

# 5

## Kriging Metamodels and Their Designs

This chapter is organized as follows. Section 5.1 introduces Kriging, which is also called Gaussian process (GP) or spatial correlation modeling. Section 5.2 details so-called *ordinary Kriging* (OK), including the basic Kriging assumptions and formulas assuming deterministic simulation. Section 5.3 discusses parametric bootstrapping and conditional simulation for estimating the variance of the OK predictor. Section 5.4 discusses *universal Kriging* (UK) in deterministic simulation. Section 5.5 surveys designs for selecting the input combinations that gives input/output data to which Kriging metamodels can be fitted; this section focuses on *Latin hypercube sampling* (LHS) and customized sequential designs. Section 5.6 presents *stochastic Kriging* (SK) for random simulations. Section 5.7 discusses bootstrapping with acceptance/rejection for obtaining Kriging predictors that are monotonic functions of their inputs. Section 5.8 discusses sensitivity analysis of Kriging models through *functional analysis of variance* (FANOVA) using Sobol's indexes. Section 5.9 discusses *risk analysis* (RA) or *uncertainty analysis* (UA). Section 5.10 discusses several remaining issues. Section 5.11 summarizes the major conclusions of this chapter, and suggests topics for future research. The chapter ends with Solutions of exercises, and a long list of references.

## 5.1 Introduction

In the preceding three chapters we focussed on linear regression metamodels (surrogates, emulators); namely, low-order polynomials. We fitted those models to the input/output (I/O) data of the—either local or global—experiment with the underlying simulation model; this simulation model may be either deterministic or random. We used these metamodels for the explanation of the simulation model’s behavior, and for the prediction of the simulation output for input combinations that were not yet simulated.

In the present chapter, we focus on *Kriging* metamodels. The name Kriging refers to Danie Krige (1919–2013), who was a South African mining engineer. In the 1960s Krige’s empirical work in *geostatistics*—see Krige (1951)—was formalized by the French mathematician George Matheron (1930–2000), using GPs—see Matheron (1963).

*Note:* A standard textbook on Kriging in geostatistics involving “spatial datan” is Cressie (1993); more recent books are Chilès and Delfiner (2012) and Stein (1999).

Kriging was introduced as a metamodel for *deterministic simulation models* or “computer models” in Sacks et al. (1989). Simulation models have  $k$ -dimensional input combinations where  $k$  is a given positive integer, whereas geostatistics considers only two or three dimensions.

*Note:* Popular textbooks on Kriging in computer models are Forrester et al. (2008) and Santner et al. (2003). A popular survey article is Simpson et al. (2001).

Kriging for *stochastic (random) simulation models* was briefly discussed in Mitchell and Morris (1992). Next, Van Beers and Kleijnen (2003) details Kriging in such simulation models, simply replacing the deterministic simulation output by the average computed from the replications that are usual in stochastic simulation. Although Kriging has not yet been frequently applied in stochastic simulation, we believe that the track record Kriging achieved in deterministic simulation holds promise for Kriging in stochastic simulation; also see Kleijnen (2014).

*Note:* Kleijnen (1990) introduced Kriging into the discrete-event simulation community. A popular review article is Kleijnen (2009). The classic discussion of Kriging in stochastic simulation is Ankenman et al. (2010). More references will follow in the next sections of this chapter.

Kriging is also studied in *machine learning*. A popular textbook is Rasmussen and Williams (2006). Web sites on GPs in machine learning are

<http://www.gaussianprocess.org/>  
<http://ml.dcs.shef.ac.uk/gpss/>  
<http://www.mlss.cc/>.

Besides the Anglo-Saxon literature, there is a vast *French* literature on Kriging, inspired by Matheron’s work; see

<http://www.gdr-mascotnum.fr/documents.html>.

Typically, Kriging models are fitted to data that are obtained for larger experimental areas than the areas used in low-order polynomial regression metamodels; i.e., Kriging models are *global* instead of local. Kriging models are used for prediction. The final goals are sensitivity analysis and risk analysis—as we shall see in this chapter—and optimization—as we shall see in the next chapter; these goals were also discussed in Sect. 1.2.

## 5.2 Ordinary Kriging (OK) in Deterministic Simulation

In this section we focus on OK, which is the simplest form of *universal Kriging* (UK), as we shall see in Sect. 5.4. OK is popular and successful in practical deterministic simulation, as many publications report.

*Note:* These publications include Chen et al. (2006), Martin and Simpson (2005), and Sacks et al. (1989). Recently, Mehdad and Kleijnen (2015a) also reports that in practice OK is likely to give better predictors than UK.

In Sect. 5.2.1 we present the basics of OK; in Sect. 5.2.2 we discuss the problems caused by the estimation of the (hyper)parameters of OK.

### 5.2.1 OK Basics

OK assumes the following metamodel:

$$y(\mathbf{x}) = \mu + M(\mathbf{x}) \quad \text{with } \mathbf{x} \in \mathbb{R}^k \quad (5.1)$$

where  $\mu$  is the constant mean  $E[y(\mathbf{x})]$  in the given  $k$ -dimensional experimental area, and  $M(\mathbf{x})$  is the additive noise that forms a Gaussian (multivariate normal) stationary process with zero mean. By definition, a *stationary process* has a constant mean, a constant variance, and covariances that depend only on the distance between the input combinations (or “points” in  $\mathbb{R}^k$ )  $\mathbf{x}$  and  $\mathbf{x}'$  (stationary processes were also defined in Definition 3.2).

Because different Kriging publications use different symbols for the same variable, we now discuss our symbols. We use  $\mathbf{x}$ —instead of  $\mathbf{d}$ —because the Kriging literature uses  $\mathbf{x}$  for the combination of inputs—even though the design of experiments (DOE) literature and the preceding chapters use  $\mathbf{d}$  for the combination of design variables (or factors);  $\mathbf{d}$  determines products such as  $x_j x_{j'}$  with  $j, j' = 1, \dots, k$ . The constant mean  $\mu$  in Eq. (5.1) is also denoted by  $\beta_0$ ; also see the section on UK (Sect. 5.4). Ankenman et al. (2010) calls  $M(\mathbf{x})$  the *extrinsic noise* to distinguish it from the *intrinsic noise* in stochastic simulation. OK assumes that the simulation output is deterministic (say)  $w$ . We distinguish between  $y$  (metamodel output) and  $w$  (simulation model output), whereas most Kriging publications do not distinguish between  $y$  and  $w$  (we also distinguished between  $y$  and  $w$  in the preceding chapters on linear regression; an example of our use of  $y$  and  $w$

is the predictor formula in Eq. (5.2) below). We try to stick to the symbols used in the preceding chapters; e.g., to denote the number of dimensions we use  $k$  (not  $d$ , which is used in some Kriging publications),  $\Sigma$  (not  $\Gamma$ ) to denote a covariance matrix, and  $\sigma$  (not  $\gamma$  or  $\Sigma(\mathbf{x}_0, \cdot)$ ) to denote a vector with covariances.

OK with its constant mean  $\mu$  does not imply a *flat* response surface. Actually, OK assumes that  $M(\mathbf{x})$  has positive covariances so  $\text{cov}[y(\mathbf{x}), y(\mathbf{x}')] > 0$ . Consequently, if it happens that  $y(\mathbf{x}) > \mu$ , then  $E[y(\mathbf{x}')] > \mu$  is “very likely” (i.e., the probability is greater than 0.50)—especially if  $\mathbf{x}$  and  $\mathbf{x}'$  lie close in  $\mathbb{R}^k$ . However, a linear regression metamodel with white noise implies  $\text{cov}[y(\mathbf{x}), y(\mathbf{x}')] = 0$ ; see the definition of white noise that we gave in Definition 2.3.

OK uses a *linear* predictor. So let  $\mathbf{w} = (w(\mathbf{x}_1), \dots, w(\mathbf{x}_n))'$  denote the  $n$  observed values of the simulation model at the  $n$  so-called *old* points (in machine learning these old points are called the “training set”). OK computes the predictor  $\hat{y}(\mathbf{x}_0)$  for a *new* point  $\mathbf{x}_0$  as a linear function of the  $n$  observed outputs at the old points:

$$\hat{y}(\mathbf{x}_0) = \sum_{i=1}^n \lambda_i w_i = \boldsymbol{\lambda}' \mathbf{w} \quad (5.2)$$

where  $w_i = f_{\text{sim}}(\mathbf{x}_i)$  and  $f_{\text{sim}}$  denotes the mathematical function that is defined by the simulation model itself (also see Eq. (2.6); the weight  $\lambda_i$  decreases with the *distance* between the new input combination  $\mathbf{x}_0$  and the old combination  $\mathbf{x}_i$ , as we shall see in Eq. (5.6); i.e., the weights  $\boldsymbol{\lambda}' = (\lambda_1, \dots, \lambda_n)$  are not constants (whereas  $\boldsymbol{\beta}$  in linear regression remains constant). Notice that  $\mathbf{x}_i = (x_{i;j})$  ( $i = 1, \dots, n; j = 1, \dots, k$ ) so  $\mathbf{X}' = (\mathbf{x}_1, \dots, \mathbf{x}_n)$  is a  $k \times n$  matrix.

To determine the *optimal* values for the weights  $\boldsymbol{\lambda}$  in Eq. (5.2), we need to specify a criterion for OK. In fact, OK (like other types of Kriging) uses the *best linear unbiased predictor (BLUP)*, which (by definition) minimizes the mean squared error (MSE) of the predictor:

$$\min \text{MSE}[\hat{y}(\mathbf{x}_0)] = \min \{E[\hat{y}(\mathbf{x}_0) - y(\mathbf{x}_0)]^2\}; \quad (5.3)$$

moreover, the predictor must be *unbiased* so

$$E[\hat{y}(\mathbf{x}_0)] = E[y(\mathbf{x}_0)]. \quad (5.4)$$

This bias constraint implies that if the new point coincides with one of the old points, then the predictor must be an *exact interpolator*; i.e.,  $\hat{y}(\mathbf{x}_i) = w(\mathbf{x}_i)$  with  $i = 1, \dots, n$  (also see Exercise 5.2 below).

*Note:* Linear regression uses as criterion the sum of squared residuals (SSR), which gives the least squares (LS) estimator. This estimator is not an exact interpolator, unless  $n = q$  where  $q$  denotes the number of regression parameters; see Sect. 2.2.1.

It can be proven that the solution of the constrained minimization problem defined by Eqs. (5.3) and (5.4) implies that  $\boldsymbol{\lambda}$  must satisfy the following

condition where  $\mathbf{1} = (1, \dots, 1)'$  is an  $n$ -dimensional vector with all elements equal to 1 (a more explicit notation would be  $\mathbf{1}_n$ ):

$$\sum_{i=1}^n \lambda_i = \mathbf{1}'\boldsymbol{\lambda} = 1. \tag{5.5}$$

Furthermore, it can be proven that the *optimal* weights are

$$\boldsymbol{\lambda}'_o = \left[ \boldsymbol{\sigma}(x_0) + \mathbf{1} \frac{1 - \mathbf{1}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\sigma}(x_0)}{\mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1}} \right]' \boldsymbol{\Sigma}^{-1} \tag{5.6}$$

where  $\boldsymbol{\Sigma} = (\text{cov}(y_i, y_{i'}))$ —with  $i, i' = 1, \dots, n$ —denotes the  $n \times n$  symmetric and positive definite matrix with the covariances between the metamodel’s “old” outputs (i.e., outputs of input combinations that have already been simulated), and  $\boldsymbol{\sigma}(x_0) = (\text{cov}(y_i, y_0))$  denotes the  $n$ -dimensional vector with the covariances between the metamodel’s  $n$  “old” outputs  $y_i$  and  $y_0$ , where  $y_0$  denotes the metamodel’s new output. Equation (5.1) implies  $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_M$ , but we suppress the subscript  $M$  until we really need it; see the section on stochastic simulation (Sect. 5.6). Throughout this book, we use Greek letters to denote unknown parameters (such as covariances), and bold upper case letters for matrixes and bold lower case letters for vectors.

Finally, it can be proven (see, e.g., Lin et al. 2004) that Eqs. (5.1), (5.2), and (5.6) together imply

$$\hat{y}(\mathbf{x}_0) = \mu + \boldsymbol{\sigma}(x_0)' \boldsymbol{\Sigma}^{-1} (\mathbf{w} - \mu \mathbf{1}). \tag{5.7}$$

We point out that this predictor varies with  $\boldsymbol{\sigma}(x_0)$ ; given are the Kriging parameters  $\mu$  and  $\boldsymbol{\Sigma}$ —where  $\boldsymbol{\Sigma}$  depends on the given old input data  $\mathbf{X}$ —and the old simulation output  $\mathbf{w}(\mathbf{X})$ . So we might replace  $\hat{y}(\mathbf{x}_0)$  by  $\hat{y}(\mathbf{x}_0|\mu, \boldsymbol{\Sigma}, \mathbf{X}, \mathbf{w})$  or  $\hat{y}(\mathbf{x}_0|\mu, \boldsymbol{\Sigma}, \mathbf{X})$ —because the output  $\mathbf{w}$  of a deterministic simulation model is completely determined by  $\mathbf{X}$ —but we do not use this unwieldy notation.

**Exercise 5.1** *Is the conditional expected value of the predictor in Eq. (5.7) smaller, equal, or larger than the unconditional mean  $\mu$  if that condition is as follows:  $w_1 > \mu, w_2 = \mu, \dots, w_n = \mu$ ?*

**Exercise 5.2** *Use Eq. (5.7) to derive the predictor if the new point is an old point, so  $\mathbf{x}_0 = \mathbf{x}_i$ .*

The Kriging predictor’s *gradient*  $\nabla(\hat{y}) = (\partial\hat{y}/\partial x_1, \dots, \partial\hat{y}/\partial x_k)$  results from Eq. (5.7); details are given in Lophaven et al. (2002, Eq. 2.18). Gradients will be used in Sect. 5.7 and in the next chapter (on simulation optimization). We should not confuse  $\nabla(\hat{y})$  (the gradient of the Kriging metamodel) and  $\nabla(w)$ , the gradient of the underlying simulation model. Sometimes we can indeed compute  $\nabla(w)$  in deterministic simulation (or estimate  $\nabla(w)$  in stochastic simulation); we may then use  $\nabla(w)$  (or  $\hat{\nabla}(w)$ )

to estimate better Kriging metamodels; see Qu and Fu (2014), Razavi et al. (2012), Ulaganathan et al. (2014), and Viana et al. (2014)’s references numbered 52, 53, and 54 (among the 221 references in that article).

If we let  $\tau^2$  denote the variance of  $y$ —where  $y$  was defined in Eq. (5.1)—then the MSE of the optimal predictor  $\hat{y}(\mathbf{x}_0)$ —where  $\hat{y}(\mathbf{x}_0)$  was defined in Eq. (5.7)—can be proven to be

$$\begin{aligned} \text{MSE} [\hat{y}(\mathbf{x}_0)] &= \tau^2 - \boldsymbol{\sigma}(\mathbf{x}_0)' \boldsymbol{\Sigma}^{-1} \boldsymbol{\sigma}(\mathbf{x}_0) \\ &\quad + \frac{[1 - \mathbf{1}' \boldsymbol{\Sigma}^{-1} \boldsymbol{\sigma}(\mathbf{x}_0)]^2}{\mathbf{1}' \boldsymbol{\Sigma}^{-1} \mathbf{1}}. \end{aligned} \tag{5.8}$$

Because the predictor  $\hat{y}(\mathbf{x}_0)$  is unbiased, this MSE equals the predictor variance—which is often called the *Kriging variance*. We denote this variance by  $\sigma_{OK}^2$ , the variance of the OK predictor. Analogously to the comment we made on Eq. (5.7), we now point out that this MSE depends on  $\boldsymbol{\sigma}(\mathbf{x}_0)$  only because the other factors in Eq. (5.8) are fixed by the old I/O data (we shall use this property when selecting a new point in sequential designs; see Sect. 5.5.2).

**Exercise 5.3** Use Eq. (5.8) to derive that  $\sigma_{OK}^2 = 0$  if  $\mathbf{x}_0$  equals one of the points already simulated; e.g.,  $\mathbf{x}_0 = \mathbf{x}_1$ .

Because  $\sigma_{OK}^2$  is zero if  $\mathbf{x}_0$  is an old point, the function  $\sigma_{OK}^2(\mathbf{x}_0)$  has many local minima if  $n > 1$ —and has many local maxima too; i.e.,  $\sigma_{OK}^2(\mathbf{x}_0)$  is nonconcave. Experimental results of many experiments suggest that  $\sigma_{OK}^2(\mathbf{x}_0)$  has local maxima at  $\mathbf{x}_0$  approximately halfway between old input combinations  $\mathbf{x}_i$ ; see part c of Fig. 5.2 below. We shall return to this characteristic in Sect. 6.3.1 on “efficient global optimization” (EGO).

Obviously, the optimal weight vector  $\boldsymbol{\lambda}_o$  in Eq. (5.6) depends on the covariances—or equivalently the correlations—between the outputs of the Kriging metamodel in Eq. (5.1). Kriging assumes that these correlations are determined by the “distance” between the input combinations. In geostatistics, Kriging often uses the *Euclidean distance* (say)  $h$  between the inputs  $\mathbf{x}_g$  and  $\mathbf{x}_{g'}$  with  $g, g' = 0, 1, \dots, n$  (so  $g$  and  $g'$  range between 0 and  $n$  and consequently  $\mathbf{x}_g$  and  $\mathbf{x}_{g'}$  cover both the new point and the  $n$  old points):

$$h_{g;g'} = \|\mathbf{x}_g - \mathbf{x}_{g'}\|_2 = \sqrt{\sum_{j=1}^k (x_{g;j} - x_{g';j})^2} \tag{5.9}$$

where  $\|\bullet\|_2$  denotes the  $L_2$  norm. This assumption means that

$$\rho[y(\mathbf{x}_g), y(\mathbf{x}_{g'})] = \frac{\sigma(h_{g;g'})}{\tau^2}, \tag{5.10}$$

which is called an *isotropic* correlation function; see Cressie (1993, pp. 61–62).

In simulation, however, we often assume that the Kriging metamodel has a correlation function—which implies a covariance function—that is

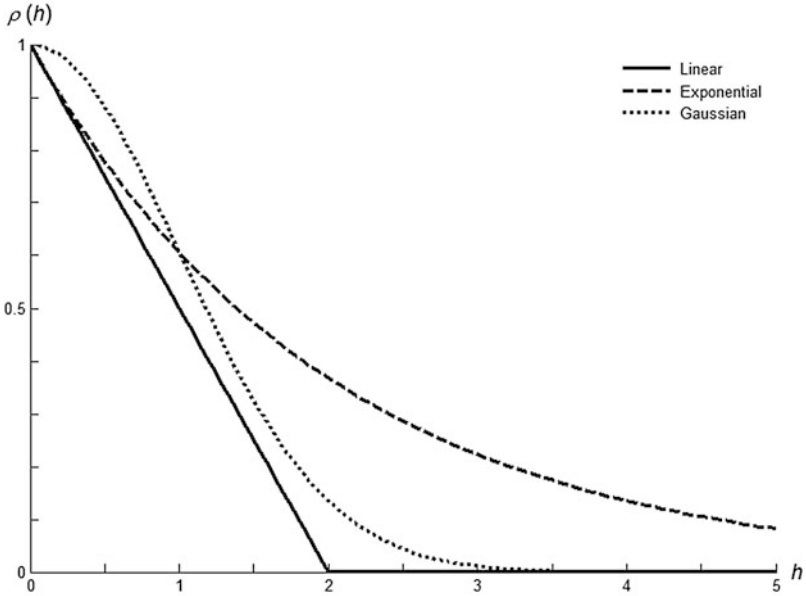


FIGURE 5.1. Three types of correlation functions  $\rho(h)$  with distance  $h$  and parameter  $\theta = 0.5$

not isotropic, but is *anisotropic*; e.g., in a *separable* anisotropic correlation function we replace Eq. (5.10) by the *product* of  $k$  one-dimensional correlation functions:

$$\rho[y(\mathbf{x}_g), y(\mathbf{x}_{g'})] = \prod_{j=1}^k \rho(x_{g;j}, x_{g';j}) \quad (g, g' = 0, 1, \dots, n). \quad (5.11)$$

Because Kriging assumes a stationary process, the correlations in Eq. (5.11) depend only on the distances in the  $k$  dimensions:

$$h_{g;g';j} = |x_{g;j} - x_{g';j}| \quad (j = 1, \dots, k); \quad (5.12)$$

also see Eq. (5.9). So,  $\rho(x_{g;j}, x_{g';j})$  in Eq. (5.11) reduces to  $\rho(h_{g;g';j})$ . Obviously, if the simulation model has a single input so  $k = 1$ , then these isotropic and the anisotropic correlation functions are identical. Furthermore, Kriging software standardizes (scales, codes, normalizes) the original simulation inputs and outputs, which affects the distances  $h$ ; also see Kleijnen and Mehdad (2013).

*Note:* Instead of correlation functions, geostatisticians use variograms, covariograms, and correlograms; see the literature on Kriging in geostatistics in Sect. 5.1.

There are several types of correlation functions that give valid (positive definite) covariance matrices for stationary processes; see the gen-

eral literature on GPs in Sect. 5.1, especially Rasmussen and Williams (2006, pp. 80–104). Geostatisticians often use so-called Matérn correlation functions, which are more complicated than the following three popular functions—displayed in Fig. 5.1 for a single input with parameter  $\theta = 0.5$ :

- Linear:  $\rho(h) = \max(1 - \theta h, 0)$
- Exponential:  $\rho(h) = \exp(-\theta h)$
- Gaussian:  $\rho(h) = \exp(-\theta h^2)$

*Note:* It is straightforward to prove that the Gaussian correlation function has its point of inflection at  $h = 1/\sqrt{2\theta}$ , so in Fig. 5.1 this point lies at  $h = 1$ . Furthermore, the linear correlation function gives correlations  $\rho(h)$  that are smaller than the exponential function gives, for  $\theta > 0$  and  $h > 0$ ; Fig. 5.1 demonstrates this behavior for  $\theta = 0.5$ . Finally, the linear correlation function gives  $\rho(h)$  smaller than the Gaussian function does, for (roughly)  $\theta > 0.45$  and  $h > 0$ . There are also correlation functions  $\rho(h)$  that do not monotonically decrease as the lag  $h$  increases; this is called a “hole effect” (see

[http://www.statistik.tuwien.ac.at/public/dutt/vorles/geost\\_03/node80.html](http://www.statistik.tuwien.ac.at/public/dutt/vorles/geost_03/node80.html)).

In simulation, a popular correlation function is

$$\rho(\mathbf{h}) = \prod_{j=1}^k \exp(-\theta_j h_j^{p_j}) = \exp\left(-\sum_{j=1}^k \theta_j h_j^{p_j}\right) \quad (5.13)$$

where  $\theta_j$  quantifies the importance of input  $j$ —the higher  $\theta_j$  is, the less effect input  $j$  has—and  $p_j$  quantifies the smoothness of the correlation function—e.g.,  $p_j = 2$  implies an infinitely differentiable function. Figure 5.1 has already illustrated an exponential function and a Gaussian function, which correspond with  $p = 1$  and  $p = 2$  in Eq. (5.13). (We shall discuss better measures of importance than  $\theta_j$ , in Sect. 5.8.)

**Exercise 5.4** What is the value of  $\rho(h)$  in Eq. (5.13) with  $p > 0$  when  $h = 0$  and  $h = \infty$ , respectively?

**Exercise 5.5** What is the value of  $\theta_j$  in Eq. (5.13) with  $p_j > 0$  when input  $j$  has no effect on the output?

*Note:* The choice of a specific type of correlation function may also affect the numerical properties of the Kriging model; see Harari and Steinberg (2014b).

Because  $\rho(\mathbf{h})$  in Eq. (5.13) decreases as the distance  $\mathbf{h}$  increases, the optimal weights  $\lambda_o$  in Eq. (5.6) are relatively high for old inputs close to the new input to be predicted.



*Note:* Some of the weights may be *negative*; see Wackernagel (2003, pp. 94–95). If negative weights give negative predictions and all the observed outputs  $w_i$  are nonnegative, then Deutsch (1996) sets negative weights and small positive weights to zero while restandardizing the sum of the remaining positive weights to one to make the predictor unbiased.

It is well known that Kriging results in bad extrapolation compared with interpolation; see Antognini and Zagoraiou (2010). Our intuitive explanation is that in *interpolation* the new point is surrounded by relatively many old points that are close to the new point; let us call them “close neighbors”. Consequently, the predictor combines many old outputs that are strongly positively correlated with the new output. In *extrapolation*, however, there are fewer close neighbors. Note that linear regression also gives minimal predictor variance at the center of the experimental area; see Eq. (6.7).

### 5.2.2 Estimating the OK Parameters

A major problem in OK is that the optimal Kriging weights  $\lambda_i$  ( $i = 1, \dots, n$ ) depend on the correlation function of the assumed metamodel—but *it is unknown which correlation function gives a valid metamodel*. In Kriging we usually select either an isotropic or an anisotropic type of correlation function and a specific type of decay such as linear, exponential, or Gaussian; see Fig. 5.1. Next we must estimate the parameter values; e.g.  $\theta_j$  ( $j = 1, \dots, k$ ) in Eq. (5.13). For this estimation we usually select the *maximum likelihood* (ML) criterion, which gives the ML estimators (MLEs)  $\hat{\theta}_j$ . ML requires the selection of a distribution for the metamodel output  $y(\mathbf{x})$  in Eq. (5.1). The standard distribution in Kriging is a multivariate normal, which explains the term GP. This gives the *log-likelihood function*

$$\begin{aligned} l(\mu, \tau^2, \boldsymbol{\theta}) &= -\ln[(2\pi)^{n/2}] \\ &\quad - \frac{1}{2} \ln[|\tau^2 \mathbf{R}(\boldsymbol{\theta})|] - \frac{1}{2} (\mathbf{w} - \mu \mathbf{1})' [\tau^2 \mathbf{R}(\boldsymbol{\theta})]^{-1} (\mathbf{w} - \mu \mathbf{1}) \\ &\quad \text{with } \boldsymbol{\theta} \geq \mathbf{0} \end{aligned} \tag{5.14}$$

where  $|\cdot|$  denotes the determinant and  $\mathbf{R}(\boldsymbol{\theta})$  denotes the correlation matrix of  $\mathbf{y}$ . Obviously, MLE requires that we minimize

$$\ln[|\tau^2 \mathbf{R}(\boldsymbol{\theta})|] + (\mathbf{w} - \mu \mathbf{1})' [\tau^2 \mathbf{R}(\boldsymbol{\theta})]^{-1} (\mathbf{w} - \mu \mathbf{1}). \tag{5.15}$$

We denote the resulting MLEs by a “hat”, so the MLEs are  $\hat{\mu}$ ,  $\hat{\tau}^2$ , and  $\hat{\boldsymbol{\theta}}$ . This minimization is a difficult mathematical problem. The classic solution in Kriging is to “divide and conquer”—called the “profile likelihood” or the “concentrated likelihood” in mathematical statistics—as we summarize in the following algorithm (in practice we use standard Kriging software that we shall list near the end of this section).

**Algorithm 5.1**

1. Initialize  $\hat{\boldsymbol{\theta}}$ , which defines  $\hat{\mathbf{R}}$ .
2. Compute the generalized least squares (GLS) estimator of the mean:

$$\hat{\boldsymbol{\mu}} = (\mathbf{1}^T \hat{\mathbf{R}}^{-1} \mathbf{1})^{-1} \mathbf{1}' \hat{\mathbf{R}}^{-1} \mathbf{y}. \quad (5.16)$$

3. Substitute  $\hat{\boldsymbol{\mu}}$  resulting from Step 2 and  $\hat{\mathbf{R}}$  resulting from Step 1 into the MLE variance estimator

$$\hat{\tau}^2 = \frac{(\mathbf{w} - \hat{\boldsymbol{\mu}} \mathbf{1})' \hat{\mathbf{R}}^{-1} (\mathbf{w} - \hat{\boldsymbol{\mu}} \mathbf{1})}{n}. \quad (5.17)$$

Comment:  $\hat{\tau}^2$  has the denominator  $n$ , whereas the denominator  $n - 1$  is used by the classic unbiased estimator assuming  $\mathbf{R} = \mathbf{I}$ .

4. Solve the remaining problem in Eq. (5.15):

$$\text{Min } \hat{\tau}^2 |\hat{\mathbf{R}}|^{-n}. \quad (5.18)$$

Comment: This equation can be found in Lophaven et al. (2002, equation 2.25). To solve this nonlinear minimization problem, Lophaven et al. (2002) applies the classic Hooke-Jeeves heuristic. Gano et al. (2006) points out that this minimization problem is difficult because of “the multimodal and long near-optimal ridge properties of the likelihood function”; i.e., this problem is not convex.

5. Use the  $\hat{\boldsymbol{\theta}}$  that solves Eq. (5.18) in Step 4 to update  $\hat{\mathbf{R}}$ , and substitute this updated  $\hat{\mathbf{R}}$  into Eqs. (5.16) and (5.17).
6. If the MLEs have not yet converged, then return to Step 2; else stop.

*Note:* Computational aspects are further discussed in Bachoc (2013), Butler et al. (2014), Gano et al. (2006), Jones et al. (1998), Li and Sudjianto (2005), Lophaven et al. (2002), Marrel et al. (2008), and Martin and Simpson (2005).

This difficult optimization problem implies that different MLEs may result from different software packages or from initializing the same package with different starting values; the software may even break down. The DACE software uses lower and upper limits for  $\theta_j$ , which are usually hard to specify. Different limits may give completely different  $\hat{\theta}_j$ , as the examples in Lin et al. (2004) demonstrate.

*Note:* Besides MLEs there are other estimators of  $\boldsymbol{\theta}$ ; e.g., *restricted MLEs* (RMLEs) and cross-validation estimators; see Bachoc (2013), Rasmussen and Williams (2006, pp. 116–124), Roustant et al. (2012), Santner et al.

(2003, pp. 66–68), and Sundararajan and Keerthi (2001). Furthermore, we may use the LS criterion. We have already shown estimators for covariances in Eq. (3.31), but in Kriging the number of observations for a covariance of a given distance  $h$  decreases as that distance increases. Given these estimates for various values of  $h$ , Kleijnen and Van Beers (2004) and Van Beers and Kleijnen (2003) use the LS criterion to fit a linear correlation function.

Let us denote the MLEs of the OK parameters by  $\hat{\boldsymbol{\psi}} = (\hat{\mu}, \hat{\tau}^2, \hat{\boldsymbol{\theta}}')'$  with  $\hat{\boldsymbol{\theta}}' = (\hat{\theta}_1, \dots, \hat{\theta}_k)$  in case of an anisotropic correlation function such as Eq. (5.13); obviously,  $\hat{\boldsymbol{\Sigma}} = \hat{\tau}^2 \hat{\mathbf{R}}(\hat{\boldsymbol{\theta}})$ . *Plugging* these MLEs into Eq. (5.7), we obtain the predictor

$$\hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}}) = \hat{\mu} + \hat{\boldsymbol{\sigma}}(\mathbf{x}_0)' \hat{\boldsymbol{\Sigma}}^{-1} (\mathbf{w} - \hat{\mu} \mathbf{1}). \quad (5.19)$$

This predictor depends on the new point  $\mathbf{x}_0$  only through  $\hat{\boldsymbol{\sigma}}(\mathbf{x}_0)$ , because  $\hat{\mu}$  and  $\hat{\boldsymbol{\Sigma}}^{-1} (\mathbf{w} - \hat{\mu} \mathbf{1})$  depend on the old I/O. The second term in this equation shows that this predictor is *nonlinear* (likewise, weighted least squares with estimated weights gives a nonlinear estimator in linear regression meta-models; see Sect. 3.4.4). However, most publications on Kriging compute the MSE of this predictor by simply plugging the MLEs of the Kriging parameters  $\tau^2$ ,  $\sigma(\mathbf{x}_0)$ , and  $\boldsymbol{\Sigma}$  into Eq. (5.8):

$$\begin{aligned} \text{MSE}[\hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}})] &= \hat{\tau}^2 - \hat{\boldsymbol{\sigma}}(\mathbf{x}_0)' \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\sigma}}(\mathbf{x}_0) \\ &\quad + \frac{(1 - \mathbf{1}' \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\sigma}}(\mathbf{x}_0))^2}{\mathbf{1}' \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{1}} \end{aligned} \quad (5.20)$$

We shall discuss a bootstrapped estimator of the true MSE of this nonlinear predictor, in the next section (Sect. 5.3).

*Note:* Martin and Simpson (2005) discusses alternative approaches—namely, validation and Akaike’s information criterion (AIC)—and finds that ignoring the randomness of the estimated Kriging parameters underestimates the true variance of the Kriging predictor. Validation for estimating the variance of the Kriging predictor is also discussed in Goel et al. (2006) and Viana and Haftka (2009). Furthermore, Thiart et al. (2014) confirms that the plug-in MSE defined in Eq. (5.20) underestimates the true MSE, and discusses alternative estimators of the true MSE. Jones et al. (1998) and Spöck and Pilz (2015) also imply that the plug-in estimator underestimates the true variance. Stein (1999) gives asymptotic results for Kriging with  $\hat{\boldsymbol{\psi}}$ .

We point out that Kriging gives a predictor plus a measure for the accuracy of this predictor; see Eq. (5.20). Some other metamodels—e.g., splines—do not quantify the accuracy of their predictor; see Cressie (1993, p. 182). Like Kriging, linear regression metamodels do quantify the accuracy; see Eq. (3.41).

The MSE in Eq. (5.20) is also used to compute a two-sided symmetric  $(1 - \alpha)$  *confidence interval* (CI) for the OK predictor at  $\mathbf{x}_0$ , where

$\hat{\sigma}_{\text{OK}}^2\{\hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}})\}$  equals  $\text{MSE}[\hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}})]$  and (say)  $a \pm b$  denotes the interval  $[a - b, a + b]$ :

$$P[w(\mathbf{x}_0) \in [\hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}}) \pm z_{\alpha/2} \sqrt{\hat{\sigma}_{\text{OK}}^2\{\hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}})\}}] = 1 - \alpha. \quad (5.21)$$

There is much *software* for Kriging. In our own experiments we have used DACE, which is a free-of-charge MATLAB toolbox well documented in Lophaven et al. (2002). Alternative free software is the R package DiceKriging—which is well documented in Roustant et al. (2012)—and the object-oriented software called the “ooDACE toolbox”—documented in Couckuyt et al. (2014). PerK programmed in C is documented in Santner et al. (2003, pp. 215–249). More free software is mentioned in Frazier (2011) and in the textbooks and websites mentioned in Sect. 5.1; also see the Gaussian processes for machine learning (GPML) toolbox, detailed in Rasmussen and Nickisch (2010). We also refer to the following four toolboxes (in alphabetical order):

MPERK on

[http://www.stat.osu.edu/~comp\\_exp/jour.club/MperkManual.pdf](http://www.stat.osu.edu/~comp_exp/jour.club/MperkManual.pdf)

STK on

<http://sourceforge.net/projects/kriging/>

<http://octave.sourceforge.net/stk/>,

SUMO on

<http://www.sumo.intec.ugent.be/>,

and Surrogates on

<https://sites.google.com/site/felipeacviana/surrogatetestoolbox>.

Finally, we refer to the commercial JMP/SAS site:

[https://www.jmp.com/en\\_us/software/feature-index.html#K](https://www.jmp.com/en_us/software/feature-index.html#K).

*Note:* For large data sets, the Kriging computations may become problematic; solutions are discussed in Gramacy and Haaland (2015) and Meng and Ng (2015).

As we have already stated in Sect. 1.2, we adhere to a frequentist view in this book. Nevertheless, we mention that there are many publications that interpret Kriging models in a *Bayesian* way. A recent article is Yuan and Ng (2015); older publications are referenced in Kleijnen (2008). Our major problem with the Bayesian approach to Kriging is that we find it hard to come up with prior distributions for the Kriging parameters  $\boldsymbol{\psi}$ , because we have little intuition about the correlation parameters  $\boldsymbol{\theta}$ ; e.g., what is the prior distribution of  $\boldsymbol{\theta}$ , in the Kriging metamodel of the  $M/M/1$  simulation model?

*Note:* Kriging seems related to so-called *moving least squares* (MLS), which originated in curve and surface fitting and fits a continuous function using a weighted least squares (WLS) criterion that gives more weight to old points close to the new point; see Lancaster and Salkauskas (1986) and also Forrester and Keane (2009) and Toropov et al. (2005).

The Kriging metamodel may also include *qualitative* inputs besides quantitative inputs. The challenge is to specify a valid covariance matrix; see Zhou et al. (2011).

## 5.3 Bootstrapping and Conditional Simulation for OK in Deterministic Simulation

In the preceding section we mentioned that in the present section we discuss a bootstrap approach to estimating the MSE of the nonlinear predictor with plugged-in estimated Kriging parameters  $\hat{\boldsymbol{\psi}}$  in Eq. (5.19). We have already discussed the general principles of bootstrapping in Sect. 3.3.5. Now we discuss parametric bootstrapping of the GP assumed in OK that was specified in Eq. (5.1). We also discuss a bootstrap variant called “conditional simulation”. Hasty readers may skip this section, because parametric bootstrapping and its variant are rather complicated and turn out to give CIs with coverages and lengths that are not superior compared with the CI specified in Eq. (5.21).

### 5.3.1 Bootstrapped OK (BOK)

For bootstrapping we use the notation that we introduced in Sect. 3.3.5. So we denote bootstrapped data by the superscript  $*$ ; e.g.,  $(\mathbf{X}, \mathbf{w}^*)$  denotes the original input and the bootstrapped output of the simulation model. We define bootstrapped estimators analogously to the original estimators, but we compute the bootstrapped estimators from the bootstrapped data instead of the original data; e.g., we compute  $\hat{\boldsymbol{\psi}}$  from  $(\mathbf{X}, \mathbf{w})$ , but  $\hat{\boldsymbol{\psi}}^*$  from  $(\mathbf{X}, \mathbf{w}^*)$ . We denote the bootstrap sample size by  $B$  and the  $b$ th bootstrap observation in this sample by the subscript  $b$  with  $b = 1, \dots, B$ .

Following Kleijnen and Mehdad (2013), we define the following  $(1 + n)$ -dimensional Gaussian or “normal” ( $N_{1+n}$ ) distribution:

$$\begin{pmatrix} y(\mathbf{x}_0) \\ y(\mathbf{x}) \end{pmatrix} \sim N_{1+n} \left[ \mu \mathbf{1}_{1+n}, \begin{pmatrix} \tau^2 & \boldsymbol{\sigma}(\mathbf{x}_0)' \\ \boldsymbol{\sigma}(\mathbf{x}_0) & \boldsymbol{\Sigma} \end{pmatrix} \right], \quad (5.22)$$

where all symbols were defined in the preceding section. Obviously, Eq. (5.22) implies  $y(\mathbf{x}) \sim N_n(\mu \mathbf{1}_n, \boldsymbol{\Sigma})$ .

Li and Zhou (2015) extends Den Hertog et al. (2006)’s bootstrap method for estimating the variance from univariate GP models to so-called “pairwise meta-modeling” of multivariate GP models assuming nonseparable covariance functions. We saw that if  $\mathbf{x}_0$  gets closer to an old point  $\mathbf{x}$ , then the predictor variance decreases and—because OK is an exact interpolator in deterministic simulation—this variance becomes exactly zero when  $\mathbf{x}_0 = \mathbf{x}$ . Furthermore,  $N_{1+n}$  in Eq. (5.22) implies that the distribution of

the new output—given the  $n$  old outputs—is the *conditional* normal distribution

$$N \left[ \hat{\boldsymbol{\mu}} + \hat{\boldsymbol{\sigma}}(\mathbf{x}_0)' \hat{\boldsymbol{\Sigma}}^{-1} [\mathbf{y}(\mathbf{x}) - \hat{\boldsymbol{\mu}}\mathbf{1}_n], \hat{\tau}^2 - \hat{\boldsymbol{\sigma}}(\mathbf{x}_0)' \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\sigma}}(\mathbf{x}_0) \right]. \quad (5.23)$$

We propose the following BOK pseudo-algorithm.

**Algorithm 5.2**

1. Use  $N_k \left( \hat{\boldsymbol{\mu}}\mathbf{1}_k, \hat{\boldsymbol{\Sigma}} \right)$   $B$  times to sample the  $n$  old outputs  $\mathbf{y}_b^*(\mathbf{X}, \hat{\boldsymbol{\psi}}) = (y_{1;b}^*(\mathbf{X}, \hat{\boldsymbol{\psi}}), \dots, y_{k;b}^*(\mathbf{X}, \hat{\boldsymbol{\psi}}))'$  where  $\hat{\boldsymbol{\psi}}$  is estimated from the old simulation I/O data  $(\mathbf{X}, \mathbf{w})$ . For each new point  $\mathbf{x}_0$  repeat steps 2 through 4  $B$  times.
2. Given the  $n$  old bootstrapped outputs  $\mathbf{y}_b^*(\mathbf{X}, \hat{\boldsymbol{\psi}})$  of step 1, sample the new output  $y_b^*(\mathbf{x}_0, \hat{\boldsymbol{\psi}})$  from the conditional normal distribution defined in Eq. (5.23).
3. Using the  $n$  old bootstrapped outputs  $\mathbf{y}_b^*(\mathbf{X}, \hat{\boldsymbol{\psi}})$  of step 1, compute the bootstrapped MLE  $\hat{\boldsymbol{\psi}}_b^*$ . Next calculate the bootstrapped predictor

$$\hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}}_b^*) = \hat{\boldsymbol{\mu}}_b^* + \hat{\boldsymbol{\sigma}}(\mathbf{x}_0, \hat{\boldsymbol{\theta}}_b^*)' \hat{\boldsymbol{\Sigma}}^{-1}(\hat{\boldsymbol{\theta}}_b^*) [\mathbf{y}_b^*(\mathbf{X}, \hat{\boldsymbol{\psi}}) - \hat{\boldsymbol{\mu}}_b^*\mathbf{1}_n]. \quad (5.24)$$

4. Given  $\hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}}_b^*)$  of step 3 and  $y_b^*(\mathbf{x}_0, \hat{\boldsymbol{\psi}})$  of step 2, compute the bootstrap estimator of the squared prediction error (SPE):

$$\text{SPE}_b^* = \text{SPE}[\hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}}_b^*)] = [\hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}}_b^*) - y_b^*(\mathbf{x}_0, \hat{\boldsymbol{\psi}})]^2.$$

5. Given the  $B$  bootstrap samples  $\text{SPE}_b^*$  ( $b = 1, \dots, B$ ) resulting from steps 1 through 4, compute the bootstrap estimator of  $\text{MSPE}[\hat{y}(\mathbf{x}_0)]$  (this MSPE was defined in Eq. (5.8):

$$\text{MSPE}^* = \frac{\sum_{b=1}^B \text{SPE}_b^*}{B}. \quad (5.25)$$

If we ignore the bias of the BOK predictor  $\hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}}^*)$ , then Eq. (5.25) gives  $\hat{\sigma}^2[\hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}}^*)]$  which is the bootstrap estimator of  $\sigma^2[\hat{y}(\mathbf{x}_0|\hat{\boldsymbol{\psi}})]$ . We abbreviate  $\hat{\sigma}^2[\hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}}^*)]$  to  $\hat{\sigma}_{\text{BOK}}^2$ . The standard error (SE) of  $\hat{\sigma}_{\text{BOK}}^2$  follows from Eq. (5.25):

$$\text{SE}(\hat{\sigma}_{\text{BOK}}^2) = \sqrt{\frac{\sum_{b=1}^B (\text{SPE}_b^* - \text{MSPE}^*)^2}{(B-1)B}}.$$

We apply  $t_{B-1}$  ( $t$ -statistic with  $B-1$  degrees of freedom) to obtain a two-sided symmetric  $(1-\alpha)$  CI for  $\sigma_{\text{BOK}}^2$ :

$$P[\sigma_{\text{OK}}^2 \in \hat{\sigma}_{\text{BOK}}^2 \pm t_{B-1, \alpha/2} \text{SE}(\hat{\sigma}_{\text{BOK}}^2)] = 1 - \alpha. \quad (5.26)$$

Obviously, if  $B \uparrow \infty$ , then  $t_{B-1;\alpha/2} \downarrow z_{\alpha/2}$  where  $z_{\alpha/2}$  denotes the  $\alpha/2$  quantile of the standard normal variable  $z \sim N(0, 1)$ ; typically  $B$  is so high (e.g., 100) that we can indeed replace  $t_{B-1;\alpha/2}$  by  $z_{\alpha/2}$ .

Figure 5.2 illustrates BOK for the following test function, taken from Forrester et al. (2008, p. 83):

$$w(x) = (6x - 2)^2 \sin(12x - 4) \text{ with } 0 \leq x \leq 1. \tag{5.27}$$

This function has one local minimum at  $x = 0.01$ , and one global minimum at  $x = 0.7572$  with output  $w = -6.02074$ ; we shall return to this function in the next chapter, in which we discuss simulation optimization. The plot shows that each of the  $B$  bootstrap samples has its own old output values  $\mathbf{y}_b^*$ . Part (a) displays only  $B = 5$  samples to avoid cluttering-up the plot. Part (b) shows less “wiggling” than part (a);  $\hat{y}(\mathbf{x}, \hat{\psi}_b^*)$ , which are the predictions at old points, coincide with  $\mathbf{y}_b^*(\mathbf{X}, \hat{\psi})$ , which are the values sampled in part (a). Part (c) uses  $B = 100$ .

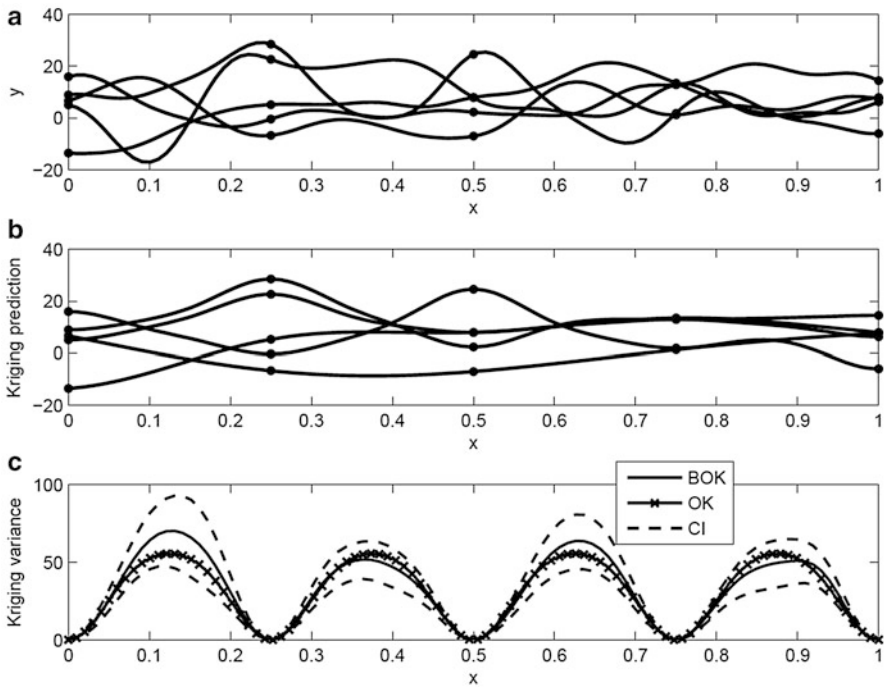


FIGURE 5.2. BOK for the test function in Forrester et al. (2008): (a) jointly sampled outputs at 5 equi-spaced old and 98 equi-spaced new points, for  $B = 5$ ; (b) Kriging predictions for 98 new points based on 5 old points sampled in (a); (c) estimated predictor variances and their 95 % CIs for  $B = 100$

To compute a two-sided symmetric  $(1 - \alpha)$  CI for the predictor at  $\mathbf{x}_0$ , we may use the OK point predictor  $\hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}})$  and  $\hat{\sigma}_{\text{BOK}}^2$  (equal to the MSE in Eq. (5.25)):

$$P\{w(\mathbf{x}_0) \in \hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}}) \pm z_{\alpha/2} \sqrt{\hat{\sigma}_{\text{BOK}}^2}\} = 1 - \alpha. \quad (5.28)$$

If  $\hat{\sigma}_{\text{OK}}^2 < \hat{\sigma}_{\text{BOK}}^2$ , then this CI is longer and gives a higher coverage than the CI in Eq. (5.21). Furthermore, we point out that Yin et al. (2010) also finds empirically that a Bayesian approach accounting for the randomness of the estimated Kriging parameters gives a wider CI—and hence higher coverage—than an approach that ignores this estimation.

### 5.3.2 Conditional Simulation of OK (CSOK)

We denote *conditional simulation* (CS) of OK by CSOK. This method ensures  $\hat{y}(\mathbf{x}, \hat{\boldsymbol{\psi}}_b^*) = w(\mathbf{x})$ ; i.e., in all the bootstrap samples the prediction at an old point equals the observed value. Part (a) of Fig. 5.3 may help understand Algorithm 5.3 for CSOK, which copies steps 1 through 3 of Algorithm 5.2 for BOK in the preceding subsection.

*Note:* Algorithm 5.3 is based on Kleijnen and Mehdad (2013), which follows Chilès and Delfiner (2012, pp. 478–650). CS may also be implemented through the R software package called “DiceKriging”; see Roustant et al. (2012).

#### Algorithm 5.3

1. Use  $N_n(\hat{\boldsymbol{\mu}}\mathbf{1}_n, \hat{\boldsymbol{\Sigma}})$   $B$  times to sample the  $n$  old outputs  $\mathbf{y}_b^*(\mathbf{X}, \hat{\boldsymbol{\psi}}) = (y_{1;b}^*(\mathbf{X}, \hat{\boldsymbol{\psi}}), \dots, y_{k;b}^*(\mathbf{X}, \hat{\boldsymbol{\psi}}))'$  where  $\hat{\boldsymbol{\psi}}$  is estimated from the old simulation I/O data  $(\mathbf{X}, \mathbf{w})$ . For each new point  $\mathbf{x}_0$ , repeat steps 2 through 4  $B$  times.
2. Given the  $n$  bootstrapped outputs  $\mathbf{y}_b^*(\mathbf{X}, \hat{\boldsymbol{\psi}})$  of step 1, sample the new output  $y_b^*(\mathbf{x}_0, \hat{\boldsymbol{\psi}})$  from the conditional normal distribution in Eq. (5.23).
3. Using the  $k$  old bootstrapped outputs  $\mathbf{y}_b^*(\mathbf{X}, \hat{\boldsymbol{\psi}})$  of step 1, compute the bootstrapped MLE  $\hat{\boldsymbol{\psi}}_b^*$ . Next calculate the bootstrapped predictor

$$\hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}}_b^*) = \hat{\boldsymbol{\mu}}_b^* + \hat{\boldsymbol{\sigma}}(\mathbf{x}_0)' \hat{\boldsymbol{\Sigma}}^{-1}(\hat{\boldsymbol{\theta}}_b^*) [y_b^*(\mathbf{X}, \hat{\boldsymbol{\psi}}) - \hat{\boldsymbol{\mu}}_b^* \mathbf{1}_n]. \quad (5.29)$$

4. Combining the OK estimator defined in Eq. (5.19) and the BOK estimator defined in Eq. (5.29), compute the CSOK predictor

$$\hat{y}_{\text{CSOK}}(\mathbf{x}_0, b) = \hat{\boldsymbol{\mu}} + \hat{\boldsymbol{\sigma}}(\mathbf{x}_0)' \hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{w} - \hat{\boldsymbol{\mu}}\mathbf{1}_n) + [y_b^*(\mathbf{x}_0, \hat{\boldsymbol{\psi}}) - \hat{y}(\mathbf{x}_0, \hat{\boldsymbol{\psi}}_b^*)]. \quad (5.30)$$



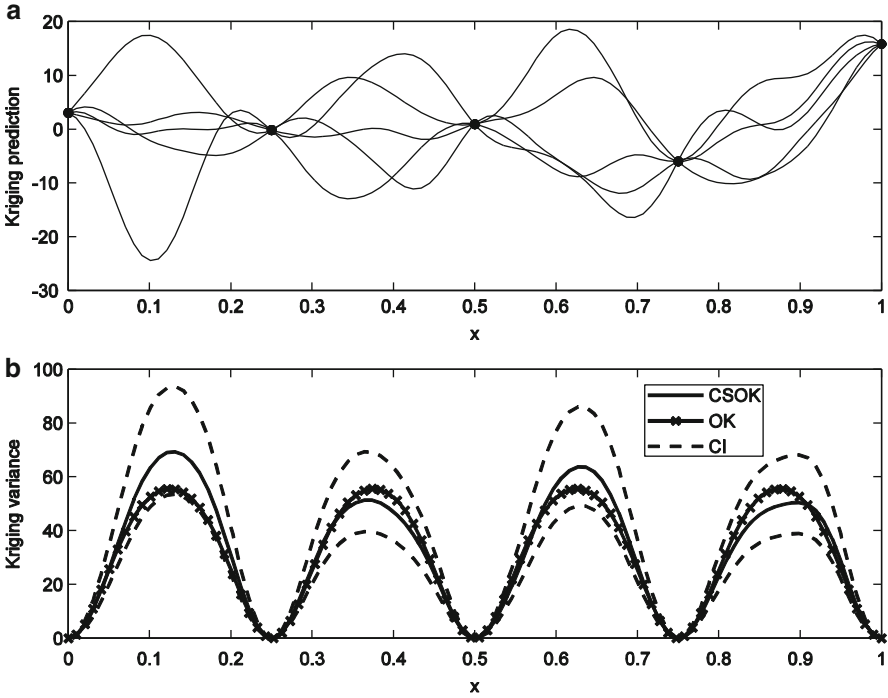


FIGURE 5.3. CSOK for the test function in Forrester et al. (2008): (a) predictions at 98 new points, for  $B = 5$ ; (b) estimated predictor variances and their 95 % CIs for  $B = 100$ , and OK's predictor variances

Given these  $B$  estimators  $\hat{y}_{\text{CSOK}}(\mathbf{x}_0, b)$  ( $b = 1, \dots, B$ ), compute the CSOK estimator of  $\text{MSPE}[\hat{y}(\mathbf{x}_0)]$ :

$$\hat{\sigma}^2[\hat{y}_{\text{CSOK}}(\mathbf{x}_0)] = \frac{\sum_{b=1}^B [\hat{y}_{\text{CSOK}}(\mathbf{x}_0, b) - \bar{\hat{y}}_{\text{CSOK}}(\mathbf{x}_0)]^2}{B - 1} \text{ with}$$

$$\bar{\hat{y}}_{\text{CSOK}}(\mathbf{x}_0) = \frac{\sum_{b=1}^B \hat{y}_{\text{CSOK}}(\mathbf{x}_0, b)}{B}. \tag{5.31}$$

We abbreviate  $\hat{\sigma}^2[\hat{Y}_{\text{CSOK}}(\mathbf{x}_0)]$  to  $\hat{\sigma}_{\text{CSOK}}^2$ . Mehdad and Kleijnen (2014) proves that  $\hat{\sigma}_{\text{CSOK}}^2 \leq \hat{\sigma}_{\text{BOK}}^2$ ; in practice, it is not known how much smaller  $\hat{\sigma}_{\text{CSOK}}^2$  is than  $\hat{\sigma}_{\text{BOK}}^2$ . We therefore apply a two-sided asymmetric  $(1 - \alpha)$  CI for  $\sigma_{\text{OK}}^2$  using  $\hat{\sigma}_{\text{CSOK}}^2$  and the chi-square statistic  $\chi_{B-1}^2$  (this CI replaces the CI for BOK in Eq. (5.28), which assumes  $B$  IID variables):

$$P \left( \frac{(B - 1)\hat{\sigma}_{\text{CSOK}}^2}{\chi_{B-1; 1-\alpha/2}^2} \leq \sigma_{\text{OK}}^2 \leq \frac{(B - 1)\hat{\sigma}_{\text{CSOK}}^2}{\chi_{B-1; \alpha/2}^2} \right) = 1 - \alpha. \tag{5.32}$$

Part (b) of Fig. 5.3 displays  $\hat{\sigma}_{\text{CSOK}}^2$  defined in Eq. (5.31) and its 95% CIs defined in Eq. (5.32) based on  $B = 100$  bootstrap samples; it also displays  $\hat{\sigma}_{\text{OK}}^2$  following from Eq. (5.20). Visual examination of this part suggests that  $\hat{\sigma}_{\text{CSOK}}^2$  tends to exceed  $\hat{\sigma}_{\text{OK}}^2$ .

Next, we display both  $\hat{\sigma}_{\text{CSOK}}^2$  and  $\hat{\sigma}_{\text{BOK}}^2$  and their CIs, for various values of  $B$ , in Fig. 5.4. This plot suggests that  $\hat{\sigma}_{\text{CSOK}}^2$  is not significantly smaller than  $\hat{\sigma}_{\text{BOK}}^2$ . These results seem reasonable, because both CSOK and BOK use  $\hat{\psi}$ , which is the *sufficient statistic* of the GP computed from the same  $(\mathbf{X}, \mathbf{w})$ . CSOK seems simpler than BOK, both computationally and conceptually. CSOK gives better predictions for new points close to old points; but then again, BOK is meant to improve the predictor variance—not the predictor itself.

We may use  $\hat{\sigma}_{\text{CSOK}}^2$  to compute a CI for the OK predictor, using the analogue of Eq. (5.28):

$$P \left\{ w(\mathbf{x}_0) \in \hat{y}(\mathbf{x}_0, \hat{\psi}) \pm z_{\alpha/2} \sqrt{\hat{\sigma}_{\text{CSOK}}^2} \right\} = 1 - \alpha. \quad (5.33)$$

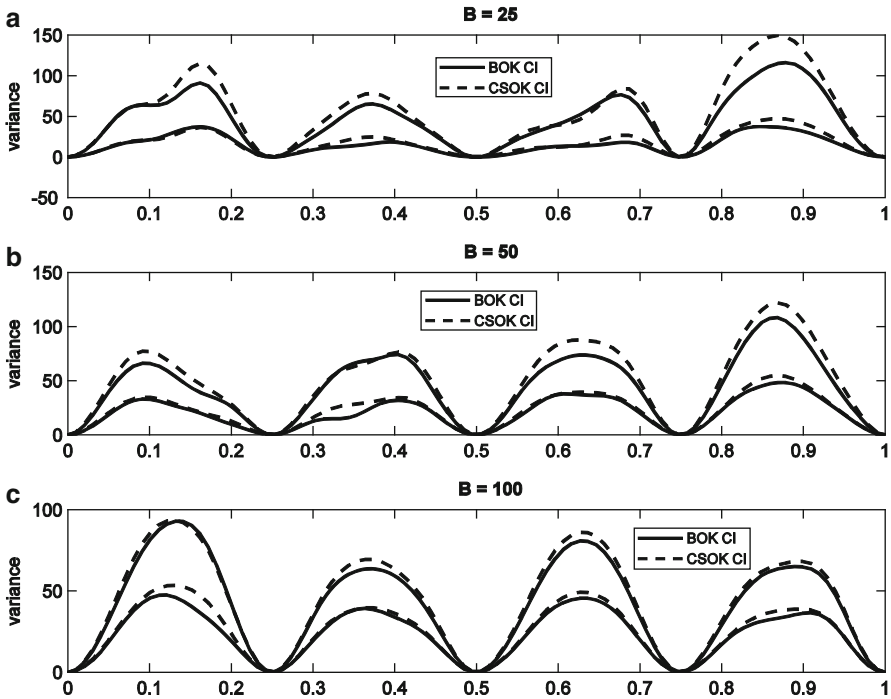


FIGURE 5.4. CIs for BOK versus CSOK for various  $B$  values, using the test function in Forrester et al. (2008)

Moreover, we can derive an alternative CI; namely, a *distribution-free* two-sided asymmetric CI based on the so-called *percentile method* (which we defined in Eq. (3.14)). We apply this method to  $\widehat{y}_{\text{CSOK}}(\mathbf{x}_0, b)$  ( $b = 1, \dots, B$ ), which are the  $B$  CSOK predictors defined in Eq. (5.30). Because the percentile method uses order statistics, we now denote  $\widehat{y}_{\text{CSOK}}(\mathbf{x}_0, b)$  by  $\widehat{y}_{\text{CSOK}; b}(\mathbf{x}_0)$ , apply the usual subscript  $(\cdot)$  (e.g.,  $(B\alpha/2)$ ) to denote order statistics (resulting from sorting the  $B$  values from low to high), and select  $B$  such that  $B\alpha/2$  and  $B(1 - \alpha/2)$  are integers:

$$P[\widehat{y}_{\text{CSOK}; (B\alpha/2)}(\mathbf{x}_0) \leq w(\mathbf{x}_0) \leq \widehat{y}_{\text{CSOK}; (B(1-\alpha/2))}(\mathbf{x}_0)] = 1 - \alpha. \quad (5.34)$$

An advantage of the percentile method is that this CI does not include negative values if the simulation output is not negative; also see Sect. 5.7 on bootstrapping OK to preserve known characteristics of the I/O functions (nonnegative outputs, monotonic I/O functions, etc.). We do not apply the percentile method to BOK, because BOK gives predictions at the  $n$  old points that do not equal the observed old simulation outputs  $w_i$ .

For OK, BOK, and CSOK Mehdad and Kleijnen (2015a) studies CIs with a nominal coverage of  $1 - \alpha$  and reports the estimated expected coverage  $1 - E(\widehat{\alpha})$  and the estimated expected length  $E(l)$  of the CIs, for a GP with two inputs so  $k = 2$  and an anisotropic Gaussian correlation function such as Eq. (5.13) with  $p = 2$ . In general, we prefer the CI with the shortest length, unless this CI gives too low coverage. The reported results show that OK with  $\widehat{\sigma}_{\text{OK}}$  gives shorter lengths than CSOK with  $\widehat{\sigma}_{\text{CSOK}}$ , and yet OK gives estimated coverages that are not significantly lower. The percentile method for CSOK gives longer lengths than OK, but its coverage is not significantly better than OK's coverage. Altogether the results do not suggest that BOK or CSOK is superior, so we recommend OK when predicting a new output; i.e., OK seems a robust method.

**Exercise 5.6** Consider the three alternative CIs that use OK, BOK, and CSOK, respectively. Do you think that the length of such a CI for a new point tends to decrease or increase as  $n$  (number of old points) increases?

## 5.4 Universal Kriging (UK) in Deterministic Simulation

UK replaces the constant  $\mu$  in Eq. (5.1) for OK by  $\mathbf{f}(\mathbf{x})'\boldsymbol{\beta}$  where  $\mathbf{f}(\mathbf{x})$  is a  $q \times 1$  vector of known functions of  $\mathbf{x}$  and  $\boldsymbol{\beta}$  is a  $q \times 1$  vector of unknown parameters (e.g., if  $k = 1$ , then UK may replace  $\mu$  by  $\beta_0 + \beta_1 x$ , which is called a “linear trend”):

$$y(\mathbf{x}) = \mathbf{f}(\mathbf{x})'\boldsymbol{\beta} + M(\mathbf{x}) \quad \text{with } \mathbf{x} \in \mathbb{R}^k. \quad (5.35)$$

The disadvantage of UK compared with OK is that UK requires the estimation of additional parameters. More precisely, besides  $\beta_0$  UK involves

$q - 1$  parameters, whereas OK involves only  $\beta_0 = \mu$ . We conjecture that the estimation of the extra  $q - 1$  parameters explains why UK has a higher MSE. In practice, most Kriging models do not use UK but OK

Note: This higher MSE for UK is also discussed in Ginsbourger et al. (2009) and Tajbakhsh et al. (2014). However, Chen et al. (2012) finds that UK in stochastic simulation with CRN may give better estimates of the gradient; also see Sect. 5.6. Furthermore, to eliminate the effects of estimating  $\beta$  in UK, Mehdad and Kleijnen (2015b) applies *intrinsic random functions* (IRFs) and derives the corresponding *intrinsic Kriging* (IK) and *stochastic intrinsic Kriging* (SIK). An IRF applies a linear transformation such that  $\mathbf{f}(\mathbf{x})' \beta$  in Eq. (5.35) vanishes. Of course, this transformation also changes the covariance matrix  $\Sigma_M$ , so the challenge becomes to determine a covariance matrix of IK that is valid (symmetric and “conditionally” positive definite). Experiments suggest that IK outperforms UK, and SIK outperforms SK. Furthermore, a refinement of UK is so-called *blind Kriging*, which does not assume that the functions  $\mathbf{f}(\mathbf{x})$  are known. Instead, blind Kriging chooses these functions from a set of candidate functions, assuming heredity (which we discussed below Eq. (4.11)) and using Bayesian techniques (which we avoid in this book; see Sect. 5.2). Blind Kriging is detailed in Joseph et al. (2008) and also in Couckuyt et al. (2012). Finally, Deng et al. (2012) compares UK with a new Bayesian method that also tries to eliminate unimportant inputs in the Kriging metamodel; the elimination of unimportant inputs we discussed in Chap. 4 on screening.

## 5.5 Designs for Deterministic Simulation

An  $n \times k$  design matrix  $\mathbf{X}$  specifies the  $n$  combinations of the  $k$  simulation inputs. The literature on designs for Kriging in deterministic simulation abounds, and proposes various design types. Most popular are Latin hypercube designs (LHDs). Alternative types are orthogonal array, uniform, maximum entropy, minimax, maximin, integrated mean squared prediction error (IMSPE), and “optimal” designs.

Note: Many references are given in Chen and Zhou (2014), Damblin et al. (2013), Janssen (2013), and Wang et al. (2014). Space-filling designs that account for statistical dependencies among the  $k$  inputs—which may be quantitative or qualitative—are given in Bowman and Woods (2013). A textbook is Lemieux (2009). More references are given in Harari and Steinberg (2014a), and Kleijnen (2008, p. 130). Relevant websites are

<http://lib.stat.cmu.edu>

and

<http://www.spacefillingdesigns.nl/>.

LHDs are specified through *Latin hypercube sampling* (LHS). Historically speaking, McKay et al. (1979) invented LHS not for Kriging but for risk

	Input 3's level				
Input 2's level	1	2	3	4	5
1	1	4	2	5	3
2	4	1	3	2	5
3	3	2	5	4	1
4	2	5	1	3	4
5	5	3	4	1	2

TABLE 5.1. A Latin square with three inputs, each with five levels

analysis using deterministic simulation models (“computer codes”); LHS was proposed as an alternative for crude Monte Carlo sampling (for Monte Carlo methods we refer to Chap. 1). LHS assumes that an adequate metamodel is more complicated than a low-order polynomial (these polynomial metamodels and their designs were discussed in the preceding three chapters). LHS does not assume a specific metamodel that approximates the I/O function defined by the underlying simulation model; actually, LHS focuses on the input space formed by the  $k$ -dimensional unit cube defined by the standardized simulation inputs. LHDs are one of the space-filling types of design (LHDs will be detailed in the next subsection, Sect. 5.5.1).

*Note:* It may be advantageous to use space-filling designs that allow sequential addition of points; examples of such designs are the *Sobol sequences* detailed on

[http://en.wikipedia.org/wiki/Sobol\\_sequence#References](http://en.wikipedia.org/wiki/Sobol_sequence#References).

We also refer to the *nested LHDs* in Qian et al. (2014) and the “sliced” LHDs in Ba et al. (2014), Li et al. (2015), and Yang et al. (2014); these sliced designs are useful for experiments with both qualitative and quantitative inputs. Furthermore, taking a subsample of a LHD—as we do in validation—destroys the LHD properties. Obviously, the most flexible method allowing addition and elimination of points is a simple random sample of  $n$  points in the  $k$ -dimensional input space.

In Sect. 5.5.1 we discuss LHS for designs with a given number of input combinations,  $n$ ; in Sect. 5.5.2 we discuss designs that determine  $n$  sequentially and are customized.

### 5.5.1 Latin Hypercube Sampling (LHS)

Technically, LHS is a type of stratified sampling based on the classic *Latin square* designs, which are square matrixes filled with different symbols such that each symbol occurs exactly once in each row and exactly once in each column. Table 5.1 is an example with  $k = 3$  inputs and five levels per input; input 1 is the input of real interest, whereas inputs 2 and 3 are nuisance inputs or block factors (also see our discussion on blocking in Sect. 2.10). This example requires only  $n = 5 \times 5 = 25$  combinations instead of  $5^3 = 125$  combinations. For further discussion of Latin (and Graeco-Latin) squares we refer to Chen et al. (2006).

	Input 3's level				
Input 2's level	1	2	3	4	5
1	1	2	3	4	5
2	5	1	2	3	4
3	4	5	1	2	3
4	3	4	5	1	2
5	2	3	4	5	1

TABLE 5.2. A systematic Latin square with three inputs, each with five levels

*Note:* Another Latin square—this time, constructed in a *systematic* way—is shown in Table 5.2. This design, however, may give a biased estimator of the effect of interest. For example, suppose that the input of interest (input 1) is wheat, and wheat comes in five varieties. Suppose further that this table determines the way wheat is planted on a piece of land; input 2 is the type of harvesting machine, and input 3 is the type of fertilizer. If the land shows a very fertile strip that runs from north-west to south-east (see the main diagonal of the matrix in this table), then the effect of wheat type 1 is overestimated. Therefore *randomization* should be applied to protect against unexpected effects. Randomization makes such bias unlikely—but not impossible. Therefore random selection may be corrected if its realization happens to be too systematic. For example, a LHD may be corrected to give a “nearly” orthogonal design; see Hernandez et al. (2012), Jeon et al. (2015), and Vieira et al. (2011).

The following algorithm details LHS for an experiment with  $n$  combinations of  $k$  inputs (also see Helton et al. (2006b)).

#### Algorithm 5.4

1. Divide the range of each input into  $n > 1$  mutually exclusive and exhaustive intervals of equal probability.  
Comment: If the distribution of input values is uniform on  $[a, b]$ , then each interval has length  $(b - a)/n$ . If the distribution is Gaussian, then intervals near the mode are shorter than in the tails.
2. Randomly select one value for  $x_1$  from each interval, without replacement, which gives  $n$  values  $x_{1,1}$  through  $x_{1,n}$ .
3. Pair these  $n$  values with the  $n$  values of  $x_2$ , randomly without replacement.
4. Combine these  $n$  pairs with the  $n$  values of  $x_3$ , randomly without replacement to form  $n$  triplets.
5. And so on, until a set of  $n$   $n$ -tuples is formed.

Table 5.3 and Fig. 5.5 give a LHD example with  $n = 5$  combinations of the two inputs  $x_1$  and  $x_2$ ; these combinations are denoted by as in fig. 5.5. The table shows that each input has five discrete levels, which are labelled 1 through 5. If the inputs are continuous, then the label (say) 1 may denote a value within interval 1; see Fig. 5.5.

LHS does not imply a strict mathematical relationship between  $n$  (number of combinations actually simulated) and  $k$  (number of simulation inputs), whereas DOE uses (for example)  $n = 2^k$  so  $n$  drastically increases with  $k$ . Nevertheless, if LHS keeps  $n$  “small” and  $k$  is “large”, then the resulting LHD covers the experimental domain  $\mathbb{R}^k$  so sparsely that the fitted Kriging model may be an inadequate metamodel of the underlying simulation model. Therefore a well-known rule-of-thumb for LHS in Kriging is  $n = 10k$ ; see Loepky et al. (2009).

	$x_1$				
$x_2$	1	2	3	4	5
1			•		
2		•			
3				•	
4					•
5	•				

TABLE 5.3. A LHS example with  $n = 5$  combinations of two inputs  $x_1$  and  $x_2$

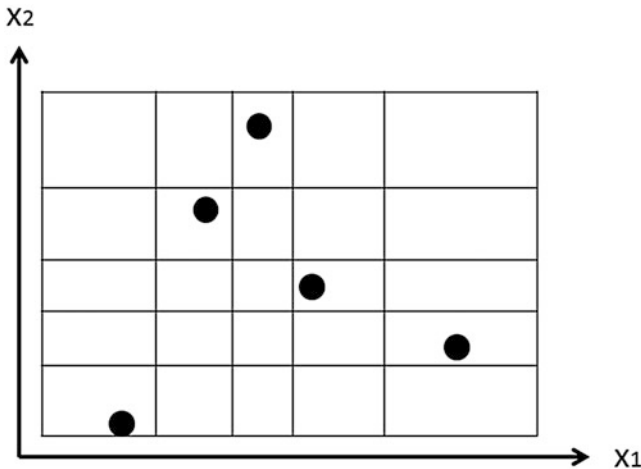


FIGURE 5.5. LHS example with  $n = 5$  combinations of two inputs  $x_1$  and  $x_2$

*Note:* Wang et al. (2014) recommends  $n = 20k$ . Furthermore, Hernandez et al. (2012) provides a table for LHDs with acceptable nonorthogonality for various  $(n, k)$  combinations with  $n \leq 1,025$  and  $k \leq 172$ .

Usually, LHS assumes that the  $k$  inputs are independently distributed—so their joint distribution becomes the product of their  $k$  individual marginal distributions—and the marginal distributions are uniform (symbol U) in the interval  $(0, 1)$  so  $x_j \sim U(0, 1)$ . An alternative assumption is a multivariate Gaussian distribution, which is completely characterized by its covariances and means. For nonnormal joint distributions, LHS may use Spearman’s correlation coefficient (discussed in Sect. 3.6.1); see Helton et al. (2006b). If LHS assumes a nonuniform marginal distribution for  $x_j$  (as we may assume in risk analysis, discussed in Sect. 5.9), then LHS defines  $n$ —mutually exclusive and exhaustive—subintervals  $[l_{j;g}, u_{j;g}]$  ( $g = 1, \dots, n$ ) for the standardized  $x_j$  such that each subinterval has the same probability; i.e.,  $P(l_{j;g} \leq x_j \leq u_{j;g}) = 1/n$ . This implies that near the mode of the  $x_j$  distribution, the subintervals are relatively short, compared with the subintervals in the tails of this distribution.

In LHS we may either *fix* the value of  $x_j$  to the middle of the subinterval  $g$  so  $x_j = (l_{j;g} + u_{j;g})/2$  or we may *sample* the value of  $x_j$  within that subinterval accounting for the distribution of its values. Fixing  $x_j$  is attractive when we wish to estimate the sensitivity of the output to the inputs (see Sect. 5.8, in which we shall discuss global sensitivity analysis through Sobol’s indexes). A random  $x_j$  is attractive when we wish to estimate the probability of the output exceeding a given threshold as a function of an uncertain input  $x_j$ , as we do in risk analysis (see Sect. 5.9).

LHDs are *noncollapsing*; i.e., if an input turns out to be unimportant, then each remaining individual input is still sampled with one observation per subinterval. DOE, however, then gives multiple observations for the same value of a remaining input—which is a waste in deterministic simulation (in stochastic simulation it improves the accuracy of the estimated intrinsic noise). Kriging with an anisotropic correlation function may benefit from the noncollapsing property of LHS, when estimating the correlation parameters  $\theta_j$ . Unfortunately, projections of a LHD point in  $n$  dimensions onto more than one dimension may give “bad” designs. Therefore standard LHS is further refined, leading to so-called maximin LHDs and nearly-orthogonal LHDs.

*Note:* For these LHDs we refer to Damblin et al. (2013), Dette and Pepelyshev (2010), Deutsch and Deutsch (2012), Georgiou and Stylianou (2011), Grosso et al. (2009), Janssen (2013), Jourdan and Franco (2010), Jones et al. (2015), Ranjan and Spencer (2014) and the older references in Kleijnen (2008, p. 130).

In a case study, Helton et al. (2005) finds that crude Monte Carlo and LHS give similar results if these two methods use the same “big” sample size. In general, however, LHS is meant to improve results in simulation applications; see Janssen (2013).



There is much software for LHS. For example, Crystal Ball, @Risk, and Risk Solver provide LHS, and are add-ins to Microsoft’s Excel spreadsheet software. LHS is also available in the MATLAB Statistics toolbox subroutine `lhs` and in the R package `DiceDesign`. We also mention Sandia’s DAKOTA software:

<http://dakota.sandia.gov/>.

### 5.5.2 Sequential Customized Designs

The preceding designs for Kriging have a given number of input combinations  $n$  and consider only the input domain  $\mathbf{x} \in \mathbb{R}^k$ ; i.e., these designs do not consider the output. Now we present designs that select  $n$  input combinations *sequentially* and consider the specific I/O function  $f_{\text{sim}}$  of the underlying simulation model so these designs are application-driven or *customized*. We notice that the importance of sequential sampling is also emphasized in Simpson et al. (2004), reporting on a panel discussion.

*Note:* Sequential designs for Kriging metamodels of deterministic simulation models are also studied in Busby et al. (2007), Crombecq et al. (2011), Koch et al. (2015), and Jin et al. (2002). Sequential LHDs ignoring the output (e.g., so-called “replicated LHDs”) are discussed in Janssen (2013). Our sequential customized designs are no longer LHDs (even though the first stage may be a LHD), as we shall see next.

The designs discussed so far in this section, are *fixed sample* or *one shot* designs. Such designs suit the needs of experiments with real systems; e.g., agricultural experiments may have to be finished within a single growing season. Simulation experiments, however, proceed sequentially—unless parallel computers are used, and even then not the whole experiment is finished in one shot. In general, sequential statistical procedures are known to be more “efficient” in the sense that they require fewer observations than fixed-sample procedures; see, e.g., Ghosh and Sen (1991). In sequential designs we learn about the behavior of the underlying system as we experiment with this system and collect data. (The preceding chapter on screening also showed that sequential designs may be attractive in simulation.) Unfortunately, extra computer time is needed in sequential designs for Kriging if we re-estimate the Kriging parameters when new I/O data become available. Fortunately, computations may not start from scratch; e.g., we may initialize the search for the MLEs in the sequentially augmented design from the MLEs in the preceding stage.

*Note:* Gano et al. (2006) updates the Kriging parameters only when the parameter estimates produce a poor prediction. Toal et al. (2008) examines five update strategies, and concludes that it is bad not to update the estimates after the initial design. Chevalier and Ginsbourger (2012) presents formulas for updating the Kriging parameters and predictors for designs that add I/O data either purely sequential (a single new point with its

output) or batch-sequential (batches of new points with their outputs). We shall also discuss this issue in Sect. 5.6 on SK.

Kleijnen and Van Beers (2004) proposes the following algorithm for specifying a customized sequential design for Kriging in deterministic simulation.

### Algorithm 5.5

1. Start with a pilot experiment using some space-filling design (e.g., a LHD) with only a few input combinations; use these combinations as the input for the simulation model, and obtain the corresponding simulation outputs.
2. Fit a Kriging model to the I/O simulation data resulting from Step 1.
3. Consider (but do not yet simulate) a set of candidate combinations that have not yet been simulated and that are selected through some space-filling design; find the “winner”, which is the candidate combination with the highest predictor variance.
4. Use the winner found in Step 3 as the input to the simulation model that is actually run, which gives the corresponding simulation output.
5. Re-fit (update) the Kriging model to the I/O data that is augmented with the I/O data resulting from Step 4.  
Comment: Step 5 refits the Kriging model, re-estimating the Kriging parameters  $\psi$ ; to save computer time, this step might not re-estimate  $\psi$ .
6. Return to Step 3 until either the Kriging metamodel satisfies a given goal or the computer budget is exhausted.

Furthermore, Kleijnen and Van Beers (2004) compares this sequential design with a sequential design that uses the predictor variance with plugged-in parameters specified in Eq. (5.20). The latter design selects as the next point the input combination that maximizes this variance. It turns out that the latter design selects as the next point the input farthest away from the old input combinations, so the final design spreads all its points (approximately) evenly across the experimental area—like space-filling designs do. However, the predictor variance may also be estimated through *cross-validation* (we have already discussed cross-validation of Kriging models below Eq. (5.20)); see Fig. 5.6, which we discuss next.

Figure 5.6 displays an example with a fourth-order polynomial I/O function  $f_{\text{sim}}$  with two local maxima and three local minima; two minima occur at the border of the experimental area. Leave-one-out cross-validation means successive deletion of one of the  $n$  old I/O observations (which are already simulated), which gives the data set  $(\mathbf{X}_{-i}, \mathbf{w}_{-i})$ . ( $i = 1, \dots, n$ ).

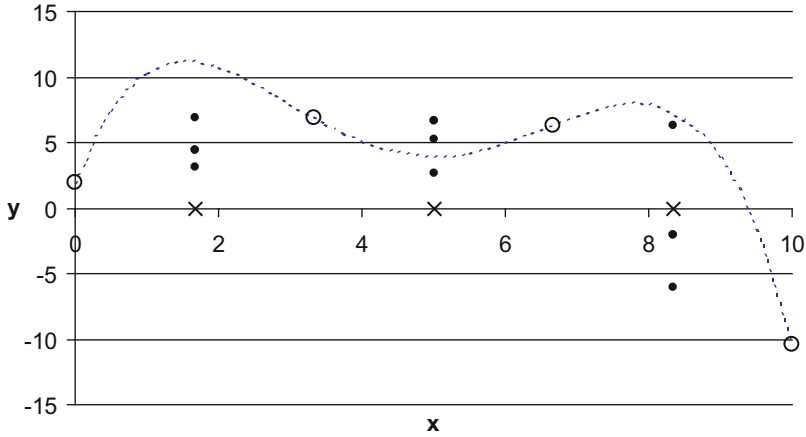


FIGURE 5.6. Cross-validation in fourth-order polynomial example with four pilot observations (see circles) and three candidate input values (see solid dots)

Next, we compute the Kriging predictor, after re-estimating the Kriging parameters. For each of three candidate points, the plot shows the three Kriging predictions computed from the original data set (no data deleted), and computed after deleting observation 2 and observation 3, respectively; the two extreme inputs ( $x = 0$  and  $x = 10$ ) are not deleted because Kriging does not extrapolate well. The point that is most difficult to predict turns out to be the candidate point  $x = 8.33$  (the highest candidate point in the plot). To quantify this prediction uncertainty, we may jackknife the predictor variances, as follows.

In Sect. 3.3.3, we have already discussed jackknifing in general (jackknifing is also applied to stochastic Kriging, in Chen and Kim (2013)). Now, we calculate the jackknife’s pseudo-value  $J$  for candidate point  $j$  as the weighted average of the original and the cross-validation predictors, letting  $c$  denote the number of candidate points and  $n$  the number of points already simulated and being deleted successively:

$$J_{j;i} = n\hat{y}_j - (n - 1)\hat{y}_{j;-i} \quad \text{with} \quad j = 1, \dots, c \text{ and } i = 1, \dots, n.$$

From these pseudo-values we compute the classic variance estimator (also see Eq. (3.12)):

$$s^2(J_j) = \frac{\sum_{i=1}^n (J_{j;i} - \bar{J}_j)^2}{n(n - 1)} \quad \text{with} \quad \bar{J}_j = \frac{\sum_{i=1}^n J_{j;i}}{n}.$$

Figure 5.7 shows the candidate points that are selected for actual simulation. The pilot sample consists of four equally spaced points; also see

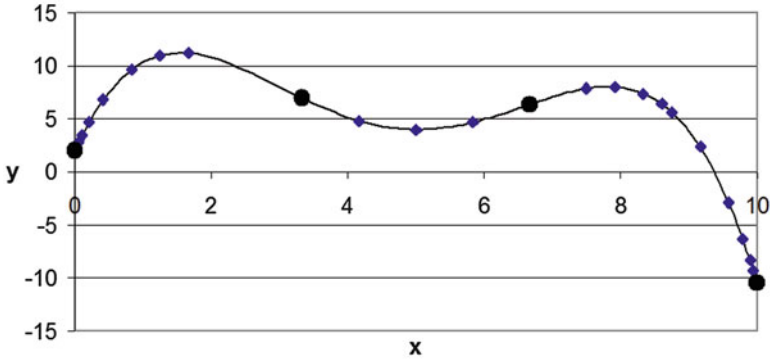


FIGURE 5.7. A fourth-degree polynomial example (see curve) of a sequential and customized design (see diamonds) and four initial points (see solid dots)

Fig. 5.6. The sequential design selects relative few points in subareas that generate an approximately linear I/O function; the design selects many points near the edges, where the function changes much. So the design favors points in subareas that have “more interesting” I/O behavior.

*Note:* Lin et al. (2002) criticizes cross-validation for the validation of Kriging metamodels, but in this section we apply cross-validation for the estimation of the prediction error when selecting the next design point in a customized design. Kleijnen and Van Beers (2004)’s method is also applied by Golzari et al. (2015).

## 5.6 Stochastic Kriging (SK) in Random Simulation

The interpolation property of Kriging is attractive in deterministic simulation, because the observed simulation output is unambiguous. In *random* simulation, however, the observed output is only one of the many possible values. Van Beers and Kleijnen (2003) replaces  $w_i$  (the simulation output at point  $i$  with  $i = 1, \dots, n$ ) by  $\bar{w}_i = \sum_{r=1}^{m_i} w_{i,r} / m_i$  (the average simulated output computed from  $m_i$  replications). These averages, however, are still random, so the interpolation property loses its intuitive appeal. Nevertheless, Kriging may be attractive in random simulation because Kriging may decrease the predictor MSE at input combinations close together.

*Note:* Geostatisticians often use a model for (random) measurement errors that assumes a so-called *nugget effect* which is white noise; see Cressie (1993, pp. 59, 113, 128) and also Clark (2010). The Kriging predictor is then no longer an exact interpolator. Geostatisticians also study noise with

heterogeneous variances; see Opsomer et al. (1999). In machine learning this problem is studied under the name *heteroscedastic GP regression*; see Kleijnen (1983) and our references in Sect. 5.1. Roustant et al. (2012) distinguishes between the nugget effect and homogeneous noise, such that the former gives a Kriging metamodel that remains an exact interpolator, whereas the latter does not. Historically speaking, Danie Krige worked in mining engineering and was confronted with the “nugget effect”; i.e., gold diggers may either miss the gold nugget “by a hair” or hit it “right on the head”. Measurement error is a fundamentally different issue; i.e., when we measure (e.g.) the temperature on a fixed location, then we always get different values when we repeat the measurement at points of time “only microseconds apart”, the “same” locations separated by nanomillimeters only, using different measurement tools or different people, etc.

In deterministic simulation, we may study *numerical* problems arising in Kriging. To solve such numerical noise, Lophaven et al. (2002, Eq. 3.16) and Toal et al. (2008) add a term to the covariance matrix  $\Sigma_M$  (also see Eq. (5.36) below); this term resembles the nugget effect, but with a “variance” that depends on the computer’s accuracy.

*Note:* Gramacy and Lee (2012) also discusses the use of the nugget effect to solve numerical problems, but emphasizes that the nugget effect may also give better statistical performance such as better CIs. Numerical problems are also discussed in Goldberg et al. (1998), Harari and Steinberg (2014b), and Sun et al. (2014).

In Sect. 5.6.1 we discuss a metamodel for stochastic Kriging (SK) and its analysis; in Sect. 5.6.2 we discuss designs for SK.

### 5.6.1 A Metamodel for SK

In the analysis of random (stochastic) simulation models—which use pseudorandom numbers (PRNs)—we may apply SK, adding the *intrinsic noise* term  $\varepsilon_r(\mathbf{x})$  for replication  $r$  at input combination  $\mathbf{x}$  to the GP metamodel in Eq. (5.1) for OK with the *extrinsic noise*  $M(\mathbf{x})$ :

$$y_r(\mathbf{x}) = \mu + M(\mathbf{x}) + \varepsilon_r(\mathbf{x}) \quad \text{with} \quad \mathbf{x} \in \mathbb{R}^k \quad \text{and} \quad r = 1, \dots, m_i \quad (5.36)$$

where  $\varepsilon_r(\mathbf{x})$  has a Gaussian distribution with zero mean and variance  $\text{Var}[\varepsilon_r(\mathbf{x})]$  and is independent of the extrinsic noise  $M(\mathbf{x})$ . If the simulation does not use CRN, then  $\Sigma_\varepsilon$ —the covariance matrix for the intrinsic noise—is diagonal with the elements  $\text{Var}[\varepsilon(\mathbf{x})]$  on the main diagonal. If the simulation does use CRN, then  $\Sigma_\varepsilon$  is not diagonal; obviously,  $\Sigma_\varepsilon$  should still be symmetric and positive definite. (Some authors—e.g. Challenor (2013)—use the term “aleatory” noise for the intrinsic noise, and the term “epistemic noise” for the extrinsic noise in Kriging; we use these alternative terms in Chaps. 1 and 6.)

Averaging the  $m_i$  replications gives the average metamodel output  $\bar{y}(\mathbf{x}_i)$  and average intrinsic noise  $\bar{\varepsilon}(\mathbf{x}_i)$ , so Eq. (5.36) is replaced by

$$\bar{y}(\mathbf{x}_i) = \mu + M(\mathbf{x}_i) + \bar{\varepsilon}(\mathbf{x}_i) \quad \text{with} \quad \mathbf{x} \in \mathbb{R}^k \quad \text{and} \quad i = 1, \dots, n. \quad (5.37)$$

Obviously, if we obtain  $m_i$  replicated simulation outputs for input combination  $i$  and we do not use CRN, then  $\Sigma_{\bar{\varepsilon}}$  is a diagonal matrix with main-diagonal elements  $\text{Var}[\varepsilon(\mathbf{x}_i)]/m_i$ . If we do use CRN and  $m_i$  is a constant  $m$ , then  $\Sigma_{\bar{\varepsilon}} = \Sigma_{\varepsilon}/m$  where  $\Sigma_{\varepsilon}$  is a symmetric positive definite matrix.

SK may use the classic estimators of  $\text{Var}[\varepsilon(\mathbf{x}_i)]$  using  $m_i > 1$  replications, which we have already discussed in Eq. (2.27):

$$s^2(w_i) = \frac{\sum_{r=1}^{m_i} (w_{i;r} - \bar{w}_i)^2}{m_i - 1} \quad (i = 1, \dots, n)$$

Instead of these point estimates of the intrinsic variances, SK may use another Kriging metamodel for the variances  $\text{Var}[\varepsilon(\mathbf{x}_i)]$ —besides the Kriging metamodel for the mean  $E[y_r(\mathbf{x}_i)]$ —to predict the intrinsic variances. We expect this alternative to be less volatile than  $s^2(w_i)$ ; after all,  $s^2(w_i)$  is a *chi-square* variable (with  $m_i - 1$  degrees of freedom) and has a large variance. Consequently,  $s^2(w_i)$  is not normally distributed so the GP assumed for  $s^2(w_i)$  is only a rough approximation. Because  $s^2(w_i) \geq 0$ , Goldberg et al. (1998) uses  $\log[s^2(w_i)]$  in the Kriging metamodel. Moreover, we saw in Sect. 3.3.3 that a logarithmic transformation may make the variable normally distributed. We also refer to Kamiński (2015) and Ng and Yin (2012).

*Note:* Goldberg et al. (1998) assumes a known mean  $E[y(\mathbf{x})]$ , and a Bayesian approach using *Markov chain Monte Carlo* (MCMC) methods. Kleijnen (1983) also uses a Bayesian approach but no MCMC. Both Goldberg et al. (1998) and Kleijnen (1983) do not consider replications. Replications are standard in stochastic simulation; nevertheless, stochastic simulation without replication is studied in (Marrel et al. 2012). Risk and Ludkovski (2015) applies SK with estimated constant mean  $\hat{\mu}$  (like OK does) and mean function  $f(\mathbf{x}; \hat{\beta})$  (like UK does), and reports several case studies that give smaller MSEs for  $f(\mathbf{x}; \hat{\beta})$  than for  $\hat{\mu}$ .

SK uses the OK predictor and its MSE replacing  $\Sigma_M$  by  $\Sigma_M + \Sigma_{\bar{\varepsilon}}$  and  $\mathbf{w}$  by  $\bar{\mathbf{w}}$ , so the SK predictor is

$$\hat{y}(\mathbf{x}_0, \hat{\psi}) = \hat{\mu} + \hat{\sigma}(\mathbf{x}_0)'(\hat{\Sigma}_M + \hat{\Sigma}_{\bar{\varepsilon}})^{-1}(\bar{\mathbf{w}} - \hat{\mu}\mathbf{1}) \quad (5.38)$$

and its MSE is

$$\begin{aligned} \text{MSE}[\hat{y}(\mathbf{x}_0, \hat{\psi})] &= \hat{\tau}^2 - \hat{\sigma}(\mathbf{x}_0)'(\hat{\Sigma}_M + \hat{\Sigma}_{\bar{\varepsilon}})^{-1}\hat{\sigma}(\mathbf{x}_0) \\ &\quad + \frac{[1 - \mathbf{1}'(\hat{\Sigma}_M + \hat{\Sigma}_{\bar{\varepsilon}})^{-1}\hat{\sigma}(\mathbf{x}_0)]^2}{\mathbf{1}'(\hat{\Sigma}_M + \hat{\Sigma}_{\bar{\varepsilon}})^{-1}\mathbf{1}}; \end{aligned} \quad (5.39)$$

also see Ankenman et al. (2010, Eq. 25).

The output of a stochastic simulation may be a *quantile* instead of an average (Eq. (5.37) does use averages). For example, a quantile may be relevant in chance-constrained optimization; also see Eq. (6.35) and Sect. 6.4 on robust optimization. Chen and Kim (2013) adapts SK for the latter type of simulation output; also see Bekki et al. (2014), Quadrianto et al. (2009), and Tan (2015).

*Note:* Salemi et al. (2014) assumes that the simulation inputs are integer variables, and uses a Gaussian Markov random field. Chen et al. (2013) allows some inputs to be qualitative, extending the approach for deterministic simulation in Zhou et al. (2011). Estimation of the whole density function of the output is discussed in Moutoussamy et al. (2014).

There is not much *software* for SK. The Matlab software available on the following web site is distributed “without warranties of any kind”:

<http://www.stochastickriging.net/>.

The R package “DiceKriging” accounts for heterogeneous intrinsic noise; see Roustant et al. (2012). The R package “mlegp” is available on

<http://cran.r-project.org/web/packages/mlegp/mlegp.pdf>.

Software in C called PErK may also account for a nugget effect; see Santner et al. (2003, pp. 215–249).

In Sect. 5.3 we have already seen that ignoring the randomness of the estimated Kriging parameters  $\hat{\psi}$  tends to underestimate the true variance of the Kriging predictor. To solve this problem in case of deterministic simulation, we may use *parametric bootstrapping* or its refinement called *conditional simulation*. (Moreover, the three variants—plugging-in  $\hat{\psi}$ , bootstrapping, or conditional simulation—may give predictor variances that reach their maxima for different new input combinations; these maxima are crucial in simulation optimization through “efficient global optimization”, as we shall see in Sect. 6.3.1). In stochastic simulation, we obtain several replications for each old input combination—see Eq. (5.37)—so a simple method for estimating the true predictor variance uses *distribution-free bootstrapping*. We have already discussed the general principles of bootstrapping in Sect. 3.3.5. Van Beers and Kleijnen (2008) applies distribution-free bootstrapping assuming no CRN, as we shall see in the next subsection (Sect. 5.6.2). Furthermore, Yin et al. (2009) also studies the effects that the estimation of the Kriging parameters has on the predictor variance.

*Note:* Mehdad and Kleijnen (2015b) applies *stochastic intrinsic Kriging* (SIK), which is more complicated than SK. Experiments with stochastic simulations suggest that SIK outperforms SK.

To estimate the true variance of the SK predictor, Kleijnen and Mehdad (2015a) applies the Monte Carlo method, distribution-free bootstrapping, and parametric bootstrapping, respectively—using an M/M/1 simulation model for illustration.

### 5.6.2 Designs for SK

Usually SK employs the same designs as OK and UK do for deterministic simulation. So, SK often uses a one-shot design such as a *LHD*; also see Jones et al. (2015) and MacCalman et al. (2013).

However, besides the  $n \times k$  matrix with the  $n$  design points  $\mathbf{x}_i \in \mathbb{R}^k$  ( $i = 1, \dots, n$ ) we need to select the *number of replications*  $m_i$ . In Sect. 3.4.5 we have already discussed the analogous problem for linear regression meta-

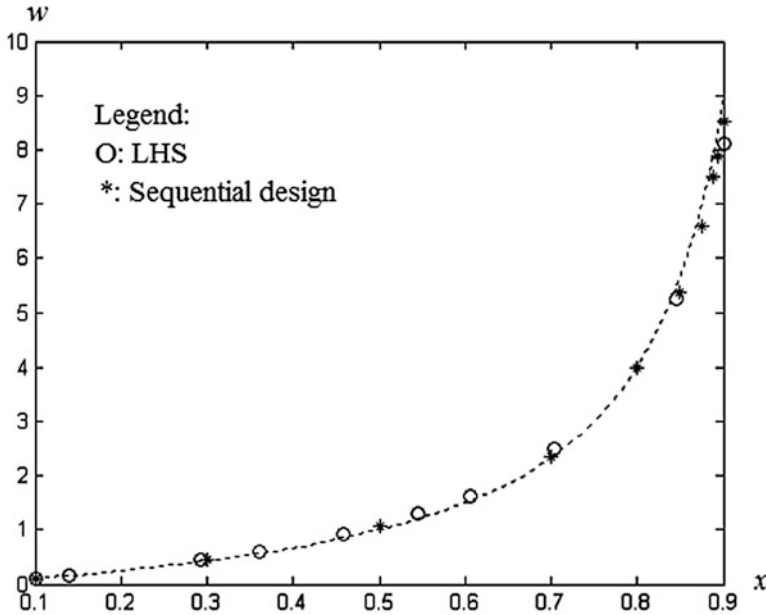


FIGURE 5.8. A LHD and a sequentialized design for the M/M/1 simulation with traffic rate  $0 < x < 1$  and  $n = 10$  points

models; a simple rule-of-thumb is to select  $m_i$  such that with  $1 - \alpha$  probability the average output is within  $\gamma\%$  of the true mean; see Eq. (3.30).

*Note:* For SK with heterogeneous intrinsic variances but without CRN (so  $\Sigma_\varepsilon$  is diagonal), Boukouvalas et al. (2014) examines *optimal designs* (which we also discussed for linear regression metamodels in Sect. 2.10.1). That article shows that designs that optimize the determinant of the so-called *Fisher information matrix* (FIM) outperform space-filling designs (such as LHDs), with or without replications. This FIM criterion minimizes the estimation errors of the GP covariance parameters (not the parameters  $\beta$  of the regression function  $\mathbf{f}(\mathbf{x})'\beta$ ). That article recommends designs with at least two replications at each point; the optimal number of replications is determined through an optimization search algorithm. Furthermore, that article proposes the logarithmic transformation of the intrinsic variance when estimating a metamodel for this variance (we also discussed such a transformation in Sect. 3.4.3). Optimal designs for SK with homogeneous intrinsic variances (or a nugget effect) are also examined in Harari and Steinberg (2014a), and Spöck and Pilz (2015).

There are more complicated approaches. In sequential designs, we may use Algorithm 5.5 for deterministic simulation, but we change Step 3—which finds the candidate point with the highest predictor variance—such that we find this point through distribution-free bootstrapping based on



replication, as we shall explain below. Figure 5.8 is reproduced from Van Beers and Kleijnen (2008); it displays a fixed LHS design with  $n = 10$  values for the traffic rate  $x$  in an M/M/1 simulation with experimental area  $0.1 \leq x \leq 0.9$ , and a sequentialized design that is stopped after simulating the same number of observations (namely,  $n = 10$ ). The plot shows that the sequentialized design selects more input values in the part of the input range that gives a drastically increasing (highly nonlinear) I/O function; namely  $0.8 < x \leq 0.9$ . It turns out that this design gives better Kriging predictions than the fixed LHS design does—especially for small designs, which are used in expensive simulations.

The M/M/1 simulation in Fig. 5.8 selects a run-length that gives a 95% CI for the mean simulation output with a relative error of no more than 15%. The sample size for the distribution-free bootstrap method is selected to be  $B = 50$ .

To estimate the predictor variance, Van Beers and Kleijnen (2008) uses *distribution-free bootstrapping* and treats the observed average bootstrapped outputs  $\bar{w}_i^*$  ( $i = 1, \dots, n$ ) as if they were the true mean outputs; i.e., the Kriging metamodel is an exact interpolator of  $\bar{w}_i^*$  (obviously, this approach ignores the split into intrinsic and extrinsic noise that SK assumes).

*Note:* Besides the M/M/1 simulation, Van Beers and Kleijnen (2008) also investigates an  $(s, S)$  inventory simulation. Again, the sequentialized design for this  $(s, S)$  inventory simulation gives better predictions than a fixed-size (one-shot) LHS design; the sequentialized design concentrates its points in the steeper part of the response surface. Chen and Li (2014) also determines the number of replications through a relative precision requirement, but assumes linear interpolation instead of Kriging; that article also provides a comparison with the approach in Van Beers and Kleijnen (2008).

*Note:* Ankenman et al. (2010) does use the SK model in Eq. (5.36), and tries to find the design that allocates a fixed computer budget such that “new points” (input combinations not yet simulated) may be selected or additional replications for old points may be obtained. Chen and Zhou (2014) uses this approach, applying a variety of design criteria based on the MSE. Plumlee and Tuo (2014) also examines the number of replications in SK. Hernandez and Grover (2010) discusses sequential designs for Kriging metamodels of random simulation models; namely, models of so-called nanoparticles. Furthermore, Forrester (2013) recommends re-estimation of the Kriging hyperparameters  $\psi$ , as the sequential design provides new I/O data. Kamiński (2015) gives various methods that avoid re-estimation of  $\psi$  in case of SK and sequential designs. Mehdad and Kleijnen (2015b) discusses sequential designs for *stochastic intrinsic Kriging* (SIK). More research on this issue is needed.

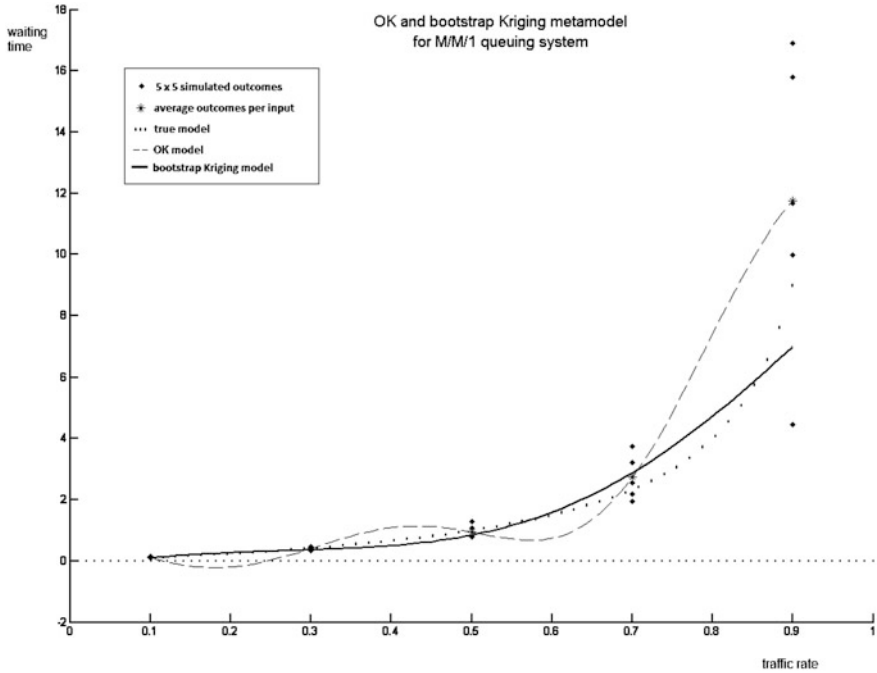


FIGURE 5.9. OK versus monotonic bootstrapped Kriging with acceptance/rejection, and the true I/O function for M/M/1 with  $n = 5$  old input values and  $m = 5$  replications

### 5.7 Monotonic Kriging: Bootstrapping and Acceptance/Rejection

In practice we sometimes know (or assume we know) that the I/O function implicitly specified by the simulation model is *monotonic*; e.g., if the traffic rate increases, then the mean waiting time increases. More examples are given in our chapter on screening (Chap. 4). We define a monotonic function as follows (as we also did in Definition 4.1):

**Definition 5.1** *The function  $w = f(x)$  is called monotonically increasing if  $w(x = x_1) \leq w(x = x_2)$  if  $x_1 \leq x_2$ .*

The Kriging metamodel, however, may show a “wiggling” (erratic) I/O function, if the sample size is small; see the wiggly curve in Fig. 5.9. To make the Kriging predictor  $\hat{y}(x_j)$  ( $j = 1, \dots, k$ ) a monotonic function of the input  $x_j$ , we propose *bootstrapping with acceptance/rejection*; i.e., we reject the Kriging metamodel fitted in bootstrap sample  $b$ —with  $b = 1, \dots, B$  and bootstrap sample size  $B$ —if this metamodel is not monotonic. In this section we summarize how Kleijnen and Van Beers (2013) uses distribution-

free bootstrapping assuming stochastic simulation with replications for each input combination; at the end of this section, we shall briefly discuss parametric bootstrapping for deterministic simulation. (The general principles of distribution-free bootstrapping and parametric bootstrapping were discussed in Sect. 3.3.5.)

*Note:* Instead of bootstrapping, Da Veiga and Marrel (2012) solves the monotonicity problem and related problems analytically. However, their solution suffers from the curse of dimensionality; i.e., its *scalability* is questionable.

Kleijnen and Van Beers (2013) uses the popular DACE Matlab Kriging software, which is meant for deterministic simulation so it gives an exact interpolator. Bootstrapped Kriging, however, is not an exact interpolator for the original observations; i.e., its predictor  $\hat{y}^*(\mathbf{x}_i)$  for the  $n$  old input combinations  $\mathbf{x}_i$  ( $i = 1, \dots, n$ ) does not necessarily equal the  $n$  corresponding original average simulated outputs  $\bar{w}_i = \sum_{r=1}^{m_i} w_{i,r}/m_i$  where  $m_i$  ( $\gg 2$ ) denotes the number of replications for input combination  $i$ . Actually, bootstrapped Kriging using DACE is an exact interpolator of the bootstrapped averages  $\bar{w}_i^* = \sum_{r=1}^{m_i} w_{i,r}^*/m_i$ , but not of  $\bar{w}_i$ . A CI is given by the well-known percentile method, now applied to the (say)  $B_a$  ( $\leq B$ ) accepted bootstrapped Kriging predictors  $\hat{y}_{b_a}^*(\mathbf{x})$  ( $b_a = 1, \dots, B_a$ ).

More precisely, a monotonic predictor implies that the estimated *gradients* of the predictor remains positive as the inputs increase; we focus on monotonically increasing functions, because monotonically decreasing functions are a strictly analogous problem. An advantage of monotonic metamodeling is that the resulting sensitivity analysis is understood and accepted by the clients of the simulation analysts so these clients have more confidence in the simulation as a decision support tool. Furthermore, we shall see that monotonic Kriging gives smaller MSE and a CI with higher coverage and acceptable length. Finally, we conjecture that estimated gradients with correct signs will improve simulation optimization, discussed in the next chapter.

Technically speaking, we assume that no CRN are used so the number of replications may vary with the input combination ( $m_i \neq m$ ). Furthermore, we assume a Gaussian correlation function. We let  $\mathbf{x}_i < \mathbf{x}_{i'}$  ( $i, i' = 1, \dots, n; i \neq i'$ ) mean that at least one component of  $\mathbf{x}_i$  is smaller than the corresponding component of  $\mathbf{x}_{i'}$  and none of the remaining components is bigger. For example, the M/M/1 queueing simulation with the traffic rate  $x$  as the single input (so  $k = 1$ ) implies that  $\mathbf{x}_i < \mathbf{x}_{i'}$  becomes  $x_i < x_{i'}$ , whereas the  $(s, S)$  inventory simulation with the  $k = 2$  inputs  $s$  and  $S$  implies that  $\mathbf{x}_i < \mathbf{x}_{i'}$  may mean  $s_i < s_{i'}$  and  $S_i \leq S_{i'}$ . The DACE software gives the estimated gradients  $\nabla \hat{y}(\mathbf{x})$ , besides the prediction  $\hat{y}(\mathbf{x})$ . We use a *test set* with  $v$  “new” points (in the preceding sections we denoted a single new point by

$\mathbf{x}_0$ ). We let  $\lceil x \rceil$  denote the integer resulting from rounding  $x$  upwards,  $\lfloor x \rfloor$  the integer resulting from rounding  $x$  downwards; the subscript  $(\cdot)$  denotes the order statistics.

We propose the following algorithm (which adapts step 1 of Algorithm 5.2, and deviates only in its details but not in its overall goal from the algorithm in Kleijnen and Van Beers 2013); we assume that a 90% CI is desired.

**Algorithm 5.6**

1. Resample the  $m_i$  original outputs  $w_{i;r}$  ( $i = 1, \dots, n; r = 1, \dots, m_i$ ) with replacement, to obtain the bootstrapped output vectors  $\mathbf{w}_{i;b}^* = (w_{i;r;b}^*, \dots, w_{i;r;b}^*)'$  ( $b = 1, \dots, B$ ), which give  $(\mathbf{X}, \bar{\mathbf{w}}_b^*)$  where  $\mathbf{X}$  denotes the  $n \times k$  matrix with the original  $n$  old combinations of the  $k$  simulation inputs and  $\bar{\mathbf{w}}_b^*$  denotes the  $n$ -dimensional vector with the bootstrap averages  $\bar{w}_{i;b}^* = \sum_{r=1}^{m_i} w_{i;r;b}^* / m_i$ .
2. Use DACE to compute  $\hat{\boldsymbol{\psi}}_b^*$ , the MLEs of the Kriging parameters  $\boldsymbol{\psi}$  computed from the bootstrapped I/O data  $(\mathbf{X}, \bar{\mathbf{w}}_b^*)$  of step 1.
3. Apply DACE using  $(\mathbf{X}, \bar{\mathbf{w}}_b^*)$  of step 1 and  $\hat{\boldsymbol{\psi}}_b^*$  of step 2 to compute the Kriging predictor  $\hat{y}_b^*$  that interpolates so  $\hat{y}_b^*(\mathbf{x}_i) = \bar{w}_{i;b}^*$ .
4. Accept the Kriging predictor  $\hat{y}_b^*$  of step 3 only if  $\hat{y}_b^*$  is monotonically increasing; i.e., all  $k$  components of the  $n$  gradients are positive:

$$\nabla \hat{y}_{i;b}^* > \mathbf{0} \quad (i = 1, \dots, n) \tag{5.40}$$

where  $\mathbf{0}$  denotes an  $n$ -dimensional vector with all elements equal to zero.

5. Use the  $B_a$  accepted bootstrapped Kriging metamodels resulting from step 4 to compute  $B_a$  predictions for  $v$  new points  $\mathbf{x}_u$  ( $u = 1, \dots, v$ ) with the point estimate equal to the sample median  $\hat{y}_{u;(\lceil 0.50B_a \rceil)}^*$  and the two-sided 90% CI equal to  $[\hat{y}_{u;(\lfloor 0.05B_a \rfloor)}^*, \hat{y}_{u;(\lceil 0.95B_a \rceil)}^*]$ .

If we find that step 5 gives a CI interval that is too wide, then we add more bootstrap samples so  $B$  increases and  $B_a$  probably increases too. For example, the M/M/1 simulation starts with  $B = 100$  and augments  $B$  with 100 until either  $B_a \geq 100$  or—to avoid excessive computational time— $B = 1,000$ . This M/M/1 example has two performance measures; namely, the mean and the 90% quantile of the steady-state waiting time distribution. Furthermore, the example illustrates both “short” and “long” simulation runs. Finally,  $n = 5$  and  $m_i = 5$  with  $0.1 \leq x \leq 0.9$  and  $v = 25$  new points; also see Fig. 5.9. This plot shows wiggling OK (so  $d\hat{y}/dx$  is negative for at least one  $x$ -value in the area of interest), whereas the bootstrap with acceptance/rejection gives monotonic predictions. This

plot also shows—for each of the  $n = 5$  input values—the  $m = 5$  replicated simulation outputs (see dots) and their averages (see stars). Furthermore, the plot shows the analytical (dotted) I/O curve. Low traffic rates give such small variability of the individual simulation outputs that this variability is hardly visible; nevertheless, the bootstrap finds a monotonic Kriging model.

To quantify the performance of the preceding algorithm, we may use the *integrated mean squared error* (IMSE) defined in Sect. 2.10.1. To estimate the IMSE, we select  $v$  test points. If we let  $\zeta_u$  ( $u = 1, \dots, v$ ) denote the true output at test point  $u$ , then the *estimated integrated mean squared error* (EIMSE) MSE averaged over these  $v$  test points is the *estimated integrated MSE* (EIMSE) is

$$\text{EIMSE} = \frac{\sum_{u=1}^v (\hat{y}_{u;(\lceil 0.50B' \rceil)}^* - \zeta_u)^2}{v}.$$

*Note:* We point out that a disadvantage of the IMSE criterion is that a high MSE at some point  $\mathbf{x}_u$  can be “camouflaged” by a low MSE at some other point  $\mathbf{x}_{u'}$  ( $u \neq u'$ ).

Furthermore, OK uses the CI defined in Eq. (5.21). This CI is symmetric around its point estimate  $\hat{y}$  and may include negative values—even if negative values are impossible, as is the case for waiting times—whether it be the mean or the 90% quantile.

A number of macroreplications (namely, 100) enable the estimation of the variance of the EIMSE estimate and the CI’s coverage and width. These macroreplications show that this algorithm gives a smaller EIMSE than OK does, but this EIMSE is not significantly smaller. Of course, the EIMSE for the 90% quantile is higher than the EIMSE for the mean. This algorithm also gives significantly higher estimated coverages, without widening the CI. Increasing  $n$  (number of old points) from 5 to 10 gives coverages close to the nominal 90%—without significantly longer CIs—whereas OK still gives coverages far below the desired nominal value.

Besides using bootstrapped Kriging with acceptance/rejection to preserve monotonicity, we may also preserve other characteristics of the simulation I/O function; e.g., the Kriging predictions should not be *negative* for waiting times, variances, and thickness. Deutsch (1996) also investigates negative predictions in OK arising when some weights  $\lambda_i$  are negative (see again Sect. 5.2); also see

<http://www.gslib.com/>.

Furthermore, we may apply bootstrapping with acceptance/rejection to other metamodeling methods besides Kriging; e.g., *linear regression* (which we detailed in Chaps. 2 and 3).

If the simulation model is *deterministic*, then there are no replications so we may replace distribution-free bootstrapping by parametric bootstrapping assuming a multivariate Gaussian distribution as implied by a GP; also see Sect. 5.3.

Kleijnen et al. (2012) applies distribution-free bootstrapping with acceptance/rejection to find Kriging metamodels that preserve the assumed *convexity* of the simulation I/O function. Checking this convexity requires extending the DACE software to compute *Hessians*. Unfortunately, it turns out that this method does not give truly convex Kriging prediction functions. On hindsight, we may argue that in practice we do not really know whether the I/O function of the simulation model is convex; e.g., is the cost function of a realistic inventory-simulation model convex? We might assume that the simulation model has a unique optimal solution; convexity implies that the global and the local optima coincide. Da Veiga and Marrel (2012, p. 5) states: “Sometimes, the practitioner further knows that  $f$  (the I/O function) is convex at some locations, due to physical insight”. Jian et al. (2014) develops a Bayesian approach for estimating whether a noisy function is convex.

## 5.8 Global Sensitivity Analysis: Sobol’s FANOVA

So far we focused on the *predictor*  $\hat{y}(\mathbf{x})$ , but now we discuss *sensitivity analysis* (SA) measuring how sensitive the simulation output  $w$  is to the individual inputs  $x_1$  through  $x_k$  and their interactions. Such an analysis may help us to understand the underlying simulation model; i.e., SA may help us to find the important simulation inputs. In the three previous chapters we used polynomials of first order or second order to approximate the simulation I/O function  $w = f_{\text{sim}}(\mathbf{x})$ , so the regression parameters  $\beta$  quantify the first-order and second-order effects of the inputs. OK gives a more complicated approximation; namely, Eq. (5.1) including the extrinsic noise term  $M(\mathbf{x})$  which makes  $y$  a *nonlinear* function of  $\mathbf{x}$ . To quantify the importance of the inputs of the simulation model—possibly approximated through a metamodel—we now apply so-called *functional analysis of variance* (FANOVA). This analysis uses variance-based indexes that were originally proposed by the Russian mathematician Sobol; see Sobol (1990) and the references in Archer et al. (1997).

FANOVA decomposes the variance of the simulation output  $w$  into fractions that refer to the individual inputs or to sets of inputs; e.g., FANOVA may show that 70% of the output variance is caused by the variance in  $x_1$ , 20% by the variance in  $x_2$ , and 10% by the interaction between  $x_1$  and  $x_2$ . As we have already seen in Sect. 5.5.1, we assume that the input  $\mathbf{x}$  has a prespecified (joint) distribution (which may be the product of  $k$  marginal distributions). Below Eq. (5.13) we stated that  $\theta_j$  denotes the importance of  $x_j$ . However, the importance of  $x_j$  is much better quantified through FANOVA, which also measures interactions—as we shall see in this section.

It can be proven that the following variance decomposition—into a sum of  $2^{k-1}$  components—holds:

$$\sigma_w^2 = \sum_{j=1}^k \sigma_j^2 + \sum_{j < j'}^k \sigma_{j;j'}^2 + \dots + \sigma_{1;\dots;k}^2 \quad (5.41)$$

with the main-effect (first order) variance

$$\sigma_j^2 = \text{Var}[E(w|x_j)] \quad (5.42)$$

and the two-factor interaction variance

$$\sigma_{j;j'}^2 = \text{Var}[E(w|x_j, x_{j'})]$$

and so on, ending with the  $k$ -factor interaction variance

$$\sigma_{1;\dots;k}^2 = \text{Var}[E(w|x_1, \dots, x_k)]. \quad (5.43)$$

In Eq. (5.42)  $E(w|x_j)$  denotes the mean of  $w$  if  $x_j$  is kept fixed while all  $k-1$  remaining inputs  $\mathbf{x}_{-j} = (\dots, x_{j-1}, x_{j+1}, \dots)'$  do vary. If  $x_j$  has a “large” main effect, then  $E(w|x_j)$  changes much as  $x_j$  changes. Furthermore, Eq. (5.42) shows  $\text{Var}[E(w|x_j)]$ , which is the variance of  $E(w|x_j)$  if  $x_j$  varies; so if  $x_j$  has a large main effect, then  $\text{Var}[E(w|x_j)]$  is high if  $x_j$  varies. We point out that in Eq. (5.43)  $\text{Var}[E(w|x_1, \dots, x_k)]$  denotes the variance of the mean of  $w$  if all  $k$  inputs are fixed; consequently, this variance is zero in deterministic simulation, and equals the intrinsic noise in stochastic simulation (the intrinsic noise in stochastic simulation may vary with  $\mathbf{x}$ , as we saw in Sect. 5.6).

The measure  $\sigma_j^2$  defined in Eq. (5.42) leads to the following variance-based measure of importance, which the FANOVA literature calls the *first-order sensitivity index* or the *main effect index* and which we denote by  $\gamma$  (we use Greek letters for parameters, throughout this book):

$$\gamma_j = \frac{\sigma_j^2}{\sigma_w^2}.$$

So,  $\gamma_j$  quantifies the effect of varying  $x_j$  alone—averaged over the variations in all the other  $k-1$  inputs;  $\sigma_w^2$  in the denominator standardizes  $\gamma_j$  to provide a fractional contribution (in linear regression analysis we standardize the inputs  $x_j$  so that  $\beta_j$  measures the relative main effect; see Sect. 2.3.1). The interaction indices  $\sigma_{j;j'}^2$  through  $\sigma_{1;\dots;k}^2$  are also divided by  $\sigma_w^2$ . The result of this standardization is the following equation:

$$\sum_{j=1}^k \gamma_j + \sum_{j=1}^{k-1} \sum_{j'=j+1}^k \gamma_{j;j'} + \dots + \gamma_{1;\dots;k} = 1. \quad (5.44)$$

As  $k$  increases, the number of measures in Eqs. (5.41) or (5.44) increases dramatically; actually, this number is  $2^k - 1$  (as we know from classic ANOVA). The estimation of all these measures may require too much computer time, as we shall see below. Moreover, such a large number of measures may be hard to interpret; also see Miller (1956). So—as we did in the immediately preceding three chapters—we might assume that only the first-order measures  $\gamma_j$ —and possibly the second-order measures  $\gamma_{j;j'}$ —are important, and verify whether they sum up to a fraction “close enough” to 1 in Eq. (5.44); i.e., do they contribute the major part of the total variance  $\sigma_w^2$ ?

Alternatively, we might compute the *total-effect index* or *total-order index* (say)  $\gamma_{j;-j}$ , which measures the contribution to  $\sigma_w^2$  due to  $x_j$  including all variance caused by all the interactions between  $x_j$  and any other input variables  $\mathbf{x}_{-j}$ :

$$\gamma_{j;-j} = \frac{E[\text{Var}(w|\mathbf{x}_{-j})]}{\sigma_w^2} = 1 - \frac{\text{Var}[E(w|\mathbf{x}_{-j})]}{\sigma_w^2}.$$

It can be proven that  $\sum_{j=1}^k \gamma_{j;-j} \geq 1$ —unless there are only first-order effects—because the interaction effect between (say)  $x_j$  and  $x_{j'}$  is counted in both  $\gamma_{j;-j}$  and  $\gamma_{j';-j'}$ .

The *estimation* of the various sensitivity measures uses Monte Carlo methods. We may improve the accuracy of the estimators, replacing the “crude” Monte Carlo method by *quasi-Monte Carlo* methods, such as LHS and Sobol sequences (which we discussed in Sect. 5.5). To save computer time, we may replace the simulation model by a metamodel such as an OK model (with a specific correlation function; e.g., the Gaussian function).

*Note:* Details are given in Saltelli et al. (2008, pp. 164–67); also see Fang et al. (2006, pp. 31–33, 193–202), Helton et al. (2006b), Le Gratiet and Cannamela (2015), and Saltelli et al. (2010). The method in Le Gratiet and Cannamela (2015) is available in the package “sensitivity” (linked to the R package DiceKriging).

*Note:* FANOVA is the topic of much current research; see Anderson et al. (2014), Borgonovo and Plischke (2015), Farah and Kottas (2014), Ginsbourger et al. (2015), Henkel et al. (2012), Jeon et al. (2015), Lamboni et al. (2013), Marrel et al. (2012), Muehlenstaedt et al. (2012), Owen et al. (2013), Quaglietta (2013), Razavi and Gupta (2015), Shahraki and Noorossana (2014), Storlie et al. (2009), Tan (2014a), Tan (2014b), Tan (2015), Wei et al. (2015), and Zuniga et al. (2013).

## 5.9 Risk Analysis

In the preceding section on global sensitivity analysis through FANOVA we assumed that the input  $\mathbf{x} \in \mathbb{R}^k$  has a given (joint) distribution. This assumption implies that even a *deterministic* simulation model gives a random output  $w$ ; by definition, a stochastic simulation model always gives a



random output. In *risk analysis* (RA) or *uncertainty analysis* (UA) we may wish to estimate  $P(w > c)$ , which denotes the probability of the output  $w$  exceeding a given threshold value  $c$ . RA is applied in nuclear engineering, finance, water management, etc. A probability such as  $P(w > c)$  may be very small—so  $w > c$  is called a *rare event*—but may have disastrous consequences (we may then apply “importance sampling”; see Kleijnen et al. 2013). In Sect. 1.1 we have already discussed the simple Example 1.1 with the net present value (NPV) as output and the discount factor or the cash flows as uncertain inputs, so the input values are sampled from given distribution functions; spreadsheets are popular software for such NPV computations.

*Note:* Borgonovo and Plischke (2015) applies FANOVA to inventory management models—such as the economic order quantity (EOQ) model—with uncertain inputs. We also refer to the publications that we gave in Sect. 1.1; namely, Evans and Olson (1998) and Vose (2000). Another type of deterministic simulation is used in project planning through the *critical path method* (CPM) and *program evaluation and review technique* (PERT), which in RA allows for uncertain durations of the project components so these durations are sampled from beta distributions; see Lloyd-Smith et al. (2004). More examples of RA are given in Kleijnen (2008, p. 125); also see Helton et al. (2014).

The uncertainty about the exact values of the input values is called *subjective* or *epistemic*, whereas the “intrinsic” uncertainty in stochastic simulation (see Sect. 5.6) is called *objective* or *aleatory*; see Helton et al. (2006a). There are several methods for obtaining subjective distributions for the input  $\mathbf{x}$  based on *expert opinion*.

*Note:* Epistemic and aleatory uncertainties are also discussed in Barton et al. (2014), Batarseh and Wang (2008), Callahan (1996), De Rocquigny et al. (2008), Helton et al. (2010), Helton and Pilch (2011), and Xie et al. (2014).

We emphasize that the goals of RA and SA do differ. SA tries to answer the question “Which are the most important inputs in the simulation model of a given real system?”, whereas RA tries to answer the question “What is the probability of a given (disastrous) event happening?”. We have already seen designs for SA that uses low-order polynomials (which are a type of linear regression metamodels) in the immediately preceding three chapters; designs for RA are samples from the given distribution of the input  $\mathbf{x}$  through Monte Carlo or quasi-Monte Carlo methods, as we discussed in the preceding section on FANOVA (Sect. 5.8). SA identifies those inputs for which the distribution in RA needs further refinement.

*Note:* Similarities and dissimilarities between RA and SA are further discussed in Kleijnen (1983, 1994, 1997), Martin and Simpson (2006), Norton (2015), Oakley and O’Hagan (2004), and Song et al. (2014).

We propose the following algorithm for RA with the goal of estimating  $P(w > c)$ .

**Algorithm 5.7**

1. Use a Monte Carlo method to sample input combination  $\mathbf{x}$  from its given distribution.  
Comment: If the inputs are independent, then this distribution is simply the product of the marginal distributions.
2. Use  $\mathbf{x}$  of step 1 as input into the given simulation model.  
Comment: This simulation model may be either deterministic or stochastic.
3. Run the simulation model of step 2 to transform the input  $\mathbf{x}$  of step 2 into the output  $w$ .  
Comment: This run is called “propagation of uncertainty”.
4. Repeat steps 1 through 3  $n$  times to obtain the estimated distribution function (EDF) of the output  $w$ .
5. Use the EDF of step 4 to estimate the required probability  $P(w > c)$ .

**Exercise 5.7** *Perform a RA of an M/M/1 simulation, as follows. Suppose that you have available  $m$  IID observations on the interarrival time, and on the service time, respectively, denoted by  $a_i$  and  $s_i$  ( $i = 1, \dots, m$ ). Actually, you sample these values from exponential distributions with parameter  $\lambda = \rho$  and  $\mu = 1$  where  $\rho$  is the traffic rate that you select. Resample with replacement (i.e., use distribution-free bootstrapping) to obtain  $m$  interarrival times and  $m$  service times, which you use to estimate the arrival and service rates  $\lambda$  and  $\mu$ . Use this pair of estimated rates as input to your M/M/1 simulation. In this simulation, you observe the output that you are interested in (e.g., the estimated steady-state mean waiting time). Perform  $M$  macroreplications, to estimate the aleatory uncertainty. Repeat the bootstrapping, to find different values for the pair of estimated rates; again simulate the M/M/1 system to estimate the epistemic uncertainty. Compare the effects of both types of uncertainty.*

Because (by definition) an *expensive* simulation model requires much computer time per run, we may perform RA as follows: do not run  $n$  simulation runs (see steps 3 and 4 in the preceding algorithm), but run its *metamodel*  $n$  times. For example, Giunta et al. (2006) uses crude Monte Carlo, LHS, and orthogonal arrays to sample from two types of metamodels—namely, Kriging and multivariate adaptive regression splines (MARS)—and finds that the true mean output can be better estimated through inexpensive sampling of many values from the metamodel, which is estimated from relatively few I/O values obtained from the expensive simulation model (because that publication estimates an expected value, it does not perform a true RA). Another example is Martin and Simpson (2006), using a Kriging metamodel to assess output uncertainty. Furthermore, Barton et al. (2014)

uses bootstrapping and stochastic Kriging (SK) to obtain a CI for the mean output of the real system. Another interesting article on RA is Lemaître et al. (2014). The British research project called *Managing uncertainty in complex models* (MUCM) also studies uncertainty in simulation models, including uncertainty quantification, uncertainty propagation, risk analysis, and sensitivity analysis; see

<http://www.mucm.ac.uk>.

Related to MUCM is the “Society for Industrial and Applied Mathematics (SIAM)”’s “Conference on Uncertainty Quantification (UQ16)”, held in cooperation with the “American Statistical Association (ASA)” and the “Gesellschaft für Angewandte Mathematik und Mechanik (GAMM)”’s “Activity Group on Uncertainty Quantification (GAMM AG UQ)”, in Lausanne (Switzerland), 5–8 April 2016; see

<http://www.siam.org/meetings/uq16/>.

We shall return to uncertainty in the input  $\mathbf{x}$  in the next chapter, in which we discuss robust optimization (which accounts for the uncertainty in some of the inputs); see Sect. 6.4.

Chevalier et al. (2013) and Chevalier et al. (2014) use a Kriging metamodel to estimate the *excursion set* defined as the set of inputs—of a deterministic simulation model—resulting in an output that exceeds a given threshold, and quantifies uncertainties in this estimate; a sequential design may reduce this uncertainty. Obviously, the volume of the excursion set is closely related to the *failure probability*  $P(w > c)$  defined in the beginning of this section. Kleijnen et al. (2011) uses a first-order polynomial metamodel (instead of a Kriging metamodel) to estimate which combinations of uncertain inputs form the frontier that separates acceptable and unacceptable outputs; both aleatory uncertainty—characteristic for random simulation—and epistemic uncertainty are included.

*Note:* Stripling et al. (2011) creates a “manufactured universe” (namely, a nuclear “particle-transport universe”) that generates data on which a simulation model may be built; next, this simulation model generates data to which a metamodel is fitted. This metamodel produces predictions, which may be compared to the true values in the manufactured universe. We may compare this approach with the Monte Carlo experiment in Exercise 5.7, in which the manufactured universe is an M/M/1 system and the metamodel is a SK model; actually, we may use an M/G/1 system—where G stands for general service time distribution (e.g., a lognormal distribution)—and the simulator builds an M/M/1 simulation model with exponential arrival and service parameters estimated from the data generated by the M/G/1 system, so model errors are made besides estimation errors.

RA is related to the *Bayesian* approach, as the latter approach also assumes that the parameters of the simulation model are unknown and assumes given “prior” distributions for these parameters. The Bayesian paradigm selects these prior distributions in a more formal way (e.g., it selects so-called conjugate priors), obtains simulation I/O data, and *calibrates*

the metamodel’s parameters; i.e., it computes the posterior distribution (or likelihood) using the well-known Bayes theorem. *Bayesian model averaging* and *Bayesian melding* formally account—not only for the uncertainty of the input parameters—but also for the uncertainty in the form of the (simulation) model itself. The Bayesian approach is very interesting, especially from an academic point of view; practically speaking, however, classic frequentist RA has been applied many more times. References to the Bayesian approach are given in Kleijnen (2008, p. 126); also see “Bayesian model averaging” in Wit et al. (2012) and the specific Bayesian approach in Xie et al. (2014).

*Note:* We present a methodology that treats the simulation model as a black box, so this methodology can be applied to any simulation model. A disadvantage, however, is that this methodology cannot make use of knowledge about the specific model under discussion; e.g., Bassamboo et al. (2010) uses knowledge about specific call-center queueing models, when examining epistemic and aleatory uncertainties.

## 5.10 Miscellaneous Issues in Kriging

Whereas we focussed on Kriging metamodels for the *mean* simulation output in the preceding sections, Plumlee and Tuo (2014) examines Kriging metamodels for a fixed *quantile* (e.g., the 90% quantile) of the random simulation output. Jala et al. (2014) uses Kriging to estimate a quantile of a deterministic simulation with random input (which results in uncertainty propagation, as we saw in Sect. 5.9). In Sect. 5.6.1 we have already mentioned that Chen and Kim (2013) adapts SK for quantiles, and we have also referred to Bekki et al. (2014), Quadrianto et al. (2009), and Tan (2015).

Another issue is *multivariate Kriging*, which may be applied in *multi-fidelity* metamodeling; i.e., we use several simulation models of the same real system, and each model has its own degree of detail representing the real system. Obviously, the various simulation models give external noises  $M(\mathbf{x})$  that are correlated. An example in *finite element modeling* (FEM) is the use of different simulation models with different meshes (grids). However, we are not aware of much multi-fidelity modeling in discrete-event simulation; however, Xu et al. (2015) does discuss multifidelity in such simulation.

*Note:* Multi-fidelity metamodeling is further discussed in Couckuyt et al. (2014), Koziel et al. (2014), Le Gratiet and Cannamela (2015), Razavi et al. (2012), Tuo et al. (2014), and Viana et al. (2014, Section III).

We may also combine the output of a simulation model with the output of the real system, so-called *field data*. For such problems Goh et al. (2013) uses a Bayesian approach.

In practice, a discrete-event simulation model usually produces *multiple responses*, which have intrinsic noises  $\varepsilon(\mathbf{x})$  that are correlated because these outputs are (different) functions of the same PRNs. For such a simulation

model we might use a multivariate Kriging metamodel. However, Kleijnen and Mehdad (2014) finds that we might as well apply univariate Kriging to each type of simulation response separately. Notice that FANOVA for multivariate Kriging is examined in Zhang (2007) and Zhang et al. (2007). Li and Zhou (2015) considers multivariate GP metamodels for deterministic simulation models with multiple output types.

We may combine Kriging metamodels, each with a different type of correlation function (e.g., Gaussian and exponential) in an *ensemble*; see Harari and Steinberg (2014b), Viana et al. (2014, Figure 5), and the other references in Sect. 1.2.

We may partition the input domain  $\mathbf{x} \in \mathbb{R}^k$  into subdomains, and fit a separate GP model within each subdomain; these subdomains may be determined through *classification and regression trees* (CART); for CART we also refer to Chap. 1. Gramacy and Lee (2008) speak of a *treed Gaussian process*. An R package for treed GPs is available on

<http://users.soe.ucsc.edu/~rbgramacy/tgp.html>.

Another issue in Kriging is the *validation* of Kriging metamodels. In deterministic simulation we may proceed analogously to our validation of linear regression metamodels in deterministic simulation, discussed in Sect. 3.6; i.e., we may compute the coefficients of determination  $R^2$  and  $R_{\text{adj}}^2$ , and apply cross-validation (as we also did in Fig. 5.6). We also refer to the free R package DiceEval; see

<http://cran.r-project.org/web/packages/DiceEval/index.html>.

Scatterplots with  $(w_i, \hat{y}_i)$ —not  $(w_i, \hat{y}_{-i})$  as in cross-validation—are used in many deterministic simulations; an example is the climate simulation in Hankin (2005). The validation of Kriging metamodels is also discussed in Bastos and O’Hagan (2009), following a Bayesian approach. An interesting issue in cross-validation is the fast re-computation of the Kriging model (analogous to the shortcut in Eq. (3.50) for linear regression that uses the hat matrix); also see Hubert and Engelen (2007), discussing fast cross-validation for principle component analysis (PCA).

For deterministic simulations Challenor (2013) and Iooss et al. (2010) examine LHDs with an extra criterion based on the distances between the points in the original and the validation designs (so no cross-validation is applied).

A final issue in Kriging is the variant that Salemi et al. (2013) introduces; namely, *generalized integrated Brownian fields* (GIBFs). Related to these GIBFs are the *intrinsic random functions* that Mehdad and Kleijnen (2015b) introduces into Kriging metamodeling of deterministic and stochastic simulation models, as we have already seen in Sect. 5.4.

## 5.11 Conclusions

In this chapter we started with an introduction of Kriging and its application in various scientific disciplines. Next we detailed OK for deterministic

simulation. For the unbiased estimation of the variance of the OK predictor with estimated Kriging parameters we discussed parametric bootstrapping and conditional simulation. Next we discussed UK for deterministic simulation. Then we surveyed designs for Kriging metamodels, focusing on one-shot standardized LHS and sequentialized, customized designs. We continued with SK for random simulation. To preserve the monotonicity of the I/O function, we proposed bootstrapping with acceptance/rejection. Next we discussed FANOVA using Sobol's sensitivity indexes. Furthermore we discussed RA. Finally, we discussed several remaining issues. Throughout this chapter we also mentioned issues requiring further research.

## Solutions of Exercises

**Solution 5.1**  $E(y|w_1 > \mu, w_2 = \mu, \dots, w_n = \mu) > \mu$  because  $\sigma(x'_0)\Sigma^{-1} > \mathbf{0}'$ .

**Solution 5.2** In general  $\Sigma\Sigma^{-1} = \mathbf{I}$ . If  $\mathbf{x}_0 = \mathbf{x}_i$ , then  $\sigma(x_0)$  is a vector of  $\Sigma$ . So  $\sigma(x_0)'\Sigma^{-1}$  equals a vector with  $n - 1$  zeroes and one element with the value one. So  $\sigma(\mathbf{x}_0)'\Sigma^{-1}(\mathbf{w} - \mu\mathbf{1})$  reduces to  $w_i - \mu$ . Finally,  $\hat{y}(\mathbf{x}_0|\mathbf{w})$  becomes  $\mu + (w_i - \mu) = w_i$ .

**Solution 5.3** If  $\mathbf{x}_0 = \mathbf{x}_1$ , then  $\lambda_1 = 1$  and  $\lambda_2 = \dots = \lambda_n = 0$  (because  $\hat{y}(\mathbf{x}_0)$  is an exact interpolator), so  $\text{Var}[\hat{y}(\mathbf{x}_0)] = 2\text{cov}(y_1, y_1) - [\text{cov}(y_1, y_1) + \text{cov}(y_1, y_1)] = 0$ .

**Solution 5.4** When  $h = 0$ , then  $\rho = 1/\exp(0) = 1/1 = 1$ . When  $h = \infty$ , then  $\rho = 1/\exp(\infty) = 1/\infty = 0$ .

**Solution 5.5** When input  $j$  has no effect on the output, then  $\theta_j = \infty$  in Eq. (5.13) so the correlation function drops to zero.

**Solution 5.6** As  $n$  (number of old points) increases, the new point has neighbors that are closer and have outputs that are more correlated with the output of the new point. So the length of the CI decreases.

**Solution 5.7** The results depend on your choice of the parameters of this Monte Carlo experiment; e.g., the parameter  $m$ .

## References

- Anderson B, Borgonovo E, Galeotti M, Roson R (2014) Uncertainty in climate change modeling: can global sensitivity analysis be of help? Risk Anal 34(2):271–293
- Ankenman B, Nelson B, Staum J (2010) Stochastic kriging for simulation metamodeling. Oper Res 58(2):371–382

- Antognini B, Zagoraiou M (2010) Exact optimal designs for computer experiments via kriging metamodelling. *J Stat Plan Inference* 140(9):2607–2617
- Archer GEB, Saltelli A, Sobol IM (1997) Sensitivity measures, ANOVA-like techniques and the use of bootstrap. *J Stat Comput Simul* 58:99–120
- Ba S, Brennenman WA, Myers WR (2014) Optimal sliced Latin hypercube designs. *Technometrics* (in press)
- Bachoc F (2013) Cross validation and maximum likelihood estimation of hyper-parameters of Gaussian processes with model misspecification. *Comput Stat Data Anal* 66:55–69
- Barton RR, Nelson BL, Xie W (2014) Quantifying input uncertainty via simulation confidence intervals. *INFORMS J Comput* 26(1):74–87
- Bassamboo A, Randhawa RS, Zeevi A (2010) Capacity sizing under parameter uncertainty: safety staffing principles revisited. *Manag Sci* 56(10):1668–1686
- Bastos LS, O’Hagan A (2009) Diagnostics for Gaussian process emulators. *Technometrics* 51(4):425–438
- Batarseh OG, Wang Y (2008) Reliable simulation with input uncertainties using an interval-based approach. In: Mason SJ, Hill RR, Mönch L, Rose O, Jefferson T, Fowler JW (eds) *Proceedings of the 2008 winter simulation conference*, Miami, pp 344–352
- Bekki J, Chen X, Batur D (2014) Steady-state quantile parameter estimation: an empirical comparison of stochastic kriging and quantile regression. In: Tolk A, Diallo SY, Ryzhov IO, Yilmaz L, Buckley S, Miller JA (eds) *Proceedings of the 2014 Winter Simulation Conference*, Savannah, pp 3880–3891
- Borgonovo E, Plischke E (2015) Sensitivity analysis: a review of recent advances. *Eur J Oper Res* (in press)
- Borgonovo E, Tarantola S, Plischke E, Morris MD (2014) Transformations and invariance in the sensitivity analysis of computer experiments. *J R Stat Soc, Ser B* 76:925–947
- Boukouvalas A, Cornford D, Stehlik M (2014) Optimal design for correlated processes with input-dependent noise. *Comput Stat Data Anal* 71:1088–1102
- Bowman VE, Woods DC (2013) Weighted space-filling designs. *J Simul* 7:249–263

- Busby D, Farmer CL, Iske A (2007) Hierarchical nonlinear approximation for experimental designs and statistical data fitting. *SIAM J Sci Comput* 29(1):49–69
- Butler A, Haynes RD, Humphries TD, Ranjan P (2014) Efficient optimization of the likelihood function in Gaussian process modelling. *Comput Stat Data Anal* 73:40–52
- Callahan BG (ed) (1996) Special issue: commemoration of the 50th anniversary of Monte Carlo. *Hum Ecol Risk Assess* 2(4):627–1037
- Challenor P (2013) Experimental design for the validation of Kriging metamodels in computer experiments. *J Simul* (7):290–296
- Chen EJ, Li M (2014) Design of experiments for interpolation-based metamodels. *Simul Model Pract Theory* 44:14–25
- Chen VCP, Tsui K-L, Barton RR, Meckesheimer M (2006) A review on design, modeling, applications of computer experiments. *IIE Trans* 38:273–291
- Chen X, Ankenman B, Nelson BL (2012) The effects of common random numbers on stochastic Kriging metamodels. *ACM Trans Model Comput Simul* 22(2):7:1–7:20
- Chen X, Kim K-K (2013) Building metamodels for quantile-based measures using sectioning. In: Pasupathy R, Kim S-H, Tolk A, Hill R, Kuhl ME (eds) *Proceedings of the 2013 winter simulation conference*, Washington, DC, pp 521–532
- Chen X, Wang K, Yang F (2013) Stochastic kriging with qualitative factors. In: Pasupathy R, Kim S-H, Tolk A, Hill R, Kuhl ME (eds) *Proceedings of the 2013 winter simulation conference*, Washington, DC, pp 790–801
- Chen X, Zhou Q (2014) Sequential experimental designs for stochastic kriging. In: Tolk A, Diallo SD, Ryzhov IO, Yilmaz L, Buckley S, Miller JA (eds) *Proceedings of the 2014 winter simulation conference*, Savannah, pp 3821–3832
- Chevalier C, Ginsbourger D (2012) Corrected Kriging update formulae for batch-sequential data assimilation. arXiv, 1203.6452v1
- Chevalier C, Ginsbourger D, Bect J, Molchanov I (2013) Estimating and quantifying uncertainties on level sets using the Vorob'ev expectation and deviation with Gaussian process models. In: Uciniski D, Atkinson AC, Patan M (eds) *mODa 10 – advances in model-oriented design and analysis; proceedings of the 10th international workshop in model-oriented design and analysis*. Springer, New York, pp 35–43



- Chevalier C, Ginsbourger D, Bect J, Vazquez E, Picheny V, Richet Y (2014) Fast parallel Kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics* 56(4): 455–465
- Chilès J-P, Delfiner P (2012) *Geostatistics: modeling spatial uncertainty*, 2nd edn. Wiley, New York
- Clark I (2010) Statistics or geostatistics? Sampling error or nugget effect? *J S Afr Inst Min Metall* 110:307–312
- Couckuyt I, Dhaene T, Demeester P (2014) ooDACE toolbox: a flexible object-oriented Kriging implementation. *J Mach Learn Res* 15:3183–3186
- Couckuyt I, Forrester A, Gorissen D, Dhaene T (2012) Blind kriging; implementation and performance analysis. *Adv Eng Softw* 49:1–13
- Cressie NAC (1993) *Statistics for spatial data*, rev edn. Wiley, New York
- Crombecq K, Laermans E, Dhaene T (2011) Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling. *Eur J Oper Res* 214:683–696
- Damblin G, Couplet M, Iooss B (2013) Numerical studies of space-filling designs: optimization of Latin hypercube samples and subprojection properties. *J Simul* 7:276–289
- Da Veiga S, Marrel A (2012) Gaussian process modeling with inequality constraints. *Annales de la faculté des sciences de Toulouse Sér. 6* 21(3):529–555
- De Rocquigny E, Devictor N, Tarantola S (2008) Uncertainty settings and natures of uncertainty. In: de Rocquigny E, Devictor N, Tarantola S (eds) *Uncertainty in industrial practice*. Wiley, Chichester
- Den Hertog D, Kleijnen JPC, Siem AYD (2006) The correct Kriging variance estimated by bootstrapping. *J Oper Res Soc* 57(4):400–409
- Deng H, Shao W, Ma Y, Wei Z (2012) Bayesian metamodeling for computer experiments using the Gaussian Kriging models. *Qual Reliab Eng* 28(4):455–466
- Dette H, Pepelyshev A (2010) Generalized Latin hypercube design for computer experiments. *Technometrics* 25:421–429
- Deutsch CV (1996) Correcting for negative weights in ordinary Kriging. *Comput Geosci* 22(7):765–773
- Deutsch JL, Deutsch CV (2012) Latin hypercube sampling with multidimensional uniformity. *J Stat Plan Inference* 142(3):763–772

- Evans JR, Olson DL (1998) Introduction to simulation and risk analysis. Prentice-Hall, Upper Saddle River
- Fang K-T, Li R, Sudjianto A (2006) Design and modeling for computer experiments. Chapman & Hall/CRC, London
- Farah M, Kottas A (2014) Bayesian inference for sensitivity analysis of computer simulators, with an application to radiative transfer models. *Technometrics* 56(2):159–173
- Forrester AIJ (2013) Comment: properties and practicalities of the expected quantile improvement. *Technometrics* 55(1):13–18
- Forrester AIJ, Keane AJ (2009) Recent advances in surrogate-based optimization. *Prog Aerosp Sci* 45(1–3):50–79
- Forrester A, Sóbester A, Keane A (2008) Engineering design via surrogate modelling: a practical guide. Wiley, Chichester
- Frazier PI (2011) Learning with dynamic programming. In: Cochran JJ, Cox LA, Keskinocak P, Kharoufeh JP, Smith JC (eds) *Encyclopedia of operations research and management science*. Wiley, New York
- Gano SE, Renaud JE, Martin JD, Simpson TW (2006) Update strategies for Kriging models for using in variable fidelity optimization. *Struct Multidiscip Optim* 32(4):287–298
- Georgiou SD, Stylianou S (2011) Block-circulant matrices for constructing optimal Latin hypercube designs. *J Stat Plan Inference* 141:1933–1943
- Ghosh BK, Sen PK (eds) (1991) *Handbook of sequential analysis*. Marcel Dekker, New York
- Ginsbourger D, Dupuy D, Badea A, Carraro L, Roustant O (2009) A note on the choice and the estimation of Kriging models for the analysis of deterministic computer experiments. *Appl Stoch Models Bus Ind* 25: 115–131
- Ginsbourger D, Iooss B, Pronzato L (2015) Editorial. *J Stat Comput Simul* 85(7):1281–1282
- Giunta AA, McFarland JM, Swiler LP, Eldred MS (2006) The promise and peril of uncertainty quantification using response surface approximations. *Struct Infrastruct Eng* 2(3–4):175–189
- Goel T, Haftka R, Queipo N, Shyy W (2006) Performance estimate and simultaneous application of multiple surrogates. In: 11th AIAA/ISSMO multidisciplinary analysis and optimization conference, multidisciplinary analysis optimization conferences. American Institute of Aeronautics and Astronautics, Reston, VA 20191–4344, pp 1–26

- Goh J, Bingham D, Holloway JP, Grosskopf MJ, Kuranz CC, Rutter E (2013) Prediction and computer model calibration using outputs from multi-fidelity simulators. *Technometrics* 55(4):501–512
- Goldberg PW, Williams CKI, Bishop CM (1998) Regression with input-dependent noise: a Gaussian process treatment. In: Jordan MI, Kearns MJ, Solla SA (eds) *Advances in neural information processing systems*, vol 10. MIT, Cambridge, pp 493–499
- Golzari A, Sefat MH, Jamshidi S (2015) Development of an adaptive surrogate model for production optimization. *J Petrol Sci Eng* (in press)
- Gramacy RB and Haaland B (2015) Speeding up neighborhood search in local Gaussian process prediction. *Technometrics* (in press)
- Gramacy RB, Lee HKH (2008) Bayesian treed Gaussian process models with an application to computer modeling. *J Am Stat Assoc* 103(483):1119–1130
- Gramacy RB, Lee HKH (2012) Cases for the nugget in modeling computer experiments. *Stat Comput* 22:713–722
- Grosso A, Jamali ARMJU, Locatelli M (2009) Finding maximin Latin hypercube designs by iterated local search heuristics. *Eur J Oper Res* 197(2):541–54
- Hankin RKS (2005) Introducing BACCO, an R bundle for Bayesian analysis of computer code output. *J Stat Softw* 14(16):1–21
- Harari O, Steinberg DM (2014a) Optimal designs for Gaussian process models via spectral decomposition. *J Stat Plan Inference* (in press)
- Harari O, Steinberg DM (2014b) Convex combination of Gaussian processes for Bayesian analysis of deterministic computer experiments. *Technometrics* 56(4):443–454
- Helton JC, Davis FJ, Johnson JD (2005) A comparison of uncertainty and sensitivity results obtained with random and Latin hypercube sampling. *Reliab Eng Syst Saf* 89:305–330
- Helton JC, Johnson JD, Oberkampf WD, Sallaberry CJ (2006a) Sensitivity analysis in conjunction with evidence theory representations of epistemic uncertainty. *Reliab Eng Syst Saf* 91:1414–1434
- Helton JC, Johnson JD, Oberkampf WD, Sallaberry CJ (2010) Representation of analysis results involving aleatory and epistemic uncertainty. *Int J Gen Syst* 39(6):605–646

- Helton JC, Johnson JD, Sallaberry CJ, Storlie CB (2006b) Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliab Eng Syst Saf* 91:1175–1209
- Helton JC, Pilch M (2011) Guest editorial: quantification of margins and uncertainty. *Reliab Eng Syst Saf* 96:959–964
- Helton JC, Hansen CW, Sallaberry CJ (2014) Conceptual structure and computational organization of the 2008 performance assessment for the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Reliab Eng Syst Saf* 122:223–248
- Henkel T, Wilson H, Krug W (2012) Global sensitivity analysis of non-linear mathematical models – an implementation of two complementing variance-based algorithms. In: Laroque C, Himmelspach J, Pasupathy R, Rose O, Uhrmacher AM (eds) *Proceedings of the 2012 winter simulation conference*, Washington, DC, pp 1737–1748
- Hernandez AF, Grover MA (2010) Stochastic dynamic predictions using Gaussian process models for nanoparticle synthesis. *Comput Chem Eng* 34(12):1953–1961
- Hernandez AS, Lucas TW, Sanchez PJ (2012) Selecting random Latin hypercube dimensions and designs through estimation of maximum absolute pairwise correlation. In: Laroque C, Himmelspach J, Pasupathy R, Rose O, Uhrmacher AM (eds) *Proceedings of the 2012 winter simulation conference*, Berlin, pp 280–291
- Hubert M, Engelen S (2007) Fast cross-validation of high-breakdown resampling methods for PCA. *Comput Stat Data Anal* 51(10):5013–5024
- Iooss B, Boussouf L, Feuillard V, Marrel A (2010) Numerical studies of the metamodel fitting and validation processes. *Int J Adv Syst Meas* 3:11–21
- Jala M, Lévy-Leduc C, Moulines É, Conil E, Wiart J (2014) Sequential design of computer experiments for the assessment of fetus exposure to electromagnetic fields. *Technometrics* (in press)
- Janssen H (2013) Monte-Carlo based uncertainty analysis: sampling efficiency and sampling convergence. *Reliab Eng Syst Saf* 109:123–132
- Jeon JS, Lee SR, Pasquinelli L, Fabricius IL (2015) Sensitivity analysis of recovery efficiency in high-temperature aquifer thermal energy storage with single well. *Energy* (in press)
- Jian N, Henderson S, Hunter SR (2014) Sequential detection of convexity from noisy function evaluations. In: Tolk A, Diallo SY, Ryzhov IO, Yilmaz L, Buckley S, Miller JA (eds) *Proceedings of the 2014 winter simulation conference*, Savannah, pp 3892–3903

- Jin, R, Chen W, Sudjianto A (2002) On sequential sampling for global metamodeling in engineering design. In: Proceedings of DET'02, ASME 2002 design engineering technical conferences and computers and information in engineering conference, DETC2002/DAC-34092, Montreal, 29 Sept–2 Oct 2002
- Jones B, Silvestrini RT, Montgomery DC, Steinberg DM (2015) Bridge designs for modeling systems with low noise. *Technometrics* 57(2): 155–163
- Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *J Glob Optim* 13:455–492
- Joseph VR, Hung Y, Sudjianto A (2008) Blind Kriging : a new method for developing metamodels. *J Mech Des* 130(3):31–102
- Jourdan A, Franco J (2010) Optimal Latin hypercube designs for the Kullback-Leibler criterion. *AStA Adv Stat Anal* 94:341–351
- Kamiński B (2015) A method for updating of stochastic Kriging metamodels. *Eur J Oper Res* (accepted)
- Kersting K, Plagemann C, Pfaff P, Burgard W (2007) Most-likely heteroscedastic Gaussian process regression. In: Ghahramani Z (ed) Proceedings of the 24th annual international conference on machine learning (ICML-07), Corvalis, pp 393–400
- Kleijnen JPC (1983). Risk analysis and sensitivity analysis: antithesis or synthesis?. *Simuletter*, 14(1–4):64–72
- Kleijnen JPC (1990) Statistics and deterministic simulation models: why not? In: Balci O, Sadowski RP, Nance RE (eds) Proceedings of the 1990 winter simulation conference, Washington, DC, pp 344–346
- Kleijnen JPC (1994) Sensitivity analysis versus uncertainty analysis: when to use what? In: Grasman J, van Straten G (eds) Predictability and non-linear modelling in natural sciences and economics. Kluwer, Dordrecht, pp 322–333
- Kleijnen JPC (1997) Sensitivity analysis and related analyses: a review of some statistical techniques. *J Stat Comput Simul* 57(1–4):111–142
- Kleijnen JPC (2008) Design and analysis of simulation experiments. Springer, New York
- Kleijnen JPC (2009) Kriging metamodeling in simulation: a review. *Eur J Oper Res* 192(3):707–716
- Kleijnen JPC (2014) Simulation-optimization via Kriging and bootstrapping: a survey. *J Simul* 8(4):241–250

- Kleijnen JPC, Mehdad E (2013) Conditional simulation for efficient global optimization. In: Pasupathy R, Kim S-H, Tolk A, Hill R, Kuhl ME (eds) Proceedings of the 2013 winter simulation conference, Washington, DC, pp 969–979
- Kleijnen JPC, Mehdad E (2014) Multivariate versus univariate Kriging metamodels for multi-response simulation models. *Eur J Oper Res* 236:573–582
- Kleijnen JPC, Mehdad E (2015) Estimating the correct predictor variance in stochastic Kriging. CentER Discussion Paper, 2015, Tilburg
- Kleijnen JPC, Mehdad E, Van Beers WCM (2012) Convex and monotonic bootstrapped Kriging. In: Laroque C, Himmelspace J, Pasupathy R, Rose O, Uhrmacher AM (eds) Proceedings of the 2012 winter simulation conference, Washington, DC, pp 543–554
- Kleijnen JPC, Pierreval H, Zhang J (2011) Methodology for determining the acceptability of system designs in uncertain environments. *Eur J Oper Res* 209:176–183
- Kleijnen JPC, Ridder AAN, Rubinstein RY (2013) Variance reduction techniques in Monte Carlo methods. In: Gass SI, Fu MC (eds) Encyclopedia of operations research and management science, 3rd edn. Springer, New York, pp 1598–1610
- Kleijnen JPC, Van Beers WCM (2004) Application-driven sequential designs for simulation experiments: Kriging metamodeling. *J Oper Res Soc* 55(9):876–883
- Kleijnen JPC, Van Beers WCM (2013) Monotonicity-preserving bootstrapped Kriging metamodels for expensive simulations. *J Oper Res Soc* 64:708–717
- Koch P, Wagner T, Emmerich MTM, Bäck T, Konen W (2015) Efficient multi-criteria optimization on noisy machine learning problems. *Appl Soft Comput* (in press)
- Koziel S, Bekasiewicz A, Couckuyt I, Dhaene T (2014) Efficient multi-objective simulation-driven antenna design using co-Kriging. *IEEE Trans Antennas Propag* 62(11):5901–5915
- Krige DG (1951) A statistical approach to some basic mine valuation problems on the Witwatersrand. *J Chem, Metall Min Soc S Afr* 52(6):119–139
- Lamboni M, Iooss B, Popelin A-L, Gamboa F (2013) Derivative-based global sensitivity measures: general links with Sobol indices and numerical tests. *Math Comput Simul* 87:45–54

- Lancaster P, Salkauskas K (1986) Curve and surface fitting: an introduction. Academic, London
- Law AM (2015) Simulation modeling and analysis, 5th edn. McGraw-Hill, Boston
- Le Gratiet L, Cannamela C (2015) Cokriging-based sequential design strategies using fast cross-validation techniques for multi-fidelity computer codes. *Technometrics* 57(3):418–427
- Lemaître P, Sergienko E, Arnaud A, Bousquet N, Gamboa F, Iooss B (2014) Density modification based reliability sensitivity analysis. *J Stat Comput Simul* (in press)
- Lemieux C (2009) Monte Carlo and quasi-Monte Carlo sampling. Springer, New York
- Li K, Jiang B, Ai M (2015) Sliced space-filling designs with different levels of two-dimensional uniformity. *J Stat Plan Inference* 157–158:90–99
- Li R, Sudjianto A (2005) Analysis of computer experiments using penalized likelihood in Gaussian Kriging models. *Technometrics* 47(2):111–120
- Li Y, Zhou Q (2015) Pairwise meta-modeling of multivariate output computer models using nonseparable covariance function. *Technometrics* (in press)
- Lin Y, Mistree F, Allen JK, Tsui K-L, Chen VCP (2004) Sequential meta-modeling in engineering design. In: 10th AIAA/ISSMO symposium on multidisciplinary analysis and optimization, Albany, 30 Aug–1 Sept, 2004. Paper number AIAA-2004-4304
- Lin Y, Mistree F, Tsui K-L, Allen JK (2002) Metamodel validation with deterministic computer experiments. In: 9th AIAA/ISSMO symposium on multidisciplinary analysis and optimization, Atlanta, 4–6 Sept 2002. Paper number AIAA-2002-5425
- Lloyd-Smith B, Kist AA, Harris RJ, Shrestha N (2004) Shortest paths in stochastic networks. In: Proceedings 12th IEEE international conference on networks 2004, Wakefield, MA, vol 2, pp 492–496
- Loeppky JL, Sacks J, Welch W (2009) Choosing the sample size of a computer experiment: a practical guide. *Technometrics* 51(4):366–376
- Lophaven SN, Nielsen HB, Sondergaard J (2002) DACE: a Matlab Kriging toolbox, version 2.0. IMM Technical University of Denmark, Kongens Lyngby

- MacCalman AD, Vieira H, Lucas T (2013) Second order nearly orthogonal Latin hypercubes for exploring stochastic simulations. Naval Postgraduate School, Monterey
- McKay MD, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21(2):239–245 (reprinted in *Technometrics*, 42(1,2000):55–61)
- Marrel A, Iooss B, Da Veiga S, Ribatet M (2012) Global sensitivity analysis of stochastic computer models with joint metamodels. *Stat Comput* 22:833–847
- Marrel A, Iooss B, Van Dorpe F, Volkova E (2008) An efficient methodology for modeling complex computer codes with Gaussian processes. *Comput Stat Data Anal* 52:4731–4744
- Martin JD, Simpson TW (2005) Use of Kriging models to approximate deterministic computer models. *AIAA J* 43(4):853–863
- Martin JD, Simpson TW (2006) A methodology to manage system-level uncertainty during conceptual design. *ASME J Mech Des* 128(4):959–968
- Matheron G (1963) Principles of geostatistics. *Econ Geol* 58(8):1246–1266
- Mehdad E, Kleijnen JPC (2015a) Classic Kriging versus Kriging with bootstrapping or conditional simulation: classic Kriging’s robust confidence intervals and optimization. *J Oper Res Soc* (in press)
- Mehdad E, Kleijnen JPC (2015b) Stochastic intrinsic Kriging for simulation metamodeling. CentER Discussion Paper, Tilburg
- Meng Q, Ng SH (2015, in press) An additive global and local Gaussian process model for large datasets. In: Yilmaz L, Chan WKV, Moon I, Roeder TMK, Macal C, Rossetti MD (eds) Proceedings of the 2015 winter simulation conference. [Will be made available on the WSC website in January 2016, after the conference in Dec. 2015]
- Miller GA (1956) The magical number seven plus or minus two: some limits on our capacity for processing information. *The Psychol Rev* 63:81–97
- Mitchell TJ, Morris MD (1992) The spatial correlation function approach to response surface estimation. In: Swain JJ, Goldsman D, Crain RC, Wilson JR (eds) Proceedings of the 1992 winter simulation conference, Arlington
- Moutoussamy V, Nanty S, Pauwels B (2014) Emulators for stochastic simulation codes. In: ESAIM: Proceedings, Azores, pp 1–10



- Muehlenstaedt T, Roustant O, Carraro L, Kuhnt S (2012) Data-driven Kriging models based on FANOVA-decomposition. *Stat Comput* 22:723–738
- Ng SH, Yin J (2012), Bayesian Kriging analysis and design for stochastic simulations. *ACM Trans Model Comput Simul* 22(3):1–26
- Norton J (2015) An introduction to sensitivity assessment of simulation models. *Environ Model Softw* 69:166–174
- Oakley J, O’Hagan A (2004) Probabilistic sensitivity analysis of complex models: a Bayesian approach. *J R Stat Soc, Ser B*, 66(3):751–769
- Opsomer JD, Ruppert D, Wand MP, Holst U, Hossjer O (1999) Kriging with nonparametric variance function estimation. *Biometrics* 55(3): 704–710
- Owen AB, Dick J, Chen S (2013) Higher order Sobol’ indices. <http://arxiv.org/abs/1306.4068>
- Plumlee M, Tuo R (2014) Building accurate emulators for stochastic simulations via quantile Kriging, *Technometrics* 56(4):466–473
- Qian PZG, Hwang Y, Ai M, Su H (2014) Asymmetric nested lattice samples. *Technometrics* 56(1):46–54
- Qu H, Fu MC (2014) Gradient extrapolated stochastic kriging. *ACM Trans Model Comput Simul* 24(4):23:1–23:25
- Quadrianto N, Kersting K, Reid MD, Caetano TS, Buntine WL (2009) Kernel conditional quantile estimation via reduction revisited. In: *IEEE 13th international conference on data mining (ICDM)*, Miami, pp 938–943
- Quaglietta E (2013) Supporting the design of railway systems by means of a Sobol variance-based sensitivity analysis. *Transp Res Part C* 34:38–54
- Ranjan P, Spencer N (2014) Space-filling Latin hypercube designs based on randomization restrictions in factorial experiments. *Stat Probab Lett* (in press)
- Rasmussen CE, Nickisch H (2010) Gaussian processes for machine learning (GPML) toolbox. *J Mach Learn Res* 11:3011–3015
- Rasmussen CE, Williams C (2006) *Gaussian processes for machine learning*. MIT, Cambridge
- Razavi S, Tolson BA, Burn DH (2012) Review of surrogate modeling in water resources. *Water Resour Res* 48, W07401:1–322

- Razavi S, Gupta HV (2015) What do we mean by sensitivity analysis? The need for comprehensive characterization of “global” sensitivity in earth and environmental systems models. *Water Resour Res* 51 (in press)
- Risk J, Ludkovski M (2015) Statistical emulators for pricing and hedging longevity risk products. Preprint arXiv:1508.00310
- Roustant O, Ginsbourger D, Deville Y (2012) DiceKriging, DiceOptim: two R packages for the analysis of computer experiments by Kriging-based metamodeling and optimization. *J Stat Softw* 51(1):1–55
- Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments (includes comments and rejoinder). *Stat Sci* 4(4):409–435
- Salemi P, Staum J, Nelson BL (2013) Generalized integrated Brownian fields for simulation metamodeling. In: Pasupathy R, Kim S-H, Tolk A, Hill R, Kuhl ME (eds) *Proceedings of the 2013 winter simulation conference*, Washington, DC, pp 543–554
- Saltelli A, Annoni P, Azzini I, Campolongo F, Ratto M, Tarantola S (2010) Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Comput Phys Commun* 181:259–270
- Saltelli A, Ratto M, Andres T, Campolongo F, Cariboni J, Gatelli D, Saisana M, Tarantola S (2008) *Global sensitivity analysis: the primer*. Wiley, Chichester
- Santner TJ, Williams BJ, Notz WI (2003) *The design and analysis of computer experiments*. Springer, New York
- Shahraki AF, Noorossana R (2014) Reliability-based robust design optimization: a general methodology using genetic algorithm. *Comput Ind Eng* 74:199–207
- Simpson TW, Booker AJ, Ghosh D, Giunta AA, Koch PN, Yang R-J (2004) Approximation methods in multidisciplinary analysis and optimization: a panel discussion. *Struct Multidiscip Optim* 27(5):302–313
- Simpson TW, Mauery TM, Korte JJ, Mistree F (2001) Kriging metamodels for global approximation in simulation-based multidisciplinary design. *AIAA J* 39(12):853–863
- Sobol IM (1990) Sensitivity estimates for non-linear mathematical models. *Matematicheskoe Modelirovanie* 2:112–118
- Song E, Nelson BL, Pegden D (2014) Advanced tutorial: input uncertainty quantification. In: Tolk A, Diallo SY, Ryzhov IO, Yilmaz L, Buckley S, Miller JA (eds) *Proceedings of the 2014 winter simulation conference*, Savannah, pp 162–176

- Spöck G, Pilz J (2015) Incorporating covariance estimation uncertainty in spatial sampling design for prediction with trans-Gaussian random fields. *Front Environ Sci* 3(39):1–22
- Stein ML (1999) *Statistical interpolation of spatial data: some theory for Kriging*. Springer, New York
- Storlie CB, Swiler LP, Helton JC, Sallaberry CJ (2009) Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models. *Reliab Eng Syst Saf* 94(11):1735–1763
- Stripling HF, Adams ML, McClarren RG, Mallick BK (2011) The method of manufactured universes for validating uncertainty quantification methods. *Reliab Eng Syst Saf* 96(9):1242–1256
- Sun L, Hong LJ, Hu Z (2014) Balancing exploitation and exploration in discrete optimization via simulation through a Gaussian process-based search. *Oper Res* 62(6):1416–1438
- Sundararajan S, Keerthi SS (2001) Predictive approach for choosing hyperparameters in Gaussian processes. *Neural Comput* 13(5):1103–1118
- Tajbakhsh DS, Del Castillo E, Rosenberger JL (2014) A fully Bayesian approach to sequential optimization of computer metamodels for process improvement. *Qual Reliab Eng Int* 30(4):449–462
- Tan MHY (2014a) Robust parameter design with computer experiments using orthonormal polynomials. *Technometrics* (in press)
- Tan MHY (2014b) Stochastic polynomial interpolation for uncertainty quantification with computer experiments. *Technometrics* (in press)
- Tan MHY (2015) Monotonic quantile regression with Bernstein polynomials for stochastic simulation. *Technometrics* (in press)
- Thiart C, Ngwenya MZ, Haines LM (2014) Investigating ‘optimal’ kriging variance estimation using an analytic and a bootstrap approach. *J S Afr Inst Min Metall* 114:613–618
- Toal DJJ, Bressloff NW, Keane AJ (2008) Kriging hyperparameter tuning strategies. *AIAA J* 46(5):1240–1252
- Toropov VV, Schramm U, Sahai A, Jones R, Zeguer T (2005) Design optimization and stochastic analysis based on the moving least squares method. In: 6th world congress of structural and multidisciplinary optimization, Rio de Janeiro, paper no. 9412
- Tuo RC, Wu FJ, Yuc D (2014) Surrogate modeling of computer experiments with different mesh densities. *Technometrics* 56(3):372–380

- Ulaganathan S, Couckuyt I, Dhaene T, Laermans E (2014) On the use of gradients in Kriging surrogate models. In: Tolk A, Diallo SY, Ryzhov IO, Yilmaz L, Buckley S, Miller JA (eds) Proceedings of the 2014 winter simulation conference, Savannah, pp 2692–2701
- Van Beers WCM, Kleijnen JPC (2003) Kriging for interpolation in random simulation. *J Oper Res Soc* 54:255–262
- Van Beers WCM, Kleijnen JPC (2008) Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. *Eur J Oper Res* 186(3):1099–1113
- Viana FAC, Haftka RT (2009) Cross validation can estimate how well prediction variance correlates with error. *AIAA J* 47(9):2266–2270
- Viana FAC, Simpson TW, Balabanov V, Toropov V (2014) Metamodeling in multidisciplinary design optimization: how far have we really come? *AIAA J* 52(4):670–690
- Vieira H, Sanchez S, Kienitz KH, Belderrain MCN (2011) Generating and improving orthogonal designs by using mixed integer programming. *Eur J Oper Res* 215:629–638
- Vose D (2000) *Risk analysis; a quantitative guide*, 2nd edn. Wiley, Chichester
- Wackernagel H (2003) *Multivariate geostatistics: an introduction with applications*, 3rd edn. Springer, Berlin
- Wang C, Duan Q, Gong W, Ye A, Di Z, Miao C (2014) An evaluation of adaptive surrogate modeling based optimization with two benchmark problems. *Environ Model Softw* 60:167–179
- Wei P, Lu Z, Song J (2015) Variable importance analysis: a comprehensive review. *Reliab Eng Syst Saf* 142:399–432
- Wit E, Van den Heuvel E, Romeijn J-W (2012) All models are wrong . . . : an introduction to model uncertainty, *Statistica Neerlandica* 66(3):217–236
- Xie W, Nelson BL, Barton RR (2014) A Bayesian framework for quantifying uncertainty in stochastic simulation. *Oper Res* (in press)
- Xu J, Zhang S, Huang E, Chen C-H, Lee H, Celik N (2014) Efficient multi-fidelity simulation optimization. In: Tolk A, Diallo SY, Ryzhov IO, Yilmaz L, Buckley S, Miller JA (eds) Proceedings of the 2014 winter simulation conference, Savannah, pp 3940–3951
- Yang X, Chen H, Liu MQ (2014) Resolvable orthogonal array-based uniform sliced Latin hypercube designs. *Stat Probab Lett* 93:108–115

- Yin J, Ng SH, Ng KM (2009) A study on the effects of parameter estimation on Kriging model's prediction error in stochastic simulation. In: Rossini MD, Hill RR, Johansson B, Dunkin A, Ingalls RG (eds) Proceedings of the 2009 winter simulation conference, Austin, pp 674–685
- Yin J, Ng SH, Ng KM (2010) A Bayesian metamodeling approach for stochastic simulations. In: Johansson B, Jain S, Montoya-Torres J, Hagan J, Yücesan E (eds) Proceedings of the 2010 winter simulation conference, Baltimore, pp 1055–1066
- Yuan J, Ng SH (2015) An integrated approach to stochastic computer model calibration, validation and prediction. *Trans Model Comput Simul* **25**(3), Article No. 18
- Zhang Z (2007) New modeling procedures for functional data in computer experiments. Doctoral dissertation, Department of Statistics, Pennsylvania State University, University Park
- Zhang Z, Li R, Sudjianto A (2007) Modeling computer experiments with multiple responses. *SAE Int* 2007-01-1655
- Zhou Q, Qian PZG, Zhou S (2011) A simple approach to emulation for computer models with qualitative and quantitative factors. *Technometrics* **53**:266–273
- Zuniga MM, Kucherenko S, Shah N (2013) Metamodelling with independent and dependent inputs. *Comput Phys Commun* **184**(6):1570–1580