Invited Review

Kriging metamodeling in simulation: A review

Jack P.C. Kleijnen *

Department of Information Management, Tilburg University, Post Box 90153, 5000 LE Tilburg, The Netherlands

Received 16 February 2007; accepted 5 October 2007
Available online 17 October 2007

Abstract

This article reviews Kriging (also called spatial correlation modeling). It presents the basic Kriging assumptions and formulas—contrasting Kriging and classic linear regression metamodels. Furthermore, it extends Kriging to random simulation, and discusses bootstrapping to estimate the variance of the Kriging predictor. Besides classic one-shot statistical designs such as Latin Hypercube Sampling, it reviews sequentialized and customized designs for sensitivity analysis and optimization. It ends with topics for future research.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Kriging; Metamodel; Response surface; Interpolation; Optimization; Design

I. Introduction

Metamodels are also known as response surfaces, surrogates, emulators, auxiliary models, etc. By definition, a metamodel is an approximation of the Input/Output (I/O) function that is implied by the underlying simulation model. Metamodels are fitted to the I/O data produced by the experiment with the simulation model. This simulation model may be either deterministic or random (stochastic). Note that simulation is applied in many different disciplines, so the terminology varies widely; therefore, this article gives several terms for the same concept.

Examples of deterministic simulation are models of airplanes, automobiles, TV sets, and computer chips—applied in Computer Aided Engineering (CAE) and Computer Aided Design (CAD) at Boeing, General Motors, Philips, etc. Detailed examples are the helicopter test example in [4], the vehicle safety example in [18], and the other examples in [30,47].

Deterministic simulations give the same output for the same input. However, deterministic simulations may show numerical inaccuracies; i.e., a minor (infinitesimal) change in the input produces a major change in the output. A well-known mathematical example is the inversion of a matrix that is nearly singular (ill-conditioned). Simulation examples are discussed in [12,15,16,50]. These inaccuracies may make deterministic simulation related to random simulation.

Random simulations use Pseudo-Random Numbers (PRNs) inside their models, so simulations of the same input combination give different outputs (unless, the PRN streams are identical; i.e., the PRN seeds are identical). Examples are models of logistic and telecommunication systems. Details are given in textbooks on discrete-event simulation, such as [1,32]. This article covers both deterministic and random simulation!

Most publications on metamodels focus on low-order polynomial regression. This type of metamodel may be used for the explanation of the underlying simulation model’s behavior, and for prediction of the expected simulation output for combinations of input values that have not yet been simulated (inputs are also called factors; combinations are also called points or scenarios). The final goals of metamodeling may be Validation and Verification (V&V) of the simulation model, sensitivity or what-if analysis of that model, and optimization of the simulated system; see [25,27,32].
This article focuses on Kriging metamodels. Typically, Kriging models are fitted to data that are obtained for larger experimental areas than the areas used in low-order polynomial regression; i.e., Kriging models are global (rather than local). These models are used for prediction; the final goals are sensitivity analysis and optimization.

Kriging was originally developed in geostatistics (also known as spatial statistics) by the South African mining engineer called Krige (who is still alive). The mathematics were further developed by Matheron; see his 1963 article [39]. A classic geostatistics textbook is Cressie’s 1993 book [9]. More recent are the references 17 through 21 in [38].

Later on, Kriging models were applied to the I/O data of deterministic simulation models. These models have k-dimensional input where k is a given positive integer (whereas geostatistics considers only two-dimensional input); see Sacks et al.’s classic 1989 article [43]. More recent publications are [24,44,49].

Only in 2003, Van Beers and Kleijnen [51] started with the application of Kriging to random simulation models. Although Kriging in random simulation is still rare, the track record that Kriging achieved in deterministic simulation holds great promise for Kriging in random simulation.

Note: Searching for ‘Kriging’ via Google on August 20, 2007 gave 661,000 hits, which illustrates the popularity of this mathematical method. Searching within these pages for ‘Operations Research’ gave 134,000 hits.

The goal of this article is to review the basics of Kriging, and some recent extensions. These basics may convince analysts in deterministic or random simulation of the potential usefulness of Kriging. Furthermore, the review of recent extensions may also interest those analysts who are already familiar with Kriging in simulation.

The rest of this article is organized as follows. Section 2 compares Kriging with linear regression, and covers the basic assumptions and formulas of Kriging. Section 3 presents some relatively new results, including Kriging in random simulation and estimating the variance of the Kriging predictor through bootstrapping. Section 4 includes one-shot and sequential statistical designs for Kriging metamodels, distinguishing between sensitivity analysis and optimization. Section 5 presents conclusions and topics for future research.

2. Kriging versus linear regression

This section first highlights the differences between classic linear regression—especially low-order polynomial regression—and modern Kriging. This article focuses on a single (univariate, scalar) simulation output, because most published Kriging models also assume such output. In practice, a simulation model has multiple (multivariate, vector) output, but univariate Kriging may then be applied per output variable.

Assuming a single output, a general black-box representation of a simulation model is

\[ w = s(d_1, \ldots, d_k, r_0), \]

where

\[ w \text{ is the output of the underlying simulation model; } \]

\[ s(\cdot) \text{ denotes the mathematical function implicitly defined by the computer code implementing this simulation model;} \]

\[ d_j \text{ with } j = 1, \ldots, k \text{ is the } j\text{th input variable of the simulation program, so } D = (d_j) \text{ is the design matrix for the simulation experiment, with } i = 1, \ldots, n \text{ and } n \text{ the number of input combinations in that experiment (bold mathematical symbols denote matrixes or vectors); } \]

\[ r_0 \text{ is the vector of PRN seeds (} r_0 \text{ may be a scalar; } r_0 \text{ is not used by a deterministic simulation).} \]

Notice that D determines the original simulation input variables (say) z and the corresponding standardized (coded, scaled) regression variables x defined below (3). This D is usually standardized; e.g., a two-level (fractional) factorial has elements that are either -1 or +1; also see [26].

2.1. Linear regression

The first-order polynomial regression metamodel for (1) is

\[ y_{reg} = \beta_0 + \beta_1 d_1 + \ldots + \beta_k d_k + e_{reg}, \]

where

\[ y_{reg} \text{ is the regression predictor of the simulation output } w \text{ in (1) (the subscript reg distinguishes this metamodel from the Kriging metamodel presented below);} \]

\[ \beta = (\beta_0, \beta_1, \ldots, \beta_k)' \text{ is the vector with the parameters of this metamodel;} \]

\[ e_{reg} \text{ is the error (residual, noise)—which includes both lack of fit of the metamodel and intrinsic noise caused by the PRNs (this intrinsic noise vanishes in deterministic simulation).} \]

The model in (2) is a special case of the general linear regression model

\[ y_{reg} = X\beta + e_{reg}, \]

where

\[ y_{reg} \text{ denotes the } n\text{-dimensional vector with the regression predictor where } n \text{ is the number of simulated input combinations;} \]

\[ X = (x_{ij}) \text{ denotes the } n \times q \text{ matrix of explanatory standardized regression variables with } x_{ij} \text{ the value of variable } j \text{ in combination } i (i = 1, \ldots, n; j = 1, \ldots, q) \text{ (e.g., (2) has } q = 1 + k \text{ because (2) includes the dummy variable or constant } x_{i0} = 1 \text{ corresponding with } \beta_0). \]
\[ \beta = (\beta_1, \ldots, \beta_q)\] denotes the \( q \)-dimensional vector of regression parameters (if there is a dummy variable, then \( \beta_1 \) denotes the intercept in the general regression model, whereas the symbol \( \beta_0 \) denoted the intercept in the specific regression model (2));

\[ e_{\text{reg}} \] is the vector of residuals in the \( n \) input combinations.

Above, it was mentioned that \( D \) determines the standardized regression variables \( x \). Indeed, the first-order polynomial model (2) implies \( X = (1, D) \) where \( 1 = (1, \ldots, 1) \) is an \( n \)-dimensional vector with each element being the number one.

The Least Squares (LS) estimator (say) \( \hat{\beta} \) of the regression parameter vector \( \beta \) in the linear regression model (3) can be derived to be

\[ \hat{\beta} = (XX)^{-1}Xw, \tag{4} \]

where \( w = (w_1, \ldots, w_n)^T \) is the \( n \)-dimensional vector with ‘the’ output of the simulation model with input \( D "; \) ‘the’ output of combination \( i \) is the average output \( \bar{w}_i \) of a constant number of replications \( m_i = m \):

\[ \bar{w}_i = \frac{\sum_{l=1}^{m_i} w_{li}}{m_i}. \tag{5} \]

In deterministic simulation, (5) has \( m = 1 \). If the number of replicates is not constant, then a slightly more complicated formulation is required; see [26]. Note that (by definition) replications give Independent and Identically Distributed (IID) simulation outputs, because they use non-overlapping replicates is not constant, then a slightly more complicated formulation is required; see [26]. Note that (by definition) replications give Independent and Identically Distributed (IID) simulation outputs, because they use non-overlapping PRN streams through the judicious selection of the vector of PRN seeds \( r_0 \).

Hence, the regression predictor for a simulation input (say) \( d = (d_1, \ldots, d_k) \) is

\[ \hat{y}_{\text{reg}}(d) = x(d)^T \hat{\beta} = x(d)^T (XX)^{-1}Xw, \tag{6} \]

where the vector of explanatory variables \( x \) is determined by the vector of simulation inputs \( d \); e.g., the first-order polynomial model (2) implies \( x(d) = (1, d_1, \ldots, d_k)^T \). The input \( d \) may be a ‘new’ or an ‘old’ combination; by definition, an ‘old’ combination is one of the rows in \( D \).

2.2. Kriging: Basics

This article focuses on the simplest type of Kriging called Ordinary Kriging, which assumes

\[ w(d) = \mu + \delta(d), \tag{7} \]

where

\( \mu \) is the simulation output averaged over the ‘experimental area’ (the ‘experimental area’ is called ‘the experimental frame’ in [55]; it might also be called the ‘domain of admissible scenarios’—given the goals of the simulation study);

\( \delta(d) \) is the additive noise that forms a ‘stationary covariance process’ with zero mean. A time series (say) \( w \) is a stationary covariance process if it has a constant mean (say) \( E[w_i] = \mu_i \), a constant variance \( \text{var}(w_i) = \sigma_i^2 \), and covariances depending only on the lag \( |t - t'| \); i.e., \( \text{cov}(w_i, w_{i'}) = \sigma_{ii'} \).

The Kriging model (7) with its constant \( \mu \) does not imply a flat response surface; see [43]. Instead of the constant \( \mu \), Universal Kriging uses a regression model. However, Ordinary Kriging often suffices in practice; see [7,37,38,43].

Kriging is used—quite successfully—in deterministic simulation. At first sight it seems strange that the random (meta)model (7) can be applied to a deterministic simulation model. A reasonable interpretation is that the deviations of the simulation output \( w \) from its mean \( \mu \) form a random process—with the characteristics of a ‘stationary covariance process’ (with zero mean); see \( \delta \) in (7).

For random simulation, a more suitable Kriging model augments (7) with a white noise term (say) \( \epsilon(d) \) that is independent of \( \delta(d) \):

\[ w(d) = \mu + \delta(d) + \epsilon(d), \tag{8} \]

where (by definition) \( \epsilon(d) \) has zero mean and constant (unknown) variance \( \sigma_\epsilon^2 \); see [44, p. 216]. The Kriging predictor is then no longer an exact interpolator (also see (19) below). This white noise assumption, however, is unrealistic: the output of a random simulation model has a variance that changes as the input changes. For example, the steady-state waiting time (output) of an M/M/1 model has a variance that is known to increase much more than its mean, as the traffic rate (input) increases. In practice, unbiased estimators of the heterogeneous variances of the simulation outputs can be easily computed from replicated simulation outputs:

\[ \text{var}(w_i) = \frac{\sum_{l=1}^{m_i} (w_{li} - \bar{w}_i)^2}{m_i - 1} \quad (i = 1, \ldots, n), \tag{9} \]

where \( w_{li} \) denotes the simulation output of replication \( r \) of input combination \( i \), and \( \bar{w}_i \) was defined in (5) for the special case \( m_i = m \); actually, (9) only requires \( m_i > 1 \). So, it might be claimed that a Kriging metamodel for a random simulation model should estimate \( \sigma_i^2 \) (\( i = 1, \ldots, n \)) through (9), and \( \sigma_\epsilon^2 \) through the so-called ‘nugget effect’ in the estimated variogram (the variogram is an alternative formulation of the correlation function, and assumes a constant variance; see [9, p. 59] and (18) below). Unfortunately, current Kriging software does not enable such estimation. Moreover, the Kriging software that accounts for the white noise \( \epsilon(d) \) in (8) is not so well documented and not so widely available as standard Kriging software is; e.g., [44]’s computer program in C running under Unix or Linux—called ‘PEnK’ (an acronym for ‘Parametric Empirical Kriging’)—allows for white noise, whereas [35]’s ‘DACE’ (the acronym for ‘Design and Analysis of Computer Experiments’) is written in Matlab and is very well documented.
but does not allow for white noise. Moreover, preliminary experiments reported in \[53\] suggest that the prediction variance of PErK is not smaller than that of DACE. Also see \[9,29,40,46,51\].

Ordinary Kriging—from now on, briefly called Kriging—uses the following linear predictor:
\[
y(d) = \xi(d, D)/w(d) = \xi^* w,
\]
where the weights \(\xi(d, D)\)—abbreviated to \(\xi\)—are not constants (whereas \(\beta\) in (3) is) but decrease with the distance between the input \(d\) to be predicted and the ‘old’ points \(D\) (this decreasing relationship is detailed below); this \(D\) determines the simulation output vector \(w\), so the explicit notation is \(w(d)\) and the simpler notation is \(w\).

To select the optimal values for the weights \(\xi\) in (10), a criterion must be selected. In linear regression, the Sum of Squared Residuals is the criterion—which gives the LS estimator (4). Kriging selects the Best Linear Unbiased Predictor (BLUP), which (by definition) minimizes the Mean Squared Error (MSE) of the predictor \(y(d)\):
\[
\min MSE[y(d)] = \min[\mathbb{E}(y(d) - w(d))^2],
\]
where \(d\) may be any point in the experimental area. Moreover, this minimization must account for the condition that the predictor is unbiased:
\[
\mathbb{E}(y(d)) = E(w(d)),
\]
where in deterministic simulation \(E(w(d))\) may be replaced by \(w(d)\). It can be proven that the solution of the constrained minimization problem defined by (11) and (12) implies that the weights of the linear predictor (10) must satisfy the following condition:
\[
\sum_{i=1}^{n} \lambda_i = 1
\]
(13)
or (in matrix notation) \(1^T \xi = 1\). Furthermore, the optimal weights can be proven to have the values
\[
\lambda_i = \Gamma^{-1} \left[ \gamma + 1 \frac{1 - \Gamma_{i, i}^{-1}}{\Gamma_{i, j} - 1} \right],
\]
(14)
where
\[
\Gamma = (\text{cov}(w_i, w_j)) \quad \text{with} \quad i, j = 1, \ldots, n \quad \text{is the} \quad n \times n \quad \text{symmetric and positive semi-definite matrix with the covariances between the 'old' outputs (i.e., outputs of input combinations that have already been simulated);}
\]
\[
\gamma = (\text{cov}(w_i, w_0)) \quad \text{is the} \quad n \times 1 \quad \text{vector with the covariances between the} \quad n \quad \text{old outputs} \quad w_i \quad \text{and} \quad w_0, \quad \text{the output of the combination to be predicted (which may be either new or old).}
\]

Finally, it can be proven (also see [34]) that (7), (10) and (14) imply
\[
y(d) = \hat{\mu} + \gamma(d) \Gamma^{-1}(w - \hat{\mu} 1),
\]
where
\[
\hat{\mu} = (1 \Gamma^{-1} 1)^{-1} 1 \Gamma^{-1} w
\]
and \(d_0\) denotes the input of the output \(w_0\) that is to be predicted.

It is easy to see that the Kriging model in (7) implies \(E(y) = \mu\) because (7) implies \(E(w) = \mu 1\). Furthermore, if (say) \(w_1 > \mu, \quad w_2 = \mu, \quad \ldots, \quad w_n = \mu\), then the conditional expected value of the predictor in (15) exceeds the unconditional mean \(\mu\) (because each element of the row vector \(y' \Gamma^{-1}\) is positive).

Obviously, the optimal values for the Kriging weights in (14) depend on the covariances—or equivalently the correlations—between the simulation outputs in the Kriging model (7). Kriging assumes that these correlations are determined by the ‘distance’ between the inputs of the outputs \(w_j\) and \(w_l\) or \(w_i\) and \(w_0\)—or (more succinctly) between \(w_i\) and \(w_g\) with \(g = 0, 1, \ldots, n\).

In simulation applications of Kriging, the usual assumption is that the correlation function for a \(k\)-dimensional input vector is the product of \(k\) one-dimensional functions:
\[
\rho(w(d), w(d')) = \prod_{j=1}^{k} \rho(d_{ij}, d_{ij}).
\]
Moreover, the Kriging assumption of a stationary covariance process for \(\delta(d)\) in (7) implies that the correlations depend only on
\[
h_j(i, g) = |d_{ij} - d_{ij}| \quad \text{with} \quad (j = 1, \ldots, k) (i = 1, \ldots, n)
\]
(16)
\[
(g = 0, 1, \ldots, n).
\]
So, \(\rho(d_{ij}, d_{ij})\) in (16) reduces to \(\rho(h_j(i, g))\). Unfortunately, transforming the standardized design points \(d_j\) into the original simulation inputs \(z_j\) makes the distances scale dependent; also see [8].

There are several types of stationary covariance processes. Three popular types for a single input (so \(h = h\) in (17)) with parameter \(\theta > 0\) are:

- Linear correlation function: \(\rho(h) = \max(1 - \theta h, 0)\).
- Exponential correlation function: \(\rho(h) = \exp(-\theta h)\).
- Gaussian correlation function: \(\rho(h) = \exp(-\theta h^2)\) (its point of inflection can be proven to be \(1/\sqrt{2\theta}\)).

In Kriging, a popular correlation function is
\[
\rho(h) = \exp \left[- \sum_{j=1}^{k} \theta_j h_j^p \right] = \prod_{j=1}^{k} \exp[- \theta_j h_j^p],
\]
(18)
where
\[
\theta_j \quad \text{denotes the importance of input} \quad j; \quad \text{i.e., the higher} \quad \theta_j \quad \text{is, the less effect input} \quad j \quad \text{has;}
\]
\(p_j\) denotes the smoothness of the correlation function; e.g., \(p_j = 2\) implies an infinitely differentiable function.
Obviously, exponential and Gaussian correlation functions have $p = 1$ and $p = 2$ respectively.

Correlation functions that decrease as the distance increases, imply that the optimal weights are relatively high for inputs close to the input to be predicted. Furthermore, some of the weights may be negative. Finally, the weights imply that for an ‘old’ input the predictor equals the observed simulation output at that input:

$$y(d_i) = w(d_i) \quad \text{if} \quad d_i \in D.$$  \hspace{1cm} (19)

so all weights are zero except the weight of the observed output; i.e., the Kriging predictor is an exact interpolator. Note that the regression predictor minimizes the Sum of Squared Residuals (SSR), so it is not an exact interpolator—unless $n = q$ (where $q$ was defined below (3)).

A major problem is that the optimal Kriging weights $\lambda_i$ in (14) depend on the correlation function of the underlying simulation model—but this correlation function is unknown. Therefore both the type and the parameter values must be estimated. (The number of observations for estimating a covariance for a given distance $h$ decreases as that distance increases.) Given these estimates for various values of the distance $h$, a correlation function is fitted. To estimate the parameters of such a correlation function, the standard software and literature uses Maximum Likelihood Estimators (MLEs). A MLE requires constrained maximization. This optimization is a hard problem, because matrix inversion is necessary, multiple local maxima may exist, etc.; see [36,38]. (Besides the MLE criterion, [38] uses cross-validation; for the linear correlation function, [28] uses the LS criterion.)

The estimation of the correlation functions and the corresponding optimal weights is made easy through DACE, which is well documented and free of charge; see [35]. Alternative free software is available via http://www.stat.ohio-state.edu/~comp_exp/ and http://endo.sandia.gov/Surpack.

If the number of simulation inputs does not exceed three, then geographical Kriging software can also be applied. An example of commercial geographical software is Isatis; see http://www.geovariances.com/.

Unfortunately, DACE uses lower and upper limits for $\theta_j$ (the correlation parameters in (18)), which the analysts usually find hard to specify. Different limits may give completely different MLEs $\hat{\theta}_j$; see the examples in [34].

Note that there are also many publications that interpret Kriging models in a Bayesian way; see [20,36], and Section 4.3.

3. Kriging: New results

This section first summarizes some new results for Kriging applied in random (not deterministic) simulation. Next, it discusses the problems caused by the estimation of the optimal Kriging weights.

3.1. Kriging in random simulation

The interpolation property in (19) is attractive in deterministic simulation, because the observed simulation output is unambiguous (ignoring numerical noise, discussed in Section 1). In random simulation, however, the observed output is only one of the many possible values. For random simulations, [51] replaces $w(d_i)$ in (11) by the average observed output $\bar{y}_i$; see (5). These averages, however, are still random, so the property in (19) loses its intuitive appeal. Nevertheless, [51] gives examples in which the Kriging predictions based on (5) are much better than the regression predictions (regression metamodels may be useful for other goals such as understanding, screening, and validation). This problem was also discussed above; see (8).

The Kriging model in (7) assumes a stationary covariance process, which implies a constant variance (say) $\sigma^2_{ij}$. However, in experiments with random simulation models such as queueing models, the output variances $\text{var}(w_i)$ are not constant at all. Fortunately, [29] demonstrates that the Kriging model is not very sensitive to this variance heterogeneity; also see [45].

3.2. Kriging with estimated weights

The Kriging literature virtually ignores the problems caused by replacing the weights $\lambda$ in (10) by the estimated optimal weights (say) $\hat{\lambda}_0$. This replacement implies that the Kriging predictor becomes a nonlinear estimator. The literature uses the predictor variance—given the Kriging weights $\lambda$. At a fixed point $d$, this variance follows directly from (14) (also see [9, p. 122]):

$$\text{var}[y(d)|\lambda] = 2 \sum_{i=1}^{n}(\lambda_i \text{cov}(w_0, w_i) - \sum_{j=1}^{n} \sum_{i=1}^{n} \lambda_j \lambda_i \text{cov}(w_0, w_j)).$$  \hspace{1cm} (20)

Using (20), it is easy to derive that the variance in case the new point $w_0$ equals an old point $w_i$ reduces to zero. Ignoring the randomness of the estimated optimal weights tends to underestimate the true variance of the Kriging predictor. Moreover, the true (unconditional) variance and the conditional variance do not reach their maxima for the same input combination (which is an important characteristic when selecting the next design point; see the next section). To estimate the true variance, random simulation is discussed first; deterministic simulation next.

In random simulation, each input combination is replicated a number of times; also see (5) and (9). Therefore a simple method for estimating the true predictor variance is distribution-free bootstrapping. The basics of bootstrapping are explained in [11,26]. To estimate the predictor var-
variance in Kriging, [52] resamples—with replacement—the m_i IID observations. This sampling results in the bootstrapped average \( \bar{y}^* \) where the superscript * is the usual symbol to denote a bootstrapped observation \( (i = 1, \ldots, n) \). From these n bootstrapped averages \( \bar{y}^*_n \), the bootstrapped estimated optimal weights \( \hat{\lambda}^*_n \) and the corresponding bootstrapped Kriging predictor \( \hat{y}^* \) are computed. To decrease sampling effects, this whole procedure is repeated B times, which gives \( y^*_b \) with \( b = 1, \ldots, B \). The variance of the Kriging predictor is estimated from these B values, analogously to (9).

Note: In a quick-and-dirty experiment (that was not published), Van Beers (Tilburg University) simulated the M/M/1 with and without Common Random Numbers (CRN), and obtained the estimated correlation parameter \( \theta \) in the Gaussian correlation function \( \rho(h) = \exp(-\theta h^2) \). When applying CRN, the simulation outputs at different traffic rates were stronger correlated. So it is to be expected that the correlation function decreases slower or its parameter \( \theta \) becomes smaller. Indeed, the experiment gave \( 0.05 < \theta < 0.10 \) in five macro-replicates with CRN, and \( 11 < \theta < 16 \) without CRN.

For deterministic simulation, [10] applies parametric bootstrapping, assuming a Gaussian stationary covariance process and Gaussian correlation functions. The ‘true’ parameter values \( \theta \) of this process are estimated from the given simulation I/O data, \( (D, w) \). Next, the Monte Carlo method is used to sample bootstrapped data \( (D, w^*_n) \) from the estimated multivariate normal distribution. Alternative approaches (including cross-validation and Akaike’s Information Criterion) are discussed in [36,38].

4. Designs for Kriging

Simulation analysts often use Latin Hypercube Sampling (LHS) to generate the I/O simulation data to which they fit a Kriging model. Note that LHS was not invented for Kriging but for Risk Analysis; see [26].

LHS assumes that an adequate metamodel is more complicated than a low-order polynomial such as (2), which is assumed by classic designs such as fractional factorials. LHS, however, does not assume a specific metamodel or simulation model. Instead, LHS focuses on the design space formed by the k-dimensional unit cube defined by the k standardized simulation inputs \( 0 \leq d_{ij} \leq 1 \) with \( i = 1, \ldots, n \) and \( j = 1, \ldots, k \). LHS is one of the space-filling types of design. Other designs related to LHS (e.g., maximin designs and orthogonal arrays) are discussed in [26], including many websites and references; also see [5,17,19,31,48]; a Bayesian view is presented in [6].

An alternative for LHS and related designs are sequentialized designs. Sequential statistical procedures are known to be more ‘efficient’; i.e., they require fewer observations than fixed-sample (one-shot) procedures; see, e.g., [42]. Obviously, sequential designs imply that observations are analyzed—so the data generating process is better understood—before the next input combination is selected. This property implies that the design depends on the specific underlying process (simulation model); i.e., the design is customized (tailored or application-driven, not generic). Moreover, computer experiments (unlike many real-world experiments) proceed sequentially. Nevertheless, sequential procedures may be less efficient computationally; e.g., re-estimating the Kriging parameters may be costly (see [13]).

There are several approaches to the sequential design of simulation experiments. It is important to distinguish between two different goals of simulation experiments, namely sensitivity analysis and optimization (also see Section 1). This section first discusses (in some detail) a specific approach for the first goal; next it discusses a related approach for optimization; finally, it summarizes alternative approaches for sensitivity analysis and optimization.

4.1. Designs for sensitivity analysis

Sequential procedures for deterministic and random simulations are developed in [28,52] respectively. These two procedures share the following steps.

1. Start with a pilot experiment, using some small generic space-filling design (e.g., a LHS design). Its (say) \( n_0 \) combinations determine the inputs into the simulation model, which give the corresponding simulation outputs.
2. Fit a Kriging model to the I/O simulation data that are available at this step (in the first pass of this procedure, these I/O data are the data resulting from Step 1).
3. Consider (but do not yet simulate) a set of candidate input combinations that have not yet been simulated and that are selected through some space-filling design; select as the next combination to be actually simulated, the candidate combination that has the highest predictor variance (the estimation of this variance is further discussed below).
4. Use the combination selected in Step 3 as the input combination to the simulation model; run the (expensive) simulation, and obtain the corresponding simulation output.
5. Re-fit a Kriging model to the I/O data that is augmented with the I/O data resulting from Step 4.
6. Return to Step 3, until the Kriging metamodel is acceptable.

The resulting designs are customized; i.e., which combination has the highest predictor variance is determined by the underlying simulation model. For example, if the simulation model has an I/O function (response surface) that is a simple hyperplane within a subspace of the total experimental area, then this procedure selects relatively few points in that part of the input space.

This procedure is applied to two random simulation models that are building blocks for more realistic simulation models. The first model is an M/M/1 simulation with
a traffic rate \( d \) that varies over the experimental area \( 0.1 \leq d \leq 0.9 \). For comparison purposes, a one-shot LHS design is also applied, with \( n = 10 \) prefixed values for the traffic rate. Next the sequential procedure is applied, until the same number of observations as in the LHS design (namely 10) are simulated. For each simulated traffic rate, enough IID observations (namely, renewal cycles) are obtained to reach a prespecified accuracy (namely, a confidence interval with a prespecified type-I error rate, \( \alpha \) and a prespecified relative error); also see [32]. To estimate the Kriging predictor variance, distribution-free bootstrapping is applied to these IID observation (see Step 3 in the procedure presented above). This sequential design turns out to select more input values in the part of the input range that gives a drastically increasing (highly nonlinear) I/O function. This design gives better Kriging predictions than the fixed LHS design does—especially for small designs, which are used in expensive simulations.

The second random simulation is an \((s, S)\) inventory simulation. Again, in the ‘rough’ region of the response surface the procedure simulates more input combinations. The other experiment is with a deterministic simulation model, namely a fourth-degree polynomial I/O function in one input variable with two local maxima and three local minima; two of these minima are at the border of the experimental area. Now no bootstrapping is used to estimate the predictor variance in Step 3 of the procedure; instead, cross-validation and jackknifing are used (as explained below). This design is compared with another sequential design based on (20), which approximates the variance of the Kriging predictor ignoring the random character of the estimated weights. The latter design selects as the next point the input value that maximizes this variance; i.e., there is no need to specify candidate points. It turns out that this approach selects as the next point the input farthest away from the old inputs, so the final design spreads all its points evenly across the experimental area—like one-shot space-filling designs do.

Step 3 (of the sequential procedure described above) for deterministic simulation estimates the true predictor variance through cross-validation. So, one of the I/O observations already simulated is deleted from the original I/O data set \((D, w)\), which gives \((D_{-i}, w_{-i})\) where the subscript \(-i\) means that \((D_i, w_i)\) is deleted. Next, the Kriging predictor is recomputed from the recomputed correlation function parameters and the corresponding optimal Kriging weights. To quantify this predictor’s uncertainty, jackknifing is used. So, calculate the jackknife’s pseudovalue (say) \( J \) for candidate input combination (say) \( j \) as the weighted average of the original and the cross-validation predictors:

\[
J_{j,i} = n\hat{y}_j - (n - 1)\hat{y}_{j,-i} \quad \text{with } j = 1, \ldots, c \\
\text{and } i = 1, \ldots, n,
\]

where \( c \) denotes the number of candidate points and \( n \) the number of points that have really been simulated so far and are deleted successively. Use these pseudovalue to compute the classic variance estimator (analogsly to (9))

\[
\text{var}(\bar{J}_j) = \frac{\sum_{i=1}^n (J_{j,i} - \bar{J}_j)^2}{(n - 1)}.
\]

Like in the two random simulation examples, the procedure results in a design that favors input combinations in subareas that have more interesting I/O behavior; i.e., the final design selects relative few input values in the subareas that generate an approximately linear I/O function, whereas it selects many input values near the borders where the function changes much.

4.2. Designs for optimization

This article focuses on constrained optimization in random simulation models with (say) \( r \) multiple outputs. This may be formalized as follows:

\[
\min \mathbb{E}(w_0 | \mathbf{x}) \tag{21}
\]

such that the other \((r - 1)\) random outputs satisfy the constraints

\[
\mathbb{E}(w_0 | \mathbf{x}) \geq a_h \quad \text{for } h = 1, \ldots, r - 1. \tag{22}
\]

An example is the following inventory simulation. The expected sum of the inventory carrying costs and ordering costs should be minimized. The expected service percentage (or fill rate) should be at least (say) 90% so \( a_1 = 0.9 \) in (22). Constrained optimization is further discussed in [26].

Analogous to the sequential design procedure for sensitivity analysis, the procedure has the following steps in optimization.

1. Start with a pilot experiment, using some small space-filling design. Now, however, the comparison of the \((r - 1)\) simulation outputs with \( a_h \) in (22) reveals whether the input gives a feasible solution. In random simulation the noisy outputs can be compared with \( a_h \) through a \( t \) test:

\[
t_{m_{h-1}} = \frac{W_h - a_h}{\sqrt{\text{var}(w_i) / m_i}},
\]

where \( W_h \) and \( \text{var}(w_i) \) were defined in (5) and (9). An advantage of a space-filling design is that it may be expected that some outputs will turn out to be feasible; e.g., [3] finds that 10 of the 20 combinations of a LHS design for the optimization of an \((s, S)\) inventory system are feasible.

2. Fit a Kriging model to the I/O simulation data that is available at this step. Because DACE does not incorporate cross-correlations between the \( r \) random simulation outputs, univariate Kriging models are fitted in [3].

3. Apply some numerical search procedure for the solution of the Kriging approximation—namely, the \( r \)
Kriging models—to the original problem—namely, (21) and (22). Matlab’s ‘fmincon’ was applied in [3].
(4) This estimated optimal input combination is simulated next.
(5) Re-fit a Kriging model to the I/O data that is augmented with the I/O data resulting from Step 4.
(6) Test whether the estimated optimum (say) \( z^0 \) from Step 4 is indeed optimal. The first-order optimality conditions are formalized by the Karush–Kuhn–Tucker (KKT) conditions; see, e.g., [14, p. 81]:

\[
\begin{align*}
\mathbf{p}_{h=0} &= \sum_{h=0}^{\mathbf{A}(\mathbf{z})} \lambda_h \mathbf{p}_{h=0}, \\
\lambda_h &= 0, \\
h &\in A(\mathbf{z}),
\end{align*}
\]

where \( \mathbf{p}_{h=0} \) denotes the \( k \)-dimensional vector with the gradient of the goal function (21); \( A(\mathbf{z}) \) is the index set with the indices of the constraints that are binding at \( \mathbf{z} \); \( \lambda_h \) is the Lagrange multiplier for binding constraint \( h \); \( \mathbf{p}_{h=0} \) is the gradient of the output in that binding constraint. Note that (23) implies that the gradient of the objective can be expressed as a non-negative linear combination of the gradients of the binding constraints, at \( \mathbf{z}^0 \). DACE also gives the gradients of the fitted Kriging models! For random simulation, a statistical procedure to test whether (23) holds is presented in [2].

(7) If the KKT conditions do not hold, then further local experimentation is recommended—until the KKT conditions do hold.
(8) To escape from a local minimum, Step 3 may be repeated with a different starting point, followed by Step 4, etc.

4.3. Alternative designs

Let’s finally summarize alternative sequentialized and customized designs for sensitivity analysis and optimization in simulation experimentation—starting with the most recent publications.

- Reference [21] derives sequential designs for the optimization of random and deterministic simulation models, using Kriging and so-called Efficient Global Optimization (EGO), which maximizes the Expected Improvement (EI) following a Bayesian approach; also see [13,22,24,45]. These publications try to balance local and global search (e.g., using the EI criterion)—assuming a single simulation output (no constrained multiple outputs) and a Gaussian distribution for the stationary covariance process (instead of distribution-free bootstrapping, jackknifing, and cross-validation).

- Reference [47] reports on a panel discussion that also emphasizes the importance of sequential and adaptive sampling.

- Reference [34] uses a Bayesian approach to derive a sequential design based on prediction errors for the optimization of deterministic simulation models. That publication includes a number of interesting references. (Moreover, the related publication [33] uses a second metamodel to predict the predictor errors.)

- Reference [23] studies sequential designs for Kriging metamodels, using (20), which assumes known parameters of the underlying covariance process.

- Reference [54] uses a Bayesian approach to derive sequential IMSE (Integrated MSE) designs.

Note: Reference [41] estimates the 95% quantile of the output of a deterministic simulation model with uncertain inputs. Consequently there is no interest in the whole experimental area. This publication combines Kriging, the Bayesian approach, and two-stage (not fully sequential) sampling.

5. Conclusions and future research

This article may be summarized as follows.

- The article emphasized the basic assumption of Kriging, namely old simulation observations closer to the new point to be predicted, should receive more weight. This assumption is formalized through a stationary covariance process with correlations that decrease as the distances between the inputs of observations increase.

- Moreover, the Kriging model is an exact interpolator; i.e., predicted outputs equal observed simulated outputs at old points—which is attractive in deterministic simulation.

- The article reviewed some more recent results for random simulation, and explained how the true variance of the Kriging predictor can be estimated through bootstrapping.

- The article finished with a discussion of one-shot and sequentialized, customized designs for sensitivity analysis and optimization of simulation models, analyzed through Kriging.

There is a need for more research:

- Kriging software needs further improvement. For example, the estimated metamodel should be less sensitive to the prespecified lower and upper limits for the correlation parameters \( \ell_j \). The usual MSE criterion may be replaced by the maximum squared error criterion. In random simulation, Kriging should allow predictors that do not equal the average outputs at the inputs already observed; see the discussion of (8).

- Sequentialized and customized design procedures may benefit from asymptotic proofs of their performance; e.g., does the design approximate the optimal design (the latter designs are briefly discussed in [26])?

- More experimentation and analyses may be done to derive rules of thumb for the sequential procedures’ parameters, such as the size of the pilot design in deter-
ministic or random simulation and the initial number of replicates in random simulation experiments. For this pilot design not only LHS but also other space-filling designs may be studied; e.g., maximin designs and orthogonal arrays.

- **Stopping rules** for sequential designs based on a measure of accuracy (or precision) may be investigated.
- Kriging should also be applied to practical random simulation models, which are more complicated than the academic M/M/1 queuing model and (s, S) inventory models.
- The designs for sensitivity analysis and optimization might be combined for robust optimization (also see [26]).
- Nearly all Kriging publications assume univariate output, whereas in practice simulation models have multi-variante output.
- Often the analysts know that the simulation’s I/O function has certain properties, e.g., monotonicity. Most metamodels (such as Kriging and regression) do not preserve these properties (also see [26]).
- Sequential and customized designs may be analyzed not only through Kriging but also through other types of metamodels.

**Acknowledgements**

I thank the anonymous referee for very useful comments on the previous version, including 20 references. I also thank my colleagues Dick den Hertog and Wim van Beers (both at Tilburg University) for their comments on earlier versions of this article.

**References**


