Simulation Output Analysis Using Integrated Paths

JAMES M. CALVIN
New Jersey Institute of Technology

This article considers the steady-state simulation output analysis problem for a process that satisfies a functional central limit theorem. We construct an estimator for the time-average variance constant that is based on iterated integrations of the sample path. When the observations are batched, the method generalizes the method of batch means. One advantage of the method is that it can be used without batching the observations; that is, it can allow for the process variance to be estimated at any time as the simulation runs without waiting for a fixed time horizon to complete. When used in conjunction with batching, the method can improve efficiency (the reciprocal of work times mean-squared error) compared with the standard method of batch means. In numerical experiments, efficiency improvement ranged from a factor of 1.5 (for the waiting time sequence in an M/M/1 queueing system with a single integrated path) up to a factor of 14 (for an autoregressive process and 19 integrated paths).

Categories and Subject Descriptors: C.4 [**Performance of Systems**]—*Performance attributes*; G.3 [**Probability and Statistics**]—*Probabilistic algorithms (including Monte Carlo), statistical computing, stochastic processes*; I.6.6 [**Simulation and Modeling**]: Simulation Output Analysis

General Terms: Algorithms, Performance, Theory

Additional Key Words and Phrases: Variance reduction, efficiency improvement

ACM Reference Format:

Calvin, J. M. 2007. Simulation output analysis using integrated paths. ACM Trans. Model. Comput. Simul. 17, 3, Article 13 (July 2007), 24 pages. DOI = 10.1145/1243991.1243994 http://doi.acm.org/10.1145/1243991.1243994

1. INTRODUCTION

In this article we describe a method for steady-state simulation output analysis that can be considered an extension of the method of batch means. The basic idea of the batch means method is to normalize the simulation output path in such a way that the normalized path converges to a Wiener process. Then the vector of values of the normalized process at equispaced points converges to a normal distribution involving the unknown variance constant. The scheme

Author's address: Computer Science Department, New Jersey Institute of Technology, Newark, NJ 07102.

we describe constructs $k+1$ normalized paths, where the rth normalized path converges to an r-fold integrated Wiener process. Compared with batch means, this leads to more efficient estimation of the variance constant, where we take efficiency to be the reciprocal of the product of work and mean-squared error. In the limit, the variance of the variance constant estimator is proportional to $1/(k+1)$. At each step of the simulation, k additions and stores are required beyond what is required for the standard method.

One of the motivations for the proposed method is that it can provide an alternative to breaking up the path into fixed length segments. In some computer simulation settings the ultimate length of the simulation is not known in advance, but is determined adaptively as the simulation progresses. For example, when comparing alternative systems to determine which has the best performance, as the simulations progress at different parameter settings it becomes apparent that the simulations at the more promising parameters should run longer than the simulations at less promising values. The adaptive rules for determining simulation run-length depend on mean and variability estimates that must be available over the entire time interval, not just at predetermined epochs. The proposed method can be used without batching; if $k+1$ integrated paths are kept, then the limiting distribution of the variance estimator is the same as that of the standard batch means method with $k+1$ batches. The advantage of the proposed method is that an estimator can be constructed at any step, without waiting for some number of batches to be simulated.

The next section describes the assumptions we make on the simulated process. Section 3 describes the output analysis algorithm and states the main result. A stochastic process that arises as the limit of processes constructed during the simulation and is used in proving the main result described in Section 3, is described in Section 4. Section 5 gives an alternative estimator with lower bias, and includes a proof of the main result from Section 3. Section 6 describes how to use the proposed method in conjunction with batching of the data (that is, splitting the simulated path up into disjoint segments). Our main interest is in simulations that do not store all the simulated data, but in Section 7 we describe how to adapt the estimator to the situation where the sample average is used to center the output process. In Section 8 we study the asymptotic bias of the estimator. Section 9 reports the results of some numerical experiments on an autoregressive process and the waiting time sequence in an M/M/1 queueing system. We end with some concluding remarks.

2. ASSUMPTIONS

Suppose that a simulation generates a real-valued sequence $Y_1, Y_2, \ldots$. We assume throughout the article that there exist constants $\mu \in \mathbb{R}$, $\sigma \in (0, \infty)$, such that if we define the process

$$X_n^*(t) = n^{-1/2} \sum_{i=1}^{\lceil nt \rceil} (Y_i - \mu), \quad 0 \le t \le 1,$$

where $\lceil x \rceil$ denotes the smallest integer not less than x, then $X_n^* \xrightarrow{\mathcal{D}} \sigma B$ as $n \to \infty$ in the space $D = D([0,1])$ of right-continuous functions on the unit

interval that have left-hand limits at each point, endowed with the Skorohod metric. Here B is a standard Wiener process and $\stackrel{\mathcal{D}}{\rightarrow}$ denotes convergence in distribution. In this case we say that the process satisfies a functional central limit theorem; see Billingsley [1999]. Most processes that arise as models for a steady-state simulation satisfy a functional central limit theorem. For example, suppose that $\{Y_n\}$ is a second order stationary sequence with mean $EY_n = \mu$ and

$$\sum_{n\geq 1} E((Y_0 - \mu)(Y_n - \mu)) < \infty.$$

Then $\{Y_n\}$ satisfies the functional central limit theorem. In particular the conclusion holds if $\{Y_n\}$ is an m-dependent sequence or an autoregressive process; see, for example, Billingsley [1999].

In discrete-event simulation one often estimates μ by the sample average

$$\mu_n^* = \frac{1}{n}\sum_{i=1}^{n} Y_i.$$

If the functional central limit theorem assumption holds, then $\mu_n^* \rightarrow \mu$ almost surely, and

$$\sqrt{n}(\mu_n^* - \mu) \stackrel{\mathcal{D}}{\rightarrow} \mathcal{N}(0, \sigma^2). \tag{1}$$

If a suitable estimate of σ^2 is available, then the central limit theorem (1) can be used to form confidence intervals for μ, or it may be used in adaptive methods for comparing the performance of a system at different parameter values.

3. SIMULATION ALGORITHM

We now outline the proposed method for estimating the parameter σ^2 of the simulated process. At first we assume that at least two independent simulations are run in parallel to center the output. In Section 7 we describe the modification for a single simulation.

Choose a parameter $k \geq 0$ (the integration count parameter). Run two or more simulations in parallel, producing independent and identically distributed output sequences $\{Y_1^1, Y_2^1, \ldots\}$ and $\{Y_1^2, Y_2^2, \ldots\}$. Define $W_0^j = 0$ for $0 \leq j \leq k$ and for $i > 0$ set

$$W_i^0 = \sum_{l=1}^{i} \left(Y_l^1 - Y_l^2\right)$$

and for $1 \leq j \leq k$ set

$$W_i^j = \sum_{l=1}^{i} W_l^{j-1}.$$

The data maintained by the simulation method after n simulated data points is $\mathbf{W}_n = (W_n^0, W_n^1, \ldots, W_n^k)$. This vector is easily updated in time $O(k)$ at each simulation step; see Figure 1.

(1) Set $W^i = 0$, $0 \le i \le k$.
(2) For $j = 1$ to n
—generate the jth data pair Y_j^1, Y_j^2,
—add $Y_j^1 - Y_j^2$ to W^0,
—for $i = 1$ to k set $W^i = W^i + W^{i-1}$.
(3) For $r = 0$ to k set

$$Z_r = \sum_{j=0}^{r} A_{rj} n^{-j-1/2} W^j.$$

(4) Return

$$V_n = \frac{1}{2(k+1)} \sum_{r=0}^{k} (Z_r)^2.$$

Fig. 1. The simulation algorithm for the estimator V_n.

Suppose that after n transitions, an estimate of σ^2 is required. For $0 \le r \le k$, set

$$Z_n^r = \sum_{j=0}^{r} A_{rj} n^{-j-1/2} W_n^j,$$

where

$$A_{ij} = \frac{(-1)^{i+j}(i+j)!\sqrt{2i+1}}{j!(i-j)!} \tag{2}$$

if $0 \le j \le i$ and $A_{ij} = 0$ otherwise.

The following theorem provides the basis for the proposed estimator.

THEOREM 3.1. *As $n \to \infty$,*

$$\mathbf{Z}_n = \left(Z_n^0, Z_n^1, \ldots, Z_n^k\right) \xrightarrow{\mathcal{D}} \sqrt{2}\,\sigma \mathcal{N}\left(\mathbf{0}, \mathbf{I}_{(k+1)\times(k+1)}\right),$$

where $\mathbf{0}$ and $\mathbf{I}_{(k+1)\times(k+1)}$ denote the zero vector in $\mathbb{R}^{k+1}$ and the $(k+1)\times(k+1)$ identity matrix, respectively.

We will prove Theorem 3.1 in Section 5.

Let V_n denote the estimator of σ^2 based on a simulation of length n, defined by

$$V_n = \frac{1}{2(k+1)} \sum_{r=0}^{k} \left(Z_n^r\right)^2. \tag{3}$$

Note that Theorem 3.1 implies that

$$V_n \xrightarrow{\mathcal{D}} \sigma^2 \frac{\chi_{k+1}^2}{k+1} \tag{4}$$

as $n \to \infty$, where χ_m^2 denotes a chi-squared random variable with m degrees of freedom. The asymptotic variance of V_n as $n \to \infty$ is therefore $2\sigma^4/(k+1)$.

Figure 1 gives the simulation algorithm for the case where the simulation run length n is known in advance and an estimate of σ^2 is only required after n transitions. However, steps 3 and 4 can be carried out at any time during the

simulation, in parallel with step 2, allowing for the computation of an estimate of σ^2 at any step along the way in time $O(k^2)$ per sample.

The number of iterated integrations k is a fixed parameter of the algorithm. The returned value is an estimate of σ^2. The algorithm can use stored values for the upper left $(k+1)\times(k+1)$ submatrix of the matrix $\mathbf{A}$ defined at (2).

The work is dominated by Step 2, which requires time roughly of order $n(s+ka)$, where s is the time to generate one step of the simulation, and a is the time to add and store a number. Steps 1 and 4 require time $O(k)$, while step 3 takes time $O(k^2)$. The storage is k numbers in addition to W^0.

Theorem 3.1 can be used to construct confidence intervals for μ using standardized time series, or cancellation methods; see Schruben [1983]. After n observations, we construct the point estimator

$$\mu_n \stackrel{\triangle}{=} \frac{\sum_{i=1}^{n}\left(Y_i^1+Y_i^2\right)}{2n}.$$

Then using (4) and (1) (recall that the variance constant is $\sigma^2/2$ in our setup),

$$\left(\frac{2n}{V_n}\right)^{1/2}(\mu_n-\mu)\stackrel{\mathcal{D}}{\to} t_{k+1},$$

where t_{k+1} denotes a Student t distribution with $k+1$ degrees of freedom. Setting $g_n=(V_n/(2n))^{1/2}$, a $100(1-\delta)\%$ confidence interval for μ is

$$[\mu_n - qt(1-\delta/2,k+1)\cdot g_n,\ \mu_n + qt(1-\delta/2,k+1)\cdot g_n]$$

where $qt(1-\delta/2,k+1)$ is the $1-\delta/2$ quantile of the t distribution with $k+1$ degrees of freedom.

The next section introduces a process that is used in the proof of Theorem 3.1.

4. *R*-FOLD INTEGRATED WIENER MEASURE

The vector process $\mathbf{W}_n$ constructed by the simulation, suitably normalized, will have a limiting distribution that is associated with a particular Gaussian process that we now briefly describe.

Let $X^0=\sigma B$ be a Wiener process with diffusion parameter σ. This is the centered, continuous Gaussian process with covariance

$$E(X^0(s)X^0(t))=\sigma^2\min(s,t).$$

For $r\geq 1$, let $T^r: D\to D$ denote the operator of r-fold integration:

$$T^r f(t)=\int_{s=0}^{t} f(s)\frac{(t-s)^{r-1}}{(r-1)!}ds.$$

Note that $T^r f$ is continuous, even when f is not. Fix $k\geq 1$ and set $X^r=T^rX^0$ for $1\leq r\leq k$. Then

$$X^r(t)=T^rX^0(t)=\int_{s=0}^{t}T^{r-1}X^0(s)\,ds=\int_{s=0}^{t}X^{r-1}(s)\,ds,\quad 0\leq t\leq 1.$$

The distribution of X^r is called the r-fold integrated Wiener measure; see Novak and Ritter [1993] for properties. The vector process

$$\mathbf{X}(t) = (X^0(t), X^1(t), \ldots, X^k(t))$$

is a Gaussian process with the Markov property. For $0 \le i \le k$ and $0 \le s < t$,

$$E(X^i(t) \mid X^0(s), X^1(s), \ldots, X^k(s)) = \sum_{l=0}^{i} \frac{(t-s)^l}{l!} X^{i-l}(s) \triangleq \mu^i_{t-s}(\mathbf{X}(s));$$

see Novak and Ritter [1993]. Set $\mu_t(x) = (\mu^0_t(x), \ldots, \mu^k_t(x))$. Also, for $0 \le i, j \le r$,

$$\begin{aligned}
\operatorname{cov}(X^i(t), X^j(t)) &= \operatorname{cov}(T^i X^0(t), T^j X^0(t)) \\
&= \int_{s=0}^{t} \int_{u=0}^{t} \operatorname{cov}(X^0(s), X^0(u)) \frac{(t-s)^{i-1}(t-u)^{j-1}}{(i-1)!(j-1)!} du\, ds \\
&= \int_{s=0}^{t} \int_{u=0}^{t} \sigma^2 \min(s,u) \frac{(t-s)^{i-1}(t-u)^{j-1}}{(i-1)!(j-1)!} du\, ds \\
&= \sigma^2 \frac{t^{i+j+1}}{(i+j+1)i!j!} \triangleq \sigma^2 (\mathbf{C}_t)_{ij}. \qquad (5)
\end{aligned}$$

Using the Cholesky decomposition of the covariance matrix $\mathbf{C}_t$, we obtain

$$\mathbf{C}_t = \mathbf{L}_t \cdot \mathbf{L}_t^T,$$

where

$$(\mathbf{L}_t)_{ij} = \begin{cases} \frac{i!\sqrt{2j+1}\, t^{i+1/2}}{(i+j+1)!(i-j)!} & \text{if } 0 \le j \le i \le k, \\ 0 & \text{if } 0 \le i < j \le k. \end{cases}$$

Then

$$\left(\mathbf{L}_t^{-1}\right)_{ij} = \begin{cases} \frac{(-1)^{i+j}(i+j)!\sqrt{2i+1}\, t^{-j-1/2}}{j!(i-j)!} & \text{if } 0 \le j \le i \le k, \\ 0 & \text{if } 0 \le i < j \le k, \end{cases}$$

and

$$\mathbf{L}_t^{-1}(\mathbf{X}(t) - \mu_t(X(0))) \sim \sigma \mathcal{N}\left(\mathbf{0}, \mathbf{I}_{(k+1)\times(k+1)}\right). \qquad (6)$$

5. IMPROVED ESTIMATOR

The vector $(W_n^0, \ldots, W_n^k)$ is easy to compute at each step of the simulation; however, as will be seen in Section 9, it gives rise to an estimator with large bias for large values of k. In this section we define an estimator that is more complex to compute, but that results in less bias and also in a simpler convergence proof.

Define

$$\widetilde{W}_n^0(t) = n^{-1/2} \sum_{i=1}^{\lceil nt \rceil} \left(Y_i^1 - Y_i^2\right), \quad 0 \le t \le 1,$$

and for $r \ge 1$,

$$\widetilde{W}_n^r(t) = T^r \widetilde{W}_n^0(t), \quad 0 \le t \le 1,$$

and set $\widetilde{\mathbf{W}}_n = (\widetilde{W}_n^0, \ldots, \widetilde{W}_n^r)$. For $0 \le r \le k$, let

$$\widetilde{Z}_n^r = \sum_{j=0}^{r} A_{rj} \widetilde{W}_n^j(1), \tag{7}$$

where $\mathbf{A}$ is defined at (2). Our new estimator of σ^2 is

$$\widetilde{V}_n = \frac{1}{2(k+1)} \sum_{r=0}^{k} \left(\widetilde{Z}_n^r\right)^2. \tag{8}$$

THEOREM 5.1. *As $n \to \infty$,*

$$\widetilde{\mathbf{Z}}_n = \left(\widetilde{Z}_n^0, \widetilde{Z}_n^1, \ldots, \widetilde{Z}_n^k\right) \xrightarrow{\mathcal{D}} \sqrt{2}\,\sigma \mathcal{N}\left(\mathbf{0}, \mathbf{I}_{(k+1)\times(k+1)}\right).$$

PROOF. By the functional central limit theorem assumption,

$$\widetilde{W}_n^0 \xrightarrow{\mathcal{D}} \sqrt{2}\sigma B.$$

Continuity of the map $f \mapsto (f, Tf, T^2 f, \ldots, T^k f)$ implies that

$$\left(\widetilde{W}_n^0, T\widetilde{W}_n^0, \cdots, T^k \widetilde{W}_n^0\right) \xrightarrow{\mathcal{D}} \sqrt{2}\sigma(B, TB, T^2 B, \ldots, T^k B);$$

in particular,

$$\begin{aligned} \widetilde{\mathbf{W}}_n(1) &= \left(\widetilde{W}_n^0(1), T\widetilde{W}_n^0(1), \ldots, T^k \widetilde{W}_n^0(1)\right) \\ &\xrightarrow{\mathcal{D}} \sqrt{2}\sigma(B(1), TB(1), T^2 B(1), \ldots, T^k B(1)). \end{aligned} \tag{9}$$

The theorem now follows from (9) and (6), since

$$\begin{aligned} \widetilde{\mathbf{Z}}_n &= \left(\mathbf{L}_1^{-1}\right)\widetilde{\mathbf{W}}_n(1) \\ &\xrightarrow{\mathcal{D}} \sqrt{2}\sigma\left(\mathbf{L}_1^{-1}\right)(B(1), TB(1), T^2 B(1), \ldots, T^k B(1)) \\ &\stackrel{\mathcal{D}}{=} \sqrt{2}\sigma\mathcal{N}\left(\mathbf{0}, \mathbf{I}_{(k+1)\times(k+1)}\right) \end{aligned}$$

by (6). □

For the remainder of the article we write $\widetilde{\mathbf{W}}_n$ for $\widetilde{\mathbf{W}}_n(1)$, and Y_i for $Y_i^1 - Y_i^2$. To compare $\widetilde{W}_n^r$ and W_n^r, observe that

$$\begin{aligned} \widetilde{W}_n^r = T^r \widetilde{W}_n^0 &= \int_{s=0}^{1} n^{-1/2} \sum_{i=1}^{\lceil ns \rceil} Y_i \frac{(1-s)^{r-1}}{(r-1)!} ds \\ &= n^{-r-1/2} \sum_{i=1}^{n} Y_i \frac{(n-i+1)^r}{r!}. \end{aligned}$$

This compares with

$$W_n^r = \sum_{i_r=1}^{n} \sum_{i_{r-1}=1}^{i_r} \cdots \sum_{i_0=1}^{i_1} \left(Y_{i_0}^1 - Y_{i_0}^2\right)$$
$$= \sum_{i=1}^{n} Y_i \frac{(n-i+1)(n-i+2)\cdots(n-i+r)}{r!} = \sum_{i=1}^{n} Y_i \frac{(n-i+1)^{\overline{r}}}{r!}, \quad r \geq 1,$$

where

$$a^{\overline{r}} = (a)(a+1)(a+2)\cdots(a+r-1)$$

is the rising factorial function; see Knuth [1997]. Note that for fixed k,

$$\widetilde{W}_n^r - n^{-r-1/2} W_n^r \xrightarrow{P} 0, \quad 0 \leq r \leq k,$$

where $\xrightarrow{P}$ indicates convergence in probability. Therefore, Theorem 5.1 implies Theorem 3.1.

For the remainder of the article, including the bias approximation in Section 8 and, with one exception, the numerical experiments in Section 9, we consider mainly the estimator $\widetilde{V}_n$.

The rising factorial function has the expansion

$$(n-i+1)^{\overline{r}} = \sum_{j=0}^{r} (-1)^{r-j} \begin{bmatrix} r \\ j \end{bmatrix} (n-i+1)^j$$

[Knuth 1997, p. 67], where the $\begin{bmatrix} n \\ i \end{bmatrix}$ are the Stirling numbers of the first kind, satisfying the recurrence

$$\begin{bmatrix} n+1 \\ m \end{bmatrix} = \begin{bmatrix} n \\ m-1 \end{bmatrix} - n \begin{bmatrix} n \\ m \end{bmatrix}, \quad 1 \leq m \leq n, \quad \begin{bmatrix} n \\ n \end{bmatrix} = 1, \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 1,$$

and $\begin{bmatrix} n \\ 0 \end{bmatrix} = 0$ for $n \geq 1$. Thus

$$n^{-r-1/2} W_n^r = n^{-r-1/2} \sum_{i=1}^{n} Y_i \frac{(n-i+1)^{\overline{r}}}{r!}$$
$$= n^{-r-1/2} \sum_{i=1}^{n} Y_i \frac{\sum_{j=0}^{r} (-1)^{r-j} \begin{bmatrix} r \\ j \end{bmatrix} (n-i+1)^j}{r!}$$
$$= n^{-r-1/2} \sum_{j=0}^{r} (-1)^{r-j} \begin{bmatrix} r \\ j \end{bmatrix} \sum_{i=1}^{n} Y_i \frac{(n-i+1)^j}{j!} \frac{j!}{r!}$$
$$= \sum_{j=0}^{r} (-1)^{r-j} \begin{bmatrix} r \\ j \end{bmatrix} \frac{j!}{r!} \widetilde{W}_n^j \, n^{j-r}$$

and so

$$\widetilde{W}_n^r = n^{-r-1/2} W_n^r - \sum_{j=1}^{r-1} (-1)^{r-j} \begin{bmatrix} r \\ j \end{bmatrix} \frac{j!}{r!} \widetilde{W}_n^j \, n^{j-r}.$$

Table I. Examples of $c(i, j)$

$i\backslash j$	0	1	2	3	4	5
0	1					
1	0	1				
2	0	1/2	1			
3	0	1/3	1	1		
4	0	1/4	11/12	3/2	1	
5	0	1/5	5/6	7/4	2	1

(1) Set $W^i = 0$, $0 \le i \le k$.
(2) For $j = 1$ to n
—generate the jth data pair Y_j^1, Y_j^2,
—add $Y_j^1 - Y_j^2$ to W^0,
—for $i = 1$ to k set $W^i = W^i + W^{i-1}$.
(3) Set $\widetilde{W}^0 = n^{-1/2}W^0$, and for $r = 1$ to k set

$$\widetilde{W}^r = n^{-r-1/2}W^r - \sum_{j=1}^{r-1}(-1)^{r-j}\left[{r \atop j}\right]\frac{j!}{r!}\widetilde{w}^j n^{j-r}$$

(4) For $r = 0$ to k set

$$\widetilde{Z}_r = \sum_{j=0}^{r} A_{rj}\widetilde{W}^j.$$

(5) Return

$$\widetilde{V}_n = \frac{1}{2(k+1)}\sum_{r=0}^{k}\widetilde{Z}_r^2.$$

Fig. 2. The simulation algorithm for the improved estimator $\widetilde{V}_n$.

Define the triangular array $\tilde{c}(i, j)$, $0 \le j \le i$, by

$$\tilde{c}(i+1, j) = \frac{j\tilde{c}(i, j-1) - i\tilde{c}(i, j)}{i+1}$$

and $\tilde{c}(i, i) = 1$, $\tilde{c}(i, 0) = 1$ if $i = 0$ and 0 otherwise. Define $c(i, j) = |\tilde{c}(i, j)|$. Then

$$\widetilde{W}_n^r = n^{-r-1/2}W_n^r - \sum_{j=1}^{r-1} c(r, j)\widetilde{W}_n^j\, n^{j-r}.$$

The first six rows of the $c(i, j)$ are given in Table I. In particular, $\widetilde{W}_n^0 = n^{-1/2}W_n^0$, $\widetilde{W}_n^1 = n^{-3/2}W_n^1$, and $\widetilde{W}_n^2 = n^{-5/2}W_n^2 - n^{-5/2}W_n^1/2$.

The $c(i, j)$'s are related to the Stirling numbers of the first kind by

$$c(i, j) = (-1)^{i-j}\left[{i \atop j}\right]\frac{j!}{i!}.$$

Figure 2 gives the simulation algorithm for the improved estimator $\widetilde{V}_n$. The main difference between the algorithm for V_n and this algorithm is step 3, which adds an additional $O(k^2)$ time. The algorithm can use stored values for the matrices $\mathbf{A}$ and $\mathbf{c}$ or they can be computed in a step 0 in time $O(k^2)$.

As in the case of V_n, the algorithm can be easily modified to allow for the computation of an estimate of σ^2 at any step along the way, in time $O(k^2)$ per sample.

6. BATCHING

Using the approach from the previous section does not result in a consistent estimator for σ^2. In this section we describe how to use batching in this context to obtain consistent estimators. The results in this section apply to either V_n or $\widetilde{V}_n$.

Let us suppose first that the simulation run length n is known before the simulation begins. We choose a parameter b, the number of batches. Assume for simplicity that $b \cdot l = n$ for an integer l (the batch size). For the simulation from 1 to l we apply the method described; but replacing n with l. We retain the parameter k from the previous section. This results in a vector

$$(Z_1^0, Z_1^1, Z_1^2, \ldots, Z_1^k).$$

Continue in this fashion, obtaining vectors

$$(Z_j^0, Z_j^1, Z_j^2, \ldots, Z_j^k)$$

for the simulated path segment $(j-1)l, (j-1)l+1, \ldots, jl$ for $j = 2, 3, \ldots, b$. For the segments after the first, we must subtract the conditional mean based on the process up to that segment. Let $V_{b,n}$ denote the estimator of σ^2 after simulation run length n, based on b batches, defined by

$$V_{b,n} = \frac{1}{2b(k+1)} \sum_{i=1}^{b} \sum_{j=0}^{k} \left(Z_i^j\right)^2. \tag{10}$$

Then

$$V_{b,n} \xrightarrow{\mathcal{D}} \sigma^2 \frac{\chi^2_{b(k+1)}}{b(k+1)}. \tag{11}$$

The limit random variable depends only on the product $b(k+1)$, so we can trade off, increasing one while decreasing the other. For $k = 0$ the method is the standard method of batch means. For $b = 1$, the path is not broken up into segments at all.

The variance of the limit random variable is

$$\frac{2\sigma^4}{b(k+1)}. \tag{12}$$

Of course this captures the limiting variance for standard batch means ($k = 0$) and shows the decrease in variance with increasing k. Other methods have been proposed with limiting variances of the form $c\sigma^4/b$ for a constant c; for example overlapping batch means ($c = 4/3$; see Meketon and Schmeiser [1984]), and weighted versions of batched estimators [Alexopoulos et al. 2006].

To allow batching, the algorithm is applied to the simulation path split up into b batches, treated independently. In situations where the simulation run length is not known in advance, dynamic batching methods are available; see for example Yeh and Schmeiser [2000].

7. SINGLE SIMULATION RUN

This method can be adapted to use a single simulation run instead of two or more. In this case the sample average at time n is used to center the processes, and we end up with the same limiting process $\mathbf{X}$, conditioned on $X^0(1) = 0$.

Specifically, for $1 \le j \le k$, set

$$\overline{W}_n^j = \widetilde{W}_n^j - \frac{n^j}{(j+1)!}\widetilde{W}_n^0.$$

Then

$$\left(n^{-3/2}\overline{W}_n^1, n^{-5/2}\overline{W}_n^2, \ldots, n^{-(2k+1)/2}\overline{W}_n^k\right) \xrightarrow{\mathcal{D}} \sigma\mathcal{N}(\mathbf{0}, \mathbf{C}),$$

where

$$(\mathbf{C})_{ij} = \frac{ij}{(i+1)!(j+1)!(i+j+1)}, \quad 1 \le i, j \le k.$$

The covariance matrix can be written $\mathbf{C} = \mathbf{L} \cdot \mathbf{L}^T$, where

$$(\mathbf{L})_{ij} = \begin{cases} \frac{i!\sqrt{2j+1}}{(i+j+1)!(i-j)!} & \text{if } i \ge j, \\ 0 & \text{if } i < j \end{cases}$$

and

$$(\mathbf{L}^{-1})_{ij} = \begin{cases} \frac{(-1)^{i+j}(i+j)!\sqrt{2i+1}}{j!(i-j)!} & \text{if } i \ge j, \\ 0 & \text{if } i < j. \end{cases}$$

The same sort of analysis used to construct V_n gives an estimator that converges to a t_k distribution. Note that we end up with one less degree of freedom.

8. BIAS ESTIMATION

In this section we estimate the bias of the estimator $\widetilde{V}_n$ to order n^{-1}; that is, we identify a quantity η associated with the simulated process such that

$$n(E\widetilde{V}_n - \sigma^2) \to \eta$$

as simulation run length $n \to \infty$. The parameter k, on which $\widetilde{V}_n$ and η depend, is fixed throughout.

We will assume that $\{Y_i\}$ is a second order stationary process with $EY_i = 0$ and $E(Y_iY_{i+k}) = \gamma_k$, $k \ge 0$. We further assume that

$$\sum_{j=1}^{\infty} j|\gamma_j| < \infty,$$

and define

$$\lambda = \sum_{j=1}^{\infty} j\gamma_j, \qquad \sigma^2 = \gamma_0 + 2\sum_{j=1}^{\infty}\gamma_j.$$

THEOREM 8.1. *Define the estimator $\widetilde{V}_n$ at (8) with some choice of $k \ge 0$. Then*

$$n(E\widetilde{V}_n - \sigma^2) \to -2(k+1)\lambda$$

as $n \to \infty$.

According to Chien et al. [1997], for the nonoverlapping batch means estimator $N(b,m)$ with b batches of size m each,

$$n(EN(b,m)-\sigma^2)\to -2(b+1)\lambda$$

as $n=bm\to\infty$.

With batching, the bias is given by Theorem 8.1, where n is the batch size.

For the proof of Theorem 8.1 we require a technical lemma, the proof of which appears in the appendix.

LEMMA 8.2. *For nonnegative integers α, β,*

$$E\left(\widetilde{W}_n^{\alpha}\widetilde{W}_n^{\beta}\right)=\frac{1}{\alpha!\beta!}\left(\frac{\sigma^2}{\alpha+\beta+1}-\frac{1}{n}\left(I_{(\alpha+\beta>0)}+2I_{(\alpha+\beta=0)}\right)\lambda+\frac{\sigma^2 I_{(\alpha+\beta>0)}}{2n}+o(1/n)\right), \tag{13}$$

where I_C is the indicator function of the set C.

PROOF OF THEOREM 8.1. From (7),

$$\widetilde{Z}_n^i=\sum_{j=0}^{i}A_{ij}\widetilde{W}_n^j,\quad 0\le i\le k.$$

Therefore,

$$\begin{aligned}
E\widetilde{Z}_i^2&=\sum_{j=0}^{i}\sum_{m=0}^{i}A_{ij}A_{im}E\left(\widetilde{W}_n^j\widetilde{W}_n^m\right)\\
&=\sum_{j=0}^{i}\sum_{m=0}^{i}(-1)^{j+m}(2i+1)\frac{(i+j)!(i+m)!}{j!(i-j)!m!(i-m)!}E\left(\widetilde{W}_n^j\widetilde{W}_n^m\right)\\
&=\sum_{j=0}^{i}\sum_{m=0}^{i}(-1)^{j+m}(2i+1)\frac{(i+j)!(i+m)!}{j!(i-j)!m!(i-m)!}\frac{1}{j!m!}\\
&\quad\times\left(\frac{\sigma^2}{j+m+1}-\frac{1}{n}-\frac{1}{n}\left(I_{(j+m>0)}+2I_{(j+m=0)}\right)\lambda+\frac{\sigma^2 I_{(j+m>0)}}{2n}+o(1/n)\right).
\end{aligned} \tag{14}$$

Now use the identities

$$\sum_{j=0}^{i}\sum_{m=0}^{i}(-1)^{j+m}(2i+1)\frac{(i+j)!(i+m)!}{j!(i-j)!m!(i-m)!}\frac{1}{j!m!}=2i+1, \tag{15}$$

which is a consequence of

$$\sum_{j=0}^{i}(-1)^j\frac{(i+j)!}{(j!)^2(i-j)!}=(-1)^i$$

(from Mathematica version 5.2), and

$$\sum_{j=0}^{i}\sum_{m=0}^{i}(-1)^{j+m}(2i+1)\frac{(i+j)!(i+m)!}{j!(i-j)!m!(i-m)!}\frac{1}{j!m!(j+m+1)}=1. \qquad (16)$$

For the latter identity, note that if $(\widetilde{W}_n^0,\ldots,\widetilde{W}_n^k)$ were normally distributed with mean 0 and covariance matrix $\mathbf{C}=\mathbf{C}_1$ (defined at (5)), then we would have $\widetilde{Z}_n^i\sim N(0,1)$ and

$$\begin{aligned}1=E\big(\widetilde{Z}_n^i\big)^2 &= E\left(\sum_{j=0}^{i}A_{ij}\widetilde{W}_n^j\right)^2=\sum_{j=0}^{i}\sum_{m=0}^{i}(-1)^{j+m}(2i+1)\frac{(i+j)!(i+m)!}{j!(i-j)!m!(i-m)!}C_{jm}\\ &= \sum_{j=0}^{i}\sum_{m=0}^{i}(-1)^{j+m}(2i+1)\frac{(i+j)!(i+m)!}{j!(i-j)!m!(i-m)!}\frac{1}{j!m!(j+m+1)}.\end{aligned}$$

Using the identities (15) and (16) in (14) gives

$$E\widetilde{Z}_i^2=\sigma^2-\frac{2\lambda}{n}(2i+1)+o(1/n).$$

Therefore,

$$\begin{aligned}E\widetilde{V}_n &= \frac{1}{k+1}\sum_{i=0}^{k}E\widetilde{Z}_i^2\\ &= \frac{\sum_{i=0}^{k}(\sigma^2-2\lambda(2i+1)n^{-1}+o(1/n))}{k+1}\\ &= \sigma^2-\frac{2(k+1)\lambda}{n}+o(1/n),\end{aligned}$$

and so

$$n(E\widetilde{V}_n-\sigma^2)\to -2(k+1)\lambda$$

as $n\to\infty$. □

9. NUMERICAL EXPERIMENTS

This section reports the results of numerical experiments on two different models; a first-order autoregressive process and the waiting time sequence for an M/M/1 queue. For each model we ran experiments calculating the sample variance, sample bias, and running time.

In addition to comparing the variance and bias with the limiting values given in previous sections, our goal was to examine the efficiency of the estimators. We take the efficiency of an estimator with bias B and variance V, that takes time T to compute, to be

$$\frac{1}{T(B^2+V)};$$

that is, the efficiency of a simulation estimator is the reciprocal of the mean-squared error multiplied by the running time. Increasing the parameter k

increases the running time, while decreasing the variance. In the experiments we compute the relative increase in efficiency compared with the base case of $k = 0$, which is the standard method of batch means. Both of the models are inexpensive to simulate; the efficiency increases would be larger for more computationally intensive simulations. The two models we chose are also used (with the same parameters) in Alexopoulos et al. [2006], so the results in this section allow the proposed method to be compared with a range of methods surveyed there, at least for the asymptotic characteristics.

9.1 The AR(1) Model

The AR(1) model is a stationary Gaussian process defined by $Y_0 \sim N(0, 1)$ and

$$Y_i = \varphi Y_{i-1} + \epsilon_i, \quad i \geq 1,$$

where $-1 < \varphi < 1$ and the $\{\epsilon_i\}$ are independent, $\epsilon_i \sim N(0, 1 - \varphi^2)$. For this process $\gamma_k = \varphi^k$,

$$\lambda = \sum_{k=1}^{\infty} k\gamma_k = \frac{\varphi}{(1-\varphi)^2},$$

and

$$\sigma^2 = \gamma_0 + 2\sum_{k=1}^{\infty} \gamma_k = \frac{1+\varphi}{1-\varphi}.$$

For the experiment we took $\varphi = 0.9$, so $\sigma^2 = 19$ and $\lambda = 90$.

The experiments consisted of 10,000 independent replications with a run length of $n = 200,000$; since we used two independent simulations, as described in Section 3, this corresponds to 400,000 simulated steps of the AR(1) process. The simulations were split into $b = 20$ batches. In Figure 3, the sample variance of the estimator for the AR(1) process is plotted for values of k from 0 to 19, and also the limiting variance calculated based on (12). The sample and asymptotic variances are virtually identical for this model. Figure 4 plots the sample bias of the estimator $\widetilde{V}_n$ and also the asymptotic first order bias given by Theorem 8.1, which are reasonably close. Figure 5 plots the actual computer running times of the experiments, which increase approximately affinely. In Figure 6 the relative efficiency (efficiency divided by efficiency for the batch means estimator, $k = 0$) is plotted for various values of k. For $k = 1$, the efficiency improves by a factor of 1.99, while for $k = 19$, the efficiency improves by a factor of 14.7. Finally, we compare the sample bias of V_n and $\widetilde{V}_n$ with the asymptotic first order bias given by Theorem 8.1. The results of the experiments (with no batching) are shown in Figure 7. For the estimator $\widetilde{V}_n$, the sample bias is close to the asymptotic bias. For small k, the bias of V_n is also close, but for large k (greater than about 10) the bias increases rapidly.

9.2 The M/M/1 Waiting Time Sequence

The second model is the waiting time sequence for a stationary M/M/1 queueing system operated on a first-come first-served basis. Arrivals constitute a Poisson

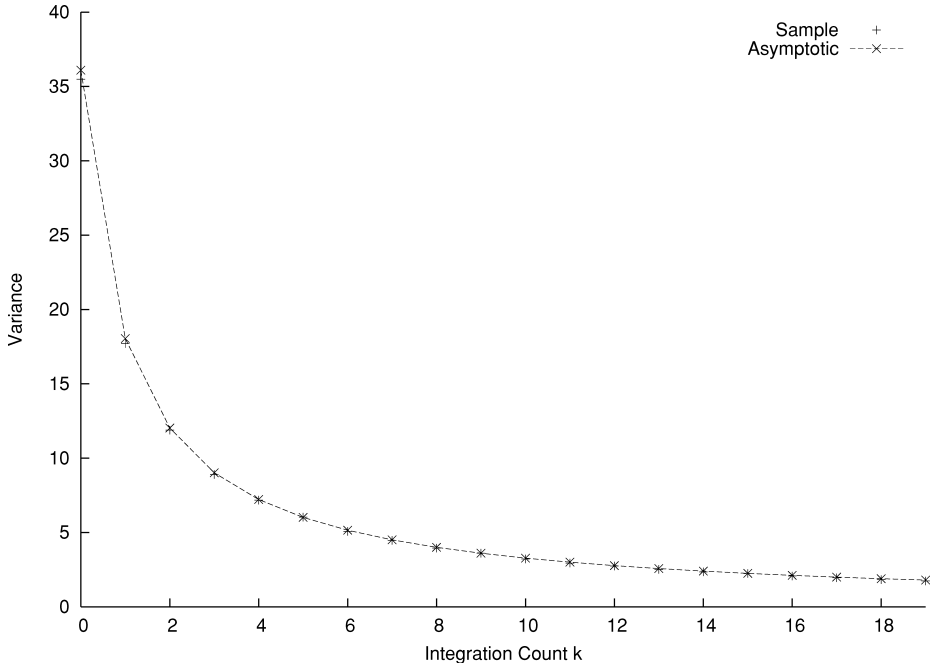

Fig. 3. Sample and asymptotic variance for AR(1) simulations, varying k.

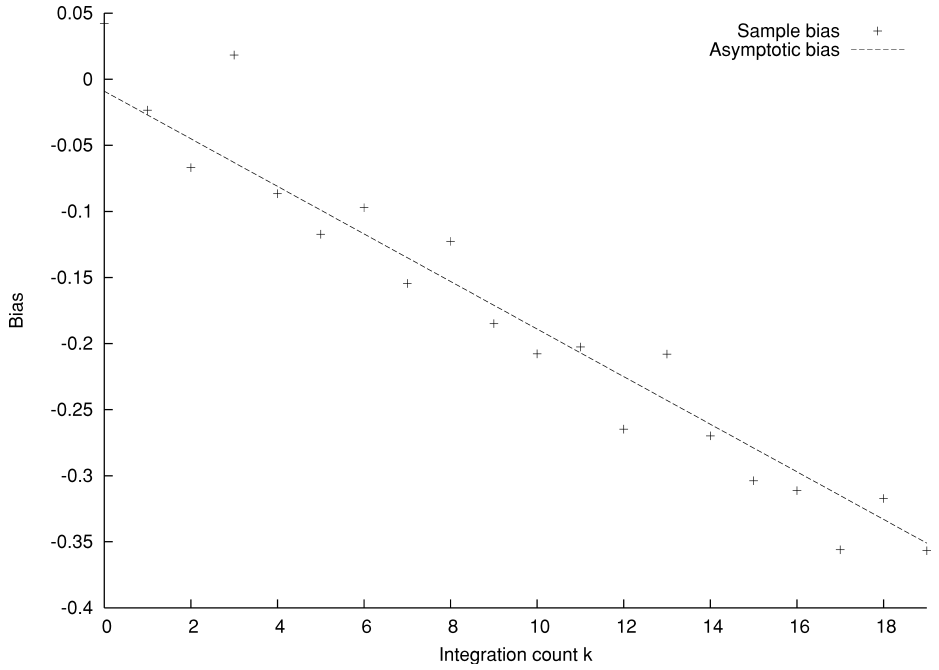

Fig. 4. Sample and asymptotic bias for AR(1) simulations, varying k.

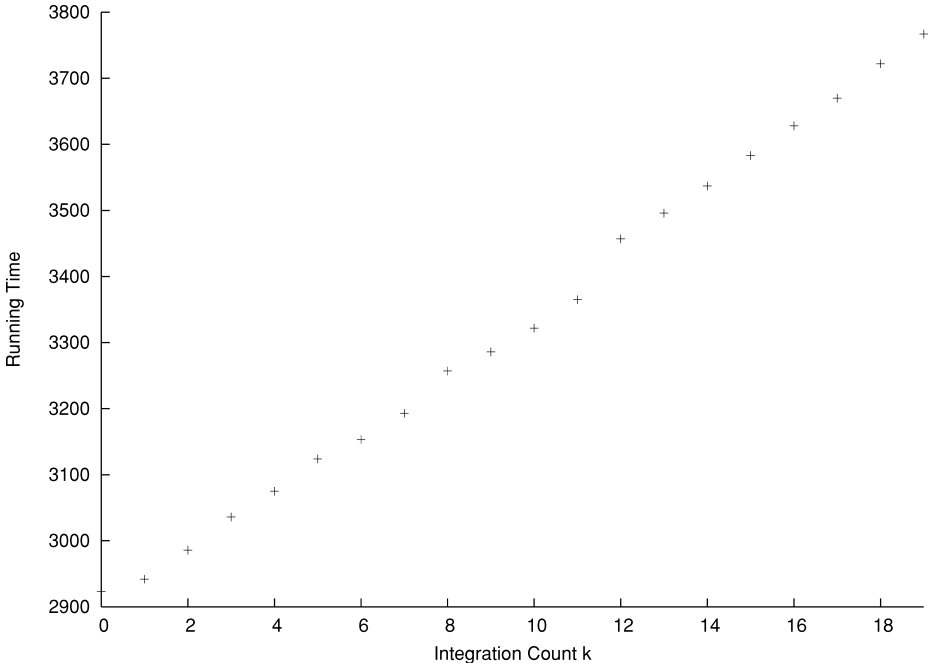

Fig. 5. Time for AR(1) simulations, varying k.

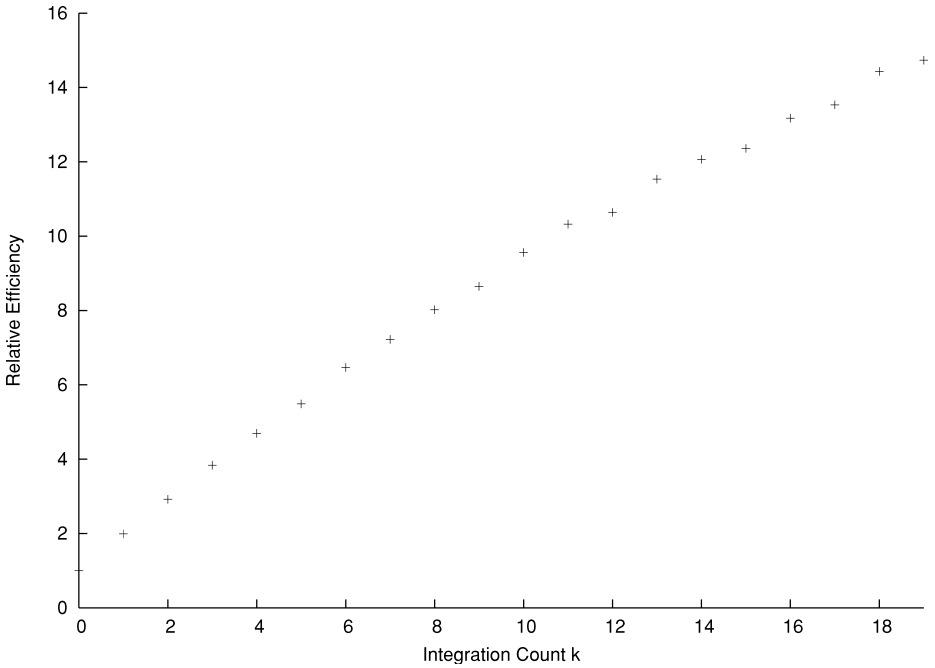

Fig. 6. Relative efficiency for AR(1) simulations, varying k.

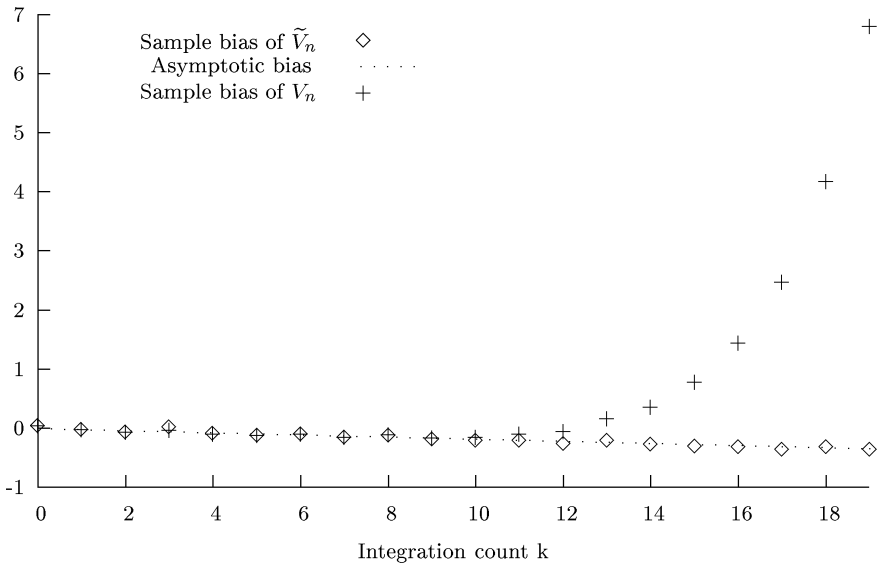

Fig. 7. Comparison of sample bias for $\widetilde{V}_n$ and V_n and asymptotic bias for AR(1) simulations, varying k, no batching.

process with unit rate, and the service times are independent, exponentially distributed with mean 0.8. For traffic intensity ρ,

$$\sigma^2 = \rho(2 + 5\rho - 4\rho^2 + \rho^3)(1 - \rho)^{-4};$$

see Whitt [1989]. For our model with $\rho = 0.8$, $\sigma^2 = 1976$. From calculations in Song and Schmeiser [1995, Table 4], $\lambda = 50,004$.

As in the case of the previous model, we ran 10,000 independent replications with run length $n = 200{,}000$ and batch count $b = 20$. Again, two independent simulations were run. The time to run the experiments increased at a constant rate from 4362 seconds for $k = 0$ to 5398 seconds for $k = 19$; the plot, which we omit, is similar to Figure 5. Figure 8 plots the sample variance for k ranging from 0 to 19, and also the limiting variances based on (12). In addition to the sample variance based on the run length $n = 200{,}000$, we also ran experiments of 1,000 replications with $n = 2{,}000{,}000$; the results are shown as "Long run length sample." Note that the sample variance from the simulations is significantly larger than the value given by (12) for the shorter run length, and the sample variances are closer to the asymptotic values with the longer run length. Figure 9 plots the bias of the estimator $\widetilde{V}_n$, which appears to increase approximately linearly with integration count k, as was the case for the AR(1) model.

Figure 10 plots the relative efficiency of the estimators for different k (with respect to $k = 0$, batch means). As in Figure 8, we plot the results for the two different run lengths. Note that with the shorter run length, the maximum efficiency improvement is only by a factor of 3.4. With the longer run length, maximum efficiency increases by a factor of 9.5 relative to batch means. For the longer run length, efficiency is roughly doubled with a single integrated path.

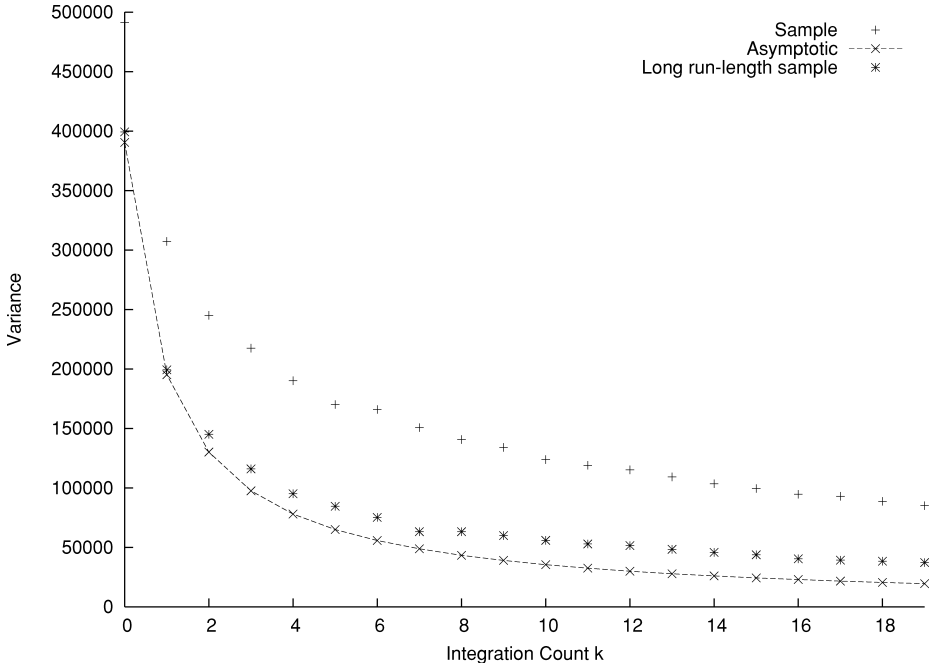

Fig. 8. Sample and asymptotic variance for M/M/1 simulations, varying k.

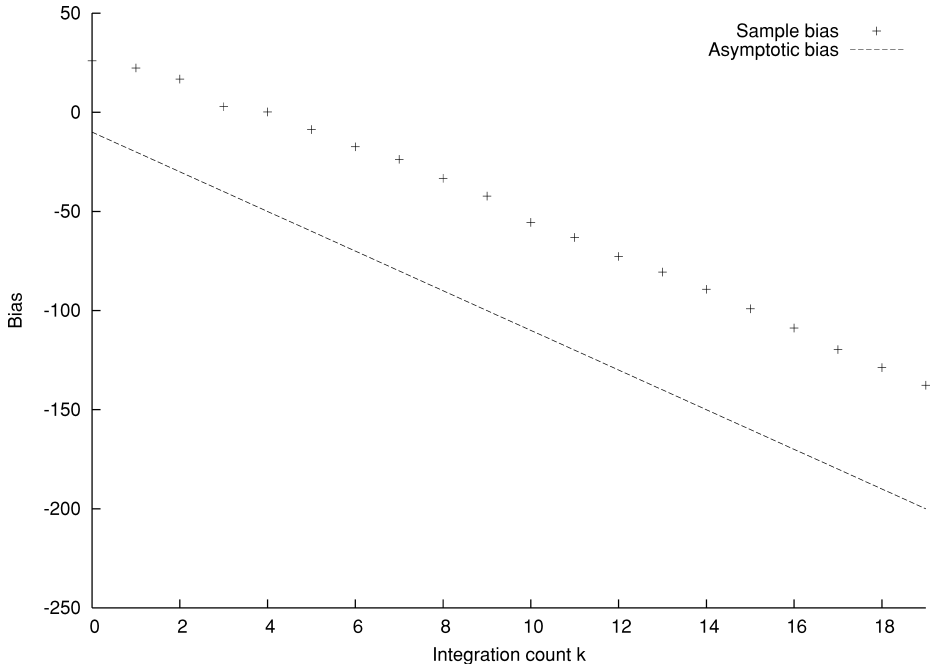

Fig. 9. Sample and asymptotic bias for M/M/1 simulations, varying k.

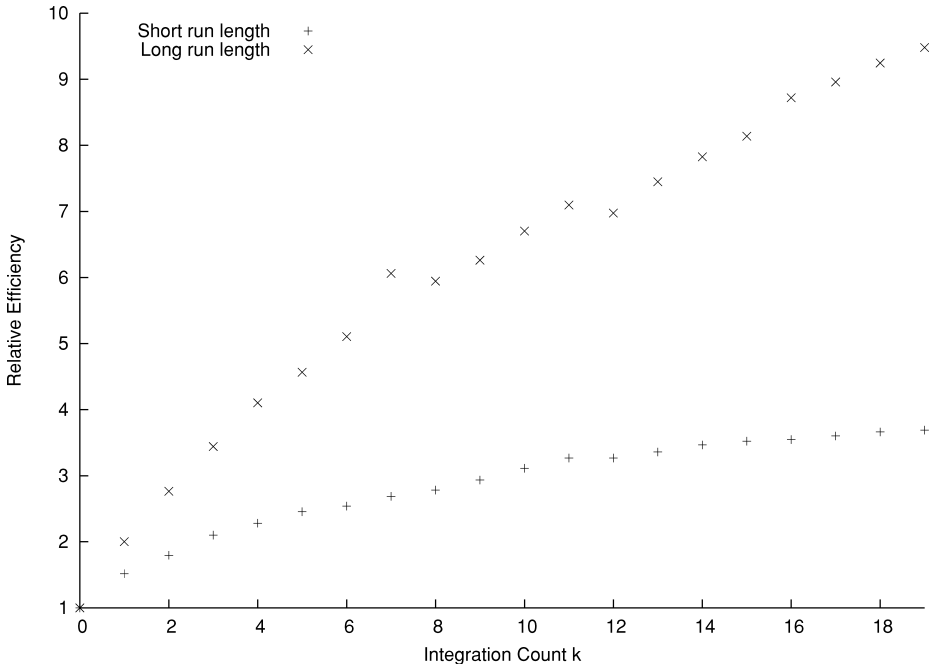

Fig. 10. Relative efficiency for M/M/1 simulations, varying k.

Table II. AR(1) Model Experiments with Short Run Lengths

	$n = 1{,}000$			$n = 5{,}000$			$n = 10{,}000$		
k	bias	var	mse	bias	var	mse	bias	var	mse
0	−3.57	24.33	37.11	−0.76	33.31	33.88	−0.38	34.83	34.98
1	−6.46	9.03	50.74	−1.44	15.57	17.64	−0.74	16.65	17.20
2	−8.62	4.64	78.98	−2.15	9.62	14.24	−1.08	10.83	12.01
3	−10.21	2.78	107.09	−2.82	6.83	14.79	−1.44	7.81	9.89
4	−11.42	1.86	132.28	−3.48	5.10	17.21	−1.79	6.09	9.32
5	−12.34	1.30	153.78	−4.11	4.06	20.96	−2.13	4.90	9.43
6	−13.08	0.97	172.01	−4.73	3.23	25.59	−2.48	4.10	10.27

9.3 Short Run Lengths

In this section we consider the same AR(1) and M/M/1 models, but with shorter run lengths. We begin with the AR(1) model, and consider run lengths of $n = 1,000$, $n = 5{,}000$, and $n = 10{,}000$. The results, based on 100,000 independent replications, are shown in Table II. Since we are running parallel simulations with 20 batches, these run lengths correspond to the rows with $m = 100$, $m = 500$, and $m = 1{,}000$, respectively, in Tables 2 and 3 of Alexopoulos et al. [2006], so we can compare our results with the results of the experiments reported there. It is clear that as k increases, the variance tends to be lower and the bias much higher, for the estimator proposed here compared with the range of alternatives considered in Alexopoulos et al. [2006]. For the shorter run lengths the bias is extremely large.

Taking into account the time needed to run the experiments, for $n = 1{,}000$, $k = 0$ gave the greatest efficiency. For $n = 5{,}000$, $k = 2$ improved efficiency by a

Table III. M/M/1 Model Experiments with Short Run Lengths

	$n = 5{,}120$			$n = 10{,}240$			$n = 20{,}480$		
k	bias	var $\times 10^6$	mse $\times 10^6$	bias	var $\times 10^6$	mse $\times 10^6$	bias	var $\times 10^6$	mse $\times 10^6$
0	−409	1.63	1.79	−205	1.56	1.60	−104	1.16	1.17
1	−693	0.67	1.15	−385	0.88	1.03	−202	0.81	0.85
2	−902	0.36	1.17	−547	0.52	0.82	−290	0.59	0.68
3	−1052	0.23	1.33	−679	0.36	0.83	−380	0.45	0.59
4	−1175	0.15	1.53	−793	0.26	0.89	−460	0.35	0.56
5	−1265	0.10	1.70	−891	0.19	0.98	−539	0.27	0.56
6	−1339	0.08	1.87	−977	0.14	1.10	−608	0.22	0.59

factor of 2.34 over batch means, and for $n = 10{,}000$, $k = 4$ improved efficiency by a factor of 3.48.

For the M/M/1 model, we used run lengths of $n = 5{,}120$, $n = 10{,}240$, and $n = 20{,}480$. These run lengths correspond to the first three rows of Tables 4 and 5 of Alexopoulos et al. [2006]. The experimental results are shown in Table III. As in the case of the AR(1) model, the bias is very large for short run lengths or large k. For $n = 5{,}120$, the maximum efficiency is achieved with $k = 1$, which gives 1.48 times the efficiency of $k = 0$ (batch means). For $n = 10{,}240$, the efficiency improvement for $k = 2$ is by a factor of 1.90, and for $n = 20{,}480$ the efficiency improvement for $k = 4$ is by a factor of 2.00.

Note that the approach we have adopted of running parallel simulations increases the bias. Since

$$E\big(Y_0^1 - Y_0^2\big)\big(Y_m^1 - Y_m^2\big) = 2E\big(Y_0^1 Y_m^1\big),$$

one would expect the bias to be about twice as large (based on the bias approximation in Section 8). On the other hand, since we are not using the sample mean to center the path, we have an additional degree of freedom. Therefore, one would expect the estimator with $k = 0$ to have a bit lower variance and about twice the bias as the batch means estimator used in Alexopoulos et al. [2006]. Tables II and III are consistent with these observations.

Based on these experiments, it appears that for short run lengths, k should be near or equal to 0 (see the next subsection on choice of k). The proposed estimator has bias of order $(k + 1)/n$, in contrast to some of the estimators considered in Alexopoulos et al. [2006], which have bias of order $o(1/n)$. The construction of low bias versions of the proposed estimator will be considered in a separate paper.

9.4 Choice of k

As seen from the analytic results and the experiments, bias increases and variance decreases with increasing k. One way of posing the question of what is the "best" k is to choose it to minimize mean-squared error. If we approximate the mean-squared error using the first-order bias approximation from Section 8 and the asymptotic variance from (4), we get the expression

$$mse(k) \approx \frac{2\sigma^4}{b(k+1)} + \frac{4\lambda^2(k+1)^2}{n^2}, \tag{17}$$

Table IV. AR(1) Model Experiments with No Batching

n	bias	var $\times 10^6$	mse $\times 10^6$
1,000	−3.47	25.97	38.04
5,000	−0.75	33.98	34.54
10,000	−0.39	34.79	34.95

Table V. M/M/1 Model Experiments with No Batching

n	bias	var $\times 10^6$	mse $\times 10^6$
5,120	−410.6	1.66	1.83
10,240	−229.2	1.56	1.61
20,480	−126.9	1.25	1.27

where b is the number of batches and n is the batch size. This suggests choosing k as the nearest integer to

$$\left(\frac{\sigma^2}{2\lambda\sqrt{b}}n\right)^{2/3} - 1. \tag{18}$$

Substituting the values for the AR(1) process, we get optimal values of k of 0, 2, and 4 for $n = 1{,}000$, $n = 5{,}000$, and $n = 10{,}000$, respectively. Comparing these with the results in Table II, we see that these are exactly the values of k giving the smallest sample mean-squared error. Substituting values for the M/M/1 model into (18), we obtain $k = 4$, 7, and 12 for run lengths $n = 5{,}120$, $n = 10{,}240$, and $n = 20{,}480$, respectively. From Table III, the values of k that minimize the mean-squared error in the experiments are 1, 2, and 5. The discrepancy is similar to that shown in Figures 8 and 9, where we see that the sample bias and variance differ significantly from the asymptotic values for the M/M/1 model.

These calculations are similar to those in Song and Schmeiser [1995] and Chien et al. [1997] for finding optimal batch sizes for the batch means method. Note that from an asymptotic point of view, we want $k = O(\sqrt{n})$ so that the cost of computing the variance estimate does not dominate the cost of the simulation; this choice does not involve considerations of bias or variance, which suggest $k = \Theta(n^{2/3})$ is better.

9.5 Integrated Path Estimator Without Batching

We repeated the experiments with $k = 20$ and no batching, to compare with the standard batch means method with 20 batches. These two estimators have the same limiting distribution. In all experiments we used 100,000 independent replications.

The results for the AR(1) model are shown in Table IV, and the results for the M/M/1 model are shown in Table V. Comparing these results with Table II and Table III, we see that the batch means estimator with 20 batches and the integrated path estimator with $k = 20$ and no batching, have similar bias and variance.

10. CONCLUSIONS

This article has described a method for estimation of the steady-state variance parameter that is intended for simulations for which the run length is not known in advance. The storage requirement is proportional to a parameter k, the integration path count. As k increases, the asymptotic efficiency of the estimator increases as run length increases.

The estimator is a particular example of a weighted estimator; that is, the estimator is based on integrating the path multiplied by a weighting function. The particular weighting function, corresponding to multiple path integrations, is chosen for computational efficiency. If the simulation run length is known in advance, then it is feasible to consider a range of weighting functions geared towards different objectives. In Alexopoulos et al. [2006], several such choices are shown to significantly reduce bias. The choice of weighting function considered in this article results in relatively large bias and low variance.

While asymptotic efficiency improves with increasing integration count k, k must be small for short run lengths. To minimize mean-square error, k should be $\Theta(n^{2/3})$ as run length n increases. However, such a choice of k implies that the computational cost of the output analysis dominates the computational cost of the simulation. If k is chosen $O(n^{1/2})$, then the output analysis cost is at most comparable to the cost of the simulation.

APPENDIX

This appendix contains the proof of Lemma 8.2.

For nonnegative integer α

$$\widetilde{W}_n^{\alpha} = n^{-1/2} \sum_{i=1}^{n} Y_i \frac{\left(1 - \frac{i-1}{n}\right)^{\alpha}}{\alpha!}.$$

Therefore,

$$\begin{aligned}
E\left(\widetilde{W}_n^{\alpha}\widetilde{W}_n^{\beta}\right) &= n^{-1}E\sum_{i=1}^{n} Y_i \frac{\left(1 - \frac{i-1}{n}\right)^{\alpha}}{\alpha!} \sum_{j=1}^{n} Y_j \frac{\left(1 - \frac{j-1}{n}\right)^{\beta}}{\beta!} \\
&= n^{-1}E\sum_{i=1}^{n} Y_i^2 \frac{\left(1 - \frac{i-1}{n}\right)^{\alpha+\beta}}{\alpha!\beta!} + n^{-1}E\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} Y_i Y_j \frac{\left(1 - \frac{i-1}{n}\right)^{\alpha}}{\alpha!} \frac{\left(1 - \frac{j-1}{n}\right)^{\beta}}{\beta!} \\
&\quad + n^{-1}E\sum_{i=2}^{n}\sum_{j=1}^{i-1} Y_i Y_j \frac{\left(1 - \frac{i-1}{n}\right)^{\alpha}}{\alpha!} \frac{\left(1 - \frac{j-1}{n}\right)^{\beta}}{\beta!} \\
&= n^{-1}\gamma_0 \sum_{i=1}^{n} \frac{\left(1 - \frac{i-1}{n}\right)^{\alpha+\beta}}{\alpha!\beta!} + n^{-1}\sum_{k=1}^{n-1}\gamma_k \sum_{i=1}^{n-k} \frac{\left(1 - \frac{i-1}{n}\right)^{\alpha}}{\alpha!} \frac{\left(1 - \frac{i+k-1}{n}\right)^{\beta}}{\beta!} \\
&\quad + n^{-1}\sum_{k=1}^{n-1}\gamma_k \sum_{i=1}^{n-k} \frac{\left(1 - \frac{i-1}{n}\right)^{\beta}}{\beta!} \frac{\left(1 - \frac{i+k-1}{n}\right)^{\alpha}}{\alpha!}
\end{aligned}$$

$$= \frac{n^{-1}}{\alpha!\beta!}\left(\gamma_0\sum_{i=1}^{n}\left(1-\frac{i-1}{n}\right)^{\alpha+\beta}+\sum_{k=1}^{n-1}\gamma_k\sum_{i=1}^{n-k}\left(1-\frac{i-1}{n}\right)^{\alpha}\left(1-\frac{i+k-1}{n}\right)^{\beta}\right.$$
$$\left.+\sum_{k=1}^{n-1}\gamma_k\sum_{i=1}^{n-k}\left(1-\frac{i-1}{n}\right)^{\beta}\left(1-\frac{i+k-1}{n}\right)^{\alpha}\right).$$

Now

$$n^{-1}\sum_{i=1}^{n}\left(1-\frac{i-1}{n}\right)^{\alpha+\beta}=\frac{1}{\alpha+\beta+1}+\frac{I_{(\alpha+\beta>0)}}{2n}+o(1/n)$$

and

$$n^{-1}\sum_{i=1}^{n-k}\left(1-\frac{i-1}{n}\right)^{\alpha}\left(1-\frac{i+k-1}{n}\right)^{\beta}$$
$$= \int_{x=0}^{1-k/n}(1-x)^{\alpha}(1-x-k/n)^{\beta}dx+\frac{(1-k/n)^{\beta}-I_{(\beta=0)}(k/n)^{\alpha+\beta}}{2n}+o(1/n)$$
$$= \sum_{j=0}^{\beta}\binom{\beta}{j}(-1)^{\beta-j}\frac{(k/n)^{\beta-j}-(k/n)^{\alpha+\beta+1}}{\alpha+j+1}$$
$$+\frac{1}{2n}\sum_{j=0}^{\beta}\binom{\beta}{j}(-1)^{\beta-j}((k/n)^{\beta-j}-(k/n)^{\alpha+\beta})+o(1/n).$$

Therefore,

$$E\left(\widetilde{W}_n^{\alpha}\widetilde{W}_n^{\beta}\right) = \frac{1}{\alpha!\beta!}\left(\gamma_0\left(\frac{1}{\alpha+\beta+1}+\frac{I_{(\alpha+\beta>0)}}{2n}+o(1/n)\right)\right.$$
$$+\sum_{k=1}^{n-1}\gamma_k\left(\sum_{j=0}^{\beta}\binom{\beta}{j}(-1)^{\beta-j}\frac{(k/n)^{\beta-j}-(k/n)^{\alpha+\beta+1}}{\alpha+j+1}\right.$$
$$+\frac{1}{2n}\sum_{j=0}^{\beta}\binom{\beta}{j}(-1)^{\beta-j}((k/n)^{\beta-j}-(k/n)^{\alpha+\beta})$$
$$+\sum_{j=0}^{\alpha}\binom{\alpha}{j}(-1)^{\alpha-j}\frac{(k/n)^{\alpha-j}-(k/n)^{\alpha+\beta+1}}{\beta+j+1}$$
$$\left.\left.+\frac{1}{2n}\sum_{j=0}^{\alpha}\binom{\alpha}{j}(-1)^{\alpha-j}((k/n)^{\alpha-j}-(k/n)^{\alpha+\beta})+o(1/n)\right)\right).$$

Using the fact that for $\theta > 1$,

$$\sum_{k=1}^{n-1}\gamma_k\left(\frac{k}{n}\right)^{\theta}=o(1/n),$$

we obtain

$$
\begin{aligned}
E\left(\widetilde{W}_n^{\alpha}\widetilde{W}_n^{\beta}\right) &= \frac{1}{\alpha!\beta!}\left(\gamma_0\left(\frac{1}{\alpha+\beta+1}+\frac{I_{(\alpha+\beta>0)}}{2n}+o(1/n)\right)\right.\\
&+ \sum_{k=1}^{n-1}\gamma_k\left(\frac{1}{\alpha+\beta+1}-\beta\frac{(k/n)}{\alpha+\beta}-(k/n)^{\alpha+\beta+1}\sum_{j=0}^{\beta}\binom{\beta}{j}(-1)^{\beta-j}\frac{1}{\alpha+j+1}\right.\\
&+ \frac{1}{2n}\left(1-\beta(k/n)-(k/n)^{\alpha+\beta}\sum_{j=0}^{\beta}\binom{\beta}{j}(-1)^{\beta-j}\right)\\
&+ \frac{1}{\alpha+\beta+1}-\alpha\frac{(k/n)}{\alpha+\beta}-(k/n)^{\alpha+\beta+1}\sum_{j=0}^{\alpha}\binom{\alpha}{j}(-1)^{\alpha-j}\frac{1}{\beta+j+1}\\
&+ \left.\left.\frac{1}{2n}\left(1-\alpha(k/n)-(k/n)^{\alpha+\beta}\sum_{j=0}^{\alpha}\binom{\alpha}{j}(-1)^{\alpha-j}\right)+o(1/n)\right)\right)\\
&= \frac{1}{\alpha!\beta!}\left(\frac{\sigma^2}{\alpha+\beta+1}-\frac{1}{n}\left(I_{(\alpha+\beta>0)}+2I_{(\alpha+\beta=0)}\right)\sum_{k=1}^{n-1}k\gamma_k+\frac{1}{2n}\sigma^2 I_{(\alpha+\beta>0)}+o(1/n)\right)\\
&= \frac{1}{\alpha!\beta!}\left(\frac{\sigma^2}{\alpha+\beta+1}-\frac{1}{n}\left(I_{(\alpha+\beta>0)}+2I_{(\alpha+\beta=0)}\right)\lambda+\frac{\sigma^2 I_{(\alpha+\beta>0)}}{2n}+o(1/n)\right).
\end{aligned}
$$

This completes the proof of Lemma 8.2.

REFERENCES

ALEXOPOULOS, C., ARGON, N., GOLDSMAN, D., STEIGER, N., TOKOL, G., AND WILSON, J. 2006. Efficient computation of overlapping variance estimators for simulation. *To Appear in INFORMS J. Comput*.

BILLINGSLEY, P. 1999. *Convergence of Probability Measures*, second ed. John Wiley & Sons, New York, New York.

CHIEN, C., GOLDSMAN, D., AND MELAMED, B. 1997. Large-sample results for batch means. *Manage. Sci. 43*, 1288–1295.

KNUTH, D. 1997. *The Art of Computer Programming, Volume 1: Fundamental Algorithms*, third ed. Addison-Wesley, Reading, Massachusetts.

MEKETON, M. S. AND SCHMEISER, B. W. 1984. Overlapping batch means: something for nothing? In *Proceedings of the 1984 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 227–230.

NOVAK, E. AND RITTER, K. 1993. Some complexity results for zero finding for univariate functions. *J. Complex. 9*, 15–40.

SCHRUBEN, L. W. 1983. Confidence interval estimation using standardized time series. *Oper. Res. 31*, 1090–1108.

SONG, W.-M. T. AND SCHMEISER, B. W. 1995. Optimal mean-squared-error batch sizes. *Manage. Sci. 41*, 110–123.

WHITT, W. 1989. Planning queueing simulations. *Manage. Sci. 35*, 1341–1366.

YEH, Y. AND SCHMEISER, B. W. 2000. Simulation output analysis via dynamic batch means. In *Proceedings of the 2000 Winter Simulation Conference*. Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 637–645.

Received August 2005; revised November 2006 and January 2007; accepted January 2007