Proceedings of the 2004 Winter Simulation Conference
R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

SIMULATION OUTPUT ANALYSIS: A TUTORIAL BASED ON ONE RESEARCH THREAD

Bruce Schmeiser

School of Industrial Engineering
Purdue University
West Lafayette, IN 47907–2023, U.S.A.

ABSTRACT

In reviewing topics in simulation output analysis, we advocate univariate analysis, micro/macro replications based on fixed sample sizes, overlapping batches, batch sizes based on mean squared error, dynamic batch sizes, and a concise format for reporting results.

1 INTRODUCTION

We follow one thread through the topic of simulation output analysis. That thread's core is composed of the Purdue Ph.D. dissertations by Kang (1984), Song (1988), Pedrosa (1994), Wood (1995), and Yeh (2002). Purdue M.S. theses by Scott (1990), Yeh (1999), and Wieland (2003) are related. As in earlier tutorials (Schmeiser and Song 1996 and Goldsman and Schmeiser 1997), the method of batching is central to this thread.

Output analysis addresses the problem of determining and reporting the quality of a given stochastic simulation experiment. Roughly, the issue centers on what is likely to happen if the experiment were run again with different random-number seeds.

This tutorial, which is a rewritten and updated discussion of some content in Schmeiser (1990), begins with a discussion of simulation experiments designed to estimate performance measures of the modeled system. The discussion then focuses on standard errors. Of the many approaches to estimating standard errors, only the approach of micro/macro replications is pursued. In the context steady-state simulations, a macro replication is called a batch; a central problem is to determine automatically a good batch size.

As in past tutorials (Schmeiser 1992b, Schmeiser 2001), what follows is relatively informal and contains opinions, many of which are widely accepted and some of which are not. Due to the space constraint, references are limited to the one research thread, ignoring substantial fine work by others.

2 SIMULATION EXPERIMENTS

The simulation experiments considered here are of two types. The first type estimates performance measures of a stochastic system; examples include many industrial, military, and financial systems. The second type estimates the solution of a deterministic problem with Monte Carlo sampling; examples include integration and Markov Chain Monte Carlo experiments (which are analogous to simulating steady-state systems Schmeiser 1992a). As does the Winter Simulation Conference (WSC), we think mostly of the former type, but the discussion throughout applies to both types.

We now review a simulation world view, performance measures, point estimation, and sources of error.

2.1 World View

As first discussed in Nelson (1983), all simulation experiments that we consider can be viewed as

$$G \to U \to X \to Y \to \widehat{\theta}.$$

Here G represents a source of randomness, typically one or more pseudorandom-number generators. The source G is used to generate *random numbers* U, almost always assumed to be independent and uniformly distributed over the unit interval. The random numbers U are used to generate *random variates* X, whose distribution is known via a given *input model*. The random variates X are transformed into *output variates* Y via a *logic model*. The output variates Y are used to compute *point estimators* $\widehat{\theta}$ of *performance measures* θ whose unknown values are properties of the distribution of Y.

The purpose of such simulation experiments is to determine the values of the performance measures, θ. As such, simulation is a competitor to closed-form and numerical analysis of the *probability model* specified by the input model and the logical model.

2.2 Performance Measures

In applications where analysis is via simulation experiments, multiple performance measures are common. Therefore, θ is typically best thought of as being a vector of scalar measures. These scalar measures are often means (such as expected number of customers in the system), with an important special case being probabilities (such as the probability the the number of customers is less then ten). Other measures are common, including other distribution moments (such as standard deviations and variances), quantiles (such as the median number of customers in the system), and dependence measures (such as the correlation between the number of customers at Station 1 and Station 2).

At the WSC, such performance measures are often defined for models with continuous time and discrete states. For our discussion, however, time can be discrete or continuous, or even absent, with no concept of time in the model. Similarly, states can be discrete or continuous or mixed. Performance measures can be defined in terms of *steady-state* behavior or *transient* behavior. Transient behavior can arise from a known initial state or because of seasonality or because the system terminates. When comparing several systems, we still think of having one simulation experiment, with a classic performance measure being the difference between some mean (such as expected cycle time) for each of a pair of systems. We are interested in methods that apply in general.

2.3 Point Estimators of Performance Measures

The "answer" provided by a simulation experiment is $\widehat{\theta}$, the vector of observed *point estimates*, which is hoped to be close to the unknown performance-measure vector θ. The vector of point estimates is an observation of the vector of point estimators; by definition there is only one such observation arising from the simulation experiment. If there appears to be more than observation $\widehat{\theta}$, these are averaged to create the single observation. The fundamental difficulty in simulation output analysis is that we wish to make a statement about the quality of the single point estimate $\widehat{\theta}$.

The point estimator $\widehat{\theta}$ is a function of the output variates Y. The function used is typically the "natural" estimator. For example, a probability is estimated with the fraction of successes, an expected value is estimated with the sample mean $\overline{Y} = \sum_{i=1}^{n} Y_i/n$, a variance is estimated with sample variances $S^2 = (\sum_{i=1}^{n} Y_i^2 - n\overline{Y}^2)/(n-1)$ and a coefficient of variation is estimated with $S/\overline{Y}$.

Such natural estimators are appropriate regardless of whether the data Y are independent. For example, the output data might be time in a steady-state system for n consecutive customers. In such an autocorrelated situation, the sample variance is biased. The bias, however, is asymptotically negligible and is small compared to the standard deviation of S^2 (Ceylan and Schmeiser 1993, Wood 1995). That the natural estimators work well regardless of correlations is reflected in their use by commercial simulation languages. The same point is supported by the histograms and empirical cumulative distribution functions reported by some commercial simulation languages.

The functional form can differ from the natural estimator in many smart ways. For example, initial data are often ignored when estimating steady-state performance measures. For example, when estimating a variance when the mean μ_Y is known, the sample variance S^2 can be replaced to gain the extra degree of freedom provided by $\sum_{i=1}^{n}(Y_i-\mu_Y)^2/n$. For example, using the midpoints of histograms cells (rather than storing all observations) can be computationally efficient at small cost in lost statistical information (Schmeiser and Deutsch 1977). Our goal is to have output analysis that does not depend upon the form of the point estimators.

2.4 Error Sources in Simulation Experiments

The statistical quality of a simulation experiment is a function of the *sampling distribution* of the vector of point estimators. For any one scalar point estimator, the mean squared error

$$\mathrm{Mse}(\widehat{\theta},\theta) = \mathrm{E}[(\widehat{\theta}-\theta)^2] = \mathrm{Bias}^2[\widehat{\theta},\ \theta] + \mathrm{Var}(\widehat{\theta})$$

is a classical measure of quality. Viewed this way, bias and variance are the two ways that a simulation experiment can fail.

Bias can arise from at least six sources. Five sources must be controlled by the construction of the experiment. First, the pseudorandom numbers U at best only appear to be independent and uniformly distributed on the unit interval. Second, the distribution of the random variates X can differ from the known input model, often for convenience, such as using $x = (u^{0.135} - (1-u)^{0.135})/0.1975$ as a simple approximation to generate a standard normal random variate. Third, initial transients and stopping rules can bias point estimators. Fourth, some good point estimators are inherently biased, such as using order statistics to estimate quantiles. Fifth, the computer-number system is only an approximation to the real-number system; for example, all distributions are bounded on a computer and computations have round-off error.

The sixth source of bias is *modeling error*. It also should be small by construction of the experiment, but unlike the other five sources of error it is quite application dependent, with an important tradeoff between model tractability and the difference between the real-world's performance measure, say θ_0, and the simulated model's performance measure θ. This modeling error can arise from error in the input model, which is often estimated from real-world data, or from error in the logical model, which is often intentional to simplify coding. Sensitivity analysis can be used to

provide a sense of the effect of the unknown modeling error. Simulation practitioners need to understand that the simulation-experiment's point estimators know nothing of θ_0; by definition the experiment's purpose is to estimate the model's θ. Others disagree (Barton et al. 2002).

The effect of the six sources of bias depend on the run length (i.e., sample size of the output data Y). Simulation run lengths are typically long, much longer than corresponding real-world experiments. Long runs can expose bias caused by flaws in random-number generators, random-variate generators, and computer arithmetic. Modeling error that is negligible at short run lengths can become dominant for long runs. On the other hand, long runs make negligible the effects of initial transients, stopping rules, and estimator bias.

Variance arises for only one reason: *sampling error*. If the experiment were to be replicated using a different source of randomness (typically different random-number seeds), the point estimate would change. Sampling error depends on the sample size of the output data Y. Ideally—with zero bias and with reasonable point estimators—as the run length goes to infinity, the sampling distribution converges to the point θ.

3 STANDARD ERROR

Sources of error do, however, exist. Bias and sampling error are always present. Simulation output analysis is concerned with telling the simulation practitioner the quality of the point estimate. Because even a bad experiment could by accident have $\widehat{\theta} = \theta$, the quality of the point estimate must be discussed in terms of the sampling distribution of the point estimator $\widehat{\theta}$.

In this section we discuss the central role of the *standard error*, the standard deviation of any one scalar component of $\widehat{\theta}$. We first explain why other properties of the sampling distribution are ignored, then argue against various classical uses of the standard error, and propose a simple way of using standard errors to report point estimators and their precision.

3.1 As a Measure of Sampling Error

With substantial generality, long simulation run length causes the sampling distribution of $\widehat{\theta}$ to be approximately normal. This central-limit-theorem effect applies directly to sample means, but also to sample standard deviations, sample variances, sample correlations, and sample quantiles.

Occasionally, non-normality is noticeable. Distribution kurtosis is

$$\alpha_4 = \mathrm{E}[(Y - \mathrm{E}(Y))^4]/\mathrm{Var}^2(Y).$$

For normal output data Y, the kurtosis is $\alpha_4 = 3$; the sample mean is normally distributed for every run length n; but the natural kurtosis point estimator,

$$\widehat{\alpha}_4 = n^{-1} \sum_{i=1}^{n} \frac{(Y_i - \overline{Y})^4}{S^4},$$

is skewed, with a mode that is far less than three, even for long run lengths. The reason is that the fourth power makes the tails quite important, but in most replications the extreme tails are under represented. Of course, $\widehat{\alpha}_4$ is a sum, so large n does induce normality.

Under the assumption of point-estimator normality, the sampling distribution is characterized by the bias and variance of $\widehat{\theta}$. Therefore, $\mathrm{Mse}(\widehat{\theta}, \theta)$ is the ideal measure of quality. By definition, however, the value of the performance measure θ is unknown, so we cannot estimate mse because the bias is unknown. We assume zero bias, which is widely acceptable (even for $\widehat{\alpha}_4$).

Therefore, the quality of the point estimator centers on $\mathrm{Var}(\widehat{\theta})$. Mean absolute deviation could be used, but as in most applications the mathematical convenience of variance leads to its dominant use. Often the variance is forsaken for its square root, $\mathrm{Ste}(\widehat{\theta})$, because the standard error has the same units as θ and it is the quantity used in many classical analyses.

So far, we have discussed only a single performance measure. The covariance matrix of $\widehat{\theta}$ is the natural extension of the variance when $\widehat{\theta}$ is a vector. The covariance matrix contains more information than the vector of standard errors (the square roots of covariance-matrix's diagonal elements). In addition, the covariance matrix can be estimated. In the next subsection, about uses of standard error, we argue that scalar standard errors suffice for most simulation applications.

3.2 Classical Uses

Before discussing standard-error estimation in the next subsection, we discuss some uses of the estimate $\widehat{\mathrm{Ste}}(\widehat{\theta})$.

The most classic use is to create a confidence interval. Under the assumption of a normal sampling distribution, a $((1-\alpha) \times 100)$-percent confidence interval for θ is

$$\widehat{\theta} \pm z_{1-(\alpha/2)} \times \widehat{\mathrm{Ste}}(\widehat{\theta}),$$

where z_p is the pth quantile of the standard-normal distribution. If the quality of the standard error is not so good, the standard-normal z_p commonly is replaced by the corresponding Student-T value $t_{1-(\alpha/2),\nu}$, with degrees of freedom ν chosen (somehow) to reflect the quality of the standard-error estimator. Because the underlying assumptions are not true, all such confidence intervals are

approximate. In addition, few readers of such confidence intervals can explain their meaning. Schmeiser (2001) discusses other reasons to avoid confidence intervals. (Given, however, that confidence interval procedures continue to be developed, see Kang and Schmeiser 1990 and Schmeiser and Yeh 2002.)

When θ is a vector, such a confidence interval can be computed for each component. If the covariance matrix were estimated, the corresponding multidimensional ellipsoid confidence region could be computed, centered at $\widehat{\theta}$. Even more so than for single-dimensional confidence intervals, we avoid confidence regions for simulation experiments. What is a typical user to do with such a region, especially in dimensions higher than two?

Hypothesis testing is another classical use of standard errors. Schmeiser (2001) discusses reasons to avoid hypothesis testing. The key reason is that in simulation experiments the model is created by the practitioner, so the null hypothesis is known to be false (or, maybe occasionally, true). The hypothesis test is then really a test of whether the simulation-experiment's run length is sufficient to detect that the null hypothesis is false. Short runs fail to reject; long runs reject. Not very interesting.

Standard errors are also used to determine when to stop running a simulation experiment. Various algorithms exist to stop when the run length is sufficient to provide some specified property, such as a confidence interval of a specified width. Most experiments' run lengths, however, are determined by the amount of time available. Graduate students notoriously run thesis simulation experiments for days and weeks, whatever time is available before the defense. Automated stopping rules can easily ask an experiment to run for years or centuries, depending upon the specified precision. Coupled with the usual situation where there are multiple performance measures, dynamic stopping rules seem applicable in few applications. Exceptions are when comparing systems, either a small number of given systems (Goldsman, Nelson, and Schmeiser 1991) or within optimization and root-finding algorithms.

3.3 Reporting Simulation Results: A Proposal

Despite seeing little purpose for Subsection 3.2's classical uses of the estimated standard error, we do think that reporting the precision of a simulation-experiment's point estimators is important. Such reporting, of course, is the purpose of simulation output analysis and the topic of this tutorial.

Song and Schmeiser (1994) argue for a simple method of reporting the results of a simulation experiment. Such experiments often have many point estimates, with space in the reporting page at a premium. Therefore, the goal is to report the point estimates—and their precisions—unambiguously and concisely.

Suppose that the simulation experiment has been run and that a vector of point estimates $\widehat{\theta}$ and a corresponding vector of standard-error estimates $\widehat{\text{Ste}}(\widehat{\theta})$ are available. Our purpose is to report each point estimate and its precision, as indicated by its standard error.

There are two underlying ideas. (1) Print only meaningful digits of the point estimate. (2) Print, at most, the two leading digits of the standard-error estimate. This leaves the issues of defining *meaningful* and of formatting the printed information. Here is one concrete proposal. Define a digit of the point estimate as being meaningful if it is not to the right of the leading digit of the standard-error estimate; the leading digit is the left-most non-zero digit. For example, suppose that $\widehat{\theta} = 1234.56789$ and $\widehat{\text{Ste}}(\widehat{\theta}) = 0.0345678$. Then the leading digit of $\widehat{\text{Ste}}(\widehat{\theta})$ is the three in the one-hundreths column, so the meaningful digits of $\widehat{\theta}$ are 1234.57, obtained by rounding the last meaningful digit. We advocate reporting these values with the format 1234.57@3, read as "$\widehat{\theta}$ is equal to 1234.57 at standard error 0.03".

4 ESTIMATING STANDARD ERRORS

We now discuss methods for estimating standard errors. For simplicity, we assume that the standard error of one component of θ is to be estimated from output identically distributed observations $Y_1, Y_2, \ldots, Y_n$. Analogous discussion applies if, for example, Y exists continuously in time, such as the number of customers in the system.

4.1 Classical Special Cases

In a first course, two special cases for independent and identically distributed (iid) data are invariably covered. In the first case, θ is the probability of an event A. Then the point estimator $\widehat{\theta}$ is $\widehat{p}$, the relative frequency of observations in A, the standard error is $\text{Ste}(\widehat{p}) = [(p(1-p))/n]^{1/2}$, and the usual standard-error estimator is $\widehat{\text{Ste}}(\widehat{p}) = [(\widehat{p}(1-\widehat{p}))/(n-1)]^{1/2}$. In the second case, θ is the mean $\text{E}(Y)$. Then the point estimator is the sample mean $\overline{Y}$, the standard error is $\text{Ste}(\overline{Y}) = \text{Std}(Y)/\sqrt{n}$, and the standard-error estimator is $\widehat{\text{Ste}}(\overline{Y}) = S/\sqrt{n}$.

Other special cases are less common and less tractable. For example, assume iid data and that the performance measure is $\theta = \text{Var}(Y)$. The point estimator is the sample variance, S^2, and its standard error is the square root of $[\text{Var}^2(Y)/n][\alpha_4 - (n-1)/(n-3)]$. A natural standard-error estimator is obtained by estimating the variance $\text{Var}(Y)$ and the kurtosis α_4.

For estimating means from steady-state data,

$$\text{Var}(\overline{Y}) = \frac{\text{Var}(Y)}{n}[1 + 2\sum_{h=1}^{n}(1 - \frac{h}{n})\rho_h], \qquad (1)$$

where $\rho_h = \text{Corr}(Y_i, Y_{i+h})$ is the lag-h autocorrelation. In the limit as the run length n goes to infinity,

$$n\text{Var}(\overline{Y}) = \text{Var}(Y)\gamma_0,$$

where $\gamma_0 = 1 + 2\sum_{h=1}^{\infty} \rho_h$ is the (asymptotic) number of dependent observations that contain the same statistical information as one independent observation. That is, the n dependent observations are equivalent to n/γ_0 independent observations. When, as typically occurs in queueing models, the autocorrelations are positive, assuming the iid value of $\gamma_0 = 1$ leads to underestimating $\text{Var}(\overline{Y})$, giving the misleading interpretation that the point estimator is better than it is.

The natural standard-error estimator obtained from Equation 1 replaces the variance $\text{Var}(Y)$ by S^2 and the autocorrelations ρ_h by an autocorrelation estimator such as

$$\widehat{\rho}_h = \frac{(n-h)^{-1}\sum_{i=1}^{n-h} Y_i Y_{i+h} - \overline{Y}^2}{S^2}.$$

Although the variance estimator S^2 cancels, and therefore does not need to be calculated, the natural standard-error estimator requires $O(n^2)$ computation. Such a computational effort is inappropriate since, unlike real-world experiments, computing effort spent in estimating the standard error could be used to reduce the standard error (by increasing the value of n). Worse, the natural estimator is not good statistically because for large values of n most of the autocorrelations are essentially zero; their estimators $\widehat{\rho}_h$, however, are subject to random errors and worse, they are themselves positively correlated, causing the random effects to accumulate.

Far better is truncating to only m autocorrelations and using an estimator such as

$$\widehat{\text{Var}}(\overline{Y}) = \frac{S^2}{n}[1 + 2\sum_{h=1}^{m}(1 - \frac{h}{m})\widehat{\rho}_h], \qquad (2)$$

which requires only $O(nm)$ computation and is statistically better—assuming that m is chosen to ignore only ρ_h values that are negligible. The weighting factor, sometimes called the *lag window*, decreases from one to zero; its triangular shape is a bit arbitrary, but the shape of the window is less important than the value of the window width m.

Equation 2 has two disadvantages: it applies only to sample means and a good value of the parameter m needs to be determined. Both disadvantages are addressed in the next section.

4.2 Micro/Macro Replications

We wish to have a standard-error estimator that is general, computationally efficient, and statistically good. General in the sense of any stationary data and for any point estimator $\widehat{\theta}$, not restricted to sample means. Computationally efficient in the sense of $O(n)$ or a bit larger, but certainly much less than $O(n^2)$. Statistically good in the sense that the standard-error estimator should have a small mean squared error.

In this section, we discuss using micro/macro replications to estimate standard errors. In addition to the three criteria of the previous paragraph, the method uses only the elementary statistics of Equation 2 and is therefore reasonably easy to implement and to explain.

The approach is to estimate the standard error not for the usual *grand* point estimator $\widehat{\theta}$, but for a related *micro/macro* estimator. Partition the experiment's run length n into k equal-size contiguous macro replications, each composed of $m = n/k$ micro replications. The micro/macro point estimator is then

$$\overline{\widehat{\theta}} = \frac{\sum_{j=1}^{k} \widehat{\theta}_j}{k},$$

where each $\widehat{\theta}_j$ is defined analogously to $\widehat{\theta}$ but is calculated using only the data in the jth of the k macro replications. If $\widehat{\theta}$ is the sample mean of the experiment's n observations, then $\widehat{\theta} = \overline{\widehat{\theta}}$ for any values of m and k. If $k = 1$ and $m = n$, then again $\widehat{\theta} = \overline{\widehat{\theta}}$.

The useful key thought is that, in general, for small numbers of macro replications k, $\widehat{\theta} \approx \overline{\widehat{\theta}}$. Why? First, the grand point estimator's bias is $O(n^{-1})$ and the micro/macro point estimator's bias is $O(m^{-1})$. Second, both the point estimator's variances are $O(n^{-1})$. Third, because they are computed from the same output data, the correlation between $\widehat{\theta}$ and $\overline{\widehat{\theta}}$ is quite high. For the first reason, the grand point estimator is preferred to the micro/macro point estimator. For the second and third reasons, we calculate the standard error of the micro/macro point estimator and report it as the standard error of the reported grand point estimator.

The micro/macro point estimator is, regardless of the functional form of $\widehat{\theta}$, the average of the k macro-replication estimators $\widehat{\theta}_j$. Because it is an average of identically distributed data, Equation 2 applies, with ρ_h now referring to the autocorrelation between macro-replication point estimators $\widehat{\theta}_j$ and $\widehat{\theta}_{j+h}$. Typically the value of k is small, however, so rather than trying to estimate the autocorrelations between batches, the value of k is chosen so that the macro-replication size $m = n/k$ is large (see Subsection 4.5). Large values of m provide almost independent macro-replication estimators $\widehat{\theta}_j$ (e.g., Kang and Schmeiser 1987), which in turn yields the standard-error estimator

$$\widehat{\text{Var}}(\overline{\widehat{\theta}}) = \frac{S^2_{\widehat{\theta}}}{k}, \qquad (3)$$

where $S_\theta^2 = [\sum_{j=1}^{k} \widehat{\theta}_j^2 - k\overline{\widehat{\theta}}^2]/(k-1)$. Maybe surprisingly, for steady-state output data, the standard-error estimators of Equations 2 and 3 are asymptotically equal, differing only in end effects.

The mathematics of micro/macro replications apply to any simulation experiment. (Scott 1990 and Schmeiser and Scott 1991 discuss SERVO, general-purpose software for obtaining standard errors for all point estimators.) In the simplest case of iid output data, the only reason for a large value of m micro replications is normality of the macro-replication estimators, since they are independent for any values of k and m. For stationary output data, the macro replications can be either truly independent replications or asymptotically independent batches from one run. The choice is a perennial topic, but the tradeoff is simple: truly independent macro replications incur the disadvantage of the initial-transient bias in each macro replication; asymptotically independent batches from one run incur the initial-transient bias only once. Because methods to deal with the initial-bias transient are ad-hoc and often difficult to implement, my preference is to use one run.

Typically WSC output-analysis tutorials discuss various other approaches, such as the regenerative method and standardized time series, for estimating standard errors. The alternatives to micro/macro replications, however, focus almost entirely on sample averages. The ability to generalize to any type of point estimator is unique to micro/macro replications.

4.3 Steady-State Batching

For steady-state output data, micro/macro replications is usually called the method of nonoverlapping batching. Using nonoverlapping batches is intuitively appealing, but alternatives are reasonable and can improve the quality of the standard-error estimator. (As in the previous subsection, the grand estimator $\widehat{\theta}$ is calculated from the simulation experiment's n output observations and does not depend upon batching strategy.) There are $n-m+1$ batches, all identically distributed with size m. Let $\mathcal{B}$ denote a subset of the integers $\{1, 2, \ldots, n-m+1\}$ and let $|\mathcal{B}|$ denote the cardinality of $\mathcal{B}$. Then for large batch sizes m, for any selection of $\mathcal{B}$ a reasonable standard-error estimator is

$$\widehat{\mathrm{Var}}(\overline{\widehat{\theta}}) = \frac{S_{\mathcal{B}}^2}{n/m}, \tag{4}$$

where $S_{\mathcal{B}}^2 = [\sum_{j\in\mathcal{B}} \widehat{\theta}_j^2 - |\mathcal{B}|\overline{\widehat{\theta}}^2]/(|\mathcal{B}|-1)$.

Nonoverlapping batches, Equation 3, is the special case of $\mathcal{B}= \{1, m+1, 2m+1, \ldots, (k-1)m+1\}$. Overlapping batches (Meketon and Schmeiser 1984) is $\mathcal{B}= \{1, 2, \ldots, n+m-1\}$. Partially overlapping batches and spaced batches are defined analogously.

Regardless of the choice of $\mathcal{B}$, the expected value of $S_{\mathcal{B}}^2$ is $\mathrm{E}[(\widehat{\theta}_j - \overline{\widehat{\theta}})^2]$, which depends only on the batch size m. Because of the correction factor n/m, which is the asymptotic ratio of variances between a batch of size n and of size m, the expected value of $\widehat{\mathrm{Var}}(\overline{\widehat{\theta}})$ in Equation 4 does not depend on the choice of $\mathcal{B}$.

A key point is that independence of the batch statistics, $\widehat{\theta}_j$, is not important for obtaining a good standard-error estimator. Because all batches are identically distributed, the best statistical performance arises with overlapping batches, despite that being the alternative with the highest autocorrelation between batch statistics $\widehat{\theta}_j$. Spacing batches, by omitting some output data from $S_{\mathcal{B}}^2$, reduces dependence among batch statistics but increases the variance of the standard-error estimator because fewer batches are used in the computation. Song and Schmeiser (1993) show the advantage of overlapping estimators by viewing graphs of various estimators' quadratic forms.

4.4 Determining Batch Size

For any batching method $\mathcal{B}$, the batch size m needs to be chosen. What are the tradeoffs? Large batch sizes yield the needed asymptotic property that the variance of batch statistics is inversely proportional to batch size. Large batch sizes yield batch statistics that are normally distributed. Small batch sizes yield more batches. That is, the choice of batch size m balances bias, due to poor asymptotics, and variance, due to few batches.

Schmeiser (1982), in considering nonoverlapping batches for confidence intervals on the mean, advocates choosing $10 \le k \le 30$, even when the run length n is quite large. This is reasonable (even for general batch statistics), since the quality of $S_{\mathcal{B}}^2$ is improved little with many batches k, but the correction factor n/m can be arbitrarily inappropriate when the asymptotics do not hold.

Given that the asymptotics hold, however, more batches are better than fewer batches, which argues that the optimal choices of m and k should go to infinity as the run length n goes to infinity. The resulting consistency of the standard-error estimator is appealing, as is having a formula to automate the choice.

4.5 Minimizing MSE

Most of the literature of batching methods is specialized to batch means. (Exceptions include Schmeiser, Avramidis, and Hashem (1990), Hashem and Schmeiser (1994), Wood (1995), Wood and Schmeiser (1994), and Wood and Schmeiser (1995), who consider batch variances and batch quantiles.) This section assumes batch means, but the high-level conclusions about batch sizes apply to all batch statistics.

Goldsman and Meketon (1986) first advocated the use of mean squared error to choose the batch size m. In particular, the mse-optimal batch size m^* minimizes

$$\text{Mse}[\widehat{\text{Var}}(\widehat{\theta}), \text{Var}(\widehat{\theta})].$$

The mse depends upon the output-data process, the choice of batching strategy, and the run length n. (So, using the same output process and run length, batching strategies can be compared using the mse criterion.)

Following Goldsman and Meketon (1986), Song (1988) and Song and Schmeiser (1995) show that the asymptotic mse-optimal batch size is

$$m^* = [2n \frac{c_\text{b}}{c_\text{v}} \frac{\gamma_1}{\gamma_0}]^{1/3} + 1. \quad (5)$$

Here c_b and c_v are bias and variance constants that depend upon the batching strategy; γ_0 and $\gamma_1 = 2\sum_{h=1}^{\infty} h\rho_h$ are characteristics of the output process.

For a given experiment, the asymptotic minimal mse depends upon the batching strategy only via the product $c_\text{b}c_\text{v}$. In this sense, overlapping batch means is the best strategy among the family of standardized-time-series and batch-means estimators.

Because the bias and variance constants are known for any given batching strategy (Goldsman and Meketon 1986, Song and Schmeiser 1994), and because the run length n is known, the problem of estimating mse-optimal batch size reduces to estimating the output process's *center of gravity* γ_1/γ_0. Estimating the individual autocorrelations ρ_h is unnecessary. Pedrosa (1994), based on Pedrosa and Schmeiser (1993), develops the 121-OBM method that uses two evaluations of the overlapping-batch-means estimator, with two batch sizes that differ by one, to estimate the center of gravity in $O(n)$ computation.

Because the center of gravity must be estimated, the optimal batch size from Equation 5 also is estimated. Song and Schmeiser (1995) use the second derivative—with respect to batch size—of the optimal mse as a measure of robustness to the effect of using estimated, rather than actual, optimal batch size. This second derivative is proportional to c_v^2/c_b. Using this measure, the linear combination of nonoverlapping batch means and the standardized-area estimator (Schruben 1983) has substantially better robustness than batch means or standardized-area estimators alone.

Because optimal batch sizes depend upon the estimator, and because estimated batch size is a function of the estimated center of gravity, Yeh and Schmeiser (2004) argue that robustness measure should be the second derivative of the mse with respect to the center of gravity. This measure is proportional to $c_\text{b}c_\text{v}$. In this sense, mse and robustness are proportional, with the result that overlapping batch means is the preferred estimator in both senses.

Yeh (1999), Yeh and Schmeiser (2000), and Yeh (2002) discuss dynamic batching, the context where the output analysis is not allowed to store the entire realization of output data for later analysis.

5 CONCLUDING THOUGHTS

The literature of estimating the standard error of the point estimator $\widehat{\theta}$ is vast, but only for the sample average. Even for that special case, no method exists to ensure that the standard error is finite. Finiteness is, however, guaranteed because the experiment is performed on a computer, where all realizations are finite.

More important, no method exists for determining whether the performance measure is finite. Certainly many queueing systems are unstable, with expected number of customers going to infinity. Such behavior looks much like an initial transient. Wieland (2003) and Wieland, Pasupathy, and Schmeiser (2003) discuss the problem of designing a method for determining whether a given simulated system is stable.

An appealing idea is to use linear combinations of batching estimators that have differing batch sizes m. Schmeiser and Song (1987) and Song and Schmeiser (1988a) provide empirical evidence. Song and Schmeiser (1988b, 1988c), Pedrosa (1994), and Pedrosa and Schmeiser (1993) provide useful results for determining the linear-combination weights.

ACKNOWLEDGMENTS

I thank Marvin Nakayama for inviting me to prepare this advanced tutorial, the editors for patience and care, and Huifen Chen, Raghu Pasupathy, Wheyming Song, and Yingchieh Yeh for suggestions to improve the presentation.

REFERENCES

Barton, R.R., R.C.H. Cheng, S.E. Chick, S.G. Henderson, A.M. Law, L.M. Leemis, B.W. Schmeiser, L.W. Schruben, and J.R. Wilson. 2002. Panel on current issues in simulation input modeling. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yucesan, C.-H. Chen, J.L. Snowdon, and J.M. Charnes, 353–369. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Ceylan, D. (Wood), and B. Schmeiser. 1993. Interlaced variance estimators. In *Proceedings of the 1993 Winter Simulation Conference*, ed. G.W. Evans, M. Mollaghasemi, E.C. Russell, and W.E. Biles, 1382–1383. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Goldsman, D.G., and M.S. Meketon. 1986. A comparison of several variance estimators. Technical Report J-

85-12, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia.

Goldsman, D.G., and B. Schmeiser. 1997. Computational efficiency of batching methods. In *Proceedings of the 1997 Winter Simulation Conference*, ed. S. Andradóttir, K.J. Healy, D.H. Withers, and B.L. Nelson, 202–207. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Goldsman, D.G., B.L. Nelson, and B. Schmeiser. 1991. Methods for selecting the best system. In *Proceedings of the 1991 Winter Simulation Conference*, ed. B.L. Nelson, W.D. Kelton, and G.M. Clark, 177–186. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Hashem, S., and B. Schmeiser. 1994. Algorithm 727: quantile estimation using overlapping batch statistics. *ACM Transactions on Mathematical Software* 20: 100–102.

Kang, K. 1984. *Confidence interval estimation via batch mean and time series modeling*. Doctoral dissertation, Purdue University.

Kang, K., and B. Schmeiser. 1987. Properties of batch means from stationary ARMA time series. *Operations Research Letters* 6 (1): 19–24.

Kang, K., and B. Schmeiser. 1990. Graphical methods for evaluating and comparing confidence-interval procedures. *Operations Research Letters* 38: 546–553.

Meketon, M.S., and B. Schmeiser. 1984. Overlapping batch means: something for nothing? In *Proceedings of the 1984 Winter Simulation Conference*, ed. S. Sheppard, U.W. Pooch, and C.D. Pegden, 227–230. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Nelson, B.L. 1983. *Variance reduction in simulation experiments: a mathematical-statistical framework*. Doctoral dissertation, Purdue University.

Pedrosa, A. 1994. *Automatic batching in simulation output analysis*. Doctoral dissertation, Purdue University.

Pedrosa, A., and B. Schmeiser. 1993. Asymptotic and finite-sample correlations between obm estimators. In *Proceedings of the 1993 Winter Simulation Conference*, ed. G.W. Evans, M. Mollaghasemi, E.C. Russell, and W.E. Biles, 481–488. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Schmeiser, B. 1982. Batch-size effects in the analysis of simulation output. *Operations Research* 30: 556–568.

Schmeiser, B. 1990. Simulation experiments. Chapter 7 in *Handbooks in Operations Research and Management Science, Volume 2: Stochastic Models*, ed. D.P. Heyman and M.J. Sobel, 295–330.

Schmeiser, B. 1992a. Discussion of 'Inference from iterative simulation using multiple sequences' by A. Gelman and D.B. Rubin and 'Practical Markov chain Monte Carlo' by C.J. Geyer. *Statistical Science* 7: 498-499.

Schmeiser, B. 1992b. Modern simulation environments: statistical issues. In *Proceedings of the First IE Research Conference*, 457–462. Norcross, Georgia: Institute of Industrial Engineers.

Schmeiser, B. 2001. Some myths and common errors in simulation experiments. In *Proceedings of the 2001 Winter Simulation Conference*, ed. B.A. Peters, J.S. Smith, D.J. Medeiros, and M.W. Rohrer, 39–46. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Schmeiser, B., T. Avramidis, and S. Hashem. 1990. Overlapping batch statistics. In *Proceedings of the 1990 Winter Simulation Conference*, ed. O. Balci, R.P. Sadowski, and R.E. Nance, 395–398. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Schmeiser, B., and S.J. Deutsch. 1977. Quantile estimation from grouped data: the cell midpoint. *Communications in Statistics: Simulation and Computation* B6: 221–234.

Schmeiser, B., and M.D. Scott. 1991. SERVO: simulation experiments with random-vector output. In *Proceedings of the 1991 Winter Simulation Conference*, ed. B.L. Nelson, W.D. Kelton, and G.M. Clark, 927–936. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Schmeiser, B., and W.T. Song. 1987. Correlation among estimators of the variance of the sample mean. In *Proceedings of the 1987 Winter Simulation Conference*, ed. A. Thesen, H. Grant and W.D. Kelton, 309–317. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Schmeiser, B., and W.T. Song. 1996. Batching methods in simulation output analysis: What we know and what we don't. In *Proceedings of the 1996 Winter Simulation Conference*, ed. J.M. Charnes, D.M. Morrice, D.T. Brunner and J.J. Swain, 122–127. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Schmeiser, B., and Y. Yeh. 2002. On choosing a single criterion for confidence-interval procedures. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J.L. Snowdon, and J.M. Charnes, 345–352. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Schruben, L.W. 1983. Confidence interval estimators using standardized time series. *Operations Research* 31: 1090–1108.

Scott, M.D. 1990. *A code generator for random-vector simulation experiments*. M.S. thesis, Purdue University.

Song, W.T. 1988. *On quadratic-form variance estimators of the sample mean in the analysis of simulation output*. Doctoral dissertation, Purdue University.

Song, W.T., and B. Schmeiser. 1988a. Minimal-mse linear combinations of variance estimators of the sample

mean. In *Proceedings of the 1988 Winter Simulation Conference*, ed. M.A. Abrams, P.L. Haigh, and J.C. Comfort, 414–421. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Song, W.T., and B. Schmeiser. 1988b. Estimating standard errors: empirical behavior of asymptotic mse-optimal batch sizes. In *Computing Science and Statistics: Proceedings of the* 20th *Symposium on the Interface*, ed. E.J. Wegman, D.T. Gantz, and J.J. Miller, 575–580.

Song, W.T., and B. Schmeiser. 1988c. On the dispersion matrix of estimators of the variance of the sample mean in the analysis of simulation output. *Operations Research Letters* 7: 259–266.

Song, W.T., and B. Schmeiser. 1993. Variance of the sample mean: properties and graphs of quadratic-form estimators. *Operations Research* 41: 501–517.

Song, W.T., and B. Schmeiser. 1994. Reporting the precision of simulation experiments. In *New directions in simulation for manufacturing and communications*, ed. S. Morito, H. Sakasegawa, K. Yoneda, M. Fushimi, and K. Nakano, 402–407. Operations Research Society of Japan.

Song, W.T., and B. Schmeiser. 1995. Optimal mean-squared-error batch sizes. *Management Science* 41: 110–123.

Wieland, J.R. 2003. *Developing a simulation approach for checking queueing-network stability.* M.S. thesis, Purdue University.

Wieland, J.R., R. Pasupathy, and B. Schmeiser. 2003. Queueing-network stability: simulation-based checking. In *Proceedings of the 2003 Winter Simulation Conference*, ed. S. Chick, P.J. Sánchez, D. Ferrin, and D.J. Morrice, 520–527. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Wood, D.C. 1995. *Variances in dynamic-system performance: point estimation and standard errors*. Doctoral dissertation, Purdue University.

Wood, D.C., and B. Schmeiser. 1994. Consistency of overlapping batch variances. In *Proceedings of the 1994 Winter Simulation Conference*, ed. J. Tew, S. Manivannan, D. Sadowski, and A. Seila, 316–319. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Wood, D.C., and B. Schmeiser. 1995. Overlapping batch quantiles. In *Proceedings of the 1995 Winter Simulation Conference*, ed. C. Alexopoulos, K. Kang, D. Goldsman, and W. Lilegdon, 303–308. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Yeh, Y. 1999. *Steady-state simulation output analysis via dynamic batch means.* M.S. thesis, Purdue University.

Yeh, Y. 2002. *Steady-state simulation output analysis: mse-optimal dynamic batch means with parsimonious storage*. Doctoral dissertation, Purdue University.

Yeh, Y., and B. Schmeiser. 2000. Simulation output analysis via dynamic batch means. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J.A. Joines, R.R. Barton, K. Kang, and P.A. Fishwick, 637–645. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Yeh, Y., and B. Schmeiser. 2004. On the mse robustness of batching estimators. *Operations Research Letters* 32: 293–298.

AUTHOR BIOGRAPHY

BRUCE SCHMEISER is professor of Industrial Engineering at Purdue University. His research interests center on developing methods for better simulation experiments. He is a member of INFORMS, is a Fellow of IIE, and has been active within the Winter Simulation Conference for many years, including being the 1983 Program Chair and chairing the Board of Directors from 1988–1990.