Journal of Machine Learning Research 20 (2019) 1-36 Submitted 1/19; Revised 7/19; Published 8/19

Gaussian Processes with Linear Operator Inequality Constraints

Christian Agrell CHRISAGR@MATH.UIO.NO
Department of Mathematics
University of Oslo
P.O. Box 1053 Blindern, Oslo N-0316, Norway

Group Technology and Research
DNV GL
P.O. Box 300, 1322 Høvik, Norway

Editor: Andreas Krause

Abstract

This paper presents an approach for constrained Gaussian Process (GP) regression where we assume that a set of linear transformations of the process are bounded. It is motivated by machine learning applications for high-consequence engineering systems, where this kind of information is often made available from phenomenological knowledge. We consider a GP f over functions on $\mathcal{X} \subset \mathbb{R}^n$ taking values in $\mathbb{R}$, where the process $\mathcal{L}f$ is still Gaussian when $\mathcal{L}$ is a linear operator. Our goal is to model f under the constraint that realizations of $\mathcal{L}f$ are confined to a convex set of functions. In particular, we require that $a \leq \mathcal{L}f \leq b$, given two functions a and b where $a < b$ pointwise. This formulation provides a consistent way of encoding multiple linear constraints, such as shape-constraints based on e.g. boundedness, monotonicity or convexity. We adopt the approach of using a sufficiently dense set of virtual observation locations where the constraint is required to hold, and derive the exact posterior for a conjugate likelihood. The results needed for stable numerical implementation are derived, together with an efficient sampling scheme for estimating the posterior process.

Keywords: Gaussian processes, Linear constraints, Virtual observations, Uncertainty Quantification, Computer code emulation

1. Introduction

Gaussian Processes (GPs) are a flexible tool for Bayesian nonparametric function estimation, and widely used for applications that require inference on functions such as regression and classification. A useful property of GPs is that they automatically produce estimates on prediction uncertainty, and it is often possible to encode prior knowledge in a principled manner in the modelling of prior covariance. Some early well-known applications of GPs are within spatial statistics, e.g. meteorology (Thompson, 1956), and in geostatistics (Matheron, 1973) where it is known as *kriging*. More recently, GPs have become a popular choice within probabilistic machine learning (Rasmussen and Williams, 2005; Ghahramani, 2015). Since the GPs can act as interpolators when observations are noiseless, GPs have also become the main approach for uncertainty quantification and analysis involving computer experiments (Sacks et al., 1989; Kennedy and O'Hagan, 2001).

Often, the modeler performing function estimation has prior knowledge, or at least hypotheses, on some properties of the function to be estimated. This is typically related to the function shape with respect to some of the input parameters, such as boundedness, monotonicity or convexity. Various methods have been proposed for imposing these types of constraints on GPs (see Section 4.1 for a short review). For engineering and physics based applications, constraints based on integral operators and partial differential equations are also relevant (Jidling et al., 2017; Särkkä, 2011). What the above constraints have in common is that they are linear operators, and so any combination of such constraints can be written as a single linear operator. For instance, the constraints $a_1(\mathbf{x}) \leq f(\mathbf{x}) \leq b_1(\mathbf{x})$, $\partial f/\partial x_i \leq 0$ and $\partial^2 f/\partial x_j^2 \geq 0$ for some function (or distribution over functions) $f : X \to Y$, can be written as $a(\mathbf{x}) \leq \mathcal{L}f(\mathbf{x}) \leq b(\mathbf{x})$ for $a(\mathbf{x}) = [a_1(\mathbf{x}), -\infty, 0]$, $b(\mathbf{x}) = [b_1(\mathbf{x}), 0, \infty]$ and $\mathcal{L} : Y^X \to (Y^X)^3$ being the linear operator $\mathcal{L}f = [f, \partial f/\partial x_i, \partial^2 f/\partial x_j^2]$.

The motivation for including constraints is usually to improve predictions and to obtain a reduced and more realistic estimate on the uncertainty, the latter having significant impact for risk-based applications. For many real-world systems, information related to constraints in this form is often available from phenomenological knowledge. For engineering systems, this is typically knowledge related to some underlying physical phenomenon. Being able to make use of these constraint in probabilistic modelling is particularly relevant for high-consequence applications, where obtaining realistic uncertainty estimates in subsets of the domain where data is scarce is a challenge. Furthermore, information on whether these types of constraints are likely to hold given a set of observations is also useful for explainability and model falsification. For a broader discussion see (Agrell et al., 2018; Eldevik et al., 2018).

In this paper, we present a model for estimating a function $\mathrm{f} : \mathbb{R}^{n_x} \to \mathbb{R}$ by a constrained GP (CGP) $f|D, a(\mathbf{x}) \leq \mathcal{L}f(\mathbf{x}) \leq b(\mathbf{x})$. Here D is a set of observations of $(\mathbf{x}_j, y_j)$, possibly including additive white noise, and $f \sim \mathcal{GP}(\mu(\mathbf{x}), K(\mathbf{x}, \mathbf{x}'))$ is a GP with mean $\mu(\mathbf{x})$ and covariance function $K(\mathbf{x}, \mathbf{x}')$ that are chosen such that existence of $\mathcal{L}f$ is ensured. Due to the linearity of $\mathcal{L}$, both $\mathcal{L}f|D$ and $f|D, \mathcal{L}f$ remain Gaussian, and our approach is based on modelling $f|D, \mathcal{L}f$ under the constraint $a(\mathbf{x}) \leq \mathcal{L}f(\mathbf{x}) \leq b(\mathbf{x})$. To model the constraint that $a(\mathbf{x}) \leq \mathcal{L}f(\mathbf{x}) \leq b(\mathbf{x})$ for all inputs $\mathbf{x}$, we take the approach of using a finite set of input locations where the constraint is required to hold. That is, we require that $a(\mathbf{x}_v) \leq \mathcal{L}f(\mathbf{x}_v) \leq b(\mathbf{x}_v)$ for a finite set of inputs $\{\mathbf{x}_v\}$ called the set of *virtual observation locations*. With this approach the CGP is not guaranteed to satisfy the constraint on the entire domain, but a finite set of points $\{\mathbf{x}_v\}$ can be found so that the constraint holds globally with sufficiently high probability.

The model presented in this paper is inspired by the research on shape-constrained GPs, in particular (Wang and Berger, 2016; Da Veiga and Marrel, 2012, 2015; Riihimki and Vehtari, 2010; Golchi et al., 2015; Maatouk and Bay, 2017; López-Lopera et al., 2018). We refer to Section 4 for further discussion on these alternatives. In the case where $\mathcal{L} = \partial/\partial x_i$, our approach is most similar to that of Wang and Berger (2016), where the authors make use of a similar sampling scheme for noiseless GP regression applied to computer code emulation. Many of the approaches to constrained GPs, including ours, rely on the constraint to be satisfied at a specified set of virtual locations. The use of virtual constraint observations may seem *ad hoc* at first, as the set of virtual observation locations has to be dense enough to ensure that the constraint holds globally with sufficiently high probability. Inversion

of the covariance matrix of the joint GP may therefore be of concern, both because this scales with the number of observations cubed and because there is typically high serial correlation if there are many virtual observations close together. The general solution is then to restrict the virtual observation set to regions where the probability of occurrence of the constraint is low (Riihimki and Vehtari, 2010; Wang and Berger, 2016). According to Wang and Berger (2016), when they followed this approach in their experiments, they found that only a modest number of virtual observations were typically needed, that these points were usually rather disperse, and the resulting serial correlation was not severe. We draw the same conclusion in our experiments. There is also one benefit with the virtual observation approach, which is that implementation of constraints that only hold on subsets of the domain is straightforward.

For practical use of the model presented in this paper, we also pay special attention to numerical implementation. The computations involving only real observations or only virtual observations are separated, which is convenient when only changes to the constraints are made such as in algorithms for finding a sparse set of virtual observation locations or for testing/validation of constraints. We also provide the algorithms based on Cholesky factorization for stable numerical implementation, and an efficient sampling scheme for estimating the posterior process. These algorithms are based on derivation of the exact posterior of the constrained Gaussian process using a general linear operator, and constitutes the main contribution of this paper.

The paper is structured as follows: In Section 2 we state the results needed on GP regression and GPs under linear transformations. Our main results are given in Section 3, where we introduce the constrained GP (CGP) and present the model for GP regression under linear inequality constraints. In particular, given some training data, we derive the posterior predictive distribution of the CGP evaluated at a finite set of inputs, which is a compound Gaussian with a truncated Gaussian mean (Section 3.1). Section 3.2 presents an algorithm for sampling from the posterior, and parameter estimation is addressed in Section 3.3. Section 3.4 and Section 3.5 are dedicated to optimization of the set of virtual observation locations needed to ensure that the constraint holds with sufficiently high probability. Some relevant alternative approaches from the literature on GP's under linear constraints are discussed in Section 4, followed up by numerical examples considering monotonicity and boundedness constraints. A Python implementation is available at `https://github.com/cagrell/gp_constr`, together with the code used for the examples. We end with some concluding remarks in Section 5.

2. Gaussian Processes and Linear Operators

We are interested in GP regression on functions $\mathrm{f} : \mathbb{R}^{n_x} \to \mathbb{R}$ under the additional inequality constraint $a(\mathbf{x}) \leq \mathcal{L}\mathrm{f}(\mathbf{x}) \leq b(\mathbf{x})$ for some specified functions $a(\mathbf{x})$ and $b(\mathbf{x})$, and the class of linear operators $\{\mathcal{L}|\mathcal{L}\mathrm{f} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_c}\}$. Here n_x and n_c are positive integers, and the subscripts are just used to indicate the relevant underlying space over $\mathbb{R}$. We will make use of the properties of GPs under linear transformations given below.

2.1. Gaussian Process Regression

We consider a Gaussian process $f \sim \mathcal{GP}(\mu(\mathbf{x}), K(\mathbf{x}, \mathbf{x}'))$ given as a prior over functions $\mathrm{f} : \mathbb{R}^{n_x} \to \mathbb{R}$, which is specified by its mean and covariance function

$$\begin{aligned} \mu(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] : \mathbb{R}^{n_x} \to \mathbb{R}, \\ K(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))] : \mathbb{R}^{n_x \times n_x} \to \mathbb{R}. \end{aligned} \tag{1}$$

Let $\mathbf{x}$ denote a vector in $\mathbb{R}^{n_x}$ and X the $N \times n_x$ matrix of N such input vectors. The distribution over the vector $\mathbf{f}$ of N latent values corresponding to X is then multivariate Gaussian with

$$\mathbf{f}|X \sim \mathcal{N}(\mu(X), K(X, X)),$$

where $K(X, X')$ denotes the Gram matrix $K(X, X')_{i,j} = K(\mathbf{x}_i, \mathbf{x}'_j)$ for two matrices of input vectors X and X'. Given a set of observations $Y = [y_1, \ldots, y_N]^T$, and under the assumption that the relationship between the latent function values and observed output is Gaussian, $Y|\mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma^2 I_N)$, the predictive distribution for new observations X^* is still Gaussian with mean and covariance

$$\begin{aligned} \mathbb{E}[\mathbf{f}^*|X^*, X, Y] &= \mu(X^*) + K(X^*, X)[K(X, X) + \sigma^2 I_N]^{-1}(Y - \mu(X)), \\ \mathrm{cov}(\mathbf{f}^*|X^*, X, Y) &= K(X^*, X^*) - K(X^*, X)[K(X, X) + \sigma^2 I_N]^{-1} K(X, X^*). \end{aligned} \tag{2}$$

Here $\mathbf{f}^*|X^*$ is the predictive distribution of $f(X^*)$ and $\mathbf{f}^*|X^*, X, Y$ is the predictive posterior given the data X, Y. For further details see e.g. Rasmussen and Williams (2005).

2.2. Linear Operations on Gaussian Processes

Let $\mathcal{L}$ be a linear operator on realizations of $f \sim \mathcal{GP}(\mu(\mathbf{x}), K(\mathbf{x}, \mathbf{x}'))$. As GPs are closed under linear operators (Rasmussen and Williams, 2005; Papoulis and Pillai, 2002), $\mathcal{L}f$ is still a GP [1]. We will assume that the operator produces functions with range in $\mathbb{R}^{n_c}$, but where the input domain $\mathbb{R}^{n_x}$ is unchanged. That is, the operator produces functions from $\mathbb{R}^{n_x}$ to $\mathbb{R}^{n_c}$. This type of operators on GPs has also been considered by Särkkä (2011) with applications to stochastic partial differential equations. The mean and covariance of $\mathcal{L}f$ are given by applying $\mathcal{L}$ to the mean and covariance of the argument:

$$\begin{aligned} \mathbb{E}[\mathcal{L}f(\mathbf{x})] &= \mathcal{L}\mu(\mathbf{x}) : \mathbb{R}^{n_x} \to \mathbb{R}^{n_c}, \\ \mathrm{cov}(\mathcal{L}f(\mathbf{x}), \mathcal{L}f(\mathbf{x}')) &= \mathcal{L}K(\mathbf{x}, \mathbf{x}')\mathcal{L}^T : \mathbb{R}^{n_x \times n_x} \to \mathbb{R}^{n_c \times n_c}, \end{aligned} \tag{3}$$

1. We assume here that $\mathcal{L}f$ exists. For instance, if $\mathcal{L}$ involves differentiation then the process f must be differentiable. See e.g. (Adler, 1981) for details on proving existence.

and the cross-covariance is given as

$$\begin{aligned}\operatorname{cov}(\mathcal{L}f(\mathbf{x}), f(\mathbf{x}')) &= \mathcal{L}K(\mathbf{x}, \mathbf{x}') : \mathbb{R}^{n_x \times n_x} \to \mathbb{R}^{n_c}, \\ \operatorname{cov}(f(\mathbf{x}), \mathcal{L}f(\mathbf{x}')) &= K(\mathbf{x}, \mathbf{x}')\mathcal{L}^T : \mathbb{R}^{n_x \times n_x} \to \mathbb{R}^{n_c}.\end{aligned} \tag{4}$$

The notation $\mathcal{L}K(\mathbf{x}, \mathbf{x}')$ and $K(\mathbf{x}, \mathbf{x}')\mathcal{L}^T$ is used to indicate when the operator acts on $K(\mathbf{x}, \mathbf{x}')$ as a function of $\mathbf{x}$ and $\mathbf{x}'$ respectively. That is, $\mathcal{L}K(\mathbf{x}, \mathbf{x}') = \mathcal{L}K(\mathbf{x}, \cdot)$ and $K(\mathbf{x}, \mathbf{x}')\mathcal{L} = \mathcal{L}K(\cdot, \mathbf{x}')$. With the transpose operator the latter becomes $K(\mathbf{x}, \mathbf{x}')\mathcal{L}^T = (\mathcal{L}K(\cdot, \mathbf{x}'))^T$. In the following sections we make use of the predictive distribution (2), where observations correspond to the transformed GP under $\mathcal{L}$.

3. Gaussian Processes with Linear Inequality Constraints

Following Section 2.1 and Section 2.2, we let $f \sim \mathcal{GP}(\mu(\mathbf{x}), K(\mathbf{x}, \mathbf{x}'))$ be a GP over real valued functions on $\mathbb{R}^{n_x}$, and $\mathcal{L}$ a linear operator producing functions from $\mathbb{R}^{n_x}$ to $\mathbb{R}^{n_c}$. The matrix X and the vector Y will represent N noise perturbed observations: $y_i = f(\mathbf{x}_i) + \varepsilon_i$ with ε_i i.i.d. $\mathcal{N}(0, \sigma^2)$ for $i = 1, \ldots, N$.

We would like to model the posterior GP conditioned on the observations X, Y, and on the event that $a(\mathbf{x}) \leq \mathcal{L}f(\mathbf{x}) \leq b(\mathbf{x})$ for two functions $a(\mathbf{x}), b(\mathbf{x}) : \mathbb{R}^{n_x} \to (\mathbb{R} \cup \{-\infty, \infty\})^{n_c}$, where $a_i(\mathbf{x}) < b_i(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^{n_x}$ and $i = 1, \ldots, n_c$. To achieve this approximately, we start by assuming that the constraint $a(\mathbf{x}) \leq \mathcal{L}f(\mathbf{x}) \leq b(\mathbf{x})$ only holds at a finite set of inputs $\mathbf{x}_1^v, \ldots, \mathbf{x}_S^v$ that we refer to as *virtual observation locations*. Later, we will consider how to specify the set of virtual observation locations such that the constraint holds for any $\mathbf{x}$ with sufficiently high probability. Furthermore, we will also assume that *virtual observations* of the transformed process, $\mathcal{L}f(\mathbf{x}_i^v)$, comes with additive white noise with variance σ_v^2. We can write this as $a(X^v) \leq \mathcal{L}f(X^v) + \varepsilon^v \leq b(X^v)$, where $X^v = [\mathbf{x}_1^v, \ldots, \mathbf{x}_S^v]^T$ is the matrix containing the virtual observation locations and ε^v is a multivariate Gaussian with diagonal covariance of elements σ_v^2.

We will make use of the following notation: Let $\widetilde{C}(X^v) \in \mathbb{R}^{S \times n_c}$ be the matrix with rows $(\widetilde{C}(X^v))_i = \mathcal{L}f(\mathbf{x}_i^v) + \varepsilon_i^v$ for i.i.d. $\varepsilon_i^v \sim \mathcal{N}(\mathbf{0}, \sigma_v^2 I_{n_c})$, and let $C(X^v)$ denote the event $C(X^v) := \cap_{i=1}^S \{a(\mathbf{x}_i^v) \leq (\widetilde{C}(X^v))_i \leq b(\mathbf{x}_i^v)\}$. $C(X^v)$ thus represents the event that the constraint $a(\mathbf{x}) \leq \mathcal{L}f(\mathbf{x}) + \varepsilon^v \leq b(\mathbf{x})$ is satisfied for all points in X^v, and it is defined through the latent variable $\widetilde{C}(X^v)$.

In summary, the process we will consider is stated as

$$f | X, Y, X^v, C(X^v) := f | f(X) + \varepsilon = Y, a(X^v) \leq \mathcal{L}f(X^v) + \varepsilon^v \leq b(X^v),$$

where f is a Gaussian process, X, Y is the training data and X^v are the locations where the transformed process $\mathcal{L}f + \varepsilon^v$ is bounded. The additive noise ε and ε^v are multivariate Gaussian with diagonal covariance matrices of elements σ^2 and σ_v^2 respectively.

Here we assume that observations of all parts of $\mathcal{L}f$ comes with i.i.d. white noise with variance σ_v^2. The reason for this is mainly for numerical stability, where we in computations will choose a tiny variance to approximate noiseless observations. Similarly, σ^2 may be chosen as a fixed small number for interpolation in the standard GP regression setting. In the following derivations, the results for exact noiseless observations can be obtained by setting the relevant variance to zero.

We also assume that any sub-operator of $\mathcal{L}$ is constrained at the same set of virtual locations X^v. This is mainly for notational convenience, and this assumption will be relaxed in Section 3.5. In the following, we let N_v denote the total number of virtual observation locations. Here $N_v = S \cdot n_c$ for now, whereas we will later consider $N_v = \sum_{i=1}^{n_c} S_i$ where the i-th sub-operator is associated with S_i virtual observation locations.

3.1. Posterior Predictive Distribution

Our goal is to obtain the posterior predictive distribution $\mathbf{f}^*|X^*, X, Y, X^v, C(X^v)$. That is: the distribution of $\mathbf{f}^* = f(X^*)$ for some new inputs X^*, conditioned on the observed data $Y = f(X) + \varepsilon$ and the constraint $a(X^v) \le \mathcal{L}f(X^v) + \varepsilon^v \le b(X^v)$.

To simplify the notation we write $\mathbf{f}^*|Y, C$, excluding the dependency on inputs X, X^* and X^v (as well as any hyperparameter of the mean and covariance function). The posterior predictive distribution is given by marginalizing over the latent variable $\widetilde{C}$:

$$\begin{aligned} p(\mathbf{f}^*, C|Y) &= p(\mathbf{f}^*|C, Y)p(C|Y), \\ p(\mathbf{f}^*|C, Y) &= \int_{a(X^v)}^{b(X^v)} p(\mathbf{f}^*|\widetilde{C}, Y)p(\widetilde{C}|Y)d\widetilde{C}, \\ p(C|Y) &= \int_{a(X^v)}^{b(X^v)} p(\widetilde{C}|Y)d\widetilde{C}, \end{aligned}$$

where the limits correspond to the hyper-rectangle in $\mathbb{R}^{N_v}$ given by the functions $a(\cdot)$ and $b(\cdot)$ evaluated at each $\mathbf{x}^v \in X^v$. The predictive distribution and the probability $p(C|Y)$ are given in Lemma 1. $p(C|Y)$ is of interest, as it is the probability that the constraint holds at X^v given the data Y.

In the remainder of the paper we will use the shortened notation $\mu^* = \mu(X^*)$, $\mu = \mu(X)$, $\mu^v = \mu(X^v)$ and $K_{X,X'} = K(X, X')$. For vectors with elements in $\mathbb{R}^{n_c}$, such as $\mathcal{L}\mu^v$, we interpret this elementwise. E.g. $\mathcal{L}\mu^v(X^v)$ is given by the column vector $[\mathcal{L}\mu(\mathbf{x}_1^v)_1, \ldots, \mathcal{L}\mu(\mathbf{x}_1^v)_{n_c}, \ldots, \mathcal{L}\mu(\mathbf{x}_S^v)_1, \ldots, \mathcal{L}\mu(\mathbf{x}_S^v)_{n_c}]$.

We start by deriving the posterior predictive distribution $\mathbf{f}^*$ at some new locations X^*. The predictive distribution is represented by a Gaussian, $\mathbf{f}^*|Y, C \sim \mathcal{N}(\mu(\mathbf{C}), \Sigma)$, for some fixed covariance matrix Σ and a mean $\mu(\mathbf{C})$ that depends on the random variable $\mathbf{C} = \widetilde{C}|Y, C$. The variable $\widetilde{C} = \mathcal{L}f(X^v) + \varepsilon^v$ remains Gaussian after conditioning on the observations Y, i.e. $\widetilde{C}|Y \sim \mathcal{N}(\nu_c, \Sigma_c)$ with some expectation ν_c and covariance matrix Σ_c that can be computed using (3, 4). Applying the constraints represented by the event C on the random variable $\widetilde{C}|Y$ just means restricting $\widetilde{C}|Y$ to lie in the hyper-rectangle defined by the bounds $a(X^v)$ and $b(X^v)$. This means that $\mathbf{C} = \widetilde{C}|Y, C$ is a truncated multivariate Gaussian, $\mathbf{C} \sim \mathcal{TN}(\nu_c, \Sigma_c, a(X^v), b(X^v))$. The full derivation of the distribution parameters of $\mathbf{C}$ and $\mathbf{f}^*|Y, C$ are given in Lemma 1 below, whereas Lemma 2 provides an alternative algorithmic representation suitable for numerical implementation.

Lemma 1 *The predictive distribution $\mathbf{f}^*|Y, C$ is a compound Gaussian with truncated Gaussian mean:*

$$\mathbf{f}^*|Y, C \sim \mathcal{N}(\mu^* + A(\boldsymbol{C} - \mathcal{L}\mu^v) + B(Y - \mu), \Sigma), \tag{5}$$

$$\boldsymbol{C} = \widetilde{C}|Y, C \sim \mathcal{TN}(\mathcal{L}\mu^v + A_1(Y-\mu), B_1, a(X^v), b(X^v)), \tag{6}$$

where $\mathcal{TN}(\cdot,\cdot,a,b)$ *is the Gaussian* $\mathcal{N}(\cdot,\cdot)$ *conditioned on the hyper-rectangle* $[a_1,b_1]\times\cdots\times[a_k,b_k]$*, and*

$$\begin{array}{ll} A_1 = (\mathcal{L}K_{X^v,X})(K_{X,X}+\sigma^2 I_N)^{-1}, & B_1 = \mathcal{L}K_{X^v,X^v}\mathcal{L}^T + \sigma_v^2 I_{N_v} - A_1 K_{X,X^v}\mathcal{L}^T, \\ A_2 = K_{X^*,X}(K_{X,X}+\sigma^2 I_N)^{-1}, & B_2 = K_{X^*,X^*} - A_2 K_{X,X^*}, \\ & B_3 = K_{X^*,X^v}\mathcal{L}^T - A_2 K_{X,X^v}\mathcal{L}^T, \end{array}$$
$$A = B_3 B_1^{-1}, \quad B = A_2 - AA_1, \quad \Sigma = B_2 - AB_3^T.$$

Moreover, the probability that the unconstrained version of $\boldsymbol{C}$ *falls within the constraint region,* $p(C|Y)$*, is given by*

$$p(C|Y) = p\left(a(X^v) \le \mathcal{N}(\mathcal{L}\mu^v + A_1(Y-\mu), B_1) \le b(X^v)\right), \tag{7}$$

and the unconstrained predictive distribution is

$$\mathbf{f}^*|Y \sim \mathcal{N}(\mu^* + A_2(Y-\mu), B_2).$$

The derivation in Lemma 1 is based on conditioning the multivariate Gaussian $(\mathbf{f}^*, Y, \widetilde{C})$, and the proof is given in Appendix A. For practical implementation the matrix inversions involved in Lemma 1 may be prone to numerical instability. A numerically stable alternative is given in Lemma 2 below.

In the following lemma, $Chol(K)$ is the lower triangular Cholesky factor of a matrix K. We also let $R = (P \setminus Q)$ denote the solution to the linear system $PR = Q$ for matrices P and Q, which may be efficiently computed when P is triangular using forward or backward substitution.

Lemma 2 *Let* $L = Chol(K_{X,X}+\sigma^2 I_N)$*,* $v_1 = L \setminus K_{X,X^v}\mathcal{L}^T$ *and* $v_2 = L \setminus K_{X,X^*}$*.*

Then the matrices in Lemma 1 can be computed as

$$\begin{array}{ll} A_1 = (L^T \setminus v_1)^T, & B_1 = \mathcal{L}K_{X^v,X^v}\mathcal{L}^T + \sigma_v^2 I_{N_v} - v_1^T v_1, \\ A_2 = (L^T \setminus v_2)^T, & B_2 = K_{X^*,X^*} - v_2^T v_2, \\ & B_3 = K_{X^*,X^v}\mathcal{L}^T - v_2^T v_1. \end{array}$$

Moreover, B_1 *is symmetric and positive definite. By letting* $L_1 = Chol(B_1)$ *and* $v_3 = L_1 \setminus B_3^T$ *we also have*

$$A = (L_1^T \setminus v_3)^T, \quad B = A_2 - AA_1, \quad \Sigma = B_2 - v_3^T v_3.$$

The proof is given in Appendix B. The numerical complexity of the procedures in Lemma 2 is $n^3/6$ for Cholesky factorization of $n \times n$ matrices and $mn^2/2$ for solving triangular systems where the unknown matrix is $n \times m$. In the derivation of Lemma 1 and Lemma 2, the order of operations was chosen such that the first Cholesky factor $L = Chol(K_{X,X}+\sigma^2 I_N)$ only depends on X. This is convenient in the case where the posterior $\mathbf{f}^*|Y, C$ is calculated multiple times for different constraints C or virtual observations X^v, but where the data X, Y remain unchanged.

3.2. Sampling from the Posterior Distribution

In order to sample from the posterior we can first sample from the constraint distribution (6), and then use these samples in the mean of (5) to create the final samples of $\mathbf{f}^*|Y, C$.

To generate k samples of the posterior at M new input locations, $[\mathbf{x}_1^*, \ldots, \mathbf{x}_M^*]^T = X^*$, we use the following procedure

Algorithm 3 *Sampling from the posterior distribution*

1. *Find a matrix Q s.t. $Q^TQ = \Sigma \in \mathbb{R}^{M\times M}$, e.g. by Cholesky or a spectral decomposition.*
2. *Generate $\widetilde{C}_k$, a $N_v \times k$ matrix where each column is a sample of $\widetilde{C}|Y, C$ from the distribution in* (6)*.*
3. *Generate U_k, a $M \times k$ matrix with k samples from the standard normal $\mathcal{N}(\mathbf{0}, I_M)$.*
4. *The $M \times k$ matrix where each column in a sample from $\mathbf{f}^*|Y, C$ is then obtained by*

$$[\mu^* + B(Y - \mu)] \oplus_{col} [A(-\mathcal{L}\mu^v \oplus_{col} \widetilde{C}_k) + QU_k],$$

where $\oplus_{col}$ means that the $M \times 1$ vector on the left hand side is added to each column of the $M \times k$ matrix on the right hand side.

This procedure is based on the well-known method for sampling from multivariate Gaussian distributions, where we have used the property that in the distribution of $\mathbf{f}^*|Y, C$, only the mean depends on samples from the constraint distribution.

The challenging part of this procedure is the second step where samples have to be drawn from a truncated multivariate Gaussian. The simplest approach is by rejection sampling, i.e. generating samples from the normal distribution and rejection those that fall outside the bounds. In order to generate m samples with rejection sampling, the expected number of samples needed is $m/p(C|Y)$, where the acceptance rate is the probability $p(C|Y)$ given in (7). If the acceptance rate is low, then rejection sampling becomes inefficient, and an alternative approach such as Gibbs sampling (Kotecha and Djuric, 1999) is typically used. In our numerical experiments (presented in Section 4.2) we made use of a new method based on simulation via minimax tilting by Botev (2017), developed for high-dimensional exact sampling. Botev (2017) prove strong efficiency properties and demonstrate accurate simulation in dimensions $d \sim 100$ with small acceptance probabilities ($\sim 10^{-100}$), that take about the same time as one cycle of Gibbs sampling. For higher dimensions in the thousands, the method is used to accelerate existing Gibbs samplers by sampling jointly hundreds of highly correlated variables. In our experiments, we experienced that this method worked well in cases where Gibbs sampling was challenging. A detailed comparison with other sampling alternatives for an application similar to ours is also given in (López-Lopera et al., 2018). An important observation in Algorithm 3 is that for inference at a new set of input locations X^*, when the data X, Y and virtual observation locations X^v are unchanged, the samples generated in step *2* can be reused.

3.3. Parameter Estimation

To estimate the parameters of the CGP we make use of the marginal maximum likelihood approach (MLE). We define the marginal likelihood function of the CGP as

$$L(\theta) = p(Y, C|\theta) = p(Y|\theta)p(C|Y, \theta), \tag{8}$$

i.e. as the probability of the data Y and constraint C combined, given the set of parameters represented by θ. We assume that both the mean and covariance function of the GP prior (1) $\mu(\mathbf{x}|\theta)$ and $K(\mathbf{x}, \mathbf{x}'|\theta)$ may depend on θ. The log-likelihood, $l(\theta) = \ln p(Y|\theta) + \ln p(C|Y, \theta)$, is thus given as the sum of the unconstrained log-likelihood, $\ln p(Y|\theta)$, which is optimized in unconstrained MLE, and $\ln p(C|Y, \theta)$, which is the probability that the constraint holds at X^v given in (7).

In (Bachoc et al., 2018) the authors study the asymptotic distribution of the MLE for shape-constrained GPs, and show that for large sample sizes the effect of including the constraint in the MLE is negligible. But for small or moderate sample sizes the constrained MLE is generally more accurate, so taking the constraint into account is beneficial. However, due to the added numerical complexity in optimizing a function that includes the term $\ln p(C|Y, \theta)$, it might not be worthwhile. Efficient parameter estimation using the full likelihood (8) is a topic of future research. In the numerical experiments presented in this paper, we therefore make use of the unconstrained MLE. This also makes it possible to compare models with and without constraints in a more straightforward manner.

3.4. Finding the Virtual Observation Locations

For the constraint to be satisfied locally at any input location in some bounded set $\Omega \subset \mathbb{R}^{n_x}$ with sufficiently high probability, the set of virtual observation locations X^v has to be sufficiently dense. We will specify a target probability $p_{\text{target}} \in [0, 1)$ and find a set X^v, such that when the constraint is satisfied at all virtual locations in X^v, the probability that the constraint is satisfied for any $\mathbf{x}$ in Ω is at least p_{target}. The number of virtual observation locations needed depends on the smoothness properties of the kernel, and for a given kernel it is of interest to find a set X_v that is effective in terms of numerical computation. As we need to sample from a truncated Gaussian involving cross-covariances between all elements in X^v, we would like the set X^v to be small, and also to avoid points in X^v close together that could lead to high serial correlation.

Seeking an optimal set of virtual observation locations has also been discussed in (Wang and Berger, 2016; Golchi et al., 2015; Riihimki and Vehtari, 2010; Da Veiga and Marrel, 2012, 2015), and the intuitive idea is to iteratively place virtual observation locations where the probability that the constraint holds is low. The general approach presented in this section is most similar to that of Wang and Berger (2016). In Section 3.5 we extend this to derive a more efficient method for multiple constraints.

In order to estimate the probability that the constraint holds at some new location $\mathbf{x}^* \in \Omega$, we first derive the posterior distribution of the constraint process.

Lemma 4 *The predictive distribution of the constraint $\mathcal{L}f(\boldsymbol{x}^*)$ for some new input $\boldsymbol{x}^* \in \mathbb{R}^{n_x}$, condition on the data Y is given by*

$$\mathcal{L}f(\boldsymbol{x}^*)|Y \sim \mathcal{N}(\mathcal{L}\mu^* + \widetilde{A}_2(Y - \mu), \widetilde{B}_2), \tag{9}$$

and when $\mathcal{L}f(\boldsymbol{x}^*)$ is conditioned on both the data and virtual constraint observations, X, Y and $X^v, C(X^v)$, the posterior becomes

$$\mathcal{L}f(\boldsymbol{x}^*)|Y, C \sim \mathcal{N}(\mathcal{L}\mu^* + \widetilde{A}(\boldsymbol{C} - \mathcal{L}\mu^v) + \widetilde{B}(Y - \mu), \widetilde{\Sigma}). \tag{10}$$

Here L, v_1, A_1, B_1 and L_1 are defined as in Lemma 2 , $\boldsymbol{C}$ is the distribution in (6) *and*

$$\begin{aligned} \widetilde{v}_2 &= L \setminus K_{X,\boldsymbol{x}^*}\mathcal{L}^T, & \widetilde{B}_2 &= \mathcal{L}K_{\boldsymbol{x}^*,\boldsymbol{x}^*}\mathcal{L}^T - \widetilde{v}_2^T\widetilde{v}_2, \\ \widetilde{A}_2 &= (L^T \setminus \widetilde{v}_2)^T, & \widetilde{B}_3 &= \mathcal{L}K_{\boldsymbol{x}^*,X^v}\mathcal{L}^T - \widetilde{v}_2^T v_1, \\ & & \widetilde{v}_3 &= L_1 \setminus \widetilde{B}_3^T, \\ \widetilde{A} &= (L_1^T \setminus \widetilde{v}_3)^T, & \widetilde{B} &= \widetilde{A}_2 - \widetilde{A}A_1, \quad \widetilde{\Sigma} = \widetilde{B}_2 - \widetilde{v}_3^T\widetilde{v}_3. \end{aligned}$$

The proof is given in Appendix D. The predictive distribution in Lemma 4 was defined for a single input $\mathbf{x}^* \in \mathbb{R}^{n_x}$, and we will make use of the result in this context. But we could just as well consider an input matrix X^* with rows $\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots$, where the only change in Lemma 4 is to replace $\mathbf{x}^*$ with X^*. In this case we also note that the variances, $\mathrm{diag}(\widetilde{\Sigma})$, is more efficiently computed as $\mathrm{diag}(\widetilde{\Sigma}) = \mathrm{diag}(\mathcal{L}K_{X^*,X^*}\mathcal{L}^T) - \mathrm{diag}(\widetilde{v}_2^T\widetilde{v}_2) - \mathrm{diag}(\widetilde{v}_3^T\widetilde{v}_3)$ where we recall that $\mathrm{diag}(v^T v)_i = \sum_j v_{i,j}^2$ for $v^T = [v_{i,j}]$.

Using the posterior distribution of $\mathcal{L}f$ in Lemma 4 we define the constraint probability $p_c : \mathbb{R}^{n_x} \to [0, 1]$ as

$$p_c(\mathbf{x}) = P\left(a(\mathbf{x}) - \nu < \xi(\mathbf{x}, X^v) < b(\mathbf{x}) + \nu\right), \tag{11}$$

where $\xi(\mathbf{x}, X^v) = \mathcal{L}f(\mathbf{x}^*)|Y$ for $X^v = \emptyset$ and $\xi(\mathbf{x}, X^v) = \mathcal{L}f(\mathbf{x}^*)|Y, C$ otherwise. The quantity ν is a non-negative fixed number that is included to ensure that it will be possible to increase p_c using observations with additive noise. When we use virtual observations $\widetilde{C}(\mathbf{x}) = \mathcal{L}f(\mathbf{x}^*) + \varepsilon^v$ that come with noise $\varepsilon^v \sim \mathcal{N}(0, \sigma_v^2)$, we can use $\nu = \max\{\sigma_v\Phi^{-1}(p_{\text{target}}), 0\}$ where $\Phi(\cdot)$ is the normal cumulative distribution function. Note that σ_v, and in this case ν, will be small numbers included mainly for numerical stability. In the numerical examples presented in this paper this noise variance was set to 10^{-6}.

In the case where $X^v = \emptyset$, computation of (11) is straightforward as $\xi(\mathbf{x}, X^v)$ is Gaussian. Otherwise, we will rely on the following estimate of $p_c(\mathbf{x})$:

$$\hat{p}_c(\mathbf{x}) = \frac{1}{m}\sum_{j=1}^{m} P\left(a(\mathbf{x}) - \nu < (\mathcal{L}f(\mathbf{x})|Y, C_j) < b(\mathbf{x}) + \nu\right), \tag{12}$$

where $C_1, \ldots, C_m$ are m samples of $\mathbf{C}$ given in (6).

We outline an algorithm for finding a set of virtual observation locations X^v, such that the probability that the constraint holds locally at any $\mathbf{x} \in \Omega$ is at least p_{target} for some specified set $\Omega \subset \mathbb{R}^{n_x}$ and $p_{\text{target}} \in [0, 1)$. That is, $\min_{\mathbf{x}\in\Omega} p_c(\mathbf{x}) \geq p_{\text{target}}$. The algorithm can be used starting with no initial virtual observation locations, $X^v = \emptyset$, or using some pre-defined set $X^v \neq \emptyset$. The latter may be useful e.g. if the data X, Y is updated, in which case only a few additions to the previous set X^v might be needed.

Algorithm 5 *Finding locations of virtual observations X^v s.t. $\hat{p}_c(\boldsymbol{x}) \geq p_{target}$ for all $\boldsymbol{x} \in \Omega$.*

1. *Compute $L = Chol(K_{X,X} + \sigma^2 I_N)$.*

2. Until convergence do:

(a) If $X^v \neq \emptyset$ compute A_1 and B_1 as defined in Lemma 2, and generate m samples $C_1, \ldots, C_m$ of $\boldsymbol{C}$ given in (6)*.*

(b) If $X^v = \emptyset$ compute $(\boldsymbol{x}^*, p^*) = (\arg\min p_c(\boldsymbol{x}), p_c(\boldsymbol{x}^*))$. Otherwise compute $(\boldsymbol{x}^*, p^*) = (\arg\min \hat{p}_c(\boldsymbol{x}), \hat{p}_c(\boldsymbol{x}^*))$ with $\hat{p}_c$ defined as in (12)*, using the samples generated in step (a).*

(c) Terminate if $p^* \geq p_{target}$, otherwise update $X^v \to X^v \cup \{\boldsymbol{x}^*\}$.

The rate of convergence of Algorithm 5 relies on the probability that the constraint holds initially, $P\left(a(\mathbf{x}) < (\mathcal{L}f(\mathbf{x})|Y) < b(\mathbf{x})\right)$, and for practical application one may monitor p^* as a function of the number of virtual observation locations, $|X^v|$, to find an appropriate stopping criterion.

With the exception of low dimensional input $\mathbf{x}$, the optimization step $\mathbf{x}^* = \arg\min \hat{p}_c(\mathbf{x})$ is in general a hard non-convex optimization problem. But with respect to how $\mathbf{x}^*$ and p^* are used in the algorithm, some simplifications can be justified. First, we note that when computing $\hat{p}_c(\mathbf{x})$ with (12) for multiple $\mathbf{x} = \mathbf{x}_1, \mathbf{x}_2, \ldots$, the samples $C_1, \ldots, C_m$ are reused. It is also not necessary to find the the absolute minimum, as long as a *small enough* value is found in each iteration. Within the global optimization one might therefore decide to stop after the first occurrence of $\hat{p}_c(\mathbf{x})$ less than some threshold value. With this idea one could also search over finite candidate sets $\Omega \subset \mathbb{R}^{n_x}$, using a fixed number of random points in $\mathbb{R}^{n_x}$. This approach might produce a larger set X^v, but where the selection of $\mathbf{x}^*$ is faster in each iteration. Some of the alternative strategies for locating $\mathbf{x}^*$ in Algorithm 5 are studied further in our numerical experiments in Section 4.2.

With the above algorithm we aim to impose constraints on some bounded set $\Omega \subset \mathbb{R}^{n_x}$. Here Ω has to be chosen with respect to both training and test data. For a single boundedness constraint, it might be sufficient that the constraint only holds at the points $\mathbf{x} \in \mathbb{R}^{n_x}$ that will be used for prediction. But if we consider constraints related to monotonicity (see Example 1, Section 4.2), dependency with respect to the latent function's properties at the training locations is lost with this strategy. In the examples we give in this paper we consider a convex set Ω, in particular $\Omega = [0, 1]^{n_x}$, and assume that training data, test data and any input relevant for prediction lies within Ω.

3.5. Separating Virtual Observation Locations for Sub-operators

Let $\mathcal{L}$ be a linear operator defined by the column vector $[\mathcal{F}_1, \ldots, \mathcal{F}_k]$, where each $\mathcal{F}_i$ is a linear operator leaving both the domain and range of its argument unchanged, i.e. $\mathcal{F}_i$ produces functions from $\mathbb{R}^{n_x}$ to $\mathbb{R}$, subjected to an interval constraint $[a_i(\mathbf{x}), b_i(\mathbf{x})]$. Until now we have assumed that the constrain holds at a set of virtual observation locations X^v, which means that $a_i(X^v) \leq \mathcal{F}_i f(X^v) \leq b_i(X^v)$ for all $i = 1, \ldots, k$.

However, it might not be necessary to constrain each of the sub-operators $\mathcal{F}_i$ at the same points $\mathbf{x}^v \in X^v$. Intuitively, constraints with respect to $\mathcal{F}_i$ need only be imposed at locations where $p(\mathcal{F}_i f(\mathbf{x}) \notin [a_i(\mathbf{x}), b_i(\mathbf{x})])$ is large. To accommodate this we let X^v be the concatenation of the matrices $X^{v,1}, \ldots, X^{v,k}$ and define $\mathcal{L}^T f(X^v) = [\mathcal{F}_1^T f(X^{v,1}), \ldots, \mathcal{F}_k^T f(X^{v,1})]^T$. This is equivalent to removing some of the rows in $\mathcal{L}(\cdot)(X^v)$, and all of the results in this paper still apply. In this setting we can improve the algorithm in Section 3.4 for finding the

set of virtual observation locations by considering each sub-operator individually. This is achieved using the estimated partial constraint probabilities, $p_{c,i}(\mathbf{x})$, that we defined as in (11) by considering only the i-th sub-operator. We may then use the estimate

$$\hat{p}_{c,i}(\mathbf{x}) = \frac{1}{m}\sum_{j=1}^{m} P\left(a_i(\mathbf{x}) - \nu < (\mathcal{L}f(\mathbf{x})|Y, C_j)_i < b_i(\mathbf{x}) + \nu\right), \tag{13}$$

where $(\mathcal{L}f(\mathbf{x})|Y, C_j)_i$ is the univariate Normal distribution given by the i-th row of $(\mathcal{L}f(\mathbf{x})|Y, C_j)$, and $C_1, \ldots, C_m$ are m samples of $\mathbf{C}$ given in (6) as before. Algorithm 5 can then be improved by minimizing (13) with respect to both $\mathbf{x}$ and $i = 1, \ldots k$. The details are presented in Appendix C, Algorithm 7.

3.6. Prediction using the Posterior Distribution

For the unconstrained GP in this paper where the likelihood is given by Gaussian white noise, the posterior mean and covariance is sufficient to describe predictions as the posterior remains Gaussian. It is also known that in this case there is a correspondence between the posterior mean of the GP and the optimal estimator in the Reproducing Kernel Hilbert Space (RKHS) associated with the GP (Kimeldorf and Wahba, 1970). This is a Hilbert space of functions defined by the positive semidefinite kernel of the GP. Interestingly, a similar correspondence holds for the constrained case. Maatouk et al. (2016) show that for constrained interpolation, the Maximum *A Posteriori* (MAP) or mode of the posterior is the optimal constrained interpolation function in the RKHS, and also illustrate in simulations that the unconstrained mean and constrained MAP coincide only when the unconstrained mean satisfies the constraint. This holds when the GP is constrained to a convex set of functions, which is the case in this paper where we condition on linear transformations of a function restricted to a convex set.

3.7. An Alternative Approach based on Conditional Expectations

Da Veiga and Marrel (2012, 2015) propose an approach for approximating the first two moments of the constrained posterior, $\mathbf{f}^*|Y, C$, using conditional expectations of the truncated multivariate Gaussian. This means, in the context of this paper, that the first two moments of $\mathbf{f}^*|Y, C$ are computed using the first two moments of the latent variable $\mathbf{C}$. To apply this idea using the formulation of this paper, we can make use of the following result.

Corollary 6 *Let the matrices A, B, Σ and the truncated Gaussian random variable $\boldsymbol{C}$ be as defined in Lemma 1, and let ν, Γ be the expectation and covariance of $\boldsymbol{C}$. Then the expectation and covariance of the predictive distribution $\boldsymbol{f}^*|Y, C$ are given as*

$$\begin{aligned} \mathbb{E}(\boldsymbol{f}^*|Y, C) &= \mu^* + A(\nu - \mathcal{L}\mu^v) + B(Y - \mu), \\ cov(\boldsymbol{f}^*|Y, C) &= \Sigma + A\Gamma A^T. \end{aligned} \tag{14}$$

Moreover, if $\widetilde{A}$, $\widetilde{B}$ and $\widetilde{\Sigma}$ are the matrices defined in Lemma 4, then the expectation and variance of the predictive distribution of the constraint $\mathcal{L}f(\boldsymbol{x}^*)|Y, C$ are given as

$$\begin{aligned} \mathbb{E}(\mathcal{L}f(\boldsymbol{x}^*)|Y, C) &= \mathcal{L}\mu^* + \widetilde{A}(\nu - \mathcal{L}\mu^v) + \widetilde{B}(Y - \mu), \\ var(\mathcal{L}f(\boldsymbol{x}^*)|Y, C) &= \widetilde{\Sigma} + \widetilde{A}\Gamma\widetilde{A}^T. \end{aligned} \tag{15}$$

The results follows directly from the distributions derived in Lemmas 1 and 4, and moments of compound distributions. A proof is included in Appendix E for completeness.

Da Veiga and Marrel (2012, 2015) make use of a Genz approximation (Genz, 1992, 1997) to compute ν, Γ for inference using (14). They also introduce a crude but faster correlation-free approximation that can be used in the search for virtual observation locations. With this approach, (15) is used where ν, Γ are computed under the assumption that $\mathrm{cov}(\widetilde{C}|Y)$ is diagonal. We can state this approximation as follows:

$$\nu_i \approx m_i + s_i \frac{\phi(\widetilde{a}_i) - \phi(\widetilde{b}_i)}{\Phi(\widetilde{b}_i) - \Phi(\widetilde{a}_i)}, \quad \Gamma_{i,i} \approx s_i^2 \left[1 + \frac{\widetilde{a}_i \phi(\widetilde{a}_i) - \widetilde{b}_i \phi(\widetilde{b}_i)}{\Phi(\widetilde{b}_i) - \Phi(\widetilde{a}_i)} - \left(\frac{\phi(\widetilde{a}_i) - \phi(\widetilde{b}_i)}{\Phi(\widetilde{b}_i) - \Phi(\widetilde{a}_i)} \right)^2 \right],$$

where m_i is the i-th component of $\mathbb{E}(\widetilde{C}|Y) = \mathcal{L}\mu^v + A_1(Y - \mu)$, $s_i = \sqrt{\mathrm{cov}(\widetilde{C}|Y)_{i,i}} = \sqrt{(B_1)_{i,i}}$, $\widetilde{a}_i = (a(X^v)_i - m_i)/s_i$, $\widetilde{b}_i = (b(X^v)_i - m_i)/s_i$, ϕ and Φ are the pdf and cdf of the standard normal distribution and Γ is diagonal with elements $\Gamma_{i,i}$. We will make use of these approximations in some of the examples in Section 4.2 for comparison.

3.8. Numerical Considerations

For numerical implementation, we discuss some key considerations with the proposed model. One of the main issues with implementation of GP models in terms of numerical stability is related to covariance matrix inversion, which is why alternatives such as Cholesky factorization are recommended in practice. This does however not alleviate problems related to ill-conditioned covariance matrices. This is a common problem in computer code emulation (zero observational noise) in particular, where training points might be 'too close to each other' in terms of the covariance function, leaving the covariance matrix close to degenerate as some of the observations become redundant. A common remedy is to introduce a 'nugget' term on the diagonal entries of the covariance matrix, in the form of additional white noise on the observations. This means using a small $\sigma > 0$ instead of $\sigma = 0$ in Equation (2), even when the observations are noiseless. In terms of matrix regularization this is equivalent to Tikhonov regularization. See for instance Ranjan et al. (2010) and Andrianakis and Challenor (2012) which give a detailed discussion and recommendations for how to choose appropriate value for σ. In practice, a fixed small value is often used without further analysis, as long as the resulting condition number is not too high. This approach can be justified since the use of a nugget term has a straightforward interpretation, as opposed to other alternatives such as pseudoinversion. In our experiments on noiseless regression we fix $\sigma^2 = 10^{-6}$, as the error introduced by adding a variance of 10^{-6} to the observations is negligible.

Similarly, for the virtual observations used in this paper we make use of the noise parameter σ_v to avoid ill-conditioning of the matrix B_1 defined in Lemma 1. B_1 is the covariance matrix of the transformed GP, $\widetilde{C}|Y$, and B_1^{-1} together with $(K_{X,X} + \sigma^2 I_N)^{-1}$ are needed for all the posterior computations that involve constraints. The virtual noise parameter σ_v has a similar interpretation as σ, but where the artificial added noise acts on observations of the transformed process. Here $\sigma_v = 0$ means that the constraints are enforced with probability 1, $\sigma_v > 0$ implies that the constraints are enforced in a soft way, and $\sigma_v \to \infty$ provides no constraint at all. In the numerical examples presented in this

paper, a fixed value $\sigma_v^2 = 10^{-6}$ has been used to approximate hard constraints with an error we find negligible.

As for computational complexity, we may start by first looking at the operations involved in computing the posterior predictive distribution at M inputs $\mathbf{x}_1^*, \ldots, \mathbf{x}_M^*$ (including covariances), using Lemma 2. We first make note of the operations needed in the unconstrained case, i.e. standard GP regression with Gaussian noise, for comparison. If there are $N \geq M$ observations in the training set, then the complexity is dominated by the Cholesky factorization $L = Chol(K_{X,X} + \sigma^2 I_N)$, which require an order of N^3 operations and N^2 in memory. The Cholesky factor may be stored for subsequent predictions. Then, to compute the posterior predictive distribution at M new inputs, the number of operations needed is dominated by matrix multiplication and solving triangular systems, of orders NM^2 and N^2M. When a number N_v of virtual observation locations are included, we are essentially dealing with the same computations as the standard GP regression, but with $N + N_v$ number of observations. I.e. the computations involved are of order $(N + N_v)^3$ in time and $(N + N_v)^2$ in memory. The order of operations in Lemma 2 was chosen such that the Cholesky factor L that only depends on the training data can be reused. For a new set X^v of size N_v, the computations needed for prediction at M new locations X^* will only require the Cholesky factorization $L_1 = Chol(B_1)$ of order N_v^3. When both L and L_1 are stored, the remaining number of operations will be of order N^2M or N_v^2M for solving triangular systems, and NM^2, N_vM^2 or NMN_v for matrix multiplications.

In order to sample from the posterior using Algorithm 3, some additional steps are required. After the computations of Lemma 2 we continue to factorize the $M \times M$ covariance matrix Σ and generate samples from the truncated Gaussian $\widetilde{C}|Y, C$. The complexity involved in sampling from this N_v-dimensional truncated Gaussian depends on the sampling method of choice, see Section 3.2. We can combine k of these samples with k samples from a standard normal $\mathcal{N}(\mathbf{0}, I_M)$ to obtain samples of the final posterior, using an order of $MN_vk + M^2k$ operations. The total procedure of generating k samples at $M \leq N$ new inputs is therefore dominated by matrix operations of order $(N + N_v)^3$, MN_vk and M^2k, together with the complexity involved with sampling from a N_v-dimensional truncated Gaussian. For subsequent prediction it is convenient to here also reuse the samples generated from the truncated Gaussian, together with results that only involve X and X^v. This means storing matrices of size $N_v \times k$, $N \times N$ and $N_v \times N_v$. The remaining computations are then dominated by operations of order N^2M, N_v^2M, NM^2, N_vM^2, NMN_v, MN_vk, and M^2k. In the algorithms used to find virtual observation locations, Algorithm 5 and 7, we make sure to reuse computations that only involve the training data in each iteration of $N_v = 1, 2, \ldots$. This means that in addition to the previously stated operations, we need to perform Cholesky factorization of order N_v^3 and generate samples from a N_v-dimensional truncated Gaussian. This is initially very cheap, but becomes the main numerical challenge when N_v grows large. As the purpose of these algorithms is to find a small set X^v, that also avoids sampling issues due to serial correlation, we found it useful to output the minimal constraint probability p^* found in each iteration to reveal if the stopping criterion used (in terms of p_{target} or a maximum number of iterations) was unrealistic in practice.

4. Gaussian Process Modelling with Boundedness and Monotonicity Constraints

In this section we present some examples related to function estimation where we assume that the function and some of its partial derivatives are bounded. This is the scenario considered in the literature on shape-constrained GPs, and alternative approaches to GPs under linear constraints are usually presented in this setting. We start by a brief discussion on related work, followed by some numerical experiments using boundedness and monotonicity constraints. The numerical experiments were performed using the Python implementation available at `https://github.com/cagrell/gp_constr`.

4.1. Related Work

We give a brief overview of some alternative and related approaches to constrained GPs. For the approaches that rely on imposing constraints at a finite set of virtual observation locations, we recall that the constraint probability can be used in the search for a suitable set of virtual observation locations. The constraint probability is the probability that the constraint holds at an arbitrary input $\mathbf{x}$, $p_c(\mathbf{x})$ given in (11). Some key characteristics of the approaches that make use of virtual observations are summarized in Table 1.

The related work most similar to the approach presented in this paper is that of Wang and Berger (2016) and Da Veiga and Marrel (2012, 2015). Wang and Berger (2016) make use of a similar sampling scheme for noiseless GP regression applied to computer code emulation. A Gibbs sampling procedure is used for inference and to estimate the constraint probability $p_c(\mathbf{x})$ in the search for virtual observation locations. The approach of Da Veiga and Marrel (2012, 2015) is based on computation of the posterior mean and covariance of the constrained GP, using the equations that are also restated in this paper in Corollary 6. They make use of a Genz approximation for inference (Genz, 1992, 1997), and also introduce a crude but faster correlation-free approximation that can be used in the search for virtual observation locations. The approach of Da Veiga and Marrel (2012, 2015) is discussed further in the numerical experiments below, where we illustrate the idea in Example 1 and in Example 2 study an approximation of the posterior constrained GP using the constrained moments with a Gaussian distribution assumption. A major component in (Da Veiga and Marrel, 2012, 2015), (Wang and Berger, 2016) and this paper is thus computation involving the truncated multivariate Gaussian. Besides the choice of method for sampling from this distribution, the main difference with our approach is that we leverage Cholesky factorizations and noisy virtual observations for numerical stability.

A different approach that also make use of virtual observations is that of Riihimki and Vehtari (2010), where a *probit* likelihood is used to represent interval observations of the derivative process to impose monotonicity. They then make use of Expectation Propagation (EP) to approximate the posterior with a multivariate Gaussian. As pointed out by Golchi et al. (2015), the Gaussian assumption is questionable if the constraint (in this case monotonicity) does not hold with high probability a priori. Golchi et al. (2015) proceeds to develop a fully Bayesian procedure for application to computer experiments by the use of Sequentially Constrained Monte Carlo Sampling (SCMC). A challenge with this approach however is that finding a suitable set of virtual observation locations is difficult. Our experience, in agreement with (Wang and Berger, 2016; Da Veiga and Marrel, 2012,

	Virtual obs. likelihood	**Inference strategy**	**Strategy for finding X^v**
Agrell (2019)	Indicator + noise	Sampling (Minimax tilting)	Based on estimating $p_c(\mathbf{x})$ from samples
Wang and Berger (2016)	Indicator	Sampling (Gibbs)	Based on estimating $p_c(\mathbf{x})$ from samples
Da Veiga and Marrel (2012, 2015)	Indicator	Moment approximation (Genz)	Based on approximating $p_c(\mathbf{x})$ assuming Gaussian posterior distribution
Riihimki and Vehtari (2010)	Probit	Expectaion Propagation	Based on approximating $p_c(\mathbf{x})$ assuming Gaussian posterior distribution
Golchi et al. (2015)	Probit	SCMC	NA

Table 1: Summary of alternative approaches that make use of virtual observations. The table compares the likelihood used for virtual observations, the method used for inference and to determine the set of virtual observation locations X^v.

2015; Riihimki and Vehtari, 2010), is that for practical applications in more than a few dimensions, such a strategy is essential to avoid numerical issues related to high serial correlation, and also to reduce the number of virtual observation locations needed. It is also worth noting that a strategy that decouples computation involving training data and virtual observation locations from inference at new locations is beneficial. For the approaches discussed herein that rely on sampling/approximation related to the truncated multivariate Gaussian, the samples/approximations can be stored and reused as discussed in Section 3.8.

There are also some approaches to constrained GPs that are not based on the idea of using virtual observations. An interesting approach by Maatouk and Bay (2017), that is also followed up by López-Lopera et al. (2018), is based on modelling a conditional process where the constraints hold in the entire domain. They achieve this through finite-dimensional approximations of the GP that converge uniformly pathwise. With this approach, sampling from a truncated multivariate Gaussian is also needed for inference, in order to estimate the coefficients of the finite-dimensional approximation that arise from discretization of the input space. The authors give examples in 1D and 2D, but note that due to the structure of the approximation, the approach will be time consuming for practical applications in higher dimensions. There are also other approaches that consider special types of shape constraints, but where generalization seems difficult. See for instance (Abrahamsen and Benth, 2001; Yoo and Kyriakidis, 2006; Michalak, 2008; Kleijnen and Beers, 2013; Lin and Dunson, 2014; Lenk and Choi, 2017).

4.2. Numerical Experiments

In this section we will make us of the following constraints:

- $a_0(\mathbf{x}) \leq f(\mathbf{x}) \leq b_0(\mathbf{x})$
- $a_i(\mathbf{x}) \leq \partial f/\partial x_i(\mathbf{x}) \leq b_i(\mathbf{x})$

for all $\mathbf{x}$ in some bounded subset of $\mathbb{R}^{n_x}$, and $i \in \mathcal{I} \subset \{1, \ldots, n_x\}$. Without loss of generality we assume that the constrains on partial derivatives are with respect to the first k components of $\mathbf{x}$, i.e. $\mathcal{I} = \{1, \ldots, k\}$ for some $k \leq n_x$.

As the prior GP we will assume a constant mean $\mu = 0$ and make use of either the RBF or Matérn 5/2 covariance function. These are stationary kernels of the form

$$K(\mathbf{x}, \mathbf{x}') = \sigma_K^2 k(r), \; r = \sqrt{\sum_{i=1}^{n_x} \left(\frac{x_i - x_i'}{l_i} \right)^2}, \tag{16}$$

with variance parameter σ_K^2 and length scale parameters l_i for $i = 1, \ldots, n_x$. The radial basis function (RBF), also called squared exponential kernel, and the Matérn 5/2 kernel are defined through the function $k(r)$ as

$$k_{\text{RBF}}(r) = e^{-\frac{1}{2}r^2} \text{ and } k_{\text{Matérn 5/2}}(r) = (1 + \sqrt{5}r + \frac{5}{3}r^2)e^{-\sqrt{5}r}.$$

In general, the kernel hyperparameters σ_K^2 and l_i are optimized together with the noise variance σ through MLE. In the examples that consider noiseless observations, the noise variance is not estimated, but set to a small fixed value as discussed in Section 3.8. With the above choice of covariance function, existence of the transformed GP is ensured. In fact, the resulting process is infinitely differentiable using the RBF kernel (see Adler, 1981, Theorem 2.2.2) and twice differentiable with the Matérn 5/2. These prior GP alternatives were chosen as they are the most commonly used in the literature, and thus a good starting point for illustrating the effect of including linear constraints. We note that although it is not in general possible to design mean and covariance functions that produce GPs that satisfy the constraints considered in this paper, one could certainly ease numerical computations by selecting a GP prior based on the constraint probability $p(C|Y, \theta)$ in (7), and for instance make us of a mean function that is known to satisfy the constraint.

If we let $\mathcal{F}^0 f = f$, $\mathcal{F}^i f = \partial f/\partial x_i$, and $X^{v,i}$ be the set of S_i virtual observations corresponding to the i-th operator $\mathcal{F}^i$, then we can make use of the formulation in Section 3.5 and equations from Appendix C to obtain

$$\mathcal{L}\mu^v = [\mu \mathbf{1}_{S_0}, \mathbf{0}_{S_{[1,k]}}]^T,$$

where $\mathbf{1}_{S_1}$ is the vector $[1, \ldots, 1]^T$ of length S_1 and $\mathbf{0}_{S_{[1,k]}}$ is the vector $[0, \ldots, 0]^T$ of length $S_{[1,k]} = -S_0 + \sum S_i$. Furthermore,

$$\begin{aligned}
K_{X,X^v}\mathcal{L}^T &= \left[K_{X,X^{v,0}}, (K^{1,0}_{X^{v,1},X})^T, \ldots, (K^{k,0}_{X^{v,k},X})^T \right], \\
K_{X^*,X^v}\mathcal{L}^T &= \left[K_{X^*,X^{v,0}}, (K^{1,0}_{X^{v,1},X^*})^T, \ldots, (K^{k,0}_{X^{v,k},X^*})^T \right], \\
\mathcal{L}K_{X^v,X^v}\mathcal{L}^T &= \begin{bmatrix}
K_{X^{v,0},X^{v,0}} & (K^{1,0}_{X^{v,1},X^{v,0}})^T & \cdots & (K^{k,0}_{X^{v,k},X^{v,0}})^T \\
K^{1,0}_{X^{v,1},X^{v,0}} & K^{1,1}_{X^{v,1},X^{v,1}} & \cdots & K^{1,k}_{X^{v,1},X^{v,k}} \\
\vdots & \vdots & \ddots & \vdots \\
K^{k,0}_{X^{v,k},X^{v,0}} & K^{k,1}_{X^{v,k},X^{v,1}} & \cdots & K^{k,k}_{X^{v,k},X^{v,k}}
\end{bmatrix},
\end{aligned}$$

where we have used the notation

$$K^{i,0}(\mathbf{x},\mathbf{x}') = \frac{\partial}{\partial x_i} K(\mathbf{x},\mathbf{x}') \text{ and } K^{i,j}(\mathbf{x},\mathbf{x}') = \frac{\partial^2}{\partial x_i \partial x_i'} K(\mathbf{x},\mathbf{x}').$$

The use of constraints related to boundedness and monotonicity is illustrated using three examples of GP regression. Example 1 considers a function $\mathrm{f}:\mathbb{R}\to\mathbb{R}$ subjected to boundedness and monotonicity constraints. In Example 2 a function $\mathrm{f}:\mathbb{R}^4\to\mathbb{R}$ is estimated under the assumption that information on whether the function is monotone increasing or decreasing as a function of the first two inputs is known, i.e. $\mathrm{sgn}(\partial \mathrm{f}/\partial x_1)$ and $\mathrm{sgn}(\partial \mathrm{f}/\partial x_2)$ are known. In Example 3 we illustrate how monotonicity constraints in multiple dimensions can be used in prediction of pressure capacity of pipelines.

4.2.1. **Example 1**: Illustration of Boundedness and Monotonicity in 1D

As a simple illustration of imposing constraints in GP regression, we first consider the function $\mathrm{f}:\mathbb{R}\to\mathbb{R}$ given by $\mathrm{f}(x) = \frac{1}{3}[\tan^{-1}(20x-10) - \tan^{-1}(-10)]$. We assume that the function value is known at 7 input locations given by $x_i = 0.1 + 1/(i+1)$ for $i = 1,\ldots,7$. First, we assume that the observations are noiseless, i.e. $\mathrm{f}(x_i)$ is observed for each x_i. Estimating the function that interpolates at these observations is commonly referred to as *emulation*, which is relevant when dealing with data from computer experiments. Our function $\mathrm{f}(x)$ is both bounded and increasing on all of $\mathbb{R}$. In this example we will constrain the GP to satisfy the conditions that for $x\in[0,1]$, we have that $\mathrm{d}\mathrm{f}/\mathrm{d}x \geq 0$ and $a(x) \leq \mathrm{f}(x) \leq b(x)$ for $a(x) = 0$ and $b(x) = \frac{1}{3}\ln(30x+1) + 0.1$. The function is shown in Figure 1 together with the bounds and the 7 observations.

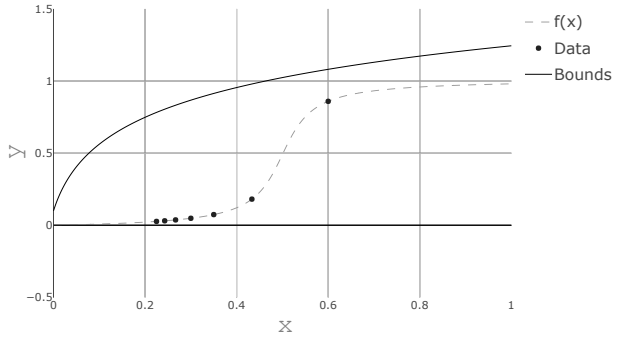

Figure 1: Function to emulate in Example 1

We select an RBF kernel (16) with parameters $\sigma_K = 0.5$ (variance) and $l = 0.1$ (length scale). To represent noiseless observations we set $\sigma^2 = 10^{-6}$, where σ^2 is the noise variance in the Gaussian likelihood. The assumed noise on virtual observations will also be set to 10^{-6}. To illustrate the effect of adding constraints we show the constrained GP using only boundedness constraint, only monotonicity constraint and finally when both constraints are imposed simultaneously. Figure 2 shows the resulting GPs. Algorithm 7 was used with a target probability $p_{target} = 0.99$ to determine the virtual observation locations that are indicated in the figures, and the posterior mode was computed by maximizing a Gaussian

kernel density estimator over the samples generated in Algorithm 3. For both constraints, 17 locations was needed for monotonicity and only 3 locations was needed to impose boundedness when the virtual locations for both constraints where optimized simultaneously. This is reasonable, as requiring $f(0) > 0$ is sufficient to ensure $f(x) > 0$ for $x \geq 0$ when f is increasing, and similarly requiring $f(x^v) < b(x^v)$ for some few points $x^v \in [0.6, 1]$ should suffice. But note that Algorithm 7 finds the virtual observation locations for both constraints simultaneously. Here $x^v = 0$ for boundedness was first identified, followed by some few points for monotonicity, followed by a new point x^v for boundedness etcetera.

For illustration purposes none of the hyperparameters of the GP were optimized. Moreover, for data sets such as the one in this example using plug-in estimates obtained from MLE generally not appropriate due to overfitting. Maximizing the marginal likelihood for the unconstrained GP gives a very poor model upon visual inspection ($\sigma_K = 0.86, l = 0.26$). However, it was observed that the estimated parameters for the constrained model (using Eq. (8)) gives estimates closer to the selected prior which seems more reasonable ($\sigma_K = 0.42, l = 0.17$), and hence the inclusion of the constraint probability, $p(C|Y, \theta)$, in the likelihood seems to improve the estimates also for the unconstrained GP.

We may also assume that the observations come with Gaussian white noise, which in terms of numerical stability is much less challenging than interpolation. Figure 3 shows the resulting GPs fitted to 50 observations. The observations were generated by sampling $x_i \in [0.1, 0.8]$ uniformly, and y_i from $\mathrm{f}(x_i)+\varepsilon_i$ where ε_i are i.i.d. zero mean Gaussian with variance $\sigma^2 = 0.04$. Both GPs were optimized using plug-in estimates of hyperparameters (σ_K, l, σ^2) given by maximizing the marginal likelihood. These are $(\sigma_K = 0.34, l = 0.32, \sigma^2 = 0.053)$ for the constrained case and $(\sigma_K = 0.34, l = 0.23, \sigma^2 = 0.040)$ for the unconstrained case. We observe that the estimated noise variance is larger in the constrained model than the unconstrained where this estimate is exact.

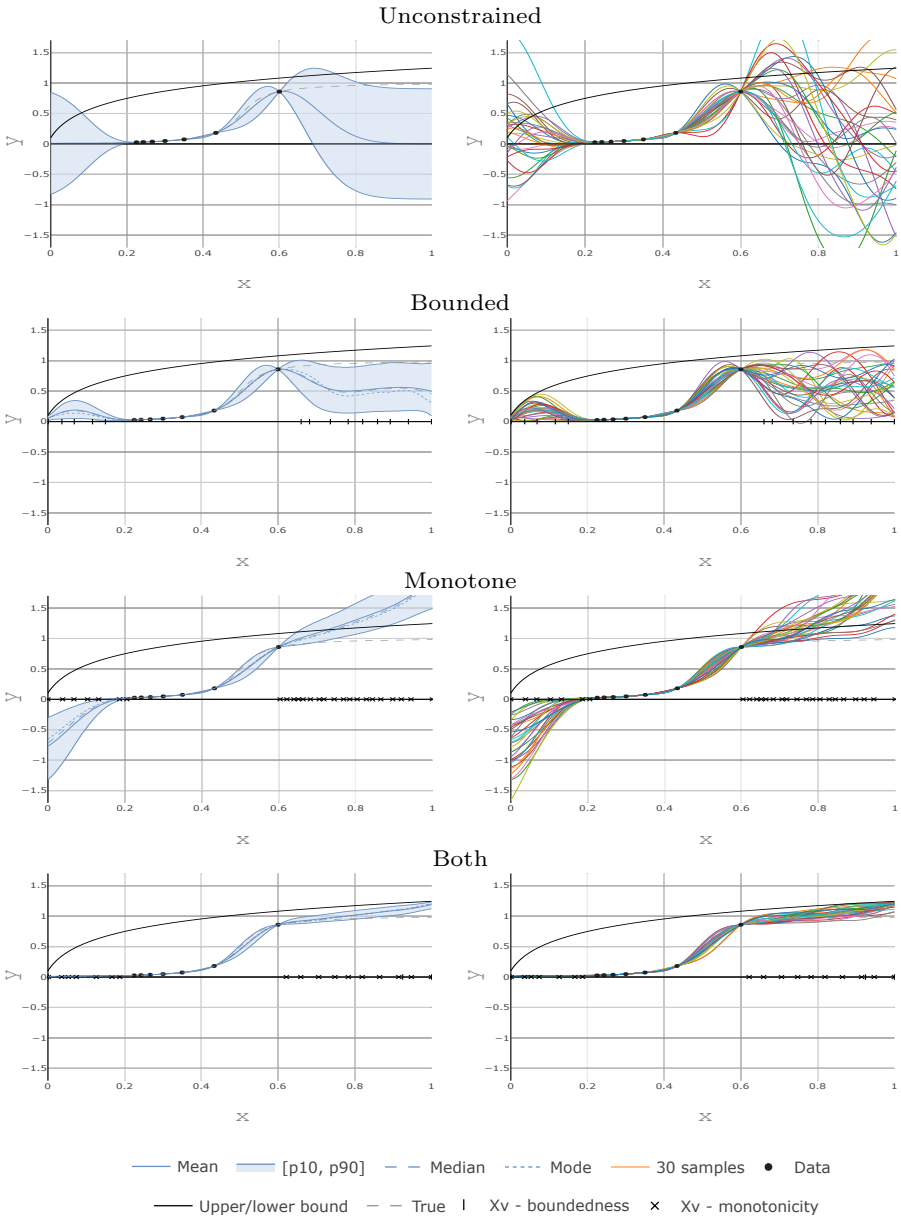

Figure 2: The GP with parameters $\sigma_K = 0.5$ (variance) and $l = 0.1$ (length scale) used in Example 1. The virtual observation locations are indicated by markers on the x-axis.

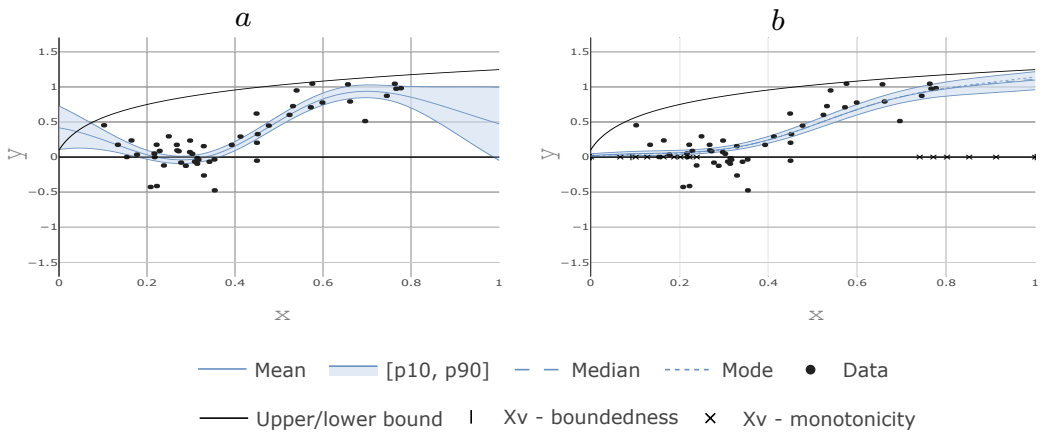

Figure 3: Unconstrained (a) and constrained (b) GPs fitted to 50 observations with Gaussian noise. The predictive distributions are shown, i.e. the distribution of $f(x)$ where $y = f(x) + \varepsilon$.

Da Veiga and Marrel (2015) propose to use estimates of the posterior mean and variance of $\mathcal{L}f(\mathbf{x})|Y, C$ to estimate the constraint probability $p_c(\mathbf{x})$ assuming a Gaussian distribution. They also introduce the faster correlation-free approximation, where the parameters are estimated under the assumption that observations of $\mathcal{L}f(\mathbf{x})|Y$ at different input locations $\mathbf{x}$ are independent (see Section 3.7). In Figure 4 we plot estimates of $p_{c,i}(\mathbf{x})$, for the boundedness and monotonicity constraint individually, using the approach in this paper (13) and the two moment based approximations. The plots were generated first after a total of 5 and then 10 virtual observations locations had been included in the model with both constraints. As we are mainly interested in finding $\mathbf{x}^* = \arg\min p_{c,i}(\mathbf{x})$, Figure 4 indicates that the moment based approximations are appropriate initially. However, as more virtual observation locations are included, the correlation-free assumption becomes questionable. But it could still serve as a useful starting point, and in a strategy based on checking the approximation error from time to time, it should still be possible to take advantage of the computational savings offered by the correlation-free approximation.

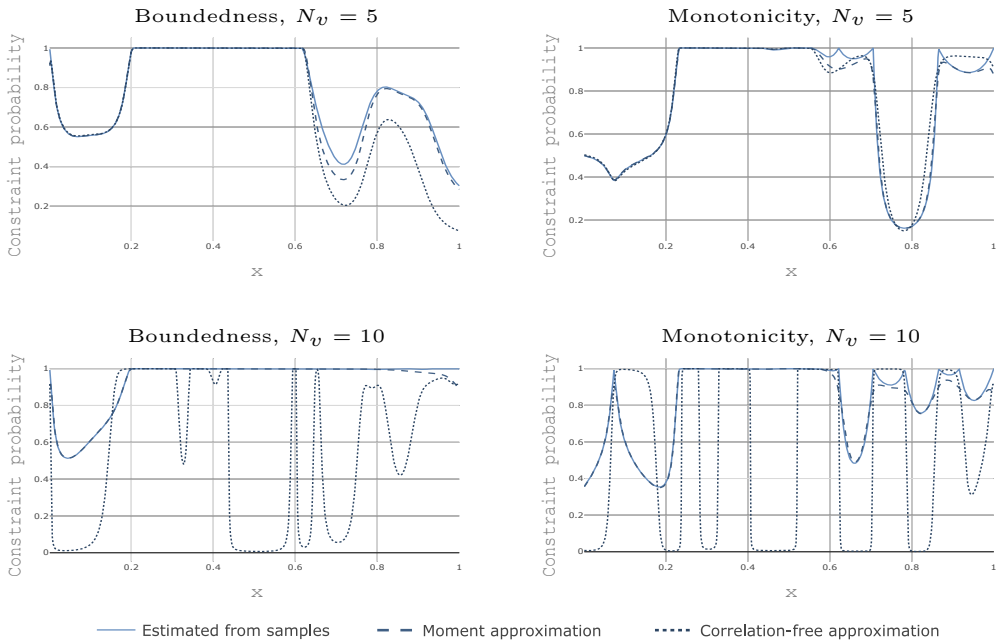

Figure 4: Constraint probability $p_c(\mathbf{x})$ computed using the estimate (13) together with the moment based approximations from Da Veiga and Marrel (2015). The constraint probability is shown for monotonicity and boundedness, where N_v is the total number of virtual observation locations used in the model.

4.2.2. **Example 2**: 4D Robot Arm Function

In this example we consider emulation of a function $\mathrm{f} : \mathbb{R}^4 \to \mathbb{R}$, where we assume that the sign of the first two partial derivatives, $\mathrm{sgn}(\partial \mathrm{f}/\partial x_1)$ and $\mathrm{sgn}(\partial \mathrm{f}/\partial x_2)$, are known. The function to emulate is

$$\mathrm{f}(\mathbf{x}) = \sum_{i=1}^{m} L_i \cos\left(\sum_{j=1}^{i} \tau_j\right),$$

for $m = 2$, and $\mathbf{x} = [L_1, L_2, \tau_1, \tau_2]$. The function is inspired by the robot arm function often used to test function estimation (An and Owen, 2001). Here $\mathrm{f}(\mathbf{x})$ is the y-coordinate of a two dimensional robot arm with m line segments of length $L_i \in [0, 1]$, positioned at angle $\tau_i \in [0, 2\pi]$ with respect to the horizontal axis. The constraints on the first two partial derivatives thus implies that it is known whether or not the arm will move further away from the x-axis, as a function of the arm lengths, L_1 and L_2, for any combination of τ_1 and τ_2.

In this experiment we first fit an unconstrained GP using 40 observations taken from a Latin hypercube sample over the input space $[0, 1]^2 \times [0, 2\pi]^2$. A Matérn 5/2 covariance function is used with plug-in MLE hyperparameters. Then, a total of 80 virtual observation locations are found using the procedure in Algorithm 7, where we search over a finite candi-

date set of 1000 locations in the minimization of the constraint probability. We repeat this procedure 100 times and report performance using the predictivity coefficient Q^2, predictive variance adequation (PVA) and the average width of 95% confidence intervals (AWoCI).

Given a set of tests $y_1, \ldots, y_{n_{test}}$ and predictions $\hat{y}_1, \ldots, \hat{y}_{n_{test}}$, Q^2 is defined as

$$Q^2 = 1 - \sum_{i=1}^{n_{test}} (\hat{y}_i - y_i)^2 / \sum_{i=1}^{n_{test}} (\bar{y} - y_i)^2,$$

where $\bar{y}$ is the mean of $y_1, \ldots, y_{n_{test}}$. In our experiments the predictions $\hat{y}_i$ are given by the posterior mean of the GP. The PVA criterion is defined as

$$\text{PVA} = \left| \log \left(\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \frac{(\hat{y}_i - y_i)^2}{\hat{\sigma}_i^2} \right) \right|,$$

where $\hat{\sigma}_i^2$ is the predictive variance. This criterion evaluates the quality of the predictive variances and to what extent confidence intervals are reliable. The smaller the PVA is, the better (Bachoc, 2013). In addition to this criterion, it is also useful to evaluate the size of confidence intervals. For this we compute the average width of 95% confidence intervals

$$\text{AWoCI} = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (p_{0.975}^{(i)} - p_{0.025}^{(i)}),$$

where $p_{0.975}^{(i)}$ and $p_{0.025}^{(i)}$ are the predicted 97.5% and 2.5% percentiles.

The result of 100 predictions for one single experiment is shown in Figure 5. As expected, the estimated prediction uncertainty is reduced significantly using the constrained model, and single predictions given by the posterior mean are also improved. In Table 2 we summarize the results from running 100 of these experiments. In each experiment, Q^2, PVA and AWoCI was computed from prediction at 1000 locations sampled uniformly in the domain. We also report the probability that the constraint holds in the unconstrained GP, $p(C|Y)$ given in (7), and the CPU time in seconds used to generate 10^4 samples from the posterior on an Intel® Core™ i5-7300U 2.6GHz CPU. For comparison, we also include predictions from moment-based approximations using the approach of Da Veiga and Marrel (2012, 2015). We study in particular their approach for finding the set of virtual observation locations, as discussed in Section 3.7 and illustrated in the previous example. In total, the following alternatives are considered:

1. **Unconstrained**: The initial GP without constraints.
2. **Constrained**: The constrained GP using the approach presented in this paper.
3. **Moment approx. 1**: Using the sampling scheme of this paper for inference, but where the moment based approximation is used in the search for virtual observation locations.
4. **Moment approx. 2**: Using moment approximation for both inference and searching for virtual observation locations. This is one of the procedures from Da Veiga and Marrel (2012, 2015).

5. **Correlation-free approx.**: Same as **Moment approx. 1** but where the correlation-free approximation is used in the search for virtual observation locations.

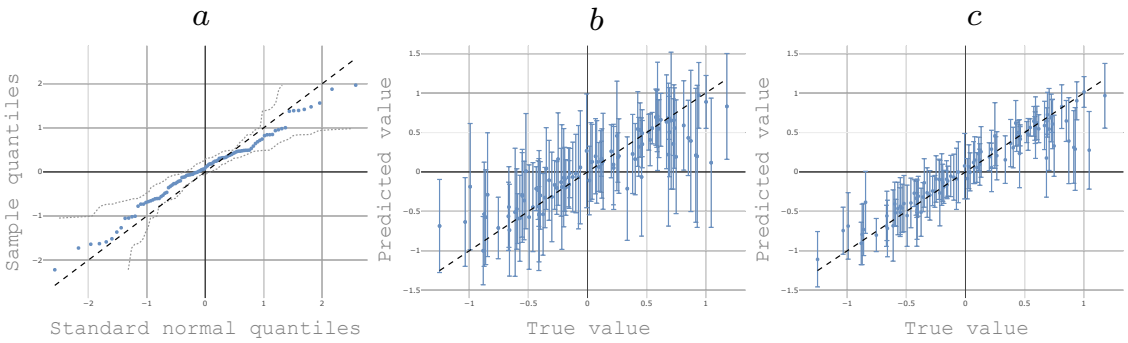

Figure 5: Figure a shows a qq-plot with 95% confidence band of 100 normalized residuals $(y_i - \mu_i)/(\sigma_i)$, where μ_i and σ_i^2 are the mean and variance of the predictive distribution of the unconstrained GP. In Figure b, predictions vs the true function value is shown together with a $[0.025, 0.975]$ (95%) percentile interval for the unconstrained GP. The same type of figure is shown in c for the constrained GP.

In Table 2 we see that the use of constraints is beneficial in terms of both a higher Q^2 (better predictive performance) and a smaller PVA (higher quality of predictive variances). With the exception of 'Moment approx. 2', the inclusion of constraints provides significant uncertainty reduction as the width of 95% confidence intervals (AWoCI) are reduce by almost a factor of 2 on average. A box plot showing AWoCI from the 100 experiments is also shown in Figure 6. We see that the different approaches for estimating the constraint probability, $p_c(\mathbf{x})$, in the search for virtual observation locations work equally well. The Gaussian assumption on the posterior $\mathbf{f}^*|Y, C$ on the other hand is not optimal, as it tends to overestimate the uncertainty in this example.

	$p(C\|Y)$	T_s	PVA	Q^2	AWoCI
Unconstrained			3.03	0.7558	0.99
Constrained	4.1E-34	24.8	2.85	0.8842	0.54
Moment approx. 1	2.4E-36	25.2	2.84	0.8844	0.54
Moment approx. 2	2.4E-36	25.2	2.84	0.8844	0.83
correlation-free approx.	8.6E-37	21.1	2.91	0.8775	0.55

Table 2: Average values from 100 experiments of the robot arm function. T_s is the CPU time in seconds used to generate 10^4 samples.

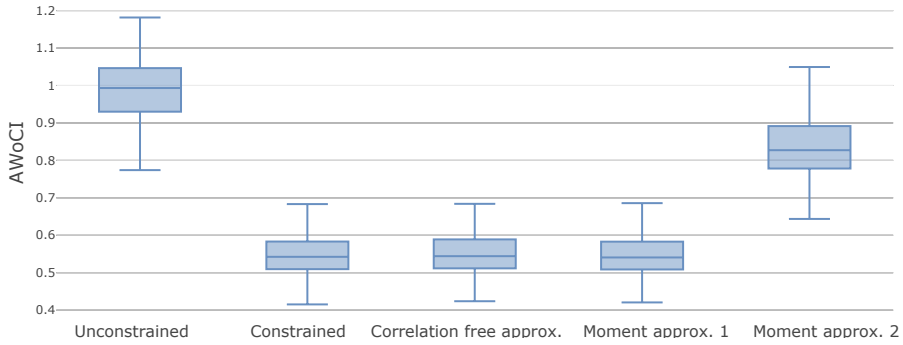

Figure 6: Average width of confidence intervals (AWoCI) from 100 experiments of the robot arm function.

4.2.3. **Example 3**: Pipeline Pressure Capacity

In this example we consider a model for predicting the pressure capacity of a steel pipeline with defects due to corrosion. As corrosion is one of the major threats to the integrity of offshore pipelines, experiments are carried out to understand how metal loss due to corrosion affects a pipeline's capacity with respect to internal pressure (Sigurdsson et al., 1999; Amaya et al., 2019). These include full scale burst tests and numerical simulation through Finite Element Analysis (FEA). Results from this type of experiments serve as the basis for current methodologies used in the industry for practical assessment of failure probabilities related to pipeline corrosion, such as ASME B31G or DNVGL-RP-F101. We consider experiments related to a single rectangular shaped defect, which is essential to these methodologies.

To simulate synthetic experiments of the burst capacity of a pipeline with a rectangular defect, we will use the simplified capacity equation given in in (RP-F101 DNV GL, 2017). The maximum differential pressure (capacity in MPa) the pipeline can withstand without

bursting is in the simplified equation given as

$$P_{cap}(\sigma_u, D, t, d, l) = 1.05\frac{2t\sigma_u}{D-t}\frac{1-d/t}{1-\frac{d/t}{Q}}, \quad Q = \sqrt{1+0.31\frac{l^2}{Dt}},$$

where $\sigma_u \in [450, 550]$ (MPa) is the ultimate tensile strength of the material, $D \in [10t, 50t]$ (mm) and $t \in [5, 30]$ (mm) are the outer diameter and wall thickness of the pipeline, and $d \in [0, t]$ (mm) and $l \in [0, 1000]$ (mm) are the depth and length of the rectangular defect.

From the physical phenomenon under consideration, we know that the capacity of the pipeline will decrease if the size of the defect were to increase. Similarly, we know that the pipeline capacity increases with a higher material strength or wall thickness, and decreases as a function of the diameter, all else kept equal. In the form of partial derivatives we can express this information as: $\frac{\partial P_{cap}}{\partial d} < 0$, $\frac{\partial P_{cap}}{\partial l} < 0$, $\frac{\partial P_{cap}}{\partial \sigma_u} > 0$, $\frac{\partial P_{cap}}{\partial t} > 0$ and $\frac{\partial P_{cap}}{\partial D} < 0$.

For convenience we will transform the input variables to the unit hypercube. Let $\mathbf{x}$ denote the transformed input vector $\mathbf{x} = [x_1, \ldots, x_5]$, where $x_1 = (\sigma_u - 450)/(550 - 450)$, $x_2 = (D/t - 10)/(50 - 10)$, $x_3 = (t - 5)/(30 - 5)$, $x_4 = d/t$ and $x_5 = l/1000$. We will make use of the function

$$f(\mathbf{x}) = P_{cap}(\mathbf{x}) \text{ for } \mathbf{x} \in [0, 1]^5,$$

and assume that the burst capacity observed in an experiment is $f(\mathbf{x}) + \varepsilon$, where ε is a zero mean Normal random variable with variance $\sigma^2 = 4$. The constraints on the partial derivatives after the transformation becomes: $\frac{\partial f}{\partial x_1} > 0$, $\frac{\partial f}{\partial x_2} < 0$, $\frac{\partial f}{\partial x_3} > 0$, $\frac{\partial f}{\partial x_4} < 0$ and $\frac{\partial f}{\partial x_5} < 0$ for $\mathbf{x} \in [0, 1]^5$.

In this example we thus have five constraints available, represented by bounds on the partial derivative of $f(\mathbf{x})$ w.r.t. x_i for $i = 1, \ldots, 5$. Besides studying the effect of including all five constraints, we will test some different alternatives using a smaller number of constraints, and also lower input dimensions. To simulate a lower dimensional version of the capacity equation, we can consider only the fist n_x input variables and keep the remaining variables fixed. We consider $n_x = 3, 4$ and 5 where we fix $x_i = 0.5$ for all $i > n_x$. For each of these scenarios we will consider n_x and $n_x - 1$ number of constraints. We let n_c denote the number of constraints, where using n_c constraints means that the bound on $\partial f/\partial x_i$ is included for $i = 1, \ldots, n_c$.

In each experiment we start by generating a training set of $N = 5n_x$ or $N = 10n_x$ LHS samples from $[0, 1]^{n_x}$. As in the previous example in Section 4.2.2, we fit a zero mean GP using a Matérn 5/2 covariance function and plug-in hyperparameters by MLE. We search over a candidate set consisting of 2500 uniform samples from $[0, 1]^{n_x}$ iteratively to update the set of virtual observation locations, until the constraint probability at all locations in the candidate set, and for each constraint, is at least 0.7. To check whether this is a reasonable stopping criterion we finish by minimizing the constraint probability for each constraint, using the differential evolution (Storn and Price, 1997) global optimization algorithm available in (SciPy Jones et al., 2001–).

Table 3 shows the results for different combinations of input dimensionality n_x, number of constraints n_c and number of training samples N, where the results in each row is computed from 100 experiments. As in the previous example we report $p(C|Y)$, PVA, Q^2 and AWoCI, and the CPU time spent generating samples for prediction (T_s). We also report the average CPU time used in the search for a new virtual observation location and

n_x	n_c	N	N_v	T_v	$p(C\|Y)$	$p_{c,\min}$	T_s	PVA	Q^2	AWoCI
3	2	15	3.6	0.6	2.6E-01	0.79	0.05	0.94 (0.89)	0.95 (0.95)	3.9 (6.2)
3	2	30	3.5	0.6	2.5E-01	0.78	0.04	0.89 (0.87)	0.97 (0.97)	3.0 (4.8)
3	3	15	5.8	0.9	1.2E-01	0.74	0.09	1.47 (1.23)	0.95 (0.95)	3.7 (6.1)
3	3	30	3.9	0.9	2.2E-01	0.76	0.04	0.79 (0.79)	0.97 (0.97)	3.1 (5.0)
4	3	20	11.8	0.9	1.5E-02	0.67	0.19	1.40 (1.29)	0.87 (0.92)	5.5 (9.4)
4	3	40	11.7	0.9	6.6E-03	0.71	0.18	0.51 (0.52)	0.97 (0.97)	4.1 (6.9)
4	4	20	13.6	1.2	6.9E-03	0.65	0.49	1.56 (1.31)	0.91 (0.91)	5.5 (9.6)
4	4	40	12.8	1.2	2.7E-03	0.69	0.19	0.50 (0.48)	0.97 (0.97)	4.0 (6.7)
5	4	25	14.8	1.2	6.3E-03	0.66	0.22	1.03 (1.08)	0.85 (0.83)	8.3 (14.3)
5	4	50	17.4	1.2	1.2E-03	0.66	0.26	0.73 (0.78)	0.90 (0.90)	6.8 (11.5)
5	5	25	15.5	1.5	3.1E-03	0.65	0.24	1.12 (1.10)	0.82 (0.81)	8.4 (14.4)
5	5	50	20.2	1.6	1.1E-03	0.61	0.35	0.67 (0.77)	0.90 (0.90)	6.5 (11.3)

Table 3: Average values from 100 experiments with input dimensionality n_x, number of constraints n_c and number of training samples N. Values in parenthesis correspond to the unconstrained model. Here $p_{c,\min}$ is the minimum of the constraint probability for any constraint over the entire domain after a total of N_v virtual observation locations have been included. T_v is the average CPU time in seconds used to find each of the N_v points using 10^3 samples, and T_s is the CPU time in seconds used to generate 10^4 samples of the final model for prediction.

the minimum constraint probability, $p_{c,\min} = \min_{i=1,\dots n_c} \min_{\mathbf{x}\in[0,1]^{n_x}} \hat{p}_{c,i}(\mathbf{x})$ (13), computed with differential evolution. Here we make use of 10^3 samples to compute the estimate $\hat{p}_{c,i}(\mathbf{x})$, whereas 10^4 samples are used for the final prediction.

From Table 3 we first notice that the number of virtual observation locations (N_v) determined by the searching algorithm is fairly low. One might interpret this as an indication that the unconstrained GP produces samples that are likely to agree with the monotonicity constraints, except for at a few locations. As a result, computation that involve sampling from the truncated multivariate Gaussian is efficient. Still, we see that inclusion of the constraints has an effect on uncertainty estimates as the AWoCI is reduced by a factor of around 1.6 in each experiment, whereas PVA and Q^2 are fairly similar for the unconstrained and constrained model overall. We also notice that the smallest constraint probability found in the domain using a global optimization technique is reduced when the number of constraints or dimensionality is increased. This is expected, as we only considered a finite candidate set and not the entire domain when searching for the location minimizing the constraint probability. Hence, if we really want to achieve a minimal constraint probability larger than 0.7 in 5 dimensions, more than 2500 samples in the candidate set would be needed with this strategy, or a global optimizer could be used to identify the remaining virtual observation locations needed.

For the application considered in this example, where uncertainty in the prediction is key to risk assessment, we argue that the effect the constraints have on uncertainty estimates makes the inclusion of constraints worthwhile. Modern engineering methodologies that

make use of capacity predictions as the one illustrated in this example are usually derived in the context of Structural Reliability Analysis (SRA), where the capacity is combined with a probabilistic representation of load (in this case differential pressure) to estimate the probability of failure (Madsen et al., 2006).

Alternative methods based on conservative estimates to ensure sufficient safety margin between load and capacity are also common. For the application considered herein, this would typically mean using a lower percentile instead of the posterior mean in order to represent a conservative capacity. The inclusion of constraints can therefore help to avoid unnecessary conservatism due to unphysical scenarios, that are not realistic but have positive probability in the unconstrained model.

Finally, we note that the constraints used in this example are not from differentiating the equation used as stand-in for experiments, but from knowledge related to the underlying physical phenomenon. The constraints therefore remain applicable, were the experiments to come from physical full-scale tests. This naturally also holds in applications to computer code emulation, where we would set the noise term ε to zero in this example if we were to assume that the capacity experiments came from a numerical (FEA) simulation. With results from this type of numerical simulation, a noise parameter is usually added to the simulation output as well, to represent model uncertainty as the numerical simulation is not a perfect representation of the real physical phenomenon. Very often the model uncertainty is represented by a univariate Gaussian. An interesting alternative here is to instead account for the model uncertainty as observational noise in the GP, where the use of constraints may help to obtain a more realistic model uncertainty as well.

5. Discussion

The model presented in this paper provides a consistent approach to GP regression under multiple linear constraints. The computational framework used is based on a sampling scheme which is exact in the limit. However, sampling strategies like the one in this paper can be too numerically demanding as opposed to approximation methods such as Laplace approximations, variational Bayesian inference, expectation propagation etcetera. The choice of using a sampling-based approach came from the author's intended use, which relates to machine learning for high-risk and safety-critical engineering applications (Agrell et al., 2018). For these applications, a proper treatment of uncertainty with respect to risks and the overall reliability of the system under consideration is essential. Making predictions based on past observations in this setting is challenging, as the consequence of wrong predictions may be catastrophic. In addition, critical consequences often relate to infrequent or low probability events, where relevant data is naturally scarce. However, there is usually additional knowledge available, and todays methods for assessing risk tend to rely heavily on understanding the underlying physical phenomenon. We gave an example in Section 4.2.3 considering prediction of the burst capacity of a pipeline, that may serve as a component in a larger model of system reliability. Such models are often graphical, e.g. Bayesian networks, that are derived from known causal dependencies. In this scenario it is essential that the accuracy of numerical estimation- or approximation methods can be assessed. In the case where simulation-based methods cannot be used due to computational limitations, they still serve as a useful benchmark that can help in the development and assessment of

suitable approximation-based algorithms. As for the simulation scheme in this paper, the only computational burden lies in sampling from a truncated multivariate Gaussian. As this is a fairly general problem, multiple good samplers exist for this purpose. We found the method of Botev (2017) to work particularly well for our applications, as it provides exact sampling in a relevant range of dimensions where many alternative sampling schemes fail. Based on a comparison made by López-Lopera et al. (2018), we see that the method based on Hamiltonian Monte Carlo by Pakman and Paninski (2012) may also be appropriate.

As we discuss briefly in Section 3.3, estimation of hyperparameters becomes challenging when the term $p(C|Y, \theta)$ enters the likelihood. Moreover, as our approach is based on the use of virtual observation locations, we are aware that the task of estimating or optimizing model hyperparameters in general is not well defined. This is because the likelihood depends both on the hyperparameters and the set of virtual observation locations (Eq. 8). This problem is neglected in the literature on shape-constrained GPs, where it is either assumed that the virtual observation locations are known a priori (for low input dimension selecting a space filling sufficiently dense design is unproblematic), or the hyperparameters are addressed independently of these. To our knowledge the problem of simultaneously estimating hyperparameters and virtual observation locations has not yet been addressed. A rather simplistic approach is to iterate between estimating hyperparameter and the set of virtual observation locations. However, for higher input dimensions this might be problematic altogether, in which case sparse approximations may be needed to deal with a large set of virtual observation locations. In this setting, it might be more fruitful to view the virtual observation locations as additional hyperparameters, in a model approximating the posterior corresponding to an sufficiently dense set of virtual observation locations, e.g. as in the inducing points framework for scaling GPs to large data sets (de G. Matthews et al., 2016). This is a topic of further research.

With the approach in this paper, we make use of the probability $p(C|Y)$, which is interesting in its own for investigating whether constraints such as e.g. monotonicity are likely to hold given a set of observations. Alternatively, inference on the constraint noise parameter σ_v can provide similar type of information. Ideally, we choose a small fixed value for σ_v to avoid numerical instability, as discussed in Section 3.8. But in extreme cases, with conflicting constraints or observations that contradict constraints with high probability, the model may still experience numerical issues. We argue that models that 'break' under these circumstances are preferred as it reveals that either 1) there is something wrong with the observations, or 2) there is something wrong with the constraints and hence our knowledge of the underlying phenomenon (Agrell et al., 2018). It would nevertheless be better if more principled ways of investigating such issues were available. In our experiments we observed that the conditional likelihood, $p(Y|C)$, in general is decreasing as a function of σ_v, whereas this was not the case for an invalid constraint assuming a monotonic *decreasing* function in Example 1. Hence, σ_v might provide useful information in this manner. The estimated partial constraint probabilities $\hat{p}_{c,i}(\mathbf{x})$ can also be useful for revealing such issues, for instance by monitoring the intermediate minimum values p_i^* computed in Algorithm 7 as new virtual observation locations are added.

Finally, we note that as the model presented in this paper relies on conditioning on a transformed GP with values in $\mathbb{R}^{n_c}$, it could be extended to multi-output GPs over functions

$\mathrm{f}:\mathbb{R}^{n_x}\to\mathbb{R}^{n_y}$ in a natural way. But for non-Gaussian likelihoods, or applications with large or high-dimensional data, other approximation based alternatives are needed.

Acknowledgments

This work has been supported by grant 276282 from the Norwegian Research Council and DNV GL Group Technology and Research. The research is part of an initiative on applying constraints based on phenomenological knowledge in probabilistic machine learning for high-risk applications, and the author would like to thank colleagues at DNV GL and the University of Oslo for fruitful discussions on the topic. A special thanks to Arne B. Huseby, Simen Eldevik, Andreas Hafver, and the editor and reviewers of JMLR for insightfull comments that have greatly improved the paper.

Appendix A. Proof of Lemma 1

Proof. We start by observing that $(\mathbf{f}^*,\widetilde{C},Y)$ is jointly Gaussian with mean and covariance

$$\mathbb{E}([\mathbf{f}^*,\widetilde{C},Y]^T)=[\mu^*,\mathcal{L}\mu^v,\mu]^T, \tag{17}$$

$$\operatorname{cov}([\mathbf{f}^*,\widetilde{C},Y]^T)=\begin{bmatrix}K_{X^*,X^*} & K_{X^*,X^v}\mathcal{L}^T & K_{X^*,X}\\ \mathcal{L}K_{X^v,X^*} & \mathcal{L}K_{X^v,X^v}\mathcal{L}^T+\sigma_v^2 I_{N_v} & \mathcal{L}K_{X^v,X}\\ K_{X,X^*} & K_{X,X^v}\mathcal{L}^T & K_{X,X}+\sigma^2 I_N\end{bmatrix}. \tag{18}$$

By first conditioning on Y we obtain

$$\begin{bmatrix}\mathbf{f}^*\\ \widetilde{C}\end{bmatrix}\Bigg| Y\sim\mathcal{N}\left(\begin{bmatrix}\mu^*+A_2(Y-\mu)\\ \mathcal{L}\mu^v+A_1(Y-\mu)\end{bmatrix},\begin{bmatrix}B_2 & B_3\\ B_3^T & B_1\end{bmatrix}\right), \tag{19}$$

for $A_1=(\mathcal{L}K_{X^v,X})(K_{X,X}+\sigma^2 I_N)^{-1}$, $A_2=K_{X^*,X}(K_{X,X}+\sigma^2 I_N)^{-1}$, $B_1=\mathcal{L}K_{X^v,X^v}\mathcal{L}^T+\sigma_v^2 I_{N_v}-A_1K_{X,X^v}\mathcal{L}^T$, $B_2=K_{X^*,X^*}-A_2K_{X,X^*}$, and $B_3=K_{X^*,X^v}\mathcal{L}^T-A_2K_{X,X^v}\mathcal{L}^T$.

Conditioning on $\widetilde{C}$ then gives

$$\mathbf{f}^*|Y,\widetilde{C}\sim\mathcal{N}\left(\mu^*+A(\widetilde{C}-\mathcal{L}\mu^v)+B(Y-\mu),\Sigma\right), \tag{20}$$

for $A=B_3B_1^{-1}$, $B=A_2-AA_1$ and $\Sigma=B_2-AB_3^T$.

Similarly, we may derive $\widetilde{C}|Y$ by observing that the joint distribution of $\widetilde{C},Y$ is given by removing the first row in (17) and the first row and column in (18). Hence,

$$\widetilde{C}|Y\sim\mathcal{N}\left(\mathcal{L}\mu^v+A_1(Y-\mu),B_1\right). \tag{21}$$

The constrained posterior of $\widetilde{C}$ is obtained by applying the constraint C to the posterior, and hence $\widetilde{C}|Y,C$ becomes a truncated Gaussian with the same mean and variance as in (21), and the bounds $a(X^v)$ and $b(X^v)$ given by C. Similarly, $\mathbf{f}^*|Y,C$ is obtained by replacing $\widetilde{C}$ in (20) with $\widetilde{C}|Y,C$. Finally, the probability $p(C|Y)$ is just the probability that $\widetilde{C}|Y$ given in (21) falls within the bounds given by C, and the unconstrained distribution remains the same as (2). ■

Appendix B. Proof of Lemma 2

Proof. The equations in Lemma 2 can be verified by simply inserting L, v_1 and v_2 and check against the expressions in Lemma 1. We show this for A_1 and B_1, and the results for the remaining matrices are proved by applying the same procedures. In order to factorize B_1, we use that B_1 is the covariance matrix of a Gaussian random variable (see Equation 21 in Appendix A), and must therefore be symmetric and positive definite.

To show that $A_1 = (L^T \setminus v_1)^T$ we use that $v_1 = L \setminus K_{X,X^v}\mathcal{L}^T \Rightarrow Lv_1 = K_{X,X^v}\mathcal{L}^T$. Hence,

$$\begin{aligned}
&A_1 = (L^T \setminus v_1)^T \\
&\Rightarrow L^T A_1^T = v_1 = L \setminus K_{X,X^v}\mathcal{L}^T \\
&\Rightarrow LL^T A_1^T = K_{X,X^v}\mathcal{L}^T \\
&\Rightarrow A_1 = ((LL^T)^{-1}K_{X,X^v}\mathcal{L}^T)^T = (\mathcal{L}K_{X^v,X})(K_{X,X} + \sigma^2 I_N)^{-1},
\end{aligned}$$

where we have used that $(K_{X,X^v}\mathcal{L}^T)^T = \mathcal{L}K_{X^v,X}$ and $LL^T = K_{X,X} + \sigma^2 I_N$.

To show that $B_1 = \mathcal{L}K_{X^v,X^v}\mathcal{L}^T + \sigma_v^2 I_{N_v} - v_1^T v_1$ we need to show that $v_1^T v_1 = A_1 K_{X,X^v}\mathcal{L}^T$, which is trivial

$$\begin{aligned}
v_1^T v_1 &= (L^{-1}K_{X,X^v}\mathcal{L}^T)^T(L^{-1}K_{X,X^v}\mathcal{L}^T) \\
&= \mathcal{L}K_{X^v,X}(LL^T)^{-1}K_{X,X^v}\mathcal{L}^T \\
&= A_1 K_{X,X^v}\mathcal{L}^T.
\end{aligned}$$

■

Appendix C. Algorithm for Finding Virtual Observation Locations based on Individual Sub-operators

We present the details of the algorithm for finding virtual observation locations introduced in Section 3.5. Here we let $\mathcal{L}$ be a linear operator defined by the column vector $[\mathcal{F}_1, \ldots, \mathcal{F}_k]$, where $\mathcal{F}_i$ produces functions from $\mathbb{R}^{n_x}$ to $\mathbb{R}$, subjected to an interval constraint $[a_i(\mathbf{x}), b_i(\mathbf{x})]$. We would like to impose constraints related to the i-th sub-operator only at locations where $p(\mathcal{F}_i f(\mathbf{x}) \notin [a_i(\mathbf{x}), b_i(\mathbf{x})])$ is not sufficiently small. For this we let X^v be the concatenation of the matrices $X^{v,1}, \ldots, X^{v,k}$ and define $\mathcal{L}^T f(X^v) = [\mathcal{F}_1^T f(X^{v,1}), \ldots, \mathcal{F}_k^T f(X^{v,1})]^T$. The matrices needed to make use of Lemma 1 and Lemma 2 are $\mathcal{L}\mu^v$, $K_{X,X^v}\mathcal{L}^T$, $K_{X^*,X^v}\mathcal{L}^T$, and $\mathcal{L}K_{X^v,X^v}\mathcal{L}^T$. Using that $\mathcal{F}_i f(X^v) = \mathcal{F}_i f(X^{v,i})$, these are given by

$$\mathcal{L}\mu^v = \begin{bmatrix} \mathcal{F}_1\mu(X^{v,1}) \\ \vdots \\ \mathcal{F}_k\mu(X^{v,k}) \end{bmatrix}, \quad K_{X,X^v}\mathcal{L}^T = \begin{bmatrix} K_{X,X^{v,1}}\mathcal{F}_1^T \\ \vdots \\ K_{X,X^{v,k}}\mathcal{F}_k^T \end{bmatrix},$$

where $K_{X^*,X^v}\mathcal{L}^T$ also is given by the above equation for $X = X^*$. Finally, $\mathcal{L}K_{X^v,X^v}\mathcal{L}^T$ is the block matrix with blocks

$$(\mathcal{L}K_{X^v,X^v}\mathcal{L}^T)_{i,j} = \mathcal{F}_i K_{X^{v,i},X^{v,j}}\mathcal{F}_j^T.$$

We want to improve the algorithm in Section 3.4 for finding the set of virtual observation locations by considering each sub-operator individually. To do this we make use estimated partial constraint probabilities (given in (13) and restated below).

$$\hat{p}_{c,i}(\mathbf{x}) = \frac{1}{m}\sum_{j=1}^{m} P\left(a_i(\mathbf{x}) - \nu < (\mathcal{L}f(\mathbf{x})|Y, C_j)_i < b_i(\mathbf{x}) + \nu\right),$$

where $(\mathcal{L}f(\mathbf{x})|Y, C_j)_i$ is the univariate Normal distribution given by the i-th row of $(\mathcal{L}f(\mathbf{x})|Y, C_j)$ and $C_1, \ldots, C_m$ are m samples of $\mathbf{C}$ given in (6) as before. For the individual sub-operators $\mathcal{F}_i$, the set of virtual observations X_i^v needed to ensure that $\hat{p}_{c,i}(\mathbf{x}) \geq p_{\text{target}}$ can then be found using the following algorithm.

Algorithm 7 *Finding locations of virtual observations X_i^v s.t. $\hat{p}_{c,i}(\boldsymbol{x}) \geq p_{target}$ for all $\boldsymbol{x} \in \Omega$ and all sub-operators $\mathcal{F}_1, \ldots, \mathcal{F}_k$.*

1. *Compute $L = Chol(K_{X,X} + \sigma^2 I_N)$.*
2. *Until convergence do:*
   (a) *If $X^v \neq \emptyset$ compute A_1 and B_1 as defined in Lemma 2, and generate m samples $C_1, \ldots, C_m$ of $\boldsymbol{C}$ given in* (6).
   (b) *If $X^v = \emptyset$ compute $(\boldsymbol{x}_i^*, p_i^*) = (\arg\min p_{c,i}(\boldsymbol{x}), p_{c,i}(\boldsymbol{x}^*))$. Otherwise compute $(\boldsymbol{x}_i^*, p_i^*) = (\arg\min \hat{p}_{c,i}(\boldsymbol{x}), \hat{p}_{c,i}(\boldsymbol{x}^*))$, for all $i = 1, \ldots, k$ with $\hat{p}_{c,i}$ defined as in* (13) *using the samples generated in step (a).*
   (c) *Let $(\boldsymbol{x}^*, p^*, j)$ correspond to the smallest probability: $p^* = p_j^* = min_i\ p_i^*$.*
   (d) *Terminate if $p^* \geq p_{target}$, otherwise update $X_j^v \to X_j^v \cup \{\boldsymbol{x}^*\}$.*

Appendix D. Proof of Lemma 4

Proof. This follows exactly from the proofs of Lemma 1 and Lemma 2 by replacing $\mathbf{f}^* \to \mathcal{L}f(\mathbf{x}^*)$, which implies $\mu^* \to \mathcal{L}\mu^*$, $K_{X^*,X} \to \mathcal{L}K_{\mathbf{x}^*,X}$, $K_{X^*,X^*} \to \mathcal{L}K_{\mathbf{x}^*,\mathbf{x}^*}\mathcal{L}^T$ and $K_{X^*,X^v}\mathcal{L}^T \to \mathcal{L}K_{\mathbf{x}^*,X^v}\mathcal{L}^T$.

■

Appendix E. Proof of Corollary 6

Proof. We show the derivation of the expectation and covariance of $\mathbf{f}^*|Y, C$ as the derivations for $\mathcal{L}f(\mathbf{x}^*)|Y, C$ are equivalent. From Lemma 1 we have that

$$\mathbf{f}^*|Y, C \sim \mathcal{N}(\mu^* + A(\mathbf{C} - \mathcal{L}\mu^v) + B(Y - \mu), \Sigma).$$

If we let ν, Γ be the expectation and covariance of $\mathbf{C}$, then

$$\begin{aligned}\mathbb{E}[\mathbf{f}^*|Y, C] = \mathbb{E}_{\mathbf{C}}\left[\mathbb{E}[\mathbf{f}^*|Y, \mathbf{C}]\right] &= \mathbb{E}_{\mathbf{C}}\left[\mu^* + A(\mathbf{C} - \mathcal{L}\mu^v) + B(Y - \mu)\right] \\ &= \mu^* + A(\nu - \mathcal{L}\mu^v) + B(Y - \mu),\end{aligned}$$

and

$$\begin{aligned}\operatorname{cov}[\mathbf{f}^*|Y,C] &= \mathbb{E}_{\mathbf{C}}\left[\operatorname{cov}[\mathbf{f}^*|Y,\mathbf{C}]\right] + \operatorname{cov}_{\mathbf{C}}[\mathbb{E}[\mathbf{f}^*|Y,\mathbf{C}]] \\ &= \mathbb{E}_{\mathbf{C}}[\Sigma] + \operatorname{cov}_{\mathbf{C}}[\mu^* + A(\mathbf{C} - \mathcal{L}\mu^v) + B(Y-\mu)] \\ &= \Sigma + \operatorname{cov}_{\mathbf{C}}[A\mathbf{C}] = \Sigma + A\Gamma A^T.\end{aligned}$$

■

References

Petter Abrahamsen and Fred Espen Benth. Kriging with inequality constraints. *Mathematical Geology*, 33(6):719–744, Aug 2001.

Robert J. Adler. *The Geometry of Random Fields*. Wiley series in probability and mathematical statistics. Probability and mathematical statistics. J. Wiley, 1981.

Christian Agrell, Simen Eldevik, Andreas Hafver, Frank Børre Pedersen, Erik Stensrud, and Arne Huseby. Pitfalls of machine learning for tail events in high risk environments. In Stein Haugen, Anne Barros, Coen van Gulijk, Trond Kongsvik, and Jan Erik Vinnem, editors, *Safety and Reliability Safe Societies in a Changing World - Proceedings of ESREL 2018*. CRC Press, june 2018.

Rafael Amaya, Mauricio Sanchez-Silva, Emilio Bastidas-Arteaga, Franck Schoefs, and Felipe Munoz. Reliability assessments of corroded pipelines based on internal pressure - A review. *Engineering Failure Analysis*, 98, 01 2019.

Jian An and Art Owen. Quasi-regression. *Journal of Complexity*, 17(4):588 – 607, 2001.

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

François Bachoc. Cross validation and maximum likelihood estimations of hyper-parameters of gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, 66:55 – 69, 2013.

François Bachoc, Agnes Lagnoux, and Andrés F. López-Lopera. Maximum likelihood estimation for gaussian processes under inequality constraints. working paper or preprint, August 2018.

Zdravko I. Botev. The normal law under linear restrictions: simulation and estimation via minimax tilting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(1):125–148, 2017.

Sébastien Da Veiga and Amandine Marrel. Gaussian process modeling with inequality constraints. *Annales de la faculté des sciences de Toulouse Mathématiques*, 21(3):529–555, 4 2012.

Sébastien Da Veiga and Amandine Marrel. Gaussian process regression with linear inequality constraints. working paper or preprint, 10 2015.

Alexander G. de G. Matthews, James Hensman, Richard Turner, and Zoubin Ghahramani. On sparse variational methods and the kullback-leibler divergence between stochastic processes. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 231–239. PMLR, 09–11 May 2016.

DNV GL. Reccomended Practice: Corroded pipelines DNVGL-RP-F101. *DNV GL, Høvik, Norway*, 2017.

Simen Eldevik, Christian Agrell, Andreas Hafver, and Frank B. Pedersen. AI + Safety: Safety implications for artificial intelligence and why we need to combine casual- and data-driven models. 08 2018. [Online position paper by DNV GL Group Technology and Research; https://ai-and-safety.dnvgl.com/, posted 28-August-2018].

Alan Genz. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1(2):141–149, 1992.

Alan Genz. Comparison of methods for the computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 11, 04 1997.

Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521 (7553):452–459, 2015.

Shirin Golchi, D R. Bingham, H Chipman, and David Campbell. Monotone emulation of computer experiments. *SIAM/ASA Journal on Uncertainty Quantification*, 3:370–392, 01 2015.

Carl Jidling, Niklas Wahlström, Adrian Wills, and Thomas B Schön. Linearly constrained gaussian processes. pages 1215–1224. Curran Associates, Inc., 2017.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

Marc C. Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

George S. Kimeldorf and Grace Wahba. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *Ann. Math. Statist.*, 41(2):495–502, 04 1970.

Jack P. C. Kleijnen and Wim C. M. Van Beers. Monotonicity-preserving bootstrapped kriging metamodels for expensive simulations. *JORS*, 64:708–717, 2013.

Jayesh H. Kotecha and Petar M. Djuric. Gibbs sampling approach for generation of truncated multivariate gaussian random variables. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, volume 3, pages 1757–1760 vol.3, March 1999.

Peter Lenk and Taeryon Choi. Bayesian analysis of shape-restricted functions using gaussian process priors. *Statistica Sinica*, 27:43–69, 2017.

Lizhen Lin and David B. Dunson. Bayesian monotone regression using gaussian process projection. *Biometrika*, 101(2):303–317, 2014.

Andrés López-Lopera, François Bachoc, Nicolas Durrande, and Olivier Roustant. Finite-dimensional gaussian approximation with linear inequality constraints. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1224–1255, 2018.

Hassan Maatouk and Xavier Bay. Gaussian process emulators for computer experiments with inequality constraints. *Mathematical Geosciences*, 49(5):557–582, Jul 2017.

Hassan Maatouk, Laurence Grammont, and Xavier Bay. Generalization of the kimeldorf-wahba correspondence for constrained interpolation. *Electronic Journal of Statistics*, 10 (1):1580–1595, 2016.

Henrik O. Madsen, Steen Krenk, and Niels C. Lind. *Methods of Structural Safety*. Dover Civil and Mechanical Engineering Series. Dover Publications, 2006.

Georges Matheron. The intrinsic random functions and their applications. *Advances in Applied Probability*, 5(3):439–468, 1973.

Anna Michalak. A gibbs sampler for inequality-constrained geostatistical interpolation and inverse modeling. *Water Resour. Res*, 44, 09 2008.

Ari Pakman and Liam Paninski. Exact hamiltonian monte carlo for truncated multivariate gaussians. *Journal of Computational and Graphical Statistics*, 23, 08 2012.

Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill Higher Education, 4 edition, 2002.

Pritam Ranjan, Ronald Haynes, and Richard Karsten. A computationally stable approach to gaussian process interpolation of deterministic computer simulation data. *Technometrics*, 53, 03 2010.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

Jaakko Riihimki and Aki Vehtari. Gaussian processes with monotonicity information. *Journal of Machine Learning Research - Proceedings Track*, 9:645–652, 01 2010.

Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments. *Statist. Sci.*, 4(4):409–423, 11 1989.

Simo Särkkä. Linear operators and stochastic partial differential equations in gaussian process regression. In Timo Honkela, Włodzisław Duch, Mark Girolami, and Samuel Kaski, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, pages 151–158, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

Gudfinnur Sigurdsson, Espen H. Cramer, Ola H. Bjørnøy, B. Fu, and D. Ritchie. Background to DNV RP-F101 Corroded pipelines. In *Proceedings of the* 18^{th} *international conference on offshore mechanics and arctic engineering, OMAE, Newfoundland, Canada.* American Society of Mechanical Engineers, U.S., 1999.

Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4): 341–359, Dec 1997.

Philip Duncan Thompson. Optimum smoothing of two-dimensional fields. *Tellus*, 8(3): 384–393, 1956.

Xiaojing Wang and James O. Berger. Estimating shape constrained functions using gaussian processes. *SIAM/ASA Journal on Uncertainty Quantification*, 4:1–25, 01 2016.

Eun-Hye Yoo and Phaedon C. Kyriakidis. Area-to-point kriging with inequality-type data. *Journal of Geographical Systems*, 8(4):357–390, Oct 2006.