Tilburg University

Kriging

Kleijnen, J.P.C.

Publication date:
2017

Document Version
Early version, also known as pre-print

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):
Kleijnen, J. P. C. (2017). *Kriging: Methods and Applications*. (CentER Discussion Paper; Vol. 2017-047). CentER, Center for Economic Research.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

No. 2017-047

KRIGING: METHODS AND APPLICATIONS

By

Jack P.C. Kleijnen

21 November 2017

ISSN 0924-7815
ISSN 2213-9532

Kriging: methods and applications

Jack P.C. Kleijnen

Tilburg University, Postbox 90153, 5000 LE Tilburg, Netherlands

November 21, 2017

Abstract

In this chapter we present Kriging—also known as a Gaussian process (GP) model—which is a mathematical interpolation method. To select the input combinations to be simulated, we use Latin hypercube sampling (LHS); we allow uniform and non-uniform distributions of the simulation inputs. Besides deterministic simulation we discuss random simulation, which requires adjusting the design and analysis. We discuss sensitivity analysis of simulation models, using "functional analysis of variance" (FANOVA)—also known as Sobol sensitivity indexes. Finally, we discuss optimization of the simulated system, including "robust" optimization.

[This chapter is prepared for "New mathematics inspired by industrial challenges", edited by Wil Schilders and Michael Günther, Special volume in ECMI book series, Springer]

Keywords: Gaussian process, Latin hypercube, deterministic simulation, random simulation, sensitivity analysis, optimization

JEL: C0, C1, C9, C15, C44

1 Introduction

Kriging is the mathematical interpolation method that is named after the South African mining-engineer Krige (1919–2013). He solved the problem of interpolating results (outputs, responses) that were obtained at a limited number of locations for gold mining; see the details on Krige's life in Minnitt and Assibey-Bonsu (2013).

Next, this method was formalized by the French mathematician Matheron (1930–2000), who developed the novel mathematical statistics called geostatistics or spatial statistics. The basis of this formalization is the stationary *Gaussian process* (GP). This stationarity means that the GP has a constant mean (expected value), a constant variance, and covariances that depend only on the distances between "points" in a (say) k -dimensional space; obviously, in spatial statistics $k \leq 3$ (length, width, height). This GP defines a multivariate normal (or Gaussian) distribution. Spatial statistics is detailed in the popular textbook (with 900 pages) Cressie (1993); books reflecting the French tradition

are Chilès and Delfiner (1999) and Wackernagel (2003) (the connection between Krige and Matérn is also discussed in Minnitt and Assibey-Bonsu (2013)).

Later on, these GPs were applied in *machine learning*, which is a subdiscipline within computer science. The best known textbook on GPs in machine learning is Rasmussen and Williams (2006).

Actually, in this chapter we focus on the development and application of GPs in experiments with computerized simulation models; this field is known as *design and analysis of computer experiments* (DACE). Obviously, these simulation models may have many inputs, which implies $k \gg 1$. The pioneering article on DACE is Sacks et al. (1989); a more recent textbook is Santner et al. (2003). DACE publications focus on *deterministic* simulation models, whereas we shall also discuss *random* or *stochastic* simulation models, which use *pseudorandom numbers* (PRNs); by definition, PRNs are uniformly distributed on $[0, 1]$. The design and analysis of experiments with random simulation models require the development of novel statistical methods such as *sample-size determination* and *stochastic Kriging* (SK); see the pioneering article Ankenman et al. (2010).

We call a GP a *metamodel* of the underlying simulation model; i.e., this metamodel is a simpler and explicit mathematical function that approximates the complicated and implicit function defined by the simulation model (that is either deterministic or random). The DACE literature often calls the metamodel a *surrogate* or an *emulator*. We shall see (in the next section) that Kriging also quantifies the uncertainty of its predictor (say) $\hat{y}$; i.e., Kriging also gives $\text{Var}(\hat{y})$, whereas many other metamodels (e.g., neural nets, splines) do not.

Kriging may have different *goals*; see Kleijnen (2015, p. 9). We shall discuss prediction, sensitivity analysis, and optimization. Besides these topics, Baudin et al. (2017) also discusses uncertainty quantification and uncertainty propagation.

Note: Simulation is applied in many scientific disciplines, which have their own terminologies and mathematical symbols. We use the terminology and symbols in Kleijnen (2015); e.g., we write Gaussian and Kriging with capitals (because these words refer to the names Gauss and Krige), and we use the symbol k (instead of d). Hasty readers may skip paragraphs that start with "Note:".

Note: Mehdad and Kleijnen (2017a) and Mehdad and Kleijnen (2017b) consider so-called *intrinsic Kriging* (IK)—which originated in geostatistics—and derive several IK types for deterministic simulations and random simulations respectively.

We base this chapter on the Chapters 5 and 6 in Kleijnen (2015), but we update these chapters, adding novel methods and applications for Kriging. Furthermore, corrections and additions of Kleijnen (2015) are available on <https://sites.google.com/site/kleijnenjackpc/home/publications/corrections-additions-of-2015-springer-book>. Besides Kleijnen (2015) we also use Kleijnen and Van Beers (2017). We point out that we do not use a Bayesian approach.

We organize the rest of this chapter as follows. In Section 2 we present so-called "ordinary Kriging" (OK), comparing OK with popular linear-regression metamodels. In Section 3 we present the "Latin hypercube sampling" (LHS)

design to select the input combinations to be simulated, which results in the input/output (I/O) data analyzed by OK. In Section 4 we consider the adjustments in the design and analysis through OK that are needed in random simulation. In Section 5 we discuss sensitivity analysis (SA) of simulation models. In Section 6 we discuss optimization of the simulated system. In Section 7 we present our conclusions, including topics that require further research.

2 Ordinary Kriging

To explain OK, we start with the following *linear regression* model (we assume that the readers are familiar with this model; otherwise, they can read Chapters 2 and 3 in Kleijnen (2015); we add the subscript "reg", to distinguish between regression and OK models):

$$\mathbf{y}_{\text{reg}} = \mathbf{X}_{\text{reg}}\boldsymbol{\beta}_{\text{reg}} + \mathbf{e}_{\text{reg}} \quad (1)$$

where $\mathbf{y}_{\text{reg}}$ denotes the n -dimensional vector with the observations on the dependent (explained) regression variable where n denotes the number of observed (simulated) input combinations or "points", $\mathbf{X}_{\text{reg}}$ is the $n \times q$ matrix of independent (explanatory) regression variables, $\boldsymbol{\beta}_{\text{reg}}$ is the q -dimensional vector with regression parameters (coefficients), and $\mathbf{e}_{\text{reg}}$ is the n -dimensional vector with the residuals $E(\mathbf{y}_{\text{reg}}) - E(\mathbf{w})$ where $\mathbf{w}$ is the n -dimensional vector with simulated outputs. $\mathbf{X}_{\text{reg}}$ consists of the n q -dimensional row vectors $\mathbf{x}_{\text{reg};i}$ with $i = 1, \dots, n$. We assume a univariate (scalar) output. There are k simulation inputs x_j ($j = 1, \dots, k$). An independent regression variable may be identical to a simulation input or it may be a function of one or more simulation inputs; e.g., such a regression variable may be the square of an input or the product of two inputs. Classic regression analysis assumes that $\mathbf{e}_{\text{reg}}$ is *white noise*; i.e., $\mathbf{e}_{\text{reg}}$ is normally (Gaussian) distributed with zero means, constant variances (say) σ^2 , and zero correlations so the covariance matrix of $\mathbf{e}_{\text{reg}}$ is $\boldsymbol{\Sigma}_{\text{reg}} = \sigma^2 \mathbf{I}_{n \times n}$ where $\mathbf{I}_{n \times n}$ denotes the $n \times n$ identity matrix.

Because white noise implies independent residuals, we cannot learn from the residuals, whereas Kriging assumes that the residuals at the two points $\mathbf{x}_i$ and $\mathbf{x}_{i'}$ have values that are more similar, as $\mathbf{x}_i$ and $\mathbf{x}_{i'}$ are closer. So, Kriging assumes that the residuals are positively correlated. The simplest Kriging model is the *OK* model

$$y(\mathbf{x}) = \mu + M(\mathbf{x}) \quad (2)$$

where μ is the constant mean $E[y(\mathbf{x})]$ and $M(\mathbf{x})$ is a zero-mean *stationary GP* (a

more complicated type of Kriging may replace this μ by a low-order polynomial; see universal Kriging or UK, discussed in the Note at the end of this section). We call $M(\mathbf{x})$ the *extrinsic noise*, because we shall use the term "intrinsic noise" for random simulation analyzed through SK. The OK model defined in (2) leads to the *linear predictor* (say) $\hat{y}(\mathbf{x}_0)$ for the *new* point $\mathbf{x}_0 = (x_{0;j})$ that combines the *n old* outputs collected in $\mathbf{w}_n$ —or briefly $\mathbf{w}$ —that are observed at the *n old*

points with k inputs, so the $n \times k$ matrix $\mathbf{X}$ has the rows $\mathbf{x}_i = (x_{i1}, \dots, x_{ik})$ with $i = 1, \dots, n$ and $\hat{y}(\mathbf{x}_0)$ uses the n weights λ_i collected in the vector $\boldsymbol{\lambda}$:

$$\hat{y}(\mathbf{x}_0) = \sum_{i=1}^n \lambda_i w(\mathbf{x}_i) = \boldsymbol{\lambda}' \mathbf{w}. \quad (3)$$

To derive the optimal values for λ_i , we use the *best linear unbiased predictor* (BLUP) criterion. By definition, the predictor is unbiased if $E[\hat{y}(\mathbf{x})] = E[y(\mathbf{x})]$. This implies that $\mathbf{x} = \mathbf{x}_0$ gives $\hat{y}(\mathbf{x}_0) = w(\mathbf{x}_0)$: *exact interpolation*. This interpolation makes perfect sense in deterministic simulation. A regression model determines the optimal β_{reg} in (1) through the criterion of the *best linear unbiased estimator* (BLUE) of β_{reg} where "best" means "minimum variance"; this BLUE is $\hat{\beta}_{\text{reg}} = (\mathbf{X}'_{\text{reg}} \mathbf{X}_{\text{reg}})^{-1} \mathbf{X}'_{\text{reg}} \mathbf{w}$.

Note: This BLUE is identical to the "maximum likelihood" (ML) estimator and the "least squares" (LS) estimator. LS is a mathematical instead of a statistical criterion; instead of this L_2 -norm, some mathematical models use either the L_1 -norm or the L_∞ -norm. The BLUE is not always equal to the MLE; e.g., the BLUE of σ^2 has the denominator $n - 1$, whereas the MLE has n . Obviously, $\hat{\mathbf{y}}_{\text{reg}}(\mathbf{x}) = \mathbf{x}'_{\text{reg}} \hat{\beta}_{\text{reg}}$ does not interpolate—unless $n = q$.

Altogether, the optimal $\boldsymbol{\lambda}'$ in (3) is

$$\lambda'_o = [\sigma_M(\mathbf{x}_0) + \mathbf{1}' \frac{1 - \mathbf{1}' \boldsymbol{\Sigma}_M^{-1} \sigma(\mathbf{x}_0)}{\mathbf{1}' \boldsymbol{\Sigma}_M^{-1} \mathbf{1}}] \boldsymbol{\Sigma}_M^{-1} \quad (4)$$

where $\boldsymbol{\Sigma}_M = (\sigma_{i;i'}) = (\text{Cov}(y_i, y_{i'}))$ ($i, i' = 1, \dots, n$) denotes the $n \times n$ matrix

with the covariances between the metamodel's "old" outputs y_i , and $\sigma_M(\mathbf{x}_0) = (\sigma_{0;i}) = (\text{Cov}(y_0, y_i))$ denotes the n -dimensional vector with the covariances between the metamodel's new output y_0 and the n old outputs y_i . Obviously, $\boldsymbol{\Sigma}_M$ is determined by the old I/O simulation data $(\mathbf{X}, \mathbf{w})$, whereas $\sigma_M(\mathbf{x}_0)$ varies with $\mathbf{x}_0$ (so we might write $\lambda_o(\mathbf{x}_0)$ if we would want to point out that λ_o varies with $\mathbf{x}_0$, whereas $\hat{\beta}_{\text{reg}}$ remains constant). Furthermore, a stationary process implies that λ_i decreases with the *distance* between $\mathbf{x}_0$ and $\mathbf{x}_i$. We substitute λ_o —defined in (4)—into (3), and we let $\mathbf{1}_n$ denote the n -dimensional vector with all elements equal to 1; this gives

$$\hat{y}(\mathbf{x}_0) = \mu + \sigma_M(\mathbf{x}_0)' \boldsymbol{\Sigma}_M^{-1} (\mathbf{w} - \mu \mathbf{1}_n). \quad (5)$$

To denote $\text{Var}(y_i) = \sigma_{i;i} = \sigma_i^2 = \sigma^2$, we use the symbol τ^2 (τ^2 is the more usual symbol in the Kriging literature). Then

$$\text{MSE} [\hat{y}(\mathbf{x}_0)] = \tau^2 - \sigma_M(\mathbf{x}_0)' \boldsymbol{\Sigma}_M^{-1} \sigma_M(\mathbf{x}_0) + \frac{[1 - \mathbf{1}'_n \boldsymbol{\Sigma}_M^{-1} \sigma_M(\mathbf{x}_0)]^2}{\mathbf{1}'_n \boldsymbol{\Sigma}_M^{-1} \mathbf{1}_n}. \quad (6)$$

Because $\hat{y}(\mathbf{x}_0)$ is unbiased, we obtain $\text{MSE} [\hat{y}(\mathbf{x}_0)] = \text{Var}[\hat{y}(\mathbf{x}_0)]$. Finally, $\text{Var}[\hat{y}(\mathbf{x}_0)] = 0$ if $\mathbf{x}_0 = \mathbf{x}_i$.

Sometimes, it is convenient to switch from covariances to correlations; i.e., we switch to the correlation matrix $\mathbf{R} = (\rho_{i;i'})$, which equals $\tau^{-2} \boldsymbol{\Sigma}_M$; obviously,

$\rho(\mathbf{x}_0) = \tau^{-2}\sigma_M(\mathbf{x}_0)$. There are several types of correlation functions; see (e.g.) Rasmussen and Williams (2006, pp. 80–104). In simulation, most popular is the *Gaussian correlation function*:

$$\rho(\mathbf{h}, \theta) = \prod_{j=1}^k \exp(-\theta_j h_j^2) = \exp\left(-\sum_{j=1}^k \theta_j h_j^2\right) \text{ with } \theta_j \geq 0 \quad (7)$$

with distance vector $\mathbf{h} = (h_j)$ where $h_j = |x_{g;j} - x_{g';j}|$ and $g, g' = 0, 1, \dots, n$,

and with $\theta = (\theta_j)$ so $\mathbf{R} = \mathbf{R}(\theta)$. We collect the *Kriging (hyper)parameters* in $\psi = (\mu, \tau^2, \theta')'$. This ψ is estimated by the *ML estimator* (MLE) $\hat{\psi}$. To compute $\hat{\psi}$, it is convenient to switch to the log-likelihood function:

$$\min_{\psi} \ln[|\tau^2 \mathbf{R}(\theta)|] + (\mathbf{w} - \mu \mathbf{1}_n)' [\tau^2 \mathbf{R}(\theta)]^{-1} (\mathbf{w} - \mu \mathbf{1}_n) \text{ with } \theta \geq \mathbf{0} \quad (8)$$

where $|\mathbf{R}|$ denotes the determinant of $\mathbf{R}$. Solving (8) is a mathematical chal-

lenge; e.g., different solutions $\hat{\psi}$ may result from different software packages or from initializing the same package with different starting values; see Erickson et al. (2017). We observe that the Kriging software *standardizes* the k inputs such that they are limited to a k -dimensional hypercube $[0, 1]^k$.

In practice, we *plug* the estimator $\hat{\psi}$ into (5) and (6), so we get $\hat{y}(\mathbf{x}_0, \hat{\psi})$ and $\text{Var}[\hat{y}(\mathbf{x}_0, \hat{\psi})]$. Most publications ignore that $\hat{y}(\mathbf{x}_0, \hat{\psi})$ becomes nonlinear, and $\text{Var}[\hat{y}(\mathbf{x}_0, \hat{\psi})]$ underestimates the ("true") Kriging variance. However, Kleijnen (2015, pp. 191–197)—and also Chevalier et al. (2014)—use bootstrapping—and the related conditional simulation—to estimate the true Kriging variance. Nevertheless, we shall use the simple plug-in estimator $\hat{\psi}$, and do not explicitly display the dependence on $\hat{\psi}$. This gives

$$s^2[\hat{y}(\mathbf{x}_0)] = \hat{\tau}^2 - \hat{\sigma}_M(\mathbf{x}_0)' \hat{\Sigma}_M^{-1} \hat{\sigma}_M(\mathbf{x}_0) + \frac{[1 - \mathbf{1}' \hat{\Sigma}_M^{-1} \hat{\sigma}_M(\mathbf{x}_0)]^2}{\mathbf{1}' \hat{\Sigma}_M^{-1} \mathbf{1}}. \quad (9)$$

We combine $\hat{y}(\mathbf{x}_0)$ and $s^2[\hat{y}(\mathbf{x}_0)]$ in a two-sided confidence interval (CI) with nominal coverage $1 - \alpha$; i.e., we assume

$$P[w \in \hat{y}(\mathbf{x}_0) \pm z_{\alpha/2} s^2[\hat{y}(\mathbf{x}_0)]] = 1 - \alpha. \quad (10)$$

The actual (true) coverage of this CI may be lower than $1 - \alpha$, because of the following three factors: (i) the plug-in predictor $\hat{y}(\mathbf{x}_0)$ is biased; (ii) the plug-in variance estimator $s^2[\hat{y}(\mathbf{x}_0)]$ underestimates $\text{Var}[\hat{y}(\mathbf{x}_0)]$; and (iii) the absolute value of the Normal quantile $|z_{\alpha/2}|$ is lower than the absolute value of the Student statistic with (say) f degrees of freedom $|t_{f;\alpha/2}|$ if $f < \infty$, where $t_{f;\alpha/2}$ with the proper (but unknown) f seems to be the correct factor for a CI that uses an estimated variance.

Note: If we replace the constant $\mu = E(y)$ by a trend (e.g., $E(y) = \beta' \mathbf{x}$), then we get UK; details on UK are found in Kleijnen (2015, pp. 197–198) and also in Chen et al. (2016) and Mukhopadhyay et al. (2016).

3 Latin hypercube sampling

Kleijnen (2015, p. 198) discusses LHS and alternative *space-filling designs*; e.g., orthogonal array, uniform, maximum entropy, minimax, maximin, integrated mean squared prediction error, and “optimal” designs. Benková et al. (2015) discusses space-filling designs that may satisfy various criteria and input constraints such that the input space is not a k -dimensional cube, including so-called bridge designs. Chen et al. (2016) shows that “there is substantial variation in prediction accuracy over equivalent designs”.

Originally, McKay et al. (1979) invented LHS (as an alternative for crude Monte Carlo sampling) for *risk analysis or uncertainty analysis* through deterministic simulation models that have (random) uncertain inputs; risk analysis estimates the probability of the output exceeding a given threshold as a function of an uncertain input x_j (for details on risk analysis we refer to Kleijnen (2015, pp. 218–222) and Baudin et al. (2017)). LHS assumes that an adequate metamodel is more complicated than a low-order polynomial, but LHS does not assume a specific metamodel (e.g., a Kriging model). LHS usually assumes that the k inputs are *independently* distributed (so their joint distribution is the product of the k individual marginal distributions); in this chapter we also use this assumption. Often LHS assumes that these distributions are *uniform* (symbol U) in the interval $[0, 1]$, so $x_j \sim U(0, 1)$. In risk analysis, however, LHS often assumes a specific *non-uniform* distribution for x_j with its mode at $x_{0;j}$ where $x_{0;j}$ denotes the j^{th} coordinate of $\mathbf{x}_0$ and $j = 1, \dots, k$ (our standardization implies $0 \leq x \leq 1$). This mode may be the value of the input that the experts think is most likely. There are many non-uniform distributions; see Law (2015, pp. 286–305). For example, we may use *beta distributions*, provided we select the correct values for the two parameters of the beta distribution; moreover, a different combination of these parameters gives a different variation around the mode; see Law (2015, pp. 295–297). We discuss a special case of the beta distributions; namely, a triangular distribution with its mode at $x_{m;j}$; we denote this distribution by $T(x_{m;j})$. We shall detail LHS for $U(0, 1)$ and for $T(x_{m;j})$ later on in this section. More details on LHS can be found in Kleijnen (2015, pp. 198–203); recent algorithms are detailed in Dong and Nakayama (2017), and Le Guiban et al. (2017)).

Whatever the marginal distributions are, LHS with a sample size (number of points) n defines n mutually exclusive and exhaustive subintervals (or classes) with equal probability (namely, $1/n$) for x_j with $j = 1, \dots, k$. We denote these subintervals by $[l_{g;j}, h_{g;j}]$ with $g = 1, \dots, n$; the standardization $0 \leq x_j \leq 1$ implies $l_{1;j} = 0$ and $h_{n;j} = 1$. Altogether, if F_j denotes the *cumulative distribution function* (CDF) of x_j , then

$$P(l_{g;j} \leq x_j \leq h_{g;j}) = F_j(h_{g;j}) - F_j(l_{g;j}) = F_j\left(\frac{g}{n}\right) - F_j\left(\frac{g-1}{n}\right) = \frac{1}{n} \quad (11)$$

where $\min(g-1)/n = (1-1)/n = 0$, so $F_j((g-1)/n) = F_j(0) = 0$ because $\min x_j = 0$; likewise, $\max(g)/n = n/n = 1$, so $F_j(g/n) = F_j(1) = 1$ because $\max x_j = 1$. Obviously, (11) implies that near the mode of $T(x_{m;j})$ the subintervals $[l_{g;j},$

$h_{g;j}]$ are relatively short, compared with the other subintervals; however, if $x_j \sim U(0, 1)$, then each interval has length $1/n$.

LHS offers the following two *options*. Option (i) fixes x_j to the n *midpoints* (symbol m_j) of its n subintervals, so $x_{g;j} = m_{g;j} = (l_{g;j} + h_{g;j})/2$; e.g., if $x \sim U(0, 1)$, then these midpoints are equispaced with distance $1/n$ over the interval $[0, 1]$ so these midpoints are $1/(2n)$, $3/(2n)$, ..., $1 - 1/(2n) = (2n - 1)/(2n)$. Option (ii) samples x_j within its subinterval, accounting for F_j . Option (i) implies that x_j has a *discrete* PDF, so (11) becomes

$$P(x_{g;j} = m_{g;j}) = F_j(h_{g;j}) - F_j(l_{g;j}) = F_j\left(\frac{g}{n}\right) - F_j\left(\frac{g-1}{n}\right) = \frac{1}{n}. \quad (12)$$

Furthermore, LHS *samples without replacement*, so the midpoint $m_{j;g}$ is sampled only once (in the sample of size n). We denote the *inverse CDF* by F_j^{-1} , so $y = F_j(x)$ with $0 \leq y \leq 1$ implies $x = F_j^{-1}(y)$; we observe that $U(0, 1)$ and $T(x_m)$ imply that F_j is continuous. Altogether, $\mathbf{x}_j$ is a permutation of the n values $F_j^{-1}(0.5/n)$, $F_j^{-1}(1.5/n)$, ..., $F_j^{-1}(1 - 0.5/n)$. Option (ii) first samples $r \sim U(a, b)$ with $a = (g-1)/n$ and $b = g/n$ where $g = 1, \dots, n$; next this option computes

$$x = F_j^{-1}(r) \text{ with } r \sim U\left(\frac{g-1}{n}, \frac{g}{n}\right). \quad (13)$$

Algorithm 1 is a (pseudo)algorithm for LHS for option (i), which gives the $n \times k$ design matrix $\mathbf{X}_L$ (the subscript L stands for LHS); also see Kleijnen (2015, p. 200).

Algorithm 1

1. Read n, k, F_j ($j = 1, \dots, k$).
2. Initialize: $j = 1$.
3. Use F_j to divide the range of x_j into n mutually exclusive and exhaustive intervals of equal probability with midpoint $m_{j;g}$ ($g = 1, \dots, n$), and find $\mathbf{x}_j = (m_{j;1}, m_{j;2}, \dots, m_{j;n})'$.
4. Randomly permute the n elements of $\mathbf{x}_j$, and save the result as column j of $\mathbf{X}_L$.
5. If $j < k$ then $j = j + 1$ and go to Step 3; else stop.

For option (ii), Step 3 becomes: Use F_j to divide the range of x_j into n mutually exclusive and exhaustive intervals of equal probability, and apply (13) to find $\mathbf{x}_j = (x_{j;1}, x_{j;2}, \dots, x_{j;n})'$.

For both options, however, the *random* permutations in Step 4 may give a "bad" $\mathbf{X}_L$; to decide on a "good" $\mathbf{X}_L$, our algorithm needs a criterion. We use the *maximin* criterion, which maximizes the minimum Euclidean distance between the n k -dimensional points in $[0, 1]^k$. We perform these random permutations (say) M times, and select the "best" design; e.g., MATLAB's default is $M = 5$.

LHS does not impose a strict mathematical relationship between n and k . We observe that in Kriging $n \geq 10$ implies that we can estimate θ_j with $j = 1, \dots, k$ reasonably accurate, because LHS implies that projection of the n k -dimensional points onto the k individual axes gives n non-collapsing values per axis. Loeppky et al. (2009) gives the rule-of-thumb $n = 10k$ for LHS in Kriging aimed at sensitivity analysis; so, $n \geq 10$ if $k \geq 1$. Nevertheless, if LHS uses a "small" n and a "large" k , then LHS covers $[0, 1]^k$ sparsely (so there are only a few old points close to the new point) and the Kriging predictor is inadequate.

There is much *software* for LHS. For example, Microsoft's Excel spreadsheet software has add-ins that include LHS; LHS is also included in Oracle's Crystal Ball, Palisade's @Risk, and Frontline Systems' Risk Solver. LHS is also available in the MATLAB Statistics toolbox, the R package, and Sandia's DAKOTA software. An interesting website is <http://www.spacefillingdesigns.nl/>. However, some software (e.g., MATLAB) does not allow a non-uniform distribution such as $T(x_m)$, so we now present LHS for $U(0, 1)$ and $T(x_m)$; for details we refer to Kleijnen and Van Beers (2017).

The CDF of $U(0, 1)$ is $F_{U;j}(x) = x$ if $0 \leq x \leq 1$. This CDF and (12) imply that option (i) samples $U(0, 1)$ through

$$x_{j;g} = m_{g;j} \text{ if } l_{g;j} < x_{g;j} < h_{g;j} \quad (g = 1, \dots, n)$$

This CDF and (13) imply that option (ii) samples $U(0, 1)$ through the PRN $r \sim U(0, 1)$ and computes

$$x_{j;g} = l_{g;j} + r(h_{g;j} - l_{g;j}) \text{ if } l_{g;j} < x_{g;j} < h_{g;j} \quad (g = 1, \dots, n)$$

The CDF of $T(x_m)$ is (see Law (2015, pp. 304–305)):

$$\begin{aligned} F_{T;j}(x) &= \frac{x^2}{x_{m;j}} \text{ if } 0 \leq x \leq x_{m;j} \\ F_{T;j}(x) &= 1 - \frac{(1-x)^2}{1-x_{m;j}} \text{ if } x_{m;j} \leq x \leq 1, \end{aligned} \tag{14}$$

so this CDF has a *kink* at $x_{m;j}$. Combining this equation with (12), option (i) samples

$$\begin{aligned} x_{j;g} &= \sqrt{\frac{x_{m;j}(2g-1)}{2n}} \text{ with } x_g \leq x_{m;j} \quad (g = 1, \dots, n) \\ x_{j;g} &= 1 - \sqrt{\frac{(1-x_{m;j})[2n-(2g-1)]}{2n}} \text{ with } x_{j;g} \geq x_{m;j} \end{aligned} \tag{15}$$

Option (ii) samples x_j (within the specific subinterval) via the first line of (14) if $h_{g;j} < x_{m;j}$. If $l_{g;j} > x_{m;j}$, then it samples x_j via the second line of (14). If $l_{g;j} < x_{m;j} < h_{g;j}$, then it first samples the PRN $r \sim U(0, 1)$; if $r < x_{m;j}$, then

it samples x_j via the first line of (14); if $r > x_{m;j}$, then it samples x_j via the second line of (14). Altogether, option (ii) samples

$$\begin{aligned} x_{j;g} &= \sqrt{x_{m;j}r} \text{ if either } x_{m;j} > h_{g;j} \text{ or } l_{g;j} < x_{m;j} < h_{g;j} \text{ and } r < x_{m;j} \\ x_{j;g} &= 1 - \sqrt{(1 - x_{m;j})(1 - r)} \text{ if either } x_{m;j} < l_{g;j} \text{ or } l_{g;j} < x_{m;j} < h_{g;j} \text{ and } r > x_{m;j}. \end{aligned} \quad (16)$$

Both options are compared in Kleijnen and Van Beers (2017).

Kleijnen and Van Beers (2004) proposes a sequential design for Kriging in deterministic simulation aimed at SA. That approach considers (but does not yet simulate) a set of candidate combinations selected through LHS, and finds the candidate with the highest predictor variance. This approach is summarized in Kleijnen (2015, pp. 204–206).

4 Random simulation: Kriging analysis and experimental design

In random simulation we wish to reduce the noise of the simulation output at point $\mathbf{x}_i$ ($i = 1, \dots, n$), so we obtain (say) $m_i \geq 1$ replications. Ankenman et al. (2010) develops *stochastic Kriging* (SK), adding the *intrinsic noise* term $\varepsilon_r(\mathbf{x}_i) \sim N(0, \text{Var}[\varepsilon(\mathbf{x}_i)])$ for replication r ($r = 1, \dots, m_i$). After averaging over these m_i replications, SK uses the formulas for the OK predictor $\hat{y}$ in (5) and $\text{Var}(\hat{y})$ in (6), but replaces $\mathbf{w}$ by $\bar{\mathbf{w}}$ and $M(\mathbf{x}_i)$ by $M(\mathbf{x}_i) + \bar{\varepsilon}(\mathbf{x}_i)$ where $\bar{\varepsilon}(\mathbf{x}_i) \sim N(0, \text{Var}[\varepsilon(\mathbf{x}_i)]/m_i)$ and $\bar{\varepsilon}(\mathbf{x}_i)$ is assumed to be independent of $M(\mathbf{x})$. We use the symbol $\psi_{+\varepsilon}$ to denote ψ augmented with $\text{Var}[\varepsilon(\mathbf{x}_i)]$. The SK predictor $\hat{y}(\psi_{+\varepsilon})$ is not an exact interpolator anymore—which makes sense in random simulation (which gives only estimates of the true simulation output). Obviously, $\Sigma_{\bar{\varepsilon}}$ is diagonal if no *common random numbers* (CRN) are used (if we applied CRN and used $m_i = m$, then $\Sigma_{\bar{\varepsilon}} = \Sigma_{\varepsilon}/m$; however, we assume no CRN in this section). To estimate $\text{Var}[\varepsilon(\mathbf{x}_i)]$, SK may use the classic unbiased estimator

$$s^2(w_i) = \frac{\sum_{r=1}^{m_i} (w_{i;r} - \bar{w}_i)^2}{m_i - 1} \quad (i = 1, \dots, n). \quad (17)$$

However, these $s^2(w_i)$ are rather noisy estimators, so SK may use a second Kriging metamodel for $\text{Var}[\varepsilon(\mathbf{x}_i)]$ —besides the Kriging metamodel for the mean $E[y(\mathbf{x}_i)]$. This second metamodel is only a rough approximation, because $s^2(w_i)$ is not normally distributed. The transformation $\log[s^2(w_i)]$ may give a normal distribution. For more details we refer to Kleijnen (2015, p.208).

An alternative for SK is *hetGP* developed in Binois et al. (2016). This alternative assumes $m_i \geq 1$, whereas SK assumes $m_i \gg 1$. Whereas SK gives a biased $\hat{\psi}_{+\varepsilon}$ because SK fits Kriging models for the mean and the intrinsic variances independently, hetGP couples these models through a joint likelihood for $\psi_{+\varepsilon}$ that is optimized in one shot. This alternative requires computational time of the same order as SK does.

The output of a random simulation may be a *quantile* (not an average); e.g., a quantile may be relevant in *chance-constrained optimization*. References are given in Kleijnen (2015, p. 208).

In our preceding discussion of Kriging, we assumed $m_i \geq 1$ replications at the point $\mathbf{x}_i$ with $i = 1, \dots, n$. In practice, we must decide *how many points* and *which points* to simulate, and *how many replications* to obtain for these points. Unfortunately, there are no simple solutions for this problem; we present the following ideas. Because we may design simulation experiments sequentially, we can start with a relatively small number of points (say) n_0 ; e.g., $n_0 = 10k$ (see again Loepky et al. (2009)). The selection of the initial value for m_i (say) m_0 is a moot issue; an obvious minimum is $m_0 = 2$, as shown by the denominator of (17). Sacks et al. (1989) introduced the criterion called the *integrated mean squared prediction error* (IMSPE), which integrates $\text{Var}[\hat{y}(\mathbf{x})]$ for $\mathbf{x}$ in $[0, 1]^k$. Later on, Ankenman et al. (2010) also used this criterion. Obviously, this criterion assumes that the goal of Kriging is SA; if the goal is optimization, then other criteria may be used (see Section 6). It is common practice to select the n_0 specific initial points through LHS. The main problem is that m_i with $i = 1, \dots, n_0$ should depend on both M (external noise) and $\Sigma_{\bar{\epsilon}}$ (internal noise). To solve this problem, Mehdad and Kleijnen (2017b) first ignores both $\Sigma_{\bar{\epsilon}}$ and the condition that m_i should be integer; the resulting allocation of the given total number of simulation observations N still depends on both M and $\Sigma_{\bar{\epsilon}}$. Because these M and $\Sigma_{\bar{\epsilon}}$ need to be estimated, Mehdad and Kleijnen (2017) uses the following two-stage approach, originally proposed in Ankenman et al. (2010, p. 378). Stage 1 obtains a pilot sample of n_0 points and allocates $m_0 > 10$ replications to each of these points, which enables the estimation of M and $\Sigma_{\bar{\epsilon}}$. Stage 2 first selects $n - n_0$ points, and then "optimally" allocates the remaining $N - n_0 m_0$ observations over these n points.

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.

5 Sensitivity analysis

SA may be one of the goals of metamodeling. Kleijnen (2015, p. 9) writes that a goal of computer experiments may be SA—either global or local—which is also called *what if* analysis of the simulation model; in practice, these goals may be ambiguous, and may be known under other names; e.g., SA may aim at either “gaining insight” or “prediction”. In this chapter we give the following *definition* of SA: SA quantifies how the simulation output changes, as one or more simulation inputs change. To start our discussion of SA, we assume that the simulation is deterministic and that we change only one of the k inputs and that this single input (say) x_1 is continuous. Then we can quantify this sensitivity at a given point $\mathbf{x}$ through $\partial w / \partial x_1 | \mathbf{x}$. We have the simplest SA if we can adequately approximate the I/O behavior of the simulation model by a first-order polynomial in $\mathbf{x}$: SA then gives $\partial w / \partial x_1 | \mathbf{x} = \beta_1$. If we use

a second-order polynomial, then interactions between x_1 and the other $k - 1$ inputs play a role and the marginal effect of x_1 is not constant. Instead of this local SA we may perform global SA: how does the simulation output change, as one or more simulation inputs change over the whole area of interest $[0, 1]^k$? Moreover, "the" output may have a distribution (instead of a single value) if the deterministic simulation has one or more inputs that are uncertain, so we assume a prespecified distribution for these uncertain inputs (as in LHS; see Section 3). Now we discuss *functional analysis of variance* (FANOVA), which uses variance-based indexes originally proposed by the Russian mathematician Sobol.

FANOVA decomposes σ_w^2 into fractions that refer to sets of inputs:

$$\sigma_w^2 = \sum_{j=1}^k \sigma_j^2 + \sum_{j < j'}^k \sigma_{j;j'}^2 + \dots + \sigma_{1;\dots;k}^2 \quad (18)$$

with the main-effect variance $\sigma_j^2 = \text{Var}[E(w|x_j)]$, the two-factor interaction variance $\sigma_{j;j'}^2 = \text{Var}[E(w|x_j, x_{j'})] - \text{Var}[E(w|x_j)] - \text{Var}[E(w|x_{j'})]$, etc. This σ_j^2 gives the *first-order sensitivity index* or the *main effect index* $\zeta_j = \sigma_j^2/\sigma_w^2$, which quantifies the effect of varying x_j alone—averaged over the variations in all the other $k - 1$ inputs—where the denominator σ_w^2 standardizes ζ_j to provide a fractional contribution. Altogether the sum of the 2^{k-1} indexes is 1. In practice, we assume that only the ζ_j —and possibly the $\zeta_{j;j'}$ —are important, and that they sum up to a fraction "close enough" to 1. To *estimate* these measures, we may use LHS and replace the simulation model by a Kriging metamodel; see Kleijnen (2015, pp. 216—218). In practice, FANOVA may show that (say) 70% of σ_w^2 is caused by x_1 , 20% by x_2 , and 10% by the interaction between x_1 and x_2 . Software for FANOVA is included in Open TURNS, discussed in Baudin et al. (2017).

6 Optimization

There are many methods for the optimization of a simulated system; see the recent references in Kleijnen (2015, p. 242). In this chapter, however, we focus on *efficient global optimization* (EGO) in Section 6.1, *Kriging and integer mathematical programming* (KIMP) in Section 6.2, and *robust optimization* (RO) in Section 6.3.

6.1 Efficient global optimization (EGO)

Jones et al. (1998) develops EGO, which uses Kriging. EGO is a *sequential* method that balances *local* and *global* search; i.e., EGO balances *exploitation* and *exploration*. We present only the *basic* EGO-variant for *deterministic* simulation. There are many more variants, for deterministic simulation and random simulation, constrained optimization, multi-objective optimization includ-

ing Pareto frontiers, RO, the "excursion set" or "admissible set", estimation of a quantile, and Bayesian approaches; see Kleijnen (2015, p. 267–269).

In basic EGO we first find the best output observed (simulated) so far. Next we select a new combination $\mathbf{x}_0$, considering both $\hat{y}(\mathbf{x}_0)$ and $s^2[\hat{y}(\mathbf{x})]$; e.g., if $\mathbf{x}_0$ and $\mathbf{x}'_0$ have $\hat{y}(\mathbf{x}_0) \approx \hat{y}(\mathbf{x}'_0)$ and $s^2[\hat{y}(\mathbf{x}_0)] > s^2[\hat{y}(\mathbf{x}'_0)]$, then we explore $\mathbf{x}_0$ because $\mathbf{x}_0$ has a higher probability of *improvement* so it may give a smaller w . We know that $s^2[\hat{y}(\mathbf{x}_0)]$ increases as $\mathbf{x}_0$ lies farther away from $\mathbf{x}_i$; see (6). Actually, we estimate the maximum of the *expected improvement* (EI), which is reached if either $\hat{y}(\mathbf{x}_0)$ is much smaller than $w_{\min} = \min_{1 \leq i \leq n} w(\mathbf{x}_i)$ or $s^2[\hat{y}(\mathbf{x}_0)]$ is relatively large so $\hat{y}(\mathbf{x}_0)$ is relatively uncertain. More precisely, we start with a pilot sample—typically selected through LHS—which results in $(\mathbf{X}, \mathbf{w})$. Next we find $w_{\min}$. We also fit a Kriging metamodel $\hat{y}(\mathbf{x})$. Together, this gives $\text{EI}(\mathbf{x}) = \text{E}[\max(w_{\min} - \hat{y}(\mathbf{x}), 0)]$. Jones et al. (1998) derives the EI estimator

$$\widehat{\text{EI}}(\mathbf{x}) = (w_{\min} - \hat{y}(\mathbf{x})) \Phi \left(\frac{w_{\min} - \hat{y}(\mathbf{x})}{s[\hat{y}(\mathbf{x}_0)]} \right) + s[\hat{y}(\mathbf{x}_0)] \phi \left(\frac{w_{\min} - \hat{y}(\mathbf{x})}{s[\hat{y}(\mathbf{x}_0)]} \right) \quad (19)$$

where Φ and ϕ denote the cumulative distribution function (CDF) and the PDF of $N(0, 1)$. Using (19), we estimate $\mathbf{x}_{opt}$, which denotes the $\mathbf{x}$ that maximizes $\widehat{\text{EI}}(\mathbf{x})$. To find this $\hat{\mathbf{x}}_{opt}$, we may use a (relatively large) set of candidate points selected through LHS (say) $\mathbf{X}_{cand}$; we do not simulate these candidates, but we find the candidate with the highest $\widehat{\text{EI}}(\mathbf{x})$ with $\mathbf{x} \in \mathbf{X}_{cand}$. Next we use this candidate $\hat{\mathbf{x}}_{opt}$ as the simulation input combination, and obtain $w(\hat{\mathbf{x}}_{opt})$. Then we fit a new Kriging model to the augmented I/O data. We update n , and return to (19)—until we satisfy a stopping criterion; e.g., $\widehat{\text{EI}}(\hat{\mathbf{x}}_{opt})$ is “close” to 0 or the computer budget is exhausted.

6.2 Kriging and integer mathematical programming (KIMP)

Kleijnen et al. (2010) develops an alternative for the various EGO variants that we mentioned in Section 6.1 This alternative considers the following *constrained optimization* problem, which has the goal (objective) function $\text{E}(w^{(1)}|\mathbf{z})$ and the $v - 1$ constrained random variables $\text{E}(w^{(l)}|\mathbf{z})$ ($l = 2, \dots, v$) where $\mathbf{z}$ denotes the vector with the k original (non-standardized) inputs z_j , which must satisfy s constraints f_g (e.g., budget constraints) and z_j must be a non-negative integer, and z_j must satisfy the box constraints formed by the lower value L_j and the upper value H_j :

$$\begin{aligned} \min_{\mathbf{z}} \quad & \text{E}(w^{(1)}|\mathbf{z}) \\ \text{E}(w^{(l)}|\mathbf{z}) \geq \quad & c_l \quad (l = 2, \dots, v) \\ f_g(\mathbf{z}) \geq \quad & c_g \quad (g = 1, \dots, s) \\ z_j \in \mathbf{N} \quad & (j = 1, \dots, k) \\ L_j \leq z_j \leq \quad & H_j \quad \text{with } j = 1, \dots, k. \end{aligned} \quad (20)$$

This problem is solved through KIMP, which combines (i) *sequentialized* designs (like EGO does); (ii) *Kriging* to obtain explicit functions $\hat{y}^{(l)}(\mathbf{z})$ for $\text{E}(w^{(l)}|\mathbf{z})$

with $l' = 1, \dots, v$ (like EGO does); (iii) *integer nonlinear programming* (INLP) to estimate the optimal solution from these explicit Kriging models (without using EI, used by EGO). Experiments with KIMP and *OptQuest*—the popular heuristic for problems like (20)—suggest that KIMP requires fewer simulated combinations and gives better estimated optima. KIMP is also summarized in Kleijnen (2015, pp. 269-272).

6.3 Robust optimization (RO)

Dellino et al. (2012) combines Krige’s model and Taguchi’s world view. Taguchi designed robust engineering products such as cars (at Toyota), emphasizing that in practice some inputs are under complete control of the engineers (e.g., the car’s design), whereas other inputs are not (the car’s driver, and the roads). He therefore distinguished between (i) *controllable* or decision variables, and (ii) *noncontrollable* or environmental noisy (or random) factors. Consequently, the estimated optimum (see the preceding two subsections) may turn out to be inferior because this optimum ignores *uncertainties* in some of the simulation inputs. Therefore we may proceed as follows.

To simplify our discussion, we sort the k simulation inputs such that the first k_C inputs are controllable, and the next k_{NC} inputs are noncontrollable. We let $\mathbf{z}_C$ and $\mathbf{z}_{NC}$ denote the vector with the k_C controllable and the k_{NC} noncontrollable original (nonstandardized) inputs $\mathbf{z}$. Taguchi assumed a single output (say) w , and focused on its mean $E(w)$ and its variance; obviously, this variance is caused by $\mathbf{z}_{NC}$, so $\text{Var}(w|\mathbf{z}_C) > 0$. Taguchi combined $E(w)$ and $\text{Var}(w|\mathbf{z}_C)$ into a scalar loss function such as the *signal-to-noise* or *mean-to-variance* ratio $E(w)/\text{Var}(w|\mathbf{z}_C)$; see the detailed discussion of Taguchi’s approach in Myers et al. (2009, pp. 486–488), which is the classic textbook on *response surface methodology* (RSM).

Taguchi’s approach is successful in production engineering, but statisticians criticize its statistical methods. We add that—compared with real-life experiments (discussed in Myers et al. (2009))—simulation experiments have more inputs, more input values, and more input combinations. Myers et al. (2009, pp. 502–506) combines Taguchi’s worldview with the statisticians’ RSM.

Dellino et al. (2012), however, uses $E(w)$ and $\sigma(w|\mathbf{z}_C)$ *separately*; obviously, $\sigma(w|\mathbf{z}_C)$ has the same scale as $E(w)$ has. *Mathematical programming* (MP) can then be used to solve the *constrained optimization*: $\min_{\mathbf{z}_C} E(w|\mathbf{z}_C)$ such that $\sigma(w|\mathbf{z}_C) \leq c_\sigma$ where c_σ is a prespecified upper threshold for σ . Constrained optimization is also discussed in Myers et al. (2009, p. 492). Whereas Myers et al. (2009) superimposes contour plots for $E(w|\mathbf{z}_C)$ and $\sigma(w|\mathbf{z}_C)$ to estimate the optimal $\mathbf{z}_C$, Dellino et al. (2012) uses MP.

This MP, however, requires specification of c_σ (threshold for σ). In practice, managers may find it hard to select a specific value for c_σ , so we may try different c_σ values and estimate the corresponding *Pareto-optimal* efficiency frontier. However, this frontier is uncertain because it depends on the estimators of $E(w|\mathbf{z}_C)$ and $\sigma(w|\mathbf{z}_C)$. To estimate the variability of the frontier, *bootstrapping* may be applied. Dellino et al. (2012) is also summarized in Kleijnen (2015, pp.

280–284).

An application of RO using Kriging to estimate the Pareto frontier is Zhang and Xu (2017). Kriging for RO is also used in Chatterjee et al. (2017), comparing this approach with several alternative metamodel types (e.g., neural networks).

Yanikoğlu et al. (2016) presents an approach that follows RO as developed in MP. RO in MP was originally developed by the Israeli mathematician Ben-Tal et al., and uses concepts such as "uncertainty sets", "robust counterparts", and "adjustable decision rules". Yanikoğlu et al.'s approach uses only experimental data combined with "phi-divergence" uncertainty sets, so this approach does not need a known distribution for the environmental inputs. This approach is applied to low-order polynomial metamodels, but may be extended to Kriging metamodels (without introducing additional complexity to the problem formulation). Stinstra and Den Hertog (2008) details RO for both linear regression and Kriging metamodels.

7 Conclusions and future research

We provided an overview of a specific metamodel type; namely, Kriging (or GP) models. Kriging has already become popular in geostatistics and machine learning, and is also gaining popularity in the analysis of simulation experiments. However, many issues remain in Kriging; e.g., should we use a low-order polynomial trend or a simple constant mean; should we select a Gaussian or a Matérn correlation function; how should we estimate the intrinsic variance for new points? Kriging in random simulation needs further research on sequential designs for the selection of additional points and replications. Kriging is used in EGO for optimization in deterministic simulation without constraints; constrained optimization is currently investigated by many researchers. Robust optimization has just started in simulation, whereas it is a hot topic in MP. Application of various types of Kriging for SA and optimization remains challenging.

Future research may try to solve the *curse of dimensionality*; currently, k is limited to (say) 20 inputs. To solve this problem, we may precede Kriging by *factor screening*; several screening methods are presented in Chapter 4 of Kleijnen (2015).

Another topic of future research is *big data*, in which n (the number of I/O points) is so high that the computation of the inverse of the estimated covariance matrix is impossible. Various solutions are discussed in Kleijnen and Van Beers (2017).

References

- Ankenman, B., B. Nelson, and J. Staum (2010), Stochastic kriging for simulation metamodeling. *Operations Research*, 58, no. 2, pp. 371382
- Baudin, M., A. Dutfoy, B. Iooss, and A-L. Popelin (2017), Open TURNS: an industrial software for uncertainty quantification in simulation. *Handbook of*

uncertainty quantification, edited by R. Ghanem, D. Higdon, and H. Owhadi, Springer (preprint: hal-01107849v2)

Benková, E., R. Harman, and W.G. Müller (2015), Privacy sets for constrained space-filling. *arXiv:1510.05284*

Chatterjee, T., S. Chakraborty, and R. Chowdhury (2017), A critical review of surrogate assisted robust design optimization. *Archives of Computational Methods in Engineering*, in press

Chen, H., J.L. Loepky, J. Sacks, and W.J. Welch (2016), Analysis methods for computer experiments: how to assess and what counts? *Statistical Science*, 31, no. 1, pp. 40-60

Chen X and K-K. Kim (2013) Building metamodels for quantile-based measures using sectioning. *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S-H. Kim, A. Tolk, R. Hill, and M.E. Kuhl, pp. 521-532

Chen X, K. Wang, and F. Yang (2013) Stochastic kriging with qualitative factors. *Proceedings of the 2013 winter simulation conference*, edited by R. Pasupathy, S-H. Kim, A. Tolk, R. Hill, and M.E. Kuhl, pp. 790-801

Chevalier, C., X. Emery, and D. Ginsbourger (2014), Fast update of conditional simulation ensembles. <hal-00984515>

Chilès, J-P and P. Delfiner (1999), *Geostatistics: modeling spatial uncertainty*. Wiley, New York

Cressie, N.A.C. (1993), *Statistics for spatial data; revised edition*. Wiley, New York

Dellino G, J.P.C. Kleijnen, and C. Meloni (2012), Robust optimization in simulation: Taguchi and Krige combined. *INFORMS Journal on Computing*, 24, no. 3, pp. 471-484

Dong, H. and M.K. Nakayama (2017), Quantile estimation with Latin hypercube sampling. *Operations Research*, 65, no. 6, pp. 1678-1695

Erickson, C.B., B.E. Ankenman, S.M. Sanchez (2017), Comparison of Gaussian process modeling software. *European Journal of Operational Research*, accepted

Jones, D.R., M. Schonlau, and W.J. Welch (1998), Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13, pp. 455-492

Kleijnen, J.P.C. (2015), *Design and analysis of simulation experiments; second Edition*. Springer

Kleijnen, J. P. C. and W.C.M. van Beers (2004), Application-driven sequential designs for simulation experiments: Kriging metamodeling. *Journal of Operational Research Society*, 55, no. 9, pp. 876-883

Kleijnen, J. P. C. and W.C.M. van Beers (2017), Prediction for big data through Kriging. CentER Discussion Paper; Center for Economic Research (CentER), Tilburg University, forthcoming

Kleijnen, J. P. C., W.C.M. van Beers, and I. van Nieuwenhuyse I (2010) Constrained optimization in simulation: a novel approach. *European Journal of Operational Research*, 202, pp. 164-174

Le Guiban, K., A. Rimmel, M-A. Weisser, and J. Tomasik (2017), The first approximation algorithm for the maximin Latin hypercube design problem. *Operations Research*, in press

- Loeppky, J.L., J. Sacks, and W.J. Welch (2009), Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, 51, pp. 366–376
- Mehdad, E. and J.P.C. Kleijnen (2017a), Efficient global optimization for black-box simulation via sequential intrinsic Kriging. *Journal of the Operational Research Society*, accepted Nov.2017
- Mehdad, E. and J.P.C. Kleijnen (2017b), Stochastic intrinsic Kriging for simulation metamodeling. Version 29 October 2017
- Myers, R.H., D.C. Montgomery, and C.M. Anderson-Cook (2009), *Response surface methodology: process and product optimization using designed experiments; third edition*. Wiley, New York
- Minnitt, R.C.A, and W. Assibey-Bonsu (2013) Professor D.G. Krige FRSSAf. *Transactions of the Royal Society of South Africa*, 68:3, 199-202, DOI: 10.1080/0035919X.2013.826748
- Mukhopadhyay, T., S Chakraborty, S Dey, S Adhikari, and R. Chowdhury (2016), A critical assessment of Kriging model variants for high-fidelity uncertainty quantification in dynamics of composite shells. *Archives of Computational Methods in Engineering*, pp. 1–24
- Rasmussen, C. and C. Williams (2006), *Gaussian processes for machine learning*. MIT Press, Cambridge, Massachusetts
- Sacks, J., W.J. Welch, T.J. Mitchell and H.P. Wynn (1989), Design and analysis of computer experiments (includes Comments and Rejoinder). *Statistical Science*, 4, no. 4, pp. 409–435
- Salemi P., J. Staum, and B.L. Nelson (2013), Generalized integrated Brownian fields for simulation metamodeling. *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S-H. Kim, A. Tolk, R. Hill, and M.E. Kuhl, pp. 543–554
- Santner, T.J., B.J. Williams, and W.I. Notz (2003), *The design and analysis of computer experiments*. Springer, New York
- Stinstra, E. and D. den Hertog (2008), Robust optimization using computer experiments. *European Journal of Operational Research*, 191, pp. 816–837
- Wackernagel, H. (2003), *Multivariate geostatistics: an introduction with applications; third edition*. Springer, Berlin,
- Yanikoğlu, I., den Hertog, D. & Kleijnen, J. P. C. (2016), Robust dual-response optimization. *IIE Transactions: Industrial Engineering Research and Development*, 48, no. 3, pp. 298–312
- Zhang, W. and W. Xu (2017), Simulation-based robust optimization for the schedule of single-direction bus transit route: the design of experiment. *Transportation Research Part E*, 106, pp. 203–230

Acknowledgement 2 *I thank Dick den Hertog (Tilburg University, the Netherlands) and Ihsan Yanikoğlu (Özyeğin University, Turkey) for their comments on a previous version of this chapter.*