A Scalable Approach to Enhancing Stochastic Kriging with Gradients

Haojun Huo[1], Xiaowei Zhang[1], and Zeyu Zheng[2]

[1]Department of Industrial Engineering and Decision Analytics, HKUST, Clear Water Bay, Hong Kong
[2]Department of Industrial Engineering and Operations Research, University of California, Berkeley, Berkeley, CA 94720, U.S.

Abstract

It is known that incorporating gradient information can significantly enhance the prediction accuracy of stochastic kriging. However, such an enhancement cannot be scaled trivially to high-dimensional design space, since one needs to invert a large covariance matrix that captures the spatial correlations between the responses and the gradient estimates at the design points. Not only is the inversion computationally inefficient, but also numerically unstable since the covariance matrix is often ill-conditioned. We address the scalability issue via a novel approach without resorting to matrix approximations. By virtue of the so-called Markovian covariance functions, the associated covariance matrix can be invertible analytically, thereby improving both the efficiency and stability dramatically. Numerical experiments demonstrate that the proposed approach can handle large-scale problems where prior methods fail completely.

1 Introduction

Stochastic kriging (SK) was proposed in Ankenman et al. (2010) and has recently become a popular metamodeling technique for constructing response surfaces of complex stochastic simulation models in a variety of disciplines including queueing simulation (Shen et al. 2018), financial risk management (Chen and Kim 2016), insurance product pricing (Risk and Ludkovski 2016), etc. The SK metamodel postulates an extrinsic spatial correlation structure on the response surface and utilizes it to predict the unknown responses based on the simulated ones at carefully chosen design points. This metamodeling technique has been used to quantify the impact of input uncertainty on output analysis of simulation models; see Barton et al. (2014) and Xie et al. (2014). It has also been used to facilitate the exploration-exploitation trade-off during the process of searching for the optimal solution of simulation optimization problems; see Quan et al. (2013) and Sun et al. (2014). We refer to Kleijnen (2015) for a recent overview.

For a variety of simulation models, the gradient estimator can be derived analytically using, among others, infinitesimal perturbation analysis (IPA) or the likelihood ratio (LR) method; see

L'Ecuyer (1990). In such situations, the additional cost required to compute the gradient estimator at a design point is usually marginal once the simulation model is already executed at the same location. It is therefore generally perceived as an effective technique to leverage the gradient estimator to enhance SK without inducing significant computational overhead. Indeed, it is shown in Chen et al. (2013) and Qu and Fu (2014), despite different ways of using the gradient information, that doing so increases the prediction accuracy of the SK metamodel considerably.

However, the SK metamodel does not scale up easily. It often suffers from numerical difficulties associated with inversion of covariance matrices. On one hand, matrix inversion generally takes $O(n^3)$ operations, which is computationally prohibitive for large n, where n is the number of design points. On the other hand, the covariance matrix involved in the SK metamodel tends to be ill-conditioned (i.e., nearly singular) for large n, making the computation of the inverse numerically unstable. The scalability issue essentially precludes the use of SK for high-dimensional response surfaces, because, due to the curse of dimensionality, n needs to grow exponentially fast in d to cover a sufficient proportion of a d-dimensional design space. Such an issue is further magnified when the gradient information is integrated, since the size of the covariance matrix would be scaled by a factor of 2 to $(d+1)$, depending on the number of partial derivatives included.

The recent work Ding and Zhang (2018) addresses the scalability issue by constructing a new class of covariance functions which result in *analytically* invertible covariance matrices. Specifically, they show that if the design points form a regular lattice (not necessarily equally spaced), then the inverse of the corresponding covariance matrix is a sparse matrix and each of the nonzero entries have closed-form expressions. The analytical invertibility both reduces the computational cost and improves the numerical stability substantially for using the SK metamodel.

In this paper, we extend their methodology to the setting where gradient estimators are available, thereby making the gradient-enhanced SK metamodel scalable. We show that with our approach, the idea of integrating gradient information with SK can now be applied to simulation models with a high-dimensional design space for which the prior approaches may end up with complete numerical failure.

The remaining of the paper is organized as follows. In Section 2, we introduce the scalability issue in the SK metamodel. In Section 3, we review the so-called Markovian covariance functions (MCFs) and their properties. In Section 4, using MCFs we develop a simple, scalable approach to integrating gradient estimators in the SK metamodel for enhancement. We present numerical experiments in Section 5 and conclude in Section 6.

2 Problem Formulation

We briefly review both the SK metamodel and prior methods for enhancing SK with gradient information, and highlight the scalability issue during the process.

2.1 Stochastic Kriging

Let $\mathsf{Z}(\boldsymbol{x})$ denote the response surface of interest, where $\boldsymbol{x} = (x^1, \ldots, x^d)$ is the design variable. Given an experimental design $\{(\boldsymbol{x}_i, m_i) : i = 1, \ldots, n\}$, the simulation model is executed at design point $\boldsymbol{x}_i$ for m_i times independently and yields the simulation outputs $\{z_j(\boldsymbol{x}_i) : j = 1, \ldots, m_i\}$. The SK metamodel is concerned with predicting the responses at an arbitrary location by interpolating the simulation outputs properly. Specifically, it assumes that $\mathsf{Z}(\boldsymbol{x})$ is a realization of a Gaussian process, which effectively introduces spatial correlations between the responses. That is,

$$\mathsf{Z}(\boldsymbol{x}) = \beta + \mathsf{M}(\boldsymbol{x}), \tag{1}$$

where β is a unknown parameter and M is a mean zero Gaussian process with covariance function $k(\boldsymbol{x}, \boldsymbol{y}) \coloneqq \mathrm{Cov}(\mathsf{M}(\boldsymbol{x}), \mathsf{M}(\boldsymbol{y}))$. For instance, a common choice is the squared exponential covariance function of the form $k(\boldsymbol{x}, \boldsymbol{y}) = \tau^2 \exp[-\sum_{r=1}^{d} \rho^r (x^r - y^r)^2]$, where $(\tau^2, \rho^1, \ldots, \rho^d)$ are unknown parameters. Then, the simulation outputs can be expressed as

$$z_j(\boldsymbol{x}_i) = \mathsf{Z}(\boldsymbol{x}_i) + \epsilon_j(\boldsymbol{x}_i), \quad j = 1, \ldots, m_i,\ i = 1, \ldots, n, \tag{2}$$

where the $\epsilon_j(\boldsymbol{x}_i)$'s are independent simulation errors having a normal distribution with mean zero. Let $\bar{z}(\boldsymbol{x}_i) \coloneqq m_i^{-1} \sum_{j=1}^{m_i} z_j(\boldsymbol{x}_i)$. The SK metamodel makes predictions by interpolating $\bar{\boldsymbol{z}} \coloneqq (\bar{z}(\boldsymbol{x}_1), \ldots, \bar{z}(\boldsymbol{x}_n))$.

Let $\boldsymbol{\Sigma}_{\mathsf{M}}$ denote the $n \times n$ covariance matrix of $(\mathsf{M}(\boldsymbol{x}_1), \ldots, \mathsf{M}(\boldsymbol{x}_n))$, and $\boldsymbol{\Sigma}_\epsilon$ denote that of $(\bar{\epsilon}(\boldsymbol{x}_1), \ldots, \bar{\epsilon}(\boldsymbol{x}_n))$. Notice that $\boldsymbol{\Sigma}_\epsilon$ is a diagonal matrix due to the independence assumption. Then, for any $\boldsymbol{x}_0 \in \mathbb{R}^d$, the SK predictor is

$$\widehat{\mathsf{Z}}(\boldsymbol{x}_0) = \beta + \boldsymbol{k}^\intercal(\boldsymbol{x}_0, \cdot)[\boldsymbol{\Sigma}_{\mathsf{M}} + \boldsymbol{\Sigma}_\epsilon]^{-1}(\bar{\boldsymbol{z}} - \beta \mathbf{1}_n), \tag{3}$$

where $\boldsymbol{k}(\boldsymbol{x}_0, \cdot) \coloneqq (k(\boldsymbol{x}_0, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}_0, \boldsymbol{x}_n))^\intercal$ and $\mathbf{1}_n$ is the $n \times 1$ vector of ones.

In practice, β and the parameters for defining the function $k(\boldsymbol{x}, \boldsymbol{y})$, say $\boldsymbol{\theta}$, are unknown. A typical treatment is the maximum-likelihood estimation (MLE), which maximizes the following log-likelihood function over the parameter space,

$$\ell(\beta, \boldsymbol{\theta}) \coloneqq -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log[\det(\boldsymbol{\Sigma}_{\mathsf{M}} + \boldsymbol{\Sigma}_\epsilon)] - \frac{1}{2}(\bar{\boldsymbol{z}} - \beta \mathbf{1}_n)^\intercal [\boldsymbol{\Sigma}_{\mathsf{M}} + \boldsymbol{\Sigma}_\epsilon]^{-1}(\bar{\boldsymbol{z}} - \beta \mathbf{1}_n), \tag{4}$$

where $\det(\cdot)$ means the determinant; see Ankenman et al. (2010). The estimates are then plugged into (3).

The bottleneck for computing (3) and (4) is the inversion of $[\boldsymbol{\Sigma}_{\mathsf{M}} + \boldsymbol{\Sigma}_\epsilon]$, whose time complexity is $O(n^3)$ and becomes prohibitive for large n. The poor scalability is also attributed to the fact that for large n, $\boldsymbol{\Sigma}_{\mathsf{M}}$ usually becomes ill-conditioned (i.e., nearly singular) while searching over the parameter space for maximizing (4), especially if the squared exponential covariance function is adopted; see Ababou et al. (1994) and references therein. In the context of stochastic simulation,

the diagonal entries of $\mathbf{\Sigma}_\epsilon$ are typically small because of the repeated sampling at each design point. Although the presence of $\mathbf{\Sigma}_\epsilon$ reduces the condition number of the matrix to be inverted, the reduction is thus not significant. The ill-conditionedness of $[\mathbf{\Sigma}_\mathsf{M} + \mathbf{\Sigma}_\epsilon]$ may cause failure of the MLE, resulting in unreliable estimates of the parameters and erroneous response predictions; see also Chapter 5.4 of Fang et al. (2006). Next, we show that the scalability issue is even more severe if gradient information is incorporated, which is supposedly to improve the prediction accuracy of the SK metamodel in the first place.

2.2 Enhancing Stochastic Kriging with Gradients

To fix the idea, suppose that the j-th run of the simulation model at each design point $\boldsymbol{x}_i$ produces an gradient estimate $\boldsymbol{g}_j(\boldsymbol{x}_i) = (g_j^1(\boldsymbol{x}_1), \ldots, g_j^d(\boldsymbol{x}_n))^\intercal$, in addition to the response estimate $z_j(\boldsymbol{x}_i)$, namely,

$$g_j^r(\boldsymbol{x}_i) = \mathsf{G}^r(\boldsymbol{x}_i) + \delta_j^r(\boldsymbol{x}_i), \quad r = 1, \ldots, d, \tag{5}$$

where $\mathsf{G}^r(\boldsymbol{x}_i)$ is the true r-th partial derivative and $\delta_j^r(\boldsymbol{x}_i)$ is the simulation error with mean zero. It is worth mentioning that the simulation errors $(\epsilon_j(\boldsymbol{x}_i), \delta_j^1(\boldsymbol{x}_i), \ldots, \delta_j^d(\boldsymbol{x}_i))$ at the same design point are correlated in general, because the gradient estimate $\boldsymbol{g}_j(\boldsymbol{x}_i)$ can be viewed as a deterministic transformation of the response estimate $z_j(\boldsymbol{x}_i)$ with the use of IPA or LR.

The *gradient extrapolated stochastic kriging* (GESK) developed in Qu and Fu (2014) leverages the gradient information in an *indirect* fashion as follow. In the neighborhood of each design point $\boldsymbol{x}_i$, it converts the gradient information to a "pseudo" observation of the response surface by linear extrapolation, i.e.,

$$z_j(\tilde{\boldsymbol{x}}_i) = z_j(\boldsymbol{x}_i) + \boldsymbol{g}_j^\intercal(\boldsymbol{x}_i)\Delta\boldsymbol{x}_i, \tag{6}$$

where $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + \Delta\boldsymbol{x}_i$ and $\Delta\boldsymbol{x}_i \in \mathbb{R}^d$ represents the direction and the step size of the extrapolation. Obviously, $\|\Delta\boldsymbol{x}_i\|$ ought to be small to make the extrapolation accurate enough. Then, SK can be performed by interpolating the augmented data $(\bar{z}(\boldsymbol{x}_1), \ldots, \bar{z}(\boldsymbol{x}_n), \bar{z}(\tilde{\boldsymbol{x}}_1), \ldots, \bar{z}(\tilde{\boldsymbol{x}}_n))$ as if they were the simulation outputs from the design points $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_n)$. The size of covariance matrix involved then becomes $2n \times 2n$ due to the doubled data volume.

Furthermore, for a multidimensional design space, multiple pseudo responses can be introduced for each design point $\boldsymbol{x}_i$ by varying $\Delta\boldsymbol{x}_i$ in order to make the most of the gradient information. For instance, a natural choice is to introduce one pseudo response along each axis of the d-dimensional design space, leading to d pseudo responses at each design point and a $(d+1)n \times (d+1)n$ covariance matrix to invert.

A different approach to integrating gradient information in SK can be found in Chen et al. (2013), where it is called *stochastic kriging with gradient estimators* (SKG). It leverages the gradient information *directly* by modeling the gradient estimator as partial derivatives of the response surface $\mathsf{Z}(\boldsymbol{x})$. This requires one to calculate the covariances between the response surface and the d surfaces for the partial derivatives, so the covariance matrix to invert is of size $(d+1)n \times (d+1)n$ as well.

Consequently, the scalability issue would become significantly more severe than it already is if

one seeks to integrate gradient estimates with SK using either of the above approaches to enhance the prediction performance.

3 Markovian Covariance Functions

In Ding and Zhang (2018), a new class of covariance functions are constructed to permit the use of SK for large datasets. They are called Markovian covariance functions (MCFs), because the Gaussian process with an MCF on a regular lattice becomes a Gaussian Markov random field (Rue and Held 2005). In this section, we briefly review the basic theory of MCFs and show why they make SK scalable. We first discuss 1-dimensional MCFs and then extend the concept to the multidimensional case.

Definition 1. Suppose that $p, q : \mathbb{R} \mapsto \mathbb{R}_+$ are two positive continuous functions that satisfy $p(x)q(y) - p(y)q(x) < 0$ for all $x < y$. Then, $k(x, y) = p(x)q(y)\,\mathbb{I}_{\{x\le y\}} + p(y)q(x)\,\mathbb{I}_{\{x>y\}}$ is called a 1-dimensional MCF.

It can be seen easily that the squared exponential covariance function that is commonly used in SK is not an MCF. We refer to Ding and Zhang (2018) for a principled approach to constructing MCFs based on ordinary differential equations. Two representative examples are

$$k(x, y; \tau^2, \rho) = \tau^2 \left[e^{\rho x} e^{-\rho y}\,\mathbb{I}_{\{x\le y\}} + e^{\rho y} e^{-\rho x}\,\mathbb{I}_{\{x>y\}}\right] = \tau^2 e^{-\rho|x-y|}, \tag{7}$$

for $x, y \in \mathbb{R}$, and

$$k(x, y; \tau^2, \nu) \coloneqq \begin{cases} \tau^2 \left[\sin(\gamma x)\sin(\gamma(1-y))\,\mathbb{I}_{\{x\le y\}} + \sin(\gamma y)\sin(\gamma(1-x))\,\mathbb{I}_{\{x>y\}}\right], & \text{if } \nu < 0, \\ \tau^2 \left[x(1-y)\,\mathbb{I}_{\{x\le y\}} + y(1-x)\,\mathbb{I}_{\{x>y\}}\right], & \text{if } \nu = 0, \\ \tau^2 \left[\sinh(\gamma x)\sinh(\gamma(1-y))\,\mathbb{I}_{\{x\le y\}} + \sinh(\gamma y)\sinh(\gamma(1-x))\,\mathbb{I}_{\{x>y\}}\right], & \text{if } \nu > 0, \end{cases}$$

for $x, y \in (0, 1)$, where $\gamma = \sqrt{|\nu|}$.

MCFs have two critical features that distinguish themselves from others: (i) the associated covariance matrix can be inverted analytically and (ii) the inverse matrix is *sparse*, i.e., most of its entries are zero. Indeed, the following theorem asserts that there are at most $(3n - 2)$ nonzero entries in the inverse matrix of size $n \times n$.

Theorem 1. *Let $k(x, y)$ be a 1-dimensional MCF and $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ be a matrix whose (i, j)-th entry is $k(x_i, x_j)$, where $x_1 < \cdots < x_n \in \mathbb{R}$ with $n \ge 3$. Let $p_i = p(x_i)$ and $q_i = q(x_i)$. Then,*

(i) $\boldsymbol{K}^{-1}$ is a tridiagonal matrix, i.e., $(\boldsymbol{K}^{-1})_{i,j} = 0$ if $|i - j| \ge 2$;

(ii) the nonzero entries of $\boldsymbol{K}^{-1}$ *are given as follows*

$$(\boldsymbol{K}^{-1})_{i,i} = \begin{cases} \dfrac{p_2}{p_1(p_2q_1 - p_1q_2)}, & \text{if } i = 1, \\ \dfrac{p_{i+1}q_{i-1} - p_{i-1}q_{i+1}}{(p_iq_{i-1} - p_{i-1}q_i)(p_{i+1}q_i - p_iq_{i+1})}, & \text{if } 2 \le i \le n-1, \\ \dfrac{q_{n-1}}{q_n(p_nq_{n-1} - p_{n-1}q_n)}, & \text{if } i = n, \end{cases}$$

and

$$(\boldsymbol{K}^{-1})_{i-1,i} = (\boldsymbol{K}^{-1})_{i,i-1} = \frac{-1}{p_iq_{i-1} - p_{i-1}q_i}, \quad i = 2, \ldots, n.$$

Proof. See Theorems 1 – 3 of Ding and Zhang (2018). □

It turns out that the two features, that is, the analytical invertibility of the covariance matrix and the sparsity in the inverse, improve substantially the computational efficiency and the numerical stability of the computation of $[\boldsymbol{\Sigma}_{\mathsf{M}} + \boldsymbol{\Sigma}_\epsilon]^{-1}$ in (3) and (4). To see this, we apply the Woodbury matrix identity (Horn and Johnson (2012), §0.7.4),

$$[\boldsymbol{\Sigma}_{\mathsf{M}} + \boldsymbol{\Sigma}_\epsilon]^{-1} = \boldsymbol{\Sigma}_{\mathsf{M}}^{-1} - \boldsymbol{\Sigma}_{\mathsf{M}}^{-1}[\boldsymbol{\Sigma}_{\mathsf{M}}^{-1} + \boldsymbol{\Sigma}_\epsilon^{-1}]^{-1}\boldsymbol{\Sigma}_{\mathsf{M}}^{-1}. \tag{8}$$

By virtue of Theorem (1), if the design space is 1-dimensional and the covariance function of the Gaussian process $\mathsf{M}(x)$ in (1) is an MCF, then $\boldsymbol{\Sigma}_{\mathsf{M}}^{-1}$ has $(3n-2)$ nonzero entries whose expressions are available in closed form, and thus it takes $O(n)$ operations to compute $\boldsymbol{\Sigma}_{\mathsf{M}}^{-1}$. Moreover, since $\boldsymbol{\Sigma}_\epsilon$ is diagonal, $[\boldsymbol{\Sigma}_{\mathsf{M}}^{-1} + \boldsymbol{\Sigma}_\epsilon^{-1}]$ is sparse as well, so it can be inverted with $O(n^2)$ operations by leveraging sparse linear algebra. The matrix multiplication in (8) can be computed with $O(n^2)$ operations because $\boldsymbol{\Sigma}_{\mathsf{M}}^{-1}$ is sparse. Therefore, the overall time complexity for computing $[\boldsymbol{\Sigma}_{\mathsf{M}} + \boldsymbol{\Sigma}_\epsilon]^{-1}$ using (8) is $O(n^2)$.

In addition to the reduction in time complexity, the use of MCFs also improves the numerical stability. To see this, notice that when computing the right-hand-side of (8), numerical procedures for matrix inversion such as Gaussian elimination are only needed for computing $[\boldsymbol{\Sigma}_{\mathsf{M}}^{-1} + \boldsymbol{\Sigma}_\epsilon^{-1}]^{-1}$, since $\boldsymbol{\Sigma}_{\mathsf{M}}^{-1}$ is analytically available. In general, we anticipate the diagonal entries of $\boldsymbol{\Sigma}_\epsilon$ to be small, or at least can be made so by increasing the number of replications. Then, the diagonal entries of $\boldsymbol{\Sigma}_\epsilon^{-1}$ ought to be sufficiently large in practice. Hence, $[\boldsymbol{\Sigma}_{\mathsf{M}}^{-1} + \boldsymbol{\Sigma}_\epsilon^{-1}]$ is expected to be far away from singularity and can be inverted in a numerically stable manner.

Definition 2. Let $k^r(\cdot,\cdot)$ be a 1-dimensional MCF for each $r = 1, \ldots, d$. Then, $k(\boldsymbol{x}, \boldsymbol{y}) = \prod_{r=1}^d k^r(x^r, y^r)$, for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ is called a d-dimensional MCF.

The product form of the d-dimensional MCFs implies that the covariance matrix $\boldsymbol{\Sigma}_{\mathsf{M}}$ can be written in the form of the Kronecker product, provided that the design points form a regular lattice.

Proposition 1. *Suppose that the design points can be expressed as a Cartesian product, i.e., there*

exist positive integers $n^1, \ldots, n^d$ *and* $x_1^r, \cdots, x_{n^r}^r \in \mathbb{R}$, $r = 1, \ldots, d$ *such that* $n = \prod_{r=1}^d n^r$ *and*

$$\mathcal{X} \coloneqq \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} = \bigtimes_{r=1}^d \{x_1^r, \ldots, x_{n^r}^r\}. \tag{9}$$

Let k *and* k^r, $r = 1, \ldots, d$, *be a* d*-dimensional MCF and its associated 1-dimensional MCFs, respectively. Let* $\boldsymbol{K}$ *denote the covariance matrix associated with* k *and* $\mathcal{X}$*, and* $\boldsymbol{K}^k$ *denote that associated with* k^r *and* $\{x_1^r, \ldots, x_{n^r}^r\}$, $r = 1, \ldots, d$. *Then,* $\boldsymbol{K} = \bigotimes_{r=1}^d \boldsymbol{K}^r$ *and* $\boldsymbol{K}^{-1} = \bigotimes_{r=1}^d (\boldsymbol{K}^r)^{-1}$.

Proof. It is straightforward by the definition and properties of the Kronecker product of matrices; see, e.g., Chapter 13 of Laub (2005). □

Hence, $\boldsymbol{K}^{-1}$ is sparse if the d-dimensional design points form a regular lattice, thanks to the tridiagonal structure of each $(\boldsymbol{K}^r)^{-1}$ by Theorem 1. Then, the Woodbury matrix identity (8) can be applied readily to reduce computational cost of multi-dimensional SK metamodels.

4 Scalable GESK

Recall that by transforming gradient estimates to pseudo observations of the response surface via linear extrapolation, GESK essentially provides the SK metamodel with an augmented set of design points and observed responses. Following Proposition 1, the computational tractability of MCFs can be adopted in GESK naturally, if the augmented set of design points form a regular lattice. In the following, we show that this can be done easily provided that the original design points form a regular lattice.

Let $\boldsymbol{\alpha} = (\alpha^1, \ldots, \alpha^d)^\intercal$ be a vector with $\alpha^r = 0, 1$, $r = 1, \ldots, d$. Then, $\boldsymbol{\alpha}$ has 2^d possible values and they form a d-dimensional unit hypercube. For each design point $\boldsymbol{x}_i$, $i = 1, \ldots, n$, we consider a design point in its neighborhood as follows,

$$\tilde{\boldsymbol{x}}_i^{\boldsymbol{\alpha}} \coloneqq \boldsymbol{x}_i + \eta \boldsymbol{\alpha}, \tag{10}$$

for some small $\eta > 0$. Following the linear extrapolation (6), we construct the pseudo observations at $\tilde{\boldsymbol{x}}_i^{\boldsymbol{\alpha}}$,

$$z_j(\tilde{\boldsymbol{x}}_i^{\boldsymbol{\alpha}}) = z_j(\boldsymbol{x}_i) + \eta \boldsymbol{g}_j^\intercal(\boldsymbol{x}_i)\boldsymbol{\alpha}, \quad j = 1, \ldots, m_i. \tag{11}$$

Clearly, $\tilde{\boldsymbol{x}}_i^{\boldsymbol{\alpha}} = \boldsymbol{x}_i$ if $\boldsymbol{\alpha} = \boldsymbol{0}$ and $\{\tilde{\boldsymbol{x}}_i^{\boldsymbol{\alpha}} : \alpha^r = 0, 1,\ r = 1, \ldots, d\}$ form a d-dimensional hypercube with $\boldsymbol{x}_i$ being one of the corners. Consequently, if the original set of design points $\mathcal{X}$ form a regular lattice (9), then the augmented set of the design points $\tilde{\mathcal{X}} \coloneqq \{\tilde{\boldsymbol{x}}_i^{\boldsymbol{\alpha}} : \alpha^r = 0, 1,\ r = 1, \ldots, d,\ i = 1, \ldots, n\}$ also form a regular lattice, i.e.,

$$\tilde{\mathcal{X}} = \bigtimes_{r=1}^d \{x_1^r, \ldots, x_{n^r}^r, x_1^r + \eta, \ldots, x_{n^r}^r + \eta\}.$$

We then use the augmented data $\{\bar{z}(\tilde{\boldsymbol{x}}_i^{\boldsymbol{\alpha}}) : \tilde{\boldsymbol{x}}_i^{\boldsymbol{\alpha}} \in \tilde{\mathcal{X}}\}$ to construct the SK metamodel with an MCF,

where

$$\bar{z}(\tilde{\boldsymbol{x}}_i^{\boldsymbol{\alpha}}) = \frac{1}{m_i}\sum_{j=1}^{m_i} z_j(\tilde{\boldsymbol{x}}_i^{\boldsymbol{\alpha}}).$$

It is easy to see that $\tilde{\mathfrak{X}}$ has $2^d n$ design points. However, we stress here that the computation of $\boldsymbol{\Sigma}_{\mathsf{M}}^{-1}$ is not done via a generic numerical inversion subroutine applied directly to $\boldsymbol{\Sigma}_{\mathsf{M}}$, a $2^d n \times 2^d n$ matrix, which would be computationally prohibitive. Instead, Proposition 1 suggests that by leveraging the lattice structure of $\tilde{\mathfrak{X}}$ and the Kronecker product, its computation is reduced to the multiplication of d matrices, each of which has a much smaller size (i.e., $2n^r \times 2n^r$ for $r = 1, \ldots, d$), and can be inverted analytically with the use of MCFs (Theorem 1). Therefore, the seemingly excessive volume of $\tilde{\mathfrak{X}}$ does not incur prohibitive computational cost.

Remark 1. It may happen in practice that for some r, a computationally efficient estimator does not exist for the partial derivative $G^r(\boldsymbol{x})$ and we need to exclude it from GESK. Our methodology remains valid – one can simply set $\alpha^r = 0$ in (10) and (11) for such r. Notice that the volume of the augmented data is reduced by a factor of 2 for each partial derivative excluded from GESK.

The choice of the step size η is crucial to the performance of GESK. Treating η as an additional parameter besides the unknown parameters for defining the MCF, we adopt the penalized maximum likelihood estimation (PMLE) approach proposed in Qu and Fu (2014) to determine their values. Specifically, we maximize the following penalized log-likelihood function

$$\ell_\lambda(\beta, \boldsymbol{\theta}, \eta) := \ell(\beta, \boldsymbol{\theta}) - \lambda\eta^{-2},$$

where $\ell(\beta, \boldsymbol{\theta})$ is the log-likelihood function (4) and λ is a regularization parameter which can be selected via a standard cross validation (CV) approach; see Qu and Fu (2014) for more details.

5 Numerical Experiments

We present two examples to demonstrate the scalability of our approach to enhancing SK with gradient information. In both examples, we use the following covariance function

$$k(\boldsymbol{x}, \boldsymbol{y}) = \tau^2 \exp\left(-\sum_{r=1}^{d} \rho^r |x^r - y^r|\right),$$

where $(\tau^2, \rho^1, \ldots, \rho^d)$ are unknown parameters. Then, this is a d-dimensional MCF by Definition 2 and (7).

5.1 Griewank Function

Consider the Griewank function which has a relatively complex response surface,

$$\mathsf{Z}(\boldsymbol{x}) = \frac{1}{10}\sum_{r=1}^{d}\left(\frac{x^r}{20}\right)^2 - \prod_{r=1}^{d}\cos\left(\frac{x^r}{\sqrt{r}}\right) + 1;$$

see Figure 1 for an illustration. Then, the r-th partial derivative is

$$\mathsf{G}^r(\boldsymbol{x}) = \frac{x^r}{2000} + \frac{1}{\sqrt{r}} \sin\left(\frac{x^r}{\sqrt{r}}\right) \prod_{s \neq r} \cos\left(\frac{x^s}{\sqrt{s}}\right), \quad r = 1, \ldots, d.$$

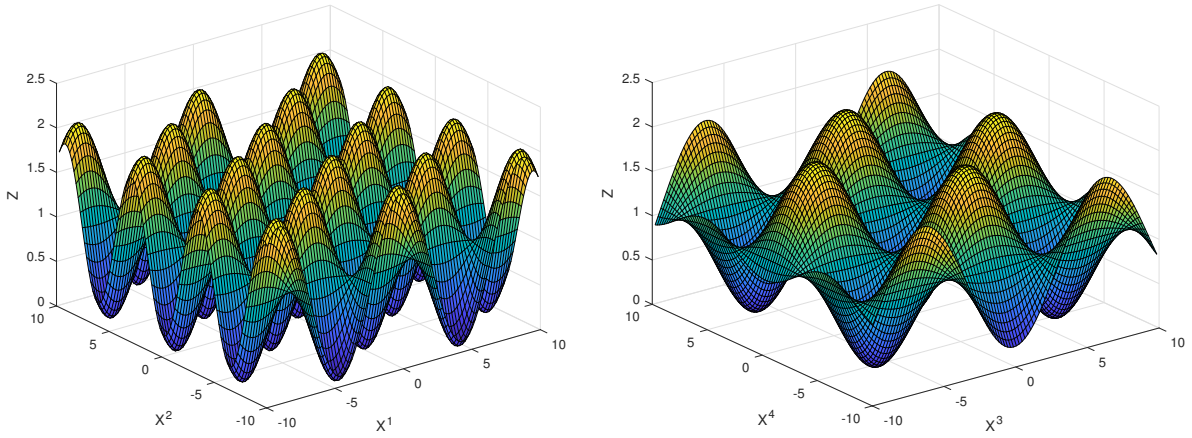

Figure 1: Two-dimensional projections of the four-dimensional Griewank function by fixing $x^3 = x^4 = 0$ (Left) and fixing $x^1 = x^2 = 0$ (Right), respectively.

We take $d = 4$ and let the design space be $[-10, 10]^4$. We choose the design points to form a regular lattice (9) as follows. We set $n^1 = \cdots = n^r$ and for each $r = 1, \ldots, 4$, we set $\{x_1^r, \ldots, x_{n^r}^r\}$ to be equally spaced on $[-10, 10]$ with $x_1^r = -10$ and $x_{n^r}^r = 10$. Since we aim to show the scalability of our approach, we set the number of design points to be large and particularly, consider three cases: $n = 5^4$, 8^4, and 10^4.

Suppose that at each design point $\boldsymbol{x}_i$ and for each replication j, the error for simulating the true response surface in (2) is $\epsilon_j(\boldsymbol{x}_i) \sim \mathcal{N}(0, 0.5)$, and the error for estimating the gradient in (5) is $\delta_j^r(\boldsymbol{x}_i) \sim \mathcal{N}(0, 1)$, $r = 1, \ldots, d$. For simplicity, we assume that the number of replications $m_i = 1000$, $i = 1, \ldots, n$, and that these simulation errors, $(\epsilon_j(\boldsymbol{x}_i), \delta_j^1(\boldsymbol{x}_i), \ldots, \delta_j^d(\boldsymbol{x}_i))$, are mutually independent.

To measure the prediction accuracy, we compute the following *empirical integrated MSE* (EIMSE),

$$\text{EIMSE} = \frac{1}{N} \sum_{i=1}^{N} (\widehat{\mathsf{Z}}(\underline{\boldsymbol{x}}_i) - \mathsf{Z}(\underline{\boldsymbol{x}}_i))^2, \tag{12}$$

where $\widehat{\mathsf{Z}}(\underline{\boldsymbol{x}}_i)$ is the response predicted by either SK or GESK, and $\{\underline{\boldsymbol{x}}_i : i = 1, \ldots, N\}$ are chosen randomly from the design space. We perform 100 macro-replications and in each we generate $N = 1000$ random points for computing EIMSE. The 100 realizations of EIMSE for a given set of design points and a given prediction method (SK or GESK) are illustrated via a boxplot in Figure 2.

That GESK has a higher prediction accuracy than SK has been shown in Qu and Fu (2014). But it was done there through small-scale examples with one-dimensional response surfaces and

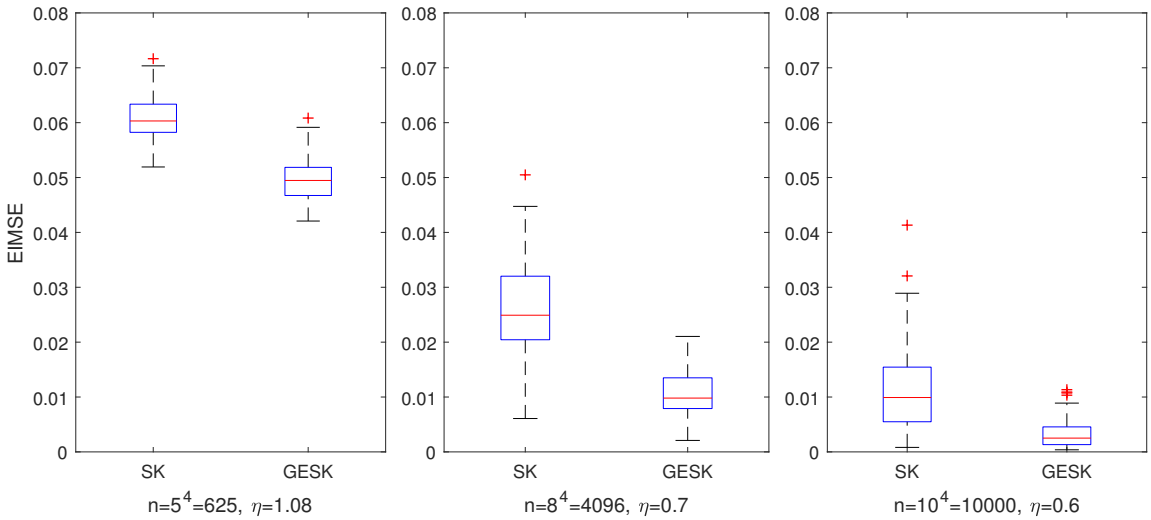

Figure 2: EIMSE for predicting the four-dimensional Griewank function.

at most 20 design points. Our experience suggests that if the covariance function is not an MCF, e.g., the widely used squared exponential covariance function, then the computation of the SK metamodel would run into severe numerical issues and mostly yield erroneous prediction results for $n > 100$. By contrast, thanks to MCFs, our scalable approach can now handle large-scale problems in a numerically stable manner. Indeed, the increased number of design points improves the prediction accuracy of both SK and GESK substantially.

5.2 Closed-Loop Flexible Assembly System

This example is adopted from Suri and Leung (1987) and Chen et al. (2013). We consider a closed-loop flexible assembly system (CLFAS) consisting of four workstations that are linked by a conveyor; see Figure 3.

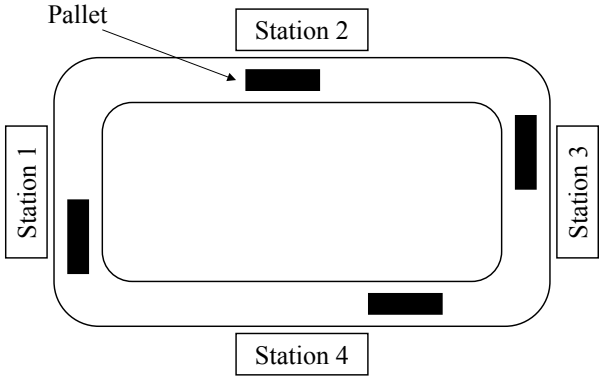

Figure 3: Schematic diagram of CLFAS.

There are four pallets carrying parts on the conveyor. A part enters the CLFAS through station 1, goes through station 2, 3, and 4 on a pallet, and leaves the system through station 1. The primary

source of randomness stems from machine jams: a part may cause a station to jam and when it happens at station r, it takes a random amount of time R^r to clear the machine there. Let T^r denote the operation time of a part at station r, $r = 1, \ldots, 4$. Then,

$$T^r = x^r + \mathbb{I}\{\text{jam at station r}\}R^r,$$

where x^r is the *deterministic* machine cycle time and $\mathbb{I}\{\cdot\}$ is the indicator function. We assume that each station jams with probability 0.5% and let $R^1, \ldots, R^4$ be independent random variables uniformly distributed on $[0.05, 0.55]$. We also assume that the times between stations are negligible.

Suppose that each station allows only one part to queue in front of it. Hence, a station may be blocked and cannot release a finished part if the downstream station is full. The response surface of interest is the expected throughput $\mathsf{Z}(\boldsymbol{x})$ of the first $P = 1000$ parts finished by the system, where $\boldsymbol{x} = (x^1, \ldots, x^4) \in [0.02, 0.1]^4$; see Figure 4 for an illustration based on extensive simulation.

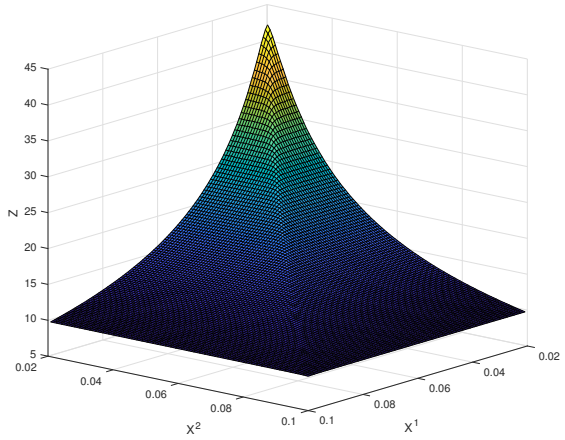

Figure 4: The expected throughput of the CLFAS as a function of x^1 and x^2 by fixing $x^3 = x^4 = 0.02$.

The algorithm for estimating both $\mathsf{Z}(\boldsymbol{x})$ and its partial derivatives $\mathsf{G}^r(\boldsymbol{x})$, $r = 1, \ldots, 4$, is given in Appendix. The setup of the numerical experiment is similar to that in §5.1. The design points are chosen to form an equally spaced regular lattice with $n = 3^4$ or 5^4. At each design point $\boldsymbol{x}_i$, we run 100 replications. The variance of $\epsilon_j(\boldsymbol{x}_i)$, the variance of $\delta_j^r(\boldsymbol{x}_i)$, $r = 1, \ldots, 4$, and their covariances are estimated from the samples; see Qu and Fu (2014) for details. We perform 100 macro-replications and in each we compute the EIMSE (12) based on $N = 1000$ random points in the design space. Because the true response is unknown, we use instead the estimated value from 1000 replications, for which the estimation error is negligible. The results are shown in Figure 5.

In Chen et al. (2013), the CLFAS example is used to demonstrate the benefit of incorporating gradient information with the SK metamodel, albeit in a different way than GESK. But similar to Qu and Fu (2014), their use of gradient information is also at a small scale, with only 25 design points. We instead show that the large-scale use of the SK metamodel integrated with gradient estimators is made feasible by our approach and it can improve prediction accuracy significantly.

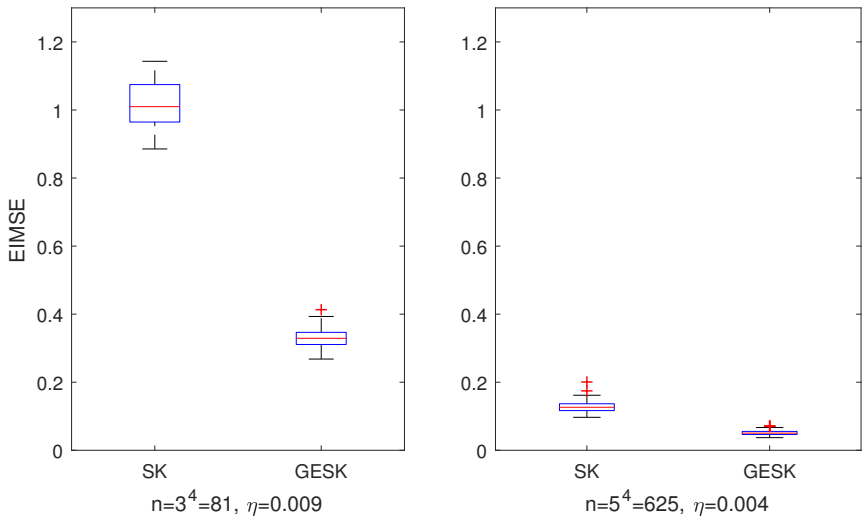

Figure 5: EIMSE for predicting the expected throughput of the CLFAS.

6 conclusions

Two distinct methods for enhancing SK with gradients have been developed in simulation literature, namely, SKG and GESK. Both suffer from the poor scalability when applied to large datasets. Taking advantage of MCFs, we have proposed in this paper a simple approach that significantly improves the scalability of GESK so that it can be used for simulation models with a high-dimensional design space.

However, the linear extrapolation in GESK introduces an approximation error that is hard to characterize in the prediction. The choice of the step size is critical to the performance of GESK and needs to be chosen carefully by virtue of cross validation, which can be computationally expensive. Moreover, to accommodate the lattice structure in the design points required by MCFs, the number of pseudo observations at each original design point is $(2^d - 1)$. In the light of the d partial derivatives, this conceivably introduces a great deal of redundancy in the use of the gradient information.

SKG, on the other hand, uses the gradient estimators directly and does not incur the computational overhead as GESK does. It is therefore of great interest to apply MCFs to improve the scalability of SKG as well. Nevertheless, SKG requires one to characterize the correlation structure of the gradient surface. It is still an open question whether MCFs would make the joint covariance matrix between the response surface and the gradient surface analytically invertible and imply certain sparsity in the inverse matrix. We leave it to future research.

Acknowledgments

The authors gratefully acknowledge the support of the Hong Kong Research Grant Council under Project No. 16211417.

A Algorithm for CLFAS

For ease of reference, we restate below the algorithm in Chen et al. (2013) for estimating both $\mathsf{Z}(\boldsymbol{x})$ and its IPA gradient estimator $\mathsf{G}^r(\boldsymbol{x})$, $r = 1, \ldots, 4$.

Step 1. Initialize variables $A_{s,r} \leftarrow 0$ for $s, r = 1, \ldots, 4$.

Step 2. At the end of an operation at station s with operation time T^s, set $A_{s,s} \leftarrow A_{s,s} + 1$.

Step 3. If a pallet leaving station s going to station t terminates an idle period of station t, then set

$$A_{t,r} \leftarrow A_{s,r}, \quad r = 1, \ldots, 4.$$

Step 4. If a pallet leaving station s going to station t terminates a blocked period of station s, then set

$$A_{s,r} \leftarrow A_{t,r}, \quad r = 1, \ldots, 4.$$

Step 5. Let L denote the total length of simulation in time units after P parts are completed by the CLFAS. Estimate the throughput and its gradient by

$$z(\boldsymbol{x}) = \frac{P}{L} \quad \text{and} \quad g^r(\boldsymbol{x}) = -\frac{z(\boldsymbol{x})}{L} A_{4,r}, \quad r = 1, \ldots, 4.$$

References

Ababou, R., A. C. Bagtzoglou, and E. F. Wood (1994). On the condition number of covariance matrices in kriging, estimation, and simulation of random fields. *Mathematical Geology 26*(1), 99–133.

Ankenman, B., B. L. Nelson, and J. Staum (2010). Stochastic kriging for simulation metamodeling. *Operations Research 58*(2), 371–382.

Barton, R. R., B. L. Nelson, and W. Xie (2014). Quantifying input uncertainty via simulation confidence intervals. *INFORMS Journal on Computing 26*(1), 74–87.

Chen, X., B. Ankenman, and B. L. Nelson (2013). Enhancing stochastic kriging metamodels with gradient estimators. *Operations Research 61*(2), 512–528.

Chen, X. and K.-K. Kim (2016). Efficient VaR and CVaR measurement via stochastic kriging. *INFORMS Journal on Computing 28*(4), 629–644.

Ding, L. and X. Zhang (2018). Scalable stochastic kriging with Markovian covariances. https://arxiv.org/abs/1803.02575.

Fang, K.-T., R. Li, and A. Sudjianto (2006). *Design and Modeling for Computer Experiments*. Boca Raton, FL: Chapman & Hall/CRC.

Horn, R. A. and C. R. Johnson (2012). *Matrix Analysis* (2nd ed.). New York: Cambridge University Press.

Kleijnen, J. P. C. (2015). Kriging metamodels and their designs. In *Design and Analysis of Simulation Experiments* (2nd ed.)., Chapter 5, pp. 179–239. Switzerland: Springer.

Laub, A. J. (2005). *Matrix Analysis for Scientists and Engineers*. Philadelphia, PA: SIAM.

L'Ecuyer, P. (1990). A unified view of the IPA, SF, and LR gradient estimation techniques. *Management Science 36*(11), 1364–1383.

Qu, H. and M. C. Fu (2014). Gradient extrapolated stochastic kriging. *ACM Transactions on Modeling and Computer Simulation 24*(4), 23:1–23:25.

Quan, N., J. Yin, S. H. Ng, and L. H. Lee (2013). Simulation optimization via kriging: A sequential search using expected improvement with computing budget constraints. *IIE Transactions 45*(7), 763–780.

Risk, J. and M. Ludkovski (2016). Statistical emulators for pricing and hedging longevity risk products. *Insurance: Mathematics and Economics 68*, 45–60.

Rue, H. and L. Held (2005). *Gaussian Markov Random Fields: Theory and Applications*. Boca Raton, FL: Chapman & Hall/CRC.

Shen, H., L. J. Hong, and X. Zhang (2018). Enhancing stochastic kriging for queueing simulation with stylized models. *IISE Transactions.* (https://doi.org/10.1080/24725854.2018.1465242).

Sun, L., L. J. Hong, and Z. Hu (2014). Balancing exploitation and exploration in discrete optimization via simulation through a Gaussian process-based search. *Operations Research 62*(6), 1416–1438.

Suri, R. and Y. T. Leung (1987). Single run optimization of a SIMAN model for closed loop flexible systems. In A. T. et al. (Ed.), *Proceedings of the 1987 Winter Simulation Conference*, Piscataway, New Jersey, pp. 738–748. IEEE.

Xie, W., B. L. Nelson, and R. R. Barton (2014). A Bayesian framework for quantifying uncertainty in stochastic simulation. *Operations Research 62*(6), 1439–1452.