Monotonicity-preserving Bootstrapped Kriging Metamodels for Expensive Simulations

Jack P.C. Kleijnen and Wim C.M. van Beers
Department of Information Management / CentER,
Tilburg University, Postbox 90153, 5000 LE Tilburg,
Netherlands, {kleijnen@uvt.nl, W.C.M.vanBeers@uvt.nl}

Abstract

Kriging (Gaussian process, spatial correlation) metamodels approximate the Input/Output (I/O) functions implied by the underlying simulation models; such metamodels serve sensitivity analysis and optimization, especially for computationally expensive simulations. In practice, simulation analysts often know that the I/O function is monotonic. To obtain a Kriging metamodel that preserves this known shape, this article uses bootstrapping (or resampling). Parametric bootstrapping assuming normality may be used in deterministic simulation, but this article focuses on stochastic simulation (including discrete-event simulation) using distribution-free bootstrapping. In stochastic simulation, the analysts should simulate each input combination several times to obtain a more reliable average output per input combination. Nevertheless, this average still shows sampling variation, so the Kriging metamodel does not need to interpolate the average outputs. Bootstrapping provides a simple method for computing a noninterpolating Kriging model. This method may use standard Kriging software, such as the free Matlab toolbox called DACE. The method is illustrated through the M/M/1 simulation model with as outputs either the estimated mean or the estimated 90% quantile; both outputs are monotonic functions of the traffic rate, and have nonnormal distributions. The empirical results demonstrate that monotonicity-preserving bootstrapped Kriging may give higher probability of covering the true simulation output, without lengthening the confidence interval.

Keywords: Queues: simulation; Simulation: statistical analysis; Statistics: nonparametric

Version: September 23, 2009

1 Introduction

Simulation models are applied in a great variety of scientific disciplines, from (say) sociology to astronomy; see Karplus (1983)'s famous overview. The goals of these simulation models may be classified as (also see Kleijnen, 2008, p. 7)

- *sensitivity analysis*—either global ('what if' analysis) or local (gradient or derivative estimation)—of the simulation model;
- *optimization* of the real system being simulated.

To realize these two goals, the simulation analysts 'experiment' with the simulation model; i.e., they run the model with different combinations of the 'inputs' of the simulation model; these inputs may be either input variables (e.g., the number of servers in a queuing simulation) or parameters (e.g., the server speed); see Zeigler, Praehofer, and Kim (2000)'s fundamental book on simulation.

Unfortunately, practical simulation often requires much computer time for obtaining the output (response) w for an input combination $\mathbf{x}$. The analysts therefore fit a *metamodel* to the relatively small number (say) n of input combinations $\mathbf{x}_i$ $(i = 1, \ldots, n)$ actually simulated; i.e., a metamodel is a model of the Input/Output (I/O) behavior of the underlying simulation model. The most popular types of metamodels are first-order and second-order polynomials. *Kriging* (also called 'Gaussian Process' or 'spatial correlation') metamodels have also become popular in deterministic simulation, which is applied in Computer Aided Engineering (CAE); see the classic article on Kriging in deterministic simulation by Sacks et al. (1989), the popular textbook by Santner, Williams, and Notz (2003) and the additional recent references in Kleijnen (2008, p. 3).

Mathematically speaking, these Kriging models are (exact) *interpolators*; i.e., the Kriging predictors equal the outputs observed for the n 'old' input combinations. Only recently, Kriging has been investigated for random simulation models; the oldest publication seems Van Beers and Kleijnen (2003) (we ignore randomness caused by numerical noise, which is an issue in simulation models used in engineering; see Forrester, Sóbester, and Keane (2008, p. 141)). The outputs of such simulation models are random; i.e., the use of different *Pseudo-Random Number* (PRN) streams give different output observations. The interpolation property of Kriging is then less desirable. Santner et al. (2003, pp. 215-249) account for the so-called *nugget effect* or *measurement error* by adding a white noise term; their Kriging predictor does not interpolate the n outputs; also see Forrester et al. (2008, p. 143). Recently, Ankenman, Nelson, and Staum (2009) introduced Kriging for random simulation, accounting for variance heterogeneity (e.g., the variance of the waiting time increases as the traffic rate in a queuing simulation increases). A similar Kriging model is developed by Yin, Ng, and Ng (2008), who speak of the 'modified nugget effect'. The Kriging predictors that account for nugget effects do not interpolate the n outputs averaged over the (say) m_i ($i = 1, \ldots, n$) replicates per input combination; these replicates are Identically, Independently Distributed (IID) if they use nonoverlapping PRN streams. Ankenman et al. also account for possible correlations between the outputs for different input combinations caused by the use of *Common Random Numbers* (CRN); CRN mean that the same PRN seed (initial value) is used to generate the same replication number (say) r for all the n input combinations, where $r = 1, \ldots, m = \min_i m_i$.

Unfortunately, there is no well-documented *software* implementing these nugget effects (e.g., Ankenman et al. use their 'own code written in S-PLUS'); popular software (either commercial or academic) assumes deterministic simulation. We focus on the free Matlab Kriging toolbox called *DACE*, which is well documented in Lophaven, Nielsen, and Sondergaard (2002).

In this article, we assume a *given* design—i.e., the n input combinations and their number of replicates m_i are given—that is so small (because the simulation model is so expensive) that the classic Kriging metamodel does not preserve the shape of the I/O function. To solve the problem of 'wiggling' or 'erratic' Kriging metamodels, we shall derive bootstrapped Kriging with the following properties:

1. Our Kriging preserves the *monotonicity* assumed for the I/O function implied by the underlying simulation model.

2. Our Kriging is *not an interpolator*; i.e., its predictor for the n old input combinations does not necessarily equal the n average simulated outputs $\overline{w_i} = \sum_{r=1}^{m_i} w_{i;r}/m_i$ (where m_i still denotes the number of replicates for input combination i, and $i = 1, \ldots, n$).

3. Our Kriging also gives a *confidence interval* for the predictor.

4. Our Kriging is *distribution-free*; i.e., it does not assume normally (Gaussian) distributed simulation outputs; all other authors on Kriging assume normality, but our M/M/1 example will show that this may be an unrealistic assumption.

5. Our Kriging accounts for *variance heterogeneity*, so $var(w_i)$ is an (unknown) function of $\mathbf{x}_i$.

As we mentioned under property 1, we assume that the I/O function is *monotonic*. For example, in queuing simulation the analysts often assume that the expected waiting time increases as the traffic increases. In medical research, the response is often assumed to be a monotonic function of the medicine dose; see Frazier, Powell, and Dayanik (2009). In regression analysis so-called 'isotonic' regression and 'rank' regression may be used for monotonic I/O functions; see the references in Kleijnen (2008, pp. 98, 162). Factor screening through sequential bifurcation assumes a monotonic I/O function; applications are random simulation of supply chains and deterministic simulation of the CO_2 greenhouse effect; see the review by Kleijnen (2009).

Monotonicity-preserving Kriging implies that the estimated *gradients* remain positive as the simulated traffic rate increases. This monotonicity preservation implies sensitivity analysis results that are understood and accepted by the clients of the simulation analysts. Furthermore, we conjecture that estimated gradients with correct signs will improve simulation optimization. Moreover, we shall investigate whether our monotonicity-preserving Kriging gives better predictions measured by the *Mean Squared Error* (MSE), which is the standard criterion in Kriging. Finally, we shall compare the estimated *coverage* and *width* of the confidence intervals based on our distribution-free bootstrapping and on the classic (or standard) Kriging variance predictor.

Technically, we realize our Kriging (with the five properties listed above) through distribution-free bootstrapping that we adapt for random simulation. This bootstrapping is conceptually simple: it resamples—with replacement—the replicated simulation outputs. It is computationally inexpensive, compared with the computer time required by many practical simulations; e.g., Simpson et al. (2004) give an example that required 36 to 160 hours of computer time to obtain a single run for a crash model at Ford; Ankenman et al. (2009) give more examples.

Note: Traditionally, bootstrapping is used to estimate the variability of some statistic; e.g., Den Hertog, Kleijnen, and Siem (2006) estimate the true variance of the Kriging predictor (accounting for the estimation of the Kriging parameters) through parametric bootstrapping assuming a Gaussian process for the deterministic simulation output. Our bootstrapping, however, has a very different goal, namely monotonicity preservation.

Our *main conclusion* is that our bootstrap Kriging gives confidence intervals with better coverage.

Note: If the simulation model is 'nearly' deterministic, then there are so few replicates that distribution-free bootstrapping gives too little variation in the outputs and we resort to parametric bootstrapping assuming a Gaussian process.

The remainder of this article is organized as follows. Section 2 details our monotonicity-preserving bootstrapped Kriging. Section 3 illustrates this Kriging through the M/M/1 simulation model, focusing on terminating simulation with either the average or the estimated 90% quantile as output; these outputs are monotonically increasing functions of the traffic rate. Section 4 presents conclusions and topics for further research.

2 Monotonicity-preserving Bootstrapped Kriging

As we mentioned above, in *deterministic* simulation Kriging is an *interpolator*:

$$y(\mathbf{x}_i) = w(\mathbf{x}_i) \; i = 1, \ldots, n \tag{1}$$

where y denotes the Kriging predictor, w the simulation output, $\mathbf{x}_i$ the i^{th} combination of the (say) $k \geq 1$ simulation inputs with $i = 1, \ldots, n$ where n denotes the number of 'old' input combinations that have already been simulated; outputs are predicted through the Kriging metamodel fitted to the n I/O combinations. ($\mathbf{x}$ is a 'point' in the k-dimensional experimental space; the 'inputs' are called 'factors' in experimental design theory.)

Random (stochastic) simulation (e.g., discrete-event simulation) gives different outputs at $\mathbf{x}_i$ whenever the PRN seed changes. We assume that the simulation analysts obtain $m_i \geq 2$ replicates; otherwise, they cannot evaluate the variability of the simulation model's output. Moreover we assume that these replicates are IID, because they use nonoverlapping PRN streams and the PRNs are assumed to be IID on the interval $(0, 1)$. The uncertainty caused by the PRN streams is called *intrinsic* by Ankenman et al. (2009) and Kleijnen (2008, p. 18). These replicates enable the following classic *unbiased* variance estimators:

$$s_i^2 = \frac{\sum_{r=1}^{m_i} (w_{i;r} - \overline{w_i})^2}{m_i - 1} \text{ with } m_i \geq 2. \tag{2}$$

Alternative popular symbols for these estimators are $s^2(w_i)$, $\widehat{\sigma^2(w_i)}$, and $\widehat{\sigma_i^2}$. (If $w_{i;r}$ is Normally IID, then s_i^2 has several well-known properties; e.g., $s_i^2(m_i - 1)$ is $\chi^2_{m_i-1}$ distributed.) Obviously $s^2(\overline{w_i}) = s^2(w_i)/m_i$.

We assume that the simulation model is *expensive* so n (number of simulated input combinations) and m_i (number of replicates for combination i) are so *small* that the fitted Kriging metamodel does not necessarily preserve the monotonicity of the I/O function defined by the underlying simulation model; an example is Figure 1, which we shall discuss later on.

Note: Kleijnen, Van Beers, and Van Nieuwenhuyse (2009) select m_i such that the sample average $\overline{w_i}$ satisfies a relative precision requirement; i.e., they select m_i such that the halfwidth of the $(1-\alpha)$ confidence interval for the average simulation output is within $\gamma\%$ of the true mean (also see the classic simulation textbook, Law 2007, pp. 500-503). Ankenman et al. (2009) mention a ratio related to signal/noise that they call $\gamma = V/\tau^2$. We recommend that if the analysts assume monotonicity for the simulation model's I/O function, then they obtain so many replicates that the n average simulation outputs also show this property. However, if the simulation model is expensive, then the analysts may not be able to follow our recommendation; our bootstrapped Kriging procedure may then preserve the monotonicity.

Ankenman et al. (2009) introduce the term *extrinsic uncertainty* for the approximation error that remains when fitting a Kriging model—even if there were no 'intrinsic uncertainty' (as is the case in deterministic simulation). Those authors—and the other authors on the nugget effect mentioned in Section 1—assume that the intrinsic and extrinsic errors are *additive*; our bootstrap does not need to make such an assumption.

Obviously, random simulation gives *average* outputs $\overline{w_i}$ that are still random; i.e., their observed realizations would change if the experiment were repeated with different PRN seeds. Therefore we do not require our Kriging predictor to equal the observed average outputs (whereas we did in previous publications; see, e.g., Kleijnen et al. 2009 and Van Beers and Kleijnen 2003).

A simple data-driven way to account for the intrinsic randomness of $w_{i;r}$ is *bootstrapping.* There are two types of bootstrapping (see the textbook by Efron and Tibshirani 1993, and the additional references given by Kleijnen 2008, p. 81):

- *Distribution-free* (non-parametric) bootstrapping, which assumes that all n old points are replicated 'enough' times: $m_i >> 2$ (in the M/M/1 example, $m_i = 5$).

- *Parametric* bootstrapping; e.g., we assume that the simulation responses are normally distributed with parameters $\mu_i = E(w_i)$ and σ_i^2 estimated from the $m_i \geq 2$ replicates (if we assumed constant variances, then only one point would need to be replicated more than once: $\exists i : m_i \geq 2$).

We focus on distribution-free bootstrapping, because we assume that each point $\mathbf{x}_i$ is replicated enough times. So we resample the m_i replicates per point $\mathbf{x}_i$, which gives the bootstrap observations $w^*_{i;r}$ which in turn give the average output $\overline{w_i}^*$ (the superscript $*$ is the usual symbol for bootstrapped observations). (Obviously, parametric bootstrapping assuming normality may give outputs between $-\infty$ and ∞, whereas distribution-free bootstrapping gives a smaller range of possible outputs.) The simulation outputs $w_{i;r}$ have different variances at different points $\mathbf{x}_i$ (e.g., the simulated steady-state mean waiting time has a variance that increases with the traffic rate) so they are not IID.

We formalize our *bootstrap procedure* through the following pseudocode, assuming that no CRN are used (also see the last paragraph of this section) and allowing a possibly non-constant number of replicates, m_i $(i = 1, \ldots, n)$:

1. Initialize the input combination: $i = 1$.

2. Initialize the replicate number: $r = 1$.

3. Resample—with replacement—a replicate number r^* from the uniform distribution defined on the integers $1, \ldots, m_i$; i.e., the uniform density function is $p(r^*) = 1/m_i$ with $r^* = 1, \ldots, m_i$.

4. Replace the r^{th} 'original' output $w_{i;r}$ by the bootstrap output $w^*_{i;r} = w_{i;r*}$.

5. If $r < m_i$ then $r = r + 1$ and return to Step 3; else proceed to the next step.

6. If $i < n$ then $i = i + 1$ and return to Step 2; else proceed to the next step.

7. Compute the Kriging predictor y^* from the bootstrapped I/O data set $(\mathbf{X},\overline{\mathbf{w}}^*)$ where $\mathbf{X}$ denotes the $n \times k$ matrix with the n old combinations of the k inputs and $\overline{\mathbf{w}}^*$ denotes the n-dimensional vector with the averages $\overline{w_i}^* = \sum\limits_{r=1}^{m_i} w^*_{i;r}/m_i$ and $i = 1, \ldots, n$.

To these bootstrapped I/O data $(\mathbf{X},\overline{\mathbf{w}}^*_i)$, we fit an *interpolating* Kriging model y^*(analogous to (1)):

$$y^*_i = \overline{w_i}^* \ (i = 1, \ldots, n). \tag{3}$$

We point out that we do not fit the Kriging model to an individual output $w^*_{i;r}$ because an individual output is noisier.

Kriging for deterministic simulation uses the extrinsic noise's covariance matrix $\mathbf{\Gamma}$, defined below (17) in Appendix 1 with basic Kriging formulas. This $\mathbf{\Gamma}$ depends on the correlation parameters θ_j defined below (19), which are computed through *Maximum Likelihood Estimation*

(MLE). Similar MLE is used by Ankenman et al. (2009) and Yin et al. (2008). We also use MLE when we use the DACE toolbox. This MLE assumes that the covariances follow a specific function—namely, the Gaussian correlation function (19)—and assumes normality, whereas we do not assume normality when bootstrapping; we accept this inconsistency because we use DACE's MLE only to estimate the nuisance parameters collected in $\mathbf{\Gamma}$. (Van Beers and Kleijnen (2003) use Least Squares instead of MLE, to estimate $\mathbf{\Gamma}$.)

Because of the randomness in bootstrapping, the resampling is repeated (say) B times, where B is called the *bootstrap sample size*. So (3) results in B bootstrapped Kriging predictors $\mathbf{y}_b^* = (y_{1;b}^*, \dots, y_{n;b}^*)'$ with $b = 1, \dots, B$. From these B predictors we select the (say) B' *monotonicity-preserving* ones. Assuming a *strictly monotonically increasing* I/O function, we select those B' bootstrapped Kriging models that are strictly monotonically increasing:

$$y_{i;b'}^* < y_{i';b'}^* \text{ if } \mathbf{x}_i < \mathbf{x}_{i'} \ (i, i' = 1, \dots, n) \ (b' = 1, \dots, B') \tag{4}$$

where $\mathbf{x}_i < \mathbf{x}_{i'}$ means 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. Obviously, we may define 'monotonically decreasing' in a strictly analogous way. Notice that (4) implies that each of the k components of the n gradients are positive; we denote this by

$$\nabla y_{i;b'}^* > \mathbf{0} \ (i = 1, \dots, n) \ (b' = 1, \dots, B'). \tag{5}$$

These gradients are provided 'for free' by DACE; see (18) in Appendix 1 and also Kleijnen (2008, p. 143).

Note: Monotonicity preservation of Kriging metamodels is also examined by Siem (2007), but he does not succeed in finding a solution. Velikova (2006) also discusses monotonicity, but she does so for neural networks instead of Kriging. Feelders (2000) discusses monotonic classification trees in data mining, using bootstrapping.

So there are B' bootstrapped monotonicity-preserving Kriging predictors $y_{b'}^*$ $(b' = 1, \dots, B')$. We use these predictors to compute B' predictions $y_{t;b'}^*$ for (say) v new input combinations $\mathbf{x}_u.(u = 1, \dots, v)$. So, we predict not a single new point $\mathbf{x}_{n+1}$ but $v >> 1$ new points; such a *test set* was also used by Sacks et al. (1989). From the B' predictions for point t we compute as *point estimate* $y_{t;(0.50B')}^*$ where $\lceil x \rceil$ denotes the integer resulting from rounding x upwards and the subscript () denotes the order statistics; so $y_{u;(\lceil 0.50B_s \rceil)}^*$ denotes the sample median (the sample median is not sensitive to outliers, whereas the sample mean is; quantiles such as the median will also be discussed in equation 9 for our M/M/1 simulation).

Note: Instead of the sample median $y^*_{u;(\lceil 0.50B' \rceil)}$ we may use the sample mean $\overline{y^*_u} = \sum y^*_{u;b'}/B'$—especially when using Kriging for optimization that uses the resulting explicit function.

Besides this point estimator, we compute the lower and upper bound of the (say) 90% *confidence interval* for the bootstrapped Kriging predictor for the true value at test point $\mathbf{x}_u$, namely $y^*_{u;(\lfloor 0.05B' \rfloor)}$ and $y^*_{u;(0.95B')}$ where $\lfloor x \rfloor$ denotes the integer resulting from rounding x downwards (more refined procedures are discussed by Efron and Tibshirani 1993). If B' is too small to give a reasonable confidence interval, we increase the bootstrap sample size B; e.g., in our M/M/1 example we start with $B = 100$ but augment B with 100 until either $B' \geq 100$ or (to avoid excessive computational time) $B = 1000$. The Kriging literature (e.g., Lophaven et al. 2002, p. 4 and Santner et al. 2003, p. 96) gives confidence intervals, assuming normality and computing an estimate of the variance $\widehat{\sigma}^2_y$ of the classic predictor y that ignores the random character of the Kriging weights resulting from estimating the parameters in the Kriging correlation functions; see θ_j in (19) in Appendix 1; Den Hertog et al. (2006) have already shown that this estimator was misleading in several deterministic simulation examples, and so will we in our M/M/1 example.

Of course, a confidence interval would have a perfect coverage of 100% if its width were infinite—but such an interval is useless. We therefore estimate both the *coverage* and the *width* of the confidence interval for bootstrapped and classic Kriging—averaged over all ν test points—in our M/M/1 example in Section 3.

Ankenman et al. (2009) recommend not using *CRN*, because this technique increases the MSE under their assumptions. We, however, claim that the use of CRN should depend on the *goal* of the metamodel, namely, sensitivity analysis and optimization (see again Section 1). Actually, these goals are related; e.g., the estimated gradient (a local sensitivity measure) may be used to estimate the optimum. Ankenman et al. focus on what-if analysis. Like Ankenman et al. we recommend avoiding CRN, but for a different reason: CRN reduces the variability of the bootstrapped simulation outputs averaged over the m_i replicates, whereas finding a monotonicity-preserving bootstrapped Kriging model is more likely when the bootstrapped averages have larger variability. Our bootstrap procedure can be easily adapted for CRN; see Kleijnen et al. (2009).

3 M/M/1 Example

In Section 3.1 we present some preliminary considerations for our M/M/1 simulation. In Section 3.2 we present results for our monotonicity-preserving Kriging in M/M/1 simulation.

3.1 Preliminaries

The M/M/1 queuing model is a popular example in random simulation; e.g., Ankenman et al. (2009) use this example to illustrate their analysis, and Law (2007, pp. 12-47, 79-83) details the single-server queuing system, including the M/M/1 model.

In an M/M/1 simulation, the waiting time of customer t (say) w_t may be computed through

$$w_{t+1} = \max(0, w_t + s_t - a_{t+1}) \ t = 1, 2, \ldots \tag{6}$$

where w_1 is determined by the initialization of the simulation run (e.g., if the simulation starts in the 'empty' state, then $w_1 = 0$); the exponentially distributed inputs s and a have service rate μ and arrival rate λ (the service rate is usually denoted by μ, and should not be confused with the mean service time $E(s) = \mu_s$). To obtain sampled values for these s and a, the simulation may use a single PRN stream $p_1, p_2, \ldots$ as follows (but alternative sampling routines do exist):

$$s_t = \frac{-\ln p_{2t-1}}{\mu} \text{ and } a_{t+1} = \frac{-\ln p_{2t}}{\lambda}. \tag{7}$$

This simulation model is of practical interest because it is a building block for more complicated queuing networks that are used in telecommunications, supply chains, etc. This model is also of academic interest because it generates a *time series* of length (say) T so we have the vector $\mathbf{w} = (w_{1,} \ldots, w_T)'$ (a positively correlated multivariate output); this vector may be used in steady-state analysis, and has components with non-constant variances ($var(w_t)$ increases as t increases, until the steady state is reached). It is well-known that various performance measures (see below) are nonlinear functions of the traffic rate (or traffic load) $x = \lambda/\mu$.

In our example we simulate this model starting not with $w_1 = 0$ (the usual initialization of equation 6) but with w_1 equal to its expected steady-state value. The reason is that we are not interested in the problem of determining whether the simulation has reached steady state or is still in transient state; i.e., we 'cheat' and use the analytical solution for the steady-state waiting time distribution for the M/M/1. In this way we can verify some of our simulation results. (Ankenman et al. 2009 use a similar trick.)

We study two *performance measures*:

1. the steady-state *mean* waiting time $E(w_t|t \to \infty) = \mu_w$;

2. the steady-state 90% *quantile* $w_{.9}$ defined by $P(w_t \leq w_{.9} | t \rightarrow \infty) = 0.9$.

The first measure is popular in academic research. The classic estimator of this mean is the *time-series average*

$$\overline{w} = \frac{\sum\limits_{t=1}^{T} w_t}{T}. \tag{8}$$

The second measure is more popular in practice; see Batur and Choobineh (2009), Hong (2009) and Jin, Fu, and Xiong (2003). To estimate a quantile, we sort the (autocorrelated) time series from low to high—which gives the (autocorrelated) *order statistics* $w_{(1)}, \ldots, w_{(T)}$. A classic point estimate of $w_{.9}$ is

$$\widehat{w_{.9}} = w_{(\lceil .9T \rceil)}. \tag{9}$$

To observe the sampling variability of the estimates of the mean and 90% quantile defined in (8) and (9), we use $m \geq 2$ *replicates* (alternative approaches are discussed in the literature; see Law 2007, p. 506). Replicate r ($r = 1, \ldots, m$) gives the average waiting time $\overline{w_r}$ and the estimated quantile $\widehat{w_{.9;r}}$. We expect these averages $\overline{w_r}$ to be *normally* distributed because of the Functional Limit Theorem; see Lehmann (1999). The quantile estimators $\widehat{w_{.9;r}}$ are only asymptotically normally distributed; see Chen (2008) and Hong (2009). We (rather arbitrarily) select a 'short' runlength $T = 1000$ and a 'long' length $T = 100000$ (Ankenman et al. 2009 select $T = 1000$ in a related example). Furthermore, in the preliminary investigation reported in this subsection, we select only two traffic rates, namely, $\lambda/\mu = 0.5$ and $\lambda/\mu = 0.9$ (in the next subsection we use more traffic rates); a high traffic rate gives stronger autocorrelation so we expect nonnormality. In this subsection, we wish to obtain accurate estimates of the true behavior of the simulated outputs, so we select $m = 1000$ replicates (our Kriging results in the next subsection will use smaller, realistic m values). We test the goodness of fit through the chi-square and the Kolmogorov-Smirnov tests; also see (Law 2007, pp. 340-351). We present the resulting p-values for the latter test only, because the former test gives similar results; see Table 1. This table shows that the estimated average and quantile are not normally distributed if the simulation run is short ($T = 1000$)—even for a relatively low traffic rate ($\lambda/\mu = 0.5$).

Note: The estimated mean may be larger than the estimated quantile ($\overline{w} > \widehat{w_{.9}}$), because the former statistic is sensitive to outliers. Indeed, for low traffic rates we sometimes observe this phenomenon.

Runlength	$T = 1000$		$T = 100000$	
Traffic rate	$\lambda/\mu = 0.5$	$\lambda/\mu = 0.9$	$\lambda/\mu = 0.5$	$\lambda/\mu = 0.9$
Average $\overline{w}$	< 0.01	< 0.01	> 0.15	0.11
90% quantile $\widehat{w_{.9}}$	< 0.01	< 0.01	> 0.15	0.116

Table 1: Kolmogorov-Smirnov Test of Normality: p Values

To evaluate the performance of our procedure, we use the following analytical results; see equation (2.21) in Gross and Harris (1998, p. 67). The steady-state waiting time W in an M/M/1 model with traffic rate $x = \lambda/\mu$ has the distribution $P(W \leq w) = 1 - x\exp(-\mu(1-x)w)$. This implies for the 90% quantile $w_{.9}$: $1 - x\exp(-\mu(1-x)w_{.9}) = 0.9$, so $w_{.9} = -\ln(0.1/x)/\mu(1-x)$. The mean μ_w is $x/[\mu(1-x)]$. We select the time units for the arrival and service times such that the service rate μ equals one.

3.2 Monotonicity-preserving Kriging Results

In practice, a small number of replicates m is used if a single simulation run takes much computer time; nevertheless, m should be large enough to obtain adequate signal/noise (see Kleijnen et al. 2009). The signal/noise also depends on the runlength T. After some experimentation, we select $T = 1000$ and $m = 5$ for our example. Furthermore, we select $n = 5$ values for the traffic rate λ/μ (for higher n values, we do not expect wiggly behavior so there is no need to bootstrap for monotonicity preservation). We select these n points such that the traffic rates $x_i = \lambda_i/\mu_i$ are inside the experimental area $0.1 \leq x \leq 0.9$. We select a bootstrap sample size $B = 100$; sometimes, this does not give enough monotonicity-preserving Kriging models (so $B' < 100$), so we bootstrap another $B = 100$ times. We assume that the users are interested in either the mean or the 90% quantile, so we bootstrap the estimated mean and quantile independently (i.e., we do not resample the m correlated pairs $(\overline{w_r}, \widehat{w_{.9;r}})$).

Besides the $n = 5$ old points, we select $v = 25$ new points that are to be predicted. We select these new points such that no extrapolation is needed, because Kriging metamodels are known to be poor extrapolators. To select these new points, we use Latin Hypercube Sampling (LHS); for an explanation of LHS, references, and software see Kleijnen (2008, p. 126-130).

To estimate whether our *point* predictor (the bootstrap median) for the true output (say) ζ is better than the classic Kriging predictor y, we estimate the Integrated MSE through

$$\widehat{IMSE}^* = \frac{\sum_{u=1}^{v}(y^*_{u;(\lceil 0.50B' \rceil)} - \zeta_u)^2}{v} \text{ and } \widehat{IMSE} = \frac{\sum_{u=1}^{v}(y_u - \zeta_u)^2}{v}, \tag{10}$$

assuming that we know the true output ζ_t—as is the case for the M/M/1 example. Notice that we use the *same* bootstrapped Kriging metamodel to obtain point predictors for the v different test points.

To estimate whether our confidence interval gives better *coverage* than the classic Kriging confidence interval does, we compute the indicator function

$$I^*_u = 1 \text{ if } y^*_{u;(\lfloor 0.05B' \rfloor)} < \zeta_u < y^*_{u;(\lceil 0.95B' \rceil)}\text{; else } I^* = 0 \ (u = 1, \ldots, v) \tag{11}$$

for our bootstrap procedure; for the classic procedure we compute

$$I_u = 1 \text{ if } y_u - 1.64\widehat{\sigma}_{y_u} < \zeta_u < y_u + 1.64\widehat{\sigma}_{y_u}\text{; else } I = 0 \ (u = 1, \ldots, v) \tag{12}$$

where $\widehat{\sigma}_{y_u}$ is provided by the classic Kriging literature and software, including DACE. We point out that—unlike our bootstrap confidence interval (11)—the classic interval (12) is symmetric around its point estimate y_t and may include negative values—even if negative values are impossible, as is the case for waiting times. Analogously to (10) we estimate the coverage averaged over all v test points:

$$\overline{I^*} = \frac{\sum_{u=1}^{v} I^*_u}{v} \text{ and } \overline{I} = \frac{\sum_{u=1}^{v} I_u}{v}. \tag{13}$$

Note: The estimators in (13) are not binomially distributed; e.g., I_u and $I_{u'}$ $(u' = 1, \ldots, v)$ are not independent because they use the same estimated (fitted) Kriging model.

In our M/M/1 example, we obtain (say) L *macro-replicates*, which differ only in their PRN seeds (in practice, the analysts obtain a single macro-replicate). Let $\overline{I^*_l}$ denote the average indicator for our bootstrap predictor in macro-replicate l, and $\overline{I_l}$ the analogue for the classic predictor; see (13). Our predictor then has the better coverage if the macro-replicates give an average $\overline{\overline{I^*}} = \sum_l \overline{I^*_l}/L$ that is closer to the nominal value 0.90 than the classic predictor's $\overline{\overline{I}} = \sum_l \overline{I_l}/L$. We obtain 100 *macro-replicates* so $L = 100$ (Ankenman et al. also use $L = 100$). From these L macro-replicates we estimate the mean IMSE through $\overline{\widehat{IMSE}}$ and the standard error $s(\widehat{IMSE})$ to obtain the following 90% confidence interval for the IMSE of the classic Kriging and our monotonicity-preserving bootstrap approach:

$$\overline{\widehat{IMSE}} \pm 1.64\frac{s(\widehat{IMSE})}{\sqrt{L}} \text{ and } \overline{\widehat{IMSE}^*} \pm 1.64\frac{s(\widehat{IMSE}^*)}{\sqrt{L}}$$

where

$$\overline{\widehat{IMSE}} = \frac{\sum_{l=1}^{L} \widehat{IMSE}_l}{L} \text{ and } s(\widehat{IMSE}) = \sqrt{\frac{\sum_{l=1}^{L} (\widehat{IMSE}_l - \overline{\widehat{IMSE}})^2}{L-1}},$$

and the formulas for the bootstrap results are analogous. For the coverage we use analogous equations:

$$\overline{\overline{I}} \pm 1.64 \frac{s(\overline{I})}{\sqrt{L}} \text{ and } \overline{\overline{I^*}} \pm 1.64 \frac{s(\overline{I^*})}{\sqrt{L}}$$

where

$$\overline{\overline{I}} = \frac{\sum_{l=1}^{L} \overline{I_l}}{L} \text{ and } s(\overline{I}) = \sqrt{\frac{\sum_{l=1}^{L} (\overline{I_l} - \overline{\overline{I}})^2}{L-1}},$$

etc. The formulas for the lengths of the confidence intervals are analogous.

In some macro-replicates the classic Kriging metamodels are monotonic, so there is no need to bootstrap. To check whether a classic Kriging metamodel gives a monotonic I/O function, we check whether all the gradients estimated at the n old points are positive: $\forall i : \nabla y_i \geq \mathbf{0}$ $(i = 1, \ldots, n)$; also see (5). If a macro-replicate satisfies this condition, then we sample a new macro-replicate; we stop after we have $L = 100$ macro-replicates with nonmonotonic classic Kriging metamodels.

Figure 1 shows a macro-replicate in which the classic Kriging metamodel shows wiggling, whereas a bootstrapped model is monotonic. This figure also shows—for each of the $n = 5$ input values—the $m = 5$ replicated simulation outputs (see dots) and their averages (see stars). Furthermore, the figure shows the analytical (dotted) I/O curve. Notice that for low traffic rates the variability of the individual simulation outputs is so small that this variability is hardly visible; nevertheless, the bootstrap finds a monotonic Kriging model. (The data of all figures in this article, and the corresponding Matlab code are available from the authors.)

Wiggling means that the derivative $d\widehat{y}/dx$ is negative for at least one x value in the area of interest. Wiggling may also occur at new points (besides the old points; see Fig. 1). In the M/M/1 example (which has a single input), we check whether Kriging gives wiggling at 100 new points, spread uniformly across the experimental range. (In applications with multiple inputs, however, such a grid search is rather expensive, so we may find the $\mathbf{x}$ point that minimizes the components of the estimated gradient $\nabla \widehat{y}$; if at least one component is negative, then the Kriging predictor $\widehat{y}$ shows wiggling so we apply bootstrapping.)

Note: We also experiment with Universal Kriging replacing the constant term μ in Ordinary Kriging by a first-order and a second-order polynomial respectively (see Appendix 1, last paragraph). This Universal Kriging, however, does not remove the wiggling so we focus on Ordinary Kriging.

Figure 2 gives the IMSE for the average and the 90% quantile estimated from these $L = 100$ macro-replicates; the symbol 'KA' stands for 'Kriging the Average', 'BQ' for 'Bootstrap the Quantile', etc. This figure shows that our bootstrap gives smaller estimated IMSEs, albeit not significantly smaller; of course, the 90% quantile has larger IMSEs than the mean has. Classic Kriging uses the MSE as the criterion when optimizing the Kriging weights λ; when we impose the monotonicity constraint, we do not expect significantly lower MSE.

Figure 3 gives the estimated coverage for the average and the 90% quantile. This figure shows that our bootstrap gives significantly higher estimated coverage for the mean and the quantile. Unfortunately, all estimated coverages are significantly lower than the nominal value of 90%.

Figure 4 gives the corresponding estimated widths. This figure shows that our bootstrap gives widths that are not significantly shorter; see the point estimate. The variability of the width is smaller for our bootstrap; see the length of the confidence interval for the width. Together the latter two figures show that our bootstrap gives better coverage without lengthening the confidence interval.

Because the coverage is significantly lower than the prespecified nominal value of 90%, we compare Kriging with classic linear regression metamodeling; i.e., we repeat our experiment with *second-order polynomial* metamodels. Because the variances of the simulation outputs vary with the input combination, we adapt *Ordinary Least Squares* (OLS) as explained in Kleijnen (2008, p. 95) and Appendix 2. This gives coverages for the mean and the quantile that are lower than Kriging (classic or bootstrapped) does. The variability of the coverage (measured by the length of the confidence interval for the estimated coverage) is smaller for the second-order polynomial; we think that this phenomenon arises because in this metamodel only the three regression coefficients may vary. Details are displayed in Table 3 in Appendix 2.

Finally, we increase n (number of old points) from 5 to 10; see Table 2. Remember that we limit our analysis to the $L = 100$ macro-replicates that show *nonmonotonic* classic Kriging metamodels. This table shows

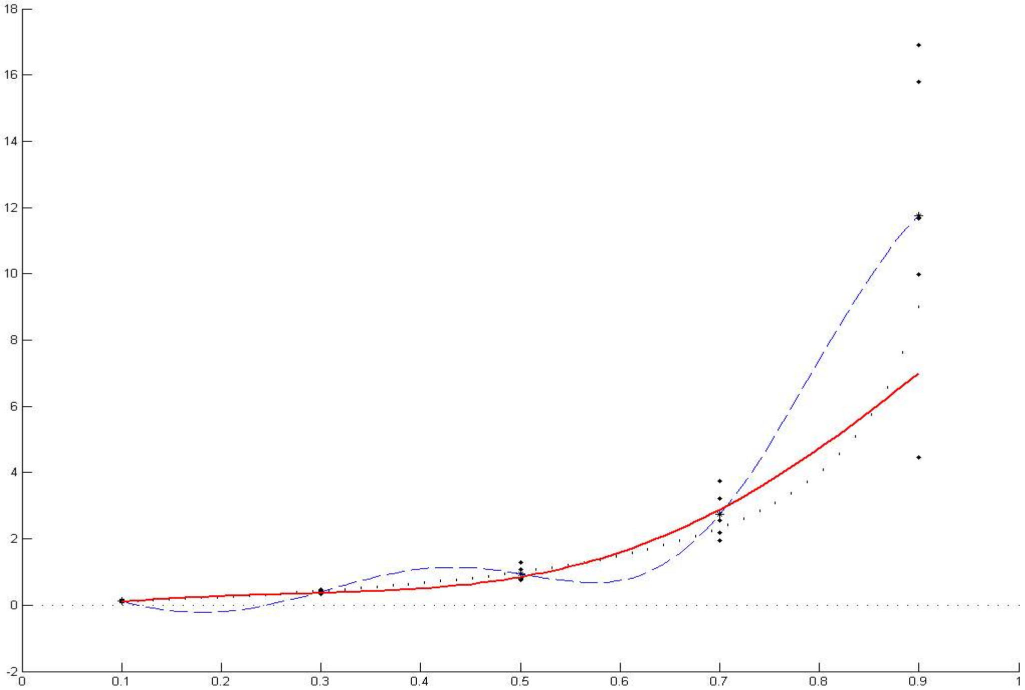

Figure 1: Nonmonotonic Classic and Monotonic Bootstrapped Kriging metamodels and True I/O Function for M/M/1 Example with $n = 5$, $m = 5$, $T = 1000$

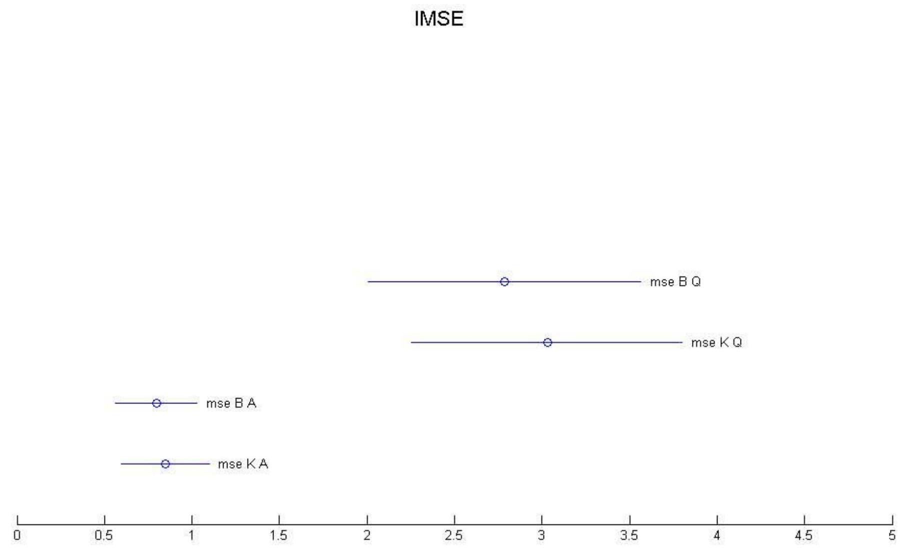

Figure 2: Estimated IMSE of Nonmonotonic Classic and Monotonic Bootstrapped Kriging for the Mean and 90% Quantile, for $n = 5,\ m = 5,\ T = 1000$

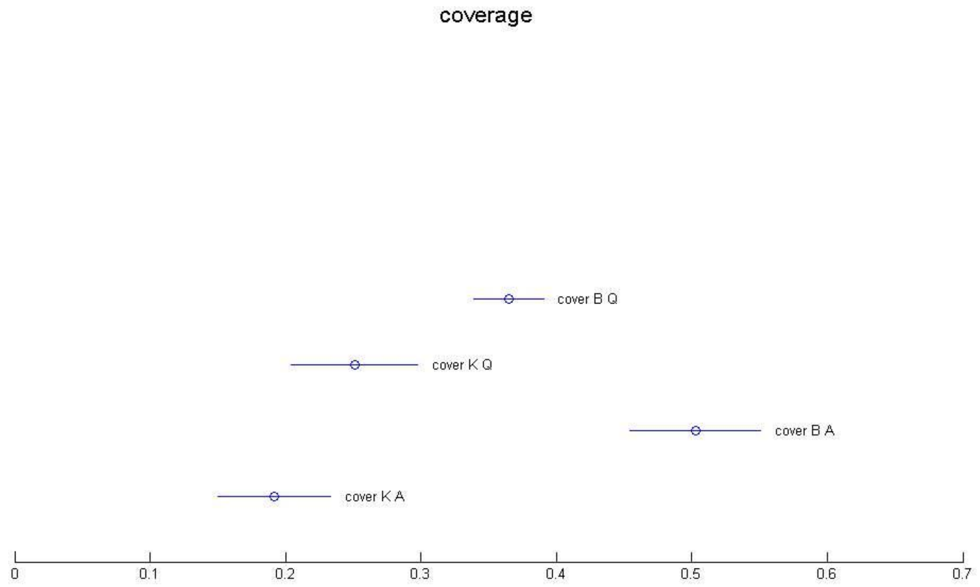

Figure 3: Coverage of Nonmonotonic Classic and Monotonic Bootstrapped Kriging for the Mean and 90% Quantile, for n = 5, m = 5, T = 1000

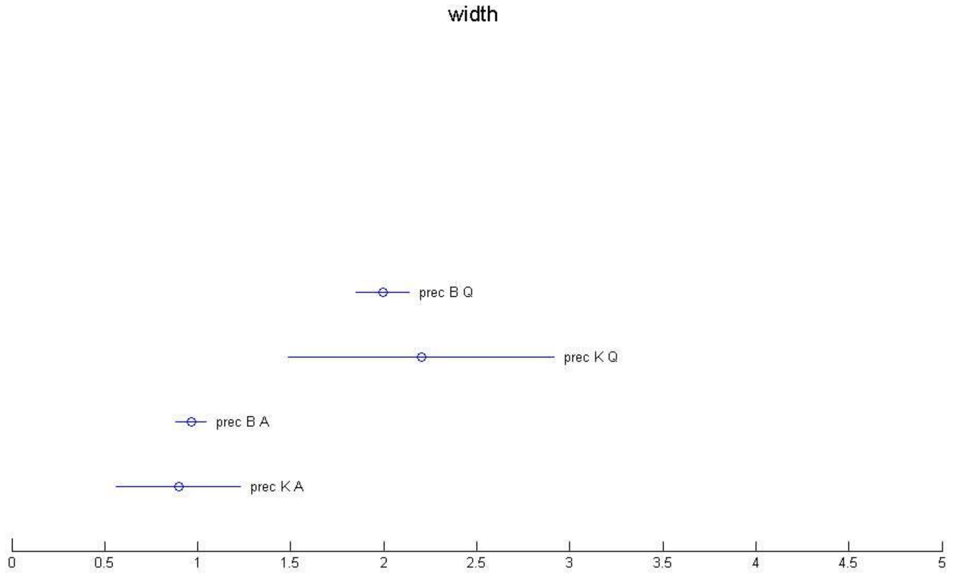

Figure 4: Width of Confidence Interval for Nonmonotonic Clasic and Monotonic Bootstrapped Kriging for the Mean and 90% Quantile, for n = 5, m = 5, T = 1000

	lower bound	median	upper bound
coverage KA	0.226	0.353	0.449
coverage BA	0.831	0.845	0.858
coverage KQ	0.331	0.442	0.552
coverage BQ	0.830	0.844	0.859
width KA	0.233	0.306	0.379
width BA	0.371	0.382	0.392
width KQ	1.023	1.047	1.070
width BQ	0.948	1.208	1.477

Table 2: Coverage in Classic Nonmonotonic Kriging (K) and Bootstrapped Monotonic Kriging (B), for the Average (A) and the 90% Quantile (Q), for n = 10, m = 5, T = 1000

that a larger number of input combinations increases the estimated coverages for both classic Kriging and bootstrapped Kriging; we think that this phenomenon is explained by the better fit of the metamodels. These coverages are close to the nominal 90% for our bootstrapped Kriging, whereas classic nonmonotonic Kriging still gives coverages far below the desired nominal value. The improved coverage of bootstrapped Kriging does not require significantly longer confidence intervals.

4 Conclusions and Future Research

In practice, simulation may be computationally *expensive*, so only a few input combinations are simulated and these few combinations are replicated only a few times (small n and m). Classic Kriging may then give wiggly, nonmonotonic metamodels. In such cases, our monotonicity-preserving Kriging gives better results; namely, better coverage without longer confidence interval.

As the number of replicates m increases, the original and the bootstrapped average simulation outputs *converge* to the true value, so the original and the bootstrapped Kriging metamodel tend to be monotonicity-preserving. However, in all our 100 macro-replicates with only $m =$ 5 replicates, we did find several monotonicity-preserving bootstrapped Kriging models, whereas the original Kriging model was erratic.

Unfortunately, a small number of simulated input combinations n may give too little information to estimate a Kriging model (classic or monotonicity-preserving) that gives the desired coverage. In such situations we would advise the analysts to spend more computer time in order to obtain reliable results. While awaiting these results, they can bootstrap the too small sample to obtain results that are better than the classic results.

In *future research* we may investigate the following topics.

- Our monotonicity-preserving bootstrap procedure may be applied to more complicated simulation models with more than one input.

- Our procedure may also be applied to the stochastic Kriging metamodels proposed by Ankenman et al. (2009) and Yin et al. (2008) instead of the classic metamodel assumed by the DACE software.

- Our procedure may be used to select a sequential experimental design (involving the stagewise selection of the number of points n and their placement in the k-dimensional input space, and the number of replicates m_i at each point); see Ankenman et al. (2009) and Kleijnen and Van Beers (2004).

- We may also study bootstrapped Kriging that preserves the know shape (monotonicity, convexity, nonnegativeness) of the underlying I/O function, which may be known or assumed in simulation optimization.

Appendix 1: Basic Kriging Formulas

Ordinary Kriging assumes

$$w(\mathbf{x}) = \mu + \delta(\mathbf{x}) \tag{14}$$

where μ is the simulation output averaged over the experimental area, and $\delta(\mathbf{x})$ is the additive external noise that forms a stationary covariance process with zero mean. For random simulation, a more suitable Kriging model augments (14) with an additive internal noise term, which is *white noise* (say) $\epsilon(\mathbf{x})$ that is independent of $\delta(\mathbf{x})$:

$$w(\mathbf{x}) = \mu + \delta(\mathbf{x}) + \epsilon(\mathbf{x}). \tag{15}$$

Ordinary Kriging uses the *linear* predictor

$$y = \lambda'\mathbf{w} \tag{16}$$

with the *optimal* weights

$$\lambda_o = \mathbf{\Gamma}^{-1}[\gamma + \mathbf{1}\frac{1 - \mathbf{1}'\mathbf{\Gamma}^{-1}\gamma}{\mathbf{1}'\mathbf{\Gamma}^{-1}\mathbf{1}}] \tag{17}$$

where $\mathbf{\Gamma} = (cov(w_i, w_{i'}))$ with $i, i' = 1, \ldots, n$ is the $n \times n$ symmetric and positive semi-definite matrix with the covariances between the n

‘old’ outputs, and $\gamma = (cov(w_i, w_0))$ is the n-dimensional vector with the covariances between the n old outputs w_i and w_0, the output of the combination to be predicted—which may be either new or old. Combining (14), (16), and (17) gives

$$y = \widehat{\mu} + \gamma' \mathbf{\Gamma}^{-1}(\mathbf{w} - \widehat{\mu}\mathbf{1}) \tag{18}$$

where $\widehat{\mu} = (\mathbf{1}'\mathbf{\Gamma}^{-1}\mathbf{1})^{-1}\mathbf{1}'\mathbf{\Gamma}^{-1}\mathbf{w}$. The correlation function for a k-dimensional input vector is assumed to be the *product* of k one-dimensional functions ρ_j $(j = 1, \ldots, k)$; a popular one-dimensional correlation function is the Gaussian one:

$$\rho_j = \exp[-\theta_j h_j^2] \tag{19}$$

where $h_j = |x_{i;j} - x_{i';j}|$ denotes the Euclidean distance between the values of the input j in the two input combinations i and i'; θ_j denotes the importance of input j; i.e., the higher θ_j is, the less effect input j has. An alternative for Ordinary Kriging is *Universal Kriging*, which replaces the constant μ in (14) by a linear combination of known functions; e.g., a low-order polynomial; see Cressie (1993, p. 151) and Lophaven et al. (2002, p. 13). Ordinary Kriging is recommended by most authors (but not all; see Blind Kriging by Joseph, Hung, and Sudjianto 2008).

Appendix 2: Basic Linear Regression Formulas

In our *linear regression* we assume

$$y(\mathbf{x}) = \sum_{j=1}^{q} \beta_j x_j + \epsilon(\mathbf{x}) = \mathbf{x}'\beta + \epsilon(\mathbf{x}) \tag{20}$$

where x_j is the j^{th} explanatory regression variable, $\mathbf{x} = (x_1, \ldots, x_q)'$, $\beta = (\beta_1, \ldots, \beta_q)'$ is the vector of regression parameters, and $\epsilon(\mathbf{x})$ is the additive noise with zero mean and variances that may vary with $\mathbf{x}$; because we do not use CRN, the noise terms at different points are independent. Because the variances of the simulation output are unknown, we proceed as follows (Kleijnen 2008, pp. 87-91 gives more alternatives). We use OLS to estimate β from the bootstrapped outputs $\overline{\mathbf{w}}_b^*$ $(b = 1, \ldots, B)$ with bootstrap sample size B (we select $B = 100$ in our experiment):

$$\widehat{\beta_b^*} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\overline{\mathbf{w}}_b^* \ (b = 1, \ldots, B)$$

where $\mathbf{X}$ is the $n \times q$ matrix of explanatory variables in the n simulated combinations.The corresponding regression predictor for point $\mathbf{x}_u$ is

$$\widehat{y_{u;b}^*} = \sum_{j=1}^{q} \widehat{\beta_{j;b}^*} x_{j;u} = \mathbf{x}_u' \widehat{\beta_b^*} \ (u = 1, \ldots, \upsilon).$$

	lower bound	median	upper bound
KA	0.150	0.192	0.234
BA	0.454	0.503	0.552
PA	0.160	0.173	0.186
KQ	0.204	0.251	0.298
BQ	0.339	0.365	0.391
PQ	0.183	0.194	0.206

Table 3: Coverage in Nonmonotonic Classic Kriging (K) and Monotonic Bootstrapped Kriging (B), and Polynomial Regression (P), for the Average (A) and the 90% Quantile (Q),), for n = 5, m = 5, T = 1000

Our corresponding $1-\alpha$ confidence interval for the true output ζ_u is

$$\widehat{y^*_{u;(\lfloor 0.05B \rfloor)}} < \zeta_u < \widehat{y^*_{u;(\lceil 0.95B \rceil)}}.$$

Note: The known shape of the polynomial regression model may be preserved through semidefinite programming and real algebraic geometry; see Siem, de Klerk, and den Hertog (2008).

References

Ankenman, B., B. Nelson, and J. Staum (2009), Stochastic kriging for simulation metamodeling. *Operations Research* (accepted)

Batur, D. and F. Choobineh (2009), A quantile-based approach to system selection. *European Journal of Operational Research*, in press

Chen, E.J. (2008), Some procedures of selecting the best designs with respect to quantile. *Simulation*, 84, pp. 275-284

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

Den Hertog, D., J.P.C. Kleijnen, and A.Y.D. Siem (2006), The correct Kriging variance estimated by bootstrapping. *Journal of the Operational Research Society*, 57, no. 4, pp. 400-409

Efron, B. and R.J. Tibshirani (1993), *An introduction to the bootstrap.* Chapman & Hall, New York

Feelders, A.J. (2000), Prior knowledge in economic applications of data mining. *Proceedings of the fourth European conference on principles and practice of knowledge discovery in data bases*, Springer, pp. 395-400

Forrester, A., A. Sóbester, and A. Keane (2008), *Engineering design via surrogate modelling: a practical guide.* Wiley, Chichester, United Kingdom

Frazier, P., W. Powell, and S. Dayanik (2009), The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing* (accepted)

Gross, D. and C.M. Harris (1998), *Fundamentals of queueing theory.* Wiley, New York

Hong, L.J. (2009), Estimating quantile sensitivities. *Operations Research*, 57. no. 1, pp. 118-130

Jin, X., M.C. Fu, and X. Xiong (2003), Probabilistic error bounds for simulation quantile estimators. *Management Science*, 14, no. 2, pp. 230-246

Joseph, V. R., Y. Hung, and A. Sudjianto, ((2008), Blind Kriging: a new method for developing metamodels. *Journal of Mechanical Design*, 130, no. 3, pp. 31-102

Karplus, W.J (1983), The spectrum of mathematical models. *Perspectives in Computing*, 3, no. 2, pp. 4-13

Kleijnen, J.P.C. (2008), *Design and analysis of simulation experiments.* Springer Science + Business Media

Kleijnen, J.P.C. (2009), Factor screening in simulation experiments: review of sequential bifurcation. *Advancing the Frontiers of Simulation: A Festschrift in Honor of George S. Fishman*, edited by C. Alexopoulos, D. Goldsman, and J. R. Wilson, Springer, New York, pp. 169-173

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

Kleijnen, J.P.C., W. van Beers, and I. van Nieuwenhuyse (2009), Constrained optimization in simulation: a novel approach. *European Journal of Operational Research* (accepted)

Law, A.M. (2007), *Simulation modeling and analysis; fourth edition.* McGraw-Hill, Boston

Lehmann, E. L. (1999), *Elements of large-sample theory*, Springer, New York

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

Sacks, J., W.J. Welch, T.J. Mitchell and H.P. Wynn (1989), Design and analysis of computer experiments (includes Comments and Rejoinder). *Statistical Science*, 4, no. 4, pp. 409-435

Santner, T.J., B.J. Williams, and W.I. Notz (2003), *The design and analysis of computer experiments.* Springer-Verlag, New York

Siem, A.Y.D. (2007), *Property preservation and quality measures in meta-models.* Ph.D. dissertation, Tilburg University, Tilburg, Netherlands

Siem, A.Y.D., E. de Klerk, and D. den Hertog (2008), Discrete least-norm approximation by nonnegative (trigonometric) polynomials and rational functions. *Structural Multidisciplinary Optimization*, 35, pp.

327–339

Simpson, T.W., A.J. Booker, D. Ghosh, A.A. Giunta, P.N. Koch, and R.-J. Yang (2004), Approximation methods in multidisciplinary analysis and optimization: a panel discussion. *Structural and Multidisciplinary Optimization*, 27, no. 5, pp. 302-313

Van Beers, W. and J.P.C. Kleijnen (2003), Kriging for interpolation in random simulation. *Journal of the Operational Research Society*, no. 54, pp. 255-262

Velikova, M. (2006), *Monotone models for prediction in data mining*, Ph.D. dissertation, Tilburg University, Tilburg, Netherlands

Yin, J., S.H. Ng, and K.M. Ng (2008), Kriging model with modified nugget effect. *Proceedings of the 2008 IEEE International Conference on Industrial Engineering and Engineering Management*, to appear

Zeigler B.P., H. Praehofer, T.G. Kim (2000), *Theory of modeling and simulation; second edition.* Academic Press, San Diego

Acknowledgment

We thank Dick den Hertog (Tilburg University) for his very useful comments on an earlier version.