Monotonic Metamodels for Deterministic Computer Experiments

Matthias Hwai Yong Tan
Department of SEEM
City University of Hong Kong
Hong Kong
(mathtan@cityu.edu.hk)

In deterministic computer experiments, it is often known that the output is a monotonic function of some of the inputs. In these cases, a monotonic metamodel will tend to give more accurate and interpretable predictions with less prediction uncertainty than a nonmonotonic metamodel. The widely used Gaussian process (GP) models are not monotonic. A recent article in Biometrika offers a modification that projects GP sample paths onto the cone of monotonic functions. However, their approach does not account for the fact that the GP model is more informative about the true function at locations near design points than at locations far away. Moreover, a grid-based method is used, which is memory intensive and gives predictions only at grid points. This article proposes the weighted projection approach that more effectively uses information in the GP model together with two computational implementations. The first is isotonic regression on a grid while the second is projection onto a cone of monotone splines, which alleviates problems faced by a grid-based approach. Simulations show that the monotone B-spline metamodel gives particularly good results. Supplementary materials for this article are available online.

KEY WORDS: B-spline metamodels; Design and analysis of experiments; Deterministic simulations; Gaussian process projections.

1. INTRODUCTION

When a computer model is time-consuming to run, a metamodel can be constructed to serve as a quick-to-evaluate approximation. In many engineering problems, it is known from physical considerations that the output is a monotonic function of some of the inputs. For example, the strength of a welding joint is increasing in the size of the weld. The critical buckling stress of a two-bar truss increases with the diameter and Young’s modulus of the truss members. When a solid body is immersed in a fluid with a higher temperature, the temperature at any point in the body increases with time and the convective heat transfer coefficient. In these cases, a metamodel that uses the monotonicity information will tend to be more accurate and interpretable than a nonmonotonic metamodel. Moreover, prediction intervals constructed with a monotonic metamodel tend to be tighter because nonmonotonic predictions are excluded. Unfortunately, monotonicity information is rarely exploited in computer experiments. This article proposes statistical approaches for building monotonic metamodels in deterministic computer experiments.

For deterministic computer experiments with more than one input, there are three existing methods that can be employed to construct monotonic metamodels that give both point and interval predictions. The first is the L2 projection approach proposed by Lin and Dunson (2014), which projects the GP sample paths onto the cone of monotone functions. Although Lin and Dunson (2014) discussed the possibility of a weighted projection approach, they focused on the unweighted projection. The second is the rearrangement approach proposed by Chernozhukov, Fernandez-Val, and Galichon (2009). This approach transforms the point predictor and credible limits of a GP into quantile functions induced by a uniform distribution for the inputs. Computational implementation of both approaches requires working with a grid in the input space. Thus, predictions at points not on a grid require a monotonic interpolator that interpolates the predictions on the grid. Selection of grid and interpolator is important to achieve small discretization error without unnecessarily high computational requirements. Lin and Dunson (2014) and Chernozhukov, Fernandez-Val, and Galichon (2009) did not address this problem. Predictors obtained from the projection and rearrangement procedures can have undesirable characteristics. These include interval predictions that are wide at the design points, point predictions with sharp corners (nondifferentiability), and prediction interval limits with sharp corners at nondesign points. Interval predictions that are wide at the design points are undesirable for deterministic data. Nondifferentiability is undesirable when the true function is known to be differentiable. The third method is the minimax predictor proposed by Beliakov (2005). This predictor often yields poor prediction accuracy, and prediction bounds that are too narrow at some regions and too wide elsewhere.

Other approaches for exploiting monotonicity information in computer experiments were proposed by Riihimäki and Vehtari (2010), Wang (2012), and Golchi et al. (2015). However, these methods only encourage the derivative of the GP to be positive at a finite set of points. The GP can be highly nonmonotonic if the set of points is small but computation can be intractable otherwise. There is a vast literature on monotonic regression but most of the methods are for observational data with large random
errors and one or two independent variables. Important methods include the pooled adjacent violators (PAV) algorithm (Barlow et al. 1972), I-splines (Ramsay 1988), B-splines (Brezger and Steiner 2008; He and Ng 1999), Bernstein bases (Tan 2016), and monotonic Kernel regression (Du, Parmeter, and Racine 2013).

In deterministic computer experiments, the output is observed without random error. Because of this, the output is known at the design points and there is less uncertainty about the response near the design points due to the known continuity or differentiability of the function. Motivated by this observation, this article proposes a weighted projection approach to construct monotonic metamodels from GP models. The weights are the inverse of the GP variances truncated to avoid infinite values. Thus, higher weights are given to regions near design points and the weighted projection would tend to be closer to the GP near the design points than the unweighted projection. Two computational implementations are considered. The first method is weighted isotonic regression on a grid. We propose a monotone interpolator and a grid selection method that aim to achieve low discretization error with a small number of grid points. In the second method, the sample paths of the GP model are projected onto a cone of monotonic spline functions. The method constructs a monotone B-spline model that approximates the weighted projection of the GP with a small number of model terms. The model is differentiable if the degrees of the B-spline bases are greater than one and is well defined everywhere. The proposed methods are computationally tractable and provide point and interval predictions for problems with several inputs. For both methods, the point and interval predictions can be forced to nearly interpolate the data points for problems with one and two inputs using small convergence tolerances. In higher dimensions, it is difficult to achieve small prediction interval lengths at the data points without incurring high computational burden. Thus, both methods are tuned to yield noninterpolating interval predictions. However, the point predictions are often nearly interpolating and the methods produce better accuracy and shorter prediction intervals than the unweighted projection.

The remainder of the article is organized as follows: Section 2 reviews GP modeling. Section 3 describes the projection approach introduced by Lin and Dunson (2014). The section also presents the weighted projection approach and a grid-based computational implementation using isotonic regression. In Section 4, we propose the monotone B-spline metamodel. A real example is given in Section 5 and conclusions are given in Section 6.

2. REVIEW OF GAUSSIAN PROCESS MODELS

In GP modeling, the prior for the functional relationship \( F : [0, 1]^d \rightarrow \mathbb{R} \) between output and inputs is given by a continuous GP:

\[
Y(x) = \mu + G(x),
\]

where \( G(x) \) is a zero mean stationary GP. Given two points \( x_i \) and \( x_j \), the covariance of \( Y(x_i) \) and \( Y(x_j) \) is

\[
\text{cov}(Y(x_i), Y(x_j)) = \sigma^2 R(x_i, x_j | \theta),
\]

where \( R \) is the correlation function. In a computer experiment, the output is evaluated at \( n \) values of inputs in the design \( D = \{x^1, \ldots, x^n\} \subset [0, 1]^d \). This yields a vector \( Y = (Y^1, \ldots, Y^n)^T \) of observed outputs. The prior process is updated with the data, giving a posterior GP (Sacks et al. 1989; Curran et al. 1991)

\[
Y(\cdot) | (Y, \lambda, \sigma^2, \theta) \sim GP \left( \mu(\cdot | \lambda, \theta), C(\cdot | \sigma^2, \theta) \right),
\]

with mean function \( \mu(\cdot | \lambda, \theta) \) and covariance function \( C(\cdot | \sigma^2, \theta) \). The mean function is

\[
\mu(x | \lambda, \theta) = \lambda + r(x)^T R^{-1}(Y - \lambda 1), \tag{3}
\]

where \( r(x) = (R(x, x^1 | \theta), \ldots, R(x, x^n | \theta))^T \), \( R = (R(x_i, x_j | \theta))_{1 \leq i \leq n, 1 \leq j \leq n} \), and \( 1 \) is an \( n \times 1 \) vector of 1’s. The covariance function is given by

\[
C(x_i, x_j | \sigma^2, \theta) = \sigma^2 \left[ R(x_i, x_j | \theta) - r(x_i)^T R^{-1} r(x_j) \right]. \tag{4}
\]

The mean and covariance functions given by (3)–(4) depend on the parameters \( \lambda, \sigma^2, \) and \( \theta \), which can be estimated from the data. One approach to estimating \( \lambda, \sigma^2, \) and \( \theta \) is the maximum likelihood method. It turns out that given \( \lambda, \theta = \hat{\lambda}, \sigma^2 = \hat{\sigma}^2 = (Y - \hat{\lambda}1)^T R^{-1}(Y - \hat{\lambda}1)/n \) maximize the likelihood. Thus, the maximum likelihood estimate \( \hat{\theta} \) of \( \theta \) is obtained by minimizing

\[
\log(\hat{\sigma}^2) + \log(|R|). \tag{5}
\]

It is common to perform statistical inference on \( \hat{F} \) using \( \hat{\theta} \) in place of \( \theta \) and \( \hat{\sigma}^2 = \frac{\sigma^2}{n-1} \hat{\sigma}^2 \) (an unbiased estimator of \( \sigma^2 \)) in place of \( \sigma^2 \), that is, using the GP

\[
Y(\cdot) | (Y, \hat{\lambda}, \hat{\sigma}^2, \hat{\theta}) \sim GP \left( \hat{\mu}(\cdot | \hat{\lambda}, \hat{\theta}), \hat{C}(\cdot | \hat{\sigma}^2, \hat{\theta}) \right). \tag{6}
\]

This can be viewed as an empirical Bayes approach. In GP modeling of computer experiments, it is common to employ the product Matérn correlation function. In this article, we employ the product Matérn correlation function with smoothness parameter 1.5:

\[
R(x_1, x_2 | \theta) = \prod_{i=1}^d \left[ \exp(-\rho_i) \right](\rho_i + 1), \quad \rho_i = \sqrt{6} |x_{1i} - x_{2i}| / \theta_i, \tag{7}
\]

where \( x_1 = (x_{11}, \ldots, x_{1d}) \). This correlation function gives once differentiable sample paths.

3. PROJECTION AND WEIGHTED PROJECTION APPROACHES

In this section, we review the projection approach proposed by Lin and Dunson (2014) and introduce a weighted modification of their approach. The weighted isotonic regression on a grid method for implementing the proposed approach is discussed. We address two important implementation issues not addressed by Lin and Dunson (2014): choice of grid and interpolator. For simplicity, we consider building metamodels that are increasing in all inputs. It is straightforward to modify the approaches in this article to construct metamodels that are increasing in some of the inputs only.
Let $\mathcal{M}$ denote the closed convex cone of all increasing functions on $[0,1]^d$ and $Y$ denote the continuous posterior process given by (6). Lin and Dunson (2014) proposed a monotonic metamodel that is the unweighted projection of $Y$ onto $\mathcal{M}$:

$$P_M Y = \arg\min_{f \in \mathcal{M}} \|Y - f\|_w^2,$$  \hspace{1cm} (8)

where $\|Y - f\|_w^2 = \int_{[0,1]^d} [Y(x) - f(x)]^2 w(x) \, dx$. There are a few problems with this approach for deterministic computer experiments. First, it does not account for the fact that the GP model is inversely proportional to the posterior variance to construct a monotonic metamodel. The weighted projection of $\bar{Y}$ may yield credible intervals that look wide at the corners (which we call a cell) that contains $x$. We do not set $w(x)$ equal to the inverse of the posterior variance $\text{var}[Y(x)]$ to avoid infinite weights at the design points $x^1, \ldots, x^n$. Since $w(x)$ is very large (and maximized) at $x^1, \ldots, x^n$, the weighted projection $P_M Y$ should tend to nearly interpolate the data points $(x^1, Y^1), \ldots, (x^n, Y^n)$. This can be seen in Figure 1(e).

In practice, only a discrete approximation of $P_M Y$ can be computed. Suppose that the values of $Y$ on an $[L_1 \times \cdots \times L_d]$ grid are denoted by $\{Y(i_1, \ldots, i_d) : i_j = 1, \ldots, L_j, j = 1, \ldots, d\}$. Then, the values of $P_M Y$ on the grid are approximated by minimizing the weighted sum of squared discrepancies $\sum w(i_1, \ldots, i_d) [Y(i_1, \ldots, i_d) - f(i_1, \ldots, i_d)]^2$, where $w(i_1, \ldots, i_d) = 1/\max \{\text{var}[Y(i_1, \ldots, i_d)], B\}$, subject to the constraint that $f(i_1, \ldots, i_d)$ is increasing on the grid. This is the well-known isotonic regression problem. For the case of a single input, that is, $d = 1$, the solution can be efficiently obtained by applying the PAV algorithm (Barlow et al. 1972). An efficient way to compute the solution when $d > 1$ is the multivariate isotonic regression algorithm proposed by Dykstra and Robertson (1982). The algorithm, which is given in Appendix A, iteratively applies the PAV algorithm to obtain increasing predictions with respect to each input. To construct point and interval predictions, the algorithm is applied to a number of simulated GP sample paths. The mean of the projected sample paths is taken as the point prediction, and the tail quantiles are used to construct credible intervals.

For prediction at points not on the grid, Beliakov’s (2005) interpolator can be used but it gives poor results. We use the following interpolator. If prediction is desired at a point $x = (x_1, \ldots, x_d)$, we find the smallest hypercube with grid points as corners (which we call a cell) that contains $x$. We then transform the inputs so that the cell is $[0,1]^d$. Denote the transformed input by $(z_1, \ldots, z_d)$ and the isotonized values on the corners of the

Figure 1. Posterior mean and 98% credible intervals for GP model (a); point and interval predictions given by unweighted projection method (b); point and interval predictions given by rearrangement method (c); point and interval predictions given by Beliakov’s interpolator (d); point and interval predictions given by grid-based weighted projection method (e); and point and interval predictions given by the monotone B-spline metamodel (f). In each figure, the true function is plotted as a dotted line and the data points are plotted as circles.
cell by \( \hat{Y}(z_1, \ldots, z_d) \). Then, we predict at \( x \) using the model constructed from Bernstein bases

\[
\sum_{j=1}^{d} \sum_{i=0}^{1} \hat{Y}(i_1, \ldots, i_d) \zeta_j^i (1 - z_1)^{1-i} \ldots \zeta_j^d (1 - z_d)^{1-i}.
\]

(10)

Note that (10) interpolates the isotonized values at the corners of the cell. Due to the monotonicity preserving property of the Bernstein bases (Floater and Peña 1998), (10) is also increasing. It can be shown that the metamodel obtained with the above procedure is a continuous and increasing function of \( x \). However, it is nondifferentiable.

We select the grid as follows: Start with an \( L_1 \times \cdots \times L_d \) grid of equally spaced values \( \prod_{i=1}^{d} [0, 1/(L_i - 1), \ldots, 1] \). Increase \( L_1 \) by one and observe the relative change in prediction on a test set \( D_t \), that is, where relative change is defined as the mean absolute difference between predictions obtained with \( L_1 + 1 \) equally spaced values and \( L_1 \) equally spaced values for the first input divided by the range of predictions in the former case. This process is repeated for \( L_2, \ldots, L_d \). Suppose that changing \( L_j \) to \( L_j + 1 \) gives the maximum mean relative change for the sample paths. Then, we reset the value of \( L_j \) to \( L_j + 1 \). If this maximum change is small, then the procedure is stopped. Because it is time consuming to select the grid based on many simulated sample paths, we select a grid based on isotonizing the posterior mean, the 0.01 quantile, and the 0.99 quantile of the posterior GP. The grid refinement is stopped when the mean relative change in the projection of these three functions is less than 0.005. Note that the tail quantiles tend to be more non-monotonic than most sample paths. This tends to lead to the selection of a finer grid than necessary. The selected grid is employed for prediction by isotonizing a number of simulated sample paths and predicting with the interpolator (10).

4. MONOTONE METAMODELS BASED ON B-SPLINES

This section presents the spline model and linear constraints on the model coefficients used to achieve monotonicity. We also give a key result, which states that the projection of a continuous GP onto the cone of monotone splines converges in mean square to the projection of the GP onto the cone of monotone functions as the knots get dense.

B-splines are widely employed for function approximation because they have many useful properties (De Boor 2001). A useful property of these bases is that it is easy to enforce monotonicity with respect to the inputs. For \( i = 1, \ldots, d \), let \( (u'_1, \ldots, u'_{n_i}) \) be B-spline bases of degree \( K \) defined by knots \( \{\zeta^i_0, \ldots, \zeta^i_{n_i+K+1}\} \), where \( \zeta^i_0 = \cdots = \zeta^i_K = 0 < \zeta^i_{K+1} < \cdots < 1 = \zeta^i_{n_i+1} = \cdots = \zeta^i_{n_i+K+1} \). Note that \( u^i_j(x_i) \geq 0 \) and \( \sum_{j=0}^{n_i} u^i_j(x_i) = 1 \) for all \( x_i \in [0, 1] \). To approximate a function defined on \([0, 1]^d\), we can employ the tensor product bases \( \prod_{i=1}^{d} u^i_{j_i} : j_i = 0, \ldots, n_i; k = 1, \ldots, d \), that is,

\[
f(x|e) = \sum_{j=0}^{n_i} \cdots \sum_{j_d=0}^{n_d} c(j_1, \ldots, j_d) \prod_{i=1}^{d} u^i_{j_i}(x_i) = \sum_{j \in T} c_j u_j(x),
\]

(11)

where \( j = (j_1, \ldots, j_d) \), \( c_j = c(j_1, \ldots, j_d) \), \( T = \prod_{i=1}^{d} [0, 1, \ldots, n_i] \), and \( u_j = \prod_{i=1}^{d} u^i_{j_i} \). It follows from the results in Floater and Peña (1998) that the tensor product bases preserve axial monotonicity, that is, if \( c(j_1, \ldots, j_d) \) is increasing in \( j_i \) for all \((j_1, \ldots, j_{i-1}, j_i+1, \ldots, j_d)\), then \( f(x|e) \) is increasing in \( x_i \). When \( K = 1 \) or when \( K = 2 \) and \( d = 1 \), the converse also holds: if \( f(x|e) \) is increasing in \( x_i \), then \( c(j_1, \ldots, j_d) \) is increasing in \( j_i \) for all \((j_1, \ldots, j_{i-1}, j_i+1, \ldots, j_d)\). Thus, the spline function \( f(x|e) \) is increasing in \( x_1, \ldots, x_d \) if \( e \) satisfies the linear constraints:

\[
c(j_1, \ldots, j_{i-1}, j_i+1, \ldots, j_d) \\
\leq c(j_1, \ldots, j_{i-1}, j_i+1, \ldots, j_d), \\
j_i = 0, \ldots, n_i - 1, j_k = 0, \ldots, n_k \forall k \neq i, \\
i = 1, \ldots, d.
\]

(12)

This section employs bases of degrees \( K = 1 \) and \( K = 2 \). For \( K = 2 \), (11) is continuously differentiable on \((0, 1)^d\) (because the multiplicity of each interior knot is one). Let \( S = \{ f(x|e) \} \) be the space of functions of the form (11), \( E \) be the set of all \( e \) that satisfies (12), and \( \mathcal{R} = \{ f(x|e) : e \in E \} \subset S \), which is a closed convex cone of increasing spline functions. Note that \( S \) is a proper subset of the space of all continuous functions. Moreover, for \( K = 2 \) and \( d > 1 \) or when \( K > 2 \), \( \mathcal{R} \) is a proper subset of \( S \cap M \). However, Theorem 1 implies that any continuous function can be approximated arbitrarily closely by a function in \( S \) and any continuous increasing function can be approximated arbitrarily closely by a function in \( S \) as the Cartesian
product of knot sequences $\prod_{i=1}^{d} (\xi_0^i, \ldots, \xi_{n_i+K+1}^i)$ gets denser in $[0, 1]^d$.

**Theorem 1.** Let $F : [0, 1]^d \to \mathbb{R}$ be a continuous function. Define

$$f(x) = \sum_{j=0}^{n} \sum_{i=0}^{1} F(x_{j}^{i}, \ldots, x_{n_i}^{i}) u_j(x),$$  \hspace{1cm} (13)

where $x_{j}^i = (\xi_0^i + \xi_{j}^i + K+1)/2$. Let $h_i = \max(\xi_{j+1}^i - \xi_{j}^i : i = K, \ldots, n_i)$ and $h = \sum_{i=1}^{d} h_i$. Then, $\|F - f\|_{\infty} \leq \omega(K+1)h/2$, where $\omega(\delta) = \sup_{x \in [0, 1]^d} |F(x_1) - F(x_2)| : x_1, x_2 \in [0, 1]^d, \|x_1 - x_2\|_1 \leq \delta$ is the modulus of continuity of $F$.

If $F$ is uniformly continuous, $\lim_{\delta \to 0} \omega(\delta) = 0$. Thus, $\lim_{h \to 0} \|F - \hat{f}\|_{\infty} = 0$. Define

$$\hat{f}_M(x) = \sum_{j=0}^{n} \sum_{i=0}^{1} \left[ P_M F(x_{j}^{i}, \ldots, x_{n_i}^{i}) \right] u_j(x).$$  \hspace{1cm} (14)

Then, it is easy to see that $\hat{f}_M(x) \in \mathcal{R}$. It seems that $P_M F$ should be continuous when $F$ is continuous. This is shown for $d = 1$ by Groeneboom and Jongbloed (2010). Lin and Dunson (2014) assumed that $P_M F$ is continuous even when $d > 1$ without giving a proof. The proof by Groeneboom and Jongbloed (2010) seems difficult to generalize to the case where $d > 1$. However, since the set of discontinuities of $P_M F$ must have Lebesgue measure zero (Lindner 2004) and $P_M F$ is bounded for continuous $F$, Theorem B gives $\lim_{h \to 0} \|P_M F - \hat{f}_M\|_w = 0$.

**Theorem 2.** Let $F : [0, 1]^d \to \mathbb{R}$ be discontinuous on a set of Lebesgue measure zero. Assume that $F$ is bounded. Define $\hat{f}$ as in Theorem 1. Then, $\lim_{h \to 0} \|F - \hat{f}\|_{w} = 0$.

It is known that $\hat{f}$ is a poorer approximation of $F$ than $P_M F$, where $P_M$ is the projection operator that projects a function onto $S$. Thus, $\hat{f}_M$ should be a poorer approximation of $P_M F$ than $P_R F$, where $P_R$ projects a function onto the closed convex cone $\mathcal{R}$. Thus, we propose to employ the monotonic metamodel $P_R Y$, which is obtained by minimizing

$$\|Y - \hat{f}(x)\|_w^2$$  \hspace{1cm} (15)

subject to the constraints (12). We now show that $\lim_{h \to 0} P_R Y = P_M Y$. Note that

$$\|Y - P_M Y\|_w \leq \|Y - P_R Y\|_w \leq \|Y - P_R P_M Y\|_w.$$  \hspace{1cm} (16)

The first inequality is because $\mathcal{R} \subset \mathcal{M}$ while the second inequality follows from the fact that $P_R P_M Y \in \mathcal{R}$. Let $\hat{f}_M$ be given by (14) with $F = Y$. Since $\hat{f}_M \in \mathcal{R}$, $\|P_M Y - P_R P_M Y\|_w \leq \|P_M Y - \hat{f}_M\|_w$. By Theorem 2, $\lim_{h \to 0} \hat{f}_M = P_M Y$. Thus, we have $\lim_{h \to 0} P_R P_M Y = P_M Y$. By this result and (16), we have $\lim_{h \to 0} \|Y - P_R Y\|_w = \|Y - P_M Y\|_w$. By an elementary result of projection onto closed convex sets,

$$\|Y - P_R Y\|_w^2 = \|Y - P_M Y - (P_R Y - P_M Y)\|_w^2 \geq \|Y - P_M Y\|_w^2 + \|P_M Y - P_R Y\|_w^2.$$  \hspace{1cm} (17)

Thus, $\lim_{h \to 0} \|P_M Y - P_R Y\|_w^2 = 0$. This implies that $\lim_{h \to 0} P_R Y = P_M Y$.

It is too time consuming to find a different $\mathcal{R}$ for each sample path $Y$ to make $\|P_M Y - P_R Y\|_w^2$ small to obtain monotonic predictions. Thus, we would like to find a closed convex cone $\mathcal{R}$ (defined by $S$ and (12)) such that $\|P_M Y - P_R Y\|_w^2$ is small on average. This is indeed possible as implied by Theorem 3, whose proof is given in Appendix B.

**Theorem 3.** Let $Y$ be a continuous GP with continuous covariance function and $\mathcal{R}$ be the cone of spline functions that satisfy (12). Then, $\lim_{h \to 0} E(\|P_M Y - P_R Y\|_w^2) = 0$.

In practice, we need to choose the knots so that $E(\|P_M Y - P_R Y\|_w^2)$ is sufficiently small. The knots define $S$, which together with (12), define $\mathcal{R}$. However, it seems that the quantity $E(\|P_M Y - P_R Y\|_w^2)$ is not possible to estimate without first computing $P_M Y$. Because it is difficult to estimate $E(\|P_M Y - P_R Y\|_w^2)$, we propose to increase the number of knots until the change in $P_R Y$ is small. In particular, if $\mathcal{R}_1, \mathcal{R}_2, \ldots$ is a sequence of monotone spline cones obtained by increasing the number of knots such that $h \to 0$, then we can select the cone $\mathcal{R}_k$ given by the smallest $k$ such that $E(\|P_{\mathcal{R}_k} Y - P_{\mathcal{R}_k} Y\|_w^2)$ is sufficiently small.

### 4.2 B-Spline Model Selection

Model (11) allows us to easily construct a metamodel that is monotonic in any input. We only need to estimate the coefficients $c(j_1, \ldots, j_d)$ subject to linear constraints for each simulated sample path $Y$. However, the number of coefficients $\prod_{i=1}^{d} n_i$ is often very large. A variable selection approach cannot be used to reduce the model size because the $c(j_1, \ldots, j_d)$'s must be increasing in $j_i$ to ensure that $f$ is monotonic with respect to input $i$.

We propose the following method to reduce the number of bases. The idea is to employ the modified B-spline bases:

$$b^i_j = \begin{cases} 1 & \text{if } j = 0 \\ u^i_j & \text{if } j \in \{1, \ldots, n_i\} \end{cases}$$  \hspace{1cm} (18)

This idea is similar to the idea of the modified Bernstein bases in Tan (2016). Note that since $\sum_{j=0}^{n_i} u^i_j(x_i) = 1$, both $\{b^0, \ldots, b^{n_i}_0\}$ and $\{u^0, \ldots, u^{n_i}_i\}$ have the same span. Thus, the tensor product bases $B = \prod_{i=1}^{d} b^i_j : j_k = 0, \ldots, n_k; k = 1, \ldots, d$ and $U = \prod_{i=1}^{d} u^i_j : j_k = 0, \ldots, n_k; k = 1, \ldots, d$ have the same span. This implies that $f$ can be rewritten as

$$f(x|a) = \sum_{j=0}^{n} \sum_{i=0}^{n} a(j_1, \ldots, j_d) \prod_{i=1}^{d} b^i_j(x_i).$$  \hspace{1cm} (19)

By equating (11) and (19), we find that (see Appendix C)

$$c(j_1, \ldots, j_d) = \sum_{k=1}^{d} \sum_{k_i \in \{0, \ldots, j_i\}} a(k_1, \ldots, k_d).$$  \hspace{1cm} (20)
Thus, monotonicity in the $i$th input is achieved by imposing the linear constraints:

$$c(j_1, \ldots, j_i + 1, \ldots, j_d) - c(j_1, \ldots, j_i, \ldots, j_d)$$

$$= \sum_{k=1}^{d} \sum_{i=0}^{n_i} \left[ \sum_{k_i \in [0, j_i]} a(k_1, \ldots, k_i) - \sum_{k_i = 0}^{j_i} a(k_1, \ldots, k_i) \right] \geq 0,$$  

$$j_i = j_1 = 0, \ldots, n_i - 1, j_d = 0, \ldots, m_d \forall k \neq i.$$  

(21)

Note that (19) immediately gives a functional ANOVA decomposition, that is, it decomposes into a sum of a constant $a(0, \ldots, 0)$, terms that depend only on a single input $f_{i}(x_i) = \sum_{j=1}^{n_i} a(0, \ldots, 0, j, 0, \ldots, 0) b_{j}^{i}(x_i),$  

(22)
terms that depend only on two inputs

$$f_{i,j}(x_i, x_j) = \sum_{k=1}^{n_i} \sum_{l=1}^{n_j} a(0, \ldots, 0, k, 0, \ldots, 0, l, 0, \ldots, 0) \times b_{k}^{i}(x_i) b_{l}^{j}(x_j)$$  

(23)

and so on. We call (22) a main effect, and (23) a two-factor interaction. These are called functional ANOVA components. In contrast, each basis $\prod_{i=1}^{d} \eta_i^{j_i}$ is a $d$-factor interaction.

In many cases, only a handful of main effects and interactions have been proposed (Wahba 1986; Reich, Storlie, and Bondell 2009). Model (19) with only nonzero main effects and two-factor interactions can contain far fewer terms than model (11).

If $n_1 = \cdots = n_d = n_i$, then the former model contains $1 + n_i d + n_i^2 d (d - 1)/2$ terms but the latter contains $(n_i + 1)^d$ terms. The number of constraints in (21) is $(n_i + 1)^{d - 1}$ but is reduced to $(n_i + 1)^{d - 1} - n_i$ when input $i$ does not interact with $k$ other inputs.

In this article, we fit models of the form:

$$f(x|a) = a(0, \ldots, 0) + \sum_{(i_1, \ldots, i_d) \in \Xi} f_{i_1, \ldots, i_d} (x_{i_1}, \ldots, x_{i_d}),$$  

(24)

where $\Xi \subseteq \{(1), (1, 2), (1, 2, 3), \ldots, (1, \ldots, d)\}$ and $f_{i_1, \ldots, i_d}$ is the interaction between inputs $i_1, \ldots, i_d$. This amounts to setting some of the coefficients, that is, the $a(j_1, \ldots, j_d)$’s, in (19) to zero. Monotonicity with respect to input $i$ is achieved by imposing the constraints (21).

A referee suggested using triogram splines (Hansen, Kooperberg, and Sardy 1998). However, the triogram splines are bi-variate splines. Thus, a triogram spline approximation cannot converge to a function with nonzero three-factor or higher order interactions. Since the GP model with a product correlation function has nonzero high-order interactions (although they may be small), Theorem 3 cannot hold for triogram splines. However, triogram splines can be generalized to splines on simplices (Neamtu 2001). This is a problem for further research.

4.3 Algorithm

This section describes the algorithm we employ to construct a monotonic B-spline model that is the weighted projection of the GP $Y$ onto a selected cone of monotone splines. The algorithm increases the number of knots and adds functional ANOVA components into the model (24) in such a way that the change in the monotone spline predictions on a set of points is maximized. When the maximum change is small, the algorithm terminates.

Let $R_1, R_2, \ldots$ be a sequence of monotone spline cones, each defined by the spline knots $\{\xi_0^i, \ldots, \xi_{n_i+1}^i\}, i = 1, \ldots, d$, choice of $\Xi$ in (24), and constraints (21). Practical computation of $P_{R_i} Y$ involves using a set of points to approximate the squared norm $\|P_{R_i} Y\|^2_w$. The projection $P_{R_i} Y$ of $Y$ onto $R_i$ is approximated by minimizing $\sum_{i=1}^{N} (Y(x_i) - f(x_i|a))^2 w(x_i)$ subject to (21) for all $i = 1, \ldots, d$, for some chosen points $\{x_1, \ldots, x_N\} \subset [0, 1]^d$. In this article, we set the first $n$ points of $\{x_1, \ldots, x_N\}$ to be from the experiment design and the remaining $N - n$ points to be from a Sobol sequence (Niederreiter 1992). We denote the discrete projection onto $R_k$ by $\hat{f}_k(x) = f(x|\hat{a}_k)$, where

$$\hat{a}_k = \arg\min \sum_{i=1}^{N} (Y(x_i) - f(x_i|a))^2 w(x_i) : (21) \text{ holds for all } i = 1, \ldots, d.$$  

(25)

As the number of knots and selected functional ANOVA component increases, the number of model terms increases. Thus, $N$ must increase also to ensure that $\hat{a}_k$ is uniquely defined.

The quantity $\|P_{R_{k-1}} Y - P_{R_k} Y\|^2_w$ is approximated with

$$\alpha_k (Y) = \sum_{i=1}^{m} (f_{k-1}(x_i') - f_k(x_i'))^2 w(x_i'),$$  

(26)

where $\{x_1', \ldots, x_m'\} = D_i \subset [0, 1]^d$ is chosen to be a maximin Latin hypercube design (MLHD). We can use $R_k$ to construct the monotone spline metamodel when the average of $\alpha_k(Y)$ over a number of simulated sample paths $Y$ is small. However, it is time consuming to simulate many sample paths and project them onto $R_k$. In this article, we set $\bar{Y}$ equal to the mean, pointwise 0.99 quantile, and pointwise 0.01 quantile of the GP, which we denote as $Y_{0.5}, Y_{0.99},$ and $Y_{0.01}$, respectively. We use the criterion

$$\alpha_k = 100 [\alpha_k(Y_{0.5}) + \alpha_k(Y_{0.99}) + \alpha_k(Y_{0.01})] / \{V(Y_{0.5}) + V(Y_{0.99}) + V(Y_{0.01})\},$$  

(27)

where $V(Y_{0.5}) = \sum_{i=1}^{m} (Y_{0.5}(x_i') - \hat{Y}_{0.5})^2 w(x_i')$ and $\hat{Y}_{0.5} = \sum_{i=1}^{m} Y_{0.5}(x_i') w(x_i') / \sum_{i=1}^{m} w(x_i')$, to select the knots and the functional ANOVA components. In (27), $\alpha_k(Y_{0.5}) + \alpha_k(Y_{0.99}) + \alpha_k(Y_{0.01})$ is expressed as a percentage of the sum of variances of $Y_{0.5}, Y_{0.99},$ and $Y_{0.01}$.

In this article, we employ equally spaced knots

$$\{\xi_0^i, \ldots, \xi_{n_i+1}^i\} = \{0, 0.1/m_i, \ldots, 1, \ldots\}. $$  

(28)

The sequence $R_1, R_2, \ldots$ is constructed as follows: Given $R_k$, we increase $m_i$ by one, which gives a tentative choice for $R_{k+1}$, and we compute the value of $\alpha_{k+1}$ for this tentative choice. This process is repeated for all $i = 1, \ldots, d$. We also add each functional ANOVA component not in the model (each addition gives a tentative choice for $R_{k+1}$) and compute the value of $\alpha_{k+1}$. If the increase in $m_i$ gives the maximum $\alpha_{k+1}$,
then this increase is adopted to define $R_{k+1}$. If the addition of a functional ANOVA component into the model gives the maximum $\alpha_k$, then this addition is adopted to define $R_{k+1}$. The choice of an $R_{k+1}$ that maximize $\alpha_k$ (which is a surrogate for $E[\|P_{R_k} Y - P_{R_{k+1}} Y\|_w^2]$) is sensible because this choice has the most significant influence on the predictions given by the monotone spline metamodel. Note that the tail quantiles tend to be more nonmonotonic than any simulated sample path. Thus, using $\alpha_k$ as criterion may lead to the selection of more knots and functional ANOVA components than necessary.

The algorithm we use to construct the monotone spline meta-model is given below:

1. Initialize $m_1 = \cdots = m_d = 1$, $f(x|\alpha) = a(0, \ldots, 0) + \sum_{i=1}^{d} f_{ij}(x_i)$, and $N = \max\{3M, 2n\}$, where $M$ is the size of the incumbent model. Set $\{x_1, \ldots, x_N\}$ to be points from the experiment design and a Sobol sequence, $D_t = \{x'_1, \ldots, x'_n\}$ to be an MLHD (design points $x^1, \ldots, x^n$ can be included in $D_t$), and $k = 1$.
2. Compute $\alpha_k$ in (27), where we set $f_0(x'_1) = \bar{y}_{0.5}$. If $\alpha_k \leq T$ and $M \geq 2n$, go to Step 7.
3. Set $k = k + 1$. Find the model sizes that would result if each $m_i$ is increased by one, and the model sizes that would result if a functional ANOVA component not in the incumbent model is added into the model. Let $M_{\text{max}}$ be the maximum of these model sizes. Set $N = \max\{3M_{\text{max}}, 2n\}$ and $\{x_1, \ldots, x_N\}$ to be points from the design and a Sobol sequence.
4. For each $i = 1, \ldots, d$, increase each $m_i$ by one. By setting $Y$ to $Y_{0.5}, Y_{0.99}$, and $Y_{0.01}$ in turn, fit the model by computing $\hat{\alpha}_k$ in (25) and compute the predictions of the new model at $[x'_1, \ldots, x'_n]$. Compute $\alpha_k$.
5. For each functional ANOVA component not in the incumbent model, add it into the model. By setting $Y$ to $Y_{0.5}, Y_{0.99}$, and $Y_{0.01}$ in turn, fit the model by computing $\hat{\alpha}_k$ in (25) and compute the predictions of the new model at $[x'_1, \ldots, x'_n]$. Compute $\alpha_k$.
6. Choice of $R_{k+1}$: If the maximum $\alpha_k$ corresponds to an increase in $m_i$, then update $m_i$ by increasing it by one and update the spline bases. If the maximum $\alpha_k$ corresponds to adding a functional ANOVA component, then add that functional ANOVA component into the model. If $\alpha_k \leq T$ and $M \geq 2n$, go to Step 7. Otherwise, return to Step 3.
7. Simulate the GP $Y$ on $\{x_1, \ldots, x_N\}$ and minimize $\sum_{i=1}^{N} [Y(x) - f(x|\hat{\alpha})]^2 w(x_i)$ subject to (21), which gives $\hat{\alpha}$. Repeat $r$ times. The sample mean of $f(x|\hat{\alpha})$ is used as point predictor whereas the sample tail quantiles are used as prediction interval limits.

**Remark 1.** We require the model size $M$ to be at least $2n$ because good prediction at the data points would require a model of around $n$ terms and additional terms are needed to quantify uncertainty about the true function so that credible intervals can be constructed.

**Remark 2.** The algorithm increases $N$ as the model size increases because $N$ must be larger than the model size so that the model can be estimated reliably via weighted least squares. Thus, we need $N \geq M_{\text{max}}$. Because the minimum model size that we allow is $2n$, we set $N = \max\{3M_{\text{max}}, 2n\}$. The constant 3 seems to work well in practice.

The parameter $T$ controls the accuracy with which the monotone spline model approximates $P_{\lambda i} Y$ while the parameter $B$ controls closeness of $P_{\lambda i}$ to the experiment data. To construct an interpolating metamodel, we need to use a small $B$, a small $T$, and also include the design points in the training set $\{x_1, \ldots, x_N\}$ used to fit the monotone spline metamodel. Very small values of $B$ and $T$ can be used when $d = 1$. However, such values can cause computational difficulties for $d > 1$ because the model size can become too large. For fixed $B$, the model size will increase as $T$ gets smaller. For fixed $T$, the monotone metamodel’s predictions at the design points will get closer to the experiment data as $B$ gets smaller. Although accuracy improves and the metamodel gets closer to interpolating the data as $B$ gets smaller, the prediction interval coverage gets poorer. The value of $B$ we recommend is $B = \max\{\text{var}(x'_1), \ldots, \text{var}(x'_n)\}/20$ for $d > 1$ based on results in Example 2. With this choice of $B$, the ratio of the maximum to the minimum weight $w(x)$ on $[0, 1]^d$ is approximately 20. This seems to give good coverage and accuracy for $T \approx 1$.

Given the selected spline knots and functional ANOVA components, the monotonic metamodel $f(x|\hat{\alpha})$ is constructed in Step 7 by simulating values of the GP $Y$ on the set of training points and minimizing $\sum_{i=1}^{N} [Y(x) - f(x|\hat{\alpha})]^2 w(x_i)$ subject to (21). The main computational burden in the algorithm is the process of solving the quadratic program (25) when the vector of model coefficients $\alpha$ is large, or when the constraints (21) are required to hold for many inputs because the program would then contain many decision variables or constraints. In Step 4 and Step 5, we solve the quadratic program three times but the steps may need to be repeated many times. In Step 7, we need to solve the program several hundred times to obtain accurate estimates of prediction intervals.

### 5. Examples

This section provides examples to demonstrate the usefulness of our approach. We compare the two proposed methods to a few alternative methods. The first is the unweighted projection method proposed by Lin and Dunson (2014). The second is the average rearrangement estimator proposed by Chernozhukov, Fernandez-Val, and Galichon (2009). The rearrangement method monotonizes the posterior mean and credible limits of the GP on a grid. Neither Lin and Dunson (2014) nor Chernozhukov, Fernandez-Val, and Galichon (2009) discussed how the grid may be selected or which interpolator should be used. Thus, we adopt the grid selection method described at the end of Section 3 (which is used to implement the grid-based weighted projection method) to implement the unweighted projection and rearrangement methods. We use the interpolator (10) for both methods. We also compare the proposed method to Belavâte’s (2005) interpolator. Belavâte’s (2005) interpolator is constructed using eq. (3.3) of his article with Lipschitz constant estimated with his eq. (5.1).

For the unweighted projection and grid-based weighted projection methods, and monotone B-spline metamodel, the interval predictions are obtained with the 0.01 and 0.99 quantiles of 300
simulated sample paths while the point predictions are the sample mean. The choice of initial grid in the grid selection procedure is as follows: For \( d = 1 \), the initial grid is \([0, 0.01, \ldots, 1]\); for \( d = 3 \), it is \([0, 1/6, \ldots, 1]\); for \( d = 4 \), it is \([0, 1/5, \ldots, 1]\). We set the target \( T \) in the monotone spline algorithm to be 0.01 for \( d = 1 \), 1 for \( d = 3 \), and 3 for \( d = 4 \).

5.1 Example 1

In this example, we construct monotonic metamodels for an increasing function plotted in Figure 1. The design is \( D = \{0, 0.25, 0.5, 0.75, 1\} \). In Figure 1(a), we plot the posterior mean and 98% credible intervals for the stationary GP model with the Matérn correlation function described in Section 2. The posterior mean is only slightly nonmonotonic. However, the credible intervals are highly nonmonotonic.

In Figure 1(b), predictions of the unweighted projection approach are plotted. We see that the interval predictions at the design points can be of nonnegligible width. This problem seems to be worse for the rearranged point predictions and credible intervals plotted in Figure 1(c). Consequently, the rearranged predictions look visually unappealing. Note that the point predictions for the unweighted projection and rearrangement approaches are nearly interpolating the data because the posterior mean of the GP is nearly monotonic. Beliakov's (2005) interpolator (Figure 1(d)) gives poor interval predictions, which have zero width in the interval \([0.25, 0.5]\). Note that there will always be an interval between two design points with zero prediction interval width if the Lipschitz constant is estimated with eq. (5.1) in Beliakov (2005). Figure 1(e) plots the predictions obtained with the grid-based weighted projection method while Figure 1(f) plots the predictions obtained with the monotone B-spline metamodel (\( K = 2 \)). We set \( B = \text{max}\{\text{var}(x_1^t), \ldots, \text{var}(x_m^t)\}/100 \) for both methods. We see that for both methods, the point and interval predictions nearly interpolate the data points.

5.2 Example 2: Displacement in a Mindlin Plate

Consider an isotropic flat plate of uniform thickness \( \mathcal{H} > 0 \) under transverse loading \( \mathcal{Q}(x, y) \). The plate is polygonal and simply supported. Assume a right-hand coordinate system (with \( z \) pointing downward). The Mindlin plate theory predicts the transverse displacement \( \mathcal{W}(x, y) \) of the plate with two Poisson equations (chap. 7 of Wang, Reddy, and Lee 2000):

\[
\Delta m(x, y) = -\mathcal{Q}(x, y), \quad m(x, y) = 0\mathcal{W}(x, y) \in \partial\Omega, \quad (29)
\]

\[
\Delta W(x, y) = -\mathcal{Q}(x, y)(6 - \mathcal{V})/(5\mathcal{G}\mathcal{H}) - m(x, y)/ \times \left[ \mathcal{G}\mathcal{H}^2/[6(1 - \mathcal{V})] \right], \quad \mathcal{V}(x, y) = 0\mathcal{W}(x, y) \in \partial\Omega, \quad (30)
\]

where \( \Delta \) is the Laplacian operator, \( \Omega \) is the plate region of the plate, \( \partial\Omega \) is the plate boundary, \( m \) is the Marcus moment, \( \mathcal{G} > 0 \) is the shear modulus, and \( \mathcal{V} \in [0, 0.5] \) is the Poisson ratio. Note that we set the shear correction factor to \( \mathcal{S}(6 - \mathcal{V}) \) following Liew et al. (1998 p. 24) and \( m \) is uniquely defined by (29). In particular, \( m \) does not depend on \( \mathcal{G}, \mathcal{H}, \) and \( \mathcal{V} \).

It follows from physical considerations or the comparison theorem at p. 177 of Mattheij et al. (2005) that if \( m^1(x, y) \) and \( m^2(x, y) \) denote the Marcus moment under loadings \( \mathcal{Q}^1(x, y) \) and \( \mathcal{Q}^2(x, y) \), respectively, then \( m^1(x, y) \leq m^2(x, y) \mathcal{V}(x, y) \in \Omega \) whenever \( \mathcal{Q}^1(x, y) \leq \mathcal{Q}^2(x, y) \mathcal{V}(x, y) \in \Omega \). Similarly, the comparison theorem implies that \( \mathcal{W} \) should be a decreasing function of the right-hand side of (30), that is, \( -\mathcal{Q}(x, y)(6 - \mathcal{V})/(5\mathcal{G}\mathcal{H}) - m(x, y)/ \times \left[ \mathcal{G}\mathcal{H}^2/[6(1 - \mathcal{V})] \right] \). Thus, \( \mathcal{W} \) and the maximum displacement \( V = \max \{\mathcal{W}(x, y): (x, y) \in \Omega\} \) is an increasing function of \( \mathcal{Q} \), and a decreasing function of \( \mathcal{G}, \mathcal{H}, \) and \( \mathcal{V} \). Note that it is intuitively clear that \( \mathcal{W} \) should decrease when \( \mathcal{H} \) or \( \mathcal{G} \) increases, that is, as the plate becomes thicker or more resistant to shear forces. However, the effect of \( \mathcal{V} \) on \( \mathcal{W} \) is not obvious to this author without the above reasoning. Finally, it is obvious from physical considerations that the maximum displacement \( Y \) is an increasing function of the length of the plate \( \mathcal{L} \) if the plate is rectangular. This can be shown mathematically by rescaling \( x \) and \( y \).

We solve (29)–(30) for a square plate with side length \( \mathcal{L} \) and uniform loading \( \mathcal{Q} = 10^6 \) using the finite element method implemented in the Matlab code given in chap. 12 of Ferreira (2008). A total of \( 30 \times 30 = 900 \) Q4 elements are used. Note that the code takes the Young's modulus \( \mathcal{E} = 2\mathcal{G}(1 + \mathcal{V}) \) instead of the shear modulus \( \mathcal{G} \) as input. We transform the variables \( \mathcal{G}, \mathcal{H}, \mathcal{V}, \mathcal{L} \) to \( x_1 = -(\mathcal{G} - 30 \times 10^9)/(10 \times 10^9), x_2 = -(H - 0.25)/0.2, x_3 = -(\mathcal{V} - 0.4)/0.2, x_4 = (\mathcal{L} - 1)/1 \). Thus, \( Y \) is increasing in \( (x_1, \ldots, x_4) \in [0, 1]^4 \).

We perform simulations to compare the alternative metamodeling methods for approximating the finite element code. In the first set of 100 simulations, we fix \( x_4 = 0.5 \) and model \( Y \) as a function of \( x_1, x_2, x_3, x_4 \). In the second set of 100 simulations, we model \( Y \) as a function of \( x_1, x_2, x_3, x_4, x_4 \) \( d = 4 \). We set \( B = \max \{\text{var}(x_1^t), \ldots, \text{var}(x_m^t)\} / 20 \). In all simulations, a generalized maximin Latin hypercube design (GMLHD) proposed by Dette and Pepelyshev (2010) of size \( 7d \) is used to fit the GP model, which is then employed to construct the unweighted projection, grid-based weighted projection, rearrangement, and monotone B-spline (\( K = 1 \) and \( K = 2 \)) metamodels. Beliakov's interpolator is constructed directly with data from the GMLHD. Each GMLHD is obtained by applying the quantile function of the arcsine density to transform each element of an MLHD.

For each of the two sets of simulation runs, we compute the mean squared error (MAE), root mean squared error (RMSE), average prediction interval length (average PI length), and coverage of nominal 98% intervals for each method on an MLHD test set \( D_t \) of size \( 100d \). The simulation means and standard deviations of these quantities for \( d = 4 \) are presented in Table 1. We only give results for \( d = 4 \) in the article because results for \( d = 3 \) (see Appendix D) lead to essentially the same conclusions regarding the relative performance of the alternative methods. Table 1 shows that Beliakov's interpolator gives the poorest prediction accuracy (MAE and RMSE) in all cases and also poor prediction interval length. The monotone spline metamodel with \( K = 2 \) provides better prediction accuracy than the other methods. It also gives the shortest prediction intervals. The average length of the intervals is only around 30% the average length of the credible intervals of the GP model. The average coverage of the monotone spline metamodel with \( K = 2 \) is 0.89, and it is only slightly lower than the average coverage achieved by the GP model, which is 0.95. It appears that the monotone spline metamodel with \( K = 1 \) is dominated by the monotone
spline metamodel with $K = 2$ in terms of average accuracy and prediction interval length.

Compared to the monotone spline metamodel with $K = 2$, the grid-based weighted projection method gives poorer accuracy, and longer intervals with better (closer to nominal) coverage. The grid-based weighted projection method gives better accuracy, shorter prediction intervals, and similar coverage compared to the unweighted projection method. Thus, the use of the weights in (9) indeed provides better results for deterministic computer experiments. The rearrangement procedure gives similar accuracy and better coverage than the grid-based weighted projection method but its prediction intervals are very wide.

The GP model, monotone spline model, grid-based weighted and unweighted projection methods, and rearrangement method give average coverage that is less than the nominal 0.98. However, the average coverages achieved by these methods are only slightly below the nominal. Note that the monotone spline metamodels and the grid-based weighted projection method are supposed to converge to the projection of the GP onto the cone of monotone functions, that is, $P^M_Y$. However, both methods yield quite different results. This is because the monotone spline metamodel uses a finite number of spline knots and is fitted with a discrete set of points, while the grid-based method uses a finite grid.

Table 1 also gives the simulation means and standard deviations of the size of the monotone spline metamodel, and the grid sizes for the grid-based methods. The monotone spline metamodels ($K = 1$ and $K = 2$) have model sizes that are far smaller than the grid sizes of the grid-based methods. In fact, such large grid sizes are necessary. The initial grid is of size 1296 but the final chosen grids are far larger. Thus, the monotone spline metamodel is less memory intensive than the grid-based methods. In seven simulations, a single run for the monotone spline model with $K = 2$ took 44 to 169 sec with an average of 74 sec on a Lenovo laptop with a 2.9GHz processor and a 4GB RAM. The grid-based weighted projection methods took 1018 to 2814 sec with an average of 1701 sec. The unweighted projection method took 209 to 646 sec with an average of 409 sec. Finally, the rearrangement method took 53 to 606 sec with an average of 271 sec.

Next, we examine the interpolation property of the methods. The MAE of the lower and upper credible limits at the design points are given in Table 1. The GP model and Beliakov’s interpolator have MAE at design points of zero because they interpolate the data. The monotone spline model and grid-based weighted projection method give tighter intervals at the design points than the unweighted projection method, and rearrangement method. The rearrangement method always gives the widest average prediction intervals at the design points. Note that the sums of the MAE of the prediction limits at the design points for the monotone spline models are somewhat less than their average prediction interval length.

Finally, we consider the effect of the choice of $B$. We fix $x_4 = 0.5$ and model $f$ as a function of $x_1, x_2$, and $x_3 (d = 3)$. A total of 100 simulations is performed with $B = V_m/100$, $B = V_m/20$, and $B = V_m/10$, where $V_m = \max \{ \var (x_1^t), \ldots, \var (x_n^t) \}$. Results for the monotone spline model and grid-based weighted projection method are given in Table D1 and Table D2 in Appendix D. We see that when $B$ is increased, the average prediction interval length decreases, and the interval predictions get closer to interpolating the experiment data. However, the average coverage of the intervals deteriorates. We recommend using $B = V_m/20$ since it appears to give a good tradeoff.

### Table 1. MAE, RMSE, average PI length, coverage, model size, and MAE at design points for GP model (GP), monotone spline model (spline), grid-based weighted projection method (weighted), unweighted projection method (unweighted), rearrangement method (rearranged), and Beliakov’s interpolator (Beliakov) for $d = 4$. Values for MAE, RMSE, average PI length, and MAE at design points are multiplied by $10^4$. Bold values are best mean values

|               | GP          | Spline, $K = 1$ | Spline, $K = 2$ | Weighted | Unweighted | Rearranged | Beliakov |
|---------------|-------------|----------------|----------------|-----------|------------|------------|----------|
| MAE           | Mean 21.16  | 15.27          | **14.33**      | 17.52     | 20.18      | 17.66      | 79.86    |
|               | Std 11.41   | 7.46           | 7.41           | 8.16      | 9.83       | 9.91       | 35.53    |
| RMSE          | Mean 41.34  | 32.09          | **30.02**      | 35.20     | 37.59      | 37.46      | 112.68   |
|               | Std 20.65   | 15.24          | 15.54          | 16.81     | 18.13      | 17.66      | 47.16    |
| Average PI length | Mean 178.28 | 56.24          | **50.15**      | 64.19     | 86.57      | 189.32     | 213.37   |
|               | Std 70.80   | 17.78          | 17.88          | 19.94     | 29.68      | 75.95      | 91.55    |
| Coverage      | Mean 0.95   | 0.91           | 0.89           | 0.92      | 0.91       | **0.96**   | 0.96     |
|               | Std 0.03    | 0.06           | 0.07           | 0.05      | 0.05       | 0.03       | 0.04     |
| Model size    | Mean –      | **64.33**      | 67.91          | 2341.80   | 2046.60    | 2082.96    | –        |
|               | Std –       | 10.33          | 17.86          | 613.54    | 409.10     | 501.16     | –        |
| MAE at design points | LCL 0.00   | 19.99          | 20.44          | 22.03     | 33.03      | 91.55      | 0.00     |
|               | UCL 0.00    | 17.39          | 19.00          | 24.83     | 34.97      | 94.04      | 0.00     |

### 6. CONCLUSIONS

This article proposes the weighted projection approach to construct monotonic metamodels for deterministic computer experiments. It is based on projecting sample paths of the posterior GP onto the cone of monotone functions with respect to a weighted norm with weight function that is inversely proportional to the posterior variance. Two computational implementations are proposed: the grid-based method and the monotone spline metamodel. In the first method, isotonic regression is performed on a grid and in the second method, the GP sample paths are projected onto a selected cone of monotone splines. The proposed monotonic metamodels tend to give more accurate predictions and substantially shorter credible intervals than the GP model with only slightly poorer credible interval cov-
verage, as demonstrated in Example 2 and Example 3 (online supplement).

The proposed methods outperform the unweighted projection method in terms of prediction accuracy and interval length. The monotone spline metamodel is computationally less demanding and more accurate than the grid-based methods. It can be used for moderately high-dimensional problems when the number of significant functional ANOVA components is small and monotonicity with respect to a few inputs only is required because the number of decision variables and constraints for the quadratic programs that need to be solved would then be small. If one is more concerned with achieving good prediction interval coverage, then the grid-based weighted projection method should be used. A fully Bayesian approach that integrates out \( \lambda, \sigma^2, \) and \( \theta \) can be implemented in a straightforward way. The grid-based weighted projection method only requires posterior sample path values at the grid points while the monotone spline metamodel only requires posterior sample path values at the training points. The posterior sample path values can be obtained with the use of Markov chain Monte Carlo methods. This approach may give coverage closer to nominal since all parameter uncertainties are fully taken into account. However, it is more time consuming because Markov chain Monte Carlo simulation needs to be performed.

**SUPPLEMENTARY MATERIALS**

Appendices.pdf: This file contains Appendices A–E. Appendix A gives the multivariate isotonic regression algorithm. Appendix B contains proofs of the theorems given in the article. Appendix C gives a proof of (20). Appendix D presents additional numerical results for Example 2. An additional example is given in Appendix E.

Online zip file: This file contains Matlab codes for all methods considered in this article.

**ACKNOWLEDGMENTS**

The author thanks an associate editor and two referees for comments that helped improve the article significantly. This research was supported by Early Career Scheme (ECS) project No. 21201414 sponsored by the Research Grants Council of Hong Kong.

[Received January 2015. Revised September 2015.]

**REFERENCES**

Barlow, R. E., Bartholomew, D. J., Bremner, J. M., and Brunk, H. D. (1972), *Statistical Inference Under Order Restrictions: The Theory and Application of Isotonic Regression*, New York: Wiley. [2,3]

Beliaikov, G. (2005), “Monotonicity Preserving Approximation of Multivariate Scattered Data,” *BIT Numerical Mathematics*, 45, 653–677. [1,3,7,8]

Brezger, A., and Steiner, W. J. (2008), “Monotonic Regression Based on Bayesian P-Splines,” *Journal of Business & Economic Statistics*, 26, 90–104. [2]

Chernozhukov, V., Fernandez-Val, I., and Galichon, A. (2009), “Improving Point and Interval Estimators of Monotone Functions by Rearrangement,” *Biometrika*, 96, 559–575. [1,7]

Curran, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991), “Bayesian Prediction of Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments,” *Journal of the American Statistical Association*, 86, 953–963. [2]

De Boor, C. (2001), *A Practical Guide to Splines* (Applied Mathematical Sciences, Vol. 27, Revised ed.), New York: Springer. [4]

Dette, H., and Pepelyshev, A. (2010), “Generalized Latin Hypercube Design for Computer Experiments,” *Technometrics*, 52, 421–429. [8]

Du, P., Parmeter, C. F., and Racine, J. S. (2013), “Nonparametric Kernel Regression With Multiple Predictors and Multiple Shape Constraints,” *Statistica Sinica*, 23, 1343–1372. [2]

Dykstra, R. L., and Robertson, T. (1982), “An Algorithm for Isotonic Regression for Two or More Independent Variables,” *The Annals of Statistics*, 10, 708–716. [3]

Ferreira, A. J. (2008), *MATLAB Codes for Finite Element Analysis: Solids and Structures* (Vol. 157), New York: Springer Science and Business Media. [8]

Floater, M. S., and Peña, J. M. (1998), “Tensor-Product Monotonicity Preservation,” *Advances in Computational Mathematics*, 9, 353–362. [4]

Golchi, S., Bingham, D. R., Chipman, H., and Campbell, D. A. (2015), “Monotone Emulation of Computer Experiments,” *SIAM/ASA Journal of Uncertainty Quantification*, 3, 370–392. [1]

Groeneboom, P., and Jongbloed, G. (2010), “Generalized Continuous Isotonic Regression,” *Statistics & Probability Letters*, 80, 248–253. [5]

Hansen, M., Kooperberg, C., and Sardy, S. (1998), “Triogram Models,” *Journal of the American Statistical Association*, 93, 101–119. [6]

He, X., and Ng, P. (1999), “COBS: Qualitatively Constrained Smoothing via Linear Programming,” *Computational Statistics*, 14, 315–338. [2]

Liew, K. M., Xiang, Y., Kitipornchai, S., and Wang, C. M. (1998), *Vibration of Mindlin Plates: Programming the p-version Ritz Method*, Oxford: Elsevier. [8]

Lin, L., and Dunson, D. B. (2014), “Bayesian Monotone Regression Using Gaussian Process Projection,” *Biometrika*, 101, 303–317. [1,2,3,5,7]

Lindner, W. (2004), “Some Properties of the Set of Discontinuity Points of a Monotonic Function of Several Variables,” *Folia Mathematica*, 11, 53–57. [5]

Matheij, R. M., Rienstra, S. W., and ten Thije Boomkamp, J. H. (2005), *Partial Differential Equations: Modeling, Analysis, Computation*, Philadelphia, PA: SIAM. [8]

Neamtu, M. (2001), “What is the Natural Generalization of Univariate Splines to Higher Dimensions?” in *Mathematical Methods for Curves and Surfaces: Oslo 2000*, eds. T. Lyche and L. L. Schumaker, Tennessee: Vanderbilt University Press, pp. 355–392. [6]

Niederreiter, H. (1992), *Random Number Generation and Quasi-Monte Carlo Methods*, Philadelphia, PA: SIAM. [6]

Ramsay, J. O. (1988), “Monotone Regression Splines in Action,” *Statistical Science*, 3, 425–441. [2]

Reich, B. J., Storlie, C. B., and Bondell, H. D. (2009), “Variable Selection in Bayesian Smoothing Spline ANOVA Models: Application to Deterministic Computer Codes,” *Technometrics*, 51, 110–120. [6]

Ribimакic, J., and Vehari, A. (2010), “Gaussian Processes With Monotonicity Information,” in *International Conference on Artificial Intelligence and Statistics*, pp. 645–652. [1]

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), “Design and Analysis of Computer Experiments,” *Statistical Science*, 4, 409–423. [2]

Tan, M. H. Y. (2016), “Monotonic Quantile Regression with Bernstein Polynomials for Stochastic Simulation,” *Technometrics*, 58, 180–190. [2,5]

Wahba, G. (1986), “Partial and Interaction Splines for the Semiparametric Estimation of Functions of Several Variables,” in *Computer Science and Statistics: Proceedings of the 18th Symposium of the Interface*, ed. T. J. Boardman, Washington, DC: American Statistical Association, pp. 75–80. [6]

Wang, C. M., Reddy, J. N., and Lee, K. H. (2000). “Shear Deformable Beams and Plates: Relationships With Classical Solutions,” Oxford: Elsevier Science. [8]

Wang, X. (2012), “Bayesian Modeling Using Latent Structures,” Ph.D. dissertation, Department of Statistics, Duke University. [1]

Wu, C. F. J., and Hamada, M. S. (2009). *Experiments: Planning, Analysis, and Optimization* (2nd ed.), New York: Wiley. [6]