

A User's Guide to the Brave New World of Designing Simulation Experiments

Jack P.C. Kleijnen • Susan M. Sanchez • Thomas W. Lucas • Thomas M. Cioppa

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

*Operations Research Department and the Graduate School of Business and Public Policy
Naval Postgraduate School, Monterey, CA 93943-5219 USA*

*Operations Research Department and the Graduate School of Business and Public Policy
Naval Postgraduate School, Monterey, CA 93943-5219 USA*

*U.S. Army Training and Doctrine Command Analysis Center, Naval Postgraduate School
PO Box 8692, Monterey, CA 93943-0692 USA*

kleijnen@uvt.nl • ssanchez@nps.navy.mil • twlucas@nps.navy.mil • tom.cioppa@trac.nps.navy.mil

Many simulation practitioners can get more from their analyses by using the statistical theory on design of experiments (DOE) developed specifically for exploring computer models. In this paper, we discuss a toolkit of designs for simulationists with limited DOE expertise who want to select a design and an appropriate analysis for their computational experiments. Furthermore, we provide a research agenda listing problems in the design of simulation experiments—as opposed to real world experiments—that require more investigation. We consider three types of practical problems: (1) developing a basic understanding of a particular simulation model or system; (2) finding robust decisions or policies; and (3) comparing the merits of various decisions or policies. Our discussion emphasizes aspects that are typical for simulation, such as sequential data collection. Because the same problem type may be addressed through different design types, we discuss quality attributes of designs. Furthermore, the selection of the design type depends on the metamodel (response surface) that the analysts tentatively assume; for example, more complicated metamodels require more simulation runs. For the validation of the metamodel estimated from a specific design, we present several procedures.

(Metamodels, Latin Hypercube Sampling, Factorial Designs, Sequential Bifurcation, Robust Design)

1. Introduction

Design of experiments (DOE) has a rich history, with a raft of both theoretical developments and practical applications in many fields. Success stories abound in agriculture, clinical trials, industrial product design, and many other areas. Yet, despite the impact DOE has had on other fields and the wealth of experimental designs that appear in the literature, we feel DOE is not used as widely or effectively in the practice of simulation as it should be. We suggest several possible explanations for this phenomenon.

One reason that DOE does not appear to be part of the standard ‘best practices’ is that many simulation analysts have not been convinced of the benefits of DOE. Instead of using even a simple experimental design, many analysts end up making runs to measure performance for only a single system specification, or they choose to vary a handful of the many potential factors one-at-a-time. Their efforts are focused on building, rather than analyzing, the simulation model. DOE benefits can be cast in terms of achieving gains (e.g., improving average performance by using DOE instead of a trial-and-error approach to finding a good solution) or avoiding losses (e.g., obtaining an ‘optimal’ result with respect to one specific environmental setting may lead to disastrous results when implemented). However, many simulation practitioners seem unaware of the additional insights that can be gleaned by effective use of designs.

A second possible reason is that papers on DOE research are often found in specialty journals, making it difficult for simulation analysts to find out about the variety of methods available. Many papers make modifications that improve efficiency or guard against specific kinds of bias, whereas the bigger picture—namely, the setting for which this class of designs is most appropriate—may not be clear to an audience more familiar with simulation modeling issues than with statistical DOE.

The primary reason, however, is that most designs originally developed for real-world experimentation have been subsequently adapted for use in simulation studies, rather than developed specifically for simulation settings. Classic DOE textbooks (e.g., Box, Hunter, and Hunter 1978, Box and Draper 1987, Montgomery 1991, or Myers and Montgomery 1995) do not focus on the needs of simulation analysts, but instead on the practical constraints and implementation issues when conducting real-world experiments. Comprehensive simulation textbooks (Law and Kelton 2001, Banks et al. 2000) do cover a broad range of topics. Though they provide detailed lists of references, they demonstrate DOE by using it

on a few simple test problems. These problems do not stretch the reader's mental framework as to the depth and breadth of insights that might be achieved via other designs. So, studying classic DOE or general simulation textbooks familiarizes analysts with only a small subset of potential designs and applications; hence analysts are likely to force their problems to fit to a particular design instead of identifying the design that best meets their needs.

Our goal is to bring together, in one place, (1) a discussion of the issues analysts *should* be aware of as they prepare to collect and analyze output from a simulation model, and (2) a guide for selecting appropriate designs. Ideally, the analysts implement the data collection by considering at least some of these issues before coding the simulation model. In particular, we assert that analysts must consider the following issues in order to come up with a truly effective analysis:

- The type of questions that they (or their clients) would like to answer
- Characteristics of their simulation setting
- Characteristics of, and constraints imposed on, the simulation data collection process
- The need to convey the results effectively

These issues seem straightforward, but we assert that there are some fundamental problems related to designing simulation experiments that are all-too-often overlooked. We discuss these more fully in the sections that follow, with a focus on the practical benefits that can be achieved through DOE. We believe that using a design suited to a particular application is much better than trial-and-error or limiting oneself to a simple, small design. Consequently, we should have no trouble convincing practitioners that DOE is a useful and necessary part of any analysis of complex simulation systems.

We do not intend this article to be a tutorial on the details for implementing specific designs, nor do we present a historical development of DOE and its application to simulation experiments. Instead, we try to provide an overview of the wide variety of situations that simulation analysts might face, the benefits and drawbacks of various designs in these contexts, and links to references for further details. Our overarching goal is to change the mindset of simulation analysts and researchers so they consider DOE to be an integral part of any simulation project.

This overview is based on our joint experience accumulated through contacts with many simulation users and researchers over the last few decades. Where we disagree with current practice and

theory, we present both sides to further stimulate reflection and discussion. Despite the wide variety of designs that are available in the literature and—in some cases—statistical or simulation packages, we identify some situations where needs are still unmet. Hopefully, this will motivate further research to address these deficiencies.

In Section 2 we define some terminology, and describe how designing simulation experiments is different from designing experiments on real-world systems. Specifically, we address the types of questions that simulation analysts or clients should ask. We also describe a number of other characteristics of simulation settings that cannot easily be handled through more traditional methods, and provide examples to motivate the need for designs that cover a wide range of simulation settings. In Section 3 we discuss some characteristics of designs, including criteria that have been used to evaluate their effectiveness. In Section 4 we describe several classes of designs, and assess their strengths and weaknesses with respect to their appropriateness for various simulation settings and their design characteristics. In Section 5 we describe ways of checking the assumptions that were made when the experimental design was chosen. We conclude, in Section 6, with a discussion of areas that merit further work in order to achieve the potential benefits—either via additional research or via incorporation into standard simulation or statistical software packages. A list with many references enables further study.

2. Why is DOE for Simulation so Different?

First we define some terminology. An *input* or a *parameter* in simulation is referred to as a *factor* in DOE. A factor can be either qualitative or quantitative. For example, consider a queueing system simulation. If queue discipline can be either LIFO (last in, first out) or FIFO (first in, first out), this is a qualitative factor. The number of servers is a discrete quantitative factor, while the rate for an exponential distribution used to model customer inter-arrival times is a continuous quantitative factor. In any case, each factor can be set to two or more values, called *factor levels*. Typically, factor levels are coded numerically for analysis purposes. A *scenario* or *design point* is obtained by specifying the complete combination of levels for all factors. We consider stochastic simulations, and hence *replicates* mean that different Pseudo Random Numbers (PRN) are used to simulate the same scenario. Unless otherwise specified, we will assume that these replicates use non-overlapping PRN streams, so that we have Independently Identically Distributed

(IID) outputs across replicates—as most statistical methods assume. The output stream from a single replicate is typically much less well behaved.

Of course, the simulation is itself a model of some real-world (or prospective) system, process, or entity. We can view the simulation code as a *black box* that implicitly transforms input (such as factor level settings and PRN) into output. A *metamodel* is a model or approximation of this implicit Input/Output (I/O) function (also called response surface, auxiliary model, emulator, etc.). Often, one result of a simulation experiment is the construction of a metamodel. Hopefully, a parsimonious metamodel can be built that describes the relationship between inputs and outputs in much simpler terms than the full simulation. The term ‘metamodel’ is commonly used when most or all of the factors are quantitative, and one goal in choosing a particular design might be estimation of certain types of metamodels. We emphasize the following chicken-and-egg problem: once the design is specified and simulated, metamodel parameters can be estimated; however, the types of metamodels that the analyst desires to investigate should guide the selection of an appropriate design.

The field of DOE developed as a way to efficiently generate and analyze data from real-world experimentation. In simulation—with its advances in computing power—we are no longer bound by some of the constraints that characterize real-world experiments. This is both an opportunity and a challenge for analysts interested in applying DOE to simulation experiments. Indeed, it is an opportunity to gain much more insight into how systems behave, and so provide assistance and information to decision-makers that might differ dramatically (in terms of its quantity and nature) from information obtainable using more traditional methods. It is a challenge because it may require a new mindset: we argue that the way simulation experiments should be approached is now fundamentally different from the way that real-world experiments—involving, say, human subjects—should be approached.

To illustrate the difference between classic DOE and simulation DOE, we consider the classic ‘bias minimizing’ designs. For example, Donohue, Houck, and Myers (1993)—assuming a first-order polynomial metamodel, but at the same time allowing for possible bias caused by second-order effects—derive designs that minimize that bias. We, however, argue that, in general, such designs are relevant in real-world experiments but not in simulation. In the former experiments, the analysts must often select a design that is executed in ‘one shot’ (say, one growing season in agriculture). In simulation, however, the

data are collected sequentially so the analysts may start with a design for a first-order metamodel; then test (validate) the adequacy of that model; and only if they reject that model, they augment their design to a design that allows the estimation of second-order effects; also see Kleijnen and Sargent (2000).

2.1 Asking Appropriate Questions

All simulation texts mention the importance of identifying the ‘right’ problem to solve—before constructing a simulation and conducting the analysis. For example, Law and Kelton (2000) state that the first step in a simulation study is to formulate the problem and plan the study. As part of this step, they mention that the problem of interest should be stated by the project manager, and that the analysts should specify the overall study objectives, specific questions to be answered, performance measures that will be used to evaluate the efficacy of different system configurations, system configurations to be modeled, and the time and resources required for the study. They go on to say that experimental design, sensitivity analysis, and optimization deal with situations in which there is ‘...less structure in the goal of the simulation study: we may want to find out which of possibly many parameters and structural assumptions have the greatest effect on a performance measure, or which set of model specifications appear to lead to optimal performance.’ (Law and Kelton 2000 Chapter 12).

We recommend an even broader view, since we have found that the type of question people most often think about concerns an *a priori* single specific performance measure for which they then try to estimate or optimize its mean. Our starting point, however, is three basic goals that simulation analysts and their clients may have:

1. *Developing a basic understanding* of a particular simulation model or system.
2. *Finding robust* decisions or policies.
3. *Comparing* the merits of various decisions or policies.

Developing a Basic Understanding

This goal covers a wide range of questions. We use this phrase rather than ‘testing hypotheses about factor effects’ for the following reason.

At one extreme, we may be developing a simulation to gain insight into situations where the underlying mechanisms are not well understood, and where real-world data are limited or even non-existent. For example, when he was Chief Scientist of the Marine Corps, Dr. Alfred Brandstein posed the question ‘When and how should command and control be centralized or decentralized?’ We do not know enough about the human mind to program a model for how decisions *are* made by an individual—let alone a group of people! Yet, ignoring these types of questions because they are ‘too hard’ or ‘inappropriate for Operations Research’ is unacceptable: our profession's roots are in finding ways to address difficult, interdisciplinary problems.

One way that similar questions are explored is via the development of *agent-based* models that try to mimic and potentially explain the behavior of complex adaptive systems. For each agent (e.g., object or person), simple ‘rules’ or ‘behaviors’ are specified instead of building detailed prescriptive models to cover the interactions among all the agents. Applications of agent-based simulations have been developed to provide insights into the evolution of organisms, behavior of crowds in stadiums, swarming behavior of insects, food distribution, and counter-terrorism activities, and more (see Horne and Leonardi 2001 for a discussion and examples of very simple agent-based models called ‘distillations’). For these types of simulations, DOE can be an integral part of the modeling development process. Indeed, we have found DOE useful in several ways: it can uncover details into how the model is behaving, cause the modeling team to discuss in detail the implications of various model assumptions, help frame questions when we may not know ahead of time what questions should be asked, challenge or confirm expectations about the direction and relative importance of factor effects, and even uncover problems in the program logic. Note that in such an exploratory environment, it does not make sense to think about using the models to numerically estimate factor effects—we are looking for tendencies rather than values.

At the other extreme, suppose we have a model that we are comfortable using for prediction. Then ‘understanding the system’ may result from performing a detailed *sensitivity analysis* of a particular system configuration. How should we proceed? Searching for effects by varying factors one-at-a-time is an ineffective means of gaining understanding for all but the simplest systems. First, when using this approach it is impossible to identify any interaction effects (synergy or redundancy) between two or more factors. Second, even in the case when one-factor-at-a-time sampling can be used to construct an unbiased

metamodel, it is inefficient (in terms of the variance of the estimated effects, given the amount of data). So, we assert that—from the outset—the analyst *must* explore factor effects concurrently in order to understand how their simulation model behaves.

Between these two extremes are situations where the analysts wish to come up with a *short list of important factors* from a long list of potential factors. Depending on the context, this situation might lead to a more thorough investigation of this short list via additional simulation experiments, a decision to forego adding enhancements or greater detail to aspects of the model that were not found to be important, or the collection of (additional) real-world data in order to home in on appropriate values of (say) influential input distributions. Alternatively, simply identifying the most influential factors (and their directional effects on performance) may suffice. It is also important to know which factors are ‘certainly’ unimportant (at least over prespecified factor level ranges) so the users are not bothered by details about these factors. Of course, the importance of factors depends on the *experimental domain* (or experimental frame, as Zeigler 1976 calls it). For example, oxygen supply is important for missions high in the sky and deep under water, but not on land at sea level. So the clients must supply information on the intended use of the simulation, including realistic ranges of the individual factors and limits on the *admissible scenarios*. This includes realistic combinations of factor values; for example, some factor values must add up to 100%.

Finding Robust Decisions or Policies

We discuss robust policies, rather than optimal policies, for a reason. It is certainly true that finding the *optimal* policy for a simulated system is a hot topic, and many methods have been proposed—including (alphabetically) Genetic Algorithms, Perturbation Analysis, Response Surface Methodology (RSM), Score Functions, Simulated Annealing, and Stochastic Approximation (see Fu 2002 for a discussion of the current research and practice of optimization for simulation). However, all these methods implicitly condition on a large number of events or environmental factors. In practice, the future environment is uncertain so that this so-called optimal policy cannot be achieved and may break down completely! Therefore, we wish to find a *robust* policy—that is, one that works well across a broad range of scenarios. Such policies have also been called ‘satisficing’ (see Simon 1981).

To illustrate this problem of classic optimization, consider laying out a small factory, where simulation is used to explore the alternatives. The project manager's decision factors relate to the type, number, position, and buffers associated with machines on the factory floor, as well as any schemes for prioritizing or expediting orders. This is a prototypical problem often analyzed using a simulation optimization method—but the result of the 'optimization' is conditioned on the assumptions of specific (typically assumed independent) distributions for order arrivals, order sizes, machine uptimes, downtimes, and service times, and many more. We argue that the use of the term 'optimum' is problematic when the probability of all these assumptions holding in practice, even for a limited time, is zero. Suggesting different possible 'optimal' layouts for (say) several potential customer order patterns may be singularly unhelpful, since the decision-maker cannot control (or perhaps even accurately predict) future order patterns.

In contrast, a *robust* design approach treats all these assumptions as additional factors when running the experiment. These factors are considered noise factors—rather than decision factors—because they are unknown or uncontrollable in the real-world environment. A robust system or policy is one that works well across the *range* of noise conditions that might be experienced. Therefore, implementing a robust solution is much less likely to result in surprising (unanticipated) results.

We do not mean to imply that an optimization approach will necessarily yield a bad answer. If sensitivity analysis of the so-called optimal solution indicates that it still performs well (in an absolute sense) when realistic departures from these assumptions occur, then the optimization algorithm has identified a solution that is likely to perform well in practice. If changes in the environment (e.g., different patterns of customer orders) impact all potential solutions in the same manner, then the relative merit of particular policies does not change. Nonetheless, there are situations where optimizing and then performing sensitivity analysis can lead (and has led) to fundamentally different answers. For example, a military problem of interest these days is finding the 'optimal' strategy for defending a high-value target (courthouse, church, monument) against a single terrorist. If the analysts condition on the route the terrorist will take approaching the building, then forces will be concentrated along this path. However, if the direction of approach is unknown, then an entirely different strategy (dispersing the protective forces) is

much more effective. Another example is Kleijnen and Gaury (2002)'s study on robustness in production planning that was published in the open literature.

This *robust design philosophy* is inspired by Taguchi (1980), who pioneered an approach of using simple, orthogonal designs as a means of identifying robust product configurations for Toyota. The results improved the quality while lowering the cost of automobiles and component systems, because the chosen product designs performed well—despite variations in incoming raw material properties, the manufacturing process, and the customers' environments. This robust design approach is discussed for simulation experiments by Sanchez (2000). Metamodels can suggest scenarios (i.e., new combinations of factor levels) that have not yet been investigated—though the analyst should make confirmatory runs before applying the results.

Comparing Decisions or Policies

We avoid the phrase 'making predictions about the performance of various decisions or policies'. Comparisons may need to be made across a number of dimensions. Rather than formal statistical methods for testing particular factor effects or estimating a specific performance measure, our goal might be to provide the decision-maker with detailed descriptive information. For example, we could present the means, variances, percentiles, and any unusual observations (see the box plots in Law and Kelton 2000) for several different performance measures, for each of the systems of interest. These measures could then be reported, in conjunction with implementation costs and other considerations not included in the simulation model.

If at least some of the factors are quantitative, and if a performance measure can be clearly stated, then it is possible to construct metamodels of the performance that describe the I/O relationships in terms of functions of various factor levels. Here, rather than running an experiment in order to gain insight into how the performance is affected by all the factors, we may focus on a few of immediate interest to the decision-maker.

However, *Ranking and Selection Procedures*, *Multiple Comparison Procedures (MCP)*, and *Multiple Ranking Procedures (MRP)* assume that the analysts wish to compare a fixed small number of 'statistical populations', representing policies or scenarios. There are two basic approaches: (1) how to

select, with high probability, a system/decision/policy that is, for practical purposes, the best of the group of potential systems; and (2) how to screen the potential systems/decisions/policies to obtain a (random-size) subset of ‘good’ ones. Nelson and Goldsman (2002) provide a review of these procedures in simulation settings (see also Hsu 1996, Chick and Inoue 2001 and Goldsman et al. 2002). Many procedures have been developed specifically to address some of the characteristics of simulation experiments we will discuss in Section 3. Some assume that all populations are compared with each other, whereas other procedures assume comparisons with a standard; see Nelson and Goldsman (2001).

The three types of questions that we have posed differ from those problems others have suggested in the literature. Sacks et al. (1989) classify problems for simulation analysts as prediction, calibration, and optimization. Kleijnen (1998) distinguishes among sensitivity analysis (global, not local), optimization, and validation of simulation models. These two classifications are related to the ones we use—for example, global sensitivity analysis can be used as a way of gaining understanding about a problem—but there is not a one-to-one mapping. For certain classes of simulations, such as many military simulation models or hazardous waste disposal models, data are extremely limited or nonexistent. This means that calibrating, optimizing, or predicting with a model may not be a meaningful goal. In the best tradition of scientific discovery, we feel that simulation experiments can, nonetheless, have a role in supporting the *development* of insights (or theories) in these situations. Dewar et al. (1996) discuss how one can credibly use models that cannot be validated due to a dearth of data or changing conditions.

The above situations contrast sharply with many of the simulation experiments that appear in the literature. These publications assume that a simulation model exists that has already been thoroughly validated and verified. The decision-makers have very specific questions; for example, about the impact on a particular performance measure that results from changing a small number of factors to specified (new) values. The users might hypothesize the nature and strength of a particular factor effect, and the analysts' charge is to run the simulation model and collect I/O data in order to test this hypothesis.

2.2 The Simulation Setting

In this section, we describe some of the characteristics of simulation settings that call for non-traditional designs as part of the analyst’s toolkit. To motivate our discussion, we include practical examples that

have been worked on for real clients in the recent past, or are currently under investigation. These examples are drawn from industrial and military applications.

Number of Potential Factors

In real-world experiments, only a small number of factors are typically varied. Indeed, it is impractical or impossible to attempt to control more than, say, ten factors; many published experiments deal with fewer than five. Academic simulations, such as single-server queuing models, are also severely limited in terms of the number of factors that can be varied. Both application domains obviate the need for a larger set of designs in the analyst's toolkit.

In contrast, for realistic simulations the list of potential factors is typically very long. For example, the MANA software platform was developed to facilitate construction of simple agent-based models (Steven and Lauren 2001). The agents' rules for movement are a function of a 'personality' or propensity to move—based on ten possible goals (toward/away from a location, a friend, an enemy, a road, etc.). The personalities can be in one of ten states, and change when a trigger event (such as detecting an enemy or being shot at) occurs. In all, over 20 factors could be modified for each agent for each of the ten personality states, so we are dealing not with ten factors, but with thousands of factors - and this is considered a 'simple' modeling platform!

Other examples abound. Bettonvil and Kleijnen (1997) describe an ecological case study involving 281 factors. Cioppa (2002) examines 22 factors in an investigation of peace-enforcement operations. Even simple queuing systems can be viewed as having a few dozen factors—if the analysts look at arrival rates and distributions that change over time, service distributions, and correlations arising when service times decrease or servers are added as long lines of customers build up.

We emphasize that good computer programming avoids fixing the factors at specific numerical values within the code; instead the computer reads factor values so that the program can be run for many combinations of values. Of course, the computer should check whether these values are admissible; that is, do these combinations fall within the experimental domain? *Such a practice can automatically provide a list of potential factors.* Next, the users should confirm whether they indeed wish to experiment with all

these factors or whether they wish to *a priori* fix some factors at nominal (or base) levels. This type of coding helps *unfreeze* the mindset of users who would otherwise be inclined to focus on only a few factors.

Choice of Performance Measures

Consider both the number and the types of performance measures. Most procedures (e.g., MCP, MRP, RSM) involve a single quantitative performance measure; the goal is to maximize or minimize the expected value of this measure.

However, in many simulation applications, it is impossible to identify a single measure of performance. For example, textbook examples of simple queuing systems often discuss minimizing the average waiting time. In practice, alternatives include minimizing the proportion of customers that wait more than a specified length of time, maximizing the number that are served within a particular length of time, improving customer satisfaction by providing information about their projected wait time and allowing them to reschedule, minimizing the number of errors in processing customer transactions, balancing workloads across servers. Another example is provided by the various performance measures in supply chain management; see Kleijnen and Smits (2002). Consequently, it is restrictive to use a DOE framework that suggests the appropriate goal of the study should be examining the expected value of a single performance measure.

In theory, a design will be used to generate multiple outputs but they will be accounted for in the analysis. For example, *multivariate regression analysis* may be applied. In practice, however, each output is usually analyzed individually. For linear regression analysis, Khuri (1996) proves that this practice suffices if all outputs are generated by the same design. The same design is indeed used when running the simulation and observing multiple outputs. More research is needed for non-linear regression analysis, such as Kriging and neural nets (see, e.g., Cressie 1993, Simpson et al. 1997).

Taguchi's robust design approach (Taguchi 1987) offers another alternative when multiple performance measures exist. If responses are converted to losses and appropriately scaled, then the analysts can construct models of overall expected loss. However, we prefer constructing separate metamodels for each performance characteristic, because it makes it easier to identify *why* certain scenarios have more or less desirable performance than others.

A few publications use a mathematical programming framework to analyze multiple simulation outputs; that is, one output is minimized whereas the remaining outputs should satisfy prefixed constraints. For example, the inventory is minimized while the service percentage meets a pre-specified level. See Angün et al. (2002).

Some problems require only *relative* answers; that is, the users want to know whether one policy is better than another. For example, in a study on the search for sea mines, the users wanted to know which tilt angle of the sonar gives better results, see Kleijnen (1995). Conversely, some problems require *absolute* answers. For example, in the same case study, users wanted to know whether the probability of mine detection exceeds a certain threshold—before deciding whether to do a mine sweep at all.

Response Surface Complexity

Assumptions about the metamodel's complexity are generally broken down into those regarding its deterministic and its stochastic components, respectively. These assumptions often drive the analysis. The standard assumptions in the DOE analysis are that the deterministic component can be fit by a polynomial model of the factor levels (perhaps after suitable transformations of the factors or responses) and that the stochastic component can be characterized as additive *white noise*. The latter assumption means that the residuals of the metamodel are Normally IID. In practice, Normality may be explained by the central limit theorem. However, the IID assumption is violated when the noise has larger variances in subspaces of the experimental area: *variance heterogeneity*. Such heterogeneity is pervasive in simulations. For example, in queuing problems the intrinsic noise increases dramatically as the traffic load approaches 100 percent (Cheng and Kleijnen 1999, Kleijnen, Cheng, and Melas 2000). Moreover, common random numbers (CRN) are often used for generating output from several simulation scenarios, since CRN can sharpen the comparison among systems. CRN, however, violate the independence assumption.

Good modeling practice means that the analyst should strive to find the simplest metamodel that captures the essential characteristics of the system (Occam's razor). Therefore, we need a suite of design tools: some appropriate for simple response surfaces, other for more complex systems. We remark that simpler metamodels are often easier to justify when only a small number of factors and performance measures are examined; yet, interpreting the results may be problematic because the analyst may easily

miss important system characteristics. As we will discuss further in Section 4, some designs provide some assessment about the suitability of the estimated metamodel. In principle, we prefer classifying factors into a continuum: those thought to be very important, those that might be important, those that are thought to be unimportant but are sampled anyway, and those that we are quite comfortable in ignoring. Designs that sample differently across these classifications make intuitive sense.

It is becoming increasingly apparent that some systems exhibit highly non-linear behavior. For example, Vinyard and Lucas (2002) made billions of runs and found that chaotic behavior was rampant across many performance measures in a simple deterministic model of combat. Adding a stochastic component often mitigated this behavior, but sometimes aggravated some measures of the non-monotonicity in the performance measures. Designs that examine only a small number of scenarios are unable to reveal such behavior: instead, the analysts may believe they are facing a simulation model with a large stochastic component.

Steady State vs. Terminating Simulations

Terminating simulations are those that run until a specific event has occurred (including the event of simulating a fixed amount of time). Examples include simulating a single day's operation of a retail establishment, or a model of a space satellite that ends when the satellite is destroyed or becomes non-functional. Steady-state simulations have no natural termination point, and can keep generating data for their analysis. The simulation type has implications on the design and analysis. For terminating simulations, it may be necessary to censor results if we are simulating rare events; see Kleijnen, Vonk Noordegraaf, and Nielen (2001). Multiple performance measures may again come into play: it may be important to know not just who wins the battle, but how long it takes to finish. For steady-state simulations, the warm-up period must be chosen carefully, and the length of the warm-up period affects the total time for experimentation.

Inclusion of Simulation-Specific Factors

The analysts have control over many things during the course of a simulation study—in addition to the factor levels they manipulate and the performance measures they collect. This control includes the

maximum run time for terminating simulations; the warm-up period, run length, and batch sizes for steady-state simulations; common and/or antithetic random number streams; and perhaps the use of other variance reduction techniques (VRT) developed for simulation output analysis such as control variates and importance sampling. However, not all designs can easily accommodate these VRT. For steady-state simulations, even the appropriate choice of run length vs. batch size for a fixed budget is not easy without some information about the transient component of the simulation (Steiger and Wilson 2001).

2.3 External Concerns and Constraints

We now discuss issues that often play a major role in the implementation of simulation experiments, though they are generally not discussed in the literature.

Sequential vs. One-shot Data Collection

In real-world experiments, the basic mindset is often that data should be taken simultaneously, unless the design is specifically identified as a sequential design. When samples must be taken sequentially, the experiment is viewed as prone to validity problems: the analysts must randomize the order of sampling to guard against time-related changes in the experimental environment (such as temperature, humidity, consumer confidence, and learning effects), and perform appropriate statistical tests to determine whether or not the results have been contaminated.

However, most simulation experiments are implemented sequentially—even if they are not (formally) analyzed that way. If a small number of design points are explored, this implementation may involve the analysts manually changing factor levels. Alternatively, and less prone to data entry errors, an input file or series of input files could be generated automatically once a particular design has been chosen. These files may be executed sequentially (and efficiently) in batch mode. Modifying simulations to run in parallel over different computers is possible, but not typical. For example, at the supercomputing clusters of the Maui High Performance Computing Center and in Woodbridge, Virginia, parallelization is being used effectively. In many cases, however, ‘parallelization’ results from an analyst manually starting different runs (or sets of runs) on a few assorted computers to cut down on the overall time to complete the data collection. For example, Vonk Noordegraaf, Nielen, and Kleijnen (2002) use five PCs @ 533 MHZ to

finish their 64 scenarios—each scenario replicated twice—in two weeks. We remark that freely available software, such as that used on literally thousands of PCs as part of the search for extraterrestrial intelligence (SETI), could be used to facilitate parallel data collection for simulation experiments, but this is not readily available for use in either the industrial or academic settings with which we are familiar.

Premature Stopping of the Experiment

Another issue arises whenever the simulation takes a non-trivial amount of time to run. The analysts may have to terminate their experiment *prematurely*—because the computer breaks down, the client gets impatient, etc. We have found this to be true of many defense simulation projects. In such cases, it is better for the analyst to have organized the list of scenarios in such a way that the experimental I/O data can provide useful information even if that list is curtailed. For example, suppose a design requires two months of CPU time to complete. Given that the scenarios could be run in any order, the analyst would want to avoid having a factor believed by the client to be extremely important held to a single level for all runs taken during the first month. With this view, even non-sequential designs can be implemented sequentially in ways that are robust to early termination. Clearly, sequential or partially sequential designs have this characteristic: after one stage of sampling the analysts indicate which configuration(s) should be examined next. Also, some single-stage designs can be viewed as augmentations of simpler designs, so there is a natural way to separate the design into two or more parts (see the resolution 4 designs in Section 4).

Data Collection Effort

The information revolution has improved our ability to run simulations quickly: those simulations that used to take months now take hours; those that used to take hours now take seconds. This change has caused some analysts to add more details *to* their simulation models: we believe it should spur us to ask more *from* our simulation models.

Within the current computing environment, the traditional concept of a fixed sampling budget is unnecessarily restrictive. The primary indication of the data collection effort is likely to be the total time required to select a design, implement the design, and make the simulation runs. Working backwards, the total run time is not a fixed constraint: it can be cut in half, for example, if the analysts have access to two

computers. The time per run is not typically fixed: different analysts might choose to use different run lengths and/or batch sizes; run times might vary across scenarios because some tend to yield fewer events in steady-state simulations, or lead to early termination for non-steady-state simulations. The implementation effort is a function of both the time required to generate a design and that associated with setting the factor levels accordingly at the beginning of each run. Implementing a design may be very easy if software is available to generate coded factor levels, convert them to original factor levels, and then generate input files so the simulation can be run in batch mode. Conversely, if the analysts must edit and recompile the code for each scenario, or make all changes manually through a graphical user interface, then the implementation time can surpass the time needed for making runs. We will discuss design choices in Section 4.

One way of describing this data collection effort has been to determine the time required to estimate the metamodel parameters to a certain level of precision. However, it is difficult to use this time in making generic recommendations, since it depends on the underlying (heterogeneous) variability. In recent experience, we have dealt with simulations where run time varies from less than a second to half a day per scenario on a single processor.

A related issue is choosing between a design with more replicates per scenario, and a design with more scenarios and fewer replicates—supposing that the total computer time remains the same for the two alternative designs. Replication enables the estimation of possibly non-constant response variances. If the primary goal of the study is *finding robust* systems or policies, then some replication is essential. If the goal is *understanding* the system, this may include understanding the variance. However, if the goal is that of *understanding* or *comparing* systems and a constant variance can be assumed, then this constant can be estimated through the mean squared residuals (MSR)—provided no CRN are used and the metamodel is correctly specified. If classic Ordinary Least Squares (OLS) is applied, it is then better to spend scarce computer time to explore more scenarios instead of getting more accurate estimators of the responses for fewer scenarios. Note that a single replicate does yield an unbiased estimator of the response of a specific scenario; additional replicates provide more accurate estimates.

2.4 Conveying Results Effectively

The best experiment will come to naught if the results are not communicated properly to the decision-maker. We refer back to the three primary goals: developing a basic understanding, identifying robust solutions, and comparing systems. For the first goal, the best analogy might be exploratory data analysis. Graphical tools that allow multi-dimensional visualization of the results may be much more helpful than equations or tables of numbers. Tools we have found useful include 3-dimensional rotatable plots and trellis plots (Sanchez and Lucas 2002); we have also found that regression trees and Bayesian networks have been effective ways of communicating which factors are most influential on the performance measures (Gentle 2002, Martinez and Martinez 2002). However, *visualization* of simulation results remains a challenge at this stage of simulation experimentation. Tufte (1990) is the seminal reference for excellence in graphical presentation; see also Meyer and Johnson (2001) for tools developed specifically for visually exploring large amounts of data from simulation experiments with multiple performance measures.

3. Criteria for Evaluating Designs

Once the simulation analysts know their situation, the question is: now what? Above we stated that there is no single prototypical situation (in terms of the type of question to be asked, or simulation characteristics) that analysts might face. In this light, it is not surprising that we cannot recommend a specific design. How, then, should analysts choose a design that is appropriate for their situation? While we do not have all the answers, we do attempt to provide some guidance. Others have listed desirable attributes for designs for experiments with real systems (see, e.g., Box and Draper 1975, Myers and Montgomery 2002). We shall describe some criteria that have been or might be used to evaluate designs in simulation settings, and discuss how they may (or may not) apply directly to the issues described earlier.

Number of Scenarios

In the literature, a major design attribute is the number of scenarios required to enable estimation of metamodel parameters. A design is called *saturated* if its number of factor combinations (say) n equals the number of metamodel parameters, q . For example, if the metamodel is a first-order polynomial in k factors,

then q equals $k + 1$ (where 1 refers to the grand or overall mean, often denoted by β_0), so a saturated design means $n = k + 1$. Actually, there are several saturated designs for a given metamodel type. For the first-order polynomial in k factors, one saturated design changes one factor at a time, whereas another design is a fractional factorial (for example, if k is 3, then the latter design is a 2^{3-1} design; see Box, Hunter, and Hunter 1978). To choose among different designs, we consider other quality attributes.

Orthogonality

If the columns of the design matrix are orthogonal, it is easier to interpret the results from fitting a metamodel (e.g., using OLS regression). Orthogonality has long been a desirable criterion for evaluating designs: it reduces the chance of improper interpretation of the results caused by (partial) confounding, it simplifies computations, and it improves the statistical efficiency of the estimated specific metamodel terms. However, requiring orthogonality can have limitations as well. It may be that in reality some factor level combinations are not permissible. For example, in the M/M/1 queue the expected steady-state waiting time is infinite if the arrival rate exceeds the service rate. A more complicated application (simulating part of the Rotterdam harbor) with exploding waiting times for the original orthogonal design appears in Kleijnen, van den Burg, and van der Ham (1979). In general, forcing the use of an orthogonal design may mean limiting many factors to narrower ranges, or figuring out a way to deal with unstable results at certain scenarios. However, in complex models it may not be possible to know *a priori* which factor level combinations are problematic.

A design may be orthogonal in the *coded* factor values (such as -1 and +1) but not in the original factor values. Simulation analysts should be aware of possible scaling effects. For example, to find the most important factors, all factors should be coded; see Bettonvil and Kleijnen (1990).

Orthogonality may improve statistical efficiency, as we stated above—and detail next.

Efficiency

The design determines the standard errors for the estimated metamodel parameters. The DOE literature uses several criteria (see Kleijnen 1987, p. 335). For example, *A-optimality* means that the sum of these standard errors is minimal. *D-optimality* considers the whole covariance matrix of the estimated

parameters (not only the main diagonal); it means that the determinant of this matrix is minimal. *G-optimality* considers the Mean Squared Error (MSE) of the output predicted through the metamodel (also see Koehler and Owen 1996). Of course, these criteria require strong *a priori* assumptions on the metamodels to be fit to the data and the nature of the response (e.g., homogeneity of variance). Consequently, they are of little value when there is substantial uncertainty *a priori* on the nature of the simulation's output.

The above criteria certainly can be—and have been—used to evaluate designs proposed for analyzing simulation experiments. However, the classic DOE assumptions (polynomials with white noise) are usually violated in simulation. Moreover, focusing on minimizing the number of design points (or maximizing the efficiency for a fixed number of design points) may not be enough to insure 'efficient' data collection, at least for steady-state simulations: does it make sense to worry about using the most efficient design if one does not also worry about using the smallest run length to achieve the desired goal? In short, efficiency is most critical when the runs are very time-consuming. When we are able to gather lots of data quickly, other criteria become more relevant.

Space-filling and Bias Protection

Conceptually, space-filling designs are those that sample not only at the edges of the hypercube that defines the experimental area, but also in the interior. Popular measures for assessing a design's space-filling property include the *maximum minimum Euclidean distance* between design points (see Johnson et al. 1990) and—from uniform design theory—the *discrepancy* (Fang and Wang 1994). Note that for computational reasons, the *modified L2 discrepancy* is often used as a surrogate for discrepancy (Fang et al. 2000). Cioppa (2002) shows that the use of both measures provides a better ability to distinguish among candidate designs.

A design with good space-filling properties means that the analysts do not need to make many assumptions about the nature of the response surface. Such designs also provide flexibility when estimating a large number of linear and nonlinear effects, as well as interactions, and so provide general bias protection when fitting metamodels of specific forms. Other designs do not have good space-filling

properties, but still protect against specific violations of model complexity assumptions: see the designs of resolution 3, 4, and 5 below.

At this point in time, space-filling designs also provide the best way of exploring surfaces where we do not expect to have smooth metamodels—but spikes, thresholds, and other chaotic behavior. That is, space-filling designs are particularly useful for fitting nonparametric models, such as locally weighted regressions.

Ability to Handle Constraints on Factor-level Combinations

As we mentioned above, in some situations (for example, chemical experiments) factor values must add up to 100 percent. The classic DOE literature presents *mixture* designs for these situations. Many designs exist for exploring experimental regions (i.e., permissible combinations of design points) that are hypercubes or spheres. In simulation experiments, however, realistic combinations of factor values complicate the design process dramatically. This is an area seriously in need of further research. Sanchez et al. (2001) propose elliptical designs, motivated by observational economic data. In many queuing situations, certain combinations of factor settings give unstable outputs (again see Kleijnen, van den Burg, and van Ham 1979, Sanchez et al. 2003). Until designs that can handle such situations are available, visual presentation of the results—and exploratory data analysis—may be the most appropriate ways of determining whether or not these situations exist.

Ease of Design Construction and Analysis

Designs should be easy to construct if they are to be used in practice. We will use this criterion in deciding which designs to recommend in Section 4. The *analysis* is easy if computer software is available for many platforms. Regression software is abundant, so the most common analysis tool is readily available and need not be discussed further. Newer surface-fitting models may also be applied in simulation: *Kriging* assumes covariance-stationary processes for the fitting errors (instead of white noise), and can fit response functions with multiple local hilltops. However, Kriging software for simulation experiments is limited to academic software—with its inherent lack of user support.

Because Kriging metamodels are not well known in our application area, we list some important publications. Cressie (1993) is a textbook of 900 pages on spatial statistics including Kriging and its origin in South Africa, where an engineer called Krige developed his technique while searching for gold. Sacks et al. (1989) is the classic paper on Kriging applied to deterministic simulation models for the design of computer chips, cars, airplanes, etc. Simpson et al. (1997) compare Kriging with Taguchi and neural nets, while Giunta and Watson (1998) compare Kriging with polynomial metamodels. Jin, Chen, and Simpson (2000) compare Kriging with polynomial metamodels, splines, and neural nets. More recently, Van Beers and Kleijnen (2002) apply Kriging to stochastic simulation.

4. Design Toolkit: What Works and When

Now that we have identified several characteristics of simulation settings and designs, it is time to match them together. Consider Figure 1, in which we chart some designs according to two dimensions that together describe the simulation setting. The horizontal axis represents a continuum from simple to complex response surfaces. Since the metamodel complexity depends on both the deterministic and stochastic components, there is not a unique mapping. However, we list some of the assumptions along the axis to inform the users about the types of metamodels that can be fit. The vertical axis loosely represents the number of factors. So, the lower left (near the origin of the figure) represents very simple response surfaces with only a handful of factors—that is, the traditional DOE setting with Plackett-Burman designs developed in the 1940s, etc. The upper right-hand corner represents very complex response surfaces with many factors. We do not present a comprehensive list of all available designs, but rather describe those that seem most promising and are either readily available or fairly easy to generate.

Recall that we hoped to change the mindset of those who might otherwise begin experimentation by focusing on a small number of factors. Therefore, we advocate using designs displayed near the top of this figure. In this way, the analysts can look broadly across the factors in the simulation study. The analysts can start—from the left-hand side of the figure—making some simplifying assumptions, which will tend to reduce the initial data collection effort. (Of course, whenever assumptions are introduced, their validity should be checked later on.) Alternatively, employing CRN or other VRT can make certain procedures more efficient, and perhaps allow the analyst to handle more factors—making fewer

assumptions—for a given computational effort. In practice, however, except in rare event simulations VRT seldom give dramatic efficiency gains.

If this initial experiment does not completely address the main goal, then the analysts can use their preliminary results to design new experiments (augmenting the current data) in order to focus on the factors or regions that appear most interesting. This focus may mean relaxing metamodel assumptions for the short list of factors selected after the initial experiment, while holding the remaining factors to only a few configurations; that is, move south-east in Figure 1.

We now provide brief descriptions of the designs in Figure 1 and their characteristics, along with references for further details.

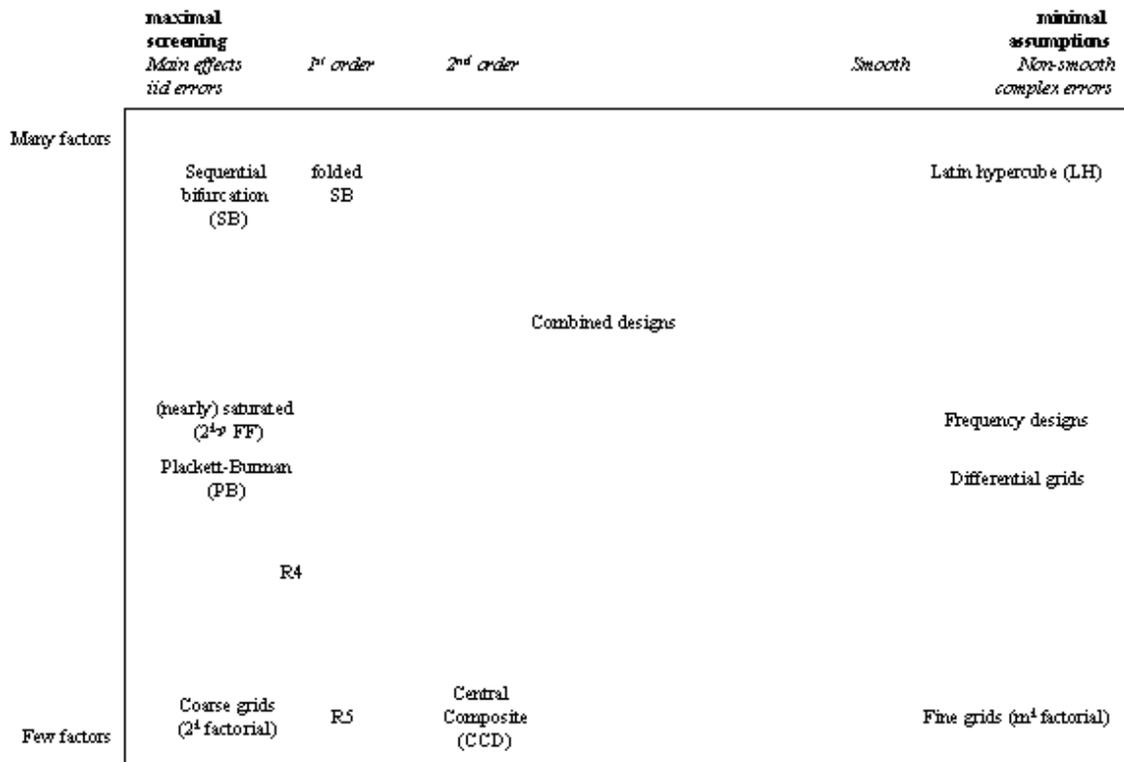


Figure 1: Recommended Designs According to the Number of Factors and System Complexity Assumptions

Gridded or Factorial Designs

Factorial designs are easy to explain to someone unfamiliar with classic DOE. A popular type of factorial is a 2^k design: each of k factors takes on one of two levels, and all resulting combinations are simulated. Then it is possible to fit a metamodel including *all* interactions—not only between pairs of factors, but also among triplets, etc. (these models are linear in the factor effects, not the factor levels).

Considering more complex metamodels (i.e., moving to the right in Figure 1), the analysts may use finer grids: three levels per factor result in 3^k designs, m levels result in m^k designs. When there are more than a few factors, the analysts may use different grids for different groups of factors—employing finer grids for those factors thought to be important.

These finer grids enable us to either view nonlinearities in the response surface or test the linearity assumption. Unfortunately, the number of scenarios n grows exponentially as the number of factors k increases, so factorial designs are notoriously inefficient when more than a handful of factors are involved. Nevertheless, these designs are an important tool since they are easy to generate, plot, and analyze. Hence, whenever individual run times are minimal, the benefit of detailed information about the nature of the response surface may easily outweigh the additional computation time relative to the more efficient designs we discuss next.

Resolution 3 (R3) and Resolution 4 (R4) Designs

For metamodels with *main effects only*, it can be proved that the most efficient designs are R3 designs—provided the *white noise* assumption holds. R3 designs are 2^{k-p} designs if $k + 1$ is a multiple of four; otherwise R3 designs are tabulated as Plackett-Burman designs. See any DOE textbook for details (e.g., Box, Hunter and Hunter 1978).

If *interactions* are assumed to be present, but the users are mainly interested in estimating first-order effects, then R4 designs are appropriate. R4 designs give unbiased estimators of main effects—even if two-factor interactions are present. These designs can be easily constructed through the *fold-over* procedure: after executing the R3 design, the analysts run the mirror design that replaces each plus sign in a specific factor's column by a minus sign; and each minus by a plus sign. In other words, the analysts can

proceed in two stages: first run an R3 design; then augment it to an R4 design. (See also the RSM designs in Donohue, Houck, and Myers 1993.)

Even if the white noise assumption does not hold, classic designs do enable the analysts to estimate the metamodel parameters—although not necessarily with minimum standard errors. If we account for the analysts' time and energy, then these designs seem acceptable. Clearly, R3 designs give smaller standard errors for the estimated first-order effects than the popular practice of *changing one factor at a time*: the former designs use all scenarios to estimate all effects, whereas the latter designs use only two scenarios per effect.

Resolution 5 (R5) Designs

If users are also interested in the individual two-factor interactions, then a R5 design is needed. Many 2^{k-p} designs of the R5 type are not saturated. Saturated designs include Rechtschaffner (1967)'s designs, which are discussed by Kleijnen (1987, pp. 310-311) and applied by Kleijnen and Pala (1999). A R5 design requires $O(k^2)$ factor combinations, so this design is less attractive if individual runs are time-consuming. Sometimes, however, the R4 design suggests that certain factors are unimportant so the R5 design can be limited to fewer factors. The 2^{k-p} designs are relatively easy to construct or can be looked up in tables; see Box et al. (1978), Kleijnen (1974-1975, 1987), and Myers and Montgomery (1995).

Fractional factorial designs (including R3, R4, and R5 designs) meet classic optimality criteria such as D-optimality for specific metamodels. Other designs that satisfy these criteria are derived in *optimal design* theory, pioneered by Fedorov and Kiefer; see Pukelsheim (1993). These 'optimal' designs typically lack the simple geometric patterns of classic designs, and are too complicated for most practitioners.

Central Composite Designs (CCD)

A second-order metamodel includes purely quadratic effects so that *non-monotonic* response functions can be handled. Best known are CCD, which have five values per factor. These values are coded as -1, +1, 0, -c, +c with $c \neq 1$ and $c \neq 0$. It is possible to determine an optimal value of c if the white noise assumption holds. However, since this assumption does not hold for most simulation experiments, we do not worry too

much about the choice of c —except to suggest the analysts choose an intermediate value for better space-filling. Details on CCD can be found in any DOE textbook.

Actually, estimation of quadratic effects requires no more than three factor levels, so to save computer time the analysts may again use *saturated* designs, which implies $n = 1 + k + k(k - 1)/2 + k$. Simple saturated designs are applied by Kleijnen and Pala (1999).

Sequential Bifurcation (SB)

In practice, there are situations with a large number of factors but only a small number of important factors. Moreover, the users may be able to specify the sign (or direction) of each potential main effect, and the metamodel is a simple polynomial with main effects - possibly augmented with two-factor interactions. In those situations, the individual factors can be aggregated into groups such that individual main effects will not cancel out. *Group screening* can be very effective at identifying the important factors. The most efficient group screening procedure seems to be SB. For example, in an ecological case study, 281 factors are screened after only 77 factor combinations are simulated. If interactions may be important, SB still gives unbiased estimators of the main effects—provided the number of combinations is doubled (similar to the fold-over principle for R3 and R4 designs discussed above). SB is also robust to premature termination of the experiment: SB can be stopped at any stage, providing upper bounds for aggregated (not individual) effects. See Bettonvil and Kleijnen (1997) for details, and Kleijnen (1998) for additional references. Cheng (1997) expands SB to output responses that are stochastic.

Other screening techniques with less restrictive metamodels are discussed by Campolongo, Kleijnen, and Andres (2000), Holcomb and Carlyle (2002), Holcomb, Montgomery, and Carlyle (2003), Lin (1995), and Trocine and Malone (2001). Their performance relative to SB needs further research.

Latin Hypercube Sampling (LHS)

For situations involving a relatively large number of factors, McKay, Beckman, and Conover (1979) proposed LHS: let n still define the number of scenarios; define n levels per factor; for each scenario, sample the factor values without replacement (giving random permutations of factor levels). LHS is so

straightforward that it is incorporated in popular add-on software (such as @Risk) for spreadsheet simulation; see Sukiyaama and Chow (1997).

LHS designs have good space-filling properties—particularly if replicates are taken—so they are efficient ways of exploring unknown, but potentially complicated response surfaces with many quantitative factors. For LHS in Kriging—which assumes smooth metamodels with many local hilltops—we refer the reader to Koehler and Owen (1996), Morris and Mitchell (1995), Simpson, Lin and Chen (2001), and also Osio and Amon (1996)’s work on multistage Bayesian surrogates methodology (MBSM).

There are numerous variants of basic LHS. Recently, Ye (1998) developed an algorithm for orthogonal LHS, assuming a linear metamodel. Cioppa (2002) extended the number of factors that can be examined in orthogonal LHS within a fixed number of runs. Moreover, he found that by giving up a small amount of orthogonality (i.e., pairwise correlations between the design columns of less than 0.03), the analysts can dramatically increase the space-filling property of these designs. His LHS designs are tabulated, thus easily used.

Frequency-based Designs

For quantitative factors, a frequency-based approach makes each factor oscillate sinusoidally between its lowest and highest value—at a unique and carefully chosen frequency. If the simulation model is coded so that factors can be oscillated during the course of a simulation run (called the *signal run*), then comparisons can be made to the *noise run* where all factors are held at nominal levels. This approach has been advocated as a screening tool for identifying important metamodel terms (Schruben and Cogliano 1987, Sanchez and Buss 1987).

More recently, frequency-based designs have been used to *externally* set factor levels for scenarios, that is, factor levels remain constant during the course of the simulation run—but they change from run to run (Lucas et al. 2002, Wu 2002). These designs have reasonably good space-filling properties, and there is a natural gradation in the granularity of sampling: factors oscillated at low frequencies are sampled at many levels, whereas factors oscillated at high frequencies are sampled at fewer levels. This property may help the analysts design an experiment to be robust to early termination, for example, by choosing higher oscillation frequencies for those factors believed *a priori* to be most important to

investigate. By carefully choosing the oscillation frequencies, it is possible to use the results to fit second and third order metamodels. The designs are relatively easy to construct and to implement (see Jacobson, Buss, and Schruben 1991, Morrice and Bardhan 1995, Saltelli, Tarantola, and Chan 1999, or Sanchez and Lucas 2002).

Combined or Crossed Designs

Selecting designs for finding *robust* solutions—following Taguchi’s approach—falls naturally into the upper middle portion of Figure 1. While there may be a large number of factors, the analysts are interested in a metamodel that captures the impact of the *decision* factors only. So their metamodel—while it may be complex—does not require estimation of all factor and interaction effects. Actually, the *noise* factors enter into the metamodel via their impact on the *variability* of the response for a particular combination of decision factor levels. This clear division of factors suggests that the analysts sample the two sets differently—for example, by combining (crossing) a 3^k or a CCD for the decision factors with a lower resolution design for the noise factors—as we now show.

Taguchi (1987) proposes a particular class of orthogonal designs, but these designs are for factory experiments and are limited to main-effects models that we find too restrictive for simulation environments. Ramberg et al. (1991) use a sequential approach, beginning with a 2^{k-p} augmented with a center point for the decision factors, and recommend a saturated or nearly saturated factorial for the noise factors. Moeeni, Sanchez and Vakharia (1996) use three levels (varied across runs) per decision factor and frequency-based oscillation (varied within a run) for 35 noise factors. Cabrera-Rios, Mount-Campbell, and Irani (2002, p. 225) propose three levels per decision factor and two levels per environmental factor. If the number of decision factors is not too large, then it is efficient to cross a CCD for the decision factors with LHS for the noise factors. If the number of decision factors is large, then orthogonal or nearly orthogonal LHS may be good choices for the decision factors. In short, these designs are easy to generate, and the two sub-designs can be chosen to achieve the characteristics (space-filling, orthogonality, efficiency) that are most pertinent to the problem at hand.

The above discussion shows how noise factors can be exploited in terms of designing efficient experiments to identify robust solutions. Even though it is often efficient to use a crossed design in order to

search for robust solutions, such a design is not necessary. Combined designs maintain orthogonality between the decision and noise factors, and retain the properties of the separate designs.

The decision/noise factor classification is not the only situation where combined and crossed designs can be exploited. For example, Lucas, Bankes, and Vye (1997) give an example of group screening within a fractional factorial design crossed with LHS. Also, Lucas et al. (2002) discuss the benefits of combining multiple designs after classifying the factors into several groups based on their anticipated impact. This allows the analyst much more flexibility than simply putting each factor into (or leaving it out of) the experiment.

Summary

We have presented several design alternatives for simulation experiments involving either a few or many factors. If runs are extremely time-consuming, then the analysts can reduce the computational effort by making some (hopefully, reasonable) assumptions about the nature of the response surface. These assumptions can be checked after the runs are completed, as we shall describe in Section 5. We contrast this approach to arbitrarily limiting the number of factors: if the analysts change only a few factors while keeping all other factors constant, then the conclusions of the simulation study may be extremely limited.

We remark that we have not attempted to list all the designs that have been proposed for simulation experiments. For example, we have not placed any simulation optimization methods in Figure 1—although we can view ‘optimization’ as a means of comparing systems under very specific conditions. Our goal was to suggest some designs that analysts can readily use.

5. Checking the Assumptions

Whichever design is used, sound practice means that the analysts check their assumptions. If the analysts selected a design from the right-hand-side of Figure 1, then they made very few assumptions about the nature of the response surface. In the process of fitting a metamodel, the analysts determine what (if any) assumptions are reasonable. However, if they started in the upper left corner of Figure 1, then the experiment was likely used to screen the factors and identify a short list as the focus of further experimentation. If so, the analysts are likely to make fewer assumptions during the next stages of

experimentation. If they started from the lower left (as traditional DOE does), then it may be essential to confirm that the resulting metamodel is sufficient—or to augment it appropriately.

One check has the *signs* of the estimated effects evaluated by ‘experts’; that is, experts on the real system being simulated. A simple example is: does a decreased traffic rate (resulting from adding or training servers) indeed reduce the average waiting time? Another example is the case study by Kleijnen (1995) on a sonar simulation experiment: naval experts evaluated the signs of the metamodel effects; because all these signs were accepted, the underlying simulation model was considered to be ‘valid’. In general, checking the signs may be particularly applicable when the goal of the simulation study is general understanding rather than prediction, as for the agent-based models discussed earlier. We, however, remark that sometimes intuition is wrong and needs to be challenged. For example, Smith and Sanchez (2003) describe a forecasting project where the model of losses (incurred for certain groups of loans) had the ‘wrong’ signs. Examination of the detailed files, however, confirmed that their patterns differed from the vast majority of loans and revealed why, so that the model ended up providing new—and valid—insights to the experts. Another example is the ecological case study in which Bettonvil and Kleijnen (1997) employ SB: the resulting short list of factors included some that the ecological experts had not expected to have important effects.

Another check *compares* the metamodel predictions to the simulation outputs for one or more new scenarios (which might be selected through a small LHS design). If the results are close, the metamodel is considered acceptable (see any textbook on linear models or forecasting; also Kleijnen, Feelders, and Cheng 1998). Kleijnen and Sargent (2000) discuss how to use output from initial simulation experiments to test the metamodel constructed from other scenarios in subsequent experiments. They refer to this as validating metamodels, not to be confused with validating a simulation model.

The assumption of *NIID errors* can be examined via residual analysis (if OLS is used to fit the metamodels), or by taking additional replications at a few design points. Tuniz and Batmaz (2000) investigated procedures for validating this and other assumptions for least-squares metamodel estimation.

Note that *higher-order interactions* are notoriously difficult to explain to the users; nevertheless, traditional DOE routinely estimates and tests these interactions. One solution *transforms* the original inputs or outputs of the simulation model. We give two generic examples. First, logarithmic transformations of

inputs and outputs may help in queuing problems; see Irizarry, Wilson, and Trevino (2001) and Kleijnen and Van Groenendaal (1992, pp. 159-162). Second, replacing two individual factors by their ratio may help in queuing where the arrival and the service rates are combined into the traffic rate; in combat models the relative strength may provide a better explanation than the individual absolute strengths of the two combatants. However, when multiple performance measures are collected, it may be difficult or impossible to transform individual factors so that all response surfaces are simple. If so, it may be best to transform certain responses in order to fit simpler models in the transformed space, and then back-transform in order to present the results to the decision-maker.

Even with careful thought and planning, it is rare that the results from a single experiment are so comprehensive that the simulation model and its metamodel(s) need never be revisited. In practice, results from simulation experiments often need to be modified; that is, expanded or thrown out to obtain more detailed information on the simulation performance for a smaller region of the factor combinations. These modifications are determined in large part by the expertise of the simulation analysts. This points out a need for semi-automatic methods for suggesting design refinements, which can be tricky. For example, suppose one has built a response surface model that accurately characterizes simulation performance over a particular region of the factor space. Over time, however, the external environment changes so that the combinations of factor levels initially studied are no longer of primary interest—so some additional experiments are conducted. The question then is: when is it appropriate to use a global metamodel (with data from all experiments) instead of focusing on several local metamodels (over more restricted ranges)? This question merits further research.

6. Conclusions and Future Research

Our primary goal in writing this paper is to help change the *mindset* of simulation practitioners and researchers: we believe practitioners should view DOE as an integral part of any simulation study, while researchers should move beyond viewing the simulation setting merely as an application area for traditional DOE methods. We advocate thinking first about the three basic goals: understanding a system, finding robust solutions, or comparing systems. We also recommend that simulation analysts select a design

suitable for the context; this selection typically means that a large number of factors—and potentially complex response surfaces—need to be considered. We also provide guidance on the appropriate use of a variety of easy designs.

In this paper, we have listed many problems that require more investigation, resulting in a *research agenda for the design of simulation experiments*. For example, it is important to further investigate sequential design and analysis since most computer architectures simulate the scenarios and replicates one after the other. The search for robust instead of ‘optimal’ solutions requires further research. Further work is needed to better match types of metamodels (and appropriate designs for developing these metamodels) to the characteristics of the simulation setting. Screening designs deserve further investigation and application, particularly if they can be incorporated into other designs to reduce the large number of factors at the start of the investigation. Non-smooth metamodels are needed to represent spikes, thresholds, and chaotic behavior; appropriate designs require more research and software. Multiple outputs might need special designs and analyses for non-linear regression metamodels—used in Kriging and neural nets—and for evaluating or comparing systems. In addition, approaches that deal with constraints on factor level combinations and/or unstable system configurations are critical if we are to explore large regions of the factor space.

In addition to the research, appropriate design and analysis methods must be readily available in software. While gains have been made in recent years, as in Kriging and visualization software, there is still much room for improvement.

Acknowledgments

This work was supported in part by a grant from the Marine Corps Combat Development Command.

References

- Angün, E., D. den Hertog, G. Gürkan, J.P.C. Kleijnen. 2002. Response surface methodology revisited. *Proceedings of the Winter Simulation Conference*, forthcoming.

- Banks, J., J.S. Carson, B.L. Nelson, D.M. Nicol. 2000. *Discrete-event Simulation*, 3rd ed. Prentice-Hall, Upper Saddle River, NJ.
- Bettonvil, B., J.P.C. Kleijnen. 1997. Searching for important factors in simulation models with many factors: sequential bifurcation. *European Journal of Operational Research* **96** 180-194.
- Bettonvil, B., J.P.C. Kleijnen. 1990. Measurement scales and resolution IV designs. *American Journal of Mathematical and Management Sciences* **10** 309-322.
- Box, G.E.P., W.G. Hunter, J.S. Hunter. 1978. *Statistics for Experimenters: An Introduction to Design, Data Analysis and Model Building*. John Wiley & Sons, New York.
- Cabrera-Rios, M., C.A. Mount-Campbell, S.A. Irani. 2002. An approach to the design of a manufacturing cell under economic considerations. *International Journal of Production Economics* **78** 223-237.
- Campolongo, F., J.P.C. Kleijnen, T. Andres. 2000. Screening methods. In A. Saltelli, K. Chan, E.M. Scott, eds. *Sensitivity Analysis*. John Wiley & Sons, New York. 65-89.
- Cheng, R.C.H. 1997. Searching for important factors: sequential bifurcation under uncertainty. *Proceedings of the Winter Simulation Conference*. 275-280.
- Cheng, R.C.H. and J.P.C. Kleijnen. 1999. Improved design of simulation experiments with highly heteroskedastic responses. *Operations Research* **47** 762-777.
- Chick, S.E., K. Inoue. 2001. New procedures to select the best simulated system using common random numbers. *Management Science* **47** 1133-1149.
- Cioppa, T.M. 2002. Efficient Nearly Orthogonal and Space-filling Experimental Designs for High-dimensional Complex Models. Ph.D. Dissertation, Operations Research Department, Naval Postgraduate School, Monterey, CA.
http://library.nps.navy.mil/uhtbin/hyperion-image/02sep_Cioppa_PhD.pdf
- Cressie, N.A.C. 1993. *Statistics for Spatial Data*. Revised ed. John Wiley & Sons, New York.
- Dewar, J.A., S.C. Bankes, J.S. Hodges, T.W. Lucas, D.K. Saunders-Newton, P. Vye. 1996. Credible Uses of the Distributed Interactive Simulation (DIS) system. MR-607-A, RAND, Santa Monica, CA.
- Donohue, J. M., E.C. Houck, R.H. Myers. 1993. Simulation designs and correlation induction for reducing second-order bias in first-order response surfaces. *Operations Research* **41** 880-902.

- Fang, K.T., D.K.J. Lin, P. Winker, Y. Zhang. 2000. Uniform design: theory and application. *Technometrics* **42** 237-248.
- Fang, K.T., Y. Wang. 1994. *Number-Theoretic Methods in Statistics*. Chapman & Hall, London.
- Fu, M.C. 2002. Optimization for simulation: theory vs. practice. *INFORMS Journal on Computing* **14** 192-215.
- Gentle, J.E. 2002. *Computational Statistics*. Springer, New York.
- Giunta, A.A., L.T. Watson. 1998. A comparison of approximating modeling techniques: polynomial versus interpolating models. AIAA-98-4758.
- Goldsman, D., S-H. Kim, W. S. Marshall, B.L. Nelson. 2002. Ranking and selection for steady-state simulation: procedures and perspectives. *INFORMS Journal on Computing* **14** 2-19.
- Holcomb, D., W.M. Carlyle. 2002. Some combinatorial aspects, construction methods, and evaluation criteria for supersaturated designs. *Quality and Reliability Engineering International* **18** 299-304.
- Holcomb, D., D.C. Montgomery, W.M. Carlyle. 2003. Analysis of supersaturated designs. *Journal of Quality Technology*, forthcoming.
- Horne, G., M. Leonardi, eds. 2001. *Maneuver Warfare Science 2001*. Marine Corps Combat Development Command, Quantico, VA.
- Hsu, J.C. 1996. *Multiple Comparisons; Theory and Methods*. Chapman & Hall, London.
- Irizarry, M., J.R. Wilson, J. Trevino. 2001. A flexible simulation tool for manufacturing-cell design, II: response surface analysis and case study. *IIE Transactions* **33** 837-846.
- Jacobson, S., A. Buss, L. Schruben. 1991. Driving frequency selection for frequency domain simulation experiments. *Operations Research* **39** 917-924.
- Jin, R., W. Chen, T. Simpson. 2000. Comparative studies of metamodeling techniques under multiple modeling criteria. AIAA-2000-4801.
- Johnson, M., L. Moore, D. Ylvisaker. 1990. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference* **26** 131-148.
- Khuri, A.I. 1996. Multiresponse surface methodology. In S. Ghosh, C.R. Rao, eds. *Handbook of Statistics Volume 13*. Elsevier, Amsterdam.

- Kleijnen, J.P.C. 1998. Design for sensitivity analysis, optimization, and validation of simulation models. In J. Banks, ed. *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*. John Wiley & Sons, New York. 173-223.
- Kleijnen, J.P.C. 1995. Case study: statistical validation of simulation models. *European Journal of Operational Research* **87** 21-34.
- Kleijnen, J.P.C. 1987. *Statistical Tools for Simulation Practitioners*. Marcel Dekker, New York.
- Kleijnen, J.P.C. 1974-1975. *Statistical Techniques in Simulation, Volumes I and II*. Marcel Dekker Inc., New York. (Russian translation, Publishing House 'Statistics', Moscow, 1978.)
- Kleijnen, J.P.C., R.C.H. Cheng. 1999. Improved design of queueing simulation experiments with highly heteroscedastic responses. *Operations Research* **47** 762-777.
- Kleijnen, J.P.C., R.C.H. Cheng, V.B. Melas. 2000. Optimal design of experiments with simulation models of nearly saturated queues. *Journal of Statistical Planning and Inference* **85** 19-26.
- Kleijnen, J.P.C., A.J. Feelders, R.C.H. Cheng. 1998. Bootstrapping and validation of metamodels in simulation. *Proceedings of the Winter Simulation Conference*. 701-706.
- Kleijnen, J.P.C., E. Gaury. 2002. Short-term robustness of production management systems: a case study. *European Journal of Operational Research*, forthcoming.
- Kleijnen, J.P.C., M.T. Smits. 2002. Performance metrics in supply chain management. Working paper, Tilburg University, Tilburg, The Netherlands.
- Kleijnen, J.P.C., A.J. van den Burg, R.Th. van der Ham. 1979. Generalization of simulation results: practicality of statistical methods. *European Journal of Operational Research* **3** 50-64.
- Kleijnen, J.P.C., A. Vonk Noordegraaf, M. Nielen. 2001. Sensitivity analysis of censored output through polynomial, logistic and tobit models: theory and case study. *Proceedings of the Winter Simulation Conference*. 486-491.
- Kleijnen, J.P.C., O. Pala. 1999. Maximizing the simulation output: a competition. *Simulation* **73** 168-173.
- Kleijnen, J.P.C., R.G. Sargent. 2000. A methodology for the fitting and validation of metamodels in simulation. *European Journal of Operational Research* **120** 14-29.
- Kleijnen, J.P.C., W. van Groenendaal. 1992. *Simulation: A Statistical Perspective*. John Wiley & Sons, Chichester, England.

- Koehler, J.R., A.B. Owen. 1996. Computer experiments. In S. Ghosh, C.R. Rao, eds. *Handbook of Statistics, Volume 13*. Elsevier, Amsterdam. 261-308.
- Lauren, M.K. R. T. Stephen. 2001. *MANA Map Aware Non-Uniform Automata Version 1.0 Users Manual*. Land Operations Division, Defence Science and Technology Organisation, Australia.
- Law, A.M., W.D. Kelton. 2000. *Simulation Modeling and Analysis*. 3rd ed. McGraw-Hill, New York.
- Lin, D.K.J. 1995. Generating systematic supersaturated designs. *Technometrics* **37** 213-225.
- Lucas, T.W., S.M. Sanchez, L. Brown, W. Vinyard. 2002. Better designs for high-dimensional explorations of distillations. In G. Horne, S. Johnson eds. *Maneuver Warfare Science 2002*, USMC Project Albert, Quantico, VA.
- Lucas, T.W., S.C. Bankes, P. Vye. 1997. Improving the Analytic Contribution of Advanced Warfighting Experiments (AWEs). *RAND*, DB-207-A.
- Martinez, W.L., A.R. Martinez. 2002. *Computational Statistics Handbook with MATLAB*. Chapman & Hall/CRC, Boca Raton, FL.
- McKay, M.D., R.J. Beckman, W.J. Conover. 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21** 239-245.
- Meyer, T., S. Johnson. 2001. Visualization for data farming: a survey of methods. In G. Horne, M. Leonardi, eds. *Maneuver Warfare Science 2001*. Marine Corps Combat Development Command, Quantico, VA.
- Moeni, F., S.M. Sanchez, A.J. Vakharia. 1997. A robust design methodology for Kanban system design. *International Journal of Production Research* **35** 2821-2838.
- Montgomery, D.C. 1991. *Design and Analysis of Experiments*. John Wiley & Sons, New York.
- Morrice, D.J., I.R. Bardhan. 1995. A weighted least squares approach to computer simulation factor screening. *Operations Research* **43** 792-806.
- Morris, M.D., T.J. Mitchell. 1995. Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference* **43** 381-402.
- Myers, R.H., D.C. Montgomery. 1995. *Response Surface Methodology: Process and Product Optimization using Designed Experiments*. John Wiley & Sons, New York.

- Nelson, B.L., D. Goldsman. 2001. Comparisons with a standard in simulation experiments. *Management Science* **47** 449-463.
- Osio, I.G., C.H. Amon. 1996. An engineering design methodology with multistate Bayesian surrogates and optimal sampling. *Research in Engineering Design* **8** 189-206.
- Pukelsheim, F. 1993. *Optimal Design of Experiments*. John Wiley & Sons, New York.
- Ramberg, J.S., S.M. Sanchez, P.J. Sanchez, L.J. Hollick. 1991. Designing simulation experiments: Taguchi methods and response surface metamodels. *Proceedings of the Winter Simulation Conference*. 167-176.
- Rechtschaffner, R.L. 1967. Saturated fractions of 2^n and 3^n factorial designs. *Technometrics* **9** 569-575.
- Sacks, J., W.J. Welch, T.J. Mitchell, H.P. Wynn. 1989. Design and analysis of computer experiments (includes Comments and Rejoinder). *Statistical Science* **4** 409-435.
- Saltelli, A., S. Tarantola, and P.S. Chan 1999. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics* **41** 39-56
- Sanchez, S.M. 2000. Robust design: seeking the best of all possible worlds. *Proceedings of the Winter Simulation Conference*. 69-76.
- Sanchez, S.M., L. D. Smith, E.C. Lawrence. 2001. Tolerance design revisited: assessing the impact of correlated noise factors. Working paper, Operations Research Department, Naval Postgraduate School, Monterey, CA.
- Sanchez, S.M, T.W. Lucas. 2002. Exploring the world of agent-based simulations: simple models, complex analyses. *Proceedings of the Winter Simulation Conference*, forthcoming.
- Sanchez, P.J., A.H. Buss. 1987. A model for frequency domain experiments. *Proceedings of the Winter Simulation Conference*. 424-427.
- Schruben, L.W., V.J. Cogliano. 1987. An experimental procedure for simulation response surface model identification. *Communications of the ACM* **30** 716-730.
- Simon, H.A. 1981. *The Sciences of the Artificial*. 2nd edition. MIT Press, Cambridge, MA.
- Simpson, T.W., J. Peplinski, P.N. Koch, J.K. Allen. 1997. On the use of statistics in design and the implications for deterministic computer experiments. DETC97/DTM-3881, ASME. 1-12.

- Simpson, T.W., D.K.J. Lin, W. Chen. 2001. Sampling strategies for computer experiments: design and analysis. *International Journal of Reliability and Applications*, forthcoming.
- Smith, L.D., S.M. Sanchez. 2003. Assessment of business potential at retail sites: empirical findings from a U.S. supermarket chain. *The International Review of Retail, Distribution and Consumer Research*, forthcoming.
- Steiger, N.M., J.R. Wilson. 2001. Convergence properties of the batch means method for simulation output analysis. *INFORMS Journal on Computing* **13** 277-293.
- Sugiyama, S.O., J.W. Chow. 1997. @Risk, Riskview and BestFit. *OR/MS Today* **24**(2) 64-66.
- Taguchi, G. 1987. *System of Experimental Designs, Volumes 1 and 2*. UNIPUB/Krauss International, White Plains, NY.
- Trocine L., L.C. Malone. 2001. An overview of newer, advanced screening methods for the initial phase in an experimental design. *Proceedings of the Winter Simulation Conference*. 169-178.
- Tunali, S. and I. Batmaz. 2000. Dealing with the least squares regression assumptions in simulation metamodeling. *Computers & Industrial Engineering* **38** 307-320.
- Tufte, E.R. 1990. *Envisioning Information*. Graphics Press, Cheshire, CT.
- Van Beers, W., J.P.C. Kleijnen. 2002. Kriging for interpolation in random simulation. *Journal Operational Research Society* (accepted).
- Ye, K.Q. 1998. Orthogonal column Latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association—Theory and Methods* **93** 1430-1439.
- Vinyard, W., T.W. Lucas. 2002. Exploring combat models for non-monotonicities and remedies. *PHALANX* **35**, March.
- Vonk Noordegraaf, A., M. Nielen, J.P.C. Kleijnen. 2002. Sensitivity analysis by experimental design and metamodeling: case study on simulation in national animal disease control. *European Journal of Operational Research*, accepted.
- Wu, H-F. 2002. Spectral analysis and sonification of simulation data generated in a frequency domain experiment. M.S. Thesis, Operations Research Department, Naval Postgraduate School, Monterey, CA. http://library.nps.navy.mil/uhtbin/hyperion-image/02sep_Wu.pdf
- Zeigler B.P., K. Praehofer, T.G. Kim. 2000. *Theory of Modeling and Simulation*. 2nd ed. Academic Press.