Proceedings of ALGORITMY 2024
pp. 214–224

ADAPTIVE GAUSSIAN PROCESS REGRESSION FOR BAYESIAN INVERSE PROBLEMS*

PAOLO VILLANI†, JÖRG F. UNGER‡, AND MARTIN WEISER†

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.

Key words. Gaussian process regression, Bayesian inverse problems, surrogate models, parameter identification, active learning

AMS subject classifications. 60G15, 62F15, 62F35, 65N21

1. Introduction. The inverse problem of inferring the posterior probability of parameters $p \in \Omega \subset \mathbb{R}^d$ 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 [8]. 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 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 [15]. A priori point sets [7, 14] are effectively supplemented by adaptive designs [4, 9, 11, 21] 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 [13], and by an adaptive choice of evaluation tolerances [16, 18, 19] in different settings. Multi-fidelity and multi-level sample allocation and model selection [3, 6, 17] face closely related problems, but focus on computing unbiased linear estimators by sampling.

The contribution of the present work is the extension of previous work on goal oriented adaptive surrogate model construction from a pure offline training for maximum posterior point estimates [18] to a representation of the whole posterior distribution

*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

by MCMC sampling. The quantity of interest is the Kullback-Leibler divergence of true and surrogated posteriors. In contrast to [18], we consider an interleaved posterior sampling and surrogate training approach, and integrate the surrogate model inaccuracy by a marginal likelihood instead of a linearized error propagation analysis.

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 $y : \Omega \to \mathbb{R}^m$ on a bounded domain $\Omega \subset \mathbb{R}^d$, 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. Thus, we can interpret the exact forward model as the limit of increasingly accurate numerical approximations: $y = \lim_{\tau\to 0} y_\tau$. Given a design $\mathcal{D} = (p_i, \tau_i)_{i=1,\dots,s}$ defining evaluation points p_i and tolerances $\tau_i > 0$, the numerical model y_τ provides training data $D(\mathcal{D}) = (p_i, y_i)_{i=1,\dots,s}$ with $y_i = y_{\tau_i}(p_i) \approx y(p_i)$ of accuracy τ_i.

Given training data D, we are interested in a prediction of $y_{s+1} \approx y(p_{s+1})$ for any $p_{s+1} \in \Omega$. To perform GPR, we assume y to be a realization of a Gaussian process $\mathcal{G}$ with prior mean $\mu_0 : \Omega \to \mathbb{R}^m$ and covariance kernel $k : \Omega \times \Omega \to \mathbb{R}^{m\times m}$. We assume μ_0 to be constant and equal to zero, while k will be specified later.

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 $T = \operatorname{diag}(\tau_1 I, \dots, \tau_s I, \infty I)$. 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 $\bar{y} : \Omega \to \mathbb{R}^m$ and covariance $\Gamma : \Omega \to \mathbb{R}^{m\times m}$ on the whole parameter space. We refer to [15, 18] for a more detailed exposition.

A surrogate model $y_D \approx y$ can be defined either as the deterministic mean $y_D^d = \bar{y}$, or as a stochastic process taking the covariance Γ into account, i.e. $y_D^s \sim \mathcal{N}(\bar{y}, \Gamma)$. Note that the latter case is not identical to the marginalized Gaussian process, since spatial correlations are partially neglected for reasons of computational efficiency.

3. Bayesian surrogate-based parameter identification. We assume measurements y^m to be random variables generated by a linear additive Gaussian noise model

$$y^m = y(p) + \eta \tag{3.1}$$

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

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

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 replace y by a GPR surrogate y_D based on some training data D. We postpone the question of how to build training designs to the next section. For simplicity, we consider a surrogate with independent output components, i.e. diagonal covariance $\Gamma(p)$.

To evaluate the likelihood, we could substitute the forward model y with the mean estimate $y_D^d = \bar{y}$, obtaining

$$\pi_D^d(y^m \mid p, D) = (2\pi)^{-m/2} \det(\Sigma_l)^{-1/2} \exp\Big(-\frac{1}{2} \|y^m - \bar{y}(p)\|^2_{\Sigma_l^{-1}} \Big). \tag{3.2}$$

This, from a decision-theoretic point of view, corresponds to the minimization of the L^1 loss [10], but ignores the uncertainty estimate given by the predictive variance Γ. Substituting y by the stochastic surrogate y_D^s, which corresponds to marginalizing over GP realizations, results in a different conditional distribution of the measurements $y^m \mid p, D \sim \mathcal{N}(\bar{y}(p), \Sigma_l + \Gamma(p))$ and in a marginal likelihood

$$\pi_D^s(y^m \mid p, D) = (2\pi)^{-m/2} \det\left(\Sigma_l + \Gamma(p)\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \|y^m - \bar{y}(p)\|^2_{(\Sigma_l + \Gamma(p))^{-1}} \right), \tag{3.3}$$

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 π_D^s is closely related to the L^2 loss [10, 20]. Including the GP variance into the likelihood can be important for avoiding overconfident yet wrong posterior approximations by surrogated forward models, see Fig. 3.1 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

$$\pi(p \mid y^m) = \frac{\pi(p)\ \pi(y^m \mid p)}{\pi(y^m)}, \tag{3.4}$$

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

$$\pi(p \mid y^m, D) = \frac{\pi(p)\ \pi_D(y^m \mid p, D)}{\pi(y^m \mid D)}, \tag{3.5}$$

corresponding to the surrogated likelihood with π_D being either π_D^d or π_D^s as given in (3.2) and (3.3), respectively.

In both cases, the normalizing constant $\pi(y^m)$ or $\pi(y^m \mid D)$, 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 [20], we do not aim at building a surrogate which is globally accurate on the whole parameter space Ω, but at finding a design $\mathcal{D}$ for evaluating training data D such that the approximate posterior is accurate, i.e. $\pi(p \mid y^m) \approx \pi(p \mid y^m, D)$. Repeatedly selecting training points randomly sampled from $\pi(p \mid y^m, D)$, updating y_D, and then iterating is sufficient for convergence of $\pi(p \mid y^m, D)$ to $\pi(p \mid y^m)$ 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 D.

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

$$\begin{aligned} D_{\mathrm{KL}}\left(\pi(\cdot \mid y^m) \mid \pi(\cdot \mid y^m, D)\right) &= \mathbb{E}_{\pi(p|y^m)} \left[\log \frac{\pi(p \mid y^m)}{\pi(p \mid y^m, D)} \right] \\ &= \int_\Omega \pi(p \mid y^m) \log \frac{\pi(p \mid y^m)}{\pi(p \mid y^m, D)}\ dp. \end{aligned} \tag{4.1}$$

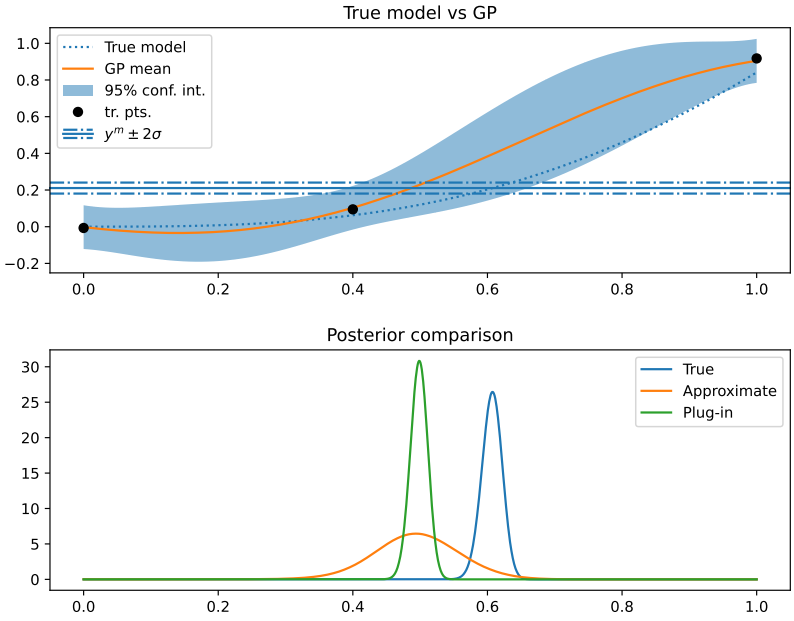

FIG. 3.1. *Impact of the plug-in likelihood* (3.2) *and marginal likelihood* (3.3) *on the posterior for an illustrative inverse problem problem with forward model* $y(p) = p^2 sin(p)$ *and uniform prior on parameter space* $[0, 1]$. *The marginal likelihood* (3.3) *is wider due to including the GP variance, and avoids overconfident posteriors.*

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 π_D^s from (3.3) and the posteriors (3.4) and (3.5), their logarithmic ratio can be written as

$$\log \frac{\pi(p \mid y^m)}{\pi(p \mid y^m, D)} = \log \frac{\pi(y^m \mid p)}{\pi_D^s(y^m \mid p, D)} - \log \frac{\pi(y^m)}{\pi(y^m \mid D)}.$$

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

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

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

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

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

we obtain

$$-\|y(p) - y^m\|^2_{\Sigma_l^{-1}} + \|\bar{y}(p) - y^m\|^2_{(\Sigma_l + \Gamma(p))^{-1}} \lesssim -\operatorname{tr}\left(\Sigma_l^{-1}\Gamma(p)\right) + 2\,|\bar{y}(p) - y^m|^{\mathsf{T}}\, v,$$

where the above absolute value and square root are to be taken element-wise and $\lesssim$ means "approximately less than or equal". We therefore define the local error quantity

$$\begin{aligned} e_D(p) &:= \frac{1}{2}\left(\log\det(I + \Sigma_l^{-1}\Gamma(p)) - \operatorname{tr}\left(\Sigma_l^{-1}\Gamma(p)\right) + 2\,|\bar{y}(p) - y^m|^{\mathsf{T}}\, v\right) \\ &\gtrsim \log\frac{\pi(y^m \mid p)}{\pi_D^s(y^m \mid p, D)} \end{aligned} \tag{4.2}$$

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

By optimistically assuming that the normalization factors are similar independent of the training data D, and thus $\log\frac{\pi(y^m)}{\pi(y^m|D)} \approx 0$, we substitute (4.2) into (4.1) and obtain the global error quantity

$$E(D) = \int_\Omega e_D(p)\pi(p \mid y^m)\; dp. \tag{4.3}$$

To create a nearly optimal surrogate model, we aim at evaluating training data $D(\mathcal{D})$ minimizing $E(D)$ under a computational work constraint, by prescribing a design $\mathcal{D}$, i.e. selecting evaluation points and tolerances. By denoting the computational work needed to realize $\mathcal{D}$ by $W(\mathcal{D})$, for a given budget W we aim at solving the optimization problem

$$\min_{\mathcal{D}} E(D(\mathcal{D})) \;\text{ subject to }\; W(\mathcal{D}) \le W. \tag{4.4}$$

We subsequently write $E(\mathcal{D})$ instead of $E(D(\mathcal{D}))$.

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 (4.4) is not needed – an approximate solution will do, even if it yields a slightly less efficient design. We follow [18, 19] 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 $j = 1, \ldots, J$, aim at solving

$$\min_{\mathcal{D}_j \le \mathcal{D}_{j-1}} E(\mathcal{D}_j) \quad \text{s.t.} \quad W(\mathcal{D}_j \mid \mathcal{D}_{j-1}) \le \Delta W_j. \tag{5.1}$$

We write $\mathcal{D} \le \mathcal{D}_{j-1}$ 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 $W(\mathcal{D} \mid \mathcal{D}_{j-1}) = W(\mathcal{D}) - W(\mathcal{D}_{j-1})$ 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 optimization of the evaluation tolerances. In the latter, we also decide about the actual inclusion of the new points in the training set.

We note that structurally similar optimization problems need to be solved in multi-fidelity and multi-level Monte Carlo methods for computing unbiased linear

estimators by sampling correlated approximate models of different accuracy [3, 17] and in relaxation approaches for the design of experiments [12]. Transferring the solution methods developed for these settings to the problem considered here could be very helpful, though a direct translation is challenging due to the nonlinearity of the work model W, see also [18].

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' [19]. This is given by

$$\begin{aligned}\frac{dE(\mathcal{D})}{dW(p')} &= \int_\Omega \frac{de_D(p)}{dW(p')}\pi(p \mid y^m)\,dp \\ &= \int_\Omega \frac{de_D(p)}{d\Gamma(p)}\frac{d\Gamma(p)}{d\tau(p')}\bigg|_{\tau=\tau'}\frac{d\tau(p')}{dW(p')}\bigg|_{\tau=\tau'}\pi(p \mid y^m)\;dp, \end{aligned} \tag{5.2}$$

where the linearization tolerance τ' is the current surrogate model standard deviation at point p'. We adopt (5.2) as a utility function and select local minimizers of $\frac{dE(\mathcal{D}_{j-1})}{dW}$ as next candidate points.

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

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

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 optimization problem.

Evaluation tolerances. Let $\mathcal{D}_j = \left\{(p_i^j, \tau_i^j) \mid i = 1, \dots, s_j\right\}$ be the training design 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 (5.1) 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

$$\min_{\tau^j\in\mathcal{T}_j} E(\tau^j) \quad \text{subject to} \quad W_{\tau^j|\mathcal{D}_{j-1}} \leq \Delta W_j, \tag{5.3}$$

with $\mathcal{T}_j = \{(\tau_1, \dots, \tau_{s_j}) \in (\mathbb{R}^+ \cup \{+\infty\})^{s_j} \mid \tau_i \leq \tau_i^{j-1} \text{ for } i \leq s_{j-1}\}$ being the set of admissible tolerances. 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 [18, 23], 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

$$W(\tau) = \tau^{-l/r}. \tag{5.4}$$

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

Problem (5.3) 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 (5.3) 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 (3.4) and the sequential approach (5.1). Similar to [22], we combine them to an interleaved strategy given as pseudocode in Alg. 1.

Both the global error quantity (4.3) and the utility function (5.2) 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 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 form $\pi(p \mid y^m, D_{j-1})$, 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 set $\mathcal{S}_j$, which is the current best available approximation of the posterior and is used to evaluate the integrals involved in the training problem (5.1) at step j.

The decisions about number of samples, number of candidates and budget fractioning are to be made according to the characteristics of the problem: for instance, in a problem with extremely expensive model evaluations, the costs of sampling are irrelevant and one can discard and redraw a full set of samples at each iteration; in other settings it will be more convenient to exploit the interleaved approach and keep part of the samples from previous iterations, as we do in the experimental section.

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.

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 Gaussian kernel with diagonal output structure [1]. The hyperparameters are tuned by marginal likelihood maximization using PyTorch's Adam optimizer, 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 (5.2), i.e. all candidates are accepted and

Algorithm 1 Surrogate-based Bayesian inversion

Require: $\mathcal{D}_0$ initial design, W budget
$\mathcal{S}_0 \leftarrow \emptyset$
$W_{\mathcal{D}} \leftarrow 0$
$j \leftarrow 1$
while $W_{\mathcal{D}} \leq W$ **do**
  decide: n_j samples to draw, h_j samples to remove
  remove h_j samples from $\mathcal{S}_{j-1}$
  draw n_j samples $\mathcal{S}$ from $\pi(p \mid y^m, D)$
  $\mathcal{S}_j \leftarrow \mathcal{S}_{j-1} \cup \mathcal{S}$
  decide: ΔW_j iteration budget, c_j number of candidates
  obtain c_j candidates by maximizing (5.2)
  optimize accuracies τ^j by solving (5.3), update $\mathcal{D}$ and D
  evaluate forward model for decreased tolerances
  $W_{\mathcal{D}} \leftarrow W_{\mathcal{D}} + \Delta W_j$
  $j \leftarrow j + 1$
end while
draw n_j samples $\mathcal{S}$ from $\pi(p \mid y^m, \mathcal{D})$
$\mathcal{S}_j \leftarrow \mathcal{S}_{j-1} \cup \mathcal{S}$

evaluated with a fixed accuracy. For comparing the approaches, the approximation errors (4.1) are computed numerically with MCMC sampling through the emcee Ensemble sampler [5] utilizing the true forward model. The implementation used for these examples is available at Zenodo[1].

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 $y :]0, 1[\to \mathbb{R}^2$ 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 $\Sigma_l = 10^{-4}\mathrm{diag}(\frac{16}{9}, \frac{4}{9})$. 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 (5.4), 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 $\pi(p \mid y^m)$ and surrogated posterior $\pi(p \mid y^m, D)$ are shown in Fig. 7.1. Optimizing evaluation tolerances provides a significant performance improvement over both other strategies.

[1] https://zenodo.org/doi/10.5281/zenodo.11066159

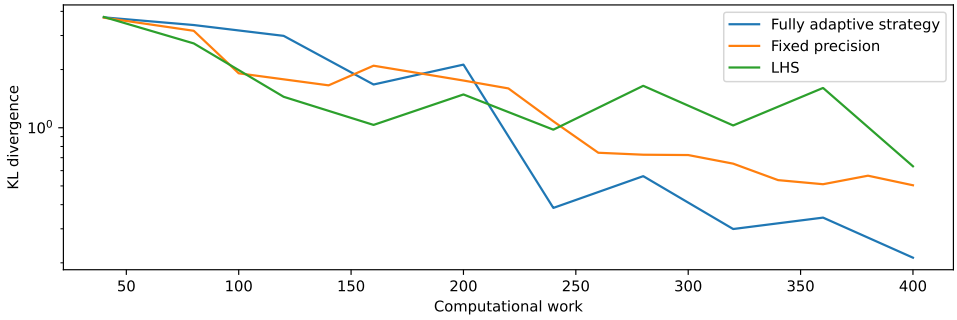

FIG. 7.1. *Kullback-Leibler divergence of surrogated posterior and true posterior for different training designs over the computational work spent in the 1D example.*

7.2. 2D analytical experiment. The second experiment considers a parameter space of two dimensions and $m = 3$ measurements. The forward model $y :]-0.5, 0.5[^2 \to \mathbb{R}^3$ is again analytical, given by

$$y(p) = \left[\sin(10k)(p_1 - p_2) \exp\left(\frac{\sin(8p_2)}{3}\right) + \cos(10k)(p_1 + p_2) \exp\left(\frac{\sin(8p_1)}{3}\right) \right]_{k\in\{0,2,3\}}. \tag{7.1}$$

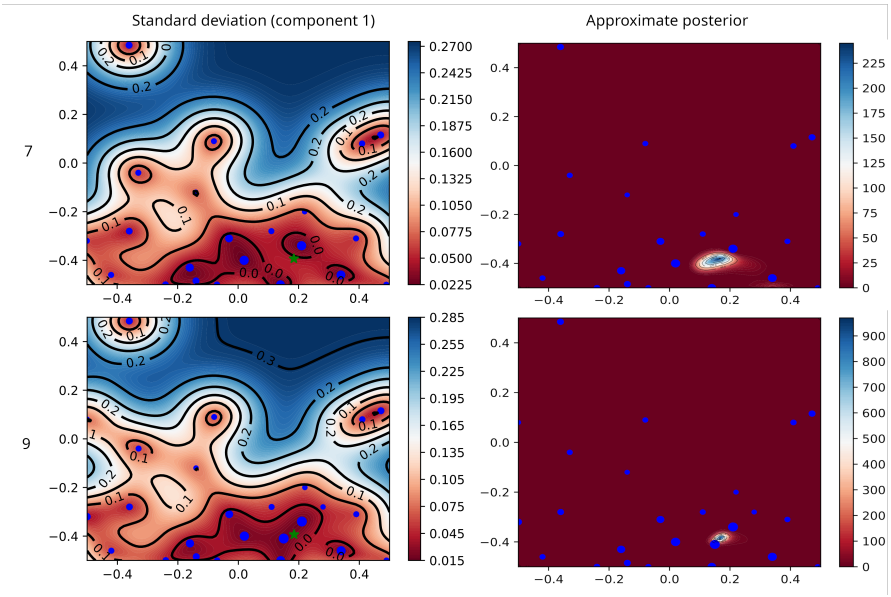

FIG. 7.2. *Reduction of surrogate standard deviation of the y_1, i.e. $k = 0$, component (left) and change of posterior distribution (right) between iterations 7 and 9. The computational work for each point is represented by its size. New points are added and some of the old points are refined. The true parameter used for creating the artificial measurements y^m is indicated by a green star.*

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 $\Sigma_l = 10^{-4}\mathrm{diag}(1,1,4)$. 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 of new samples added into $\mathcal{S}_j$ is gradually increased according to $n_j = 200 + \lfloor 26.4j^2 \rfloor$. Similarly, the number of discarded samples is $h_j = 200 + \lfloor 12.5j^2 \rfloor$ for $j > 1$. The error reduction by adding new points and decreasing tolerances is illustrated in Fig. 7.2 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. 7.3. Again, a substantial performance improvement is achieved by 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.

The sequential extension of the design $\mathcal{D}$ implicitly provides a hierarchy of GPR surrogate models of different accuracy and evaluation complexity, which could also be an interesting building block for multi-fidelity Monte Carlo methods. In that use case, however, the quantity of interest defining the error model and acquisition function would need to be defined differently.

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] M. Croci, K.E. Willcox, and S.J. Wright. Multi-output multilevel best linear unbiased estimators via semidefinite programming. *Comput. Meth. Appl. Mech. Eng.*, 413:116130, 2023.

[4] 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.

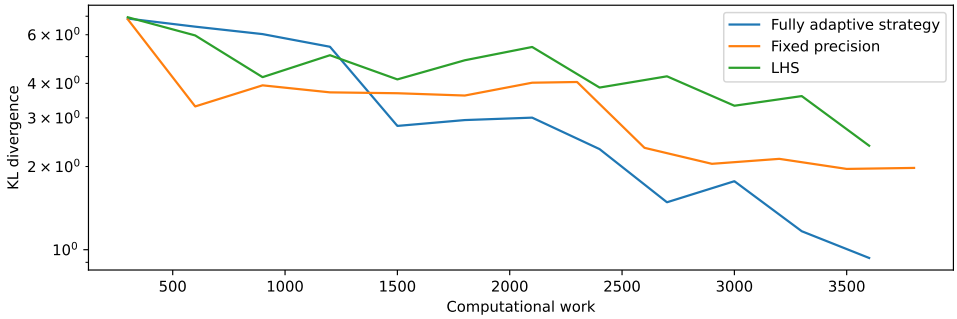

FIG. 7.3. *Kullback-Leibler divergence of surrogated posterior and true posterior for different training designs over the computational work spent in the 2D example.*

[5] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman. emcee: The MCMC hammer. *PASP*, 125:306–312, 2013.
[6] M. B. Giles. Multilevel monte carlo methods. *Acta Numerica*, 24:259–328, 2015.
[7] 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.
[8] 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.
[9] V. Joseph and Y. Hung. Orthogonal-maximin latin hypercube designs. *Statistica Sinica*, 18:171–186, 2008.
[10] 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.
[11] 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.
[12] I. Neitzel, K. Pieper, B. Vexler, and D. Walter. A sparse control approach to optimal sensor placement in PDE-constrained parameter estimation problems. *Numer. Math.*, 143(4):943–984, 2019.
[13] 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.
[14] 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.
[15] C. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
[16] 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.
[17] D. Schaden and E. Ullmann. Asymptotic analysis of multilevel best linear unbiased estimators. *SIAM/ASA J. Uncertainty Quantification*, 9(3):953–978, 2021.
[18] P. Semler and M. Weiser. Adaptive Gaussian process regression for efficient building of surrogate models in inverse problems. *Inverse Problems*, 39:125003, 2023.
[19] 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).
[20] 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.
[21] M. Sugiyama. Active learning in approximately linear regression based on conditional expectation of generalization error. *Journal of Machine Learning Research*, 7:141—166, 2006.
[22] 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.
[23] M. Weiser and S. Ghosh. Theoretically optimal inexact spectral deferred correction methods. *Commu. Appl. Math. Comp. Sci.*, 13(1):53–86, 2018.