

Adaptive Gaussian Process Regression for Bayesian inverse problems*

Paolo Villani[†] Jörg Unger[‡] Martin Weiser[†]

May 1, 2024

Abstract

We introduce a novel adaptive Gaussian Process Regression (GPR) methodology for efficient construction of surrogate models for Bayesian inverse problems with expensive forward model evaluations. An adaptive design strategy focuses on optimizing both the positioning and simulation accuracy of training data in order to reduce the computational cost of simulating training data without compromising the fidelity of the posterior distributions of parameters. The method interleaves a goal-oriented active learning algorithm selecting evaluation points and tolerances based on the expected impact on the Kullback-Leibler divergence of surrogated and true posterior with a Markov Chain Monte Carlo sampling of the posterior. The performance benefit of the adaptive approach is demonstrated for two simple test problems.

Keywords: Gaussian process regression, Bayesian inverse problems, surrogate models, parameter identification, active learning

MSC 2010: 60G15, 62F15, 62F35, 65N21

1 Introduction

The inverse problem of inferring the posterior probability of parameters $p \in \mathbb{R}^d$ in a forward model $y(p)$ from measurements $y^m \in \mathbb{R}^m$ is often addressed by sampling with Markov Chain Monte Carlo (MCMC) methods [5]. The large number of forward evaluations required for a faithful representation of the posterior density renders this inapplicable in case of computationally expensive forward models such as large finite element (FE) simulations. The forward model is thus often replaced by a fast surrogate model when sampling the posterior. Here, we focus on the efficient construction of Gaussian Process Regression (GPR) surrogates.

Surrogate models are learned from values $y(p_i)$ at specific evaluation points p_i as training data. The accuracy of the resulting surrogate depends on the number and position of the sample points. Constructing an accurate surrogate

*This work has been supported by Bundesministerium für Bildung und Forschung – BMBF, project number 05M20ZAA (siMLOpt) and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 436400679.

[†]Zuse Institut Berlin, {weiser,villani}@zib.de

[‡]Bundesanstalt für Materialforschung und -prüfung, joerg.unger@bam.de

model can become computationally expensive when a large number of evaluations is required. Consequently, strategies for selecting near-optimal evaluation points have been proposed for various settings [11]. A priori point sets [4, 10] are effectively supplemented by adaptive designs [3, 6, 8, 16] selecting the most beneficial evaluation points p_i .

When using FE simulations for computing training data, the evaluations of $y(p_i)$ are affected by discretization and truncation errors. The trade-off between accuracy and cost has been investigated using different low and high fidelity models [9], and by an adaptive choice of evaluation tolerances [12, 13, 14] in different settings. Here, we extend [13] from an offline training for maximum posterior point estimates to an interleaved posterior sampling and surrogate training driven by a goal-oriented approach.

2 Gaussian Process regression

Gaussian process regression is a regression technique which allows to approximate any function, naturally fits the Bayesian framework, and provides an uncertainty estimate of its prediction.

We consider a forward model $y : \mathbb{R}^d \rightarrow \mathbb{R}^m$, which we assume to be a realisation of a Gaussian process \mathcal{G} with mean $\mu_0 : \mathbb{R}^d \rightarrow \mathbb{R}^m$ and covariance kernel $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^{m \times m}$ to be defined later.

For training points $(p_i, y_i)_{i=1, \dots, s}$ with $y_i \approx y(p_i)$ of accuracy $\tau_i \geq 0$, we are interested in a prediction of $y_{s+1} \approx y(p_{s+1})$ for any p_{s+1} . The GPR posterior covariance block matrix is $\Gamma = (K^{-1} + T^{-2})^{-1} \in \mathbb{R}^{m(s+1) \times m(s+1)}$ with prior covariance blocks $K_{ij} = k(p_i, p_j)$ and formally likelihood covariance $T = \text{diag}(\tau_1 I, \dots, \tau_s I, \infty I)$. The GPR posterior mean is $\bar{Y} = \Gamma(K^{-1}M_0 + T^{-2}Y)$ with $Y = (y_1, \dots, y_s, 0)$. Then, the GPR prediction is the marginal normal distribution $y_{s+1} \sim \mathcal{N}(\bar{Y}_{s+1}, \Gamma_{s+1, s+1})$. As $p_{s+1} \in \Omega$ is arbitrary, this defines mean $\bar{y} : \Omega \rightarrow \mathbb{R}^m$ and covariance $\Gamma : \Omega \rightarrow \mathbb{R}^{m \times m}$ on the whole parameter space. We refer to [11, 13] for a more detailed exposition.

3 Bayesian surrogate-based parameter identification

We consider the forward model $y : \Omega \subset \mathbb{R}^d \rightarrow \mathbb{R}^m$, which cannot be evaluated directly, but can be approximated through a numerical procedure y_τ with arbitrary precision in exchange of computational work: We assume that for any $\tau > 0$, we obtain an evaluation $y_\tau(p) \sim \mathcal{N}(y(p), \tau I)$, with cost W_τ .

We assume measurements y^m to be random variables generated by a linear additive Gaussian noise model

$$y^m = y(p) + \eta \tag{1}$$

with $\eta \sim \mathcal{N}(0, \Sigma_l)$. For simplicity, we consider a diagonal covariance structure $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$, corresponding to independent noise components. The conditional distribution of the measurements is then $y^m | p \sim \mathcal{N}(y(p), \Sigma_l)$,

$$\pi(y^m | p) = (2\pi)^{-m/2} \det(\Sigma_l)^{-1/2} \exp\left(-\frac{1}{2} \|y^m - y(p)\|_{\Sigma_l^{-1}}^2\right)$$

is the likelihood of the problem. Evaluating the likelihood requires evaluating the forward model y , which we assume to be computationally expensive.

To reduce costs, we assume that y is a realisation of a GP, and introduce a GP surrogate model \mathcal{G} of predictive mean \bar{y} and variance Γ . For simplicity, we consider a surrogate with independent output components, i.e. diagonal covariance $\Gamma(p)$. The training points for this GP are given by numerical evaluations $y_{\tau_i}(p_i)$ of the forward model. These points and the corresponding evaluation tolerances τ_i form the training design \mathcal{D} . We postpone the question of how to build training designs to the next section.

To evaluate the likelihood, we could substitute the forward model with the mean estimate \bar{y} , obtaining

$$\pi_{\text{plug-in}}(y^m | p, \bar{y}) = (2\pi)^{-m/2} \det(\Sigma_l)^{-1/2} \exp\left(-\frac{1}{2}\|y^m - \bar{y}(p)\|_{\Sigma_l^{-1}}^2\right). \quad (2)$$

This, from a decision-theoretic point of view, corresponds to the minimisation of the L^1 loss [7], but ignores the uncertainty estimate given by the predictive variance: since y is assumed to be a realisation of \mathcal{G} , the measurement noise model (1) becomes $y^m = \mathcal{G}(p) + \eta$. Marginalizing over GP realizations results in a different conditional distribution of the measurements $y^m | p, \mathcal{G} \sim \mathcal{N}(\bar{y}(p), \Sigma_l + \Gamma(p))$ and in a marginal likelihood:

$$\pi_{\mathcal{D}}(y^m | p, \mathcal{D}) = (2\pi)^{-m/2} \det(\Sigma_l + \Gamma(p))^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\|y^m - \bar{y}(p)\|_{(\Sigma_l + \Gamma(p))^{-1}}^2\right), \quad (3)$$

see, e.g., [2]. Note that the conditional distribution is still Gaussian due to the normality of both the noise and the GP. Moreover, the likelihood $\pi_{\mathcal{D}}$ is closely related to the L^2 loss [7, 15]. Including the GP variance into the likelihood can be important for avoiding overconfident yet wrong posterior approximations by surrogated forward models, see Fig. 1 for an illustration.

By adopting a Bayesian point of view, we express prior belief on the parameter by assigning a prior distribution $\pi(p)$. Then, by Bayes' theorem, we obtain a true posterior distribution

$$\pi(p | y^m) = \frac{\pi(p) \pi(y^m | p)}{\pi(y^m)}, \quad (4)$$

corresponding to the true likelihood $\pi(y^m | p)$ and an approximate posterior

$$\pi(p | y^m, \mathcal{D}) = \frac{\pi(p) \pi(y^m | p, \mathcal{D})}{\pi(y^m | \mathcal{D})}, \quad (5)$$

corresponding to the likelihood approximation $\pi(y^m | p, \mathcal{D})$ as given in (2) and (3), respectively.

In both cases, the normalising constant $\pi(y^m)$ or $\pi(y^m | \mathcal{D})$, respectively, will not be computationally available, as it requires integration over the parameter space Ω : fortunately, it is not needed for posterior sampling by Markov-Chain Monte Carlo (MCMC) methods.

4 Posterior-oriented surrogate model

As in [15], we do not aim at building a surrogate which is globally accurate on the whole parameter space Ω , but at finding a design \mathcal{D} such that the approximate posterior is accurate, i.e. $\pi(p | y^m) \approx \pi(p | y^m, \mathcal{D})$. Repeatedly

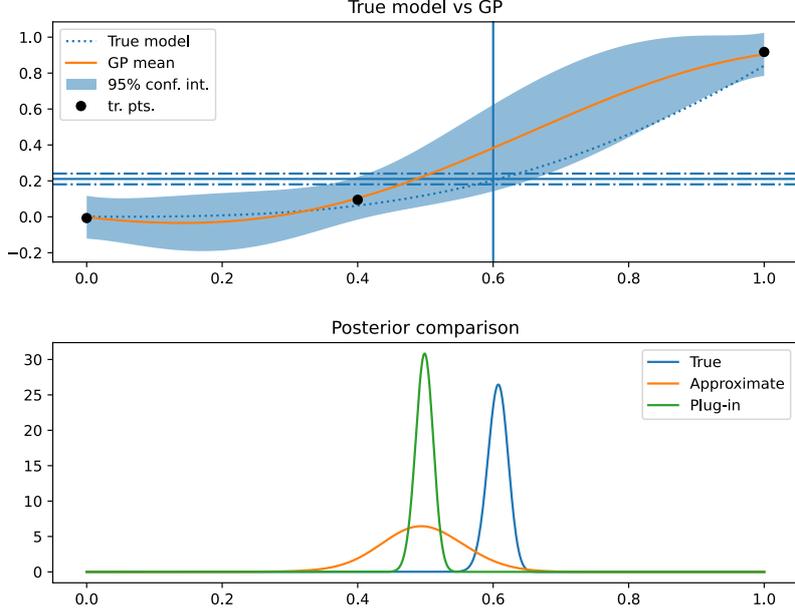


Figure 1: Impact of the likelihoods (2) and (3) on the posterior for an illustrative inverse problem with forward model $y(p) = p^2 \sin(p)$ and uniform prior on parameter space $[0, 1]$. The horizontal lines show the actual measurement and the 2σ range of measurement noise. The marginal likelihood (3) is wider due to including the GP variance, and avoids overconfident posteriors.

selecting training points randomly sampled from $\pi(p | y^m, \mathcal{D})$, updating \mathcal{G} and then iterating is sufficient for convergence of $\pi(p | y^m, \mathcal{D})$ to $\pi(p | y^m)$ in the Hellinger metric [2]. Here, we also aim at finding a design \mathcal{D} which incurs a small computational cost of evaluating training data y_{τ_i} .

We measure the deviation of the surrogated and the true posterior densities by the Kullback-Leibler (KL) divergence

$$\begin{aligned} D_{\text{KL}}(\pi(\cdot | y^m) | \pi(\cdot | y^m, \mathcal{D})) &= \mathbb{E}_{\pi(p | y^m)} \left[\log \frac{\pi(p | y^m)}{\pi(p | y^m, \mathcal{D})} \right] \\ &= \int_{\Omega} \pi(p | y^m) \log \frac{\pi(p | y^m)}{\pi(p | y^m, \mathcal{D})} dp. \end{aligned} \quad (6)$$

Since computing the KL divergence requires evaluating the full model, we derive a numerical approximation which relies on the surrogate only. Using the marginal likelihood (3) and the posteriors (4) and (5), their logarithmic ratio can be written as

$$\log \frac{\pi(p | y^m)}{\pi(p | y^m, \mathcal{D})} = \log \frac{\pi(y^m | p)}{\pi_{\mathcal{D}}(y^m | p, \mathcal{D})} - \log \frac{\pi(y^m)}{\pi(y^m | \mathcal{D})}.$$

The first term, the logarithmic ratio of true and surrogated likelihood, equals

$$\begin{aligned} & \log \frac{\pi(y^m | p)}{\pi_{\mathcal{D}}(y_m | p, \mathcal{D})} \\ &= \frac{1}{2} \left(\log \frac{\det(\Sigma_l + \Gamma(p))}{\det(\Sigma_l)} - \|y(p) - y^m\|_{\Sigma_l^{-1}}^2 + \|\bar{y}(p) - y^m\|_{(\Sigma_l + \Gamma(p))^{-1}}^2 \right). \end{aligned}$$

As $\Sigma_l^{-1} - (\Sigma_l + \Gamma(p))^{-1} \succeq 0$, we can upper bound the difference between norms by

$$\begin{aligned} & -\|y(p) - y^m\|_{\Sigma_l^{-1}}^2 + \|\bar{y}(p) - y^m\|_{(\Sigma_l + \Gamma(p))^{-1}}^2 \\ & \leq -\|y(p) - y^m\|_{\Sigma_l^{-1}}^2 + \|\bar{y}(p) - y^m\|_{\Sigma_l^{-1}}^2 \\ & = -\|y(p) - \bar{y}(p)\|_{\Sigma_l^{-1}}^2 - 2(\bar{y}(p) - y^m)^T \Sigma_l^{-1} (y(p) - \bar{y}(p)). \end{aligned}$$

By assuming that y is a realisation of \mathcal{G} , $\mathbb{E} \left[(y^{(i)}(p) - \bar{y}^{(i)}(p))^2 \right] = \Gamma^{(i,i)}(p)$ and therefore $\|y(p) - \bar{y}(p)\|_{\Sigma_l^{-1}}^2 \approx \text{tr}(\Sigma_l^{-1} \Gamma(p))$ hold. Defining $v = \Sigma_l^{-1} \sqrt{\text{diag}(\Gamma(p))} \in \mathbb{R}^m$, we obtain

$$-\|y(p) - y^m\|_{\Sigma_l^{-1}}^2 + \|\bar{y}(p) - y^m\|_{(\Sigma_l + \Gamma(p))^{-1}}^2 \lesssim -\text{tr}(\Sigma_l^{-1} \Gamma(p)) + 2|\bar{y}(p) - y^m|^T v.$$

We therefore define the local error quantity

$$e_{\mathcal{D}}(p) := \frac{1}{2} \left(\log \det(I + \Sigma_l^{-1} \Gamma(p)) - \text{tr}(\Sigma_l^{-1} \Gamma(p)(I - \Sigma_l^{-1})) + 2|\bar{y}(p) - y^m|^T v \right) \quad (7)$$

$$\gtrsim \log \frac{\pi(y^m | p)}{\pi_{\mathcal{D}}(y_m | p, \mathcal{D})}$$

as an approximate upper bound on the log ratio of true and surrogated likelihood.

By optimistically assuming that the normalisation factors are similar independent of the design \mathcal{D} , and thus $\log \frac{\pi(y^m)}{\pi(y^m | \mathcal{D})} \approx 0$, we substitute (7) into (6) and obtain the global error quantity

$$E(\mathcal{D}) = \int_{\Omega} e_{\mathcal{D}}(p) \pi(p | y^m) dp. \quad (8)$$

To create an optimal surrogate model, we aim at a training design \mathcal{D} which minimises $E(\mathcal{D})$ under a computational work constraint. By denoting the computational work needed to realize \mathcal{D} by $W(\mathcal{D})$, for a given budget W we aim at solving the optimisation problem

$$\min_{\mathcal{D}} E(\mathcal{D}) \quad \text{subject to} \quad W(\mathcal{D}) \leq W. \quad (9)$$

5 Sequential design of experiments

It is far from trivial to predict a priori how design choices impact the error quantity E , especially when a large budget W is available or the initial surrogate is

unreliable. Fortunately, an exact solution of (9) is not needed – an approximate solution will do, even if it yields a slightly less efficient design. We follow [13, 14] and adopt a greedy sequential approach, where the budget $W = \sum_{j=1}^J \Delta W_j$ is partitioned and sequentially spent.

We start from an initial design \mathcal{D}_0 and then, for $j = 1, \dots, J$, aim at solving

$$\min_{\mathcal{D}_j \leq \mathcal{D}_{j-1}} E(\mathcal{D}_j) \quad \text{s.t.} \quad W(\mathcal{D}_j | \mathcal{D}_{j-1}) \leq \Delta W_j. \quad (10)$$

We write $\mathcal{D} \leq \mathcal{D}_{j-1}$ for any design \mathcal{D} which refines \mathcal{D}_{j-1} in the sense that it includes all evaluation points p_i contained in \mathcal{D}_{j-1} with lesser or equal tolerances τ_i . We write $W(\mathcal{D} | \mathcal{D}_{j-1}) = W(\mathcal{D}) - W(\mathcal{D}_{j-1})$ for the work needed to obtain \mathcal{D} from \mathcal{D}_{j-1} .

Even this sequential formulation is highly non-linear and non-convex. An accurate solution would require a considerable amount of computational work, possibly exceeding the savings in computational budget possible with a better design. Consequently, we adopt the heuristic approach of separating the selection of new candidate evaluation points from the optimisation of the evaluation tolerances. In the latter, we also decide about the actual inclusion of the new points in the training set.

Candidate points. We choose points where spending computational budget is likely to reduce the error most. In order to do so, we look at the sensitivity of the global error E with respect to a reduction of training error at a candidate position p' [14]. This is given by

$$\begin{aligned} \frac{dE(\mathcal{D})}{dW(p')} &= \int_{\Omega} \frac{de_{\mathcal{D}}(p)}{dW(p')} \pi_{\mathcal{D}}(p | y_m) dp \\ &= \int_{\Omega} \frac{de_{\mathcal{D}}(p)}{d\Gamma(p)} \frac{d\Gamma(p)}{d\tau(p')} \bigg|_{\tau=\tau'} \frac{d\tau(p')}{dW(p')} \bigg|_{\tau=\tau'} \pi_{\mathcal{D}}(p | y_m) dp, \end{aligned} \quad (11)$$

where the linearization tolerance τ' is the current GP standard deviation at point p' . We adopt (11) as a utility function and select local minimizers of $\frac{dE(\mathcal{D}_{j-1})}{dW}$ as next candidate points.

The optimisation problem is solved approximately via a multistart pattern search. Quadrature is performed by Monte Carlo integration on samples \mathcal{S}_j to be defined in Sec. 6 below. This results in the numerical utility function

$$\frac{dE(\mathcal{D}_{j-1})}{dW(p')} \approx \frac{1}{|\mathcal{S}_j|} \sum_{p \in \mathcal{S}_j} \frac{de_{\mathcal{D}_{j-1}}(p)}{dW(p')}.$$

If more than c_j local maxima are found, the best c_j ones are selected as candidates; if less are found, all of them are included. A larger number of candidates allows more points to be considered, but results in a harder accuracy optimisation problem.

Evaluation tolerances. Let $\mathcal{D}_j = \{(p_i^j, \tau_i^j) \mid i = 1, \dots, s_j\}$ be the set of training points at step j . By the selection of candidate points, $s_j \geq s_{j-1}$ and $p_i^j = p_i^{j-1}$ for $i = 1, \dots, s_{j-1}$ hold.

Optimal tolerances τ_i^j are given by the solution of (10) as a function of the tolerances. In order to be able to solve the problem, we ignore the shifts in the mean \bar{y} as they cannot be predicted before evaluating the model. Consequently,

we only consider the impact of evaluation tolerances on the predictive variance and, for evaluation tolerances $\tau^j = (\tau_1^j, \dots, \tau_{s_j}^j)$, write $E(\tau^j)$. As already spent computational budget cannot be recovered by forgetting previously acquired information, we impose the constraint $\tau_i^j \leq \tau_i^{j-1}$ for $i = 1, \dots, s_{j-1}$.

This results in the problem

$$\min_{\tau^j \in \mathcal{T}_j} E(\tau^j) \quad \text{subject to} \quad W_{\tau^j | \mathcal{D}_{j-1}} \leq \Delta W_j, \quad (12)$$

where the set of admissible tolerances is

$$\mathcal{T}_j = \{(\tau_1, \dots, \tau_{s_j}) \in (\mathbb{R}^+ \cup \{+\infty\})^{s_j} \mid \tau_i \leq \tau_i^{j-1} \text{ for } i \leq s_{j-1}\}.$$

If after optimization $\tau_i^j = +\infty$ holds for some $i > s_{j-1}$, p_i^j is excluded from the training set.

Before we can numerically solve the problem, we need to notice that computational costs are not available before the evaluation is performed, such that we need to resort to a priori work models. Following [13, 18], we make use of established a priori asymptotic estimates for finite elements of degree r in space dimension l and an optimal solver such as multigrid, and define

$$W(\tau) = \tau^{-l/r}. \quad (13)$$

This estimate is asymptotic for $\tau \rightarrow 0$. Consequently, despite being inaccurate for low-accuracy evaluations, it is usually accurate for the expensive high-accuracy ones.

Problem (12) is solved by multistart gradient descent with projection and backtracking linesearch. The integral in E is approximated again by Monte Carlo integration on the samples \mathcal{S}_j , resulting in a numerical objective

$$E(\tau^j) \approx \frac{1}{|\mathcal{S}_j|} \sum_{p \in \mathcal{S}_j} e_{\tau^j}(p).$$

To implement gradient descent with projection, we adopt the coordinate change

$$\tau^j = (\tau_1, \dots, \tau_{s_j}) \mapsto \left(\tau_1^{-l/r}, \dots, \tau_{s_j}^{-l/r} \right) = W^j,$$

such that the constraint in (12) becomes linear, transforming the set of admissible tolerances \mathcal{T}^j into a simplex and enabling efficient projection.

6 Solution of the inverse problem

The previous sections established the inverse problem (4) and the sequential approach (10) to surrogate model training. Similar to [17], we combine them to an interleaved strategy given as pseudocode in Alg. 1.

Both the global error quantity (8) and the utility function (11) require integration with respect to the posterior $\pi(p \mid y^m)$. We perform the integration through an MCMC sampling of the posterior, which is at the same time the ultimate goal of the inversion.

We start with an empty sample chain $\mathcal{S}_0 = \emptyset$. At iteration j , we draw a number n_j of samples from $\pi(p \mid y^m, \mathcal{D}_{j-1})$, append them to \mathcal{S}_{j-1} , and remove

the oldest $h_j < n_j$ elements of the chain, as they have been drawn from a less accurate posterior approximation. This results in the sample chain \mathcal{S}_j , which is used to evaluate the integrals involved in the training problem (10) at step j .

As the sample size $|\mathcal{S}_j|$ may be too large for an efficient evaluation of the integrals in (8) and (11), we use a sufficiently large randomly extracted subset of \mathcal{S}_j instead of the whole chain for Monte Carlo integration.

When the computational budget is exhausted, the training of the surrogate model terminates. A last round of samples is added to the chain, obtaining the final set of samples from the posterior.

Algorithm 1 Surrogate-based Bayesian inversion

Require: \mathcal{D}_0 initial design, W budget

- 1: $\mathcal{S}_0 \leftarrow \emptyset$
 - 2: $W_{\mathcal{D}} \leftarrow 0$
 - 3: $j \leftarrow 1$
 - 4: **while** $W_{\mathcal{D}} \leq W$ **do**
 - 5: **decide:** n_j samples to draw, h_j samples to remove
 - 6: remove h_j samples from \mathcal{S}_{j-1}
 - 7: draw n_j samples \mathcal{S} from $\pi(p \mid y^m, \mathcal{D})$
 - 8: $\mathcal{S}_j \leftarrow \mathcal{S}_{j-1} \cup \mathcal{S}$
 - 9: **decide:** ΔW_j iteration budget, c_j number of candidates
 - 10: obtain c_j candidates
 - 11: optimize accuracies τ^j , update \mathcal{D}
 - 12: evaluate forward model for decreased tolerances
 - 13: $W_{\mathcal{D}} \leftarrow W_{\mathcal{D}} + \Delta W_j$
 - 14: $j \leftarrow j + 1$
 - 15: **end while**
 - 16: draw n_j samples \mathcal{S} from $\pi(p \mid y^m, \mathcal{D})$
 - 17: $\mathcal{S}_j \leftarrow \mathcal{S}_{j-1} \cup \mathcal{S}$
-

7 Numerical experiments

We present two illustrative experiments based on a Python implementation of Alg. 1, where GPR is implemented with PyTorch. We adopt a separable kernel with diagonal output structure and a Gaussian kernel as base [1]. The hyperparameters are tuned by marginal likelihood maximisation using PyTorch’s Adam optimiser, with the kernel’s correlation length scale constrained to $[0, 0.15]$.

As a benchmark, the results are compared with a non-adaptive space filling approach, Latin Hypercube Sampling, and the position-adaptive-only training strategy given by candidate point selection according to (11), i.e. all candidates are accepted and evaluated with a fixed accuracy. For comparing the approaches, the approximation errors (6) are computed numerically with MCMC sampling utilising the true forward model. The implementation used for these examples is available at Zenodo¹.

¹<https://zenodo.org/doi/10.5281/zenodo.11066159>

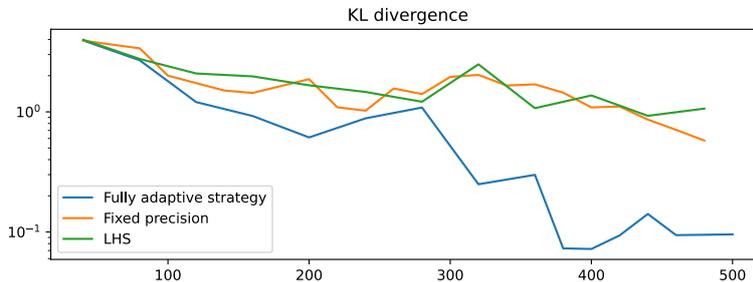


Figure 2: Kullback-Leibler divergence of surrogated posterior and true posterior for different training designs over the computational work spent in the 1D example.

7.1 1D analytical experiment

The first experiment is performed on a one-dimensional parameter space, with $m = 2$ measurements. We consider an analytical forward model $y :]0, 1[\rightarrow \mathbb{R}^2$ given by

$$y(p) = \left[\frac{1}{2}p + \frac{1}{2}p^2 \exp\left(\frac{1}{3}\sin(12p - i)\right) \right]_{i=0,1}.$$

This mimics the evaluation of a FE model on a 2D domain with quadratic elements, i.e. $l/r = 1$. The discretization error is simulated via a zero mean Gaussian noise and the measurement likelihood is $\Sigma_l = 10^{-4} \text{diag}(\frac{16}{9}, \frac{4}{9})$.

A budget of 500 is considered: at each iteration two candidate points are considered and a budget of 20 is assigned to each point. With the work model (13), this results in a default tolerance of 0.05 per point in the non-adaptive strategies and a total of 12 iterations.

The number n_j of new samples added into \mathcal{S}_j is gradually increased from 200 samples at the first iteration to 2000 in the last, according to $n_j = 200 + 1800 \left(\frac{j}{12}\right)^2$. Similarly, the number of discarded samples ranges from 200 to 1000, with $h_1 = 0$ as in the first iteration the chain is empty, and $h_j = 200 + 800 \left(\frac{j}{12}\right)^2$ for $j > 1$.

The obtained accuracies in terms of the Kullback-Leibler divergence between true posterior $\pi(p | y^m)$ and surrogated posterior $\pi(p | y^m, \mathcal{D})$ are shown in Fig. 2. Optimizing evaluation tolerances provides a significant performance improvement over both other strategies.

7.2 2D analytical experiment

The second experiment considers a parameter space of two dimensions and $m = 3$ measurements. The forward model $y :]-0.5, 0.5[^2 \rightarrow \mathbb{R}^3$ is again analytical,

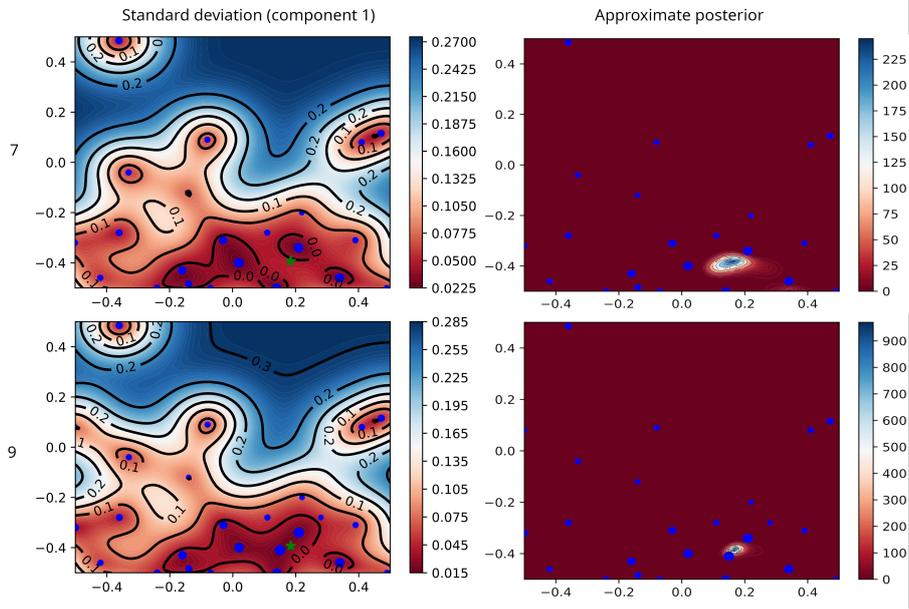


Figure 3: Reduction of surrogate standard deviation of the y_1 , i.e. $k = 0$, component (left) and change of posterior distribution (right) between iterations 7 and 9. The computational work for each point is represented by its size. New points are added and some of the old points are refined. The true parameter used for creating the artificial measurements y^m is denoted by a green star.

given by

$$y(p) = \left[\sin(10k)(p_1 - p_2) \exp\left(\frac{1}{3}\sin(8p_2)\right) + \cos(10k)(p_1 + p_2) \exp\left(\frac{1}{3}\sin(8p_1)\right) \right]_{k \in \{0,2,3\}}.$$

The underlying model is assumed to be a quadratic FE scheme on a 3D domain, i.e. $l/r = 1.5$. The discretization error is again simulated via zero mean Gaussian noise and the measurement likelihood is $\Sigma_l = 10^{-4}\text{diag}(1, 1, 4)$.

A working budget of 3600 is considered: at each iteration, 3 candidate points are considered and a fixed budget of 100 corresponding to a fixed tolerance $\tau = 0.046$ is assigned to each point in the non-adaptive strategies for a total of 12 iterations.

The number n_j of new samples added into \mathcal{S}_j is gradually increased from 200 samples at the first iteration to 4000 in the last, according to $n_j = 200 + \lfloor 26.4j^2 \rfloor$. Similarly, the number of discarded samples ranges from 200 to 2000, with $h_1 = 0$ as in the first iteration the chain is empty, and then $h_j = 200 + \lfloor 12.5j^2 \rfloor$ for $j > 1$. The error reduction by adding new points and decreasing tolerances is illustrated in Fig. 3 for a single iteration. The performance in terms of the Kullback-Leibler divergence between true and surrogated posteriors over computational work is shown in Fig. 4. Again, a substantial performance improvement is achieved by

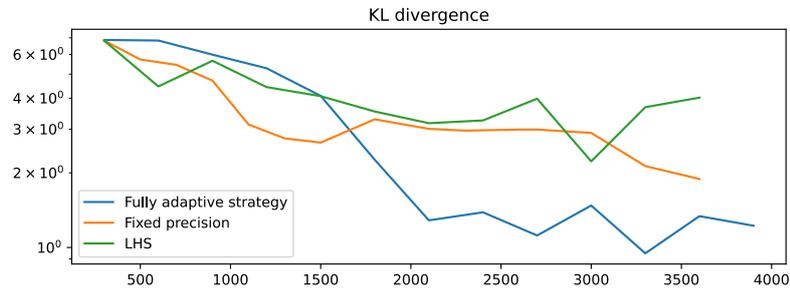


Figure 4: Kullback-Leibler divergence of surrogate posterior and true posterior for different training designs over the computational work spent in the 2D example.

optimizing evaluation tolerances in addition to the evaluation positions.

Conclusions

When learning GPR surrogate models with numerically simulated training data as a replacement for the true forward model in posterior sampling, significant reductions of computational effort can be achieved with adaptive approaches. With numerical forward models that allow exploiting accuracy-work trade-offs, such as finite element simulations, the goal-oriented adaptive selection of simulation tolerances appears to be particularly effective.

References

- [1] M.A. Álvarez, L. Rosasco, and N.D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.
- [2] T. Bai, A.L. Teckentrup, and K.C. Zygalakis. Gaussian processes for Bayesian inverse problems associated with linear partial differential equations. Technical report, arXiv:2307.08343, 2023.
- [3] K. Crombecq, E. Laermans, and T. Dhaene. Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. *European Journal of Operational Research*, 214:683–696, 2011.
- [4] A. Giunta, S. Wojtkiewicz, and M. Eldred. Overview of modern design of experiments methods for computational simulations (invited). In *41st Aerospace Sciences Meeting and Exhibit, AIAA 2003-649*, pages 1–17, 2003.
- [5] P.J. Green, K. Łatuszyński, M. Pereyra, and C.P. Robert. Bayesian computation: a summary of the current state, and samples backwards and forwards. *Stat. Comput.*, 25:835–862, 2015.
- [6] V. Joseph and Y. Hung. Orthogonal-maximin latin hypercube designs. *Statistica Sinica*, 18:171–186, 2008.

- [7] M. Järvenpää, M. U. Gutmann, A. Vehtari, and P. Marttine. Parallel Gaussian process surrogate Bayesian inference with noisy likelihood evaluations. *Bayesian Analysis*, 16, pp. 147–178., 2021.
- [8] R. Lehmensiek, P. Meyer, and M. Müller. Adaptive sampling applied to multivariate, multiple output rational interpolation models with application to microwave circuits. *International Journal of RF and Microwave Computer-Aided Engineering*, 12(4):332–340, 2002.
- [9] J. Nitzler, J. Biehler, N. Fehn, P.-S. Koutsourelakis, and A. Wall. A generalized probabilistic learning approach for multi-fidelity uncertainty quantification in complex physical simulations. *Comp. Meth. Appl. Mech. Eng.*, 400:115600, 2022.
- [10] N. Queipo, R. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1–28, 2005.
- [11] C. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [12] G. Sagnol, H.-C. Hege, and M. Weiser. Using sparse kernels to design computer experiments with tunable precision. In *Proceedings of COMPSTAT 2016*, pages 397–408, 2016.
- [13] P. Semler and M. Weiser. Adaptive Gaussian process regression for efficient building of surrogate models in inverse problems. *Inverse Problems*, 39:125003, 2023.
- [14] P. Semler and M. Weiser. Adaptive gradient enhanced gaussian process surrogates for inverse problems. In *Proceedings of the MATH+ Thematic Einstein Semester 2023*, 2024 (submitted).
- [15] M. Sinsbeck and W. Nowak. Sequential Design of Computer Experiments for the Solution of Bayesian Inverse Problems. *SIAM/ASA Journal on Uncertainty Quantification*, 5:1, 640-664., 2017.
- [16] M. Sugiyama. Active learning in approximately linear regression based on conditional expectation of generalization error. *Journal of Machine Learning Research*, 7:141–166, 2006.
- [17] Z. Wang and M. Broccardo. A novel active learning-based Gaussian process metamodelling strategy for estimating the full probability distribution in forward UQ analysis. *Struct. Safety*, 84:101937, 2020.
- [18] M. Weiser and S. Ghosh. Theoretically optimal inexact spectral deferred correction methods. *Comm. Appl. Math. Comp. Sci.*, 13(1):53–86, 2018.