EOR 6153
26 February 2004 Disk used
ARTICLE IN PRESS
No. of Pages 14, DTD = 4.3.1

Available online at www.sciencedirect.com
SCIENCE DIRECT®

ELSEVIER

European Journal of Operational Research xxx (2004) xxx–xxx

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

An overview of the design and analysis of simulation experiments for sensitivity analysis

Jack P.C. Kleijnen *

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

Received 1 August 2003

Abstract

Sensitivity analysis may serve validation, optimization, and risk analysis of simulation models. This review surveys 'classic' and 'modern' designs for experiments with simulation models. Classic designs were developed for real, non-simulated systems in agriculture, engineering, etc. These designs assume 'a few' factors (no more than 10 factors) with only 'a few' values per factor (no more than five values). These designs are mostly incomplete factorials (e.g., fractionals). The resulting input/output (I/O) data are analyzed through polynomial metamodels, which are a type of linear regression models. Modern designs were developed for simulated systems in engineering, management science, etc. These designs allow 'many factors (more than 100), each with either a few or 'many' (more than 100) values. These designs include group screening, Latin hypercube sampling (LHS), and other 'space filling' designs. Their I/O data are analyzed through second-order polynomials for group screening, and through Kriging models for LHS.

Keywords: Simulation; Regression; Scenarios; Risk analysis; Uncertainty modelling

Contents

* Tel.: +31-13-4662029; fax: +31-13-4663069.
E-mail address: kleijnen@uvt.nl (J.P.C. Kleijnen).

0377-2217/$ - see front matter © 2004 Published by Elsevier B.V.
doi:10.1016/j.ejor.2004.02.005

1. Introduction

Once simulation analysts have programmed a simulation model, they may use it for sensitivity analysis, which in turn may serve validation, optimization, and risk (or uncertainty) analysis for finding robust solutions. In this paper, I discuss how these analyses can be guided by the statistical theory on *Design Of Experiments* (DOE).

I assume that the reader is familiar with simulation—at the level of a textbook such as Law and Kelton (2000), including their chapter 12 on 'Experimental design, sensitivity analysis, and optimization'. This assumption implies that the reader's familiarity with DOE is restricted to elementary DOE for simulation. In this article, I try to summarize that elementary DOE, and extend it.

Traditionally, experts in statistics and stochastic systems have focused on *tactical* issues in simulation; i.e., issues concerning the runlength of a steady-state simulation, the number of runs of a terminating simulation, variance reduction techniques (VRT), etc. I find it noteworthy that in the related area of *deterministic* simulation—where these tactical issues vanish—statisticians have been attracted to DOE issues; see the standard publication by Koehler and Owen (1996). Few statisticians have studied random simulations. And only some simulation analysts have focused on *strategic* issues, namely which scenarios to simulate, and how to analyze the resulting I/O data.

Note the following terminology. Statisticians speak of 'factors' with 'levels' whereas simulation analysts speak of inputs or parameters with values. Statisticians talk about design points or runs, whereas simulationists talk about scenarios.

Two textbooks on classic DOE for simulation are Kleijnen (1975, 1987). An update is Kleijnen (1998). A bird-eye's view of DOE in simulation is Kleijnen et al. (2003a), which covers a wider area than this review—without using any equations, tables, or figures; this review covers a smaller area—in more detail. An article related to this one is Kleijnen (2004), focusing on Monte Carlo experiments in mathematical statistics instead of (dynamic) simulation experiments in Operations Research.

Classic articles on DOE in simulation are Schruben and Margolin (1978) and Donohue et al. (1993). Several tutorials have appeared in the *Winter Simulation Conference Proceedings*.

Classic DOE for real, non-simulated systems was developed for agricultural experiments in the 1930s, and—since the 1950s—for experiments in engineering, psychology, etc. In those real systems, it is impractical to experiment with 'many' factors; $k = 10$ factors seems a maximum. Moreover, it is then hard to experiment with factors that have more than 'a few' values; five values per factor seems a maximum.

The remainder of this article is organized as follows. Section 2 covers the black box approach to simulation, and corresponding metamodels (especially, polynomial and Kriging models); note that 'metamodels' are also called response surfaces, emulators, etc. Section 3 starts with simple metamodels with a single factor for the M/M/1 simulation; proceeds with designs for multiple factors including Plackett–Burman designs for first-order polynomial metamodels, and concludes with screening designs for (say) hundreds of factors. Section 4 introduces Kriging metamodels, which provide exact interpolation in deterministic simulation. These metamodels often use space-filling designs, such as Latin hypercube sampling (LHS). Section 5 discusses cross-validation of the metamodel, to decide whether the metamodel is an adequate approximation of the underlying simulation model. Section 6 gives conclusions and further research topics.

2. Black boxes and metamodels

DOE treats the simulation model as a black box—not a white box. To explain the difference, I consider an example, namely an M/1/1/ simulation. A *white box* representation is

$$\bar{\boldsymbol{w}} = \frac{\sum_{i=1}^{I} \boldsymbol{w}_i}{I}, \tag{1a}$$

$$\boldsymbol{w}_i = \max(\boldsymbol{w}_{i-1} + \boldsymbol{s}_{i-1} - \boldsymbol{a}_i, 0), \tag{1b}$$

$$\boldsymbol{a}_i = -\ln(\boldsymbol{r}_{2i})/\lambda, \tag{1c}$$

$$\boldsymbol{s}_{i-1} = -\ln(\boldsymbol{r}_{2i-1})/\mu, \tag{1d}$$

$$w_1 = 0, \tag{1e}$$

with average waiting time as output in (1a); interarrival times a in (1c); service times s in (1d); pseudo-random numbers (PRN) r in (1c) and (1d); empty starting (or initial) conditions in (1e); and the well-known single-server waiting-time formula in (1b).

This white box representation may be analyzed through perturbation analysis and score function analysis in order to estimate the gradient (for local sensitivity analysis) and use that estimate for optimization; see Spall (2003). I, however, shall not follow that approach.

A *black box* representation of this M/M/1 example is

$$\bar{\boldsymbol{w}} = \boldsymbol{w}(\lambda, \mu, \boldsymbol{r}_0), \tag{2}$$

where $\boldsymbol{w}(.)$ denotes the mathematical function implicitly defined by the computer simulation program implementing (1a)–(1e); $\boldsymbol{r}_0$ denotes the seed of the PRN.

One possible *metamodel* of the black box model in (2) is a Taylor series approximation—cut off after the first-order effects of the two factors, λ and μ:

$$\boldsymbol{y} = \beta_0 + \beta_1\lambda + \beta_2\mu + \boldsymbol{e}, \tag{3}$$

where $\boldsymbol{y}$ is the metamodel predictor of the simulation output $\bar{\boldsymbol{w}}$ in (2); $\boldsymbol{\beta}' = (\beta_0, \beta_1, \beta_2)$ denotes the parameters of the metamodel in (3), and e is the noise—which includes both *lack of fit* of the metamodel and *intrinsic noise* caused by the PRN.

Besides (3), there are many alternative metamodels. For example, a simpler metamodel is

$$\boldsymbol{y} = \beta_0 + \beta_1\boldsymbol{x} + \boldsymbol{e}, \tag{4}$$

where $\boldsymbol{x}$ is the traffic rate—in queueing theory usually denoted by ρ:

$$\boldsymbol{x} = \rho = \frac{\lambda}{\mu}. \tag{5}$$

This combination of the two original factors (λ, μ) into a single factor (ρ)—inspired by queueing theory—illustrates the use of *transformations*. Another useful transformation may be a logarithmic one: replacing $\boldsymbol{y}$, μ, and λ by, $\log(\boldsymbol{y})$, $\log(\lambda)$,and $\log(\mu)$ in (3) makes the first-order polynomial approximate relative changes; i.e., the regression parameters β_1 and β_2 become elasticity coefficients.

There are many—more complex—types of metamodels. Examples are Kriging models, neural nets, radial basis functions, splines, support vector regression, and wavelets; see Clarke et al. (2003) and Antioniadis and Pham (1998). I, however, will focus on two types that have established a track record in random and deterministic simulation respectively:

- linear regression models (see Section 3)
- Kriging (see Section 4).

To estimate the parameters of whatever metamodel, the analysts must *experiment* with the simulation model; i.e., they must change the inputs (or factors) of the simulation, run the simulation, and analyze the resulting I/O data. This experimentation is the topic of the next sections.

3. Linear regression metamodels and DOE

3.1. Simplest metamodels for M/M/1 simulations

I start with the simplest metamodel, namely a first-order polynomial with a single factor; see (4). Elementary mathematics proves that—to fit such a straight line—it suffices to have two I/O observations; also see 'local area 1' in Fig. 1. It can be proven that selecting those two values as far apart as 'possible' gives the 'best' estimator of the parameters in (4). In other words, if within the local area the fitted first-order polynomial gives an error—denoted by e in (4)—that has zero mean (so the polynomial is an 'adequate' or 'valid' approximation; i.e., it shows no 'lack of fit'), then the parameter estimator is unbiased with minimum variance.

In practice, the analysts do not know over which *experimental area* a first-order polynomial is a 'valid' metamodel. This validity depends on the goals of the simulation study; see Kleijnen and Sargent (2000).

So the analysts may start with a *local* area, and simulate the two (locally) extreme input values. Let us denote these two extreme values by −1 and +1, which implies the following *standardization* (also called coding, linear transformation):

$$\boldsymbol{x} = \frac{\rho - \bar{\rho}}{(\rho_{\max} - \rho_{\min})/2}, \tag{6}$$

where $\bar{\rho} = (\rho_{\max} + \rho_{\min})/2$ denotes the average traffic rate in the (local) experiment.

The Taylor series argument implies that—as the experimental area gets bigger (see 'local area 2' in Fig. 1)—a better metamodel may be a second-order polynomial

$$\boldsymbol{y} = \beta_0 + \beta_1 \boldsymbol{x} + \beta_2 \boldsymbol{x}^2 + \boldsymbol{e}. \tag{7}$$

Obviously, estimation of the three parameters in (7) requires at least the simulation of three input values. Indeed, DOE provides designs with three values per factor (for example, 3^k designs; see Section 3). However, most publications on DOE in simulation discuss *Central Composite Designs* (CCD), which have five values per factor; see Kleijnen (1975).

I emphasize that the second-order polynomial in (7) is nonlinear in $\boldsymbol{x}$ (the regression variables), but *linear* in $\boldsymbol{\beta}$ (the parameters to be estimated). Consequently, such a polynomial metamodel is a type of *linear regression* model.

Finally, when the experimental area covers the *whole* area in which the simulation model is valid ($0 < \rho < 1$), then other *global* metamodels become relevant. For example, Kleijnen and Van Beers (2003a) find that a *Kriging* metamodel outperforms a second-order polynomial.

Note that Zeigler et al. (2000) call the experimental area the 'experimental frame'. I call it the

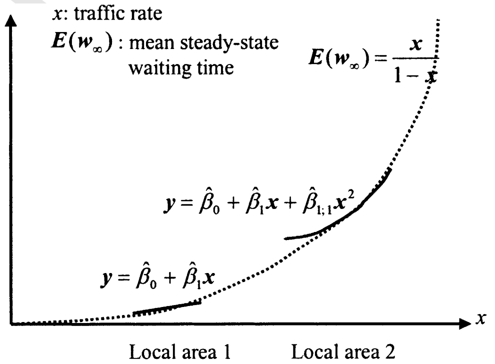

Fig. 1. M/M/1 example.

domain of admissible scenarios, given the goals of the simulation study.

I conclude that *lessons* learned from this simple M/M/1 model, are:

(i) The analysts should decide whether they want to experiment *locally or globally*.
(ii) Given that decision, they should select a specific *metamodel type*; for example, a low-order polynomial or a Kriging model.

3.2. Metamodel with multiple factors

Let us now consider a metamodel with k factors; for example, (4) implies $k = 1$, whereas (3) implies $k = 2$. The following design is most popular, even though it is inferior: *change one factor at a time*; see Fig. 2 and the columns denoted by $\mathbf{x}_1$ and $\mathbf{x}_2$ in Table 1. In that design the analysts usually start with the 'base' scenario, denoted by the row $(0, 0)$. Then the next two scenarios that they run are $(1, 0)$ and $(0, 1)$.

In such a design, the analysts cannot estimate the *interaction* between the two factors. Indeed, Table 1 shows that the estimated interaction (say) $\beta_{1;2}$ is *confounded* with the estimated intercept $\hat{\beta}_0$; i.e., the columns for the corresponding regression variables are linearly dependent. (Confounding remains when the base values are denoted not by zero but by one; then these two columns become identical.)

In practice, analysts often study each factor at *three levels* (denoted by −1, 0, +1) in their one-at-a-time design. However, two levels suffice to estimate a first-order polynomial metamodel—as we saw in Section 3.1.

Table 1
One-factor-at-a-time design for two factors, and possible regression variables

Scenario	$\mathbf{x}_0$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_1\mathbf{x}_2$
1	1	0	0	0
2	1	1	0	0
3	1	0	1	0

To enable the estimation of *interactions*, the analysts must change factors *simultaneously*. An interesting problem arises if k increases from two to three. Then Fig. 2 becomes Fig. 3—which does not show the output w, since it would require a fourth dimension; instead x_3 replaces w. And Table 1 becomes Table 2. The latter table shows the 2^3 factorial design; i.e., each of the three factors has two values, and the analysts simulate all the combinations of these values. To simplify the notation, the table shows only the signs of the factor values, so − means −1 and + means +1. The table further shows possible regression variables, using the symbols '0' through '1.2.3'—to denote the indexes of the regression variables $\mathbf{x}_0$ (which remains 1 in all scenarios) through $\mathbf{x}_1\mathbf{x}_2\mathbf{x}_3$. Further, I point out that each column is *balanced*; i.e., each column has four plusses and four minuses —except for the dummy column.

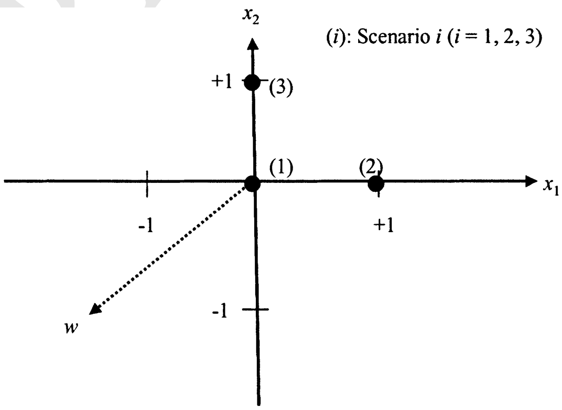

Fig. 2. One-factor-at-a-time design for two factors.

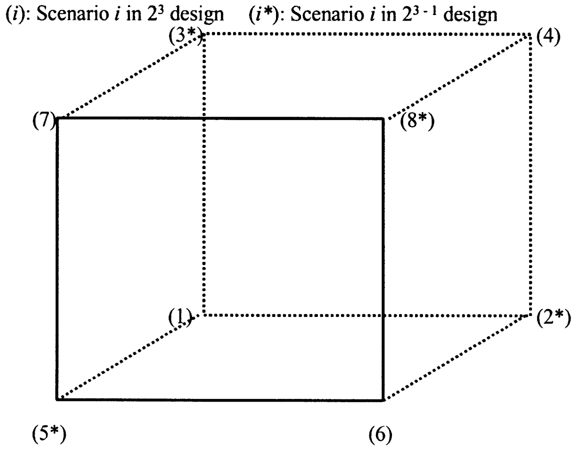

Fig. 3. The 2^3 design.

Table 2
The 2^3 design and possible regression variables

Scenario	0	1	2	3	1.2	1.3	2.3	1.2.3
1	+	−	−	−	+	+	+	−
2	+	+	−	−	−	−	+	+
3	+	−	+	−	−	+	−	+
4	+	+	+	−	+	−	−	−
5	+	−	−	+	+	−	−	+
6	+	+	−	+	−	+	−	−
7	+	−	+	+	−	−	+	−
8	+	+	+	+	+	+	+	+

The 2^3 design enables the estimation of all eight parameters of the following metamodel, which is a third-order polynomial that is *incomplete*; i.e., some parameters are assumed zero:

$$\boldsymbol{y} = \beta_0 + \sum_{j=1}^{3} \beta_j \boldsymbol{x}_j + \sum_{j=1}^{2} \sum_{j'>j}^{3} \beta_{j;j'} \boldsymbol{x}_j \boldsymbol{x}_{j'} + \beta_{1;2;3} \boldsymbol{x}_1 \boldsymbol{x}_2 \boldsymbol{x}_3 + \boldsymbol{e}. \quad (8)$$

Indeed, the 2^3 design implies a matrix of regression variables $\boldsymbol{X}$ that is *orthogonal*

$$(\boldsymbol{X}'\boldsymbol{X}) = \boldsymbol{nI}, \quad (9)$$

where n denotes the number of scenarios simulated; for example, Table 2 implies $n = 8$. Hence the *ordinary least squares* (OLS) estimator

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{w} \quad (10)$$

simplifies for the 2^3 design—which implies (9)—to $\hat{\boldsymbol{\beta}} = \boldsymbol{X}'\boldsymbol{w}/8$.

The *covariance matrix* of the (linear) OLS estimator given by (10) is

$$\mathrm{cov}(\hat{\boldsymbol{\beta}}) = [(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}']\mathrm{cov}(\boldsymbol{w}) \times [(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}']'. \quad (11)$$

In case of *white noise*; i.e.,

$$\mathrm{cov}(\boldsymbol{w}) \in \sigma^2 \boldsymbol{I}, \quad (12)$$

(11) reduces to the well-known formula

$$\mathrm{cov}(\hat{\boldsymbol{\beta}}) = \sigma^2 (\boldsymbol{X}'\boldsymbol{X})^{-1}. \quad (13)$$

However, I claim that in practice this white noise assumption does not hold:

(i) The output variances change as the input (scenario) changes so the assumed common variance σ^2 in (12) does not hold. This is called *variance heterogeneity*. (Well-known examples are queueing models, which have both the mean and the variance of the waiting time increase as the traffic rate increases; see Cheng and Kleijnen, 1999.)

(ii) Often the analysts use *common random numbers* (CRN) so the assumed diagonality of the matrix in (12) does not hold.

Therefore I conclude that the analysts should choose between the following two options.

(i) Continue to apply the OLS *point* estimator (10), but use the *covariance* formula (11) instead of (13).
(ii) Switch from OLS to *generalized least squares* (GLS) with estimated cov($\boldsymbol{w}$) based on $m > n$ replications (using different PRN); for details see Kleijnen (1992, 1998).

The variances of the estimated regression parameters—which are on the main diagonal in (11)—can be used to test statistically whether some factors have zero effects. However, I emphasize that a *significant* factor may be *unimportant*—practically speaking. If the factors are scaled between −1 and +1 (see the transformation in (6)), then the estimated effects quantify the order of importance. For example, in a first-order polynomial metamodel the factor estimated to be the most important factor is the one with the highest absolute value for its estimated effect. Also see Bettonvil and Kleijnen (1990).

3.3. Fractional factorials and other incomplete designs

The incomplete third-order polynomial in (8) included a third-order effect, namely $\beta_{1;2;3}$. Standard DOE textbooks include the definition and estimation of such high-order interactions. However, the following claims may be made:

1. High-order effects are hard to interpret.
2. These effects often have negligible magnitudes.

Claim # 1 seems obvious. If claim #2 holds, then the analysts may simulate fewer scenarios than specified by a full factorial (such as the 2^3 design). For example, if indeed $\beta_{1;2;3}$ is zero, a 2^{3-1} fractional factorial design suffices. A possible 2^{3-1} design is shown in Table 2, deleting the four rows (scenarios) that have a minus sign in the 1.2.3 column (rows 1, 4, 6, 7). In other words, only a *fraction*—namely 2^{-1} of the 2^3 full factorial design—is simulated. This design corresponds with the points denoted by the symbol * in Fig. 3. Note that this figure has the following geometrically property: each scenario corresponds with a vertex that cannot be reached via a single edge of the cube.

This 2^{3-1} design has two identical columns, namely the 1.2.3 column (which has four plusses) and the dummy column 0 (which obviously has four plusses). Hence, the corresponding two effects are confounded—but β_0 is assumed zero, so this confounding can be ignored!

Sometimes a *first-order polynomial* metamodel suffices. For example, in the (sequential) optimization of black-box simulation models the analysts may use a first-order polynomial to estimate the local gradient; see Angün et al. (2002). Then a 2^{k-p} design suffices with the biggest p value such that $2^{k-p} > k$. An example is: $k = 7$ and $p = 4$ so only eight scenarios are simulated; see Table 3. This table shows that the first three factors (labeled 1, 2, and 3) form a full factorial 2^3 design; the symbol '4 = 1.2'

Table 3
A 2^{7-4} design

Scenario	1	2	3	4 = 1.2	5 = 1.3	6 = 2.3	7 = 1.2.3
1	−	−	−	+	+	+	−
2	+	−	−	−	−	+	+
3	−	+	−	−	+	−	+
4	+	+	−	+	−	−	−
5	−	−	+	+	−	−	+
6	+	−	+	−	+	−	−
7	−	+	+	−	−	+	−
8	+	+	+	+	+	+	+

means that the values for factor 4 are specified by multiplying the elements of the columns for the factors 1 and 2. Note that the design is still balanced and orthogonal. Because of this orthogonality, it can be proven that the estimators of the metamodel's parameters have smaller variances than one-factor-at-a time designs give. How to select scenarios in 2^{k-p} designs is discussed in many DOE textbooks, including Kleijnen (1975, 1987).

Actually, these designs—i.e., fractional factorial designs of the 2^{k-p} type with the biggest p value still enabling the estimation of first-order metamodels—are a subset of *Plackett–Burman designs*. The latter designs consists of $k+1$ scenarios rounded upwards to a multiple of four. For example, if $k=11$, then Table 4 applies. If $k=8$, then the Plackett–Burman design is a 2^{7-4} fractional factorial design; see Kleijnen (1975, pp. 330–331) Plackett–Burman designs are tabulated in many DOE textbooks, including Kleijnen (1975). Note that designs for first-order polynomial metamodels are called *resolution III* designs.

Resolution IV designs enable unbiased estimators of first-order effects—even if two-factors interactions are important. These designs require double the number of scenarios required by resolution III designs; i.e., after simulating the scenarios of the resolution III design, the analysts simulate the *mirror scenarios*; i.e., multiply by −1 the factor values in the original scenarios.

Resolution V designs enable unbiased estimators of first-order effects plus two-factor interactions. To this class belong certain 2^{k-p} designs with small enough p values, and *saturated* designs developed by Rechtschaffner (1967); saturated designs are designs with the minimum number of scenarios—that still allow unbiased estimators of the metamodel's parameters. Saturated designs are attractive for *expensive* simulations; i.e., simulations that require relatively much computer time per scenario.

CCD augment Resolution V designs with the base scenario and $2k$ scenarios changing factors one at a time; this changing means increasing and decreasing each factor in turn. Saturated designs (smaller than CCD) are discussed in Kleijnen (1987, pp. 314–316).

3.4. Designs for screening

Most practical, non-academic simulation models have many factors; for example, Kleijnen et al. (2003a,b)—denoted below as Kleijnen et al. (2003b)—experiment with a supply-chain simulation model with nearly 100 factors. Even a Plackett–Burman design would then take 102 scenarios. Because each scenario needs to be replicated several times, the total computer time may then be prohibitive. For that reason, many analysts keep a lot of factors fixed (at their base values), and experiment with only a few remaining factors. An example is a military (agent-based) simulation that was run millions of times for just a few scenarios—changing only a few factors; see Horne and Leonardi (2001).

However, statisticians have developed designs that require fewer than k scenarios—called *supersaturated designs*; see Yamada and Lin (2002).

Table 4
The Placket–Burman design for 11 factors

Scenario	1	2	3	4	5	6	7	8	9	10	11
1	+	−	+	−	−	−	+	+	+	−	+
2	+	+	−	+	−	−	−	+	+	+	−
3	−	+	+	−	+	−	−	−	+	+	+
4	+	+	+	−	+	+	−	−	−	+	+
5	+	−	+	+	−	−	−	−	−	−	+
6	+	+	−	+	+	+	+	+	−	−	−
7	−	+	+	+	−	+	+	−	+	−	−
8	−	−	+	+	+	−	+	+	−	+	−
9	−	−	−	+	+	+	−	+	+	−	+
10	+	−	−	−	+	+	+	−	+	+	−
11	−	+	−	−	−	+	+	+	−	+	+
12	−	−	−	−	−	−	−	−	−	−	−

Some designs *aggregate* the k individual factors into groups of factors. It may then happen that the effects of individual factors cancel out, so the analysts would erroneously conclude that all factors within that group are unimportant. The solution is to define the −1 and +1 levels of the individual factors such that all first-order effects are *non-negative*. As an example, let us return to the metamodel for the M/M/1 simulation in (3), which treats the arrival and service rates as individual factors. Then the value −1 of the arrival rate denotes the lowest value in the experimental area so waiting time tends to be low. The value −1 of the service rate is its high value, so again waiting time tends to be low. My experience is that in practice the users do know the direction of the first-order effects of individual factors—not only in queueing simulations but also in other types (e.g., an ecological simulation with nearly 400 factors discussed by Bettonvil and Kleijnen, 1996).

There are several types of group screening designs; for a recent survey including references, I refer to Kleijnen et al. (2003b). Here I focus on the most efficient type, namely *sequential bifurcation* (abbreviated to SB).

SB is so efficient because it is a *sequential* design. SB starts with only two scenarios, namely, one scenario with all individual factors at −1, and a second scenario with all factors at +1. Comparing the outputs of these two extreme scenarios requires only two replications because the aggregated effect of the group factor is huge compared with the intrinsic noise (caused by the PRN). In the next step, SB splits—*bifurcates*—the factors into two groups. There are several heuristic rules to decide on how to assign factors to groups (again see Kleijnen et al., 2003b). Comparing the outputs of the third scenario with the outputs of the preceding scenarios enables the estimation of the aggregated effect of the individual factors within a group. Groups—and all its individual factors—are eliminated from further experimentation as soon as the group effect is statistically unimportant. Obviously, the groups get smaller as SB proceeds sequentially. SB stops when the first-order effects of all important individual factors are estimated. In the supply-chain simulation only 11 of the 92 factors are classified as important. This shortlist of important factors is further investigated to find a robust solution.

4. Kriging metamodels

Let us return to the M/M/1 example in Fig. 1. If the analysts are interested in the I/O behavior within 'local area 1', then a first-order polynomial such as (4) may be adequate. Maybe, a second-order polynomial such as (7) is required to get an valid metamodel in 'local area 2', which is larger and covers a steeper part of the I/O function. However, Kleijnen and Van Beers (2003a) show that the latter metamodel gives very poor predictions compared with a Kriging metamodel.

Kriging has been often applied in deterministic simulation models. Such simulations are used for computer aided engineering (CAE) in the development of airplanes, automobiles, computer chips, computer monitors, etc.; see Sacks et al. (1989)'s pioneering article, and—for an update—see Simpson et al. (2001).

For random simulations (including discrete-event simulations) there are hardly any applications yet. First, I explain the basics of Kriging; then DOE aspects.

4.1. 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 process; see the classic Kriging textbook Cressie (1993). More precisely, a Kriging prediction is a weighted linear combination of all output values already observed. These weights depend on the distances between the input for which the output is to be predicted and the inputs already simulated. Kriging assumes that *the closer the inputs are*, *the more positively correlated the outputs are*. This assumption is modeled through the correlogram or the related variogram, discussed below.

Note that in deterministic simulation, Kriging has an important advantage over linear regression analysis: Kriging is an *exact* interpolator; that is, predicted values at observed input values are exactly equal to the simulated output values.

The simplest type of Kriging (to which I limit this review) assumes the following *metamodel* (also see (4) with $\mu = \beta_0$ and $\beta_1 = 0$):

$$y = \mu + e \tag{14a}$$

with

$$E(e) = 0, \quad \text{var}(e) = \sigma^2, \tag{14b}$$

where μ is the mean of the stochastic process $\boldsymbol{y}(.)$, and $\boldsymbol{e}$ is the additive noise, which is assumed to have zero mean and constant finite variance σ^2 (furthermore, many authors assume normality). Kriging further assumes a *stationary covariance process*; i.e., the process $\boldsymbol{y}(.)$ has constant mean and constant variance, and the covariances of $\boldsymbol{y}(\mathbf{x}+\mathbf{h})$ and $\boldsymbol{y}(\mathbf{x})$ depend only on the distance between their inputs, namely the lag $|\mathbf{h}| = |(\mathbf{x}+\mathbf{h}) - (\mathbf{x})|$.

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 n simulation output data:

$$\hat{\boldsymbol{y}}(\mathbf{x}_0) = \sum_{i=1}^{n} \lambda_i \cdot \boldsymbol{y}(\mathbf{x}_i) = \boldsymbol{\lambda}' \cdot \boldsymbol{y} \tag{15a}$$

with

$$\sum_{i=1}^{n} \lambda_i = 1, \tag{15b}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)'$ and $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)'$.

To quantify the weights $\boldsymbol{\lambda}$ in (15), Kriging derives the *best linear unbiased estimator* (BLUE), which minimizes the mean squared error (MSE) of the predictor:

$$\text{MSE}(\hat{\boldsymbol{y}}(\mathbf{x}_0)) = \boldsymbol{E}((\boldsymbol{y}(\mathbf{x}_0) - \hat{\boldsymbol{y}}(\mathbf{x}_0))^2)$$

with respect to $\boldsymbol{\lambda}$. Obviously, these weights depend on the covariances mentioned below (14). Cressie (1993) characterizes these covariances through the *variogram*, defined as $2\gamma(\mathbf{h}) = \boldsymbol{var}(\boldsymbol{y}(\mathbf{x}+\mathbf{h}) - \boldsymbol{y}(\mathbf{x}))$. (I follow Cressie (1993), who uses variograms to express covariances, whereas Sacks et al. (1989) use correlation functions.) It can be proven that the *optimal* weights in (15) are

$$\boldsymbol{\lambda}' = \left(\boldsymbol{\gamma} + \mathbf{1}\frac{1 - \mathbf{1}'\boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma}}{\mathbf{1}'\boldsymbol{\Gamma}^{-1}\mathbf{1}}\right)' \boldsymbol{\Gamma}^{-1} \tag{16}$$

with the following symbols: $\boldsymbol{\gamma}$: vector of the n (co)variances between the output at the new input $\mathbf{x}_0$ and the outputs at the n old inputs, so $\boldsymbol{\gamma} = (\gamma(\mathbf{x}_0 - \mathbf{x}_1), \ldots, \gamma(\mathbf{x}_0 - \mathbf{x}_n))'$, $\boldsymbol{\Gamma}$: $n \times n$ matrix of the covariances between the outputs at the n old inputs—with element (i,j) equal to $\gamma(\boldsymbol{x}_i - \boldsymbol{x}_j)$, and $\mathbf{1}$: vector of n ones.

I point out that the optimal weights in (16) vary with the input value for which output is to be predicted (see $\boldsymbol{\gamma}$), whereas linear regression uses the same estimated metamodel (with $\hat{\boldsymbol{\beta}}$) for all inputs to be predicted. (A forthcoming paper discusses the fact that the weights $\boldsymbol{\lambda}$ are estimated via the estimated covariances $\boldsymbol{\gamma}$ and $\boldsymbol{\Gamma}$, so the Kriging predictor is actually a non-linear random variable; see, Kleijnen et al., 2004.)

4.2. Designs for Kriging

The most popular design type for Kriging is *LHS*. This design type was invented by McKay et al. (1979) for deterministic simulation models. Those authors did not analyze the I/O data by Kriging (but they did assume I/O functions more complicated than the polynomial models in classic DOE). Nevertheless, LHS is much applied in Kriging nowadays, because LHS is a simple technique (it is part of spreadsheet add-ons such as @Risk).

LHS offers *flexible* design sizes n (number of scenarios simulated) for any number of simulation inputs, k. An example is shown for $k = 2$ and $n = 4$ in Table 5 and Fig. 4, which are constructed as follows.

1. The table illustrates that LHS divides each input range into n intervals of equal length, numbered from 1 to n (the example has $n = 4$; see the numbers in the last two columns); i.e., the number of values per input can be much larger than in Plackett–Burman designs or CCD.

Table 5
A LHS design for two factors and four scenarios

Scenario	Interval factor 1	Interval factor 2
1	2	1
2	1	4
3	4	3
4	3	2

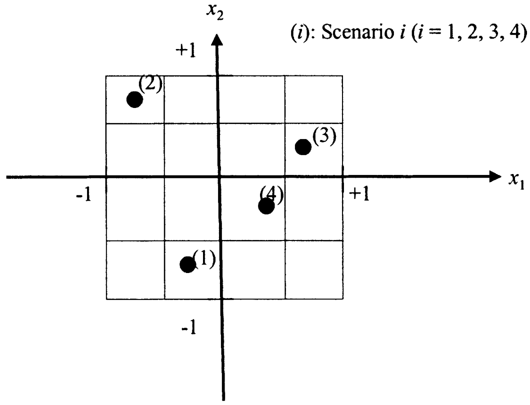

Fig. 4. A LHS design for two factors and four scenarios.

2. Next, LHS places these integers $1, \ldots, n$ such that each integer appears exactly once in each row and each column of the design. (This explains the term 'Latin hypercube': it resembles Latin squares in classic DOE.)

Within each cell of the design in the table, the exact input value may be sampled uniformly; see Fig. 4. (Alternatively, these values may be placed systematically in the middle of each cell. In risk analysis, this uniform sampling may be replaced by sampling from some other distribution for the input values.)

Because LHS implies randomness, its result may happen to be an *outlier*. For example, it might happen—with small probability—that in Fig. 4 all scenarios lie on the main diagonal, so the values of the two inputs have a correlation coefficient of -1. Therefore, the LHS may be adjusted to become (nearly) orthogonal; see Ye (1998).

We may also compare classic designs and LHS geometrically. Fig. 3 illustrates that many classic designs consist of corners of k-dimensional cubes. These designs imply simulation of *extreme scenarios*. LHS, however, has better *space filling* properties. (In risk analysis, the scenarios fill the space according to a—possibly non-uniform—distribution.)

This space filling property has inspired many statisticians to develop related designs. One type maximizes the minimum Euclidean distance between any two points in the k-dimensional experimental area. Other designs minimize the maximum distance. See Koehler and Owen (1996), Santner et al. (2003), and also Kleijnen et al. (2003a).

5. Cross-validation of metamodels

Whatever metamodel is used (polynomial, Kriging, etc.), the analysts should validate that model—once its parameters have been estimated. Kleijnen and Sargent (2000) discuss many criteria. In this review, I focus on the question: does the metamodel give *adequate predictions*? To answer this question, I discuss cross-validation for linear regression; after that discussion, it will be obvious how cross-validation also applies to other metamodel types. I explain a different validation procedure for linear regression models in the Appendix A.

I assume that the analysts use OLS to estimate the regression parameters; see (10). This yields the n classic regression predictors for the n scenarios implied by $\boldsymbol{X}$ in (10):

$$\hat{\boldsymbol{Y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} \tag{17}$$

However, the analysts can also compute regression predictors through *cross-validation*, as follows.

1. Delete I/O combination i from the complete set of n combinations. I suppose that this i ranges from 1 through n, which is called *leave-one-out cross-validation*. I assume that this procedure re-

sults in n non-singular matrixes, each with $n-1$ rows (say) $\boldsymbol{X}_{-i} (i = 1, 2, \ldots, n)$. To satisfy this assumption, the original matrix $\boldsymbol{X}(=\boldsymbol{X}_{-0})$ must satisfy $n > q$ where q denotes the number of regression parameters. Counter-examples are the saturated designs in Tables 3 and 4; the solution is to experiment with one factor less or to add one scenario (e.g., the scenario with all coded x-values set to zero, which is the base scenario).

2. Next the analysts *recompute* the OLS estimator of the regression parameters $\boldsymbol{\beta}$; i.e., they use (10) with $\boldsymbol{X}_{-i}$ and (say) $\boldsymbol{w}_{-i}$ to get $\hat{\boldsymbol{\beta}}_{-i}$.
3. Substituting $\hat{\boldsymbol{\beta}}_{-i}$ (which results from step 2) for $\hat{\boldsymbol{\beta}}$ in (17) gives $\hat{y}_i$, which denotes the *regression predictor* for the scenario deleted in step 1.
4. Executing the preceding three steps for all scenarios gives n predictions $\hat{y}_i$.
5. These $\hat{y}_i$ can be compared with the corresponding simulation outputs $\boldsymbol{w}_i$. This comparison may be done through a *scatter plot*. The analysts may eyeball that plot to decide whether they find the metamodel acceptable.

Case studies using this cross-validation procedure are Vonk Noordegraaf (2002) and Van Groenendaal (1998).

6. Conclusion and further research

Because simulation—treated as a black box—implies *experimentation* with a model, DOE is essential. In this review, I discussed both *classic* DOE for *polynomial* regression metamodels and modern DOE (including LHS) for other metamodels such as Kriging models. The simpler the metamodel is, the fewer scenarios need to be simulated.

I did not discuss so-called *optimal designs* because these designs use statistical assumptions (such as white noise) that I find too unrealistic. A recent discussion of optimal DOE—including references—is Spall (2003).

Neither did I discuss the designs in Taguchi (1987), as I think that the classic and modern designs that I did discuss are superior. Nevertheless, I believe that Taguchi's concepts—as opposed to his statistical techniques—are important. In practice, the 'optimal' solution may break down because the environment turns out to differ from the environment that the analysts assumed when deriving the optimum. Therefore they should look for a 'robust' solution. For further discussion I refer to Kleijnen et al. (2003a).

Because of space limitations, I did not discuss *sequential* DOE, except for SB and two-stage resolution IV designs—even though the sequential nature of simulation experiments (caused by the computer architecture) makes such designs very attractive. See Jin et al. (2002), Kleijnen et al. (2003a), and Kleijnen and Van Beers (2003b).

An interesting research question is: how much computer time should analysts spend on *replication*; how much on exploring *new* scenarios?

Another challenge is to develop designs that explicitly account for *multiple outputs*. This may be a challenge indeed in SB (depending on the output selected to guide the search, different paths lead to the individual factors identified as being important). In practice, multiple outputs are the rule in simulation; see Kleijnen and Smits (2003) and also Kleijnen et al. (2003a).

The application of *Kriging* to *random* simulation models seems a challenge. Moreover, corresponding software needs to be developed. Also see Lophaven et al. (2002).

Comparison of various metamodel types and their designs remains a major problem. For example, Meckesheimer et al. (2001) compare radial basis, neural net, and polynomial metamodels. Clarke et al. (2003) compare low-order polynomials, radial basis functions, Kriging, splines, and support vector regression. Alam et al. (2003) found that LHS gives the best neural-net metamodels. Comparison of screening designs has hardly been done; see Kleijnen et al. (2003a,b).

Acknowledgements

The following colleagues provided comments on a first draft of this paper: Russell Barton (Pennsylvania State University) and Wim van Beers (Tilburg University).

Appendix A. Alternative validation test of linear regression metamodel

Instead of the cross-validation procedure discussed in Section 5, I propose the following test—which applies only to linear regression metamodels (not to other types of metamodels); also see Kleijnen (1992).

The accuracy of the predictor for the new scenario $\mathbf{x}_{n+1}$ based on (17) may be quantified by its variance

$$\mathrm{var}(\hat{\mathbf{y}}_{n+1}) = \mathbf{x}'_{n+1}\ \mathrm{cov}(\hat{\boldsymbol{\beta}})\mathbf{x}_{n+1}, \qquad \text{(A.1)}$$

where $\mathrm{cov}(\hat{\boldsymbol{\beta}})$ was given by (13) in case of white noise. For more realistic cases, I propose that analysts replicate each scenario (say) m times with non-overlapping PRN and $m > 1$, and get m estimates (say) $\hat{\boldsymbol{\beta}}_r (r = 1, \ldots, m)$ of the regression parameters. From these estimates they can estimate $\mathrm{cov}(\hat{\boldsymbol{\beta}})$ in (A.1). (The non-overlapping PRN reduce the $q \times q$ matrix $\mathrm{cov}(\hat{\boldsymbol{\beta}})$ to a diagonal matrix with the elements $\mathrm{var}(\hat{\beta}_j)$, $j = 1, \ldots, q$, on the main diagonal; CRN is allowed.) Note that this validation approach requires replication, whereas cross-validation does not.

Next, the analysts *simulate* this new scenario with new non-overlapping PRN, and get $\mathbf{w}_{n+1}$. To estimate the variance of this simulation output, the analysts may again use m replicates, resulting in $\bar{\mathbf{w}}_{n+1}$ and $\widehat{\mathrm{var}}(\bar{\mathbf{w}}_{n+1})$.

I recommend comparing the regression prediction and the simulation output through a Student t test

$$\mathbf{t}_{m-1} = \frac{\hat{\mathbf{y}}_{n+1} - \bar{\mathbf{w}}_{n+1}}{\{\widehat{\mathrm{var}}(\hat{\mathbf{y}}_{n+1}) + \widehat{\mathrm{var}}(\hat{\mathbf{w}}_{n+1})\}^{1/2}}. \qquad \text{(A.2)}$$

The analysts should reject the metamodel if this test statistic exceeds the $1 - \alpha$ quantile of the $\mathbf{t}_{m-1}$ distribution.

If the analysts simulate *several* new scenarios, then they can still apply the t test in (A.2)—now combined with Bonferroni's inequality.

References

Alam, F.M., McNaught, K.R., Ringrose, T.J., 2003. A comparison of experimental designs in the development of a neural network simulation metamodel. Simulation Modelling: Practice and Theory (accepted conditionally).

Angün, E., den Hertog, D., Gürkan, G., Kleijnen, J.P.C., 2002. Response surface methodology revisited. In: Yücesan, E., Chen, C.H., Snowdon, J.L., Charnes, J.M. (Eds.), Proceedings of the 2002 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 377–383.

Antioniadis, A., Pham, D.T., 1998. Wavelet regression for random or irregular design. Computational Statistics and Data Analysis 28, 353–369.

Bettonvil, B., Kleijnen, J.P.C., 1996. Searching for important factors in simulation models with many factors: sequential bifurcation. European Journal of Operational Research 96 (1), 180–194.

Bettonvil, B., Kleijnen, J.P.C., 1990. Measurement scales and resolution IV designs. American Journal of Mathematical and Management Sciences 10 (3–4), 309–322.

Cheng, R.C.H., Kleijnen, J.P.C., 1999. Improved design of simulation experiments with highly heteroskedastic responses. Operations Research 47 (5), 762–777.

Clarke, S.M., Griebsch, J.H., Simpson, T.W., 2003. Analysis of support vector regression for approximation of complex engineering analyses. Proceedings of DETC '03, ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago.

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

Donohue, J.M., Houck, E.C., Myers, R.H., 1993. Simulation designs and correlation induction for reducing second-order bias in first-order response surfaces. Operations Research 41 (5), 880–902.

Horne, G., Leonardi, M. (Eds.), 2001. Maneuver Warfare Science 2001. Defense Automatic Printing Service, Quantico, VA.

Jin, R., Chen, W., Sudjianto, A., 2002. On sequential sampling for global metamodeling in engineering design. In: Proceedings of DETC '02, ASME 2002 Design Engineering Technical Conferences and Computers and Information in Engineering Conference, DETC2002/DAC-34092, September 29–October 2, Montreal, Canada.

Kleijnen, J.P.C., 2004. Design and analysis of Monte Carlo experiments. In: Gentle, J.E., Haerdle, W., Mori Y. (Eds.), Handbook of Computational Statistics; vol. I: Concepts and Fundamentals. Springer, Heidelberg.

Kleijnen, J.P.C., 1998. Experimental design for sensitivity analysis, optimization, and validation of simulation models. In: Banks, J. (Ed.), Handbook of Simulation. Wiley, NewYork, pp. 173–223.

Kleijnen, J.P.C., 1992. Regression metamodels for simulation with common random numbers: comparison of validation tests and confidence intervals. Management Science 38 (8), 1164–1185.

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

Kleijnen, J.P.C., 1975. Statistical techniques in simulation, vol. II. Marcel Dekker, New York [Russian translation, Publishing House "Statistics", Moscow, 1978].

Kleijnen, J.P.C., Sanchez, S.M., Lucas, T.W., Cioppa, T.M., 2003a. A user's guide to the brave new world of designing simulation experiments. INFORMS Journal on Computing (accepted conditionally).

Kleijnen, J.P.C., Bettonvil, B., Person, F., 2003b. Finding the important factors in large discrete-event simulation: sequential bifurcation and its applications. In: Dean, A.M., Lewis, S.M. (Eds.), Screening, Springer, New York (forthcoming; preprint: http://center.kub.nl/staff/kleijnen/papers.html).

Kleijnen, J.P.C., Sargent, R.G., 2000. A methodology for the fitting and validation of metamodels in simulation. European Journal of Operational Research 120 (1), 14–29.

Kleijnen, J.P.C., Siem, A.Y.D., den Hertog, D., 2004. The correct Kriging variance. (Preliminary title) Working Paper, CentER, Tilburg University, Tilburg, The Netherlands (preprint: http://center.kub.nl/staff/kleijnen/papers.html).

Kleijnen, J.P.C., Smits, M.T., 2003. Performance metrics in supply chain management. Journal Operational Research Society 54 (5), 507–514.

Kleijnen, J.P.C., Van Beers, W.C.M., 2003a. Robustness of Kriging when interpolating in random simulation with heterogeneous variances: some experiments. European Journal of Operational Research (accepted conditionally).

Kleijnen, J.P.C., Van Beers, W.C.M., 2003b.Application -driven sequential designs for simulation experiments: Kriging metamodeling. Working Paper, CentER, Tilburg University, Tilburg, The Netherlands (submitted for publication; preprint: http://center.kub.nl/staff/kleijnen/papers.html).

Koehler, J.R., Owen, A.B., 1996. Computer experiments. In: Ghosh, S., Rao, C.R. (Eds.), Handbook of Statistics, vol. 13. Elsevier, Amsterdam, pp. 261–308.

Law, A.M., Kelton, W.D., 2000. Simulation Modeling and Analysis, third ed. McGraw-Hill, New York.

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

McKay, M.D., Beckman, R.J., Conover, W.J., 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21 (2), 239–245 (reprinted in 2000: Technometrics 42 (1), 55–61).

Meckesheimer, M., Barton, R.R., Limayem, F., Yannou, B., 2001. Metamodeling of combined discrete/continuous responses. AIAA Journal 39, 1950–1959.

Rechtschaffner, R.L., 1967. Saturated fractions of 2n and 3n factorial designs. Technometrics 9, 569–575.

Sacks, J., Welch, W.J., Mitchell, T.J., Wynn, H.P., 1989. Design and analysis of computer experiments. Statistical Science 4 (4), 409–435.

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

Schruben, L.W., Margolin, B.H., 1978. Pseudorandom number assignment in statistically designed simulation and distribution sampling experiments. Journal American Statistical Association 73 (363), 504–525.

Simpson, T.W., Mauery, T.M., Korte, J.J., Mistree, F., 2001. Kriging metamodels for global approximation in simulation-based multidisciplinary design optimization. AIAA Journal 39 (12), 2233–2241.

Spall, J.C., 2003. Introduction to stochastic search and optimization; estimation, simulation, and control. Wiley, New York.

Taguchi, G., 1987. System of Experimental Designs, vol. 1 and 2. UNIPUB/Krauss International, White Plains, NY.

Van Groenendaal, W.J.H., 1998. The Economic Appraisal of Natural Gas Projects. Oxford University Press, Oxford.

Vonk Noordegraaf, A., 2002. Simulation modelling to support national policy making in the control of bovine herpes virus 1. Doctoral dissertation, Wageningen University, Wageningen, The Netherlands.

Yamada, S., Lin, D.K.J., 2002. Construction of mixed-level supersaturated design. Metrika (56), 205–214, Available from interrefhttp://springerlink.metapress.com/app/home/content.asp?wasp = 4h8ac83qyg2rvm9lwa7w&refer-qyg2rvm9lwa7w&referrer = contribution&for-contribution&format = 2&page = 1url<http://springer-link.metapress.com/app/home/con-tent.asp?wasp = 4h8ac83qyg2rvm9lwa7w&referrer = contri-error = contribution&format = 2&page = 1>.

Ye, K.Q., 1998. Orthogonal column Latin hypercubes and their application in computer experiments. Journal Association Statistical Analysis, Theory and Methods (93), 1430–1439.

Zeigler, B.P., Praehofer, K., Kim, T.G., 2000. Theory of Modeling and Simulation, second ed. Academic Press, New York.