), described briefly in Section 2. Although this has the downside of requiring a minimum amount of replication, the calculations are straightforward and computations are thrifty. However, sequential design requires predicting the variance at new locations, and to accommodate that within SK Ankenman et al., recommend fitting a second, independent, GP for $\hat{r}(\mathbf{x})$ to smooth the empirical variances.

An alternative is to model the (log) variance as a latent process, jointly with the original "mean process" (Goldberg et al., 1998; Kersting et al., 2007). However these methods can

be computationally cumbersome, and are not tailored to leverage the computational savings that come with replication. Here we rely on the hybrid approach detailed by Binois et al. (2016), leveraging replication and learning the latent log-variance GP based on a joint log-likelihood with the mean GP. We offer the following by way of a brief review.

For common choices of stationary kernel $k(\mathbf{x}, \mathbf{x}') = \nu c(\mathbf{x} - \mathbf{x}')$, the covariance matrix for the "mean GP" may be characterized as $\mathbf{K}_n = \nu(\mathbf{C}_n + \mathbf{\Lambda}_n)$ with $\mathbf{C}_n = (c(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j))_{1 \le i,j \le n}$; and for the "noise GP" we take the analog $\log \mathbf{\Lambda}_n = \mathbf{C}_{(g)}(\mathbf{C}_{(g)} + g\mathbf{A}_n^{-1})^{-1}\mathbf{\Delta}_n$ where $\mathbf{C}_{(g)}$ is the equivalent of $\mathbf{C}_n$ for the second GP with kernel $k_{(g)}$. That is, $\log \mathbf{\Lambda}_n$ is the prediction given by a GP based on latent variables $\mathbf{\Delta}_n = (\delta_1, \ldots, \delta_n)$ that can be learned as additional parameters, alongside hyperparameters of $k_{(g)}$ and nugget g.

Based on this representation, the MLE of ν is

$$\hat{\nu}_N := N^{-1}\left(N^{-1}\sum_{i=1}^{n}\frac{a_i}{\lambda_i}s_i^2 + \bar{\mathbf{Y}}^\top(\mathbf{C}_n + \mathbf{A}_n^{-1}\mathbf{\Lambda}_n)^{-1}\bar{\mathbf{Y}}\right)$$

with $s_i^2 = \frac{1}{a_i}\sum_{j=1}^{a_i}(y_i^{(j)} - \bar{y}_i)^2$ whereas the rest of the parameters and hyperparameters can be optimized based on the concentrated joint log-likelihood:

$$\begin{aligned}\log \tilde{L} = & -\frac{N}{2}\log\hat{\nu}_N - \frac{1}{2}\sum_{i=1}^{n}\left[(a_i - 1)\log\lambda_i + \log a_i\right] - \frac{1}{2}\log|\mathbf{C}_n + \mathbf{A}_n^{-1}\mathbf{\Lambda}_n| \\ & - \frac{n}{2}\log\hat{\nu}_{(g)} - \frac{1}{2}\log|\mathbf{C}_{(g)} + g\mathbf{A}_n^{-1}| + \text{Const},\end{aligned}$$

with $\hat{\nu}_{(g)} = n^{-1}\mathbf{\Delta}_n^\top(\mathbf{C}_n + g\mathbf{A}_n^{-1})^{-1}\mathbf{\Delta}_n$. Closed form derivatives are given in Binois et al. (2016), while an R (R Core Team, 2017) package with embedded C++ subroutines is available as `hetGP` on CRAN.

Notice that for stationary kernels, the Eq. (3) reduces to $\text{IMSPE}(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \nu(1 - \text{tr}(\mathbf{C}_n^{-1}\mathbf{W}_n))$. The look-ahead IMSPE over replicates (7) becomes $I_{N+1}(\bar{\mathbf{x}}_k) = \nu(1 - \text{tr}(\mathbf{B}_k'\mathbf{W}_n))$ with $\mathbf{B}_k' = \frac{\left((\mathbf{C}_n + \mathbf{A}_n^{-1}\mathbf{\Lambda}_n)^{-1}\right)_{\cdot,k}\left((\mathbf{C}_n + \mathbf{A}_n^{-1}\mathbf{\Lambda}_n)^{-1}\right)_{k,\cdot}}{a_k(a_k+1)/\lambda_k - \left(\mathbf{C}_n + \mathbf{A}_n^{-1}\mathbf{\Lambda}_n\right)^{-1}_{k,k}}$. Also, the gradient of $I_{N+1}(\tilde{\mathbf{x}})$ from (5) involves $\partial r(\tilde{\mathbf{x}})/\partial\tilde{\mathbf{x}}_{(p)}$, which for `hetGP` reduces to

$$\frac{\partial \mathbf{k}_{(g)}(\tilde{\mathbf{x}})(\mathbf{C}_{(g)} + g\mathbf{A}_n^{-1})^{-1}\mathbf{\Delta}_n}{\partial\tilde{\mathbf{x}}_{(p)}} = \frac{\partial \mathbf{k}_{(g)}(\tilde{\mathbf{x}})}{\partial\tilde{\mathbf{x}}_{(p)}}(\mathbf{C}_{(g)} + g\mathbf{A}_n^{-1})^{-1}\mathbf{\Delta}_n.$$

4.2 Sequential heteroskedastic modeling

Optimizing IMSPE with lookahead over replication [Section 3.3] is only practical if the `hetGP` model can be updated efficiently when new simulations are performed. Two different update schemes are necessary: one for potential new designs, considered during the process of evaluating alternatives under the criteria [Eqs. (5–7)]; and another for the actual update with new simulation $y(\mathbf{x}_{N+1})$.

When looking-ahead, no new y-value is entertained, so hyperparameters of both GPs stay fixed and only the latents may need to be augmented. Updating $\mathbf{K}_n$ follows (4) or (6), depending on whether the candidate $\tilde{\mathbf{x}}$ is new or a replicate. In the latter case, only $\mathbf{A}_n$ is updated for the "noise GP". Conversely, if a new location is added, an estimate of $r(\bar{\mathbf{x}}_{n+1})$ is required, which can come from the noise GP via exponentiating the usual GP predictive equations. That is, the new latent δ_{n+1} is taken as the predicted value by the noise GP.

The second update scheme—using the $y(\mathbf{x}_{N+1})$ observation—will require updating all the GPs' hyperparameters (including latents). Optimizing all hyperparameters of our heteroskedastic GP model is a potentially costly $\mathcal{O}(n^3)$ procedure. Instead of starting from scratch, a warm start of the MLE optimization is performed. Where they exist, previous values can be re-used as starting values, leaving only the latent $\tilde{\delta}$ at the newest design point, that is $\tilde{\delta} = \delta_{n+1}$ for a new location or $\tilde{\delta} = \delta_k$ for a replicate, requiring special attention.

As in the first case, $\tilde{\delta}$ may be initialized at its predicted value. But taking into account the new $y(\mathbf{x}_{N+1})$ makes it possible to combine information from the latent noise GP with results from empirical estimation of the log-variances. Kamiński (2015) explores this for updating SK models when new observations are added—a special case of the typical GP update formulas. The resulting combination of two predictions is via the geometric mean and can be summarized by the Gaussian $\mathcal{N}(\tilde{\delta}, V_{\tilde{\delta}})$ with

$$\tilde{\delta} = \left(\frac{\mu_{(g)}(\mathbf{x}_{N+1})}{\check{\sigma}^2_{(g)}(\mathbf{x}_{N+1})} + \frac{\hat{\delta}}{V_{\hat{\delta}}}\right)\left(\frac{1}{\check{\sigma}^2_{(g)}(\mathbf{x}_{N+1})} + \frac{1}{V_{\hat{\delta}}}\right)^{-1}, \quad V_{\tilde{\delta}} = \left(\frac{1}{\check{\sigma}^2_{(g)}(\mathbf{x}_{N+1})} + \frac{1}{V_{\hat{\delta}}}\right)^{-1},$$

where $\mu_{(g)}(\mathbf{x}_{N+1})$ and $\check{\sigma}^2_{(g)}(\mathbf{x}_{N+1})$ are the prediction from the noise GP[1] while $\hat{\delta}$ is the empirical estimate of the log variance at $\mathbf{x}_{N+1}$, itself with variance $V_{\hat{\delta}}$. We take $\hat{\sigma}^2 = \hat{\nu}_N^{-1}\frac{1}{\tilde{a}}\sum_{j=1}^{\tilde{a}}\left(y^{(j)}(\mathbf{x}_{N+1}) - \mu_n(\mathbf{x}_{N+1})\right)^2$, i.e., the uncorrected sample variance estimator that exists even for $\tilde{a} = 1$, i.e., the number of observations at $\mathbf{x}_{N+1}$. Supposing that the $y^{(j)}(\mathbf{x}_{N+1})$'s are i.i.d. Gaussian, we have $\tilde{a}\hat{\sigma}^2/\sigma^2 \sim \chi^2_{\tilde{a}}$. Accounting for the log-transformation, as in Boukouvalas (2010), we get $\hat{\delta} = \log(\hat{\sigma}^2) - \Psi((\tilde{a})/2) - \log(2) + \log(\tilde{a})$ and $V_{\hat{\delta}} = \Psi_2(\tilde{a}/2)$ with Ψ and Ψ_2 the digamma and trigamma functions.

Finally, the quick updates described above are predicated on improving local searches, and are thus not guaranteed to globally optimize the likelihood, which is always a challenge in MLE settings. The risk of becoming trapped in an inferior local mode is greater at earlier stages in the sequential design, i.e., when n is small. In practice, we find it beneficial to periodically restart the optimization with conservative (potentially random) initializations, which is cheap in that (small n) setting. As n increases, and the likelihood becomes more peaked, we find that costly restarts are of limited practical value. Local refinements, as described above, are fast and reliable.

[1]To avoid predictive variances close to zero for replicates, i.e., $\tilde{\delta} = \delta_k$, such that $\sigma^2_{(g)}(\mathbf{x}_{N+1}) \approx 0$ (g should be small), the variance is given by the "downdated" GP instead (i.e., the predicted variance if removing the replicated design), that are usually used for Leave-One-Out estimations and can be found, e.g., in Bachoc (2013), giving $\sigma^2_{(g)}(\mathbf{x}_{N+1}) = \left(\left(\mathbf{C}_{(g)} + g\mathbf{A}_n^{-1}\right)^{-1}_{k,k}\right)^{-1}$.

4.3 Defining the horizon

Although the horizon h in the lookahead criteria in Section 3.3 could be fixed arbitrarily, or chosen based on computational considerations (smaller being faster), here we propose two heuristics to set it adaptively based on design goals. The adaptiveness means that $h \equiv h_N$ is now indexed by the current design size.

The first heuristic involves managing surrogate modeling costs, targeting a fixed ratio $\rho = n/N$ of unique to full design size. The goal is to ensure that each new unique location is, "worth its weight" in replicates from a computational perspective. The choice of n/N is arbitrary —other targets will do—but we focus on this particular one because its magnitude is easy to intuit. The *Target* heuristic we use to "maintain ρ" as sequential design steps progress is as simple as it is effective:

$$h_{N+1} \leftarrow \begin{cases} h_N + 1 & \text{if } n/N > \rho \quad \text{and a new point } \bar{\mathbf{x}}_{n+1} \text{ is chosen;} \\ \max\{h_N - 1, -1\} & \text{if } n/N < \rho \quad \text{and a replicate is chosen;} \\ h_N & \text{otherwise.} \end{cases} \tag{13}$$

If the current ratio is too high and a new point $\bar{\mathbf{x}}_{n+1}$ was recently added, making the ratio even higher, the horizon is increased to encourage future replication. If rather a replicate has been added while the current ratio was too low, then the horizon is decreased, encouraging exploration. Otherwise the evolution is on the right trajectory and the horizon is unchanged. Observe that (13) allows a horizon of -1, which is described shortly.

To implement the continuous search (5), we deploy a limited multistart scheme over `optim` searches in R with `method="lbfgsb"` and closed form gradients (12). In parallel, a discrete search over $\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_n$ is carried out via (7). The two solutions thus obtained are compared against thresholds, and if the $\tilde{\mathbf{x}}$ found via continuous search (or its relative objective value) is within (say) $\varepsilon = 10^{-6}$ of that of the best $\bar{\mathbf{x}}_{k^*}$, the replicate is preferred on computational grounds. The horizon $h_N = -1$ is an exception, adding a new $\tilde{\mathbf{x}}$ no matter how close it is to the replicate candidate. Thus, $h \equiv -1$ can be roughly thought of as incrementing n by 1 along with N at each iteration; in practice it still occasionally generates replicates, primarily at the corners of the input space, if the corresponding multistart scheme determines that the I_{N+1}-minimizer lies at the boundary of D. On the other hand, $h \equiv 0$ obtains many replicates due to thresholding, which yields a "soft" clustering mechanism for $(\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_n)$. Indeed, every iteration where we have a situation resembling Figure 1, the $h = 0$ rule will select a replicate and not increment n. In contrast, $h = -1$ will only "stumble" into a replicate if the optimizer finds a global minimum at the edge of the domain. It does not explicitly entertain replicates via (7).

Our empirical work [Section 5] illustrates how horizon targeting effectively manages computational costs. Although at times the horizon h_N can reach quite high values (upwards of $h_N = 20$), the computational cost of search is negligible compared to updating the GP fits. Meanwhile high horizons represent a "light touch" preference for replication: they do not preclude exploration, rather they somewhat discourage it. Thus, while the ultimate number of unique locations n is dependent on the entire history of the simulations, and hence comes with a sampling distribution, the corresponding search heuristic is much simpler than one

that would impose a hard constraint on the final n.

When accuracy is the ultimate goal we prefer a different adaptation of h, making a more explicit link between ρ and the signal-to-noise ratio in the data. In *linear* regression contexts, one way to deal with heterogeneity is to allocate replications on unique designs such that the ratio of the empirical variance over number of replicates are close to each other, i.e., to enforce homogeneity of $\hat{\sigma}_i^2/a_i$ (Kleijnen, 2015). This approach captures the basic idea that more replicates are needed where $r(\mathbf{x})$ is high, but applicability to our setup is not direct because such a scheme does not factor in correlations estimated by GPs. Ankenman et al. (2010) address this within SK by considering the allocation of the remaining budget of evaluations over existing designs, i.e., to determine where to augment with additional replicates. In particular, they show that the optimal allocation of the N simulations across n unique designs is summarized by $\mathbf{A}_n^*$, a diagonal matrix with components

$$a_i^* \approx N \frac{\sqrt{r(\bar{\mathbf{x}}_i)K_i}}{\sum_{j=1}^{n} \sqrt{r(\bar{\mathbf{x}}_j)K_j}}, \quad \text{where} \quad K_i = (\mathbf{K}_n^{-1}\mathbf{W}_n\mathbf{K}_n^{-1})_{i,i}. \tag{14}$$

We emphasize that (14) only addresses the replication aspect—the designs $\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_n$ must be entered *a priori* by the user. Thus, this recipe is not directly implementable in a sequential design setting. One solution could be to generate (e.g., by space-filling) a candidate design of pre-determined size n and $\underline{\mathrm{r}}$ replicates per design and then, after learning $\mathbf{K}_n$ and $r(\bar{\mathbf{x}}_i)$'s, apply (14). However, in that case one may end up with $a_i^* < \underline{\mathrm{r}}$, as is the case in Figure 3. This illustrative 1d example highlights that in areas with low noise, a lower number of replicates would have been better, while in more noisy areas, more points are necessary. The right panel shows the a_i^* at this stage (referred to as *batch*) compared to the greedy sequential allocation of 105 replicates. The latter is more realistic because it acknowledges that design decisions cannot be undone[2].

Instead of such two-stage design, we utilize (14) in a sequential fashion, by making a comparison between the allocation a_i^* via (14) (employing the current estimates of the noise $r(\bar{\mathbf{x}})$ at that particular stage) and the actual a_i's collected so far from the sequential design. The existing number of replicates a_i is then either too high, in which case no more replicates should be added, or too low, and could benefit from more replication. We use this information in the *Adapt* scheme to adjust the horizon by sampling

$$h_{N+1} \sim \mathrm{Unif}\{a_1', \ldots, a_n'\} \quad \text{with} \quad a_i' := \max(0, a_i^* - a_i). \tag{15}$$

Hence, if there are locations that require many more replicates according to (14), h_{N+1} could be large to encourage replication.

5 Experiments

Here we illustrate our methods and simpler variants on a suite of examples spanning synthetic and real data from computer simulation experiments. Our main metric is out-of-sample

[2]The batch scheme recommends fewer than five replicates after five replicates where already used.

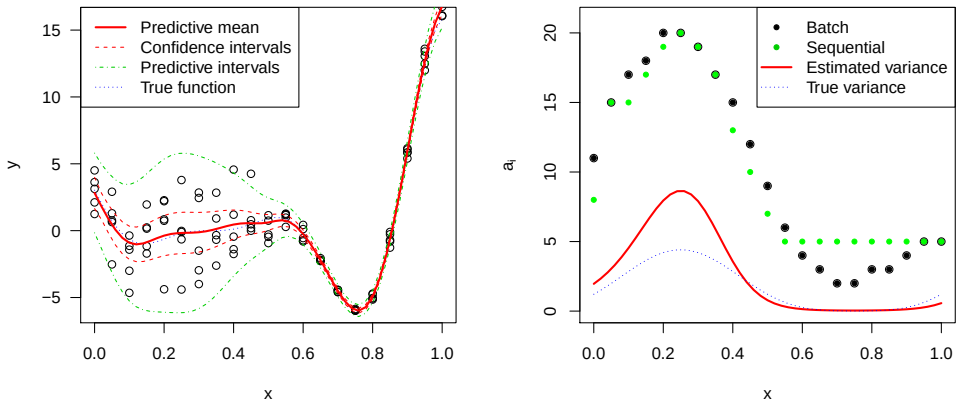


Figure 3: Left: toy example with 5 replicates at each of the 21 uniformly spaced unique design points. Right: proposed allocation of new 105 replicates (total 210 observations) based on (14) and a greedy sequential approach.

root mean-square (prediction) error (RMSE) over the sequential design iterations, and in particular after the final iteration. Since accurate estimation of variances over the input space is also an important consideration (especially in the heteroskedastic context)—even though our IMSPE criteria does not explicitly target learning variances—we consider RMSE to the true log variance, when it is known, and when it is not we use a proper scoring rule (Gneiting and Raftery, 2007, Eq. (27)) combining mean and variance forecasts out-of-sample. Our main comparators are non-sequential (space-filling) designs, homoskedastic GP predictors, and combinations thereof.

5.1 Illustrative one-dimensional example

We start by reusing the 1d toy example from above [surrounding Figure 2 and 3] to show qualitatively the effect of the horizon choice on the resulting designs. The underlying function is $f(x) = (6x-2)^2 \sin(12x-4)$, from Forrester et al. (2008), and the noise function is $r(x) = (1.1 + \sin(2\pi x))^2$. The experiment starts with an initial maximin LHS with 10 points, no replicates, and the GPs use a Gaussian kernel.

Results are presented in Figure 4 for a total budget of $N_{\max} = 500$. Each panel in the figure corresponds to a different look-ahead horizon h, with the final two involving Adaptive and Target schemes. There are several noteworthy observations. Notice that as the horizon is increased, more replicates are added. See $\rho = n/N$ reported in the main title of each panel. The design density is greatest in the high variance parts of the space, and that density is increasingly replaced by replication when the horizon is increased. The effect is most drastic from $h = -1$ to $h = 0$, with the ratio of unique designs over total designs dropping by more than half without impact on performance. Notice that replicates are added even with $h = -1$, at the extremities of the space. Results with high horizons and Adapt and Target schemes end up having both fewer unique designs and a higher accuracy. The very best RMSE results are provided by $h = 4$ and the Adapt (15) scheme. In the latter case just 60 unique locations are used ($\rho = 0.12$), with some design points replicated as many as thirty

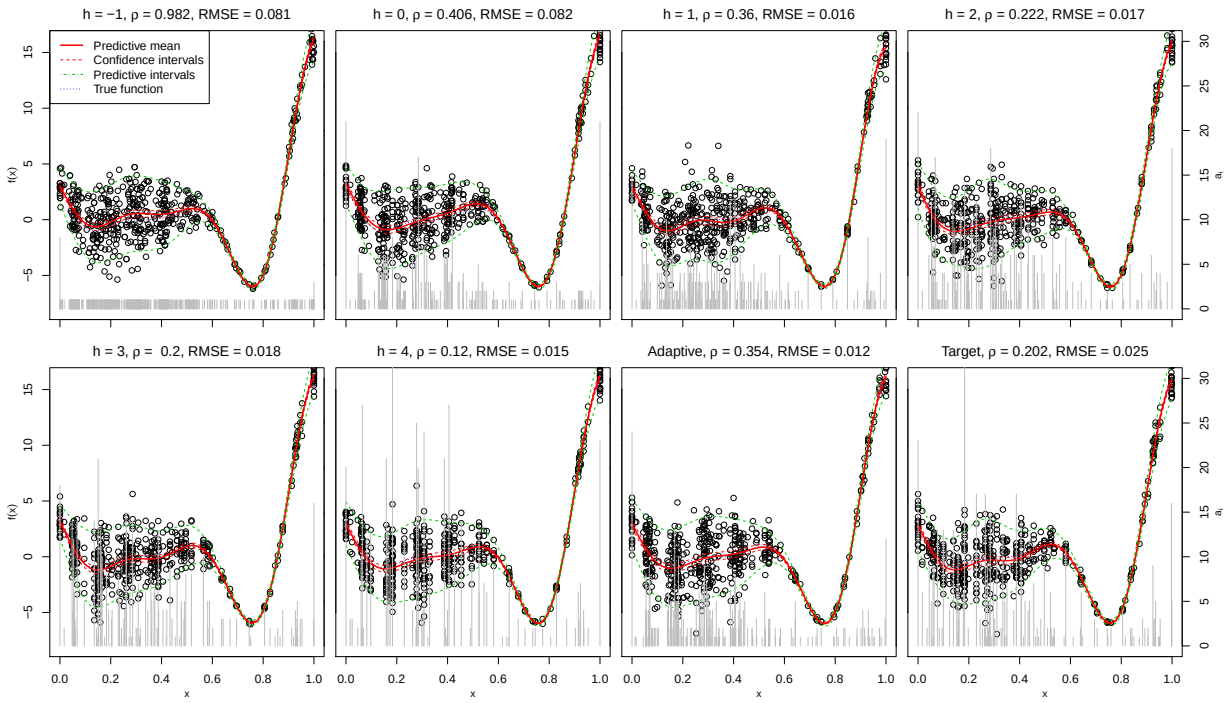


Figure 4: Results on the 1d example by varying the horizon. Grey vertical segments indicate the number of replicates at a given location.

times.

5.2 Synthetic simulation experiment

Here we expand previous 1-d illustrative example by exploring variation over data-generating mechanisms via Monte Carlo (MC) with input space $\mathbf{x} \in [0, 1]$. Using the hyperparameter setting outlined in Section 4.1, we consider a process with noise structure $\mathbf{\Lambda}_n$ sampled as $\log \mathbf{\Lambda}_n \sim \mathrm{GP}(0, \nu_g \mathbf{C}_{(g)})$, where $\mathbf{C}_{(g)}$ is stationary with Matérn 5/2 kernel $k_{(g)}$. Then observations are drawn via $Y|\mathbf{\Lambda}_n \sim \mathrm{GP}(0, \mathbf{K}_n)$, where $\mathbf{K}_n = \nu(\mathbf{C}_n + \mathbf{\Lambda}_n)$ and $\mathbf{C}_n$ is again Matérn 5/2. We set $\theta = 0.1$, and $\nu = 1$ for the mean GP, and $\theta_{(g)} = 0.5$ and $\nu_{(g)} = 7^2$ for the noise GP. To manage the MC variance between runs we normalized the $\mathbf{\Lambda}_n$-values thus obtained so that the average signal-to-noise ratio was one.

We considered a budget of $N = 200$ and studied various strategies for design—comparing one-shot space-filling designs without or with replication to sequential designs with a lookahead horizon of $h = 0$—and for modeling, testing both homoskedastic and heteroskedastic GPs. These are enumerated as follows: (i) homoskedastic GP without replication using an $n = 200$ grid design; (ii) `hetGP` without replication, again with an $n = 200$ grid; (iii) `hetGP` with one-shot space-filling design with random replication on an $n = 40$ grid with random $a_i \in \{1, \ldots, 10\}$; (iv) sequential learning and design using a homoskedastic GP initialized with a single-replicate $n = 40$ grid, iterating until $N = 200$; and (v) sequential learning and

design using `hetGP` initialized with a single-replicate $n = 40$ grid, iterating until $N = 200$.

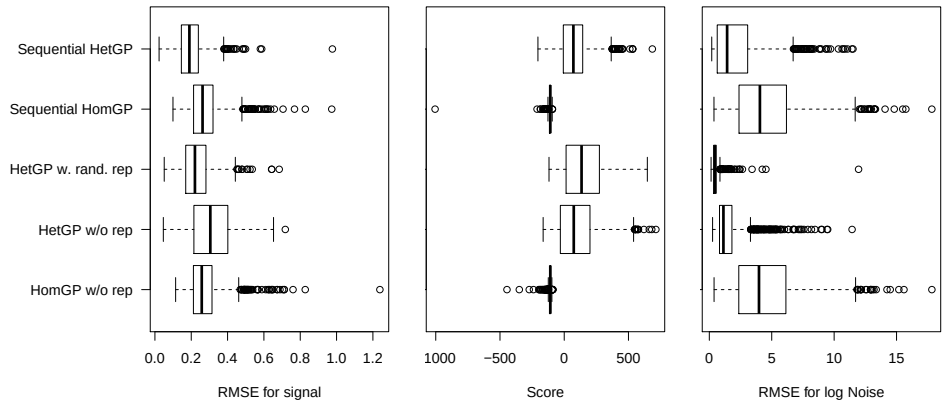


Figure 5: Result from on-dimensional synthetic Monte Carlo experiment in terms of RMSE to the true mean (left), proper scores (middle) and RMSE to true log noise (right).

Figure 5 summarizes the results from 1000 MC replicates, illustrating the subtle balance between replication and exploration. As can be seen in the left panel, our proposed sequential `hetGP` performs the best in terms of out-of-sample RMSE. To investigate the statistical significance of RMSE differences we conducted one-sided matched Wilcoxon signed-rank tests of adjacent performers (better v. next-best) with the order based on median RMSE. The corresponding p-values are 4.80×10^{-24}, 9.66×10^{-26}, 0.0956 and 2.66×10^{-12}. For example, the test involving our best method, `hetGP` with sequential design, versus the second best, `hetGP` with random replication, suggests that the former significantly out-performs the latter. Since our IMSPE design criteria emphasized mean-squared prediction error, it is refreshing that our proposed method wins (significantly) on that metric. The only such comparison which did not "reject the null at the 5% level" involved pitting sequential versus uniform design with a homoskedastic GP. The value of proceeding sequentially is much diminished without the capability to learn a differential noise level.

The center and right panels of the figure show that other design variations may be preferred for other performance metrics. Observe that one-shot space-filling design with random replication using `hetGP` wins when using proper scores. Apparently, random replication yields better estimates of predictive variance when comparing to the truth. See the right panel. Space-filling and uniform replication are easily achieved in this one-dimensional case, but may not port well to higher dimension as our later, more realistic, examples show. Our sequential `hetGP`, coming in second here on score and log noise RMSE, offers more robustness as the input dimension increases.

5.3 Susceptible-Infected-Recovered (SIR) epidemic model

Our first real example deals with estimating the future number of infecteds in a stochastic Susceptible-Infected-Recovered (SIR) epidemic model. This is a standard model for cost-benefit analysis of public health interventions related to communicable diseases, such as

influenza or dengue. For our purposes we treat it as a 2d input space indexed by the count $I_0 \in \mathbb{N}$ of initial infecteds and $S_0 \in \mathbb{N}$ of initial susceptibles (the total population size $M \geq I_0 + S_0$ is pre-fixed; the rest of the population is viewed as immune to the disease). The pair $(I_t, S_t) \in \{S + I \leq M\}$ evolves as a continuous-time Markov chain (easily simulated) following certain non-linear (hence analytically intractable) transition rates, until eventually $I_t = 0$ and the epidemic dies out. The response $f(S, I)$ is the expected aggregate number of infected-days, $\int_0^\infty I_t \, dt$ averaged across the Markov chain trajectories; determining $f(S, I)$ is a first step towards constructing adaptive epidemic response policies. It is important to note that the signal-to-noise ratio is varying drastically over D, with a zero variance at $I = 0$ (where $Y \equiv 0$) and up to $r(\mathbf{x}) \approx 90^2$ on the left part of the domain, in the critical region where the stochasticity in infections leads to either a quick burn-out in infecteds or a rapid infection of a significant population fraction.

Whereas Binois et al. (2016) considered static space-filling designs with random numbers of replicates, with a favorable comparison to SK, here we focus on aspects of sequential design, in particular the effect of horizon h in the IMSPE with lookahead over replication. We perform a Monte Carlo experiment wherein designs are initialized with $n = N = 10$ unique design locations (just one observation each), and grown to size $N = 500$ over sequential design iterations, disregarding how many unique locations n are chosen along the way. A Matérn kernel with $\nu = 5/2$ is used. The experiment is repeated 30 times and averages of various statistics are reported in Figures 6, 7 and Table 1 based on a testing set placed on a dense grid with a thousand replications each.

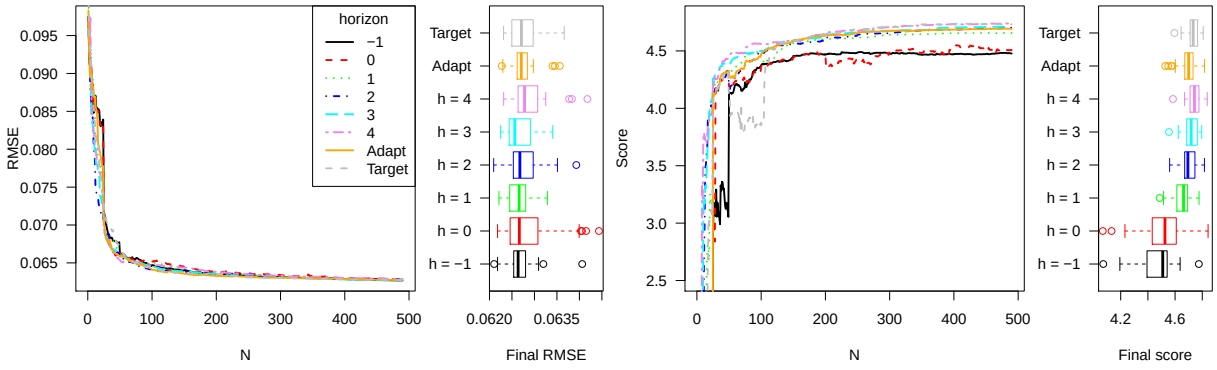


Figure 6: RMSE and score results on the SIR test case over sequential design iterations, via summaries 30 MC repetitions. The Target scheme aims for $\rho = n/N = 0.2$.

In Figure 6, the results in terms of RMSEs and score are presented. While the RMSEs are barely distinguishable, the scores exhibit more spread, and the best results are obtained by the methods leaning the most toward replication (i.e., $h = 4$ and Target scheme). Since the signal-to-noise ratio is low in some parts of the input space, replication is beneficial in terms of RMSE *and score*. One reason for the RMSEs not to be very different between the alternatives is that the underlying function is very smooth. However, the variance surface is more challenging, such that having more replicates is helpful in this case, as highlighted by

the differences in score, shown in the final panel of Figure 6.

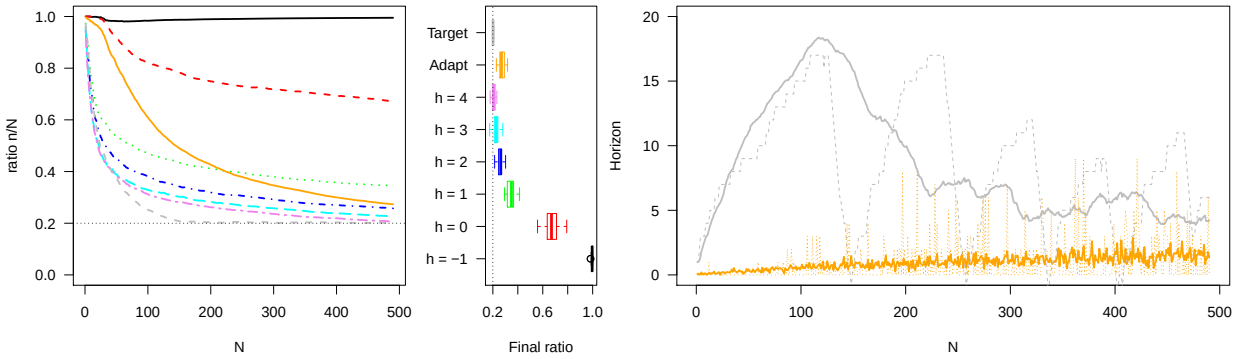


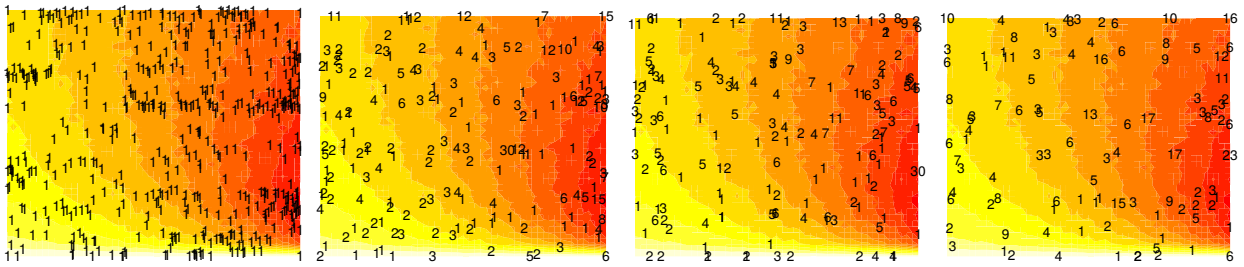
Figure 7: Ratio n/N and horizon evolution on the SIR test case over sequential design iterations, via summaries over 30 MC repetitions. The thin dotted line indicates the Target ratio of 0.2. Right: thin dotted (resp. thick) lines represent one iteration (resp. average) of the horizons for the Adapt and Target schemes. Colors are the same as in Figure 6.

Table 1: Average percentage of designs points with no replicates, with more than five, and running time on the SIR test problem.

Horizon	-1	0	1	2	3	4	Adapt	Target
Percentage of 1s	99.2	49.6	13.6	6.9	4.8	3.5	8.8	4.1
Percentage of 5s and more	0.04	1.6	5.7	6.9	7.7	7.9	6.9	7.6
Time (s)	812	473	278	257	259	271	306	288

Moving on to Figure 7, the left and center panels show the ratio of unique locations over the total design size: n/N. As expected, as the horizon h increases, more replicates are selected. In turn, this lowers the computation time, as reported in Table 1. In particular, observe that the computational cost of looking ahead is negligible next to the cost saved by having smaller n relative to N. The final panel in Figure 7 shows how the horizon h evolves when fixing a Target ratio of $\rho = 0.2$ in (13), i.e., an average of 5 replicates per unique design location) or learning it with the adaptive scheme (15). Notice that the Target scheme with $\rho = 0.2$ sometimes utilizes horizons higher than $h \geq 15$, yet the computational cost is never higher than the high-fixed-horizon results, which offer the best performance for this problem. Due to its random nature, the Adapt scheme changes abruptly between algorithm runs, but its horizon h_N is increasing on average in N.

Figure 8 provides a visual indication of the density of design throughout the input space for fixed and tuned horizons. As expected, in all panels the density of inputs in the design is higher in high variance parts of the input space. The numbers in the plot indicate the numbers of replicates a_i. Observe that low-horizon heuristics result in mostly $a_i = 1$, whereas for the longer horizons clusters of tightly grouped unique locations are replaced with replicates. Table 1 demonstrates that this feature is consistent over MC repetitions. Thus,



(a) $h = -1$ (b) $h = 1$ (c) Adapt (d) Target

Figure 8: Designs obtained with different strategies for the horizon, where numbers indicate how many replicates a_i are performed at a given location $\bar{\mathbf{x}}_i$. Darker colors indicate higher variance. The x-axis is the number of susceptibles, from 1200 to 2000 while the y-axis is the initial number of infecteds, from, 0 to 200.

our heuristic is adept at capturing the basic logic of amalgamating singleton design locations into replicates, which apparently maintains essentially the same statistical efficiency while reducing computational overhead by a factor of more than 3.

5.4 Inventory management

The assemble to order (ATO) simulation, first introduced by Hong and Nelson (2006) with implementation in MATLAB later provided by Xie et al. (2012), comes from inventory management. The inputs determine stocks and replenishment schedules for key items in assembled products, and the simulator estimates revenue by combining inventory costs with profits obtained from orders which come in following a compound Poisson random process. Binois et al. (2016) showed the benefit of heteroskedastic modeling, versus several homoskedastic alternatives, on random space-filling designs with $n = 1000$ unique locations with a random number of replications (uniform in $1, \ldots, 10$) so that the average full data size was $N = 5000$. Here, one of our aims is to illustrate that by building a better design (sequentially), a much lower N is possible without sacrificing accuracy. Binois et al. used a proper scoring rule (Gneiting and Raftery, 2007, Eq. (27)) as their main metric. Since our IMSPE criterion targets accuracy via squared-error loss we report RMSEs, but include scores to facilitate comparison to those space-filling designs. The best average score reported in Figure 2 of that paper was 3.3, with a min and max of 2.8 and 3.6 respectively.

Similar to the SIR experiment, we perform the following variations on sequential IMSPE design, varying the horizon, h, of lookahead and offering the two adaptive horizon schemes outlined in Section 3.3. We initialize with $n = 100$ unique space-filling locations and a random number of replicates, uniform in $\{1, \ldots, 10\}$ so that the starting size is $N = 500$ on average. Subsequently, sequential design iterations are performed until $N = 2000$ total samples are gathered, irregardless of how many unique locations, n, result. The experiment is repeated in a Monte Carlo fashion, with thirty repeats.

Figure 9 summarizes the results of the experiment in a format similar to Figure 6. The take-home message is fairly evident: in contrast to the SIR example, shorter lookahead

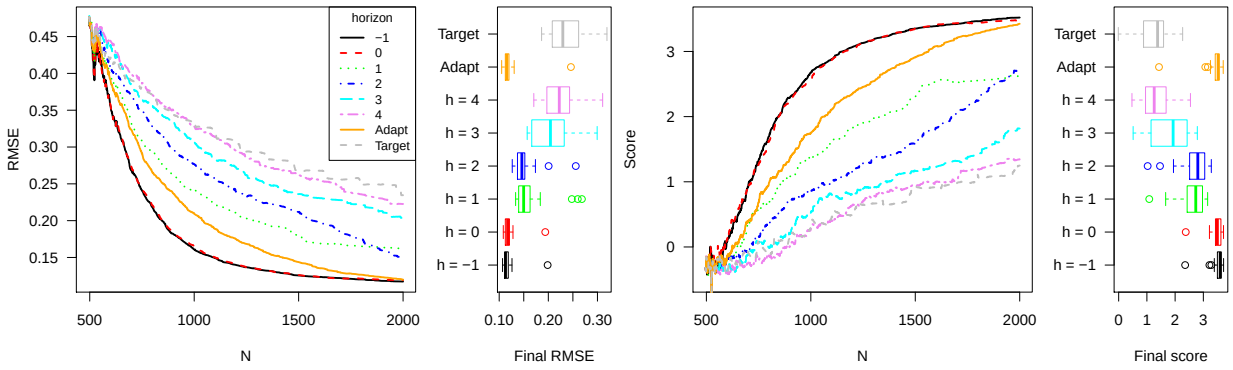


Figure 9: RMSE and score results on the ATO problem, via thirty Monte Carlo iterations, in a format similar to Figure 6.

horizon is better, owing to the relatively higher signal-to-noise ratio. Observe that our average score is 3.5, and the min and max are 2.4 and 3.7 respectively. So scores based on just $N = 2000$ samples are higher than in the space-filling $N = 5000$ experiment, however the spread is a little wider. Finally, note that the Adapt heuristic (15) eventually performs as well as the best horizon ($h = -1$). Targeting $\rho = 0.2$ via (13), by contrast, leads to far too little exploration. The Adapt scheme required an average of 682 minutes to build up to a design of size $N = 2000$ with an average of $n = 1086$ unique sites (min and max of 465 and 1211 respectively), whereas Target took only 183 minutes thanks to using $n = 400$ locations on average (399 to 405).

6 Conclusion and perspectives

This paper addresses a design question which has been around the surrogate modeling literature, and informally in the community, for many years. There is general agreement that the "fully batched" version of the problem, of finding n locations and numbers of replicates $a_1, \ldots, a_n$ on each, is not computationally tractable, although there are some attempts in the recent literature. We therefore consider the simpler task of deciding whether the *next* sample should explore or replicate in a sequential design context. The condition we derive is simple to express, and leads to an intuitive suite of visualizations. Proceeding sequentially has merits, not only computationally but also facilitating "as you go" adjustments to help avoid pathologies arising from feedbacks between design and inference. However the procedure is still myopic. To help correct this we introduced a computationally tractable lookahead scheme that emphasizes the role of replication in design. Tuning the horizon of that scheme allows the user to trade off the dual roles of replication in surrogate modeling design: computational thriftiness of inference against out-of-sample accuracy, although as we show these are not always at odds.

Our presentation focused on the integrated mean-squared prediction error (IMSPE) criteria. We chose IMSPE because it is popular, but also because it leads to closed form

derivatives for optimization, updating equations for search over look-ahead horizons, and simplifications for entertaining replicates. There is, of course, a vast literature on model-based design criteria (see, e.g., Chen and Zhou, 2017; Kleijnen, 2015) targeting alternative quantities of interest, such as entropy or information for unknown model parameters. Although designs for prediction and estimation sometimes coincide, like for linear regression, the correlation structure for GPs can be a game-changer (Müller et al., 2012). It may well be that other criteria lead to strategies similar to ours, which may be an interesting avenue for future research.

Our implementation and empirical work leveraged a new heteroskedastic Gaussian process modeling library called `hetGP`, available for R on CRAN (Binois and Gramacy, 2017). Our IMSPE, updates, lookahead procedures, and more are provided in a recently updated version of the package. To aid in reproducibility, our supplementary material contains codes using that library to reproduce the smaller examples from the paper [Figures 2–4]. The other examples require rather more computing, and/or linking between R and MATLAB for simulation [ATO], which somewhat challenges ease of replication. However, we are happy to provide those codes upon request.

Processes (i.e., data generating mechanisms) benefiting from a heteroskedastic feature bring out the best in our sequential design schemes, demanding a greater degree of replication in high-noise regions relative to low-noise ones, confirming the intuition that replication becomes more valuable for separating signal from noise as the data get noisier (e.g., Wang and Haaland, 2017). However, the results we provide are just as valid in the homoskedastic setting, albeit with somewhat less flair. In that context, inferring the right level of replication is a global affair, except perhaps at the edges of the input space which tend to prefer a slightly higher degree.

Our three sets of examples illustrated that the method both does what it is designed to do, and that designs with the right trade-off between exploration and replication perform better than ones which are designed more naïvely. These examples span a range of features, from low to high noise (and slow to rapid change in noise), low to moderate input dimension, and synthetic to real simulation experiments. The behavior is diverse but the results are consistent: sequential design with lookahead-based IMSPE leads to accurate prediction, and the slight bias toward replication yields computationally more thrifty predictors without a compromise on accuracy. However, sequential design might not always be appropriate. Sometimes batching, at least to a small degree, cannot be avoided. Addressing this situation represents an exciting avenue for further research.

Acknowledgments

We thank the anonymous reviewers for helpful comments on the earlier version of the paper. All four authors are grateful for support from National Science Foundation grant DMS-1521702 and DMS-1521743.

References

Anagnostopoulos, C. and Gramacy, R. (2013). "Information-Theoretic Data Discarding for Dynamic Trees on Data Streams." *Entropy*, 15, 12, 5510–5535. ArXiv:1201.5568.

Ankenman, B., Nelson, B. L., and Staum, J. (2010). "Stochastic kriging for simulation metamodeling." *Operations research*, 58, 2, 371–382.

Antognini, A. B. and Zagoraiou, M. (2010). "Exact optimal designs for computer experiments via Kriging metamodelling." *Journal of Statistical Planning and Inference*, 140, 9, 2607–2617.

Bachoc, F. (2013). "Cross Validation and Maximum Likelihood estimations of hyper-parameters of Gaussian processes with model misspecification." *Computational Statistics & Data Analysis*, 66, 55–69.

Barnett, S. (1979). *Matrix Methods for Engineers and Scientists*. McGraw-Hill.

Binois, M. and Gramacy, R. B. (2017). hetGP*: Heteroskedastic Gaussian Process Modeling and Design under Replication*. R package version 1.0.0.

Binois, M., Gramacy, R. B., and Ludkovski, M. (2016). "Practical heteroskedastic Gaussian process modeling for large simulation experiments." *arXiv preprint arXiv:1611.05902*.

Boukouvalas, A. (2010). "Emulation of random output simulators." Ph.D. thesis, Aston University.

Boukouvalas, A., Cornford, D., and Stehlík, M. (2014). "Optimal design for correlated processes with input-dependent noise." *Computational Statistics & Data Analysis*, 71, 1088–1102.

Burnaev, E. and Panov, M. (2015). "Adaptive design of experiments based on Gaussian processes." In *Statistical Learning and Data Sciences*, 116–125. Springer.

Chen, X. and Zhou, Q. (2014). "Sequential experimental designs for stochastic kriging." In *Proceedings of the 2014 Winter Simulation Conference*, 3821–3832. IEEE Press.

— (2017). "Sequential design strategies for mean response surface metamodeling via stochastic kriging with adaptive exploration and exploitation." *European Journal of Operational Research*, 262, 2, 575–585.

Chevalier, C., Ginsbourger, D., and Emery, X. (2014). "Corrected kriging update formulae for batch-sequential data assimilation." In *Mathematics of Planet Earth*, 119–122. Springer.

Cioffi-Revilla, C. (2014). *Introduction to computational social science*. Berlin/New York: Springer.

Das, A. and Kempe, D. (2008). "Algorithms for subset selection in linear regression." In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, 45–54. ACM.

Forrester, A., Sobester, A., and Keane, A. (2008). *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons.

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

Ginsbourger, D. and Le Riche, R. (2010). "Towards Gaussian process-based optimization with finite time horizon." In *mODa 9–Advances in Model-Oriented Design and Analysis*, 89–96. Springer.

Gneiting, T. and Raftery, A. E. (2007). "Strictly proper scoring rules, prediction, and estimation." *Journal of the American Statistical Association*, 102, 477, 359–378.

Goldberg, P. W., Williams, C. K., and Bishop, C. M. (1998). "Regression with input-dependent noise: A Gaussian process treatment." In *Advances in Neural Information Processing Systems*, vol. 10, 493–499. Cambridge, MA: MIT press.

Gonzalez, J., Osborne, M., and Lawrence, N. (2016). "GLASSES: Relieving The Myopia Of Bayesian Optimisation." In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 790–799.

Gorodetsky, A. and Marzouk, Y. (2016). "Mercer kernels and integrated variance experimental design: connections between Gaussian process regression and polynomial approximation." *SIAM/ASA Journal on Uncertainty Quantification*, 4, 1, 796–828.

Gramacy, R. and Polson, N. (2011). "Particle Learning of Gaussian Process Models for Sequential Design and Optimization." *Journal of Computational and Graphical Statistics*, 20, 1, 102–118.

Gramacy, R. B. and Lee, H. K. H. (2009). "Adaptive Design and Analysis of Supercomputer Experiment." *Technometrics*, 51, 2, 130–145.

Hong, L. and Nelson, B. (2006). "Discrete optimization via simulation using COMPASS." *Operations Research*, 54, 1, 115–129.

Horn, D., Dagge, M., Sun, X., and Bischl, B. (2017). "First Investigations on Noisy Model-Based Multi-objective Optimization." In *International Conference on Evolutionary Multi-Criterion Optimization*, 298–313. Springer.

Huan, X. and Marzouk, Y. M. (2016). "Sequential Bayesian optimal experimental design via approximate dynamic programming." *arXiv preprint arXiv:1604.08320*.

Jalali, H., Nieuwenhuyse, I. V., and Picheny, V. (2017). "Comparison of Kriging-based algorithms for simulation optimization with heterogeneous noise." *European Journal of Operational Research*, 261, 1, 279 – 301.

Johnson, L. (2008). "Microcolony and Biofilm Formation as a Survival Strategy for Bacteria." *Journal of Theoretical Biology*, 251, 24–34.

Kamiński, B. (2015). "A method for the updating of stochastic Kriging metamodels." *European Journal of Operational Research*, 247, 3, 859–866.

Kersting, K., Plagemann, C., Pfaff, P., and Burgard, W. (2007). "Most likely heteroscedastic Gaussian process regression." In *Proceedings of the International Conference on Machine Learning*, 393–400. New York, NY: ACM.

Kleijnen, J. P. (2015). *Design and Analysis of Simulation Experiments*, vol. 230. Springer.

Krause, A. and Guestrin, C. (2007). "Nonmyopic active learning of gaussian processes: an exploration-exploitation approach." In *Proceedings of the 24th international conference on Machine learning*, 449–456. ACM.

Krause, A., Singh, A., and Guestrin, C. (2008). "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies." *Journal of Machine Learning Research*, 9, Feb, 235–284.

Lam, R., Willcox, K., and Wolpert, D. H. (2016). "Bayesian Optimization with a Finite Budget: An Approximate Dynamic Programming Approach." In *Advances In Neural Information Processing Systems*, 883–891.

Law, A. M. (2015). *Simulation Modeling and Analysis*. 5th ed. McGraw-Hill.

Leatherman, E. R., Santner, T. J., and Dean, A. M. (2017). "Computer experiment designs for accurate prediction." *Statistics and Computing*, 1–13.

Liu, M. and Staum, J. (2010). "Stochastic kriging for efficient nested simulation of expected shortfall." *The Journal of Risk*, 12, 3, 3.

Mehdad, E. and Kleijnen, J. P. (2018). "Stochastic intrinsic Kriging for simulation metamodeling." *Applied Stochastic Models in Business and Industry*, in press.

Müller, W. G., Pronzato, L., and Waldl, H. (2012). "Relations between designs for prediction and estimation in random fields: an illustrative case." In *Advances and Challenges in Space-time Modelling of Natural Events*, 125–139. Springer.

Petersen, K. B., Pedersen, M. S., et al. (2008). "The matrix cookbook." *Technical University of Denmark*, 7, 15.

Picheny, V. and Ginsbourger, D. (2013). "A nonstationary space-time Gaussian process model for partially converged simulations." *SIAM/ASA Journal on Uncertainty Quantification*, 1, 57–78.

Plumlee, M. and Tuo, R. (2014). "Building accurate emulators for stochastic simulations via quantile Kriging." *Technometrics*, 56, 4, 466–473.

Pratola, M. T., Harari, O., Bingham, D., and Flowers, G. E. (2017). "Design and Analysis of Experiments on Nonconvex Regions." *Technometrics*, 1–12.

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

Quan, N., Yin, J., Ng, S. H., and Lee, L. H. (2013). "Simulation optimization via kriging: a sequential search using expected improvement with computing budget constraints." *IIE Transactions*, 45, 7, 763–780.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rasmussen, C. E. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). "Design and analysis of computer experiments." *Statistical science*, 4, 4, 409–423.

Santner, T. J., Williams, B. J., and Notz, W. I. (2013). *The design and analysis of computer experiments*. Springer Science & Business Media.

Seo, S., Wallat, M., Graepel, T., and Obermayer, K. (2000). "Gaussian Process Regression: Active Data Selection and Test Point Rejection." In *Proceedings of the International Joint Conference on Neural Networks*, vol. III, 241–246. IEEE.

Wang, W. and Haaland, B. (2017). "Controlling Sources of Inaccuracy in Stochastic Kriging." *arXiv preprint arXiv:1706.00886*.

Weaver, B. P., Williams, B. J., Anderson-Cook, C. M., Higdon, D. M., et al. (2016). "Computational enhancements to Bayesian design of experiments using Gaussian processes." *Bayesian Analysis*, 11, 1, 191–213.

Xie, J., Frazier, P., and Chick, S. (2012). "Assemble to Order Simulator."

A Detailed gradient expressions

Here we provide expressions for the Section 3.2 discussion on the gradient of the IMSPE.

$$\frac{\partial \mathbf{K}_{n+1}^{-1}}{\partial \tilde{\mathbf{x}}} = \frac{\partial}{\partial \tilde{\mathbf{x}}} \begin{bmatrix} \mathbf{K}_n^{-1} + \mathbf{g}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top \sigma_n^2(\tilde{\mathbf{x}}) & \mathbf{g}(\tilde{\mathbf{x}}) \\ \mathbf{g}(\tilde{\mathbf{x}})^\top & \sigma_n^2(\tilde{\mathbf{x}})^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{H}(\tilde{\mathbf{x}}) & \mathbf{h}(\tilde{\mathbf{x}}) \\ \mathbf{h}(\tilde{\mathbf{x}})^\top & v_1(\tilde{\mathbf{x}}) \end{bmatrix} \quad \text{as in Eq. (10),}$$

$$\text{where } v_1(\tilde{\mathbf{x}}) := \frac{\partial \sigma_n^2(\tilde{\mathbf{x}})^{-1}}{\partial \tilde{\mathbf{x}}_{(p)}} = \frac{2\mathbf{d}(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1}\mathbf{k}(\tilde{\mathbf{x}}) + \frac{\partial r(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}_{(p)}}}{\left(k(\tilde{\mathbf{x}},\tilde{\mathbf{x}}) - \mathbf{k}(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1}\mathbf{k}(\tilde{\mathbf{x}}) + r(\tilde{\mathbf{x}})\right)^2}, \quad \mathbf{d}(\tilde{\mathbf{x}}) := \frac{\partial \mathbf{k}(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}_{(p)}}$$

$$\mathbf{h}(\tilde{\mathbf{x}}) := \frac{\partial \mathbf{g}(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}_{(p)}} = -\mathbf{K}_n^{-1}\left(v_1(\tilde{\mathbf{x}})\mathbf{k}(\tilde{\mathbf{x}}) + \sigma_n^2(\tilde{\mathbf{x}})^{-1}\mathbf{d}(\tilde{\mathbf{x}})\right)$$

$$\begin{aligned} \mathbf{H}(\tilde{\mathbf{x}}) &:= \frac{\partial \mathbf{g}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top \sigma_n^2(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}_{(p)}} = \frac{\partial \sigma_n^2(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}_{(p)}}\mathbf{g}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top + \sigma_n^2(\tilde{\mathbf{x}})\mathbf{h}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top + \sigma_n^2(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})\mathbf{h}(\tilde{\mathbf{x}})^\top \\ &= v_2(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top + \sigma_n^2(\tilde{\mathbf{x}})\left(\mathbf{h}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top + (\mathbf{h}(\tilde{\mathbf{x}})\mathbf{g}(\tilde{\mathbf{x}})^\top)^\top\right), \end{aligned}$$

and $v_2(\tilde{\mathbf{x}}) = -2\mathbf{d}(\tilde{\mathbf{x}})^\top \mathbf{K}_n^{-1}\mathbf{k}(\tilde{\mathbf{x}})$. Similarly, since $\mathbf{W}_n$ does not depend on $\tilde{\mathbf{x}}$:

$$\frac{\partial \mathbf{W}_{n+1}}{\partial \tilde{\mathbf{x}}_{(p)}} = \begin{bmatrix} \mathbf{0}_{n\times n} & \mathbf{c}_1(\tilde{\mathbf{x}}) \\ \mathbf{c}_1(\tilde{\mathbf{x}})^\top & c_2(\tilde{\mathbf{x}}) \end{bmatrix}, \qquad \text{as presented in Eq. (11).}$$

Expressions for $\mathbf{c}_1(\cdot)$ and $c_2(\cdot)$ for particular kernels may be found in Appendix B.

B Expressions for common kernels

We consider here four kernels common in practice: Gaussian (or squared exponential) and Matérn with parameter $\alpha = 5/2,\ 3/2,\ 1/2$ (the last one being the exponential kernel) and give the corresponding expressions for E, w, $\mathbf{d}$, $\mathbf{c}_1$ and c_2 as introduced in Section 3. Notice that all these kernels are stationary, i.e., $k(\mathbf{x},\mathbf{x}') = \nu c(\mathbf{x}-\mathbf{x}')$ with ν the process variance and c the correlation function. As a consequence, $E = \int_{\mathbf{x}\in D} k(\mathbf{x},\mathbf{x})\, d\mathbf{x} = \int_{\mathbf{x}\in D} \nu c(\mathbf{0}) d\mathbf{x} = \nu$.

In their separable form, over $D = [0,1]^d$, these kernel write $k(\mathbf{x},\mathbf{x}') = \nu \prod_{p=1}^d k_i(x_p, x'_p)$ with k_i one of the aforementioned kernel. By using the separability we get:

$$\nu w(\mathbf{x}_i,\mathbf{x}_j) = \int_{\mathbf{x}\in D} k(\mathbf{x}_i,\mathbf{x})k(\mathbf{x}_j,\mathbf{x})d\mathbf{x} = \nu \prod_{p=1}^d \int_{x\in[0,1]} k_i(x_{i,p},x)k_i(x_{j,p},x)dx = \nu \prod_{p=1}^d w_p(x_{i,p},x_{j,p}).$$

Below we provide our parameterization of these kernels in univariate form along with the corresponding expressions for w, $\mathbf{d}$, $\mathbf{c}_1$ and c_2.

B.1 Gaussian kernel

The univariate Gaussian kernel is $k_G(x,x') = \exp\left(-\frac{(x-x')^2}{\theta}\right)$. Therefore:

$$d_i = \frac{\partial k_G(x_i,x)}{\partial x} = \frac{2(x_i-x)}{\theta}\exp\left(-\frac{(x_i-x)^2}{\theta}\right)$$

$$w(x_i, x_j) = \frac{\sqrt{2\pi\theta}}{4} \exp\left(-\frac{(x_i - x_j)^2}{2\theta}\right)\left(\text{erf}\left(\frac{2 - (x_i + x_j)}{\sqrt{2\theta}}\right) + \text{erf}\left(\frac{x_i + x_j}{\sqrt{2\theta}}\right)\right),\ 1 \le i, j \le n$$

with erf the error function. In addition:

$$c_2 = \frac{\partial w(x_i, x_i)}{\partial x_i} = \exp\left(-\frac{2x_i^2}{\theta}\right) - \exp\left(-\frac{(1 - 2x_i)^2}{\theta}\right)$$

and, for the vector $\mathbf{c}_1$, $1 \le i \le n$:

$$\frac{\partial w(x, x_i)}{\partial x} = \sqrt{\frac{\pi}{8\theta}} \exp\left(-\frac{(x - x_i)^2}{2\theta}\right)\left[(x - x_i)\left(\text{erf}\left(\frac{x + x_i - 2}{\sqrt{2\theta}}\right) - \text{erf}\left(\frac{x + x_i}{\sqrt{2\theta}}\right)\right)\right.$$
$$\left. + \sqrt{\frac{2\theta}{\pi}}\left(\exp\left(-\frac{(x + x_i)^2}{2\theta}\right) - \exp\left(\frac{-(x + x_i - 2)^2}{2\theta}\right)\right)\right].$$

Remark: this is the kernel used in Figure 1, with hyperparameters $\nu = 1$, $\theta = 0.01$.

B.2 Matérn kernels with $\alpha = \{1, 3, 5\}/2$

We use the following parameterization of the Matérn kernel for specific values of α:

$$k_{M,1/2}(x, x') = \exp\left(-\frac{|x - x'|}{\theta}\right)$$

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

$$k_{M,5/2}(x, x') = \left(1 + \frac{\sqrt{5}|x - x'|}{\theta} + \frac{5(x - x')^2}{2\theta^2}\right)\exp\left(-\frac{\sqrt{5}|x - x'|}{\theta}\right)$$

The derivatives with respect to x, i.e., in $\mathbf{d}$ are:

$$\frac{\partial k_{M,1/2}(x, x')}{\partial x} = \frac{(-1)^{\delta_{x<x'}}}{\theta}\exp\left(-\frac{|x - x'|}{\theta}\right)$$

$$\frac{\partial k_{M,3/2}(x, x')}{\partial x} = \frac{(-1)^{\delta_{x<x'}} \times 3|x - x'|}{\theta^2}\exp\left(-\frac{\sqrt{3}|x - x'|}{\theta}\right)$$

$$\frac{\partial k_{M,5/2}(x, x')}{\partial x} = (-1)^{\delta_{x<x'}}\frac{\left(\frac{10}{3} - 5\right)|x - x'| - \frac{5\sqrt{5}}{3\theta}(x - x')^2}{\theta^2}\exp\left(-\frac{\sqrt{5}|x - x'|}{\theta}\right)$$

To get closed form derivatives of $w(x_i, x_j)$ in Lemma 3.1, first consider $x_i \le x_j$ to drop absolute values, then divide integration into components p_1 $(0 \to x_i)$, p_2 $(x_i \to x_j)$, p_3 $(x_j \to 1)$. We rely on symbolic solvers for the most tedious components, see e.g., `https:`

//www.integral-calculator.com/. To reduce expression, define $\beta = \exp\left(\frac{2\sqrt{3}}{\theta}\right)$ and $\gamma = \exp\left(\frac{2\sqrt{5}}{\theta}\right)$.

The first term is given by:

$$p_{1,1/2} = \int_0^{x_i} \exp\left(-\frac{(x_i - x)}{\theta}\right) \exp\left(-\frac{(x_j - x)}{\theta}\right) dx = \frac{\theta}{2}\left(\exp\left(\frac{2x_i}{\theta}\right) - 1\right) \exp\left(\frac{-x_j - x_i}{\theta}\right),$$

and similarly

$$p_{1,3/2} = \frac{1}{12\theta}\left[\left(\theta\left(5\sqrt{3}\theta + 9x_j - 9x_j\right)\exp\left(\frac{2\sqrt{3}x_i}{\theta}\right) - 5\sqrt{3}\theta^2 - 9\left(x_j + x_i\right)\theta \right.\right.$$
$$\left.\left. -2{\cdot}3^{\frac{3}{2}}x_i x_j\right)\exp\left(-\frac{\sqrt{3}\left(x_i + x_j\right)}{\theta}\right)\right]$$

$$p_{1,5/2}\cdot t_1 = \theta^2\left(63\theta^2 + 9{\cdot}5^{\frac{3}{2}}x_j\theta - 9{\cdot}5^{\frac{3}{2}}x_i\theta + 50x_j^2 - 100x_i x_j + 50x_i^2\right)\exp\left(\frac{2\sqrt{5}x_i}{\theta}\right)$$
$$- 63\theta^4 - 9{\cdot}5^{\frac{3}{2}}\left(x_j + x_i\right)\theta^3 - 10\left(5x_j^2 + 17x_i x_j + 5x_i^2\right)\theta^2 - 8{\cdot}5^{\frac{3}{2}}x_i x_j\left(x_j + x_i\right)\theta - 50x_i^2 x_j^2,$$

with $t_1 = 36\sqrt{5}\theta^3\exp\left(\frac{\sqrt{5}(x_j + x_i)}{\theta}\right)$.

The second term:

$$p_{2,1/2} = \int_{x_i}^{x_j} \exp\left(-\frac{(x - x_i)}{\theta}\right)\exp\left(-\frac{(x_j - x)}{\theta}\right) = \left(x_j - x_i\right)\exp\left(-\frac{x_j - x_i}{\theta}\right)$$

$$p_{2,3/2} = \frac{\left(x_j - x_i\right)\left(2\theta^2 + 2\sqrt{3}\left(x_j - x_i\right)\theta + x_j^2 - 2x_i x_j + x_i^2\right)\exp\left(-\frac{\sqrt{3}(x_j - x_i)}{\theta}\right)}{2\theta^2}$$

$$p_{2,5/2}\cdot t_2 = \left(x_j - x_i\right)54\theta^4 + \left(54\sqrt{5}x_j - 54\sqrt{5}x_i\right)\theta^3 + \left(105x_j^2 - 210x_i x_j + 105x_i^2\right)\theta^2$$
$$+ \left(3{\cdot}5^{\frac{3}{2}}x_j^3 - 9{\cdot}5^{\frac{3}{2}}x_i x_j^2 + 9{\cdot}5^{\frac{3}{2}}x_i^2 x_j - 3{\cdot}5^{\frac{3}{2}}x_i^3\right)\theta + 5x_j^4 - 20x_i x_j^3 + 30x_i^2 x_j^2 - 20x_i^3 x_j + 5x_i^4$$

with $t_2 = 54\theta^4\exp\left(\frac{\sqrt{5}(x_i - x_j)}{\theta}\right)$.

The third term:

$$p_{3,1/2} = \int_{x_j}^{1} \exp\left(-\frac{(x - x_i)}{\theta}\right)\exp\left(-\frac{(x - x_j)}{\theta}\right)$$
$$= \frac{\theta}{2}\left(\exp\left(\frac{x_i - x_j}{\theta}\right) - \exp\left(\frac{x_j + x_i - 2}{\theta}\right)\right)$$

$$p_{3,3/2}\cdot t_3 = \theta\left(5\theta + 3^{\frac{3}{2}}\left(x_j - x_i\right)\right)\beta$$
$$- \left(\theta\left(5\theta - 3^{\frac{3}{2}}\left(x_j + x_i - 2\right)\right) + 6\left(x_i - 1\right)x_j - 6x_i + 6\right)\exp\left(\frac{2\sqrt{3}x_j}{\theta}\right)$$

$$p_{3,5/2}\cdot t_4 = \exp\left(\frac{2\sqrt{5}x_j}{\theta}\right)\cdot\Big[\theta\left(\theta\left(9\theta\left(7\theta - 5^{\frac{3}{2}}\left(x_j + x_i - 2\right)\right) + 10x_j\left(5x_j + 17x_i - 27\right)\right.\right.$$
$$\left.\left. +10\left(5x_i^2 - 27x_i + 27\right)\right) - 8{\cdot}5^{\frac{3}{2}}\left(x_i - 1\right)\left(x_j - 1\right)\left(x_j + x_i - 2\right)\right) + 50\left(x_i - 1\right)^2\left(x_j - 2\right)x_j$$
$$+50\left(x_i - 1\right)^2\Big] - \theta^2\left(63\theta^2 + 9{\cdot}5^{\frac{3}{2}}x_j\theta - 9{\cdot}5^{\frac{3}{2}}x_i\theta + 50x_j^2 - 100x_ix_j + 50x_i^2\right)\gamma$$

with $t_3 = 4\theta\sqrt{3}\exp\left(\frac{\sqrt{3}(x_j - x_i + 2)}{\theta}\right)$, $t_4 = -36\sqrt{5}\theta^3\exp\left(\frac{\sqrt{5}(x_j - x_i + 2)}{\theta}\right)$.

The case when $x_i > x_j$ is obtained by swapping x_i and x_j above. Derivatives with respect to x_i and x_j, to account for both of these cases, are provided as follows:

$$\frac{\partial w_{1/2}(x_i, x_j)}{\partial x_i} = -\frac{\left(2\left(x_i + \theta - x_j\right)\exp\left(\frac{2x_i}{\theta}\right) + \theta\exp\left(\frac{2x_j}{\theta}\right) - \theta\right)\exp\left(-\frac{x_i + x_j}{\theta}\right)}{2\theta}$$

$$\frac{\partial w_{1/2}(x_i, x_j)}{\partial x_j} = \frac{\left(\theta\exp\left(\frac{2x_j}{\theta}\right) - 2\exp\left(\frac{2x_i}{\theta}\right)x_j + 2\left(\theta + x_i\right)\exp\left(\frac{2x_i}{\theta}\right) + \theta\right)\exp\left(-\frac{x_j + x_i}{\theta}\right)}{2\theta}$$

$$\frac{\partial w_{3/2}(x_i, x_j)}{\partial x_i}t_5 = \exp\left(\frac{2\sqrt{3}x_i}{\theta}\right)\Big[2\sqrt{3}\beta x_i^3 + \left(-6\theta - 2{\cdot}3^{\frac{3}{2}}x_j\right)\beta x_i^2 +$$
$$+\left(\left(\left(6x_j - 6\right)\theta - 3^{\frac{3}{2}}\theta^2\right)\exp\left(\frac{2\sqrt{3}x_j}{\theta}\right) + \left(2\sqrt{3}\theta^2 + 12x_j\theta + 2{\cdot}3^{\frac{3}{2}}x_j^2\right)\beta\right)x_i +$$
$$\left(2\theta^3 + \left(4\sqrt{3} - \sqrt{3}x_j\right)\theta^2 + \left(6 - 6x_j\right)\theta\right)\exp\left(\frac{2\sqrt{3}x_j}{\theta}\right) + \left(-2\sqrt{3}x_j\theta^2 - 6x_j^2\theta - 2\sqrt{3}x_j^3\right)\beta\Big]$$
$$+\left(-3^{\frac{3}{2}}\theta^2 - 6x_jx_i\right)\beta x_i + \left(-2s^3 - \sqrt{3}x_j\theta^2\right)\beta$$

$$\frac{\partial w_{3/2}(x_i, x_j)}{\partial x_j}t_6 = \theta\left[\left(3^{\frac{3}{2}}\theta - 6x_i + 6\right)x_j - \theta\left(2\theta - \sqrt{3}\left(x_i - 4\right)\right) + 6x_i - 6\right]\exp\left(\frac{2\sqrt{3}\left(x_j + x_i\right)}{\theta}\right)$$
$$- 2\sqrt{3}\exp\left(\frac{2\sqrt{3}\left(x_i + 1\right)}{\theta}\right)x_j^3 - 2\left(3\theta - 3^{\frac{3}{2}}x_i\right)\exp\left(\frac{2\sqrt{3}\left(x_i + 1\right)}{\theta}\right)x_j^2 -$$
$$\beta\left(2\sqrt{3}\theta^2\exp\left(\frac{2\sqrt{3}x_i}{\theta}\right) - 12x_i\theta\exp\left(\frac{2\sqrt{3}x_i}{\theta}\right) + 2{\cdot}3^{\frac{3}{2}}x_i^2\exp\left(\frac{2\sqrt{3}x_i}{\theta}\right) - 3^{\frac{3}{2}}\theta^2 - 6x_i\theta\right)x_j$$
$$+ 2x_i\left(\sqrt{3}\theta^2 - 3x_i\theta + \sqrt{3}x_i^2\right)\exp\left(\frac{2\sqrt{3}\left(x_i + 1\right)}{\theta}\right) + \theta^2\left(2\theta + \sqrt{3}x_i\right)\beta$$

with $t_5 = -4\theta^3 \exp\left(\frac{\sqrt{3}(x_i+x_j+2)}{\theta}\right)$, $t_6 = -t_5$.

$$\frac{\partial w_{5/2}(x_i,x_j)}{\partial x_i} t_7 = \Big[2\cdot 5^{\frac{3}{2}}\gamma x_i^5 + \left(-100\theta - 2\cdot 5^{\frac{5}{2}}x_j\right)\gamma x_i^4 + \left(18\cdot 5^{\frac{3}{2}}\theta^2 + 400x_j\theta + 4\cdot 5^{\frac{5}{2}}x_j^2\right)\gamma x_i^3 + \left(\left(150\theta^3 + \left(24\cdot 5^{\frac{3}{2}} - 24\cdot 5^{\frac{3}{2}}x_j\right)\theta^2 + \left(150x_j^2 - 300x_j + 150\right)\theta\right)\exp\left(\frac{2\sqrt{5}x_j}{\theta}\right) + \left(-210\theta^3 - 54\cdot 5^{\frac{3}{2}}x_j\theta^2 - 600x_j^2\theta - 4\cdot 5^{\frac{5}{2}}x_j^3\right)\gamma\right)x_i^2 + \Big(\Big(-3\cdot 5^{\frac{5}{2}}\theta^4 + (270x_j - 570)\theta^3 + \left(-12\cdot 5^{\frac{3}{2}}x_j^2 + 72\cdot 5^{\frac{3}{2}}x_j - 12\cdot 5^{\frac{5}{2}}\right)\theta^2 + \left(-300x_j^2 + 600x_j - 300\right)\theta\Big)\exp\left(\frac{2\sqrt{5}x_j}{\theta}\right) + \left(42\sqrt{5}\theta^4 + 420x_j\theta^3 + 54\cdot 5^{\frac{3}{2}}x_j^2\theta^2 + 400x_j^3\theta + 2\cdot 5^{\frac{5}{2}}x_j^4\right)\gamma\Big)x_i + \Big(54\theta^5 + \left(108\sqrt{5} - 33\sqrt{5}x_j\right)\theta^4 + \left(30x_j^2 - 330x_j + 450\right)\theta^3 + \left(12\cdot 5^{\frac{3}{2}}x_j^2 - 48\cdot 5^{\frac{3}{2}}x_j + 36\cdot 5^{\frac{3}{2}}\right)\theta^2 + \left(150x_j^2 - 300x_j + 150\right)\theta\Big)\exp\left(\frac{2\sqrt{5}x_j}{\theta}\right) + \left(-42\sqrt{5}x_j\theta^4 - 210x_j^2\theta^3 - 18\cdot 5^{\frac{3}{2}}x_j^3\theta^2 - 100x_j^4\theta - 2\cdot 5^{\frac{3}{2}}x_j^5\right)\gamma\Big]\exp\left(\frac{2\sqrt{5}x_i}{\theta}\right) + \left(-150\theta^3 - 24\cdot 5^{\frac{3}{2}}x_j\theta^2 - 150x_j^2\theta\right)\gamma x_i^2 + \left(-3\cdot 5^{\frac{5}{2}}\theta^4 - 270x_j\theta^3 - 12\cdot 5^{\frac{3}{2}}x_j^2\theta^2\right)\gamma x_i + \left(-54\theta^5 - 33\sqrt{5}x_j\theta^4 - 30x_j^2\theta^3\right)\gamma$$

with $t_7 = -108\theta^5 \exp\left(\frac{\sqrt{5}(x_j+x_i+2)}{\theta}\right)$.

$$\frac{\partial w_{5/2}(x_i,x_j)}{\partial x_j}t_7 = \Bigg(\Big(150\theta^3 + 24\cdot 5^{\frac{3}{2}}\left(1-x_i\right)\theta^2 + \left(150x_i^2 - 300x_i + 150\right)\theta\Big)\exp\left(\frac{2\sqrt{5}x_i}{\theta}\right)x_j^2 +$$

$$\Big(-3\cdot 5^{\frac{5}{2}}\theta^4 + \left(270x_i - 570\right)\theta^3 - 12\cdot 5^{\frac{3}{2}}\left(x_i^2 - 6x_i + 1\right)\theta^2 - 300\left(x_i^2 - 2x_i + 1\right)\theta\Big)\exp\left(\frac{2\sqrt{5}x_i}{\theta}\right)x_j$$

$$+\Big(54\theta^5 + \Big(108\sqrt{5} - 33\sqrt{5}x_i\Big)\theta^4 + \left(30x_i^2 - 330x_i + 450\right)\theta^3 + \Big(12\cdot 5^{\frac{3}{2}}x_i^2 - 48\cdot 5^{\frac{3}{2}}x_i + 36\cdot 5^{\frac{3}{2}}\Big)\theta^2 +$$

$$\left(150x_i^2 - 300x_i + 150\right)\theta\Big)\exp\left(\frac{2\sqrt{5}x_i}{\theta}\right)\Bigg)\exp\left(\frac{2\sqrt{5}x_j}{\theta}\right) + 2\cdot 5^{\frac{3}{2}}\exp\left(2\sqrt{5}x_i/\theta + 2\sqrt{5}/\theta\right)x_j^5 +$$

$$\Big(100\theta - 2\cdot 5^{\frac{5}{2}}x_i\Big)\exp\left(\frac{2\sqrt{5}(x_i+1)}{\theta}\right)x_j^4 + \Big(18\cdot 5^{\frac{3}{2}}\theta^2 - 400x_i\theta + 4\cdot 5^{\frac{5}{2}}x_i^2\Big)\exp\left(\frac{2\sqrt{5}\left(x_i+1\right)}{\theta}\right)x_j^3$$

$$+\Bigg(\Big(210\theta^3 - 54\cdot 5^{\frac{3}{2}}x_i\theta^2 + 600x_i^2\theta - 4\cdot 5^{\frac{5}{2}}x_i^3\Big)\exp\left(\frac{2\sqrt{5}\left(x_i+1\right)}{\theta}\right) +$$

$$\Big(-150\theta^3 - 24\cdot 5^{\frac{3}{2}}x_i\theta^2 - 150x_i^2\theta\Big)\gamma\Bigg)x_j^2 + \Bigg(\Big(42\sqrt{5}\theta^4 - 420x_i\theta^3 + 54\cdot 5^{\frac{3}{2}}x_i^2\theta^2 - 400x_i^3\theta +$$

$$2\cdot 5^{\frac{5}{2}}x_i^4\Big)\exp\left(\frac{2\sqrt{5}\left(x_i+1\right)}{\theta}\right) + \Big(-3\cdot 5^{\frac{5}{2}}\theta^4 - 270x_i\theta^3 - 12\cdot 5^{\frac{3}{2}}x_i^2\theta^2\Big)\gamma\Bigg)x_j +$$

$$\Big(-42\sqrt{5}x_i\theta^4 + 210x_i^2\theta^3 - 18\cdot 5^{\frac{3}{2}}x_i^3\theta^2 + 100x_i^4\theta - 2\cdot 5^{\frac{3}{2}}x_i^5\Big)\exp\left(\frac{2\sqrt{5}\left(x_i+1\right)}{\theta}\right) +$$

$$\Big(-54\theta^5 - 33\sqrt{5}x_i\theta^4 - 30x_i^2\theta^3\Big)\gamma$$

Finally, we provide expressions for c_2 from (11):

$$c_{2,1/2} = \exp\left(-\frac{2x_i}{\theta}\right);$$

$$c_{2,3/2}\cdot t_8 = \Big(3x_i^2 - 2\left(\sqrt{3}\theta + 3\right)x_i + \theta^2 + 2\sqrt{3}\theta + 3\Big)\exp\left(\frac{4\sqrt{3}x_i}{\theta}\right) - 3\beta x_i^2$$

$$- 2\sqrt{3}\theta\beta x_i - \theta^2\beta;$$

$$c_{2,5/2}\cdot t_9 = \exp\left(\frac{4\sqrt{5}x_i}{\theta}\right)\cdot\Big[25\theta^4 - 2\left(3\cdot 5^{\frac{3}{2}}\theta + 50\right)x_i^3 + 3\left(\theta\left(25\theta + 6\cdot 5^{\frac{3}{2}}\right) + 50\right)x_i^2 -$$

$$2\left(3\theta\left(\theta\left(3\sqrt{5}\theta + 25\right) + 3\cdot 5^{\frac{3}{2}}\right) + 50\right)x_i + 9\theta^4 + 18\sqrt{5}\theta^3 + 75\theta^2 + 6\cdot 5^{\frac{3}{2}}\theta + 25\Big] -$$

$$25\gamma x_i^4 - 6\cdot 5^{\frac{3}{2}}\theta\gamma x_i^3 - 75\theta^2\gamma x_i^2 - 18\sqrt{5}\theta^3\gamma x_i - 9\theta^4\gamma$$

with $t_8 = -\theta^2\exp\left(\frac{2\sqrt{3}(x_i+1)}{\theta}\right)$, $t_9 = -9\theta^4\exp\left(\frac{2\sqrt{5}(x_i+1)}{\theta}\right)$.