Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, eds.

BETTER THAN A PETAFLOP: THE POWER OF EFFICIENT EXPERIMENTAL DESIGN

Susan M. Sanchez

Operations Research Department
Naval Postgraduate School
Monterey, CA 93943, U.S.A.

Hong Wan

School of Industrial Engineering
Purdue University
West Lafayette, IN 47907

ABSTRACT

Recent advances in high-performance computing have pushed computational capabilities to a petaflop (a thousand trillion operations per second) in a single computing cluster. This breakthrough has been hailed as a way to fundamentally change science and engineering by letting people perform experiments that were previously beyond reach. But for those interested in exploring the I/O behavior of their simulation model, efficient experimental design has a much higher payoff at a much lower cost. A well-designed experiment allows the analyst to examine many more factors than would otherwise be possible, while providing insights that cannot be gleaned from trial-and-error approaches or by sampling factors one at a time. We present the basic concepts of experimental design, the types of goals it can address, and why it is such an important and useful tool for simulation. Ideally, this tutorial will entice you to use experimental designs in your upcoming simulation studies.

1 INTRODUCTION

In June 2008, a new supercomputer called the ”Roadrunner” was unveiled. This bank of machines was assembled from components originally designed for the video game industry; it costs $133 milion, and is capable of doing a petaflop (a thousand trillion operations per second). The New York Times coverage included the following description: “*By running programs that find a solution in hours or even less time. compared with as long as three months on older generations of computers, petaflop machines like Roadrunner have the potential to fundamentally alter science and engineering, supercomputer experts say. Researchers can ask questions and receive answers virtually interactively and can perform experiments that would previously have been impractical*” (Markoff 2008).

Yet let’s take a closer look at the practicality of a brute-force approach to simulation experiments. Suppose a simulation has 100 factors, each factor has two levels (say, low and high) of interest, and we decide to look at each combination of these 100 factors. Even with a petaflop computer and a simulation that runs as fast as a single operation, running a single replication of this experiment would take over 40 million years!

Efficient design of experiments can break this curse of dimensionality at a tiny fraction of the cost. For example, suppose we want study 100 factors and all their two-way interactions. We can use a resolution 5 fractional factorial (described in Section 3.3). How quickly can we finish the experiment? On a desktop computer with a simulation that takes a full second to run, each replication of this experiment takes under 9.5 hours; even if the simulation takes a more reasonable one minute to run, we can finish this experiment on an 8-core desktop (under $3,000) in 2.85 days. Other designs are even more efficient, and provide more detailed insights into the simulation model’s behavior.

The field called Design of Experiments (DOE) has been around for a long time. Many of the classic experimental designs can be used in simulation studies. We discuss a few in this paper to explain the concepts and motivate the use of experimental design. However, the settings in which real-world experiments are performed can be quite different from the simulation environment, so a framework specifically geared toward simulation experiments is beneficial.

Before undertaking a simulation experiment, it is useful to think about *why* this the experiment is needed. Simulation analysts and their clients might seek to (i) *develop a basic understanding* of a particular simulation model or system, (ii)

978-1-4244-5771-7/09/$26.00 ©2009 IEEE

find robust decisions or policies, or (iii) *compare the merits* of various decisions or policies (Kleijnen et al. 2005). The goal will influence the way the study should be conducted.

We focus on setting up single-stage experiments to address the first goal, and touch briefly on the second. Although the examples in this paper are very simple simulation models, the same types of designs have been extremely useful for investigating more complex simulation models in a variety of application areas. For a detailed discussion of the philosophy and tactics of simulation experiments, a more extensive catalog of potential designs (including sequential approaches), and a comprehensive list of references, see Kleijnen et al. (2005), Kleijnen (2007), Chapter 12 of Law (2007), or Sanchez (2008).

The benefits of experimental design are tremendous. Once you realize how much insight and information can be obtained in a relatively short amount of time from a well-designed experiment, DOE should become a regular part of the way you approach your simulation projects.

2 BASIC CONCEPTS

2.1 Definitions and Notation

One of the first things an experimenter or tester must do to design a good experiment is identify the experimental factors. In DOE parlance, *factors* are the input (or independent) variables that are thought might have some impact on *responses* (i.e., experimental outputs). In general, an experiment might have many factors, each of which might assume a variety of values, called *levels* of the factor in DOE. A primary goal of many DOEs is to identify which of the factors are really important for which responses, and which are not and can thus be dropped from further consideration, greatly reducing the experimental effort and simplifying the task of interpreting the results. Also, of the important factors, we would like to identify the nature of the impact on the responses (e.g., increasing, linear, quadratic), and whether the levels of some factors influence the effects that other factors have (called factor *interactions*).

To identify good (or even appropriate) designs, it is often useful to classify the factors along several dimensions:

- *Quantitative* or *qualitative*. Quantitative factors naturally take on numerical values, while qualitative factors do not (though they might be assigned numeric coded values).
- *Discrete* or *continuous* (quantitative factors only). Discrete factors can have levels only at certain separated values; an example would be the number of x-ray machines in a hospital, which would have to be a non-negative integer, presumably with some upper bound. Continuous factors can assume any real value, perhaps within some range, such as the speed at which a vehicle is operated.
- *Binary* or not. Binary factors are naturally constrained to just two levels, like the classification of a part as either defective or non-defective. Non-binary factors could take on more than two values, but might still be tested at only two levels, typically “low” and “high,” or might be allowed to assume (many) more than two levels in the experiment.
- *Controllable* or *uncontrollable*. In a simulation experiment all factors are manipulated and controlled, but in reality factors might be controllable or not. For example, the degree or nature of enemy jamming of a communications system would be controlled in a simulation, but not in an actual fight. This can affect how the experimenter interprets the estimates of the effects of factors.

Throughout this paper, *simulation model* denotes any model that is evaluated using a computer. Simulation models come in many flavors. There are deterministic simulations (e.g., numerical solutions of differential equations, where the same set of inputs always produces the same output) and stochastic simulations (where the same set of simulation inputs may produce different output unless the random-number streams are carefully controlled). Simulations that model a process that occurs over time can also be characterized as terminating or non-terminating, depending on the stopping conditions. For ease of presentation we assume that terminating simulations are used; the simulation stops after either a pre-specified amount of simulation time has elapsed, or when a specific event or condition occurs.

Mathematically, let $X_1, \ldots, X_k$ denote the k factors in our experiment, and let Y denote a response of interest. Sometimes graphical methods are the best way to gain insight about the Y’s, but often we are interested in constructing *response surface metamodels* that approximate the relationships between the factors and the responses with statistical models (typically regression models).

First, suppose that the X_i's are all quantitative, although they can be discrete or continuous. A *main-effects model* means we assume

$$Y = \beta_0 + \sum_{i=1}^{k} \beta_i X_i + \varepsilon, \tag{1}$$

where the ε's are independent random errors with mean zero. Ordinary least-squares regression assumes that the ε's in (1) are also identically distributed, but the regression coefficients are still unbiased estimators of the β_i even if the underlying variance is not constant.

To explore any quadratic effects, we will include terms like X_1^2 as potential explanatory variables for Y. Similarly, two-way interactions are terms like X_1X_2. A *second-order model* includes quadratic effects and two-way interactions, although it is best for numerical stability to fit this after centering the quadratic and interaction terms, as in (2):

$$Y = \beta_0 + \sum_{i=1}^{k} \beta_i X_i + \sum_{i=1}^{k} \beta_{i,i}(X_i - \overline{X}_i)^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \beta_{i,j}(X_i - \overline{X}_i)(X_j - \overline{X}_j) + \varepsilon. \tag{2}$$

Some statistical packages do this centering automatically.

It is worth noting that regression can also be used when some of the X's are qualitative—in fact, the ANOVA (analysis of variance) technique commonly used for experimental designs with qualitative X's is a special case of regression.

A *design* is a matrix where every column corresponds to a factor, and the entries within the column are settings for this factor. Each row represents a particular combination of factor levels, and is called a *design point*. If the row entries correspond to the actual settings that will be used, these are called *natural levels*. Coding the levels in a standardized way is a convenient way to characterize a design. Different codes are possible, but for quantitative data the low and high levels are often coded as -1 and $+1$, respectively, for arithmetic convenience. Table 1 shows a simple design, in both natural and coded levels, that could be used for an experiment involving two factors.

Table 1: Experimental design in natural and coded levels.

	Natural Levels		Coded Levels	
Design Point	X_1	X_2	X_1	X_2
1	16	20	-1	-1
2	18	20	$+1$	-1
3	16	22	-1	0
4	18	22	$+1$	0
5	16	24	-1	$+1$
6	18	24	$+1$	$+1$

Each repetition of the whole design matrix is called a *replication* and we generally assume that the replications are independent. Let n_d be the number of design points, and n_r be the number of replications. Then the total number of experimental units is $n_{tot} = n_d n_r$.

2.2 Pitfalls to Avoid

Two common types of simulation studies are ill-designed experiments. The first can occur if several people each suggest an "interesting" combination of factor settings, so a handful of design points end up being explored where many levels change simultaneously. Consider an agent-based simulation model of the child's game, where two teams (blue and red) each try to "capture the flag" of the opposition. Suppose that only two design points are used, corresponding to different settings for the speed (X_1) and stealth (X_2) of the blue team, with the results in Figure 1. (Instead of providing numerical response values, a blue circle is used to represent a "good" average outcome for the blue team, while a red square represents a "bad" average outcome.) One person might claim these results show that high stealth is of primary importance, another that speed is the key to success, and a third that they are equally important. There is *no way* to resolve these differences of opinion without collecting more data. In statistical terms, the effects of stealth and speed are said to be *confounded*. In practice, simulation models easily have dozens or hundreds of potential factors. A handful of haphazardly chosen scenarios, or a trial-and-error approach, can use up a great deal of time without addressing the fundamental questions.

Speed	Stealth	Success?
Low	Low	No
High	High	Yes

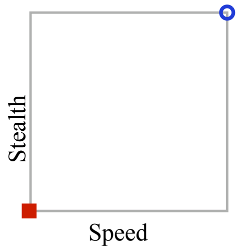

Figure 1: Confounded factor effects for capture-the-flag.

The second type of study that can be problematic occurs when people start with a "baseline" scenario and vary one factor at a time. Revisiting the capture-the-flag example, suppose the baseline corresponds to low stealth and low speed. Varying each factor, in turn, to its high level yields the results of Figure 2. It appears that *neither* factor is important, so someone using the simulation results to decide how to choose a team would not know how (or if) to proceed.

Speed	Stealth	Success?
Low	Low	No
High	Low	No
Low	High	No

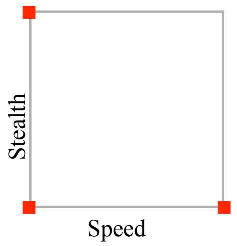

Figure 2: One-at-a-time sampling for capture-the-flag.

If all four combinations of speed and stealth (low/low, low/high, high/low, and high/high) are sampled, it is clear that success requires both high speed and high stealth. This means the that factors interact—and if there are interactions, one-at-a-time sampling will never uncover them!

The pitfalls of using a poor design seem obvious on this toy problem, but the same mistakes are made far too often in larger studies of more complex models. When only a few variations from a baseline are conducted, there may be many factors that change but a few that decision makers think are "key." If they are mistaken, changes in performance from the baseline scenario may be attributed to the wrong factors. Similarly, many analysts change one factor at a time from their baseline scenario, but fail to understand that this approach implicitly assumes that there are no interaction effects. This assumption may be unreasonable unless the region of exploration is very small.

2.3 Choosing Factors

Potential factors in simulation experiments include the *input parameters* or *distributional parameters* of a simulation model. For example, a simulation model of a repair facility might have both quantitative factors (such as the number of mechanics of different types, or the mean time for a particular task) and qualitative factors (such as priority rules).

Generating a list of the potential inputs to a simulation model is one way of coming up with an initial factor list. However, factors need not correspond directly to simulation inputs. For example, suppose two inputs are the mean times μ_1 and μ_2 required for a specific agent to process messages from class 1 and class 2, respectively, where message class 1 is considered more complex than message class 2. Varying μ_1 and μ_2 independently may either result in unrealistic situations where $\mu_1 < \mu_2$, or require the analyst to select narrow factor ranges. Instead, we could use μ_1 as one factor to represent the capabilities of the agent, and vary the ratio μ_2/μ_1 over a range of interesting values (say, 0.4 to 0.9) to represent the relative difference in message complexity.

2.4 Sample-Size Issues

In live experiments, where data are extremely expensive, the total sample size is often very small. This affects the choice of an experimental design as well as the number of replications.

In simulation experiments, where a major portion of the effort often occurs in model development, the total sampling budget may not be so constrained. This increases the set of potential designs that can be used, and it may be possible to generate a great deal of information (even hundreds of thousands of runs) in a relatively short time. We discuss this further in Section 3.

2.5 Non-terminating Simulations

Different types of simulation studies involve different types of *experimental units*. For a static Monte Carlo simulation, where no aspect of time is involved, the experimental unit is a single observation. For time-stepped or discrete-event stochastic simulation studies, it more often is a run or a batch, yielding an averaged or aggregated output value. When runs form the experimental units for non-terminating simulations, and steady-state performance measures are of interest, care must be taken to delete data during the simulation's warm-up period before performing the averaging or aggregation. Details may be found in any simulation textbook, such as Law (2007), Kelton et al. (2007), or Banks et al. (2005).

3 POTENTIAL EXPERIMENTAL DESIGNS

Many designs are available in the literature. We focus on a few basic types that we have found particularly useful for simulation experiments. Factorial or gridded designs are straightforward to construct and readily explainable—even to those without statistical backgrounds. Coarse grids (2^k factorials) are most efficient if we can assume that the simulation response is well-fit by a model with only linear main effects and interactions, while fine grids (more than two levels for factors) provide greater detail about the response and greater flexibility for constructing metamodels of the responses. When the number of factors is large, then more efficient designs are required. We have found Latin hypercubes to be good general-purpose designs for exploring complex simulation models when little is known about the response surfaces. Two-level designs called *resolution 5 fractional factorials* (R5-FFs) allow the linear main effects and interactions of many factors to be investigated simultaneously; they are potential choices either when factors have only two qualitative settings, or when practical considerations dictate that only a few levels be used for quantitative input factors. Expanding these R5-FFs to central composite designs provides some information about nonlinear behavior in simulation response surfaces.

Factorials (or gridded designs) are perhaps the easiest to discuss: they examine all possible combinations of the factor levels for each of the X_i's. A shorthand notation for the design is m^k, which means k factors are investigated, at m levels for each factor, in a total of m^k design points. We can write designs where different sets of factors are investigated at different numbers of levels as, e.g., $m_1^{k_1} \times m_2^{k_2}$, where k_1 factors are evaluated at m_1 levels each, and another k_2 factors are evaluated at m_2 levels each. These are sometimes called *crossed* designs. For example, the design in Table 1 is a $2^1 \times 3^1$ factorial experiment.

3.1 2^k Factorial Designs (Coarse Grids)

The simplest factorial design is a 2^k because it requires only two levels for each factor. These can be low and high, often denoted -1 and $+1$ (or $-$ and $+$). 2^k designs are very easy to construct. Start by calculating the number of rows $N = 2^k$. The first column alternates -1 and $+1$, the second column alternates -1 and $+1$ in groups of 2, the third column alternates in groups of 4, and so forth by powers of 2. If you are using a spreadsheet, you can easily move from a design for k factors to a design for $k+1$ factors by copying the 2^k design, pasting it below to obtain a $2^k \times k$ matrix, and then adding a column for factor $k+1$ with the first 2^k values set to -1 and the second set of 2^k values set to $+1$. Conceptually, 2^k factorial designs sample at the corners of a hypercube defined by the factors' low and high settings. Figure 3 shows examples for 2^2 and 2^3 designs. Envisioning a 2^4 or larger design is left to the hyperimaginative reader.

Factorial designs have several nice properties. They let us examine more than one factor at a time, so they can be used to identify important interaction effects. They are also *orthogonal* designs: the pairwise correlation between any two columns (factors) is equal to zero. This simplifies the analysis of the output (Y's) we get from running our experiment, because estimates of the factors' effects ($\hat{\beta}_i$'s) and their contribution to the explanatory power (R^2) of the regression metamodel will not depend on what other explanatory terms are present in the regression metamodel.

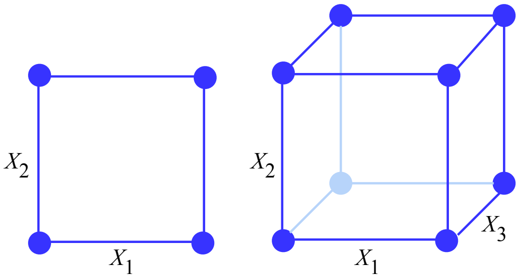

Figure 3: 2^2 and 2^3 factorial designs.

From Table 2, there are seven different terms (three main effects, two two-way interactions, and one three-way interaction) that we could consider estimating from a 2^3 factorial experiment. But since we also want to estimate the intercept (overall mean), that means there are eight things we could try to estimate from eight data points. That will not work—we will always need at least one degree of freedom (d.f.) for estimating error (and preferably, a few more).

Table 2: Terms for a 2^3 factorial design

Design	Term						
Point	1	2	3	1,2	1,3	2,3	1,2,3
1	−1	−1	−1	+1	+1	+1	−1
2	+1	−1	−1	−1	−1	+1	+1
3	−1	+1	−1	−1	+1	−1	+1
4	+1	+1	−1	+1	−1	−1	−1
5	−1	−1	+1	+1	−1	−1	+1
6	+1	−1	+1	−1	+1	−1	−1
7	−1	+1	+1	−1	−1	+1	−1
8	+1	+1	+1	+1	+1	+1	+1

A similar relationship holds as we increase the number of factors k. There will be k main effects, $\binom{k}{2} = \frac{k!}{2!(k-2)!}$ two-way interactions, $\binom{k}{3}$ three-way interactions, and so forth, up to a single k-way interaction. Adding all these up yields $2^k - 1$ terms plus the intercept. Once again, there will not be any d.f. left over for error estimation.

So, what do people do with a factorial design? One possibility is to *replicate* the design to get more d.f. for error. Estimating eight effects from eight observations (experimental units) is not possible, but estimating eight effects from 16 observations is simple. Replication also makes it easier to detect smaller effects by reducing the underlying standard errors associated with the estimates of the β's.

Another option is to *make simplifying assumptions.* The most common approach is to assume that some higher-order interactions do not exist. In the 2^3 factorial of Table 2, one d.f. would be available for estimating error if the three-way interaction could safely be ignored. We could then fit a second-order regression model to the results. Similarly, if we have data for a single replication of a 2^4 factorial design but can assume there is no four-way interaction we have one d.f. for error; if we can assume there are no three-way or four-way interactions, we have five d.f. for error estimation.

Making simplifying assumptions sounds dangerous, but it is often a good approach. Over the years, statisticians conducting field experiments have found that often, if there are interactions present, the main effects also show up unless you "just happen" to set the low and high levels so the effects cancel. There is also a rule of thumb stating that the magnitudes of two-way interactions are at most about 1/3 the size of main effects, and the magnitudes of three-way interactions are at most about 1/3 the size of the two-way interactions, etc. Whether or not this holds for experiments on simulations of complex systems is not yet certain. We may expect to find stronger interactions in a simulation of a supply chain or humanitarian assistance operations than when growing potatoes.

3.2 m^k Factorial Designs (Finer Grids)

Examining each of the factors at only two levels (the low and high values of interest) means we have no idea how the simulation behaves for factor combinations in the interior of the experimental region. Finer grids can reveal complexities in the landscape. When each factor has three levels, the convention is to use -1, 0 and 1 (or $-$, 0, and $+$) for the coded levels. Consider the capture-the-flag example once more. Figure 4 shows the (notional) results of two experiments: a 2^2 factorial (on the left) and an 11^2 factorial (on the right). For the 2^2 factorial, all that can be said is that when speed and stealth are both high, the agent is successful. Much more information is conveyed by the 11^2 factorial: here we see that if the agent can achieve a minimal level of stealth, then speed is more important. In both subgraphs the blue circles—including the upper right-hand corner—represent good results, the tan triangles in the middle represent mixed results, and the red squares on the left-hand side and bottom represent poor results.

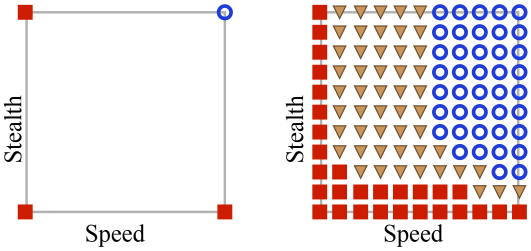

Figure 4: 2^2 and 11^2 factorial designs for capture-the-flag.

A scatterplot matrix of the design points shows projections of the full design onto each pair of factors. Consider the left-most graph in Figure 5 for a 2^4 factorial. This graph contains cells of subplots of the design points for pairs of factors at a time. For instance, the third cell over in the top row plots the (X_3, X_1) factor combinations; the third cell down in the left column is just its transpose, plotting the pairs (X_1, X_3), so carries the same information. The second graph in Figure 5 contains the scatterplot matrix for a 4^4 factorial.

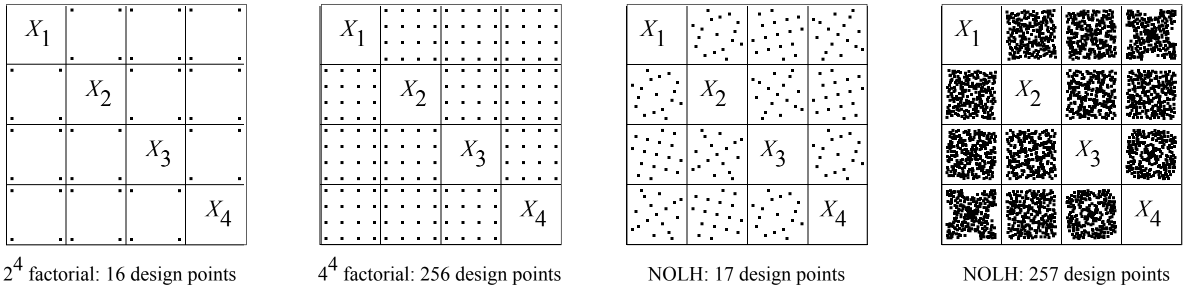

Figure 5: Scatterplot matrices for selected factorial and NOLH designs.

The larger the value of m for an m^k factorial design, the better its space-filling properties. Yet despite the greater detail provided, and the ease of interpreting the results, fine grids are not suitable for more than a handful of factors because of their massive data requirements. Even 2^k designs have this problem, as Table 3 shows.

Considering the number of high-order interactions we *could* fit but may not believe are important (relative to main effects and two-way or possibly three-way interactions), this seems like a lot of wasted effort. It means we need *smarter, more efficient* types of experimental designs if we are interested in exploring many factors.

3.3 2^{k-p} Resolution 5 Fractional Factorial Designs

Sometimes many factors take on only a few levels. In these cases, we can consider varations of gridded designs. If we are willing to assume that some high-order interactions are not important, we can cut down (perhaps dramatically) the number of runs required. This will be illustrated using a 2^k factorial, but the same ideas hold for other situations. Consider the 2^3

Table 3: Data requirements for factorial designs.

No. of factors	10^k factorial	5^k factorial	2^k factorial
1	10	5	2
2	$10^2 = 100$	$5^2 = 25$	$2^2 = 4$
3	$10^3 = 1{,}000$	$5^3 = 125$	$2^3 = 8$
5	100,000	3,125	32
10	10 billion	9,765,625	1,024
20	*don't even think of it!*	9.5×10^9	1,048,576
40		9.1×10^{21}	1.0×10^9

design in Table 2, and suppose that we are willing to assume that no interactions exist. We could call the $X_1X_2X_3$ column X_4, and investigate four factors in $2^3 = 8$ runs instead of four factors in 16 runs! This is called a 2^{4-1} fractional factorial. The potential for reducing the total number of runs increases with k.

Better yet, as long as we are assuming no interactions, we can squeeze a few more factors into the study. Take Table 2, which shows all the interaction patterns for a 2^3 factorial, and substitute in a new factor for each interaction term. The resulting design (Table 4) is called a 2^{7-4} fractional factorial, because the base design varies seven factors in only $2^{7-4} = 8$ runs instead of $2^7 = 128$ runs! X_4 uses the column that would correspond to an X_1X_2 interaction, X_5 uses the column that would correspond to an X_1X_3 interaction, and so on. The design is said to be *saturated* since we cannot squeeze in any other factors. If we ignore the last column (i.e., we do not have an X_7) then we can examine six factors in only eight runs. If we take $b = 2$ replications, we can examine seven factors in only 16 runs.

Table 4: Terms for a 2^{7-4} fractional factorial design.

Design	X_1	X_2	X_3	X_4	X_5	X_6	X_7
Point				(1,2)	(1,3)	(2,3)	(1,2,3)
1	-1	-1	-1	$+1$	$+1$	$+1$	-1
2	$+1$	-1	-1	-1	-1	$+1$	$+1$
3	-1	$+1$	-1	-1	$+1$	-1	$+1$
4	$+1$	$+1$	-1	$+1$	-1	-1	-1
5	-1	-1	$+1$	$+1$	-1	-1	$+1$
6	$+1$	-1	$+1$	-1	$+1$	-1	-1
7	-1	$+1$	$+1$	-1	-1	$+1$	-1
8	$+1$	$+1$	$+1$	$+1$	$+1$	$+1$	$+1$

Graphically, fractional factorial designs sample at a carefully-chosen fraction of the corner points on the hypercube. The left-most cube in Figure 6 shows the sampling for a 2^{3-1} factorial design, i.e., investigating three factors, each at two levels, in only $2^{3-1} = 4$ runs. There are two points on each of the left and right faces of the cube, and each of these faces has one instance of X_2 at each level and one instance of X_3 at each level, so we can isolate the effect for factor X_1. Similarly, averaging the results for the top and bottom faces allows us to estimate the effect for factor X_2, and averaging the results for the front and back faces allows us to estimate the effect for factor X_3.

Saturated or nearly-saturated fractional factorials are often called *screening designs* because they can be useful for eliminating factors that are unimportant. They very efficient (relative to full factorial designs) when there are many factors. For example, 64 runs could be used for a single replication of a design involving 63 factors, or two replications of a design involving 32 factors. Screening designs that allow only main effects to be estimated are called resolution 3 fractional factorials (R3-FFs); designs that provide valid estimates of main effects in the presence of two-way interactions (without allowing the analyst to estimate the interaction effects) are called resolution 4 fractional factorials (R4-FFs).

Saturated or nearly saturated fractional factorials are also very easy to construct. However, these designs will not do a good job of revealing the underlying structure of the response surface if there truly are strong interactions but we ignore them when setting up the experiment. A compromise is to use half-fractions or R5 fractional factorials. These allow two-way interactions to be explored but can require many fewer design points than full factorials.

It is easy to create a 2^{k-1} factorial (called a *half fraction*) by setting up the first 2^{k-1} columns as if we just had $k-1$ factors, and then constructing a column for the last factor by taking the interaction (product) of the first $k-1$ columns. Except for the special cases when $k \leq 4$, we will also be able to estimate two-way interactions with the 2^{k-1} designs. Unfortunately, a half-fraction is still inefficient if k is large. Until recently it was difficult to find a very efficient R5-FF for more than about a dozen factors. The largest R5-FF in Montgomery (2005) is a 2^{10-3}; the largest in Box, Hunter, and Hunter (2005) and NIST/Sematech (2006) is a 2^{11-4}. Sanchez and Sanchez (2005) recently developed a method, based on discrete-valued Walsh functions, for rapidly constructing very large R5-FFs—a short program generates designs up to a $2^{120-105}$ in under a minute. These allow all main effects and two-way interactions to be fit, and may be more useful for simulation analysts than saturated or nearly-saturated designs. The gains in efficiency (as compared to full factorials) are dramatic enough to be worth mentioning again: running a 2^{100} full factorial would require over 40 million years on the world's fastest supercomputer in 2009, while a R5-FF requires only $2^{100-85} = 32768$ design points.

3.4 Central Composite Designs

Because 2^k factorials or fractional factorials sample each factor at only two levels, they are very efficient at identifying slopes for main effects or two-way interactions. Unfortunately, sampling at only two levels means the analyst has no idea about what happens to the simulation's response in the middle of the factor ranges. Going to a 3^k factorial would let us estimate quadratic effects, but it takes quite a bit more data—especially if k is large!

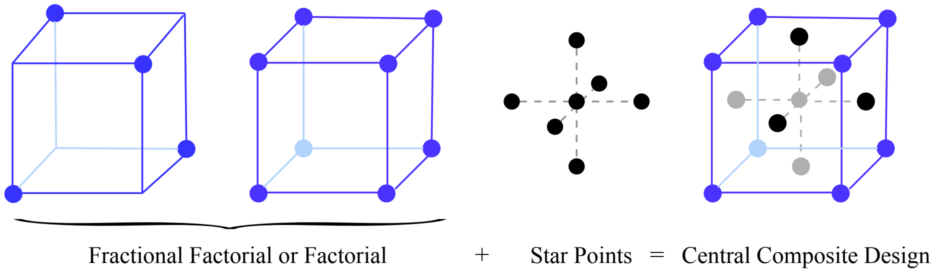

Figure 6: Construction of central composite designs.

Another classic design that lets the analyst estimate all full second-order models (i.e., main effects, two-way interactions, and quadratic effects) is called a *central composite design* (CCD). Start with a 2^k factorial or R5 2^{k-p} fractional factorial design. Then add a center point and two "star points" for each of the factors. In the coded designs, if -1 and $+1$ are the low and high levels, respectively, then the center point occurs at $(0,0,...,0)$, the first pair of star points are $(-c,0,...,0)$ and $(c,0,...,0)$; the second pair of star points are $(0,-c,0,...,0)$ and $(0,+c,0,...,0)$, and so on. A graphical depiction of a CCD for three factors appears in Figure 6. If $c = 1$ the star points will be on the face of the cube, but other values of c are possible.

Although the CCD adds more star points when there are more factors, using a fractional factorial as the basic design means the CCD has dramatically fewer design points than a 3^k factorial design for the same number of factors. For example, using the efficient R5-FFs of Sanchez and Sanchez (2005) as the base designs, a CCD for 10 factors requires 152 design points, while a 3^{10} factorial requires over 59000 design points. The additional requirements grow only linearly with k. Once again, it is clear that a brute force (full factorial) approach is impossible when k is large, but efficient experimental designs allow the analyst to conduct an experiment.

3.5 Nearly Orthogonal Latin Hypercube Designs

Latin hypercube (LH) designs provides a flexible way of constructing efficient designs for quantitative factors. They have some of the space-filling properties of factorial designs with fine grids, but require orders of magnitude less sampling. Once again, let k denote the number of factors, and let $m \geq k$ denote the number of design points. The factor levels can be coded as m equally-spaced values $\{-1, -\frac{m-2}{m-1}, -\frac{m-3}{m-1}, \ldots, \frac{m-2}{m-1}, 1\}$. A *random LH design* means that each column of the design matrix is a random permutation of these m values, and can be constructed for any number of factors k provided that $m \geq k$, but collinearity problems often arise unless $m >> k$.

Figure 7 lists a random LH with $k = 2$ and $m = 11$, and provides a picture of results that might arise by using this experimental design for our capture-the-flag simulation. Compare this design to those of Figure 4. Unlike the 2^2 factorial design, the LH design provides some information about what happens in the center of the experimental region, but requires far less effort than the 11^2 factorial design.

Speed	Stealth	Speed	Stealth
1	11	10	1
3	5	4	2
7	7	11	8
2	3	8	9
5	10	9	6
6	4		

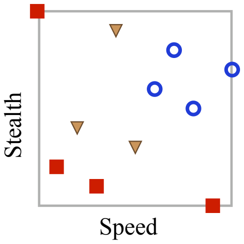

Figure 7: Random Latin hypercube for capture-the-flag.

Cioppa and Lucas (2007) construct *nearly orthogonal Latin hypercube* (NOLH) designs that have good space-filling and orthogonality properties for small or moderate k ($k \leq 29$). These designs are not square, but the number of design points are radically fewer than the numbers for the gridded designs in Table 3. For example, 20 factors can be explored in an NOLH with only 129 design points, as compared to over one million design points needed for a 2^{20} factorial.

Scatterplot matrices of four different designs are shown in Figure 5. These are a 2^4 factorial design, a 4^4 factorial design, an NOLH design with 17 design points, and an NOLH design with 257 design points. The two-dimensional space-filling behavior of the NOLH compares favorably with that of the 4^4 factorial for roughly 1/15 the computational effort, so experimenters concerned about the level of computational effort might prefer the latter. Alternatively, experimenters considering the use of the 4^4 factorial (and thus willing to run 256 design points) might prefer the NOLH with 257 design points (just one more)—and gain the ability to examine a much denser set of factor-level combinations, as well as explore up to 25 additional factors using the same design! The benefits of LH sampling are greatest for large k. Assuming that a single design point takes one second to run, each replication of a 29-factor experiment would take under five minutes using an NOLH design, but over 17 years using a 2^{29} factorial design.

Replicating the design allows us to determine whether or not a constant error variance is a reasonable characterization of the simulation's performance, and is highly recommended. If we have the time and budget for even more sampling, then two or more different Latin hypercubes or NOLHs can be *stacked* to obtain a larger design with better space-filling properties. Stacking two designs means running both sets of design points; one way to obtain two different designs from the same NOLH matrix is to reassign the factors to different columns of the experimental design matrix. More recently, Hernandez et al. (2008) use a mixed integer programming approach to generate sets of Latin hypercubes that are saturated or nearly-saturated. These extend Latin hypercube designs for simulation studies with larger numbers of factors.

3.6 Robust Design Methods

A distinction can be made between decision factors that can be controlled in the real world, and noise factors that cannot be controlled during actual operations. For example, in a simulation of search-and-rescue operations after a natural disaster, the decision factors might include the communication systems, available equipment, or number of people on the rescue team. Noise factors might include weather conditions, the number and location of those in need of rescue, and the skill levels of the emergency medical technicians. An alternative to an exploratory analysis that seeks to understand how these noise factors affect the responses is a *robust design* approach, where the goal of the experiment(s) is to identify design points that yield good performance across the range of noise factor settings—in other words, to identify *robust* systems, rather than systems that are effective only against specific threat and environmental conditions—particularly if these correspond to the most favorable settings for threat and environmental factors. The robust design philosophy was pioneered by Taguchi (1987) for manufactured-product design, where it has been successfully used to achieve high-quality products while keeping costs in line; it also facilitates the evaluation of trade-offs between quality and cost. An important consideration for the simulation community is that the robust design philosophy explicitly requires analysts to consider variances, as well as means, in assessing system performance.

The classification of factors as either decision or noise factors may affect the choice of the design. Generally, we are interested in fitting metamodels that explain the relationship between the decision factors (and their interactions, etc.) and the response. Interactions among noise factors may affect the variability of the response but are not of direct interest, while (noise factor)×(decision factor) interactions show up as unequal response variances across different decision-factor combinations.

Applying robust design principles to simulation experiments is discussed in Sanchez (2000). A more detailed discussion and examples appear in Kleijnen et al. (2005), where *identifying robust systems and processes* is considered one of three primary goals of simulation experiments.

3.7 Sequential Screening Methods

When the number of factors is very large, then sequential screening approaches may be of interest. These typically make stronger assumptions about the nature of the response surface, but are useful for quickly eliminating unimportant factors so that future experiments can focus on those that seem important. Sequential screening procedures can be more efficient than single-stage screening procedures. Two procedures of particular interest are *controlled sequential bifurcation* (CSB) procedure (Wan et al. 2006) and a variant called CSB-X (Wan et al. 2008). These procedures have the important property of providing guaranteed limits on the probabilities of observing false positives and false negatives when screening for important factors. Sequential approaches we find particularly useful for simulation experiments are *fractional factorial controlled sequential bifurcation* (F-CSB) and a variant called FFCSB-X (Sanchez, Wan, and Lucas 2009), and the *hybrid method* (Wan et al. 2009). Although these methods are heuristic, they nonetheless have been shown to have very good properties in terms of both efficiency and effectiveness. Unlike CSB and CSB-X, these latter procedures do not require *a priori* knowledge of the direction of factor effects, which makes them suitable for screening factors in simulation models of complex systems where little subject-matter expertise exists. Screening experiments are often followed up with more detailed experiments involving those factors identified as important.

3.8 Design-of-Experiment Based Simulation Optimization

Response Surface Methodology (RSM) was introduced in the early 50's by Box and Wilson (1951). It has been extensively used in industry to select the optimal operating conditions or product designs. The fundamental RSM framework can be briey described as follows for a minimization problem (Myers and Montgomery 2002):

- Stage I
    1. Plan and run a factorial (or fractional factorial) design plus the center points in the region of interest.
    2. Test for curvature (i.e., a quadratic effect). If the curvature test is not statistically significant, go to Step 3. Otherwise go to Stage II.
    3. Fit a linear regression model to the data. Determine the path of steepest descent. Move along the path of steepest descent until the response no longer improves. Go back to Step 1.
- Stage II
    1. Run a central composite design and fit a quadratic model.
    2. Based on the quadratic model, find the optimal solution.

In summary, RSM sequentially explores local regions and performs line searches along the steepest decent to approach the optimal region. Stage II (the quadratic model) is only performed once, therefore the obtained "optimal" solution is usually not the true optimal solution of the underlying system—and may not even be close. One of the biggest advantages of RSM is its generality. An arsenal of well-studied statistical tools such as regression analysis, design of experiments, and ANOVA can be incorporated in its framework, and it is widely used despite its drawbacks.

Since simulation models representing real world systems can be very complex, the local simplified metamodel approach is appealing. Early applications of RSM in simulation were reported in Biles (1975) and Kleijnen (1975) . However, two issues need to be solved. Firstly, RSM is not automated. Human intervention is required to determine the local region and appropriate design for each iteration. Secondly, RSM is heuristic, and the quality of the solution cannot be quantified. To mitigate these problems, Chang et al. (2007, 2009) propose the Stochastic Trust Region Response Surface Method (STRONG) for simulation optimization. It combines the RSM framework with the trust region method (developed for deterministic optimization). Like RSM, the method relies on a series of experimental designs for repetitive curve fitting and subsequent optimization. The difference lies on that at each iteration, the local optimization is restricted within a trust region to guarantee the reliability of the solution. If the metamodel does not fit the response well or the new solution fails to give sufficient improvement, the trust region will shrink, and vice versa. This approach eliminates the requirement of human intervention and leads to competitive asymptotic convergence property of STRONG. More importantly, the framework allows the incorporation of various experimental designs to improve the efficiency of optimization. Numerical evaluations show that this can significantly improve the efficiency of simulation optimization (Chang et al. 2007, 2009).

4 DESIGN COMPARISONS

In Figure 8 (from Sanchez 2008) we provide some guidance about experimental designs for simulation experiments. This list is not intended to be exhaustive, but we hope that it will help experimenters identify some suitable designs for particular contexts. We focus on designs discussed in Section 3 that we believe are relatively easy to generate, implement, and analyze, as well as to interpret and explain the results. However, if circumstances dictate that none of the designs are suitable, alternatives are often available from a statistician or a detailed look at some of the experimental design references. A version of this chart is maintained at the SEED Center web pages (<http://harvest.nps.edu>), and updated as new designs become available to fill some of the gaps. All acronyms are defined on the web site.

Selecting a design is an art, as well as a science. Clearly, the number of factors and the mix of different factor types (binary, qualitative or discrete with a limited number of levels, discrete with many levels, or continuous) play important roles. But these are rarely cast in stone—particularly during exploratory analysis. The experimenter has control over how factors are grouped, how levels are determined, etc. Even if these are specified, different experimenters may prefer different designs.

5 GAINING INSIGHT

We believe the following are three basic goals of simulation experiments: (i) *developing a basic understanding* of a particular simulation model or system; (ii) *finding robust* decisions or policies; and (iii) *comparing the merits* of various decisions or policies (Sanchez and Lucas 2002; Kleijnen et al. 2005). Experimental design approaches, coupled with analytic and graphical methods such as response-surface methodology and data-mining techniques, can be useful for all these goals. They help the experimenter develop a better understanding of the system. This process typically follows an iterative cycle, where insights gained from simulation experiments can be used in many ways. Results can be used to evaluate or improve the simulation model. By identifying important factors, interactions, and nonlinear effects, the experimenter can improve their understanding, find robust solutions, or raise questions to be explored in subsequent experiments. Thresholds, plateaus, or other interesting features of the response surfaces might provide guidance about situations that are particularly good (or particularly bad).

6 FINDING OUT MORE

For more on the philosophy and tactics of designing simulation experiments, examples of graphical methods that facilitate gaining insight into the simulation model's performance, and an extensive literature survey, we refer the reader to Kleijnen et al. (2005).

Books that discuss experimental designs for simulation include Santner, Williams, and Notz (2003), Law (2007), and Kleijnen (2007). Note that their goals for those performing simulation experiments may differ from those in this paper. For experiments where it is very time-consuming to run a single replication, there are other single-stage designs (often used for physical experiments) that require fewer runs than fractional factorial designs. Some of these designs appear in the above references; others can be found in experimental design texts such as Box, Hunter and Hunter (2005), Montgomery (2006), or Ryan (2007).

A more detailed discussion of how simulation experiments might be used to assist with planning live tests (physical experiments) appears in Sanchez (2008), which is the source of Figure 8 . This also contains a flowchart of the initial design process, and specific examples of different types of designs that could be used for an experiment involving 14 factors, as a way of illustrating the tradeoffs made in the "art" of choosing an appropriate experimental design.

Finally, the benefits of efficient experimental design are often more tangible if you see how they are used in practice. Designs like the ones described in this paper have assisted the U.S. military and several allied countries in a series of international data farming workshops (Horne and Meyer 2004; SEED Center for Data Farming 2008). Interdisciplinary teams of officers and analysts develop and explore agent-based simulation models to address questions of current interest to the U.S. military and allies, such as network-centric operations, effective use of unmanned vehicles, peace support operations, and more. Sanchez and Lucas (2002) provide an overview of issues in modeling and analysis aspects of agent-based simulation. A humanitarian assistance scenario is discussed in Kleijnen et al. (2005). Lucas et al. (2007) describe several defense

	2^k factorials	m^k factorials, 3 <= m <= 5	m^k factorials, 6 <= m <= 10	R3FF, orthogonal arrays	Foldover designs	R4FF, Plackett-Burman designs	R5FF	Central composite with full factorial	Central composite with R5FF	random LH with $n >> k$	smallest possible NOLH (i.e., very few extra columns)	larger NOLH	crossed NOLHs	FFCSB (main effects)	FFCSB-X or Hybrid method
FACTOR CHARACTERISTICS															
Total number of factors: 2-6	B*●	L*●	●	●	●	●	●	●	●	●	●	C*■	◍[3]	◍	◍
Total number of factors: 7-10	●	○[1]		●	●	●	B*●	●	●	●	●	C*■	◍[3]	◍	◍
Total number of factors: 11-29				●	●	●	B*●		●	●	●	C*■	◍[3]	◍	◍
Total number of factors: 30-99				●	B*●	●	○[2]		○[2]	●[4]			●	●[1]	●
Total number of factors: 100-300				●	B*●	●	○[2]		○[2]	●[4]			○[3]	●[1]	●
Total number of factors: 300-1000				●	○									●[1]	B*C* ●
Total number of factors: 1000-2000				●	○									●[1]	B*C* ●
Binary factors	B*●			●	●	●	●							●	
Qualitative factors with 3 or more levels		L*●	●												
Discrete or continuous factors treated as binary	◍[1]			◍[1]	◍[1]	◍[1]	◍[1]							●	
Discrete factors, 3-5 levels of interest	◍[1]	●						●	L*●						
Continuous factors, or discrete with many levels		●	●					●	●	●	●	●	●		●[2]
Decision factors (controllable in real world)	●	●					●				●	●			
Noise factors (uncontrollable in real world)				●	◍[4]	◍[4]	◍[4]			◍[5]	●				
RESPONSE CHARACTERISTICS															
Main effects only (initial screening)	◍[2]			●[1]	■	■	■		■		●	◍[6]		●[1]	■
Main effects (valid w/ 2-way interactions exist)	◍[2]				●			▨[2]	■						●
Main effects and all 2-way interactions	◍[2]	■	■				●	■	●				●		
Main effects and many interactions	●	◍[2]	◍[2]					●		●[5]	●[5]	●[5]	●		
Quadratic effects		◍[2]	◍[2]					●	●	●	●	●	●		
Thresholds / non-smooth effects		◍[2]	◍[2]							●	●	●	●		
Flexible modeling - not all pre-specified		◍[2]	◍[2]							●	●	●	●		
OTHER CONSIDERATIONS															
Batch mode unavailable - all runs through GUI		◍[2]		●[3]					●[3]		●[3]				

■ Provides additional modeling flexibility or allows some assumptions to be assessed
B* Good design choice for binary factors
L* Good design choice for factors with a limited number of qualitative or discrete levels
C* Good design choice for continuous factors, discrete factors with many levels
● Works well
●[1] Assumes that interactions are negligible or that they'll show up with the main effects - must follow up with confirmation runs
●[2] For FFCSB-X, "many" means 2 or 3 levels
●[3] Smaller designs are the only ones feasible until this gets "fixed" - work with the developer
●[4] Design's correlation structure must be checked - stacking many designs may be an alternative
●[5] Degrees of freedom limit the number of terms that can be estimated simultaneously, so not all main effects and two-way interactions can be estimated simultaneously.
○ Consider these designs if additional computing resources are available
○[1] These require many more runs than other designs unless *k* is small. Consider NOLH designs.
○[2] Start with 2 replications and see if you can eliminate any factors - each time you do, you effectively double the number of replications for factors that remain.
○[3] Same as above, but to avoid overly-large designs you may want to consider saturated or nearly-saturated NOLH
▨ Potential designs that provide additional modeling flexibility or allow some assumptions to be assessed, but typically require many more design points
◍ Potential designs, but better designs exist for this purpose
◍[1] Unless used for initial screening, it may be a good idea to explore at least 3 levels
◍[2] These require many more runs than other designs unless *k* is small. Consider R5FF (for binary) or NOLH designs
◍[3] Easier to use a larger NOLH (if all factors are quantitative) or else cross a full factorial for factors with just a few levels with an NOLH
◍[4] Since you do not need to estimate interactions among noise factors, use a screening design like R3FF or a small NOLH
◍[5] In the spirit of keeping noise factor designs small, you might prefer an NOLH
◍[6] If you're interested in screening and want to keep the number of runs down, go for one of the smaller LH designs

Figure 8: Design comparison chart.

and homeland security applications: critical infrastructure protection, non-lethal capabilities in a maritime environment, and emergency first response to a crisis event. The website of the SEED (Simulation Experiments & Efficient Design) Center for Data Farming (at `<harvest.nps.edu>`) also has links to many papers, both methodological and application-oriented, as well as spreadsheets and software for NOLH and R5-FF designs; this website is updated on a fairly regular basis.

7 CONCLUSIONS

The process of building, verifying, and validating a simulation model can be arduous—but once complete, then it is time to have the model work for you. One extremely effective way of accomplishing this is to use experimental designs to help explore your simulation model. This tutorial has touched on a few designs that we have found particularly useful, but other design and analysis techniques exist. Our intent was to open your eyes to the benefits of DOE, and convince you to make your next simulation study a simulation *experiment*. As we have shown, if you are interested in exploring the behavior of a simulation model with more than a handful of input factors, efficient experimental designs are readily available—and much more powerful—than a petaflop supercomputer.

ACKNOWLEDGMENTS

This work was supported in part by the U.S. Army Training and Doctrine Command Analysis Center Monterey (TRAC-MTRY), the Department of Defense's Modeling & Simulation Coordination Office (M&SCO), and the Netcentric Systems Test Science & Technology focus area (NST S&T). Portions of this tutorial appeared earlier in Sanchez (2008a, 2008b). Thanks to David Kelton and Paul Sanchez for helpful comments.

REFERENCES

Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2005. *Discrete-event simulation*, 4th ed. Upper Saddle River, New Jersey: Prentice-Hall.

Biles, W. E. 1975. A response surface method for experimental optimization of multi-response process. *Industrial and Engineering Chemistry Process Design and Development*, 14(2):152158.

Box, G. E. P., W. G. Hunter, and J. S. Hunter. 2005. *Statistics for experimenters: An introduction to design, data analysis and model building*. 2nd ed. New York: Wiley.

Box, G., and K. Wilson. 1951. On the experimental attainment of optimum conditions. *Journal of Royal Statistical Society*, 13:138.

Chang, K-H., J. L. Hong, and H. Wan. 2009. Stochastic trust region method (STRONG)a new response-surface-based algorithm in simulation optimization. School of Industrial Engineering, Purdue Unviersity. Working paper.

Chang, K-H., J. L. Hong, and H. Wan. 2007. Stochastic trust region gradient-free method (STRONG)-a new response-surface-based algorithm in simulation optimization. In *Proceedings of the 2007 Winter Simulation Conference*, ed. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 346–354. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Cioppa, T. M., and T. W. Lucas. 2007. Efficient nearly orthogonal and space-filling Latin hypercubes. *Technometrics* 49(1): 45–55.

Fu, M. 2002. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14:192227.

Hernandez, A. S., W. M. Carlyle, and T. W. Lucas. 2008. Using integer programming to generate flexible nearly orthogonal Latin hypercubes. Working Paper, Operations Research Department, Naval Postgraduate School, Monterey, CA.

Horne, G. E., and T. E. Meyer. 2004. Data farming: Discovering surprise. In *Proceedings of the 2004 Winter Simulation Conference*, eds. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 171–180. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Kelton, W. D., R. P. Sadowski, and D. P. Sturrock. 2007. *Simulation with Arena*. 4th ed. New York: McGraw-Hill.

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

Kleijnen, J. P. C., S. M. Sanchez, T. W. Lucas, and T. M. Cioppa. 2005. A user's guide to the brave new world of simulation experiments. *INFORMS Journal on Computing* 17 (3): 263–289 (with online companion).

Kleijnen, J. P. C. 1975. *Statistical Techniques in Simulation, part II.* New York: Dekker.

Law, A. M. 2007. *Simulation modeling and analysis*. 4th ed. New York: McGraw-Hill.

Lucas, T. W., S. M. Sanchez, F. Martinez, L. R. Sickinger, and J. W. Roginski. 2007. Defense and homeland security applications of multi-agent simulations. In *Proceedings of the 2007 Winter Simulation Conference*, ed. S. G. Henderson,

B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 138–149. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Markoff, J. 2008. Military supercomputer sets record. *New York Times*, June 9, 2008. Available via <http://www.nytimes.com/2008/06/09/technology/09petaflops.html> [accessed July 12, 2008].

Montgomery, D. C. 2005. *Design and analysis of experiments*. 6th ed. New York: Wiley.

Myers, R. and D. Montgomery. 2002. *Response Surface Methodology*. New York: Wiley.

NIST/Sematech. 2006. *e-Handbook of statistical methods*. Available via <http://www.itl.nist.gov/div898/handbook/> [accessed July 12, 2008].

Ryan, T. P. 2007. *Modern experimental design.* Hoboken, New Jersey: Wiley.

Sanchez, S. M. 2000. Robust design: Seeking the best of all possible worlds. In *Proceedings of the 2000 Winter Simulation Conference*, eds. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 69–76. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Sanchez, S. M. 2008a. Better than a petaflop: The power of efficient experimental design. In *Proceedings of the 2008 Winter Simulation Conference*, ed. S. J. Mason, R. R. Hill, L. Moench, O. Rose, 73–84. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Sanchez, S. M. 2008b. Simulation experiments and efficient designs in support of testing in a joint environment. Appendix to *Joint Mission Effectiveness Analysis Handbook (draft version 2.1)*, 42 pages. Suffolk, Virginia: Joint Test & Evaluation Methodology (JTEM) Joint Test & Evaluation (JT&E).

Sanchez, S. M., and T. W. Lucas. 2002. Exploring the world of agent-based simulations: Simple models, complex analyses. In *Proceedings of the 2002 Winter Simulation Conference*, eds. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. Charnes, 116–126. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Sanchez, S. M., and P. J. Sanchez. 2005. Very large fractional factorials and central composite designs. *ACM Transactions on Modeling and Computer Simulation* 15(4): 362–377.

Sanchez, S. M., H. Wan, and T. W. Lucas. 2009. Two-phase screening procedures for simulation experiments. *ACM Transactions on Modeling and Computer Simulation* 19(2): Article 7. 1-24.

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

SEED Center for Data Farming. 2008. Simulation experiments & efficient designs [online]. Available via <http://harvest.nps.edu> [accessed July 12, 2008].

Taguchi, G. 1987. *System of experimental design, vols. 1 and 2*. White Plains, New York: UNIPUB/Krauss International.

Wan, H., B. E. Ankenman, and B. L. Nelson. 2006. Controlled sequential bifurcation: A new factor-screening method for discrete-event simulation. *Operations Research* 54(4): 743–755.

Wan, H., B. E. Ankenman, and B. L. Nelson. 2008. Improving the efficiency and efficacy of CSB for simulation factor screening. *INFORMS Journal on Computing*, forthcoming.

Wan, H., H. Shen, and S. M. Sanchez. 2008. A hybrid method for simulation factor screening. Working paper, School of Industrial Engineering, Purdue University, West Lafayette, Indiana.

AUTHOR BIOGRAPHY

SUSAN M. SANCHEZ is a Professor in Operations Research at the Naval Postgraduate School, and Co-Director of the Simulation Experiments & Efficient Design (SEED) Center for Data Farming. She also holds a joint appointment in the Graduate School of Business & Public Policy. She has a B.S. in Industrial & Operations Engineering from the University of Michigan, and a Ph.D. in Operations Research from Cornell. She has been active in various service capacities within the simulation community over many years, and is currently on the WSC Board of Directors. Her web page is <http://faculty.nps.edu/smsanche/> and her email is <ssanchez@nps.edu>.

HONG WAN is an Assistant Professor in the School of Industrial Engineering at Purdue University. She is currently visiting the department of Operations Research at Naval Postgraduate School. Her research interests include design and analysis of simulation experiments, simulation optimization, applied statistics, quality management, and healthcare engineering. She has taught a variety of courses and is a member of INFORMS and ASA. Her email address is <hwan@purdue.edu> and her web page is <http://web.ics.purdue.edu/hwan>.