A:\TutorEU4.wpd
Version 4
Written: June 27, 2001
Printed: June 27, 2001 (11:46AM)

EXPERIMENTAL DESIGNS FOR SENSITIVITY ANALYSIS OF SIMULATION MODELS

Jack P.C. Kleijnen

Department of Information Systems (BIK)/Center for Economic Research (CentER),
Tilburg University (KUB), Postbox 90153, 5000 LE Tilburg, Netherlands
Phone: +3113-4662029; Fax: +3113-4663377; E-mail: kleijnen@kub.nl
Web: center.kub.nl/staff/kleijnen

Eurosim 2001 Conference, Delft, 26-29 June 2001

Note:
This version corrects errors in the version published in the
Proceedings of EUROSIM 2001
(edited by A.W. Heemink et al., ISBN: 90-806441-1-0)

EXPERIMENTAL DESIGNS FOR SENSITIVITY ANALYSIS OF SIMULATION MODELS

Jack P.C. Kleijnen

Department of Information Systems (BIK)/Center for Economic Research (CentER),
Tilburg University (KUB), Postbox 90153, 5000 LE Tilburg, Netherlands
Phone: +3113-4662029; Fax: +3113-4663377; E-mail: kleijnen@kub.nl
Web: center.kub.nl/staff/kleijnen

ABSTRACT

This introductory tutorial gives a survey on the use of statistical designs for what-if or sensitivity analysis in simulation. This analysis uses regression analysis to approximate the input/output transformation that is implied by the simulation model; the resulting regression model is also known as metamodel, response surface, compact model, emulator, etc. This regression analysis gives better results when the simulation experiment is well designed, using classical statistical designs (such as fractional factorials, including 2^{k-p} designs). These statistical techniques reduce the ad hoc character of simulation; that is, these techniques can make simulation studies give more general results, in less time.

KEYWORDS

Design of experiments; regression; simulation; metamodel; interpolation

1 INTRODUCTION

A primary goal of simulation is *what if* analysis: what happens if inputs of the simulation model change? Therefore we run a given simulation program for (say) n different combinations of the k simulation inputs. These combinations are often called *scenarios*. The simulation inputs are called *factors* in design of experiments (DOE); factors may be a parameter, an input variable, or a module of the simulation model (or computer simulation program or code). By definition, a factor takes at least two levels or 'values' during the experiment. The factor may be qualitative, for example, a factor may represent different priority rules corresponding with different computer modules. A detailed discussion of qualitative factors and various measurement scales is given in Kleijnen (1987, pp. 138-142). Note that the terminology varies so much because simulation is applied in so many disciplines.

Given the set of n scenarios, we run the simulation model and observe the outputs. Most simulation models have *multiple outputs*. In practice, however, these outputs are handled through the application of the techniques surveyed in this tutorial, *per* output type. Also see Khuri (1996) and Kleijnen (1987).

Why is DOE needed? To answer this question, let us consider the following case study. A decision support system (DSS) was developed for production planning in a specific Dutch steel tube factory. The DSS and the factory were modeled through a stochastic, discrete-event simulation. The DSS was to be optimized. This DSS, however, had fourteen input or decision variables; there were two output variables (namely, productive hours and lead time). The simulation of one combination of these fourteen inputs took six hours of computer time, so searching for the optimal combination had to be

performed with care. Details are given in Kleijnen (1993).

So, the central problem in DOE is the *astronomically great number of possible factor combinations*. In the case study, at least 2^{14} combinations may be distinguished. In general, DOE can be defined as selecting the combinations of factor levels that will be actually simulated in an experiment with the simulation model. In the DSS example, a few sequences - each consisting of 16 combinations - were actually run.

The other side of the coin is: How to *analyze* the simulation's input/output (I/O) data? Classic analysis uses regression modeling, also known in DOE as Analysis of Variance (ANOVA). Actually, design and analysis is a well-known chicken-and-egg problem: Before we decide on the design, we specify (explicitly or implicitly) a regression metamodel. The simpler this model is, the fewer scenarios we need to simulate. For example, suppose we assume an additive model (ANOVA with main effects only, no interactions at all). Then the number of scenarios may equal the number of factors, plus one (that one is needed because of the overall effect or grand mean): $n = k + 1$. In the DSS example with $k = 14$ we had $n = 16 \approx 14 + 1$.

Note that a *metamodel* is an approximation of the I/O transformation implied by the underlying simulation program. Many other terms are popular in certain (sub)disciplines: response surface, compact model, emulator, etc. Also see the monograph by Friedman (1996).

A metamodel treats the simulation model as a *black box*; that is, the simulation model's I/O is observed, and the factor effects in the metamodel are estimated. This black-box approach has the following advantages and disadvantages.

An *advantage* is that DOE can be applied to all simulation models, either deterministic or stochastic, either in steady-state or in transient state. Further, DOE gives better estimates of the factor effects than does the intuitive approach often followed in practice, namely change one factor at a time (see section 3). A *disadvantage* is that DOE cannot take advantage of the specific structure of a given simulation model (whereas other techniques such as perturbation analysis and score function can).

What do we precisely mean by *what-if analysis* in a simulation context? Unfortunately, the vast literature on simulation does not provide a standard definition. In this paper we interpret sensitivity analysis as the systematic investigation of the reaction of the simulation outputs to *extreme* values of the model's input or to *drastic* changes in the model's structure. For example, what happens when a parameter doubles; what happens if a module changes? (So we do not discuss marginal changes or perturbations in the input values.)

Sensitivity analysis can also help in optimization and validation of the simulation model. In this tutorial, however, we do not explicitly discuss the latter two topics, but refer to Kleijnen (1998, 2000). Further, if the simulation model has hundreds of factors, then the designs discussed in this tutorial require too much computer time, and special 'screening' designs are needed, discussed in Campolongo, Kleijnen, and Andres (2000).

To summarize this section, we claim that DOE is an important practical method for answering what-if questions in simulation. This claim is not surprising: By definition, simulation means that a model is used, not for mathematical analysis or numerical methods, but for experimentation. But, *experimentation requires a good design and a good analysis*.

The remainder of this paper is organized as follows. In section 2 we discuss regression metamodels. In section 3 we present designs to estimate these models. In section 4 we give conclusions. A list of references enables the reader to study the topic in depth.

2 REGRESSION MODELS

In DOE applied to simulation, the regression metamodels typically fall into one of the following three classes: (i) a first-order polynomial, which consists of main effects only, besides an overall mean; (ii) a first-order polynomial augmented with interactions between pairs of factors (two-factor interactions); and (iii) a second-order polynomial, which also includes purely quadratic effects. (In equation 1 we shall give an exact formulation.) Whatever metamodel we choose initially, this metamodel should be validated.

In *stochastic* simulation (deterministic simulation will be discussed below equation 1), scenarios are often run for different, non-overlapping pseudo-random number sequences, which gives identically and independently distributed (IID) *replicated* outputs (say) $W_{i;\,r}$ where $i = 1, ..., n$ refers to the scenario (combination of k factors) and $r = 1, ..., m_i$ to the replicate (obviously, the integers must satisfy $n > k$ and $m_i \geq 1$). In other words, let the simulation model have the single output (say) W and the k inputs z_h ($h = 1, ..., k$). Then the *linear regression (meta)model* is

$$Y = \sum_{j=1}^{q} \beta_j x_j + E \text{ or } Y = \boldsymbol{\beta}\, \boldsymbol{x} + E \tag{1}$$

with the following symbols:

Y: the regression estimator of a function of the simulation output W (for example, $\log(W)$),
x_j: a function of the simulation input (for example, $z_h z_h$ with $h = 1, ..., k$),
β_j: the j^{th} regression parameter, and
E: the noise (discussed next).

In *deterministic* simulation m_i equals one and E reduces to pure fitting error, whereas in stochastic simulation E is the fitting error plus the so-called intrinsic simulation noise (generated through pseudo-random numbers; see equation 3 later).

The regression parameters in (1) can be estimated through *least squares* (LS), which uses a mathematical criterion. Statistics become relevant if we introduce assumptions regarding the noise E in (1). Indeed, *ordinary* LS (OLS) gives the *best linear unbiased estimator* (BLUE) - with 'best' meaning minimum variance - if this E represents *white noise*: E is IID with zero mean. Moreover, OLS gives the *maximum likelihood* (ML) estimator if this E has a Gaussian (normal) distribution. So OLS is optimal under many criteria if

$$\boldsymbol{E} \sim N_p(\boldsymbol{0}, \; {}^{2}_{E}\boldsymbol{I}) \tag{2}$$

where $\boldsymbol{E}$ denotes the vector of E's, $N_p(\boldsymbol{a}, \boldsymbol{b})$ denotes a p-variate normal distribution with mean $\boldsymbol{a}$ and covariance matrix $\boldsymbol{b}$, $\boldsymbol{0}$ denotes a vector of zeroes, and ${}^{2}_{E}$ denotes the common (constant) variance of E; the number of components of all these vectors is $p = \sum_1^n m_i$.

In practice, however, not only the mean but also the variance of W is not constant; for example, in many simulated queueing systems W has a much larger variance as the traffic rate (say, x_2) increases. Then, the covariance matrix of $\boldsymbol{E}$ does not involve the identity matrix $\boldsymbol{I}$, but some diagonal matrix. Then BLUE results if *weighted* LS (WLS) is applied. It is convenient to present WLS as a special case of Generalized LS (GLS), discussed next.

In practice, simulationists often use *common random numbers* (CRN) to simulate all the n scenarios. So, replicate #1 uses a set of seeds (say) $\boldsymbol{s}_1$ (seed $\boldsymbol{s}_{1;\,1}$ is used for process 1 in replicate 1, seed $\boldsymbol{s}_{1;\,2}$ for process 2 in replicate 1, etc.), replicate #2 uses the set $\boldsymbol{s}_2$, etc.; no sequence of pseudo-random numbers should overlap; see Law and Kelton (2000).

For CRN, (2) is replaced by

$$\boldsymbol{E} \sim N_p(\boldsymbol{0}, \; {}_{p\times p}(\boldsymbol{E})) \tag{3}$$

where we sometimes abbreviate ${}_{p\times p}(\boldsymbol{E})$ to ${}_{\boldsymbol{E}}$. The goal of CRN is to create positive non-diagonal elements in this ${}_{\boldsymbol{E}}$. Given (3), we get the BLUE if we use the GLS estimator

$$\tilde{\boldsymbol{B}} = [\boldsymbol{x} \; {}_{E}^{\;1} \boldsymbol{x}]^{\;1} \boldsymbol{x} \; {}_{E}^{\;1} \boldsymbol{W} \tag{4}$$

where $\boldsymbol{W}$ denotes the p-dimensional vector with simulation outputs.

Note that if we insert a diagonal matrix for $_{\boldsymbol{E}}$ into the GLS formula (4), then we get the WLS formula. For $_{\boldsymbol{E}}$ shown in (2), we get the well-known OLS formula, namely (say) $\hat{\boldsymbol{B}} = (\boldsymbol{x}\ \boldsymbol{x})^{\ 1}\boldsymbol{x}\ \boldsymbol{W}$.

Actually, the covariance matrix $_{\boldsymbol{E}}$ in (4) is unknown. The replications, however, enable us to compute *estimated covariances* (including variances: $i = i$ in the next equation):

$$\hat{C}_{i;\ i} = \sum_{r\ =\ 1}^{m} \frac{(W_{i;\ r} \quad \overline{W}_i)(W_{i\ ;\ r} \quad \overline{W}_i)}{(m \quad 1)} \text{ with } i;\ i = 1,\ ...,\ n \tag{5}$$

where for simplicity of presentation we assume equal number of replicates ($m_i = m$); further, $\overline{W}_i = \sum_{r\ =\ 1}^{m} W_{i;\ r}/m$ denotes the overall average of combination i. To form $\hat{}_{\boldsymbol{E}}$ - the matrix of estimated error covariances - we use the fact that each of the first m elements of $\boldsymbol{E}$ has the same distribution, namely $N(0,\ ^2_1)$, ..., and each of the last m elements has the distribution $N(0,\ ^2_n)$. Dykstra(1970) proved that the resulting matrix is not singular if $m \geq n$. If this condition holds, then substituting the covariance estimates defined in (5) into the GLS formula (4) gives *estimated GLS* (EGLS), analyzed extensively by Kleijnen (1992). If this condition is not satisfied (so $m < n$), then the mean squared residuals (MSR) can be used; see Kleijnen (1987).

To validate the estimated metamodel, we may use *cross-validation*, as follows. We eliminate one combination (say combination i) and re-estimate the regression model from the remaining n - 1 combinations, giving (say) $\hat{\boldsymbol{\beta}}_{\ i}$ (we focus on OLS; EGLS is similar). This yields the regression predictor

$$\hat{y}_i = \hat{\boldsymbol{\beta}}_{\ i}\ \boldsymbol{x}_i\ (i = 1,\ ....,\ n). \tag{6}$$

We repeat this for all values of i (with i = 1, ..., n). We make a scatter plot of the simulation outputs W_i versus the predicted values computed through (6). We may quantify the metamodel's validity through the (Pearson) correlation coefficient. Details are discussed in Kleijnen and Van Groenendaal (1992) and Van Groenendaal and Kleijnen (1998).

If the metamodel is rejected, we may try other transformations of the simulation I/O, or reduce the experimental area; see Kleijnen and Sargent (2000).

If the metamodel fits, then we may check whether the *signs* of the estimated effects agree with prior, qualitative expert knowledge.

If we ignore interactions and quadratic effects, then the *relative importance* of a factor is obtained by sorting the absolute values of the first-order (or main) effects, provided the factors are *standardized*, as follows.

Suppose that in the simulation experiment the original (non-standardized) factor z_h ranges between a lowest value l_h and an upper value u_h; that is, the simulation model is not valid outside that range or in practice that factor can range over that domain only (for example, the number of servers can vary only between one and five). The variation (or spread) of that factor is measured by $a_h = (u_h - l_h)/2$; its location (or mean) by $b_h = (u_h + l_h)/2$. Then the following standardization is appropriate:

$$\begin{aligned} x_{ih} &= (z_{ih} \quad b_h)/a_h \\ (i = 1,\ ...,\ n)\ &(h = 1,\ ...,\ k)\ . \end{aligned} \tag{7}$$

Software for OLS is abundant; for EGLS it is more limited. However, for EGLS we can use OLS software, after we have transformed the I/O data linearly; see Kleijnen and Van Groenendaal (1992, pp. 140, 157).

3 SOME CLASSICAL DESIGNS

To get unique, unbiased estimators of the q regression effects β_j in (1), it is necessary to simulate at least $n \geq q$ factor combinations. *Which n* combinations to simulate, can be determined such that the accuracy of the estimated factor effects is maximized (variance minimized). This is the goal of the statistical theory on DOE. Unfortunately, classic DOE assumes white noise for E (so OLS gives BLUE). If this assumption does not hold, then the resulting estimators are still unbiased, but we do not know whether they have minimum variance. Details, including *simulation applications* are presented in Kleijnen (1998).

3.1 Main Effects Only

Consider a first-order polynomial in the k simulation inputs z_h, which is a metamodel with only k main effects, plus the intercept (overall mean). In practice, analysts usually first simulate the 'base' situation, and next they change *one factor at a time*; so, all together they simulate $1 + k$ runs. However, DOE derives *orthogonal* designs, that is, designs that satisfy

$$\boldsymbol{x}\,\boldsymbol{x} = n\boldsymbol{I} \tag{8}$$

where $\boldsymbol{x} = (x_{i;j})$ denotes the matrix with $i = 1,\ldots, n$; $j = 1,\ldots, k+1$; $n > k$; $x_{i;1} = 1$ is the dummy factor corresponding with the intercept; and $\boldsymbol{I}$ is the identity matrix with appropriate dimensions (namely $n \times n$). It can be proven that orthogonal designs give estimators of β_j that are unbiased and have smaller variances than the estimators resulting from designs that change one factor a a time.

Orthogonal designs are tabulated in many publications. The analysts may also learn how to construct those designs; see Kleijnen (1987).

A well-known class of orthogonal designs are 2^{k-p} *fractional* factorials. An example is a simulation experiment with $k = 7$ factors in only $n = 2^{7-4} = 8$ factor combinations; see Table 1 where **4** = **1.2** means that we multiply the elements of columns 1 and 2 pairwise, etc. Note that the column for the dummy factor is not displayed, since it is known to be a column of one's, whatever design we select; so **1** refers to the first simulation input z_1. This table can be easily checked for typos: each column has four elements -1; of course it is easy to check that all columns are indeed orthogonal.

Table 1: A 2^{7-4} Design

1	**2**	**3**	**4 = 1.2**	**5 = 1.3**	**6 = 2.3**	**7 = 1.2.3**
-1	-1	-1	1	1	1	-1
1	-1	-1	-1	-1	1	1
-1	1	-1	-1	1	-1	1
1	1	-1	1	-1	-1	-1
-1	-1	1	1	-1	-1	1
1	-1	1	-1	1	-1	-1
-1	1	1	-1	-1	1	-1
1	1	1	1	1	1	1

In practice, however, it is unknown whether only main effects are important. Therefore orthogonal designs with $n \approx k + 1$ should be used only in optimization through RSM; see Kleijnen (1993). Moreover

these designs are useful as building blocks if interactions are accounted for, as we shall see next.

3.2 Main Effects Unbiased by Interactions

It seems prudent to assume that interactions between pairs of factors may be important. Then the k main effects can still be estimated without bias caused by these interactions. However, the number of simulated factor combinations must be *doubled*; for example, $k = 7$ requires $n = 2 \times 8 = 16$: double the number of rows in Table 1 by adding the negative values of the original rows; for example row 9 becomes the negative of row 1 or (1, 1, 1, -1, -1, -1, 1). These designs may also give an indication of the importance of interactions; see Kleijnen (1987).

3.3 Individual Interactions

Suppose the analysts also wish to estimate the individual two-factor interactions between z_h and $z_{h'}$ ($h < h' = 2, ..., k$). There are $k(k - 1)/2$ such interactions. Then many more simulation runs are necessary. An example is $k = 7$, which means 21 interactions besides 7 main effects and one grand mean. However, the standard 2^{k-p} design requires $n = 2^{7-1} = 64$ combinations; see Kleijnen (1987) for details. Therefore only small values for k are studied in practice.

Of course, if k is really small (say, $k = 3$), then all 2^k combinations are simulated, so all interactions (not only two-factor interactions) can be estimated. In practice, these *full factorial* designs are sometimes used indeed. High-order interactions, however, are hard to interpret.

3.4 Quadratic Effects: Curvature

If the k quadratic effects of z_h^2 are to be estimated, then at least k extra runs are needed; moreover, each factor must be simulated for more than two values.

Popular in statistics and in simulation are *central composite designs*. They have five values per factor, and require relatively many runs ($n >> q$). For example, if there are $k = 2$ factors, then $q = 6$ effects are to be estimated but $n = 9$ factor combinations are actually simulated.

4 CONCLUSIONS

In the Introduction (§1) the *question* was raised: What happens if we change parameters, input variables, or modules of a given simulation model?

We can use *regression analysis* to generalize the results of the simulation experiment, since regression metamodels characterize the input/output behavior of the underlying simulation model.

We can apply *design of experiments (DOE)* to get good estimators of the main effects, interactions, and quadratic effects that occur in the regression metamodel. These designs require relatively few simulation runs, and are much better than one-at-a-time designs.

These statistical techniques have already been *applied* many times in practical simulation studies, in many domains. Hopefully, this tutorial will stimulate even more analysts to apply these techniques. The goal of these techniques is to make simulation studies give more general results, in less time.

In the mean time the *research* on statistical techniques adapted to simulation, continues in both Europe and the USA.

REFERENCES

Campolongo, F. J.P.C. Kleijnen, and T. Andres (2000), Screening methods. Chapter 4 in: *Sensitivity analysis*, edited by A. Saltelli, K. Chan, and E.M. Scott, Wiley, Chichester (England), pp. 65-89

Dykstra, R.L. (1970), Establishing the positive definiteness of the sample covariance matrix. *The Annals of Mathematical Statistics*, 41, no. 6, pp. 2153-2154.

Friedman, L.W. (1996), The simulation metamodel. Kluwer, Dordrecht, Netherlands.

Khuri, A.I. (1996), Analysis of multiresponse experiments: a review. *Statistical design and analysis of industrial experiments*, edited by S. Ghosh, Marcel Dekker, New York, pp. 231-246.

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

Kleijnen, J.P.C. (1992) Regression metamodels for simulation with common random numbers: comparison of validation tests and confidence intervals. *Management Science*, 38, no. 8, pp. 1164-1185.

Kleijnen, J.P.C. (1993) Simulation and optimization in production planning: a case study, *Decision Support Systems* 9: 269-280.

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

Kleijnen, J.P.C. (2000), Strategic directions in verification, validation, and accreditation research: a personal view. *Proceedings of the 2000 Winter Simulation Conference* (edited by J.A. Joines, R.R. Barton, K Kang, and P.A. Fishwick), pp. 909-916

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, January 2000, pp. 14-29.

Kleijnen J.P.C. and W. Van Groenendaal (1992) *Simulation: a statistical perspective*. Chichester (U.K.): Wiley.

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

Van Groenendaal W. and J.P.C. Kleijnen (1998) Identifying important factors in deterministic investment problems using design of experiments. (Together with *Proceedings of the 1998 Winter Simulation Conference* (edited by D.J. Medeiros, E.F. Watson, J.S. Carson, M.S. Manivannan), pp. 713-718.