CentER

Discussion Paper

No. 2006–10

REGRESSION MODELS AND EXPERIMENTAL DESIGNS: A TUTORIAL FOR SIMULATION ANALYSTS

By Jack P.C. Kleijnen

February 2006

ISSN 0924-7815

TILBURG ◆ UNIVERSITY

Regression models and experimental designs: a tutorial for simulation analysts

Jack P.C. Kleijnen

Tilburg University, Faculty of Economics and Business Administration,Tilburg, the Netherlands

Abstract

This tutorial explains the basics of linear regression models—especially low-order polynomials—and the corresponding statistical designs—namely, designs of resolution III, IV, V, and Central Composite Designs (CCDs). This tutorial assumes 'white noise', which means that the residuals of the fitted linear regression model are normally, independently, and identically distributed with zero mean. The tutorial gathers statistical results that are scattered throughout the literature on mathematical statistics, and presents these results in a form that is understandable to simulation analysts.

Key words: metamodels, fractional factorial designs, Plackett-Burman designs, factor interactions, validation, cross-validation

JEL codes: C0, C1, C9, C15,, C44

1 Introduction

This tutorial is an introduction to the Design and Analysis of Simulation Experiments (DASE).The *goals* of DASE are verification and validation (V & V) of the simulation model, its sensitivity (or what-if) analysis, optimization, and risk (or robustness) analysis. These goals require that the simulation analysts pay attention to the *design* of their experiments; for example, if the experimenters keep an input of the simulation model constant, then they cannot estimate the effect of that input on the output. In practice, however, most analysts keep many inputs constant, and experiment with a few factors only. This tutorial shows that there are better ways to run simulation experiments with many factors. Another example of poor practice is changing only one input at a time (while keeping all other inputs fixed at their so-called base values). This article proves that such an approach does not enable the estimation of interactions among inputs.

The design of the experiment is intimately related to its *analysis*; indeed, it is a chicken-and-egg problem. Consider the following example. Suppose the analysts assume that the input has a 'linear' effect on the output; i.e., they assume a first-order polynomial approximation (which is justified by the Taylor series argument in mathematics) or main effects only (which is the statistical terminology). Then it suffices to experiment with only two values per factor. Moreover, the analysts may assume that there are (say) $k > 1$ factors and that these factors have main effects only. Then a good design requires a relatively small experiment (of order k). For example, changing only one factor at a time does give unbiased estimators of the main effects. This tutorial, however, shows that to minimize the variances of these estimators, a different design is needed—with approximately the same number of simulation runs as required by the one-factor-at-a-time design.

A first-order polynomial approximation may be called a metamodel (see [19]), because it is an approximation of the Input/Output (I/O) behavior of the underlying simulation model. Metamodels are also called response surfaces, emulators, auxiliary models, repromodels, etc. There are different *types* of metamodels, but polynomials of first or second order (degree) have established a track record in both random and deterministic simulations. In deterministic simulation, another metamodel type is popular, namely Kriging models. Less popular are non-linear regression models, Classification and Regression Trees (CART), Multivariate Adaptive Regression Splines (MARS), (artificial) neural networks, radial functions, rational functions, splines, support vector regression, wavelets, and other types; see [7], [16], [23], and [44].

The term 'response surface' is used for *local* metamodels in *Response Surface Methodology* (RSM) and for *global* metamodels in deterministic simulation. RSM was introduced in 1951 by Box and Wilson (see [6]) as an iterative heuristic for optimizing real (physical) systems; a recent textbook is [34]. This tutorial includes RSM designs for the optimization of simulated systems. The oldest references to the term 'response surface' in deterministic simulation seem a 1985 American article ([10]) and a 1984 European monograph ([37]).

DASE has *strategic* and *tactical* aspects. Traditionally, researchers in Discrete Event Dynamic Simulation (DEDS) have focused on *tactical* issues, such as the runlength of a steady-state simulation, the number of runs of a terminating simulation, and Variance Reduction Techniques; see the classic 1963 article by Conway [9] and the 2004 review article [35]. In deterministic simulation—where these tactical issues vanish— statisticians have been attracted to *strate-*

gic issues, namely which scenarios to simulate and how to analyze the resulting data; see the classic 1996 publication by Koehler and Owen [31] and the 2003 textbook by Santner, Williams, and Notz [42]. Few statisticians have studied random simulation. And only some simulation analysts have focused on strategic issues. This tutorial focuses on strategic issues; it discusses only those tactical issues that are closely related to strategic issues.

The statistical theory called *Design Of Experiments* (DOE) was developed for real, non-simulated experiments in agriculture in the 1920s ([7] references a 1926 publication by Fisher), and in engineering, psychology, etc. since the 1950s. In real experiments it is impractical to investigate *many* factors; ten factors seems a maximum. Moreover, in real-life experiments it is hard to experiment with factors that have more than *a few* values; five values per factor seems the limit. In simulated experiments, however, these restrictions do not apply. So a *change of mindset* of the simulation experimenter is necessary. A more detailed discussion of simulation versus real experiments is presented in the 2005 survey article [27].

In summary, DASE is needed to improve the efficiency and effectiveness of simulation; i.e., DASE is crucial in the overall process of simulation (also see [32]).

Before proceeding, it is necessary to define some *symbols and terms* because DASE is a combination of mathematical statistics and linear algebra that is applied to experiments with deterministic and random simulation models; these models are applied in many scientific fields—ranging from sociology to astronomy. An excellent survey of this spectrum of applications is Karplus's classic 1983 paper [17].

This article uses Greek letters to denote *parameters*, which are model quantities that have values that cannot be directly observed in the real world so these values must be inferred from other real data; see [51]. For example, the service rate μ in a single-server queueing simulation is estimated from the (say) c observations on the service time s (a classic estimate is $\widehat{\mu} = 1/\overline{s}$ with $\overline{s} = \sum_{i=1}^{c} s_i/c$). Note that an estimator (for example, the sample average) is a random variable, which has a specific value—once it has been computed; this value is called an estimate.

Unlike a parameter, a *variable* can be directly observed in the real world. For example, the input variable service time s can be measured in a straightforward way. A variable may be either an input or an output of a model. For example, besides the input s, the queueing simulation may have the output w, waiting time.

Both parameters and input variables may be changed in a simulation experiment; i.e., they have at least two *values* or *levels* in the experiment. Parameters and input variables together are called *factors,* in DOE. For example, a simple design in DOE is a 2^k factorial experiment; i.e., there are k factors, each with two levels; all their combinations are simulated. These combinations are often called *scenarios* in simulation and modeling. Scenarios are usually called *design points* or *runs* by statisticians. This article reserves the term 'run' for a *simulation run*, which starts the simulation program in the initial conditions (for example, the empty state in a queueing simulation) and stops the simulation program once a specific state has been reached (for example, c customers have been simulated).

Factors (inputs) and responses (outputs) may be either *qualitative* or *quantitative.* In the queueing example, quantitative factors are the arrival and service rates; a qualitative factor may be the priority rule—which may have (say) three levels, namely First-In-First-Out (FIFO, Last-In-First-Out (LIFO), or Shortest-Processingtime-First (SPT).

This tutorial is based on Chapters 1 and 2 of a forthcoming book; see [23]. The book adds many more mathematical and statistical details, examples, and exercises to this article.

The remainder of this article is organized as follows. Section 2 discusses black box versus white box approaches in DASE. Section 3 covers the basics of linear regression analysis. Section 4 focuses on first-order polynomial regression. Section 4 presents designs for such first-order polynomials, namely so-called resolution-III designs. Section 6 augments the first-order polynomial regression model with interactions among the factors. Section 7 discusses designs that give unbiased estimators of the main effects—even if there are two-factor interactions: resolution-IV designs. Section 8 presents designs that also estimate the individual two-factor interactions: resolution-V designs. Section 9 extends the regression model to second-order polynomials. Section 10 presents designs for these second-degree polynomials, focussing on Central Composite Designs (CCDs). Section 11 discusses validation of the assumed regression model, including the coefficient of determination R^2 and the adjusted coefficient $R^2_{adjusted}$, Pearson's and Spearman's correlation coefficients, and cross-validation. Section 12 gives conclusions and topics for further research. The tutorial ends with 51 references enabling the readers to further explore topics that are relevant to their particular applications; some older references are included to provide a historical perspective.

2 White box versus black box approaches

This tutorial treats the simulation model as a black box—not as a white box. To explain the difference, consider the following queueing model. A *white box* representation of this model is presented in the next equations. A popular performance measure (response variable, output) of *any* queueing simulation is

$$\overline{w} = \frac{\sum_{i=1}^{c} w_i}{n} \tag{1}$$

where $\overline{w}$ denotes the average waiting time, w_i the waiting time of customer i, and c the number of customers that stops the simulation run. An alternative output may be the estimated 90% quantile, $w_{(\lceil .90n+0.5 \rceil)}$ where $w_{(i)}$ denotes the order statistics and $\lceil .90n + 0.5 \rceil$ means that $0.90n$ is rounded to the next integer. The dynamics of any single server queueing simulation with First-In-First-Out (FIFO) queueing discipline is specified by

$$w_i = \max(0, w_{i-1} + s_{i-1} - a_i) \tag{2}$$

where a_i denotes the inter-arrival time between customers i and $i-1$, and s_{i-1} denotes the service time of customer $i-1$. Suppose, the simulation starts in the empty state, so $w_1 = 0$. An M/M/1 simulation model samples the random input variables s and a such that these variables have a service rate μ and an arrival rate λ, using a single stream of Pseudo-Random Numbers (PRNs) r with seed (initial PRN) r_0, and exponential (Markovian, symbol M) service and interarrival times:

$$s_i = \frac{-\ln r_{2i-1}}{\mu} \text{ and } a_{i+1} = \frac{-\ln r_{2i}}{\lambda}. \tag{3}$$

Such a white-box representation is used by *Perturbation Analysis* (PA) and *Score Function* (SF) analysis (to estimate the gradient for local sensitivity analysis and optimization). PA and SF are discussed in (for example) Spall's recent textbook [45]; also see Rubinstein and Shapiro's classic SF book, [40], and Ho and Cao's classic PA book, [14]. (The estimation of the gradient will be further discussed in Section 4.)

DASE, however, does not follow a white-box approach; instead it uses a *black-box* approach, which is also used by DOE for real-world experiments (see, for example, [34]) and by Design and Analysis of Computer Experiments (DACE) for deterministic simulation experiments (see, for example, [42]). A black box representation of any single server simulation model with arrival and service rates λ and μ—and a fixed queueing discipline (for example, FIFO), a fixed waiting room capacity, etc.—is

$$\overline{w} = w(\lambda, \mu) \tag{4}$$

where $w(.)$ denotes the mathematical function implicitly defined by the computer simulation program that implements the equations (1) through (3). The dependence of $\overline{w}$ on the seed r_0 might have been explicitly shown, resulting in $\overline{w} = w(\lambda, \mu, r_0)$. The representation in (4), however, is better because the metamodel's goal is to predict and explain a characteristic of the distribution of the output (such as the mean or the 90% quantile), which is not random.

One possible metamodel of the black box model in (4) is a *first-order polynomial* in the two input variables λ and μ:

$$y = \beta_0 + \beta_1\lambda + \beta_2\mu + e \tag{5}$$

where y is the metamodel predictor of the simulation output $\overline{w}$ in (4); β_0, β_1,and β_2 are the parameters of this metamodel—which may be collected in the vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)'$; and e is the residual or noise—which includes both *lack of fit* of the metamodel (this metamodel is a Taylor series approximation cut off after the first-order effects) and *intrinsic noise* (caused by the PRNs).

Besides 5), there are many alternative metamodels. For example, a simpler metamodel is

$$y = \beta_0 + \beta_1 x + e \tag{6}$$

where x is the traffic rate—in queueing theory usually denoted by ρ (statisticians often use this symbol to denote a correlation coefficient; in this tutorial, the context should clarify what this symbol means):

$$x = \rho = \frac{\lambda}{\mu}. \tag{7}$$

This combination of the two original factors λ and μ into a single factor ρ (inspired by queueing theory) illustrates the use of *transformations.* Another useful transformation may be a logarithmic one: replacing y, λ, and μ by $\log(y)$, $\log(\lambda)$, and $\log(\mu)$ in (5) makes the first-order polynomial approximate relative changes; i.e., the regression parameters $\boldsymbol{\beta}$ become elasticity coefficients (by definition, the elasticity coefficient of y with respect to x is $(\partial y/y)/(\partial x/x)$). These transformations illustrate that simulation analysts should be guided by knowledge of the real system and corresponding analytical models.

3 Linear regression analysis: basics

It is convenient to use matrix representation for a *linear regression model* with multiple inputs and a single output. (Assuming a single output obviates the need for multivariate regression, which is beyond the scope of this tutorial; the univariate regression model of this tutorial may be applied to each individual output of a given simulation model.) The matrix notation of the general linear regression model is

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \tag{8}$$

where $\mathbf{y} = (y_1, \ldots, y_n)'$ denotes the n-dimensional vector with the regression predictor (or dependent variable) y with n the number of simulation runs (or observations); $\mathbf{X} = (\mathbf{x}_{ij})$ denotes the $n \times q$ matrix of explanatory (independent) regression variables with $\mathbf{x}_{ij}$ the value of explanatory variable j in run i ($i = 1, \ldots, n; j = 1, \ldots, q$); $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_q)'$ denotes the q regression parameters—including the effect of a possible dummy variable (if there is such a dummy variable, then β_1 is the intercept in the regression model); and $\mathbf{e} = (e_1, \ldots, e_n)'$ denotes the residuals in the n runs.

To select specific values (say) $\widehat{\boldsymbol{\beta}} = (\widehat{\beta}_1, \ldots, \widehat{\beta}_q)'$ for the regression parameters, the *Least Squares* (LS)—also called the Ordinary LS—criterion is often used; i.e., $\widehat{\boldsymbol{\beta}}$ is selected such that it minimizes the *Sum of Squared Residuals*, *SSR*:

$$\min_{\widehat{\boldsymbol{\beta}}} SSR = \sum_{i=1}^{n} (\widehat{e}_i)^2 = \sum_{i=1}^{n} (\widehat{y}_i - w_i)^2 = (\widehat{\mathbf{y}} - \mathbf{w})'(\widehat{\mathbf{y}} - \mathbf{w}) \tag{9}$$

where $\widehat{e}_i = \widehat{y}_i - w_i$ is the estimated residual for input combination i,

$$\widehat{y}_i = \sum_{j=1}^{q} x_{ij} \widehat{\beta}_j = \mathbf{x}_i' \widehat{\boldsymbol{\beta}}, \tag{10}$$

and w_i denotes the simulation output of run i (for example, the average waiting time of that run; see (4)).

The solution of (9) gives the LS estimate $\widehat{\boldsymbol{\beta}}$ of the regression parameter vector $\boldsymbol{\beta}$ in the regression model (8). This solution can be derived to be

$$\widehat{\boldsymbol{\beta}} \ = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{w}. \tag{11}$$

Obviously, this LS estimate exists only if $\mathbf{X}$ is not collinear, so the inverse $(\mathbf{X}'\mathbf{X})^{-1}$ does exist. The selection of a 'good' $\mathbf{X}$ in (8)—and hence in (11)—is discussed in the next sections.

The LS criterion, which is used in (9), is a mathematical (not a statistical) criterion. This criterion is also known as the L_2 norm (other popular mathematical criteria are the L_1 and the L_∞ norms). However, adding statistical assumptions about the simulation I/O data implies that the LS estimator has interesting statistical properties. Therefore this tutorial assumes *white noise*; i.e., the noise is Normally, Independently, and Identically Distributed (NIID) with zero mean. This definition deserves some comments:

- The simulation output w is indeed *normally* (or Gaussian) distributed if this output is an average; for example, (1) defines the simulation output as the average of c individual waiting times. These individual times are (positively) autocorrelated, so the classic Central Limit Theorem (CLT) does not apply. Yet it can be proven that for large c (i.e., a long simulation run) this average tends to be normally distributed.

- The simulation outputs w_i and $w_{i'}$ with $i \neq i'$ are indeed *independent* if they use non-overlapping PRN streams. So the use of Common Random Numbers (CRNs) violates this assumption.
- '*Identically* distributed' implies a constant variance. In practice, however, the simulation outputs do not have the same variance; in other words, the variances are heterogeneous or heteroscedastic instead of homogeneous. For example, for the M/M/1 it is well-known that the variances increase as the traffic rate increases (actually, the variances increase much more than the means). This practical problem is further discussed in [23].

This tutorial assumes that the simulation outputs w are indeed normally and independently distributed with the same variance (say) σ_w^2. Obviously, the simulation outputs may have different means. Furthermore, the linear regression model may be a *valid* metamodel for the variation in these means; i.e., the regression residuals may have zero means: $E(e) = 0$. By definition, a metamodel has *perfect fit* if and only if all its estimated residuals are zero: $\forall i : \widehat{e}_i = 0$ $(i = 1, \ldots n)$. This also deserves some comments:

- The metamodel is *biased* if $E(e) \neq 0$. If $E(e) > 0$, then it overestimates the expected simulation output; if $E(e) < 0$, then it underestimates.
- A *perfectly* fitting metamodel indicates that n (number of simulation runs) is too small. (Also see the discussion of the special case $R^2 = 1$ in Section 11.1).

If the residuals are white noise, then LS gives the *Best Linear Unbiased Estimator* (BLUE). The LS estimator is indeed a *linear* transformation of the random simulation response $\mathbf{w}$:

$$\hat{\boldsymbol{\beta}} = \mathbf{Lw} \tag{12}$$

where $\mathbf{L}$ is not random since $\mathbf{L} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ in (11). The linear estimator (12) has the following two properties:

$$E(\hat{\boldsymbol{\beta}}) = \mathbf{L}[E(\mathbf{w})] \tag{13}$$

and

$$\mathbf{cov}(\hat{\boldsymbol{\beta}}) = \mathbf{L}[\mathbf{cov}(\mathbf{w})]\mathbf{L}'. \tag{14}$$

It is easy to prove that (13) implies that the LS estimator $\hat{\boldsymbol{\beta}}$ is unbiased. And

the property in (14) implies that—in case of white noise—$\hat{\boldsymbol{\beta}}$ has the following covariance matrix:

$$\mathbf{cov}(\hat{\boldsymbol{\beta}}) = (\mathbf{X}'\mathbf{X})^{-1}\sigma_w^2. \tag{15}$$

Furthermore, it can be proven that among all linear unbiased estimators, $\hat{\boldsymbol{\beta}}$ is *best*; i.e., $\hat{\boldsymbol{\beta}}$ has the minimum variance. Obviously, the variances of the individual regression estimators $\widehat{\beta}_j$ are given by the main diagonal elements of (15);

their covariances are given by the off-diagonal elements of this (symmetric) matrix.

The *linear* estimator $\widehat{\boldsymbol{\beta}}$ has another interesting property if the simulation outputs $\mathbf{w}$ are *normally* distributed: $\widehat{\boldsymbol{\beta}}$ is then also normally distributed:

$$\widehat{\boldsymbol{\beta}} \sim N[\boldsymbol{\beta}, (\mathbf{X}'\mathbf{X})^{-1}\sigma_w^2]. \tag{16}$$

Consequently, the individual estimated regression parameters $\widehat{\beta}_j$ may be tested through the following *Student t* statistic:

$$t_{n-q} = \frac{\widehat{\beta}_j - \beta_j}{s(\widehat{\beta}_j)} \text{ with } j = 1, \ldots, q \tag{17}$$

where $s(\widehat{\beta_j})$ is the square root of the j^{th} element on the main diagonal of (14)where σ_w^2 is estimated through the *Mean Squared Residuals* (MSR):

$$MSR = \frac{SSR}{n-q} = \frac{(\widehat{\mathbf{y}} - \mathbf{w})'(\widehat{\mathbf{y}} - \mathbf{w})}{n-q} \tag{18}$$

where SSR was given in (9). The MSR in (18) assumes that *degrees of freedom* are left over after fitting the regression (meta)model: $n > q$. (An alternative estimator of the simulation output's variance uses replications; see (23)).The t statistic in (17) may be used to test whether a specific regression parameter is zero:

$$H_0 : \beta_j = 0. \tag{19}$$

Besides testing a single parameter, the analysts may hypothesize that *several* parameters have specific values; for example, the effects of both the arrival rate and the service rate may be hypothesized to be zero: $\beta_1 = 0$ and $\beta_2 = 0$ in (5). More generally,

$$H_0 : \beta_{j'} = \ldots = \beta_q = 0 \tag{20}$$

where the q parameters are arranged such that the last $q - j' + 1$ parameters are hypothesized to be zero. To test this *composite* hypothesis, the following F-statistic can be used (see, for example, Searle's general regression textbook [43]):

(1) Compute the SSR without the null-hypothesis; this is called the SSR of the *full* regression model: SSR_{full}.
(2) Compute the SSR under the null-hypothesis, called the SSR of the *reduced* regression model: $SSR_{reduced}$.
(3) Compute

$$F_{q-j'+1;n-q} = \frac{SSR_{reduced} - SSR_{full}}{SSR_{full}}. \tag{21}$$

The composite null-hypothesis is rejected if this statistic exceeds $F_{q-j'+1;n-q;1-\alpha}$, which denotes the $1-\alpha$ quantile of the $F_{q-j'+1;n-q}$ distribution.

The preceding linear regression formulas apply to I/O data obtained through

(1) *passive* observation of a real system
(2) *active* experimentation with either a real system or a simulation model of a real system.

The following formulas, however, apply only if the data are obtained through controlled experimentation; i.e., at least one combination of the explanatory variables $\mathbf{x}_i = (x_{i1}, \ldots, x_{iq})'$ in (8) is observed more than once. A *replicate* (or replication) means that a given combination of the explanatory variables $\mathbf{x}_i$ is observed (say) $m_i > 1$ times. The classic assumption is that these replicates are IID. This assumption is satisfied in DEDS simulation if the replicates use non-overlapping PRN streams. If the output is the response of a non-terminating simulation, then IID implies that the subrun outputs have negligible autocorrelation. If the subruns are actually renewal cycles, then the IID assumption is satisfied by definition. Details on this IID property can be found in any textbook on DEDS simulation; fore example, [32].

Replication implies that at least one input combination $\mathbf{x}_i$ is repeated in the matrix of explanatory variables, $\mathbf{X}$. Hence, the number of rows of $\mathbf{X}$ increases from n to (say)

$$N = \sum_{i=1}^{n} m_i. \tag{22}$$

It is possible to keep the number of rows limited to the n *different* combinations. The output of the i^{th}combination then becomes the output averaged over the m_i replicates (also see (24)). If the number of replicates is a constant ($m_i = m$), then the LS estimate may be computed from these averages. Otherwise, these averages should be weighted by the number of replicates; see (26) and also [21], p. 195.

If input combination $\mathbf{x}_i$ is replicated $m_i > 1$ times, then the classic unbiased variance estimator is

$$\widehat{var}(w_i) = \widehat{\sigma}^2(w_i) = s_i^2(w) = \frac{\sum_{r=1}^{m_i}(w_{ir} - \overline{w}i)^2}{m_i - 1}(i = 1, \ldots n) \tag{23}$$

with

$$\overline{w}i = \frac{\sum_{r=1}^{m_i} w_{ir}}{m_i}. \tag{24}$$

Because of the common variance assumption implied by the white noise assumption, the n variance estimators in (23) may be *pooled* using their degrees of freedom as weights:

$$\widehat{var}(w) = \widehat{\sigma}_w^2 = s^2(w) = \frac{\sum_{i=1}^{n}(m_i - 1)s_i^2}{\sum_{i=1}^{n}(m_i - 1)}. \tag{25}$$

If and only if the regression model is valid, there are now the following two variance estimators:

- The MSR (which was defined in (18) for non-replicated combinations), which is now defined in (26) for the current situation with replicated combinations. MSR uses the fitted regression model. If the regression model is not valid, then obviously MSR overestimates the true variance.
- The *pooled* variance estimator in (25), which uses replicates. This estimator does not use the fitted regression model; it is unbiased assuming the simulation outputs for a replicated combination are IID (not necessarily NIID; however, the F statistic does assume normality).

These two estimators can be compared through the following so-called *lack-of-fit F-statistic* (see. for example, [34], p. 52):

$$F_{n-q;N-n} = \frac{\sum_{i=1}^{n} m_i(\overline{w}_i - \widehat{y}_i)^2/(n-q)}{\sum_{i=1}^{n}\sum_{r=1}^{m_i}(w_{ir} - \overline{w}_i)^2/(N-n)}. \tag{26}$$

The numerator uses the MSR computed from the *average* simulation output per combination; at least one combination is replicated (the center of the simulation area is often replicated, when applying classic DOE to simulation). Obviously, the regression model is rejected if this statistic is significantly high. (An alternative validation test will be presented in Section 11.2).

4 Linear regression analysis: first-order polynomials

To estimate the parameters of whatever black-box metamodel (for example, $\boldsymbol{\beta}$ in the linear regression model (8)), the analysts must experiment with the simulation model; i.e., they must change the inputs of the simulation program, run the program, and analyze the resulting I/O data. This section assumes that a first-order polynomial is a valid metamodel.

The simplest metamodel is a *first-order polynomial with a single factor*; see (6). To fit such a straight line, it obviously suffices to have only *two* I/O observations. It is easy to prove that the white noise assumption implies that selecting those two values *as far apart as possible* gives the 'best' estimator of the parameters in (6). The validity of the fitted polynomial, however, becomes questionable as the experimental area gets bigger. (Zeigler et al. [51] call this area the *experimental frame*; it might also be called the domain of admissible scenarios—given the goals of the simulation study (various goals are discussed in [28] and [32].)

A first-order polynomial with *multiple factors* (namely, $k > 1$) may be represented as follows (denoting the dummy factor by $x_0 = 1$ and its effect by β_0):

$$E(y) = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k. \tag{27}$$

So the general linear regression model (8) now has q (number of regression parameters) equal to $k + 1$. An example is the first-order polynomial for the two factors λ and μ in (5).

In practice, such a first-order polynomial may be very useful when trying to estimate the *optimal* values for the inputs of a simulation model. For example, the analysts may wish to find the input values that maximize the profit of the simulated company. There are many methods for estimating the optimal input combination. Some of these methods use the gradient, which is the vector with the first-order derivatives: $\nabla(w) = (\partial w/x_1, \ldots, \partial w/\partial x_k)$. To estimate the gradient, many mathematical publications change one factor at a time—using two or three values per factor (see [45]). From the statistical theory on DOE, however, it follows that it is more efficient to estimate the gradient through a (full or fractional) factorial design and to fit a first-order polynomial to the resulting I/O data; also see [2].

More general (not only in optimization), the LS estimation of the $k + 1$ parameters $\boldsymbol{\beta}$ in (27) often uses either one-factor-at-a-time designs or full factorial designs. In practice, analysts often change each factor one at a time. DOE, however, may use a 2^k design where each of the k factors has two levels (values). Obviously, two values suffice for the first-order polynomial metamodel (27). Also see below (Section 5).

It is convenient and traditional in DOE to use *coded*—also called *standardized* or *scaled*—factor values. If each factor has only two levels in the whole experiment, then these levels may be denoted by -1 and +1. This implies the following linear transformation with z_j denoting the quantitative factor j measured on the original scale, l_j its lower value in the experiment, u_j its upper value, $j = 1, \ldots, k$; and $i = 1, \ldots n$:

$$x_{ij} = a_j + b_j z_{ij} \text{ with } a_j = \frac{l_j + u_j}{l_j - u_j} \text{ and } b_j = \frac{2}{u_j - l_j}. \tag{28}$$

This transformation implies

$$x_{ij} = \frac{z_{ij} - \bar{z}_j}{(u_j - l_j)/2} \tag{29}$$

where $\bar{z}_j$ denotes the average value of factor j in a *balanced* experiment, which means that each factor has the lower value in half of the n runs; the denominator $(u_j - l_j)$ is known as the *range* of factor j.

If the original variable z has either a *nominal* or an *ordinal* scale and it has only two levels, then the coding remains simple: arbitrarily associate one level with -1 and the other level with $+1$. For example, one level means that some patients have preemptive priority (say, emergency patients in a hospital simulation), whereas the other level means that this priority does not apply (so all patients are served FIFO); therefore -1 may mean that a rule does not apply or is *switched off.*

In practice, simulation analysts also consider inputs with *nominal scales with more than two levels.* For example, [22] presents a simulation study on the use of sonar to search for mines at the bottom of the sea. This bottom consists of clay, sand, or rocks—which affects the sonar's output. The simulation analysts erroneously coded these three bottom types as -1, 0, and $+1$. The correct coding of a nominal scale with two or more levels may be done through *multiple binary* variables—each coded as 0 and $+1$—instead of a single variable that is coded as -1 and $+1$; see [23].

Standardization such that each factor (either quantitative or qualitative) varies between -1 and $+1$ is useful when *comparing* the effects of multiple factors. For example, two quantitative factors may have different ranges (assuming the same scale) and the marginal effect of factor 2 may be higher than the marginal effect of factor 1; nevertheless, if the range of factor 1 is much bigger, then 'the' effect of this factor is larger. To *rank* the factor effects, the absolute values of the standardized effects β_j should be sorted; also see [23].

Note that a factor may be significant when tested through the t statistic defined in (17), but may be unimportant—especially when compared with other factors in the experiment.

A 2^kdesign results in an *orthogonal* matrix of explanatory variables for the first-order polynomial (27); that is,

$$\mathbf{X}'\mathbf{X} = n\mathbf{I}. \tag{30}$$

This property follows directly from the way a 2^kdesign is constructed. This property simplifies the LS estimator: substituting (30) into (11) gives

$$\widehat{\boldsymbol{\beta}} = (\widehat{\beta}_j) = (n\mathbf{I})^{-1}\mathbf{X}'\mathbf{w} = \mathbf{X}'\mathbf{w}/n = (\mathbf{x}_j\mathbf{w}/n) = \left(\frac{\sum_{i=1}^{n} x_{ij}w_i}{n}\right) \; (j = 1, \ldots q). \tag{31}$$

In this equation, half of the x_{ij} is -1 and the other half is $+1$, so $\widehat{\beta}_j$ is simply the difference between two averages that vary with j:

$$\widehat{\beta}_j = \frac{\sum_{i=1}^{n} x_{ij} w_i/(n/2)}{2} = \frac{\overline{w}_{1j} - \overline{w}_{2j}}{2} \tag{32}$$

where $\overline{w}_{1j}$ is the average output when factor j is $+1$; and $\overline{w}_{2j}$ is the average output when factor j is -1.

Furthermore, (30) simplifies the covariance matrix (15) to

$$\mathbf{cov}(\hat{\boldsymbol{\beta}}) = (n\mathbf{I})^{-1}\sigma_w^2 = \mathbf{I}\frac{\sigma_w^2}{n}. \tag{33}$$

So all estimators have the same variance σ_w^2/n, and they are independent. In 1952, Box ([3]) proved that the variances of $\widehat{\beta}_j$ are minimal if $\mathbf{X}$ is orthogonal.

Altogether, 2^k designs have many attractive properties. Unfortunately, the number of combinations ($n = 2^k$) grows exponentially with the number of factors (k). At the same time, the number of effects is only $q = k + 1$, so these designs become inefficient for high values of k. The solution is designs that require only a fraction of these 2^k combinations.

5 Designs for first-order polynomials: Resolution-III

A *resolution-III design* gives unbiased estimators of the parameters of a first-order polynomial, assuming that a first-order polynomial is indeed a valid metamodel of the underlying (simulation) experiment; see [4]. These designs are also known as *Plackett-Burman* designs, published back in 1946; see [**?**]. Plackett-Burman designs have as a subclass *fractional factorial two-level* or 2^{k-p} designs. Obviously, the latter subclass has its number of combinations equal to a power of two; Plackett-Burman designs have their number of combinations equal to a multiple of four (for example, $n = 12$, which is not a power of two).

The simplest example of a 2^{k-p} design assumes $k = 3$. A 2^3 design would require $n = 8$ combinations. The number of parameters is only $q = k + 1 = 4$. A 2^{3-1} design requires only $n = 4$ combinations. Because this design has resolution-III, it is denoted as a 2_{III}^{3-1}design. The three columns denoted by **1**, **2**, and $\mathbf{3} = \mathbf{1.2}$ in Table 1 together give one of the two possible 2^{3-1} designs; the heading "Combi." stands for 'Factor combination'; the heading '$\mathbf{3} = \mathbf{1.2}$' is a shorthand notation for $x_{i3} = x_{i1}x_{i2}$ with $i = 1, \ldots n$. Hence, the first element ($i = 1$) in the column $\mathbf{3} = \mathbf{1.2}$ is $x_{13} = x_{11}x_{12} = (-1)(-1) = +1$ so the entry is a plus (+). It is easy to verify that Table 1 gives an orthogonal $\mathbf{X}$. The design is also balanced. The DOE literature calls '$\mathbf{3} = \mathbf{1.2}$' a design *generator* (also see the next section).

Combi.	1	**2**	**3** $= 1.2$	**3** $= -1.2$
1	$-$	$-$	$+$	$-$
2	$+$	$-$	$-$	$+$
3	$-$	$+$	$-$	$+$
4	$+$	$+$	$+$	$-$

Table 1
Two fractional-factorial two-level designs for three factors

An alternative 2^{3-1} design is formed by the three columns denoted by **1**, **2**, and **3** $= -$**1.2** in Table 1; obviously, '**3** $= -$**1.2**' stands for $x_{i3} = -x_{i1}x_{i2}$. This design belongs to the same *family* as the design with generator **3** $=$ **1.2**. The choice between these two designs is *arbitrary* (random).

The next simplest example of a 2^{k-p} design is a design with $n = 2^3 = 8$ combinations. The number of factors follows from $2^{k-p} = 8$ or $k - p = 3$ with positive integers k and p, and $2^{k-p} > k$. The solution is $k = 7$ and $p = 4$. This gives the analogue of Table 1, now with the generators **4** $=$ **1.2**, **5** $=$ **1.3**, **6** $=$ **2.3**, and **7** $=$ **1.2.3**. This design belongs to a family formed by substituting a minus sign for the (implicit) plus sign in one or more generators; for example, substituting **4** $= -$ **1.2** for **4** $=$ **1.2** gives one other member of the family. All the 128 family members together form the unique full-factorial two-level 2^7 design.

Table 1 gives two *saturated* designs for three factors; i.e., the number of combinations equals the number of parameters to be estimated: $n = q$ in (8). Hence, no degrees of freedom are left in the MSR in (18), so the lack-of-fit F-test in (26) cannot be applied. This problem can be solved easily: select one or more combinations from another member of the family, and also simulate this combination; the easiest selection is random.

Intermediate k *values* such as $4 \leq k \leq 6$ can be handled easily: for $k = 4$ delete three columns (for example, the last three columns) of the 2^{7-4} design; for $k = 5$ delete two columns; for $k = 6$ delete one column. Obviously, the resulting designs are not saturated anymore. (Of course, the analysts may also add one or more extra factors to their original list with $4 \leq k \leq 6$ factors; these extra factors do not require a bigger experiment: n remains eight.)

The next example has $n = 2^{k-p} = 16$. So a saturated design for a first-order polynomial implies $k = 15$. Hence $k - p = 4$ implies $p = 15 - 4 = 11$. The construction of this 2^{15-11} design remains quite simple:

(1) Construct the (full factorial two-level) 2^4 design; i.e., write down the 16×4 design matrix.

(2) Add all (that is, $4 \times (4-1)/2 = 6$) pairwise generators: **5** $=$ **1.2**, **6** $=$ **1.3**,

7 = **1**.**4**, ..., **10** = **3**.**4**.

(3) Add the following four triplet generators: **11** = **1**.**2**.**3**, **12** = **1**.**2**.**4**, **13** = **1**.**3**.**4**, **14** = **2**.**3**.**4**.

(4) Add the following quadruple generator: **15** = **1**.**2**.**3**.**4**.

A last example has $n = 32$. So a saturated design implies $k = 31$. Hence $k - p = 5$ (so $2^5 = 32$) implies $p = 31 - 5 = 26$. The construction of this 2^{31-26} design remains quite simple—but tedious. A computer procedure is therefore helpful. It is simple to write such a procedure. Also see [41], p. 366 for a different procedure (based on so-called Walsh functions).

As mentioned above, Plackett-Burman designs have 2^{k-p} designs as a subclass. *Plackett-Burman designs in the narrow sense* have their number of combinations equal to a multiple of four, but not a power of two. Actually those two authors published such designs for $12 \leq n \leq 96$; also see ([18], pp. 332-333) and [34], p. 170. Plackett-Burman designs are again balanced and orthogonal.

6 Regression analysis: factor interactions

Interaction means that the effect of one factor depends on the levels of one or more other factors; i.e., $E\left(w|\, x_j = -1\right) - E\left(w|\, x_j = +1\right) = f(x_{j'})$ with $j \neq j'$ $(j, j' = 1, \ldots, k)$. If the I/O function is continuous, then $\partial E(w)/\partial dx_j = f(x_{j'})$ with $j \neq j'$. Interaction implies that the response curves with $E\left(w|\, x_j, x_{j'} = c\right)$ versus x_j are not parallel for different c values. If the interaction between two factors is positive, the factors are called *complementary*; if this interaction is negative, the factors are *substitutes* for each other. Augmenting the first-order polynomial in (27) with two-factor (also called two-way or pairwise) interactions yields

$$E(y) = \beta_0 + \sum_{j=1}^{k} \beta_j x_j + \sum_{j=1}^{k-1} \sum_{j'=j+1}^{k} \beta_{j;j'} x_j x_{j'}. \tag{34}$$

The total number of interactions is $k(k-1)/2$, so the total number of parameters is $q = 1 + k + k(k-1)2 = 1 + k(k+1)/2$. The formulation of $\mathbf{X}$ (matrix of explanatory variables) for the metamodel (34) follows straightforwardly from $\mathbf{D}$ (design matrix):

$$\mathbf{X} = (x_{ij}) = (1, d_{i1}, \ldots, d_{ik}, d_{i1}d_{i2}, \ldots, d_{i;k-1}d_{ik})\ (i = 1, \ldots, n)\ . \tag{35}$$

A first-order polynomial may not give a valid metamodel, whereas augmenting this polynomial with two-factor interactions may give an adequate approximation. An example is the Flexible Manufacturing System (FMS) case study in [29].

The ANOVA (ANalysis Of VArinance) literature uses higher-order interactions, for example, three-factor interactions:

$$E(y) = \beta_0 + \sum_{j=1}^{k} \beta_j x_j + \sum_{j=1}^{k-1} \sum_{j'=j+1}^{k} \beta_{j;j'} x_j + \sum_{j=1}^{k-2} \sum_{j'=j+1}^{k-1} \sum_{j''=j'+1}^{k} \beta_{j;j'j''} x_j x_{j'} x_{j''}. \quad (36)$$

However, high-order interactions are hard to interpret, and are often unimportant in practice. This tutorial therefore assumes that interactions among three or more factors are unimportant. Of course, this assumption should be checked; see the 'lack of fit' and 'validation' discussed in this article.

7 Designs allowing two-factor interactions: Resolution-IV

A *resolution-IV design* gives unbiased estimators of the parameters of a first-order polynomial, even if two-factor interactions are non-zero. Back in 1951, Box and Wilson [6] proved the *foldover* theorem, which may be reformulated as follows (the full theorem is quoted in [18], p. 343): If a resolution-III design (say) $\mathbf{D}_{III}$ is augmented with its 'mirror' design $-\mathbf{D}_{III}$, then the resulting design is a resolution-IV design. So the price for proceeding from a resolution-III to a resolution-IV design is that n (number of combinations simulated) doubles. The foldover gives unbiased estimators of the first-order (or main) effects, but does not always enable unbiased estimation of the individual two-factor interactions.

Consider the following example with $k = 7$ factors. Combining a 2_{III}^{7-4} design with its mirrored design gives a design with $n = 16$ combinations, namely, a 2_{IV}^{7-3} design. So $\mathbf{X}$ corresponding with the regression model (34) has $n = 16$ rows and $q = 1 + 7(7+1)/2 = 29$ columns, so this $\mathbf{X}$ is collinear (since $n < q$). Hence, LS estimation of the 29 individual regression parameters is impossible. It is possible, however, to compute the LS estimator of the intercept and the seven first-order effects. For example, it is easy to verify that the column for the interaction between the factors 6 and 7 is orthogonal to the columns for the first-order effects of the factors 6 and 7; also see (40). Obviously, the 2_{IV}^{7-3} design remains balanced.

Useful manipulations with the generators (such as $\mathbf{3} = \mathbf{1}.\mathbf{2}$ in the 2_{III}^{3-1} design of Table 1) are explained in [18] and [23].These manipulations show how estimated effects are *confounded* or *aliased*; for example, it is easy to prove that the generator $\mathbf{3} = \mathbf{1}.\mathbf{2}$ implies $E(\widehat{\beta}_1) = \beta_1 + \beta_{2;3}$, $E(\widehat{\beta}_2) = \beta_2 + \beta_{1;3}$, and (of course) $E(\widehat{\beta}_3) = \beta_3 + \beta_{1;2}$; in other words, only if $\beta_{2;3} = 0$, the estimator $\widehat{\beta}_1$ is unbiased, etc. But resolution-III designs indeed assume that all interactions are zero!

It can be shown that adding the mirror design to a resolution-III design for k factors gives a resolution-IV design for $k+1$ factors (with $n_{IV} = 2n_{III}$ and n_{III} a multiple of four, possibly a power of two). For example, $k = 11$ requires a Plackett-Burman (resolution-III) design with $n_{III} = 12$ combinations, so a resolution-IV design with $n_{IV} = 24$ combinations enables the estimation of $k = 12$ main effects unbiased by two-factor interactions.

So, the construction of resolution-IV designs is easy, once a resolution-III design is available. A $\mathbf{D}_{III}$ design (a Plackett-Burman design) is simply augmented with its mirror design, $-\mathbf{D}_{III}$. For the Plackett-Burman subclass of $2_{III}^{(k-1)-p}$ designs, the 2_{IV}^{k-p} designs may be constructed by first defining the full-factorial design in $k-p$ factors, and then aliasing the remaining p factors with high-order interactions among these first $k-p$ factors. For example, $k = 8$ and $n = 16 = 2^4$ leads to a 2^{8-4} design, so first a 2^4 design in (say) the first four factors is written down; next, the following main generators may be used: $\mathbf{5} = \mathbf{1}.\mathbf{2}.\mathbf{3}$, $\mathbf{6} = \mathbf{1}.\mathbf{2}.\mathbf{4}$, $\mathbf{7} = \mathbf{2}.\mathbf{3}.\mathbf{4}$, and $\mathbf{8} = \mathbf{1}.\mathbf{2}.\mathbf{3}.\mathbf{4}$. (It is easy to prove that the main effect estimators are unbiased, and the $8(8-1)/2 = 28$ two-factor interactions are confounded in seven groups with three interactions per group—assuming that all high-order interactions are zero.)

The resolution-IV designs discussed so far imply that the number of combinations increases with jumps of eight ($n_{IV} = 8,16,24,32,40, ...$), because the underlying resolution-III designs have a number of combinations that jump with four ($n_{III} = 4, 8, 12, 16, 20, ...$). Webb [50] derived resolution-IV designs with number of combinations that increase in smaller jumps: $n_{IV} = 2k$ where k does not need to be a multiple of four. He also used the foldover theorem. Also see [18], pp.344-348

This section on resolution-IV designs is concluded with a general discussion of confounding. Suppose that a valid linear regression metamodel is

$$E(w) = E(y) = \mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{X}_2\boldsymbol{\beta}_2. \tag{37}$$

An example is an $\mathbf{X}_1$ corresponding with the intercept and the main effects collected in $\boldsymbol{\beta}_1$, and an $\mathbf{X}_2$ corresponding with the two-factor interactions $\boldsymbol{\beta}_2$. Suppose that the analysts use the simple metamodel without these interactions. Then they estimate the first-order polynomial coefficients through

$$\widehat{\boldsymbol{\beta}}_1 = (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'\mathbf{w}. \tag{38}$$

So (38) gives

$$E(\widehat{\boldsymbol{\beta}}_1) = (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'E(\mathbf{w}). \tag{39}$$

Substitution of (37) into (39) gives

$$E(\widehat{\boldsymbol{\beta}}_1)=(\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'(\mathbf{X}_1\boldsymbol{\beta}_1 + \mathbf{X}_2\boldsymbol{\beta}_2) = \boldsymbol{\beta}_1 + (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'\mathbf{X}_2\boldsymbol{\beta}_2. \tag{40}$$

k	n	generators
5	$2_V^{5-1} = 16$	$\mathbf{5} = \mathbf{1.2.3.4}$
6	$2_{VI}^{6-1} = 32$	$\mathbf{6} = \mathbf{1.2.3.4.5}$
7	$2_{VII}^{7-1} = 64$	$\mathbf{7} = \mathbf{1.2.3.4.5.6}$
8	$2_V^{8-2} = 64$	$\mathbf{7} = \mathbf{1.2.3.4}$; $\mathbf{8} = \mathbf{1.2.5.6}$
9	$2_{VI}^{9-2} = 128$	$\mathbf{9} = \mathbf{1.4.5.7.8}$; $\mathbf{10} = \mathbf{2.4.6.7.8}$
10	$2_V^{10-3} = 128$	$\mathbf{8} = \mathbf{1.2.3.7}$; $\mathbf{9} = \mathbf{2.3.4.5}$; $\mathbf{10} = \mathbf{1.3.4.6}$
11	$2_V^{11-4} = 128$	$\mathbf{8} = \mathbf{1.2.3.7}$; $\mathbf{9} = \mathbf{2.3.4.5}$; $\mathbf{10} = \mathbf{1.3.4.6}$; $\mathbf{11} = \mathbf{1.2.3.4.5.6.7}$

Table 2
Generators for fractional-factorial two-level designs of resolutionV and higher (VI, VII)

This gives an unbiased estimator of $\boldsymbol{\beta}_1$ if either $\boldsymbol{\beta}_2 = \mathbf{0}$ or $\mathbf{X}_1'\mathbf{X}_2 = \mathbf{0}$. Indeed, resolution-III designs assume that $\boldsymbol{\beta}_1 = \mathbf{0}$ where $\boldsymbol{\beta}_2$ consists of the two-factor interactions; resolution-IV designs ensure that $\mathbf{X}_1'\mathbf{X}_2 = \mathbf{0}$ (the two-factor interaction columns are orthogonal to the main effects and intercept columns).

8 Designs for two-factor interactions: Resolution-V

A *resolution-V* design enables LS estimation of the parameters of a first-order polynomial plus its two-factor interactions. For example, a 2_V^{8-2} design (so $n = 64$) enables LS estimation of the $q = 37$ regression parameters. Such a design has two generators. To avoid aliasing among the relevant effects (namely, the two-factors interactions, the main effects, and the intercept), these generators should multiply more than two factors; for example, it is easy to derive that a good choice is $\mathbf{7} = \mathbf{1.2.3.4}$ and $\mathbf{8} = \mathbf{1.2.5.6}$ (these generators imply confounding of two-factor interactions with interactions among three or more factors—the latter high-order interactions are assumed to be zero).

In general, the first-order polynomial augmented with all the two-factor interactions implies that q (number of regression parameters) becomes $1+k+k(k-1)/2 = 1 + (k^2 + k)/2$, so the number of parameters becomes order k^2 and many more combinations need to be simulated compared with a first-order polynomial. In 1961, Box and Hunter [5] published a table with generators for 2^{k-p} designs of resolution V and higher; their table is reproduced in [18], p. 349, and in Table 2.

Recently Sanchez and Sanchez [41] published a computer procedure for constructing resolution-V designs for $k \leq 120$; for example, a $2_V^{120-105}$ design. Unfortunately, 2^{k-p} designs—except for the 2_V^{5-1} design in Table 2—require

Effect type	Generator
Intercept	$(-1, \ldots, -1)$ for all k factors
Main effect	$(-1, +1, \ldots, +1)$ for all k factors
Two-factor Interaction	$(1, 1, -1, \ldots, -1)$ for $k > 3$ factors

Table 3
Generators for Rechtschaffner's resolution-V designs

Combi.	Generator	**1**	**2**	3	**4**
1	$(-1, \ldots, -1)$	-1	-1	-1	-1
2	$(-1, +1, \ldots, +1)$	-1	$+1$	$+1$	$+1$
3		$+1$	-1	$+1$	$+1$
4		$+1$	$+1$	-1	$+1$
5		$+1$	$+1$	$+1$	-1
6	$(+1, +1, -1, \ldots, -1)$	$+1$	$+1$	-1	-1
7		$+1$	-1	$+1$	-1
8		$+1$	-1	-1	$+1$
9		-1	$+1$	$+1$	-1
10		-1	$+1$	-1	$+1$
11		-1	-1	$+1$	$+1$

Table 4
Rechtschaffner's design for four factors

relatively many combinations to estimate the regression parameters. For example, the 2^{9-2}_{VI} design in Table 2 requires 128 combinations, to estimate $q = 1 + 9(9+1)/2 = 46$ parameters, so its 'efficiency' is only $46/128 = 0.36$; and the $2^{120-105}_{V}$ design requires $n = 32,768$ whereas $q = 7,261$ so its efficiency is only $7261/32768 = 0.22$. There are resolution-V designs that require fewer runs; see [41], pp. 372-373.

Actually, if a simulation run takes much computer time, then *saturated* designs are attractive. In 1967, Rechtschaffner [39] published simple saturated non-orthogonal fractions of two-level (and three-level) designs; see Table 3 (and also [18], p. 352). Their construction is simple: the *generators* are permuted in the different factor combinations; see the design for $k = 4$ factors in Table 4 and for $k = 5$ factors in [18], p. 352. An application of these designs is presented in [26], involving $k = 6$ factors and using Rechtschaffner's design, which implies only $n = q = 1 + 6 + 6(6-1)/2 = 22$ combinations.

9 Regression analysis: second-order polynomials

The classic mathematical Taylor series argument implies that—as the experimental area gets bigger or the I/O function gets more complicated—a better metamodel may be a *second-order polynomial*. An example is the M/M/1 simulation: a valid metamodel for the I/O behavior in an area with relatively high traffic rate x may be

$$E(y) = \beta_0 + \beta_1 x + \beta_2 x^2. \tag{41}$$

Obviously, estimation of the three parameters in (41) requires at least the simulation of *three* input values. Indeed, practitioners often use a one-factor-at-a-time design with three values per factor (they even do so, when fitting a first-order polynomial). DOE also provides designs with three values per factor; for example, 3^k designs. However, more popular in simulation are Central Composite Designs (CCDs), which have five values per factor (see Section 10 below).

Note: The second-order polynomial in (41) is nonlinear in $\mathbf{x}$ (explanatory regression variables), but linear in $\boldsymbol{\beta}$ (regression parameters). Consequently, such a metamodel remains a linear regression model, which was specified in (8).

The formula for the general second-order polynomial in k factors is

$$E(y) = \beta_0 + \sum_{j=1}^{k} \beta_j x_j + \sum_{j=1}^{k} \sum_{j' \geq j}^{k} \beta_{j;j'} x_j x_{j'}. \tag{42}$$

So this metamodel adds k *purely quadratic* effects $\beta_{j;j}$ to (34). In practice, second-order polynomials are applied either locally or globally. *Local* fitting may be used when searching for the optimum input combination; see [2]. *Global* fitting (for $0 < x < 1$ in the queueing simulation) using second-order polynomials has been applied, but Kriging provides better metamodels; see [47].

10 Designs for second-degree polynomials: Central composite designs

A *CCD* augments a resolution-V design such that the purely quadratic effects can also be estimated. More specifically, a CCD adds the *central* point and $2k$ *axial* points that form a *star design*, where—in the coded factors—the central point is $(0, \ldots 0)'$, and the 'positive' axial point for factor j (with $j = 1, \ldots, k$) is the point with $x_j = +c$ and all other $k - 1$ factors fixed at the center (so $x_{j'} = 0$ with $j' = 1, \ldots, k$ and $j' \neq j$) and the 'negative' axial point for factor

j is the point with $x_j = -c$ and $x_{j'} = 0$ (so the axial points are a one-at-a-time design).

Selecting $c = k^{1/2}$ results in a *rotatable* design; that is, this design gives a *constant* variance for the predicted output at a *fixed* distance from the origin (so the contour functions are circles).

A CCD does not give an orthogonal $\mathbf{X}$; hence, the estimated parameters of the second-degree polynomial are correlated.

Furthermore, $n_{CCD} = n_V + 1 + 2k$ where n_{CCD} denotes the total number of combinations in a CCD; for example, $k = 2$ implies $n_{CCD} = 2^2 + 1 + 2 \times 2 = 9$. For $k = 120$, [41] implies $n_{CCD} = 32,768 + 1 + 2 \times 120 = 33,009$. Often only the central point is replicated, to estimate the common variance and to compute the lack-of-fit F-statistic defined in (26). CCDs are further discussed in [34] and [36].

Obviously, CCDs are rather inefficient. Therefore, [**?**] simulated only half of the star design. Classic resolution-V designs are very inefficient, whereas Rechtschaffner's designs are saturated. Finally, [21], pp. 314-316 discusses three other types of saturated designs for second-order polynomials (due to Koshall, Scheffé, and Notz respectively), but there seem to be no simulation applications of these designs.

11 Validation

Section 3 included the lack-of-fit F-test, which assumes white noise. This section drops this assumption, and presents the following statistics: R^2 and $R^2_{adjusted}$, Pearson's and Spearman's correlation coefficients, and cross-validation. These statistics may be computed for both deterministic and random simulation, and for other metamodels than linear regression models; for example, Kriging and neural networks.

11.1 Coefficients of determination and correlation coefficients

R^2 is a very popular statistic in the analysis of passively observed real systems; in active experimentation including replication, the lack-of-fit F-statistic is more popular. Whether or not replications are available, R^2 may be defined as follows (also see, for example, [11], p. 33):

$$R^2 = \frac{\sum_{i=1}^{n}(\widehat{y}_i - \overline{\overline{w}})^2}{\sum_{i=1}^{n}(\overline{w}_i - \overline{\overline{w}})^2} = 1 - \frac{\sum_{i=1}^{n}(\widehat{y}_i - \overline{w}_i)^2}{\sum_{i=1}^{n}(\overline{w}_i - \overline{\overline{w}})^2} \tag{43}$$

where $\widehat{y}_i$ denotes the metamodel predictor defined in (10), $\overline{w}_i$ denotes the simulation response of combination i averaged over its $m_i \geq 1$ replicates defined in (24), and $\overline{\overline{w}} = \sum_{i=1}^{n} \overline{w}_i/n$ denotes the overall average simulation response. The right-most equality in (43) shows that $R^2 = 1$ if $\widehat{y}_i = \overline{w}_i$ for all i-values. R^2 measures how much of the variation in the simulation response is explained by the regression model; see the denominator in (43), which is the numerator of the classic variance estimator computed over the n combinations—analogous to (23).

Note: R^2 is not defined as a function of w_{ir} (individual outputs per combination), because the metamodel is valid if it adequately predicts the *expected* output of the simulation model. Defining R^2 as a function of the individual outputs would decrease the value of R^2 because of the large variability of the simulation output per combination.

R^2 may also be used in deterministic simulation. In such a type of simulation, the analysts do not obtain any replicates so in (43) $\overline{w}_i$ becomes w_i and $\overline{\overline{w}}$ becomes $\overline{w}$.

If $n = q$ (no degrees of freedom left; saturated design), then $R^2 = 1$. This high value is misleading. Therefore R^2 *adjusted* for the number of explanatory variables is defined as follows:

$$R^2_{adjusted} = 1 - \frac{n-1}{n-q}(1 - R^2). \tag{44}$$

Hence, if $q = 1$, then $R^2_{adjusted} = R^2$.

Lower critical values for either R^2 or $R^2_{adjusted}$ are unknown, because these statistics do not have well known distributions. Analysts therefore use subjective lower thresholds. Recently, [24] demonstrated how the distributions of these two statistics can be obtained through *bootstrapping* (bootstrapping was invented by [12]; the classic textbook on bootstrapping is [13].)

R^2 is also called the *multiple correlation coefficient.* However, R^2 should be distinguished from *Pearson's correlation coefficient*—usually denoted by ρ. As any statistics textbook explains, this ρ measures the strength of the linear relationship between two random variables (say) x and w (in the classic designs presented in the preceding sections, x is deterministic; however, Latin Hypercube Sampling or LHS indeed samples x; see [1] [7], [15], [42]). Like R^2, the statistic ρ ranges between -1 and $+1$. A value of $+1$ implies that the two variables are related perfectly by an increasing (positive slope) linear relationship. A value of -1 implies a perfect, decreasing linear relationship—as is explained next.

Assume that the (vector) random variable (x, w) is *Bivariate Normally Independently Distributed* with parameters $E(x) = \mu_x$, $E(w) = \mu_w$, $var(x) = \sigma_x^2$, $var(w) = \sigma_w^2$, and $cor(x, w) = \rho(x, w) = \rho$ (so $cov(x, w) = \rho\sigma_x\sigma_w$):

$$(x, w) \backsim NID_2(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \text{ with } \boldsymbol{\mu} = (\mu_x, \mu_w)', \, \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_x^2 & \rho\sigma_x\sigma_w \\ \rho\sigma_x\sigma_w & \sigma_w^2 \end{bmatrix}$$

where the subscript 2 in NID_2 means that it concerns a *two*-dimensional vector variable. It can be derived that

$$E\left(w|\, x\right) = \beta_0 + \beta_1 x \text{ with } \beta_0 = \mu_w - \beta_1\mu_x \text{ and } \beta_1 = \rho\frac{\sigma_w}{\sigma_x}. \tag{45}$$

The parameters μ_x, μ_w, σ_x^2, and σ_w^2 can be estimated in the classic way, analogous to (24) and (23) respectively. The covariance is then estimated through

$$\widehat{cov}(x, w) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(w_i - \overline{w})}{n-1}, \tag{46}$$

so the correlation is estimated through

$$\widehat{\rho}(x, w) = \widehat{\rho} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(w_i - \overline{w})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(w_i - \overline{w})^2}}. \tag{47}$$

If $\rho = 0$, then x and w are independent (zero correlation does not imply independence for non-normally distributed variables!). To test $H_0 : \rho = 0$, the classic Student t distribution can be used:

$$t_{n-2} = \frac{\widehat{\rho}}{\sqrt{1 - \widehat{\rho}^2}}\sqrt{n-2}. \tag{48}$$

Note that when the simulation's input x and output w are uncorrelated, then x may be called an unimportant factor in this simulation experiment: $\beta_1 = 0$ if $\rho = 0$, because $\beta_1 = \rho\sigma_w/\sigma_x$; see (45)). If the input x is not sampled, but fixed through one of the classic designs presented in the preceding sections, then the t statistic defined in (17) can be applied; also see the null-hypothesis in (19).

It may happen that the two variables x and w are related, but not through the linear relationship $E\left(w|\, x\right) = \beta_0 + \beta_1 x$ in (45). An alternative relationship may be (say) $E\left(w|\, x\right) = \beta_0 x^{\beta_1}$. Such an increasing monotonic relationship may be quantified through *Spearman's rank correlation coefficient* (say) η. This coefficient is Pearson's coefficient computed—not from the original pairs (x_i, w_i)—but from the ranked pairs $(r(x_i), r(w_i))$, as follows:

(1) The smallest value of x_i is assigned a rank of $1, \ldots,$ the largest value gets rank n.
(2) If a tie occurs, then the average rank is assigned to the members of that tie.
(3) The ranks for w_i are computed in the same manner as for x_i.

To test the null-hypothesis $H_0 : \eta = 0$, Table A10 in Conover's excellent textbook [8] can be used. If $n \geq 30$, then this hypothesis may also be tested through $z = \widehat{\eta}\sqrt{n-1}$ with $z \backsim N(0,1)$; again see [8], p. 456. More details on the use of ρ and τ for identifying important factors in simulation (not for quantifying the adequacy of a metamodel) are given in [25].'

11.2 Cross-validation

Cross-validation is applied not only in linear regression analysis, but also in non-linear regression, Kriging, neural networks, etc. The basic idea of cross-validation is quite old; see [46]. This tutorial focuses on so-called *leave-one-out cross-validation*. Assume that $\mathbf{X}_i$ has only n rows (not $N = \sum_{i=1}^{n} m_i$ rows); i.e., assume that the number of replicates is constant, possibly one: $m_i = m \geq 1$. If the number of replicates is indeed a constant ($m > 1$), then the LS estimate may replace w_{ir} (individual simulation output for combination i) by $\overline{w}_i$ (average simulation output for combination i). However, if $m_i > 1$ and $m_i \neq m$ (different replication numbers), then the white noise assumption implies $var(\overline{w}_i) = \sigma_w^2/m_i$; in other words, the variance of $\overline{w}_i$ is not constant. In case of such variance heterogeneity, the LS formulas need correction; see [30], p. 157. The procedure runs as follows.

(1) Delete I/O combination i from the complete set of n combinations, to obtain the remaining I/O data set $(\mathbf{X}_{-i}, \overline{\mathbf{w}}_{-i})$. Assume that this step results in a non-collinear matrix $\mathbf{X}_{-i}$ $(i = 1, \ldots, n)$; see (49) below. To satisfy this assumption, the original matrix $\mathbf{X}$ must satisfy the condition $n > q$. Counter-examples are saturated designs; a simple solution is to simulate one more combination, for example, the center point if the original design is not a CCD.
(2) Recompute the LS estimator of the regression parameters from the remaining I/O data:

$$\hat{\boldsymbol{\beta}}_{-i} = (\mathbf{X}'_{-i}\mathbf{X}_{-i})^{-1}\mathbf{X}'_{-i}\mathbf{w}_{-i}. \tag{49}$$

(3) Use this recomputed estimator $\hat{\boldsymbol{\beta}}_{-i}$ to compute the regression prediction for the combination deleted in step 1:

$$\widehat{y}_{-i} = \mathbf{x}'_i\hat{\boldsymbol{\beta}}_{-i}. \tag{50}$$

(4) Repeat the preceding three steps, until all n combinations have been processed. This results in n predictions $\widehat{y}_{-i}$ $(i = 1, \ldots, n)$.
(5) Use a scatter plot with the n pairs $(w_i, \widehat{y}_{-i})$ to judge whether the metamodel is valid.

Case studies using this cross-validation procedure are presented in [48] and [49].

The following alternative for the subjective judgment in step 5 is proposed in [20]: Compute

$$t_{m-1}^{(i)} = \frac{\overline{w}_i - \widehat{y}_{-i}}{\sqrt{\widehat{var}(\overline{w}_i) + \widehat{var}(\widehat{y}_{-i})}} \; (i = 1, \ldots, n) \tag{51}$$

where $\widehat{var}(\overline{w}_i) = \widehat{var}(w_i)/m$ (and $\widehat{var}(w_i)$ was given in (23)) and $\widehat{var}(\widehat{y}_i)$ follows from (50) and the analogue of (14):

$$\widehat{var}(\widehat{y}_{-i}) = \mathbf{x}_i'\widehat{cov}(\widehat{\boldsymbol{\beta}}_{-i})\mathbf{x}_i \tag{52}$$

where

$$\widehat{cov}(\widehat{\boldsymbol{\beta}}_{-i}) = \widehat{var}(\overline{w}_i)(\mathbf{X}_{-i}'\mathbf{X}_{-i})^{-1}. \tag{53}$$

Note that $\overline{w}_i$ and $\widehat{y}_{-i}$ are independent because $\widehat{y}_{-i}$ does not use $\overline{w}_i$. Since (51) gives n values (because $i = 1, \ldots, n$), the regression metamodel is rejected if

$$\max_i t_{m-1}^{(i)} > t_{m-1;1-[\alpha/(2n)]} \tag{54}$$

where the right-hand side follows from *Bonferroni's inequality*, which implies that the classic type-I error rate (in this case $\alpha/2$) is replaced by the same value divided by the number of tests (in this case n).

There is a *shortcut* for the n computations in the cross-validation procedure given above; modern software applies this shortcut. The technique uses the so-called *hat matrix* $\mathbf{H}$ (see, for example, [33], pp. 201-202, and also [30], pp. 156-157):

$$\mathbf{H} = (\mathbf{h}_{ii'}) = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \text{ with } i, i' = 1, \ldots, n. \tag{55}$$

It can be proven that the numerator of (51) can be written as

$$\overline{w}_i - \widehat{y}_{-i} = \frac{\overline{w}_i - \widehat{y}_i}{1 - h_{ii}}$$

and (51) itself can be written as

$$t_{m_i-1} = \frac{\overline{w}_i - \widehat{y}_i}{\sqrt{\widehat{var}(\overline{w}_i)}\sqrt{1 - h_{ii}}} \; (i = 1, \ldots, n) \tag{56}$$

so the cross-validation computations can be based solely on the *original* I/O data, $(\mathbf{X}, \mathbf{w})$, which give $\widehat{y}_i$ and h_{ii}.

In *deterministic* simulation, the statistic defined in (54) should not be applied, for the following reason. Obviously, the term $\widehat{var}(\overline{w}_i)$ in (51) may be set to zero. The term $\widehat{var}(\widehat{y}_{-i})$ may be computed from (52) where (53) uses the factor $\widehat{var}(\overline{w}_i)$, which may now be computed from the MSR in (18). But the worse the metamodel fits, the bigger this MSR gets—so the smaller the test statistic in (51) becomes, so the smaller the probability of rejecting this false metamodel becomes! Therefore the analysts may compute the *relative* prediction errors $\widehat{y}_{-i}/w_i$, and decide whether they find these errors acceptable—practically speaking. An alternative remains the scatter plot described in Step 5 of the cross-validation procedure above.

Cross-validation not only affects the regression predictions $(\widehat{y}_{-i})$, but also the estimated regression parameters $(\hat{\boldsymbol{\beta}}_{-i})$; see (49). So the analysts may be interested not only in the predictive performance of the metamodel, but also in its *explanatory* performance; see the FMS case study in [29].

The regression literature proposes several so-called *diagnostic* statistics that are related to (56); for example, PRESS, DEFITS, DFBETAS, and Cook's D; see [30], p. 157.

12 Conclusions and further research

This tutorial explained the basics of linear regression models—especially low-order polynomials—and the corresponding statistical designs—namely, designs of resolution III, IV, V, and CCDs. The tutorial assumed white noise, meaning that the residuals of the fitted linear regression models are normally, independently, and identically distributed (NIID) with zero mean. The white noise assumption is dropped in Chapter 3 of [29], explaining the consequences.

References

[1] Alam, F.M., K.R. McNaught, and T.J. Ringrose (2003), A comparison of experimental designs in the development of a neural network simulation metamodel. *Simulation Modelling Practice and Theory*, 12, issues 7-8, pp. 559-578

[2] Angün, E., G. Gürkan, D. den Hertog, 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.

[3] Box, G.E.P. (1952), Multi-factor designs of first order. *Biometrika*, 39, no. 1, pp. 49-57

[4] Box, G.E.P. and J.S. Hunter (1961), The 2^{k-p} fractional factorial designs, Part I. *Technometrics*, 3, pp. 311-351

[5] Box, G.E.P. and J.S. Hunter (1961), The 2^{k-p} fractional factorial designs, Part II. *Technometrics*, 3, pp. 449-458

[6] Box, G.E.P. and K.B. Wilson (1951), On the experimental attainment of optimum conditions. Journal Royal Statistical Society, Series B, 13, no. 1, pp. 1-38

[7] Chen, V.C.P., K.-L. Tsui, R. R. Barton and J.K. Allen (2003), A review of design and modeling in computer experiments. In: *Handbook of Statistics; Volume 22*, edited by R. Khattree and C.R. Rao, Elsevier, Amsterdam, pp. 231-261

[8] Conover, W.J. (1980), *Practical non-parametric statistics: second edition.* Wiley, New York

[9] Conway,R.W. (1963), Some tactical problems in digital simulation. *Management Science*, 10, no 1, pp. 47-61

[10] Downing, D.J. , R.H. Gardner and F.O. Hoffman (1985), An examination of Response-Surface Methodologies for uncertainty analysis in assessment models. *Technometrics*, 27, no. 2, pp. 151-163 (Discussion: 1986, 28, no. 1, pp. 91-93)

[11] Draper, N.R. and H. Smith (1981),*Applied regression analysis; second edition.* Wiley, New York

[12] Efron, B. (1982), *The jackknife , the bootstrap and other resampling plans.* CBMS-NSF Series, Siam, Philadelphia

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

[14] Ho, Y. and X. Cao (1991), *Perturbation analysis of discrete event dynamic systems.* Kluwer, Dordrecht (Netherlands)

[15] Holland, W. (2003), @Risk version 4.5 Pro. *OR/MS Today*, 30, no. 1, pp. 52-55

[16] Hussain, M.F., R.R. Barton, and S.B. Joshi (2002), Metamodeling: radial basis functions versus polynomials. *European Journal of Operational Research*, 138, no.1, pp. 142-154

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

[18] Kleijnen, J.P.C. (1974/1975), *Statistical techniques in simulation (in two parts).* Marcel Dekker, New York (Russian translation, Publishing House 'Statistics', Moscow, 1978)

[19] Kleijnen, J.P.C. (1975), A comment on Blanning's metamodel for sensitivity analysis: the regression metamodel in simulation. *Interfaces*, 5, no. 3, pp. 21-23

[20] Kleijnen, J.P.C. (1983), Cross-validation using the t statistic. *European Journal of Operational Research*, 13, no. 2, pp. 133-141

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

[22] Kleijnen, J.P.C. (1995), Case study: statistical validation of simulation models. *European Journal of Operational Research*, 87, no. 1, pp. 21-34

[23] Kleijnen, J.P.C. (2007), *DASE: Design and analysis of simulation experiments.* Springer, Heidelberg, Germany

[24] Kleijnen, J.P.C. and D. Deflandre (2006), Validation of regression metamodels in simulation: Bootstrap approach.*European Journal of Operational Research*, 170, issue 1, pp. 120-131

[25] Kleijnen, J.P.C. and J. Helton (1999), Statistical analyses of scatter plots to identify important factors in large-scale simulations,1: review and comparison of techniques. *Reliability Engineering and Systems Safety*, 65, no. 2, pp. 147-185

[26] Kleijnen, J.P.C. and O. Pala (1999), Maximizing the simulation output: a competition. *Simulation*, 73, no. 3, pp. 168-173

[27] Kleijnen, J.P.C., S.M. Sanchez, T.W. Lucas, and T.M. Cioppa (2005), State-of-the-art review: a user's guide to the brave new world of designing simulation experiments. *INFORMS Journal on Computing* , 17, no. 3, pp. 263-289

[28] 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

[29] Kleijnen, J.P.C. and C. Standridge (1988), Experimental design and regression analysis: an FMS case study. *European Journal of Operational Research*, 33, no. 3, pp. 257-261

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

[31] Koehler, J.R. and A.B. Owen (1996), Computer experiments. *Handbook of statistics, volume 13*, edited by S. Ghosh and C.R. Rao, Elsevier, Amsterdam, pp. 261-308

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

[33] Miller, A.J. (1990, *Subset selection in regression.* Chapman and Hall, London

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

[35] Nelson, B.L. (2004), Stochastic simulation research in *Management Science. Management Science*, 50, no. 7, pp. 855-868

[36] NIST/SEMATECH (2006), *e-Handbook of statistical methods*, http://www.itl.nist.gov/div898/handbook/, 7 February 2006

[37] Olivi, L. (1984), editor, *Response surface methodology; handbook for nuclear reactor safety.* EUR 9600, Commission of the European Communities, Joint Research Centre, Ispra, Italy

[38] Plackett, R.L. and J.P. Burman (1946), The design of optimum multifactorial experiments. *Biometrika*, 33, pp. 305-325

[39] Rechtschaffner, R.L. (1967), Saturated fractions of 2^n and 3^n factorial designs. *Technometrics*, 9, pp. 569-575

[40] Rubinstein, R.Y.; A. Shapiro (1993), *Discrete-event systems: sensitivity analysis and stochastic optimization via the score function method.* Wiley, New York1993

[41] Sanchez, S.M. and P.J. Sanchez (2005), Very large fractional factorial and central composite designs. *ACM Transactions of Modeling and Computer Simulation*, 15, no. 4, pp. 362-377

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

[43] Searle, S.R. (1971), *Linear models.* John Wiley & Sons, Inc., New York

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

[45] Spall, J.C. (2003), *Introduction to stochastic search and optimization; estimation, simulation, and control.* Wiley, New York

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

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

[48] Van Groenendaal, W.J.H. (1998), *The economic appraisal of natural gas projects.* Oxford University Press, Oxford

[49] 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

[50] Webb, S. (1968), Non-orthogonal designs of even resolution. *Technometrics*, 10, pp. 291-299

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