Proceedings of the 1989 Winter Simulation Conference
E.A. MacNair, K.J. Musselman, P. Heidelberger (eds.)

Graphical methods for the design and analysis of simulation experiments

Russell R. Barton and Lee W. Schruben
School of Operations Research and Industrial Engineering
Cornell University
Ithaca, NY 14853

ABSTRACT

In this paper we develop some intuitive graphical methods for designing, running, and analyzing simulation experiments. We first concentrate on the analysis of output plots from a single run; the concept of a standardized simulation output plot is presented and its use illustrated. The remainder of the paper involves the design of experiments that may involve several (sequential) simulation runs. Experimental design discussions in simulation often focus on special topics unique to the field. Control over random variables permits design techniques such as antithetic variates and common random number streams. Yet many of the general topics of experiment design including confounding, fractional designs, and sample size determination are important in a simulation setting. The second part of this tutorial will focus on these general topics and present some graphical tools for generating experiment designs: causal diagrams and multidimensional point plots.

1. BACKGROUND AND ADVICE

When running simulation experiments several decisions must be made.

1. What values for controllable input variables should be selected for each of the runs?

This is discussed in detail in the second half of this paper.

2. In what order should the runs be made?

Simulation experiments are typically run sequentially, one run at a time. The information from past runs is available to help in making decisions concerning future runs. This is different from, say, agricultural experiments, where the inability to compress time makes sequential experiments impractical. However, care must be taken in sequential experiments not to introduce undesirable dependencies among the different runs. We give an example of this when we discuss initialization bias. The running of simulation experiments sequentially will most likely change as multiple processor computers become more widely available, so simultaneous experiments are also discussed in this paper.

3. For a particular run:
   What initial values should be chosen for the simulation input variables?
   How long should the simulation be run?
   What output should be collected during the run?

Finally,
   What should be done with the output?

The first part of this paper addresses the third set of questions concerned with a single run. The second part of the paper is concerned with the first two questions. We discuss what to do with a single run before considering multiple runs.

However, before we address any of the above questions, our single most important advice on simulation experimentation is given: budget considerable time and money for running the model and studying the output. This, of course, includes experiments for the purposes of validating the simulation. Trying to control project costs by skimping on the experiments is shortsighted; the major payoff (although not the only payoff) from a simulation study comes during experimentation. The costs of running a simulation and studying the output are typically insignificant compared to the costs of purchasing simulation modeling software, learning the system and the language, and coding and debugging the model.

Sometimes many runs of the simulation are made but only average values of output measurements are collected; this is at best wasteful and at worst misleading. Unfortunately most simulation languages encourage this practice with built-in summary statistic computation. Individual observations should be collected and plotted until the experimenter has a feel for the dynamics of the simulation model.

It is not unusual to find practitioners devoting considerable resources to building a simulation model and relatively little time seriously studying the output. Two problems inherent in the field of simulation probably contribute to this misappropriation of simulation study effort. First, properly designed experiments and appropriate output analysis appear to be more complicated than building a simulation model. Articles on simulation methodology are often irritatingly incomprehensible to the average practitioner while simulation language documentation is, for the most part, easy to understand. The second problem is that, unlike model building and language support, support for experimental analysis from competent consultants is not widely available.

We hope to address the first problem in this paper and help make simulation experimental design and analysis more intuitive. As for the second problem, we can only encourage simulation practitioners to seek advice in the analysis of their models. This would help create a market for simulation software vendors and consultants to offer the user some real support in this area. In addition, we continue to admonish college professors to include significant simulation methodology in their courses and not merely teach simulation model building while giving their students no clue as to what they should do with the models they create. Assuming that the reader has taken this advice to heart we move on to the main body of our paper.

PART I: GRAPHICAL METHODS FOR OUTPUT ANALYSIS

2. CONVENTIONAL OUTPUT PLOTS

We begin by considering the third set of decisions presented at the beginning of this article. We will assume that the decision of what data to collect during the run has been made. Suppose that during a simulation run we observe values of a variable which we denote as,

$$Y_1, Y_2, \ldots, Y_n \;.$$

A typical situation may have Y_i represent the processing time for the ith part in a factory. Alternatively, we might continuously observe an output process, Y(t), t = 1 to n. An example: Y(t) might be the length of a waiting line for a service system at time t. We will discuss only discrete output series here but the material extends in an obvious way to continuous output (see Nozari, 1986 for details on dealing with continuous output series).

A simple plot of the data in the sequence that it was generated can give a good feel for the dynamics of the system. In particular, the effects of run duration and truncation point selection on the model output might be detected. The value of such plots (after smoothing) is illustrated quite well in Welch (1983). The reader is strongly urged to consult this reference.

Smoothing the output plot tends to reduce the rapid fluctuations of a noisy or "jittery" output series. This permits one to better see the dynamics of the system. Methods for smoothing data include taking a moving average (averaging equal-sized overlapping groups of observations) of the output. One can also average observations across several independently seeded replications of the simulation. Better still, one can average antithetic pairs of replications made at a particular setting of the input factors. Consult any good simulation text for a discussion of antithetic replication. When several independent antithetic pairs of runs are made the experimenter might want to give more weight to the average of pairs of observations that are close together then to pairs of runs where the two output measurements are far apart. When the two observations from an antithetic pair of replications are close together, the output from the two runs are likely to be near the center of their probability distribution.

Batching the data (taking averages of non-overlapping, adjacent, equal-sized groups of observations) can help smooth the output. Batching the output (a technique called batched means) can also reduce the volume of data to be plotted without losing too much valuable information. By starting with a raw output plot on the computer screen and sequentially doubling the batch size, the experimenter can observe the output plot as the plotted points tend to behave more and more like familiar independent, normally distributed data. Computationally, this simply involves recursively averaging adjacent pairs of observations. Ultimately the single value for sample average is the result of this recursion. If the total number of observations is not a power of 2, simply discard observations from the beginning of the run. Sequential batching is much more informative than just seeing the final average.

A plot of the cumulative average of the output computed during a run is sometimes recommended. Define the cumulative average of the first k observations as,

$$\bar{Y}_k = \frac{1}{k}\sum_{i=1}^{k} Y_i \quad .$$

Plotting cumulative averages is one of the more common types of output plots. However, this plot can be misleading as it will necessarily become smoother during the run, since the variance of the cumulative mean decreases with run duration.

Histograms of the output can also be informative but also might be misleading. Histograms of observations from a single run do not account for the serial dependencies typical in simulation output. Histograms of raw data across different replications of a simulation do not account for the initialization bias in each run. Initialization bias is discussed later in this paper.

3. STANDARDIZED OUTPUT PLOTS

We now discuss a different kind of output plot, the standardized output series. These plots may seem a bit strange at first but they are nevertheless very simple and powerful.

Standardizing simulation output exploits the central feature of statistical analysis: the standardization (or "normalization") of information. With standardized statistics, standard analysis can be applied to a great range of problems. The same tables and tests can be used in many fields. Statisticians can be "experts" in agriculture, sociology, economics, engineering, medicine, etc. since all normalized statistics tend to look and behave in a familiar manner no matter what the source of the data. The most pervasive example of this is the tendency toward a "bell shaped" or "normal" curve for frequency plots or histograms of averages. This tendency is validated by the familiar central limit theorem of statistics that tells us that properly scaled and centered averages of independent and identically distributed observations have a limiting standard normal distribution as the sample size increases. The measurement of magnitude of standardized statistics is (what else?) the standard deviation. In fact, standardization is pervasive throughout all objective statistics.

Here we standardize the entire output series from a simulation run. This will permit us to use standard analysis techniques for different simulations and gain cumulative experience that is transferable to new simulation studies. This is because, after standardization, the output from any simulation will tend to look familiar to us no matter the source. (see Schruben, 1983 for some technical details not necessary here.)

For n output observations, Y_i, $i=1,2,...,n$, we define the (unscaled) standardized output series, $S_n(k)$, as follows. Let $S_n(0)=0$ and define

$$S_n(k) = \sum_{i=1}^{k} (Y_i - \bar{Y}_n)$$

or equivalently, using the previous definition of cumulative means,

$$S_n(k) = k\,(\bar{Y}_k - \bar{Y}_n)$$

or recursively, after observation Y_m,

$$S_m(k) = S_{m-1}(k) + (k/(m-1))\,(\bar{Y}_m - Y_m)$$

Since we are concentrating on graphical presentation we do not need to scale the length or magnitude of the output. Most plotting software routines do this in a standard manner that is satisfactory for our needs. Typically, plots are scaled to fill one window on the screen. Also, since we wish to emphasize the study of the dynamics of the simulation, we suggest that the recursion given above be used. The standardized output series is updated after each observation. Of course, it makes sense here as it did earlier to average batches of the raw output data and standardize the sequence of batched means. We will drop the subscript on run length, n, when it is not needed. S(k) can be interpreted as the cumulative deviation of the output series about the sample mean. It is typically uncorrelated and asymptotically independent of the sample mean; it focuses on the dynamics of the output.

Several characteristics of the standardized output, S(k), are useful, the most important being that all standardized output plots look similar no matter what is being simulated. Two quite different simulation outputs tend to have very much the same standardized plots. Just like the familiar use of normal summary statistics, experience in the study of standardized output plots is cumulative. The more standardized plots one studies the better one becomes in their analysis.

Standardization permits experimenters to focus on any unusual aspects of the run since any definition of "unusual" requires a standard for comparison. When there is a standard simulation output behavior to compare against (like the bell-shaped curve) analysis becomes much easier.

4. ASSESSING THE ACCURACY OF THE OUTPUT: CORRECTING INITIALIZATION BIAS

The starting conditions for each run of a simulation must be completely specified. Often these conditions are arbitrarily set at some convenient value. The simulation is then allowed to run for a "warm-up" period. The sequence of events near the beginning of a run is in general strongly influenced by these initial conditions. The early behavior of the model is therefore atypical of the system being modeled. In fact, the initial conditions for a run can have a greater influence on the accuracy of the results than any other system factor. This problem is referred to as simulation initialization bias. The point in a run which data is discarded is called the output truncation point.

Most simulation languages offer an easy method for prospectively truncating the simulation output but none offer any guidance as to what might be an appropriate warm-up period. Also, retrospective data truncation after the run has progressed for a time is much more sensible. Either way, practitioners are left to their own devices for this problem or referred to the sometimes obscure simulation methodology literature for help. We will see that the standardized output plot makes initialization bias control a very easy task.

Initialization bias can be very subtle. If a simulated factory is initialized with no work in progress, parts banks fully stocked, and all machines in perfect repair, then the simulated system behavior is biased. The system is not initially congested, so the performance of the factory will tend to look unrealistically good. Measures like part make-span (delay) and throughput are biased. Bias due to having no initial work-in-process is obvious but bias due to initial parts inventory levels, machine repair status and maintenance schedules, etc. are often overlooked or modeled incorrectly.

Even less obvious is the effect of the initial conditions on a service center such as a walk-in medical clinic or bank that closes every evening. The experimenter might think that starting each day with "no customers waiting" is sufficient. Indeed, it may be appropriate for the system as modeled. However, real service systems typically can experience a backlogging of demand (eg. follow-up appointments for the walk-in clinic, or customers returning the next day if they find too long a line at the bank). To simulate accurately the behavior of a system where there is a potential for backlogging, a backlog must initially be set (e.g. appointments scheduled) or be allowed to build up naturally.

Sometimes attempts to control initialization bias can introduce unrecognized dependencies between runs that otherwise could be assumed to be independent. For example, selecting a truncation point for a simulation run based on the output of a previous run will make the two runs dependent. Since it is not uncommon for the initial conditions to have a greater influence on the output than many of the system design factors, dependencies unwittingly introduced in this manner can be very problematic.

The central question is: how long should the simulation be run? The theoretical answer for many systems is that no run duration is long enough to allow the influence of the initial conditions to completely dissipate (there are some hypothetical systems that are exceptions). For an estimator to converge, the run duration must go to infinity so that both an infinite number of observations are kept and an infinite number of observations are discarded (the truncation point goes to infinity more slowly than the run duration.) For some details on this see Glynn and Heidelberger (1989). Also, if the system being simulated is inherently unstable then no warm-up period will make it stable; however, some warm-up period for unstable simulations is still to be recommended since the initial conditions still influence the behavior of the system and these conditions are most likely selected arbitrarily as a matter of convenience. Fortunately, in practice, it is usually possible to warm-up a simulation sufficiently.

The standardized output plots can help determine if the system is warmed-up. A characteristic feature of the standardized output, S(k), is that if the simulation output has a constant mean (say it has "warmed-up"), then its expected value is zero. That is,

$$E[Y_i] = E[Y_j] \text{ for all } i,j \Rightarrow E[S(k)] = 0$$

for all k; by definition $S(0) = S(n) = 0$. This tells us that if the standardized output tends to vary about zero throughout the run, it is likely that a sufficient warm-up period has passed before data collection has begun. If the warm-up period is too short the standardized plot will be pulled to one side of zero. Bias is very clear since the standardized plot is off to one side a power faster than underlying initialization bias. In fact, the standardized plots suggested here are similar to the CUSUM plots used in statistical quality control, the difference being that we are looking at the sum of deviations about the cumulative sample mean rather than some quality target. One of the criticisms of CUSUM control charts has been that they tend to react to a change in the mean too quickly. Here this sensitivity is good as it makes bias detection much easier than with plots of the raw output series.

From the standardized plot we can easily make a correction for initialization bias. This perhaps seems a bit like magic: we will estimate the bias in a particular value of a sample statistic without knowledge of the true parameter being estimated and using data that is independent of the estimator. If the simulation has warmed up by observation, say τ, then the expected slope of S(k) from τ to the end of the output is equal to the bias in the sample mean. That is, if the actual mean being estimated is μ then

$$E[\text{ slope } S(k),\ k=\tau,\dots,n\ ] = \text{bias}[\ \overline{Y}_n\ ]$$

This result can easily be verified by substituting μ for $E[Y_i]$ in the expected value of $S(k+1) - S(k)$ for $k > \tau$. This slope is simply added to the sample mean as a bias correction. This is actually equivalent to truncating the output at the last point the dotted line crosses the standardized output series.

In summary: if there is a drift in the mean of the simulation output (likely due to initialization effects) then the standardized output will be pulled off to one side of zero. If the standardized output looks linear near the end or the run then add an estimate of the slope to the sample mean as a bias correction. Estimation by eye is usually sufficient. If the standardized output does not appear to be linear near the end of the run then it is likely that the run duration is not long enough for the process to stabilize (it may never stabilize if the system is not itself stable).

5. ASSESSING THE PRECISION OF THE OUTPUT: INTERVAL ESTIMATION

It is widely recognized in the better simulation methodology texts that a simple estimate of some system performance measure is in itself not very helpful. Some measure of the precision of the estimate is also needed. Knowing that the observed output has a value of, say 48, is not nearly as meaningful as knowing that it is likely to be between 47 and 49 if the simulation is run again. A different decision might result if all that is known is that the true value of the quantity being estimated as equal to 48 is (loosely speaking) likely to fall somewhere between 5 and 91. Just as the experimenter needs to establish faith in the validity of a simulation model and the input data, some idea of the confidence that should be placed in the experimental results is needed.

A confidence interval for the true value of the output or a prediction interval for the output for the next simulation run (or k of the next m runs) value is important. Probably the best practical interval estimator available today is derived from the standardized output series. This estimator is known by the overly descriptive moniker, the weighted-area standardized time series interval estimator pooled with the classical batched means interval estimator. Expressions for this interval estimator are given in Goldsman and Schruben (1989) and will not be repeated here. This interval estimator is easily computed; it is recommended that all simulation output series be used to compute this interval estimator once initialization bias effects are dealt with satisfactorily.

A careful experimenter can with a single run of a simulation produce an accurate estimate of measures of system performance and also an estimate of the variance of the estimator. If care is taken then it is not necessary to have extra runs to permit "degrees of freedom for error" as in classical experiments. It is also easy to estimate the correlation between estimators at different input factor settings (details are given in Schruben and Margolin, 1978). Thus in simulation experiments, it is possible to use Generalized Least Squares analysis (with estimated error dispersion matrix) with what would ordinarily be a saturated experimental design (as many runs as unknown regression model parameters). We discuss some of the concepts of the design of multiple run experiments next.

PART II: GRAPHICAL METHODS FOR DESIGN

6. MOTIVATION

What is the purpose of experiment design for a series of runs? At the highest level, one wishes to choose parameter settings for a set of simulation runs in a way that will either:

i) maximize the amount of *relevant* information generated for a fixed number of runs

or

ii) minimize the number of runs required to gain the relevant information with a given degree of accuracy.

The ability to control the nature of random variation in simulation models provides unique opportunities in experiment design, and these are discussed regularly in WSC sessions. General topics in experiment design are given relatively little coverage in our presentations, yet many are relevant in a simulation setting. This section of the tutorial will focus on traditional topics of experiment design, treated in a non-traditional way. We will present graphical tools for designing experiments. The emphasis will be on two particular tools, causal diagrams and multidimensional point plots, and their role in the overall design process.

Historically, graphical methods for experiment design have not been recognized as an entity. A computerized literature search of scientific journals gave zero entries with keywords *graphical* and *experiment design* in the last ten years. Yet graphical methods have been used by outstanding statisticians to develop well known designs, including Box's central composite design.

Before presenting the graphical techniques, we define some fundamental concepts in experiment design and briefly discuss some of its important topics. The concepts are presented in a concise format for easy reference in the following section. The next section discusses the steps in designing an experiment, and how graphical techniques can be useful in several of these steps. Later sections give examples of causal diagrams and multidimensional point plots, and instructions for their construction.

7. TERMS AND TOPICS

Definition:
An **experiment** is a set of one or more runs (of a simulation model) made to meet a particular set of objectives.

Definition:
An **independent variable** is a parameter of the (simulation) system that can be explicitly adjusted by the experimenter.

Definition:
A **dependent variable** is an output of the (simulation) system that can be measured by the experimenter.

Definition:
An **intermediate variable** is a parameter of the (simulation) system that is affected by the settings of independent variables, and in turn affects the dependent variable(s) of interest.

Definition:
A **design factor** is an independent variable that will have its value changed during the course of an experiment.

Definition:

A **design frame** is a specification of

•which independent variables will be held fixed(& their values)

•design factors(& their ranges)

•what system outputs will be measured.

Definition:

An **experiment design** is a set of specifications of design factors for an experiment, along with a single specification vector for the settings of the independent variables that are not design factors. Experiment designs can be classified in several ways:

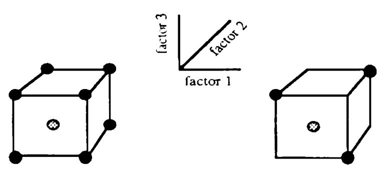
• purpose (pilot, screening, explanatory, confirmatory)

• run conditions (sequential or simultaneous specification)

• kinds of factors (continuous, discrete, mixed)

Definition:

A **factorial** design is an experiment with one or more runs for each possible combination of factor levels. If all factors are at two levels, then runs correspond graphically to vertices of an n-dimensional cube. For **fractional factorial** designs not all vertices are included in the design (see Figure 1 below). Two-level factorial designs can be used with both continuous and discrete design factors.



Full Factorial

2^3 (3 factors)

Fractional Factorial

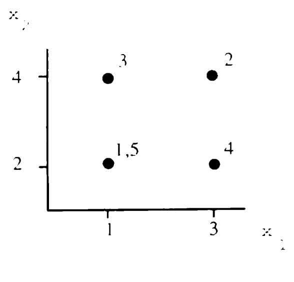
2^{3-1} (3 factors)

Figure 1. Graphical display of fractional and full factorial designs.

Definition:

A **design matrix** depends on the model to be fitted, and it is an experiment description usually reserved for general linear models. These models are of the form $y_i = \sum \beta_j f_j(x_{i1},x_{i2},\ldots,x_{ip})+\text{error}_i$, where the values of the constants β_j are to be estimated but the functions $f_j(\cdot)$ are given. There is a row for each (simulation model) run and a column in the matrix for each β_j term in the model (usually including β_0, where $f_0(x_{i1},x_{i2},\ldots,x_{ip}) \equiv 1$ for every run). The ith row of the matrix holds the values of the model terms $f_j(x_{i1},x_{i2},\ldots,x_{ip})$ for the corresponding run. This is illustrated by the example in Figure 2. The design points on the left are numbered according to their order in the design matrix. This order serves only to identify the points, and may or may not relate to the order in which the runs were made.

The variance-covariance of least-squares estimates of the β_j's is given by the matrix $\sum_{\hat{\beta}} = \sigma^2(X^TX)^{-1}$. Thus one can assess the quality of the estimates that a design will produce before ever running the experiment.



$$\begin{array}{cccc} 1 & x_1 & x_2 & x_1x_2 \\ \hline 1 & 1 & 2 & 2 \\ 1 & 3 & 4 & 12 \\ 1 & 1 & 4 & 4 \\ 1 & 3 & 2 & 6 \\ 1 & 1 & 2 & 2 \end{array}$$

DESIGN

DESIGN MATRIX
(CALLED 'X')

(Model: $y_i = \beta_0 + \beta_1x_{i1} + \beta_2x_{i2} + \beta_3x_{i1}x_{i2} + \text{error}_i$)

Figure 2. Example of graphical representation and the design matrix.

In addition to these definitions, there are several topics that will be important in our discussion of graphical methods. These are highlighted briefly below.

Nuisance variables: randomization and blocking

Nuisance variables are known to influence the results of the run, but are not controllable and/or not of direct interest. When the nature of the nuisance variable's influence is understood, it is treated like any other design factor. This approach is called **blocking**. When the nature of the influence is not understood or not repeatable, the run conditions should be assigned to values of the nuisance variable at random. This approach is called **randomization**. Examples might include the value of the random number seed, starting conditions, and time.

Confounding:

A particular experiment design may not allow the independent estimation of two or more model parameters. For example, if two design factors have their values changed from the previous run, and simulation output improves, which change (or both) caused the improvement? The effects of these factors are confounded. **Resolution** is a mathematical measure of the nature of confounding for various fractional factorial designs.

Bias:

Most experiment designs are constructed with a particular model of system response in mind. What if the postulated model is wrong? What impact will this have on parameter estimates?

Optimal Design:

The concept of optimality in experiment designs is only applied in a limited mathematical sense. Definitions of optimality are usually based on properties of the matrix X^TX, where X is the design matrix. For example, a **D-optimal** design is one which maximizes the determinant of X^TX (for a fixed number of rows in X and constraints on the x_{ij} values). Again, this concept usually refers to designs for estimating parameters from general linear models. As the D-optimal example shows, optimality depends on the particular model that is assumed.

8. STEPS IN EXPERIMENT DESIGN

Developing an experiment design involves five steps:

1. establish purpose,
2. identify and classify variables,
3. entertain possible models and select design factors,
4. choose a design, and
5. validate the design.

The goodness of a design depends on the goals of the experiment. These must be established clearly at the outset. Next, it is necessary to identify which variables are important for this experiment, and to classify them as independent, dependent, nuisance, or intermediate variables. The independent variables are divided into two groups; those that will be varied in the experiment (the design factors) and those that will be held fixed. Some design factors may be quantitative, others may be qualitative. One or more tentative models of system response are entertained to help decide which variables to include as design factors in step 3. The goodness of designs considered in step 4 depends on the postulated response model as well.

Given the number and kinds of design factors and a target number of runs, one attempts to find a suitable standard design. Often no known design will precisely fit the goals that the experimenter has in mind. Mathematical and graphical techniques have been proposed which can be used to create new designs with specific properties. For example, the DETMAX program constructs D-optimal designs for a given model (Mitchell, 1974). Graphical methods include Taguchi's linear graphs (Taguchi and Wu, 1980) and multidimensional point plots.

The final validation step includes checking the number and kinds of runs for feasibility, verifying that the variance of the estimates will be acceptable, and checking any confounding of effects allowed by the design. For general linear models, the latter two steps require a check of X^TX and/or $(X^TX)^{-1}$.

Graphical methods are particularly useful in steps 2,3 and 4 of this process. For steps 2 and 3, we will describe the use of causal diagrams. For step 4, we will show how to create and manipulate multidimensional point plots.

9. CAUSAL DIAGRAMS

Causal diagrams are particularly useful for establishing the design frame. Ishikawa 'fish-bone' diagrams are easy to construct and provide a view of the relationships of design factors, intermediate variables, and dependent variables. Figure 3 illustrates the use of this diagram to describe factors affecting vinyl disk warp in a vinyl videodisk pressing process. The final measure, disk warp, depends on a number of factors, which are presented as branches off the horizontal main. Each of these factors may in turn be due to other conditions, drawn as secondary bones or branches. Only factors at the end of these chains are directly controllable. All others are intermediate variables which cannot be used as design factors.

Fish-bone diagrams provide a graphical representation for the study of variable interactions and dependencies. These diagrams can also be a vehicle for communicating model structure to clients, and so they are useful in the development and validation phases of simulation modeling as well. A detailed description of this diagramming technique can be found in Ishikawa (1982). Horace Andrews also used high level diagrams for developing experiment designs, with more artistic content than the abstract fish-bone tool. For examples, see Andrews (1964).

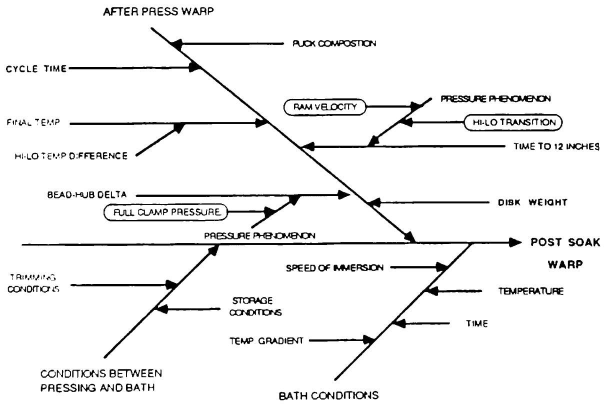


Figure 3. An Ishikawa fishbone diagram for variables affecting videodisk warp. From Young, Moore, and Girard (1987).

10. MULTIDIMENSIONAL POINT PLOTS

Why should we be interested in graphical methods for generating experiment designs? One reason is that standard designs often don't fit circumstances exactly. For example, one point in a 3^3

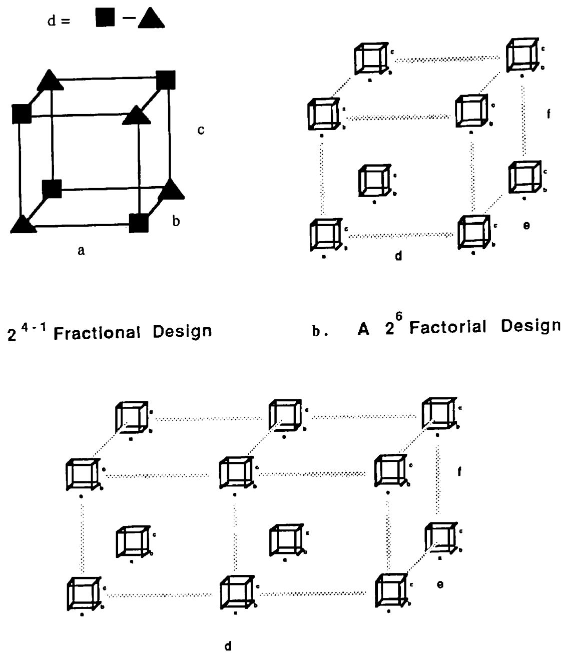
factorial design may correspond to infeasible operating conditions. How should the design be changed? With a graphical view of the design, such changes are easy to explore. For example, see Snee (1985b). It is easier to create new designs or change existing ones using graphical tools that interact with the creative right side of the brain (Edwards, 1979). Finally, the graphical representation of the design can be used as a frame on which to display the resulting simulation model output, making interpretation of results simpler. See, for example, the plots in Snee (1985a). In this section we describe the mechanics of multidimensional plots, and graphical rules for generating good designs.

Each experimental run is described by the values of the independent variables. Since only the (k) design factors will change over the course of the experiment, one can represent a single run by a point in the k-dimensional design space. Since k is often greater than two or three, we need techniques for graphically representing high dimensional sets of points. For points in general position, this is a difficult task. The general position problem is an important one in data analysis, and a number of attempts have been made to find useful representations, including the use of icons, multiple projections, and dynamic rotation of point sets.

Projections of point sets into two or three dimensional views is easier if the points are located on a regular grid rather than in general position. Fortunately this is the case for many useful experiment designs, including the factorial, fractional factorial, and central composite designs. By our definition above, latin squares, Plackett-Burman designs, and incomplete block designs can be viewed as examples of fractional factorial designs, and so they are good candidates for graphical presentation. Even designs created mathematically by programs like DETMAX can be viewed graphically if the set of candidate points for the design are limited to a relatively coarse grid.

In Figure 1 we showed multidimensional point plots for 2^3 full and fractional designs, but these can naturally be represented in three dimensions. What about 2^4 and higher designs? We will incorporate two tools for point sets in higher dimensions: icons and compound plots. Figure 4a shows a representation for a 2^4 fractional design. The icons ■ and ▲ are used to represent the high and low levels of the fourth factor. Figure 4b gives a template for a full or fractional design for six variables at two levels. Here we use a compound plot with nested cubes to represent the levels of all six factors.

The second and third parts of the figure can be used as templates for generating new fractional designs. Placing a circle at a vertex indicates that it is one of the experimental conditions that will be run. A number inside the circle can indicate run order, but for checking



a. A 2^{4-1} Fractional Design   b. A 2^6 Factorial Design

c. A $2^5 3^1$ Factorial Design

Figure 4. Multidimensional point plot templates.

the confounding of run order with main effects, a graphical representation of run order (shading or size of circle) is probably better. The design can easily be extended to several factors by using the icons from a. Figure 4c shows a $2^5 3^1$ design template. A $2^5 3^2$ design template would consist of three of these plots side by side.

Checking confounding of main effects with up to third order interactions is easy with multidimensional point plots. Figure 5 shows three 2^{4-1} fractional designs with different confounding patterns. These three patterns are easy to see and easy to create for factors at two levels.

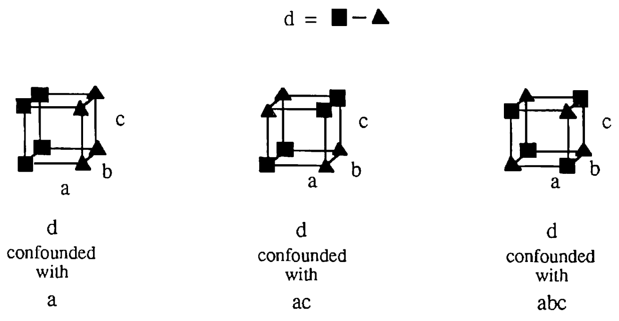


Figure 5. Confounding patterns for main effects and two and three way interactions.

			Block			
+	+	−	1	+	−	−
+	−	+		−	+	+
−	+	−	2	−	+	−
−	−	+		+	−	+
−	−	−	3	+	+	−
−	+	+		−	−	+
+	−	−		−	−	−
+	+	+	4	+	+	+

Figure 6. Good and bad blocked designs in traditional format. Can you tell which is bad?

Next we will illustrate a more complicated blocking design with three factors and four blocks of size 2. Box, Hunter, and Hunter (1978) present two designs for this situation, one good and one bad. These are illustrated in the traditional way in Figure 6. Can you tell which is good and which is bad? Figure 7 presents these designs graphically. We see that the blocks are confounded with the main effect of factor a in the bad design, but with the good design, effects due to differences in blocks are confounded with the two-way interactions, higher order terms that are presumably of less interest. The difference in the designs is particularly evident when one looks at projections of the designs along each axis. These projections are presented above, behind, and to the left of each cube.

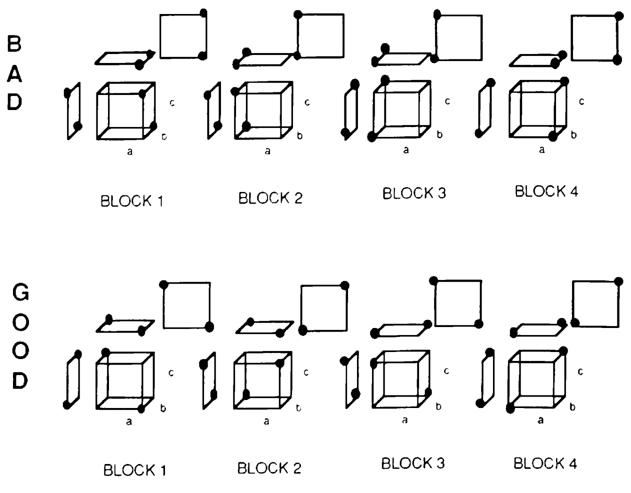


Figure 7. Good and bad blocked designs presented graphically.

Two primary issues that must be resolved in the choice of running conditions (i.e. design factor values) are *leverage* and *bias/estimability*. In general these are conflicting objectives, as illustrated in Figure 8. Consider the case of estimating the relationship between a simulation model response variable, y, and a single design parameter, x. If a linear relationship between x and y is postulated ($y = \beta_0 + \beta_1 x$ + error), placing half the runs each at the lower and upper limits of x will provide the greatest leverage. That is, the variance of $\hat{\beta}_1$ will be as small as possible for a fixed number of runs. This design, design A in the figure, is not reasonable if there is some uncertainty about the linear relationship. In this case, one or more intermediate points of x must be run, as illustrated in design B. For a simple linear relationship, however, design B is less efficient than design A.

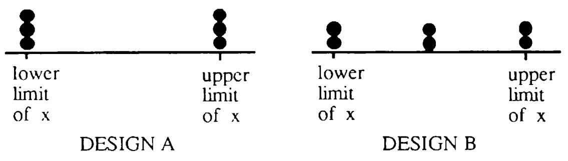


Figure 8. Two designs for estimating parameters of a simple linear model.

These issues can be described in graphical terms as well: maximizing leverage corresponds to placing the points at the extremes of the design space. Minimizing bias corresponds to distributing the points uniformly over the design space. How uniformly must the points be placed to minimize bias with minimal sacrifice in leverage? The answer is given by Box and Draper (1959):

"*...it is proved (Appendix 1) that if a polynomial of any degree d_1 is fitted by the method of least squares over any region of interest R in the k variables, when the true function is a polynomial of any degree $d_2 > d_1$, then the bias averaged over R is minimized for all values of the coefficients of the neglected terms by making the moments of order $d_1 + d_2$ and less of the design points equal to the corresponding moments of a uniform distribution over R.*"

Thus the degree to which points must be spread uniformly across the design space depends on the highest order model we wish to guard against. The bad blocking example above illustrates how projections are constructed. In the bad design, the points were not placed as far apart as possible. For the good design they were. How do we check graphically for this property? For designs involving many factors, it is important to check projections of the design space for uniform coverage.

A balanced design will have equal or roughly equal numbers of design points in each region of a projection. An aid in constructing such designs is to build them up from smaller geometric sets whose balance is easy to understand. The points of a central composite design, for example, can be constructed from two component sets of points: the vertices of a cube and points on the coordinate axes. All points are taken to be equidistant from the origin.

We have described several useful concepts for generating good designs from multidimensional point plots. These are summarized below:

RULES FOR GRAPHICAL DESIGN

1. COVER THE DESIGN SPACE UNIFORMLY
2. DECOMPOSE COMPLICATED DESIGNS INTO GRAPHICAL SUBCOMPONENTS
3. SPAN THE WHOLE DESIGN SPACE: ADDED POINTS SHOULD BE FAR FROM EXISTING POINTS TO MINIMIZE VARIANCE OF FIRST ORDER EFFECTS
4. CHECK PROJECTIONS OF THE DESIGN FOR BALANCE

11. GRAPHICAL METHODS; SUMMARY

Multidimensional point plots and causal diagrams are only two of many graphical tools for experiment design. There are nomographs for estimating sample size, network models relating connectivity to estimability of contrasts (Butz, 1982), graphical representations for nested designs (Andrews, 1964), mixture experiments (Snee, 1981), and many more. All of these techniques share some common strengths: a creative advantage from right-brain activity, easy to use and remember, easy to communicate to non-statisticians, robust to a degree, and easy to teach. It is also easier to make tradeoffs and accommodate design constraints graphically.

These methods also share a number of weaknesses. They are not usually quantitative; when they are (e.g. nomographs) they have little numerical precision. For multidimensional point plots, there are dimensional limitations which icons, compound plots, rotation, and projection can only relieve but not remove.

Graphical methods for experiment design should supplement rather than replace other methods for generating designs such as selection of existing designs and computer generated 'optimal' designs. When a new or modified design is necessary, the creative step of graphical design should always be paired with quantitative checks on the design validity.

REFERENCES

Andrews, H.P. (1964), "The Role of Statistics in Setting Food Specifications," *Proceedings of the Sixteenth Research Conference of the Research Council of the American Meat Institute*, 43-56. Reprinted in *Experiments in Industry: Design, Analysis, and Interpretation of Results*, eds. R.D. Snee, L.B. Hare, and J.R. Trout, Milwaukee: American Society for Quality Control, 1985.

Box, G.E.P., and Draper, N.R. (1959), "A Basis for the Selection of a Response Surface Design," *Journal of the American Statistical Association*, 54, 622-653.

Box, G.E.P., Hunter, W.G., and Hunter, J.S. (1978), *Statistics for Experimenters*, New York: Wiley.

Butz, L. (1982), "Connectivity in Multi-Factor Designs: A Combinatorial Approach," *Research and Education in Mathematics 3*, Berlin: Helderman Verlag.

Edwards, B. (1979), *Drawing on the Right Side of the Brain*, Boston: Houghton-Mifflin.

Glynn, P.W., and Heidelberger, P. (1989), "Analysis of Initial Transient Deletion for Replicated Steady State Simulations", presented at the Fall 1989 joint ORSA/TIMS conference, New York, NY, October, 1989.

Goldsman, D.M., and L. Schruben (1989), "New Confidence Interval Estimators Using Standardized Time Series", to appear in *Management Science*.

Ishikawa, K. (1982), *Guide to Quality Control*, Tokyo: Asian Productivity Organization, (second edition).

Mitchell, T.J. (1974), "An Algorithm for the Construction of D-Optimal Experimental Designs," *Technometrics*, 16, 203-210.

Nozari, A. (1986), "Confidence Intervals Based on Steady-State Continuous-Time Statistics", *Operations Research Letters*, 4.5, pp. 213-219.

Schruben, L. (1983), "Confidence Interval Estimation Using Standardized Time Series", *Operations Research*, 31.6, pp. 1090-1107.

Schruben, L., and Margolin, B. (1978), "Pseudorandom Number Assignment in Statistically Designed Simulation and Distribution Sampling Experiments", *Journ. Amer. Stat. Assoc.*, 73.363, pp. 504-525.

Snee, R.D. (1981), "Developing Blending Models for Gasoline and Other Mixtures," *Technometrics*, 23, 119-130.

Snee, R.D. (1985a), "Experimenting with a Large Number of Variables," in *Experiments in Industry: Design, Analysis, and Interpretation of Results*, eds. R.D. Snee, L.B. Hare, and J.R. Trout, Milwaukee: American Society for Quality Control.

Snee, R.D. (1985b), "Computer-Aided Design of Experiments -- Some Practical Experiences," *Journal of Quality Technology*, 17, 222-236.

Taguchi, G., and Wu, Y. (1980), *Introduction to Off-Line Quality Control*, The Central Japanese Quality Control Association (available from American Supplier Institute, 32100 Detroit Industrial Expressway, Romulus, MI 48174.

Welch, P. D. (1983), "The Statistical Analysis of Simulation Results", Chpt. 6 of *The Computer Performance Modeling Handbook*, Academic Press, pp. 267-329.

Young, A., Moore, M., and Girard, T. (1987), "A Warped Disk," OR516: Case Studies, Project Report, School of Operations Research and Industrial Engineering, Cornell University.

AUTHORS' BIOGRAPHIES

RUSSELL R. BARTON is a visiting associate professor in the School of Operations Research and Industrial Engineering at Cornell University. He received a B.S. in Electrical Engineering from Princeton University in 1973, and M.S. and Ph.D. degrees in operations research from Cornell University in 1975 and 1978, respectively. He was a member of technical staff at RCA Laboratories from 1978-1987. There he was involved in design of experiments, process optimization, and the optimization of differential equation-based simulation models. His current research interests include robust optimization methods for simulation models, simulation experiment design, and the interface of optimization and statistics. He is a member of ORSA, the Mathematical Programming Society, IEEE, Eta Kappa Nu, and Sigma Xi.

Russell R. Barton
SORIE
Upson Hall
Cornell University
Ithaca, NY 14853, U.S.A.
(607) 255-9899

LEE W. SCHRUBEN is a professor in the School of Operations Research and Industrial Engineering at Cornell University. He received his undergraduate degree in Engineering from Cornell University and a Masters degree from the University of North Carolina. His Ph.D. is from Yale University. Before going to graduate school he was a manufacturing system engineer with the Emerson Electric Company in St. Louis, Mo. His research interests are in the statistical design and analysis of large scale simulation experiments. His consulting activities have been primarily focused in the area of manufacturing systems simulation. He is a member of ASA, ORSA, SCS, and TIMS. He currently serves on several editorial boards for several journals..

Lee W. Schruben
SORIE
Upson Hall
Cornell University
Ithaca, NY 14853, U.S.A.
(607) 255-9133