

Confidence-interval estimation using quasi-independent sequences

E. JACK CHEN¹ and W. DAVID KELTON²

¹*BASF Corporation, Rockaway, NJ 07866-0909, USA*

E-mail: e.jack.chen@basf.com

²*Department of Quantitative Analysis and Operations Management, University of Cincinnati, Cincinnati, OH 45221-0130, USA*

E-mail: david.kelton@uc.edu

Received December 7, 2006

A *quasi-independent* (QI) subsequence is a subset of time-series observations obtained by systematic sampling. Because the observations appear to be independent, as determined by the runs tests, classical statistical techniques can be used on those observations directly. This paper discusses implementation of a sequential procedure to determine the simulation run length to obtain a QI subsequence, and the batch size for constructing confidence intervals for an estimator of the steady-state mean of a stochastic process. Our QI procedure increases the simulation run length and batch size progressively until a certain number of essentially independent and identically distributed samples are obtained. The only (mild) assumption is that the correlations of the stochastic process output sequence eventually die off as the lag increases. An experimental performance evaluation demonstrates the validity of the QI procedure.

1. Introduction

When estimating the steady-state mean μ of some discrete-time stochastic output process $\{X_i : i \geq 1\}$ from a simulation, we would like an algorithm to determine the simulation run length N so that a variance estimator of the sample mean ($\bar{X}(N) = \frac{1}{N} \sum_{i=1}^N X_i$) is asymptotically unbiased (i.e., the asymptotic approximation is valid), the confidence interval (CI) is no more than a pre-specified width, and the actual coverage probability of the CI is close to the nominal coverage probability $1 - \alpha$. Because we assume the underlying distribution is stationary, i.e., the joint distribution of the X_i 's is insensitive to time shifts, the mean estimator will be unbiased; alternatively, we assume that the simulation has been “warmed up” sufficiently for any initialization bias to have dissipated. But an appropriate run length must be determined dynamically to attain the validity (i.e., coverage probability) and precision required of a CI from a simulation.

The usual method of CI construction from classical statistics, which assumes independent and identically distributed (IID) observations, is not directly applicable since simulation output data are generally correlated. Several methods of determining the simulation run length and estimating the variance of the sample mean have been proposed. In particular, Crane and Iglehart (1975) use regenerative processes; Fishman (1971) and Schriber and Andrews (1984) use time series; Priestley (1981) uses classical spectral analysis; Heidelberger and Welch (1981) and Lada and Wilson (2006) use a spectral-based analysis; Schmeiser (1982), Meketon and Schmeiser (1984), and Steiger and Wilson (1999) use batch means (BM); Schruben (1983) and Nakayama (1994) use standardized time series. For an overview of existing methods of determining the simulation run length and estimating the variance of the sample mean, see Sargent, Kang and Goldsman (1992). However, many existing techniques are either too restrictive to be applied to general cases or too complicated to be implemented for practical use. Therefore, there is a need for robust and relatively simple procedures to determine appropriate simulation run lengths, and to estimate the variance of the sample mean (average of the observations in the run), which can be applied to generic processes.

We propose *quasi-independent-variance* (QIV) procedures (see Section 2.3) in concert with the well-known batch-means method for constructing a confidence interval for the mean μ of the stationary process. The proposed procedures progressively increase the length of a simulation run so that the mean estimate satisfies a pre-specified precision requirement, and produce asymptotically valid confidence intervals. The QIV procedures use systematic sampling to obtain a subsequence of samples that appear to be independent, determined by the *runs* test (see Section 2.2). The main advantage of the QIV approach is that, by using a subsequence of quasi-independent samples to represent the underlying distribution, we can apply classical statistical techniques directly. That is, the variance of the sample mean can be estimated indirectly by dividing the sample variance of near-independent samples by the number of near-independent samples.

In the non-overlapping batch-means method, the simulation output sequence is divided into b adjacent non-overlapping batches, each of size m . For simplicity, we assume that the simulation run length N is a multiple of m so that $N = bm$. The sample mean, $\hat{\mu}_j$, for the j^{th} batch is

$$\hat{\mu}_j = \frac{1}{m} \sum_{i=m(j-1)+1}^{mj} X_i, \quad \text{for } j = 1, 2, \dots, b. \quad (1)$$

Then the grand mean $\bar{\hat{\mu}}$ of the individual BMs, given by

$$\bar{\hat{\mu}} = \frac{1}{b} \sum_{j=1}^b \hat{\mu}_j, \quad (2)$$

is used as a point estimator for μ . Here $\bar{\hat{\mu}} = \bar{X}(N)$, the sample mean of all N individual X_i 's, and we seek to construct a CI based on the point estimator (2).

The use of BM is a well-known technique for estimating the variance of point estimators computed from simulation experiments. The batch-means variance estimator is simply the sample variance of the estimator computed on means of subsets of consecutive subsamples. Asymptotic validity of BM depends on both the assumption of approximate independence of the BMs, and the assumption that the BMs are approximately normally distributed. However, some BM methods (for example, ASAP3 by Steiger et al., 2005) construct a CI for the mean by adjusting the CI half-width based on the strength of the autocorrelations between batch means. For BM methods to be practical, a good choice of the batch size is necessary. We propose a method to estimate the batch size using only observed data. In contrast to results that model the unknown underlying dependence structure in terms of a few unknown parameters, our method is completely nonparametric. The only (mild) assumption is that the autocorrelations of the stochastic process output sequence eventually die off as the lag between observations increase.

The main advantage of our approach is that by using a subsequence of quasi-independent samples to determine the batch size, we do not require complicated computation or advanced theory. Moreover, the batch size is determined automatically without any user intervention and there is no need to store the entire sequence of observations. Although our quasi-independent procedures have many desirable properties, there are also some drawbacks. First, our methods suffer from one of the problems that afflicts any sequential stopping rule: the run length of the simulation may be inappropriately short or long. The simulation may terminate inappropriately early due to statistical variability, which may cause difficulties such as the asymptotic approximation not yet being valid. However, by specifying an appropriate stopping rule, we can reduce the chance that the simulation terminates early. To avoid the simulation's running longer than necessary, though, is somewhat difficult, which arises from the fact that we double the simulation run length every other iteration.

In Section 2 we present some basis of our methodologies and the proposed procedures for mean and variance estimation: the assumption on the output correlations, using tests

of independence to determine the lag l of the systematic samples, variance estimation via QI subsequences, implementation of three procedures for constructing a CI for μ , and some discussion of the procedures. In Section 3 we show our empirical-experiment results. In Section 4 we give concluding remarks.

2. Methodologies

In this section we introduce the methodologies we used in our procedures: assumptions on the output correlations, systematic sampling, the runs test, and present our quasi-independent procedures.

2.1 Assumptions on the output correlations

For our procedures to work correctly, we assume that the autocorrelations of the stochastic process output sequence eventually die off as the lag between them increases. Such processes typically obey a Central Limit Theorem (CLT) for dependent processes of the form

$$\frac{\sqrt{N}[\bar{X}(N) - \mu]}{\sigma} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1) \text{ as } N \rightarrow \infty,$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution with mean μ and variance σ^2 , and $\xrightarrow{\mathcal{D}}$ denotes convergence in distribution. Furthermore, if $\sum_{i=1}^{\infty} i|\gamma_i| < \infty$, then the steady-state variance constant (SSVC)

$$\sigma^2 \equiv \lim_{N \rightarrow \infty} N\text{Var}[\bar{X}(N)] = \sum_{i=-\infty}^{\infty} \gamma_i.$$

If the sequence is independent, then the SSVC is equal to the single-point process variance $\sigma_x^2 = \text{Var}(X_i)$. For a finite sample N , let

$$\sigma^2(N) = \gamma_0 + 2 \sum_{i=1}^{N-1} (1 - i/N)\gamma_i.$$

It follows that $\lim_{N \rightarrow \infty} \sigma^2(N) = \sigma^2$ and $\text{Var}[\hat{\mu}_j] = \sigma^2(N)/N \approx \sigma^2/N$, provided that N is sufficiently large.

Although asymptotic results are often applicable when the amount of data is “large enough,” the point at which the asymptotic results become valid generally depends on unknown factors. An important practical decision must be made regarding the sample size N required to achieve the desired precision. Therefore, both asymptotic theory and workable

finite-sample approaches are needed by the practitioner. We propose our quasi-independent method; the procedure uses a subsequence of n samples (taken from the original output sequence of N observations) that appear to be independent to determine the batch size. We accomplish this by *systematic sampling*, i.e., select a number l , choose that observation and then every l^{th} observation thereafter. Here l will be chosen sufficiently large so that the observations selected are essentially independent. This is possible because we assume that the autocorrelations of the output sequence eventually die off with increasing lag.

We compute the size l for our systematic sampling based on the runs tests (see Section 2.2) and choose the number of required independent observations $n = 4000$, the minimum recommended sample size for the runs tests (Knuth, 1998). The simulation run length is then $N = nl$. Let b be the number of batches (see Section 2.4 for how b is determined in our procedure); then the batch size is $m = N/b$. Here b will be chosen so that m will be an integer.

2.2 Test of independence

Because the required run lengths are drastically different between IID and correlated sequences, it is beneficial to check whether the input data appear to be independent. We use the *runs-up* and *runs-down* tests for this purpose; see Knuth (1998). Briefly, a run up (run down) is a monotonically increasing (decreasing) subsequence. We compare the proportion of different run lengths to what would be expected under independence in order to check whether the sequence appears to be independent. If the output data sequence appears to be dependent, then a sequential procedure will be used. The runs tests are used in our procedure to determine the lag length l so that observations that are at least $l - 1$ observations apart will be essentially independent.

The runs tests of independence require the sequences to be absolutely continuous. To adapt for discontinuous points, we will increase the run length with probability $1/(\tau+1)$ when two successive observations are equal, where τ is the current run length. If the distribution is continuous, the probability of run length τ is $\tau/(\tau + 1)!$, under the null hypothesis that the sequence consists of IID random variables; see the Appendix for a proof. Moreover, if there are $\tau + 1$ observations, the possibility of the $(\tau + 1)^{\text{th}}$ observation being the largest is $1/(\tau + 1)$. For example, if $X_i = X_{i+1}$ in a runs-up test and the run length τ up to X_i is 2, we will generate a uniform (0,1) random variable U ; if $U < 1/3$, we will let $X_i \leq X_{i+1}$

Table 1: Lag l for correlated sequence to appear independent

MA1(θ)			AR1(φ)			MM1(ρ)		
θ	avg	std	φ	avg	std	ρ	avg	std
0.50	2.11	0.36	0.50	4.13	1.25	0.50	7.18	21.6
0.75	2.12	0.38	0.75	10.2	3.01	0.75	35.1	12.6
0.90	2.12	0.37	0.90	27.5	7.38	0.90	295	106

and increase the run length by one, otherwise we will let $X_i \geq X_{i+1}$ and start a new run-up sequence. Our preliminary experimental results show that this adjustment works well even for discrete distributions.

We use both the runs-up and runs-down tests simultaneously. Thus, if the probability of an independent sequence passing the runs-up test is p_1 and passing the runs-down test is p_2 , the probability of an independent sequence passing the combined runs test is $p = p_1 p_2$. We carry out some experiments with the runs tests to get an idea of the relationship between the lag l and the degree of correlation. We tested three stochastic processes: the MA1(θ), AR1(φ), and the MM1(ρ) processes; see Section 3 for definitions of the AR1(φ) and MM1(ρ) processes. The MA1(θ) is the steady-state *first-order moving average* process, generated by the recurrence

$$X_i = \mu + \epsilon_i + \theta \epsilon_{i-1} \quad \text{for } i = 1, 2, \dots,$$

where ϵ_i are IID $\mathcal{N}(0, 1)$. We set μ to 2 in our experiments. In these experiments, the sample sizes are doubled every two iterations. The steps between generating new observations is considered as an iteration; see Section 2.4. We use the A and B sub-iteration notation to simplify the description of sample size. We use a 95% confidence level for both the runs-up and runs-down tests. Consequently, the procedure will commit a Type I error (i.e., independence sequences do not pass the tests) about 10% (i.e., $1 - 0.95^2$) of the time.

Table 1 lists the experimental results. The *avg* column lists the average lag at which the output sequence first appears to be independent, as determined by the runs tests. The *std* column lists the standard deviation of the lag of the output sequence where it first appears to be independent, as determined by the runs tests. For example, the average lag at which the AR1(0.5) process first appears to be independent is 4.13. In general, the lag l for a subsequence to appear independent is positively correlated with the degree of correlation of the sequence. See Chen and Kelton (2003) for more experimental results of the runs tests.

Table 2: Properties of the QI subsequence at each iteration

Iteration	0	1 _A	1 _B	2 _A	2 _B	3 _A	3 _B	...	$k_A(k > 0)$	$k_B(k > 0)$
Total Observations	n	$2n$	$3n$	$4n$	$6n$	$8n$	$12n$...	$2^k n$	$2^{k-1} 3n$
Samples in Buffer	n	$2n$	$3n$	$2n$	$3n$	$2n$	$3n$...	$2n$	$3n$
l_0	2^0	2^0	2^0	2^1	2^1	2^2	2^2	...	2^{k-1}	2^{k-1}
l'	1	2	3	2	3	2	3	...	2	3
l	$2^0 1$	$2^0 2$	$2^0 3$	$2^1 2$	$2^1 3$	$2^2 2$	$2^2 3$...	$2^{k-1} 2$	$2^{k-1} 3$

To eliminate the need to store the entire observations or the need to read in the observations repeatedly, we allocate a buffer, QI, with size $t_1 = 3n$ to store our systematic samples y_i , $1 \leq i \leq t_1$ that will be used by the runs tests of independence. Note that lag l' ($=1,2,3$) of the systematic samples is used to refer to systematic samples $y_{kl'+1}$, for $k = 0, 1, 2, \dots, n-1$. Table 2 shows the total number of observations and other properties at each iteration. The *Total Observations* row shows the total number of observations at a certain iteration. The *Samples in Buffer* row shows the number of systematic samples stored in the buffer. The l_0 row shows the lag used to obtain the systematic samples stored in the buffer. The l' row shows the lag of the systematic samples that will be used for the runs tests of independence. The l row shows the lag l at which the sequence appears to be independent if the lag- l' systematic samples passed the test of independence at that iteration.

2.3 Quasi-independent-variance estimator

We propose two simple *quasi-independent-variance* (QIV) algorithms to estimate the mean and variance. The aim of the QIV methods is to obtain approximately independent random samples by skipping highly correlated observations. Note that we assume that the autocorrelations eventually die off as the lag between them becomes large.

2.3.1 Quasi-independent-variance method one

Recall that the sample mean of all observations is

$$\bar{X}(N) = \frac{1}{N} \sum_{i=1}^N X_i. \tag{3}$$

where $N = nl$ is the simulation run length and X_i is the value of the i^{th} observation. There are $n' = nl'$ samples in the QI buffer. Recall that l' could be 1, 2, or 3. Let Y_i be the value

of the i^{th} sample in the QI buffer, i.e., the samples obtained by systematic sampling. The sample mean of the QI samples is

$$\hat{\mu}_Y = \frac{1}{n'} \sum_{i=1}^{n'} Y_i.$$

Because the observations are sampled from the steady-state distribution of a stationary process, both $\bar{X}(N)$ and $\hat{\mu}_Y$ are unbiased estimators of the unknown true mean μ ; i.e., theoretically $\hat{\mu}_Y$ will in expectation be equal to the expected mean of all observations $\bar{X}(N)$. Our experimental results confirm this. However, $\bar{X}(N)$ is a better (less variable) estimator than $\hat{\mu}_Y$ because it is computed from more observations.

The variance of each Y_i can be estimated by the sample variance

$$\widehat{\text{Var}}(Y) = \frac{1}{n' - 1} \sum_{i=1}^{n'} (Y_i - \hat{\mu}_Y)^2.$$

Because the $Y_{kl'+1}$'s appear to be independent, the variance estimator of the sample mean $\hat{\mu}_Y$ can be computed by

$$S_1^2 = \widehat{\text{Var}}(\hat{\mu}_Y) = \widehat{\text{Var}}(Y)/n;$$

n is used (instead of n') because there are only n statistically independent samples. By the CLT

$$\frac{\hat{\mu}_Y - \mu}{S_1} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1) \text{ as } n \rightarrow \infty.$$

This would then lead to the $100(1 - \alpha)\%$ CI for μ

$$\hat{\mu}_Y \pm z_{1-\alpha/2} S_1,$$

where $z_{1-\alpha/2}$ is the $1 - \alpha/2$ quantile of the standard normal distribution. Since $\bar{X}(N)$ is readily available and is at least as good as $\hat{\mu}_Y$ as an estimator of μ , we construct the CI as

$$\bar{X}(N) \pm z_{1-\alpha/2} S_1. \tag{4}$$

The first method estimates the variance of the sample mean indirectly from the variance of the individual samples of the QI subsequence, i.e., $\widehat{\text{Var}}(\hat{\mu}_Y) = \widehat{\text{Var}}(Y)/n$. The second method estimates the variance of the sample mean directly from several mean estimators, i.e., the $\hat{\mu}_Y$'s.

The QIV1 algorithm

1. Remark: itmax is the maximum number of iterations the algorithm will try, t_1 is the size of the QI buffer, l_0 is the lag between observations used to obtain systematic samples in the QI buffer, δ is the incremental sample size, and k is the index of iterations. Each iteration k contains two sub-iterations k_A and k_B .
2. Initialization: set $n = 4000$, $\text{itmax} = 10$, $t_1 = 3n$, $l_0 = 1$, $\delta = n$, and $k = 0$.
3. Generate δ observations and store lag l_0 observations in the latter portion of the QI buffer.
4. If this is the initial iteration, set $l' = 1$. If this is a k_A iteration, set $l' = 2$. If this is a k_B iteration, set $l' = 3$. Carry out the runs tests to determine whether lag l' systematic samples appear to be independent.
5. If the lag l' systematic samples appear to be independent, go to step 10.
6. If the current iteration is the initial or k_B iterations, set $k = k + 1$. If $k = 1$, go to step 3. If $k > \text{itmax}$, then go to step 10; else start a k_A iteration.
7. If this is a k_A iteration ($k > 1$), then discard every other observation in the QI buffer, and reindex the rest of the $t_1/2$ observations in the first half of the buffer. Set $l_0 = 2^{k-1}$ and $\delta = nl_0/2$.
8. If this is a k_B iteration ($k > 1$), set $\delta = nl_0$.
9. Go to step 3.
10. Compute the mean estimator according to (3).
11. Compute the confidence interval of the mean estimator according to (4).
12. Let ϵ be the desired absolute half width, and let $r|\bar{X}(N)|$ be the desired relative half width. If the CI half-width $H_1 = z_{1-\alpha/2}S_1$ is less than ϵ or $r|\bar{X}(N)|$, then terminate the algorithm.
13. Increase the required independent samples n to $(H_1/\epsilon)^2n$ or $(H_1/(r\bar{X}(N)))^2n$. Go to step 10. Note that we keep the sample size for the runs test at 4000.

Even though QIV1 can also be considered as a special case of spaced batch means (Fox, Goldsman, and Swain, 1991), with batch size of 1 and spacer size $s = l_0$, the underlying spirit is completely different. We aim to obtain independent samples, not the independent mean estimators. As mentioned previously, the average of all observations and the average of observations in the buffer are equal in expectation.

2.3.2 Quasi-independent-variance method two

The second method of estimating the variance of the sample mean is computed directly from several mean estimators. We divide the retained QI subsequence $\{y_i\}$ into b' consecutive batches with batch size m' . Here b' will be chosen such that $b' = n'/m'$ is an integer. The initial value of b' is 100. Let $\hat{\mu}_j = \frac{1}{m'} \sum_{i=(j-1)m'+1}^{jm'} Y_i$ for $j = 1, 2, \dots, b'$ be the mean estimator from the j^{th} batch. Recall that Y_i is the i^{th} systematic sample.

Note that

$$\bar{\hat{\mu}}_q = \frac{1}{b'} \sum_{j=1}^{b'} \hat{\mu}_j = \hat{\mu}_Y$$

is an unbiased estimator of μ . Because each $\hat{\mu}_j$ has a limiting normal distribution, by the CLT a CI for μ using the IID $\hat{\mu}_j$'s can be approximated by using standard statistical procedures. That is, the ratio

$$T = \frac{\bar{\hat{\mu}}_q - \mu}{S_2/\sqrt{b'}}$$

would have an approximate t distribution with $b' - 1$ DF (degrees of freedom), where

$$S_2^2 = \frac{1}{b' - 1} \sum_{j=1}^{b'} (\hat{\mu}_j - \bar{\hat{\mu}}_q)^2$$

is the usual unbiased estimator of the variance of μ . This would then lead to the $100(1 - \alpha)\%$ CI for μ

$$\bar{\hat{\mu}}_q \pm t_{b'-1, 1-\alpha/2} \frac{S_2}{\sqrt{b'}},$$

where $t_{b'-1, 1-\alpha/2}$ is the $1 - \alpha/2$ quantile for the t distribution with $b' - 1$ DF ($b' \geq 2$). Again, we substitute $\bar{\hat{\mu}}_q$ with $\bar{X}(N)$ to construct CI for μ

$$\bar{X}(N) \pm t_{b'-1, 1-\alpha/2} \frac{S_2}{\sqrt{b'}}. \quad (5)$$

The QIV2 algorithm

The first ten steps are the same as algorithm one. Steps 11, 12, and 13 will be replaced by the following three steps:

- 11'. Compute the confidence interval of the mean estimator according to (5).
- 12'. If the CI half-width $H_2 = t_{b'-1, 1-\alpha/2} S_2 / \sqrt{b'}$ is less than ϵ or $r|\bar{X}(N)|$, then terminate the algorithm.
- 13'. Increase the number of batches b' to $(H_2/\epsilon)^2 b'$ or $(H_2/(r\bar{X}(N)))^2 b'$ (with batch size m'). Go to step 10.

These procedures must store the entire quasi-independent subsequence, which is needed for the runs tests. The QIV procedures skip more observations as the autocorrelations become stronger. But an important part of simulation experiments is sample generation. We should use samples that can appropriately represent the whole population, instead of samples that are conveniently available to us. For instance, it is necessary to delete the transient-state observations to obtain a good estimate of steady-state characteristics since those early transient data often cause more harm than good—they contribute more bias than information. The same can be said regarding the observations we discard (at least in the current context), especially when the experimental results show that $\hat{\mu}_Y$ will be equal in expectation to the mean of all observations $\bar{X}(N)$.

2.4 Quasi-independent batch means

One possible criticism of QIV is that these procedures skip more samples as the autocorrelations become stronger. To ensure that all information is processed, we propose a simple QI algorithm to determine the simulation run length and batch size for the batch-means method. The aim of the QI method is to continue the simulation run until we have obtained a pre-specified number of essentially independent random samples by skipping highly correlated observations.

Note that all observations are used to compute the BM. We discard samples in the QI sequence so that the size of the QI sequence will be no more than the buffer size t_1 . Samples in the QI sequence are used by the runs tests to determine batch sizes and are not used to compute the BM. We assume that the batch size determined by our algorithm is sufficiently large so that the BM $\{\hat{\mu}_j : 1 \leq j \leq b\}$, computed by (1), are approximately IID normal.

Figure 1 displays a high-level flow chart of QI batch-means method. We allocate a buffer, PB, with size $t_2 = 3b$ to keep primitive batches (PB), which will be used to compute BM. The PB are the sample means of p_b consecutive observations. The initial value of p_b is set to $t = t_1/t_2$. Table 3 shows the total number of observations and the number of observations

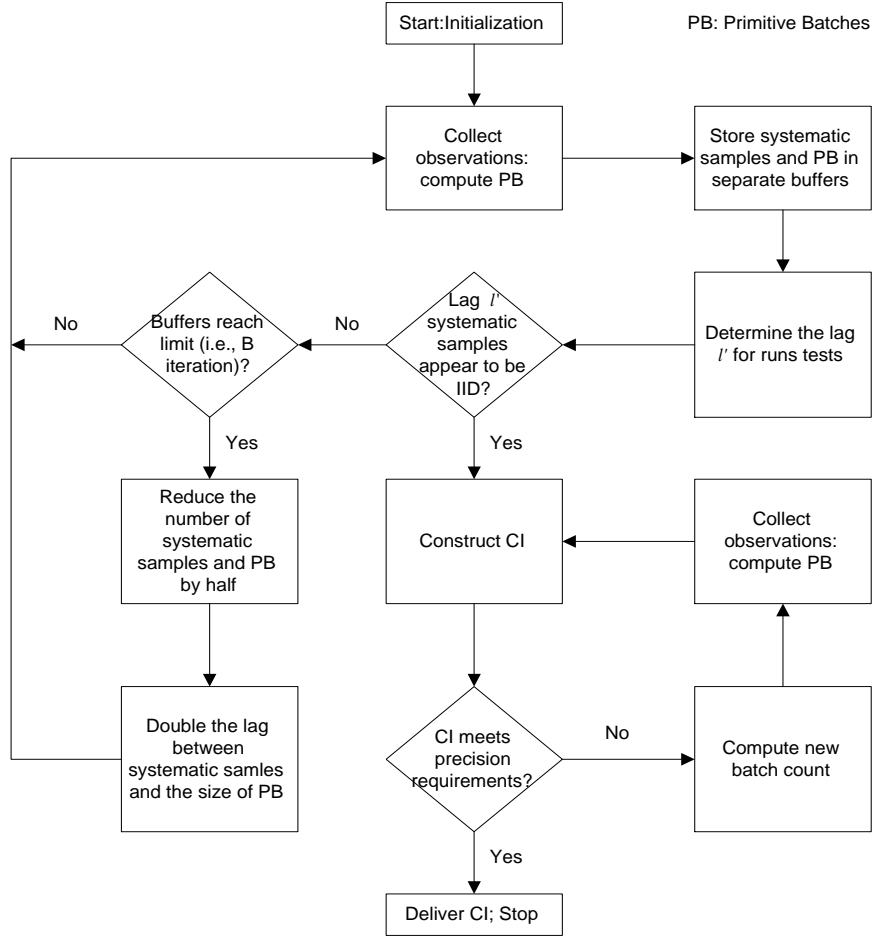


Figure 1: High-level flow chart of QI batch means

Table 3: Batch size at each iteration

Iteration	0	1_A	1_B	2_A	2_B	3_A	3_B	...	$k_A(k > 0)$	$k_B(k > 0)$
PB	b	$2b$	$3b$	$2b$	$3b$	$2b$	$3b$...	$2b$	$3b$
p_b	t	t	t	2^1t	2^1t	2^2t	2^2t	...	$2^{k-1}t$	$2^{k-1}t$
m	t	$2t$	$3t$	2^12t	2^13t	2^22t	2^23t	...	$2^{k-1}2t$	$2^{k-1}3t$

used to compute PB and BM at each iteration. The *Iteration* row shows the index of the iteration. The *PB* row shows the number of PB obtained. The p_b row shows the number of observations used to obtain the PB stored in the buffer. The m row shows the batch size m if the procedure stopped at that iteration.

This method of estimating the variance of the mean estimator $\bar{X}(N)$ is obtained directly through multiple mean estimators. Because $\hat{\mu}_j$ as defined in (1) has a limiting normal distribution, a CI for μ using the approximately IID $\hat{\mu}_j$'s can be approximated by the CLT using standard statistical procedures. That is, if there are b batches, the point estimate is computed by (2) and the ratio

$$T = \frac{\bar{\hat{\mu}} - \mu}{S/\sqrt{b}}$$

would have an approximate t distribution with $b - 1$ DF, where

$$S^2 = \frac{1}{(b-1)} \sum_{j=1}^b (\hat{\mu}_j - \bar{\hat{\mu}})^2$$

is the usual unbiased estimator of the variance of μ . This would then lead to the $100(1-\alpha)\%$ CI for μ ,

$$\bar{\hat{\mu}} \pm t_{b-1, 1-\alpha/2} \frac{S}{\sqrt{b}}. \quad (6)$$

This CI is approximately valid when the batch size m becomes large because the BMs $\hat{\mu}_1, \hat{\mu}_2, \dots, \hat{\mu}_b$ become essentially independent (since the underlying process satisfies the condition that autocorrelations eventually die off with increasing lag) and almost normally distributed (from an appropriate CLT for such sequences).

The quasi-independent batch-means algorithm

1. Remark: itmax is the maximum number of iterations the algorithm will try, t_1 is the size of the QI buffer, t_2 is the size of the PB buffer, b is the initial number of batches, l_0 is the lag of observations used to obtain systematic samples in the QI buffer, δ is the incremental sample size, and k is the index of iterations. Each iteration k contains two sub-iterations k_A and k_B .
2. Initialization: set $n = 4000$, $\text{itmax} = 10$, $t_1 = 3n$, $b = 100$, $t_2 = 3b$, $l_0 = 1$, $\delta = n$, and $k = 0$.
3. Generate δ observations and store lag l_0 observations in the QI buffer. Compute PB and store the values in the PB buffer.

4. If this is the initial iteration, set $l' = 1$. If this is a k_A iteration, set $l' = 2$. If this is a k_B iteration, set $l' = 3$. Carry out the runs tests to determine whether lag l' systematic samples appear to be independent.
5. If the lag l' systematic samples appear to be independent, go to step 10.
6. If the current iteration is the initial or k_B iterations, set $k = k + 1$. If $k = 1$, go to step 3. If $k > \text{itmax}$, then go to step 10; else start a k_A iteration.
7. If this is a k_A iteration ($k > 1$), then discard every other observation in the QI buffer, and reindex the rest of the $t_1/2$ observations in the first half of the buffer. Reduce the number of PB from t_2 to $t_2/2$ by taking the average of two adjacent PB. Set $l_0 = 2^{k-1}$ and $\delta = nl_0/2$.
8. If this is a k_B iteration ($k > 1$), set $\delta = nl_0$.
9. Go to step 3.
10. Compute the mean point estimator according to (2).
11. Compute the confidence interval of the mean estimator according to (6).
12. If the CI half-width H is less than ϵ or $r|\bar{\mu}|$, terminate the algorithm.
13. Increase the number of batches b to $(H/\epsilon)^2b$ or $(H/r|\bar{\mu}|)^2b$ (with batch size m). Go to step 10.

Our QI procedure addresses the problem of determining the batch size m , and the number of batches b , that are required to satisfy the assumptions of independence and normality. Theoretically, if these assumptions are satisfied, then the actual coverage of the CIs should be close to the pre-specified level. The QI procedure must store the entire quasi-independent sequence, which is needed for the runs tests. As a rule of thumb, the means of 30 to 40 independent samples will be approximately normal; hence, we set the initial number of batches $b = 100$.

Let the half-width be

$$H = t_{b-1, 1-\alpha/2} \frac{S}{\sqrt{b}}.$$

The final step in the QI procedure is to determine whether the CI meets the user’s requirement for precision, a maximum absolute half-width ϵ , or a maximum relative fraction r of the magnitude of the final grand mean $\bar{\mu}$. If the relevant requirement, $H \leq \epsilon$ or $H \leq r|\bar{\mu}|$ for the precision of the confidence interval is satisfied, then the QI procedure terminates: return the sample mean $\bar{\mu}$ and the CI with half-width H . If the precision requirement is not satisfied with b batches, then the QI procedure will increase the number of batches to $(H/\epsilon)^2b$ or $(H/r|\bar{\mu}|)^2b$. This step can be repeated iteratively until the pre-specified precision is achieved. This procedure is denoted QIBatch in the remainder of the paper.

2.5 Properties of estimators

Goldsman and Schmeiser (1997) list some properties that a good estimator should possess. We use these properties to assess the desirability of our algorithm. The following describes the performance of our algorithm under each property.

- Statistical performance. The batch-means estimator is asymptotically unbiased. Based on our experiments, the asymptotic approximation is valid when the batch size is determined by our QI algorithm.
- Ease of computation. Our algorithm involves only a little more than $O(N)$ operations. The runs tests are relatively inexpensive computationally.
- Parsimonious storage requirements. Our data storage is 12000 for the quasi-independent sequence and another 300 for the primitive batches if the half-width-reduction phase is not required. We only need to process each observation once and do not require storing the entire sequence of N observations.
- Ease of understanding. Our algorithm requires relatively simple statistics: autocorrelations that eventually die off with increasing lag, systematic sampling, the runs tests, and the CLT.
- Numerical stability. The limits of machine precision are the limits of our algorithm precision.
- User-specified parameters. We require only two parameters: the confidence level α and either the absolute precision ϵ or the relative precision r . We do not require the user to enter the number of batches or the batch size.

- Amenability for use in algorithms. Our algorithm can easily be incorporated with other procedures.

3. Empirical experiments

In this section we present empirical results obtained from simulations using the quasi-independent procedures proposed in this paper. The purpose of the experiments was to demonstrate the interdependence between the correlation of simulation output sequences and simulation run lengths, and the validity of our methods.

3.1 Performance measures

A stochastic model that has an appropriate covariance structure and admits an exact analysis of performance criteria is the *first-order auto-regressive* (AR1(φ)) process, generated by the recurrence relation

$$X_i = \mu + \varphi(X_{i-1} - \mu) + \epsilon_i \text{ for } i = 1, 2, \dots,$$

where

$$E(\epsilon_i) = 0, \quad E(\epsilon_i \epsilon_j) = \begin{cases} \sigma^2 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases},$$

$$-1 < \varphi < 1,$$

and X_0 is specified as a random variate drawn from the steady-state distribution. The ϵ_i 's are commonly called *error terms*. The AR1(φ) process shares many characteristics observed in simulation output processes, including first- and second-order nonstationarity and autocorrelations that decline exponentially with increasing lag. Here, the ϵ_i 's are IID $\mathcal{N}(0, 1)$. It can be shown that X_∞ has a $\mathcal{N}(\mu, \frac{1}{1-\varphi^2})$ distribution, and the SSVC of the AR1(φ) process is $1/(1-\varphi)^2$. We set $\mu = 2$.

The parameter φ is set to 0.50, 0.75, and 0.90, respectively. Each design point is based on 10,000 independent replications. The half-width reduction phase is skipped in these experiments since the QI procedures deliver a tight CI by default. The results of our mean estimations of the AR1(φ) process are summarized in Tables 4 and 5. The φ column lists this parameter of the underlying AR1(φ) process. The $\hat{\mu}$ column lists the grand sample mean, i.e., the average of these 10,000 mean estimators. The *avg samp* column lists the average of the simulation run lengths, i.e., sample sizes, determined by the QI algorithm. The *std samp* column lists the standard deviation of the simulation run length. The *avg rp*

Table 4: Sample size and relative precision for the AR1(φ) process

φ	$\hat{\mu}$	avg samp	std samp	avg rp	std rp
0.50	2.000019	16520	5014	0.0064	0.0049
0.75	2.000133	40700	12043	0.0082	0.0063
0.90	2.000248	109923	29518	0.0122	0.0094

Table 5: Coverage of confidence estimators for the AR1(φ) process

	QIV1	QIV2	QIBatch
$\varphi = 0.50$			
avg hw	0.0300	0.0340	0.0264
std hw	0.0003	0.0055	0.0038
coverage	0.9336	0.9541	0.8869
$\varphi = 0.75$			
avg hw	0.0393	0.0470	0.0337
std hw	0.0004	0.0083	0.0051
coverage	0.9414	0.9744	0.8903
$\varphi = 0.90$			
avg hw	0.0597	0.0691	0.0508
std hw	0.0006	0.0090	0.0067
coverage	0.9451	0.9716	0.8895

column lists the average of the observed relative precision of the estimator, i.e., the average of $(\bar{X}(N) - \mu)/\mu$. The *std rp* column lists the standard deviation of the relative precision. The *avg hw* row lists the average of the CI half widths of the mean estimators. The *std hw* row lists the standard deviation of the CI half width. The *coverage* row lists the proportion of the CIs that cover the true mean. The *QIV1*, *QIV2*, and *QIBatch* columns list the results for the respective methods.

That the simulation run length increases as φ increases indicates that the QI algorithm works well in determining the required sample size. Except for QIBatch, the obtained coverages are much higher than the specified nominal value of 0.9. The average half width of QIBatch is the smallest among these methods, so its coverage is the smallest, just below the nominal value. The QIV1 method estimates the variance of the sample means indirectly through the variance of individual samples and has smaller standard deviation of the half width.

Table 6: Sample size and relative precision for the mean waiting-time of an M/M/1 delay in queue

ρ	μ	$\hat{\mu}$	avg samp	std samp	avg rp	std rp
0.50	1.0	0.998308	28705	86225	0.0294	0.0236
0.75	3.0	2.997541	140520	50424	0.0202	0.0160
0.90	9.0	8.988886	1180442	422519	0.0162	0.0126

Another stochastic model that we tested is the steady-state waiting time of the M/M/1 delay in queue. Waiting times in queueing systems are usually positively correlated and sometimes strongly so. Furthermore, the skewness of the exponential distribution causes exponential-servers queuing models to yield output data that might be considered “worst case.” Some feel that if an output-analytic method works well for an exponential-servers model, it is likely to work well in practice. We tested three M/M/1 queuing models. The service rate (ν) is set to 1.0 per period for all models. The arrival rate (λ) is set to 0.50, 0.75, and 0.90 per period, respectively. We denote the M/M/1 delay in queue process by $MM1(\rho)$, where $\rho = \lambda/\nu$ is the traffic intensity.

Let $\{A_n\}$ denote the IID interarrival-time sequence and $\{S_n\}$ denote the IID service-time sequence. Then the waiting-time sequence $\{W_n\}$ is defined by Lindley’s formula (Gross and Harris, 1998)

$$W_{n+1} = (W_n + S_n - A_{n+1})^+ \text{ for } n \geq 1,$$

where $w^+ = \max(w, 0)$. In order to eliminate the initial bias, W_1 is set to a random variate drawn from the steady-state distribution.

Tables 6 and 7 list the experimental results of the waiting-time of the M/M/1 delay in queue. All columns are as defined previously. The observed coverages of QIV1 are less than the nominal value of 0.9, especially when $\rho = 0.5$. We believe this is because the waiting time of the M/M/1 queuing process with $\rho = 0.50$ has a (large) jump at the discontinuous point 0 and the distribution is extremely skewed. On the other hand, QIV2 has high coverage with all three design points. However, the half width of QIV2 is much larger than for QIV1. The observed coverages of QIBatch are close to the specified nominal value of 0.9, indicating the effectiveness of the algorithm. If the half-width is longer than desired, we can increase the simulation run length until the required precision is achieved. As the simulation run length increases, we increase the number of batches with the same batch size.

Table 7: Coverage of confidence estimators for the mean waiting-time of an M/M/1 delay in queue

	QIV1	QIV2	QIBatch
$\rho = 0.50$			
avg hw	0.0450	0.0760	0.0587
std hw	0.0025	0.0181	0.0137
coverage	0.7778	0.9480	0.8790
$\rho = 0.75$			
avg hw	0.1006	0.1641	0.1233
std hw	0.0047	0.0337	0.0237
coverage	0.8174	0.9610	0.8796
$\rho = 0.90$			
avg hw	0.2583	0.3947	0.2945
std hw	0.0109	0.0797	0.0559
coverage	0.8410	0.9600	0.8929

Table 8: Coverage of 90% confidence intervals from ASAP, ASAP2, ASAP3, and WASSP

Procedure	ASAP	ASAP2	ASAP3	WASSP
avg samp	815755	281022	287568	388000
coverage	0.940	0.920	0.895	0.904
avg rp	0.046	0.070	0.070	-
avg hw	0.413	0.628	0.627	0.587
std hw	0.134	0.045	0.045	0.085

3.2 Performance comparisons

There are many well-known BM procedures, such as ABatch and LBatch (Fishman and Yarberrry, 1997, Fishman, 1998), ASAP (Automated Simulation Analysis Procedure, Steiger and Wilson, 1999), ASAP2 (Steiger et al., 2002), ASAP3 (Steiger et al., 2005), and WASSP (Lada and Wilson, 2006). Instead of working in the time domain with the original output process, WASSP works in the frequency domain by exploiting a spectral-analysis approach. The WASSP transforms the covariance function to a power spectrum, which is then estimated by a periodogram.

We compare QIBatch with other automated procedures (i.e., no user intervention is required during the execution of the procedure). Table 8 lists the experimental results of ASAP, ASAP2, ASAP3, and WASSP for the M/M/1 process with traffic intensity 0.9 with

the required relative precision set to 7.5%. These results are extracted directly from the corresponding papers cited above. Hence, the underlying random number streams are different. Since these procedures terminate when they detect normality among batch means and deliver a correlation-adjusted CI half width, they are able to provide valid CIs with relatively small sample sizes. Note that ASAP, ASAP2, ASAP3, and WASSP require storing the entire sequence of observations and reading in the observations repeatedly. On the other hand, QIBatch generally requires larger sample sizes and delivers tighter CIs by default because it terminates only after it has obtained large enough samples to obtain pre-determined systematic samples that appear to be independent. Note that the observed default relative precisions from QIBatch are no more 3%. Hence, QI procedures likely do not need to invoke the half-width-reduction phase, unless the required relative precision is less than 3%. We don't think this is a major drawback since wide half widths provide little useful information. Furthermore, the QI subsequence can provide insight about the underlying steady-state distribution. Moreover, batch means generated from QIBatch can be used as input of simulation procedures that require IID normal data.

The estimated required sample size for ASAP, ASAP2, ASAP3, and WASSP to obtain the average CI half widths of the MM1(0.9) process to be within 0.2945 are approximately 1604314 (i.e., $(0.413/0.2945)^2 815755$), 1277877, 1304730, and 1539378, which are larger than the average sample size 1180442 used by QIBatch to obtain the average CI half widths of 0.2945. In this regard, QIBatch is efficient in terms of sample size.

4. Concluding remarks

We have presented three confidence-interval algorithms for the mean μ of a stationary process. The QIV1 and QIV2 methods estimate the variance of the mean with the QI subsequence, which are obtained through systematic sampling. On the other hand, QIBatch uses all the observations to construct a CI and uses the QI subsequence to determine the simulation run length and batch size. Some CIs require more observations than others before the asymptotics necessary for CIs become valid. Our proposed quasi-independent algorithm works well in determining the required simulation run length, and the batch size for the asymptotic approximation to be valid. Although it is heuristic, the QI procedure has a strong theoretical basis. The results from our empirical experiments show that the QI procedure is excellent in achieving the intended coverage, not only for mildly correlated

processes but also for highly correlated processes. The QI procedure does not require extensive computation; therefore, it is able to estimate highly correlated processes with good precision in a reasonable amount of run time. However, the variance of the simulation run length from our sequential procedure is large. This is not only because of randomness of the output sequence but also because we double the lag length l every two iterations. Further research would then be to develop new algorithms so that the simulation run length does not need to be doubled every other iteration.

Our proposed quasi-independent algorithm requires storing a sequence of QI observations that most likely represent the underlying distribution. This sequence is used by the runs tests to check for independence. Moreover, our procedures only needs to process each observation once and does not require storing the entire sequence of observations. Our procedure is completely automatic and has the desirable properties that it is a sequential procedure and does not require the user to have *a priori* knowledge of values that the data might assume. This allows the user to apply this method without having to make a separate pilot run to determine the range of values to be expected, or guess and risk having to re-run the simulation, either of which represents potentially large costs because many realistic simulations are time-consuming to run. The main advantage of our approach is that by using a straightforward runs tests to determine the simulation run length and the batch size, we do not require more advanced statistical theory, thus making it easy to understand, simple to implement, and fast to run. The simplicity of this method should make it attractive to simulation practitioners and software developers. Moreover, the QI simulation run-length-determination mechanism can also be used when estimating other parameters of the simulation output sequence. Chen and Kelton (2006) use this QI algorithm to estimate quantiles.

ACKNOWLEDGEMENTS

The authors thank the anonymous referees for their detailed and helpful comments, which improved the quality of the paper, as well as the Editor, Dr. David Goldsman, for his comments as well. Dr. Marty Levy also provided valuable input to our proof of Proposition 1.

References

- Chen, E.J. and Kelton, W.D. (2003) Determining simulation run length with the runs test. *Simulation Modelling Practice and Theory*, **11**, 237–250.
- Chen, E.J. and Kelton, W.D. (2006) Estimating steady-state distributions via simulation-generated histograms. *Computer and Operations Research*. To Appear.
- Crane, M.A. and Iglehart, D.L. (1975) Simulating stable stochastic systems, III: regenerative processes and discrete-event simulations. *Operations Research*, **23**, 33–45.
- Fishman, G.S. (1971) Estimating sample size in computing simulation. *Management Science*, **18**, 21–38.
- Fishman, G.S. (1998) LABATCH.2: Software for statistical analysis of simulation sample path data. In *Proceedings of the 1998 Winter Simulation Conference*, pp. 131-139.
- Fishman, G.S. and Yarberr, L.S. (1997) An implementation of the batch means method. *INFORMS Journal on Computing* **9** (3), 296-310.
- Fox, B.L., Goldsman, D. and Swain, J.J. (1991) Spaced batch means. *Operations Research Letters*, **10**, 255–263.
- Goldsman, D. and Schmeiser, B.W. (1997) Computational efficiency of batching methods. In *Proceedings of the 1997 Winter Simulation Conference*, pp. 202–207.
- Gross, D. and Harris, C.M. (1998) *Fundamentals of Queueing Theory*. 3rd ed., John Wiley & Sons, New York.
- Heidelberger, P. and Welch, P.D. (1981) A spectral method for confidence interval generation and run length control in simulation. *Communications of the ACM*, **24**, 233–245.
- Knuth, D.E. (1998) *The Art of Computer Programming*, Vol. 2., 3rd ed., Addison-Wesley, Reading, MA.
- Lada, E.K. and Wilson, J.R. (2006) A wavelet-based spectral procedure for steady-state simulation analysis. *European Journal of Operational Researches*. To Appear.
- Meketon, M.S. and Schmeiser, B.W. (1984) Overlapping batch means: something for nothing? In *Proceedings of the 1984 Winter Simulation Conference*, pp. 227–230.
- Nakayama, M.K. (1994) Two-stage stopping procedures based on standardized time series. *Management Science*, **40**, 1189–1206.
- Priestley, M.B. (1981) *Spectral Analysis and Time Series*, Academic Press, New York.
- Sargent, R.G., Kang, K. and Goldsman, D. (1992) An investigation of finite-sample behavior of confidence interval estimators. *Operations Research*, **40**, 898–913.

- Schmeiser, B.W. (1982) Batch-size effects in the analysis of simulation output. *Operations Research*, **30**, 556–568.
- Schriber, T.J. and Andrews, R.W. (1984) ARMA-based confidence interval procedures for simulation output analysis. *Amer. J. Math. and Management Science*, **4**, 345–373.
- Schruben, L.W. (1983) Confidence interval estimation using standardized times series. *Operations Research*, **31**, 1090–1108.
- Steiger, N.M., Lada, E.K., Wilson, J.R., Alexopoulos, C., Goldsman, D., and Zouaoui, F. (2002) ASAP2: An improved batch means procedure for simulation output analysis. In *Proceedings of the 2002 Winter Simulation Conference*, pp. 36–344.
- Steiger, N.M., Lada, E.K., Wilson, J.R., Joines, J.A., Alexopoulos, C., and Goldsman, D. (2005) ASAP3: A batch means procedure for steady-state simulation output analysis. *ACM Transactions on Modeling and Computer Simulation*, **15**, 39–73.
- Steiger, N.M. and Wilson, J.R. (1999) Improved batching for confidence interval construction in steady-state simulation. In *Proceedings of the 1999 Winter Simulation Conference*, pp. 42–451.

Appendix

Proposition 1: The probability of a subsequence with run length τ for the simple runs-up test is $\frac{\tau}{(\tau+1)!}$, under the null hypothesis that the input data sequence are IID continuous random variables.

Proof: For the run length to be τ , we need to have $\tau + 1$ observations in a subsequence (with probability zero that any two values are equal, as is the case for continuous random variables).

Because these are IID random variables, there are exactly $(\tau + 1)!$ orderings of $X_1, X_2, \dots, X_\tau, X_{\tau+1}$, each with equal probability of occurring. We count the number of these $(\tau + 1)!$ orderings that produce a run (up) of length τ . Imagine “slots” $1, 2, \dots, \tau, \tau + 1$ for the X_i ’s. For a run up of length τ , slot τ must be occupied by the (unique) maximum value of $X_1, \dots, X_{\tau+1}$, so there is only one way to do that. Next, slot $\tau + 1$ can be occupied by any of the remaining τ X_i ’s, as they are all less than the maximum X_i already in slot τ , so there are τ ways to do that. Finally, the remaining $\tau - 1$ X_i ’s must occupy slots $1, 2, \dots, \tau - 1$ in their (unique) increasing order, so there is only one way to do that. Thus, the number of ways to have a run up of length τ is $1 \times \tau \times 1 = \tau$. Therefore, the probability of a run up

of length τ is $\frac{\tau}{(\tau+1)!}$, completing the proof. ■

Biographies

E. Jack Chen is a Senior Staff Specialist with BASF Corporation. He received a Ph.D. in Quantitative Analysis from the University of Cincinnati. His research interests are in the area of computer simulation, statistical data analysis, and stochastic processes.

W. David Kelton is a Professor in the Department of Quantitative Analysis and Operations Management at the University of Cincinnati. He received a B.A. in mathematics from the University of Wisconsin-Madison, an M.S. in mathematics from Ohio University, and M.S. and Ph.D. degrees in industrial engineering from Wisconsin. His research interests and publications are in the probabilistic and statistical aspects of simulation, applications of simulation, and stochastic models. He served as Editor-in-Chief of the *INFORMS Journal on Computing* from 2000 through 2006.