*Proceedings of the 2007 Winter Simulation Conference*
*S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds.*

# 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 metamodels—especially low-order polynomials—and the corresponding statistical designs—namely, fractional factorial designs of resolution III (Plackett-Burman designs), IV (accounting for interactions), V (estimating individual interactions), and Central Composite Designs (CCDs, for second-order polynomial metamodels). This tutorial assumes 'white noise', which means that the residuals of the fitted linear regression metamodel are normally, independently, and identically distributed with zero mean. This metamodel requires validation. 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.

## 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; e.g., 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 contribution 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 minimization of the variances of these estimators requires a different design—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 Kleijnen 1975), 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.

The term 'response surface' is used for *local* metamodels in *Response Surface Methodology* (RSM). RSM was introduced by Box and Wilson (1951) as an iterative heuristic for optimizing real (physical) systems; also see the many references in Del Castillo (2007), Kleijnen (2007), and Myers and Montgomery (2002). This tutorial includes RSM designs for the optimization of simulated systems.

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 Conway (1963) and Nelson (2004). In deterministic simulation—where these tactical issues vanish— statisticians have been attracted to *strategic* issues, namely which scenarios to simulate and how to analyze the resulting data; see Santner, Williams, and Notz (2003). This tutorial fo-

**1-4244-1306-0/07/$25.00 ©2007 IEEE** 

cuses 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, 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 Kleijnen et al. (2005).

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 Law 2007).

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.

In this contribution, Greek letters 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 Zeigler et al. (2000). For example, the service rate $\mu$ in a single-server queueing simulation is estimated from the (say) $c$ observations on the service time $s$.

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 contribution reserves the term 'run' for a *simulation run*, which starts the simulation program in the initial conditions (e.g., the empty state in a queueing simulation) and stops the simulation program once a specific state has been reached (e.g., $c$ 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 Kleijnen (2007). That book adds many more mathematical and statistical details, alternative designs, case studies, and exercises to this article.

## 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 well-known M/M/1 queueing model. A popular performance measure (response variable, output) of *any* queueing simulation is

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

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.

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 (e.g.) Spall (2003). (The estimation of the gradient will be further discussed in Section 4.)

DASE, however, uses a *black-box* approach, which is also used by DOE for real-world experiments (see Myers and Montgomery 2002) and by Design and Analysis of Computer Experiments (DACE) for deterministic simulation experiments (see Santner et al. 2003). A black-box representation of any *single-server* simulation model with output $\overline{w}$ (average waiting time) and inputs $\lambda$ and $\mu$ (arrival and service rates) and $r_0$ (PRN seed)—and a fixed queueing discipline (e.g., FIFO), a fixed waiting room capacity, etc.—is

$$\overline{w} = w(\lambda, \mu, r_0) \tag{2}$$

where $w(.)$ denotes the mathematical function implicitly defined by the computer program that implements the simulation model.

One possible metamodel of the black box model in (2) is a *first-order polynomial* in the two input variables $\lambda$ and $\mu$:

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

where $y$ is the metamodel predictor of the simulation output $\overline{w}$ in (2); $\beta_0$, $\beta_1$,and $\beta_2$ are the parameters of this metamodel—which may be collected in the vector $\beta = (\beta_0, \beta_1, \beta_2)'$; 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 (3), there are many alternative metamodels. For example, a simpler metamodel is

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

where $x$ is the traffic rate—in queueing theory usually denoted by $\rho$:

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

This combination of the two original factors $\lambda$ and $\mu$ into a single factor $\rho$ (inspired by queueing theory) illustrates the use of *transformations*. Another useful transformation may be a logarithmic one: replacing $y$, $\lambda$, and $\mu$ by $\log(y)$, $\log(\lambda)$, and $\log(\mu)$ in (3) makes the first-order polynomial approximate relative changes; i.e., the regression parameters $\beta$ become elasticity coefficients. 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. The univariate regression model may be applied to each individual output of a given simulation model. The matrix notation of the general linear regression model is

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{e} \tag{6}$$

where $\mathbf{y} = (y_1, \ldots, y_n)'$ is the $n$-dimensional vector with the regression predictor (or dependent variable) $y$ with $n$ the number of simulation runs (or observations); $\mathbf{X} = (\mathbf{x}_{ij})$ is the $n \times q$ matrix of explanatory (independent) regression variables with $\mathbf{x}_{ij}$ the value of explanatory variable $j$ in run $i$ $(i = 1, \ldots, n; j = 1, \ldots, q)$; $\beta = (\beta_1, \ldots, \beta_q)'$ is the vector with $q$ regression parameters—including the effect of a possible dummy variable so $\beta_1$ 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) $\widehat{\beta} = (\widehat{\beta}_1, \ldots, \widehat{\beta}_q)'$ for the regression parameters, the *Least Squares* (LS)—also called the Ordinary LS—criterion is often used; i.e., $\widehat{\beta}$ is selected such that it minimizes the *Sum of Squared Residuals*, *SSR*:

$$\min_{\widehat{\beta}} SSR = \sum_{i=1}^{n} (\widehat{e}_i)^2 = \sum_{i=1}^{n} (\widehat{y}_i - w_i)^2 = (\widehat{\mathbf{y}} - \mathbf{w})'(\widehat{\mathbf{y}} - \mathbf{w}) \tag{7}$$

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

$$\widehat{y}_i = \sum_{j=1}^{q} x_{ij}\widehat{\beta}_j = \mathbf{x}_i'\widehat{\beta}, \tag{8}$$

and $w_i$ denotes the simulation output of run $i$ (e.g., the average waiting time of that run; see (2)).

The solution of (7) gives the LS estimate $\widehat{\beta}$ of the regression parameter vector $\beta$ in the regression model (6):

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{w}. \tag{9}$$

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 (6)—and hence in (9)—is discussed in the next sections.

The LS criterion is a mathematical (not a statistical) criterion—also known as the $L_2$ norm. 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:

(i) The simulation output $w$ is indeed *normally* (or Gaussian) distributed if this output is an average; e.g., (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.

(ii) 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.

(iii) '*Identically* distributed' implies a constant variance. In practice, however, the simulation outputs do not have the same variance; i.e., the variances are heterogeneous or heteroscedastic instead of homogeneous. For example, for the M/M/1 the variances increase as the traffic rate increases. This practical problem is further discussed in Kleijnen (2006, 2007).

This tutorial assumes that the simulation outputs $w$ are indeed normally and independently distributed with the same variance (say) $\sigma_w^2$. 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: $\forall i : \widehat{e}_i = 0$. This also deserves some comments:

(i) The metamodel is *biased* if $E(e) \neq 0$.

(ii) A *perfectly* fitting metamodel indicates that $n$ (number of 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 $\mathbf{w}$:

$$\hat{\beta} = \mathbf{L}\mathbf{w} \tag{10}$$

where $\mathbf{L}$ is not random since $\mathbf{L} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ in (9). The linear estimator (10) has the following two properties:

$$E(\hat{\beta}) = \mathbf{L}[E(\mathbf{w})] \tag{11}$$

and

$$\mathbf{cov}(\hat{\beta}) = \mathbf{L}[\mathbf{cov}(\mathbf{w})]\mathbf{L}'. \tag{12}$$

It is easy to prove that (11) implies that the LS estimator $\hat{\beta}$ is unbiased. And the property in (12) implies that—in case of white noise—$\hat{\beta}$ has the following covariance matrix:

$$\mathbf{cov}(\hat{\beta}) = (\mathbf{X}'\mathbf{X})^{-1}\sigma_w^2. \tag{13}$$

Furthermore, it can be proven that among all linear unbiased estimators, $\hat{\beta}$ is *best*; i.e., $\hat{\beta}$ has the minimum variance. Obviously, the variances of the individual regression estimators $\widehat{\beta}_j$are given by the main diagonal elements of (13); their covariances are given by the off-diagonal elements of this (symmetric) matrix.

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

$$\widehat{\beta} \sim N[\beta, (\mathbf{X}'\mathbf{X})^{-1}\sigma_w^2]. \tag{14}$$

Consequently, the individual estimated regression parameters $\widehat{\beta}_j$ may be tested through the $t$ statistic with $n-q$ degrees of freedom:

$$t_{n-q} = \frac{\widehat{\beta}_j - \beta_j}{s(\widehat{\beta}_j)} \text{ with } j = 1, \ldots, q \tag{15}$$

where $s(\widehat{\beta}_j)$ is the square root of the $j^{th}$ element on the main diagonal of (13) where $\sigma_w^2$ is estimated through the *Mean Squared Residuals* (*MSR*):

$$MSR = \frac{SSR}{n-q} = \frac{(\widehat{\mathbf{y}} - \mathbf{w})'(\widehat{\mathbf{y}} - \mathbf{w})}{n-q} \tag{16}$$

where SSR was given in (7). This *MSR* 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 replicates; see (21)). The $t$ statistic in (15) may be used to test whether a specific regression parameter is zero:

$$H_0 : \beta_j = 0. \tag{17}$$

Besides testing a single parameter, the analysts may hypothesize that *several* parameters have specific values; e.g., the effects of both the arrival rate and the service rate are zero: $\beta_1 = 0$ and $\beta_2 = 0$ in (3). More generally,

$$H_0 : \beta_{j'} = \ldots = \beta_q = 0 \tag{18}$$

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:

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

$$F_{q-j'+1;n-q} = \frac{SSR_{reduced} - SSR_{full}}{SSR_{full}}. \tag{19}$$

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 either *passive* observation of a real system or *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 (6) is observed more than once. A *replicate* 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 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. See Law (2007).

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)

$$N = \sum_{i=1}^{n} m_i. \tag{20}$$

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 (22)). 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 (24) and also Kleijnen (1987, p. 195).

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

$$\widehat{var(w_i)} = \widehat{\sigma^2(w_i)} = s_i^2(w) = \frac{\sum_{r=1}^{m_i}(w_{ir} - \overline{w_i})^2}{m_i - 1} (i = 1, \ldots n) \tag{21}$$

with

$$\overline{w_i} = \frac{\sum_{r=1}^{m_i} w_{ir}}{m_i}. \tag{22}$$

Because of the common variance assumption implied by the white noise assumption, the $n$ variance estimators in (21) may be *pooled* using their degrees of freedom as weights:

$$\widehat{var(w)} = \widehat{\sigma_w^2} = s^2(w) = \frac{\sum_{i=1}^{n}(m_i - 1)s_i^2}{\sum_{i=1}^{n}(m_i - 1)}. \tag{23}$$

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

(i) The *MSR* (defined in (16) for non-replicated combinations), which is now defined in (24) 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.

(ii) The *pooled* variance estimator in (23), 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 may be compared through the following so-called *lack-of-fit F-statistic* (see Myers and Montgomery 2000, p. 52):

$$F_{n-q;N-n} = \frac{\sum_{i=1}^{n} m_i(\overline{w_i} - \widehat{y_i})^2/(n-q)}{\sum_{i=1}^{n}\sum_{r=1}^{m_i}(w_{ir} - \overline{w_i})^2/(N-n)}. \tag{24}$$

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, 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 (4). 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 (4). The validity of the fitted polynomial, however, becomes questionable as the experimental area gets bigger. Zeigler et al. (2000) 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 Kleijnen and Sargent 2000 and Law 2007).

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 $\beta_0$):

$$E(y) = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k. \tag{25}$$

So the general linear regression model (6) now has $q$ (number of regression parameters) equal to $k+1$. An example is the first-order polynomial for the two factors $\lambda$ and $\mu$ in (3).

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 Spall 2003). 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 with two levels per factor and to fit a first-order polynomial to the resulting I/O data; see Angün et al. (2002) and Section 5.

It is convenient and traditional in DOE to use *standardized* 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 $i = 1,\ldots n$:

$$x_{ij} = \frac{z_{ij} - \overline{z_j}}{(u_j - l_j)/2} \tag{26}$$

where $\overline{z_j}$ 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 $+1$.

In practice, simulation analysts also consider inputs with *nominal scales with more than two levels*. For example, the sea has three types of bottom (namely clay, sand, or rocks) in a simulation study on the sonar search for mines. The analysts erroneously coded these three surface types as $-1$, 0, and $+1$. The correct coding 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 Kleijnen (2007).

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 estimated effects $\widehat{\beta}_j$ should be sorted.

A factor may be significant when tested through the $t$ statistic defined in (15), but may be unimportant—especially when compared with other factors in the experiment.

A $2^k$ design results in an *orthogonal* matrix of explanatory variables for the first-order polynomial (25); i.e.,

$$\mathbf{X}'\mathbf{X} = n\mathbf{I}. \tag{27}$$

This property follows directly from the way a $2^k$ design is constructed. This property simplifies the LS estimator, because substituting (27) into (9) gives

$$\hat{\beta} \;= (n\mathbf{I})^{-1}\mathbf{X}'\mathbf{w} = \mathbf{X}'\mathbf{w}/n = \left(\frac{\sum_{i=1}^{n} x_{ij} w_i}{n}\right). \tag{28}$$

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 $j$:

$$\widehat{\beta}_j = \frac{\sum_{i=1}^{n} x_{ij} w_i/(n/2)}{2} = \frac{\overline{w_{1j}} - \overline{w_{2j}}}{2} \tag{29}$$

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, (27) simplifies the covariance matrix (13) to

$$\mathbf{cov}(\hat{\beta}) \;= (n\mathbf{I})^{-1}\sigma_w^2 = \mathbf{I}\frac{\sigma_w^2}{n}. \tag{30}$$

So all estimators have the same variance $\sigma_w^2/n$, and they are independent. Box (1952) proves 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 grows exponentially with the number of factors: $n = 2^k$. 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

The term *resolution* describes the degree of confounding (or aliasing) among estimated main effects, two-factor interactions, three-factor interactions, etc. (these effects are further discussed below). The lower the resolution is (denoted by the Roman numerals III, IV, V, etc.), the more aliasing there is.

A design of Resolution-III (R-III) 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 Box and Hunter (1961a). These designs are also known as *Plackett-Burman* designs. These designs have as a subclass *fractional factorial two-level* or $2^{k-p}_{III}$ 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 (e.g., $n = 12$).

Let's consider a simple example with $k = 3$ factors. 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 R-III, it is denoted as a $2^{3-1}_{III}$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', and '$\mathbf{3} = \mathbf{1.2}$' 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).

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

| Combi. | **1** | **2** | $\mathbf{3}=1.2$ | $\mathbf{3}=-1.2$ |
|---|---|---|---|---|
| 1 | − | − | + | − |
| 2 | + | − | − | + |
| 3 | − | + | − | + |
| 4 | + | + | + | − |

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

Another simple 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 $(n=)2^{k-p}>k(=q-1)$. The solution is $k=7$ and $p=4$. This gives the analogue of Table 1, now with the generators $\mathbf{4}=\mathbf{1.2}$, $\mathbf{5}=\mathbf{1.3}$, $\mathbf{6}=\mathbf{2.3}$, and $\mathbf{7}=\mathbf{1.2.3}$. This design belongs to a family formed by substituting a minus sign for the (implicit) plus sign in one or more generators; e.g., substituting $\mathbf{4}=-\mathbf{1.2}$ for $\mathbf{4}=\mathbf{1.2}$ 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 (6). Hence, no degrees of freedom are left in the *MSR* in (16), so the lack-of-fit $F$-test in (24) 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 (e.g., 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.

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; see Kleijnen (2007). Also see Sanchez and Sanchez (2005) for a different procedure (based on Walsh functions).

*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, Plackett and Burman published such designs for $12\leq n\leq 96$; also see Kleijnen (1975, pp. 332-333) and Myers and Montgomery (2002, 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. 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 $E(w|x_j,x_{j'}=c)$ 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 (25) with two-factor (also called two-way or pairwise) interactions yields

$$E(y)=\beta_0+\sum_{j=1}^{k}\beta_j x_j+\sum_{j=1}^{k-1}\sum_{j'=j+1}^{k}\beta_{j;j'}x_j x_{j'}. \qquad (31)$$

The total number of interactions is $k(k-1)/2$, so the total number of parameters is $q=1+k(k+1)/2$. The formulation of $\mathbf{X}$ for the metamodel (31) follows straightforwardly from $\mathbf{D}$ (design matrix):

$$\mathbf{X}=(x_{ij})=(1,d_{i1},\ldots,d_{ik},d_{i1}d_{i2},\ldots,d_{i;k-1}d_{ik})\ (i=1,\ldots,n)\ . \qquad (32)$$

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 Kleijnen and Standridge (1988).

The ANOVA (ANalysis Of VArinance) literature uses higher-order interactions, e.g., three-factor interactions. 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 'lack of fit' and 'validation' in this contribution.

## 7 DESIGNS ALLOWING TWO-FACTOR INTERACTIONS: RESOLUTION-IV

A design of Resolution-IV (R-IV) gives unbiased estimators of the parameters of a first-order polynomial, even if two-factor interactions are non-zero. Box and Wilson (1951) prove the *foldover* theorem, which may be reformulated as follows: If a R-III design (say) $\mathbf{D}_{III}$ is augmented with its 'mirror' design $-\mathbf{D}_{III}$, then the resulting design is a R-IV design. So the price for augmenting a R-III to a R-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 $2^{7-4}_{III}$ design with its mirrored design gives a

design with $n = 16$ combinations, namely, a $2^{7-3}_{IV}$ design. So $\mathbf{X}$ corresponding with the regression model (31) has $n = 16$ rows and $q = 1+7(7+1)/2 = 29$ columns, so this $\mathbf{X}$ is collinear. 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 (36). Obviously, the $2^{7-3}_{IV}$ design remains balanced.

Useful manipulations with the generators (such as $\mathbf{3} = \mathbf{1.2}$ in the $2^{3-1}_{III}$ design of Table 1) are explained in Kleijnen (2007).These manipulations show how estimated effects are *confounded* or *aliased*; e.g., it is easy to prove that the generator $\mathbf{3} = \mathbf{1.2}$ implies $E(\widehat{\beta}_1) = \beta_1 + \beta_{2;3}$, $E(\widehat{\beta}_2) = \beta_2 + \beta_{1;3}$, and (of course) $E(\widehat{\beta}_3) = \beta_3 + \beta_{1;2}$; i.e., only if $\beta_{2;3} = 0$, the estimator $\widehat{\beta}_1$is unbiased, etc. But R-III designs indeed assume that all interactions are zero!

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

The R-IV designs discussed so far imply that the number of combinations increases with jumps of eight, because the underlying R-III designs have a number of combinations that jump with four. Webb (1968) derived R-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. Kleijnen (1975, pp.344–348) gives details.

This section is concluded with a general discussion of *confounding*. Suppose that a valid linear regression metamodel is

$$E(w) = E(y) = \mathbf{X}_1\beta_1 + \mathbf{X}_2\beta_2. \tag{33}$$

An example is an $\mathbf{X}_1$ corresponding with the intercept and the main effects collected in $\beta_1$, and an $\mathbf{X}_2$ corresponding with the two-factor interactions $\beta_2$. Suppose that the analysts use the simple metamodel without these interactions. Then they estimate the first-order polynomial coefficients through

$$\widehat{\beta}_1 = (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'\mathbf{w}. \tag{34}$$

So (34) gives

$$E(\widehat{\beta}_1) = (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'E(\mathbf{w}). \tag{35}$$

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

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

Substitution of (33) into (35) gives

$$E(\widehat{\beta}_1) = \beta_1 + (\mathbf{X}_1'\mathbf{X}_1)^{-1}\mathbf{X}_1'\mathbf{X}_2\beta_2. \tag{36}$$

This gives an unbiased estimator of $\beta_1$ if either $\beta_2 = \mathbf{0}$ or $\mathbf{X}_1'\mathbf{X}_2 = \mathbf{0}$. Indeed, R-III designs assume that $\beta_2 = \mathbf{0}$ where $\beta_2$ consists of the two-factor interactions; R-IV designs ensure that $\mathbf{X}_1'\mathbf{X}_2 = \mathbf{0}$ (interaction columns orthogonal to main effects and intercept columns).

## 8 DESIGNS FOR TWO-FACTOR INTERACTIONS: RESOLUTION-V

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

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

Sanchez and Sanchez (2005) give a computer procedure for constructing R-V designs for $k \leq 120$; e.g., a $2^{120-105}_{V}$ design. Unfortunately, such $2^{k-p}$ designs—except for the $2^{5-1}_{V}$ design in Table 2—require 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

Table 3: Generators for Rechtschaffner's R-V designs

| 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 4: Rechtschaffner's design for four factors

| 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 | | $-1$ | $-1$ | $+1$ | $+1$ |

requires $n=32,768$ whereas $q=7,261$ so its efficiency is only 0.22. There are R-V designs that require fewer runs; see Sanchez and Sanchez (2005, pp. 372-373).

Actually, if a simulation run takes much computer time, then *saturated* designs are attractive. Rechtschaffner (1967) gives simple saturated non-orthogonal fractions of two-level (and three-level) designs; see Table 3 (and also Kleijnen 1975, 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 Kleijnen (1975, p. 352). An application of these designs is presented by Kleijnen and Pala (1999), involving $k$ = 6 factors and Rechtschaffner's design with only $n$ = $q$ = 22 combinations.

## 9 REGRESSION ANALYSIS: SECOND-ORDER POLYNOMIALS

The classic 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

$$E(y)=\beta_0+\beta_1 x+\beta_2 x^2. \tag{37}$$

Obviously, estimation of the three parameters in (37) 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. DOE also provides designs with three values per factor; e.g., $3^k$ designs. However, more popular in simulation are Central Composite Designs (CCDs), which have five values per factor (see Section 10 below).

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

$$E(y)=\beta_0+\sum_{j=1}^{k}\beta_j x_j+\sum_{j=1}^{k}\sum_{j'\geq j}^{k}\beta_{j;j'}x_j x_{j'}. \tag{38}$$

So this metamodel adds $k$ *purely quadratic* effects $\beta_{j;j}$ to (31). In practice, second-order polynomials are applied either locally or globally. *Local* fitting may be used when searching for the optimum input combination; see Angün et al. (2002). *Global* fitting (for $0<x<1$ in the queueing simulation) using second-order polynomials has been applied, but Kriging provides better metamodels; see Van Beers and Kleijnen (2003).

## 10 CCD FOR SECOND-DEGREE POLYNOMIALS

A *CCD* augments a R-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 standardized factors—the central point is $(0,\ldots 0)'$, and the 'positive' axial point for factor $j$ is the point with $x_j=+c$ and all other $k-1$ factors fixed at the center 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; i.e., 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; e.g., $k=2$ implies $n_{CCD}$ = $2^2+1+2\times 2$ = 9. For $k=120$, Sanchez and Sanchez (2005) give $n_{CCD}$ = 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 (24). CCDs are further discussed in Myers and Montgomery (2002) and NIST (2006).

Obviously, CCDs are rather inefficient. Therefore, Kleijnen and Pala (1999) simulate only half of the star design. Classic R-V designs are very inefficient, whereas Rechtschaffner's designs are saturated. Finally, Kleijnen (1987, 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; e.g., Kriging and neural networks.

### 11.1 $R^2$ and $\rho$

$R^2$ may be defined as follows:

$$R^2 = \frac{\sum_{i=1}^{n}(\widehat{y}_i - \overline{\overline{w}})^2}{\sum_{i=1}^{n}(\overline{w_i} - \overline{\overline{w}})^2} = 1 - \frac{\sum_{i=1}^{n}(\widehat{y}_i - \overline{w_i})^2}{\sum_{i=1}^{n}(\overline{w_i} - \overline{\overline{w}})^2} \qquad (39)$$

where $\widehat{y}_i$ denotes the metamodel predictor defined in (8), $\overline{w_i}$denotes the simulation response of combination $i$ averaged over its $m_i \geq 1$ replicates defined in (22), and $\overline{\overline{w}} = \sum_{i=1}^{n}\overline{w_i}/n$ denotes the overall average simulation response. The rightmost equality implies 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, which is the numerator of the classic variance estimator computed over the $n$ combinations—analogous to (21).

In (39), $R^2$ 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.

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

$$R^2_{adjusted} = 1 - \frac{n-1}{n-q}(1 - R^2). \qquad (40)$$

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 known distributions. Analysts therefore use subjective lower thresholds. Kleijnen and Deflandre 2006) demonstrate how the distributions of these two statistics can be obtained through *bootstrapping*; the classic textbook on bootstrapping is Efron and Tibshirani (1993).

$R^2$ is also called the *multiple correlation coefficient.* However, $R^2$ should be distinguished from *Pearson's correlation coefficient*—usually denoted by $\rho$. This $\rho$ measures the strength of the *linear* relationship between two *random* variables (say) $x$ and $w$. Like $R$, this $\rho$ ranges between $-1$ and $+1$. A value of $+1$ implies that the two variables are related perfectly by an increasing linear relationship (with positive slope). This $\rho$ is estimated through

$$\widehat{\rho(x,w)} = \widehat{\rho} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(w_i - \overline{w})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(w_i - \overline{w})^2}}. \qquad (41)$$

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

$$t_{n-2} = \frac{\widehat{\rho}}{\sqrt{1 - \widehat{\rho}^2}}\sqrt{n-2}. \qquad (42)$$

It may happen that the two variables $x$ and $w$ are related, but not through the linear relationship $E(w|x) = \beta_0 + \beta_1 x$. An alternative relationship may be (say) $E(w|x) = \beta_0 x^{\beta_1}$. Such an increasing monotonic relationship may be quantified through *Spearman's rank correlation coefficient* (say) $\eta$. This coefficient is Pearson's coefficient computed—not from the original pairs $(x_i, w_i)$—but from the ranked pairs $(r(x_i), r(w_i))$; see Conover (1999).

More details on the use of $\rho$ and $\eta$ for identifying important factors in simulation (not for quantifying the adequacy of a metamodel) are given by Kleijnen and Helton (1999).

### 11.2 Cross-Validation

*Cross-validation is* applied not only in linear regression analysis, but also in nonlinear regression, Kriging, neural networks, etc. 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: $m_i = m \geq 1$. The LS estimate may then replace $w_{ir}$ (individual simulation output) by $\overline{w_i}$(average simulation output). The procedure runs as follows.

(i) 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 noncollinear matrix $\mathbf{X}_{-i}$. 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, e.g., the center point if the original design is not a CCD.

(ii) Recompute the LS estimator from the remaining I/O data: $\widehat{\beta_{-i}} = (\mathbf{X}'_{-i}\mathbf{X}_{-i})^{-1}\mathbf{X}'_{-i}\mathbf{w}_{-i}$.

(iii) Use this recomputed estimator to compute the regression prediction for the combination deleted in

step (i):

$$\widehat{y_{-i}} = \mathbf{x}_i' \widehat{\beta_{-i}}. \tag{43}$$

(iv) Repeat the preceding three steps, until all $n$ combinations have been processed. This results in $n$ predictions $\widehat{y_{-i}}$ $(i = 1, \ldots, n)$.

(v) 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 Van Groenendaal (1998) and Vonk Noordegraaf (2002).

The following alternative for the subjective judgment in step 5 is proposed in Kleijnen (1983): Compute

$$t_{m-1}^{(i)} = \frac{\overline{w_i} - \widehat{y_{-i}}}{\sqrt{\widehat{var}(\overline{w_i}) + \widehat{var}(\widehat{y_{-i}})}} \quad (i = 1, \ldots, n) \tag{44}$$

where $\widehat{var}(\overline{w_i}) = \widehat{var}(w_i)/m$ (and $\widehat{var}(w_i)$ was given in (21)) and $\widehat{var}(\widehat{y_i})$ follows from (43) and the analogue of (12):

$$\widehat{var}(\widehat{y_{-i}}) = \mathbf{x}_i' \widehat{cov}(\widehat{\beta_{-i}}) \mathbf{x}_i \tag{45}$$

where

$$\widehat{cov}(\widehat{\beta_{-i}}) = \widehat{var}(\overline{w_i}) (\mathbf{X}_{-i}' \mathbf{X}_{-i})^{-1}. \tag{46}$$

Note that $\overline{w_i}$ and $\widehat{y_{-i}}$ are independent because the latter does not use the former.

Since (44) gives $n$ values, the regression metamodel is rejected if

$$\max_i t_{m-1}^{(i)} > t_{m-1;1-[\alpha/(2n)]} \tag{47}$$

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

There is a *shortcut* for the $n$ computations above. Modern software uses the so-called *hat matrix*

$$\mathbf{H} = (\mathbf{h}_{ii'}) = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \text{ with } i, i' = 1, \ldots, n. \tag{48}$$

It can be proven that the numerator of (44) can be written as

$$\overline{w_i} - \widehat{y_{-i}} = \frac{\overline{w_i} - \widehat{y_i}}{1 - h_{ii}}$$

and (44) itself can be written as

$$t_{m_i-1} = \frac{\overline{w_i} - \widehat{y_i}}{\sqrt{\widehat{var}(\overline{w_i})}\sqrt{1 - h_{ii}}} \quad (i = 1, \ldots, n) \tag{49}$$

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 $h_{ii}$.

Cross-validation not only affects the regression predictions $(\widehat{y_{-i}})$, but also the estimated regression parameters $\widehat{\beta_{-i}}$. So the analysts may be interested not only in the predictive performance of the metamodel, but also in its *explanatory* performance; an example is given in the FMS case study mentioned above.

The regression literature proposes several so-called *diagnostic* statistics that are related to (49); e.g., PRESS, DEFITS, DFBETAS, and Cook's $D$; see Kleijnen and Van Groenendaal (1992, 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, and V, and CCDs. The tutorial assumed white noise, meaning that the residuals of the fitted linear regression models are NIID with zero mean. The white noise assumption is dropped in Kleijnen (2006, 2007), explaining the consequences.

Note that the Internet gives DOE software; see, e.g., http://www.scientific-computing.com.

## REFERENCES


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

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

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

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

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

Conover, W.J. (1999), *Practical non-parametric statistics: third edition.* Wiley, New York

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

Del Castillo, E. (2007), *Process optimization: a statistical approach*. Springer, New York

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

Kleijnen, J.P.C. (1975), *Statistical techniques in simulation; part II*. Marcel Dekker, New York (Russian translation, Publishing House 'Statistics', Moscow, 1978)

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

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

Kleijnen, J.P.C. (2006), White noise assumptions revisited: Regression metamodels and experimental designs for simulation practice. *Proceedings of the 2006 Winter Simulation Conference*, edited by L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, pp. 107 –117

Kleijnen, J.P.C. (2007), *DASE: Design and analysis of simulation experiments.* Springer, New York

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

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

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

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

Kleijnen, J.P.C. and R.G. Sargent (2000). A methodology for the fitting and validation of metamodels in simulation. *European Journal of Operational Research*, 120, no. 1, pp. 14 –29

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## AUTHOR BIOGRAPHY

**JACK P.C. KLEIJNEN** is professor of 'simulation and information systems' at Tilburg University. He is a member of the Department of Information Systems and Management and the Operations Research Group of the Center for Economic Research (CentER). He also teaches at the Eindhoven University of Technology, in the Postgraduate International Program in Logistics Management Systems. He is an 'external fellow' of the Mansholt Graduate School of Social Sciences of Wageningen University. His research concerns the statistical design and analysis of simulation experiments, information systems, and supply chains. He has been a consultant for several organizations in the USA and Europe, and serves on many international editorial boards and scientific committees. He spent several years in the USA, at universities and private companies. He received a number of international awards, including the INFORMS Simulation Society's 'Lifetime Professional Achievement Award (LPAA)' of 2005. More information can be found on <http://center.uvt.nl/staff/kleijnen/>.