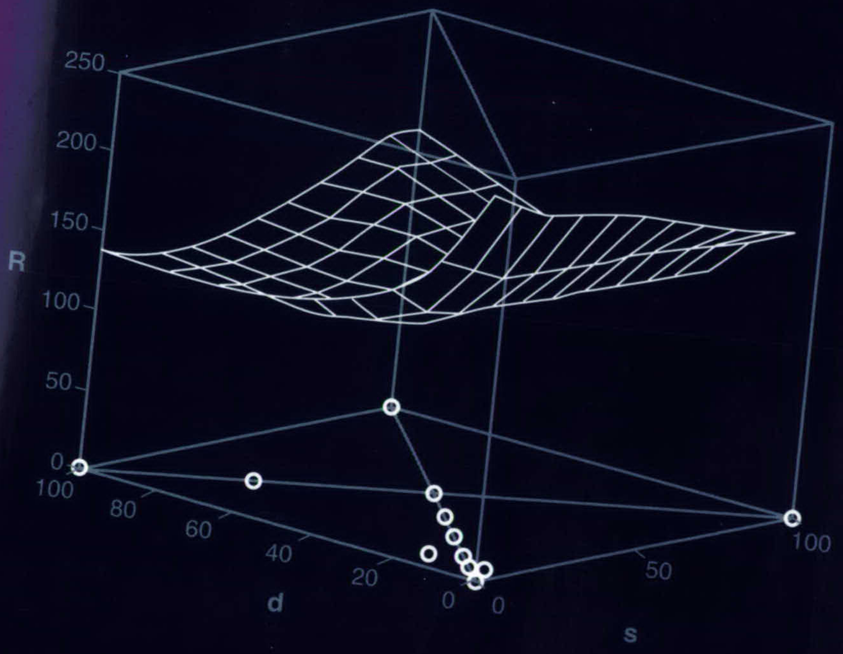
Kriging Metamodeling for Simulation

Wim C.M. van Beers



STELLINGEN

behorende bij het proefschrift

Kriging Metamodeling for Simulation

door Wim C.M. Van Beers

1. In 'discrete-event' simulatie geeft Kriging betere voorspellingen dan polynomiale regressiemodellen.
(dit proefschrift)

2. Kriging interpolatie is robuust ten aanzien van de klassieke modelveronderstelling dat de respons een constante variantie heeft.
(dit proefschrift)

3. Voor tijdrovende simulaties zijn modelafhankelijke sequentiële proefopzetten te prefereren boven klassieke proefopzetten.
(dit proefschrift)

4. Hoewel het simuleren van de gemiddelde wachttijden voor een gegeven bezettingsgraad van een M/M/1 model door middel van een niet-lineaire stochastische differentievergelijking uit principiële overwegingen de voorkeur heeft, is het voor toetsdoeleinden efficiënt en effectief om de analytisch verwachte wachttijden te verhogen met *NIID*-trekkingen.
(dit proefschrift)

KRIGING METAMODELING FOR SIMULATION

KRIGING METAMODELING FOR SIMULATION

Proefschrift

ter verkrijging van de graad van doctor
aan de Universiteit van Tilburg,
op gezag van de rector magnificus, prof. dr. F.A. van der Duyn Schouten,
in het openbaar te verdedigen ten overstaan van
een door het college voor promoties aangewezen commissie
in de aula van de Universiteit
op vrijdag 28 oktober 2005 om 14.15 uur door

Wilhelmus Cornelis Maria van Beers,

geboren op 10 november 1952 te Loon op Zand

Promotor: Prof. dr. J.P.C. Kleijnen

Acknowledgment

This thesis is the result of my research during six years in the Operations Research group of the Department of Information Systems and Management in the Faculty of Economics and Business Administration at Tilburg University. During that time I met many people who supported me in various ways. They made my stay productive and—not in the least—pleasant. To some of them I owe special thanks.

First of all, I am extraordinarily grateful to my supervisor, Professor Jack Kleijnen. He guided me through the fascinating world of random simulation and taught me to search for the intuition ('wiedes') behind the results. We had many valuable discussions, and he encouraged me to develop the ideas that we discussed in such a way that they could be published in international journals. I am grateful for the many critical comments he made—in red, green, and blue. The existence of this thesis is largely due to him.

Also special thanks are due to the members of my Ph.D. committee: Professors H. Daniels, D. den Hertog, T. Dhaene, J. Rooda, and H. Wynn for their careful reading, thoughtful suggestions, and corrections. I feel honored to have them in my committee.

Among all the others who supported me during my Ph.D. study, I particularly wish to thank our daughter, Miranda. Her typical mental support was a real encouragement to me.

Last but certainly not least, I am most grateful to my wife, Marianne. She patiently gave me all the time and space I needed to venture into those things that I found fascinating. There are no words to describe the emotional and loving way she supported me. Undoubtedly, the biggest thanks are for her!

Contents

Chapter 1

Introduction to Kriging Metamodeling for Simulation

1.1 Real systems and mathematical models

Many scientific disciplines use mathematical models (including simulation; see below) to describe complicated real systems. The goal of such models is getting more *insight* into the real system, to answer questions such as: What is the output's sensitivity to the inputs; what is the optimal combination of the input values?

To obtain such insight, analytical methods (e.g., differential calculus) often turn out to fail. In such cases, *numerical methods* may be tried; i.e., experimentation with the model may answer the questions about the real system. This experimentation often implies that the model is converted into a computer code (or computer program) that is run for a number of different input combinations. Next, the resulting input/output (I/O) behavior of the model should be analyzed.

We emphasize that the selection of the input combinations should be guided by scientific principles (e.g., changing one input at a time can be proven to be ineffective if inputs interact, and inefficient else). These principles are investigated in mathematical statistics under the name *Design Of Experiments* (DOE). In numerical experiments (as opposed to experiments with the real system) these principles need adjustment; see Kleijnen et al. (2005) and Chapters 4 and 5.

We focus on *computer expensive* simulation experiments; i.e., experiments that require much computer time. In these situations, DOE is certainly needed.

Moreover, the *analysis* of these I/O data should also be guided by scientific principles. Making scatter plots and other graphs may be the beginning of such an analysis. Objective analysis requires formal methods. Most popular are Least Squares (LS) curves fitted to the I/O data. We, however, focus on an alternative method known as Kriging; see Section 1.4.

Next, we shall discuss the key terms in the title of this chapter and this thesis.

1.2 Simulation types

A simulation model might be a physical model, e.g., a scale model of a racing car in a wind tunnel. We, however, limit ourselves to mathematical models (also see the preceding section). Following Kleijnen (1974), we define simulation as *experimenting with a model over time*. This definition indicates that the variable time plays a special role in simulation. Indeed, Law and Kelton (2000) also emphasize the role of time, by using the term *dynamic* simulation. Besides this simulation type, there is *static* simulation, in which time plays no role; an example is the *Monte Carlo* (MC) method. By definition, the MC method uses Pseudo-Random Numbers (PRN); i.e., computer generated numbers between zero and one that are independent and uniformly distributed over this interval. We shall also use MC models in this dissertation (see Chapters 2, 3, and 4).

There are deterministic and random simulation models. If the simulation model contains no random components, the simulation model is called *deterministic*. For example, such simulation might solve a set of complicated differential equations describing the airflow around air wings. Simpson et al. (2001) mention applications of deterministic simulation in various disciplines. Typically, in this type of simulation, an input combination needs simulation only once (repeated computer runs with the same input combination give exactly the same output value, provided we do not make any changes in the computer software or hardware).

Whenever there are probabilistic components (or modules) in the simulation model, the simulation is called *random* or *stochastic*. Then, an input combination should be simulated several times, and the outputs' average or quantile may be computed to estimate the model's output of interest. Typical examples are queueing and inventory simulations. Table 1 summarizes the four different simulation types; each cell refers to a chapter of this thesis that uses the specific simulation type. Note that we do not study dynamic, deterministic simulation models (often used in Computer Aided Engineering, CAE; see for example De Geest et al. (1999)).

Table 1: Simulation types with appropriate dissertation chapters

	Deterministic	*Random*
Static	Chapter 4	Chapters 2 and 3
Dynamic		Chapter 5

Finally, we discuss a third way to distinguish various simulation types: continuous versus discrete-event simulation. *Continuous* simulation experiments with models consisting of differential equations (so state variables change continuously). Computer codes approximate these differential equations by difference equations. These models are usually deterministic (see the preceding discussion, summarized in Table 1). Obviously, random elements may be added to the differential equations; for example, econometric models may consist of difference equations plus additive noise.

Discrete-event simulation has state variables that change instantaneously, at points in time that are not necessarily equidistant; for example, customers arrive at a server at random points of time. An *event* is the change in the system's state; for example, the number of waiting customers increases. Random simulation includes discrete-event event simulation; see Van Beers and Kleijnen (2005).

Simulation—in its many forms—is applied in many scientific disciplines. We focus on the discipline known as Operations Research/Management Science (OR/MS). In OR/MS, simulation is also often applied—even though simulation is described as method of last resort 'when all else fails …' in the famous Chapter 21 in Wagner (1975).

1.3 Metamodels

A metamodel is also called a *response surface*, *auxiliary model*, *emulator*, etc. Metamodeling originates from neuro-linguistics in the beginning of the twentieth century; see Korzybski and Meyers 1958. Kleijnen et al. (2005) define a metamodel as an approximation of the true I/O function implicitly defined by the given simulation model. Obviously, a metamodel is much simpler than the underlying simulation model; an example is a first-order polynomial regression (meta)model of the I/O function of a queueing simulation. A metamodel treats the simulation model as a *black box*; i.e., it uses the I/O data without knowledge of the way the simulation model processes these inputs to get the outputs.

A metamodel is a tool for the systematic experimentation and analysis of a simulation model, to gain insight (again see Section 1.1). Kleijnen (1998) discusses the use of metamodeling for

(i) sensitivity analysis

(ii) optimization

(iii) validation and verification.

We shall focus on sensitivity analysis.

There are many types of metamodels; see Kleijnen (2005). Most popular are low-order polynomial regression models. However, *Kriging* is also applied frequently in deterministic simulation; see Den Hertog and Stehouwer (2002), and Simpson et al. (1998). Because other types of metamodels do not play a role in this thesis and we wish to avoid errors of omission, we refrain from listing (and defining) these types and giving key references. In this thesis, we focus on Kriging, and compare its performance with classical regression.

Atkinson (1989, ch.3) defines *interpolation* as the selection of a function in such a way that its graph passes through a finite set of given data points. Kriging applied to deterministic simulation meets that definition, as we shall see in the next section. In practice, the term interpolation is also used to mean a function that approximates the I/O data; for example, a regression model may fit the I/O data such that it minimizes the sum of squared approximation errors (LS criterion).

1.4 Kriging

In the 1950s, the South African mining engineer *D.G. Krige* (born in 1919, and still alive) devised an interpolation method to determine true ore-bodies, based on samples. The basic idea is that these predictions are weighted averages of the observed outputs, where the weights depend on the distances between the input location to be predicted and the input locations already observed. The weights are chosen so as to minimize the prediction variance, i.e., the weights should provide a Best Linear Unbiased Estimator (BLUE) of the output value for a given input. Therefore, Kriging is also called Optimal Interpolation.

The dependence of the interpolation weights on the distances between the inputs was mathematically formalized by the French mathematician Georges Matheron (1930-2000) in his monumental 'Traité de géostatistique appliquée' (1962). He introduced a function, which he called a *variogram*, to describe the variance of the difference between two observations; also see Chapter 2.

So Kriging originated in geostatistics to answer concrete questions in the gold mining industry: Drilling for ore—deep under the ground—is expensive, so efficient prediction methods are necessary. Later on, Kriging was successfully introduced into deterministic simulation by Sacks et al. (1989). In this thesis we introduce Kriging interpolation into the area of random simulation.

Kriging provides 'exact' interpolation, i.e., predicted output values at inputs already observed equal the observed output values. Such interpolation is attractive in deterministic simulation, and Kriging is often applied in CAD (mentioned in Section 1.2). In discrete-event simulation, however, Kriging has just started.

Above, we mentioned that the variogram is the cornerstone in Kriging. Hence, accurate estimation of the variogram, based on the observed data, is essential. Journel and Huijbregts (1978, pp. 161-195) present various parametric variogram models. The values of its parameters are obtained by either Weighted Least Squares (WLS) or Maximum Likelihood Estimation (MLE); see Cressie (1993). In this thesis, we shall use both methods: WLS in Chapters 2, 3, and 4, and MLE in Chapter 5.

1.5 Kriging applications area

After the introduction of Kriging in the mining industry, it became very popular in other areas of geostatistics, such as meteorology, oceanography, agriculture and environment studies; see Cressie (1993).

After the pioneering article of Sacks (1989), Kriging has also been widely applied in deterministic simulation for engineering, aimed at the design of better computer chips, television screens, aircraft, and automobiles.
In this thesis we introduce Kriging for simulation in OR/MS, covering both deterministic and random simulations (in different chapters). We limit ourselves to sensitivity analysis; we feel that sensitivity analysis precedes optimization, so we leave Kriging for optimization as future research (also see Chapter 6).

1.6 Summary of thesis

This thesis contains the full text of four papers that either have already been published or have been submitted for publication. It is the verbatim text, except for three sentences in Chapter 2, indicated by footnotes. In this section we summarize each paper.

In Chapter 2, we introduce Kriging interpolation for *random* simulation. We develop a novel type of Kriging interpolation, and call it Detrended Kriging. We demonstrate Kriging through two numerical examples:

(i) a hyperbole inspired by the single-server queueing simulation with Poisson (Markov) arrival and service processes (known as the M/M/1 queueing model)

(ii) an artificial model, namely a fourth degree polynomial with multiple modes plus additive noise.

We test our novel method through cross-validation, and compare it to low-order regression metamodels fitted through Ordinary Least Squares (OLS). The main conclusion of this chapter is that Kriging gives better predictions than these regression models. Furthermore, tests show that the Kriging's so-called *nugget effect* equals the variance of the additive noise.

In Chapter 3, we drop the classic Kriging assumption of outputs with *constant* variances. In practice, this assumption is not realistic. Therefore we investigate the consequences of Kriging in case of a true I/O function that is a hyperbole plus noise with variances differing with the input. The main conclusion of this chapter is that Kriging is not sensitive to variance heterogeneity; i.e, Kriging is a robust method that still outperforms low-order polynomial regression.

In Chapter 4, we propose a novel method to select *experimental designs* for Kriging. Our method is sequential, because we focus on expensive deterministic simulation. Our design differs from traditional designs such as Latin Hypercube Sampling (LHS); see McKay, Beckman, and Conover (1979), and also Koehler and Owen (1996), and Kleijnen et al. (2005). More specifically, our method accounts for the specific I/O function implied by the underlying simulation model. Our customized, tailor-made designs are constructed through cross-validation and jackknifing. We test our method through the two academic applications that we also used in Chapter 2. Our main conclusions are that the novel method simulates relatively more inputs in the more interesting parts of the underlying I/O function, and it gives better predictions than traditional LHS designs.

In Chapter 5, we extend the method of Chapter 4 to random simulation, especially discrete-event simulation. However, customization is now based on bootstrapping instead of cross-validation. We test our method through two discrete-event simulation models that are classic in OR/MS, namely the M/M/1 queueing model and an (*s, S*) inventory management model. We again compare the performance of our method with classical LHS designs. It turns out that our design indeed gives better results.

Finally, in Chapter 6 we summarize the conclusions of the preceding chapters. We also discuss the advantages and the disadvantages of Kriging. We finish with possible topics for future research.

References

Atkinson, K.E. (1989), *An introduction to numerical analysis*. John Wiley & Sons, Inc., New York

Cressie, N.A.C. (1993), *Statistics for spatial data*. John Wiley & Sons, Inc., New York

De Geest, J., T. Dhaene, N. Fache, and D. De Zutter (1999), Adaptive CAD-model building algorithm for general planar microwave structures. *IEEE Transactions on Microwave Theory and Techniques*, 47, no. 9, pp. 1801-1809

Den Hertog, D., and H.P. Stehouwer (2002), Optimizing color picture tubes by high-cost non-linear programming. *European Journal of Operational Research,* 140(2), 197-211

Journel, A.G. and Huijbregts, C.J. (1978), *Mining Geostatistics.* Academic Press, London

Kleijnen, J.P.C. (1974), *Statistical techniques in simulation, part I.* Marcel Dekker Inc., New York

Kleijnen, J.P.C. (1998), Experimental design for sensitivity analysis, optimization, and validation of simulation models. Chapter 6 in: *Handbook of Simulation,* edited by J. Banks, Wiley, New York, pp. 173-223

Kleijnen, J.P.C. (2005), 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 (2005), A user's guide to the brave new world of designing simulation experiments. *INFORMS Journal on Computing* (accepted as State-of-the-Art Review)

Koehler, J.R. and A.B. Owen (1996), Computer experiments. *Handbook of statistics,* by S. Ghosh and C.R. Rao, vol. 13, pp. 261-308

Korzybski, A. and R. Meyers (1958), *Science and sanity: an introduction to non-aristotelian systems and general semantics; fourth edition.* International Non-Aristotelian Library Publishing Company, Lakeville

Law, A.M. and W.D. Kelton (2000), *Simulation modeling and analysis; third edition.* McGraw-Hill, Boston

Matheron, G. (1962), Traité de géostatistique appliquée. *Memoires du Bureau de Recherches Geologiques et Minieres,* Editions Technip, Paris, no.14: pp. 57-59

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, no. 2, pp. 239-245 (reprinted in 2000: *Technometrics*, 42, no. 1, pp. 55-61

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

Simpson, T.W., J.J. Korte, T.M. Mauery, and F. Mistree (1998), Comparison of response surface and Kriging models for multidisciplinary design optimization. *AIAA Journal*, vol. 1, pp. 381-391

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, no. 12, 2001, pp. 2233-2241

Van Beers, W.C.M. and J.P.C. Kleijnen (2005), Customized sequential designs for random simulation experiments: Kriging metamodeling and bootstrapping. *Submitted for publication*

Wagner, H. M.(1975), *Principles of operations research with applications to managerial decisions*. Prentice Hall, London

Chapter 2

Kriging for Interpolation in Random Simulation

Abstract

Whenever simulation requires much computer time, interpolation is needed. Simulationists use different interpolation techniques (for example, linear regression), but this paper focuses on Kriging. This technique was originally developed in geostatistics by D. G. Krige, and has recently been widely applied in deterministic simulation. This paper, however, focuses on random or stochastic simulation. Essentially, Kriging gives more weight to 'neighbouring' observations. There are several types of Kriging; this paper discusses—besides Ordinary Kriging —a novel type, which 'detrends' data through the use of linear regression. Results are presented for two examples of input/output behaviour of the underlying random simulation model: Ordinary and Detrended Kriging give quite acceptable predictions; traditional linear regression gives the worst results.

2.1 Introduction

A primary goal of simulation is *what if* or sensitivity analysis: What happens if inputs of the simulation model change? Therefore simulationists run a given simulation program—or computer code—for (say) n different combinations of the k simulation inputs. We assume that

Paper by Van Beers, W.C.M. and J.P.C. Kleijnen, *Journal of the Operational Research Society*, no. 54, 2003, pp. 255-262 these inputs are either parameters or quantitative input variables of the simulation model. Typically, Kriging assumes that the number of values per input variable is quite ‘big’, certainly exceeding two (two values are used in simulation experiments based on 2^{k-p} designs).

Given this set of *n* input combinations, the analysts run the simulation and observe the outputs. Note that most simulation models have multiple outputs, but in practice these outputs are analysed *per* output type.

The crucial question of this paper is: How to *analyse* this simulation input/output (I/O) data? Classic analysis uses linear-regression (meta)models; see Kleijnen (1998). A *metamodel* is an approximation of the I/O transformation implied by the underlying simulation program. Many other terms are popular in certain disciplines: Response surface, compact model, emulator, etc. Such a metamodel treats the simulation model as a *black box*; that is, the simulation model's I/O is observed, and the parameters of the metamodel are estimated. This black-box approach has the following advantages and disadvantages.

An *advantage* is that the metamodel can be applied to all types of simulation models, either deterministic or random, either in steady-state or in transient state. A *disadvantage* is that it cannot take advantage of the specific structure of a given simulation model, so it may take more computer time compared with techniques such as perturbation analysis and score function.

Metamodelling can also help in optimization and validation of the simulation model. In this paper, however, we do not discuss these two topics, but refer to the references of this paper. Further, if the simulation model has hundreds of inputs, then special ‘screening’ designs are needed, discussed in Campolongo, Kleijnen, and Andres (2000). In our examples—but not in our methodological discussion—we limit the number of inputs to the minimum, namely a single input.

Whereas polynomial-regression metamodels have been applied extensively in discrete-event simulation (such as queueing simulation), *Kriging has hardly been applied to random simulation*: A search of IAOR (International Abstracts of Operations Research) gave only two hits. However, in deterministic simulation (applied in many engineering disciplines; see our references), Kriging has been applied frequently, since the pioneering article by Sacks et al. (1989). In such simulation, Kriging is attractive because it can ensure that the metamodel’s prediction has exactly the same value as the observed simulation output as we shall see below.

In random simulation, however, this Kriging property may not be so desirable, since the observed (average) value is only an estimate of the true, expected simulation output. Unfortunately, Kriging requires extensive computation, so adequate software is needed. We discovered that for random simulation no software is available, so we developed our own software, in *Matlab*.

Note that several types of *random simulation* may be distinguished:
(i) Deterministic simulation with randomly sampled inputs. For example, in investment analysis we can compute the cash flow development over time through a spreadsheet such as Excel. Next, we sample the random values of inputs—such as the cash flow growth rate—by means of either Monte Carlo or Latin Hypercube Sampling (LHS) through an add-on such as @Risk or Crystal Ball; see Van Groenendaal and Kleijnen (1997)
(ii) Discrete-event simulation. For example, classic queueing simulation is applied in logistics and telecommunications.
(iii) Combined continuous/discrete-event simulation. For example, simulation of nuclear waste disposal represents the physical and chemical processes through deterministic non-linear difference equations and models the human interventions as discrete events (see Kleijnen and Helton, 1999).

Our research contribution consists in the development of a novel (namely, detrended) Kriging type, and the exploration of how well this Kriging type performs compared with Ordinary Kriging and traditional polynomial-regression modeling. The main conclusion of our examples is: A perfectly specified detrending function gives best predictions; Ordinary Kriging is acceptable; the usual linear regression gives the worst results.

We organize the remainder of this paper as follows. First we sketch the history of Kriging and its application in geology, metereology, and deterministic simulation. Then we describe the basics of Kriging, and give a formal Kriging model. Next we introduce our novel model for detrending the I/O data through low-order polynomial regression, including a classic cross-validation test. We illustrate this Kriging through two simple examples. In a separate section we give a third random simulation example to study the so-called nugget effect in Kriging. Finally, we present conclusions and mention possible future research topics.

2.2 Kriging

2.2.1 History of Kriging

Kriging is an interpolation technique originally developed by D. G. Krige, a South African mining engineer. In the 1950s he devised this method to determine true ore-grades, based on samples. Next, he improved the method in cooperation with G. Matheron, a French mathematician at the 'Ecole des Mines'. At the same time, in meteorology L. Gandin (in the former Soviet Union) worked on similar ideas, under the name 'optimum interpolation' (see Cressie, 1993).

Nowadays, Kriging is also applied to I/O data of *deterministic simulation* models; we refer again to Sacks et al. (1989)'s pioneering article. Many more publications followed; for example, Meckesheim et al. (2001) give 35 references. Also see Koehler and Owen (1996), and Jones, Schonlau, and Welch (1998).

2.2.2 Basics of Kriging

Kriging is an approximation method that can give predictions of unknown values of a random function, random field, or random process. These predictions are *best linear unbiased estimators*, under the Kriging assumptions presented in the next subsection.

Actually, these predictions are *weighted* linear combinations of the observed values. Kriging assumes that *the closer the input data are, the more positively correlated the prediction errors are*. Mathematically, this assumption is modeled through a *second-order stationary covariance process*: The expectations of the observations are constant and do not depend on the location (the input values), and the covariances of the observations depend only on the 'distances' between the corresponding inputs. In fact, these covariances decrease with the distance between the observations. The prediction criterion is minimum mean squared prediction errors. The result is an estimated metamodel such that observations closer to the prediction point get more weight in the predictor. When predicting the output for a location that has already been observed, then the prediction equals the observed value. (In deterministic simulation this property is certainly attractive, as we said above.)

In Kriging, a crucial role is played by the *variogram*: A diagram of the variance of the difference between the measurements at two input locations; also see Figure 1, which has symbols explained in the next subsection. The assumption of a second-order stationary covariance process implies that the variogram is a function of the distance (say) *h* between two locations. Moreover, the further apart two inputs are, the smaller this dependence is—until the effect is negligible.

2.2.3 Formal model for Kriging

A *random process* Z(•) can be described by $\{Z(\mathbf{s}): \mathbf{s} \in D\}$ where *D* is a fixed subset of R^d and Z(**s**) is a random function at location $\mathbf{s} \in D$; see Cressie (1993, p. 52).

There are several types of Kriging, but we limit this subsection to *Ordinary Kriging*, which makes the following two assumptions (already mentioned above, but not yet formalized):

(i) The *model assumption* is that the random process consists of a constant μ and an error term $\delta(\mathbf{s})$:

$$Z(\mathbf{s}) = \mu + \delta(\mathbf{s}) \quad \text{with} \quad \mathbf{s} \in D, \mu \in R \tag{1}$$

(ii) The *predictor assumption* is that the predictor for the point $\mathbf{s}_0$—denoted by $p(Z(\mathbf{s}_0))$—is a weighted linear function of all the observed output data:

$$p(Z(\mathbf{s}_0)) = \sum_{i=1}^{n} \lambda_i Z(\mathbf{s}_i) \quad \text{with} \quad \sum_{i=1}^{n} \lambda_i = 1 \tag{2}$$

To select the weights λ_i in (2), the *criterion* is minimal mean-squared prediction error (say) σ_e^2 defined as

$$\sigma_e^2 = E[\{Z(\mathbf{s}_0) - p(Z(\mathbf{s}_0))\}^2] \tag{3}$$

To minimize (3) given (2), the constraint $\sum_{i=1}^{n} \lambda_i = 1$ is added to the objective function through the Lagrangian multiplier *m*[1]. Then we can write the prediction error as

[1] Sentence in original paper revised

$$E[\{Z(\mathbf{s}_0)-\sum\nolimits_{i=1}^{n}\lambda_i Z(\mathbf{s}_i)\}^2]-2m[\sum\nolimits_{i=1}^{n}\lambda_i-1]. \tag{4}$$

To minimize (4), we utilize the *variogram*; also see Figure 1. By definition, the variogram is $2\gamma(\mathbf{h})=\mathrm{var}[Z(\mathbf{s}+\mathbf{h})-Z(\mathbf{s})]$, where $\mathbf{h}=\mathbf{s}_i-\mathbf{s}_j$ as explained by the stationary covariance process assumption with $\mathbf{h}\in R^d$ and $\mathrm{i}, j = 1, \ldots, n$. Obviously, we have $\mathrm{var}[Z(\mathbf{s}+\mathbf{h})-Z(\mathbf{s})]=2\gamma(\mathbf{s}_i-\mathbf{s}_j)=2\gamma(\mathbf{h})$. The spacing **h** is also called the lag.

After some tedious manipulations, (4) gives

$$-\sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{n}\lambda_i\lambda_j\,\gamma(\mathbf{s}_i-\mathbf{s}_j)+2\sum\nolimits_{i=1}^{n}\lambda_i\,\gamma(\mathbf{s}_0-\mathbf{s}_i)-2m\left(\sum\nolimits_{i=1}^{n}\lambda_i-1\right). \tag{5}$$

Differentiating (5) with respect to $\lambda_1, \ldots, \lambda_n$ and *m*, gives the optimal $\lambda_1, \ldots, \lambda_n$:

$$\boldsymbol{\lambda}'=\left(\boldsymbol{\gamma}+\mathbf{1}\frac{1-\mathbf{1}'\boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma}}{\mathbf{1}'\boldsymbol{\Gamma}^{-1}\mathbf{1}}\right)'\boldsymbol{\Gamma}^{-1} \quad \text{and} \quad m=-(1-\mathbf{1}'\boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma})/(\mathbf{1}'\boldsymbol{\Gamma}^{-1}\mathbf{1}), \tag{6}$$

where $\boldsymbol{\gamma}$ denotes the vector of (co)variances $(\gamma(\mathbf{s}_0-\mathbf{s}_1), \ldots, \gamma(\mathbf{s}_0-\mathbf{s}_n))'$, $\boldsymbol{\Gamma}$ denotes the $n\times n$ matrix whose $(i, j)^{\text{th}}$ element is $\gamma(\mathbf{s}_i-\mathbf{s}_j)$, $\mathbf{1}=(1, \ldots, 1)'$ is the vector of ones; also see Cressie (1993, p. 122).

We emphasize that these optimal Kriging weights λ_i depend on the specific point $\mathbf{s}_0$ that is to be predicted, whereas linear-regression metamodels use fixed estimated parameters (say) $\hat{\boldsymbol{\beta}}$. Note further that some of the weights λ_i may be negative[2].

The optimal weights (6) give the minimal mean-squared prediction error: (3) becomes (also see Cressie (1993, p. 122)

$$\begin{aligned}\sigma_e^2 &= \sum\nolimits_{i=1}^{n}\lambda_i\,\gamma(\mathbf{s}_0-\mathbf{s}_i)+m\\ &= \boldsymbol{\gamma}'\boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma}-\frac{(\mathbf{1}'\boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma}-1)^2}{\mathbf{1}'\boldsymbol{\Gamma}^{-1}\mathbf{1}}\quad .\end{aligned} \tag{7}$$

However, in (6) and (7) $\gamma(\mathbf{h})$ is *unknown*. The usual *estimator* is

[2] Sentence added to original paper

$$2\hat{\gamma}(\mathbf{h}) = \frac{1}{|N(\mathbf{h})|}\sum\nolimits_{N(\mathbf{h})}\left(Z(\mathbf{s}_i) - Z(\mathbf{s}_j)\right)^2 \tag{8}$$

where $|N(\mathbf{h})|$ denotes the number of distinct pairs in $N(\mathbf{h}) = \{(\mathbf{s}_i, \mathbf{s}_j) : \mathbf{s}_i - \mathbf{s}_j = \mathbf{h}\,; i, j = 1, \ldots, n\}$; see Matheron (1962). The estimator in (8) is unbiased, if the process $Z(\bullet)$ is indeed second-order stationary; see Cressie (1993, p. 71).

Given (8) for different $\|\mathbf{h}\|$ values, the variogram is estimated by *fitting* a curve through the estimated values $2\hat{\gamma}(\mathbf{h})$. This curve displays the following important *characteristics* (see Figure1):

(i) For large values of $\|\mathbf{h}\|$, the variogram $2\hat{\gamma}(\mathbf{h})$ approaches a constant $C(\mathbf{0})$, called the *sill*: For these large $\|\mathbf{h}\|$ values, all variances of the differences $Z(\mathbf{s}+\mathbf{h}) - Z(\mathbf{s})$ are invariant with respect to **h**.

To prove this property, we define the *covariogram* $C(\mathbf{h}) = Cov(Z(\mathbf{s}), Z(\mathbf{s}+\mathbf{h}))$. Obviously, $Cov(Z(\mathbf{s}), Z(\mathbf{s})) = Var(Z(\mathbf{s}))$. Then it is easy to derive

$$2\gamma(\mathbf{h}) = 2(C(\mathbf{0}) - C(\mathbf{h})). \tag{9}$$

Because $C(\mathbf{h}) \downarrow 0$ as $\|\mathbf{h}\| \uparrow \infty$, the variogram has the upper limit $2C(\mathbf{0})$.

(ii) The interval of $\|\mathbf{h}\|$ on which the curve does increase (to the sill), is called the *range* (say) *r*; that is, $C(\mathbf{h}) < \varepsilon$ for $\|\mathbf{h}\| > r + r_\varepsilon$. We shall give a specific model in (10).

(iii) Although (9) implies gamma $\gamma(\mathbf{0}) = 0$, the fitted curve does not always pass through zero: It may have a positive intercept—called the *nugget* variance. This variance estimates noise.

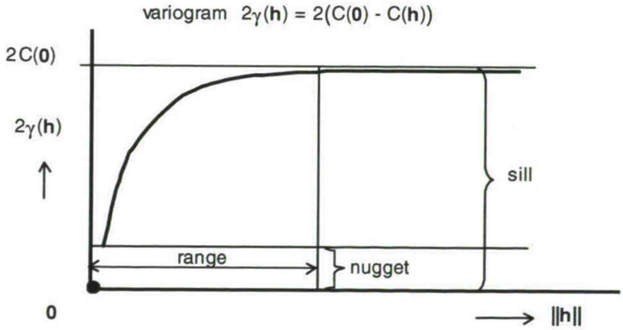


Figure 1: An example variogram

For example, in geostatistics this nugget effects means that when going back to the 'same' spot, a completely different output (namely, a gold nugget) is observed.

We add that in *random* simulation, the same input (say, the same traffic rate in queueing simulation) gives different outputs because different pseudo-random numbers are used. Below we shall return to this issue.

To fit a variogram curve through the estimates resulting from (8), analysts usually apply the *exponential model*

$$\gamma(\mathbf{h}) = \begin{cases} c_0 + c_1(1 - e^{-\|\mathbf{h}\|/a}) & \text{if} \quad \mathbf{h} \neq \mathbf{0} \\ 0 & \text{if} \quad \mathbf{h} = \mathbf{0} \end{cases} \tag{10}$$

where obviously c_0 is the nugget, $c_0 + c_1$ the sill, and *a* the range. However, other models are also fitted; for example, the *linear model*

$$\gamma(\mathbf{h}) = \begin{cases} c_0 + b\,\|\,\mathbf{h}\,\| & \text{if} \quad \mathbf{h} \neq \mathbf{0} \\ 0 & \text{if} \quad \mathbf{h} = \mathbf{0} \end{cases} \tag{11}$$

where again c_0 is the nugget; see Cressie (1993, p. 61). Actually, we shall apply (11) in our experiments, because it is the simplest model and yet gives acceptable results (for example, it estimates the nugget effect very well).

In *deterministic simulation*, analysts use more general distance formulas than (8). For example, Sacks et al. (1989, p. 413) and Jones et al. (1998, p. 5) use for the *k*-dimensional inputs $\mathbf{x}_i = (x_{i(1)}, \ldots, x_{i(k)})'$ and $\mathbf{x}_j = (x_{j(1)}, \ldots, x_{j(k)})'$ the weighted distance formula

$$h(\mathbf{x}_i, \mathbf{x}_j) = \sum\nolimits_{g=1}^{k} \vartheta_g \mid x_{i(g)} - x_{j(g)} \mid^{p_g} \tag{12}$$

where ϑ_g (with $\vartheta_g \geq 0$) measures the importance of the input x_g, and p_g controls the smoothness of the distance function. To estimate ϑ_g, maximum likelihood estimation (MLE) is used. The p_g are fixed such that $0 < p_g \leq 2$. (We shall briefly return to (12) in our section Conclusions and Future Research.)

2.3 Detrended Kriging

Ordinary Kriging was defined by (1), where $\mu \in R$ was the *constant* mean of the random process Z(•). This assumption, however, limits the application of Ordinary Kriging to rather simple models of the process Z(•). A more general assumption is that μ is not a constant, but an unknown linear combination of known functions $\{f_0(\mathbf{s}), \ldots, f_n(\mathbf{s})\}$, $\mathbf{s} \in D$. This is called *Universal Kriging*; see Huijbregts and Matheron (1971, p. 160) and also Cressie (1993, p. 151). Cressie (1993) discusses real (non-simulated) coal-ash data, and Regniere and Sharov (1999) discuss simulated spatial and temporal output data of a random simulation model for ecological processes.

Now we introduce a novel type of Kriging that we call *Detrended Kriging*. Detrended Kriging pre-processes the original data, and then applies Ordinary Kriging to the resulting data so we can apply software for Ordinary Kriging. For Universal Kriging, however, software is available only for spatial and temporal data, not for simulation with an arbitrary number of inputs—to the best of our knowledge[3].

We assume that the process mean $\mu(\mathbf{s})$ satisfies the decomposition

$$\mu(\mathbf{s}) = S(\mathbf{s}) + \eta(\mathbf{s}) \tag{13}$$

where $S(\mathbf{s})$ is a known signal function (see, however, the text below (14)) and $\eta(\mathbf{s})$ is a *white noise* process that models the measurement error; that is, $\eta(\mathbf{s})$ is normally identically and independently distributed with zero mean (NIID). So, we replace (1) by

$$Z(\mathbf{s}) = S(\mathbf{s}) + \eta(\mathbf{s}) + \delta(\mathbf{s})\,. \tag{14}$$

In practice, the signal function $S(\mathbf{s})$ in (14) is unknown. Therefore we estimate $S(\mathbf{s})$ through $\hat{S}(\mathbf{s})$, from the set of observed (noisy) I/O data $\{(\mathbf{s}_i, Z(\mathbf{s}_i)) : i = 1, \ldots, n\}$. Because of the assumed white noise, we use ordinary least squares (OLS) to obtain the estimator $\hat{S}(\mathbf{s})$.

Next we apply Ordinary Kriging to the detrended set $\{(\mathbf{s}_i, Z(\mathbf{s}_i) - \hat{S}(\mathbf{s}_i)) : i = 1, \ldots, n\}$. Our predictor for the output of location $\mathbf{s}_0$ is the sum of this Ordinary Kriging prediction and the

[3] After this paper was accepted, the Matlab Kriging toolbox of Lophaven, Nielsen, and Søndergaard became available.

estimator $\hat{S}(\mathbf{s}_0)$.

To test our new Detrended Kriging, we apply classic *cross-validation*; see Kleijnen and van Groenendaal (1992, p. 156). Cross-validation eliminates one I/O combination, say $(\mathbf{s}_k, Z(\mathbf{s}_k))$, from to the original data set $\{(\mathbf{s}_i, Z(\mathbf{s}_i)) : i = 1, \dots, n\}$, so the remaining $n-1$ data combinations are $\{(\mathbf{s}_i, Z(\mathbf{s}_i)) : i = 1, \dots, k-1, k+1, \dots, n\}$. This new set gives a prediction $p(Z(\mathbf{s}_k))$. This process of elimination and prediction is repeated for (say) c different combinations ($c \leq n$). Obviously, if we sort the original set such that the first c observations are deleted one at a time, then we get $k = 1, 2, \dots, c$.

To summarize the resulting prediction accuracy, we use the L_2 norm of the difference vector $\| p(Z(\mathbf{s}_k)) - Z(\mathbf{s}_k) \|$ (the L_2 norm $\| x \|$ is defined as $\left(\sum_{k=1}^{c} x_k^2\right)^{1/2}$). In our experiments we find that the L_1 and L_∞ norms give similar conclusions.

Note that in Kriging, all prediction errors may be zero at the I/O points that are actually used to estimate the Kriging model. Therefore we use cross-validation.

2.4 Two examples and five metamodels

We are interested in the application of Kriging to discrete-event simulation models, such as simulated queueing systems. As Law and Kelton (2000)—the best selling textbook on simulation—states (on page 12), a *single server* queueing system is quite representative of more complex, dynamic, stochastic simulation models. For further simplification, we suppose that the output of interest is the mean waiting time in the steady state, $\mathrm{E}(W)$. This output can be estimated through a simulation that uses the following non-linear stochastic difference equation:

$$W(i) = \max\{W(i-1) + S(i-1) - A(i), 0\} \quad \text{with} \quad i = 1, 2, \dots \tag{15}$$

where $W(i)$ denotes the waiting time of customer i, $S(i)$ the service time of customer i, and $A(i)$ the interarrival time between customers i and i -1. It is standard to start the simulation run in the empty (idle) state: $W(0) = 0$. For additional simplification, we assume that the arrival times form a Poisson process, and so do the service times. This gives the well-known *M/M/1* (which can actually be solved analytically; see equation 18 below). By definition, M/M/1 implies that both

$S(i)$ and $A(i)$ are identically and independently distributed (IID), so simulation is straightforward. The output E(W) is usually estimated through the simulation run's *average*

$$\overline{W} = \sum\nolimits_{i=b}^{n} \frac{W(i)}{n-b+1} \quad \text{with} \quad 0 \leq b < n \tag{16}$$

where b denotes the length of the initialization (start-up, transient) phase (which may be zero), and n the run length. (In M/M/1 analysis and simulation through renewal analysis, this initialization is no issue; in practical simulations, however, it is a major problem; see Law and Kelton (2000, pp. 496-552).) In other words, the dynamic simulation model generates the time series (15), but this series is characterized through the single statistic (16).

Actually, simulation is done for sensitivity analysis (possibly followed by optimization). Such an analysis aims at estimating the *input/output (I/O) function* (say)

$$E(Z(\mathbf{s})) = S(\mathbf{s}) \tag{17}$$

where—following (14)—Z denotes the (multiple) output and **s** the (multiple) input. In the M/M/1 example we have $Z = \overline{W}$ and $s = \lambda/\mu$ with arrival rate λ and service rate μ.

In general, $S(\mathbf{s})$ in (17) has unknown shape and parameters. However, when studying the performance of a specific simulation methodology, researchers often use the M/M/1 simulation model because the I/O function $S(\mathbf{s})$ is then known—assuming that the methodology has selected an appropriate initialization length b in (16) (obviously, knowledge of $S(\mathbf{s})$ may not be used by the methodology itself):

$$E(W) = \frac{\lambda}{\mu(\mu - \lambda)} \quad \text{with} \quad \frac{\lambda}{\mu} < 1. \tag{18}$$

Unfortunately, the latter assumption is very questionable: it is well known that selecting an appropriate transient-phase length b and run length n in (16) is difficult.

Moreover, Kriging assumes that—in general—the simulation observations Z have additive *white noise*; see (14). In the M/M/1 example, (14) gives (i) $Z = \overline{W}$, (ii) $S(\mathbf{s}) = \lambda/(\mu(1-\lambda))$, (iii) normality holds if $\overline{W}$ in (16) uses a sample size (n - b +1) such that an asymptotic central limit theorem holds, (iv) constant variances result if different simulated traffic rates use different and appropriate sample sizes—see Kleijnen and Van Groenendaal (1995)—and (v) independence results if no common pseudorandom numbers are used. Altogether, Kriging's (1) or (14) applies to the M/M/1 example only if a slew of assumptions hold!

Hence, it is much more efficient and effective to generate Kriging test data through sampling from (13) with $S(\mathbf{s}) = \lambda/(\mu(1-\lambda))$ instead of (15) and (16). Indeed, our approach requires less computer time, and guarantees that the white noise assumption holds, including the desired value for the variance of the white noise. The alternative using (15) and (16) would require very long runs, especially for high traffic rates $\lambda/\mu \uparrow 1$ this alternative requires $n \uparrow \infty$.

In conclusion, to test the Kriging methodology we generate data through a static, random Monte Carlo model like (13) instead of a dynamic stochastic simulation model such as (15) combined with (16). So, the Monte Carlo technique is both efficient and effective.

Besides Example I, we study Example II representing simulations with *multiple local maxima*, which are interesting when optimizing simulation outputs. Example I represents queueing simulations that show 'explosive' mean waiting times as the traffic rate approaches the value one. Example II has no specific interpretation.

We sample the white noise-term $\eta(\mathbf{s})$ in (14) through the Matlab function called 'randn', which gives standard NIID variates; that is, $\eta(\mathbf{s})$ has zero mean and unit variance. We also experiment with a larger variance namely 25; this results in larger error terms, but not in other conclusions.

To estimate possible values of the L_2 norm (defined above), we use 100 *macro-replications*. From these macro-replications we estimate L_2's median, 0.10 quantile $Q_{0.1}$, and 0.90 quantile $Q_{0.9}$.

In both examples we take $n = 21$ equally spaced input values: s_i with $i = 1, ..., 21$. For cross-validation we select (rather arbitrarily) $c = 5$ inputs values: We eliminate $i = 2, 8, 9, 15, 16$ respectively. We compared the following five metamodels.

i) Ordinary Kriging

ii) second-degree detrending: $\hat{S}(\mathbf{s})$ is a second-degree polynomial

iii) perfectly specified detrending function: $\hat{S}(\mathbf{s})$ is a hyperbola in Example I, and a fourth degree polynomial in Example II

iv) fifth-degree detrending: $\hat{S}(\mathbf{s})$ is a fifth-degree polynomial (overfitting)

v) linear regression model that is a second-degree polynomial estimated through OLS.

Note that we also study a perfectly specified detrending function (see iii), because this function provides a utopian situation: This function gives the minimum prediction error; in practice, this function is unknown (otherwise simulation would not be used). Furthermore, it helps us to verify the correctness of our computer program.

2.4.1 Example I: M/M/1 hyperbola

We take $S(s) = s/(1-s)$ on $D = [0.01, 0.99] \subset R^1$, this hyperbolic function may represent the mean steady-state waiting time for a traffic rate s in an M/M/1 queueing system. This function gives Figure 2, which also displays an example of the noisy output $Z(s)$. The input locations are $s_i \in \{0.01, 0.05, 0.10, \ldots, 0.95, 0.99\}$. The cross-validation is carried out at $s = 0.05, 0.35, 0.40, 0.70 \text{ and } 0.75$.

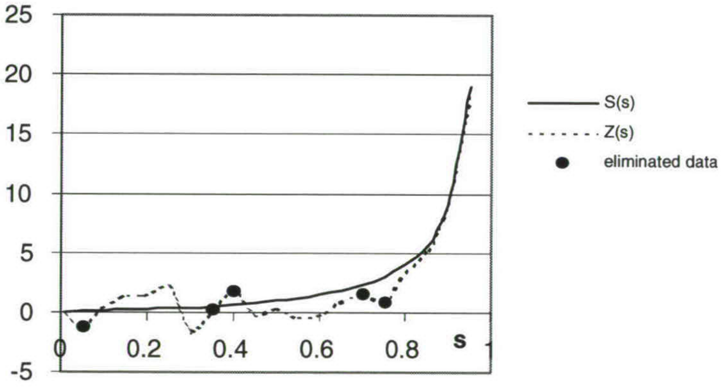


Figure 2: $S(s) = s/(1-s)$ and example sample output $Z(s)$

The estimated distribution of the prediction accuracy L_2 is summarized in Table 1. This example suggests that metamodel iii (perfectly specified detrending function) gives the best results. Model i (Ordinary Kriging) is not too bad. Model v (OLS) is simply bad.

Table 1: Prediction accuracy: estimated quantiles of L_2 distribution for Example I (M/M/1)

	metamodel				
L_2	i	ii	iii	iv	v
$Q_{0.1}$	1.2429	1.3622	1.1972	2.7411	17.593
median	1.8832	2.1522	1.8419	3.6678	18.17
$Q_{0.9}$	2.5698	2.925	2.5829	4.4677	18.652

2.4.2 Example II: fourth-degree polynomial

We take the following specific polynomial: $S(s) = -0.0579\,s^4 + 1.11\,s^3 - 6.845\,s^2 + 14.1071\,s + 2$ on $D = [0, 10] \subset R^1$. This polynomial has two maxima: A local one and a global one; see Figure 3. We obtain output for the following 21 input locations $s_i \in \{0, 0.5, 1, \ldots, 10\}$. We cross-validate at $s = 0.5, 3.5, 4, 7$ and 7.5.

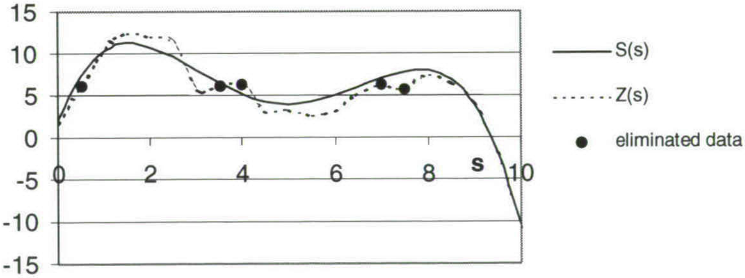


Figure 3: $S(s) = -0.0579\,s^4 + 1.11\,s^3 - 6.845\,s^2 + 14.1071\,s + 2$ and example sample output *Z*(*s*)

The estimated distribution of L_2 is summarized in Table 2. This example suggests that metamodel iii (perfectly specified detrending function) gives the best results. Model i (Ordinary Kriging) is not too bad. Model v (OLS) is simply bad.

Table 2: Example II ; also see Table 1

	metamodel				
L_2	i	ii	iii	iv	v
$Q_{0.1}$	1.5976	1.515	1.2094	1.2173	5.5965
median	2.4713	2.41	1.8748	1.9117	6.0363
$Q_{0.9}$	3.3226	3.246	2.6424	2.6959	6.5048

2.5 Nugget effect

We also wish to better understand the relationship between the nugget effect in (11) and the variance of the noise $\eta(\mathbf{s})$ in (13). Therefore we perform a simple Monte Carlo experiment: We take $Z(s) = 10 + \eta(s)$ where $\eta(s)$ is NIID with $\mu = 0$ and σ^2 = 1, 4, 9, 16, and 25 respectively. We sample two macro-replicates, setting the seed of Matlab's 'randn'—rather arbitrary—to the values 10 and 20. In the various Kriging metamodels, we fit the linear variogram of (11); see Figure 4 (we display results for the seed value of 10 only; note the different scales for the y-axis in the four plots).

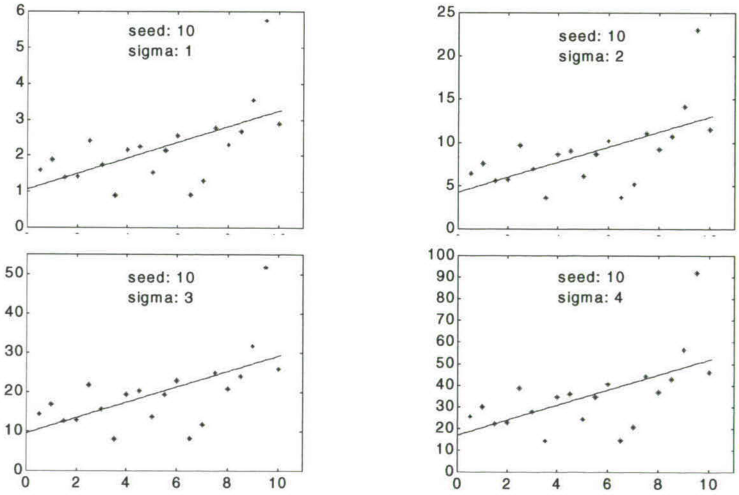


Figure 4: Variogram estimates for different variances

The intercept in (11) estimates the nugget effect; this intercept is presented for different σ^2 values in Table 3. Obviously, these results confirm our conjecture: The nugget effect is the variance of the noise.

Table 3: Estimated nugget effects for different white noise variances σ^2

σ^2	seed 10	seed 20
1	1.1	0.9
4	4	4
9	9.6	8.5
16	17.1	15.5
25	26.5	24.1

2.6 Conclusions and future research

We assume that in practice the mean μ of the Kriging metamodel (1) is not a constant, but is a composition of a signal function and white noise. We presented results for two examples of input/output behavior of the underlying random simulation model: Ordinary Kriging and Detrended Kriging gives quite acceptable predictions, whereas traditional linear regression gives the worst results.

OLS predicts so poorly, because OLS assumes that the fitting errors are white noise, whereas Kriging allows errors that are correlated; more specifically, the closer the inputs are, the more positive correlation. Moreover, OLS uses a single estimated function for all input values, whereas Kriging adapts its predictor as the input changes. Note that OLS 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)?

OLS is also compared with (universal) Kriging by Regniere and Sharov (1999). Their OLS model, however, is a rather complicated metamodel (involving terms of order six). The resulting prediction accuracies are similar for OLS and Kriging.

Further, we found that the nugget effect equals the noise variance.

We restricted our examples to a single input. Therefore we gave each weight ϑ_g in the more general distance formula (12), the fixed value of one. In design optimization, however, these parameters are used to control the importance of the input variable x_g ; see for example Simpson et al. (2001, p. 8) and Jones et al. (1998, p. 5). In future work we shall investigate multiple inputs.

Further, we shall relax the assumption of white noise: We shall investigate the effects of non-constant variances (which occur in queueing simulations), common random number usage (which creates correlations among the simulation outputs), and non-normality (Kriging uses maximum likelihood estimators of the weights ϑ_g , which assumes normality). Finally, we shall apply Kriging to practical queueing and inventory simulations.

Acknowledgment

We thank the two anonymous referees for their most useful comments on an earlier version.

References

Campolongo, F., J.P.C. Kleijnen, and T. Andres (2000), Screening methods. In: *Sensitivity Analysis*, edited by A. Saltelli, K. Chan, and E.M. Scott, Wiley, Chichester (England), pp. 65-89

Cressie, N.A.C. (1993). *Statistics for Spatial Data*, Wiley, New York

Huijbregts, C.J. and Matheron, G. (1971). Universal Kriging (An optimal method for estimating and contouring in trend surface analysis). In *Proceedings of Ninth International Symposium on Techniques for Decision-making in the Mineral Industry*

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

Kleijnen, J.P.C. and W. van Groenendaal (1992). *Simulation, a Statistical Perspective*, Wiley, Chichester (Engeland)

Kleijnen, J.P.C. and W. van Groenendaal (1995), Two-stage versus sequential sample-size determination in regression analysis of simulation experiments. *American Journal of Mathematical and Management Sciences*, 15, nos. 1&2, 1995, pp. 83-114

Kleijnen, J.P.C. (1998), Experimental design for sensitivity analysis, optimization, and validation of simulation models. In: *Handbook of Simulation*, edited by J. Banks, Wiley, New York, pp. 173-223

Kleijnen, J.P.C. and J.Helton (1999), Statistical analyses of scatter plots to identify important factors in large-scale simulations. *Reliability Engineering and Systems Safety,* 65, no. 2, pp. 147-197

Koehler, J.R. and A.B. Owen (1996), Computer experiments. In: *Handbook of Statistics*, Volume 13, edited by S. Ghosh and C.R.Rao, Elsevier, Amsterdam, pp. 261_308

Law, A.M. and W.D. Kelton (2000), *Simulation modeling and analysis, third edition*, McGraw-Hill, Boston

Matheron, G. (1962) *Traite de Geostatistique Appliquee*, Memoires du Bureau de Recherches Geologiques et Minieres, No. 14, Editions Technip, Paris , (pp.57-59)

Meckesheimer, M., R.R. Barton, T.W. Simpson, and A.J. Booker (2001), Computationally inexpensive metamodel assessment strategies. *American Institute of Aeronautics and Astronautics Journal* (submitted)

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, 1999, pp. 146-152

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

Simpson, T.W. , Mauery, T.M. , Korte, J.J., and Mistree, F. (2001). *Kriging Metamodels for Global Approximation in Simulation-Based Multidisciplinary Design Optimization. American Institute of Aeronautics and Astronautics Journal* (submitted)

Van Groenendaal, W.J.H. and J.P.C. Kleijnen (1997), On the assessment of economic risk: factorial design versus Monte Carlo methods. *Reliability Engineering and Systems Safety,* 57, pp. 91-102

Chapter 3

Robustness of Kriging when Interpolating in Random Simulation with Heterogeneous Variances: Some Experiments

Abstract

This paper investigates the use of Kriging in random simulation when the simulation output variances are not constant. Kriging gives a response surface or metamodel that can be used for interpolation. Because Ordinary Kriging assumes constant variances, this paper also applies Detrended Kriging to estimate a non-constant signal function, and then standardizes the residual noise through the heterogeneous variances estimated from replicated simulation runs. Numerical examples, however, suggest that Ordinary Kriging is a robust interpolation method.

3.1 Introduction

Given a set of input/output (I/O) data, Kriging fits a metamodel (also called response surface, emulator, auxiliary model, etc.). Kriging is an *interpolation* method; that is, at the observed I/O values its predictions are exactly equal to the observed output values. However, classic least

Paper by Kleijnen, J.P.C. and W.C.M. Van Beers, *European Journal of Operational Research* Volume 165, Issue 3 , 16 September 2005, pp. 826-834 squares (LS)—used in regression—typically fits a low-degree polynomial, which results in non-zero fitting errors (or residuals) such that the sum of these squared errors is minimized.

Originally, the South African mining engineer D.G. Krige modeled real world I/O data to predict gold quantities; see Cressie (1993). Later on, Kriging was applied to deterministic simulation I/O data (modeling the performance of computer chips, cars, etc.); see Sacks et al. (1989). More recently, the application of Kriging to random simulation (for queuing problems, etc.) was proposed by Barton (1994), and was investigated in detail by Van Beers and Kleijnen (2003). The latter authors suggested that it may be better to replace classic Ordinary Kriging by their novel method called Detrended Kriging, which corrects or 'detrends' the original output data through an estimated signal function (see $\hat{S}(\mathbf{s})$ below). However, both Ordinary and Detrended Kriging assume that the outputs have *constant variances*.

In practice, this constant variance assumption is often completely unrealistic. For example, the so-called M/M/1 queue is an important building block in discrete-event simulation; see Law and Kelton (2000). For the M/M/1, Cohen (1969) proves analytically that the steady-state waiting time (say) Z for a traffic load s with $0 < s < 1$ has mean

$$E\left(Z(s)\right) = \frac{s}{1-s} \tag{1}$$

and variance

$$Var\left(Z(s)\right) = \frac{s(2-s)}{(1-s)^2} \tag{2}$$

so this variance is definitely not constant when the traffic load changes. Therefore we now investigate the consequences of *variance heterogeneity* when applying Kriging to the I/O of random simulation.

Whereas linear-regression analysis of low-order polynomial metamodels has been applied extensively in discrete-event simulation (such as queueing simulation), *Kriging has hardly been applied to random simulation.*

The main conclusion of our (limited) experiments will be that Ordinary Kriging seems a *robust* interpolation method; that is, it predicts better than its competitors do.

The remainder of this paper is organized as follows. Section 3.2 summarizes the basics of Kriging. Section 3.3 discusses solutions for the problems of non-constant and exploding variances in random simulation. Section 3.4 proposes a novel Kriging method, which estimates a non-constant signal function and the non-constant variances. Section 3.5 presents numerical examples, which suggest that Ordinary Kriging is a robust prediction method whereas the novel Kriging method is better only in utopian situations. Section 3.6 gives conclusions and proposes future research topics.

3.2 Kriging basics

Kriging assumes the following model:

$$Z(\mathbf{s}) = S(\mathbf{s}) + \eta(\mathbf{s}) \quad \text{with} \quad \eta(\mathbf{s}) \sim NID(\mathbf{0}, \sigma^2(\mathbf{s})) \tag{3}$$

where $S(\mathbf{s})$ is the so-called *signal function* and $\eta(\mathbf{s})$ the additive *noise*; we use the notation $x \sim NID(a, b)$ when x is normally and independently distributed (NID) with mean a and variance b. Below, we shall discuss the classic assumption of *white noise*, which means that $\sigma^2(\mathbf{s})$ in (3) reduces to a constant (say) σ^2.

Note: The model in (3) may generate negative output values, whereas M/M/1 generates non-negative values only. Nevertheless, we use (3) because it provides better control over our experiment and it is faster than M/M/1 simulation; see Van Beers and Kleijnen (2003).

Ordinary Kriging assumes a *stationary covariance process* for $Z(\mathbf{s})$ in (3) (see Cressie 1993). That assumption implies that the expected values $E(Z(\mathbf{s}))$ are constant, and the covariances of $Z(\mathbf{s}_i + \mathbf{h})$ and $Z(\mathbf{s}_i)$ depend only on the distance (or lag) **h**. The *predictor* for the point $\mathbf{s}_0$—denoted by $p(Z(\mathbf{s}_0))$—is a weighted linear combination of all the (say) *n* observed output data:

$$p(Z(\mathbf{s}_0)) = \sum\nolimits_{i=1}^{n} \lambda_i Z(\mathbf{s}_i) \quad \text{with} \quad \sum\nolimits_{i=1}^{n} \lambda_i = 1. \tag{4}$$

To select the weights λ_i in (4), Kriging minimizes the mean-squared prediction error. The technique uses the *variogram*, defined as $2\gamma(\mathbf{h}) = \mathrm{var}(Z(\mathbf{s}+\mathbf{h}) - Z(\mathbf{s}))$, see Figure 1 discussed below. The optimal weights turn out to be

$$\boldsymbol{\lambda}' = \left(\boldsymbol{\gamma} + \mathbf{1}\frac{1-\mathbf{1}'\boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma}}{\mathbf{1}'\boldsymbol{\Gamma}^{-1}\mathbf{1}}\right)' \boldsymbol{\Gamma}^{-1} \tag{5}$$

where $\boldsymbol{\gamma}$ denotes the vector $(\gamma(\mathbf{s}_0 - \mathbf{s}_1), \ldots, \gamma(\mathbf{s}_0 - \mathbf{s}_n))'$, $\boldsymbol{\Gamma}$ denotes the $n \times n$ matrix whose $(i,\ j)^{\text{th}}$ element is $\gamma(\mathbf{s}_i - \mathbf{s}_j)$, and $\mathbf{1}$ denotes a vector of ones; also see Cressie (1993, p. 122). Note that the weights λ_i in (5) vary with the prediction point, whereas polynomial-regression metamodels use the same estimated metamodel for all these points.

3.3 Random simulation with heterogeneous variances

In this paper, we allow heterogeneous variances; hence, the Kriging assumption of a stationary covariance process does not hold. Therefore, we do not expect a smooth variogram. Furthermore, Van Beers and Kleijnen (2003) found that the intercept of the fitted linear variogram estimates the so-called *nugget effect*. The term 'nugget' originated in the Kriging of gold mining data: when going back to the 'same' spot (lag $\mathbf{h} \downarrow \mathbf{0}$; see the text above equation 4), a completely different output—namely, a gold nugget—may be observed. In general, Kriging interprets a nugget effect as an estimate of noise; for example, measurement noise in geostatistics. Random simulation models produce noisy data—by definition. In this paper we shall investigate whether the intercept of the variogram still estimates a nugget effect in case of heterogeneous variances.

Moreover, in simulation the variance may be so large that simulation is actually impractical: as the traffic load approaches the value one, the variance *explodes*—as the M/M/1 illustrates; see (2). Therefore Cheng and Kleijnen (1999) recommend that

(i) in such a situation we should simulate a load much lower than one, and extrapolate;

(ii) the higher the simulated load is, the larger the number of replicated simulation runs should be.

Likewise, Asmussen (1992) recommends not to simulate the M/M/1 for a traffic rate $s > 0.9$; he too suggests to extrapolate in those cases. Van Beers and Kleijnen (2003), however, simulated loads up to 0.99—but they *artificially* kept the variances constant and small over their whole range of simulated values, $0.01 \le s \le 0.99$.

Initially, we sample a single output value for each of 21 input (traffic rate) values, as follows. We use (3) where we insert (1) and (2), which model the signal and the variance functions of the M/M/1. We do so for 21 traffic rates $s \in \{0.01, 0.05, 0.10, \dots, 0.95, 0.99\}$. We ignore the variance heterogeneity and the variance explosion, and proceed as in Van Beers and Kleijnen (2003). We then get estimated variograms like Figure 1. This figure shows a wildly oscillating variogram; obviously, the linear approximation is poor. This poor fit gives an inaccurate Kriging metamodel with the weights λ_i defined in (5).

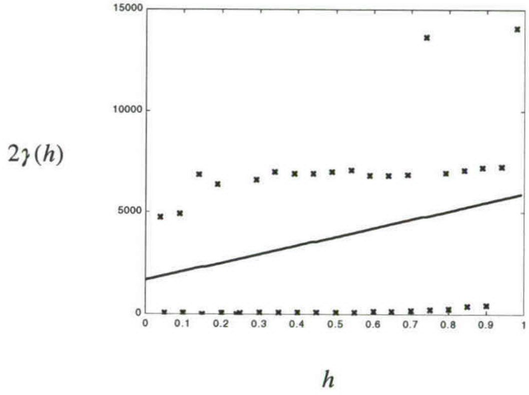


Figure 1: Estimated variogram for $Z(s) = s/(1-s) + \eta(s)$ with $\eta(s) \sim NID\left(0,\ s(2-s)/(1-s)^2\right)$

Figure 1 illustrates that in random simulation we should ensure that the *signal/noise ratio* is 'adequate'. To realize such a signal/noise ratio, we should—first of all—avoid variance explosion; that is, we should simulate a *non-saturated traffic load*; that is, a load much lower than the value one. We therefore eliminate the observations for $s = 0.99$; this yields Figure 2.

Comparison with the values on the y-axis of Figure 1 shows that the range of the variogram is much smaller, so variances have indeed been decreased.

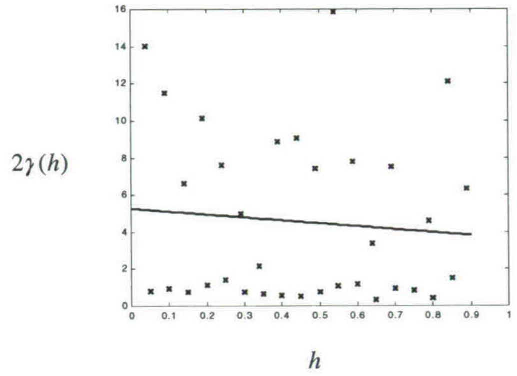


Figure 2: Variogram after elimination of $s = 0.99$ in Figure 1

Besides avoiding variance explosion, we may reduce the magnitudes of the remaining variances through increasing the *number of replicates* (say) *m*—whenever the noise of a single replicate is too big. We denote the number of identically and independently distributed (IID) replicates for input value s by $m(s)$. Then one option is to select these $m(s)$ such that the variances of the average responses become a constant (say) $\overline{\sigma}^2$:

$$Var(\overline{Z}(s)) = \frac{Var(Z(s))}{m(s)} = \overline{\sigma}^2; \tag{6}$$

see Kleijnen and Van Groenendaal (1995).

Note: In steady-state simulation—as opposed to terminating simulation—we may make a single long run and partition that run into subruns that play the role of replicates; see Law and Kelton (2000).

In practice, however, the experimental design implied by (6) may be impractical (see Kleijnen and Van Groenendaal 1995). Therefore we allow unequal variances of the sample averages $\overline{Z}(s)$. These averages are 'the' observations at the inputs *s* to which we fit a Kriging metamodel, as follows.

3.4 Novel Kriging method: Studentization

To solve the variance heterogeneity problem, we first apply Van Beers and Kleijnen (2003) 's *Detrended Kriging*, which assumes

$$Z(\mathbf{s}) = S(\mathbf{s}) + \eta(\mathbf{s}) + \delta(\mathbf{s}) \tag{7}$$

where $S(\mathbf{s})$ is the signal function, $\eta(\mathbf{s}) \sim NID(\mathbf{0}, \boldsymbol{\sigma}^2)$ is the *white noise* model for measurement error, and $\delta(\mathbf{s})$ denotes fitting error. However, unlike Van Beers and Kleijnen (2003) we now allow $\eta(\mathbf{s})$ to have heterogeneous variances determined by the input **s**; see (3).

Unbiased estimators of these variances—not depending on a metamodel such as (7)—are the classic estimators

$$\hat{\sigma}^2(\mathbf{s}_i) = \frac{\sum_{r=1}^{m(\mathbf{s}_i)} \left(Z_r(\mathbf{s}_i) - \overline{Z}(\mathbf{s}_i)\right)^2}{m(\mathbf{s}_i) - 1} \tag{8}$$

where $Z_r(\mathbf{s}_i)$ denotes the r^{th} replicated observation on $Z(\mathbf{s}_i)$ and

$$\overline{Z}(\mathbf{s}_i) = \frac{\sum_{r=1}^{m(\mathbf{s}_i)} Z_r(\mathbf{s}_i)}{m(\mathbf{s}_i)} \tag{9}$$

denotes its sample average.

Like Van Beers and Kleijnen (2003), we may estimate the signal function $S(\mathbf{s})$ through different models $\hat{S}(\mathbf{s})$ (see below). Unlike Van Beers and Kleijnen (2003), we now permit unequal variances, so the best linear unbiased estimator (BLUE) is given by *weighted least squares* (WLS)—not ordinary least squares (OLS).

Next, we apply Ordinary Kriging to the detrended and standardized observations

$$\widetilde{Z}(\mathbf{s}) = \frac{\overline{Z}(\mathbf{s}) - \hat{S}(\mathbf{s})}{\hat{\sigma}(\mathbf{s})/\sqrt{m}}. \tag{10}$$

Because (10) resembles Student's *t* statistic, we call the transformation in (10) 'Studentizing'.

If we neglect the random character of $\hat{S}(\mathbf{s})$, then we conjecture that the transformed data $\widetilde{Z}(\mathbf{s})$ in (10) have a constant mean (namely zero) and a constant variance, namely $m/(m-2)$. *So* $\widetilde{Z}(\mathbf{s})$ *then satisfies the Ordinary Kriging assumptions.*

Finally, the predictor for $\mathbf{s}_0$ (say) $p(Z(\mathbf{s}_0))$ follows from (10):

$$p(Z(\mathbf{s}_0)) = p(\widetilde{Z}(\mathbf{s}_0)) \times \frac{\hat{\sigma}(\mathbf{s}_0)}{\sqrt{m}} + \hat{S}(\mathbf{s}_0), \tag{11}$$

where $p(\widetilde{Z}(\mathbf{s}_0))$ is the Ordinary Kriging predictor for $\mathbf{s}_0$ based on the transformed data in (10); $\hat{\sigma}(\mathbf{s}_0)$ is given by (8) if $\mathbf{s}_0 = \mathbf{s}_i$; otherwise, we use piecewise linear interpolation between the variances at the two neighboring points that have already been observed. (This interpolation avoids negative estimated variances.)

3.5 Numerical examples

As the *signal function* in (3) we first select $S(s) = s/(1-s)$, which equals the M/M/1 hyperbola in (1). Following Van Beers and Kleijnen (2003), we experiment with five detrending models, $\hat{S}(s)$:

(i) Ordinary Kriging: $\hat{S}(s)$ is a constant

(ii) second-degree detrending: $\hat{S}(s)$ is a second-degree polynomial

(iii) perfect detrending: $\hat{S}(s) = S(s) = s/(1-s)$

(iv) fifth-degree detrending: $\hat{S}(s)$ is a fifth-degree polynomial

(v) second-degree polynomial regression model, estimated through WLS.

Note that '(i) Ordinary Kriging' does use the *m* replications, but not the Studentization in (10). Note further that we also study '(iii) perfect detrending', even though this is a utopian situation: in practice, this detrending function is unknown; otherwise simulation would not be used. This function, however, gives the minimum prediction error; furthermore, it helps us to verify the correctness of our computer program.

We examine four *variance heterogeneity cases*; we select the coefficients such that $\sigma(s)$ in the cases I, II, and III have a mean value of one for *s* in the interval $0 < s < 1$:

I $\sigma(s) = 1$ (constant standard deviation)

II $\sigma(s) = 1/2 + s$ (linearly increasing standard deviation)

III $\sigma(s) = 1/4 + 1/2s + 3/2s^2$ (parabola)

IV $\sigma(s) = (s(2-s))^{1/2}/(1-s)$ (hyperbola; see M/M/1).

To generate the noise in (3), we first generate standard normal variables (say) x through the Matlab function 'randn'; then $\eta(s) = x \times \sigma(s)$ where $\sigma(s)$ is quantified by the cases I through IV.

To estimate this noise, we experiment with various *replication numbers* $m(s)$. We assume constant replication numbers: $m(s) = m$. We vary this *m* between its minimum—namely, $m = 2$—and its maximum—namely $m = \infty$ (known variances). We also examine $m = 25$ and $m = 100$. When we consider known variances, we compute $\overline{Z}(s_i)$ from only 100 replicates.

We take 21 'old' observation points $s \in \{0.01, 0.05, 0.10, \ldots, 0.95, 0.99\}$, as mentioned in section 3.1.

We *evaluate* the Kriging models (i) through (v) at ten new input values (or 'test set'), namely at $s_k \in \{0.095, 0.185, 0.275, \ldots, 0.905\}$. We quantify the prediction accuracy of these five Kriging models through the L_2 norm

$$\| p(Z(s_k)) - E(Z(s_k)) \| = \left(\sum_k \left(p(Z(s_k)) - E(Z(s_k)) \right)^2 \right)^{1/2}.$$

To estimate the statistical distribution of this L_2 norm, we use 100 macro-replicates, and characterize the resulting distribution by its median, its 0.10 quantile, and its 0.90 quantile.

Note: Many criteria have been used to compare performance; see Kleijnen and Sargent (2000). Here we select the L_2 norm, which is classic in mathematics. Anyhow, we select a criterion that does not seem to bias our results in favor of any particular metamodel. Moreover, only relative values of the norm for the various metamodels are important.

In Table 1a we summarize the results of our hyperbola (M/M/1) experiments for case II ($\sigma(s) = 1/2 + s$); we obtain similar results for the other cases. This table shows that:

- In Kriging the accuracy increases as the number of replicates m increases; for example, Ordinary Kriging (model i) has a median L_2 value of 1.7772 when $m = 2$, which decreases to 0.5519 when $m = 100$.
- Polynomial regression (model v), however, does not improve with m: its median L_2 remains 29.1.
- Utopian detrending (model iii) does not always give minimum inaccuracy. For example, when $m = 2$ Ordinary Kriging (model i) has a median L_2 of 1.7772, whereas utopian detrending has 3.5355. However, when $m = 100$ Ordinary Kriging (model i) has a median L_2 of 0.5519, whereas utopian detrending has 0.2490.
- The practical detrending models (ii) and (iv) do not give significantly lower inaccuracies than Ordinary Kriging (model i).

Table 1a: Hyperbola experiments: Quantiles of distribution of prediction accuracy L_2, for variance heterogeneity case II, $\sigma(s) = 1/2 + s$, for different number of simulation replicates m, detrending models $\hat{S}(s)$ (i) through (v), estimated from 100 macro-replicates

$m = 2$					
L_2	(i)	(ii)	(iii)	(iv)	(v)
$Q_{0.10}$	1.2978	5.1792	1.7786	3.5183	28.6030
Median	1.7772	25.0581	3.5355	12.5414	29.1393
$Q_{0.90}$	2.4125	96.7104	11.5393	106.1171	29.7689

$m = 25$					
L_2	(i)	(ii)	(iii)	(iv)	(v)
$Q_{0.10}$	0.4517	0.4867	0.3664	0.7160	28.8867
Median	0.6681	0.7064	0.5169	0.9951	29.0978
$Q_{0.90}$	0.9769	1.0500	0.7071	1.3466	29.2946

$m = 100$					
L_2	(i)	(ii)	(iii)	(iv)	(v)
$Q_{0.10}$	0.4109	0.4004	0.1768	0.6859	29.0104
Median	0.5519	0.5502	0.2490	0.8522	29.1054
$Q_{0.90}$	0.6632	0.6571	0.3415	1.0093	29.1949

$m = 100$, theoretical variances					
L_2	(i)	(ii)	(iii)	(iv)	(v)
$Q_{0.10}$	0.4109	0.4041	0.1743	0.7173	29.0104
Median	0.5519	0.5429	0.2477	0.8147	29.1054
$Q_{0.90}$	0.6632	0.6516	0.3399	0.9577	29.1949

Like Van Beers and Kleijnen (2003), we add a second set of experiments: we replace the hyperbola by a fourth-degree polynomial that has two local tops (hills): $S(s) = -0.0579s^4 + 1.11s^3 - 6.845s^2 + 14.107s + 2$ with $0 \le s \le 10$.

We again take 21 observation points, now spread over the experimental domain as follows: $s \in \{0,\ 0.5,\ 1,\ \ldots,\ 9.5,\ 10\}$. We again estimate the Kriging models (i) through (v). The first two models represent different degrees of underfitting, whereas model (iv) represents overfitting. We evaluate these five alternative models at the test set $s_k \in \{0.95,\ 1.85,\ 2.75,\ \ldots,\ 9.05\}$. We present our results for a heterogeneity case similar to II—namely $\sigma(s) = 0.5 + 0.1s$—in Table 1b.

Table 1b: Fourth-degree polynomial experiments: see Table 1a for remaining symbols

			$m = 2$		
L_2	(i)	(ii)	(iii)	(iv)	(v)
$Q_{0.10}$	1.2583	3.3673	1.3326	1.6863	9.1334
Median	1.8657	8.4217	1.8517	2.8940	9.6088
$Q_{0.90}$	2.4607	60.1548	2.4381	9.9289	10.0154

			$m = 25$		
L_2	(i)	(ii)	(iii)	(iv)	(v)
$Q_{0.10}$	0.3997	0.4032	0.3334	0.3335	9.3816
Median	0.5297	0.5409	0.4867	0.4840	9.5065
$Q_{0.90}$	0.7179	0.7381	0.6897	0.6979	9.6165

			$m = 100$		
L_2	(i)	(ii)	(iii)	(iv)	(v)
$Q_{0.10}$	0.2818	0.2651	0.1798	0.1795	9.4574
Median	0.3754	0.3579	0.2543	0.2557	9.5094
$Q_{0.90}$	0.4584	0.4563	0.3287	0.3284	9.5734

			$m = 100$, theoretical variances		
L_2	(i)	(ii)	(iii)	(iv)	(v)
$Q_{0.10}$	0.2818	0.2684	0.1732	0.1813	9.4574
Median	0.3754	0.3560	0.2470	0.2555	9.5094
$Q_{0.90}$	0.4584	0.4449	0.3401	0.3283	9.5734

From Table 1, we conclude that Ordinary Kriging is a robust method; polynomial regression analysis gives the worst prediction results.

Finally, we examine the *nugget* effect. It is well-known that in case of a known and constant variance σ^2, the *intercept* of the variogram estimates this σ^2; also see Section 3.3. Now we conjecture that the Studentized observations $\tilde{Z}(s)$ in (10) behave like white noise, so that the intercept estimates the variance of Student's t_{m-1}, namely $\sigma^2 = m/(m-2)$ where m denotes the number of simulation replicates per input value. Therefore we formulate the following null-hypothesis:

$$H_0: \quad E(\hat{\beta}_0) = \frac{m}{m-2} \tag{12}$$

where $\hat{\beta}_0$ denotes the estimated intercept of the linear variogram. Note that in (10) we now use $\hat{S}(s) = \overline{Z}(s)$. To test the hypothesis in (12), we experiment with various m values, namely 25, 100 and 500. Further, the random numbers for the case of 500 replicates include those for 100 and 25 replicates, so the results are not statistically independent. Because the same number of replicates implies that the studentized data have the same mean and the same standard deviation in all variance heterogeneity cases, we examine only one case—namely case III ($\sigma(s) = 1/4 + 1/2\,s + 3/2\,s^2$). We use 500 macro-replicates to estimate the mean and standard deviation of $\hat{\beta}_0$; see Table 2. Because we obtain these estimates for more than one value of m, we apply Bonferroni's inequality: we select the estimate that deviates most from the hypothesized value $m/(m-2)$, and reject H_0 in (12) if its t value is significant at $\alpha/2k$ where k denotes the number of values of m (here $k = 3$) and the factor 2 is used because we have a two-sided test. H_0 in (12) is rejected at the significance level $\alpha = 0.20: t_{500-1} = -2.605$ while the critical value is -1.83. Therefore we repeat the experiment with other random numbers, now we find a non-significant t value. So we conclude that $m/(m-2)$ in (12) provides an adequate approximation.

Table 2: Mean and standard deviation of the variogram intercepts $\hat{\beta}_0$, for $m = 25$, 100 and 500 simulation replicates (estimated from $n = 500$ macro-replicates)

$m = 25$		$m = 100$		$m = 500$	
mean	st.dev.	mean	st.dev.	mean	st.dev.
1.079	0.44512	0.97367	0.40125	0.99095	0.3835

We also examine the nugget effect in case of Ordinary Kriging with constant and non-constant signal function respectively, namely $S(s) = 10$ and $S(s) = s/(1-s)$. Furthermore, we experiment with several variance functions. From Table 3 we conclude that in case of a constant signal function the intercept of the linear variogram estimates the square of the average standard deviation, $(\overline{\sigma})^2$ (also see Van Beers and Kleijnen 2003). However, in case of non-constant signal function we do not know what the intercept estimates!

Table 3: Estimated nugget effect (variogram intercept)—estimated from 200 macro-replicates—with Ordinary Kriging for various variance cases $\sigma^2(s)$, constant and non-constant signal $S(s)$

Constant signal $S(s) = 10$ and Homogeneous variances			
noise variance	$\sigma(s) = 0.5$	$\sigma(s) = 1$	$\sigma(s) = 4$
variogram intercept	0.2307	0.9227	14.7629

Constant signal $S(s) = 10$ and Heterogeneous variances			
noise variance	$\sigma(s) = 0.5 + s$	$\sigma(s) = 0.5 + 10s$	$\sigma(s) = \tfrac{1}{4} + \tfrac{1}{2}s + 1\tfrac{1}{2}s^2$
variogram intercept	0.9385	32.4869	1.0844

Non-constant signal $S(s) = s/(1-s)$ and Homogeneous variances			
noise variance	$\sigma(s) = 0.5$	$\sigma(s) = 1$	$\sigma(s) = 4$
variogram intercept	782.3001	783.6247	800.9322

Non-constant signal $S(s) = s/(1-s)$ and Heterogeneous variances			
noise variance	$\sigma(s) = 0.5 + s$	$\sigma(s) = 0.5 + 10s$	$\sigma(s) = \tfrac{1}{4} + \tfrac{1}{2}s + 1\tfrac{1}{2}s^2$
variogram intercept	784.9352	822.6294	787.7666

3.6 Conclusions and further research

We examined whether Kriging is a practical interpolation method for random simulation models with non-constant output variances. Based on our (limited) numerical results, we draw the following conclusions.

1. Ordinary Kriging is an interpolation method that seems not very sensitive to variance heterogeneity.
2. In practice, the signal function $S(s)$ is unknown so Detrended Kriging must use an approximation (for example, a polynomial). However, this performs worse than Ordinary Kriging.
3. A second-degree polynomial regression model estimated through least squares—ordinary or weighted—is the worst of the metamodels considered for a hyperbolic I/O function, which applies in M/M/1. (This regression analysis may be useful if only a few observations are available—say, two or three instead of twenty—and a rough sensitivity analysis suffices. The role of classic regression metamodels in simulation is also discussed by Van Beers and Kleijnen 2003.)
4. Increasing the number of replications improves the accuracy of any Kriging prediction (as is to be expected).
5. For both homogeneous and heterogeneous variances, the intercept of the estimated linear variogram estimates a nugget effect only if the signal functions remain constant as the input changes.

In future research, we might try to generalize Ordinary Kriging in case of random simulation with heterogeneous variances; that is, we might relax the assumption of a stationary covariance process. We might then assume that correlations decrease with the distance between the observations:

$$\rho\left(Z(\mathbf{s}), Z(\mathbf{s}')\right)\downarrow 0 \quad \text{as} \quad |\mathbf{s}-\mathbf{s}'|\uparrow \infty$$

We might still try to compute the Kriging predictor as a weighted linear function of the observed outputs. These weights would decrease as the distance between observations or their variances increase. It is unclear whether an explicit formula for the optimal weights can be derived or whether a numerical procedure must be resorted to.

Our results may be further generalized by applying Kriging to other types of simulation models with known I/O functions, such as M/G/1 and M/M/1/K.

Acknowledgment

The authors thank three anonymous referees, especially one referee who raised many challenging questions.

References

Asmussen, S. (1992), Queuing simulation in heavy traffic. *Mathematics of Operations Research*, 17, no. 1, pp.84-111

Barton, R.R. (1994), Metamodeling: a state of the art review. *Proceedings of the 1994 Winter Simulation Conference*, eds. J.D. Tew, S. Manivannan, D.A. Sadowski, and A.F. Seila, pp. 237_244

Cheng, R.C.H. and J.P.C. Kleijnen (1999), Improved design of queuing simulation experiments with highly heteroscedastic responses. *Operations Research*, 47, no. 5, pp. 762-777

Cohen, J.W. (1969), *The single server queue*. North-Holland Publishing Company, Amsterdam

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

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, no. 1, pp. 14-29

Kleijnen, J.P.C. and W. van Groenendaal (1995), Two-stage versus sequential sample-size determination in regression analysis of simulation experiments. *American Journal of Mathematical and Management Sciences*, 15, nos. 1&2, pp. 83-114

Law, A.M. and W.D. Kelton (2000), *Simulation modeling and analysis; third edition*. McGraw-Hill, Boston

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

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

Chapter 4

Application-driven Sequential Designs for Simulation Experiments: Kriging Metamodeling

Abstract

This paper proposes a novel method to select an experimental design for interpolation in simulation. Though the paper focuses on Kriging in deterministic simulation, the method also applies to other types of metamodels (besides Kriging), and to stochastic simulation. The paper focuses on simulations that require much computer time, so it is important to select a design with a small number of observations. The proposed method is therefore sequential. The novelty of the method is that it accounts for the specific input/output function of the particular simulation model at hand; i.e., the method is application-driven or customized. This customization is achieved through cross-validation and jackknifing. The new method is tested through two academic applications, which demonstrate that the method indeed gives better results than either sequential designs based on an approximate Kriging prediction variance formula or designs with prefixed sample sizes.

4.1 Introduction

We are interested in *expensive simulations*; that is, we assume that a single simulation run takes ‘much’ computer time (say, its time is measured in days, not minutes). Therefore we devise a

Paper by Kleijnen, J.P.C. and W.C.M. Van Beers, *Journal of the Operational Research Society*, 55, issue 9, August 2004, pp. 876-883 method meant to minimize the number of simulation runs—that number is called the 'sample size' in statistics or the 'design size' or 'scheme size' in design of experiments (DOE).

We *tailor* our design to the actual simulation; that is, we do not derive a generic design such as a classic 2^{k-p} design or a Latin Hypercube Sampling (LHS) design. We can explain the differences between our designs on one hand and classic and LHS designs on the other hand, as follows.

Classic designs assume a simple 'metamodel' (also called approximate model, emulator, response surface, surrogate, etc.). A *metamodel* is a model of an input/output (I/O) function. We denote the metamodel by $Y(\mathbf{x})$ where $\mathbf{x}$ denotes the k -dimensional vector of the k inputs—called 'factors' in classic DOE. In simulation, the true I/O function is implicitly defined by the simulation model itself (in real-life experiments, 'nature' defines this function). Classic 2^{k-p} designs of resolution III assume a first-order polynomial function (optimal resolution-III designs are orthogonal matrices, under various criteria). Central composite designs (CCD) assume a second-order polynomial function. See, for example, the well-known textbook Box, Hunter, and Hunter (1978) or the recent textbook, Myers and Montgomery (2002).

LHS—much applied in Kriging—assumes I/O functions more complicated than classic designs do—but LHS does not specify a specific function for $Y(\mathbf{x})$. Instead, LHS focuses on the design space formed by the k–dimensional unit cube, defined by $0 \le x_j \le 1$ ($j = 1, \ldots, k$) after standardizing (scaling) the inputs. LHS tries to sample that space according to some prior distribution for the inputs, such as independent uniform distributions on $[0, 1]$ (or some non-uniform distribution in risk or uncertainty analysis); see McKay, Beckman, and Conover (1979), and also Koehler and Owen (1996) and Kleijnen et al. (2002).

Unlike LHS, we explicitly account for the I/O function; unlike classic DOE, we use a more realistic I/O function than a low-order polynomial. Therefore we estimate the true I/O function through *cross-validation*; i.e., we successively delete one of the I/O observations already simulated (for cross-validation see Stone (1974); for an update see Meckesheimer et al. (2002), Mertens (2001)). In this way we estimate the uncertainty of output at input combinations not yet observed. To measure this uncertainty, we use the *jackknifed* variance. For jackknifing see the classic article by Miller (1974); for an update see again Meckesheimer et al. (2002) and

Mertens (2001). We also compare our designs (based on cross-validation and jackknifing) to sequential designs based on a formula that approximates the variance of the Kriging predictor.

It turns out that our procedure concentrates on input combinations (design points, simulation scenarios) in sub-areas that have *more interesting* I/O behavior. In our Example I, we spend most of our simulation time on the challenging 'explosive' part of a hyperbolic function (which may represent mean steady-state waiting time of single-server waiting systems). In Example II, we avoid spending much time on the relatively flat part of the fourth-degree polynomial I/O function with multiple local hills. (The reader may take a peek at Figures 3 and 6 discussed later.)

We make our procedure *sequential* for the following two reasons:

1. Sequential procedures are known to be more 'efficient'; that is, they require fewer observations than fixed-sample procedures; see the statistics literature, for example, Ghosh and Sen (1991) and Park et al. (2002).
2. Simulation experiments proceed sequentially (unless parallel computers are used).

Our Application-Driven Sequential Design (ADSD) does not provide tabulated designs; instead, we present a procedure for generating a sequential design for the actual (simulation) experiment.

Note that a different ADSD is developed by Sasena, Papalambros, and Govaerts (2002). They, however, focus on optimization instead of sensitivity analysis (we think that optimization is more applied in engineering sciences than in management sciences, because the latter sciences involve softer performance criteria). Moreover, they use the 'generalized expected improvement function' assuming a Gaussian distribution, as proposed by Jones, Schonlau, and Welch (1998). We, however, use distribution-free jackknifing and cross-validation for a set of candidate input combinations. Sasena et al. (2002) examine several criteria for selecting the next input combination to be simulated, including the 'maximum variance' criterion; the latter criterion is the one we use. (An alternative to their single, globally fitted Kriging metamodel for constrained optimization is a sequence of locally fitted first-order polynomials; see Angün et al. (2002)) Related to Sasena et al. (2002) is Watson and Barnes (1995). More research is needed to compare our method with Sasena et al.'s method (also see our final section, called 'Conclusions and further research').

The remainder of this paper is organized as follows. First we summarize the basics of Kriging. Then we summarize DOE and Kriging. Subsequently we explain our method, which uses cross-validation and jackknifing to select the next input combination to be simulated; this section also discusses sequentialization and stopping. Next we demonstrate the procedure through two academic applications, which shows that our method gives better results than a design with a prefixed sample size; moreover, estimated Gaussian and linear correlation functions (variograms)—used in Kriging—give approximately the same results. The final section presents conclusions and topics for further research.

4.2 Kriging basics

Kriging is named after the South-African mining engineer D.G. Krige. It is an interpolation method that predicts unknown values of a random function or random process; see Cressie (1993)'s classic Kriging textbook and equation (1) below. More precisely, a Kriging prediction is a weighted linear combination of all output values already observed. These weights depend on the distances between the location to be predicted and the locations already observed. Kriging assumes that *the closer the input data are, the more positively correlated the prediction errors are*. This assumption is modeled through the correlogram or the related variogram, discussed below.

Nowadays, Kriging is also popular in *deterministic simulation* (to model the performance of computer chips, television screens, etc.); see Sacks et al. (1989)'s pioneering article, and—for an update—see Simpson et al. (2001a). Compared with linear regression analysis, Kriging has an important advantage in deterministic simulation: Kriging is an *exact interpolator*; that is, predicted values at observed input values are exactly equal to the observed (simulated) output values.

Kriging assumes the following *metamodel*:

$$Y(\mathbf{x}) = \mu(\mathbf{x}) + \delta(\mathbf{x}) \text{ with } \delta(\mathbf{x}) \sim NID(0, \sigma^2(\mathbf{x})) \qquad (1)$$

where μ is the mean of the stochastic process $Y(\cdot)$, and $\delta(\mathbf{x})$ is the additive *noise*, which is assumed normally independently distributed (NID) with mean zero and variance $\sigma^2(\mathbf{x})$. *Ordinary Kriging* further assumes a *stationary covariance process* for $Y(\mathbf{x})$ in (1): the expected values $\mu(\mathbf{x})$ are constant and the covariances of $Y(\mathbf{x}+\mathbf{h})$ and $Y(\mathbf{x})$ depend only on the distance (or lag) $|\mathbf{h}| = |(\mathbf{x}+\mathbf{h})-(\mathbf{x})|$.

As we mentioned above, the Kriging *predictor* for the unobserved input $\mathbf{x}_0$—denoted by $\hat{Y}(\mathbf{x}_0)$—is a weighted linear combination of all the (say) *n* observed output data:

$$\hat{Y}(\mathbf{x}_0) = \sum_{i=1}^{n} \lambda_i \cdot Y(\mathbf{x}_i) = \boldsymbol{\lambda}' \cdot \mathbf{Y} \tag{2}$$

with $\sum_{i=1}^{n} \lambda_i = 1$, $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)'$ and $\mathbf{Y} = (y_1, \ldots, y_n)'$. To choose these weights, the 'best' linear unbiased estimator (BLUE) is derived: this estimator minimizes the mean-squared prediction error $\mathrm{MSE}\left(\hat{Y}(\mathbf{x}_0)\right) = E\left(\left(Y(\mathbf{x}_0) - \hat{Y}(\mathbf{x}_0)\right)^2\right)$, with respect to $\boldsymbol{\lambda}$. Obviously, this solution depends on the covariances, which may be characterized by the *variogram*, defined as $2\gamma(\mathbf{h}) = var(Y(\mathbf{x}+\mathbf{h}) - Y(\mathbf{x}))$. (We follow Cressie (1993), who uses variograms, whereas Sacks et al. (1989) use correlation functions; also see our discussion on the estimation of variograms in the section called 'Two examples'.) An example variogram is given in Figure 1.

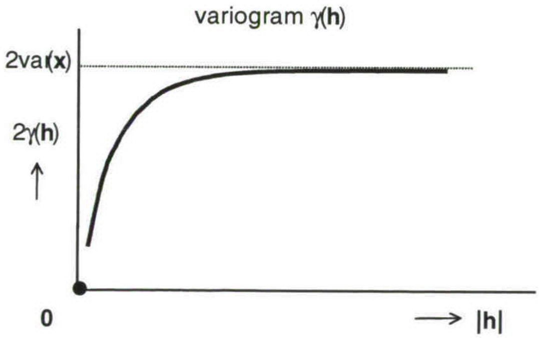


Figure 1: An example variogram

It can be proven that the *optimal* weights in (2) are

$$\boldsymbol{\lambda}' = \left(\boldsymbol{\gamma} + \mathbf{1} \frac{1 - \mathbf{1}' \boldsymbol{\Gamma}^{-1} \boldsymbol{\gamma}}{\mathbf{1}' \boldsymbol{\Gamma}^{-1} \mathbf{1}} \right)' \boldsymbol{\Gamma}^{-1} \tag{3}$$

where $\boldsymbol{\gamma}$ is the vector of (co)variances $(\gamma(\mathbf{x}_0 - \mathbf{x}_1), \ldots, \gamma(\mathbf{x}_0 - \mathbf{x}_n))'$; $\boldsymbol{\Gamma}$ is the $n \times n$ matrix whose $(i, j)^{\text{th}}$ element is $\gamma(\mathbf{x}_i - \mathbf{x}_j)$; $\mathbf{1} = (1, \ldots, 1)'$ is the vector of ones. We point out that the weights in (3) vary with the prediction point, whereas regression analysis uses the same estimated metamodel for all prediction points.

Because the (co)variances in (3) are unknown, they are based on the estimated variogram. If the *random* character of the resulting estimated optimal weights $\hat{\boldsymbol{\lambda}}$ is ignored, then the variance of the resulting linear estimator at a fixed point $\boldsymbol{x}_0$ is

$$\sigma_k^2(\mathbf{x}_0 \mid \hat{\lambda} = \lambda) = 2 \cdot \sum_i^n \lambda_i \gamma(\mathbf{x}_0 - \mathbf{x}_i) - \sum_i^n \sum_j^n \lambda_i \lambda_j \gamma(\mathbf{x}_i - \mathbf{x}_j); \tag{4}$$

see Cressie (1993, p. 122), who does not explicitly mention the conditional character of (4).

Further details on Kriging are provided by Cressie (1993); an update is Van Beers and Kleijnen (2003).

4.3 DOE and Kriging

A *design* is a set of (say) *n* combinations of the *k* factor values. These combinations are usually bounded by 'box' constraints: $a_j \le x_j \le b_j$, where $a_j, b_j \in R$ with $j = 1, \ldots, k$. The set of all feasible combinations is called the *experimental region* (say) *H*. We suppose that *H* is a *k*-dimensional unit cube, after rescaling the original rectangular area (also see the Introduction).

Our goal is to find a design—for Kriging predictions within *H*—with the *smallest size* that satisfies a certain criterion. The literature proposed several criteria: see Sacks et al. (1989, p. 414). Most of these criteria are based on the Mean Squared prediction Error,

$\mathrm{MSE}\left(\hat{Y}(\mathbf{x})\right) = E\left(\hat{Y}(\mathbf{x}) - Y(\mathbf{x})\right)^2$ where the predictor $\hat{Y}(\mathbf{x})$ follows from (2) and the true output $Y(\mathbf{x})$ was defined in (1). (An alternative considers $100(1-\alpha)\%$ prediction regions for $y(\mathbf{x})$ and inter-quantile ranges for $\hat{y}(\mathbf{x})$; see Cressie 1993, p. 108.) However, most progress has been made through the *Integrated Mean Squared Error* (IMSE); see Bates et al. (1996): choose the design that minimizes

$$IMSE = \int_H \mathrm{MSE}\left(\hat{Y}(\mathbf{x})\right)\phi(\mathbf{x})d\mathbf{x} \tag{5}$$

for a given weight function $\phi(\mathbf{x})$.

To validate the design, Sacks et al. (1989, p. 416) compare the predictions with the known true values in a *test set* of size (say) *m*. They assume $\phi(\mathbf{x})$ to be uniform, so IMSE in (5) can be estimated by the Empirical Integrated Mean Squared Error (EIMSE):

$$EIMSE = \frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}_i(\mathbf{x}) - y_i(\mathbf{x})\right)^2 . \tag{6}$$

Note that criteria such as (5) are more appropriate in sensitivity analysis than in simulation optimization; see Sasena et al. (2002) and also Kleijnen and Sargent (2000) and Kleijnen (1998).

4.4 Application-driven sequential design

4.4.1 Pilot input combinations

We start with a *pilot design* of size (say) n_0. To select n_0 *specific* points, we notice that Kriging gives very bad predictions in case of *extrapolation* (i.e., predictions outside the convex hull of the observations obtained so far). Indeed, in our examples we find very bad results (not displayed). Therefore, we select the 2^k vertices of *H* as a subset of the pilot design. In our two examples with a *single* input ($k = 1$), this choice implies that one input value is the minimum and one is the maximum of the input's range; see Figure 2 (other parts of this figure will be explained below, in next subsections).

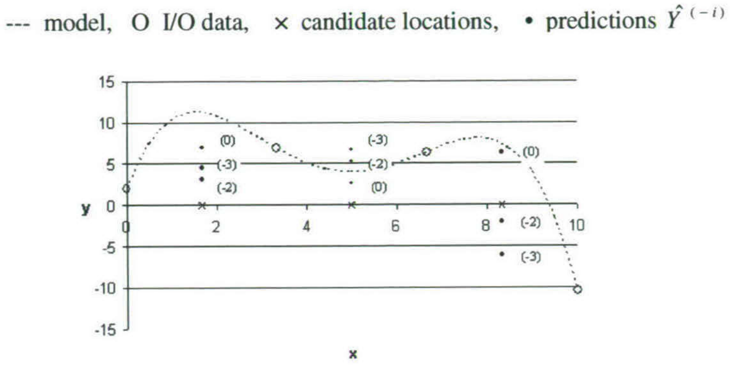


Figure 2: Fourth-order polynomial example, including four pilot observations and three candidate inputs with predictions based on cross-validation, where (-*i*) denotes which observation *i* is dropped in the cross validation

Besides these 2^k vertices, we must select some more input combinations to *estimate the variogram*. Like Cressie (1993) we assume either a Gaussian variogram

$$\gamma(h) = c_0 + c_1(1 - \exp(-h/a)) \tag{7}$$

or a linear variogram

$$\gamma(h) = c_0 + c \cdot h\,. \tag{8}$$

Obviously, estimation of the variogram (7) requires at least three different values of *h* (for example, the values $0, \frac{1}{2}, 1$); thus at least three different I/O combinations. Moreover—as we shall see—our approach uses cross-validation, which implies that we drop one of the n_0 observations and re-estimate the variogram; i.e., cross-validation necessitates one extra I/O combination.

In practice, we may select a 'small' set of additional observations—besides the 2^k corner points—using a standard *space-filling design*, which ensures that no two design points are too close to each other. More specifically, we propose a *maximin* design, which packs all design points in hyper spheres with maximum radius; see Koehler and Owen (1996, p. 288). In our examples, we take—besides the two endpoints of the factor's range—two additional points. The latter points we place such that all four observed points are equidistant; see again Figure 2. (Future research may investigate alternative sizes n_0 and components $\boldsymbol{x}$.)

4.4.2 Candidate input combinations

After selecting and actually simulating a 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 simulation model, we choose (say) *c candidate* points—without actually running any expensive simulations for these candidates (as we shall see in next subsection).

First we must select a *value* for *c*. In Figure 2 we select three candidate input values (had we taken more candidates, then we would have to perform more Kriging calculations; in general, the latter calculations are small compared with the 'expensive' simulation computations).

Next we must select *c specific* candidates. Again, we use a space-filling design (as we did for the pilot sample). In Figure 2 we select the three candidates *halfway* between the four input values already observed. (Future research may investigate how to use a space filling design to select candidates, ignoring candidates that are too close to the points already observed. In practice, LHS designs are attractive since they are so simple: LHS is part of spreadsheet add-ons such as @Risk.)

4.4.3 Cross-validation

To select a 'winning' candidate for actual (expensive) simulation, we estimate the variance of the predicted output at each candidate input—without any actual simulation. Therefore we use cross-validation and jackknifing, as follows.

Given a set of observed I/O data (x_i, y_i) with $i = 1, \ldots, n$ (initially, $n = n_0$), we eliminate observation *i* and obtain the *cross-validation* sample (with only $n - 1$ observations):

$$S^{(-i)} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_{i-1}, y_{i-1}), (x_{i+1}, y_{i+1}), \ldots, (x_n, y_n)\}. \tag{9}$$

From the sample in (9), we could compute the Kriging prediction for the output for each candidate. However, to avoid extrapolation (see previous subsection 'Pilot input combinations'), we do not eliminate the observations at the vertices: of the cross-validation sample in (9) we use only (say) n_c observations. The predictions are analogous to (2) replacing *n* by n_c; in case of

$k = 1$ we take $n_c = n_0 - 1$. Obviously, we must re-estimate the optimal weights in (2), using (3) (also see the 'binning' discussion at the end of next subsection). Figure 2 shows the $n_c = n_0 - 1 = 3$ Kriging predictions (say) $\hat{Y}^{(-i)}$ after deleting observation *i* as in (9), for each of the $c = 3$ candidates.

Figure 2 suggests that it is most difficult to predict the output at the candidate point $x = 8.33$. To quantify this prediction uncertainty, we use jackknifing.

4.4.4 Jackknifing

First, we calculate the jackknife's *pseudo-value* for candidate *j*, which is defined as the following weighted average of the original and the cross-validation predictors:

$$\tilde{y}_{j;i} = n_c \times \hat{Y}_j^{(-0)} - (n_c - 1) \times \hat{Y}_j^{(-i)} \quad \text{with} \quad j = 1, \ldots, c \text{ and } i = 1, \ldots, n_c \tag{10}$$

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 (10), we estimate the *jackknife variance* for candidate *j*:

$$\tilde{s}_j^2 = \frac{1}{n_c(n_c - 1)} \sum_{i=1}^{n_c} (\tilde{y}_{j;i} - \bar{\tilde{y}}_j)^2 \quad \text{with} \quad \bar{\tilde{y}}_j = \frac{1}{n_c} \sum_{i=1}^{n_c} \tilde{y}_{j;i} \, . \tag{11}$$

Note that we also experimented with other measures of variability, for example, the 90% interquantile; all these measures gave the same type of design.

Finally, to select the *winning* candidate (say) *w* for actual simulation, we find the maximum of the jackknife variances in (11):

$$w = \arg(\underset{j}{max}\{\tilde{s}_j^2\}) \, . \tag{12}$$

Note that a *candidate* location close to a *deleted* observation lies relative far away from the remaining observations. Hence, such a candidate is less correlated with its neighboring points.

Consequently, its Kriging predictor becomes rather uncertain. However, this phenomenon holds for each deleted observation.

Note further that to reduce the computer time needed by our procedure (not by the simulation itself), we estimate the variogram from *binned* distances: for *n* inputs, we classify the $n(n-1)/2$ possible distances *h* in (say) $n_b < n$ equally sized intervals or 'bins'. These intervals should be as small as possible to retain spatial resolution, yet large enough to stabilize the variogram estimator. Journel and Huijbregts (1978) recommend at least thirty distinct pairs in each interval. For the n_b midpoints of these intervals, we calculate the average squared difference to estimate the variogram; see Cressie (1993, p. 69). In our examples we use $n_b = 15$.

4.4.5 Sequentialization

Once we have simulated the 'winning' candidate selected through (12), we add the new observation to the set of observations; see *S* in (9)—now with superscript (-0) and with $n+1$ members.

Next, we choose a new set of candidates with respect to this augmented set. For example, in Figure 2 we add as new candidates $x = 1.67$, $x = 5$, $x = 7.5$ and $x = 9.17$; these candidates are not shown in Figure 2, but the winning candidate is shown as part of Figure 3.

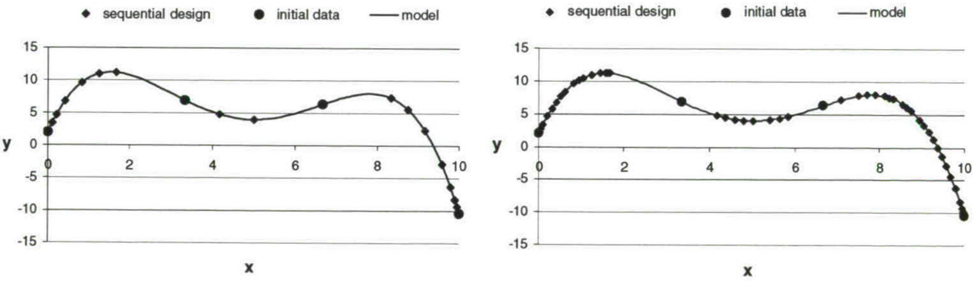


Figure 3a

Figure 3b

Figures 3: Figure 2 continued with $n = 19$ (3a) or $n = 54$ (3b) observations

The 'dynamics' of our procedure is demonstrated by Figure 4, which shows the *order* in which input values are selected—in a total sample size $n = 50$.

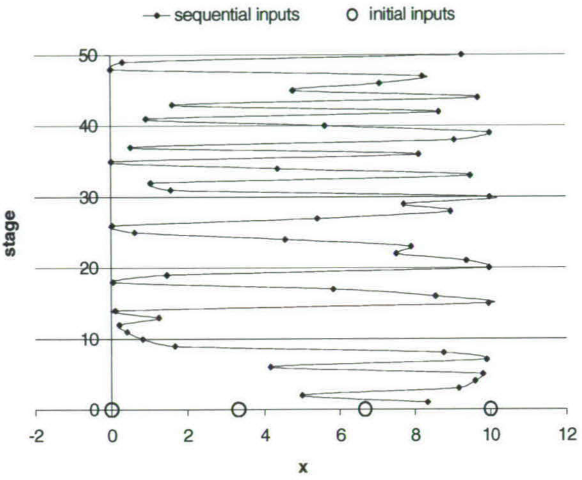


Figure 4: Dynamics of sequential sampling for Example 1

4.4.6 Stopping rule

To stop our sequential procedure, we measure the *Successive Relative Improvement* (SRI) after *n* observations:

$$\text{SRI}_n = \left| \max_j \{\tilde{s}_j^2\}_n - \max_j \{\tilde{s}_j^2\}_{n-1} \right| \Big/ \max_j \{\tilde{s}_j^2\}_{n-1} \tag{13}$$

where $\max_j\{\tilde{s}_j^2\}_n$ denotes the maximum jackknife variance (see (12)) after *n* observations.

Figure 5 shows SRI for up to $n = 50$ in Example I (detailed in next subsection). There are no essential changes in (13) beyond $n = 15$. In the literature (including Sasena et al. (2002) and Jones et al. (1998)), we did not find an appealing stopping criterion for our sequential design; future research may be needed.

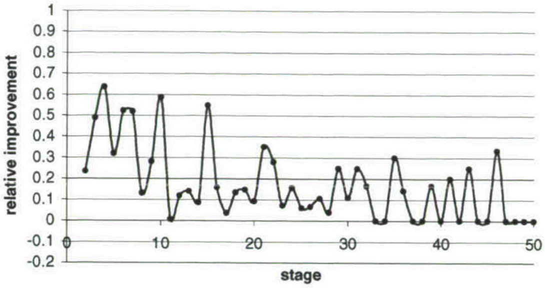


Figure 5: Successive relative improvements for 50 observations in hyperbole example

We *stop* our sequential procedure as soon as we find no 'substantial' reduction for SRI. However, SRI may fluctuate greatly in the first stages, so we might stop *prematurely*. To avoid such stopping, we select a minimum value (say) $n_{\min}$ so that the complete design contains $n = n_0 + n_{\min}$ observations. Figure 3a used $n_{\min} = 15$, whereas Figure 3b used $n_{\min} = 50$ (Figure 2 is the part of Figure 3 that corresponds with $n = 4$.)

In practice—as Kleijnen et al. (2002) point out—simulation experiments may stop prematurely (e.g., the computer may break down). Our procedure then still gives useful information.

4.5 Two examples

4.5.1 Example I: a hyperbolic I/O function

Consider the following hyperbole:

$$y = \frac{x}{1-x} \text{ with } 0 < x < 1. \tag{14}$$

We are interested in this example, because *y* in (14) equals the expected waiting time in the steady state of a single-server system with Markovian (Poisson) arrival and service times (denoted by M/M/1). This system has a single input parameter, namely the traffic load *x*, which is the ratio of the arrival rate and the service rate. This system is a building block in many realistic discrete-event simulation models; see Law and Kelton (2000, p. 12) and also Van Beers and Kleijnen (2003).

When applying our approach to (14), we decided to select a pilot sample size $n_0 = 4$ and a minimum sample size value $n_{\min} = 10$. We stop the sequential procedure as soon as the SRI in (13) drops below 5%; this results in a total sample size $n = 19$. Also see Figure 6a. Replacing 5% by 1% gives $n = 36$; see Figure 6b.

Figures 6 demonstrate that our final design selects relative few input values in the area that generates an approximately linear I/O function, whereas it selects many input values in the exploding part (where *x* approaches one).

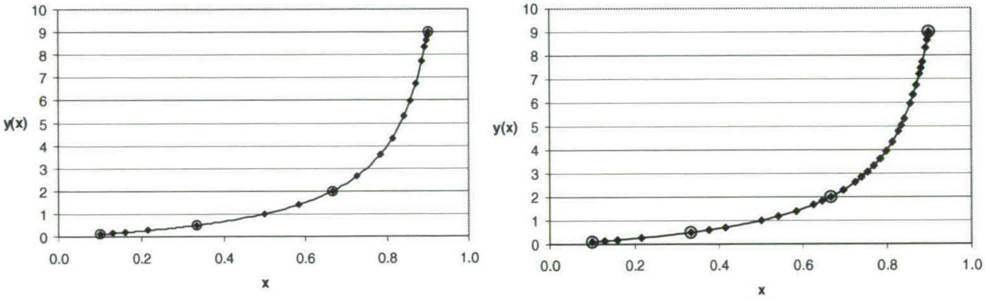


Figure 6a

Figure 6b

Figures 6: Hyperbole example, including four pilot observations and with $n = 19$ (6a) or $n = 36$ observations

We think that our design is intuitively appealing—but 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 ($n = 19$ or $n = 36$):

i. A *sequential* design based on the *approximate* Kriging variance formula (4). We then select 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 as the next point the input farthest away from the old inputs, namely $x = 0.5$ (also see Goovaerts's statement on http://www.sph.umich.edu/geomed/mods/geostats_lite/lec/krigvariance.html). 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*. LHS divides the total range of the input variable into *n* mutually exclusive and exhaustive intervals of equal length. Within each interval, LHS samples a uniformly distributed value. To estimate the resulting variability, we decided to obtain ten LHS samples, from which we estimate the mean and the standard deviation (standard error).

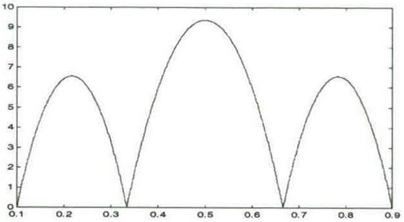


Figure 7: Approximate Kriging variance in initial design

From the *n o*bservations per design we compute the Kriging predictors for the 32 true test values, and calculate the squared error per test value. From the 32 values we compute the average —see EIMSE in (6), which corresponds with the L_2 norm—and the maximum or L_∞ norm. We find substantially better results for our designs; see Table 1.

Table 1: IMSE of three design types for hyperbole (Example I)

	ADSD		Krig Var		LHS	
	EIMSE	L_∞	EIMSE	L_∞	EIMSE	L_∞
$n = 19$	$8.90 * 10^{-4}$	0.0759	$80.08 * 10^{-4}$	0.3460	$61.4 * 10^{-4}$	0.3559
					$(48.1 * 10^{-4})$	(0.1740)
$n = 36$	$1.19 * 10^{-4}$	0.0303	$8.11 * 10^{-4}$	0.1501	$2.76 * 10^{-4}$	0.0791
					$(0.98 * 10^{-4})$	(0.0185)

4.5.2 Example II: a fourth-order polynomial I/O function

As Van Beers and Kleijnen (2003) did, we consider

$$y = -0.0579x^4 + 1.11x^3 - 6.845x^2 + 14.1071x + 2, \tag{15}$$

which is a multi-modal function; see again Figure 2 .

For our design, we select $n_0 = 4$, $n_{\min} = 10$, and a SRI smaller than 5%. This gives a sequential design with 18 observations. A SRI smaller than 1% gives a final (sequential) design with 24 observations (Example I resulted in 36 observations).

Figure 8 demonstrates that our final design selects relative few input values in the area that generates an approximately linear I/O function, whereas it selects many input values near the edges, where the function changes much.

We again compare our design with the two alternative designs discussed above. We find substantially better results for our designs; see Table 2.

Table 2: IMSE for three types of designs for fourth degree polynomial (Example II)

	ADSD		Krig Var		LHS	
	EIMSE	L_∞	EIMSE	L_∞	EIMSE	L_∞
$n = 18$	0.1741	1.0470	0.5793	0.6718	0.5855 (0.5574)	3.3011 (1.9706)
$n = 24$	0.0121	0.2503	0.2690	0.5133	0.2473 (0.2112)	2.1212 (1.3837)

Note that we focus on sensitivity analysis, not optimization. For example, our method selects input values—not only near the 'top'—but also near the 'bottom' of (15). If we were searching for a maximum, we would adapt our procedure such that it would not collect data near an obvious minimum.

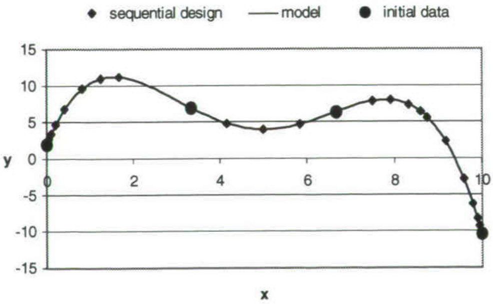


Figure 8: Final design for fourth-order polynomial example with SRI < 1% and $n = 24$

4.5.3 Estimated variograms: Gaussian versus linear

We also investigate the influence of the assumed variogram, namely a Gaussian variogram and a linear variogram; see (7) and (8). We use a single-stage design with 21 observations. We use ordinary least squares for these estimators (whereas Sacks et al. (1989) assume a Gaussian correlation function and use maximum likelihood estimation, which takes much more computer time and may involve numerical problems).

The Gaussian and the linear variograms result in two designs that look very similar, for both Example I and Example II. More precisely, when using a test set of nine equidistant input values, Kriging predictions based on a Gaussian variogram give an EIMSE of 0.3702, whereas a linear variogram gives 0.3680 for Example I. Analogously, Example II gives 0.0497 and 0.0482. So the Gaussian and linear variograms give similar values for EIMSE. The linear variogram, however, is simpler: no data transformation is needed.

4.6 Conclusions and further research

To avoid expensive simulation runs, we propose cross-validation and jackknifing to estimate the variances of the outputs for *candidate* input combinations. We actually simulate only the candidate with the *highest* estimated variance. This procedure we apply *sequentially*.

Our two examples show that our procedure simulates relatively many input combinations in those sub-areas that have interesting I/O behavior. Our design gives smaller prediction errors than either sequential designs based on the approximate variance formula in (4) or single-stage designs do.

In future research, we may extend our approach to

1. alternative *pilot-sample* sizes n_0 with alternative space-filling input combinations $\boldsymbol{x}$ (Jones et al. (1998, p. 21), propose $n_0 = 10k$ and an adjusted LHS design)
2. alternative space-filling designs for the selection of *candidate* input combinations, ignoring candidates that are too close to the points already observed in any preceding

stages (such an alternative design may be a nearly-orthogonal LHS design; see Kleijnen et al. (2002))

3. a *stopping criterion* for our sequential design
4. *multiple* inputs ($k > 1$)
5. *realistic* simulation models (instead of our Examples I and II)
6. comparison of our approach with *Sasena et al.* (2002)'s approach
7. *stochastic* simulation models (focus of our current research)
8. *other metamodels*, such as linear regression models (see Kleijnen and Sargent (2000)) and neural nets (see Simpson et al. (2001b)).

Acknowledgment

Bert Bettonvil (Tilburg University) and Paul Switzer (Stanford University) provided very useful comments on an earlier version.

References

Angün, E. D. den Hertog, G. Gürkan, and J.P.C. Kleijnen (2002), Response surface methodology revisited. *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yücesan, C.H. Chen, J.L. Snowdon and J.M. Charnes, pp. 377-383

Bates, R.A., R.J. Buck, E. Riccomagno and H.P. Wynn (1996), Experimental design and observation for large systems. *Royal Statistical Society*. 58, no. 1, pp. 77-94

Box, G.E.P., W.G. Hunter and J.S. Hunter (1978), *Statistics for experimenters: an introduction to design, data analysis and model building*. John Wiley & Sons, Inc., New York

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

Ghosh, B.K. and P.K. Sen (editors), 1991, *Handbook of sequential analysis*. Marcel Dekker, Inc., New York

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

Journel, A.G. and C.J. Huijbregts (1978), *Mining geostatistics*, Academic Press, London

Kleijnen, J.P.C. (1998), Experimental design for sensitivity analysis, optimization, and validation of simulation models. Chapter 6 in: *Handbook of simulation*, edited by J. Banks, Wiley, New York, pp. 173-223

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, no. 1, pp. 14-29

Kleijnen, J.P.C., S.M. Sanchez, T.W. Lucas and T.M. Cioppa (2002), A user's guide to the brave new world of designing simulation experiments. Working Paper (preprint: http://center.kub.nl/staff/kleijnen/papers.html)

Koehler, J.R. and A.B. Owen (1996), Computer experiments. *Handbook of statistics*, by S. Ghosh and C.R. Rao, vol. 13, pp. 261-308

Law, A.M. and W.D. Kelton (2000), *Simulation modeling and analysis, third edition*, McGraw-Hill, Boston

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, no. 2, pp. 239-245 (reprinted in 2000: *Technometrics*, 42, no. 1, pp. 55-61

Meckesheimer, M., R.R. Barton, T.W. Simpson, and A.J. Booker (2002), Computationally inexpensive metamodel assessment strategies. *AIAA Journal*, 40, no. 10, pp. 2053-2060

Mertens, B.J.A. (2001), Downdating: interdisciplinary research between statistics and computing. *Statistica Neerlandica*, 55, no. 3, pp. 358-366

Miller, R.G. (1974), The jackknife - a review. *Biomatrika*, 61, pp. 1-15

Myers, R.H. and D.C. Montgomery (2002). *Response surface methodology: process and product optimization using designed experiments; second edition*. Wiley, New York

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, no. 6, pp. 981-990

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

Sasena, M.J, P. Papalambros, and P. Goovaerts (2002), Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization* 34, no.3, pp. 263-278

Simpson, T.W., T.M. Mauery, J.J. Korte, and F. Mistree (2001a), Kriging metamodels for global approximation in simulation-based multidisciplinary design optimization. *AIAA Journal*, 39, no. 12, 2001, pp. 2233-2241

Simpson, T.W., J. Peplinski, P.N. Koch, and J.K. Allen (2001b), Metamodels for computer-based engineering design: survey and recommendation. *Engineering with Computers*, 17, no. 2, pp. 129-150

Stone, M. (1974), Cross-validatory choice and assessment of statistical predictions, *Journal Royal Statistical Society, Series B*, 36, no. 2, pp. 111-147

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

Watson, A.G. and R.J. Barnes (1995), Infill sampling criteria to locate extremes. *Mathematical Geology*, 27, no. 5, pp. 589-608

Chapter 5

Customized Sequential Designs for Random Simulation Experiments: Kriging Metamodeling and Bootstrapping

Abstract

This paper proposes a novel method to select an experimental design for interpolation in random simulation, especially discrete event simulation. (Though the paper focuses on Kriging, this design approach may also apply to other types of metamodels such as linear regression models.) Assuming that simulation requires much computer time, it is important to select a design with a small number of observations (or simulation runs). The proposed method is therefore sequential. Its novelty is that it accounts for the specific input/output behavior (or response function) of the particular simulation at hand; i.e., the method is customized or application-driven. A tool for this customization is bootstrapping, which enables the estimation of the variances of predictions for inputs not yet simulated. The new method is tested through two classic simulation models: example 1 estimates the expected steady-state waiting time of the M/M/1 queueing model; example 2 estimates the mean costs of a terminating (*s, S*) inventory simulation. For these simulations the novel design indeed gives better results than Latin Hypercube Sampling (LHS) with a prefixed sample of the same size.

Paper by Van Beers, W.C.M. and J.P.C. Kleijnen, submitted to *INFORMS Journal on Computing*

5.1 Introduction

In this paper, we focus on *expensive simulations*; that is, we assume that a single simulation run takes 'much' computer time. Consequently, 'interpolation' is needed; i.e., from the simulated input/output (I/O) data, the outputs are predicted for input combinations not yet simulated. We devise a method that is meant to minimize the number of simulation runs for such interpolation. We *tailor* our design of experiments (DOE) to the actual simulation; that is, we do not derive a generic design such as a classic design (for example, a 2^{k-p} design) or a LHS design. The differences between customized and generic designs are as follows (also see Kleijnen and Van Beers (2004), who focus on deterministic simulation).

A *metamodel* is a model of the I/O function (or 'response function') implied by the underlying simulation model. We denote the metamodel by $Y(\mathbf{x})$ where $\mathbf{x}$ denotes the k-dimensional vector of the k inputs (factors) so $\mathbf{x} = (x_1, \ldots, x_j, \ldots, x_k)'$. *Classic DOE* assumes a simple metamodel. For example, designs of resolution III (including certain 2^{k-p} designs) assume a first-order polynomial I/O function. Composite designs (CCD) assume a second-order polynomial. These designs are discussed for physical experiments in (for example) the well-known textbook Box, Hunter, and Hunter (1978) and the recent textbook Myers and Montgomery (2002); for simulation experiments we refer to Kleijnen (1987).

LHS (much applied in Kriging, described below) assumes that an adequate metamodel is more complicated than a low-order polynomial. LHS, however, does not assume a specific metamodel. Instead, LHS focuses on the design space formed by the k–dimensional unit cube, defined by $0 \leq x_j \leq 1$ ($j = 1, \ldots, k$) after standardizing (scaling) the inputs. LHS is one of the *space filling* designs: LHS samples that space according to some prior distribution for the inputs, such as independent uniform distributions on $[0, 1]$; see McKay, Beckman, and Conover (1979), and also Kleijnen et al. (2005), Koehler and Owen (1996), and Santner, Williams, and Notz (2003).

Unlike LHS, we explicitly account for the I/O function. Unlike classic DOE, we assume that a low-order polynomial (estimated through regression analysis) gives an inadequate approximation of the I/O function. In our method we estimate the uncertainty of predicted outputs at unobserved input combinations (these combinations are also called scenarios, design points, combinations of factor levels, or simulation inputs). To estimate the uncertainty of these predictions—caused by the noise and the shape of the I/O function—we use *bootstrapping*; i.e., we resample the outputs for each scenario already simulated (for bootstrapping in general see the classic textbook, Efron and Tibshirani 1993; for bootstrapping in the validation of regression metamodels in simulation see Kleijnen and Deflandre 2005).

We make our procedure *sequential* for the following two reasons.

1. Sequential statistical procedures are known to be more 'efficient'; that is, they require fewer observations than fixed-sample (one-shot) procedures; see, for example, the handbook by Ghosh and Sen (1991) and the recent article by Park et al. (2002).
2. Simulation experiments proceed sequentially (unless parallel computers are used; our procedure also fits parallel computers).

The literature on *deterministic* simulation shows several designs that—like ours—account for the specific simulation's I/O function, and are sequential. For example, Crary (2002) discusses G-optimal and I-optimal designs, which the DOE literature defines as follows. G-optimal designs minimize the *maximum* Mean Squared Error (MSE) of the predicted output; I-optimal or Integrated MSE (IMSE) designs minimize the *average* MSE (obviously, the MSE reduces to the variance if the predictor is unbiased; see (5) and (6) below). Williams, Santner, and Notz (2000, 2002) use a Bayesian approach to derive sequential IMSE designs. Sasena, Papalambros, and Govaerts (2002) derive sequential designs for the optimisation of deterministic simulation models. Kleijnen and Van Beers (2004) derive customized sequential designs for deterministic simulations. We, however, focus on DOE for random simulations, and we seem to be the first to apply bootstrapping for this problem. (Random simulation includes Discrete Event Dynamic Systems or DEDS simulation such as M/M/1 simulation, but also simulation models consisting of stochastic difference equations.)

We shall see that our designs select most of their input combinations in sub-areas that have *more interesting* I/O behavior. In our first example we spend most of our computer simulation time on the challenging 'explosive' part of the metamodel that estimates the mean steady-state waiting time for various traffic rates of single-server queueing systems with Markovian (Poisson) arrival and service times—known as the M/M/1 model. (The reader may take a peek at Figure 1, discussed in subsection 5.1.) In our second example, we estimate the average total costs in an (*s*, *S*) inventory model; there are several variations on this model, but we take the specification given by Law and Kelton (2000). Again, we find a concentration of the input combinations in the sub-area where the metamodel shows steep slopes. (See Figure 5, detailed in subsection 5.2.) In both examples, we compare our designs with LHS; our designs give better predictions.

The remainder of this paper is organized as follows. Section 2 summarizes the basics of Kriging. Section 3 summarizes DOE and Kriging. Using the M/M/1 model, section 4 explains our method, which applies bootstrapping—to estimate the variances of the Kriging predictions for candidate inputs not yet simulated—and sequentially selects as the next input to be simulated, the one with the largest bootstrap variance. Section 5 demonstrates the procedure through two classic examples: subsection 5.1 uses M/M/1 simulations, and subsection 5.2 uses an (*s*, *S*) inventory model with two inputs. For both examples our method gives better results than LHS with a prefixed sample size. Section 6 presents conclusions and topics for further research.

5.2 Kriging basics

Kriging (named after the South-African mining engineer Krige) is an interpolation method that predicts unknown values of a random function or random process; see Journel and Huijbregts (1978) and Cressie's (1993) classic Kriging textbook on spatial (geo)statistics. Whereas spatial statistics considers the two-dimensional 'location' as the known input of this process, simulation considers the *k*–dimensional 'scenario' as input; see Sacks et al.'s (1989) classic article on the Design and Analysis of Computer Experiments (DACE)—these computer experiments concern deterministic simulation. Random (stochastic) simulation—including DEDS simulations—is the topic of our paper.

More precisely, a Kriging prediction is a weighted linear combination of all output values already observed. The weights depend on the distances between the new input to be predicted and the old inputs already observed. Kriging assumes that *the closer the inputs are, the more* *positively correlated the outputs are*. Mathematical formulations follow in equations (1) through (4).

Currently, Kriging is frequently applied in *deterministic simulation*, which is much used in engineering; again see Sacks et al. (1989); for an update see Simpson et al. (2001). In deterministic simulation, Kriging has an important advantage over regression analysis: the predicted values at old inputs are exactly equal to the observed (simulated) outputs.

In *random simulation*, however, this property disappears. Now, each scenario is simulated several times—with non-overlapping pseudo-random number (PRN) streams. Van Beers and Kleijnen (2003) show that Kriging interpolates the *average* output per scenario. These averages, however, are still random, so the property that at scenarios already simulated the Kriging predictions equal the averages, loses its intuitive appeal. Still, Kriging may be attractive because it may decrease the prediction *bias* (and hence the MSE) at scenarios close together. Indeed, in the examples presented by Van Beers and Kleijnen (2003) the Kriging predictions are much better than the regression predictions (regression analysis may be useful for other goals such as screening and validation; see Kleijnen et al. 2004). Therefore we do not further discuss regression analysis in this paper.

Mathematically formulated, Kriging assumes the following metamodel:

$$Y(\mathbf{x}) = \mu(\mathbf{x}) + \delta(\mathbf{x}) \text{ with } \delta(\mathbf{x}) \sim \text{IID}(0, \sigma^2(\mathbf{x})) \tag{1}$$

where $\mu(x)$ is the mean of the stochastic process $Y(x)$, and $\delta(\mathbf{x})$ is the additive *noise*, which is assumed independently and identically distributed (IID) with mean zero and variance $\sigma^2(\mathbf{x})$. 'Ordinary' Kriging—to which we limit ourselves—further assumes a *stationary covariance process* for $Y(\mathbf{x})$ in (1); i.e., the expected values $\mu(\mathbf{x})$ are a constant μ and the covariances of $Y(\mathbf{x}+\mathbf{h})$ and $Y(\mathbf{x})$ depend only on the Euclidean distance (lag) $\|\mathbf{h}\| = \|(\mathbf{x}+\mathbf{h}) - (\mathbf{x})\|$. (The assumption $\mu(\mathbf{x}) = \mu$ is standard in Ordinary Kriging, and does not imply a flat response surface; see Sacks et al. 1989.)

The Kriging *predictor* for the unobserved (non-simulated) input (say) $\mathbf{x}_0$—denoted by $\hat{Y}(\mathbf{x}_0)$— is a weighted linear combination of all the *n* observed outputs:

$$\hat{Y}(\mathbf{x}_0) = \sum_{i=1}^{n} \lambda_i \cdot Y(\mathbf{x}_i) = \boldsymbol{\lambda}' \cdot \mathbf{Y} \tag{2}$$

with $\sum_{i=1}^{n} \lambda_i = 1$, $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)'$ and $\mathbf{Y} = (y_1, \ldots, y_n)'$. To select these weights, Kriging derives the Best Linear Unbiased Predictor (BLUP), which (by definition) minimizes the MSE of the predictor:

$$\min_{\lambda}\left\{\mathrm{MSE}\left(\hat{Y}(\mathbf{x}_0)\right)\right\} = \min_{\lambda}\left\{E\left(Y(\mathbf{x}_0) - \hat{Y}(\mathbf{x}_0)\right)^2\right\}. \tag{3}$$

Obviously, this solution depends on the output's covariances. It can be proven that the optimal weights in (2) resulting from (3) are

$$\boldsymbol{\lambda}' = \left(\boldsymbol{\gamma} + \mathbf{1}\frac{1 - \mathbf{1}'\boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma}}{\mathbf{1}'\boldsymbol{\Gamma}^{-1}\mathbf{1}}\right)'\boldsymbol{\Gamma}^{-1} \tag{4}$$

with the following symbols:

$\boldsymbol{\gamma}$ is the vector of covariances between the outputs at the input to be predicted and at the inputs already observed, so $\boldsymbol{\gamma} = (\gamma(\mathbf{x}_0 - \mathbf{x}_1), \ldots, \gamma(\mathbf{x}_0 - \mathbf{x}_n))'$;

$\mathbf{1} = (1, \ldots, 1)'$ is the vector with *n* ones;

$\boldsymbol{\Gamma}$ is the $n \times n$ matrix whose element (*i*, *j*) is the (co)variance at the inputs already observed $\gamma(\mathbf{x}_i - \mathbf{x}_j)$ with $i, j = 1, \ldots, n$.

Note that the weights in (4) vary with $\mathbf{x}_0$ (input to be predicted), whereas regression analysis uses the same estimated metamodel for all inputs $\mathbf{x}$.

Note further that the literature on (deterministic) simulation speaks of covariances and corresponding correlations, whereas the geostatistics literature speaks of the *variogram*, defined as $2\gamma(\mathbf{h}) = var(Y(\mathbf{x}+\mathbf{h}) - Y(\mathbf{x}))$. Since we shall use the Matlab Kriging toolbox DACE—made available free of charge by Lophaven, Nielsen, and Søndergaard (2002)—we avoid the term variogram. (Recent alternative free software is made available via http://www.stat.ohio-state.edu/~comp_exp/; see Santner, Williams, and Notz 2003.)

We emphasize that in practice the covariances $\boldsymbol{\gamma}$ and $\boldsymbol{\Gamma}$ in (4) are unknown so they must be *estimated*. The classical estimator for $\gamma(\mathbf{h})$ is $\hat{\gamma}(\boldsymbol{h}) = \sum_{N(\boldsymbol{h})} (Y(\boldsymbol{x}_i) - Y(\boldsymbol{x}_j))^2 / (2N(\boldsymbol{h}))$, where $|N(\mathbf{h})|$ denotes the number of distinct pairs in $N(\mathbf{h}) = \{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i - \mathbf{x}_j = \mathbf{h}\}$. Consequently, the weights in (4) become random variables (say) $\hat{\boldsymbol{\lambda}}$. These weights make the Kriging predictor resulting from (2) *non-linear*. This characteristic is often neglected in the Kriging literature. In general, non-linear functions of random variables are hard to analyze—a simple computer-intensive solution is bootstrapping; see Efron and Tibshirani (1993).

Ignoring the randomness of the estimated optimal weights $\hat{\boldsymbol{\lambda}}$ tends to *underestimate* the true variance of the Kriging predictor. For example, in the bivariate normal case this follows from the formula for the conditional variance, namely $\text{var}(Y \mid X) = (1-\rho^2) \cdot \text{var}(Y)$; see, for example, Kreyszig (1970, p. 343). To tackle this problem, Cressie (1993, p. 146) proposes *cross-validation*. Cross-validation is also used by Kleijnen and Van Beers (2004) for deterministic simulation. For deterministic simulation, Den Hertog, Kleijnen, and Siem (2005) apply parametric bootstrapping—assuming normally distributed prediction errors—and find that ignoring the randomness of the Kriging weights leads to serious errors. Because random simulation may have non-normal outputs (for example, queueing simulations have distributions with heavy right-hand tails), we use distribution-free bootstrapping—as we shall explain in Section 4.

5.3 DOE and Kriging

By definition, an experimental *design* is a set of *n* combinations of *k* factor values. These combinations are usually bounded by 'box' constraints: $a_j \le x_j \le b_j$ with $a_j, b_j \in R$ and

$j = 1, \ldots, k$. The set of all feasible combinations is called the *experimental region* (say) *H*. We suppose that *H* is a *k*-dimensional unit cube, after rescaling the original rectangular area (see Section 1).

Our goal is to find the 'best' design for Kriging predictions within *H*; the Kriging literature proposed several criteria (see Sacks et al. 1989, p. 414). Most of these criteria are based on the predictor's MSE. Most progress has been made for the IMSE (see Bates et al. 1996):

$$IMSE = \int_H \mathrm{MSE}\left(\hat{Y}(\mathbf{x})\right)\phi(\mathbf{x})d\mathbf{x} \tag{5}$$

where MSE follows from (3), and $\phi(\mathbf{x})$ is a given weight function—usually assumed to be a constant.

To evaluate a design, Sacks et al. (1989, p. 416) compare the predictions with the known output values of a *test set* consisting of (say) *N* inputs. Assuming a constant $\phi(\boldsymbol{x})$ in (5), the IMSE can then be estimated by the Empirical IMSE (EIMSE):

$$EIMSE = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i(\mathbf{x}) - y_i(\mathbf{x})\right)^2 . \tag{6}$$

Besides this EIMSE, we will also study the *maximum* MSE; that is, we also consider risk-averse users (also see Van Groenigen, 2000). So IMSE—defined in (5)—is replaced by

$$MaxMSE = \max_{\mathbf{x}\in H}\left\{MSE\left(\hat{Y}(\mathbf{x})\right)\right\} \tag{7}$$

and EIMSE in (6) by

$$EMaxIMSE = \max_{i\in\{1,\ldots,m\}}\left\{\left(\hat{y}_i(\mathbf{x}) - y_i(\mathbf{x})\right)^2\right\}. \tag{8}$$

5.4 Sequential DOE

We devise the following sequential DOE procedure with eight steps, which we illustrate through the M/M/1 model with experimental region $H = \{\rho : 0.1 \le \rho \le 0.9\}$ where ρ denotes the traffic rate.

Step 1. We start with a small *pilot design* with (say) n_0 input combinations; for example, $n_0 = 5$. We select the specific n_0 values such that they are equally spread over the experimental region. There are various 'space filling' designs; for example, LHS designs. In the first example in Section 5—namely the M/M/1—we use a *maximin* design, which (by definition) maximizes the minimum distance between any two points of the design; see Koehler and Owen (1996, p. 288). So in this example, we select the traffic rates $x_i \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ ($i = 1, \ldots, 5$).

Step 2: For each input value x_i, we initially generate (say) m_0 IID *replicates*—because bootstrapping requires IID observations; see Efron and Tibshirani (1993). To obtain IID observations in our M/M/1 simulation example, we apply *renewal* (regenerative) analysis (see, for example, Kleijnen and Van Groenendaal 1992, and Law and Kelton 2000). As 'the' renewal state, we choose the idle (empty) state. We therefore start the simulation run in the empty state—for each traffic rate x_i. Next we observe m_0 cycles—each with (random) *cycle lengths* (say) L_i (the higher x_i, the higher L_i tends to be). Besides the m_0 cycle lengths $L_{i;\,j}$ ($j = 1, \ldots, m_0$) per traffic rate x_i, we observe the sum of the waiting times over that cycle:

$$sw_{i;\,j} = \sum_{t=1}^{L_{i;\,j}} w_{i;\,j;\,t} \quad (i = 1, \ldots, n_0;\ j = 1, \ldots, m_0). \tag{9}$$

To reduce the variance when comparing the (random) outputs for different inputs (i.e., to improve the signal/noise ratio), we use *common random numbers* (CRN). This is a popular variance reduction technique (VRT). It is well known that—in M/M/1 simulation—the variance decreases substantially if the PRN (say) r_t are manipulated as follows: successive PRN are used alternatively to simulate the arrival time (say) a and the service time s; in other words,

$a_t = -\ln r_{2t-1} E(a)$ and $s_t = -\ln r_{2t} E(s)$ $(t = 1, 2, \ldots)$. The correlation coefficients for the average waiting times of two neighboring traffic rates turn out to be very high, namely roughly 0.99.

To generate the PRN, we use the Matlab command 'rand'. To initialize the PRN, we set the Matlab generator (rather arbitrarily) to its initial state $s_0 = 0$. The Matlab web site further states: 'The uniform random number generator in MATLAB 5 (and above) uses a lagged Fibonacci generator, with a cache of 32 floating point numbers, combined with a shift register random integer generator. The integer generator uses shifts and exclusive OR's.'; see (http://www.mathworks.com/support/solutions/data/8542.shtml) and also Moler (1995).

For further details on CRN, VRT, and PRN we refer to Law and Kelton (2000).

Step 3. Based on these m_0 bivariate IID outputs ($L_{i;\,j}, sw_{i;\,j}$) ($j = 1, \ldots, m_0$) per input value x_i, we estimate the mean waiting times through

$$\bar{y}_i(m_0) = \frac{\sum_{j=1}^{m_0} sw_{i;\,j}}{\sum_{j=1}^{m_0} L_{i;\,j}} \,. \tag{10}$$

This *ratio estimator* is consistent; for references see again Kleijnen and Van Groenendaal (1992) and Law and Kelton (2000). We do not try to improve the small-sample performance of this estimator (for example, through jackknifing—which is closely related to bootstrapping), because this estimator suffices for our Kriging metamodel.

To estimate the *precision* of the estimate defined in (10), we use the following probability statement that holds asymptotically per input value x_i:

$$P\left\{\bar{y}_i(m_0) - t_{m_0-1;\,1-\alpha/2} \cdot \frac{\hat{\sigma}_i/\sqrt{m_0}}{\bar{L}_i} \le E(w_i) \le \bar{y}_i(m_0) + t_{m_0-1;\,1-\alpha/2} \cdot \frac{\hat{\sigma}_i/\sqrt{m_0}}{\bar{L}_i}\right\} = 1-\alpha \tag{11}$$

where $\hat{\sigma}_i^{\ 2} = \hat{\mathrm{var}}(sw_i) + \bar{y}_i^{\ 2} \cdot \hat{\mathrm{var}}(L_i) - 2\bar{y}_i \cdot \hat{\mathrm{cov}}(sw_i, L_i)$ and $\bar{L}_i = \sum_{j=1}^{m_0} L_{i;j} / m_0$; again see Kleijnen and Van Groenendaal (1992). Note that this interval does not have an asymptotic *joint* (or experimentwise) probability $(1-\alpha)$ over all simulated input values.

Next, we add replicates one-at-a-time—*sequential sampling*—until the desired half-width of the interval in (11) has reduced to a prefixed relative error (say) δ ; for example, $\delta = 0.15$ (again see Kleijnen and Van Groenendaal 1992 and Law and Kelton 2000). We denote the final number of replicates per input x_i by m_i . This gives the average output $\bar{y}_i(m_i)$ per input x_i based on m_i replicates; see (10) with m_0 replaced by m_i .

Step 4. Based on these n_0 average outputs $\bar{y}_i(m_i)$ for the n_0 inputs x_i, we compute the *Kriging predictors* for the expected outputs of a new set of (say) n^c *candidate* input values x_g^c ($g = 1, \ldots, n^c$). We again select these candidates in a *space-filling* way; in the M/M/1 example, we choose the candidate inputs halfway between two old neighboring inputs so we avoid extrapolation: $x_g^c = (x_g + x_{g+1})/2$ (with $g = 1, \ldots, n_0 - 1$).

By definition, the Kriging predictor is a weighted linear combination of all outputs already observed; see (2). So now Kriging weights the n_0 values already observed in steps 1 through 3:

$$\hat{y}(\mathbf{x}_g^c) = \sum_{i=1}^{n_0} \lambda_i \cdot \bar{y}(\mathbf{x}_i) \tag{12}$$

with $\sum_{i=1}^{n_0} \lambda_i = 1$. To estimate the weights λ_i in (12), Kriging uses the old data set $(x_i, \bar{y}_i(m_i))$ ($i = 1, \ldots, n_0$). To estimate the variance of this non-linear predictor, we use bootstrapping—as follows.

Step 5. Per input x_i, we *bootstrap* the m_i bivariate IID outputs ($L_{i;j}$, $sw_{i;j}$); i.e., we resample—with replacement—the outputs resulting from steps 1 through 3. We denote these bootstrap observations by the superscript * (as is traditional in the bootstrap literature):

$$\{(sw_{i;1}^*, L_{i;1}^*), \ldots, (sw_{i;m_i}^*, L_{i;m_i}^*)\}. \tag{13}$$

Using these bootstrapped observations and (10), we compute the bootstrap averages:

$$\overline{y}_i^*(m_i) = \frac{\sum_{j=1}^{m_i} sw_{i;\,j}^*}{\sum_{j=1}^{m_i} L_{i;\,j}^*}\,. \tag{14}$$

Using the bootstrapped I/O data $(x_i, \overline{y}_i^*(m_i))$ $(i = 1, \ldots, n_0)$ and (12), we compute the bootstrapped Kriging predictor:

$$\hat{y}^*(\mathbf{x}_g^c) = \sum_{i=1}^{n_0} \lambda_i^* \cdot \overline{y}^*(\mathbf{x}_i)\,. \tag{15}$$

We again estimate the bootstrap weights λ_i^* in (15) through the Matlab Toolbox DACE; see Section 2.

Note that DACE aims to obtain the maximum likelihood estimator (MLE) of the Kriging weights λ_i^* in (15). For the numerical search that leads to this MLE, DACE uses starting values. As starting values, we use the MLE for λ_i based on the original I/O data in (12).

Step 6. The resampling per input x_i in step 5 is repeated (say) B times (this B is called the bootstrap sample size). Hence, (13) through (15) give $\hat{y}_b^*(\mathbf{x}_g^c)$ with $b = 1, \ldots, B$.

For each of the n^c candidate inputs x_g^c, we compute the bootstrap variance of the Kriging predictor $\hat{y}_g^{c*}$ at x_g^c:

$$\hat{\mathrm{var}}(\hat{y}_g^{c*}) = \frac{1}{B-1}\sum_{b=1}^{B}(\hat{y}_{g;\,b}^{c*} - \overline{\hat{y}}_g^{c*})^2 \tag{16}$$

where $\hat{y}_{g;\,b}^{c*}$ is the predicted value at candidate input x_g^c based on the bootstrapped I/O data $(x_i, \overline{y}_{i;b}^*(m_i))$ $(i = 1, \ldots, n_0)$ and $\overline{\hat{y}}_g^{c*} = \sum_{b=1}^{B} \hat{y}_{g;\,b}^{c*} / B$.

Step 7. We determine which candidate input has the *largest* bootstrap prediction variance (16):

$$v = \arg\left(\max_{g \in \{1, \ldots, n^c\}} \left\{ \widehat{\mathrm{var}}(\hat{y}_g^{c*}) \right\} \right), \tag{17}$$

and we add this 'winning' input x_v^c to the old design.

Now, we run the simulation model with this input x_v^c—until we have m_0 replicates for this input. We still apply CRN (so we initialize the PRN with the seed s_0). Furthermore, we again start with the empty system as the renewal state. We continue the simulation until the confidence interval reaches the threshold δ ; see (11).

Step 8. We *repeat* the steps 4 through 7—until we have reached a stopping criterion. In other words, we bootstrap the old I/O set augmented with the candidate selected in step 7. We select a new set of candidates. For these candidates, we compute the Kriging predictors and their bootstrap variances. Alternative stopping criteria may be: (i) the computer budget has been exhausted, (ii) the project has reached its deadline, (iii) the precision of the Kriging metamodel is acceptable.

We observe that adding one point at a time—as we do in our sequential DOE—is not necessarily optimal. However, it is a simple—albeit myopic—heuristic; also see Banjevic and Switzer (2002), who refer to Ferri and Piccioni (1992).

5.5 Two examples

We test our customized sequential design (CSD) through two classic academic simulation models, namely the M/M/1 model and an (*s*, *S*) model.

5.5.1 M/M/1 model

An M/M/1 has as true I/O function the hyperbole

$$y = \frac{x}{1-x} \text{ with } 0 < x < 1 \tag{18}$$

where y denotes the expected steady-state waiting time assuming a unit service rate, and x denotes the traffic rate .

We apply the procedure described in section 4, selecting the following parameters.

Step 1: We select a pilot design of size $n_0 = 5$.

Step 2: We obtain $m_0 = 10$ replicates to get initial estimates of the variances; we select as the initial PRN seed $s_0 = 0$.

Step 3: We experiment with two values for the precision, namely $\delta = 0.05$ and $\delta = 0.15$, and two values for the type-I error rate, namely $\alpha = 0.01$ and 0.05—so (11) gives four confidence intervals. For higher traffic rates (say, $x > 0.7$), the numbers of cycles and the cycle lengths may be very large. To limit computer time, we limit the number of cycles ($L_{i;\,j}$) to 1000. This limit preserves the renewal property, but may decrease the precision δ.

Step 6: We experiment with the bootstrap sample sizes: $B = 50$ and $B = 100$.

Step 8: We experiment with a stopping criterion that specifies that the total design size is either $n = 15$ or $n = 100$.

Figure 1 displays simulation results for both our design and a LHS design. This figure is based on the confidence intervals in (11) with $\alpha = 0.05$ and $\delta = 0.15$. The bootstrap sample size is only $B = 50$. The stopping criterion is that $n = 15$ traffic rates have been simulated. This figure corresponds with one scenario (labeled 7) of the eight scenarios in our experiment; see Table 1 below. LHS turns out to simulate fewer 'challenging' inputs; i.e., high traffic rates.

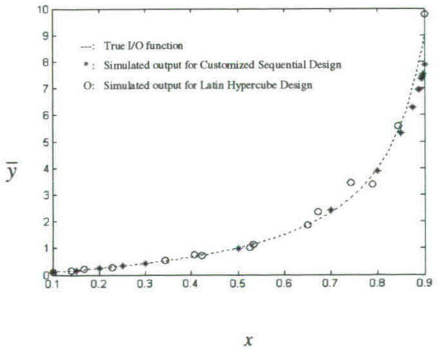


Figure 1: Two designs for M/M/1 with 15 traffic rates x and average simulation outputs $\bar{y}$

To evaluate our procedure, we use a *test set* with $N = 32$ equidistant traffic rates, namely $\{0.1125, 0.1375, \ldots, 0.8875\}$ (Sacks et al. 1989 also use test sets to evaluate their procedure). We compare the Kriging predictions of the two designs with the 'true' outputs of the test set, computed from (18). (The two designs may contain some members of the test set, but we ignore this phenomenon.) Figure 2 illustrates the 32 predictions for replicate 1 of scenario 1.

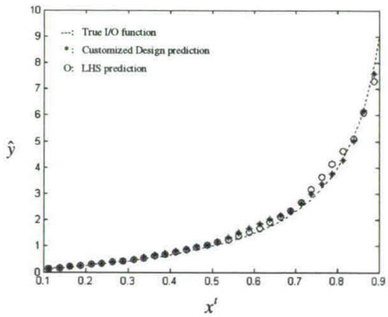


Figure 2: Predictions $\hat{y}$ for the test set for M/M/1, for two designs in replicate 1 of scenario 1

To compare the predictions of our design and LHS, we might use the *EIMSE* criterion, defined in (6). However, the final numbers of replicates in the two designs may differ, so we calculate the *corrected EIMSE*, denoted by *e* later on:

$$e = CEIMSE = C \times \frac{1}{n_t} \sum_{i=1}^{n_t} \left(\hat{y}(x_i^t) - y(x_i^t)\right)^2 , \tag{19}$$

where *C* is the ratio of the total number of replicates in the LHS design and in our design, n_t is the number of I/O combinations in the test set (so $n_t = 32$), and x_i^t is the i^{th} input of the test set.

We compute this criterion for eight scenarios; i.e., eight combinations of values of the type-I error rate α, the relative error δ, the bootstrap sample size *B*, and the final design size *n*. These scenarios are specified through a 2^{k-p} design with $k = 4$ and $p = 1$. This design is expressed in standardized values in Table 1a (see Kleijnen and Van Groenendaal 1992); note that all columns are orthogonal. The original values are displayed in Table 1b.

Table 1a: A 2^{4-1} design expressed in standardized factor values

factor	α	δ	B	n
scenario	1	2	3	$4 = 1\cdot 2\cdot 3$
1	-	-	-	-
2	-	-	+	+
3	-	+	-	+
4	+	-	-	+
5	-	+	+	-
6	+	-	+	-
7	+	+	-	-
8	+	+	+	+

Table 1b: Eight scenarios or combinations of type-I error rate α, relative error δ, bootstrap sample size *B*, and final design size *n*

scenario	α	δ	B	n
1	0.01	0.05	50	10
2	0.01	0.05	100	50
3	0.01	0.15	50	50
4	0.05	0.05	50	50
5	0.01	0.15	100	10
6	0.05	0.05	100	10
7	0.05	0.15	50	10
8	0.05	0.15	100	50

To decrease the randomness of *CEIMSE* in (19), we replicate each scenario in Table 1 $R = 5$ times. To ensure that the PRN streams do not overlap, we start Matlab's PRN generator in the initial state $s_0 = 0$ (using the command RAND('state', 0)) in the first replication of each scenario. Next we save the generator's state of the scenario that requires the largest number of simulation runs; we use that state as the initial state for each of the eight scenarios in the next replication, and so on. Table 2a shows the $R = 5$ CEIMSEs per scenario, denoted by e_r $(r = 1, \ldots, R)$, for the Customized Sequential Design; Table 2b shows e_r for LHS.

Table 2a: CEIMSE e_r for Customized Sequential Designs
in 8 scenarios replicated 5 times, computed from test set with 32 values

scenario	e_1	e_2	e_3	e_4	e_5
1	0.015026	0.028725	0.005305	0.15052	0.11056
2	0.010669	0.027213	0.010518	0.17480	0.12000
3	0.011028	0.027209	0.010513	0.17480	0.11951
4	0.010669	0.028481	0.010518	0.17636	0.11762
5	0.014915	0.029568	0.005417	0.15051	0.11044
6	0.015026	0.028725	0.005305	0.15052	0.11056
7	0.014645	0.028676	0.004749	0.12363	0.10993
8	0.011019	0.027314	0.010347	0.17486	0.12167

Table 2b: CEIMSE e_r for LHS designs
in 8 scenarios replicated 5 times, computed from test set with 32 values

scenario	e_1	e_2	e_3	e_4	e_5
1	0.045243	0.036466	0.024428	0.15382	0.126451
2	0.003626	0.026059	0.008152	0.17114	0.12551
3	0.003649	0.025891	0.007919	0.17000	0.12209
4	0.003626	0.026059	0.008152	0.17114	0.12551
5	0.041051	0.035814	0.023546	0.15164	0.124033
6	0.045243	0.036466	0.024428	0.15382	0.126451
7	0.037169	0.033886	0.018249	0.12648	0.112403
8	0.002993	0.024671	0.007233	0.14924	0.10047

We analyze the results in Table 2 as follows. Comparing Tables 2a and 2b shows that our designs do not have smaller CEIMSE than LHS designs, in *all* cases (scenarios and replicates). More precisely, our designs give better results only if the design size *n* is 'small'; see the scenarios 1, 5, 6, and 7. But it is exactly these cases that we are interested in, since (as we stated in Section 1) we focus on 'expensive' simulations, which imply that big design sizes are infeasible. So, we compute the differences

$$d_{i;r} = e_{i;r;LHS} - e_{i;r;CSD} \text{ with } i = 1, \dots, 8; \quad r = 1, \dots, 5. \tag{20}$$

Lumping all scenarios together, the Student *t* test does not give significant differences at a type-I error rate of 5% (the variation of the differences $d_{i;r}$ is large). However, Figure 3 suggests that each of the four scenarios with small *n* (design size) gives significantly positive differences. We therefore investigate which factors explain the performance of our design relative to LHS, as follows.

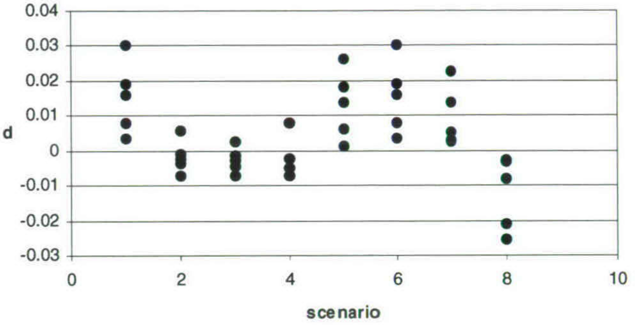


Figure 3: Differences $d_{i;r} = e_{i;r;LHS} - e_{i;r;CSD}$ for scenario $i = 1, \ldots, 8$ and replicate $r = 1, \ldots, 5$

Remember that we have the $k = 4$ factors corresponding with α , δ , *B*, and *n*. So we estimate the first-order polynomial, which has the main effects β_j :

$$d_{ir} = \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \varepsilon_{ir} \; . \tag{21}$$

We wish to account for variance heterogeneity: $\text{var}(\varepsilon_i) \neq \sigma^2$. Moreover we use CRN, so d_{ir} and $d_{i'r}$ (*i*, *i'* = 1, …, 8) are not independent. Therefore we compute the OLS estimator of the parameters in (21) per replication:

$$\hat{\boldsymbol{\beta}}_r = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{d}_r \tag{22}$$

where $\boldsymbol{X}$ is the 8×5 matrix following from (21) and Table 1a. This gives the average OLS estimator based on all $R = 5$ replications:

$$\bar{\hat{\boldsymbol{\beta}}} = \frac{1}{R}\sum_{r=1}^{R} \hat{\boldsymbol{\beta}}_r \, . \tag{23}$$

Hence the standard error for the j^{th} main effect is

$$s(\bar{\hat{\beta}}_j) = \frac{s(\hat{\beta}_j)}{\sqrt{R}} = \frac{\sqrt{\sum_{r=1}^{R}(\hat{\beta}_{j;r} - \bar{\hat{\beta}}_j)^2 \Big/ (R-1)}}{\sqrt{R}} \quad , \tag{24}$$

so the Student statistic with $\nu = R - 1$ degrees of freedom is

$$t_{j;\nu} = \frac{\bar{\hat{\beta}}_j - \beta_j}{s(\bar{\hat{\beta}}_j)} \, . \tag{25}$$

This statistic assumes normality, which probably holds because the Central Limit Theorem may be applied.

The classic null-hypothesis is that $\beta_j = 0$ ($j = 1, \dots, 4$) in (21). We display the corresponding t-statistics defined by (25) in Table 3 for three values of the type-I error rate, namely 0.10, 0.05, and 0.01.

Table 3: Significance of estimated main effects $\hat{\beta}_j$

	t-statistic	two-sided significance level		
	$t_{j;\nu}$	$\alpha = 0.10$	$\alpha = 0.05$	$\alpha = 0.01$
β_1	-2.2962201	significant	significant	not signif.
β_2	-2.4742393	significant	significant	not signif.
β_3	-1.079914	not signif.	not signif.	not signif.
β_4	-3.8774691	significant	significant	significant

Table 3 shows that the design size n (factor 4) has a significant negative effect on the difference d (for any of the three type-I error rates); i.e., the advantage of our design becomes smaller as the design size n increases. Further, the bootstrap sample size B (factor 3) has no significant effect: our procedure uses the bootstrap only to estimate which candidate input has the largest variance of the Kriging predictor; see (17). So in practice the smaller size, $B = 50$, may be used. (Most bootstrap applications require the estimation of the whole distribution function, so B is much higher than 50; for example, $B = 1000$.) Changes in α and δ (factors 1 and 2) affect the number of replicates, but this effect is incorporated in CEIMSE via the factor C; see (19).

Risk-averse users may be guided by EMaxIMSE, defined in (8). Again, our designs outperform LHS designs for the smaller design sizes n. Table 4a shows the five EMaxIMSE values for scenario i, denoted by $e^{\max}_{i;r}$ for our design, and Table 4b shows the analogous values for LHS; Figure 4 shows the differences, $d^{\max}_{i;r} = e^{\max}_{i;r;LHS} - e^{\max}_{i;r;CSD}$.

Table 4a: EMaxIMSE $e^{\max}_{r}$ for Customized Sequential Designs
in 8 scenarios replicated 5 times, computed from test set with 32 values

	EMaxIMSE e_i for Customize Sequential Designs				
scenario	e_1	e_2	e_3	e_4	e_5
1	0.068502	0.52477	0.024872	0.22247	1.0878
2	0.047377	0.52477	0.1374	0.22247	1.2755
3	0.047378	0.52477	0.1374	0.22247	1.2755
4	0.047377	0.52477	0.1374	0.22247	1.2755
5	0.068502	0.52477	0.024872	0.22247	1.0878
6	0.068502	0.52477	0.024872	0.22247	1.0878
7	0.068502	0.52477	0.024872	0.22247	1.0878
8	0.049059	0.52477	0.1374	0.22247	1.2755

Table 4b: EMaxIMSE $e^{\max}_{r}$ for LHS designs
in 8 scenarios replicated 5 times, computed from test set with 32 values

	EMaxIMSE e_i for LHS				
scenario	e_1	e_2	e_3	e_4	e_5
1	0.57114	0.34262	0.1845	0.2689	1.80351
2	0.023245	0.52006	0.11484	0.27878	1.183
3	0.023245	0.52006	0.11484	0.27878	1.183
4	0.023245	0.52006	0.11484	0.27878	1.183
5	0.57114	0.34262	0.1845	0.2689	1.80351
6	0.57114	0.34262	0.1845	0.2689	1.80351
7	0.57114	0.34262	0.1845	0.2689	1.80351
8	0.023245	0.52006	0.11484	0.27878	1.183

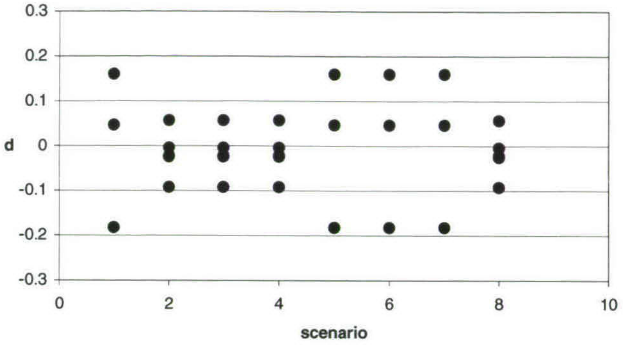


Figure 4: Differences $d_{i;r}^{\max} = e_{i;r;LHS}^{\max} - e_{i;r;CSD}^{\max}$ for scenario $i = 1, \dots, 8$ and replicate $r = 1, \dots, 5$

Note that *m (*number of required cycles) indeed increases with *x* (traffic rate). For example, for the precision requirements $\alpha = 0.05$ and $\delta = 0.15$, $x = 0.1$ requires 489 cycles, whereas $x = 0.9$ requires the maximum number of cycles, namely 1000; see Figure 5. Moreover, a cycle is likely to be longer as the traffic rate increases. For example, if $x = 0.1$ then the average cycle length is $\overline{L} = 4.8$ for $m_0 = 10$ replicates; if $x = 0.9$ then $\overline{L} = 45.9$. For a high traffic rate, the maximum number of cycles (1000) is reached, in this figure. For higher accuracy ($\delta = 0.05$) this maximum is also reached for moderate traffic rates.

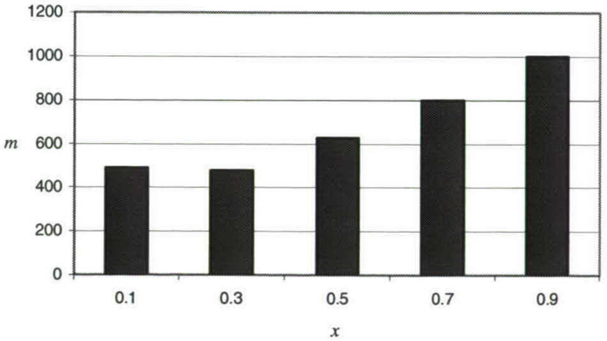


Figure 5: Number of cycles *m* per traffic rate *x* for M/M/1, given $\alpha = 0.05$ and $\delta = 0.15$

A question about our design might be: is the concentration of the simulation runs in the input range with high traffic rates caused by the high signal ($E(y)$) or the high noise ($\mathrm{var}(y)$) (both the mean and the variance of the M/M/1‘s steady-state waiting time increase with the traffic rate)? To answer this question, we run some Monte Carlo experiments inspired by the M/M/1 model. In these experiments we use the relative precision $\delta = 0.15$, the type-I error rate $\alpha = 0.05$, and the final design size $n = 15$. We use the same PRN seed for the same macro-replicate of the four experiments. We run six macro-replicates; the results across the six macro-replicates look very much alike, so—to save space—we do not display the figures for all macro-replicates; Figure 6 gives results for one macro-replicate.

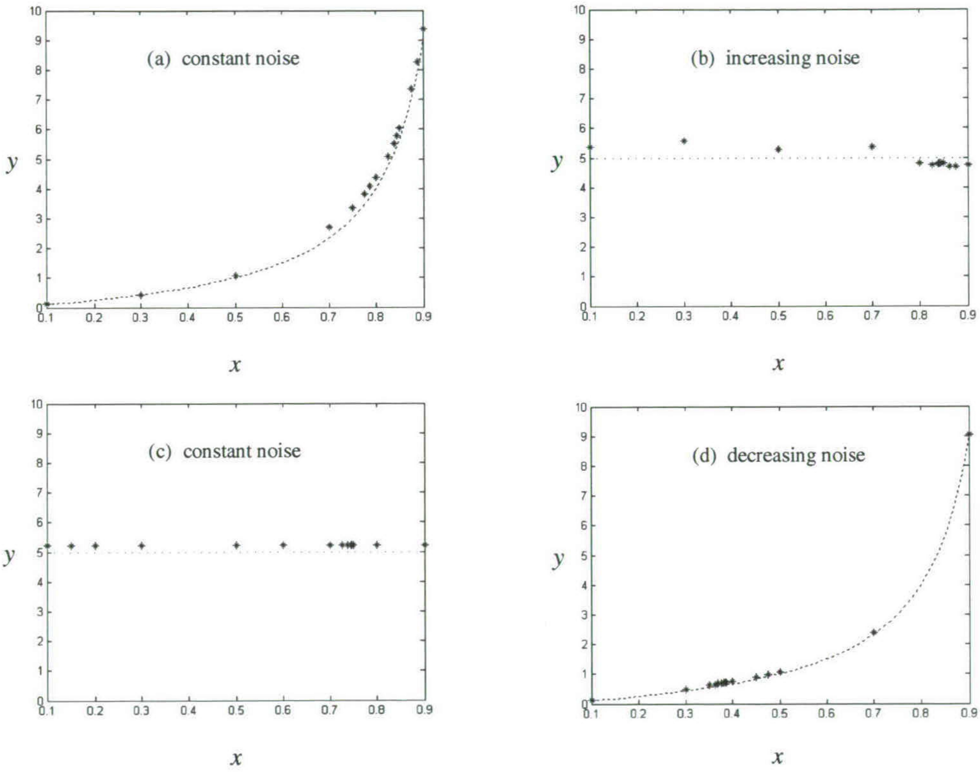


Figure 6: Monte Carlo experiments with four combinations of signal and noise functions; --- denotes signal and *** denotes I/O of Customized Sequential Design

(a) *Increasing signal and constant noise*: $y = x/(1-x) + r$ with $0.1 \le x \le 0.9$ and $r \in U(-1, 1)$; in other words, the signal follows (18), but the noise is uniformly distributed between -1 and 1, for any input value *x*. Figure 6(a) shows that our design allocates its runs to the area with rapidly changing signal—as our design did for the M/M/1 in Figure

(b) *Constant signal and increasing noise*: $y = 5 + 10rx$. Figure 6(b) shows that our design again allocates its runs to the high input values with high noise.

(c) *Constant signal and constant noise*: $y = 5 + r$. Figure 6(c) shows that now our design spreads its runs uniformly across the experimental area.

(d) *Increasing signal and decreasing noise*: $y = x/(1-x) + r/(10x)$. Figure 6(d) shows that now our design allocates most of its runs to the middle of the experimental area. Our explanation is that the increasing signal pulls the runs to the high input values, whereas the decreasing noise pulls them to the low values—so that the net result is a 'compromise'.

5.5.2 (*s*, *S*) inventory model

In an (*s*, *S*) model (with $s < S$) with random demand *D*, the inventory *I* is replenished to the order up-to level *S* whenever the inventory decreases to a value smaller than the reorder level *s*; i.e., the order quantity *Q* is

$$Q = \begin{cases} S - I & \text{if } I < s \\ 0 & \text{if } I \ge s\,. \end{cases}$$

There are several variations on this basic model, but we simulate Law and Kelton (2000, p. 60, 651)'s example 12.9—which has the following features. Times between demands are IID exponential random variables with a mean of 0.1 month. If a demand arrives, its size is given by the probability function

D	1	2	3	4
$\Pr\{D\}$	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

The inventory is reviewed at the beginning of each month. Law and Kelton define an auxiliary variable $d = S - s$ to estimate the optimal values for *s* and *S*; the (re)order quantity, however, is not a fixed quantity (the order quantity *Q* varies with the actual 'inventory position', defined as stock on hand, minus customer backorders, plus outstanding supplier orders; see Bashyam and Fu (1998)). The lead-time of an order is uniformly distributed between 0.5 and 1 month. Demand is satisfied immediately if the inventory level *I* is at least as large as the demand size *D*. Otherwise, the demand is—possibly partly—backlogged and delivered as soon as the inventory is replenished. The backlog costs are $5 per month per item backlogged. Holding costs per item per month are $1. Ordering costs consist of a setup cost of $32 per order plus incremental costs of $3 per item.

Law and Kelton simulate the system for 120 months, starting with an initial inventory $I(0) = 60$; i.e., this simulation model is *terminating* (example 1 estimates a steady-state mean of an M/M/1). Law and Kelton obtain five replicates for each of the 36 combinations formed by $s = 0, 20, 40, 60, 80, 100$ and $d = 0, 20, 40, 60, 80, 100$. Based on these 180 I/O data, they fit the following *second-order polynomial* regression (meta) model for the average monthly total costs called *R*:

$$\hat{R}(s, d) = 188.51 - 1.49s - 1.24d + 0.014sd + 0.007s^2 + 0.010d^2. \qquad (26)$$

They compare this model's predictions with the 'true' $E(R)$ estimated from 10 replicates for each of 420 new and old combinations formed by $s = 0, 5, 10, \ldots, 100$, and $d = 5, 10, 15, \ldots, 100$.

We, however, replace (26) by a Kriging model, fitted to the same I/O data (implying 36 average outputs), and compare our Kriging predictions with the 'true' outputs. We find that our Kriging model gives more accurate predictions than the regression model (26); see the Appendix for details.

Next, we change the design from Law and Kelton's *grid* (with 16 combinations of the two inputs *s* and *d* with $(s, d) \in [20, 80] \times [20, 80]$) into our design (with the same final design size, namely 16); see Figure 7.

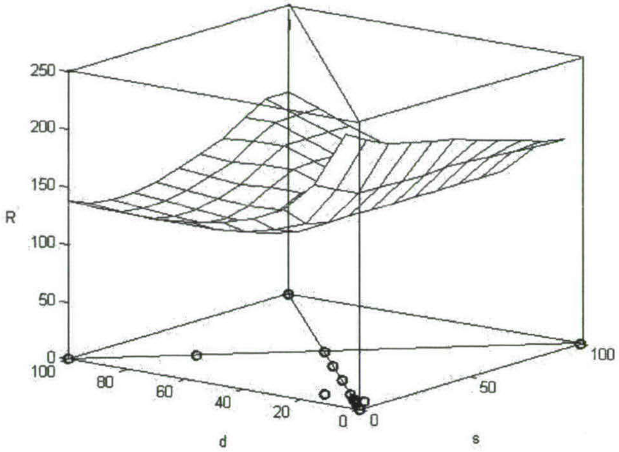


Figure 7: I/O simulation data for (*s*, *S*) inventory model with 16 scenarios denoted by ○

Like Law and Kelton, we obtain 5 replications per input combination. Next, we fit a Kriging model, and predict 81 'true' outcomes for the test set $(s, d) \in \{10, 20, \ldots, 90\} \times \{10, 20, \ldots, 90\}$ (a subset of Law and Kelton's 'true' set). Again, we calculate EIMSE and EMaxIMSE defined in (6) and (8). To reduce noise, we repeat this procedure 5 times (using non-overlapping PRN streams) for our designs and LHS. Our designs give substantial better EIMSE and EMaxIMSE; see Table 5.

Table 5: EIMSE and EMaxIMSE for CSD and LHS for (*s*, *S*) inventory simulation, based on test set with 81 true values

	CSD		LHS	
replicate	EIMSE	EMaxIMSE	EIMSE	EMaxIMSE
1	234.2	1724.4	432.9	4282.6
2	319.3	2536.9	686.9	6293.1
3	262.2	1933.3	726.4	6031.1
4	236.2	1732.9	554.5	5017.1
5	213.2	1546.5	666.5	5909.8

We conclude that in this example, our sequential design also gives more accurate Kriging predictions than LHS with a fixed design size.

5.6 Conclusions and future research

In practice, simulation often requires much computer time per run (or replicate)—so it is desirable to have an efficient experimental design for interpolation. It is well known in mathematical statistics that sequential designs are more efficient than fixed-sample designs. Our specific sequential designs add as the next input to be simulated, the input with the maximum estimated variance for the output predicted at specific candidate inputs. To obtain such predictions, we use Kriging; to estimate the variances of the Kriging predictors, we use bootstrapping. We applied this procedure to estimate (i) the expected steady-state waiting time in M/M/1 simulation, and (ii) the expected cost in terminating inventory (*s*, *S*) simulation. We compared the Kriging prediction errors of our sequential designs and those of fixed-sample LHS. Our results show that our procedure gives indeed smaller prediction errors.

In future research, (asymptotic) proofs of the performance of our procedure might be derived. More experimentation and analyses may be done to derive rules of thumb for our procedure's parameters, such as the initial design size n_0 and the initial number of replicates m_0. Our procedure may be applied to examples more complicated than the M//M/1 queueing model or the (*s*, *S*) inventory model. Stopping rules based on a measure of accuracy or precision may be investigated. Besides LHS, other designs with prefixed sizes may be explored; for example, min-max designs. Besides Ordinary Kriging, other metamodels may be used to analyze the I/O data. For example, the 'optimal' weights in Ordinary Kriging assume that the predictors equal the average outputs at the inputs already observed; dropping this constraint implies that new Kriging software must be developed. New Kriging weights may be derived, replacing the IMSE criterion by the maximum squared error criterion. Besides Kriging, other interpolation models may be used; for example, linear or nonlinear regression metamodels. We focus on sensitivity analysis; searching for the optimal input of the simulation model requires further research.

5.7 Appendix

Law & Kelton's (2000, p. 651) data set consists of 5 replicates for each of the 16 input combinations formed by $s_i \in \{20, 40, 60, 80\}$ and $d_j \in \{20, 40, 60, 80\}$ (this set is a subset of the one in the main text). Based on this input set, we find the following estimates

$\hat{\boldsymbol{\beta}} = (130.6285, -0.2630, -0.5303, 0.0088, 0.0052, 0.0038)'$, which agrees with their values up to two decimals.

As a test set (used to compare regression and Kriging metamodels), we use their 'true' I/O set, which consists of 10 replicates of each of $420 = 21 \times 20$ input combinations with $s_i \in \{0, 5, 10, \ldots, 100\}$ and $d_j \in \{5, 10, 15, \ldots, 100\}$. For the regression model we find an EIMSE of 1450.5, whereas for the Kriging model we find an EIMSE of 1200.7. So Kriging does result in a smaller EIMSE. This EIMSE, however, is still rather large, because we have to extrapolate the data outside the region $[20,80] \times [20,80]$. In general, we strongly recommend avoiding extrapolation when fitting a metamodel; indeed, in simulation it is easy to avoid extrapolation because we can select our own input combinations.

Law and Kelton also use a data set consisting of 180 I/O combinations, namely 5 replicates for each of 36 input combinations with $s_i \in \{0, 20, 40, 60, 80, 100\}$ and $d_j \in \{0, 20, 40, 60, 80, 100\}$. We use their computer program (imported from their web page http://www.mhhe.com/engcs/industrial/lawkelton/student/code.mhtml) to generate the output. Again, we fit both a second-order regression model and a Kriging model. We compare the two fitted models via the 'true' data set. For the regression model, we find an EIMSE of 152.0, whereas for the Kriging model we find an EIMSE of only 14.0 (in this case extrapolation is indeed avoided.

Acknowledgements

We thank Dick den Hertog (Tilburg University) for his comments on an earlier version, which lead to the additional Monte Carlo experiments reported in Section 5, and Ruud Brekelmans (Tilburg University) for helping us to import Law and Kelton's C-program codes into our Matlab program.

References

Banjevic, M. and P. Switzer (2002), Bayesian network designs for variance as a function of the location. *Proceedings of the 2002 JSM Conference, Section on Statistics and the Environment*, New York, NY

Bashyam, S. and M.C. Fu (1998), Optimization of (s, S) inventory systems with random lead times and a service level constraint. *Management Science*. 44, no. 12, pp. 243-256

Bates, R.A., R.J. Buck, E. Riccomagno and H.P. Wynn (1996), Experimental design and observation for large systems. *Royal Statistical Society*. 58, no. 1, pp. 77-94

Box, G.E.P., W.G. Hunter and J.S. Hunter (1978), *Statistics for experimenters: an introduction to design, data analysis and model building*. John Wiley & Sons, Inc., New York

Crary, S.B. (2002), Design of computer experiments for metamodel generation, *Analog Integrated Circuits and Signal Processing*, 32, pp. 7-16

Cressie, N.A.C. (1993), *Statistics for spatial data*. John Wiley & Sons, Inc., New York

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

Ferri, M. and M. Piccioni (1992), Optimal selection of statistical units. *Computational Statistics & Data Analysis*, 13, pp. 47-61

Ghosh, B.K. and P.K. Sen (editors), 1991, *Handbook of sequential analysis*. Marcel Dekker, Inc., New York

Den Hertog, D., J.P.C. Kleijnen, and A.Y.D. Siem (2005), The correct Kriging variance estimated by bootstrapping. *Journal of the Operational Research Society* (accepted; preprint: http://center.kub.nl/staff/kleijnen/papers.html)

Journel, A.G. and C.J. Huijbregts (1978), *Mining geostatistics*, Academic Press, London

Kleijnen, J.P.C. (1987), *Statistical tools for simulation practitioners*. Marcel Dekker, Inc., New York

Kleijnen, J.P.C. and D. Deflandre (2005), Validation of regression metamodels in simulation: Bootstrap approach. *European Journal of Operational Research* (in press)

Kleijnen, J.P.C., S.M. Sanchez, T.W. Lucas and T.M. Cioppa (2005), A user's guide to the brave new world of designing simulation experiments. *INFORMS Journal on Computing* (accepted as State-of-the-Art Review)

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*, no. 55, pp. 876-883

Kleijnen, J.P.C. and W. van Groenendaal (1992), *Simulation: a statistical* perspective. John Wiley, Chichester (England)

Koehler, J.R. and A.B. Owen (1996), Computer experiments. *Handbook of statistics*, by S. Ghosh and C.R. Rao, vol. 13, pp. 261-308

Kreyszig, E. (1970), *Introductory mathematical statistics: principles and methods.* John Wiley & Sons, Inc., New York

Law, A.M. and W.D. Kelton (2000), *Simulation modeling and analysis, third edition*, McGraw-Hill, Boston

Lophaven, S.N., H.B. Nielsen and J. Søndergaard (2002), A Matlab Kriging toolbox. *Technical report IMM-TR-2002-12*, Technical University of Denmark

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, no. 2, pp. 239-245 (reprinted in 2000: Technometrics, 42, no. 1, pp. 55-61

Moler, C. (1995), Random thoughts. *MATLAB News & Notes*, pp. 12-13

Myers, R.H. and D.C. Montgomery (2002). *Response surface methodology: process and product optimization using designed experiments; second edition.* Wiley, New York

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, no. 6, pp. 981-990

Sacks, J., W.J. Welch, T.J. Mitchell and H.P. Wynn (1989), Design and analysis of computer experiments. *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

Sasena, M.J, P. Papalambros, and P. Goovaerts (2002), Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization* 34, no.3, pp. 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, no. 12, 2001, pp. 2233-2241

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

Van Groenigen, J.W. (2000), The influence of variogram parameters on optimal sampling schemes for mapping by Kriging. *Geoderma*, no. 97, pp. 223-236

Williams, B.J., T.J. Santner, and W.I. Notz (2000), Sequential design of computer experiments to minimize integrated response functions, *Statistica Sinica*, 10, 1133-1152

Williams, B.J., T.J. Santner, and W.I. Notz (2002), Sequential design of computer experiments for constrained optimization of integrated response functions, Working Paper. Ohio State University

Chapter 6

Conclusions and future research

6.1 Conclusions

In this thesis, we studied *Sensitivity Analysis* (SA) of expensive discrete-event simulation. Running a simulation model for different input combinations may be time-consuming. Therefore, interpolation is applied. The number of necessary simulation runs may be reduced through accurate *interpolation methods* and appropriate *experimental designs* (run plans).

To realize this efficiency, we applied *Kriging interpolation* and introduced a new type of Kriging interpolation, which we named *Detrended Kriging*. We compared both the usual Ordinary Kriging and our new Detrended Kriging with classical low-order polynomial regression metamodels estimated through Ordinary Least Squares (OLS). Tests on two random models—namely a hyperbole inspired by the M/M/1 queueing model and a fourth degree polynomial, both augmented with additive noise—showed that Ordinary Kriging gives good predictions, perfectly Detrended Kriging gives the best predictions, and OLS gives the worst predictions. Obviously, these results are based on examples with a single input; in later research (discussed below), however, we found that Kriging also gives better predictions for an *(s, S)* inventory model with a two inputs. Further, we found that the intercept of the estimated linear variogram—a variogram is a basic element in Kriging—estimates the so-called *nugget effect*; this result confirmed our conjecture that the nugget effect is the variance of the simulation model's noise. Both, Ordinary Kriging and Detrended Kriging assume that the outputs have constant variances. In practice, however, this assumption is not realistic. For example, the steady-state waiting times in an M/M/1 queueing model have variances that increase with the traffic load. Therefore, we investigated the importance of this assumption. We used a hyperbole plus noise with variances changing with the input values. Our conclusion was that Kriging is not sensitive to variance heterogeneity; i.e., it is a *robust* technique.

For expensive simulation, it is important to find an efficient design for the experiments with the simulation model. Classic standard designs—such as 2^{k-p} or LHS designs—are general designs that do not account for the characteristics of the input/output (I/O) function that is implied by the simulation model at hand. As an alternative design we derived a *Customized Sequential Design* (CSD) for metamodeling in simulation. Our design is sequential, because in general it is known that sequential procedures are more 'efficient' than fixed-sample procedures; our tests confirmed that property. Moreover, our method generates a design that is specific for the given simulation model: it is customized (tailor-made). For deterministic simulation, this customization is achieved through *cross-validation* and *jackknifing*—which are two general statistical techniques. For that simulation type, our method is tested through two academic applications, namely a hyperbolic I/O function and a fourth degree polynomial. For random simulation experiments, our customization uses *bootstrapping*—which is also a general statistical technique (related to jackknifing). We tested our procedure for this simulation type through two classic Operations Research/Management Science (OR/MS) applications, namely the M/M/1 queueing model and an (*s, S*) inventory management model. Our tests showed that for both deterministic simulation and random simulation, our customized designs performed better than the classic LHS designs with the same sample size. An interesting property of our procedure is that it simulates relatively many input combinations in those sub-areas that have interesting I/O behavior.

We summarize the main contributions of our research as follows:

- Kriging metamodels give more accurate predictions than low-order polynomial regression models do,
- Customized Sequential Designs for Kriging metamodels give smaller prediction errors than standard one-shot LHS designs of the same size.

6.2 Future research

During our study, we raised several questions that we did not answer yet. Future research may concern the following three topics.

Kriging technique

Ordinary Kriging assumes a stationary covariance process. We might give up the *stationarity assumption* in case of random simulation with heterogeneous variances. We might then still assume that the Kriging predictor is a weighted linear combination of the observed outputs, and that the correlation function decreases with the distance between the input locations. We do not know whether an explicit formula for the Kriging weights may be derived; maybe a numerical search will need to be used.

Furthermore, the correlation function is often estimated through maximum likelihood estimation assuming *normality*. We estimated this correlation function without the normality assumption, using WLS. Which estimation method is better?

We focused on sensitivity analysis, not on *optimization*. Kriging for optimization in deterministic simulation has already been widely applied. Optimization in random simulation certainly deserves future research.

In our tests, we compared the performance of Kriging metamodels with classical low-order polynomial regression metamodels. Further tests may estimate the performance of Kriging compared with more sophisticated metamodels, such as *rational functions* and *neural nets*.

Customized Designs

The procedure for our customized designs starts with a space-filling pilot-sample of a rather arbitrary size. It is unclear whether this pilot-size is an efficient procedure. Further research may yield more *efficient methods* for sampling the pilot set of minimal size. Moreover, similar considerations hold for the selection of the candidate set, i.e. it cost not much computer time to expand the candidate set.

In the first stage of experimenting with random simulation models, our procedure simulates a candidate input combination a few times; in the second stage, this number is augmented until a given prediction accuracy is reached. We experimented with several numbers of simulation runs for the first stage, but we did not derive an *optimal value* to start with. Further research may tackle this issue.

We terminated our customized procedure when a given number of I/O combinations was reached, or when no relative improvement of the prediction error was found. We did not find satisfying *stopping rules* in literature, so more research on this topic might be profitable.

We demonstrated the advantage of our customized designs by applying Kriging metamodels. Further tests may show that *alternative metamodels* also may benefit from customized designs.

Furthermore, Kriging assumes a single output per input combination (a simple solution computes the Kriging predictor per output). Multivariate Kriging has already been developed in geostatistics. It is unclear whether customized designs may be developed for *multivariate* Kriging.

As mentioned above, we focused on SA; we do not know whether *optimization* may profit from customized designs. While searching for an optimum, our procedure might select more observations in sub domains centered around the true optimum. This might be a challenging topic to further research.

We demonstrated the advantage of our customized designs through several tests on classic OR/MS models. However, we did not derive theoretical proofs of the superiority of our design over classic designs.

Our customized design is a sequential method; i.e. an I/O combination is added successively to the current design. Each time the design is augmented with a new I/O combination, a new correlation function was estimated. It might be profitable to implement updating techniques for the correlation function into the Kriging program codes.

Case studies

We demonstrated and verified the results of our study through simple academic simulation models. Our results might be generalized by applying Kriging to other types of simulation models with known I/O functions, such as OR/MS models known as M/G/1 and M/M/1/K. Moreover, we did not yet apply our methods to *realistic models* with many input factors and uncertain statistical behavior. Future studies might demonstrate the performance of Kriging metamodeling for complicated OR/MS simulations.

Samenvatting

Dit proefschrift gaat over Kriging als metamodel voor computersimulaties. Het onderzoek richt zich op computerintensieve simulatie-experimenten in Operations Research/ Management Science (OR/MS); bijvoorbeeld wachttijd- en voorraadbeheersingsystemen.

Om gecompliceerde reële systemen te beschrijven worden vaak wiskundige modellen gebruikt. Die modellen beogen meer inzicht te verschaffen in de reële systemen. Modellen kunnen bijvoorbeeld informatie geven over de gevoeligheid van de uitvoer bij verandering van de invoer. Voor het oplossen van die modellen blijken analytische methoden vaak niet toereikend; numerieke methoden – zoals simulatie – moeten dan gebruikt worden. Tijdrovende simulatie-experimenten noodzaken tot interpolatie tussen berekende waarden.

Kriging is een interpolatietechniek die is ontwikkeld omstreeks 1950 in de Zuid-Afrikaanse mijnbouw door D.G. Krige. In 1989 is Kriging succesvol geïntroduceerd voor deterministische simulatie door J. Sacks, W. Welch, T. Mitchell en H. Wynn. In dit proefschrift wordt Kriging geïntroduceerd voor stochastische simulatie, met nadruk op simulatie van discrete gebeurtenissen – zoals aankomsten van klanten in een wachttijdsysteem of van orders in een voorraadsysteem.

Kriging-interpolaties voorspellen de uitvoerwaarde voor een nog niet gesimuleerde invoercombinatie. Die voorspelde waarde is een gewogen gemiddelde van de al eerder gesimuleerde uitvoerwaarden. De gewichten in die voorspelling hangen af van de 'afstanden' tussen enerzijds de invoercombinatie waarvoor een voorspelde uitvoerwaarde verlangd wordt en anderzijds de invoercombinaties waarvoor de uitvoerwaarden gesimuleerd zijn. De gewichten worden zo gekozen dat de variantie van de voorspelfout wordt geminimaliseerd. De voorspelde uitvoerwaarde is dan de 'Beste Lineaire Zuiver Schatter' van de werkelijke uitvoerwaarde. De afhankelijkheid tussen de uitkomsten en de afstanden tussen hun invoercombinaties wordt beschreven door de correlatiefunctie. De correlatiefunctie wordt op basis van de beschikbare

waarnemingen geschat volgens òf de 'kleinste kwadratenmethode' òf volgens de methode van de 'meest aannemelijke schatting'.

Dit proefschrift bevat de volledige tekst van vier artikelen die al eerder zijn gepubliceerd of zijn ingediend voor publicatie in internationale tijdschriften.

Hoofdstuk 1 geeft een toelichting op begrippen die in dit proefschrift gebruikt worden, zoals 'metamodel' en 'simulatie'. Dit hoofdstuk beschrijft verder de historie van Kriging en geeft een overzicht van de toepassingsgebieden.

Hoofdstuk 2 introduceert Kriging-interpolatie als metamodel voor stochastische simulatie. Tevens wordt een nieuw type Kriging geïntroduceerd, namelijk Detrended Kriging. Kriging wordt gedemonstreerd via twee numerieke voorbeelden: (1) een hyperbolische functie, geïnspireerd door het M/M/1 wachtrij-model, en (2) een artificieel model, namelijk een meertoppig vierdegraads-polynoom met additieve ruis. De nieuwe methode wordt getest door middel van 'kruis-validatie'. De voorspelfout van Kriging wordt vergeleken met de voorspelfout van de gebruikelijke polynomiale regressiemodellen die geschat zijn volgens de 'gewone kleinste kwadratenmethode'. De conclusie is dat Kriging betere voorspellingen geeft dan die regressiemodellen. Dit hoofdstuk toont ook aan dat het zogenaamde 'nugget effect' in Kriging inderdaad gelijk is aan de variantie van de additieve ruis.

Hoofdstuk 3 laat een klassieke veronderstelling in Kriging vervallen, namelijk de veronderstelling dat de uitvoerwaarden een constante variatie hebben. De gevolgen voor de Kriging voorspellingen worden bestudeerd in het geval de werkelijke invoer/uitvoer functie een hyperbool is met additieve ruis waarvan de variantie niet constant is, maar afhangt van de invoerwaarde. De conclusie is dat Kriging niet gevoelig is voor heterogene varianties; dat wil zeggen, Kriging is een robuuste interpolatiemethode die beter voorspelt dan polynomiale regressie.

Hoofdstuk 4 introduceert een nieuwe methode voor proefopzetten voor Kriging in deterministische simulatie. De methode is bedoeld voor tijdrovende simulaties en is daarom sequentieel, d.w.z. een initiële proefopzet wordt stap voor stap uitgebreid met steeds één invoercombinatie. De proefopzetten worden geconstrueerd met behulp van 'kruisvalidatie' en 'jackknifing'. In vergelijking met traditionele proefopzetten – zoals Latin Hypercube Sampling (LHS) – houdt de nieuwe methode rekening met de karakteristiek van de invoer/uitvoer functie van het betreffende simulatiemodel. De nieuwe methode wordt getest aan de hand van dezelfde twee academische toepassingen die ook in Hoofdstuk 2 gebruikt werden. De nieuwe methode simuleert relatief meer invoercombinaties in de 'interessante' gedeelten van de invoer/uitvoer-functie. Bovendien geeft de nieuwe methode betere voorspellingen dan traditionele LHS.

Hoofdstuk 5 breidt de methode van hoofdstuk 4 uit voor stochastische simulatie. De aanpassing aan de invoer/uitvoer-functie wordt nu verkregen door 'bootstrapping'. De methode wordt getest door een M/M/1-wachttijdmodel en een (*s*, *S*)-voorraadmodel. De nieuwe methode geeft weer betere voorspellingen dan LHS.

Hoofdstuk 6 vat de conclusies samen van de voorgaande hoofdstukken. De voor- en nadelen van Kriging worden besproken. Daarnaast worden onderwerpen voor toekomstig onderzoek voorgesteld.

Wim van Beers graduated in mathematics from the "Universiteit van Amsterdam" (UvA) in 1993. He performed his Ph.D. research in the Operations Research group of the Department of Information Systems and Management at Tilburg University, which he joined in 1999.

Many scientific disciplines use mathematical models to describe complicated real systems. Often, analytical methods are inadequate, so simulation is applied. This thesis focuses on computer intensive simulation experiments in Operations Research/Management Science. For such experiments it is necessary to apply interpolation. In this thesis, Kriging interpolation for random simulation is proposed and a novel type of Kriging—called Detrended Kriging—is developed. Kriging turns out to give better predictions in random simulation than classic low-order polynomial regression. Kriging is not sensitive to variance heterogeneity; i.e., Kriging is a robust method. Moreover, the thesis develops a novel method to select experimental designs for expensive simulation. This method is sequential, and accounts for the specific input/output function implied by the underlying simulation model. For deterministic simulation the designs are constructed through cross-validation and jackknifing, whereas for random simulation the customization is achieved through bootstrapping. The novel method simulates relatively more input combinations in the interesting parts of the input/output function, and gives better predictions than traditional Latin Hypercube Sample designs with prefixed sample sizes.