Sequential Design with Mutual Information for Computer Experiments (MICE): Emulation of a Tsunami Model*

Joakim Beck†  Serge Guillas‡

Abstract

Computer experiments are often used as substitutes for real-life experiments that are too costly, or difficult to perform. However, computer experiments themselves can be too time consuming to run for a large number of parameter configurations, as required for uncertainty quantification and optimization. We present a sequential design algorithm based on mutual information, named MICE (Mutual Information for Computer Experiments). The Gaussian Process emulator is used, which allows the algorithm to make fast estimation of the prediction error over the parameter space. Furthermore, our detailed analysis of the computational complexity includes all the costs associated with sequential design. The superior computational efficiency of the MICE algorithm is demonstrated for synthetic examples and a tsunami model, with overall gains of up to 20% in that case over other methods.

1 Introduction

Computer experiments have been widely replacing physical experiments as they are typically less resource- and time-consuming. A computer experiment is designed to learn about one or multiple outputs of interest of a computer model (also known as a simulator), using a number of runs of the simulator at different input parameter values, over a parameter space called the design space. However, advanced simulators themselves can be quite demanding in terms of computing resources and run time. Hence, in order to facilitate uncertainty quantification (UQ) for simulators, we want to run computer experiments for well chosen values of the various parameters. A natural approach is to evaluate the output of the simulator on a space-filling design, and then build a surrogate model from the collected input-output data. A surrogate model, or emulator, is a fast approximation of the simulator and is used to predict the model output for untried design points. The design should be chosen judiciously in order to obtain a good approximation of the simulator by the surrogate. The most common space-filling designs include Latin hypercube designs (LHD), maximin(Mm)- and minimax(mM)-distance designs, multi-layer designs, and orthogonal arrays [2, 23]. We denote by MmLHD a Mm-distance design within the class of LHDs, and mM LHD a mM-distance design within the class of LHDs.

Sacks et al. [25] fit a Gaussian Process (GP) models to data from computer experiments given a “small” budget. GP models are commonly used surrogate models for emulating expensive-to-evaluate simulators in many fields of simulation [26], and for global optimization [18]. In recent years, Gaussian process approaches to UQ for simulators have become increasingly popular [4, 27].

*This work was supported by the NERC grant NE/J017434/1 "Probability, Uncertainty and Risk in the Natural Environment"

†Department of Statistical Science, University College London, London, UK. (joakim.beck@ucl.ac.uk)
‡Department of Statistical Science, University College London, London, UK. (s.guillas@ucl.ac.uk)
Applications of GP emulation include CFD simulation of a rocket booster \cite{14}, and climate simulation \cite{7}. In this work, we are interested in employing such an approach to obtain the maximum information about the simulator’s output over the design space, here only for deterministic simulators. At untried design points we are uncertain about the discrepancy between simulator and GP model. Fortunately, GPs possess a closed-form expression for the expected prediction error at any untried design point, which quantifies the uncertainty in prediction. A widely used criterion to determine the most informative design is the entropy criterion \cite{29}. Another popular design criterion is the integrated mean square error \textit{(IMSE)} \cite{25}, which minimizes this expected prediction error over the design space. IMSE designs generally lead to more accurate predictions than LHDs. A more recent design criterion is the A-criterion \cite{5} \textit{(based on the alias matrix)}, which shares similar design features to IMSE, and is less computationally demanding. The two most popular sequential design strategies driven by GPs are ALM and ALC \cite{14}. ALC generally performs better than ALM for prediction, at the price of a higher computational complexity. An alternative that resembles ALC can be found in \cite{7}, which exhibits a slightly lower computational complexity.

In this work we propose a new sequential algorithm, named MICE (Mutual Information for Computer Experiments). MICE extends the sensor placement algorithm proposed by Krause et. al. \cite{20}, which uses a design criterion based on mutual information. Our analysis of the MICE algorithm includes a comparison with traditional sequential and non-sequential algorithms. Critically, we examine the computational cost of the algorithms, as some can be time consuming. In addition to the testing of the algorithms on some standard test functions, in lieu of computer models, we present an application to a tsunami model that solves nonlinear shallow water equations. Our case study is for a landslide-generated tsunami, and the quantity of interest is the sea surface elevation at the opposing shoreline of a landslide event.

The paper is organized as follows. Section 2 reviews Gaussian process modeling for prediction. In Section 3 we provide a background on two traditional sequential design algorithms for the design and analysis of deterministic computer experiments, namely ALM and ALC. Sections 4.1–4.2 present the MI algorithm (for which our algorithm is based upon), and in Section 4.3 we highlight some practical issues with MI. Then, in Section 4.4 we present the MICE algorithm. Section 5 discusses the computational complexity of sequential algorithms. A numerical comparison of MICE with other sequential and non-sequential algorithms for a few standard test functions is presented in Section 6, and for a tsunami simulator in Section 7. Finally, in Section 8, we present our conclusions. Proofs of theorems are in the appendix.

## 2 Gaussian process modeling for prediction

Let us treat the simulator \( \mathcal{X} \rightarrow \mathbb{R} \), \( x \in \mathcal{X} \subseteq \mathbb{R}^p \) as an uncertain function, \( Y(x) \), except for the points \( x \) where the output, \( y(x) \), has already been observed. Then, following \cite{25}, \( Y(x) \) is modeled as a Gaussian random field, \( Y(x)|\beta, \sigma^2, \xi \sim GP (h(x)^T \beta, \Sigma(\cdot, \cdot)) \), with \( E(Y(x)^2) < \infty \). GPs are characterized by a mean function \( h(x)^T \beta \), where \( \beta \in \mathbb{R}^q \), and a covariance function \( \Sigma(x, x'; \sigma^2) = \sigma^2 K(x, x') \) for \( x, x' \in \mathcal{X} \). \( K(x, x') \) is the correlation function, and \( \sigma^2(>0) \) is the process variance. Note that the process variance is independent of \( x \). To simplify notation, let \( \theta = (\sigma^2, \xi)^T \) be the parameters of the covariance structure of the GP. The mean function \( h(x)^T \beta \) is traditionally written as a linear combination of \( q \) regressors, \( h(x) = (h_1(x), h_2(x), \ldots, h_q(x))^T : \mathcal{X} \rightarrow \mathbb{R}^q \). In general, a constant mean is considered an adequate choice, but, a more complex mean function may be used to capture certain key trends. The correlation is typically expressed as a product of one-dimensional correlation functions. The most widely used is the squared-exponential: \( K(x, x'; \xi) = \prod_{i=1}^p \exp \left(- (x_i - x_i')^2/(2\ell_i^2) \right) \), where \( \xi = (\ell_1, \ell_2, \ldots, \ell_p)^T \), with \( \ell_i \) is defining the
correlation length parameter for the $i$-th input dimension. Since this correlation function only depends on $x$ and $x'$ through the distance between them, it is stationary. When using the squared-exponential, one implicitly assumes that the process $Y(x)$ has mean square derivatives of all orders, which is a quite strong assumption, especially when modeling complex physical processes. An alternative is to instead consider a more general set of correlation functions, the Matérn family [10]:

$$K(x, x'; \xi) = \prod_{i=1}^{p} \frac{1}{2^{\nu-1} \Gamma_{\nu}} \left(\frac{2\nu^{\frac{1}{2}}|x_i - x'_i|}{\ell_i}\right)^{\nu} J_{\nu} \left(\frac{2\nu^{\frac{1}{2}}|x_i - x'_i|}{\ell_i}\right),$$

where $\xi = (\ell_1, \ldots, \ell_p, \nu)^T$, $\Gamma_{\nu}$ is the gamma function for $\nu$, and $J_{\nu}$ a modified Bessel function of order $\nu > 0$. The parameter $\nu$ regulates the smoothness of the process; for instance, Gaussian random fields with Matérn covariance are $[\nu-1]$ times mean-square differentiable [31]. The Matérn correlation function has become increasingly popular because of its ability to model data of different degrees of smoothness. The squared-exponential correlation is a special case of a Matérn correlation function.

For prediction the mean of the posterior distribution of $Y(x)$, conditional on the known data $(y_j, x_j)_{j=1}^{m}$, can be used as a predictor for $y(x)$ at any untried point. The known design $D$ is $(X_D, y_D)$, where $X_D = (x_j)_{j=1}^{m}$, $y_D = (y_1, y_2, \ldots, y_m)^T$, and $y_i = y(x_i)$. Given that the underlying function $y(x)$ is a realization of $Y(x)\mid \beta, \theta$, the joint distribution of $Y(x_1), Y(x_2), \ldots, Y(x_m)\mid \beta, \theta$ is known to follow a multivariate normal distribution $N_m(H\beta, \Sigma_{\theta})$ where $H = (h(x_j))_{j=1}^{m}$, and $\Sigma_{\theta}$ is the $m \times m$ variance-covariance matrix whose $(j,k)$-th entry is given by $\Sigma_{x_j, x_k; \theta} = \sigma^2 K(x_j, x_k; \xi)$ for $x_j, x_k \in X_D$. The correlation matrix is $K_{\xi} = \sigma^{-2} \Sigma_{\theta}$, and must be positive semidefinite. Next, $y_D = [y_1, y_2, \ldots, y_m]^T$ is assumed to be a realization of $m$-variate normal distribution $[Y(x_1), Y(x_2), \ldots, Y(x_m)]\mid \beta, \sigma^2, \xi$. An exact marginalization of $\beta$ and $\sigma^2$ is possible using the non-informative prior $p(\beta, \sigma^2) \propto \sigma^{-2}$. Then, as shown in [10], the posterior distribution of $Y(x)\mid y_D, \xi$ follows a shifted Student’s t-distribution on $m - q$ degrees of freedom with mean

$$\hat{y}(x; \xi) = h^T(x)\hat{\beta} + k^T(x)K_{\xi}^{-1}(y_D - H\hat{\beta}),$$

and variance-covariance

$$\hat{V}(x, x'; \xi) = \sigma^2 \left(1 - k^T(x)K_{\xi}^{-1}k(x') + \gamma^T(x; \xi) \left(H^T_K \xi^{-1}H\right)^{-1}\gamma(x'; \xi)\right),$$

where the $m \times 1$ vector $k(x; \xi)$ has entry $j$ given by $K(x, x_j; \xi)$ for $x_j \in X_D$, and $\gamma(x; \xi) = H^T K_{\xi}^{-1}k(x) - h(x)$, $\hat{\beta} = (H^T K_{\xi}^{-1}H)^{-1}H^T K_{\xi}^{-1}y_D$, and $\hat{\sigma}^2 = \frac{1}{m-q}(y_D - H\hat{\beta})^T K_{\xi}^{-1}(y_D - H\hat{\beta})$. For practical computation, the inputs $X_D$ are scaled to lie in $[0,1]^p$, and the outputs $y_D$ are scaled to have zero mean and unit variance. The predictive (posterior) mean at $x$, given by Equation (2), is the best linear unbiased predictor (BLUP) [25], which interpolates the data $D$. The regularity of the correlation function $K(x, x')$ for $x = x'$ determines the regularity of the predictor $\hat{y}(x)$ [31], which means that the regularity of $y(x)$ should be reflected in the choice of $K(x, x')$. The predictive (posterior) variance at $x$, given by Equation (3) for $x' = x$, is an estimate of the MSPE ($E (y(x) - \hat{y}(x; \xi))^2$). Let us denote the predictive variance $\hat{V}(x, x; \xi)$ by $\hat{s}^2(x; \xi)$. The predictive variance $\hat{s}^2(x; \xi)$ can be used to produce confidence bands for the predictive mean $\hat{y}(x; \xi)$.

2.1 Estimation of uncertain parameters in the correlation function

So far, we have assumed that the correlation function $K(\cdot, \cdot; \xi)$ is known, which rarely is the case, particularly in the context of computer experiments. We briefly discusses approaches that deal with
the situation where the form of $K(\cdot, \cdot; \xi)$ is assumed to be known, but where $\xi$ is uncertain. Priors can be used to express one’s uncertainty about the correlation parameters, followed by sampling of the posterior of those parameters by, e.g., Markov chain Monte Carlo (MCMC). This is a fully Bayesian approach that can be very computationally expensive as the convergence to stationarity can be challenging. A somewhat less computationally demanding alternative is to find estimates of the parameters that best fit the GP to the known data, $(X_D, y_D)$, by using maximum likelihood estimation (MLE). MLE provides point estimates for $\xi$, denoted by $\hat{\xi}$, and those estimates are then plugged into the GP as if they were the true values. The vector $\hat{\xi}$ is the solution to

$$
\hat{\xi} = \arg\max_{\xi \in \Xi} \ln(L(\xi | y_D)) = \arg\min_{\xi \in \Xi} \ln(\det(\mathbf{K}_\xi)),
$$

where $\ln(L(\xi | y_D))$ is the marginal log-likelihood, and $\Xi$ is the search domain of interest. Given that the underlying distribution is Gaussian, the following relationship holds:

$$
L(\xi | y_D) \propto |\mathbf{K}_\xi|^{-\frac{1}{2}}|H^T\mathbf{K}_\xi^{-1}H|^{-\frac{1}{2}}(\sigma^2)^{-\frac{(m-\ell)}{2}}.
$$

The estimated predictor, $\hat{y}(x; \xi)$, is known as the estimated BLUP (EBLUP) [32]. A drawback with this approach is that the variance estimator $\hat{s}^2(x; \xi)$ tends to underestimate the MSPE, especially if the correlation is weak. This is because the estimator has been derived under the assumption that the correlation is fully known. In the following, we drop $\xi$’s in our notation for brevity’s sake, whenever there is no ambiguity.

### 2.2 On the numerical stability

In this section, we discuss a few practical implementations we have adopted:

i) Nugget parameter for regularization: To fit the GP model with MLE we require the inverse and the determinant of the $m \times m$ correlation matrix $\mathbf{K}_\xi$ for a large number of $\xi$’s. The Cholesky decomposition ($\mathbf{K} = \mathbf{L}\mathbf{L}^T$) is typically used for matrix inversion, which costs $O(m^3)$, and is hence computationally intensive for large $m$. $\mathbf{L}$ is here a lower triangular matrix. The Cholesky decomposition is known to suffer from numerical instabilities whenever the matrix $\mathbf{K}$ is near singular (due to ill-conditioning). The closer a matrix is to being singular, the larger its condition number is. Whenever the correlation matrix is ill-conditioned a non-zero nugget parameter can be used for regularization: $\mathbf{K}_\xi$ is replaced in (2) and (3) by $\mathbf{K}_{\xi, \tau^2} = \mathbf{K}_\xi + \tau^2 \mathbf{I}$, where $\mathbf{I}$ is the $m \times m$ identity matrix. Typically $\tau^2$ is chosen to be very small. $\mathbf{K}_{\xi, \tau^2}$ becomes better conditioned as the size of the nugget increases. In practice, to prevent the correlation matrix from becoming ill-conditioned, the nugget should be chosen so that the condition number of $\mathbf{K}_{\xi, \tau^2}$ does not exceed $\Delta = 1/(10\varepsilon)$ [22], where $\varepsilon$ is the machine epsilon for double precision floating point numbers. However, by introducing a non-zero nugget, we are sacrificing the interpolatory property of the predictive mean $\hat{y}(x)$, see Theorem 1. The nugget may also affect the integrated likelihood profile substantially [1]. Nevertheless, the use of a nugget to improve stability can be very beneficial [15]. Below is our result on the effect of the nugget on the predictive variance at any point in the design $D$.

Obviously, if $\tau^2 = 0$, $\hat{s}^2(x_i) = 0$ for $x_i \in X_D$.

**Theorem 1** For a GP model with a constant mean, and with design $D$, the predictive variance given by (3) can be written as

$$
\hat{s}^2_{\tau^2}(x_i) = \sigma^2 \left( \tau^2 - \tau^4 \mathbf{e}_i^T(\mathbf{K} + \tau^2 \mathbf{I})^{-1}\mathbf{e}_i + \tau^4 \frac{(\mathbf{e}_i^T(\mathbf{K} + \tau^2 \mathbf{I})^{-1}\mathbf{I})^2}{\mathbf{I}^T(\mathbf{K} + \tau^2 \mathbf{I})^{-1} \mathbf{I}} \right),
$$

where $\mathbf{e}_i$ is the $i$-th unit vector.
for any $x_i \in X_D$, where $\tau^2$ is a non-negative nugget parameter, and $e_i$ is the $i$-th unit vector.

According to Theorem 1 whenever $\tau^2 > 0$ is added to the correlation matrix diagonal, the predictive variance, at a known design point, is $\sigma^2 \tau^2$ with some variability proportional to $\sigma^2 \tau^4$. The magnitude of the last term in the round brackets tends to be much smaller than the other two; hence, the predictive variance at a known design point is typically smaller than $\sigma^2 \tau^2$. Moreover, as $\tau^2$ increases, the second term approaches $\tau^2$, and the last term approaches $\tau^2/M_D$, where $M_D$ is the number of points in the design $D$. This follows from the fact that with the increase of $\tau^2$, the inverse matrix reduces to $(K + \tau^2I)^{-1} \approx \tau^{-2}I$. Hence, for a very large $\tau^2$, following some simple calculations, we observe that $s^2_{\tau^2}(x_i) \approx \sigma^2 \tau^2/M_D$ for $x_i \in X_D$.

ii) QR decomposition: To improve numerical stability, we follow [21] with the Cholesky decomposition $LL^T$. Then, the following transformations are made: $H = L^{-1}H$, $\hat{k} = L^{-1}k$, and $\hat{y}_D = L^{-1}y_D$. Also, as a precaution if $D$ is large, the QR decomposition can be applied to decompose $H$ into $QR^T$, where $Q \in \mathbb{R}^{m \times p}$ has orthonormal columns, and $R \in \mathbb{R}^{p \times p}$ is an upper triangular matrix. $\beta$ can in this way be directly solved from the least square problem $R^T \beta = Q^T \hat{y}_D$. Furthermore, the direct computation of the predictive variance-covariance, using Equation (3), could, due to unstable computations, produce negative values. Thus we improve the stability of Equation (3) as follows:

$$
\hat{V}(x, x')/\hat{\sigma}^2 = 1 - k^T(x)K^{-1}k(x') + \gamma^T(x)(H^T K^{-1} H)^{-1} \gamma(x') \\
= 1 - \hat{k}^T(x)\hat{k}(x') + (\hat{H}^T \hat{k}(x) - h(x))^T (\hat{H}^T \hat{H})^{-1} (\hat{H}^T \hat{k}(x') - h(x')) \\
$$

$$
= 1 - \hat{k}^T(x)\hat{k}(x') + (\hat{H}^T \hat{k}(x) - h(x))^T (QR^T QR^T)^{-1} (\hat{H}^T \hat{k}(x') - h(x')) \\
= 1 - \hat{k}^T(x)\hat{k}(x') + (\hat{H}^T \hat{k}(x) - h(x))^T (RR^T)^{-1} (\hat{H}^T \hat{k}(x') - h(x')) \\
$$

iii) Arithmetic underflow: $\ln(\det(K))$ in (4) may suffer from floating-point underflow problems, that is, its value becomes so small that it cannot be represented in double floating-point precision. To mitigate the risk of this happening, we perform the following calculation [21]: $\ln(\det(K)) = m \ln(\det(K))^{1/m} = m \ln((\prod_j L_{jj})^{2/m})$, which uses that $\det(K) = \det(L) \det(L^T) = (\prod_j L_{jj})^2$ (recall $L$ is a triangular matrix).

3 Sequential design of computer experiments

A well-known (non-sequential) design strategy for prediction is maximum entropy sampling [29], where the goal is to identify the most informative subset of size $M$, determined using the entropy [9], from a finite candidate set $X_{cand} \subset X$:

$$
X^*_M = \arg \max_{X_M \subset X_{cand}} \mathcal{H}(\bar{Y}_M), \hspace{1cm} (7)
$$

where $\bar{Y}_M := \bar{Y}[X_M] = [Y(x_1), Y(x_2), \ldots, Y(x_M)]$ with joint pdf $p_{\bar{Y}_M}(y)$, and $\mathcal{H}(\bar{Y}_M)$ is the entropy,

$$
\mathcal{H}(\bar{Y}_M) = E(-\log_b(\bar{Y}_M)) = -\int \cdots \int \log_b(p_{\bar{Y}_M}(y))p_{\bar{Y}_M}(y) \, dy. \hspace{1cm} (8)
$$
Here $- \log_2(\bar{Y})$ is a measure of the information content of $\bar{Y}$. If $b = 2$, $[\mathcal{H}(\bar{Y})]$ is the number of bits needed to represent $\bar{Y}$. When the underlying distribution is a multivariate normal with correlation matrix $K$, the entropy (in bits) is

$$\mathcal{H}(\bar{Y}_M) = \frac{1}{2} \log_2(2\pi e)^m \det(K).$$

(9)

Ko et al. [19] showed that this optimization problem is NP-hard. In other words, this optimization problem quickly becomes computationally intractable when $M$ grows large. Instead, a sequential approach can be adopted. Sequential design algorithms start with an empty or sparse design, and then add points into the existing design in a sequential manner until the design is complete. When a certain level of accuracy is desired within a limited time budget, a sequential algorithm offers the opportunity to stop the algorithm whenever a target accuracy level has been reached, which can result in significant computational savings. The two most widely used GP-based sequential design strategies are active learning-MacKay (ALM) and active learning-Cohn (ALC), see, e.g., [28].

### 3.1 ALM algorithm

ALM (Active learning-MacKay) is a greedy algorithm for maximum entropy sampling that selects $x \in X_{\text{cand}}$ that yields the largest improvement in entropy:

$$x^* = \arg \max_{x \in X_{\text{cand}}} \mathcal{H}((\bar{Y} \cup Y(x))|D) = \arg \max_{x \in X_{\text{cand}}} \mathcal{H}(Y(x)|D) = \arg \max_{x \in X_{\text{cand}}} \hat{s}_D^2(x),$$

(10)

at each step. The last identity follows from Equation (9). We use the subscript $D$ to denote a quantity that depends on design $D$. At the end of each greedy step the new point, $x^*$, is included in the existing design $D$, $(X_D, y_D)$. This procedure is repeated until the stop criterion is satisfied. A drawback is that ALM places too many points on the boundary of the design space, especially in the early stages of the selection process. This selection behavior is rather wasteful as boundary points generally are less informative than points in the interior [23]. Here, by boundary we mean the convex hull of the points in $X_{\text{cand}}$. The ratio of boundary points grows rapidly with the dimension size, $p$. Given a regular grid with $n^p$ points, $n^p - (n - 2)^p$ points are on the boundary. This means that the ratio of the number of points on the boundary to the total is $(1 - (1 - \frac{2}{n})^p)$. As an example, if $p = 4$ and $n = 10$ the ratio is approximately 0.59, and if $p = 6$ and $n = 10$, the ratio is approximately 0.74.

### 3.2 ALC algorithm

ALC (Active learning-Cohn) is a sequential algorithm that selects $x \in X_{\text{cand}}$ that yields the largest expected reduction in predictive variance over $\mathcal{X}$:

$$x^* = \arg \max_{x \in X_{\text{cand}}} \int_{\mathcal{X}} \hat{s}_D^2(x') - \hat{s}_{D \cup \{x\}}^2(x') \, dx',$$

(11)

at each step. ALC solves (11) iteratively, and at the end of each iteration the point $x^*$ is added to the existing design $D$. In practice, the ALC criterion is often simply defined as the average reduction in predictive variance over a set of trial points spread out in $\mathcal{X}$:

$$x^* = \arg \max_{x \in X_{\text{cand}}} \frac{1}{M_r} \sum_{k=1}^{M_r} \frac{\hat{s}_D^2(x_k) - \hat{s}_{D \cup \{x\}}^2(x_k)}{M_r},$$

(12)
where \( M_f \) is the number of trial points. The Cholesky decomposition of \( K_{D \cup \{x\}} \) is performed for each \( x \in X_{\text{cand}} \), which leads to a time complexity of \( \mathcal{O}(M_{\text{cand}}M_fm^3) \), which is computationally infeasible even for moderately large \( m \). Fortunately, by first inverting \( K_D \) using the Cholesky decomposition, which is in \( \mathcal{O}(m^3) \), it is possible to compute \( K_{D \cup \{x\}}^{-1} \) for any \( x \), in \( \mathcal{O}(m^3) \), by exploiting the fact that \( K_{D \cup \{x\}}^{-1} \) can be expressed in terms of \( K_D^{-1} \) (and \( k_D(x) \)):

\[
K_{D \cup \{x\}}^{-1} = \left( K_D^{-1} + \frac{1}{k} k_D^{-1} k_D(x) k_D^T(x) K_D^{-1} - \frac{1}{k} k_D^{-1} k_D(x) \right) \tag{13}
\]

where \( k = 1 - k_D^T(x)^T K_D^{-1} k_D(x) \). Furthermore, as shown in [14], by using Equation (13), it is possible to recast (12) into:

\[
x^* = \arg \max_{x \in X_{\text{cand}}} \frac{1}{M_f} \sum_{k=1}^{M_f} \tilde{V}_D^2(x, x_k) / s_D^2(x), \tag{14}
\]

which reduces the time complexity for a single ALC step from \( \mathcal{O}(M_{\text{cand}}M_fm^3) \) to \( \mathcal{O}(m^3 + M_{\text{cand}}M_fm^2) \). ALC designs generally perform better than ALM designs in terms of MSPE for a given design size [14] [28]. Moreover, ALC does not exhibit the tendency to select points on the boundary of \( \mathcal{X} \), contrary to ALM. On the other hand, ALM is occasionally preferred over ALC due to its relatively low computational complexity [4]; more on this in Section 5.

### 4 Mutual information for the design and analysis of computer experiments

Mutual information, which, like entropy, is a standard measure in classical information theory, has been shown to be useful for sensor network design [6] [20], experimental design [17], and optimization [8].

In this work, we develop a sequential design algorithm for the design and analysis of computer experiments, called MICE (Mutual Information for Computer Experiments). MICE is based on the greedy mutual information (MI) algorithm proposed by Krause et al. [20], for optimal sensor placement. We begin this section with a brief background on mutual information. Let us consider two random vectors \( \bar{Y} \) and \( \bar{Y}' \) with marginal pdfs \( p_{\bar{Y}}(y) \) and \( p_{\bar{Y}'}(y') \), and joint pdf \( p_{\bar{Y},\bar{Y}'}(y, y') \). The mutual information between them, denoted by \( I(\bar{Y}; \bar{Y}') \), is equivalent to the Kullback-Leibler divergence \( D_{KL}(\cdot \| \cdot) \) between \( p_{\bar{Y},\bar{Y}'} \) and \( p_{\bar{Y}} p_{\bar{Y}'} \): \( \bar{Y} \)

\[
I(\bar{Y}; \bar{Y}') = D_{KL}(p_{\bar{Y},\bar{Y}'}(\cdot, \cdot) \| p_{\bar{Y}}(\cdot) p_{\bar{Y}'}(\cdot)) = \int \int \ldots \int \log \left( \frac{p_{\bar{Y},\bar{Y}'}(y, y')}{p_{\bar{Y}}(y) p_{\bar{Y}'}(y')} \right) p_{\bar{Y},\bar{Y}'}(y, y') \, dy \, dy', \tag{15}
\]

with \( \log(0)0 = 0 \). The mutual information between \( \bar{Y} \) and \( \bar{Y}' \) can be linked to entropy [7]:

\[
I(\bar{Y}; \bar{Y}') = \mathcal{H}(\bar{Y}) - \mathcal{H}(\bar{Y} | \bar{Y}'). \tag{16}
\]

Caselton and Zidek [6] proposed a mutual information-based criterion for the design of spatial sampling networks, where the goal is to select from a candidate set \( X_{\text{cand}} \subseteq X_F \) the most informative subset of size \( M \) determined by the mutual information between \( \bar{Y}[X_M] \) and \( \bar{Y}[X_{M^c}] \):

\[
X_M^* = \arg \max_{X_M \subseteq X_{\text{cand}}} \mathcal{H}(\bar{Y}[X_M]) - \mathcal{H}(\bar{Y}[X_{M^c}] | \bar{Y}[X_M]), \tag{17}
\]

where \( X_{M^c} = X_F \setminus X_M \) is the complement of \( X_M \) in \( X_F \). Here \( X_F \) is a finite subset of \( \mathbb{R}^2 \). This optimization problem is NP-hard [20].
4.1 A greedy mutual information (MI) criterion for prediction

Krause et al. [20] proposed a greedy algorithm for optimization problem (17), that is, maximize $I(\hat{Y}[X_D \cup \{x\}]; \hat{Y}[X_{D^c}\setminus\{x\}]) - I(\hat{Y}[X_D]; \hat{Y}[X_{D^c}])$ at each step, and showed that this criterion can be recast into a more simple form, i.e.,

$$\mathcal{H}(Y(x)|\hat{Y}[X_D]) - \mathcal{H}(Y(x)|\hat{Y}[X_{D^c}\setminus\{x\}]),$$

(18)

for any $x \in X_{\text{cand}}$, where $D^c = X_F \setminus X_D$, and $X_F$ is a discretization of the region of interest $X \subseteq \mathbb{R}^2$. The design $D$ is the current design. Krause et al. [20] proved that this is a constant-factor approximation algorithm for optimization problem (17) under some mild conditions, and showed its efficiency for a sensor placement problem. In this case, $X_{\text{cand}}$ is the set of admissible sensor sites. The approximation is within $(1 - 1/e)$ of the optimum, given that discretization $X_F$ is fine enough in $X$ (i.e., a discretization level larger than $2M$ points, where $M$ is the desired design size), and that the covariance function is fully known. The proof relies on the fact that the set function $I(\hat{Y}[X_D]; \hat{Y}[X_{D^c}])$ is submodular over $X_F$, with $I(\emptyset; \hat{Y}[X_F]) = 0$. Greedy algorithms are known to be quite efficient for submodular set functions. Furthermore, as we assume that $\hat{Y}[X_D]$ and $\hat{Y}[X_{D^c}]$ follow a multivariate normal distribution, we can rewrite the objective function (18) of the optimization problem into a computationally-efficient form, as follows:

$$\mathcal{H}(Y(x)|\hat{Y}[X_D]) - \mathcal{H}(Y(x)|\hat{Y}[X_{D^c}\setminus\{x\}]) = \frac{1}{2} \log \left( 2\pi e s_D^2(x) \right) - \frac{1}{2} \log \left( 2\pi e s_{D^c\setminus\{x\}}^2(x) \right) = \log \left( \frac{s_D^2(x)}{s_{D^c\setminus\{x\}}^2(x)} \right) = s_D^2(x) / s_{D^c\setminus\{x\}}^2(x)$$

(19)

Thus, the goal is to maximize the ratio of the predictive variances using the design $D$, to that using the design $D^c\setminus\{x\}$. The greedy MI algorithm is given below:

**MI algorithm (Krause et al. [20]):**

- **Require:** GP model $(h(\cdot), \Sigma(\cdot, \cdot)), \tau^2$, $X_F$, $X_{\text{cand}} \subseteq X_F$, $X_D \subset X_{\text{cand}}$, desired design size $M$
- **Step 1.** Let $X_{D^c} = X_F \setminus X_D$
- **Step 2.** Solve $x^* = \arg \max_{x \in X_{\text{cand}}} s_D^2(x) / s_{D^c\setminus\{x\}}^2(x)$
- **Step 3.** Let $X_D = X_D \cup \{x^*\}$, and $X_{\text{cand}} = X_{\text{cand}} \setminus \{x^*\}$
- **Step 4.** If design $D$ is of size $M$, then stop; otherwise go to Step 1
- **Output:** $X_D$ of size $M$

Observe that, in Step 2, a GP model is assigned to the design $D$, and for each candidate, $x$, a GP model is assigned to $D^c\setminus\{x\}$. Note that, in our case, we need to discretize the design space $X \subseteq \mathbb{R}^p$, where $p$ may be large. Next, we compare MI to ALM and ALC for a simple example.

4.2 MI example

Let us consider a realization of a stationary Gaussian random field with zero mean on a $21 \times 21$ regular grid over $[0, 1]^2$. The covariance is defined by $\sigma^2 = 1$ and the squared-exponential correlation
function with $\xi = (0.8, 0.5)^T$. The discretization $X_F$ is a regular subgrid of size $11 \times 11$ with equidistant points. We also make all points in $X_F$ available for selection, i.e., $X_{\text{cand}} = X_F$. The remaining 320 design points constitute the hold-off set used to calculate the prediction quality measured by the RMSPE (root mean squared prediction error). Figure 1 displays the prediction performance for ALM, ALC and MI, as averages over ten runs with different initial two-point designs, randomly picked from $X_{\text{cand}}$. MI offers a prediction quality comparable to ALC. ALM is the worst. Indeed, at least 20 boundary points of the total 40 were selected no later than the 25-th iteration, for all the runs with ALM. The MI criterion is indeed a promising design criteria. However, we have identified some practical issues with the MI algorithm, which are presented below.

![Image](image.png)

Figure 1: Left: A realization of the stationary Gaussian random field. Right: Prediction errors.

### 4.3 Practical issues with the MI algorithm

The results for the above example suggest that MI may be able to compete with ALM and ALC. Krause et al. [20] arrived at the same conclusion for an optimal sensor placement application, and showed that MI has a lower run time than ALC (for their application with two design parameters). Unfortunately, the MI algorithm, in its existing form, struggles for the design and analysis of computer experiments.

So far, the MI algorithm has been used for spatial data ($p = 2$) to select the most informative subset $X_D$ of size $M$ from a set $X_{\text{cand}}$ of size $M_{\text{cand}}$, where typically $M_{\text{cand}} \gg k$. Here $X_{\text{cand}} \subseteq X_F$, and $X_D = X_F \setminus X_D$. The design of computer experiments is a more challenging design problem, e.g., the candidate points are typically selected by the algorithm to explore the design space $\mathcal{X}$, and one often considers more than two parameters.

It is clear from observing the MI criterion (19) that $D^c$ plays a major role. For each $x \in X_{\text{cand}}$, a GP is built on $D^c \setminus \{x\}$, which is an operation in $\mathcal{O}(M_{D^c}^3)$, where $M_{D^c}$ is the number of points in $D^c$. The run time of the MI algorithm thus grows exponentially fast with the size of $D^c$. Hence, $D^c$ should preferably not be too large, but note that this also affects the allowed size of the candidate set, since $X_{\text{cand}} \subseteq X_{D^c}$. In addition, the distribution of $D^c$ in $\mathcal{X}$ has a significant effect on the greedy MI criterion. $D^c$ should represent the complement of $D$ in $\mathcal{X}$, and should thus be chosen in such a way that $F = D \cup D^c$ becomes a well-spaced discretization of $\mathcal{X}$. For points close to each other in $D^c$, $s_{D^c \setminus \{x\}}^2(x)$ may become very small, and hence produce a high MI score regardless of their location in $\mathcal{X}$. This issue can be observed in Figure 2. A high MI score is marked in red, a low score is in blue, an intermediate score is in yellow, and the black dots are the points of the
design $D$. The detrimental effect an additional point to $D^c$, in close proximity to points already in $D^c$, may have on the MI scores, shows that the distribution of points in $D^c$ has a strong influence on the MI criterion.

![Figure 2](image1.png)

Figure 2: The score value of the MI criterion over a $7 \times 7$ regular grid with (Left) and w/o (Right) an additional candidate point, $(2/3, 0.15)^T$. Matérn covariance with $\sigma^2 = 1$, $\nu = 5/2$ and $\xi = (0.4, 1)^T$ is employed.

Hence, there is an apparent need to examine, and, if possible, improve the robustness of the MI criterion. For small $p$, we can often ensure that $D^c (D^c \subset F)$ is well-spaced by using a regular grid with equidistant points as the discretization $F$. For large $p$, however, regardless of sampling technique employed to sample $F$, it would be an immense task to ensure that the resulting distribution of points is evenly distributed without clustering. Even with sampling techniques that are space-filling, such as MmLHD, we are unable to guarantee consistency in the results with MI. See Figure 3 for the comparison between the MI scores for two $D^c$ designs with different distribution. The MI scores for the two cases are highly conflicting.

![Figure 3](image2.png)

Figure 3: Given a design $D$, of size 5, the score value of the MI criterion is displayed for two MmLHD candidate sets of size 100. Matérn covariance with $\sigma^2 = 1$, $\nu = 5/2$ and $\xi = (0.4, 1)^T$ is employed.

The MI algorithm also assumes that the covariance is known, but this is rarely the case. MLE can be used to adaptively find point estimates for the uncertain correlation parameters. The choice of correlation parameters is crucial: a poor choice may lead to a poor GP model, see Figure 4. For MI-MLE, the guess $\xi = (1, 1)^T$ is used until the design $D$ is of size 10, after that, MLE is used
at each iteration to update the estimates of the correlation parameters. Two reasonable starting guesses, $\xi = (1, 1)^T$ and $(0.5, 1)^T$, have been included, along with a “good” choice $(0.36, 1.35)^T$ (approximately the same as those determined by MI-MLE). The underlying problem is the Branin function, $y(x) = (x_2 - 5.1x_1^2/(4\pi^2) + (5/\pi)x_1 - 6)^2 + 10(1 - 1/(8\pi))\cos(x_1) + 10$, on a $21 \times 21$ regular grid over $[-5, 10] \times [0, 15] \subset \mathbb{R}^2$.

4.4 MICE algorithm

In this section our new algorithm, called MICE, which overcomes the shortcomings of MI, is introduced. The MICE algorithm is presented below, followed by details on the individual steps of the algorithm.

| MICE algorithm: |
|-----------------|
| **Require:** $y(x), \mathcal{X}$, GP model $(h(\cdot), K(\cdot, \cdot; \xi)), \tau^2$ and $\tau^2_s$, initial data $(X_D, y_D)$, $M_{D_c}, M_{\text{cand}}$, desired design size $M$ |
| **Step 1.** MLE to obtain estimates $\hat{\xi}$ used in $K(\cdot, \cdot; \hat{\xi})$ |
| **Step 2.** Fit GP model to data $(X_D, y_D)$ |
| **Step 3.** Generate $X_{D_c}$ with respect to $X_D$, and then choose $X_{\text{cand}}(\subseteq X_{D_c})$ |
| **Step 4.** Solve $x^* = \arg \max_{x \in X_{\text{cand}}} \frac{s_D^2(x; \hat{\xi}, \tau^2)}{s_{D_c}^2(x; \hat{\xi}, \tau^2_s)}$ |
| **Step 5.** Evaluate $y^* = y(x^*)$ and let $X_D = X_D \cup \{x^*\}$ and $y_D = y_D \cup \{y^*\}$ |
| **Step 6.** If design $X_D$ is of size $M$, then stop; otherwise go to step 1 |
| **Output:** $X_D$ of size $M$ with output data $y_D$ |

In step 1 MLE is used for estimating the uncertain correlation parameters. In MICE a fully Bayesian approach could also be used. Step 2, 5, and 6, are typical steps performed by any GP-based sequential design algorithm. In step 3, we argue in favor of generating $X_{D_c}$ as a LHD. The
LHD can be selected from a set of LHDs. We want to select the LHD that has the large minimum distance to the current design $X_D$, since then $X_{D^c}$ becomes an approximation of the complement of $X_D$ in $X$. Clearly, as the dimension of the problem, $p$, grows, so does the required size of $D^c$. One way to overcome this issue is to generate a new $D^c$ for each MI step to keep $D^c$ at a moderate size, to improve the design exploration. In step 4 our MICE criterion is used. In contrast to the MI criterion, we added a “large” nugget parameter $\tau^2_s$ is added to the diagonal elements of the correlation matrix in the GP assigned to $D^c \setminus \{x\}$. If the size of the nugget is large enough, the predictive variance is flattened as the GP model becomes much more relaxed at the points in $D^c$, see Theorem 1. A moderately large value for $\tau^2_s$ is needed to achieve the desired effect; the default choice is $\tau^2_s = 1$. In this way, one overcomes the issue of uneven distribution of points in $D^c$, which we identified as a major limitation for the MI criterion. The new criterion is defined by $\hat{s}^2_D(x; \hat{\xi}, \tau^2)/\hat{s}^2_{D^c \setminus \{x\}}(x; \hat{\xi}, \tau^2_s)$ (note the two nuggets $\tau^2$ and $\tau^2_s$). Figure 5 and 6 show score values of the MICE with $\tau^2_s = 1$ for a direct comparison with the corresponding figures for MI (shown in Figure 2 and 3 respectively).

Figure 5: The score value of the MICE criterion using $\tau^2_s = 1$ for a $7 \times 7$ regular grid with one additional candidate point $(2/3, 0.15)^T$.

Figure 6: Given a design $D$, of size 5, the score value of the MICE criterion using $\tau^2_s = 1$ is shown for two MmLHD candidate sets of size 100.

Observe that MICE with $\tau^2_s = 1$ performs as well as MI for the regular grid (the simple case), whereas for the case when $D^c$ has an uneven distribution of points (and clustering), MICE shows robustness to the design $D^c$. The choice of $\tau^2_s$ is investigated further in Section 6. Note that the GP
for the design \( D \), in MICE, does not rely on \( \tau^2 \). Although, if necessary, another nugget parameter can be introduced. For that GP we have introduced \( \tau^2 = 10^{-12} \) for the purpose of improving numerical stability. The estimates \( \xi \) for the GP over \( D^c \) are the same as those for \( D \).

The optimality results provided by Krause et al. [20] for the MI algorithm are for known \( \xi \). Therefore, we generalize this result in Theorem 2, which covers the case of different nugget parameters for the GPs over \( D \) and \( D^c \), as well as the case of estimated correlation. This theorem enables us to derive an approximative bound of optimality for MICE as in [20].

**Theorem 2** Let \( Y(\mathbf{x}) \) be a GP model on a compact set \( \mathcal{X} \subset \mathbb{R}^p \) with a continuous correlation function \( K(\mathbf{x}, \mathbf{x}'): \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_0^+ \). Moreover, assume that the correlation matrices for \( D \) and \( D^c \) have included the nuggets \( \tau^2 \) and \( \tau^2_0 \), respectively. We assume that the MICE algorithm use estimates \( \hat{\xi}_i \), for greedy step \( i \), that satisfy, for some constant \( \alpha > 0 \), \(|K(\mathbf{x}, \mathbf{x}'; \xi) - K(\mathbf{x}, \mathbf{x}'; \hat{\xi}_i)| \leq \alpha \).

Then, for any \( \varepsilon > 0 \), and any finite number \( M \), there exists a discretization \( F_\delta \) of mesh width \( \delta > 0 \) such that MICE is guaranteed to select a design \( D_M \) of \( M \) design points, where \( M \leq 2|F_\delta| \), for which

\[
MI(D_M) \geq (1 - 1/e)(OPT - M\varepsilon - 2(\alpha s^{-1}\tau^{-1} - 1)^2 M^4(1 + M^{3/2})^2 - M^{7/2}\tau^2_0^2 - \tau^2_0^2/\tau^2_0),
\]

where \( e \) is the base of the natural logarithm, and \( OPT \) is the value of the mutual information for the optimal design of size \( M \).

Under the optimal conditions this upper bound guarantees a performance within 63% of the optimum. The term \( M\varepsilon \) is unavoidable, but small if the discretization \( F \) is fine enough. The term \( 2(\alpha s^{-1}\tau^{-1} - 1)^2 M^4(1 + M^{3/2})^2 - M^{7/2}\tau^2_0^2 - \tau^2_0^2/\tau^2_0 \) is related to the parameter uncertainty, and the term \( M^3\sqrt{M}\tau^2_0 - \tau^2_0^2/\tau^2_0 \) is the penalty for using \( \tau^2_0 \) instead of \( \tau^2 \) for the GP over \( D^c \). Our extension of the approximative bound of optimality, to MICE, shows the effect of the nugget \( \tau^2_0 \) on the performance, and tells us that our default choice \( \tau^2_0 = 1 \) is not causing the algorithm to diverge too much from mutual information. Observe that the near-optimal performance can be shattered by a poor choice of values for the correlation parameters (with \( \alpha \) large). To better understand the behavior of the MICE criterion for large \( D^c \), we also provide the following theorem:

**Theorem 3** Let \( Y(\mathbf{x}) \) be a second-order stationary Gaussian process with constant mean on a compact subset \( \mathcal{X} \) of \( \mathbb{R}^p \) with a Lipschitz-continuous correlation function. The expected MSPE, \( \hat{s}_{\tau^2}^2(\mathbf{x}) \), is given by \([3]\) with a correlation matrix using a non-negative nugget parameter \( \tau^2 \). Then, for any \( \varepsilon > 0 \), there exists a regular grid \( \mathbf{X}_\delta \subset \mathcal{X} \) with grid spacing \( \delta = 2\varepsilon/(\sqrt{p}K_L) \) so that for any untried point \( \mathbf{x}^* \in \mathcal{X} \) the predictive variance is bounded as

\[
-\sigma^2\tau^4 b_1(\tau^2) - \varepsilon < \hat{s}_{\tau^2}^2(\mathbf{x}^*) - \sigma^2\tau^2 < \sigma^2\tau^4 b_2(\tau^2) + \varepsilon,
\]

where

\[
b_1(\tau^2) = \max \left\{ e_i^T \left( K + \tau^2 I \right)^{-1} e_i : \mathbf{x}_i \in \mathbf{X}_\delta \right\},
\]

and

\[
b_2(\tau^2) = \max \left\{ \left( e_i^T \left( K + \tau^2 I \right)^{-1} I \right)^2 : \mathbf{x}_i \in \mathbf{X}_\delta \right\},
\]

where \( I \) is the identity matrix, and \( e_n \) the \( n \)-th unit vector for member \( \mathbf{x}_n \) of \( \mathbf{X}_\delta \). Here \( K_L \) is the Lipschitz constant for \( \hat{s}_{\tau^2}^2(\mathbf{x}) \) over \( \mathcal{X} \).
Theorem 3 shows that if design \( X_{D^c} \) is dense enough in \( X \), and \( \tau_s^2 \) is small enough, MICE is equivalent to ALM, since \( \hat{s}_{\tau_s^2}(\mathbf{x}^*) \) is within \( (\varepsilon + b\tau^4) \) to the constant \( \sigma^2 \tau_s^2 \) for any \( \mathbf{x}^* \), where \( \varepsilon > 0 \) can be made arbitrary small. The size of \( b \) decreases rapidly with \( \tau_s^2 \). MICE behaves as ALM, since according to Theorem 3, if \( D^c \) is dense enough so that \( \varepsilon > 0 \) can be made arbitrary small, and \( \tau_s^2 \) large enough so that \( (K + \tau^2 I)^{-1} = \tau^{-2} I \) (approximately), then \( 0 < \hat{s}_{\tau_s^2}^2 < \varepsilon \). However, with our default value \( \tau_s^2 = 1 \), MICE is not expected to behave as ALM. MI, on the other hand, behaves as ALM whenever \( X_{D^c} \) is dense in \( X \), and \( \tau^2 \) is very small. Note that both the squared-exponential correlation function and Matérn with \( \nu = 5/2 \) are continuously differentiable \([16]\), hence Lipschitz continuous.

Figure 7 shows, for a two dimensional example, the designs obtained for the different sequential design algorithms. Observe that MICE with \( \tau_s^2 = 1 \) displays a centered and well-spaced design. ALM selects only boundary points, since the two points of the interior constitute the initial design. MI is the criterion that appears most reluctant to select boundary points.

![Design selection with ALM, ALC, MI and MICE on \([0, 1]^2\) with fixed Matérn covariance using \( \sigma^2 = 1 \), \( \nu = 5/2 \) and \( \xi = (0.4, 1)^T \). The black-solid dots are design points, and the remaining points are candidate points, with the color representing the score value (red-high, blue-low) for the corresponding design criterion. The initial design consists of the points \( x_1 = (0.3, 0.6)^T \) and \( x_2 = (0.7, 0.4)^T \).](image-url)

In MICE, we need to compute \( \hat{s}_{\tau_s^2}^2(D^c \setminus \{x\})(\mathbf{x}) \) for each \( \mathbf{x} \in X_{\text{cand}} \), which typically requires the Cholesky decomposition of the \( (M_{D^c} - 1) \times (M_{D^c} - 1) \) correlation matrix \( K_{D^c \setminus \{x\}} \). This adds up to a computational burden if \( D^c \) is large, and we thus propose the following implementation that only requires a single Cholesky decomposition. First, invert the \( M_{D^c} \times M_{D^c} \) correlation matrix \( K_{D^c} \). Then exploit the partitioned inverse formula for matrices in block-form: the inverse of

\[
K_{D^c}^{-1} = \begin{pmatrix}
K_{D^c}^{-1} & k_{D^c}(x) \\
k_{D^c}(x)^T & K(x, x)
\end{pmatrix}^{-1}
\]  

(20)
can be written as:

\[ K_{D^c}^{-1} = \begin{pmatrix} B & b_{12} \\ b_{21} & b \end{pmatrix} \tag{21} \]

where \( D^* := D^c(x) = D^c \setminus \{x\} \), \( B = K_{D^c}^{-1} + \frac{1}{k} K_{D^c}^{-1} k_{D^c}(x) k_{D^c}^T(x) K_{D^c}^{-1} \), \( b_{12} = -\frac{1}{k} K_{D^c}^{-1} k_{D^c}(x) \), \( b_{21} = -\frac{1}{k} k_{D^c}(x) K_{D^c}^{-1} \), and \( b = \frac{1}{k} \), where \( k = K(x,x) - k_{D^c}^T(x) K_{D^c}^{-1} k_{D^c}(x) \). This relates \( K_{D^c}^{-1} \) to \( K_{D^c}^{-1} \) and \( k_{D^c}(x) \) for any \( x \in X_{D^c} \) as follows: given \( K_{D^c}^{-1} \), we can obtain \( B, b_{12}, b_{21} \) and \( b \), directly from Equation (21), and then we find that \( K_{D^c}^{-1} = B - \frac{1}{b} b_{12} b_{21} \). Therefore, \( K_{D^c}^{-1} \) can be obtained from \( K_{D^c}^{-1} \) in \( \mathcal{O}(M_{Dc}^2) \).

5 Computational complexity

To assess the performance of a sequential design strategy one should not only look at the prediction accuracy as the number of design points increases, but also take computational complexity into account. The overall computational cost is the sum of the computational cost of of fitting the GP, generating the candidate set \( X_{cand} \), evaluating the design criterion over the candidate set, and finally evaluating the simulator for those design points included in the design \( D \), that is,

\[ T_{total} = T_{mle} + T_{cand} + T_{select} + T_{model}. \tag{22} \]

The estimation of the correlation parameters with MLE, is in \( \mathcal{O}(pm^2 + M_{mle}m^\omega) \), where \( \omega > 0 \) is related to the efficiency of the algorithm used for matrix inversion: for naïve Gaussian elimination \( \omega = 3 \), and for Strassen’s algorithm \( \omega = \log_2(7) \). The term \( pm^2 \) is the number of operations needed to determine the distances between distinct pairs of points in \( X_D \) (which is \( m(m - 1)/2 \)). The second term \( M_{mle}m^\omega \) is the time complexity for MLE, which is equal to the cost as inverting the correlation matrix \( K_\xi \), \( M_{mle} \) times. \( M_{mle} \) is the number of trial points explored to obtain the MLE estimates \( \hat{\xi} \). To train the GP, which entails the determination of the weights of the linear predictor, only matrix multiplications are required, of order \( \mathcal{O}(m^2) \). The time complexity for evaluating the predictive mean at an untried point is \( \mathcal{O}(pm) \), and for the predictive variance it is \( \mathcal{O}(pm^2) \).

| Algorithm | Total time complexity for design size \( M_D \) |
|-----------|-----------------------------------------------|
| ALM       | \( \mathcal{O} \left( M_{mle}M_1^{1+\omega} + M_{cand}pM_2^2 \right) \) |
| ALC       | \( \mathcal{O} \left( M_{mle}M_1^{1+\omega} + M_{cand}M_3pM_2^3 \right) \) |
| MICE      | \( \mathcal{O} \left( M_{mle}M_1^{1+\omega} + M_DM_2^{\omega} + M_{cand}pM_3^2 + M_{cand}MDpM_4^2 \right) \) |

The computational complexity of the different algorithms are given in Table 1. For ALC, we used the formulation given by Equation (11), which is the one with lowest computational cost. The time complexity for a single ALM step is \( \mathcal{O} \left( pm^2 + M_{mle}m^\omega + M_{cand}pm \right) \), where \( M_{cand} \) is the number of points in the candidate set. The total cost for ALM is \( \mathcal{O} \left( pM_3^2 + M_{mle}M_1^{1+\omega} + M_{cand}pM_2^2 \right) \), where \( M_D \) is the number of points in the final design. Here \( M_r \) is the number of trial points used in ALC for averaging over the design space. To simplify the expressions given in Table 1 let \( M_{cand} = M_r = M_D \) and \( M_{Dc} = M_D \), and one obtains: \( \mathcal{O} \left( M_{mle}M_1^{1+\omega} + pM_3^2 \right) \) for ALM; \( \mathcal{O} \left( M_{mle}M_1^{1+\omega} + pM_4^2 \right) \) for ALC, and \( \mathcal{O} \left( M_{mle}M_1^{1+\omega} + pM_5^4 \right) \) for MICE. ALC clearly has the lowest computational complexity, and MICE is less expensive than ALC. However, MICE is less expensive than ALC as long as the ratio \( M_{Dc}/M_D \) is not too large. ALC becomes computationally prohibitive for large \( M_D \). The term \( M_{mle}M_1^{1+\omega} \) is the most expensive term for small to moderately large \( M_D \). Computational savings could be made if estimates for the correlation parameters only are updated for certain carefully
selected iteration steps (e.g. every 10 or so, or adaptively), not at each iteration. It would give a much bigger relative advantage to MICE against ALC, with a lower $T_{mle}$, see Figure 8.

Figure 8: Stacked histogram of the computational cost of the different sequential design algorithms, divided into $T_{mle}$, $T_{cand}$ and $T_{select}$, in terms of the total elapsed time for selecting designs of different sizes, applied to the Oscillatory function over $[0, 1]^4$. MLE for estimation of the correlation parameters is applied whenever the current design is greater than 10, otherwise fixed. The candidate sets generated are LHDs, selected based on the maximin distance to the current design from 500 LHD candidates.

The cost to generate candidate sets depends greatly on the desired size, and the sampling technique used. For instance, minimax designs are more computationally intensive than maximin designs [2].

6 A numerical comparison

In this section, we present a numerical comparison between MICE, ALM and ALC. MmLHD and mMLHD, which are widely used, have been included to represent affordable non-sequential strategies. MmLHDs tend to cover the parameter space better than Mm-distance designs, which are not restricted to the class of LHDs, but at the expense of lower Mm-distance scores. Hence, MmLHD can be seen as a compromise between Mm- and mM-distance designs [2]. The computational budget is limited to 150 design points; we found this to be reasonable budget. The results presented will provide a good foundation to understand the different sequential design strategies. The metric of success is primarily the RMSPE against design size. The normalized RMSPE is calculated using a hold-off design set which includes 1000 pre-evaluated design points. The hold-off set is a LHD. The test functions have been carefully selected in order to achieve diversity in both input dimension size $p$ and difficulty level. All results are presented as averages of ten runs. For the sequential strategies, what differs between the runs is the initial data of size 2, which is generated by mMLHD with 1000 reference points. Note that all strategies are compared using the same initial training data for the runs. For MmLHD and mMLHD, we simply generated ten different designs for each specific design size (50,75,100,120). The variability in the results is not always presented to increase visibility; it is however similar across the different strategies, see, e.g., the 95%-credible bands provided in Figure 13. We have also included the run time of the algorithms to express algorithm efficiency.

A stationary GP with a Matèrn covariance with $\nu = 5/2$ is used. The size of the candidate set has a significant effect on the results, and therefore we decided to declare the number of candidate
points in the method name, for instance we denoted by MICE-150 the MICE algorithm with $M_{\text{cand}} = 150$. Because the algorithm cost of ALC is significantly higher than ALM and MICE, the size of the candidate set is set as $M_{\text{cand}} = 150$. We let MICE and ALC have the same candidate set size, in order to ensure a fair comparison in terms of RMSPE. For ALM we use $M_{\text{cand}} = 1000$ due to its low algorithm cost for point selection ($T_{\text{select}}$). The candidate sets generated are LHDs, selected based on the maximin criterion with respect to the current design (from a set of 500 LHD candidates).

The ALC algorithm is given by (14), with $M_r = M_{\text{cand}}$, as in [14, 28]. MmLHD and mLMLHD select a LHD from a pool of 1000 LHDs. The mM-distance is measured using 1000 reference points over $X$ on a LHD. The optimiser employed for maximum likelihood estimation is a real-coded genetic algorithm [11] with settings that require 1024 calls to the log-likelihood. The values for the uncertain correlation parameters in the correlation function are fixed until the current design is of a specific size ($20$ if $p > 4$, else $10$) to avoid unstable estimates. The nugget parameter for the GP model assigned to current design $D$ is $\tau_s^2 = 10^{-12}$. For MICE, the smoothing nugget is typically $\tau_s^2 = 1$ but we also included results for a range of different choices of $\tau_s^2$. For the comparison we also included $\tau_s^2 = 10^{-12}$, which would be the value used in the greedy MI algorithm, since then $\tau_s^2 = \tau_s^2$.

### 6.1 Alan Genz's Oscillatory function

The “Oscillatory” function belongs to a family of test functions [13] proposed by Alan Genz for the study of quadrature methods. The function is $y(x) = \cos(c \cdot x + 2\pi w)$, $x \in [0,1]^p$. The vector $c = (c_1, c_2, \ldots, c_p)^T$ determines the level of difficulty along the different coordinate directions of $X \subset \mathbb{R}^p$, and $w$ is the displacement. To study the impact of dimension size $p$ on the difficulty to predict untried points, $c$ is constrained as $\sum_{i=1}^p c_i = h$, $c_i > 0$, where $h$ can be held fixed in order to maintain the difficulty level of the problem for different choices of $p$. Two case examples are considered: $c = (1.85, 2.51, 1.94, 2.70)^T$ and $w = 0.43$ over $[0,1]^4$, and $c = (0.14, 1.69, 0.81, 1.73, 2.10, 0.42, 0.14, 1.97)^T$ and $w = 0.4$ over $[0,1]^8$, where $h = 9$.

![Figure 9: Left: comparison between algorithms for the Oscillatory function over $[0,1]^4$. Right: the performance with MICE-150 for different choices of $\tau_s^2$.](image)

As can be observed in Figure 9, the sequential strategies outperform the ones based on LHDs, as expected. MmLHD and mLMLHD, even if well spaced, do not take into account that $y(x)$ is anisotropic. The worst performing sequential design strategy presented is MICE-150 with $\tau_s^2 = 10^{-12}$, which in fact is equal to the MICE algorithm a criterion equal to the MI criterion. The
Figure 10: Left: comparison between algorithms for the Oscillatory function over $[0,1]^8$. Right: the performance with MICE-150 for different choices of $\tau_s^2$.

Poor performance is evidently due to the MI criterion as explained in Section 4.3. In the 4-D case, ALM-1000, ALC-150, and MICE-150 (with the recommended nugget level $\tau_s^2 = 1$) produce similar result in terms of prediction error, but as shown in Figure 8, the run time for ALC is significantly higher than for ALM and MICE, which makes it the least attractive of them all if $y(x)$ is relatively inexpensive to evaluate. Even if one assumes that ALC-50 would produce a similar performance as ALC-150, it is still quite expensive and therefore not competitive. Observe that $\tau_s^2 = 1$ performs the best.

### 6.2 Piston simulation function

Here we consider a 7-dimensional example given in [3], where the output describes the circular motion of a piston within a cylinder; it obeys the following equations:

$$y(x) = 2\pi \sqrt{\frac{x_1}{x_2 + x_3^2 x_4 x_5 x_6 x_7 g_1(x)}},$$

where

$$g_1(x) = \frac{x_3}{2x_2} \left( \sqrt{g_2(x)^2 + 4x_2 \frac{x_4 x_5}{x_6} x_7 - g_2(x)} \right)$$

$$g_2(x) = x_3 x_4 + 19.62 x_1 - \frac{x_2 x_5}{x_3}$$

Here $y(x)$ is the cycle time (s) which varies with seven input variables. The design space is given by $x_1 \in [30, 60]$ (piston weight, kg), $x_2 \in [1000, 5000]$ (spring coefficient, N/m), $x_3 \in [0.005, 0.020]$ (piston surface area, m$^2$), $x_4 \in [90000, 110000]$ (atmospheric pressure, N/m$^2$), $x_5 \in [0.002, 0.010]$ (initial gas volume, m$^3$), $x_6 \in [340, 360]$ (filling gas temperature, K) and $x_7 \in [290, 296]$ (ambient temperature, K). For the simulation, the input values have been normalized using the min-max ranges. The nonlinearity makes this deterministic computer experiment problem very challenging to emulate. In addition, we included MICE-300, which showed a slight improvement over MICE-150, see Figure 11. MICE with 300 candidate points is not that much more expensive than with 150; in fact, it is significantly cheaper computationally than ALC with 150. Again, the proposed algorithm MICE performs the best. For high-dimensional problems, ALM tends to be the worst, probably due to the high percentage of points on the boundary.
7 Application to a tsunami computer model

There is a pressing need for UQ in tsunami modeling in order to provide accurate risk maps or issue informative warnings. Sarri, Guillas and Dias [27] were the first to demonstrate that statistical emulators can be used for this purpose. Recently, Sraj et al. [30] studied the propagation of uncertainty in Manning’s friction parameterization to the prediction of sea surface elevations, for the Tohoku 2011 tsunami event. They used a polynomial chaos (PC) expansion as the surrogate model of a low resolution tsunami simulator. However, for UQ, Bilionis and Zabaras [4] showed that GPs can outperform PC expansions, especially for small to moderate-sized designs that are only available for high resolution (e.g. 100-200m) tsunami simulations.

We consider here the problem of predicting the maximum free-surface elevation of a tsunami wave at the shoreline, for a wide range of untried conditions, following a subaerial landslide at an adjoining beach across a large body of shallow water. A tsunami wave simulator is used, whose cost is included in our study. We demonstrate the efficiency of the different sequential design methods for the design of a realistic computer experiments. This problem, is inspired by a benchmark problem, given at the Catalina 2004 workshop on long-wave runup models used in the validation of tsunami models. A landslide of seafloor sediments, initially at the beach, has a Gaussian shaped mass distribution, and generates tsunami waves that propagates towards the opposite shoreline across from the beach (see Figure 12). The sea-floor bathymetry is changing over time, and is used as input to the tsunami model simulator. The beach floor motion is described by the change in bathymetry of the sloping beach over time, $h(x, t) = H(x) - h_0(x, t)$, where $H(x) = x \tan \beta$ is the static uniformly sloping beach, and $h_0(x, t) = \delta \exp \left(-\left(\frac{x}{\delta} - \bar{x}\right)^2\right)$ is the perturbation with respect to $H(x, t)$. Here $\bar{x} = 2\frac{x \mu}{\delta \tan \phi_1}, \bar{t} = \sqrt{\frac{\delta}{\mu}} t, \delta$ is the maximum vertical slide thickness, $\mu$ is the ratio of the thickness and the slide length, and $\tan \phi_1$ is the beach slope. The free surface elevation is defined as $z(x, t) = -h(x, t)$. It is assumed the initial water surface is undisturbed, that is, $z(x, 0) = 0$ for all $x$. The slope $\tan \phi_2$ of the beach at the opposite shoreline is chosen so that the distance between the shorelines is 2800 m. This is a shallow water problem, which means that $\tan \phi_1 \ll 1$, and that the translating mass movement is thin ($\mu = \delta/L \ll 1$).

We use the state-of-the-art numerical code VOLNA [12] to simulate all stages of this landslide-generated tsunami event, based on nonlinear shallow water equations. We run VOLNA on a single GPU on the cluster Emerald. The bathymetry defined above is given only along one spatial coordinate, but in the code implementation of VOLNA a second spatial dimension (in this case, along the shoreline) is added to cover 10 meters of shoreline. The mesh is defined on $[-5, 5] \times [0, 3000]$.
Figure 12: Case example: landslide-generated tsunami event.

The domain of interest for emulation is $\phi_1 \in [35^\circ, 70^\circ]$, $\phi_2 \in [35^\circ, 70^\circ]$, $h \in [500.0, 1000.0]$, and $\mu = [0.01, 0.1]$.

Figure 13: Error convergence with 95%-credible bands, and runtime of the sequential design strategies for a simple tsunami problem. Note the log-scale in the lower right figure.

For MICE and ALC, $M_{\text{cand}} = 150$, and for MICE $\tau_s^2 = 1$ is used. As before, ALM uses $M_{\text{cand}} = 1000$ as it is relatively cheap computationally. The mean function is assumed to be a constant. The results are averages of ten runs. As before, the GPs have a constant mean, and use the Matèrn covariance $\nu = \frac{5}{2}$. The normalized RMSPE and the maximum error are calculated using...
a hold-off set of size 500. As observed in Figure 13, MICE performs better than ALC, especially as the design increases in size. Observe that, for design size 120, the maximum error in sea surface elevation is less than 1 meter for waves up to almost 10 meters. Note that in the bottom right figure the total run time is given in logarithmic scale with base 10, and the computational savings that can be made by using MICE are thus around 10-20% (and much higher if MLE were not used at each iteration). A single run of VOLNA takes on average 850 seconds. The time consumed by the simulator is represented by a grey dashed line in Figure 13 (bottom left figure). For a fully realistic tsunami scenario we expect a substantially longer run time for the tsunami simulator, which will make the MICE algorithm even more computationally advantageous.

8 Conclusion

We have shown for a few test functions, and a realistic tsunami simulator, that MICE is able to outperform traditional sequential and non-sequential algorithms, such as ALC, ALM, and LHDs. It naturally extends MI from a spatial context to computer experiments, where several parameters are considered and the covariance structure of the GP is not fully known. MICE is particularly attractive in terms of the time complexity of the whole process (MLE to fit the GP surrogate, draws of candidates, selection of inputs amongst these candidates, and run of the simulator at those inputs). In addition, MICE may outperform even more other designs with less frequent MLE estimates, and this will be investigated in the future. Our theoretical results enhance our understanding of the effect of the nugget parameter on the estimation of the variance, a key ingredient in MICE, and also demonstrate near-optimality of our approach. We envision a large impact of our design method in several fields of research, such as realistic tsunami modeling. Finally, an area for further research could be to use the MICE algorithm in a non-stationary setting, e.g., in the treed GP form [14].

Appendix. Proofs of theorems

Proof. [Theorem 1] Given a GP model with constant mean, known correlation function, and a non-negative nugget parameter of size $\tau^2$, then the predictive variance for $Y(\cdot)|D$ for any point $x_i \in X_D$ can be written as:

$$s^2_{\tau^2}(x_i) = \sigma^2 \left( 1 - k^T(x_i) \left( K + \tau^2 I \right)^{-1} k(x_i) \right) + \left( 1^T (K + \tau^2 I)^{-1} k(x_i) \right)^2,$$

where $I$ is the $m \times m$ identity matrix, then

$$k^T(x_i) \left( K + \tau^2 I \right)^{-1} k(x_i) = k^T(x_i) \left( K + \tau^2 I \right)^{-1} (k(x_i) + \tau^2 e_i) - \tau^2 k^T(x_i) \left( K + \tau^2 I \right)^{-1} e_i$$

$$= k^T(x_i)e_i - \tau^2 k^T(x_i)(e_i^T(K + \tau^2 I)^{-1})$$

$$= 1 - \tau^2 k^T(x_i)(e_i^T(K + \tau^2 I)^{-1})$$

$$= 1 - \tau^2 e_i^T(1)(K + \tau^2 I)^{-1}$$

$$= 1 - \tau^2 e_i^T(1)$$

$$= 1 - \tau^2 + \tau^4 e_i^T(1),$$

where $e_i$ is the $i$-th unit vector. Similarly, $1^T K^{-1} k(x_i) = 1 - \tau^2 e_i^T(1)$. Insert these results into $s^2_{\tau^2}(x^*)$, with $x^* = x_i \in X$, and we obtain

$$s^2_{\tau^2}(x_i) = \sigma^2 \left( \tau^2 - \tau^4 e_i^T(1) + \tau^4 \frac{(e_i^T(K + \tau^2 I)^{-1})^2}{1^T(K + \tau^2 I)^{-1}} \right).$$

21
Proof. [Theorem 2] Let \( X_\delta \subset \mathcal{X} \) be a regular grid with spacing \( \delta > 0 \), where \( \mathcal{X} \) is a compact subset of \( \mathbb{R}^p \). Assume the correlation function \( K(\cdot, \cdot) \) is Lipschitz continuous, then there exists a constant \( K_L > 0 \) such that \( |s^2(x_1) - s^2(x_2)| \leq K_L \|x_1 - x_2\|_2 \) for all \( x_1, x_2 \in X_\delta \), where \( \| \cdot \|_2 \) is the Euclidean norm. Assume a Gaussian random field model with constant mean, and a non-negative nugget parameter of size \( \tau^2 \). Then, for any \( \varepsilon > 0 \), assuming \( X_\delta \) has grid spacing \( \delta \leq 2\varepsilon/(\sqrt{3}K_L) \), \( s^2(x^*) \) is \( \varepsilon \)-close to \( \hat{s}^2(x_n) \) for any untried point \( x^* \in \mathcal{X} \), where \( x_n \) is the member of \( X_\delta \) closest to \( x^* \). According to Theorem 1 for any point \( x_i \in X_\delta \) the predictive variance \( \delta \) can be written as:

\[
\hat{s}^2(x_i) = \sigma^2 \left( \tau^2 - \sigma^2 \frac{e_i^T (K + \tau^2 I)^{-1} e_i + \tau^4 \frac{(e_i^T (K + \tau^2 I)^{-1} e_i)^2}{1^T (K + \tau^2 I)^{-1} 1}}{1^T (K + \tau^2 I)^{-1} 1} \right),
\]

where \( I \) is the identity matrix, and \( e_i \) the \( i \)-th unit vector. Hence, for any \( \varepsilon > 0 \) there exists a grid spacing \( \delta \geq \delta \) so that \(-\sigma^2\tau b_1(\tau^2) - \varepsilon < s^2(x^*) - \sigma^2\tau^2 < \sigma^2\tau b_1(\tau^2) + \varepsilon\), where \( b_1(\tau^2) = \max\{e_i^T (K + \tau^2 I)^{-1} e_i : x_i \in X_\delta\} \), and \( b_2(\tau^2) = \max\left\{ \frac{(e_i^T (K + \tau^2 I)^{-1} e_i)^2}{1^T (K + \tau^2 I)^{-1} 1} : x_i \in X_\delta\right\} \).

Proof. [Theorem 2] This proof follows the proof of Lemma 5 in [20]. Let \( F_1, F_2 \) be regular grids with spacing \( 2\delta \) that are assumed to cover \( \mathcal{X} \) in terms of compactness. \( F_1, F_2 \) are finite subsets of \( \mathcal{X} \), and \( F_2 \) is obtained by translating \( F_1 \) by distance \( \delta \) in Euclidean norm. \( F_1 \) is the set of points admissible for selection. Let us use the notation \( \tilde{x} \) to represent the translated point in \( F_2 \) of \( x \) in \( F_1 \). Hence, \( \|x - \tilde{x}\| \geq \delta, \forall x \in F_1 \). Moreover, since \( \mathcal{X} \) is compact, and \( K(\cdot, \cdot) \) is continuous, \( K(\cdot, \cdot) \) is uniformly continuous over \( \mathcal{X} \). Thus, \(|K(x, x') - K(\tilde{x}, \tilde{x}')| \leq \varepsilon_1, \forall x, x' \in F_1 \). Let \( D \) be a subset of \( F_1 \), and \( x \in F_1 \setminus D \).

Assume that \( \mathcal{H}(x|D) \geq \mathcal{H}(x|\tilde{D}) \) for \( |D| \leq 2M \), which is empirically justified [20]. For MICE, we use a GP for \( D \) with a nugget parameter \( \tau_1^2 > 0 \), and a GP for \( \tilde{D} \) with a nugget \( \tau_2^2 > 0 \). First, let us determine an upper bound for \( |s^2_D(x) - s^2_{\tilde{D}}(x)| \):

\[
|s^2_D(x) - s^2_{\tilde{D}}(x)| = \sigma^2 |k^T_D(x)K^{-1}k_D(x) - k^T_{\tilde{D}}(x)K^{-1}k_D(x)| \\
\leq \sigma^2 \|k^T_D(x) - k^T_{\tilde{D}}(x)\|_2 \|K^{-1}\|_2 \|k_D(x)\|_2 + \|k_{\tilde{D}}(x)\|_2 \|K^{-1}\|_2 \|k_D(x)\|_2.
\]

Since \( K(\cdot, \cdot) \) is uniformly continuous over \( \mathcal{X} \), we know that \( \forall \varepsilon_2 > 0 \) there exists a spacing \( \delta \) such that, for \( \|x - \tilde{x}\| \leq \delta \), \( |K(x, x') - K(x, x')| \leq \varepsilon_1 \) for \( x \neq x' \), and \( \|K_{D,\tau_2} - K_{\tilde{D},\tau_2}\|_2 \leq \sqrt{M}\varepsilon_1 + \sqrt{M}\tau_1^2 - \tau_2^2 \). We also derive \( \|k^T_D(x) - k^T_{\tilde{D}}(x)\|_2 \leq \varepsilon_1 \sqrt{M} \), and similarly, \( \|k^T(x)\|_2 \leq C \sqrt{M} \), where \( \| \|_2 \) is the Euclidean norm, and \( C = \max_{x \in \mathcal{X}} K(x, x) \). We assume wlog that \( C = 1 \). Furthermore:

\[
\|K_{D,\tau_1}^{-1} - K_{\tilde{D},\tau_2}^{-1}\|_2 = \left\|K_{D,\tau_1}^{-1}(K_{D,\tau_2} - K_{\tilde{D},\tau_2})K_{\tilde{D},\tau_2}^{-1}\right\|_2 \leq \left\|K_{D,\tau_1}^{-1}\right\|_2 \left\|K_{D,\tau_2} - K_{\tilde{D},\tau_2}\right\|_2 \left\|K_{\tilde{D},\tau_2}^{-1}\right\|_2 \leq (1 + \tau_2^2)^{-1}(1 + \tau_2^2)^{-1}\sqrt{M}(M\varepsilon_1 + |\tau_2^2 - \tau_1^2|) \leq \sqrt{M}\varepsilon_1 + \sqrt{M}|\tau_2^2 - \tau_1^2|, \\
\]

where we used that \( K \) is positive semidefinite, which means that \( \|K^{-1}\|_2 \leq \lambda_{\min}(K)^{-1} \leq (1 + \tau_2^2)^{-1} \), where \( \lambda_{\min}(K) \) is the smallest eigenvalue. We thus obtain the following bound:

\[
|\hat{s}^2_{D,\tau_1}(x) - \hat{s}^2_{\tilde{D},\tau_2}(x)| \leq \sigma^2 \varepsilon_1 M(1 + \tau_1^2)^{-1} + M(1 + \tau_2^2)^{-1}(1 + \tau_2^2)^{-1}\sqrt{M}(M\varepsilon_1 + |\tau_2^2 - \tau_1^2|) \\
\leq \sigma^2 \varepsilon_1 M + M\sqrt{M}(M\varepsilon_1 + |\tau_2^2 - \tau_1^2|).
\]

Then, for any \( \varepsilon > 0 \) we can choose the grid spacing \( \delta > 0 \) such that \( \varepsilon \geq \varepsilon_1\tau_2^2 \sigma^2 M(2M + M^3/2) \). Hence, \( |\hat{s}^2_{D,\tau_1}(x) - \hat{s}^2_{\tilde{D},\tau_2}(x)| \leq \varepsilon\tau_2^2 + \sigma^2 M^3/2|\tau_1^2 - \tau_2^2| \), and, in turn,

\[
\mathcal{H}_{\tau_1}(x|D) - \mathcal{H}_{\tau_2}(x|\tilde{D}) = \frac{1}{2} \log \left( \frac{s^2_{D,\tau_1}(x)}{s^2_{\tilde{D},\tau_2}(x)} \right) = \frac{1}{2} \log \left( 1 + \frac{(s^2_{D,\tau_1}(x) - \hat{s}^2_{D,\tau_1}(x))/\hat{s}^2_{D,\tau_1}(x)}{s^2_{\tilde{D},\tau_2}(x)} \right) \\
\leq \frac{1}{2} \log \left( 1 + \varepsilon + M^{5/2}|\tau_1^2 - \tau_2^2|^2/\tau_2^2 \right) \leq \varepsilon + M^{5/2}/\tau_2^2 - \tau_1^2/\tau_2^2.
\]

22
We used that \( s_D^2(x) \geq \sigma^2 \tau^2_1 / M \) (see Theorem 1). Suppose that estimates are available for the correlation parameters \( \xi \); replacing \( K(x, x) \) by \( K(x, x'; \hat{\xi}) \) throughout the calculations above. Then, an extra term is added to \( s^2(x) \) to account for the parameter uncertainty \([32]\): \( s^2(x; \hat{\xi}) = \sigma^2 (1 - k^T(x; \hat{\xi})K^{-1}k(x; \hat{\xi})) + E \left( \left( \hat{y}(x; \xi) - \hat{y}(x; \hat{\xi}) \right)^2 \right) \). The estimates are updated at each greedy step, denoted by \( \hat{\xi}_i \), for greedy step \( i \). Using (2), with zero-mean, \( \hat{y}(x; \xi) - \hat{y}(x; \hat{\xi}) = k^T(x)K^{-1}y_D - k^T(x)K^{-1}\hat{\xi}_i y_D \). Let us assume that \( \|y_D\|_2 \leq \sqrt{M} \) (normalized). We know that there exists a constant \( \alpha \geq 0 \) such that, for all \( \{\hat{\xi}_i\}_{i=1}^M \), and for all, \( x, x' \in \mathcal{X} \), \( |K(x, x'; \xi) - K(x, x'; \hat{\xi})| \leq \alpha \). Then, \( E(\hat{y}(x; \hat{\xi}_i) - (\hat{y}(x; \xi))^2) = E((k^T(x)K^{-1}y_D - k^T(x)K^{-1}\hat{\xi}_i y_D)^2) \leq E((\|k^T(x) - k^T_{\xi_i}(x)\|_2\|K^{-1}_{\xi_i}\|_2\|y_D\|_2 + \|k^T(x)\|_2\|K^{-1}_{\xi_i} - K^{-1}(\hat{\xi}_i)\|_2\|y_D\|_2)^2) \leq \alpha^2 M^2(1 + M^{3/2})^2 \). As a result, using similar calculations, \( \hat{\mathcal{H}}(x|D) - \mathcal{H}(x|D, \hat{\xi}) \leq \frac{1}{2} \log((s_D^2(x) + \alpha^2 M^2(1 + M^{3/2})^2)/s_D^2(x)) \leq \frac{1}{2} \log(1 + (\alpha\sigma^{-1}\tau_1^{-1})^2 M^3(1 + M^{3/2})^2) \). Hence,

\[
\mathcal{H}(x|D, \hat{\xi}, \tau_1^2) - \mathcal{H}(x|\hat{D}, \hat{\xi}, \tau_1^2) = (\mathcal{H}(x|D, \tau_1^2) - \mathcal{H}(x|\hat{D}, \tau_1^2)) + (\mathcal{H}(x|D, \hat{\xi}, \tau_1^2) - \mathcal{H}(x|\hat{D}, \hat{\xi}, \tau_1^2))
\]

\[
+ (\mathcal{H}(x|D, \hat{\xi}, \tau_1^2) - \mathcal{H}(x|D, \tau_1^2)) + (\mathcal{H}(x|\hat{D}, \hat{\xi}, \tau_1^2) - \mathcal{H}(x|\hat{D}, \hat{\xi}, \tau_1^2)) \leq \varepsilon + 2(\alpha\sigma^{-1}\tau_1^{-1})^2 M^3(1 + M^{3/2})^2 + M^{5/2}(\tau_1^2)^{-2} - \tau_1^2/\tau_2^2.
\]

Both GPs (for \( D \) and \( \hat{D} \)) use the same estimates \( \hat{\xi} \). Finally, by following the same the proof of Theorem 7 in [20], we can easily get the result of this theorem.

Acknowledgments

The authors would like to acknowledge that the work presented here made use of the Emerald High Performance Computing facility made available by the Centre for Innovation. The Centre is formed by the universities of Oxford, Southampton, Bristol, and University College London in partnership with the STFC Rutherford-Appleton Laboratory.

References

[1] A. Andrianakis and P.G. Challenor. The effect of the nugget on gaussian process emulators of computer models. *Computational Statistics & Data Analysis*, 56(12):4215 – 4228, 2012.

[2] S. Ba and V.R. Joseph. Multi-layer designs for computer experiments. *J. Am. Stat. Assoc.*, 106(495), 2011.

[3] E.N. Ben-Ari and D.M. Steinberg. Modeling data from computer experiments: an empirical comparison of kriging with mars and projection pursuit regression. *Qual. Eng.*, 19(4):327–338, 2007.

[4] I. Bilionis and N. Zabaras. Multi-output local gaussian process regression: Applications to uncertainty quantification. *Journal of Computational Physics*, 231(17):5718 – 5746, 2012.

[5] D. Bursztyn and D.M. Steinberg. Comparison of designs for computer experiments. *Journal of Statistical Planning and Inference*, 136(3):1103 – 1119, 2006.

[6] W.F. Caselton and J.V. Zidek. Optimal monitoring network designs. *Statistics & Probability Letters*, 2(4):223–227, 1984.
[7] J.A. Christen and B. Sansó. Advances in the sequential design of computer experiments based on active learning. *Communications in Statistics-Theory and Methods*, 40(24):4467–4483, 2011.

[8] E. Contal and N. Vayatis. Gaussian process optimization with mutual information. *arXiv preprint arXiv:1311.4825*, 2013.

[9] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience, 2 edition, July 2006.

[10] C. Currin, T. Mitchell, M. Morris, and D. Ylvisaker. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *J. Am. Stat. Assoc.*, 86(416):953–963, 1991.

[11] K. Deb and R.B. Agrawal. Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148, 1995.

[12] D. Dutykh, R. Poncet, and F. Dias. The volna code for the numerical modeling of tsunami waves: Generation, propagation and inundation. *Eur. J. Mech. B-Fluid.*, 30(6):598–615, 2011.

[13] A. Genz. An adaptive numerical integration algorithm for simplices. In *Computing in the 90’s*, pages 279–285. Springer, 1991.

[14] R.B. Gramacy and Herbert K.H. Lee. Adaptive design and analysis of supercomputer experiments. *Technometrics*, 51(2), 2009.

[15] R.B. Gramacy and Herbert K.H. Lee. Cases for the nugget in modeling computer experiments. *Statistics and Computing*, 22(3):713–722, 2012.

[16] M.S. Handcock and M.L. Stein. A bayesian analysis of kriging. *Technometrics*, 35(4):403–410, 1993.

[17] X. Huan and Y.M. Marzouk. Simulation-based optimal bayesian experimental design for nonlinear systems. *Journal of Computational Physics*, 2012.

[18] D.R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

[19] C-W. Ko, J. Lee, and M. Queyranne. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691, 1995.

[20] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *J. Mach. Learn. Res.*, 9:235–284, 2008.

[21] S.N. Lophaven, H.B. Nielsen, and J. Søndergaard. A matlab kriging toolbox. 2002.

[22] C-Y. Peng and J. Wu. On the choice of nugget in kriging modeling for deterministic computer experiments. *Journal of Computational and Graphical Statistics*, (just-accepted), 2012.

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

[24] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[25] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical science*, 4(4):409–423, 1989.
[26] T.J. Santner, B.J. Williams, and W.I. Notz. The design and analysis of computer experiments. Springer, 2003.

[27] A. Sarri, S. Guillas, and F. Dias. Statistical emulation of a tsunami model for sensitivity analysis and uncertainty quantification. Nat. Hazards Earth Syst. Sci, 12:2003–2018, 2012.

[28] S. Seo, M. Wallat, T. Graepel, and K. Obermayer. Gaussian process regression: Active data selection and test point rejection. In Neural Networks, IEEE-INNS-ENNS International Joint Conference on, volume 3, pages 3241–3241. IEEE Computer Society, 2000.

[29] M.C. Shewry and H.P. Wynn. Maximum entropy sampling. J. Appl. Stat., 14(2):165–170, 1987.

[30] I. Sraj, K.T. Mandli, O.M. Knio, C.N. Dawson, and I. Hoteit. Uncertainty quantification and inference of manning’s friction coefficients using dart buoy data during the tohoku tsunami. Ocean Modelling, 2014. Accepted.

[31] M.L. Stein. Interpolation of spatial data: some theory for kriging. Springer Verlag, 1999.

[32] D.L. Zimmerman and N. Cressie. Mean squared prediction error in the spatial linear model with estimated covariance parameters. Annals of the Institute of Statistical Mathematics, 44(1):27–43, 1992.