No. 2015-035

REGRESSION AND KRIGING METAMODELS WITH THEIR EXPERIMENTAL DESIGNS IN SIMULATION: REVIEW

By

Jack P.C. Kleijnen

6 July 2015

ISSN 0924-7815
ISSN 2213-9532

Regression and Kriging metamodels with their experimental designs in simulation: review

Jack P.C. Kleijnen
Tilburg University, Postbox 90153, Tilburg, Netherlands
kleijnen@tilburguniversity.edu

July 6, 2015

Abstract

This article reviews the design and analysis of simulation experiments. It focusses on analysis via either low-order polynomial regression or Kriging (also known as Gaussian process) metamodels. The type of metamodel determines the design of the experiment, which determines the input combinations of the simulation experiment. For example, a first-order polynomial metamodel requires a "resolution-III" design, whereas Kriging may use Latin hypercube sampling. Polynomials of first or second order require resolution III, IV, V, or "central composite" designs. Before applying either regression or Kriging, sequential bifurcation may be applied to screen a great many inputs. Optimization of the simulated system may use either a sequence of low-order polynomials known as response surface methodology (RSM) or Kriging models fitted through sequential designs including efficient global optimization (EGO). The review includes robust optimization, which accounts for uncertain simulation inputs.

Keywords: robustness and sensitivity, simulation, metamodel, design, regression, Kriging

JEL: C0, C1, C9, C15, C44

1 Introduction

In this article we review the design and analysis of simulation experiments. This *design* depends on the *metamodel*—also called surrogate model or emulator—that we use to analyze f_{sim}, which denotes the input/output (I/O) function implicitly defined by the underlying simulation model. For example, if we assume that a first-order polynomial is an "adequate" or "valid" metamodel, then changing one factor (simulation input) at a time enables unbiased estimators of the first-order or "main" effects of these factors. However, these estimators do not have minimum variances; a fractional factorial design with two values or "levels" per factor—denoted as a 2^{k-p} design—gives estimators with minimum variances—as we shall see in Section 2. There are several types of metamodels, but we focus on the most popular types; namely, *low-order polynomial regression* and *Kriging*—or Gaussian process (GP)—metamodels.

We focus on simulation that has as its goals *sensitivity analysis* (SA) and *optimization* of the underlying real system. For such SA and optimization there are many methods, but we focus on methods that use either low-order polynomials or Kriging metamodels and their corresponding designs. SA is an ambiguous term; e.g., SA may be either global or local, but we focus on global SA or "what if" analysis. Notice that many SA methods are reviewed in Borgonovo and Plischke (2015).

Because Kriging metamodels and simulation-based optimization are very active fields of research, we update two previous reviews; namely, Kleijnen (2005, 2009). We base our update on Kleijnen (2015), which includes many website addresses for software and hundreds of additional references. We assume that the readers have a basic knowledge of simulation and mathematical statistics.

We organize this article as follows. Section 2 summarizes classic linear regression metamodels—including polynomials—and their designs. Section 3 presents solutions when the classic assumptions do not hold in practice. Section 4 explains how sequential bifurcation (SB) can screen hundreds of inputs of realistic simulation models; this section uses the two preceding sections. Section 5 summarizes Kriging and its designs. Section 6 explains simulation optimization through either low-order polynomials or Kriging; this section includes robust optimization. This article ends with 46 carefully selected references, including 30 references published in 2010 or more recently.

2 Classic linear regression metamodels and their designs

Classic linear regression models have the following form:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}+\mathbf{e} \tag{1}$$

with $\mathbf{y}$ the n-dimensional vector with the dependent variable and n the number of simulated input combinations; $\mathbf{X}$ the $n \times q$ matrix of independent regression variables with $\mathbf{x}_{i;j}$ the value of the independent variable j in combination i (i = 1, ..., n; j = 1, ..., q); $\boldsymbol{\beta}$ the q-dimensional vector with regression parameters; and $\mathbf{e}$ the n-dimensional vector with the residuals in the n combinations. In this review we focus on a special case of (1); namely, a second-order polynomial with k simulation inputs:

$$y = \beta_0 + \sum_{j=1}^{k}\beta_j x_j + \sum_{j=1}^{k}\sum_{j'\geq j}^{k}\beta_{j;j'} x_j x_{j'} + e \tag{2}$$

with the intercept β_0, the k first-order effects β_j (j= 1, ..., k), the $k(k-1)2$ two-factor interactions (cross-products) $\beta_{j;j'}$ (j'> j), and the k purely quadratic effects $\beta_{j;j}$. Obviously, (2) implies $q = (k+1)(k+2)/2$. Furthermore, interaction means that the effect of an input depends on the values of one or more other inputs. A purely quadratic effect means that effect of the input is not constant, but diminishes or increases. This metamodel is nonlinear in $\mathbf{x}$ but linear in β, as in (1).

In this review, we assume that interactions among three or more inputs are unimportant. Our reason is that such interactions are hard to interpret, and are often unimportant in practice. Of course, we should check this assumption; i.e., we should "validate" the estimated metamodel, as we shall see below.

To estimate $\boldsymbol{\beta}$ in (1), we apply the *least squares* (LS) criterion and obtain

$$\hat{\beta} \ = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{w} \tag{3}$$

where $\mathbf{w}$ denotes the n-dimensional vector with the simulation outputs w_i that correspond with $\mathbf{x}_{i;j}$. Obviously, $\hat{\beta}$ exists only if $\mathbf{X}$ is not *collinear*; e.g. $\mathbf{X}$ is collinear if two simulation inputs change simultaneously by the same amount. To select a specific $\mathbf{X}$, we may decide to minimize $\mathrm{Var}(\widehat{\beta_j})$. To derive $\mathrm{Var}(\widehat{\beta_j})$, classic regression analysis assumes that $\mathbf{e}$ in (1) is *white noise*: $\mathbf{e}$ is normally, independently, and identically distributed (NIID) with zero mean and a constant variance σ_e^2. If the metamodel (1) is valid, then obviously $\sigma_e^2 = \sigma_w^2$. Altogether, $\hat{\beta}$ has the following covariance matrix:

$$\mathbf{\Sigma}_{\hat{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\sigma_w^2. \tag{4}$$

The unknown parameter σ_w^2 in this equation can be estimated through

$$\mathrm{MSR} = \frac{(\widehat{\mathbf{y}} - \mathbf{w})'(\widehat{\mathbf{y}} - \mathbf{w})}{n - q} \tag{5}$$

where $\widehat{\mathbf{y}} = \mathbf{X}\hat{\beta}'$; obviously, (5) assumes $n - q > 0$. This MSR gives $\widehat{\mathbf{\Sigma}}_{\hat{\beta}}$. Next we may derive *confidence intervals* (CIs) and tests for the individual elements of $\hat{\beta}$ through the variances given by the main diagonal of $\widehat{\mathbf{\Sigma}}_{\hat{\beta}}$ and the Student statistic t_v with $v = n - q$.

It can be proven that the variances of the $\hat{\beta}$-elements are minimal if $\mathbf{X}$ is orthogonal. Often, the theory on *design of experiments* (DOE) assumes that we *standardize* (scale) the simulation inputs such that -1 $\leq x_{i;j} \leq$ 1. If each input has only two values in the whole experiment with its n input combinations, then standardization implies the following linear transformation where z_j denotes the quantitative input j measured on the original scale, l_j denotes the lower value of z_j in the experiment, u_j the upper value, $\overline{z}_j$ the average value of input j in a *balanced* experiment in which each input is observed at its lower value in $n/2$ combinations:

$$x_{i;j} = \frac{z_{i;j} - \overline{z}_j}{(u_j - l_j)/2} \ (i = 1, ..., n; \ j = 1, ..., k). \tag{6}$$

Consequently, an *orthogonal* standardized $\mathbf{X}$ implies $\mathbf{X}'\mathbf{X} = n\mathbf{I}$, so $\mathbf{\Sigma}_{\hat{\beta}}$ in (4) becomes

$$\mathbf{\Sigma}_{\hat{\beta}} = (n\mathbf{I})^{-1}\sigma_w^2 = \mathbf{I}\frac{\sigma_w^2}{n}. \tag{7}$$

Input combination	**1**	**2**	**3**	**4** = **1.2**	**5** = **1.3**	**6** = **2.3**	**7** = **1.2.3**
1	-	-	-	+	+	+	-
2	+	-	-	-	-	+	+
3	-	+	-	-	+	-	+
4	+	+	-	+	-	-	-
5	-	-	+	+	-	-	+
6	+	-	+	-	+	-	-
7	-	+	+	-	-	+	-
8	+	+	+	+	+	+	+

Table 1: A one-sixteenth fractional factorial design for seven inputs

This implies that the q estimators have the same variance σ_w^2/n, and are statistically independent. Because these estimators have the same estimated variances, we can rank these estimates in order of importance, using either these estimates themselves or their t-values. Because all q estimators are independent, the "full" regression model with q effects and the "reduced" model with nonsignificant effects eliminated have identical values for those estimated effects that occur in both models. If $\mathbf{X}$ were not orthogonal, then this so-called "backwards elimination" of nonsignificant effects would change the remaining estimates.

The selection of $\mathbf{X}$ that gives a "good" $\mathbf{\Sigma}_{\hat{\beta}}$ is the goal of DOE, discuss next. In this discussion we initially assume that no input combination is replicated. We discuss the following special cases of (2): (i) all second-order effects $\beta_{j;j'}$ are zero (so the metamodel becomes a first-order polynomial); (ii) all purely quadratic effects $\beta_{j;j}$ are zero; we discuss (a) we estimate the first-order effects β_j unbiased by the two-factor interactions $\beta_{j;j'}$ with $j \neq j'$, and (b) we obtain unbiased estimators of both β_j and $\beta_{j;j'}$. These cases require designs of different *resolution* (denoted by R); e.g., R-III designs for first-order polynomials.

2.1 R-III designs for first-order polynomials

By definition, a R-III design gives unbiased estimators of β_j ($j = 1, ..., k$), assuming a first-order polynomial is a valid metamodel. These designs are also known as *Plackett-Burman* designs. A subclass of these designs are *fractional factorial two-level* 2_{III}^{k-p} designs with positive integer p such that $p < k$ and 2^{k-p} $\geq 1 + k$. Plackett-Burman designs have n equal to a multiple of four and at least equal to $k + 1$; e.g., for $8 \leq k \leq 11$ implies $n = 12$.

An example is the 2_{III}^{7-4} design in Table 1. The symbol - stands for -1, and + for 1. The symbol **1** stands for $(x_{1;1}, ..., x_{n;1})'$ where in this example $n = 8$; likewise, **2** and **3** correspond with inputs 2 and 3. The symbol **4** = **1.2** stands for $x_{i;4} = x_{i;1}x_{i;2}$ with $i = 1, ..., n$, so the first element ($i = 1$) in this column is $x_{1;4} = x_{1;1}x_{1;2} = (-1)(-1) = +1$. The DOE literature calls "**4** = **1.2**" a *design generator.* A 2^{k-p} design requires p generators; e.g., Table 1 specifies these generators in the last four columns. Obviously, this example gives a balanced design and an orthogonal $\mathbf{X}$.

If $4 \leq k \leq 6$, then we still use Table 1, but we ignore columns; e.g., if $k= 6$ we may ignore the last column. If $k = 7$, then Table 1 implies a *saturated* design: $n = q$ (obviously, a first-order polynomial means $q = 1 + k$). A saturated design implies $\upsilon = n - q = 0$ in (5). To solve this problem, we may add one or more combinations to Table 1; e.g., either combinations from the *full factorial* 2^7 design excluding the combinations in Table 1 or the combination at the *center* of the experimental area where $x_j = 0$ if z_j is quantitative and x_j is randomly selected as -1 or 1 if z_j is qualitative.

Kleijnen (2015) also details a 2^{15-11} design, including a simple algorithm for constructing this design. Algorithms for the construction of 2_{III}^{k-p} designs with high k values are presented in Ryan and Bulutoglu (2010) and Shrivastava and Ding (2010). We do not detail 2_{III}^{k-p} designs with such high k values, because in practice such values are rare. In Section 4 we shall discuss so-called screening designs that are more efficient than 2_{III}^{k-p} designs.

There are also Plackett-Burman designs for $12 \leq n \leq 96$ with n not a power of two but a multiple of four; e.g., Kleijnen (2015) gives such a design with $n = 12$ and $k = 11$. Montgomery (2009, p. 326) and Myers et al. (2009, pp. 165) tabulate such designs for $12 \leq n \leq 36$. These designs are balanced and orthogonal, like 2^{k-p}designs are.

2.2 R-IV designs

By definition, a R-IV design gives unbiased estimators of the β_j (j = 1, ..., k) in a first-order polynomial—even if *two-factor interactions* are nonzero, but all higher-order effects are zero. Remembering that the number of two-factor interactions in (2) is $k(k-1)/2$, we obtain $q = 1 + k + k(k-1)2 = 1 + k(k+1)/2$. Furthermore, $\mathbf{X}$ follows from the $n \times k$ design matrix $\mathbf{D} = (d_{i;j})$:

$$\mathbf{X} = (\mathbf{x}_i) = (1, d_{i;1}, \ldots, d_{i;k}, d_{i;1}d_{i;2}, \ldots, d_{i;k-1}d_{i;k}) \; (i = 1, \ldots, n) \tag{8}$$

To construct a R-IV design, we apply the *foldover theorem*, which states that augmenting a R-III design $\mathbf{D}$ with its *mirror* design $-\mathbf{D}$ gives a R-IV design. Obviously, a R-IV design doubles the number of combinations. A R-IV design does not always enable unbiased estimators of the *individual* two-factor interactions. For example, Table 1 gave a 2_{III}^{7-4} design, so the R-IV design has $n = 16$. $\mathbf{X}$ follows from (8) with $k = 7$, so $\mathbf{X}$ has $q = 1 + 7(7 + 1)/2 = 29$ columns; because $n < q$ we know that $\mathbf{X}$ is collinear. Hence, it is impossible to apply (3) and compute the LS estimators of the 29 individual regression parameters. Kleijnen (2015) explains that we can estimate sums of these interactions; e.g., the 2_{IV}^{8-4} design with the generators $\mathbf{5} = \mathbf{1.3.4}$, $\mathbf{6} = \mathbf{2.3.4}$, $\mathbf{7} = \mathbf{1.2.3}$, and $\mathbf{8} = \mathbf{1.2.4}$ gives estimators of the $8(8-1)/2 = 28$ two-factor interactions that are *aliased* or *confounded* in seven groups of size four. In general, let us assume that a valid linear regression metamodel is

$$y = \mathbf{X}_1\beta_1 + \mathbf{X}_2\beta_2 + e. \tag{9}$$

An example is an $\mathbf{X}_1$ corresponding with the intercept and the first-order effects collected in β_1, and an $\mathbf{X}_2$ corresponding with the two-factor interactions

β_2. Suppose that we start with a tentative simple metamodel without these interactions, so

$$\widehat{\beta}_1 = (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'\mathbf{w}. \tag{10}$$

Assuming that a valid metamodel is (9) gives $E(w) = E(y)$. Combining (10) and (9) then gives

$$\begin{aligned} &E(\widehat{\beta}_1) = (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'E(\mathbf{w}) =(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'(\mathbf{X}_1\beta_1 + \mathbf{X}_2\beta_2) \\ &= \beta_1 + (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'\mathbf{X}_2\beta_2. \end{aligned} \tag{11}$$

This equation includes the *alias matrix* $\mathbf{A} = (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'\mathbf{X}_2$. Equation (11) implies an unbiased estimator of β_1 if either $\beta_2 = \mathbf{0}$ or $\mathbf{X}_1'\mathbf{X}_2 = \mathbf{0}$. Indeed, R-III designs assume that $\beta_2 = \mathbf{0}$ where β_2 consists of the two-factor interactions, and R-IV designs ensure that $\mathbf{X}_1'\mathbf{X}_2 = \mathbf{0}$ (the two-factor interaction columns are orthogonal to the columns for the first-order effects and the intercept).

2.3 Resolution-V designs for two-factor interactions

By definition, a R-V design enables LS estimation of the first-order effects, the two-factor interactions, and the intercept; higher-order effects are assumed to be zero. Obviously, this implies $q = 1 + k + k(k-1)/2 = (k^2 + k)/2 + 1$. The DOE literature gives tables with generators for 2_V^{k-p} designs; e.g., the 2_V^{8-2}design with $\mathbf{7} = \mathbf{1.2.3.4}$ and $\mathbf{8} = \mathbf{1.2.5.6}$. Unfortunately, 2_V^{k-p} designs are not saturated at all; e.g., the 2_V^{8-2}design implies $n = 64 \gg q = 37$. So-called Rechtschaffner designs include saturated R-V designs, but they are not orthogonal; the statistical properties of these designs are further investigated in Qu (2007).

2.4 CCDs for second-degree polynomials

By definition, a CCD or *central composite design* enables LS estimation of all the effects in a second-order polynomial, assuming all higher-order effects are zero. A CCD consists of the following combinations: (i) a R-V design (see Section 2.3); (ii) the *central* combination $\mathbf{0}_k'$, which denotes a row-vector with k zeroes; (iii) the $2k$ *axial* combinations—which form a *star design*—where the "positive" axial combination for input j is $x_j = c$ while all other (k - 1) inputs are fixed at the center so $x_{j'} = 0$ with $j' \neq j$, and the "negative" axial combination for input j is $x_j = -c$ and $x_{j'} = 0$. Notice that $c \neq 1$ implies a CCD with five values per input, whereas $c = 1$ implies a CCD with only three values per input. The usual choice is $c \neq 1$; the optimal choice of c assumes white noise, which does not hold in practice; see the next section.

Obviously, a CCD does not give an orthogonal $\mathbf{X}$, and is not saturated. CCDs are popular in *response surface methodology* (RSM), which we shall discuss in Section 6.1 (Section 6 discusses simulation-optimization). For further discussion of CCDs we refer to Khuri and Mukhopadhyay (2010), Kleijnen (2015), and Myers et al. (2009, pp. 296–317).

3 Classic assumptions versus simulation practice

In the preceding section we detailed the *classic* assumptions of linear regression metamodels and their concomitant designs; these assumptions stipulate a single type of simulation output (univariate output) and white noise. In practice, however, these assumptions usually do not hold. Indeed, a practical simulation model may give a *multivariate* output. White noise implies (i) *normally* distributed output; (ii) no *common random numbers* (CRN); (iii) *homogeneous* variances of the simulation output; (iv) a *valid* metamodel. In this section, we try to answer the following questions: (a) How realistic are these classic assumptions? (b) How can we test these assumptions? (c) If an assumption is violated, can we then transform the simulation's I/O data such that the assumption holds for the transformed data? (d) If we cannot find such a transformation, which statistical methods can we then apply?

3.1 Multivariate output

Examples of multivariate output are inventory simulations with two outputs; namely, (i) the sum of the holding costs and the ordering costs; (ii) the service (or fill) rate. Analogous to (1), we now assume that we use r univariate linear regression metamodels:

$$\mathbf{y}^{(l)} = \mathbf{X}^{(l)}\beta^{(l)}+\mathbf{e}^{(l)} \text{ with } l = 1, \ldots r \tag{12}$$

where $\mathbf{y}^{(l)}$ is the n-dimensional vector with the dependent variable corresponding with simulation output type l; n is the number of simulated input combinations; $\mathbf{X}^{(l)} = (\mathbf{x}_{i;j}^{(l)})$ is the $n \times q_l$ matrix of independent regression variables with $\mathbf{x}_{i;j}^{(l)}$ denoting the value of independent variable j in combination i for metamodel l ($i = 1, ..., n$; $j = 1, ..., q_l$); $\beta^{(l)} = (\beta_1^{(l)}, \ldots, \beta_{q_l}^{(l)})'$ is the vector with the q_l regression parameters for metamodel l; $\mathbf{e}^{(l)}$ is the n-dimensional vector with the residuals of metamodel l, in the n combinations. In our review, we assume that all the r fitted regression metamodels are polynomials of the same order (e.g.,second-order), so $\mathbf{X}^{(l)} = \mathbf{X}$ and $q_l = q$. The literature calls the metamodel a *multiple* regression model if $q > 1$ and the metamodel has an intercept, and calls it a *multivariate* regression model if $r > 1$.

The $\mathbf{e}^{(l)}$ have the following two *properties*: (i) Their variances may vary with l; e.g., the estimated inventory costs and service percentages have very different variances. (ii) The residuals $e^{(l)}$ and $e^{(l')}$ are not independent for a given input combination i, because they are (different) transformations of the same pseudorandom number (PRN) stream. Consequently, it might seem that we need to replace classic *ordinary* LS (OLS) by—rather complicated—*generalized LS* (GLS); see Khuri and Mukhopadhyay (2010). Fortunately, if $\mathbf{X}^{(l)} = \mathbf{X}$, then GLS reduces to OLS computed per output; see Markiewicz and Szczepańska (2007). Consequently, the *best linear unbiased estimator* (BLUE) of $\beta^{(l)}$ is

$$\widehat{\beta}^{(l)} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{w}^{(l)} \; (l = 1, \ldots, r). \tag{13}$$

Using this equation, we can easily obtain CIs and tests for $\widehat{\beta}^{(l)}$; i.e., we may use the classic formulas presented in the preceding section. There seem to be no *general* designs for multivariate output; see Khuri and Mukhopadhyay (2010).

3.2 Nonnormal output

In simulation, the normality assumption often holds *asymptotically*; i.e., if the simulation run is long, then the sample average of those autocorrelated data gives nearly normal output. Estimated quantiles, however, may be very nonnormal, especially if they are rather extreme; an example is the 99% quantile. The t-statistic is known to be quite insensitive to nonnormality (whereas the F-statistic is not). Whether the actual simulation run is long enough to make the normality assumption hold, is always hard to know. Therefore it seems good practice to test whether the normality assumption holds, as follows.

To test whether a set of observations has a Gaussian *probability density function* (PDF), we may use various *residual plots* and *goodness-of-fit statistics*; e.g., the chi-square, Kolmogorov-Smirnoff, Anderson-Darling, and Shapiro-Wilk statistics. A basic assumption of these statistics is that the observations are IID. We may therefore obtain "many" (say, 100) replications for a specific input combination (e.g., the base scenario) if the simulation is not computationally expensive. However, if a single simulation run takes relatively much computer time, then we can obtain only "a few" (between 2 and 10) replications, so the plots are too rough and the statistical tests lack power.

Actually, the white-noise assumption concerns the metamodel's *residuals* e, not the simulation model's outputs w. If we assume that there are $m_i \geq 1$ replications for combination i (i = 1, ..., n), then $\overline{w}_i = \sum_{r=1}^{m_i} w_{i;j}/m_i$ and $\widehat{\overline{e}}_i = \widehat{y}_i - \overline{w}_i$. For simplicity of presentation, we further assume that m_i is a constant m. If $w_{i;j}$ has a constant variance σ_w^2, then $\overline{w}_i$ also has a constant variance $\sigma_{\overline{w}}^2 = \sigma_w^2/m$. Unfortunately, even if $\overline{w}_i$ has a constant variance $\sigma_{\overline{w}}^2$ and is independent of $\overline{w}_{i'}$ with $i \neq i'$ (no CRN), then $\widehat{\overline{e}}_i$ does not have a constant variance and $\widehat{\overline{e}}_i$ and $\widehat{\overline{e}}_{i'}$ are not independent; i.e., w can prove that

$$\mathbf{\Sigma}_{\widehat{\overline{\mathbf{e}}}} = [\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}']\sigma_{\overline{w}}^2. \tag{14}$$

We may apply *normalizing transformations* to w; e.g., $v = \log(w)$ may be more normally distributed than w. Unfortunately, the metamodel now explains the behavior of the transformed output—not the original output. We also refer to Bekki et al. (2009) and Kleijnen (2015).

Another transformation is *jackknifing*, which is a general statistical method for solving the following two types of problems: (i) constructing CIs for nonnormal responses; (ii) reducing the bias of some estimators. To explain jackknifing, we use the regression problem in (1). Suppose we want CIs for β in case of nonnormal w. For simplicity, we assume $m_i = m > 1$ (i = 1, ..., n). The original estimator $\hat{\beta}$ was given in (3).The jackknife then deletes replication r ($r = 1$, ..., m) for each combination i, and computes

$$\widehat{\beta}_{-r} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\overline{\mathbf{w}}_{-r} \ (r = 1, \ldots, m) \tag{15}$$

with the n-dimensional vector $\overline{\mathbf{w}}_{-r} = (\overline{w}_{i;-r})$ where $\overline{w}_{i;-r}$ denotes the average of the $m-1$ simulation outputs when excluding output r. The m estimators $\widehat{\beta}_{-r}$ in (15) are correlated because they share $m-2$ elements. For ease of presentation, we focus on the scalar β_q (last element of β). Jackknifing then uses the *pseudovalue*

$$J_r = m\widehat{\beta}_q - (m-1)\widehat{\beta}_{q;-r}. \tag{16}$$

In this example, both the original and the jackknifed estimators are unbiased so the pseudovalues also remain unbiased; otherwise, the bias is reduced by the *jackknife point estimator* $\overline{J} = \sum_{r=1}^{m} J_r/m$. Examples of biased estimators are ratio estimators and nonlinear estimators; see Section 5. To compute a CI, jackknifing treats the pseudovalues as if they were NIID. So if $t_{m-1;1-\alpha/2}$ denotes the $1-\alpha/2$ quantile of the t_{υ}-distribution with $\upsilon = m-1$, and $\widehat{\sigma}^2_{\overline{J}} = \sum_{r=1}^{m}(J_r - \overline{J})^2/[m(m-1)]$, then jackknifing gives a two-sided $1-\alpha$ CI forβ_q:

$$P(\overline{J} - t_{m-1;1-\alpha/2}\widehat{\sigma}_{\overline{J}} < \beta_q < \overline{J} + t_{m-1;1-\alpha/2}\widehat{\sigma}_{\overline{J}}) = 1-\alpha. \tag{17}$$

Applications of jackknifing in simulation are numerous; see Gordy and Juneja (2010) and Kleijnen (2015).

Distribution-free bootstrapping or *nonparametric bootstrapping* is another general statistical method that does not assume normality. This bootstrapping may be used to solve two types of problems; namely, problems caused by (i) nonnormal distributions or (ii) nonstandard statistics. Let us return to the example that lead to (17). Now we distinguish between the *original observations* w and the *bootstrapped observations* w^*. Standard bootstrapping assumes that the original observations are IID; indeed, $w_{i;1}$, ..., $w_{i;m}$ are IID because the m replications use nonoverlapping PRN. Distribution-free bootstrapping means *resampling with replacement* from the m original IID observations. We apply this resampling to each of the n combinations. The resulting $w^*_{i;1}$, ..., $w^*_{i;m}$ give $\overline{\mathbf{w}}^*$, which upon substitution into (3) gives

$$\widehat{\beta}^* = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\overline{\mathbf{w}}^*. \tag{18}$$

To reduce sampling variation, we repeat this resampling (say) B times; B is known as the *bootstrap sample size*. A typical value for B is 100 or 1,000. This sample size gives $\widehat{\beta}^*_b$ with $b = 1$, ..., B. The so-called *percentile method* gives

$$P(\widehat{\beta}^*_{q;(B\alpha/2)} < \beta_q < \widehat{\beta}^*_{q;(B[1-\alpha/2])}) = 1-\alpha \tag{19}$$

where $\widehat{\beta}^*_{q;(B\alpha/2)}$ denotes the $\alpha/2$ quantile of the *empirical density function* (EDF) of $\widehat{\beta}^*_q$ obtained through the order statistics denoted by the subscript $(\cdot)$ where—for notational simplicity—we assume that $B\alpha/2$ is integer; $\widehat{\beta}^*_{q;(B[1-\alpha/2])}$ is defined analogously. Another example is given in Turner et al. (2013); namely, a CI for s_w^2 (sample variance of w) if w does not have a Gaussian distribution (so s_w^2 is not χ^2). We shall mention more examples; namely, CIs for a quantile below (26), for R^2 below (28), and for cross-validation statistics below (33).

3.3 Heterogeneous output variances

In practice, $\text{Var}(w_i)$ changes as $\mathbf{x}_i$ changes. In some applications, however, we may hope that this variance heterogeneity is negligible. Unfortunately, $\text{Var}(w_i)$ is unknown so we must estimate it. Given m_i replications, the classic estimator is

$$s_i^2 = \frac{\sum_{r=1}^{m_i}(w_{i;r} - \overline{w}_i)^2}{m_i - 1} \; (i = 1,\ldots, n) \tag{20}$$

with $\overline{w}_i = \sum_{r=1}^{m_i} w_{i;r}/m_i$. This s_i^2 itself has high variance; i.e., if $w_{i;r}$ is normally distributed with $\text{Var}(w_{i;r}) = \sigma_i^2$, then $Var(s_i^2) = 2\sigma_i^4/m_i$. In practice, we need to compare n estimators s_i^2, and we may apply many tests; see Kleijnen (2015).

The logarithmic transformation may be used not only to obtain Gaussian output but also to obtain outputs with constant variances. We, however, prefer accepting heterogeneous variances and adapting our analysis, as follows. If $E(\mathbf{e})$ $= 0$, then $\widehat{\beta}$ is still *unbiased*. However, if $\text{Var}(w_i)$ in not constant, then $\mathbf{\Sigma}_{\widehat{\beta}}$ is no longer given by (4), but by

$$\mathbf{\Sigma}_{\widehat{\beta}} = (\mathbf{X}_N'\mathbf{X}_N)^{-1}\mathbf{X}_N'\mathbf{\Sigma}_{\mathbf{w}}\mathbf{X}_N(\mathbf{X}_N'\mathbf{X}_N)^{-1} \tag{21}$$

where $\mathbf{X}_N$ is the $N \times q$ matrix of independent variables with $N = \sum_{i=1}^{n} m_i$ and $\mathbf{\Sigma}_{\mathbf{w}}$ is the $N \times N$ matrix with the first m_1 elements on its main diagonal all equal to σ_1^2, ..., the last m_n elements on its main diagonal equal to σ_n^2.

In *jackknifed estimated weighted LS* (JEWLS) assuming $m_i = m$ and no CRN, we proceed analogously to (15):

$$\widehat{\widehat{\beta}}_{-r} = (\mathbf{X}\widehat{\mathbf{\Sigma}}^{-1}_{\overline{\mathbf{w}};-r}\mathbf{X})^{-1}\mathbf{X}'\widehat{\mathbf{\Sigma}}^{-1}_{\overline{\mathbf{w}};-r}\overline{\mathbf{w}}_{-r} \; (r = 1,\ldots,m) \tag{22}$$

where $\overline{\mathbf{w}}_{-r}$ is the vector with the n averages of the $m-1$ replications after deleting replication r, and $\widehat{\mathbf{\Sigma}}_{\overline{\mathbf{w}};-r}$ is the diagonal matrix with s_i^2 computed from the same $m-1$ replications. Let $\widehat{\widehat{\beta}}$ denote the estimator that deletes no replication. Because $\widehat{\widehat{\beta}}$ and $\widehat{\widehat{\beta}}_{-r}$ use estimated (random) weights $\widehat{\mathbf{\Sigma}}_{\mathbf{w}}$ and $\widehat{\mathbf{\Sigma}}_{\overline{\mathbf{w}};-r}$, they are nonlinear estimators. Nevertheless, using $\widehat{\widehat{\beta}}$ and $\widehat{\widehat{\beta}}_{-r}$ we compute the pseudovalues, which give the desired CI.

The DOE literature ignores designs for heterogeneous output variances. We propose classic designs with m_i such that the resulting $V\widehat{a}r(\overline{w}_i) = s_i^2/m_i$ with i = 1, ..., n are approximately constant. First we take a *pilot sample* of size $m_0 \geq 2$ for each combination, which gives $s_i^2(m_0)$. Next we select a number of additional replications $\widehat{m}_i - m_0$ with

$$\widehat{m}_i = m_0 \times nint \; \left[\frac{s_i^2(m_0)}{min_i \; s_i^2(m_0)}\right] \tag{23}$$

where *nint* $[x]$ denotes the integer closest to x. We use the $\widehat{m}_i$ replications of both stages to compute $\overline{w}_i$ and s_i^2. From $\overline{w}_i$we compute $\widehat{\beta}$. We estimate $\mathbf{\Sigma}_{\widehat{\beta}}$ through (21) with $\mathbf{\Sigma}_{\mathbf{w}}$ estimated by a diagonal matrix with elements $s_i^2(\widehat{m}_i)/\widehat{m}_i$. We compute CIs for $\widehat{\beta}_j$ using t_v with $v = m_0 - 1$.

Actually, (23) guides the *relative* number of replications $\widehat{m}_i/\widehat{m}_{i'}$. To select absolute numbers $\widehat{m}$, we recommend the following rule derived in Law (2015, p. 505) with a relative estimation error γ:

$$\widehat{m} = min \left[r \geq m : \frac{t_{r-1;1-\alpha/2}\sqrt{s_i^2(m)/i}}{|\overline{w}(m)|} \leq \frac{\gamma}{1+\gamma} \right] \tag{24}$$

In Section 5 we shall return to the selection of m_i.

3.4 Common random numbers (CRN)

CRN are the default in software for discrete-event simulation. CRN are meant to compare the outputs of different input combinations while all other "circumstances" are the same; e.g., waiting times are compared for one or two servers while random customer arrivals are the same. Obviously, CRN create correlation between $w_{i;r}$ and $w_{i';r}$. Moreover, two different replications use nonoverlapping PRN streams, so $w_{i;r}$ and $w_{i;r'}$ with $r \neq r'$ are independent; i.e., $\mathbf{w}_r$ and $\mathbf{w}_{r'}$ are independent. The final goal of CRN is to reduce $Var(\widehat{\beta}_j)$ and $Var(\widehat{y})$; actually, CRN increase the variance of the estimated intercept.

If we continue to use OLS, then $\mathbf{\Sigma}_{\widehat{\beta}}$ is given by (21) but now $\mathbf{\Sigma}_{\mathbf{w}}$ is not a diagonal matrix. The estimator $\widehat{\mathbf{\Sigma}}_{\mathbf{w}}$ is *singular* if $m \leq n$; else we may compute CIs for $\widehat{\beta}_j$ from t_{m-1}. An alternative method requires only $m > 1$, and uses replication r to estimate β through

$$\hat{\beta}_r = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{w}_r \ (r = 1, \ldots, m). \tag{25}$$

Obviously, the n elements of $\mathbf{w}_r$ are correlated because of CRN, and they may have different variances. The m estimators $\widehat{\beta}_{j;r}$ (j = 1, ..., q; r = 1, ..., m) are independent and have a common standard deviation (say) $\sigma_{\widehat{\beta}_j}$, so we get

$$t_{m-1} = \frac{\overline{\widehat{\beta}}_j - \beta_j}{s(\overline{\widehat{\beta}}_j)} \text{ with } j = 1, \ldots, q \tag{26}$$

where $\overline{\widehat{\beta}}_j = \sum_{r=1}^{m} \widehat{\beta}_{j;r}/m$ and $s^2(\overline{\widehat{\beta}}_j) = \sum_{r=1}^{m} (\widehat{\beta}_{j;r} - \overline{\widehat{\beta}}_j)^2/[m(m-1)]$. Unfortunately, we cannot apply this alternative when estimating a *quantile* instead of a mean. In case of a quantile, we recommend distribution-free bootstrapping; see Kleijnen (2015) and Kleijnen et al. (2011).

3.5 Validation of metamodels

In practice, we do not know whether $E(\widehat{y}_i) = E(w_i)$; e.g., given a "small" experimental area, the estimated first-order polynomial may be adequate for estimating the gradient when searching for the optimum; see Section 6. We discuss the following validation methods: (i) two related coefficients of determination; namely, R^2 and R^2_{adj}; (ii) cross-validation. These methods may also be used to compare first-order against second-order polynomials, or linear regression against Kriging metamodels.

R^2 may be defined as

$$R^2 = \frac{\sum_{i=1}^{n}(\widehat{y}_i - \overline{\overline{w}})^2}{\sum_{i=1}^{n}(\overline{w}_i - \overline{\overline{w}})^2} = 1 - \frac{\sum_{i=1}^{n}(\widehat{y}_i - \overline{w}_i)^2}{\sum_{i=1}^{n}(\overline{w}_i - \overline{\overline{w}})^2} \tag{27}$$

where $\overline{\overline{w}} = \sum_{i=1}^{n}\overline{w_i}/n$ and $m_i \geq 1$. If $n = q$ (saturated design), then $R^2 = 1$—even if $E(\widehat{y}_i) \neq E(w_i)$. If $n > q$ and q increases, then R^2 increases—whatever the size of $|E(\widehat{y}_i) - E(w_i)|$ is. Because of possible *overfitting*, the regression literature *adjusts* R^2:

$$R^2_{\text{adj}} = 1 - \frac{n-1}{n-q}(1 - R^2). \tag{28}$$

Critical values for R^2 or R^2_{adj} are unknown, because these statistics do not have well-known distributions. We might use subjective lower thresholds. However, Kleijnen and Deflandre (2006) demonstrates how to estimate the distributions of these two statistics through *distribution-free bootstrapping*.

Cross-validation—or more precisely—*leave-one-out* cross-validation—may be defined as follows. For ease of presentation, we assume that $\mathbf{X}$ has only n instead of $N = \sum_{i=1}^{n}m_i$ rows, because $m_i = m \geq 1$ so in (3) we may replace $\mathbf{w}$ by $\overline{\mathbf{w}}$. In this cross-validation we delete I/O combination i to obtain $(\mathbf{X}_{-i}, \overline{\mathbf{w}}_{-i})$, and compute

$$\widehat{\beta}_{-i} = (\mathbf{X}'_{-i}\mathbf{X}_{-i})^{-1}\mathbf{X}'_{-i}\overline{\mathbf{w}}_{-i}, \tag{29}$$

which gives $\widehat{y}_{-i} = \mathbf{x}'_i\widehat{\beta}_{-i}$ with $i = 1., ..., n$. We may "eyeball" the *scatterplot* with $(\overline{w}_i, \widehat{y}_{-i})$, and decide whether the metamodel is valid. Alternatively, we may compute

$$t^{(i)}_{m-1} = \frac{\overline{w}_i - \widehat{y}_i}{\sqrt{s^2(\overline{w}_i) + s^2(\widehat{y}_{-i})}} \; (i = 1, \ldots, n) \tag{30}$$

where $s^2(\overline{w}_i) = s_i^2/m$ and

$$s^2(\widehat{y}_{-i}) = \mathbf{x}'_i\widehat{\mathbf{\Sigma}}_{\widehat{\beta}_{-i}}\mathbf{x}_i \text{ with } \widehat{\mathbf{\Sigma}}_{\widehat{\beta}_{-i}} = s^2(\overline{w}_i)(\mathbf{X}'_{-i}\mathbf{X}_{-i})^{-1}. \tag{31}$$

We reject the regression metamodel if

$$max_i \; |t^{(i)}_{m-1}| > t_{m-1;1-[\alpha/(2n)]} \tag{32}$$

where *Bonferroni's inequality* implies that $\alpha/2$ is replaced by $\alpha/(2n)$—resulting in the *experimentwise* or *familywise* type-I error rate α. Cross-validation affects not only $\widehat{y}_{-i}$, but also $\widehat{\beta}_{-i}$; see (29). Actually, we may be interested not only in the predictive performance of the metamodel, but also in its *explanatory* performance.

Related to cross-validation are several *diagnostic* statistics in the regression literature; e.g., DEFITS, DFBETAS, and Cook's D. The most popular diagnostic statistic, however, is the *prediction sum of squares* (PRESS):

$$\text{PRESS} = \sqrt{\frac{\sum_{i=1}^{n}(\widehat{y}_{-i} - w_i)^2}{n}}. \tag{33}$$

Regression software uses a shortcut to avoid the n recomputations in cross-validation. We may apply bootstrapping to estimate the distribution of these validation statistics; see Bischl et al. (2012).

If the validation suggests a big fitting error e, then we may consider various *transformations*. For example, in queueing simulations we may combine the arrival rate (say) λ and the service rate μ into the traffic rate $x = \lambda/\mu$. Another transformation replaces y, λ, and μ by $\log(y)$, $\log(\lambda)$, and $\log(\mu)$ so that the first-order polynomial approximates relative changes through elasticity coefficients. If we assume that f_{sim} is monotonic, then we may replace w and x_j by their ranks: *rank regression*. In the preceding subsections, we also considered transformations that make w better satisfy the assumptions of normality and variance homogeneity; unfortunately, different goals of a transformation may conflict with each other.

In Section 2 we discussed designs for low-order polynomials. If such a design does not give a valid metamodel, then we do not recommend routinely adding higher-order terms: these terms are hard to interpret. However, if the goal is not to better *understand* the simulation model but to better *predict* its output, then we may add high-order terms; e.g., a 2^k design enables the estimation of all interactions. In the discussion of (28), we have already mentioned the danger of overfitting. Note that adding more explanatory variables is called *stepwise regression*; eliminating nonsignificant variables (see (26)) is called *backwards elimination*.

4 Screening the many inputs of realistic simulation models

Screening means searching for the really important inputs among the many inputs that can be varied in a simulation experiment. It is realistic to assume that effects are *sparse*; i.e., only a few inputs among these many inputs are really important. Indeed, the *Pareto* principle or *20-80* rule states that only a few inputs—say, 20%—are really important. Kleijnen (2015) presents two examples, with 281 and 92 inputs, respectively; screening finds only 15 and 11 inputs to be really important.

The (rather scarce) literature presents several types of screening designs. We focus on designs that treat the simulation as a *black box*; i.e., only the I/O of the simulation is observed. Kleijnen (2015) summarizes four types of screening designs; these types use different mathematical assumptions. We focus on *sequential bifurcation* (SB), because SB is very efficient and effective if its assumptions are satisfied. SB selects the next input combination after analyzing the preceding I/O data. SB is *customized*; i.e., SB accounts for the specific simulation model. Notice that Borgonovo and Plischke (2015) also summarizes SB, besides Morris's method; Morris's method is also discussed in Fédou and Rendas (2015).

4.1 SB for deterministic simulations and low-order polynomial metamodels

To explain the basic idea of SB, we assume deterministic simulation and a *first-order* polynomial with approximation error e where $E(e) = 0$:

$$y = \gamma_0 + \gamma_1 z_1 + \ldots + \gamma_k z_k + e. \tag{34}$$

Furthermore, we assume that the *signs* of γ_j ($j = 1, ..., k$) are known so that we can define the lower and upper bounds l_j and u_j of z_j such that $\gamma_j \geq 0$. Together (34) and $\gamma_j \geq 0$ imply that we may rank the inputs such that the most important input has the largest first-order effect; the least important inputs have effects close to zero. Input j is called *important* if $\gamma_j > \Delta$ where $\Delta \geq 0$ is specified by the users.

In its first step, SB aggregates all k inputs into a single group, and checks whether or not that group has an important effect. So in this step, SB obtains $w(\mathbf{z} = \mathbf{l})$ where $\mathbf{z} = (z_j)$ and $\mathbf{l} = (l_j)$. In this step, SB also obtains $w(\mathbf{z} = \mathbf{h})$ where $\mathbf{h} = (h_j)$. Obviously, if all inputs have zero effects, then $w(\mathbf{z} = \mathbf{l}) = w(\mathbf{z} = \mathbf{h})$. However, if one or more inputs have positive effects, then $w(\mathbf{z} = \mathbf{l}) < w(\mathbf{z} = \mathbf{h})$. In practice, not all k inputs have zero effects. It may happen that all effects are unimportant ($0 \leq \gamma_j < \Delta$), but $w(\mathbf{z} = \mathbf{h}) - w(\mathbf{z} = \mathbf{l}) > \Delta$.

If SB finds that the group has an important effect, then the next step splits the group into two subgroups: *bifurcation.* Let k_1 denote the size of subgroup 1 and k_2 the size of subgroup 2 (so $k_1 + k_2 = k$). SB obtains w_{k_1} which denotes w when all k_1 inputs in subgroup 1 are "high". SB compares this w_{k_1} with $w_0 = w(\mathbf{z} = \mathbf{l})$; if $w_{k_1} - w_0 < \Delta$, then none of the individual inputs within subgroup 1 is important and SB *eliminates* this subgroup from further experimentation. SB also compares w_{k_1} with $w_k = w(\mathbf{z} = \mathbf{h})$. The result w_k - $w_{k_1} < \Delta$ is unlikely, because we expect that at least one input is important and that this input is a member of subgroup 2.

SB continues splitting important subgroups into smaller subgroups, and eliminating unimportant subgroups. It may happen that SB finds both subgroups to be important, which leads to further experimentation with two important subgroups. Finally, SB identifies and estimates all individual inputs that are not in eliminated subgroups.

Obviously, assuming $\gamma_j \geq 0$ ensures that first-order effects do not cancel each other within a group. Furthermore, we can define the inputs such that if f_{sim} is monotonically decreasing in z_j, then this function becomes monotonically increasing in the standardized inputs x_j. Experience shows that in practice the users often do know the signs of γ_j; e.g., some inputs may be transportation speeds so the higher these speeds, the lower the cost which is the output of interest in one SB case study. Nevertheless, if in a specific case study it is hard to specify the signs of a few specific inputs, then we should treat these inputs *individually*; i.e., we should not group these inputs with other inputs in SB. Treating such inputs individually is safer than assuming a negligible probability of cancellation within a subgroup.

The *efficiency* of SB—measured by the number of simulated input combinations—improves if the individual inputs are labeled such that inputs are placed in increasing order of importance. Such labeling implies that the important inputs are *clustered*. The efficiency further improves when placing similar inputs within the same subgroup. So, splitting a group into subgroups of *equal* size is not necessarily optimal. Academic and practical examples are given in Kleijnen (2015).

Now we assume a *second-order* polynomial plus approximation error e with $E(e) = 0$. Moreover, we assume that if $\gamma_j = 0$, then $\gamma_{j;j} = 0$ and $\gamma_{j;j'} = 0$ with $j' \neq j$: *heredity* assumption; see Wu and Hamada (2009). In Section 2.2 we discussed the *foldover* principle for constructing R-IV designs from R-III designs; likewise, SB enables the estimation of γ_j unbiased by $\gamma_{j;j}^2$ and $\gamma_{j;j'}$ if SB simulates the mirror input of the original input in its sequential design. We let w_{-j} denote w when the first j standardized inputs are -1. Kleijnen (2015) shows that an unbiased estimator of $\beta_{j'-j} = \sum_{h=j'}^{j} \beta_h$ is

$$\widehat{\beta}_{j'-j} = \frac{(w_j - w_{-j}) - (w_{j'-1} - w_{-(j'-1)})}{4}. \tag{35}$$

4.2 SB for random simulations and second-order polynomials

Initially we assume a *fixed* number of replications m per simulated input combination. Let $w_{j;r}$ with $r = 1, ..., m$ denote observation r on w_j. Furthermore we assume a second-order polynomial metamodel. We then obtain the analogue of (35)): $\widehat{\beta}_{(j'-j);r} = (w_{j;r} - w_{(-j);r}) - (w_{(j'-1);r} - w_{-(j'-1);r})/4$. This enables estimation of the mean and the variance of $\widehat{\beta}_{(j'-j);r}$, which gives a t_{m-1}-test—analogously to (26). In SB we apply a one-sided test because SB assumes $\gamma_j > 0$ so $\beta_j > 0$.

Whereas the preceding t-test assumes a favorite null-hypothesis, the *sequential probability ratio test* (SPRT) in Wan et al. (2010) considers two comparable hypotheses, and selects m such that the SPRT controls the type-I error probability through the whole procedure and holds the type-II error probability at each step.

This SPRT classifies inputs with $\beta_j \leq \Delta_0$ as *unimportant* and inputs with $\beta_j \geq \Delta_1$ as *important* where Δ_0 and Δ_1 are specified by the users; for *intermediate* inputs—with $\Delta_0 < \beta_j < \Delta_1$—the power should be "reasonable". The *initial* number of replications when estimating $\beta_{j'-j}$ is $m_{0;j'-j}$. We expect that $m_{0;j'-j}$ may be smaller in the early stages, because those stages estimate the sum of the positive first-order effects of bigger groups so the signal-noise ratio is larger. These $m_{0;j'-j}$ replications give

$$s^2(\widehat{\beta}_{j'-j}) = \frac{\sum_{r=1}^{m_{0;j'-j}} (\widehat{\beta}_{(j'-j);r} - \overline{\widehat{\beta}}_{j'-j})^2}{m_{0;j'-j} - 1} \text{ with } \overline{\widehat{\beta}}_{j'-j} = \frac{\sum_{r=1}^{m_{0;j'-j}} \widehat{\beta}_{(j'-j);r}}{m_{0;j'-j}}. \quad (36)$$

Statistical details of the SPRT and a Monte Carlo experiment are given in Kleijnen (2015) and Shi et al. (2014).

4.3 Multiresponse SB: MSB

In practice, simulation models have *multiple response types* (also see Section 3.1). Shi et al. (2014) extends SB to *multiresponse SB* (MSB). This MSB selects groups of inputs such that within a group all inputs have the same sign for a specific type of output, so no cancellation of first-order effects occurs. More precisely, denote the number of simulation outputs by n (not to be confused with n in the preceding sections). By definition, changing the value of input j from $L_j^{(l)}$ to $H_j^{(l)}$ increases output l ($l = 1, ..., n$). This change, however, may decrease another output $l' \neq l$; e.g., if there are k inputs and $n = 2$ outputs, then inputs 1 through k_1 may have the same signs for both outputs, whereas inputs $k_1 + 1$ through k have opposite signs for the two outputs. MSB also applies the SPRT summarized in Section 4.2. For more details including extensive Monte Carlo experiments and a case study concerning a logistic system in China we refer to Kleijnen (2015) and Shi et al. (2014).

4.4 Validating the SB and MSB assumptions

By definition, "screening" means that k is too big to enable the estimation of all the q individual effects of a second-order polynomial; e.g., the Chinese case study has only $k = 26$, and yet $q = 378$. We denote the number of *unimportant* inputs identified through SB or MSB by k_{U} ("U" stands for unimportant), and the number of important inputs by k_{I} ("I"stands for important); obviously, $k_{\mathrm{U}} + k_{\mathrm{I}} = k$. Each of these k_{U} inputs has nearly the same magnitude for $\widehat{\beta}_j^{(l)}$; namely, smaller than $\Delta_0^{(l)}$. So we do not estimate the many individual—first order and second order—effects of the unimportant inputs, but test whether these inputs indeed have virtually no effects. So we simulate only a few *extreme* combinations of these inputs. To explain this validation method, we consider a simulation with a single output type so SB suffices. We then simulate only the following two extreme combinations of unimportant inputs (for simplicity we assume that the k_{I} inputs are quantitative): (a) all k_{U} inputs that SB identified as unimportant are fixed at -1, while all k_{I} remaining inputs are fixed at 0; (b) all k_{U} inputs are fixed at 1, while the k_{I} remaining inputs are fixed at the same values as in combination (a). We relabel the k inputs such that the first k_{U} inputs are declared to be unimportant. We let $\mathbf{x}_{\mathrm{U}}$ denote the k_{U}-dimensional vector with the values of the unimportant inputs, and $\mathbf{1}$ denote the k_{U}-dimensional vector with all elements equal to 1. Consequently, combinations (a) and (b) together with (2) give

$$E(y \mid \mathbf{x}_{\mathrm{U}} = \mathbf{1}) - E(y|\mathbf{x}_{\mathrm{U}} = -\mathbf{1}) = 2\sum_{j=1}^{k_{\mathrm{U}}} \beta_j = 2\beta_{1-k_{\mathrm{U}}}. \tag{37}$$

Let m_{val} denote the number of replications for these two combinations (to choose m_{val}, we may examine the final m that the SPRT needed to test the significance of individual inputs). This gives

$$d_r = w_r(\mathbf{x}_{\mathrm{U}} = \mathbf{1}) - w_r(\mathbf{x}_{\mathrm{U}} = -\mathbf{1}) \ (r = 1, ..., m_{\mathrm{val}}).$$

If we use CRN, then we get so-called *paired differences* and use

$$t_{m_{\mathrm{val}}-1} = \frac{\overline{d} - E(d)}{s(d)/\sqrt{m_{\mathrm{val}}}}. \tag{38}$$

This statistic gives a CI for $\delta = E(d)$, which we may use to test

$$H_0 : E(d) \leq \Delta \text{ versus } H_1 : E(d) > \Delta \tag{39}$$

where $\leq$ implies a one-sided hypothesis—which we use because $\beta_j \geq 0$. We reject this H_0 only if the observed value of the statistic in (38) with $E(d)$ replaced by Δ is higher than $t_{m_{\mathrm{val}}-1;1-\alpha}$. We select $\Delta = 2k_{\mathrm{U}}\Delta_0$ where Δ_0 was used to define unimportant inputs, and the k_{U} unimportant inputs might have a total effect of $2k_{\mathrm{U}}\Delta_0$; see the factor 2 in (37). Altogether, we accept bigger differences between the outputs for the extreme input combinations, as k_{U} increases.

Finally, we test the *heredity* assumption (if $\beta_j = 0$, then $\beta_{j;j} = 0$ and $\beta_{j;j'} = 0$ with $j \neq j'$). Our test of the combinations (a) and (b) is insensitive to these $\beta_{j;j}$ and $\beta_{j;j'}$. Therefore we now consider the *center combination* $\mathbf{x}_0 = \mathbf{0}$ where $\mathbf{0}$ denotes the k_{U}-dimensional vector with all elements equal to zero. Obviously, if the heredity assumption does not apply, then $E(y \mid \mathbf{x}_{\mathrm{U}} = \mathbf{0}) \neq E(y \mid \mathbf{x}_{\mathrm{U}} = -\mathbf{1}) = E(y \mid \mathbf{x}_{\mathrm{U}} = \mathbf{1})$. So we compute

$$d_{0;r} = w_r(\mathbf{x}_{\mathrm{U}} = \mathbf{0}) - [\frac{w_r(\mathbf{x}_{\mathrm{U}} = -\mathbf{1}) + w_r(\mathbf{x}_{\mathrm{U}} = \mathbf{1})}{2}] \ (r = 1, ..., m_{\mathrm{val}}).$$

If the second-order polynomial holds for the k_{U} unimportant inputs, then $E(d_0) = -\sum_{j=1}^{k_{\mathrm{U}}} \sum_{j'=j}^{k_{\mathrm{U}}} \beta_{j;j'}$ where some of the $k_{\mathrm{U}}(k_{\mathrm{U}} - 1)/2 + k_{\mathrm{U}}$ second-order effects $\beta_{j;j'}$ may be negative and some may be positive so we do not make any assumptions about the magnitude of this sum. These $d_{0;r}$ give the analogue of (38), which we use to test

$$H_0 : E(d_0) = 0 \text{ versus } H_1 : E(d_0) \neq 0 \tag{40}$$

where we now use a two-sided hypothesis, because the second-order effects may be negative or positive. We reject this H_0 if $|t_{0;m_{\mathrm{val}}-1}| > t_{m_{\mathrm{val}}-1;1-\alpha/2}$.

Next we briefly discuss MSB with two output types. If there were a single input group, then the method would be the same as the method for the SB explained above. If there are two input groups, then we simulate the two extreme combinations for each of the two output types.; i.e., we simulate four combinations, as explained in Kleijnen (2015).

Shi and Kleijnen (2015) also presents a validation method that takes advantage of the existence of *input groups*. These input groups enable us to estimate the effects of an input group for all n output types *simultaneously*, so we save on simulation effort. This method may be more efficient than the method detailed above.

5 Kriging metamodels and their designs

Kriging metamodels are fitted to simulation I/O data obtained for larger or *global* experimental areas than the *local* areas in low-order polynomial metamodels.

5.1 Ordinary Kriging (OK) in deterministic simulation

In this section we focus on OK, which is popular and successful in practical deterministic simulation; see Chen et al. (2006). In deterministic simulation, OK assumes

$$y(\mathbf{x}) = \mu + M(\mathbf{x}) \text{ with } \mathbf{x} \in \mathbb{R}^k \tag{41}$$

where μ is the constant mean $E[y(\mathbf{x})]$, $M(\mathbf{x})$ is is a Gaussian stationary process with zero mean. By definition, such a process has covariances that depend only on the distance between the input combinations $\mathbf{x}$ and $\mathbf{x}'$ (we use $\mathbf{x}$ instead of $\mathbf{d}$ because the Kriging literature uses $\mathbf{x}$, whereas DOE and the preceding sections use $\mathbf{d}$). $\mathbf{X}$ denotes an $n \times k$ matrix with the n combinations $\mathbf{x}_i$ ($i = 1, ..., n$). $M(\mathbf{x})$ is called the *extrinsic noise* to distinguish it from the *intrinsic noise* in stochastic simulation; see Section 5.3.

OK computes the predictor $\widehat{y}(\mathbf{x}_0)$ for the *new* combination $\mathbf{x}_0$ as a *linear* function of the n *old* outputs $\mathbf{w}$ at $\mathbf{X}$:

$$\widehat{y}(\mathbf{x}_0) = \textstyle\sum_{i=1}^{n} \lambda_i w_i = \boldsymbol{\lambda}'\mathbf{w}. \tag{42}$$

The weight λ_i decreases with the *distance* between $\mathbf{x}_0$ and $\mathbf{x}_i$, so λ is not a constant vector (whereas β in linear regression is constant). The *optimal* weight-vector λ_o gives the *best linear unbiased predictor* (BLUP*).* "Unbiased" implies that if $\mathbf{x}_0 = \mathbf{x}_i$, then the predictor is an *exact interpolator*: $\widehat{y}(\mathbf{x}_i) = w(\mathbf{x}_i)$. We can prove that

$$\lambda_o' = [\sigma_M(\mathbf{x}_0) + \mathbf{1}\frac{1 - \mathbf{1}'\boldsymbol{\Sigma}_M^{-1}\sigma(\mathbf{x}_0)}{\mathbf{1}'\boldsymbol{\Sigma}_M^{-1}\mathbf{1}}]'\boldsymbol{\Sigma}_M^{-1} \tag{43}$$

where $\boldsymbol{\Sigma}_M = (\text{cov}(y_i, y_{i'}))$ denotes the $n \times n$ matrix with the covariances between the metamodel's "old" outputs, and $\sigma_M(\mathbf{x}_0) = (\text{cov}(y_i, y_0))$ denotes the n-dimensional vector with the covariances between the metamodel's n old outputs

y_i and the new output y. Furthermore, we can prove that λ_o implies

$$\widehat{y}(\mathbf{x}_0) = \mu + \sigma_M(\mathbf{x}_0)'\mathbf{\Sigma}_M^{-1}(\mathbf{w}-\mu\mathbf{1}) \tag{44}$$

where $\mathbf{1}$ is an n-dimensional vector with all elements equal to 1. Obviously, $\widehat{y}(\mathbf{x}_0)$ varies with $\sigma_M(\mathbf{x}_0)$; given are μ, $\mathbf{\Sigma}_M$, and $\mathbf{w}$.

The *gradient* $\nabla(\widehat{y})$ follows from (44); see Lophaven et al. (2002, Eq. 2.18). We should not confuse $\nabla(\widehat{y})$ and $\nabla(w)$; sometimes we can indeed estimate $\nabla(w)$, and use $\widehat{\nabla}(w)$ to estimate a better OK model; see Qu and Fu (2014) and Ulaganathan et al. (2014).

Let τ^2 denote Var(y). The *mean squared error* (MSE) of $\widehat{y}(\mathbf{x}_0)$ can be proven to be

$$\begin{aligned}\text{MSE }[\widehat{y}(\mathbf{x}_0)] &= \tau^2 - \sigma_M(\mathbf{x}_0)'\mathbf{\Sigma}_M^{-1}\sigma_M(\mathbf{x}_0)\\ &\quad + \frac{[1-\mathbf{1}'\mathbf{\Sigma}_M^{-1}\sigma_M(\mathbf{x}_0)]^2}{\mathbf{1}'\mathbf{\Sigma}_M^{-1}\mathbf{1}}.\end{aligned} \tag{45}$$

Because $\widehat{y}(\mathbf{x}_0)$ is unbiased, this MSE equals Var$[\widehat{y}(\mathbf{x}_0)]$. Obviously, Var$[\widehat{y}(\mathbf{x}_0)]$ $= 0$ if $\mathbf{x}_0 = \mathbf{x}_i$. So, Var$[\widehat{y}(\mathbf{x}_0)]$ has many local minima. Experimental results suggest that Var$[\widehat{y}(\mathbf{x}_0)]$ has local maxima at $\mathbf{x}_0$ approximately halfway between old input combinations. Kriging gives bad extrapolations compared with interpolations (linear regression also gives minimal Var$[\widehat{y}(\mathbf{x}_0)]$ when $\mathbf{x}_0 = \mathbf{0}$).

Obviously, λ_o depends on $\mathbf{\Sigma}_M$ and $\sigma_M(\mathbf{x}_0)$—or switching from covariances to correlations—on $\mathbf{R} = \tau^{-2}\mathbf{\Sigma}_M$ and $\rho(\mathbf{x}_0) = \tau^{-2}\sigma_M(\mathbf{x}_0)$. There are several types of correlation functions; see Rasmussen and Williams (2006, pp. 80–104). Most popular is the *Gaussian* correlation function:

$$\rho(\mathbf{h}) = \prod_{j=1}^{k} \exp\ \left(-\theta_j h_j^2\right) = \exp\ (-\sum_{j=1}^{k}\theta_j h_j^2) \tag{46}$$

with distance vector $\mathbf{h} = (h_j)$ where $h_j = |x_{g;j} - x_{g';j}|$ and $g, g' = 0, 1, ..., n$. Obviously, $\rho(\mathbf{h})$ implies that λ_o in (43) has relatively high elements for $\mathbf{x}_i$ close to $\mathbf{x}_0$. Kriging software standardizes the original simulation inputs and outputs, which affects the values in $\mathbf{h}$.

To estimate the *Kriging parameters* $\psi = (\mu, \tau^2, \theta')'$ with $\widehat{\theta} = (\widehat{\theta}_j)'$, the most popular criterion is *maximum likelihood* (ML)—but LS and cross-validation are also used. We denote the resulting estimator by $\widehat{\psi}$. The estimation of ψ is a mathematical challenge; different values for $\widehat{\psi}$ may result from different software packages or from initializing the same package with different starting values.

Plugging $\widehat{\psi}$ into (44), we obtain

$$\widehat{y}(\mathbf{x}_0, \widehat{\psi}) = \widehat{\mu} + \widehat{\sigma}(\mathbf{x}_0)'\widehat{\mathbf{\Sigma}}^{-1}(\mathbf{w}-\widehat{\mu}\mathbf{1}). \tag{47}$$

Obviously, $\widehat{y}(\mathbf{x}_0, \widehat{\psi})$ is a *nonlinear* predictor. In practice, we simply *plug* $\widehat{\psi}$ into (45) to obtain MSE$[\widehat{y}(\mathbf{x}_0, \widehat{\psi})]$; moreover, we ignore possible bias of $\widehat{y}(\mathbf{x}_0)$ so $s^2\{\widehat{y}(\mathbf{x}_0)\} = $ MSE$[\widehat{y}(\mathbf{x}_0, \widehat{\psi})]$. We use this $s^2[\widehat{y}(\mathbf{x}_0)$ to compute a CI:

$$P[w(\mathbf{x}_0) \in [\widehat{y}(\mathbf{x}_0, \widehat{\psi}) \pm z_{\alpha/2}s\{\widehat{y}(\mathbf{x}_0)\}] = 1 - \alpha. \tag{48}$$

There is much *software* for Kriging; see Kleijnen (2015).

Note: There are many publications that interpret Kriging in a *Bayesian* way.; see Yuan and Ng (2015). However, we find it hard to come up with a prior distribution for ψ, because we have little intuition about θ.

Note: We may apply *parametric bootstrapping* to estimate the true MSE of $\widehat{y}(\mathbf{x}_0, \widehat{\psi})$ in (47). We may also apply a bootstrap variant called *conditional simulation*. These two methods are rather complicated, and yet they give CIs with coverages and lengths that are not superior compared with the CI specified in (48); see Kleijnen (2015).

Note: *Universal Kriging* (UK) replaces μ in (41) by $\mathbf{f}(\mathbf{x})'\beta$ where $\mathbf{f}(\mathbf{x})$ is a q-dimensional vector of known functions of $\mathbf{x}$ and β is the corresponding q-dimensional vector of unknown parameters. The disadvantage of UK is that UK requires the estimation of additional parameters: besides $\mu = \beta_0$. We conjecture that the estimation of these $q-1$ extra parameters explains why UK has a higher MSE. In practice, most Kriging models do not use UK but OK.

5.2 Designs for deterministic simulation

There is an abundant literature on various design types for Kriging in deterministic simulation; e.g., orthogonal array, uniform, maximum entropy, minimax, maximin, integrated mean squared prediction error, and "optimal" designs. However, most popular is *Latin hypercube sampling* (LHS), which we explain next.

LHS assumes that an adequate metamodel is more complicated than a low-order polynomial; LHS does not assume a specific type of metamodel, but focuses on $\mathbb{R}^k$, the input space formed by the k–dimensional unit cube defined by the standardized simulation inputs. LHS does not imply a strict mathematical relationship between n and k, 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 "space filling" LHS covers $\mathbb{R}^k$ so sparsely that the fitted Kriging model may be inadequate. Therefore a well-known rule-of-thumb for LHS in Kriging is $n = 10k$; see Loeppky et al. (2009).

Technically, LHS divides the range of each input into n mutually exclusive and exhaustive intervals of equal probability. LHS gives a *noncollapsing* design; i.e., if an input turns out to be unimportant, then each remaining individual input is still sampled with one observation per interval (the estimation of the correlation function may benefit from this noncollapsing property). Unfortunately, projections of a LHS-combination in $\mathbb{R}^k$ onto more than one dimension may give "bad' designs. Therefore standard LHS is further refined, leading to so-called maximin LHS and nearly-orthogonal LHS.

Now we switch from *fixed sample* or *one shot* designs to *sequential* designs that account for f_{sim} so the sequential designs are application-driven or *customized*. Sequential procedures require fewer observations than fixed-sample procedures. In sequential designs we learn about the behavior of the underlying system as we experiment with this system and collect data (also see Section 4 on SB). Unfortunately, extra computer time is needed if we re-estimate ψ when new I/O data become available.

Kleijnen (2015) details sequential designs for Kriging, to perform either *SA* (so the whole experimental area is of interest) or *optimization* (so only the global optimum is interesting). In such a design we may start with a *pilot* experiment using LHS, and use this experiment to obtain simulation I/O data. Next we fit a Kriging model to these data. Then we may consider—but not yet simulate—$\mathbf{X}_c$ which denotes a larger set of *candidate* combinations selected through LHS, and find the "winning" candidate. In SA this winner has the highest $s^2\{\widehat{y}(\mathbf{x})\}$ with $\mathbf{x} \in \mathbf{X}_c$; for optimization the winner is discussed in Section 6.2. Experiments show that sequential designs for SA select relatively few combinations in subareas with an approximately linear f_{sim}, and many combinations in the other subareas; in optimization the winner looks "promising". Next we use this winner as the input for the next simulation run, which gives additional I/O data. We may now re-fit (update) the Kriging model to the augmented I/O data. We stop if either the Kriging metamodel satisfies a given goal or the computer budget is exhausted.

5.3 Stochastic Kriging (SK)

SK adds the *intrinsic noise* term $\varepsilon_r(\mathbf{x})$ for replication r at combination $\mathbf{x}$ to (41), assuming $\varepsilon_r(\mathbf{x}) \in N(0,\text{Var}[\varepsilon_r(\mathbf{x})])$ and $\varepsilon_r(\mathbf{x})$ independent of $M(\mathbf{x})$. Averaging over replications, SK replaces (41) by

$$\overline{y}(\mathbf{x}_i) = \mu + M(\mathbf{x}_i) + \overline{\varepsilon}(\mathbf{x}_i) \text{ with } \mathbf{x} \in \mathbb{R}^k \text{ and } i = 1, \dots, n. \tag{49}$$

Obviously, m_i replications without CRN make $\mathbf{\Sigma}_{\overline{\varepsilon}}$ diagonal with main-diagonal elements $\text{Var}[\varepsilon(\mathbf{x}_i)]/m_i$. CRN and $m_i = m$ give $\mathbf{\Sigma}_{\overline{\varepsilon}} = \mathbf{\Sigma}_{\varepsilon}/m$.

SK may use s_i^2 defined in (20). Alternatively, SK may use another Kriging metamodel for $\text{Var}[\varepsilon(\mathbf{x}_i)]$—besides the Kriging metamodel for the mean $E[y_r(\mathbf{x}_i)]$—to predict $\text{Var}[\varepsilon(\mathbf{x}_i)]$. This alternative may give less volatile estimates than the point-estimates s_i^2. Because s_i^2 is not normally distributed, the GP is only a rough approximation. We might also replace s by $\log(s_i^2)$ in the Kriging metamodel; also see Section 3.3 and Kamiński (2015).

To get the SK predictor, Ankenman et al. (2010, Eq. 25) replaces $\mathbf{\Sigma}_M$ in OK by $\mathbf{\Sigma}_M + \mathbf{\Sigma}_{\overline{\varepsilon}}$ and $\mathbf{w}$ by $\overline{\mathbf{w}}$:

$$\widehat{y}(\mathbf{x}_0, \widehat{\psi}) = \widehat{\mu} + \widehat{\sigma}(\mathbf{x}_0)'(\widehat{\mathbf{\Sigma}}_M + \widehat{\mathbf{\Sigma}}_{\overline{\varepsilon}})^{-1}(\overline{\mathbf{w}} - \widehat{\mu}\mathbf{1}) \tag{50}$$

and

$$\begin{aligned} s^2\{\widehat{y}(\mathbf{x}_0)\} &= \widehat{\tau}^2 - \widehat{\sigma}(\mathbf{x}_0)'(\widehat{\mathbf{\Sigma}}_M + \widehat{\mathbf{\Sigma}}_{\overline{\varepsilon}})^{-1}\widehat{\sigma}(\mathbf{x}_0) \\ &\quad + \frac{[1 - \mathbf{1}'(\widehat{\mathbf{\Sigma}}_M + \widehat{\mathbf{\Sigma}}_{\overline{\varepsilon}})^{-1}\widehat{\sigma}(\mathbf{x}_0)]^2}{\mathbf{1}'(\widehat{\mathbf{\Sigma}}_M + \widehat{\mathbf{\Sigma}}_{\overline{\varepsilon}})^{-1}\mathbf{1}}. \end{aligned} \tag{51}$$

SK for a *quantile* instead of an average is discussed in Chen and Kim (2013).

In the discussion of (48) we mentioned the problems caused by the randomness of $\widehat{\psi}$. To solve this problem we may apply *distribution-free bootstrapping*; see Van Beers and Kleijnen (2008) and Yin et al. (2009).

Usually SK employs the same designs as OK does for deterministic simulation. So, SK often uses one-shot *LHS*. However, we also need to select m_i. In Section 3.3 we have already discussed the analogous problem for regression metamodels; a simple rule-of-thumb is (24).

In sequential designs, we may select the "winner" defined in Section 5.2. In SK we may select this winner, using distribution-free bootstrapping. Van Beers and Kleijnen (2008) selects more input values in the subdomain of the experimental area that gives a highly nonlinear estimated I/O function; this design gives better Kriging predictions than the fixed LHS design—especially for small designs, which are used in expensive simulations.

5.4 Monotonic Kriging

In practice we sometimes know that f_{sim} is *monotonic*; e.g., if the traffic rate increases, then the mean waiting time increases. More examples were given in Section 4 on screening. The Kriging metamodel, however, may be *wiggling* if n is small. To make $\widehat{y}$ monotonic, we may apply *distribution-free bootstrapping* with *acceptance/rejection*; i.e., we reject the Kriging metamodel fitted in bootstrap sample b—with $b = 1, ..., B$ and bootstrap sample size B—if this metamodel is not monotonic. A monotonic predictor means that the estimated *gradients* of the predictor remain positive as the inputs increase. An advantage of monotonic Kriging is that in practice the resulting SA is better understood and accepted. Furthermore, monotonic Kriging may give a smaller MSE and a CI with higher coverage and acceptable length. Finally, the estimated gradients with correct signs may improve optimization.

Methodologically, we assume that no CRN are used, and m_i ($i = 1, ..., n'$) is not necessarily a constant m. Let $\mathbf{x}_i < \mathbf{x}_{i'}$ with $i \neq i'$ 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. Between (44) and (45) we have already mentioned that $\widehat{y}(\mathbf{x})$ enables the computation of $\nabla\widehat{y}(\mathbf{x})$. We use a *test set* with v "new" combinations. We assume that a 90% CI is desired. We start with resampling $w_{i;r}$ ($r = 1, ..., m_i$) with replacement, to obtain the m_i-dimensional vector $\mathbf{w}^*_{i;b} = (w^*_{i;r;b})$ ($b = 1, ..., B$); resampling all n combinations gives $(\mathbf{X},\overline{\mathbf{w}}^*_b)$ where $\overline{\mathbf{w}}^*_b$ has the n elements $\overline{w}^*_{i;b} = \sum_{r=1}^{m_i} w^*_{i;r;b}/m_i$. Using this $(\mathbf{X},\overline{\mathbf{w}}^*_b)$, we compute $\widehat{\boldsymbol{\psi}}^*$. Using $(\mathbf{X},\overline{\mathbf{w}}^*_b)$ and $\widehat{\boldsymbol{\psi}}^*_b$, we compute $\widehat{y}^*_b$. We *accept* this $\widehat{y}^*_b$ only if $\nabla\widehat{y}^*_{i;b'} > \mathbf{0}$. We use the B_a accepted $\widehat{y}^*_b$ to compute B_a predictions for the test set $\mathbf{x}_u$ ($u = 1, ..., v$). These B_a predictions $\widehat{y}^*_u$ give the sample median $\widehat{y}^*_{u;(0.50B_a)}$ as the point estimate and $[\widehat{y}^*_{u;(0.05B_a)}, \widehat{y}^*_{u;(0.95B_a)}]$ as the two-sided 90% CI; see (19). If we find the resulting CI too wide, then we increase B so B_a probably increases too.

To quantify the performance of this method in SA, we may use the *estimated integrated mean squared error* (EIMSE). where we average over $\mathbf{x}_u$. Furthermore, OK uses the CI defined in (48), which is symmetric around its point estimate $\widehat{y}$ and may include negative values—even if negative values are impossible, as is the case for waiting times. Experiments show that—compared with OK—monotonic Kriging gives a smaller—but not significantly smaller—EIMSE, and significantly higher estimated coverages of the CI without widening the CI.

We may also apply Kriging with acceptance/rejection to preserve other I/O characteristics; e.g., *positive* values for waiting times, variances, and thickness. Furthermore, we may apply this method to other types of metamodels such as the *linear regression* metamodel.

5.5 Global sensitivity analysis

So far we focused on the *predictor* $\widehat{y}(\mathbf{x})$, but now we measure how sensitive $\widehat{y}(\mathbf{x})$—and hence $w(\mathbf{x})$—are to the individual inputs x_1 through x_k and their interactions. We assume that $\mathbf{x}$ has a prespecified distribution, as in LHS; see Section 5.2. We apply *functional analysis of variance* (FANOVA), using variance-based indexes originally proposed by Sobol. FANOVA decomposes the output variance σ_w^2 into fractions that refer to the individual inputs or to sets of inputs; e.g., FANOVA may show that 70% of σ_w^2 is caused by the variance in x_1, 20% by x_2, and 10% by the interaction between x_1 and x_2. We can prove the following variance decomposition into a sum of 2^{k-1} components:

$$\sigma_w^2 = \sum_{j=1}^{k} \sigma_j^2 + \sum_{j<j'}^{k} \sigma_{j;j'}^2 + \ldots + \sigma_{1;\ldots;k}^2 \tag{52}$$

with the main-effect variance $\sigma_j^2 = \mathrm{Var}[E(w|x_j)]$ and the two-factor interaction variance $\sigma_{j;j'}^2 = \mathrm{Var}[E(w|x_j, x_{j'})]$ etc., ending with the k-factor interaction variance $\sigma_{1;\ldots;k}^2 = \mathrm{Var}[E(w|x_1,\ldots,x_k)]$. Note that $\mathrm{Var}[E(w|x_1,\ldots,x_k)]$ denotes the variance of the mean of w if all k inputs are fixed; consequently, this variance equals the intrinsic noise in stochastic simulation.

The measure σ_j^2 leads to the *first-order sensitivity index* or the *main effect index* $\gamma_j = \sigma_j^2/\sigma_w^2$. So, γ_j quantifies the effect of varying x_j alone—averaged over the variations in all the other $k-1$ inputs; σ_w^2 in the denominator standardizes γ_j to provide a fractional contribution (in linear regression we standardize x_j so that β_j measures the relative main effect; see (6)). Likewise, $\sigma_{j;j'}^2$ through $\sigma_{1;\ldots;k}^2$ are divided by σ_w^2. Altogether we get

$$\sum_{j=1}^{k} \gamma_j + \sum_{j=1}^{k-1} \sum_{j'=j+1}^{k} \gamma_{j;j'} + \ldots + \gamma_{1;\ldots;k} = 1. \tag{53}$$

As k increases, the number of measures in (52) or (53) increases dramatically. So—as we assumed for regression metamodels—we may assume that only γ_j—and possibly $\gamma_{j;j'}$—are important, and verify whether they sum up to a fraction "close enough" to 1 in (53). The *estimation* of the various sensitivity measures may use LHS, and replace the simulation model by a Kriging metamodel; see Borgonovo and Plischke (2015) and Saltelli et al. (2008, pp. 164- 67).

5.6 Risk analysis (RA) or uncertainty analysis (UA)

In FANOVA we assume a given distribution for $\mathbf{x} \in \mathbb{R}^k$ with $w = f_{\text{sim}}(\mathbf{x})$. In RA we may wish to estimate $P(w > c)$ with a given threshold value c. RA is applied in nuclear engineering, finance, water management, etc. Actually, $P(w > c)$ may be very small—so $w > c$ is called a *rare event*—but may have disastrous consequences. An example of RA may be w denoting the net present value (NPV) and $\mathbf{x}$ the cash flows so $\mathbf{x}$ is sampled from given distribution functions; spreadsheets are popular software for such NPV computations. The uncertainty about the exact values of $\mathbf{x}$ is called *subjective* or *epistemic*, whereas the "intrinsic" uncertainty in stochastic simulation is called *objective* or *aleatory*.

SA and RA address different questions; namely, "Which are the most important inputs in the simulation model of a given real system?" and "What is the probability of a given (disastrous) event happening?". So, SA may identify those inputs for which the distribution in RA needs further refinement. RA and SA are also detailed in Borgonovo and Plischke (2015).

Methodologically, we propose the following method for RA aimed at estimating $P(w > c)$. We use a Monte Carlo method to sample input combination $\mathbf{x}$ from its given distribution. Next we use this $\mathbf{x}$ as input into the given simulation model. We run the simulation model to transform $\mathbf{x}$ into w, which is called *propagation of uncertainty*. We repeat these steps n times to obtain the EDF of w. Finally, we use this EDF to estimate $P(w > c)$.

In *expensive* simulation, we do not run n simulation runs, but run its *metamodel* n times. The true $P(w > c)$ may 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.

The British community *Managing uncertainty in complex models* (MUCM) also studies uncertainty in simulation models, including RA, and SA. Chevalier et al. (2014) uses Kriging to estimate the *excursion set* defined as the set of inputs that give 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* $P(w > c)$. Kleijnen et al. (2011) uses a first-order polynomial to estimate which combinations of uncertain inputs form the frontier that separates acceptable and unacceptable outputs; both aleatory and epistemic uncertainty are included.

RA is related to the *Bayesian* approach assuming the parameters of the simulation model to be unknown with a given *prior* distributions for these parameters—usually, *conjugate* priors. After obtaining simulation I/O data, this approach computes the posterior distribution 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. Frequentist RA, however, has been applied many more times in practice. We also refer to the specific Bayesian approach in Xie et al. (2014).

6 Simulation optimization

In *practice*, optimization of real-world systems is important. We also emphasize the crucial role of *uncertainty* in the input data for simulation models, which implies that *robust* optimization is important. In *academic research*, the importance of optimization is demonstrated by the many publications on this topic.

The simplest optimization has no constraints for the inputs or the outputs, has no uncertain inputs, and concerns the expected value of a single output, $E(w)$. This $E(w)$ may represent the probability of a binary variable: $P(w = 1) = p$ and $P(w = 0) = 1 - p$ imply $E(w) = p$. However, $E(w)$ excludes quantiles and the mode of the output distribution. Furthermore, the simplest optimization assumes continuous inputs, excluding *ranking and selection* (R&S) and *multiple comparison* procedures.

There are so many optimization methods that we do not try to summarize these methods. Instead, we focus on optimization using metamodels, especially linear regression and Kriging. Jalali and Van Nieuwenhuyse (2015) claims that metamodel-based optimization is "relatively common" and that RSM is the most popular metamodel-based method, while Kriging is popular in theoretical publications. Furthermore, we focus on *expensive* simulations; for these simulations, it is impractical to apply optimization methods such as *OptQuest*.

In many applications, a single simulation run is computationally inexpensive, but there are extremely many input combinations. In practice, most simulation models have a high k: *curse of dimensionality*; e.g., $k = 7$ with 10 values per input requires 10^7 combinations. Moreover, a single run may be expensive if we wish to estimate the steady-state performance of a queueing system with a high traffic rate. Finally, if we wish to estimate a small $E(w) = p$ (e.g. $p = 10^{-7}$), then we need to simulate extremely many customers (unless we succeed in applying importance sampling).

6.1 Linear regression for optimization: RSM

RSM treats the simulation model as a *black box*. RSM is *sequential*; i.e., it uses a sequence of local experiments that is meant to lead to the optimum input combination. RSM has gained a good track record; see Kleijnen (2015), Law (2015, pp. 656–679). and Myers et al. (2009).

We assume that RSM is applied only after the important inputs and their experimental area have been identified; i.e., before RSM starts, we may need *screening*; see Section 4. However, Chang et al. (2014) integrates RSM and screening.

Methodologically, the goal of RSM is to minimize $E(w|\mathbf{z})$ where $\mathbf{z}$ denotes the vector with the k *original* (nonstandardized) inputs. To initialize RSM, we select a starting point; e.g., the combination currently used in practice. In the *neighborhood* of this starting point we fit a first-order polynomial, assuming white noise; however, in a next step, RSM allows $\mathrm{Var}(w)$ to change. Unfortunately, there are no general guidelines for determining the appropriate *size* of the local area in each step; Chang et al. (2014), however, selects this size through a so-called trust region. To fit a first-order polynomial, we use a *R-III design*; see Section 2.1. In the next steps, we fit local first-order *polynomials.* In each of these steps, we use the *gradient* implied by the first-order polynomial fitted in that step: $\nabla(\widehat{y}) = \beta_{-0}$ where -0 means that the intercept β_0 is removed from β. This $\nabla(\widehat{y})$ estimates the *steepest descent* direction. We take a step in the steepest-descent direction, trying intuitively selected values for the step size. After a number of steps in the steepest-descent direction, the output will increase instead of decrease because the first-order polynomial is only a local approximation. When such deterioration occurs, we simulate the $n > k$ combinations of the R-III design that is now centered around the best combination found so far. To quantify the *adequacy* of the local first-order polynomial, we may compute R^2 or cross-validation statistics; see Section 3.5. Obviously, a *plane* implied by a first-order polynomial cannot adequately represent a *hill top* when searching to maximize the output or—equivalently—minimize the output. So, in the neighborhood of the estimated optimum, we fit a *second-order* polynomial using a CCD. Next we use the derivatives of this polynomial to estimate the optimum. We may also apply *canonical analysis* to examine the shape of the optimal subregion: unique minimum, saddle point, or ridge with stationary points? If time permits, then we may try to escape from a possible local minimum and *restart* the search from a different initial local area.

We recommend not to eliminate inputs with *nonsignificant* effects in a local first-order polynomial: these inputs may become significant in a next local area. The selection of m_i is a moot issue; see Section 3.3 and the SPRT for SB in Section 4.2. A higher-order polynomial is more accurate than a lower-order polynomial, but may give a predictor $\widehat{y}$ with lower bias but higher variance so its MSE increases; moreover, a higher-order polynomial requires the simulation of many more input combinations.

Assuming $m_i = m$ and CRN, we may compute $\widehat{\beta}_r$ ($r = 1, ..., m$). So, replication r gives $\nabla(\widehat{y})$—if a first-order polynomial is used—or the optimum—if a second-order polynomial is used. If we find the estimated accuracy too low, then we may simulate additional replications so m increases. We may also use either parametric or distribution-free bootstrapping to derive CIs for $\nabla(\widehat{y})$ and the optimum.

Kleijnen (2015) also discusses the *scale-independent* adapted steepest-descent direction that accounts for $\mathbf{\Sigma}_{\widehat{\beta}}$. Experimental results imply that this direction performs better than the classic steepest-descent direction.

In practice, simulation models have *multiple* (say, r) responses types. The RSM literature offers several approaches for such situations, but we focus on *generalized RSM* (GRSM); see Angün (2004). GRSM addresses the following *constrained nonlinear random optimization problem*:

$$\begin{array}{c} \min_{\mathbf{x}} E(w_0|\mathbf{z}) \\ E(w_{h'}|\mathbf{z}) \geq c_h \ (h' = 1, \ldots, r-1) \\ l_j \leq z_j \leq u_j \text{ with } j = 1, \ldots, k. \end{array} \tag{54}$$

GRSM combines RSM and mathematical programming, avoiding creeping along the boundary of the feasible area that is determined by the constraints on the random outputs and the deterministic inputs. So, GRSM moves faster to the optimum than steepest descent. GRSM is scale independent. For details we refer to Kleijnen (2015).

Obviously, it is uncertain whether the optimum estimated by GRSM is close to the true optimum. The first-order necessary optimality conditions are known as the *Karush-Kuhn-Tucker* (KKT) conditions. The KKT conditions may be tested through parametric bootstrapping; see Bettonvil et al. (2009).

6.2 Kriging metamodels for optimization

Efficient global optimization (EGO) is a well-known *sequential* method that uses Kriging to balance *local* and *global* search; i.e., it balances *exploitation* and *exploration.* When EGO selects a new combination $\mathbf{x}_0$, it estimates the maximum of the *expected improvement* (EI) comparing $w(\mathbf{x}_0)$ and—in minimization— $\min_i$ $w(\mathbf{x}_i)$ with $i = 1$, ..., n. We saw below (45) that $s^2\{\widehat{y}(\mathbf{x}_0)\}$ increases as $\mathbf{x}$ lies farther away from $\mathbf{x}_i$. So, EI reaches its maximum if either $\widehat{y}$ is much smaller than min $w(\mathbf{x}_i)$ or $s^2\{\widehat{y}(\mathbf{x}_0)\}$ is large so $\widehat{y}$ shows much uncertainty. We present only *basic* EGO for *deterministic* simulation; also see the classic EGO reference, Jones et al. (1998).

Note: There are many *variants* for deterministic and random simulations, constrained optimization, multi-objective optimization including Pareto frontiers, the "admissible set" or "excursion set", robust optimization, estimation of a quantile, and Bayesian approaches; see the references in Kleijnen (2015).

We start with a pilot sample, typically selected through LHS. To the resulting I/O data $(\mathbf{X}, \mathbf{w})$, we fit a Kriging metamodel $y(\mathbf{x})$ Next we find $f_{\min} = \min_{1 \leq i \leq n}$ $w(\mathbf{x}_i)$. This gives

$$\text{EI}(\mathbf{x}) = E\left[\max\left(f_{\min} - y(\mathbf{x}), 0\right)\right]. \tag{55}$$

We can derive the following closed-form expression for its estimator:

$$\widehat{\text{EI}}(\mathbf{x}) = \left(f_{\min} - \widehat{y}(\mathbf{x})\right)\Phi\left(\frac{f_{\min} - \widehat{y}(\mathbf{x})}{s\{\widehat{y}(\mathbf{x}_0)\}}\right) + s\{\widehat{y}(\mathbf{x}_0)\}\phi\left(\frac{f_{\min} - \widehat{y}(\mathbf{x})}{s\{\widehat{y}(\mathbf{x}_0)\}}\right) \tag{56}$$

where Φ and ϕ denote the cumulative distribution function (CDF) and the PDF of the Gaussian variable with mean 0 and variance 1. Using (56), we find $\widehat{\mathbf{x}}_{opt}$, the estimate of $\mathbf{x}$ that maximizes $\widehat{\text{EI}}(\mathbf{x})$. (We may apply a global optimizer; a *local* optimizer is undesirable because $s\{\widehat{y}(\mathbf{x}_i)\} = 0$ so $\text{EI}(\mathbf{x}_i) = 0$. Alternatively, we use a set of candidate points selected through LHS.) Next we run the simulation with this $\widehat{\mathbf{x}}_{opt}$, and find $w(\widehat{\mathbf{x}}_{opt})$. Then we fit a new Kriging model to the augmented I/O data (Kamiński (2015) presents methods for avoiding re-estimation of the Kriging parameters). We update n and return to (56)—until we satisfy a stopping criterion; e.g., $\widehat{\mathrm{EI}}(\widehat{\mathbf{x}}_{opt})$ is "close" to 0.

Kleijnen et al. (2010) derives the heuristic *Kriging and integer mathematical programming* (KrIMP), addressing the problem already presented in (54), but now augmented with s constraints f_g for $\mathbf{x}$ while $\mathbf{x}$ must belong to the set of non-negative integers $\mathbf{N}$:

$$\begin{aligned} &\min_{\mathbf{x}} E(w_0|\mathbf{x}) \\ &E(w_{h'}|\mathbf{x}) \geq c_h \ (h' = 1, \ldots, r-1) \\ f_g(\mathbf{x}) &\geq c_g \ (g = 1, \ldots, s) \\ x_j &\in \mathbf{N} \ (j = 1, \ldots, d). \end{aligned} \tag{57}$$

To solve this problem, KrIMP combines (i) *sequentialized* DOE to specify the next combination; (ii) *Kriging* to analyze the resulting I/O data, and obtain explicit functions for $E(w_h|\mathbf{x})$ ($h = 0$, 1, ..., $r-1$); (iii) *integer nonlinear programming* (INLP) to estimate the optimal solution from these explicit Kriging models. Experiments with KrIMP and OptQuest suggest that KrIMP requires fewer simulated combinations and gives better estimated optima.

6.3 Robust optimization (RO)

Robustness is crucial in today's uncertain world. The optimum solution for the decision variables—that we may estimate through RSM or Kriging—may turn out to be inferior when ignoring uncertainties in the noncontrollable environmental variables; i.e., these uncertainties create a *risk*.

Taguchians emphasize that in practice some inputs of a manufactured product are under complete control of the engineers, whereas other inputs are not. In simulation, $\widehat{\mathbf{x}}_{opt}$ may be completely wrong when we ignore uncertainties in some inputs. Taguchians therefore distinguish between (i) *decision variables*, which we now denote by $\mathbf{d} = (d_1, ..., d_k)'$, and (ii) *environmental inputs* or *noise factors* $\mathbf{e} = (e_1, ..., e_c)'$; in this section we denote the residual by ϵ instead of e.

Note: The goal of RO is the design of robust products or systems, whereas the goal of *risk analysis* is to quantify the risk of a given engineering design; that design may turn out to be not robust at all. For example, Kleijnen and Gaury (2003) simulates production-management, using RSM assuming a specific—namely the most likely—combination of e-values. Next, the robustness of $\widehat{\mathbf{x}}_{opt}$ is estimated when $\mathbf{e}$ changes; technically, $\mathbf{e}$ is generated through LHS. In RO, however, we wish to find a solution that—from the start of the analysis—accounts for all possible environments, including their likelihood; i.e., whereas Kleijnen and Gaury (2003) performs an *ex post* robustness analysis, we wish to perform an *ex ante* analysis.

Taguchians assume a single output (say) w, focusing on its mean μ_w and its variance caused by the noise factors $\mathbf{e}$ so $\sigma^2(w|\mathbf{d}) > 0$. These two outputs are combined in a *scalar loss function* such as the *signal-to-noise* or *mean-to-variance* ratio μ_w/σ^2_w; also see Myers et al. (2009, pp. 486-488). We, however, use μ_w and σ^2_w separately; i.e., given a threshold T, we try to solve:

$$\min E(w|\mathbf{d}) \text{ such that } \sigma(w|\mathbf{d}) \leq T; \tag{58}$$

also see Myers et al. (2009, pp. 488-495).

The Taguchian worldview is successful in production engineering, but statisticians criticize the statistical techniques. Moreover—compared with real-life experiments—simulation experiments have many more inputs, input values, and input combinations. Myers et al. (2009, pp. 502-506) combines the Taguchian worldview with the statisticians' RSM. Whereas Myers et al. (2009) assumes $\mathbf{\Sigma_e} = \boldsymbol{\sigma}^2_e\mathbf{I}$, w assume a general $\mathbf{\Sigma_e}$. Whereas Myers et al. (2009) superimposes contour plots for $E(w|\mathbf{d})$ and $\sigma(w|\mathbf{d})$ to find $\widehat{\mathbf{x}}_{opt}$, we use MP. This MP, however, requires specification of T in (58). Unfortunately, managers may find it hard to select a specific value for T, so we may try different T-values and estimate the corresponding *Pareto-optimal* efficiency frontier. To estimate the variability of this frontier resulting from the estimators of $E(w|\mathbf{d})$ and $\sigma(w|\mathbf{d})$, we may use bootstrapping.

More precisely, Myers et al. (2009) fits a *second-order polynomial* for $\mathbf{d}$ that is to be optimized; possible effects of $\mathbf{e}$ are modelled through a first-order polynomial; control-by-noise two-factor interactions are also considered:

$$\begin{aligned} y &= \beta_0 + \sum_{j=1}^{k}\beta_j d_j + \sum_{j=1}^{k}\sum_{j'\geq j}^{k}\beta_{j;j'} d_j d_{j'} + \sum_{g=1}^{c}\gamma_g e_g + \sum_{j=1}^{k}\sum_{g=1}^{c}\delta_{j;g} d_j e_g + \epsilon \\ &= \beta_0 + \boldsymbol{\beta}'\mathbf{d} + \mathbf{d}'\mathbf{B}\mathbf{d} + \boldsymbol{\gamma}'\mathbf{e} + \mathbf{d}'\mathbf{\Delta}\mathbf{e} + \epsilon. \end{aligned} \tag{59}$$

If $E(\mathbf{e}) = \boldsymbol{\mu}_\mathbf{e}$ and $E(\epsilon) = 0$, then (59) implies

$$\mu_y = \beta_0 + \boldsymbol{\beta}'\mathbf{d} + \mathbf{d}'\mathbf{B}\mathbf{d} + \boldsymbol{\gamma}'\boldsymbol{\mu}_\mathbf{e} + \mathbf{d}'\mathbf{\Delta}\boldsymbol{\mu}_\mathbf{e}. \tag{60}$$

Given $\mathbf{\Sigma_e}$, 59) implies

$$\sigma^2_y = (\boldsymbol{\gamma}' + \mathbf{d}'\mathbf{\Delta})\mathbf{\Sigma_e}(\boldsymbol{\gamma} + \mathbf{\Delta}'\mathbf{d}) + \sigma^2_\epsilon = \mathbf{l}'\mathbf{\Sigma_e}\mathbf{l} + \sigma^2_\epsilon \tag{61}$$

where $\mathbf{l} = (\boldsymbol{\gamma} + \mathbf{\Delta}'\mathbf{d})$ is the *gradient* $(\partial y/\partial e_1, \ldots, \partial y/\partial e_c)'$. Obviously, the larger the gradient's elements are, the larger σ^2_y is. Furthermore, if $\mathbf{\Delta} = \mathbf{0}$, then we cannot control σ^2_y through $\mathbf{d}$.

To estimate the parameters in (59), Taguchians usually apply a *crossed design*, which combines the design or *inner array* for $\mathbf{d}$ with n_d combinations and the design or *outer array* for $\mathbf{e}$ with n_e combinations so the crossed design has $n_d \times n_e$ combinations. Analogously, we combine a CCD for $\mathbf{d}$ and a R-III design for $\mathbf{e}$; see Section 2.4 and Section 2.1, respectively. Obviously, this combined design enables estimation of $\mathbf{\Delta}$.

Reformulating (59) as $y = \boldsymbol{\zeta}'\mathbf{x} + \epsilon$ with the q-dimensional vector $\boldsymbol{\zeta} = (\beta_0, ..., \delta_{k;c})'$ and $\mathbf{x}$ defined corresponding with $\boldsymbol{\zeta}$, gives

$$\widehat{\zeta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{w} \text{ and } \mathbf{\Sigma}_{\widehat{\boldsymbol{\zeta}}} = (\mathbf{X}'\mathbf{X})^{-1}\sigma^2_w \tag{62}$$

where σ_w^2 equals σ_ϵ^2 because we assume $\epsilon \sim \mathrm{NIID}(0, \sigma_\epsilon^2)$ in (59). To estimate μ_y in (60), we simply plug $\widehat{\zeta}$ into (60) with known $\mathbf{d}$ and $\boldsymbol{\mu}_\mathbf{e}$. To estimate σ_y^2 we also plug $\widehat{\zeta}$ into (61) with known $\mathbf{\Sigma}_\mathbf{e}$. Obviously, this plugging-in of $\widehat{\zeta}$ creates bias, which we ignore. Next we solve (58), using (for example) Matlab's *fmincon*. The resulting $\widehat{\mathbf{d}}_{opt}$ implies corresponding values for μ_w and σ_w^2. Examples are presented in Myers et al. (2009) and also in Dellino et al. (2012) and Yanikoğlu et al. (2015).

Dellino et al. (2012) combines the Taguchian world view and *Kriging*, so this approach replaces the polynomial in (59) by OK metamodels. Moreover, Dellino et al. uses bootstrapping to quantify the variability of the estimated Kriging metamodels. Dellino et al. again combines Kriging and NLP. Changing T in the NLP-model (58) enables estimation of the Pareto frontier.

To estimate the Kriging models for $E(w|\mathbf{d})$ and $\sigma(w|\mathbf{d})$ in (58), Dellino et al. proposes the following two approaches: (i) Fit one Kriging model for $E(w|\mathbf{d})$ and one Kriging model for $\sigma(w|\mathbf{d})$—estimating both models from the same *simulation* I/O data. (ii) Fit one Kriging model to a relatively small number n of combinations of $\mathbf{d}$ and $\mathbf{e}$, and use this metamodel to compute *predictions* for w for $N \gg n$ combinations of $\mathbf{d}$ and $\mathbf{e}$, accounting for the distribution of $\mathbf{e}$. We detail these approaches, as follows.

In approach (i) we select the n_d combinations of $\mathbf{d}$ *space-filling*; e.g., we use LHS. The n_e combinations of $\mathbf{e}$, however, we *sample* from the distribution of $\mathbf{e}$. If we have no prior information about the likelihood of specific values for $\mathbf{e}$, then we use independent uniform distributions for each element of $\mathbf{e}$. To sample $\mathbf{e}$, we may again use LHS. The result is an $n_d \times n_e$ crossed design specifying the $k+$ c simulation inputs. Running the simulation with these $n_d \times n_e$ input combinations gives $w_{i;j}$, which give

$$\overline{w}_i = \frac{\sum_{j=1}^{n_e} w_{i;j}}{n_e} \text{ and } s_i^2 = \frac{\sum_{j=1}^{n_e} (w_{i;j} - \overline{w}_i)^2}{n_e - 1} \text{ with } i = 1, ..., n_d. \tag{63}$$

The estimators in (63) are unbiased, as they do not use any metamodels.

In approach (ii) we select $n \ll n_d \times n_e$ combinations of the $k + c$ inputs d_j and e_g through a space-filling design (using e.g. max-min LHS). Next, we use this $n \times (k+c)$ matrix as simulation input, and obtain the n-dimensional vector with simulation outputs w. To these I/O simulation data we fit a Kriging model, which approximates w as a function of d_j and e_g. Finally, we use a design with $N \gg n$ combinations, crossing a space-filling design with N_d combinations of $\mathbf{d}$ and LHS with N_e combinations of $\mathbf{e}$ accounting for the distribution of $\mathbf{e}$. We use the Kriging model to compute the predictors $\widehat{y}$ of the $N_d \times N_e$ simulation outputs. We then compute the N_d conditional means $\overline{w}_i$ and standard deviations s_i^2 using (63) replacing n_d and n_e by N_d and N_e and replacing w by $\widehat{y}$. We use these $\overline{w}_i$ and s_i^2 $(i = 1, ..., N_d)$ to fit one Kriging model for $\overline{w}_i$ and one for s_i.

Finally, we summarize *Ben-Tal et al.*'s RO. If MP ignores the uncertainty in the coefficients of the MP model, then the resulting *nominal solution* may easily violate the constraints in the given model. RO may give a slightly worse value for the goal variable, but RO increases the probability of satisfying the constraints; i.e., a robust solution is "immune" to variations of the variables within the *uncertainty set* U. Given historical data on $\mathbf{e}$, Yanikoğlu et al. (2013) derives a specific U for $\mathbf{p}$ where $\mathbf{p}$ denotes the unknown density function of $\mathbf{e}$ that is compatible with the historical data on $\mathbf{e}$. This type of RO develops a computationally tractable *robust counterpart* of the original problem. RO may give better worst-case performance and also better average performance than the nominal solutions give.

References

Angün, M.E. (2011), *Black box simulation optimization: generalized response surface methodology.* VDM Verlag Dr. Müller, Saarbrücken, Germany (also published by CentER Dissertation Series, Tilburg University, Tilburg, Netherlands, 2004)

Ankenman, B., B. Nelson, and J. Staum (2010), Stochastic Kriging for simulation metamodeling. *Operations Research*, 58, no. 2, pp. 371–382

Bekki, J.M., J W Fowler, G T Mackulak and M Kulahci (2009), Simulation-based cycle-time quantile estimation in manufacturing settings employing non-FIFO dispatching policies. *Journal of Simulation*, June 2009, Volume 3, Number 2, pp. 69–128

Bettonvil, B.W.M., E. Del Castillo, and J.P.C. Kleijnen (2009). Statistical testing of optimality conditions in multiresponse simulation-based optimization. *European Journal of Operational Research*, 199, no. 2, pp. 448–458

Bischl, B., O. Mersmann, H. Trautmann, and C. Weihs (2012), Resampling methods for meta-model validation with recommendations for evolutionary computation. *Evolutionary Computation*, 20, no. 2, pp. 249–275

Borgonovo, E. and E. Plischke (2015), Sensitivity analysis: a review of recent advances. *European Journal of Operational Research*, in press

Chang, K.-H. M-K Li and H. Wan (2014), Combining STRONG with screening designs for large-scale simulation optimization, *IIE Transactions*, 46, no. 4, pp. 357–373

Chen, V.C.P., K.-L. Tsui, R.R. Barton, and M. Meckesheimer (2006), A review on design, modeling, and applications of computer experiments. *IIE Transactions*, 38, pp. 273–291

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

Chevalier, C., D. Ginsbourger, J. Bect, E. Vazquez, V. Picheny, and Y. Richet (2014), Fast parallel Kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, 56, no. 4, pp. 455–465

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

Fédou, J.-M. and M.-J. Rendas (2015), Extending Morris method: identification of the interaction graph using cycle-equitable designs. *Journal of Statistical Computation and Simulation*, 85, no. 7, pp. 1398-1419

Gordy, M.B. and S. Juneja (2010), Nested simulation in portfolio risk measurement. *Management Science*, 56, no. 11, pp. 1833–1848

Jalali H. and I. Van Nieuwenhuyse (2015), Simulation optimization in inventory replenishment: a classification. *IIE Transactions*, 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

Kamiński, B. (2015), A method for updating of stochastic Kriging metamodels. *European Journal of Operational Research*, accepted

Khuri, A.I. and S. Mukhopadhyay (2010), Response surface methodology. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2, pp. 128–149

Kleijnen, J. P. C. (2005), An overview of the design and analysis of simulation experiments for sensitivity analysis, *European Journal of Operational Research*, 164, no. 2, pp. 287–300

Kleijnen, J. P. C. (2009), Kriging metamodeling in simulation: a review. *European Journal of Operational Research*, 192, no. 3, pp. 707-716

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

Kleijnen, J.P.C. and D. Deflandre (2006), Validation of regression metamodels in simulation: bootstrap approach. *European Journal of Operational Research*, 170, no. 1, pp. 120–131

Kleijnen, J.P.C. and E.G.A. Gaury (2003), Short-term robustness of production-management systems: a case study. *European Journal of Operational Research*, 148, no. 2, pp. 452–465

Kleijnen, J.P.C., H. Pierreval, and J. Zhang (2011). Methodology for determining the acceptability of system designs in uncertain environments. *European Journal of Operational Research*, 209, no. 2, pp. 176–183

Kleijnen, J.P.C., W.C.M. Van Beers, and I. Van Nieuwenhuyse (2010). Constrained optimization in simulation: a novel approach. *European Journal of Operational Research*, 202, no. 1, pp. 164–174

Law, A.M. (2015), *Simulation modeling and analysis; fifth edition.* McGraw-Hill, Boston

Loeppky, J. L., Sacks, J., and Welch, W. (2009), Choosing the sample size of a computer experiment: a practical guide. *Technometrics*, 51, no. 4, pp. 366–376

Lophaven, S.N., H.B. Nielsen, and J. Sondergaard (2002), DACE: a Matlab Kriging toolbox, version 2.0. IMM Technical University of Denmark, Kongens Lyngby

Markiewicz, A. and A. Szczepańska (2007), Optimal designs in multivariate linear models. *Statistics & Probability Letters*, 77, pp. 426–430

Montgomery, D. C. (2009). *Design and analysis of experiments; 7th edition*, Wiley, Hoboken, NJ

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

Qu, H. and M.C. Fu (2014), Gradient extrapolated stochastic kriging. *ACM Transactions on Modeling and Computer Simulation*, 24, no. 4, 23:1–23:25

Qu, X. (2007), Statistical properties of Rechtschaffner designs. *Journal of Statistical Planning and Inference*, 137, pp. 2156–2164

Rasmussen, C.E. and C. Williams (2006), *Gaussian processes for machine learning*, The MIT Press, Cambridge, Massachusetts

Ryan, K.J. and D.A. Bulutoglu (2010), Minimum aberration fractional factorial designs with large N. *Technometrics*, 52, no. . 2, pp. 250–255

Saltelli, A., M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola (2008), *Global sensitivity analysis: the primer.* Wiley

Shi, W., J.P.C. Kleijnen, and Z. Liu (2014), Factor screening for simulation with multiple responses: sequential bifurcation. *European Journal of Operational Research*, 237, no. 1, pp. 136–147

Shrivastava, A.K. and Y. Ding (2010), Graph based isomorph-free generation of two-level regular fractional factorial designs. *Journal of Statistical Planning and Inference*, 140, pp. 169–179

Turner, A.J., S. Balestrini-Robinson, and D. Mavris (2013), Heuristics for the regression of stochastic simulations. *Journal of Simulation*, 7, pp. 229–239

Ulaganathan, S., I. Couckuyt, T. Dhaene, and E. Laermans (2014), On the use of gradients in Kriging surrogate models. *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, pp. 2692–270

Van Beers, W.C.M. and J.P.C. Kleijnen (2008), Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. *European Journal of Operational Research*, 186, no. 3,pp. 1099–1113

Wan, H., B.E. Ankenman, and B.L. Nelson (2010), Improving the efficiency and efficacy of controlled sequential bifurcation for simulation factor screening. *INFORMS Journal on Computing*, 22. no. 3, pp. 482–492

Wu, C.F.J. and M. Hamada (2009), *Experiments; planning, analysis, and parameter design optimization; second edition.* Wiley, New York

Xie, W., B.L. Nelson, and R.R. Barton (2014) A Bayesian framework for quantifying uncertainty in stochastic simulation. *Operations Research*, 62, no. 6, pp. 1439–1452

Yanikoğlu, İ., D. den Hertog, and J.P.C. Kleijnen (2015), Robust dual response optimization, *IIE Transaction*s, in press

Yin, J., S.H. Ng, and K.M. Ng (2009), A study on the effects of parameter estimation on Kriging model's prediction error in stochastic simulation, *Proceedings of the 2009 Winter Simulation Conference*, edited by M.D. Rossini, R.R. Hill, B. Johansson, A. Dunkin, and R.G. Ingalls, pp. 674–685

Yuan, J and S.H. Ng (2015), An integrated approach to stochastic computer model calibration, validation and prediction. *Transactions on Modeling and Computer Simulation*, 25 no. 3, article 18