

KRIGING INTERPOLATION IN SIMULATION: A SURVEY

Wim C.M. van Beers

Department of Information Systems and Management
Tilburg University
Postbox 90153, 5000 LE Tilburg
THE NETHERLANDS

Jack P.C. Kleijnen

Department of Information Systems and Management/
Center for Economic Research (CentER)
Tilburg University
Postbox 90153, 5000 LE Tilburg
THE NETHERLANDS

ABSTRACT

Many simulation experiments require much computer time, so they necessitate interpolation for sensitivity analysis and optimization. The interpolating functions are ‘metamodels’ (or ‘response surfaces’) of the underlying simulation models. Classic methods combine low-order polynomial regression analysis with fractional factorial designs. Modern Kriging provides ‘exact’ interpolation, i.e., predicted output values at inputs already observed equal the simulated output values. Such interpolation is attractive in deterministic simulation, and is often applied in Computer Aided Engineering. In discrete-event simulation, however, Kriging has just started. Methodologically, a Kriging metamodel covers the whole experimental area; i.e., it is global (not local). Kriging often gives better global predictions than regression analysis. Technically, Kriging gives more weight to ‘neighboring’ observations. To estimate the Kriging metamodel, space filling designs are used; for example, Latin Hypercube Sampling (LHS). This paper also presents novel, customized (application driven) sequential designs based on cross-validation and bootstrapping.

1 INTRODUCTION

In practice, experiments with simulation models (also called computer codes) often require much computer time; for example, we have heard of a simulation that requires 50 hours per run. Consequently, sensitivity analysis and optimization require the analysts to *interpolate* the observed input/output (I/O) data. The interpolating function is called a *metamodel* of the underlying simulation model (it is also called a response surface, emulator, compact model, auxiliary model, etc.) Interpolation treats the simulation model as a *black* box model (unlike Perturbation Analysis and Score Function methods). Classic interpolation methods use first-order or second-order polynomials; see Kleijnen (2004). In this paper, however, we survey an alternative type of metamodel, namely *Kriging* (named after Krige, the South-African mining engineer who developed this method in the 1950s; see

Cressie (1993) for historical details). Kriging has already realized a track record in geostatistics—see the classic textbook, Cressie (1993)—and in deterministic simulation—see the classic paper, Sacks et al. (1989) and the recent textbook, Santner, Williams, and Notz (2003).

Kriging provides *exact* interpolation, i.e., the predicted output values at ‘old’ input combinations already observed are equal to the simulated output values at those inputs (‘inputs’ are also called ‘factors’; ‘input combinations’ are also called ‘scenarios’). Obviously, such interpolation is appealing in deterministic simulation. Kriging and deterministic simulation are often applied in Computer Aided Engineering (CAE) for the (optimal) design of airplanes, automobiles, computer chips, computer monitors, etc.; see again Sacks et al. (1989) and—for updates—Meckesheimer et al. (2002) and Simpson et al. (2001).

However, Kriging for *discrete-event* simulation has just started. In 2002, a search of the ‘International Abstracts of Operations Research’ gave only two hits. A Google search for the term Kriging in the WSC proceedings gave only seven hits for 1998 through 2003; for example, Chick (1997) and Barton (1998). Earlier, Barton (1994) also proposed the application of Kriging to random simulations (such as discrete-event queueing simulations). Régnière and Sharov (1999) discuss spatial and temporal output data of a random simulation model for ecological processes. In this paper, we survey our recent research on Kriging in random simulation; details are given by Kleijnen and Van Beers (2004a) and Van Beers and Kleijnen (2003).

Methodologically speaking, a Kriging interpolator covers the whole experimental area; i.e., it is a *global* (not local) metamodel. Our research indicates that Kriging may give better global predictions than low-order polynomial regression metamodels.

Note: Regression may be attractive when looking for an explanation—not a prediction—of the simulation’s I/O behavior; for example, which inputs are most important; does the simulation show a behavior that the experts find acceptable (also see Kleijnen 1998)? Regression models such as polynomials are useful in the local search for optimization and in

screening for identification of the most important factors among hundreds of factors; again see Kleijnen (2004).

To estimate the *parameters* (or coefficients) of these polynomials, the analysts must run their simulation model for a set of (say) n combinations of the k input values; this set is called a *design* (or an $n \times k$ design matrix or plan). The analysts usually apply *classic designs* such as fractional factorials (for example, 2^{k-p} designs); see Myers and Montgomery (2002). Kriging, however, estimates its parameters through *space filling designs*, which we shall detail in Section 3. The simplest and most popular designs use Latin Hypercube Sampling (LHS), which is already part of spreadsheet add-on software, such as Crystal Ball and @Risk. We shall summarize our own novel *customized* sequential design approach (tailored to the actual simulation; i.e., application driven)—first presented for deterministic simulation in Kleijnen and Van Beers (2004b), and for random simulation in Van Beers and Kleijnen (2004).

We organize the remainder of our paper as follows. Section 2 presents Kriging basics (including its stationary covariance process), and compares Kriging with regression. Section 3 summarizes standard designs for Kriging metamodels, emphasizing LHS designs. Section 4 presents novel designs that are sequential and customized, using either cross-validation or bootstrapping. Section 5 gives conclusions and further research topics.

2 BASICS OF ORDINARY KRIGING

There are several types of Kriging, but we limit our discussion to so-called *Ordinary Kriging*. Its predictor for the ‘new’, unobserved (non-simulated) input \mathbf{x}_{n+1} —denoted by $\hat{Y}(\mathbf{x}_{n+1})$ —is a weighted linear combination of all the n ‘old’, already observed outputs \mathbf{x}_i :

$$\hat{Y}(\mathbf{x}_{n+1}) = \sum_{i=1}^n \lambda_i \cdot Y(\mathbf{x}_i) = \lambda' \cdot \mathbf{Y} \quad (1)$$

with $\sum_{i=1}^n \lambda_i = 1$, $\lambda = (\lambda_1, \dots, \lambda_n)'$ and $\mathbf{Y} = (Y(\mathbf{x}_1), \dots, Y(\mathbf{x}_n))'$; capital letters denote random variables. This Kriging assumes a single output per input combination; in case of multiple outputs, the predictor may be computed per output. For multivariate Kriging we refer to Wackernagel (2003). (Each input vector has k components; see Section 1.)

To quantify the Kriging weights λ in (1), Kriging derives the best linear unbiased estimator (BLUE). The bias criterion implies that the predictor for an input that has already been observed, equals the observed output (this property makes Kriging an exact interpolator). To account for the output (co)variances, Kriging assumes that *the*

closer the input data are, the more positively correlated their outputs are.

This assumption is modeled through a covariance process that is *second-order stationary*; i.e., the means and variances are constants (say) μ_Y^2 and σ_Y^2 , and the covariances of the outputs depend only on the ‘distances’ (say) h between the inputs. In fact, these covariances decrease as these distances. increase.

Kriging may use different types of covariance functions. For example, if there is a single input factor ($k = 1$), the *exponential* covariance function with parameter θ is

$$\text{cov}[Y(x_i), Y(x_j)] = \sigma_Y^2 \exp(-\theta |x_i - x_j|) \quad (2)$$

so the second factor in (2) represents the correlation function. Another example is the Gaussian function, which replaces the absolute value $|x_i - x_j|$ in (2) by the square $(x_i - x_j)^2$ (so it becomes smoother). A third example is the linear covariance function, which is used in Kleijnen and Van Beers (2004a, b) and Van Beers and Kleijnen (2003). In two examples, Kleijnen and Van Beers (2004b) find that it does not matter much whether linear or exponential covariance functions are used.

On one hand, the covariance functions tend to zero as the distance increases; again see (2) for an example. On the other hand, at zero distance there is still variation in the output, namely σ_Y^2 . In geostatistics this is called the *nugget* effect; for example, when going back to the ‘same’ spot, a completely different output (namely, a gold nugget) may be observed. In deterministic simulation, this variance is harder to interpret. Den Hertog, Kleijnen, and Siem (2004) experiment with deterministic simulations, and demonstrate how a given covariance function with a specific θ parameter may give particular output patterns; see Figure 1. Obviously, this pattern is not *white noise*, which implies uncorrelated outputs (white noise is assumed by classic regression analysis). This figure demonstrates what it means to model the I/O function of a *deterministic* simulation model by means of a *random* metamodel, namely, the random process specified by (1) and (2). More details follow in Section 3.2.

In practice there are multiple input factors ($k > 1$). The Kriging literature then assumes *product form* correlation functions; for example, (2) gives

$$\text{cov}[Y(x_i), Y(x_j)] = \sigma_Y^2 \prod_{g=1}^k \exp(-\theta_g |x_{i,g} - x_{j,g}|) \quad (3)$$

where the parameters θ_g quantify the relative importance of the various inputs, and $\mathbf{x}_{i,g}$ denotes the value of input g in input combination i .

Note: The geostatistics literature uses so-called *variograms* to model the covariances, whereas the deterministic simulation literature uses covariance functions such as (3). In our previous publications, we used the variogram, but currently we use the software developed by Lophaven, Nielsen, and Sondergaard (2002), which uses covariance functions in its Matlab toolbox.

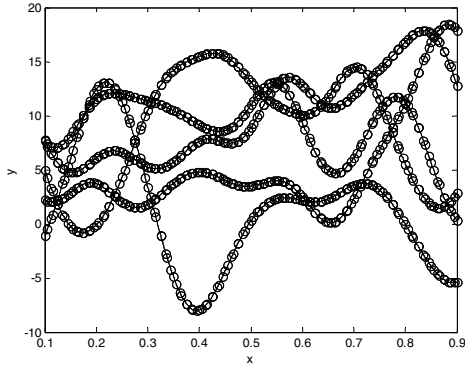


Figure 1: Realizations of a Covariance Stationary Process Model

Using the co-variances between the ‘old’ outputs and the covariances between the old and the new outputs, the optimal weights in (1) can be derived. We refer to the literature—and the corresponding software—for the rather complicated formula. The result is twofold:

- (i) observations closer to the input to be predicted get more weight (if the new input equals an old input, then the corresponding weight becomes one and all the other $n - 1$ weights become zero);
- (ii) the optimal weights vary with the specific input combination \mathbf{x}_{n+1} that is to be predicted (i.e., a different ‘new’ input has different neighbors with ‘old’ outputs that get more weight), whereas regression metamodels use fixed estimated parameters (say) $\hat{\beta}$.

Figure 2 illustrates Kriging versus second-order polynomial regression in a simulation experiment. More numerical illustrations—with constant and heterogeneous variances respectively—are given by Van Beers and Kleijnen (2003) and Kleijnen and Van Beers (2004a). In all these examples, Kriging gives better predictions than regression metamodels.

The literature (e.g., Cressie 1993, p. 122) and software also give—rather complicated—formulas for the predictor variance, $\text{var}(\hat{Y}(\mathbf{x}_{n+1}))$. However, the formulas for the optimal weights and the corresponding predictor variance are wrong! They neglect the fact that the covariances must be estimated; i.e., given a function type such as (3), the parameters θ_g must be estimated. (Most publications assume

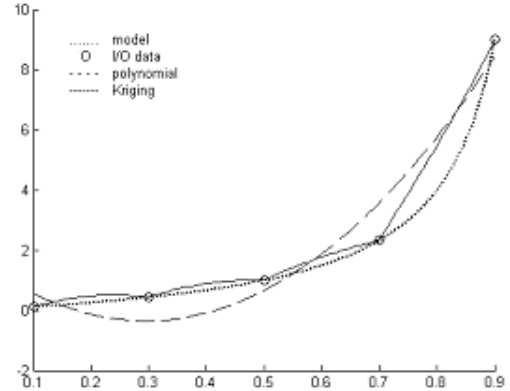


Figure 2: Kriging versus Second-Order Polynomial Regression in a Simulation Experiment

normally distributed output Y , and use maximum likelihood estimation—MLE—for these parameters; we, however, use ordinary least squares—OLS—in our linear covariance functions.) But these estimators make the predictor in (1) a non-linear predictor: replace λ by (say) L . Therefore we shall reconsider Kriging in Section 4.

3 STANDARD DESIGNS FOR KRIGING

In the preceding section, we focused on the Kriging analysis. Now we focus on the design of the experiment with a simulation model—that is going to be analyzed by Kriging. (When the analysts are going to analyze their I/O data by means of low-order polynomials, then they use designs such as fractional factorial designs.) ‘Design and analysis’ is a ‘chicken and egg’ problem.

In random simulations, we must confront the issue of *replication*. Even in *expensive* simulations—which require much computer time per replicate (or simulation run)—it is non-sense to obtain a few replicates if the signal/noise is then too small (so the analysts better toss a coin rather than develop and run a simulation model). Kleijnen and Van Beers (2004a, Figure 1) demonstrate that—in case of too few replicates—the estimated correlation function is very noisy—and hence the Kriging weights and predictions are very inaccurate.

After simulating a reasonable number of replicates (also see Law and Kelton 2000), Kriging is applied to the *average* output per input combination. Kleijnen and Van Beers (2004a) demonstrate that as the number of replicates increases, the Kriging predictor’s accuracy also increases. (Regression models, however, did not give better predictions; an explanation might be that regression models use the average output per input combination, but—whatever the number of replicates is—their average has the same expected value). Their conclusion is that ordinary Kriging works well in case of random simulation—provided replicates are obtained such that the signal/noise is acceptable; it is not necessary to take so many replicates that the sample averages get a constant variance.

The simplest and most popular designs use LHS; see Section 1 and Kleijnen (2004). LHS was invented by McKay, Beckman, and Conover (1979) for deterministic simulation models. Those authors did not analyze the I/O data by Kriging (but they did assume I/O functions more complicated than the polynomial models in classic DOE). LHS offers *flexible* design sizes n (number of input combinations actually simulated) for any k (number of simulation inputs). LHS proceeds as follows; also see the example for $k = 2$ factors in Figure 3.

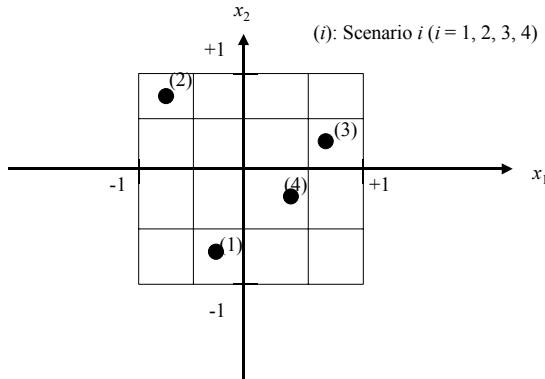


Figure 3: A LHS Design for Two Factors and Four Scenarios

- (i) LHS divides each input range into n intervals of equal length, numbered from 1 to n (so the number of values per input can be much larger than in designs for low-order polynomials).
- (ii) Next, LHS places these integers $1, \dots, n$ such that each integer appears exactly once in each row and each column of the design matrix.
- (iii) Within each cell of the design matrix, the exact input value may be sampled uniformly. (Alternatively, these values may be placed systematically in the middle of each cell. In risk analysis, this uniform sampling may be replaced by sampling from some other distribution for the input values.)

Because LHS implies randomness, its result may happen to be an *outlier*. For example, it might happen—with small probability—that two input factors have a correlation coefficient of -1 (all their values lie on the main diagonal of the design matrix). Therefore the LHS may be adjusted to become (nearly) orthogonal; see Ye (1998).

Classic designs simulate *extreme scenarios*—namely the corners of a k -dimensional square—whereas LHS has better *space filling* properties; again see Figure 3. This space filling property has inspired many statisticians to develop related designs. One type maximizes the minimum Euclidean distance between any two points in the k -dimensional experimental area. Other designs minimize the maximum distance. See Koehler and Owen (1996), Santner, Williams, and Notz (2003), and also Kleijnen et al. (2004).

4 NOVEL DESIGNS FOR KRIGING

The *classic Kriging* variance formula for the prediction error neglects the random character of the ‘optimal’ weights (see Section 2). The formula implies zero variances at all the n old input combinations—which is correct, as Kriging implies exact interpolation. (Also see Figure 7, which we discuss below.) To correct this formula, we propose three approaches to quantify the uncertainty of the Kriging predictor:

- (i) cross-validation for deterministic simulation;
- (ii) parametric bootstrapping for deterministic simulation;
- (iii) distribution-free bootstrapping for random simulation.

In the next three subsections, we summarize these three approaches. We focus on the application of these approaches to derive novel *designs* for the Kriging analysis of simulation experiments (for classic Kriging designs see Section 3). Two of these approaches have already been applied to derive *customized sequential* designs; see Kleijnen and Van Beers (2004a) and Van Beers and Kleijnen (2004).

Sequentialization means that—after a small pilot design—the I/O data obtained so far, are analyzed and used to decide on the next scenario to be simulated. In general, sequential procedures are known to be more ‘efficient’; that is, they require fewer observations than fixed-sample procedures. Moreover, simulation experiments proceed sequentially—unless run in batch mode or run on parallel computers. See Park et al. (2002) and also Sacks et al. (1989).

Moreover, our approach derives designs that are *customized*; that is, they are not *generic* designs (such as a classic 2^{k-p} designs or LHS).

To compare different designs for academic simulations with *known* outputs, we follow Sacks et al. (1989, p. 416) and compare the calculated predictions $\hat{y}(\mathbf{x}_j)$ with the known true values $y(\mathbf{x}_j)$ in a *test set* of size (say) m (so $j = 1, \dots, m$) through the Empirical Integrated Mean Squared Error (EIMSE):

$$EIMSE = \frac{1}{m} \sum_{i=1}^m (\hat{y}_i(\mathbf{x}) - y_i(\mathbf{x}))^2 \quad (4)$$

Note that such a criterion is more appropriate in sensitivity analysis than in optimization; see Kleijnen and Sargent (2000) and Sasena, Papalambros, and Goovaerts (2002).

4.1 Customized Sequential Designs for Deterministic Simulation

To customize our design, we estimate the true I/O function through a type of *cross-validation*; i.e., we successively delete one of the I/O observations already simulated (for cross-validation see again Meckesheimer et al. 2002 and

also Mertens 2001). In this way, we estimate the uncertainty of the output at input combinations not yet observed. To quantify this uncertainty, we use the *jackknifed* variance; see (6) below (for jackknifing in general, again see Meckesheimer et al. and Mertens).

It turns out that customized designs concentrate on scenarios in sub-areas that have *more interesting* I/O behavior. In Figure 4 (further discussed below), we avoid spending much time on the relatively flat part of the I/O function with two local hills; in Figure 5, we spend most of our simulation time on the ‘explosive’ part of the hyperbolic I/O function.

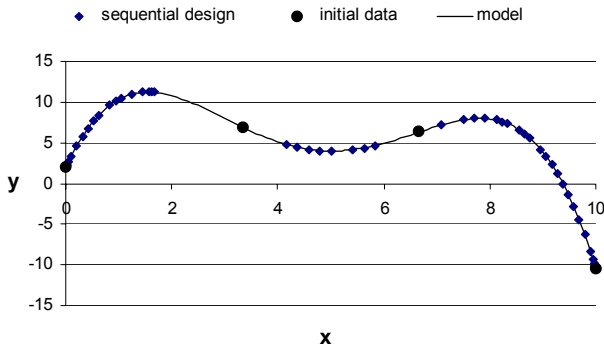


Figure 4: Customized Design for Example with Two Local Hilltops, Including Four Pilot Observations and 36 Additional Observations

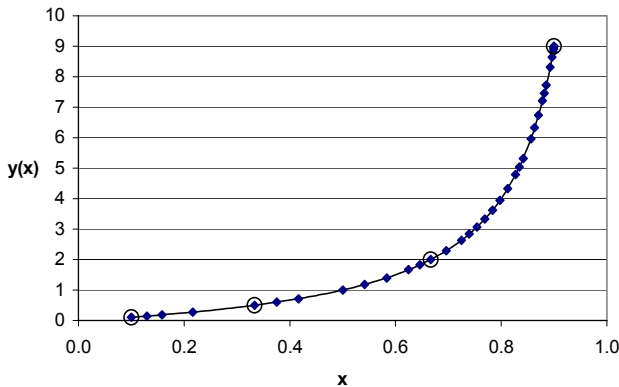


Figure 5: Customized Design for Hyperbolic Example, Including Four Pilot Observations and 36 Additional Observations

The procedure that leads to Figures 4 and 5, runs as follows. We start with a *pilot design* of size (say) n_0 . Kriging may give bad predictions in case of *extrapolation*. Therefore, we propose to start with the 2^k scenarios of a (classic) full factorial as a subset of the pilot design. Our two examples have a single input ($k = 1$), so one input value is the minimum and one is the maximum of the input’s range; see Figure 6 (other parts of this figure will be explained below). (Recently, however, we realized that the covariance function used in Kriging implies that values

‘close’ to the extreme scenarios might be better than the extremes themselves.)

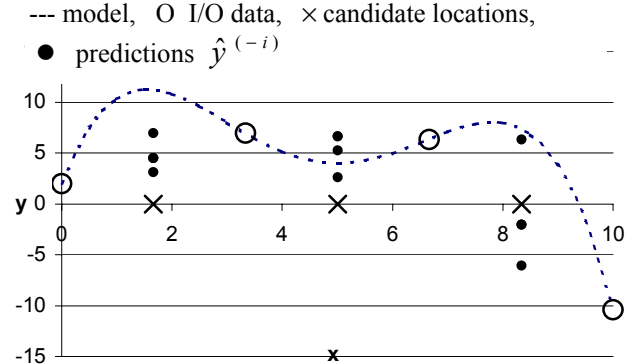


Figure 6: Customized Design for Example with Two Local Hilltops

Besides these 2^k scenarios, we must select some more input combinations to *estimate the covariance function*. Obviously, estimation of a linear covariance function requires at least three different values of the distance h ; thus at least three different I/O combinations. Moreover, cross-validation drops one of the n_0 observations and re-estimates the covariance function; i.e., cross-validation necessitates one extra I/O combination.

In the general case of more than one input, we may use a small *space-filling design* (see above). In our two examples, we take—besides the two endpoints of the factor’s range—two additional points such that all four observed points are equidistant; again see Figure 6.

After simulating the pilot design, we choose additional input combinations—accounting for the particular simulation model at hand. Because we do not know the I/O function of this model, we choose (say) c *candidate points*—without actually running any expensive simulations for these candidates!

First we must select a *value* for c . In the example of Figure 6 we select three candidate input values. In general, as we consider more candidates, we must calculate more Kriging predictions; the latter calculations take little computer time compared with the ‘expensive’ simulation computations.

Next we must select c *specific* candidates. Again, we use a space-filling design. In Figure 6 we select the three candidates *halfway* between the four input values already observed.

To select a ‘winning’ candidate for actual (expensive) simulation, we estimate the variance of the predicted output at each candidate input—using *cross-validation*. There are several types of cross-validation; for example, each time a single observation is dropped. To avoid extrapolation, we do not drop the observations at the vertices: we calculate the Kriging predictions from only (say) n_c observations. Next, we re-estimate the ‘optimal’ Kriging

weights. Figure 6 shows the re-computed $n_c = n_0 - 1 = 3$ Kriging predictions (say) $\hat{y}^{(-i)}$ after deleting observation i for each of the $c = 3$ candidates. This figure suggests that it is most difficult to predict the output at the candidate point $x = 8.33$ (see the solid circles or heavy dots).

To quantify this prediction uncertainty, we use *jackknifing*. First, we calculate the jackknife’s *pseudo-value* for candidate j :

$$\tilde{y}_{j,i} = n_c \times \hat{y}_j^{(-0)} - (n_c - 1) \times \hat{y}_j^{(-i)} \quad (5)$$

where $\hat{y}_j^{(-0)}$ is the original Kriging prediction for candidate input j based on the complete set of observations (zero observations eliminated: see the superscript -0).

From the pseudo-values in (5), we compute the *jackknife variance* for candidate j :

$$\tilde{s}_j^2 = \frac{1}{n_c(n_c - 1)} \sum_{i=1}^{n_c} (\tilde{y}_{j,i} - \bar{\tilde{y}}_j)^2 \quad \text{with} \quad \bar{\tilde{y}}_j = \frac{1}{n_c} \sum_{i=1}^{n_c} \tilde{y}_{j,i}. \quad (6)$$

To select the *winning* candidate among the c candidates for actual simulation, we find the candidate with the maximum jackknife variances in (6).

Once we have simulated the ‘winning’ candidate, we add the new observation to the set of observations. Next, we choose a new set of candidates with respect to this augmented set.

In the literature, we could not find an appealing stopping criterion for our sequential design; more research is needed. In practice, simulation experiments may stop prematurely (e.g., the computer may break down); our procedure then still gives useful information! Moreover, in one-shot (non-sequential) designs such as LHS, the users must *a priori* decide on the sample size.

Two simple examples with a single input ($k = 1$) were displayed in Figures 4 and 5. We saw that our design is intuitively appealing—but now we also use a *test set* to quantify its performance. In this test, we compare our design with two alternative design types of the same size ($n = 36$):

i. A *sequential* design based on the *approximate* Kriging variance formula discussed in Section 2. This design selects as the next point the input value that maximizes this variance (we do not need to specify candidate points); see Figure 7.

The figure illustrates that this approach selects the new input farthest away from the old inputs, namely $x = 0.5$. This results in a final design that spreads all its points evenly across the experimental area—so it resembles the next design.

ii. A *single-stage LHS design*. To reduce the variability in our test results, we obtain ten LHS samples—and average our results over these ten designs.

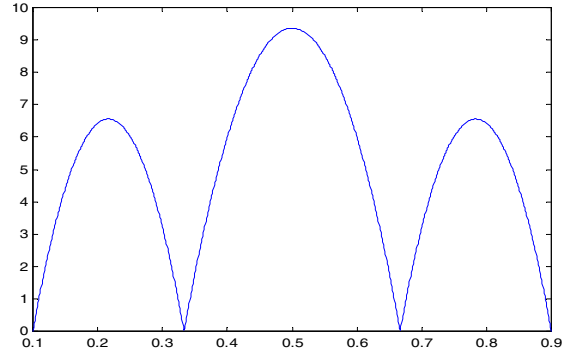


Figure 7: Approximate Variance of Kriging Prediction Error

Each design gives the Kriging predictors for the 32 true test values—which gives the EIMSE, defined in (4). Our customized sequential designs give substantially better results; for example, our EIMSE is only 15% of design (i) and 43% of design (ii) in the first example (for more information see Kleijnen and Van Beers 2004b, Table 2).

Note: We focus on sensitivity analysis, not optimization. For example, our design selects input values—not only near the ‘top’—but also near the ‘bottom’ in Figure 4. If we were searching for a maximum, we would adapt our procedure such that it would not collect data near an obvious minimum. Also see our discussion on further research in Section 5.

4.2 Parametric Bootstrap for Deterministic Simulation

To derive an unbiased estimator of the variance of the Kriging prediction error, Den Hertog et al. (2004) use *Monte Carlo* sampling from the assumed stationary Gaussian process with a type of covariance function specified by (say) equation (3). As parameters for this process, they use the MLE computed from a (pilot) sample of I/O observations. This Monte Carlo experiment is actually a parametric bootstrap; see Efron and Tibshirani (1993). They find that the classic formula *underestimates* the true variance of the Kriging predictor. Moreover, the true variance of the Kriging function may be less symmetric than the classic Figure 7.

Moreover, this bootstrapping demonstrates what it means to model the I/O function of a *deterministic* simulation by means of a *random* process; again see Figure 1. (More details will be reported orally at the WSC 2004 conference.)

4.3 Distribution-Free Bootstrap for Random Simulation

In random simulation, the analysts must obtain several replicates in order to get a clear signal/noise picture; again see Section 3. We propose to *resample* these replicates—with replacement; i.e., we propose distribution-free bootstrap—

ping; again see Efron and Tibshirani (1993). This gives the resampled I/O data (say) $(\mathbf{x}_i, Y_r^*(\mathbf{x}_i))$ where $Y_r^*(\mathbf{x}_i)$ denotes the r^{th} resampled value for input \mathbf{x}_i and $r=1, \dots, p$ where p denotes the number of replicates (for simplicity of notation we assume a constant number of replicates—as is often the case when common random numbers are used).

Next, we recompute the estimated Kriging weights \hat{L}^* and the corresponding predictor $\hat{Y}^*(\mathbf{x})$. Repeating this bootstrapping (say) b times gives the bootstrap variance:

$$\text{var}(\hat{Y}_i^*) = \frac{1}{(b-1)} \sum_{g=1}^b (\hat{Y}_{i,g}^* - \bar{\hat{Y}}_i^*)^2, \quad (7)$$

which is analogous to (6) (the latter equation is the jackknife variance estimator based on cross-validation). The variance estimator in (7) can be used to select the next scenario to be simulated—analogous to the customized sequential design in Section 4.1 with (6) now replaced by (7). More details can be found in Van Beers and Kleijnen (2004) (and will be reported orally at the WSC 2004 conference).

5 CONCLUSIONS AND FUTURE RESEARCH

Kriging gives *more accurate predictions* than regression does, because regression assumes that the prediction errors are white noise whereas Kriging allows errors that are correlated; i.e., the closer the inputs are, the more positive are the output correlations. Moreover, regression uses a single estimated parameter set for all input values, whereas Kriging adapts its parameters (Kriging weights) as the input to be predicted changes.

Besides a proper analysis, an appropriate *design* of the simulation experiments is essential. We discussed several alternatives for classic designs for polynomial regression analysis, namely LHS and our own customized sequential designs for Kriging analysis.

The development of Kriging analysis and designs for *random* simulation need much more research. More empirical work is required on Kriging in realistic random simulations, to see if the track record in deterministic simulation may be matched.

Recently, Van Beers and Kleijnen (2004) used distribution-free bootstrapping, and investigated the effects of non-constant variances (which occur in queuing simulations), common random numbers (which create correlations among the simulation outputs), and non-normality (Kriging uses maximum likelihood estimators of the weights, which assumes normality).

Future research may investigate alternative pilot-samples—sizes n_0 and components \mathbf{x} —and alternative sets of candidate points, for sequential customized designs. A good stopping criterion for these designs is also needed.

It seems interesting to compare our customized *sequential designs* with the alternatives of Cox and John (1995), Jin, Chen, and Sudjianto (2002), and Sasena et al. (2002). For example, Jin et al. (2002) also use ‘cross-validation’, but they do not use jackknifing; they do not use a set of space filling candidate input combinations—our empirical results are more promising than their results. Distinguishing between sensitivity analysis and optimization is also important.

Comparisons of different *metamodel types* (polynomial regression, Kriging, splines, rational functions, etc.) remains a challenging problem. For example, Clarke, Griebisch, and Simpson (2003) compare five metamodel types. Keys and Rees (2004) present a sequential design using splines (instead of Kriging). Kleijnen (2004) gives more references.

Multivariate outputs also deserve more research, since realistic simulation models give multiple responses; again see Wackernagel (2003).

REFERENCES

- Barton, R.R. (1998), Simulation metamodels. In Proceedings of the 1998 Winter Simulation Conference, eds. D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, 167-174. Los Alamitos, CA: IEEE Computer Society Press.
- Barton, R.R. (1994), Metamodeling: a state of the art review. In Proceedings of the 1994 Winter Simulation Conference, eds. J.D. Tew, S. Manivannan, D.A. Sadowski, and A.F. Seila, 237-244. New York: ACM Press.
- Chick, S.E. (1997), Bayesian analysis for simulation input and output. In Proceedings of the 1997 Winter Simulation Conference, eds. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, 253-259. New York: ACM Press.
- Clarke, S.M., J.H. Griebisch, and T.W., Simpson (2003), Analysis of support vector regression for approximation of complex engineering analyses. In Proceedings of DETC '03, ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago.
- Cox, D.D. and S. John (1995), SDO: a statistical method for global optimisation. In Proceedings of the ICASE/LaRC Workshop on Multidisciplinary Design Optimization, Philadelphia: eds. N.M. Alexandrov and M.Y. Hussaini, Siam, 315-329.
- Cressie, N.A.C. (1993). Statistics for Spatial Data, New York: Wiley.
- Den Hertog, D., J.P.C. Kleijnen, and A.Y.D. Siem (2004), The correct Kriging variance estimated by bootstrapping. Working Paper. Available online via <<http://center.kub.nl/staff/kleijnen/papers.html>> (accessed August 24, 2004).
- Efron B. and R.J. Tibshirani (1993). An introduction to the bootstrap. New York: Chapman & Hall.

- Jin, R, W. Chen, and A. Sudjianto (2002), On sequential sampling for global metamodeling in engineering design. In *Proceeds of DET '02, ASME 2002 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC2002/DAC-34092*, September 29-October 2, 2002, Montreal, Canada.
- Keys, A.C. and L.P. Rees (2004), A sequential-design metamodeling strategy for simulation optimization. *Computers & Operations Research* 31: 1911-1932.
- Kleijnen (1998), Experimental design for sensitivity analysis, optimization, and validation of simulation models. *Handbook of Simulation*, ed. J. Banks, 173-223, New York: Wiley.
- Kleijnen (2004), Invited review: An overview of the design and analysis of simulation experiments for sensitivity analysis. *European Journal of Operational Research* (in press).
- Kleijnen, J.P.C. S.M. Sanchez, T.W. Lucas, and T.M. Cioppa (2004), A user's guide to the brave new world of designing simulation experiments. Working Paper. Available online via <http://center.kub.nl/staff/kleijnen/papers.html/> (accessed August 24, 2004).
- Kleijnen, J.P.C and R.G. Sargent (2000), A methodology for the fitting and validation of metamodels in simulation. *European Journal of Operational Research*, 120, (1): 14-29.
- Kleijnen, J.P.C. and W.C.M. van Beers (2004a), Robustness of Kriging when interpolating in random simulation with heterogeneous variances: some experiments. *European Journal of Operational Research* (in press)
- Kleijnen, J.P.C. and W.C.M. van Beers (2004b), Application-driven sequential designs for simulation experiments: Kriging metamodeling. *Journal of the Operational Research Society* 55: 876-883.
- Koehler, J.R. and A.B. Owen (1996), Computer experiments. In *Handbook of Statistics*, Volume 13, eds. S. Ghosh, C.R. Rao, 261-308. Amsterdam: Elsevier.
- Law, A.M. and W.D. Kelton (2000), *Simulation modeling and analysis*. 3rd edition. New York: McGraw-Hill.
- Lophaven, S.N., H.B. Nielsen, and J. Sondergaard (2002), *DACE: a Matlab Kriging toolbox*, version 2.0. IMM Technical University of Denmark, Lyngby.
- McKay, M.D., R.J. Beckman, and W.J. Conover (1979), A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21 (2): 239-245 (reprinted in 2000: *Technometrics* 42 (1): 55-61).
- Meckesheimer, M., R.R. Barton, T.W. Simpson, and A.J. Booker (2002), Computationally inexpensive metamodel assessment strategies. *AIAA Journal* 40 (10): 2053-2060.
- Mertens, B.J.A (2001), DOWNDATING: interdisciplinary research between statistics and computing. *Statistica Neerlandica* 55 (3): 358-366.
- Myers, R.H. and D.C. Montgomery (2002), *Response surface methodology: process and product optimization using designed experiments*. 2nd. edition. New York: Wiley.
- Park, S., J.W. Fowler, G.T. Mackulak, J.B. Keats, and W.M. Carlyle (2002), D-optimal sequential experiments for generating a simulation-based cycle time-throughput curve. *Operations Research* 50 (6): 981-990.
- Régnière, J. and A. Sharov (1999), Simulating temperature-dependent ecological processes at the sub-continental scale: male gypsy moth flight phenology. *International Journal of Biometereology* 42: 146-152.
- Sacks, J., W.J. Welch, and T.J. Mitchell, H.P. Wynn (1989), Design and analysis of computer experiments. *Statistical Science* 4 (4): 409-435.
- Santner, T.J., B.J. Williams, and W.I. Notz (2003), *The design and analysis of computer experiments*. New York: Springer-Verlag.
- Sasena, M.J, P. Papalambros, and P. Goovaerts (2002), Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization* 34 (3): 263-278.
- Simpson, T.W., T.M. Mauery, J.J. Korte, and F. Mistree (2001), Kriging metamodels for global approximation in simulation-based multidisciplinary design optimization. *AIAA Journal* 39 (12): 2233-2241.
- Van Beers, W.C.M. and J.P.C. Kleijnen (2003), Kriging for interpolation in random simulation. *Journal of the Operational Research Society* 54: 255-262.
- Van Beers, W.C.M. and J.P.C. Kleijnen (2004), Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. Available online via <http://center.kub.nl/staff/kleijnen/papers.html/> (accessed August 24, 2004).
- Wackernagel, W. (2003), *Multivariate Geostatistics: An Introduction with Applications*. 3rd edition. Berlin: Springer-Verlag.
- Ye, K.Q. (1998), Orthogonal column Latin hypercubes and their application in computer experiments. *Journal Association Statistical Analysis, Theory and Methods* (93): 1430-1439.

AUTHORS BIOGRAPHIES

WIM VAN BEERS is a Ph.D. student in the Department of Information Management of Tilburg University. His research topic is Kriging in simulation. He will defend his thesis in the Spring of 2005. His e-mail address is wvbeers@uvt.nl.

JACK P.C. KLEIJNEN is a Professor of Simulation and Information Systems in the Department of Information Management; he is also a member of the Operations Research group of CentER (Center for Economic Research) of Tilburg University. His research concerns computer simulation (especially the statistical design and analysis of simulation ex-

periments), information systems, and supply chains. He has been a consultant for several organizations in the USA and Europe, and served on many international editorial boards and scientific committees. He spent several years in the USA, at universities and private companies. He received a number of international awards. More information can be found on his web page, including details on his publications (nearly 200), seminars, simulation course, academic honors, previous employment, education, 'who is who' listings, etc. His e-mail address is <kleijnen@uvt.nl> and his web address is <<http://center.uvt.nl/staff/kleijnen/>>.