

ABSTRACT

TAFAZZOLI YAZDI, ALI. Skart: A Skewness- and Autoregression-Adjusted Batch-Means Procedure for Simulation Analysis. (Under the direction of Professor James R. Wilson).

We discuss Skart, an automated batch-means procedure for constructing a skewness- and autoregression-adjusted confidence interval (CI) for the steady-state mean of a simulation output process in either discrete time (i.e., observation-based statistics) or continuous time (i.e., time-persistent statistics). Skart is a sequential procedure designed to deliver a CI that satisfies user-specified requirements concerning not only the CI's coverage probability but also the absolute or relative precision provided by its half-length. Skart exploits separate adjustments to the half-length of the classical batch-means CI so as to account for the effects on the distribution of the underlying Student's t -statistic that arise from skewness (nonnormality) and autocorrelation of the batch means. The skewness adjustment is based on a modified Cornish-Fisher expansion for the classical batch-means Student's t -ratio, and the autocorrelation adjustment is based on an autoregressive approximation to the batch-means process for sufficiently large batch sizes. Skart also delivers a point estimator for the steady-state mean that is approximately free of initialization bias. The duration of the associated warm-up period (i.e., the statistics clearing time) is based on iteratively applying von Neumann's randomness test to spaced batch means with progressively increasing batch sizes and interbatch spacer sizes. In an experimental performance evaluation involving a wide range of test processes, Skart compared favorably with other simulation analysis methods—namely, its predecessors ASAP3, WASSP, and SBatch as well as ABATCH, LBatch, the Heidelberger-Welch procedure, and the Law-Carson procedure. Specifically, Skart exhibited competitive sampling efficiency and substantially closer conformance to the given CI coverage probabilities than the other procedures.

Also presented is a nonsequential version of Skart, called N-Skart, in which the user supplies a single simulation-generated series of arbitrary length and specifies a coverage probability for a CI based on that series. In the same set of test processes previously mentioned and for a range of data-set sizes, N-Skart also achieved close conformance to the specified CI coverage probabilities.

Skart: A Skewness- and Autoregression-Adjusted Batch-Means
Procedure for Simulation Analysis

by
Ali Tafazzoli Yazdi

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Industrial Engineering

Raleigh, North Carolina

2009

APPROVED BY:

Dr. David A. Dickey

Dr. Yahya Fathi

Dr. Emily K. Lada

Dr. Stephen D. Roberts

Dr. Natalie M. Steiger

Dr. James R. Wilson
Chair of Advisory Committee

BIOGRAPHY

Ali Tafazzoli Yazdi is a Ph.D. student in the Edward P. Fitts Department of Industrial and Systems Engineering at North Carolina State University. He was born on August 22, 1978, in Tehran, Iran. In May 2001, he graduated from Sharif University of Technology with a bachelor of science degree in industrial engineering. To pursue advanced degrees, he came to the United States in January 2003. He received a master of science degree with a dual major in industrial engineering and operations research from North Carolina State University in December 2004. His research interests include simulation modeling, analysis, and optimization. While working on his Ph.D., he also served as a research assistant and was a student intern in the Operations Research Department of the SAS Institute.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Dr. James R. Wilson, for his encouragement, thoughtful advice, kindness, and patience throughout my Ph.D. study. His guidance not only was a good inspiration to me in the past, but also will be a very valuable asset to me in the future. Special thanks go to Dr. Steve Roberts for serving on my committee and for all the enlightening advice he has given me over the years. I thank Dr. Yahya Fathi for serving on my committee and for all the guidance he has given me since I have been at NC State. My thanks also go to Dr. David Dickey for serving on my committee and for his constructive input. I am particularly indebted to Dr. Natalie Steiger and Dr. Emily Lada for serving on my committee and for the extensive insight they provided in developing Skart. Their prior work on ASAP3, WASSP, and SBatch were fundamental to the development of this dissertation. I also thank Dr. Charlie Smith and Dr. Len Stefanski for serving at different times as the Graduate School representative on my committee. Finally, I dearly owe my parents, my brother and sister, and my friends for their emotional and moral support during the time when I was working on this dissertation.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	ix
1 Introduction	1
1.1 Motivation and Problem Statement	2
1.2 Scope and Objectives of Research	5
1.3 Organization of the Dissertation	7
2 Literature Review	9
2.1 Overview of Methods for Steady-State Analysis	9
2.1.1 Replication/Deletion	9
2.1.2 Regenerative Method	11
2.1.3 Standardized Time Series	14
2.1.4 Spectral Analysis	17
2.1.5 Method of Autoregressive Representation	19
2.2 Review of Automated Simulation Output Analysis Procedures	21
2.2.1 ASAP3	21
2.2.2 LBATCH and ABATCH	23
2.2.3 Law-Carson Procedure	25
2.2.4 SBatch	26
2.2.5 WASSP	28
2.2.6 Heidelberger-Welch Procedure	29
2.3 Relative Performance of Automated Simulation Output Analysis Procedures	32
2.4 Skewness Adjustment to the Student's t -Statistic for Independent Observations	32
2.4.1 Johnson's Modification of Student's t -Statistic	33
2.4.2 Hall's Modification of Student's t -Statistic	36
2.4.3 Willink's Modification of Student's t -Statistic	37
3 Skart: A Skewness- and Autoregression-Adjusted Procedure for Steady-State Simulation Analysis	39
3.1 Overview of Skart	39

3.2	Formal Algorithmic Statement of Skart	43
3.3	Detailed Description of Steps in Skart	43
3.3.1	Initialization Step	43
3.3.2	The Test for Randomness	50
3.3.3	Autocorrelation Adjustment for the Variance Estimator	53
3.3.4	Skewness Adjustment to Student's t -Statistic	54
3.3.5	Fulfilling the Precision Requirement	56
3.4	N-Skart: A Nonsequential Version of Skart	57
4	Performance Evaluation of Skart	64
4.1	Performance of Skart	65
4.1.1	The $M/M/1$ Queue-Waiting-Time Process	65
4.1.2	The $M/M/1$ Number-in-Queue Process	71
4.1.3	The $M/H_2/1$ Queue-Waiting-Time Process	72
4.1.4	The First-Order Autoregressive (AR(1)) Process	73
4.1.5	The AR(1)-to-Pareto (ARTOP) Process	75
4.1.6	The $M/M/1/LIFO$ Queue-Waiting-Time Process	77
4.1.7	The $M/M/1/SIRO$ Queue-Waiting-Time Process	78
4.1.8	The $M/M/1/M/1$ Queue-Waiting-Time Process	79
4.1.9	Two-State Discrete-Time Markov Chains	80
4.2	Performance Evaluation of N-Skart	83
4.3	Efficiency Analysis	85
5	Conclusions and Future Research	90
5.1	Main Conclusions of the Research	90
5.2	Directions for Future Research	92
	Bibliography	94
	Appendices	99
	Appendix A Asymptotic Properties of Batch Means from Stationary, Invertible ARMA Processes with Increasing Batch Size	99
A.1	Motivation, Assumptions, and Basic Results about Batch-Means Processes	100
A.2	Proof of (A.17) and (A.32) When the Original (Unbatched) Process Is ARMA	105
A.3	Consequences of Assumption (A.32) When the Batch-Means Process Is ARMA	110

A.4	Proof of the Main Result When the Batch-Means Process Is ARMA	112
Appendix B Distribution of First Regular $M/M/1$ Queue Waiting Time X_1 given $N(0) = c$ Initial Nonregular Customers		118
Appendix C Skart and N-Skart User's Manual		121
C.1	Downloading Skart and N-Skart	121
C.2	Running Skart in Visual Basic	121
C.3	Running N-Skart in Visual Basic	123
C.4	Visual Basic Codes for Skart and N-Skart Functions and Subroutines	129

LIST OF TABLES

Table 4.1 Performance of Skart in the $M/M/1$ queue-waiting-time process for different levels of server utilization (SU) and for 90% and 95% CIs for the no precision case	66
Table 4.2 Performance of Skart, SBatch, WASSP, and ASAP3 in the $M/M/1$ queue-waiting-time process with 90% server utilization and empty-and-idle initial condition	67
Table 4.3 Performance of Skart, SBatch, and ASAP3 in the $M/M/1$ queue-waiting-time process with 90% server utilization and 113 customers initially in the system	69
Table 4.4 Performance of Skart, SBatch, WASSP, and ASAP3 in the $M/M/1$ queue-waiting-time process with 80% server utilization and empty-and-idle initial condition	70
Table 4.5 Performance of Skart in the $M/M/1$ number-in-queue process with 80% and 90% server utilization computed over 1,000 independent replications	72
Table 4.6 Performance of Skart, SBatch, WASSP, and ASAP3 in the $M/H_2/1$ queue-waiting-time process with 80% server utilization and empty-and-idle initial condition	73
Table 4.7 Performance of Skart, SBatch, WASSP, and ASAP3 in the AR(1) process	74
Table 4.8 Performance of Skart, SBatch, WASSP, and ASAP3 in the ARTOP process	76
Table 4.9 Performance of Skart, SBatch, WASSP, and ASAP3 in the $M/M/1/LIFO$ queue-waiting-time process with 80% server utilization	78
Table 4.10 Performance of Skart in the $M/M/1/SIRO$ queue-waiting-time process with 80% server utilization based on 1,000 replications of the process	79

Table 4.11 Performance of Skart, ASAP3, and WASSP in the $M/M/1/M/1$ queue-waiting-time process with 80% server utilization	80
Table 4.12 Performance of Skart, ASAP3, and SBatch in the two-state discrete-time Markov chain with marginal skewness $\mathcal{B}(1) = 0.0$ based on 1,000 replications	81
Table 4.13 Performance of Skart, ASAP3, and SBatch in the two-state discrete-time Markov chain with marginal skewness $\mathcal{B}(2) = 2.1$ based on 1,000 replications	82
Table 4.14 Performance of Skart, ASAP3, and SBatch in the two-state discrete-time Markov chain with marginal skewness $\mathcal{B}(3) = 4.8$ based on 1,000 replications	83
Table 4.15 Performance of N-Skart for selected test problems based on 1,000 replications	84
Table 4.16 Asymptotic efficiency of Skart and ASAP3 in the sense of Chow and Robbins based on 1,000 replications of each test process	87
Table 4.17 Asymptotic efficiency of WASSP and SBatch in the sense of Chow and Robbins based on 1,000 replications of each test process	88
Table C.1 Input arguments required for the SkarProcedure subroutine	122
Table C.2 Input arguments required for the NSkarProcedure subroutine	124

LIST OF FIGURES

Figure 3.1 High-level flowchart of Skart.....	40
Figure 3.2 Algorithmic statement of Skart	44
Figure 3.3 High-level flowchart of N-Skart	58
Figure 3.4 Algorithmic statement of N-Skart.....	59
Figure C.1 Screen-shot of the Notepad output window showing the result of applying Skart procedure to an $M/M/1$ queue-waiting-time process	123
Figure C.2 Screen-shot of the Notepad output window showing the result of applying N-Skart procedure to an $M/M/1$ queue-waiting-time process when the user decides to quit the procedure due to insufficient data	126
Figure C.3 Screen-shot of the Notepad output window showing the result of applying N-Skart procedure to an $M/M/1$ queue-waiting-time process when the user chooses to continue with delivering a CI knowing that the randomness test could not be passed.....	127
Figure C.4 Screen-shot of the Notepad output window showing the result of applying N-Skart procedure to an $M/M/1$ queue-waiting-time process when there is enough observation available to deliver a CI.....	128

Chapter 1

Introduction

Discrete-event simulation concerns the modeling of a system as it evolves over time by a numerical (computer-based) representation in which the state variables change at only a countable number of points in simulation time. This technique can be used to model a real or proposed system so that the behavior of the system can be studied under specific conditions. Most simulations can be classified as either finite-horizon or steady-state. Finite-horizon (terminating) simulation models are ended at a specific time or by the occurrence of a specific condition. On the other hand, steady-state (nonterminating) simulations operate (at least conceptually) into the indefinite future; and in this case interest centers on long-run average performance. Analysis of the outputs generated by steady-state simulation models is the focus of this research.

Let $\{X_1, X_2, \dots\}$ be the discrete-time stochastic process representing the output from a single run of a nonterminating simulation. If the simulation is in steady-state operation, then the random variables X_1, X_2, \dots will all have the same steady-state distribution function, $F_X(x) = \Pr\{X_i \leq x\}$ for all real x and for $i = 1, 2, \dots$. In discrete-event stochastic simulation, we are usually interested in constructing point and confidence-interval (CI) estimates for some parameter, or characteristic, of the steady-state distribution function $F_X(x)$ of a discrete- or continuous-time output process. In this research, we are primarily interested in estimating the steady-state mean $\mu_X = E[X] = \int_{-\infty}^{+\infty} x dF_X(x)$ of a selected univariate simulation output process $\{X_i : i = 1, 2, \dots\}$ generated by a single prolonged simulation run; and we limit the discussion to output processes for which $E[X_i^2] < \infty$ so that the process mean μ_X and the process variance $\sigma_X^2 = \text{Var}[X] = E[(X_i - \mu_X)^2]$ are well defined. Given an output sequence of length n generated by a simulation in steady-state operation, we know that a natural

point estimator of μ_X is the sample mean

$$\bar{X} = \bar{X}(n) = n^{-1} \sum_{i=1}^n X_i.$$

The inherent variability associated with this point estimator of μ_X leads us to the idea of estimating the variance of \bar{X} and of constructing a CI estimator for μ_X . An estimator of the process variance σ_X^2 is the sample variance,

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2.$$

When we require some indication of precision of the point estimator \bar{X} , if the X_i 's are independent and identically distributed (i.i.d.) according to $N(\mu_X, \sigma_X^2)$, a normal distribution with mean μ_X and variance σ_X^2 , then we typically construct a valid CI for μ_X as follows,

$$\bar{X} \pm t_{1-\alpha/2, n-1} \frac{S}{\sqrt{n}}, \quad (1.1)$$

with a user-specified coverage probability $1 - \alpha$, where $0 < \alpha < 1$. In the above equation, $t_{1-\alpha/2, n-1}$ is the $(1 - \alpha/2)$ -quantile of Student's t -distribution with $n - 1$ degrees of freedom. In general, an appropriate CI for μ_X should satisfy two criteria: (a) it is narrow enough to be informative; and (b) its actual probability of covering the point μ_X is equal (or at least sufficiently close) to the nominal level $1 - \alpha$. In general we say a CI is *valid* for all stochastic processes satisfying certain conditions if criterion (b) is satisfied *exactly* for all $\alpha \in (0, 1)$ when the CI is computed from any process satisfying those conditions. As elaborated in the rest of this dissertation, the condition that the $\{X_i\}$ are i.i.d. $N(\mu_X, \sigma_X^2)$ is rarely satisfied by a steady-state simulation output process; and the classical CI (1.1) is frequently not valid, or even approximately valid, in steady-state simulation experiments.

1.1. Motivation and Problem Statement

Three fundamental problems arise in analyzing output from a stochastic steady-state simulation [41]. The first problem is caused by a transient in the initial sequence of responses that is due to the system's starting condition. It is usually impossible to start a simulation in steady-state operation, thereby making it necessary to do the following: (a) start the simulation in some convenient initial condition that may not be typical of steady-state operation; and (b) select the duration of the warm-up period (i.e., the data-truncation point or statistics clearing time) so that beyond the warm-up period, the

mean of each simulation-generated observation is sufficiently close to the steady-state mean. If observations prior to the end of the warm-up period are included in the analysis, then the resulting estimator may be biased [41]; and such bias in the point estimator can severely degrade not only the accuracy of the point estimator but also the probability that the associated CI will cover the steady-state mean. In other words, initialization bias can cause not only a grossly misleading simulation-based point estimate of long-run system performance but also wildly optimistic indications of the inherent accuracy and reliability of the simulation-based results, where the term *accuracy* refers to the magnitude of the estimation error and the term *reliability* refers to the CI's nominal (user-specified) coverage probability. This is known as the start-up, or initialization bias, problem.

The second fundamental problem in steady-state simulation analysis is the correlation problem caused by pronounced stochastic dependencies among successive responses generated within a single simulation run. This phenomenon complicates the construction of a CI for the steady-state mean because standard statistical methods require i.i.d. normal observations to yield a valid CI.

The third fundamental problem in steady-state simulation analysis is closely related to the second problem mentioned above—specifically, it is the nonnormality problem caused by pronounced departures from normality in the simulation-generated responses.

Several methods have been proposed for solving the previously mentioned problems in the analysis of steady-state simulation experiments. For reasons elaborated in Chapter 2, the method of nonoverlapping batch means (NBM) is by far the most widely used and most efficient output analysis procedure in practical applications for which initialization bias, correlation, or nonnormality are significant effects [32]. In the NBM method, the sequence of simulation-generated outputs $\{X_i : i = 1, \dots, n\}$ is divided into k adjacent nonoverlapping batches, each of size m , where both k and m are sufficiently large to ensure that the resulting batch means are at least approximately i.i.d. normal random variables. The sample mean for the j th batch is

$$Y_j(m) = \frac{1}{m} \sum_{i=m(j-1)+1}^{mj} X_i \quad \text{for } j = 1, \dots, k; \quad (1.2)$$

and the grand mean of the individual batch means,

$$\bar{Y} = \bar{Y}(m, k) = \frac{1}{k} \sum_{j=1}^k Y_j(m), \quad (1.3)$$

is used as the point estimator for μ_X . The objective is to construct a CI estimator for μ_X that is centered

on a point estimator as in Equation (1.3), where in practice some initial observations (or batches) may be deleted (truncated) to eliminate the effects of initialization bias.

We assume that the original simulation-generated process $\{X_i\}$ is stationary (in the strict sense) so that the joint distribution of the X_i 's is insensitive to time shifts. We also assume the process is weakly dependent—that is, X_i 's widely separated from each other in the sequence are almost independent (in sense of ϕ -mixing; see [8]) so that the lag- q covariance,

$$\gamma_X(q) \equiv E[(X_{i+q} - \mu_X) \cdot (X_i - \mu_X)] \quad \text{for } q = 0, \pm 1, \pm 2, \dots,$$

satisfies $\gamma_X(q) \rightarrow 0$ sufficiently fast as $|q| \rightarrow \infty$. These weakly dependent processes typically obey a Central Limit Theorem of the form

$$\sqrt{n} \left[\bar{X}(n) - \mu_X \right] \xrightarrow[n \rightarrow \infty]{D} N(0, \gamma_X), \quad (1.4)$$

where

$$\gamma_X \equiv \lim_{n \rightarrow \infty} n \text{Var}[\bar{X}(n)] = \sum_{q=-\infty}^{\infty} \gamma_X(q)$$

is the steady-state variance parameter (SSVP) (as distinguished from the process variance $\sigma_X^2 \equiv \text{Var}[X_i] = \gamma_X(0)$), and the symbol $\xrightarrow[n \rightarrow \infty]{D}$ denotes convergence in distribution. A sufficient condition for γ_X to exist is that $\sum_{q=-\infty}^{\infty} |\gamma_X(q)| < \infty$. General conditions under which (1.4) holds are given, for example, in Theorem 20.1 of Billingsley [8].

Although some output analysis methods attempt to estimate the steady-state variance parameter γ_X in constructing a CI for μ_X , the classical NBM method does not. A key assumption of the NBM method is that the batch size m is sufficiently large so that the batch means $\{Y_j(m) : j = 1, \dots, k\}$ are approximately i.i.d. normal variates,

$$\{Y_j(m) : j = 1, \dots, k\} \stackrel{\text{i.i.d.}}{\sim} N[\mu_X, \sigma_X^2(m)/m], \quad (1.5)$$

where the symbol $\stackrel{\text{i.i.d.}}{\sim}$ is read “is independent and identically distributed as”; and

$$\sigma_X^2(m) = m \cdot \text{Var}[Y_j(m)] = \gamma_X(0) + 2 \sum_{\ell=1}^{m-1} \left(1 - \frac{\ell}{m}\right) \gamma_X(\ell).$$

It follows that $\lim_{m \rightarrow \infty} \sigma_X^2(m) = \gamma_X$ and $\text{Var}[Y_j(m)] \approx \gamma_X/m$, provided that m is sufficiently large. Then we can apply the classical results concerning Student's t -distribution to compute a CI for μ_X by computing the sample variance of the batch means,

$$S_{m,k}^2 = \frac{1}{k-1} \sum_{j=1}^k [Y_j(m) - \bar{Y}(m,k)]^2. \quad (1.6)$$

Under the same weak dependence conditions that are sufficient to ensure (1.4) as $m \rightarrow \infty$ with k fixed so that $n = mk \rightarrow \infty$, an asymptotically valid $100(1 - \alpha)\%$ CI for μ_X is

$$\bar{Y}(m,k) \pm t_{1-\alpha/2, k-1} \frac{S_{m,k}}{\sqrt{k}}. \quad (1.7)$$

The main difficulty with conventional NBM procedures such as the Law-Carson [40] procedure, ABATCH, and LBATCH [19] is the reliable determination of an adequate batch size m so that the batch means $\{Y_j(m)\}$ are approximately uncorrelated and normal. For an elaboration of this issue, see for example Steiger et al. [54].

1.2. Scope and Objectives of Research

As explained in the previous section and also by Damerджи [17] in more detail, general conditions are known under which the batch-means method leads to a strongly consistent estimator of the variance of the sample mean and to an asymptotically valid CI for the steady-state mean. However, these conditions cannot be easily verified in practice. The principal purpose of this research is to develop a practical and efficient implementation of the batch-means method that is accessible to a large audience of potential users and that can fill the gap between theory and practice.

In this research, we develop and evaluate Skart (stands for: **S**kewness- and **a**utoregression-adjusted Student's t analysis; *skart*, or *scart*, is also an obsolete English word that means “to gather together carefully” [44])—a new procedure for steady-state simulation output analysis which can be considered as an extension of the classical method of NBM. Skart addresses the start-up problem by successively applying the randomness test of von Neumann [61] to spaced batch means with progressively increasing batch sizes and interbatch spacer sizes. Skart addresses the normality problem by a modified Cornish-Fisher expansion for the classical batch-means Student's t -ratio that incorporates a term due to Willink [60] accounting for any skewness in the set of truncated batch means that are

finally delivered. Skart addresses the correlation problem by using an autoregressive approximation to the autocorrelation function of the delivered set of truncated, adjacent batch means.

Skart takes into account not only any deterministic trends or stochastic dependencies in the sequence of batch means but also their marginal skewness so as to determine sufficiently large values of the data-truncation point (statistics clearing time) and the batch size m such that the truncated batch means approximately constitute a stationary first-order autoregressive process with mean μ_X . In our experience, this condition was much easier to achieve in practice than the conditions required to apply other batch-means procedures. Beyond the data-truncation point, we compute the sample variance of k' adjacent batch means for batch size m using Equation (1.6) with $k = k'$. We deliver an asymptotically valid $100(1 - \alpha)\%$ skewness- and autoregression-adjusted CI for μ_X having the form

$$\left[\bar{Y}(m, k') - G(L) \sqrt{\frac{AS_{m,k''}^2}{k'}}, \bar{Y}(m, k') - G(R) \sqrt{\frac{AS_{m,k''}^2}{k'}} \right], \quad (1.8)$$

where

$$G(\zeta) \equiv \frac{\sqrt[3]{1 + 6\beta(\zeta - \beta)} - 1}{2\beta}, \quad \text{with } \beta = \frac{\hat{B}_{m,k''}}{6\sqrt{k'}} \quad (1.9)$$

and $S_{m,k''}^2$ and $\hat{B}_{m,k''}$ are approximately unbiased estimators of the variance and skewness of the $\{Y_j(m)\}$ computed from k'' approximately i.i.d. spaced batch means separated by spacers having roughly the same size as the warm-up period (see (3.14) below), and finally

$$L = t_{1-\alpha/2, k''-1} \quad \text{and} \quad R = t_{\alpha/2, k''-1}. \quad (1.10)$$

(Note that in Equation (1.9), the indicated cube root $\sqrt[3]{1 + 6\beta(\zeta - \beta)}$ is understood to have the same sign as the quantity $1 + 6\beta(\zeta - \beta)$.) Thus we see that $G(L)$ and $G(R)$ are skewness-adjusted quantiles of Student's t -distribution for the left and right half-lengths of the proposed CI (this point is elaborated in §3.3.4); and the autoregression (correlation) adjustment A is applied to the sample variance $S_{m,k'}^2$ to compensate for any residual correlation between the batch means. The correlation adjustment A is computed as

$$A = \left[1 + \hat{\varphi}_{Y(m)} \right] / \left[1 - \hat{\varphi}_{Y(m)} \right], \quad (1.11)$$

where the standard estimator of the lag-one correlation of the batch means is

$$\hat{\varphi}_{Y(m)} = \widehat{\text{Corr}}[Y_j(m), Y_{j+1}(m)] = \frac{1}{k'-1} \sum_{j=1}^{k'-1} \left[Y_j(m) - \bar{Y}(m, k') \right] \left[Y_{j+1}(m) - \bar{Y}(m, k') \right] / S_{m,k'}^2. \quad (1.12)$$

The specific objectives of this research can be summarized as follows:

- Formulate and justify algorithms to address the start-up, nonnormality, and correlation problems;
- Develop Skart, an automated sequential procedure for on-the-fly or offline simulation output analysis that integrates all three algorithms mentioned above into a portable, robust software package that can be easily invoked from popular general-purpose simulation platforms such as Arena [32];
- Identify a suite of test problems to evaluate the performance of Skart, including some problems specifically designed to “stress-test” the procedure and some problems with characteristics that are typical of large-scale practical simulation studies;
- Design, perform, and analyze a set of experiments using the selected test problems to compare the performance of Skart with that of other state-of-the-art procedures for steady-state simulation analysis;
- Develop an automated nonsequential version of Skart and implement it in a portable, robust, standalone software package that can easily be applied to a group of one or more data sets each of arbitrary fixed size, delivering for each data set a CI for its associated steady-state mean with a user-specified coverage probability; and finally
- Design, perform, and analyze a set of experiments using the selected set of test problems to evaluate comprehensively the performance of the nonsequential version of Skart.

Complementing the extensive experimental performance evaluation of Skart is a rigorous theoretical justification of the asymptotic time-series models of the batch means on which Skart is based.

1.3. Organization of the Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 reviews several methods for steady-state simulation output analysis, briefly explains six well-known automated steady-state simulation output analysis procedures, and summarizes the advantages and disadvantages of each of these procedures. Chapter 2 also contains a discussion of three different skewness adjustments to Student’s t -statistic for i.i.d. observations. Chapter 3 describes the design of Skart, a new batch-means

procedure for constructing a CI on the steady-state mean of an output process; and Chapter 3 also explains the theoretical and experimental basis underlying both the sequential and nonsequential versions of Skart. Chapter 4 contains discussion and analysis of the experimental results for both versions of Skart when they are applied to a series of test problems, along with an empirical evaluation of Skart's sampling efficiency as the required relative precision of its CIs tends to zero. Finally, conclusions and recommendations for future studies are explained in Chapter 5.

The appendices include details on technical and theoretical formulation, evaluation, and practical application of Skart. Appendix A provides a rigorous theoretical justification for the time-series model of the batch means that is used in Skart to compute the correlation adjustment (1.11)–(1.12) to the estimator $S_{m,k}^2/k'$ of the variance of the grand mean $\bar{Y}(m, k')$ of the batch means. Appendix B contains the derivation and numerical evaluation of the conditional moment generating function and first two conditional moments of the waiting time in the $M/M/1$ queue for the first regular customer to arrive after time zero, given a specified nonnegative number of initial, “nonregular” customers in the system at time zero. Finally, Appendix C provides a user's manual for Skart, including screen shots of running the Skart and N-Skart software and the Visual Basic code for Skart and N-Skart.

Chapter 2

Literature Review

In this chapter we will initially review different methods for steady-state simulation output analysis. Then, we will explain six well-known automated steady-state simulation output analysis procedures, and summarize the advantages and disadvantages of each of these procedures. Finally, we will discuss three different skewness adjustments to the Student's t -statistic for i.i.d. observations.

2.1. Overview of Methods for Steady-State Analysis

A number of different approaches have already been developed for steady-state simulation output analysis. The following subsections provide a brief overview of the most popular ones.

2.1.1. Replication/Deletion

The replication/deletion method is based on independent replications (runs) of the simulation, and it consists of two phases. In the initial phase, k independent simulation-generated output processes each of length n are recorded, where X_{ij} denotes the i th observation within the j th replication for $i = 1, \dots, n$ and $j = 1, \dots, k$. The sample mean for each replication would then be computed after removing the observations in the warm-up period of length w at the beginning of each replication as follows:

$$\bar{X}_j = \frac{\sum_{i=w+1}^n X_{ij}}{n-w} \quad \text{for } j = 1, \dots, k,$$

where \bar{X}_j denotes the truncated sample mean of the j th replication. Notice that the \bar{X}_j 's are i.i.d. random variables with mean $E[\bar{X}_j]$, which may not equal μ_X ; and if the data-truncation point (statistics clearing time) w and the run length n are both big enough, then from the Central Limit Theorem (1.4) it can be claimed that $\{\bar{X}_j\} \stackrel{\text{i.i.d.}}{\sim} N(\mu_X, \text{Var}(\bar{X}_j))$ at least approximately, and

$$\text{Var}(\bar{X}_j) = \frac{1}{n-w} \left\{ \gamma_X(0) + 2 \sum_{\ell=0}^{n-w} \left(1 - \frac{\ell}{n-w}\right) \gamma_X(\ell) \right\} \quad (2.1)$$

$$\approx \frac{\gamma_X}{n-w} \quad \text{as } n \rightarrow \infty. \quad (2.2)$$

So a formulation similar to (1.7) based on sample means computed from truncated replications rather than adjacent batches can be used to deliver an approximately valid CI with the grand mean of all truncated replications,

$$\bar{\bar{X}} = \frac{1}{k} \sum_{j=1}^k \bar{X}_j,$$

as a point estimate for μ_X and

$$\widehat{\text{Var}}[\bar{\bar{X}}] = \frac{\sum_{j=1}^k [\bar{X}_j - \bar{\bar{X}}]^2}{k(k-1)}$$

as an estimate for the variance of $\bar{\bar{X}}$. The CI

$$\bar{\bar{X}} \pm t_{1-\alpha/2, k-1} \sqrt{\widehat{\text{Var}}[\bar{\bar{X}}]} \quad (2.3)$$

is approximately valid as $n \rightarrow \infty$ if the deletion phase appropriately removes the potential transients in each replication.

In the replication/deletion method, the warm-up period occurs k times, whereas in the single-run methods such as NBM the warm-up period occurs only once. The observations in each replication are not usually started in the steady-state phase due to the simulation's initial condition; and hence the resulting sample mean can be biased if the transients are not properly removed. In particular, if the truncation point w is not large enough to delete all the observations contaminated by significant initialization bias or if n is not large enough to swamp (overwhelm) any residual initialization bias that persists beyond the truncation point and if we take k too large, then we run the risk of a narrow CI centered on a biased estimator of μ_X . For some general conditions under which (2.3) is an asymptotically valid CI for μ_X as the run length $n \rightarrow \infty$, the warm-up period duration $w \rightarrow \infty$, and the replication count $k \rightarrow \infty$, see Theorem 6.4 Fishman [20].

Welch [59] developed a graphical method for determining the truncation point w which involves running k' pilot runs. In this method Welch takes averages across the k' i.i.d. occurrences of the observation with the “time” index i as follows,

$$\hat{\mu}_i = \frac{1}{k'} \sum_{j=1}^{k'} X_{ij} \quad \text{for } i = 1, \dots, n.$$

Thus $\{\hat{\mu}_i : i = 1, \dots, n\}$ is an estimate of the transient mean function $E[X_{ij}|X_{0j} = x_0]$ for $i = 1, \dots, n$, where we use the same initial condition $X_{0j} = x_0$ on each replication (run) j for $j = 1, \dots, k'$. Then, the M -point moving average, $\hat{\mu}_i(M)$, of these $\hat{\mu}_i$'s is plotted at various values of $M > 0$; and the truncation point w is chosen to be the value of i beyond which the plot of the $\{\hat{\mu}_i(M) : i = 1, 2, \dots\}$ seems to converge.

The replication/deletion method is very appealing because of its simplicity and because it can provide truly independent observations $\{\bar{X}_j : 1 \leq j \leq k\}$. This method also has strong theoretical support. On the other hand, this method is not very computationally efficient, considering that we are throwing away wk observations. Furthermore, there is no definitive, readily automatable approach to determine the sample size n and the truncation point w for each replication. Overall, this method is more attractive to those who do not have the statistical background necessary to use some of the more complicated analysis approaches. Unfortunately many of the users of the replication/deletion method fail to appreciate fully the risks of obtaining grossly misleading results by naive application of the procedure.

2.1.2. Regenerative Method

The regenerative method of output analysis can be applied to processes which exhibit a renewal-reward type of probabilistic behavior, meaning that at an increasing sequence of random times T_1, T_2, \dots , called regeneration times, the process starts afresh probabilistically independent of the past, resulting in i.i.d. blocks of data [16]. An example of a regenerative process is the sequence of queue waiting times for the $G/G/1$ queue, where a new block of data begins each time a new customer arrives at an empty-and-idle system, thus terminating the server's current idle period and starting a new busy period independently of all previous busy-idle cycles.

Consider a simulation-generated regenerative process $\{X_i\}$ and let $\{X_\ell : T_j \leq \ell < T_{j+1}\}$ denote the j th regeneration cycle for $j = 1, 2, \dots$. The following conditions are assumed to hold for

such a process [28]:

- (a) $\Pr[T_1 < \infty] = 1$ and $\Pr[0 < \tau_j < \infty] = 1$, where τ_j is defined as the length of the j th regenerative cycle $\tau_j \equiv T_{j+1} - T_j$ for $j \geq 1$;
- (b) the sequence $\{X_\ell : 0 \leq \ell < T_1\}$ is independent of $\{X_\ell : \ell \geq T_1\}$; and
- (c) the sequence $\{X_\ell : T_j \leq \ell < T_{j+1}\}$ is an i.i.d. replicate of the sequence $\{X_\ell : T_1 \leq \ell < T_2\}$.

Letting

$$Y_j = \sum_{i=T_j}^{T_{j+1}-1} X_i$$

denote the accumulated “reward” on the j th regenerative cycle, we see that

$$\{(Y_j, \tau_j), j \geq 1\}$$

is an i.i.d. sequence of two-dimensional random vectors. If $E[|Y_1|] < \infty$ and $E[\tau_1] < \infty$, then by the renewal-reward theorem [47], the steady-state mean of $\{X_i\}$ is

$$\mu_X = \frac{E[Y_1]}{E[\tau_1]}.$$

To construct an asymptotically valid CI for μ_X using a regenerative approach, we simulate, in one long run, k complete regenerative cycles and obtain k observations of the bivariate random vector (Y_j, τ_j) . If we run the simulation until we have accumulated k complete regenerative cycles, then the total sample size is

$$n(k) = \sum_{j=1}^k \tau_j$$

individual observations. The point estimate for μ_X using this method is

$$\hat{\mu}_X = \frac{\bar{Y}}{\bar{\tau}}, \tag{2.4}$$

where

$$\bar{Y} = \frac{1}{k} \sum_{j=1}^k Y_j$$

and

$$\bar{\tau} = \frac{1}{k} \sum_{j=1}^k \tau_j.$$

To formulate an asymptotically valid CI for μ_X centered on the regenerative point estimator (2.4), we must have $E[Y_j^2] < \infty$ and $E[\tau_j^2] < \infty$; then to obtain an unbiased estimator for the variance of $\hat{\mu}_X$, we define

$$V_j = Y_j - \mu_X \tau_j \quad \text{for } j = 1, \dots, k, \quad (2.5)$$

and

$$\bar{V} = \bar{Y} - \mu_X \bar{\tau}.$$

Notice that $E[V_j] = 0$ and

$$\sigma_V^2 = \text{Var}[V_j] = E[(Y_j - \mu_X \tau_j)^2] = E[Y_j^2] - 2\mu_X E[\tau_j Y_j] + \mu_X^2 E[\tau_j^2]. \quad (2.6)$$

Although the V_j 's in (2.5) are not observable quantities because they involve the unknown parameter μ_X that we are trying to estimate, the V_j 's are functions of i.i.d. random vectors; therefore, the V_j 's are also i.i.d. The Cauchy-Schwarz inequality yields

$$E[|\tau_j Y_j|] \leq \sqrt{E[\tau_j^2]E[Y_j^2]} < \infty; \quad (2.7)$$

and thus from (2.6) and (2.7), we see that $\sigma_V^2 < \infty$. Applying the Central Limit Theorem to the $\{V_j\}$, we have

$$\frac{\sqrt{k} \bar{V}}{\sigma_V} \xrightarrow[k \rightarrow \infty]{D} N(0, 1). \quad (2.8)$$

To estimate σ_V^2 , we compute the following sample estimates from k complete regenerative cycles,

$$S_Y^2 = \frac{1}{k-1} \sum_{j=1}^k (Y_j - \bar{Y})^2,$$

$$S_\tau^2 = \frac{1}{k-1} \sum_{j=1}^k (\tau_j - \bar{\tau})^2,$$

and

$$S_{Y\tau} = \frac{1}{k-1} \sum_{j=1}^k (Y_j - \bar{Y})(\tau_j - \bar{\tau}).$$

Using these sample estimates and equation (2.4), we obtain a strongly consistent estimate for σ_V^2 ,

$$\hat{\sigma}_V^2 = S_Y^2 - 2\hat{\mu}_X S_{Y\tau} + \hat{\mu}_X^2 S_\tau^2 = \frac{1}{k-1} \sum_{j=1}^k (Y_j - \hat{\mu}_X \tau_j)^2.$$

Now, substituting $\hat{\sigma}_V^2$ into equation (2.8) and applying Slutsky's theorem [7], we have

$$\frac{\sqrt{k}(\hat{\mu}_X - \mu_X)}{\hat{\sigma}_V/\bar{\tau}} \xrightarrow[k \rightarrow \infty]{D} N(0, 1);$$

and for k sufficiently large, an asymptotically valid $100(1 - \alpha)\%$ CI for the steady-state mean μ_X is

$$\hat{\mu}_X \pm z_{1-\alpha/2} \frac{\hat{\sigma}_V}{\sqrt{k} \bar{\tau}}, \quad (2.9)$$

where $z_{1-\alpha/2}$ is the $(1 - \alpha/2)$ -quantile of the standard normal distribution.

The regenerative method is appealing for several reasons: it is simple to understand and implement in practice; the initial bias problem is no longer an issue; and the technique provides truly i.i.d. observations. However, the regenerative method is not in widespread use because of the following disadvantages: (a) identifying regeneration times $\{T_j\}$ is extremely difficult in general; and (b) even if regeneration times can be identified, the corresponding expected cycle length $E[\tau_j]$ is usually so large that we require an excessively long simulation run in order to get a sufficiently large number of regenerative cycles. Because the regenerative point estimator (2.4) and CI estimator (2.9) both involve *ratio estimators* whose stochastic behavior is highly erratic for small values of k , it is recommended to accumulate $k \geq 100$ regenerative cycles so as to obtain reasonably stable results [27]. However in many large-scale simulation experiments involving heavily congested systems, it is simply infeasible to perform runs of sufficient length to accumulate 100 or more regenerative cycles.

2.1.3. Standardized Time Series

The method of Standardized Time Series (STS) was introduced by Schruben [48] for dependent stationary time series whose mean value is a parameter of interest. The STS method yields an asymptotically valid CI for μ_X if the target process $\{X_i\}$ satisfies the following Functional Central Limit Theorem (FCLT):

ASSUMPTION FCLT. The steady-state variance parameter $\gamma_X \in (0, \infty)$, and the sequence of random functions

$$W_n(t) = \frac{\lfloor nt \rfloor \{ \bar{X}(\lfloor nt \rfloor) - \mu_X \}}{\sqrt{n\gamma_X}} \quad \text{for } t \in [0, 1] \text{ and } n = 1, 2, \dots \quad (2.10)$$

satisfies

$$W_n(\cdot) \xrightarrow[n \rightarrow \infty]{D} \mathcal{W}(\cdot) \quad \text{on } [0, 1], \quad (2.11)$$

where $\lfloor \cdot \rfloor$ is the greatest-integer (floor) function; and $\mathcal{W}(\cdot)$ is a standard Brownian motion process on $[0, \infty)$ (see p. 4 of Billingsley [8]).

Within one steady-state simulation run that generates a time series $\{X_\ell : \ell = 1, \dots, n\}$ of length n , the corresponding standardized time series is the random function

$$T_n(t) \equiv \frac{\lfloor nt \rfloor \{ \bar{X}(n) - \bar{X}(\lfloor nt \rfloor) \}}{\sqrt{n\gamma_X}} \quad \text{for } t \in [0, 1], \quad (2.12)$$

where $n = 1, 2, \dots$. Under Assumption FCLT, it can be shown that we have the key limit theorem

$$\left[\sqrt{n} \{ \bar{X}(n) - \mu_X \}, \sqrt{\gamma_X} \times T_n(\cdot) \right] \xrightarrow[n \rightarrow \infty]{D} \sqrt{\gamma_X} [\mathcal{W}(1), \mathcal{B}(\cdot)], \quad (2.13)$$

where $\mathcal{B}(\cdot)$ is a standard Brownian bridge process on $[0, 1]$ (see p. 64 of Billingsley [8]) so that $\bar{X}(n)$ and $T_n(\cdot)$ are asymptotically independent as $n \rightarrow \infty$. The result (2.13) is the basis for the following development of a CI for μ_X using the STS method.

We let $\{X_\ell : \ell = 1, 2, \dots, n\}$ denote the output of a single steady-state simulation run, and we organize the data into k nonoverlapping (adjacent) batches each of size m so that we always have $n = km$. Thus the j th nonoverlapping batch of size m consists of the observations

$$\{X_{(j-1)m+i} : i = 1, \dots, m\} \quad \text{for } j = 1, \dots, k; \quad (2.14)$$

and we let

$$\bar{X}_j = \frac{1}{m} \sum_{i=1}^m X_{(j-1)m+i} \quad \text{for } j = 1, \dots, k$$

denote the j th batch mean of the process. The sample mean for the entire data set is

$$\bar{\bar{X}} = \frac{1}{k} \sum_{j=1}^k \bar{X}_j.$$

From (2.14), for the j th adjacent batch of size m ($j = 1, \dots, k$), we compute the STS

$$T_{j,m}(t) = \frac{1}{\sqrt{m\gamma_X}} \sum_{i=1}^{\lfloor mt \rfloor} (\bar{X}_j - X_{(j-1)m+i}) \quad \text{for } 0 \leq t \leq 1 \quad (2.15)$$

similar to the STS (2.12) defined for the entire output of the run. Consider

$$A_j = \sqrt{m\gamma_X} \sum_{u=1}^m T_{j,m}\left(\frac{u}{m}\right) = \sum_{u=1}^m \sum_{i=1}^u (\bar{X}_j - X_{(j-1)m+i}),$$

and

$$\mathcal{A} = \frac{12}{m^3 - m} \sum_{j=1}^k \mathcal{A}_j^2 + m \sum_{j=1}^k \left(\bar{X}_j - \bar{\bar{X}} \right)^2. \quad (2.16)$$

Since each STS $T_{j,m}(\cdot)$ converges to a Brownian bridge process as $m \rightarrow \infty$ and the STSs $\{T_{j,m}(\cdot) : j = 1, \dots, k\}$ computed from different nonoverlapping batches are asymptotically independent as $m \rightarrow \infty$, it can be shown that

$$\frac{12}{m^3 - m} \sum_{j=1}^k \mathcal{A}_j^2 \xrightarrow[m \rightarrow \infty]{D} \gamma_X \chi_k^2, \quad (2.17)$$

where χ_k^2 denotes a chi-squared random variable with k degrees of freedom;

$$m \sum_{j=1}^k \left(\bar{X}_j - \bar{\bar{X}} \right)^2 \xrightarrow[m \rightarrow \infty]{D} \gamma_X \chi_{k-1}^2; \quad (2.18)$$

and finally that (2.17) and (2.18) are independent. These results imply that

$$\frac{\mathcal{A}}{\gamma_X(2k-1)} \xrightarrow[m \rightarrow \infty]{D} \frac{\chi_{2k-1}^2}{2k-1}.$$

If the batch size m is sufficiently large so that the batch means $\{\bar{X}_j : j = 1, \dots, k\}$ are approximately i.i.d. normal random variables, then

$$\frac{\bar{\bar{X}} - \mu_X}{\sqrt{\frac{\gamma_X}{km}}} \quad \text{and} \quad \frac{\mathcal{A}}{\gamma_X(2k-1)}$$

are stochastically independent; and we have

$$\frac{\bar{\bar{X}} - \mu_X}{\sqrt{\frac{\gamma_X}{km}}} \sim N(0, 1).$$

Thus, a Student's t -ratio can be defined as follows,

$$\frac{\frac{\bar{\bar{X}} - \mu_X}{\sqrt{\frac{\gamma_X}{km}}}}{\sqrt{\frac{\mathcal{A}}{\gamma_X(2k-1)}}} = \frac{\bar{\bar{X}} - \mu_X}{\sqrt{\frac{\mathcal{A}}{km(2k-1)}}} \sim t_{2k-1} \quad \text{as } m \rightarrow \infty, \quad (2.19)$$

where t_{2k-1} denotes a Student's t -distribution random variable with $2k - 1$ degrees of freedom. This shows that

$$\lim_{m \rightarrow \infty} \Pr \left\{ -t_{1-\alpha/2, 2k-1} \leq \frac{\bar{\bar{X}} - \mu_X}{\sqrt{\frac{A}{km(2k-1)}}} \leq t_{1-\alpha/2, 2k-1} \right\} = 1 - \alpha.$$

Therefore, an asymptotically valid $100(1 - \alpha)\%$ CI for the steady-state mean μ_X is

$$\bar{\bar{X}} \pm t_{1-\alpha/2, 2k-1} \sqrt{\frac{A}{km(2k-1)}}. \quad (2.20)$$

The STS method as explained above has a firm mathematical foundation, and it is capable of delivering asymptotically valid CIs. The appeal of the STS method is that the SSVP cancels out in the calculation of the corresponding Student's t -statistic (2.19); and thus there is no need to estimate γ_X when using this method. The main disadvantage of the STS method is that it is based on the property that as $m \rightarrow \infty$, the standardized process $T_{j,m}(\cdot)$ converges to a Brownian bridge process. However, in many situations extremely large values of m are required to achieve this convergence, resulting in unrealistically large sample sizes. The complicated structure of this method also makes this approach difficult to understand and implement in practice.

Recent extensions of the STS method [2, 3] exploit the additional information about the variance parameter that can be obtained from *overlapping* batches within the same run so that the final Student's t -ratio analogous to (2.19) has substantially more degrees of freedom than (2.19) has; and thus the resulting CI for μ_X has improved precision (that is, smaller expected half-length) and improved stability (that is, smaller variance of the half-length) in comparison with (2.20). As for the version of the STS method based on nonoverlapping batches, the main problem in practical applications of the STS method based on overlapping batches is the lack of a definitive, automatable method for determining a batch size m sufficiently large so that the STSs computed from separate batches behave approximately like a Brownian Bridge process.

2.1.4. Spectral Analysis

The spectral analysis approach to steady-state output analysis is a consistent estimation method that seeks to estimate the SSVP in order to construct an asymptotically valid CI for μ_X centered on the sample mean \bar{X} . This method works in the frequency domain under the assumption that the process $\{X_i : i = 1, \dots, n\}$ is stationary.

The spectral density function $p_X(\omega)$ corresponding to the process $\{X_i\}$ is defined as

$$p_X(\omega) = \sum_{\ell=-\infty}^{\infty} \gamma_X(\ell) \cos(2\pi\omega\ell) \quad \text{for } -\frac{1}{2} \leq \omega \leq \frac{1}{2}, \quad (2.21)$$

where $\gamma_X(\ell)$ is the autocovariance function of $\{X_i\}$, as defined in Section 1.1. At frequency $\omega = 0$, we have

$$p_X(0) = \gamma_X = \sum_{\ell=-\infty}^{\infty} \gamma_X(\ell). \quad (2.22)$$

If $\sum_{\ell=-\infty}^{\infty} |\gamma_X(\ell)| < \infty$ and n is sufficiently large, then the variance of $\bar{X}(n)$ can be approximated by

$$\text{Var}[\bar{X}(n)] \approx \frac{p_X(0)}{n}. \quad (2.23)$$

Thus, an estimate $\hat{p}_X(0)$ of the spectral density of the underlying time series $\{X_i : i = 1, \dots, n\}$ at zero frequency is required to construct an asymptotically valid CI for the steady-state mean μ_X .

In classical spectral methods, the following estimator of the covariance function,

$$\hat{\gamma}_X(\ell) = \frac{1}{n-\ell} \sum_{j=1}^{n-\ell} (X_j - \bar{X}(n))(X_{j+\ell} - \bar{X}(n)), \quad \ell = 0, \pm 1, \pm 2, \dots, \pm(n-1), \quad (2.24)$$

is used to estimate at most $n-1$ covariances at various lags from a given sample of size n , so that a naive spectral estimator of the variance parameter γ_X would be

$$\hat{p}_X(0) = \hat{\gamma}_X = \sum_{\ell=-(n-1)}^{n-1} \hat{\gamma}_X(\ell).$$

Because of the large number of sample covariances in this naive spectral estimator of γ_X , this estimator's variance does not go to zero as $n \rightarrow \infty$ and tends to be highly variable. A truncated estimator

$$\hat{p}_X(0) = \hat{\gamma}_X = \sum_{\ell=-M}^M \hat{\gamma}_X(\ell) \quad (2.25)$$

has been suggested, with $M \ll n$ (here \ll means “much less than”) to reduce the variance of the estimator, but at the cost of increasing the estimator's bias. We can improve the estimator in (2.25) even further by weighing the estimated covariances $\{\hat{\gamma}_X(\ell) : |\ell| \leq M\}$, resulting in a less biased estimate as follows:

$$\hat{p}_X(0) = \hat{\gamma}_X = \sum_{\ell=-M}^M g(\ell) \hat{\gamma}_X(\ell), \quad (2.26)$$

where $g(\ell)$ is a positive, even function, called the *lag window*, with $g(0) = 1$ and $g(\ell)$ decreases as $|\ell|$ increases. If the spectral density function $p_X(\omega)$ is approximately linear near zero, then for the estimator $\hat{p}_X(0)$ in (2.26) we have $E[\hat{p}_X(0)] \approx p_X(0)$, i.e., the estimate is approximately unbiased and therefore, an asymptotically valid $100(1 - \alpha)\%$ CI for μ_X can be computed as follows:

$$\bar{X}(n) \pm z_{1-\alpha/2} \sqrt{\frac{\hat{p}_X(0)}{n}}.$$

The spectral density $p_X(\omega)$ is an even function, i.e., it is symmetric around zero. Thus in general at zero, $p_X(\omega)$ has either a peak or a valley and is not most often approximately linear. This implies that a weighted average of the spectrum about the point zero will usually result in a biased estimator for $p_X(0)$. For this reason, the determination of the truncation point M and the lag window $g(\ell)$ are very tricky and can result in excess variability or bias in the estimation of $p_X(0)$.

In practice, instead of sample covariances, the fast Fourier transform is used to estimate the SSVP from the values of the periodogram in the region near zero as elaborated in Equation (2.32) below; and for this reason the spectral method is considered by many to be computationally more efficient than the replication/deletion and regenerative methods. However, the spectral method is complicated, requiring a sophisticated background on the part of the analyst. Moreover, there is no definitive procedure for choosing the values for truncation point M and the lag window $g(\ell)$.

2.1.5. Method of Autoregressive Representation

The method of autoregressive representation, originated in simulation analysis by Fishman [18], assumes that the simulation-generated output $\{X_i : i = 1, 2, \dots\}$ is covariance stationary and can be represented by the p th-order autoregressive model

$$X_j - \mu_X = \sum_{i=1}^p \varphi_i (X_{j-i} - \mu_X) + \varepsilon_j, \quad (2.27)$$

for an appropriate (finite) value of p , where the φ_i 's are the autoregressive coefficients, μ_X is the unknown process mean, and the ε_j 's are i.i.d. normal residuals with zero mean and finite variance σ_ε^2 .

It is possible to show that if the Equation (2.27) holds for the process $\{X_i\}$ and each of the roots of the *characteristic equation*

$$1 - \sum_{v=1}^p \varphi_v z^v = 0$$

has absolute value greater than one, then

$$\lim_{n \rightarrow \infty} n \text{Var}[\bar{X}(n)] = \frac{\sigma_\varepsilon^2}{(1 - \sum_{v=0}^p \varphi_i)^2}.$$

Fishman gives a procedure in [18] which employs the sample estimators of the process covariances $\gamma_X(\ell)$ for $\ell = 0, \pm 1, \pm 2, \dots$ to determine the estimated order \hat{p} , the estimated autoregressive coefficients $\hat{\varphi}_i$ for $i = 1, \dots, \hat{p}$, and the estimated residual variance $\hat{\sigma}_\varepsilon^2$.

Yuan and Nelson [62] revisited and updated the autoregressive method proposed by Fishman and suggested to use Rissanen's predictive least-squares criterion [46] to estimate the autoregressive order of the process. Yuan and Nelson [62] also proposed to use the conditional-least-squares estimator (CLSE) [11] to estimate the coefficients of the autoregressive model.

Let $\hat{\varphi} = 1 - \sum_{i=1}^{\hat{p}} \hat{\varphi}_i$. Then, for large n , an estimate of $\text{Var}[\bar{X}(n)]$ and an approximate 100(1 - α)% CI for μ_X are given by

$$\widehat{\text{Var}}[\bar{X}(n)] = \frac{\hat{\sigma}_\varepsilon^2}{n(\hat{\varphi})^2}$$

and

$$\bar{X}(n) \pm t_{1-\alpha/2, \hat{f}} \sqrt{\widehat{\text{Var}}[\bar{X}(n)]}, \quad (2.28)$$

respectively, where Fishman estimated the degrees of freedom \hat{f} by

$$\hat{f} = \frac{n\hat{\varphi}}{(2\hat{p} + 1)\hat{\varphi} + 4 \sum_{v=1}^{\hat{p}} v\hat{\varphi}_v}.$$

The main drawback of the method of autoregressive representation is the lack of a definitive method for estimating the order p of the autoregressive model. In principle a large class of stationary processes (namely, the so-called invertible processes) have an infinite-order autoregressive representation; see, for example, Theorem 7.6.9 of Anderson [5] or §A7.2 of Koopmans [35]. Unfortunately, the practical issue of finding an adequate finite-order approximation to the infinite-order autoregressive representation of a simulation-generated process has yet to be resolved. For example, Yuan and Nelson [62] apply their autoregressive method to system sojourn times in an $M/M/1$ queue with mean inter-arrival time 10 and mean service time 9 so that the server utilization is 0.9 and the steady-state mean sojourn time $\mu_X = 90$. For fixed sample sizes ranging from 500 to 5,000 observations, they compute nominal 90% CIs for μ_X of the form (2.28). Yuan and Nelson [62] find noticeable degradation in CI coverage as well as CI half-lengths with relatively large mean and standard deviation. Specifically for

sample size $n = 5,000$, nominal 90% CIs have an empirical coverage probability of 69%; and the sample mean and standard deviation of the associated CI half-lengths are 28.19 and 30.44, respectively.

2.2. Review of Automated Simulation Output Analysis Procedures

In this section we provide a brief explanation of six automated steady-state simulation output analysis procedures that have received the most attention in the literature. The NBM method was used in developing the Law-Carson procedure [40] and the ABATCH and LBATCH [19] procedures. ASAP3 [54] and SBatch [39] are two advanced batch-means procedures in which the size of the warm-up period and the size of all subsequent batches are taken separately to be just large enough to yield adjacent nonoverlapping batch means (in the case of ASAP3) or spaced batch means (in the case of SBatch) that approximately constitute a stationary first-order autoregressive process. Suitable adjustments are then applied to the classical NBM Student's t -ratio (in the case of ASAP3) or to the sample variance of the spaced batch means (in the case of SBatch) to construct the final CI. WASSP [37] and the Heidelberger-Welch procedure [24] seek to deliver valid CIs by estimating the power spectrum of the output process generated by a steady-state simulation model. The chief advantage of these methods is that most of them are completely automated, requiring no intervention by the user.

2.2.1. ASAP3

ASAP3 is a sequential nonoverlapping batch-means procedure for constructing CIs for the mean of a steady-state simulation output process. This procedure is a refinement of ASAP [52] and ASAP2 [53]. ASAP3 retains the advantages of its predecessors while being mainly designed to address their weaknesses such as the excessive variability in the half-length of their delivered CIs—especially in situations for which there is no precision requirement. ASAP3 is based on the observation that the batch means usually achieve approximately joint multivariate normality at a batch size that is substantially smaller than the batch size required to achieve approximate independence of the batch means.

ASAP3 starts by dividing the simulation outputs into $k = 256$ nonoverlapping batches of user-specified size m (by default $m = 16$) and computing the corresponding batch means. Next ASAP3 truncates the first four batches to reduce the transient effects in the data, and ASAP3 organizes the remaining 252 batches into 63 adjacent groups each consisting of four batches. Then from every other

group of four adjacent batches, ASAP3 computes the corresponding batch means and forms a four-dimensional vector from these statistics. The resulting sample of 32 four-dimensional vectors is next tested for stationary multivariate normality. If the normality test is failed, then ASAP3 performs the following operations: the batch size m is increased by the factor $\sqrt{2}$; additional observations are collected; and the new set of truncated batch means is tested iteratively for multivariate normality until this test is finally passed. Each time ASAP3 applies the multivariate normality test, it deflates the significance level δ of the test to avoid excessive variability in the final sample size in applications with no precision requirements (see Section 3.1 of Steiger et al. [54] for further explanation).

In the next step, ASAP3 fits a first-order autoregressive (AR(1)) model to all the batch means computed beyond the first four ignored batches. Thus ASAP3 sets the truncation point $w = 4m$, where m is the current sample size delivered by the multivariate normality test; and in subsequent steps, ASAP3 uses all batch means after the truncation point and not merely alternating groups of four adjacent batch means. Let $\{Y_j(m) : j = 1, \dots, k'\}$ denote the resulting set of $k' = 252$ truncated, adjacent batch means. The next step of ASAP3 is to test the following hypothesis:

$$\varphi = \text{Corr}[Y_j(m), Y_{j+1}(m)] \leq 0.8. \quad (2.29)$$

The main reason for inserting the threshold test (2.29) into ASAP3 is to prevent the excessive variability observed in the CIs delivered by ASAP and ASAP2 when $\varphi > 0.8$, especially in the absence of a precision requirement. ASAP3 applies a normalizing arc-sine transformation to the autoregressive parameter estimator $\hat{\varphi}$ in order to test if the null hypothesis (2.29) holds with high probability. If the null hypothesis (2.29) is rejected, then ASAP3 performs the following steps:

- (a) it increases the batch size m by a factor projected to reduce the lag-one correlation between the batch means to an acceptable level below the 0.8 threshold;
- (b) it obtains the required additional observations;
- (c) it recomputes 256 batch means using the new batch size m ;
- (d) it skips the initial four batches and uses the remaining $k' = 252$ batches to recompute the new autoregressive parameter φ by fitting an AR(1) model to the resulting set of truncated, adjacent batch means; and finally
- (e) it retests the hypothesis (2.29).

These steps are repeated until we obtain a batch size that is big enough to pass the null hypothesis of nonexcessive correlation between the successive batch means.

The resulting batch means are then used to build a CI that has been adjusted to account for the remaining nonexcessive correlations between the batch means. For this purpose, ASAP3 corrects the classical NBM Student's t -ratio by using a Cornish-Fisher expansion whose terms are computed from the parameter estimates $\hat{\varphi}$ and $\hat{\sigma}_a^2$ that are obtained by fitting an AR(1) model to the current set of batch means. A correlation-adjusted $100(1 - \alpha)\%$ CI for μ_X is

$$\bar{Y}(m, k') \pm \left[\left(\frac{1}{2} + \frac{\hat{\kappa}_2}{2} - \frac{\hat{\kappa}_4}{8} \right) z_{1-\alpha/2} + \frac{\hat{\kappa}_4}{24} z_{1-\alpha/2}^3 \right] \times \sqrt{\frac{\widehat{\text{Var}}[Y(m)]}{k'}}, \quad (2.30)$$

where: $\hat{\kappa}_2$ and $\hat{\kappa}_4$ respectively denote estimators of the second and fourth cumulants of the classical NBM Student's t -ratio; $\widehat{\text{Var}}[Y(m)]$ denotes an estimator of the variance of the batch means; and the statistics $\hat{\kappa}_2$, $\hat{\kappa}_4$, and $\widehat{\text{Var}}[Y(m)]$ are computed from $\hat{\varphi}$ and $\hat{\sigma}_a$ as detailed in Section 3.3 of Steiger *et al.* [54].

The half-length of the delivered CI is then checked to see if it satisfies the precision requirement provided by the user (for the no precision case this does not apply). If the precision requirement is not satisfied, then the batch size m and batch count k are inflated by a factor based on the ratio of the current iteration's CI half-length to the desired CI half-length. ASAP3 is reapplied to the simulation-generated data until finally it delivers a CI with the required precision.

2.2.2. LBATCH and ABATCH

LBATCH and ABATCH are two procedures developed by Fishman and Yarberry [19] to analyze the output of steady-state simulation experiments. These variants of the method of batch means are designed to be applied to a fixed-length simulation-generated time series, where the user is trying to verify sufficient conditions to ensure strong consistency for the estimator of the variance of the sample mean; and ultimately the user seeks to construct asymptotically valid CIs for the mean. Neither LBATCH nor ABATCH provides a means of eliminating initialization bias or a method for determining a sample size that is sufficiently large to yield a CI for μ_X of user-specified precision.

These two procedures are designed based on two specific rules: (i) the fixed-number-of-batches rule (FNB); and (ii) the square-root rule (SQRT). The FNB rule fixes the number of batches and lets the batch size increase as the total sample size increases. This rule, however, cannot provide a

consistent estimator for the SSVP, resulting in CIs which tend to be wider than CIs constructed using a consistent estimator for the SSVP. The SQRT rule considers a scheme in which both the number of batches and the batch size grow as the total sample size increases. The SQRT rule tends to underestimate the variance of the sample mean for a fixed total sample size, which results in poorer coverage (see Fishman [19]). In general, for a given sample size the FNB rule provides CIs with better coverage than the SQRT rule, while the CI half-lengths generated by the FNB rule are larger than those generated by the SQRT rule.

To determine an appropriate batch size and batch count, Fishman [19] developed dynamic assignment rules that shift between the FNB and SQRT rules based on a test of independence of the batch means to retain the advantages of both the FNB and SQRT rules and avoid their major disadvantages. The FNB rule aims at dissipating the correlation between the batch means as fast as possible, whereas the SQRT rule aims at making the distribution of the statistic $(\bar{X}(n_j) - \mu_X) / \sqrt{S_{n_j, k_j}^2 / k_j}$ converge to $N(0, 1)$ as rapidly as possible, where n_j and k_j are the sample size and the number of batches on the j th iteration of the LBATCH or ABATCH procedure.

The LBATCH procedure begins with a fixed number of batches and tests for correlation between the batch means using von Neumann's randomness test [61]. If this test is failed, then the FNB rule is employed to reduce the correlation as fast as possible. The batch size and hence the total number of observations are doubled while the number of batches remains fixed; and additional observations are collected if necessary. The correlation test is applied to the set of batch means until the test no longer detects high correlation between the batch means based on the latest batch size. Then, the SQRT rule is applied iteratively until a predetermined number of observations is reached. The total sample size doubles on each iteration of the SQRT rule; and the batch size and the number of batches grow by a factor of approximately $\sqrt{2}$ on each application of the SQRT rule.

The ABATCH procedure is similar to the LBATCH procedure, except that ABATCH can switch back and forth between the FNB and SQRT rules. The test for correlation is applied on each iteration; and if this test is failed after the SQRT rule has been invoked, then ABATCH switches back to the FNB rule for the next iteration and all subsequent iterations until the hypothesis of independence is accepted again, causing the procedure to switch to the SQRT rule again. For detailed steps of the LBATCH and ABATCH procedures see Section 2 of Fishman and Yarberry [19].

2.2.3. Law-Carson Procedure

The Law-Carson procedure [40] is based on the method of nonoverlapping batch means. This procedure discounts the effect of nonnormality for a number of batches $k \geq 20$ and mainly focuses on the problem caused by the correlation between the batch means for small batch sizes. The FNB rule explained in Section 2.2.2 is also the driving force for this procedure. A comprehensive description of the Law-Carson procedure is given in [40].

This procedure starts by fixing the number of batches and sequentially increasing the batch size m until the sample lag-one correlation $\hat{\rho}_1(m)$ between the batch means is less than some specified level c and the resulting CI achieves a user-specified relative precision level $r > 0$. However, the correlation estimator $\hat{\rho}_1(m)$ is generally biased; and for small k , it also has a large variance. Law and Carson [40] found that some systems require as many as 400 batches for a precise estimate of $\rho_1(m)$, the true lag-one correlation between adjacent batch means for batches of size m .

This issue encouraged them to look at the possibility of using ℓk batches of size m to infer whether k batches of size ℓm are approximately uncorrelated. Law and Carson [40] noticed three types of behavior for $\rho_1(m)$ as a function of m after studying 34 processes for which $\rho_1(m)$ could be computed analytically:

- (i) the lag-one correlation strictly decreases to zero;
- (ii) the lag-one correlation changes direction one or more times and then strictly decreases to zero;
and
- (iii) the lag-one correlation is less than zero.

To complete its initialization step, the Law-Carson procedure requires the user to set some positive integers ℓ, k, n_0, n_1 (ℓk even, $\ell k/2$ even, $n_0 < n_1 < 2n_0$ each divisible by ℓk). Then, given a simulation-generated time-series of length n_1 , the observations are divided into ℓk batches of size m . The lag-one correlation between the batch means is estimated using the jackknifed estimator

$$\tilde{\rho}_1(\ell k, m) = 2\hat{\rho}_1(\ell k, m) - \left[\hat{\rho}_1^{(1)}(\ell k/2, m) + \hat{\rho}_1^{(2)}(\ell k/2, m) \right] / 2,$$

where $\hat{\rho}_1^{(1)}(\ell k/2, m)$ and $\hat{\rho}_1^{(2)}(\ell k/2, m)$ are the usual lag-one correlation estimators based on the first and last $\ell k/2$ batches, respectively. This jackknife estimator is used because in general it is less biased than the conventional sample correlation estimator.

If $\tilde{\rho}_1(\ell k, m) > c$, meaning that an acceptable level of correlation between the batch means is not reached, then the total number of observations is updated according to $n_i \leftarrow 2n_{i-2}$, where i is the iteration number. Therefore, n_i doubles on every other iteration. If the desired level of correlation is achieved, i.e., $\tilde{\rho}_1(\ell k, m) \leq c$, then the overall data set of size n_i is divided into k batches of size ℓm and a classical CI is constructed, using (1.7).

Next, the Law-Carson procedure checks whether the latest CI satisfies the given relative precision requirement. If not, then the iteration counter i is incremented, $i \leftarrow i + 1$; the total sample size is updated according to $n_i \leftarrow 2n_{i-2}$; additional observations are collected; and another iteration of the Law-Carson procedure is performed. Successive iterations of the Law-Carson procedure are performed until the precision requirement is finally satisfied.

2.2.4. SBatch

SBatch [39] is a batch-means procedure in which the size of the warm-up period and the size of all subsequent batches are taken separately to be just large enough to yield spaced batch means that approximately constitute a stationary first-order autoregressive process. A correlation adjustment is then applied to the sample variance of the spaced batch means to construct the final CI. This procedure is explained in more detail in the following paragraphs.

SBatch begins by dividing the total sample size of $n = 16,384$ into $k = 1,024$ adjacent (non-spaced) batches of size $m = 16$ with an initial spacer of size $s = 0$ observations preceding each batch. Then it applies the randomness test of von Neumann [61] to the initial set of k batch means to determine the data-truncation point beyond which all computed batch means are approximately independent not only of each other but also of the simulation's initial conditions.

Each time the randomness test is failed, SBatch adds a new batch to each spacer (up to 14 batches are allowed per spacer); then SBatch reapplies the randomness test to the new, reduced set of spaced batch means. If the number of batches in each spacer reaches the maximum of 14 batches and the randomness test is not yet passed, then SBatch resets the spacer size s to zero and inflates the batch size m by a factor of $\sqrt{2}$; additional observations are collected by resuming the simulation if necessary; and all the observations are rebatched into $k = 1,024$ adjacent (nonspaced) batches of the new size m . A new set of k batch means is computed and tested for randomness. The preceding steps in this paragraph may be repeated several times before the spaced batch means finally pass the randomness

test. Once the randomness test is passed, the observations $\{X_i : i = 1, \dots, s\}$ comprising the first spacer are skipped (ignored) to account for system warm-up; and the size of the spacer separating each batch is fixed.

Next, the resulting set of spaced batch means are tested for univariate normality using the method of Shapiro-Wilk [49]. Each time the normality test is failed, SBatch increases the batch size by a certain multiplicative factor (i.e., $\sqrt{2}$ for the first six test failures and $\lfloor 2^{1/(\ell-4)} \rfloor$ for the ℓ th failure of the normality test, $\ell = 7, 8, \dots$). Section 4.3 of Lada et al. [39] explains the reason for reducing the batch-size inflation factor. Additional observations are collected if required; and a new set of batch means is then computed using the final spacer size s provided by the randomness test and the new inflated batch size m . The normality test is repeated for the new set of spaced batch means.

After the normality test is passed, SBatch applies a threshold test to make sure that the lag-one correlation of the resulting set of approximately normal, spaced batch means is less than 0.8. Each time the correlation test is failed, the batch size is increased by 10%, additional observations are collected, a new set of spaced batch means is computed, and the correlation test is repeated for the new set of spaced batch means.

Once the correlation test is passed, SBatch will deliver a correlation-adjusted $100(1 - \alpha)\%$ CI on the steady-state mean using the current set of k' approximately i.i.d. normal spaced batch means as follows,

$$\bar{X} \pm t_{1-\alpha/2, k'-1} \sqrt{\frac{A \hat{\sigma}_{\bar{X}(m,s)}^2}{k'}}$$

where the midpoint \bar{X} is the average of all observations including those in the spacers (except the first); $\hat{\sigma}_{\bar{X}(m,s)}^2$ is the sample variance of the k' spaced batch means; $t_{1-\alpha/2, k'-1}$ is the $1 - \alpha/2$ quantile of Student's t -distribution with $k' - 1$ degrees of freedom; and

$$A = \frac{1 + \hat{\varphi}_{\bar{X}(m,s)}^2}{1 - \hat{\varphi}_{\bar{X}(m,s)}^2}$$

is the correlation adjustment based on $\hat{\varphi}_{\bar{X}(m,s)}^2$, the usual sample estimator of the lag-one correlation between the k' spaced batch means with batch size m and spacer size s . The adjustment A is applied to the sample variance $\hat{\sigma}_{\bar{X}(m,s)}^2$ to account for any residual correlation that may exist between the spaced batch means.

SBatch checks the half-length of the provided CI in the next step. If the user-specified precision requirement is satisfied, then SBatch delivers the latest computed CI and terminates. Otherwise, it

estimates the number of spaced batch means needed to meet the precision requirement. If the estimated number of spaced batch means exceeds 1,024, then SBatch reduces this number to 1,024 and increases the batch size instead. If necessary, additional observations are collected by resuming the current run of the simulation model. Then a new CI is computed using the recalculated correlation adjustment, and the precision requirement is retested. SBatch repeats the procedure outlined above until a CI satisfying the precision requirement is finally delivered.

2.2.5. WASSP

WASSP uses a wavelet-based spectral method to analyze the stochastic output from a steady-state simulation [37]. This procedure addresses both the start-up and correlation problems, delivering a CI for the steady-state mean using a wavelet estimator of the SSVP that satisfies the user-specified precision requirements. The steps of WASSP are summarized in this section.

WASSP starts by dividing the initial simulation-generated time series of length $n = 4,096$ into $k = 256$ adjacent batches of size $m = 16$, with an initial interbatch spacer of size $s = 0$ observations preceding each batch. The approach WASSP uses to determine the warm-up period and to achieve approximately i.i.d. spaced batch means is similar to the approach used by SBatch (see the von Neumann test in Subsection 2.2.4). However, in its randomness test WASSP allows up to nine batches in each spacer, whereas SBatch allows up to 14 batches per spacer.

Once the randomness test is passed so that the spacer size s is fixed, WASSP takes the spacer preceding the first batch to be the warm-up period; and like SBatch, WASSP applies the univariate Shapiro-Wilk test for normality to the truncated, spaced batch means to determine a batch size sufficiently large to yield approximately normal spaced batch means. If the normality test is failed, then WASSP progressively increases the batch size while keeping the same spacer size, deflates the Shapiro-Wilk test level of significance, and retests the newly computed set of spaced batch means for normality at the newly computed level of significance.

After the normality next is passed, WASSP computes a set of adjacent (nonspaced) batch means of the current batch size beyond the truncation point. Using these batch means, WASSP computes a wavelet-based estimator of the batch means log-spectrum over its full frequency range (i.e. from $-1/2$ to $1/2$ cycles per unit of time). Then, WASSP smooths the periodogram of the batch means by computing a multipoint moving average; WASSP applies a logarithmic transformation to the smoothed

periodogram; and then WASSP corrects for the bias induced by this transformation. Following this step, WASSP computes the discrete wavelet transform of the bias-corrected log-smoothed-periodogram of the batch means and applies a soft-thresholding scheme to obtain a parsimonious, denoised set of wavelet coefficient estimators. Finally, WASSP computes the inverse discrete wavelet transform of the thresholded wavelet coefficients to recover the wavelet-based approximation to the batch-means log-spectrum and ultimately an estimator $\hat{\gamma}_X$ of the SSVP for the original process (an estimator of the spectrum of the unbatched process at zero frequency). WASSP delivers a CI of the form

$$\bar{X}(n') \pm H, \text{ with half-length } H = t_{1-\alpha/2, \nu} \sqrt{\hat{\gamma}_X/n'}, \quad (2.31)$$

where: (i) the grand mean $\bar{X}(n')$ is computed from the truncated output process of size n' ; and (ii) ν denotes the effective degrees of freedom associated with $\hat{\gamma}_X$.

WASSP then checks the half-length of the provided CI. If the user-specified precision requirement is not satisfied, then WASSP estimates the total number of batches of the current batch size that are needed to fulfill the precision requirement. If the estimated number of batch means exceeds 4,096, then WASSP reduces this number to 4,096 and increases the batch size instead. The additional simulation-generated observations are obtained by restarting the simulation, and then the next iteration of WASSP is performed.

2.2.6. Heidelberger-Welch Procedure

The Heidelberger-Welch (HW) procedure employs a spectral method for steady-state simulation output analysis in which it uses standard regression techniques to estimate the power spectrum of the simulation-generated output process $\{X_j : j = 1, \dots, n\}$ at zero frequency [24, 25, 26]. This procedure uses the fast Fourier transformation to compute the periodogram values $\{I(\frac{\ell}{n}) : 0 < \ell < \frac{n}{2}\}$ of the simulation-generated time series which are approximately independent, unlike the original responses which are often strongly correlated, as follows:

$$\begin{aligned} I\left(\frac{\ell}{n}\right) &= \frac{1}{n} \left\{ \left[\sum_{j=1}^n X_j \cos\left(\frac{2\pi(j-1)\ell}{n}\right) \right]^2 + \left[\sum_{j=1}^n X_j \sin\left(\frac{2\pi(j-1)\ell}{n}\right) \right]^2 \right\} \\ &= \frac{|a(\ell)|^2}{n}, \end{aligned} \quad (2.32)$$

where $a(\ell)$ is the discrete Fourier transform of the output process $\{X_i : i = 1, \dots, n\}$. The periodogram has the following properties:

$$\left. \begin{aligned} \mathbb{E}\left[I\left(\frac{\ell}{n}\right)\right] &\approx p_X\left(\frac{\ell}{n}\right), & 0 < \ell < \frac{n}{2}, \\ \text{Var}\left[I\left(\frac{\ell}{n}\right)\right] &\approx p_X^2\left(\frac{\ell}{n}\right), & 0 < \ell < \frac{n}{2}, \\ \text{Cov}\left[I\left(\frac{\ell}{n}\right), I\left(\frac{j}{n}\right)\right] &\approx 0, & 0 < \ell \neq j < \frac{n}{2}, \text{ and} \\ I\left(\frac{\ell}{n}\right) &\sim \frac{1}{2}p_X\left(\frac{\ell}{n}\right)\chi_2^2, & 0 < \ell < \frac{n}{2}, \end{aligned} \right\}$$

where χ_2^2 denotes a chi-squared random variable with two degrees of freedom—that is, an exponential random variable with mean one.

After computing the periodogram (2.32) of the time series $\{X_i : i = 1, \dots, n\}$, HW computes the logarithm of averages of pairs of adjacent periodogram values and fits a quadratic polynomial to the log-smoothed-periodogram in a small neighborhood, with zero frequency as its lower boundary, to estimate the SSVP by extrapolating the fitted polynomial to zero frequency.

On each iteration of the HW procedure, prior to applying the spectral method discussed above, this procedure performs a scheme for batching the simulation-generated data in an attempt to determine an appropriate truncation point for eliminating initialization bias. First the user must specify an upper limit t_{\max} on the allowable length of a given test process; thus like LBATCH and ABATCH, the HW procedure is a fixed-sample-size procedure rather than a truly sequential procedure. The batch count k for the HW procedure is always in the range $L \leq k \leq 2L$, where the value $L = 200$ is recommended in practice [26]. Following Heidelberger and Welch's notion, we let t_i denote the current (untruncated) sample size at the i th checkpoint in the analysis of a given output process, where

$$t_1 = \lceil 0.15 t_{\max} \rceil \text{ and } t_i = \min \{ \lceil 1.5 t_{i-1} \rceil, t_{\max} \} \text{ for } i = 2, 3, \dots .$$

If $t_i \geq L$ and we take

$$b_i = \lfloor \log_2 \{(t_i - 1)/L\} \rfloor,$$

then at the i th checkpoint the batch size m_i and the number of batches k_i are given by

$$m_i = 2^{b_i} \text{ and } k_i = \lfloor t_i / m_i \rfloor,$$

respectively.

At each checkpoint $i = 1, 2, \dots$ of the HW procedure, the simulation-generated process is initially organized into k_i batches of size m_i ; and then the untruncated batch means $\{\bar{X}_j(m_i) : j =$

$1, \dots, k_i\}$ are computed and tested for covariance stationarity by computing the Cramér–von Mises (CVM) test statistic,

$$\text{CVM}(m_i, k_i) = \left[\sum_{j=0}^{k_i-1} D_{ij}^2 \right] / [k_i^2 \hat{p}_{\bar{X}(m_i)}(0)],$$

where $\hat{p}_{\bar{X}(m_i)}(0)$ denotes the HW estimator of the power spectrum $p_{\bar{X}(m_i)}(\omega)$ of the untruncated batch means process at frequency $\omega = 0$; and

$$D_{i0} = 0 \text{ and } D_{ij} = \sum_{u=1}^j [\bar{X}_u(m_i) - \bar{\bar{X}}(m_i, k_i)] \text{ for } j = 1, \dots, k_i.$$

A failure of the CVM test is an indication of the presence of nonstationarity (initialization bias) in the sequence of untruncated batch means.

If the CVM test is failed at a given checkpoint i , then the HW procedure removes the first 10% of the current untruncated batch-means sequence and reapplies the CVM test to the truncated batch means. If the CVM test again detects nonstationarity in the sequence of the truncated batch means, then the HW procedure removes the next 10% of the original (untruncated) sequence of batch means of the current batch size m_i . This process is repeated until (i) the CVM test is passed; or (ii) the CVM stationarity test is failed for six consecutive times, resulting in a warm-up period of length equal to 50% of the original untruncated sequence of batch means of the current batch size m_i . If case (ii) occurs, then the HW procedure advances to the next checkpoint by increasing the untruncated sample size by 50% and updating the batch size m_i , batch count k_i , and batch means $\{\bar{X}_j(m_i) : j = 1, \dots, k_i\}$. The CVM test is repeated at successive checkpoints until one of the following conditions occur: (a) the CVM test is passed, and the CI of the form given in (2.31) satisfies the user-specified precision level; or (b) the new checkpoint's untruncated sample size exceeds t_{\max} , the maximum allowable length of the given test process. If condition (b) occurs, then the HW procedure terminates without delivering a CI with the desired level of precision level. The possibility that HW procedure might simply fail to deliver a CI with the user-specified level of precision has turned out to be a significant drawback in practical applications of the procedure. A more comprehensive description of CVM test and the HW procedure is given in Lada *et al.* [38].

2.3. Relative Performance of Automated Simulation Output Analysis Procedures

Evaluations of some recent procedures for steady-state simulation analysis in Lada *et al.* [36] show that ASAP3 and WASSP compare favorably with the Law-Carson procedure with respect to coverage probability and sampling efficiency in all the considered cases. The results from Steiger *et al.* [54] also indicate that ASAP3 outperforms the ABATCH and LABATCH procedures not only with regard to conformance to the precision and coverage-probability requirements but also with regard to the mean and variance of the half-length of the delivered CIs. Lada *et al.* [38] compare the performance of WASSP, ASAP3, and HW with respect to sampling efficiency and the robustness of the delivered CIs. Based on their results, WASSP outperforms the HW procedure in many respects and compares favorably with ASAP3. Most recently, Lada *et al.* [39] compare the performance of SBatch, ASAP3, and WASSP. Their performance evaluation results are not very clear-cut, and no single procedure dominates the others in all the considered experiments.

In Section 4, considering the results given in the above paragraphs, we decided to compare the performance of our newly proposed batch-means procedure, Skart, with that of ASAP3, WASSP and SBatch, as the three procedures in the steady-state simulation output analysis literature that currently seem to dominate all the other competing procedures.

2.4. Skewness Adjustment to the Student's t -Statistic for Independent Observations

Consider a sample of n i.i.d. observations $\{X_i : i = 1, \dots, n\}$ with sample mean \bar{X} and sample variance S^2 , where the cumulative distribution function (c.d.f.) $F(\cdot)$ of the X_i 's has finite moments at least up to order four so that the mean $\mu = E[X_i] = \int_{-\infty}^{+\infty} x dF(x)$ and the central moments

$$\mu_k = E[(X_i - \mu)^k] = \int_{-\infty}^{+\infty} (x - \mu)^k dF(x) \quad \text{for } k = 2, 3, 4 \quad (2.33)$$

are well-defined and finite. Thus the variance, skewness, and kurtosis of the X_i 's are given by

$$\sigma^2 = \mu_2, \quad \text{Sk} = \frac{\mu_3}{\sigma^3}, \quad \text{and} \quad \text{Ku} = \frac{\mu_4}{\sigma^4}, \quad (2.34)$$

respectively. In this section if we want to emphasize the dependence of the moments (2.33) and (2.34) on the random variable X , then we will write $\mu(X)$, $\sigma^2(X)$, $\mu_k(X)$, $\text{Sk}(X)$, $\text{Ku}(X)$, etc.

The Student's ratio $t = \sqrt{n}(\bar{X} - \mu)/S$ is commonly used to make hypothesis tests about the mean μ of the underlying population and also to form CIs for μ . The main problem is that when the underlying population is not normal, the usual Student's t -distribution cannot be used to compute exact CIs for μ ; and in particular if the underlying population has pronounced positive or negative skewness, then the coverage probability of the usual CI based on Student's t -distribution is severely degraded. Several modifications have been suggested to eliminate skewness from the distribution of a Studentized statistic. In the following subsections, we will review some of the popular modifications and clarify the reason for using the modified t -statistic that has been implemented in Skart.

2.4.1. Johnson's Modification of Student's t -Statistic

Johnson [29] suggested a modification of Student's t -statistic using properties of the data so that the modified statistic has Student's t -distribution to a sufficiently good approximation. Because the skewness of the parent population affects the behavior of the original Student's t -statistic more than the kurtosis of the parent population does, Johnson's modified t -statistic provides a transformation to reduce the effect of skewness by exploiting the terms of the Cornish-Fisher expansion of the original Student's t -statistic involving the third and fourth central moments of the underlying population. The general form of the Cornish-Fisher expansion for the random variable X with moments (2.33) and (2.34) is given by [58]:

$$\text{CF}(X) = \mu + \sigma\zeta + \frac{1}{6} \cdot \frac{\mu_3}{\sigma^2}(\zeta^2 - 1) + \frac{1}{24} \cdot \frac{\mu_4}{\sigma^3}(\zeta^3 - 3\zeta) + \dots, \quad (2.35)$$

where ζ is a standard normal random variable.

Remark 1. For an alternative derivation of (2.35), see §6.25 of Stuart and Ord [55]. In terms of the analysis (6.47)–(6.56) of Stuart and Ord, we seek to develop a Cornish-Fisher expansion not for the random variable X directly but instead for the *standardized* random variable

$$X' = (X - \mu)/\sigma$$

as a function of the standard normal random variable $\zeta \sim N(0, 1)$; then taking

$$\left. \begin{aligned} m &= \text{E}[\zeta] = 0, \\ \sigma^2 &= \text{Var}[\zeta] = 1 \end{aligned} \right\} \quad (2.36)$$

in Equations (6.47)–(6.56) of Stuart and Ord [55], we must have

$$\left. \begin{aligned} \ell_1 &= \{E[X'] - m\}/\sigma^2 = 0, \\ \ell_2 &= \{\text{Var}[X'] - \sigma^2\}/\sigma^2 = 0 \end{aligned} \right\} \quad (2.37)$$

in the Cornish-Fisher expansion of X' given by Equation (6.56) of Stuart and Ord [55]. Solving the resulting equation for X expressed in terms of ζ and retaining only the first four terms on the right-hand side of Equation (6.56) of Stuart and Ord [55], we obtain (2.35).

For the sample mean $\bar{X} = \bar{X}(n)$ of a random sample of size n from $F(\cdot)$, we have

$$\mu(\bar{X}) = \mu, \quad \sigma(\bar{X}) = \frac{\sigma}{\sqrt{n}}, \quad \mu_k(\bar{X}) = \frac{\mu_k}{n^{k-1}} \text{ for } k = 3, 4, \dots; \quad (2.38)$$

and combining (2.35) and (2.38), we see that a valid representation of \bar{X} by a Cornish-Fisher expansion paralleling (2.35) up to terms of order n^{-1} is

$$\text{CF}(\bar{X}) = \mu + \frac{\sigma}{\sqrt{n}}\zeta + \frac{\mu_3}{6n\sigma^2}(\zeta^2 - 1) + O(n^{-\frac{3}{2}}), \quad (2.39)$$

since the first neglected term in (2.39) is

$$\frac{1}{24} \cdot \frac{\mu_4(\bar{X})}{\sigma^3(\bar{X})}(\zeta^3 - \zeta) = \frac{1}{24} \cdot \frac{\mu_4(X)/n^3}{[\sigma(X)/\sqrt{n}]^3} = \frac{1}{24} \frac{\mu_4(X)}{\sigma(X) \cdot n^{3/2}} = O(n^{-3/2}). \quad (2.40)$$

Here we remark that $\mu_3 = \mu_3(X)$, the third central moment of the underlying distribution, is the coefficient of the term $(\zeta^2 - 1)$ in Equation (2.39). The quantity μ_3 also appears in the coefficients of other terms, but they are of smaller order in the sample size n .

The key to obtaining a modified t -statistic in Johnson's approach is to eliminate the term involving μ_3 in the Cornish-Fisher expansion of the modified t -statistic, which is defined by

$$t_1 = \frac{(\bar{X} - \mu) + \lambda + \gamma\{(\bar{X} - \mu)^2 - (\sigma^2/n)\}}{(S^2/n)^{1/2}}. \quad (2.41)$$

The numerator of (2.41) is suggested by looking at the inverse Cornish-Fisher expansion of ζ expressed in terms of the first few powers of $(\bar{X} - \mu)$ in (2.39). The expansion for t_1 is similar to that given in (2.39). The constant λ is chosen so that constant terms (i.e., the terms that do not involve the random variable $(\bar{X} - \mu)$ or the sample size n) in the Cornish-Fisher expansion of t_1 sum to zero, and the terms of order \sqrt{n} and $1/\sqrt{n}$ that do not involve $(\bar{X} - \mu)$ are eliminated; moreover, γ is chosen so that the

coefficient of the term ζ^2 in the Cornish-Fisher expansion of t_1 is zero. Johnson [29] shows that if we substitute the values

$$\gamma = \frac{\mu_3}{3\sigma^4} \text{ and } \lambda = \frac{\mu_3}{2n\sigma^2} \quad (2.42)$$

into Equation (2.41), then in the Cornish-Fisher expansion for t_1 any residual effects due to skewness of the original observations $\{X_i\}$ have the form $O(1/n)$ and thus become negligible as the sample size n increases. Making the substitutions (2.42) in (2.41), we obtain

$$t_1 = \frac{(\bar{X} - \mu) + \mu_3/(6\sigma^2n) + [\mu_3/(3\sigma^4)](\bar{X} - \mu)^2}{(S^2/n)^{1/2}}, \quad (2.43)$$

which has approximately the Student t -distribution with $n - 1$ degrees of freedom—provided the X_i 's are randomly sampled from a distribution whose first four moments exist, and n is sufficiently large. In this equation the dominant effects of population skewness are eliminated, although because of the terms in the numerator of (2.43) that are of second order (quadratic) in the unknown μ , this formulation is neither monotone in μ nor invertible and therefore cannot result in a simple expression for CIs. Consequently, the corresponding CIs may consist of two disjoint subintervals, which is highly counterintuitive and therefore not desirable.

Johnson [29] suggested a method for constructing a simpler CI by removing the term involving $(\bar{X} - \mu)^2$ in (2.43), claiming the effect of this omitted term is of small order. This new variable t'_1 and its corresponding CI have the following form:

$$t'_1 = \frac{(\bar{X} - \mu) + \mu_3/(6\sigma^2n)}{(S^2/n)^{1/2}}, \quad (2.44)$$

$$[\bar{X} + \hat{\mu}_3/(6S^2n)] \pm t_{1-\alpha/2, n-1} S/\sqrt{n}, \quad (2.45)$$

where

$$\hat{\mu}_3 = \frac{n}{(n-1)(n-2)} \sum_{i=1}^n (X_i - \bar{X})^3.$$

The experimental results of Kleijnen *et al.* [33], however, did not support the use of (2.45) in preference to the usual CI given in (1.1); instead these results indicated that neglecting the term involving $(\bar{X} - \mu)^2$ in (2.43) adversely affects the performance of hypothesis tests and CIs based on (2.44).

2.4.2. Hall's Modification of Student's t -Statistic

Hall [23] suggested a modification to Johnson's method so that the calculated CIs are not disjoint while providing good coverage. He modified the test statistic to become a monotonic function of the mean. The Edgeworth expansion of the c.d.f. of the test statistic $Q = Q(X_1, \dots, X_n)$ based on a sample of size n , if it exists, is

$$\Pr(n^{1/2}Q \leq x) = \Phi(x) + n^{-1/2}\gamma(a_2 + a_1x^2)\phi(x) + O(n^{-1}), \quad (2.46)$$

where a_1 and a_2 are known constants that depend on the function $Q(\cdot)$, γ is an estimable constant that depends on the underlying c.d.f. $F(\cdot)$ of the $\{X_i\}$, and $\phi(x)$ and $\Phi(x)$ denote the density and distribution function of the standard normal distribution, respectively. The second term in the expansion (2.46) describes the error in the usual normal approximation and depends on the skewness in the underlying population through the constant γ .

Hall defines a transformation of the statistic Q to obtain a new statistic R with the following Edgeworth expansion,

$$\Pr(n^{1/2}R \leq x) = \Phi(x) + O(n^{-1}). \quad (2.47)$$

In the new statistic R , the term involving the skewness of the $\{X_i\}$ is eliminated. The transformation suggested by Hall has the form

$$R(Q) \equiv a_2\gamma/n + Q + a_1\gamma Q^2 + a_1^2\gamma^2 Q^3/3, \quad (2.48)$$

with a_1 , a_2 , γ , and n being the constants appearing in (2.46). The function $R(Q)$ is monotonic and has a unique inverse

$$Q(R) = (a_1\gamma)^{-1}(\{1 + 3a_1\gamma(R - a_2\gamma/n)\}^{1/3} - 1) \quad (2.49)$$

for any real value of R . After transforming the Q statistic to R , whose distribution is nearly symmetric, Hall applies the normal approximation to the new statistic R and then regains the asymmetry of the original statistic by the inverse transformation (2.49). This method is more promising than the method of Johnson because it uses a symmetric statistic in computing CIs.

To build a CI for the population mean μ , Hall set $Q = (\bar{X} - \mu)/\tilde{\sigma}$, and $\tilde{\sigma} \equiv S\sqrt{(n-1)/n}$, with $a_1 = 1/3$, $a_2 = 1/6$, and $\gamma = \mu_3/\sigma^3$; and he estimated the latter quantity by

$$\tilde{\gamma} \equiv n^{-1} \sum_{i=1}^n (X_i - \bar{X})^3 / \tilde{\sigma}^3.$$

It follows that an approximate $100 \times (1 - 2\alpha)\%$ two-sided CI for μ is

$$\left[\bar{X} - \tilde{\sigma} \tilde{Q}(\Phi^{-1}(1 - \alpha)/\sqrt{n}), \bar{X} - \tilde{\sigma} \tilde{Q}(\Phi^{-1}(\alpha)/\sqrt{n}) \right], \quad (2.50)$$

where

$$\tilde{Q}(r) \equiv \frac{3}{\tilde{\gamma}} \left(\left\{ 1 + \tilde{\gamma} \left(r - \frac{\tilde{\gamma}}{6n} \right) \right\}^{1/3} - 1 \right). \quad (2.51)$$

The results shown in Tables 1–3 of [23] indicate that the relative performance of Hall's transformed CIs are better than the modified CIs proposed by Johnson [29].

2.4.3. Willink's Modification of Student's t -Statistic

Willink [60] suggested a modification to Johnson's method by adding a cubic term in $(\bar{X} - \mu)$ to the numerator of t_1 in (2.43) to ensure existence of a unique inverse of t_1 as a function of μ . The added term has the form $o(n^{-1})$, which has the same order as the first neglected term in t_1 . Thus, we do not expect this extra term to have a large effect on the asymptotic performance of the approximation as $n \rightarrow \infty$. The modified t -statistic proposed by Willink is

$$t_2 = \frac{(\bar{X} - \mu) + \hat{\mu}_3/(6S^2n) + [\hat{\mu}_3/(3S^4)](\bar{X} - \mu)^2 + [\hat{\mu}_3^2/(27S^8)](\bar{X} - \mu)^3}{(S^2/n)^{1/2}}. \quad (2.52)$$

The same transformation given by Hall in (2.48) is recovered if we make the following substitutions in (2.52):

$$Q = \bar{X} - \mu, \quad a_2\gamma = \hat{\mu}_3/(6S^2), \quad \text{and} \quad a_1\gamma = \hat{\mu}_3/(3S^4).$$

The t_2 statistic can also be expressed in terms of the standard Student's t -statistic $t = \sqrt{n}(\bar{X} - \mu)/S$ as follows:

$$t_2 = t + a + 2at^2 + 4a^2t^3/3, \quad \text{where} \quad a = \frac{\hat{\mu}_3/S^3}{6\sqrt{n}}. \quad (2.53)$$

Therefore, an approximate $100 \times (1 - 2\alpha)\%$ two-sided CI for μ is

$$\left[\bar{X} - G(t_{1-\alpha, n-1}) \times S/\sqrt{n}, \bar{X} - G(t_{\alpha, n-1}) \times S/\sqrt{n} \right], \quad (2.54)$$

where

$$G(r) \equiv (2a)^{-1} \left(\{1 + 6a(r - a)\}^{1/3} - 1 \right) \quad \text{and} \quad a \equiv \frac{\hat{\mu}_3/S^3}{6\sqrt{n}}. \quad (2.55)$$

The performance evaluation of Willink in [60] shows that when the distribution of the $\{X_i\}$ is skewed, the intervals proposed by Willink appear to be considerably more reliable (in terms of approximately achieving the nominal coverage probability $1 - 2\alpha$) than those based on the standard Student's t -statistic and other CIs in the literature. When the distribution of the $\{X_i\}$ is normal, the proposed intervals are slightly wider on average than the standard intervals. In general, the coverage of Willink's CI (2.54) increases as the magnitude of the skewness of the X_i 's decreases, the sample size n increases, or the nominal confidence coefficient $1 - 2\alpha$ increases. Moreover, the results in Tables 1–3 of [60] show that when the sample size is big enough, (2.54) performs well for distributions with skewness as high as 4.0. For higher skewness values, Willink's CI (2.54) often fails to achieve an acceptable coverage probability.

Chapter 3

Skart: A Skewness- and Autoregression-Adjusted Procedure for Steady-State Simulation Analysis

In this chapter we first discuss Skart, an automated sequential procedure for on-the-fly or offline steady-state simulation output analysis, and we provide a detailed explanation and justification for each of the steps of Skart. Next we explain the steps of a nonsequential version of Skart, called N-Skart, that has been specifically designed for arbitrary fixed-size simulation-generated data sets; and we highlight the key differences between the original version of Skart and the nonsequential version of the procedure.

3.1. Overview of Skart

Figure 3.1 depicts a high-level flowchart of Skart. To invoke this procedure, the user must provide a desired CI coverage probability $1 - \alpha$, where $0 < \alpha < 1$, and an upper bound H^* on the CI half-length, where H^* is either expressed in absolute terms as the maximum acceptable half-length, or in relative terms as the maximum acceptable fraction r^* of the magnitude of the CI midpoint.

Skart requires 1,280 observations from a simulation model to launch. Then it computes the sample skewness of these observations to determine the initial batch size. If the sample skewness is

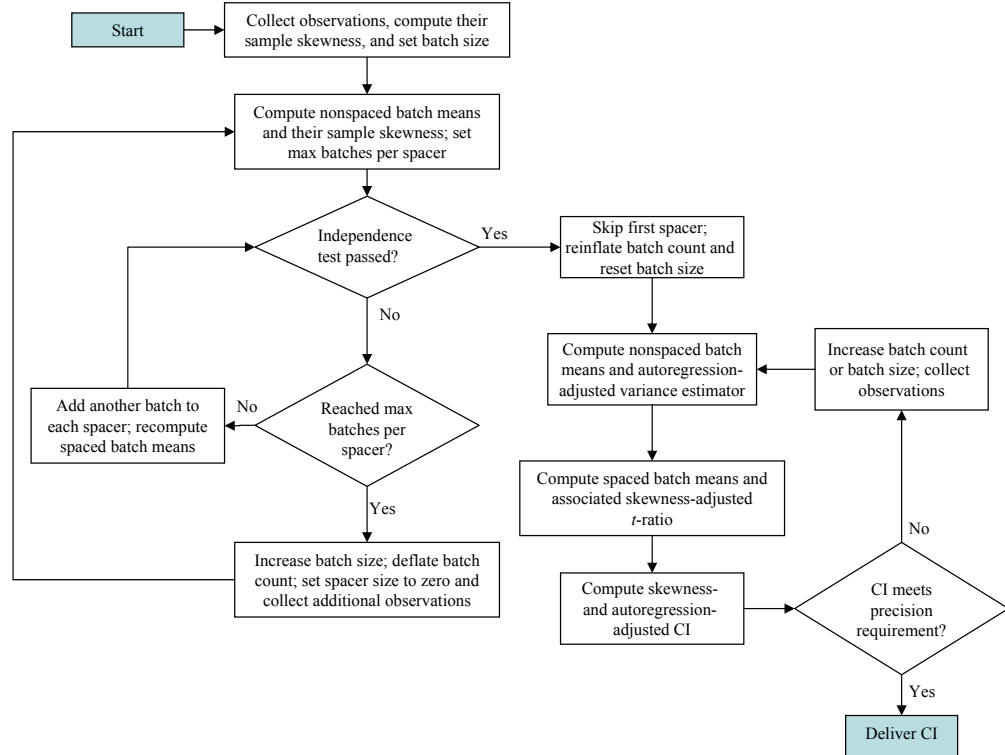


Figure 3.1: High-level flowchart of Skart

greater than 4.0, then Skart sets the batch size $m = 16$, so that the initial sample size must be increased to $16 \times 1,280 = 20,480$. This extreme case only happens when the observations are highly nonnormal. Usually, the sample skewness of the initial data set is less than 4.0, and Skart assigns the batch size $m = 1$.

Having set an appropriate value for the initial batch size, Skart divides the initial sample into $k = 1,280$ nonspaced (adjacent) batches of size m with an initial spacer consisting of $d \leftarrow 0$ ignored batches preceding each “spaced” batch. Then to determine d^* , the maximum number of batches allowed in each spacer, Skart computes the sample skewness of the corresponding batch means after skipping the first 20% of the batch means to reduce the effect of any initialization bias that may be present. If the absolute value of the sample skewness computed from the last 80% of the current set of k batch means exceeds 0.5, then Skart sets $d^* \leftarrow 3$, otherwise Skart sets $d^* \leftarrow 10$.

In the next step, Skart applies the randomness test of von Neumann [61] to the current set of k batch means to determine the required batch count, batch size, and data-truncation point beyond which all computed spaced batch means are approximately independent not only of each other but also of the simulation's initial conditions. After each iteration of the randomness test, we let k' denote the current number of spaced batch means, where each spacer consists of d ignored batches. Each time the randomness test is failed, Skart adds an additional batch to each spacer and increments the number of batches per spacer, $d \leftarrow d + 1$ (up to the computed limit of d^* batches per spacer), and updates the count of spaced batch means, $k' \leftarrow \lfloor n/\{(d + 1)m\} \rfloor$; then Skart reapplies the randomness test to the new reduced set of k' spaced batch means. If the randomness test is failed with a spacer consisting of d^* batches, then d is reset to zero, the nonspaced batch count k is deflated by the factor 0.9, the batch size m is increased by the factor $\sqrt{2}$, and additional observations are collected if required (by restarting the simulation if necessary). The augmented sample is rebatched into the reduced set of k nonspaced batches of the enlarged batch size m ; the maximum number of batches d^* is updated to take the value 3 or 10 depending on the absolute value of the sample skewness computed from the last 80% of the latest set of k nonspaced batch means as described previously; and the randomness testing procedure outlined in this paragraph is reperformed starting with the latest set of k nonspaced batch means.

Once the randomness test is passed, the observations $\{X_i : i = 1, \dots, dm\}$ in the first spacer are skipped (ignored) to account for the system warm-up, and the remaining $n' = n - dm$ observations are used to build a set of nonspaced (adjacent) batches. The number of batches $k' \leftarrow \lceil k'(1/0.9)^b \rceil$ in this set is obtained by reinflating the batch count resulting from the randomness test by the factor $(1/0.9)^b$, where b is the number of times the batch count was deflated in the randomness test; and the corresponding batch size is reset to $m \leftarrow \max\{\lfloor n'/k' \rfloor, m\}$, the maximum of the number of truncated observations divided by the computed batch count and the current value of the batch size. If $mk' > n'$, then Skart requests $mk' - n'$ additional observations, which may require resuming the simulation. Then, Skart computes the resulting truncated, nonspaced batch means $\{Y_j(m) = m^{-1} \sum_{i=1}^m X_{dm+(j-1)m+i} : j = 1, \dots, k'\}$ and the associated grand mean $\bar{Y}(m, k')$ and sample variance $S_{m,k'}^2$ of the truncated, nonspaced batch means.

Skart exploits separate adjustments to the classical batch-means CI based on the corresponding effects of nonnormality and correlation of the batch means on the distribution of the usual Student's t -ratio that underlies the batch-means method. To do this, Skart first must compute approximately unbiased estimator of the marginal variance and skewness of the truncated, nonspaced batch means

$\{Y_j(m) : j = 1, \dots, k'\}$ with the current batch size m . From all the individual observations in the current simulation-generated data set, Skart temporarily forms a set of approximately i.i.d. spaced batch means with batch size m , where the spacer size is the smallest multiple of m exceeding the size of the warm-up period. Let k'' denote the resulting number of spaced batch means. From this approximate random sample of size k'' , Skart computes $S_{m,k''}^2$ and $\hat{B}_{m,k''}$, the usual unbiased estimators of the associated marginal variance and skewness of batch means with batch size m . Skart exploits the statistics $S_{m,k''}^2$ and $\hat{B}_{m,k''}$ to compute a skewness- and correlation-adjusted CI for μ_X from $\{Y_j(m) : j = 1, \dots, k'\}$, the current set of k' truncated, nonspaced batch means with batch size m .

Skart computes the correlation adjustment A of Equation (1.11) using $\hat{\varphi}_{Y(m)}$, the usual sample estimator of the lag-one correlation of the truncated nonspaced batch means. Thus $AS_{m,k''}^2/k'$ is an approximately unbiased estimator of $\bar{Y}(m, k')$ that has $k'' - 1$ degrees of freedom and that accounts for any residual correlation between the current set $\{Y_j(m) : j = 1, \dots, k'\}$ of truncated, nonspaced batch means with batch size m .

Skart incorporates the method developed by Willink [60] to adjust the classical Student's t -statistic underlying the NBM method for any skewness present in the distribution of the batch means. For this correction, Skart computes the Willink's modified t -ratio (2.52) by making the following substitutions in (2.52):

- S^2 is replaced by $S_{m,k''}^2$; and
- $\hat{\mu}_3$ is replaced by $\hat{B}_{m,k''} S_{m,k''}^3$.

A correlation- and skewness-adjusted CI is then computed, where the midpoint of the CI is $\bar{Y}(m, k')$. The CIs generated by Skart are not usually equal-tailed, meaning that the intervals can be asymmetric around $\bar{Y}(m, k')$. When the sample skewness of the batch means is high, the difference in the lengths of the left and right tails is more conspicuous in the delivered CIs.

If no precision level is specified, then Skart simply delivers the latest CI and stops. Otherwise, the CI is tested to determine if it satisfies a user-specified absolute or relative precision requirement. In the precision requirement test, Skart assumes that the interval's half-length is equal to the length of the wider tail. If the precision requirement is satisfied, then Skart terminates and delivers the latest CI. Otherwise, Skart estimates a new batch count or batch size (or both) to meet the precision requirement.

If necessary, additional observations are collected by resuming the current run of the simulation model. Then a new CI is computed using the recalculated correlation and skewness adjustments, and the precision requirement is retested. Skart repeats the procedure outlined in the last two paragraphs until a CI satisfying the precision requirement is finally delivered.

3.2. Formal Algorithmic Statement of Skart

To invoke Skart the user must provide the following: (i) a desired CI coverage probability $1 - \alpha$, where $0 < \alpha < 1$; and (ii) an upper bound H^* on the CI half-length, where H^* is either expressed in absolute terms as the maximum acceptable half-length, or in relative terms as the maximum acceptable fraction r^* of the magnitude of the CI midpoint.

Subsequently Skart delivers one of the following: (i) a nominal $100(1 - \alpha)\%$ CI for μ_X that satisfies the specified absolute or relative precision requirement, provided no additional data are required; or (ii) a new, larger sample size n to be supplied to Skart when it is executed again. If additional observations must be generated by resuming (continuing) the current run of the user's simulation model before a CI with the required precision can be delivered, then Skart must be called again with the additional data; and this cycle may be repeated several times before Skart finally delivers a CI.

A formal algorithmic statement of Skart is given on the succeeding pages. In §3.3.1–3.3.5, we describe the steps in the algorithm in more detail.

3.3. Detailed Description of Steps in Skart

3.3.1. Initialization Step

To initiate Skart, the user must provide a data set that is at least large enough to satisfy Skart's minimal requirement for a simulation-generated time series $\{X_j : j = 1, \dots, n\}$ of length $n \leftarrow 1,280$. This requirement is arguably not too far from the minimal sample size required for meaningful analysis of a time series with any of the following properties: (i) a nontrivial deterministic trend (initialization bias); (ii) a nontrivial stochastic dependency structure (autocorrelation function); or (iii) nonnormal distributional characteristics.

Skart is designed so that it can deliver CIs using either observation-based or time-persistent

Figure 3.2: Algorithmic statement of Skart

[1] Take the initial sample of size $n \leftarrow 1,280$ and compute the sample skewness \hat{B} of the last 1,024 observations in the sample. If $|\hat{B}| > 4.0$, then set the initial batch size $m \leftarrow 16$ and increase the initial sample size to 20,480; otherwise set $m \leftarrow 1$. Then divide the initial sample into $k \leftarrow 1,280$ nonspaced (adjacent) batches of size m . Compute the batch means $\{Y_j(m) : j = 1, \dots, k\}$. Set the current number of batches in a spacer, $d \leftarrow 0$, and the maximum number of batches allowed in a spacer, $d^* \leftarrow 10$. Set the randomness test size, $\alpha_{\text{ran}} \leftarrow 0.20$, and the number of times the batch count has been deflated in the randomness test, $b \leftarrow 0$.

[2] Compute the sample skewness \hat{B}_m of the last 80% of the current set of k nonspaced batch means with size m . If $|\hat{B}_m| > 0.5$, then reset the maximum number of batches per spacer, $d^* \leftarrow 3$.

[3] Apply the von Neumann test for randomness to the current set of k batch means with significance level α_{ran} .

[3a] If the randomness test is passed, then set $k' \leftarrow k$ and go to **[5]**; otherwise go to **[3b]**.

[3b] Insert spacers each with m observations (one ignored batch) between the $k' \leftarrow \lfloor k/2 \rfloor$ remaining batches; assign the values of the k' spaced batch means; and set the total number of batches in a spacer, $d \leftarrow 1$.

[3c] Apply the randomness test to the current set of k' spaced batch means with significance level α_{ran} . If the randomness test passed, then go to **[5]**; otherwise go to **[3d]**.

[3d] If $d = d^*$ so that the current number of batches per spacer equals the maximum number of batches per spacer, then go to **[4]**; else add another ignored batch to each spacer so that the total number of batches per spacer and the number of spaced batches are respectively updated according to

$$d \leftarrow d + 1 \text{ and } k' \leftarrow \lfloor n / \{(d + 1)m\} \rfloor.$$

Reassign the values of the k' spaced batch means, and go to **[3c]**.

[4] Update the batch size m , the total batch count k , the overall sample size n , and Skart's other status variables according to

$$m \leftarrow \lceil \sqrt{2m} \rceil, \quad k \leftarrow \lceil 0.9k \rceil, \quad n \leftarrow km, \quad d \leftarrow 0, \quad b \leftarrow b + 1, \text{ and } d^* \leftarrow 10.$$

Obtain the required additional simulation-generated observations, recompute the k adjacent batch means, and go to **[2]**.

[5] Skip the first $w = d \times m$ observations in the overall sample of size n . Divide the remaining $n' \leftarrow n - w$ observations into $k' \leftarrow \lceil k'(1/0.9)^b \rceil$ nonspaced batches of size $m \leftarrow \max\{\lfloor n'/k' \rfloor, m\}$. If $mk' > n'$, then obtain $mk' - n'$ additional observations. Compute the current set of truncated, nonspaced batch means $\{Y_j(m) \leftarrow m^{-1} \sum_{i=1}^m X_{w+(j-1)m+i} : j = 1, \dots, k'\}$.

[6] Compute the grand average $\bar{Y}(m, k')$ and the sample variance $S_{m, k'}^2$ of the current set of truncated, nonspaced batch means $\{Y_j(m) : j = 1, \dots, k'\}$. Use the sample estimator of the lag-one correlation of the truncated, nonspaced batch means

$$\hat{\varphi}_{Y(m)} \leftarrow \frac{1}{k' - 1} \sum_{j=1}^{k'-1} [Y_j(m) - \bar{Y}(m, k')] [Y_{j+1}(m) - \bar{Y}(m, k')] / S_{m, k'}^2$$

to compute the correlation adjustment

$$A \leftarrow \frac{1 + \hat{\varphi}_{Y(m)}}{1 - \hat{\varphi}_{Y(m)}}$$

to the variance of the grand mean.

[7] Compute the correlation-adjusted $100(1 - \alpha)\%$ CI for μ_X using the skewness-adjusted critical values $G(t_{1-\alpha/2, k''-1})$ and $G(t_{\alpha/2, k''-1})$ of Student's t -ratio for $k'' - 1$ degrees of freedom,

$$\left[\bar{Y}(m, k') - G(t_{1-\alpha/2, k''-1}) \sqrt{\frac{AS_{m, k'}^2}{k'}}, \bar{Y}(m, k') + G(t_{\alpha/2, k''-1}) \sqrt{\frac{AS_{m, k'}^2}{k'}} \right], \quad (3.1)$$

where to evaluate (3.1) we must first compute spaced batch means with $d' = \lceil w/m \rceil$ batches per spacer so we have $k'' = 1 + \lfloor (k' - 1)/(d' + 1) \rfloor$ spaced batches of size m with corresponding spaced batch means $Y_j(m, d') \equiv Y_{(j-1)(d'+1)+1}(m)$ for $j = 1, \dots, k''$ with grand mean

$$\bar{Y}(m, k'', d') \leftarrow \frac{1}{k''} \sum_{j=1}^{k''} Y_j(m, d')$$

and sample variance and sample third central moment respectively given by

$$S_{m, k''}^2 \leftarrow \frac{1}{k'' - 1} \sum_{j=1}^{k''} [Y_j(m, d') - \bar{Y}(m, k'', d')]^2 \quad (3.2)$$

and

$$\mathcal{J}_{m,k'',d'} \leftarrow \frac{k''}{(k''-1)(k''-2)} \sum_{j=1}^{k''} [Y_j(m, d') - \bar{Y}(m, k'', d')]^3. \quad (3.3)$$

From the latter statistics (3.2) and (3.3), we compute

$$\hat{\mathcal{B}}_{m,k''} = \frac{\mathcal{J}_{m,k'',d'}}{S_{m,k''}^3}, \quad \text{and} \quad \beta \equiv \frac{\hat{\mathcal{B}}_{m,k''}}{6\sqrt{k'}}, \quad \text{and} \quad G(\zeta) \equiv \frac{\sqrt[3]{1 + 6\beta(\zeta - \beta)} - 1}{2\beta}.$$

If no precision level is specified, then deliver the CI (3.1) and stop; else go to **[8]**.

[8] Apply the appropriate absolute or relative precision stopping rule.

[8a] If the half-width

$$H \leftarrow \max \{ |G(t_{1-\alpha/2, k''-1})|, |G(t_{\alpha/2, k''-1})| \} \times \sqrt{\frac{AS_{m,k''}^2}{k'}}$$

of the current CI satisfies the user-specified precision requirement

$$H \leq H^*,$$

where

$$H^* \leftarrow \begin{cases} r^* |\bar{Y}(m, k')|, & \text{for a user-specified relative precision level } r^*, \\ h^*, & \text{for a user-specified absolute precision level } h^*, \end{cases}$$

then deliver both the CI $\bar{Y}(m, k') \pm H$ and the alternative form (3.1) and stop; otherwise go to **[8b]**.

[8b] Estimate the number of batches of the current size that will be required,

$$k^* \leftarrow \lceil (H/H^*)^2 k' \rceil.$$

[8c] Update the number of nonspaced batch means k' , the batch size m , and the total sample size n as follows:

$$\begin{aligned} k' &\leftarrow \min\{k^*, 1024\}; \\ \text{If } k' < k^* \text{ then } m &\leftarrow \lceil m \times \text{mid}\{1.02, (H/H^*)^2, 2\} \rceil; \text{ and} \\ n &\leftarrow d + k'm. \end{aligned}$$

Obtain the additional simulation-generated observations; recompute the nonspaced batch means and go to **[6]**.

statistics. For observation-based statistics, initially the simulation model generates the required initial sample size and temporarily suspends the current run (stands by), waiting for Skart to do one of the following: (i) return a CI with the required precision so that the current simulation run may be terminated; or (ii) return the number of additional observations that must be generated after the current simulation run is resumed and before the next invocation of Skart. This cycle might be repeated for several times before Skart finally delivers a CI.

For time-persistent statistics, the situation is more complicated. First Skart asks the user for a sampling interval value Δ expressed in the basic time units of the simulation clock. Skart then computes time-weighted statistics at multiples of Δ during a pilot simulation run. Usually the value of Δ is set by the user based on knowledge of the dynamics of the given simulation model. A larger sampling interval might be more appropriate in a system in which the value of the statistic does not change too often during the simulation run. Having the value of Δ , next Skart computes the total simulation time required to generate the minimum initial sample size. For instance, if the user sets $\Delta = 4$, then to generate a data set of size 1,280 in the initialization step of Skart, the simulation model has to run up to time $1,280 \times 4 = 5,120$. The required simulation run length is updated similarly each time Skart asks for a new sample size. As in the case of discrete-time output processes, this cycle might be repeated several times before Skart finally delivers a CI for the steady-state mean of a time-persistent statistic.

To determine the initial value of the batch size m , Skart computes the sample skewness of the

last $\ell \leftarrow 1,024$ observations in the initial sample,

$$\hat{\mathcal{B}} \leftarrow \frac{\ell}{(\ell-1)(\ell-2)} \sum_{j=n-\ell+1}^n (X_j - \bar{X})^3 / S^3,$$

where

$$\bar{X} \leftarrow \frac{1}{\ell} \sum_{j=n-\ell+1}^n X_j, \text{ and } S^2 \leftarrow \frac{1}{\ell-1} \sum_{j=n-\ell+1}^n (X_j - \bar{X})^2$$

respectively denote the sample mean and variance of the last ℓ observations in the initial sample.

If $|\hat{\mathcal{B}}| \leq 4.0$, then Skart takes the initial batch size $m \leftarrow 1$; otherwise, Skart takes $m \leftarrow 16$. Then Skart initializes the current batch count $k \leftarrow 1,280$ and computes the current sample size $n \leftarrow km$, requesting additional observations from the user if the initially supplied data set does not contain at least n observations. This method for assigning the initial batch size and then increasing the initial sample size from $n = 1,280$ to $n = 16 \times 1,280 = 20,480$ if necessary is designed to ensure that in subsequent steps of Skart, the skewness of the batch means has sufficiently small magnitude so that the proposed skewness adjustment to the classical batch-means Student's t -statistic will be effective; see §§3.3.2–3.3.5.

Next Skart divides the data set $\{X_i : i = 1, \dots, n\}$ into k batches of size m , with the initial spacer consisting of $d \leftarrow 0$ ignored batches preceding each “spaced” batch from which we compute a batch mean; and the corresponding “spaced” batch means $\{Y_1(m), \dots, Y_k(m)\}$ are computed using (1.2). (The index d will be used in §3.3.2, when Skart determines the number of batches per spacer that are required to ensure the spaced batch means are approximately i.i.d.) Then Skart determines d^* , the maximum number of batches per spacer, to be used in the test for randomness of the spaced batch means in §3.3.2 based on the sample skewness of the current set of batch means. By default we take $d^* \leftarrow 10$. To get a more accurate estimator $\hat{\mathcal{B}}_m$ of the marginal skewness of the batch means with the current batch size m and to reduce the effect of any initial transient on the sample skewness computation, Skart skips the first 20% of the current set of batch means and computes the skewness only using the last 80% of the batch means as follows:

$$\ell \leftarrow \lfloor 0.8k \rfloor, \quad \bar{Y}(m, \ell) \leftarrow \frac{1}{\ell} \sum_{j=k-\ell+1}^k Y_j(m), \quad (3.4)$$

$$S_{m,\ell}^2 \leftarrow \frac{1}{\ell-1} \sum_{j=k-\ell+1}^k [Y_j(m) - \bar{Y}(m, \ell)]^2, \quad (3.5)$$

$$\hat{B}_m \leftarrow \frac{\ell}{(\ell-1)(\ell-2)} \sum_{j=k-\ell+1}^k \frac{[Y_j(m) - \bar{Y}(m, \ell)]^3}{S_{m, \ell}^3}. \quad (3.6)$$

To handle effectively an output process whose marginal skewness has excessive magnitude, Skart exploits the broadly applicable property that as $m \rightarrow \infty$, the $\{Y_j(m)\}$ are asymptotically i.i.d. normal; and since the normal distribution is determined by its moments [9], the skewness of the batch means tends to zero as $m \rightarrow \infty$. Hence, if the sample skewness \hat{B}_m of the $\{Y_j(m)\}$ in (3.6) satisfies

$$|\hat{B}_m| > 0.5, \quad (3.7)$$

then Skart sets $d^* \leftarrow 3$ as the maximum number of batches allowed per spacer to be used in the test for randomness of the spaced batch means. By doing this, Skart forces the randomness test to increase the batch size more frequently for skewed processes as explained in the next section.

To complete initialization, we take $\alpha_{\text{ran}} \leftarrow 0.2$ as the randomness test size and $b \leftarrow 0$ as the number of times the batch count has been deflated in the randomness test.

3.3.2. The Test for Randomness

In this step, Skart applies the randomness test of von Neumann [61] to the current set of $k' \leftarrow k = 1,280$ batch means by computing the ratio of the mean square successive difference of the batch means to the sample variance of the batch means. Skart applies this test iteratively to determine the size of an interbatch spacer that is sufficiently large to yield approximate independence of the corresponding spaced batch means and consequently to determine a proper batch count, batch size, and data-truncation point beyond which all the computed batch means are approximately independent of the simulation's initial conditions as well as being i.i.d.—that is, the spaced batch means constitute a random sample from a common distribution.

At the significance level α_{ran} , we test the null hypothesis that the current spaced batch means are i.i.d.,

$$\{Y_{j(d+1)}(m) : j = 1, \dots, k'\} \text{ are i.i.d.}, \quad (3.8)$$

by computing their grand mean,

$$\bar{Y}(m, k', d) = \frac{1}{k'} \sum_{j=1}^{k'} Y_{j(d+1)}(m),$$

and the corresponding randomness test statistic,

$$C_{k'} \leftarrow 1 - \frac{\sum_{j=1}^{k'-1} [Y_{j(d+1)}(m) - Y_{(j+1)(d+1)}(m)]^2}{2 \sum_{j=1}^{k'} [Y_{j(d+1)}(m) - \bar{Y}(m, k', d)]^2}, \quad (3.9)$$

which is a relocated and rescaled version of the ratio of the mean square successive difference of the spaced batch means to the sample variance of the spaced batch means. Since Skart's test for randomness usually involves a large number of spaced batch means, we use a normal approximation to the null distribution of the test statistic (3.9); see ([19], p. 303). Let z_ω denote the ω quantile of the standard normal distribution for $0 < \omega < 1$. If

$$|C_{k'}| \leq z_{1-\alpha_{\text{ran}}/2} \sqrt{(k' - 2) / [(k')^2 - 1]}, \quad (3.10)$$

then the hypothesis (3.8) is accepted; otherwise (3.8) is rejected so that Skart must increase the spacer size before retesting (3.8). We found that setting $\alpha_{\text{ran}} = 0.2$ works well in practice and provides an effective balance between errors of type I and II in testing the hypothesis (3.8).

If the randomness test is passed the first time it is executed so that the current set of batch means is $\{Y_1(m), Y_2(m), \dots, Y_{1280}(m)\}$, then we fix the batch count $k' \leftarrow 1,280$ and the batch size m and proceed to the variance-adjustment step as detailed in §3.3.3. Otherwise after executing the randomness test for the first time, we insert spacers each consisting of one ignored batch between the $k' \leftarrow 1,280/2 = 640$ remaining batches and increment the number of batches per spacer, $d \leftarrow d + 1$. Thus every other batch, beginning with the second batch, is retained as one of the spaced batch means; and the alternate batches are ignored. Now, we retest the corresponding set of $k' = 640$ spaced batch means $\{Y_2(m), Y_4(m), \dots, Y_{1280}(m)\}$ for randomness by reperforming (3.8)–(3.10) with batch size m and $d = 1$ batch per spacer.

If the latest retest of (3.8) using (3.10) is passed, then we move to the variance-adjustment step of §3.3.3 with the current values of k' , m , and d ; otherwise, we add another ignored batch to each spacer so that the total number of batches per spacer and the number of spaced batches are updated according to

$$d \leftarrow d + 1 \text{ and } k' \leftarrow \lfloor n / \{(d + 1)m\} \rfloor, \quad (3.11)$$

respectively.

If the update step (3.11) has been executed immediately after the first randomness test is performed, then we now have $k' = 426$ spaced batch means $\{Y_3(m), Y_6(m), \dots, Y_{1278}(m)\}$ with batch size m and $d = 2$ batches per spacer so that the spacer size is $dm = 2m$. Each time (3.11)

is executed, the sequence of steps in the three immediately preceding paragraphs (i.e., (3.8)–(3.10)) is reperformed until one of the following conditions occurs:

- (i) The randomness test (3.10) is failed and in the update step (3.11), we get $d > d^*$ so that the total number of batches per spacer exceeds its upper limit; or
- (ii) The randomness test is passed.

When condition (i) occurs, the batch size m is increased (inflated), the batch count k is decreased (deflated), and Skart's other status variables are updated as follows:

$$m \leftarrow \lceil \sqrt{2m} \rceil, \quad k \leftarrow \lceil 0.9k \rceil, \quad n \leftarrow km, \quad d \leftarrow 0, \quad b \leftarrow b + 1, \quad \text{and} \quad d^* \leftarrow 10. \quad (3.12)$$

The required additional observations are obtained (from the original data set, or by restarting the simulation if necessary) to complete the overall sample $\{X_i : i = 1, \dots, n\}$; and then k nonspaced batch means are computed from the overall sample according to (1.2). The updated sample skewness of the new nonspaced batch means is computed using (3.4)–(3.6). If (3.7) is satisfied, then Skart takes $d^* \leftarrow 3$ as the maximum number of batches per spacer; otherwise Skart takes $d^* \leftarrow 10$.

If condition (i) above occurs so that the update step of the previous paragraph (including (3.12)) is executed, then Skart reperforms the entire randomness-testing procedure (3.8)–(3.11), starting with the current set of k nonspaced batch means of the current batch size m so that we take $k' \leftarrow k$ and $d \leftarrow 0$ in (3.8). Skart repeats the steps outlined in the five preceding paragraphs (starting with (3.8)) until condition (ii) above finally occurs.

If the condition (ii) above occurs, then we proceed to the correlation-adjustment step detailed in §3.3.3 with the current value of d as the number of batches per spacer and the following values for the batch size, the truncated sample size, and the batch count:

$$k' \leftarrow \lceil k'(1/0.9)^b \rceil, \quad n' \leftarrow n - dm, \quad \text{and} \quad m \leftarrow \max \{ \lfloor n'/k' \rfloor, m \}, \quad (3.13)$$

where the batch count k' is reinfated by the factor $(1/0.9)^b$ to compensate for the total number of times the batch count was deflated in successive iterations of (3.12). If $mk' > n'$, then Skart requests $mk' - n'$ additional observations, which may require resuming the simulation.

Skart's approach for determining a data-truncation point (statistics clearing time, warm-up period) and appropriate values for the batch size and batch count is similar to the approaches used in WASSP and Sbatch. The observations $\{X_1, X_2, \dots, X_{dm}\}$ constituting the first spacer can be regarded

as containing the warm-up period because the spaced batch means beyond the first spacer do not exhibit significant departures from randomness—that is, they do not exhibit a deterministic trend or any type of stochastic dependence on the simulation’s initial conditions. Moreover, the spaced batch means computed beyond the first spacer are approximately i.i.d. and thus can be meaningfully used for computing the skewness adjustment to the Student’s t -statistic underlying the classical method of batch-means.

However, there are a few key differences between the randomness-testing methods used in WASSP and Sbatch on the one hand and Skart on the other. In Skart, the value of d^* , the maximum number of batches per spacer, is dynamically adjusted based on the sample skewness of the most recently computed set of nonspaced batch means, whereas WASSP and Sbatch use (different) fixed values for d^* . Skart also decreases the initial count of the nonspaced batch means each time the randomness testing is restarted with (3.12) to control the sample size growth, but in WASSP and Sbatch the batch count is constant.

3.3.3. Autocorrelation Adjustment for the Variance Estimator

Constructing a valid CI for μ_X requires an approximately unbiased estimator for the variance of the grand mean $\bar{Y}(m, k')$, where we take $k = k'$ in (1.3) to indicate that the batch means have been suitably truncated to eliminate initialization bias. In the classical NBM method, $S_{m, k'}^2/k'$ is taken as the estimator of $\text{Var}[\bar{Y}(m, k')]$; and thus the conventional $100(1 - \alpha)\%$ CI (1.7) centered at $\bar{Y}(m, k')$ is taken to have half-length $t_{1-\alpha/2, k'-1} S_{m, k'}/\sqrt{k'}$ on the assumption that the batch means are approximately i.i.d. normal variates. In practice, however, we have found that the batch means are rarely uncorrelated even when they are approximately normal [54]; and in general $S_{m, k'}^2/k'$ can be a badly biased estimator of $\text{Var}[\bar{Y}(m, k')]$ —especially when the original simulation output process $\{X_i\}$ has a positive autocorrelation function that declines slowly with increasing lags. Skart applies the autocorrelation adjustment A given by (1.11) to the variance estimator $S_{m, k'}^2/k'$ to compensate for any residual correlation between the truncated batch means, where $S_{m, k''}^2$ is an approximately unbiased estimator of the variance of the $\{Y_j(m)\}$ computed from k'' approximately i.i.d. spaced batch means separated by spacers having approximately the same size as the warm-up period (see (3.3.4) below).

To compute the autocorrelation adjustment, Skart first uses the batch count k' and the batch size m determined in (3.13) to perform the following operations on the data set $\{X_i : i = 1, \dots, n\}$ accumulated so far: (a) Skart skips the first $w = d \times m$ observations to eliminate the effects of initial-

ization bias; and (b) Skart computes the corresponding nonspaced batch means

$$\{Y_j(m) = m^{-1} \sum_{i=1}^m X_{w+(j-1)m+i} : j = 1, \dots, k'\}$$

from the truncated data set. In general the resulting nonspaced batch means will have a nonnegligible autocorrelation structure; moreover, they sometimes exhibit significant departures from normality. If the $\{Y_j(m)\}$ constitute a stationary process (a property that the randomness test was designed to ensure), then usually the batch-means process can be adequately modeled by an autoregressive–moving average (ARMA) process, at least for the purpose of estimating the autocorrelation structure of the batch means [11, 52]. In Appendix A, we prove that the lag-one autoregressive and moving-average parameters are the dominant factors in an ARMA(p, q) model for the $\{Y_j(m)\}$ as $m \rightarrow \infty$, although we ended up using a first-order autoregressive (AR(1)) model instead of an ARMA(1, 1) model to compute the autocorrelation adjustment in Skart for the reasons discussed at the end of Appendix A. Hence Skart fits an AR(1) model to the $\{Y_j(m)\}$, estimating the associated autoregressive parameter by $\hat{\varphi}_{Y(m)}$ as given in (1.12). Then $\hat{\varphi}_{Y(m)}$ is used to compute the correlation adjustment A as given in (1.11) so that $AS_{m,k'}^2/k'$ is an approximately unbiased estimator of $\text{Var}[\bar{Y}(m, k')]$.

3.3.4. Skewness Adjustment to Student's t -Statistic

When the truncated, nonspaced batch means $\{Y_j(m) : j = 1, \dots, k'\}$ exhibit significant departures from normality, Skart applies an appropriate adjustment to the usual critical value of Student's t -distribution to yield a valid CI for μ_X . As detailed in §§3.3.1–3.3.2, Skart inflates the batch size for a highly skewed process to mitigate at least partially the effect of nonnormality of the batch means on the associated NBM Student's t -statistic. In some simulation applications, however, conditions such as high congestion can induce substantial skewness in the batch means (of flow times, for example) even for batch sizes that are sufficiently large to ensure the batch means are nearly uncorrelated. Therefore, the skewness adjustment that Skart applies in this step can be crucial in delivering a CI with good coverage. Moreover, we have found that the batch-size increases imposed in previous steps of Skart are necessary to ensure that the skewness of the batch means has sufficiently small magnitude so the skewness adjustment is effective.

In this step of Skart, we adapt the skewness adjustment developed by Willink [60]. The adjustment is based on the modified t -statistic of Johnson [29] in which key terms of a Cornish-Fisher expansion involve the marginal skewness of the basic data items going into the t -statistic (in this case, batch means). It must be recognized that Willink's skewness adjustment is based on the assumption

that the basic data items going into the t -statistic are i.i.d.; but the experimental performance evaluation shows the effectiveness of this adjustment in the operation of Skart when the basic data items are correlated batch means.

To obtain an approximately unbiased estimator of the marginal skewness of the current set of truncated, nonspaced batch means $\{Y_j(m) : j = 1, \dots, k'\}$, Skart computes this skewness estimator from approximately i.i.d. spaced batch means constituting a subset of the current set of nonspaced batch means. From the randomness test in §3.3.2, we concluded that spacers consisting of w observations are sufficiently large to ensure approximate independence of the corresponding spaced batch means. Thus from the current set of k' nonspaced batch means, we can extract the spaced batch means $\{Y_1(m), Y_{d'+2}(m), \dots, Y_{(k''-1)(d'+1)+1}(m)\}$, where $d' = \lceil w/m \rceil$ is the number of batches per spacer and $k'' = 1 + \lfloor (k' - 1)/(d' + 1) \rfloor$ is the total number of spaced batch means. To simplify the following discussion, for $j = 1, \dots, k''$ we let $Y_j(m, d') \equiv Y_{(j-1)(d'+1)+1}(m)$ denote the associated spaced batch means with the following approximately unbiased estimators of their required marginal moments: the grand mean

$$\bar{Y}(m, k'', d') \leftarrow \frac{1}{k''} \sum_{j=1}^{k''} Y_j(m, d');$$

the sample variance

$$S_{m,k''}^2 \leftarrow \frac{1}{k'' - 1} \sum_{j=1}^{k''} [Y_j(m, d') - \bar{Y}(m, k'', d')]^2;$$

and sample third central moment

$$\mathcal{J}_{m,k'',d'} \leftarrow \frac{k''}{(k'' - 1)(k'' - 2)} \sum_{j=1}^{k''} [Y_j(m, d') - \bar{Y}(m, k'', d')]^3.$$

Skart uses these statistics to calculate $G(L)$ and $G(R)$, the skewness-adjusted quantile of Student's t -ratio for the left and right half-lengths of the proposed CI. The function $G(\cdot)$ is defined by taking

$$\hat{B}_{m,k''} \leftarrow \mathcal{J}_{m,k'',d'} / S_{m,k''}^3 \tag{3.14}$$

in Equation (1.9); and then L and R are defined by Equation (1.10). Thus Skart provides the correlation- and skewness-adjusted CI in Equation (1.8).

3.3.5. Fulfilling the Precision Requirement

The final step of Skart is to determine whether the constructed CI satisfies the user-specified precision requirement. The half-length of the CI (1.8) is given by

$$H \leftarrow \max\{|G(L)|, |G(R)|\} \times \sqrt{\frac{AS_{m,k''}^2}{k'}},$$

where L and R are defined in (1.10), the maximum of the two half-lengths. If the CI (1.8) satisfies the precision requirement

$$H \leq H^*, \quad (3.15)$$

where H^* is given by

$$H^* \leftarrow \begin{cases} \infty, & \text{for no user-specified precision level,} \\ r^* |\bar{X}|, & \text{for a user-specified relative precision level } r^*, \\ h^*, & \text{for a user-specified absolute precision level } h^*, \end{cases} \quad (3.16)$$

then Skart terminates, delivering the confidence interval (1.8).

If the precision requirement $H \leq H^*$ is not satisfied, then Skart estimates the total number of nonspaced batches of the current batch size that are needed to satisfy the precision requirement,

$$k^* \leftarrow \lceil (H/H^*)^2 k' \rceil;$$

and thus k^*m is our latest estimate of the total sample size beyond the truncation point that is needed to satisfy the precision requirement. The batch count k' is set for the next iteration of Skart as follows:

$$k' \leftarrow \min\{k^*, 1,024\},$$

where 1,024 is the upper bound on the number of batch means used in Skart. Our experiments showed that in those situations requiring more than 1,024 batches to achieve the desired precision, we could generally obtain better performance (in terms of the final required sample size) by increasing the batch size rather than increasing the batch count. If the projected total number of batches $k^* > 1,024$, then on the next iteration of Skart we take $k' \leftarrow 1,024$ and we update the batch size according to

$$m \leftarrow \lceil m \times \text{mid}\{1.02, (k^*/k'), 2.0\} \rceil,$$

which is assigned to satisfy the precision requirement based on an approximation to the (complicated) way in which the half-length H of the CI (1.8) depends on the batch size. This approximation is

explained in detail in §3.4 of Steiger et al. [54]. We constrain the batch-size inflation factor to lie between the limits 1.02 and 2.0 to avoid an excessive number of iterations of Skart or an excessive total sample size.

On the next iteration of Skart, the total sample size including the warm-up period is thus given by

$$n \leftarrow (d + k')m,$$

where d was finalized in the randomness test. The additional simulation-generated observations are obtained by restarting the simulation or by retrieving extra data from storage; and then the next iteration of Skart is performed by computing a new CI for μ_X using Equation (1.8) after recomputing the following: (a) the grand mean $\bar{Y}(m, k')$ of the new set of k' truncated, nonspaced batch means; (b) the associated correlation and skewness adjustments; and (c) the sample variance $S_{m, k''}^2$ of the new set of k'' truncated, spaced batch means.

3.4. N-Skart: A Nonsequential Version of Skart

Skart also has a nonsequential mode of operation in which the user merely supplies a single simulation-generated time series of an arbitrary fixed length and requests a CI with a specific coverage probability based on the available data. This procedure, called N-Skart, is designed for simulation experiments in which the size of the output data set is fixed because of a limited computing budget, a constraint on the time available for the user to complete the simulation study, or other restrictions that prevent the user from resuming the current run of the simulation model. N-Skart requires a data set of size at least 1,280, the same as for the sequential version of Skart. Figure 3.3 depicts a high-level flowchart of N-Skart.

A formal algorithmic statement of N-Skart for a data set of fixed size N is given in Figure 3.4. This algorithm has nearly the same design as the sequential algorithm given in Figure 3.2; only Steps [1], [4], and [5] are slightly changed, and Step [8] is removed completely. The modified steps are highlighted with a star next to their numbers in Figure 3.4. The necessary changes for N-Skart are summarized below:

- [1*]– This step is adjusted so that if the magnitude of the computed sample skewness \hat{B} is greater than 4.0, then the increase in the batch size m is bounded in proportion to N , the available sample size, resulting in 1,280 batches of size $m \leftarrow \min\{16, \lfloor N/1,280 \rfloor\}$.

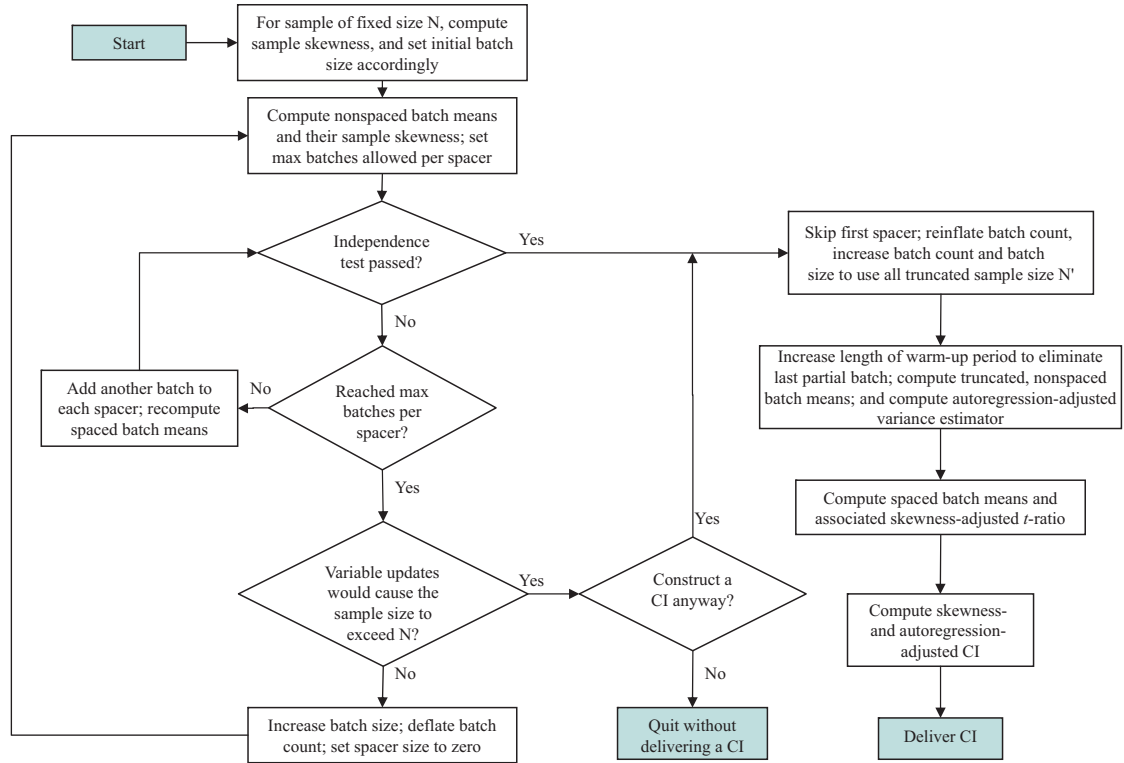


Figure 3.3: High-level flowchart of N-Skart

[4*]– In the sequential version of Skart, this step is designed to update the status variables of the procedure, such as batch size, batch count, and sample size before reapplying the randomness test. Since only limited simulation-generated observations are available in N-Skart, a feasibility check is done in this step of N-Skart to verify if the updates on the batch size and batch count would cause the sample size to exceed the available sample size N . If $\lceil \sqrt{2}m \rceil \times \lceil 0.9k \rceil \leq N$, then we update the batch size, batch count, and other status variables of the randomness test; we recompute the nonspaced batch means, and we return to Step [2]. Otherwise, N-Skart issues a warning to the user, stating that the randomness test could not be passed because of insufficient data. The warning also notes that if the user decides to continue the procedure under the given circumstances, the delivered CI might not provide the target confidence level. Here the user has two choices: (a) quit the procedure without delivering a CI; or (b) continue with constructing the

Figure 3.4: Algorithmic statement of N-Skart

[1*] From the given sample data set of size N , compute the sample skewness \hat{B} of the last 80% of the observations. If $|\hat{B}| > 4.0$, then set the initial batch size $m \leftarrow \min\{16, \lfloor N/1,280 \rfloor\}$; otherwise set $m \leftarrow 1$. Then divide the initial sample into $k \leftarrow 1,280$ nonspaced (adjacent) batches of size m . Compute the batch means $\{Y_j(m) : j = 1, \dots, k\}$. Set the current number of batches in a spacer, $d \leftarrow 0$, and the maximum number of batches allowed in a spacer, $d^* \leftarrow 10$. Set the randomness test size, $\alpha_{\text{ran}} \leftarrow 0.20$, and the number of times the batch count has been deflated in the randomness test, $b \leftarrow 0$.

[2] Compute the sample skewness \hat{B}_m of the last 80% of the current set of k nonspaced batch means with size m . If $|\hat{B}_m| > 0.5$, then reset the maximum number of batches per spacer, $d^* \leftarrow 3$.

[3] Apply the von Neumann test for randomness to the current set of k batch means with significance level α_{ran} .

[3a] If the randomness test is passed, then set $k' \leftarrow k$ and go to **[5a*]**; otherwise go to **[3b]**.

[3b] Insert spacers each with m observations (one ignored batch) between the $k' \leftarrow \lfloor k/2 \rfloor$ remaining batches; assign the values of the k' spaced batch means; and set the total number of batches in a spacer, $d \leftarrow 1$.

[3c] Apply the randomness test to the current set of k' spaced batch means with significance level α_{ran} . If the randomness test passed, then go to **[5a*]**; otherwise go to **[3d]**.

[3d] If $d = d^*$ so that the current number of batches per spacer equals the maximum number of batches per spacer, then go to **[4*]**; else add another ignored batch to each spacer so that the total number of batches per spacer and the number of spaced batches are respectively updated according to

$$d \leftarrow d + 1 \text{ and } k' \leftarrow \lfloor n / \{(d + 1)m\} \rfloor,$$

respectively. Reassign the values of the k' spaced batch means, and go to **[3c]**.

[4*] If $\lceil \sqrt{2m} \rceil \times \lceil 0.9k \rceil \leq N$, then update the batch size m , the total batch count k , the overall sample size n , and Skart's other status variables according to

$$m \leftarrow \lceil \sqrt{2m} \rceil, \quad k \leftarrow \lceil 0.9k \rceil, \quad n \leftarrow km, \quad d \leftarrow 0, \quad b \leftarrow b + 1, \text{ and } d^* \leftarrow 10;$$

recompute the k adjacent (nonspaced) batch means, and go to **[2]**. Otherwise, issue a warning that the randomness test could not be passed due to insufficient data, and ask if user wishes to continue. If the user chooses to continue with constructing a CI, then go to **[5a*]**; otherwise quit

the procedure without delivering a CI.

[5a*] Skip the first $w = d \times m$ observations in the overall sample of size N . First reinflate the batch count $k' \leftarrow \min\{\lceil k'(1/0.9)^b \rceil, k\}$; then compute the additional inflation factor $f \leftarrow \sqrt{N'/(k'm)}$ for both the batch size and batch count, and reset the truncated batch count $k' \leftarrow \min\{\lfloor f k' \rfloor, 1,024\}$. If $k' < 1024$, then reset the batch size according to $m \leftarrow \lfloor f m \rfloor$; otherwise take $m \leftarrow \lfloor N'/1,024 \rfloor$ because the maximum of $k' = 1,024$ truncated, nonspaced batch means has been reached.

[5b*] Compute the corresponding truncated, nonspaced batch means $\{Y_j(m) : j = 1, \dots, k'\}$ so that there is no partial batch left at the end,

$$Y_j(m) \leftarrow \frac{1}{m} \sum_{i=N-(k'-j)m+1}^{N-(k'-j)m} X_i \quad \text{for } j = 1, \dots, k'; \quad (3.17)$$

and update the length of the warm-up period according to $w \leftarrow w + (N' - k'm)$ so that the initial observations $\{X_i : i = 1, \dots, w\}$ are the only unused items in the entire data set of size N .

[6] Compute the grand average $\bar{Y}(m, k')$ and the sample variance $S_{m,k'}^2$ of the current set of truncated, nonspaced batch means $\{Y_j(m) : j = 1, \dots, k'\}$. Use the sample estimator of the lag-one correlation of the truncated, nonspaced batch means

$$\hat{\varphi}_{Y(m)} \leftarrow \frac{1}{k'-1} \sum_{j=1}^{k'-1} \left[Y_j(m) - \bar{Y}(m, k') \right] \left[Y_{j+1}(m) - \bar{Y}(m, k') \right] / S_{m,k'}^2$$

to compute the correlation adjustment,

$$A \leftarrow \frac{1 + \hat{\varphi}_{Y(m)}}{1 - \hat{\varphi}_{Y(m)}}$$

to the variance of the grand mean.

[7] Compute the correlation-adjusted $100(1 - \alpha)\%$ CI for μ_X using the skewness-adjusted critical values $G(t_{1-\alpha/2, k''-1})$ and $G(t_{\alpha/2, k''-1})$ of Student's t -ratio for $k'' - 1$ degrees of freedom,

$$\left[\bar{Y}(m, k') - G(t_{1-\alpha/2, k''-1}) \sqrt{\frac{AS_{m,k''}^2}{k'}}, \bar{Y}(m, k') - G(t_{\alpha/2, k''-1}) \sqrt{\frac{AS_{m,k''}^2}{k'}} \right], \quad (3.18)$$

where to evaluate (3.18) we must first compute spaced batch means with $d' = \lceil w/m \rceil$ batches per spacer so we have $k'' = 1 + \lfloor (k' - 1)/(d' + 1) \rfloor$ spaced batches of size m with corresponding spaced batch means $Y_j(m, d') \equiv Y_{(j-1)(d'+1)+1}(m)$ for $j = 1, \dots, k''$ with grand mean

$$\bar{Y}(m, k'', d') \leftarrow \frac{1}{k''} \sum_{j=1}^{k''} Y_j(m, d');$$

and sample variance and sample skewness respectively given by

$$S_{m,k''}^2 \leftarrow \frac{1}{k'' - 1} \sum_{j=1}^{k''} [Y_j(m, d') - \bar{Y}(m, k'', d')]^2; \quad \text{and} \quad (3.19)$$

$$\mathcal{J}_{m,k'',d'} \leftarrow \frac{k''}{(k'' - 1)(k'' - 2)} \sum_{j=1}^{k''} [Y_j(m, d') - \bar{Y}(m, k'', d')]^3. \quad (3.20)$$

From the latter statistics (3.19) and (3.20), we compute

$$\hat{\mathcal{B}}_{m,k''} = \frac{\mathcal{J}_{m,k'',d'}}{S_{m,k''}^3}, \quad \text{and} \quad \beta \equiv \frac{\hat{\mathcal{B}}_{m,k''}}{6\sqrt{k'}}, \quad \text{and} \quad G(\zeta) \equiv \frac{\sqrt[3]{1 + 6\beta(\zeta - \beta)} - 1}{2\beta}.$$

Deliver the CI (3.18) and stop.

requested CI by ignoring the warning.

[5a*]– In this step N-Skart skips the first $w = dm$ observations in the warm-up period, so that $N' = N - w$ approximately steady-state observations are available to build a CI for μ_X . Next the batch count k' is reinflated according to the formula $k' \leftarrow \min\{\lceil k'(1/0.9)^b \rceil, k\}$ to compensate for the total number of times the batch count was deflated in successive iterations of the randomness test. Then N-Skart computes a multiplier

$$f = \sqrt{N'/(k'm)}$$

to increase both the batch count k' and the batch size m so as to use all of the available sample size N' , subject to the constraint that $k' \leq 1,024$. Thus, N-Skart updates the count of truncated,

nonspaced batch means according to

$$k' \leftarrow \min\{\lfloor fk' \rfloor, 1,024\};$$

and the associated batch size is updated as follows:

$$m \leftarrow \begin{cases} \lfloor fm \rfloor, & \text{if } k' < 1,024, \\ \lfloor N'/1,024 \rfloor, & \text{if } k' = 1,024. \end{cases}$$

- [5b*]**– This step is specific to N-Skart. N-Skart uses formula (3.17) to compute k' batches of size m and adds the extra $N' - mk'$ observations, where $(N' - mk') < m$, to the end of warm-up period. This step enlarges the warm-up period and enhances the removal of the transients, especially in the problem instances in which the provided sample size N is fairly small and the randomness test of N-Skart is exited without being passed, as explained in Step **[4*]**.
- [8*]**– This step is removed since no absolute or relative precision requirement is considered in N-Skart and the user just expects to get the best coverage on the delivered CIs.

Chapter 4

Performance Evaluation of Skart

We conducted an extensive performance evaluation of Skart for CI estimation, including a comparison of this procedure with other well-known steady-state simulation analysis procedures—namely, ASAP3, WASSP, and SBatch. A broad diversity of test processes are considered in the experimentation, including the following: (a) processes resembling practical applications with realistic levels of complexity; and (b) processes exhibiting extremes of stochastic behavior that are commonly used to “stress-test” simulation analysis procedures. For each of the test processes, the steady-state mean is known; therefore for a given test process, we can compute the empirical coverage probabilities for the CIs delivered by each output procedure in order to evaluate the performance of the procedure and compare its performance with that of Skart. Furthermore, some specific simulation output statistics are used in the experimentation to assess the quality of Skart’s CIs for the steady-state mean of time-persistent statistics.

Beyond CI coverage probability, the performance of each output procedure is reported with respect to the following criteria: total sample size, average CI relative precision, average CI half-length, and variance of the CI half-length. Each experiment includes either 400 or 1,000 independent replications of the selected output analysis procedures for constructing 90% and 95% CIs. To provide an indication of the asymptotic performance of these output procedures, decreasing values of the relative-precision level r^* were considered for each test problem.

The standard error of each CI coverage estimator varies, depending on the number of independent replications of that CI. In case of 400 replications, for CIs with nominal 90% (respectively, 95%) coverage probability, this standard error is approximately 1.5% (respectively, 1.1%). For 1,000 replications, the standard error of the coverage estimator is approximately 0.95% (respectively, 0.69%).

To get a reasonable level of precision in the estimation of coverage probabilities, Skart was run for 1,000 replications in all the experiments. The resulting coverage error is fairly small and therefore enables us to compare the performance of Skart with that of the other output procedures and consequently find significant differences in their performance.

4.1. Performance of Skart

4.1.1. The $M/M/1$ Queue-Waiting-Time Process

Table 4.1 shows the results of applying 1,000 replications of Skart to an $M/M/1$ queue-waiting-time process with an empty-and-idle initial condition, interarrival rates ranging from 0.1 up to 0.8 customers per-time-unit, and a service rate of 1.0 customers per-time-unit. Comparable results for the other procedures are not available for traffic intensities over the entire range from 10% to 80%. These eight test problems are considered to study the performance of Skart as the interarrival rate increases, or in other words as server utilization grows, so that the correlation between successive queue waiting times increases and the system becomes more congested.

The results of the no precision case for both 90% and 95% CIs are reported in Table 4.1. These results suggest that Skart provides very close to the nominal coverage probabilities for all the reported server utilization rates. Looking at the average sample sizes in Table 4.1, we see a noticeable U-shaped trend in their values as the server utilization increases. When the server utilization is small, the sample size value is fairly high, and then it follows a decreasing trend; but later on, the sample size rapidly increases as the server utilization gets closer to one.

The logic behind this trend in the sample size highlights the basic principles on which Skart is built. For the smaller server utilizations such as 10% and 20%, the queue-waiting-time statistics have a high sample skewness due to the pronounced nonnormality of the observations—namely, the high relative frequency with which the value zero occurs among the series of observed queue waiting times. Therefore, Skart inflates the batch size in the initialization and randomness tests to mitigate the effects of nonnormality on the batch means. In queueing systems with server utilizations between 30% and 60%, neither the correlation between successive queue-waiting-times nor the skewness of those waiting times is significantly different from zero; thus Skart does not inflate the sample size very often and delivers CIs with fairly small sample sizes. As the server utilization surpasses 70%, the correlation and nonnormality effects again deviate significantly from zero, resulting in an increase in the sample size.

Table 4.1: Performance of Skart in the $M/M/1$ queue-waiting-time process for different levels of server utilization (SU) and for 90% and 95% CIs for the no precision case

Performance Measure	10% SU		20% SU		30% SU		40% SU	
	90%	95%	90%	95%	90%	95%	90%	95%
CI coverage	89.8%	94.6%	89.7%	96.1%	89.5%	94.3%	89.3%	94.1%
Avg. sample size	20,650	20,650	14,597	14,597	4,571	4,571	3,213	3,213
Avg. rel. precision	6.74%	8.09%	12.21%	14.78%	19.29%	23.83%	17.83%	21.64%
Avg. CI half-length	0.0077	0.0093	0.0331	0.0413	0.0872	0.1088	0.1261	0.1579
Var. CI half-length	0	0	0.0007	0.0013	0.0011	0.0019	0.0013	0.0028
Performance Measure	50% SU		60% SU		70% SU		80% SU	
	90%	95%	90%	95%	90%	95%	90%	95%
CI coverage	89.2%	94.9%	89.3%	94.1%	90.5%	94.3%	90.4%	94.4%
Avg. sample size	4,014	4,014	5,837	5,837	9,025	9,025	15,495	15,495
Avg. rel. precision	16.33%	19.91%	15.29%	18.69%	15.48%	18.92%	16.48%	19.73%
Avg. CI half-length	0.1762	0.2224	0.2533	0.3232	0.3964	0.4990	0.7214	0.9082
Var. CI half-length	0.0031	0.0077	0.0051	0.0167	0.0119	0.0273	0.0648	0.1561

From Table 4.1 we also observe that the total sample sizes required by Skart for 30%, 40%, and 50% server utilizations are almost equal to, or even smaller than, the *initial* sample sizes required by SBatch (16,384), WASSP (4,096), or ASAP3 (4,096).

Table 4.2 summarizes the experimental performance of Skart, ASAP3, WASSP, and SBatch when they are applied to an $M/M/1$ queue-waiting-time process for a system with an empty-and-idle initial condition, an interarrival rate of $\lambda = 0.9$ customers per-time-unit, and a service rate of $\mu = 1.0$ customers per-time-unit. In this system the steady-state server utilization is $\tau = 0.9$, the steady-state expected waiting time is $\mu_X = 9.0$ time units, and the steady-state standard deviation of the waiting time is $\sigma_X = 9.950$.

This process reaches the steady-state phase rather fast, and consequently the effect of initial-ization bias on the sample mean waiting time is insignificant. However, this process is a particularly interesting problem because in steady-state operation, we observe the following anomalies: (a) the autocorrelation function of the waiting time process decays very slowly with increasing lags; and (b) the marginal distribution of the waiting times has an exponential tail and is therefore markedly nonnormal. These characteristics result in slow convergence to the classical requirement that the batch means are i.i.d. normal as the batch size increases.

From the results in Table 4.2, it is evident that all four procedures performed reasonably well in terms of conformance to the nominal coverage probability. This was to be expected, since

Table 4.2: Performance of Skart, SBatch, WASSP, and ASAP3 in the $M/M/1$ queue-waiting-time process with 90% server utilization and empty-and-idle initial condition

Prec. Req.	Performance Measure	Nominal 90% CIs				Nominal 95% CIs			
		Skart	SBatch	WASSP	ASAP3	Skart	SBatch	WASSP	ASAP3
None	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	87.9%	87.1%	87.7%	87.5%	92.4%	91.6%	93.4%	91.5%
	Avg. sample size	41,855	54,371	18,090	31,181	41,855	54,371	17,971	31,181
	Avg. CI half-length	1.6741	1.3864	3.0715	2.0719	2.0118	1.6578	3.9987	2.5209
	Var. CI half-length	0.2282	0.2603	2.0026	0.3478	0.3607	0.3725	3.6999	0.5350
$\pm 15\%$	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	87.5%	86.6%	87.2%	91%	93.6%	91.2%	93%	95.5%
	Avg. sample size	70,473	66,719	92,049	103,742	101,730	88,447	143,920	140,052
	Avg. CI half-length	1.1905	1.1556	1.1103	1.1820	1.2284	1.2046	1.1342	1.2059
	Var. CI half-length	0.0217	0.0396	0.0387	0.0259	0.0161	0.0263	0.0314	0.0205
$\pm 7.5\%$	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	90%	88.8%	90.4%	89.5%	95.3%	94%	97%	94%
	Avg. sample size	273,540	278,642	388,000	287,568	401,816	403,844	598,020	382,958
	Avg. CI half-length	0.6396	0.6141	0.5866	0.6273	0.6397	0.6160	0.5950	0.6324
	Var. CI half-length	0.0013	0.0055	0.0072	0.0023	0.0011	0.0056	0.0056	0.0020
$\pm 3.75\%$	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	90.5%	89.8%	94%	89.5%	95.5%	95.2%	97.7%	93.5%
	Avg. sample size	1,057,080	1,151,178	1,518,400	969,011	1,492,458	1,618,147	2,361,300	1,341,522
	Avg. CI half-length	0.3223	0.3081	0.3060	0.3200	0.3232	0.3076	0.3060	0.3210
	Var. CI half-length	0.0002	0.0014	0.0008	0.0004	0.0002	0.0014	0.0007	0.0004

virtually all simulation-analysis procedures have been “tuned” to this test problem at least to some extent. As the precision level r^* became progressively smaller, Skart, ASAP3, and SBatch delivered CIs whose coverage probabilities converged to their nominal levels, while WASSP delivered CIs with some overcoverage; moreover, in this situation WASSP appeared to require substantially larger sample sizes than were required by Skart, ASAP3, or SBatch.

In the case of the $\pm 15\%$ precision requirement, Skart outperformed WASSP and ASAP3 with respect to the average sample size, while for smaller precision levels Skart’s performance was inferior to ASAP3 with respect to sampling efficiency. The results also reveal that SBatch suffered from some undercoverage at the $\pm 15\%$ precision requirement. In the no precision case while Skart still experienced some minor undercoverage, its CIs were much better behaved than those of ASAP3, WASSP, and SBatch—to be more specific, ASAP3 suffered from more serious undercoverage compared with that of Skart; WASSP delivered a CI with much higher variance of the half-length compared with that of Skart; and SBatch’s average sample size exceeded Skart’s by 30%.

To put these figures in the proper perspective, note that the corresponding results for LBTCH, ABATCH, the Law-Carson procedure, and the Heidelberg-Welch procedure are inferior to most of the

results in Table 4.2. From Table 2 of Steiger and Wilson [52] for example, ABATCH delivered the following coverage probabilities for nominal 90% CIs with the indicated relative precision levels: (i) no precision, 60%; (ii) $\pm 15\%$ precision, 72%; and (iii) $\pm 7.5\%$ precision, 82%. From Table 2 of Lada et al. [38], the corresponding coverage probabilities for the Heidelberger-Welch procedure are 67.8%, 76%, and 77%.

Next, we studied the $M/M/1$ queueing system with 90% server utilization as described above with arrival rate $\lambda = 0.9$ and service rate $\mu = 1.0$ but with an extreme initial condition in which $c = 113$ customers are assumed to be in the queue at time zero; and the first “regular” customer arrives as usual after an exponentially distributed interarrival time. This new initial condition was carefully designed to introduce a long transient in the queue-waiting-time statistic and to test the robustness of Skart, SBatch, and ASAP3 in removing severe initialization bias. Queue-waiting-time statistics are accumulated only for the “regular” customers arriving after time zero.

Let $N(t)$ denote the number of customers in the system at time t . If the initial number of customers $N(0) = c$, then it can be shown that the conditional moment generating function of the queue-waiting-time for the first regular customer is

$$M_{X_1}(t) \equiv E[e^{tX_1} | N(0) = c] = \psi^c + \frac{(1 - \psi)[1 - \psi^c(1 - t/\mu)^c]}{(1 - t/\mu)^c[1 - \psi(1 - t/\mu)]} \text{ for } t < \mu, \quad (4.1)$$

where $\psi \equiv \mu/(\lambda + \mu)$. From (4.1) it follows that

$$E[X_1 | N(0) = c] = \frac{d}{dt} M_{X_1}(0) = \frac{c}{\mu} - \frac{\psi(1 - \psi^c)}{\mu(1 - \psi)}, \quad \text{and} \quad (4.2)$$

$$\text{Var}[X_1 | N(0) = c] = \frac{d^2}{dt^2} M_{X_1}(0) - E^2[X_1 | N(0) = c], \quad (4.3)$$

where the expression for $(d^2/dt^2)M_{X_1}(0)$ in (4.3) is too complicated to display in (4.3) but is easily evaluated using Maple [42]. The derivation of (4.1) and (4.2) is given in Appendix B. The Maple code to evaluate (4.3) is also listed in Appendix B. Thus for the $M/M/1$ queue-waiting-time process with $N(0) = 113$, we have

$$E[X_1 | N(0) = 113] = 111.889, \quad \sqrt{\text{Var}[X_1 | N(0) = 113]} = 10.6881. \quad (4.4)$$

Thus we see that with the initial condition $N(0) = 113$, the expected value of the queue waiting time for the first regular customer is

$$\frac{E[X_1 | N(0) = 113] - \mu_X}{\sigma_X} \doteq 10.34 \quad (4.5)$$

standard deviations above the steady-state mean queue waiting time. It is clear from the results in (4.4) and (4.5) that this proposed $M/M/1$ queue is highly contaminated by severe initialization bias.

As Table 4.3 illustrates, Skart, SBatch, and ASAP3 all performed relatively well as far as conformance to the nominal CI coverage probabilities, although Skart showed some overcoverage for the $\pm 15\%$ precision requirement. SBatch experienced overcoverage for the $\pm 7.5\%$ precision level. Overall, ASAP3 outperformed Skart and SBatch with respect to the required average sample size. Also, as the precision requirement became smaller, Skart required relatively smaller sample sizes than SBatch required.

Table 4.3: Performance of Skart, SBatch, and ASAP3 in the $M/M/1$ queue-waiting-time process with 90% server utilization and 113 customers initially in the system

Prec. Req.	Performance Measure	Nominal 90% CIs			Nominal 95% CIs		
		Skart	SBatch	ASAP3	Skart	SBatch	ASAP3
None	# replications	1,000	1,000	1,000	1,000	1,000	1,000
	CI coverage	90.5%	92.1%	93.1%	95%	96.1%	95.7%
	Avg. sample size	84,661	111,198	57,880	84,661	111,198	57,876
	Avg. CI half-length	1.3422	1.1843	1.6330	1.6315	1.4165	1.9699
	Var. CI half-length	0.2751	0.1795	0.3970	0.4145	0.2570	0.6033
$\pm 15\%$	# replications	1,000	1,000	1,000	1,000	1,000	1,000
	CI coverage	92.5%	91.9%	91.8%	94.5%	96.3%	95.5%
	Avg. sample size	106,688	124,040	93,296	123,686	143,172	127,885
	Avg. CI half-length	1.1009	1.0729	1.1688	1.1606	1.1710	1.1967
	Var. CI half-length	0.0565	0.0544	0.0339	0.0362	0.0413	0.0240
$\pm 7.5\%$	# replications	1,000	1,000	1,000	1,000	1,000	1,000
	CI coverage	91%	93.7%	90.1%	94.5%	97.3%	95.6%
	Avg. sample size	290,403	358,533	300,386	399,023	514,722	390,574
	Avg. CI half-length	0.6367	0.6115	0.6219	0.6397	0.6101	0.6268
	Var. CI half-length	0.0025	0.0062	0.0027	0.0012	0.0063	0.0022
$\pm 3.75\%$	# replications	1,000	1,000	1,000	1,000	1,000	1,000
	CI coverage	91.5%	92.1%	89.3%	95.8%	95.3%	94.9%
	Avg. sample size	1,011,328	1,329,144	968,361	1,367,011	1,887,500	1,338,628
	Avg. CI half-length	0.3204	0.3079	0.3193	0.3166	0.3042	0.3214
	Var. CI half-length	0.0002	0.0014	0.0004	0.0001	0.0015	0.0003

Comparing the results in Tables 4.2 and 4.3, for the cases of $\pm 7.5\%$ and $\pm 3.75\%$ precision levels, we see that both Skart and ASAP3 required nearly the same average sample sizes for both the empty-and-idle and extreme initial conditions, whereas SBatch required average sample sizes that were 15%–40% larger for the initial condition $N(0) = 113$ compared with the average required sample sizes for the initial condition of $N(0) = 0$. For the no precision case the sample sizes given in Table 4.3 for Skart, ASAP3, and SBatch were roughly twice as large as the corresponding sample sizes for the empty-

Table 4.4: Performance of Skart, SBatch, WASSP, and ASAP3 in the $M/M/1$ queue-waiting-time process with 80% server utilization and empty-and-idle initial condition

Prec. Req.	Performance Measure	Nominal 90% CIs				Nominal 95% CIs			
		Skart	SBatch	WASSP	ASAP3	Skart	SBatch	WASSP	ASAP3
None	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	90.4%	89.7%	91.3%	88.8%	94.4%	95.3%	96.5%	93.3%
	Avg. sample size	15,495	67,434	38,275	41,326	15,495	67,434	35,074	41,326
	Avg. CI half-length	0.6551	0.2602	0.5400	0.3480	0.7812	0.3107	0.7060	0.4170
	Var. CI half-length	0.0264	0.0157	0.1110	0.0310	0.0373	0.0224	0.1870	0.0460
$\pm 15\%$	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	89.1%	90.4%	89%	88.5%	93.5%	95.3%	96.3%	93%
	Avg. sample size	21,006	68,778	42,497	43,796	29,983	70,225	54,578	46,106
	Avg. CI half-length	0.5327	0.2400	0.3720	0.3100	0.5402	0.2827	0.3840	0.3490
	Var. CI half-length	0.0046	0.0046	0.0070	0.0090	0.0035	0.0054	0.0060	0.0070
$\pm 7.5\%$	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	91.1%	90.3%	88.5%	86.8%	95.4%	94.2%	96.3%	93.3%
	Avg. sample size	77,050	89,434	117,540	72,060	112,206	115,037	179,840	97,643
	Avg. CI half-length	0.2845	0.2079	0.2000	0.2200	0.2842	0.2193	0.2020	0.2240
	Var. CI half-length	0.0002	0.0010	0.0010	0.0004	0.0002	0.0007	0.0008	0.0003
$\pm 3.75\%$	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	93.6%	90.3%	94%	89.5%	95.7%	95.2%	97.2%	93%
	Avg. sample size	286,566	328,768	465,160	256,186	412,386	474,520	710,070	365,353
	Avg. CI half-length	0.144	0.1116	0.1030	0.1140	0.1437	0.1109	0.1040	0.1150
	Var. CI half-length	0	0.0002	0.0002	0	0	0.0002	0.0002	0

and-idle initial condition. From the results in Tables 4.2 and 4.3, it is evident that the average sample size for the process with the extreme initial condition is higher compared with that of the process having the empty-and-idle initial condition for the cases involving the higher (coarser) levels of precision (namely, $\pm 15\%$ and $\pm 7.5\%$); and then the difference between these two average sample sizes decreases as the precision level decreases. This convergence occurs because as the precision level gets smaller, the required average sample size must increase as the inverse square of the precision level; and this inverse-square-law growth in the sample size rapidly “swamps” any effects arising from initialization bias.

Table 4.4 shows the experimental performance of Skart, ASAP3, WASSP, and SBatch when they are applied to the same $M/M/1$ queue-waiting-time process with the same empty-and-idle initial condition and service rate $\mu = 1.0$ customers per-time-unit as discussed above but with a smaller interarrival rate of $\lambda = 0.8$ customers per-time-unit. The steady-state server utilization for this system is $\tau = 0.8$, and the steady-state expected waiting time is $\mu_X = 3.2$ time units.

Based on the results in Table 4.4, we see that although all four methods delivered comparable results in terms of CI coverage, Skart had a much better sampling efficiency compared with the other procedures for the no precision case and the case of $\pm 15\%$ precision. In the latter situation, the average sample sizes required to deliver nominal 90% CIs by SBatch, WASSP, and ASAP3 exceeded Skart's average sample size by the following percentages: 227%, 102%, and 108%, respectively. ASAP3 showed a better sampling efficiency than Skart for higher precision levels; however in the case of $\pm 7.5\%$ precision with nominal 90% CIs, ASAP3 delivered 86.8% coverage, while the corresponding coverage probability for Skart was 91.1%. For this test problem, all four procedures delivered close conformance to the nominal CI coverage probabilities. From Table 4 of Lada, Steiger, and Wilson [36], the Law-Carson procedure delivered the following coverage probabilities for nominal 90% CIs: (i) no precision, 85%; (ii) $\pm 15\%$ precision, 85%; and (iii) $\pm 7.5\%$ precision, 87%. These results are inferior to most of the results in Table 4.4.

4.1.2. The $M/M/1$ Number-in-Queue Process

Table 4.5 shows the result of applying Skart to construct CIs on the number-in-queue statistic for two $M/M/1$ queueing systems, each with an empty-and-idle initial condition; interarrival rates of 0.8 and 0.9 customers per-time-unit, respectively; and each with a service rate of 1.0 customers per-time-unit. It should be mentioned here that SBatch, ASAP3, and WASSP do not have facilities for handling time-persistent statistics; but it is equally true that all these procedures can easily be adapted to yield CIs based on time-persistent statistics. The sampling interval for the $M/M/1$ number-in-queue process was set at $\Delta = 1.0$, enforcing the collection of the time-weighted (or time-averaged) number-in-queue statistic every 1.0 time units during the simulation run. Considering that the service rate is 1.0 customers per-time-unit and the interarrival rates are fairly high, it was reasonable to monitor changes in the time-weighted number-in-queue statistic every 1.0 time units.

It is evident from the results in Table 4.5 that the delivered CIs for both 80% and 90% server utilizations were in close conformance with the nominal coverage values for all the precision levels. The average simulation run time for each precision level was equal to the reported average sample size multiplied by the sampling interval Δ . Comparing the results in Table 4.5 to the results in Tables 4.2 and 4.4, we see Skart almost required the same average sample sizes to deliver CIs for the number-in-queue and queue-waiting-time statistics in both considered $M/M/1$ queueing systems.

Table 4.5: Performance of Skart in the $M/M/1$ number-in-queue process with 80% and 90% server utilization computed over 1,000 independent replications

Prec. Req.	Performance Measure	80% server utilization		90% server utilization	
		90% CIs	95% CIs	90% CIs	95% CIs
None	CI coverage	89.20%	94.30%	88.80%	93.80%
	Avg. sample size	14,700	14,700	42,211	42,211
	Avg. rel. prec.	17.7%	21.51%	20.09%	24.35%
	Avg. CI half-length	0.5631	0.6843	1.6053	1.9459
	Var. CI half-length	0.0184	0.0284	0.226	0.3401
$\pm 15\%$	CI coverage	89.50%	94.30%	88%	94.40%
	Avg. sample size	22,814	33,119	85,083	129,766
	Avg. rel. prec.	13.67%	13.78%	13.56%	13.61%
	Avg. CI half-length	0.4288	0.4343	1.07	1.101
	Var. CI half-length	0.0027	0.002	0.0176	0.0142
$\pm 7.5\%$	CI coverage	91%	94%	91.50%	95.70%
	Avg. sample size	83,268	117,855	324,574	470,936
	Avg. rel. prec.	7.14%	7.16%	7.14%	7.15%
	Avg. CI half-length	0.2269	0.2281	0.5736	0.5532
	Var. CI half-length	0.0001	0.0001	0.001	0.001
$\pm 3.75\%$	CI coverage	91.30%	96.10%	92.1%	96%
	Avg. sample size	298,774	427,169	1,022,624	1,498,953
	Avg. rel. prec.	3.54%	3.55%	3.56%	3.55%
	Avg. CI half-length	0.1142	0.1142	0.3004	0.3005
	Var. CI half-length	0	0	0.0001	0.0001

4.1.3. The $M/H_2/1$ Queue-Waiting-Time Process

Table 4.6 includes the results for the queue-waiting-time process in an $M/H_2/1$ queueing system with an empty-and-idle initial condition, a mean interarrival time of 1.0, and a hyperexponential service-time distribution that is a mixture of two exponential distributions with a common mean such that the service times have a mean of 0.8 and a coefficient of variation of 2.0. Thus in steady-state operation this system has a server utilization of $\tau = 0.8$ and a mean queue-waiting-time of $\mu_X = 8.0$ (see [36] for more details).

We concluded from Table 4.6 that in the case of no precision requirement, Skart and WASSP outperformed SBatch and ASAP3 with respect to average required sample size, while all four procedures achieved close conformance to the user-specified CI coverage probability. In the case of $\pm 15\%$ precision, all four procedures performed about the same. In the $\pm 7.5\%$ and $\pm 3.75\%$ precision cases, Skart, SBatch, and ASAP3 delivered comparable CI coverages; however, the average sample size re-

Table 4.6: Performance of Skart, SBatch, WASSP, and ASAP3 in the $M/H_2/1$ queue-waiting-time process with 80% server utilization and empty-and-idle initial condition

Prec. Req.	Performance Measure	Nominal 90% CIs				Nominal 95% CIs			
		Skart	SBatch	WASSP	ASAP3	Skart	SBatch	WASSP	ASAP3
None	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	88.2%	89.5%	91%	87.8%	93%	94.3%	93%	91.8%
	Avg. sample size	27,148	50,777	23,221	42,022	27,148	50,777	22,230	42,022
	Avg. CI half-length	1.6703	1.2135	2.7040	1.6140	2.038	1.4504	3.4560	1.9500
	Var. CI half-length	0.1972	0.2106	1.7720	0.5960	0.3431	0.3015	2.9820	0.9080
$\pm 15\%$	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	89.4%	89.2%	88.3%	88%	94.7%	93.3%	94.5%	93.3%
	Avg. sample size	62,754	65,149	78,691	76,214	88,015	84,363	138,960	96,706
	Avg. CI half-length	1.0875	1.0286	0.9930	1.0330	1.1167	1.0804	0.9940	1.0690
	Var. CI half-length	0.0161	0.0339	0.0300	0.0270	0.0097	0.0227	0.0290	0.0170
$\pm 7.5\%$	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	90%	89.6%	91%	90%	95.4%	94.7%	95.7%	94.5%
	Avg. sample size	239,976	254,400	330,580	228,482	336,319	364,154	519,990	309,560
	Avg. CI half-length	0.5686	0.5478	0.5160	0.5620	0.5693	0.5512	0.5280	0.5650
	Var. CI half-length	0.0011	0.0048	0.0060	0.0020	0.0009	0.0048	0.0020	0.0003
$\pm 3.75\%$	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	91.2%	89.5%	93%	90%	94.9%	94.7%	98%	94.7%
	Avg. sample size	875,718	1,028,683	1,283,400	798,234	1,236,649	1,396,922	2,006,800	1,115,986
	Avg. CI half-length	0.2886	0.2722	0.2700	0.2870	0.2879	0.2729	0.2700	0.2880
	Var. CI half-length	0.0001	0.0012	0.0009	0.0003	0.0001	0.0011	0.0009	0.0002

quired by ASAP3 was smaller than the average sample sizes required by Skart, SBatch, and WASSP. At all levels of precision, the CI coverage provided by Skart and SBatch were close to the nominal levels.

4.1.4. The First-Order Autoregressive (AR(1)) Process

The results shown in Table 4.7 are for applying Skart, ASAP3, WASSP, and SBatch to an AR(1) process with autoregressive parameter $\rho = 0.995$, steady-state mean $\mu_X = 100$, and the initial condition $X_0 = 0$. This AR(1) process is represented by

$$X_t = \mu_X + \rho(X_{t-1} - \mu_X) + \varepsilon_t \text{ for } t = 1, 2, \dots, \quad (4.6)$$

where $\{\varepsilon_t : t = 1, 2, \dots\} \stackrel{\text{i.i.d.}}{\sim} N(0, \sigma_\varepsilon^2)$ with $\sigma_\varepsilon^2 = 1$.

The high correlation between successive observations in this process makes it a severe test of Skart's ability to handle high correlation and to deliver an approximately valid correlation-adjusted CI.

The steady-state marginal standard deviation of the AR(1) process (4.6) is

$$\sigma_X = \sigma_\varepsilon / \sqrt{1 - \rho^2} = 10.0125; \quad (4.7)$$

and thus like the $M/M/1$ queue-waiting-time process with initial condition $N(0) = 113$, the AR(1) process (4.6) with initial condition $X_0 = 0$ starts approximately ten standard deviations away from the steady-state mean. In both processes, there is a high level of positive correlation between successive observations, and the magnitude of the resulting initialization bias is very large; however this bias is positive for the $M/M/1$ queue-waiting-time and negative for the AR(1) process. The extremely long transient in the simulation-generated realizations of this AR(1) process is also of particular interest and was purposely designed to “stress-test” Skart’s ability to eliminate initialization bias.

Table 4.7: Performance of Skart, SBatch, WASSP, and ASAP3 in the AR(1) process

Prec. Req.	Performance Measure	Nominal 90% CIs				Nominal 95% CIs			
		Skart	SBatch	WASSP	ASAP3	Skart	SBatch	WASSP	ASAP3
None	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	91%	91.5%	90.9%	95.5%	95.5%	95.6%	94.5%	98.8%
	Avg. sample size	20,632	29,831	9,866	41,076	20,632	29,831	9,824	41,076
	Avg. CI half-length	2.6618	2.1468	5.3000	2.3300	3.2097	2.5678	6.7300	2.8300
	Var. CI half-length	0.284	0.0901	1.8300	0.1700	0.4245	0.1292	2.8800	0.2700
$\pm 3.75\%$	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	91.8%	91.5%	87%	95.5%	94.9%	95.6%	95%	98.8%
	Avg. sample size	20,565	29,831	13,535	41,076	21,781	29,857	21,099	41,208
	Avg. CI half-length	2.7242	2.1468	3.2100	2.3300	3.1033	2.5653	3.2800	2.8200
	Var. CI half-length	0.24	0.0901	0.1420	0.1700	0.2095	0.1233	0.1530	0.2570
$\pm 1.875\%$	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	93.2%	91.2%	93.5%	95.5%	95.9%	95.8%	97.7%	99.3%
	Avg. sample size	48,077	42,182	57,449	68,474	68,762	61,001	90,371	101,526
	Avg. CI half-length	1.7273	1.7764	1.6500	1.7600	1.7238	1.7717	1.6600	1.7700
	Var. CI half-length	0.0226	0.0084	0.0423	0.0134	0.0236	0.0100	0.0429	0.0120
$\pm 0.9375\%$	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	93.4%	92.7%	94%	94.3%	97%	96.9%	98%	97.3%
	Avg. sample size	168,190	175,257	229,730	213,826	230,460	249,387	333,050	254,920
	Avg. CI half-length	0.9063	0.8861	0.8300	0.8940	0.9034	0.8855	0.8670	0.8960
	Var. CI half-length	0.001	0.0035	0.0105	0.0026	0.0008	0.0039	0.0115	0.0021

Table 4.7 shows that for all precision levels Skart’s sampling efficiency was better than that of SBatch and ASAP3. Notice that for this test problem, ASAP3-generated CIs exhibited significant overcoverage. In particular, at the $\pm 1.875\%$ precision level, the CI coverage of 99.3% was significantly higher than the nominal 95% level for ASAP3. The results in Table 4.7 show that while Skart experienced some overcoverage, its CIs exhibited closer conformance to the nominal coverage levels than did the CIs delivered by ASAP3 and WASSP.

For the no precision case, WASSP had the best sampling efficiency, with an average sample size of 9,866 and an empirical coverage probability of 90.9% for nominal 90% CIs, although the mean

and variance of the CI half-lengths delivered by WASSP were significantly higher than the mean and variance of the CI half-lengths provided by Skart, SBatch, and ASAP3. WASSP also had the best sampling efficiency at the $\pm 3.75\%$ relative precision level for nominal 90% CIs, requiring an average sample size of 13,535; by contrast, Skart, SBatch, and ASAP3 had average sample sizes of 21,273, 29,831 and 41,076, respectively. On the other hand, in this case WASSP delivered an empirical coverage probability of only 87%, which we judged to be unacceptable; and by contrast the empirical coverage probabilities in this case for Skart, SBatch, and ASAP3 were 91.9%, 91.5%, and 95.5%, respectively.

4.1.5. The AR(1)-to-Pareto (ARTOP) Process

The next test process we considered was the AR(1)-to-Pareto, or ARTOP, process. To generate an ARTOP time series, we initially require an AR(1) process $\{Z_j : j = 1, 2, \dots\}$ represented by

$$Z_j = \rho Z_{j-1} + b_j, \quad (4.8)$$

where $Z_0 \sim N(0, 1)$, and $\{b_j : j = 1, 2, \dots\} \stackrel{\text{i.i.d.}}{\sim} N(0, \sigma_b^2)$ is a white-noise process with variance $\sigma_b^2 = \sigma_Z^2(1 - \rho^2) = 1 - \rho^2$. This AR(1) time-series is then fed into the standard normal c.d.f. to get a sequence of correlated, Uniform(0,1), random variables $\{U_j = \Phi(Z_j) : j = 1, 2, \dots\}$, where

$$\Phi(z) = \int_{-\infty}^z \frac{1}{\sqrt{2\pi}} e^{-\xi^2/2} d\xi \text{ for all real } z$$

denotes the $N(0, 1)$ c.d.f. Next, we feed the process $\{U_j : j = 1, 2, \dots\}$ into the inverse of the Pareto c.d.f.

$$F_X(x) \equiv \Pr\{X \leq x\} = \begin{cases} 1 - (\xi/x)^\psi, & x \geq \xi, \\ 0, & x < \xi, \end{cases} \quad (4.9)$$

where $\xi > 0$ is a location parameter and $\psi > 0$ is a shape parameter, to generate the ARTOP process $\{X_j : j = 1, 2, \dots\}$ as follows,

$$X_j = F_X^{-1}(U_j) = F_X^{-1}[\Phi(Z_j)] = \xi/[1 - \Phi(Z_j)]^{1/\psi} \text{ for } j = 1, 2, \dots \quad (4.10)$$

The mean and variance of the ARTOP process (4.10) are given by

$$\mu_X = E[X_j] = \psi \xi (\psi - 1)^{-1}, \text{ for } \psi > 1,$$

and

$$\sigma_X^2 = \text{Var}[X_j] = \xi^2 \psi (\psi - 1)^{-2} (\psi - 2)^{-1}, \text{ for } \psi > 2,$$

respectively [30].

The parameters of the Pareto distribution (4.9) are set according to $\psi = 2.1$ and $\xi = 1$; and the lag-one correlation in the base process (4.8) is set to $\rho = 0.995$. This provides an ARTOP process $\{X_j : j = 1, 2, \dots\}$ whose marginal distribution has mean, variance, skewness, and kurtosis, respectively, given by

$$\mu_X = 1.9091, \quad \sigma_X^2 = 17.3554, \quad \text{E} \left[\left(\frac{X_j - \mu_X}{\sigma_X} \right)^3 \right] = \infty, \quad \text{and} \quad \text{E} \left[\left(\frac{X_j - \mu_X}{\sigma_X} \right)^4 \right] = \infty.$$

This process is particularly difficult because its marginals are highly nonnormal—in fact infinite values of the marginal skewness and kurtosis are well beyond the type of nonnormality that Skart was designed to handle. We set $Z_0 = 3.4$ to generate an ARTOP process with a fairly long transient period (i.e., to make the first observation $X_0 = \mu_X + 10\sigma_X = 43.5689$, which is ten standard deviations bigger than the steady-state mean). The results obtained for the ARTOP process are summarized in Table 4.8. It should be mentioned that the ARTOP process used for computing results of SBatch, WASSP and ASAP3 in Table 4.8 were begun in steady-state operation and thus had no transient effect.

Table 4.8: Performance of Skart, SBatch, WASSP, and ASAP3 in the ARTOP process

Prec. Req.	Performance Measure	Nominal 90% CIs				Nominal 95% CIs			
		Skart	SBatch	WASSP	ASAP3	Skart	SBatch	WASSP	ASAP3
None	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	85.8%	85.3%	79%	85.5%	90.2%	90.1%	87%	90.8%
	Avg. sample size	36,821	47,423	22,512	114,053	36,821	47,423	19,012	114,053
	Avg. CI half-length	0.4034	0.3012	0.4480	0.1730	0.5606	0.0576	0.0830	0.0144
	Var. CI half-length	1.5853	0.0403	0.0540	0.0098	1.0658	0.0576	0.0830	0.0144
±15%	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	86.2%	84.4%	71.5%	85.5%	90.5%	89.6%	81%	90.8%
	Avg. sample size	67,667	85,077	66,158	117,092	83,067	109,473	95,488	120,660
	Avg. CI half-length	0.2163	0.1940	0.2230	0.1030	0.2319	0.2121	0.2230	0.1900
	Var. CI half-length	0.0024	0.0030	0.0020	0.0025	0.0018	0.0031	0.0020	0.0024
±7.5%	# replications	1,000	1,000	1,000	400	1,000	1,000	1,000	400
	CI coverage	86.9%	82.3%	85.3%	84%	92.9%	88%	91.5%	90.3%
	Avg. sample size	213,151	306,781	345,870	186,517	370,773	460,613	520,750	255,512
	Avg. CI half-length	0.1282	0.1154	0.1160	0.1270	0.1272	0.1144	0.1200	0.1310
	Var. CI half-length	0.0002	0.0007	0.0005	0.0002	0.0002	0.0007	0.0004	0.0001
±3.75%	# replications	1,000	1,000	-	400	1,000	1,000	-	400
	CI coverage	89.7%	86.3%	-	88.8%	94.3%	93.7%	-	91%
	Avg. sample size	937,053	1,366,856	-	734,312	1,389,168	1,943,033	-	1,044,259
	Avg. CI half-length	0.0655	0.0576	-	0.0665	0.0657	0.0580	-	0.0668
	Var. CI half-length	0	0.0002	-	0	0	0.0002	-	0

Table 4.8 indicates that Skart's performance on the ARTOP process was significantly better

than that of ASAP3, SBatch, and WASSP with respect to both conformance to the nominal CI coverage probabilities and sampling efficiency. ASAP3, SBatch, and WASSP all experienced substantial undercoverage, while Skart provided coverage probabilities that were remarkably close to their nominal levels, especially for the precision requirements of $\pm 7.5\%$ and $\pm 3.75\%$. These results suggest that Skart should be robust against nonnormal marginals in many types of practical applications. Moreover, these results demonstrate the effectiveness of Skart's skewness adjustment to the classical batch-means Student's t -ratio, so that the remaining deviations of the final batch means from normality did not cause a loss of CI coverage that was either practically or statistically significant.

4.1.6. The $M/M/1/LIFO$ Queue-Waiting-Time Process

The next test process was the sequence of queue waiting times for the $M/M/1/LIFO$ queue, with customers in the queue being served in last-in-first-out (LIFO) order, a mean interarrival time of 1.0, a mean service time of 0.8, and an empty-and-idle initial condition. Thus in steady-state operation this system has a server utilization of $\tau = 0.8$ and a mean queue waiting time $\mu_X = 3.20$.

The $M/M/1/LIFO$ queue-waiting-time process was mainly selected because in steady-state operation, batch means computed from the waiting times are highly skewed—even for batch sizes that are sufficiently large to ensure the batch means are nearly uncorrelated [36].

Table 4.9 summarizes the experimental performance of Skart, SBatch, WASSP, and ASAP3 for the queue-waiting-time process in the $M/M/1/LIFO$ queueing system. These results show that Skart had a better sampling efficiency compared with that of WASSP and SBatch, especially at the less-stringent precision levels (that is, no precision and $\pm 15\%$ relative precision). The high skewness of the distribution of the batch means in the $M/M/1/LIFO$ queue-waiting-time process caused the normality test in SBatch and WASSP to be passed only after the significance level of the test had become practically negligible (that is, less than 10^{-30}), resulting in excessively inflated sample sizes. For the nominal coverage probability of 90%, Skart also demonstrated better conformance to the nominal coverage level compared with ASAP3 while usually requiring substantially smaller sample sizes. From Table 2 of Lada, Steiger, and Wilson [36], the Law-Carson procedure delivered the following coverage probabilities for nominal 90% CIs: (i) no precision, 64%; (ii) $\pm 15\%$ precision, 76%; and (iii) $\pm 7.5\%$ precision, 84%. All in all, we judged the performance of Skart to be superior to its competitors in this test problem.

Table 4.9: Performance of Skart, SBatch, WASSP, and ASAP3 in the $M/M/1/LIFO$ queue-waiting-time process with 80% server utilization

Prec. Req.	Performance Measure	Nominal 90% CIs				Nominal 95% CIs			
		Skart	SBatch	WASSP	ASAP3	Skart	SBatch	WASSP	ASAP3
None	# replications	1,000	1,000	400	400	1,000	1,000	400	400
	CI coverage	87.1%	91.4%	93%	87%	92.5%	95.9%	96%	92.5%
	Avg. sample size	23,943	117,416	125,517	53,958	23,943	117,416	124,202	53,958
	Avg. CI half-length	0.4318	0.1891	0.2650	0.1060	0.5522	0.2255	0.3350	0.3120
	Var. CI half-length	0.0326	0.0057	0.0230	0.1060	0.1015	0.0080	0.0310	0.0080
$\pm 15\%$	# replications	1,000	1,000	400	400	1,000	1,000	400	400
	CI coverage	89.2%	91.3%	90.7%	86.8%	93.9%	94%	95.2%	92.8%
	Avg. sample size	27,671	118,209	124,512	54,017	32,985	119,903	126,682	54,265
	Avg. CI half-length	0.3703	0.1812	0.2490	0.2600	0.4021	0.2117	0.2960	0.3080
	Var. CI half-length	0.0051	0.0021	0.0110	0.0040	0.0035	0.0022	0.0110	0.0050
$\pm 7.5\%$	# replications	1,000	1,000	400	400	1,000	1,000	400	400
	CI coverage	91.9%	89.5%	90.2%	87.5%	95.6%	95.4%	96.2%	92.5%
	Avg. sample size	73,986	126,961	152,355	68,325	110,520	134,123	194,590	90,911
	Avg. CI half-length	0.2263	0.1734	0.1860	0.2190	0.2253	0.1996	0.1990	0.2260
	Var. CI half-length	0.0002	0.0012	0.0020	0.0005	0.0002	0.0011	0.0010	0.0003
$\pm 3.75\%$	# replications	1,000	-	-	-	1,000	-	-	-
	CI coverage	92.4%	-	-	-	95.1%	-	-	-
	Avg. sample size	284,637	-	-	-	417,745	-	-	-
	Avg. CI half-length	0.1149	-	-	-	0.1151	-	-	-
	Var. CI half-length	0	-	-	-	0	-	-	-

4.1.7. The $M/M/1/SIRO$ Queue-Waiting-Time Process

For the next test problem, we chose an $M/M/1$ queue-waiting-time process with customers in the queue being served in random order, a mean interarrival time of 1.0, a mean service time of 0.8, and an empty-and-idle initial condition. Thus in steady-state operation this system has a server utilization of $\tau = 0.8$ and a mean queue waiting time $\mu_X = 3.20$.

Table 4.10 summarizes the performance results of applying Skart to the $M/M/1/SIRO$ queue-waiting-time process. Skart showed close conformance to the nominal CI coverage probabilities at all reported precision levels. No results are available for the performance of ASAP3, WASSP, and SBatch on this test problem.

Table 4.10: Performance of Skart in the $M/M/1/SIRO$ queue-waiting-time process with 80% server utilization based on 1,000 replications of the process

Prec. Req.	Performance Measure	Nominal 90% CIs Skart	Nominal 95% CIs Skart
None	CI coverage	88.40%	94.30%
	Avg. sample size	18,438	18,438
	Avg. rel. prec.	15.43%	18.74%
	Avg. CI half-length	0.4892	0.5952
	Var. CI half-length	0.0194	0.0267
$\pm 15\%$	CI coverage	89.60%	93.00%
	Avg. sample size	22,620	32,093
	Avg. rel. prec.	13.13%	13.46%
	Avg. CI half-length	0.4123	0.4236
	Var. CI half-length	0.0039	0.0025
$\pm 7.5\%$	CI coverage	92.90%	96.00%
	Avg. sample size	79,279	109,846
	Avg. rel. prec.	7.10%	7.16%
	Avg. CI half-length	0.2263	0.2282
	Var. CI half-length	0.0002	0.0001
$\pm 3.75\%$	CI coverage	92.10%	95.30%
	Avg. sample size	289,717	414,379
	Avg. rel. prec.	3.54%	3.55%
	Avg. CI half-length	0.1137	0.1133
	Var. CI half-length	0	0

4.1.8. The $M/M/1/M/1$ Queue-Waiting-Time Process

The next test process we considered in our experimentation was the overall queue waiting time in a system consisting of two $M/M/1$ queues in series; this is usually called the $M/M/1/M/1$ queueing system. This system has a mean interarrival time of 1.0, a mean service time of 0.8 at each server, and an empty-and-idle initial condition. Thus in steady-state operation, each server has a utilization of $\tau = 0.8$; and the expected total waiting time in both queues is 6.4.

Table 4.11 summarizes the results of applying Skart, ASAP3, and WASSP to the $M/M/1/M/1$ queue-waiting-time process. Skart and ASAP3 achieved close conformance to the specified CI coverage probabilities at all reported precision levels while requiring smaller sample sizes than WASSP required.

Table 4.11: Performance of Skart, ASAP3, and WASSP in the $M/M/1/M/1$ queue-waiting-time process with 80% server utilization

Prec. Req.	Performance Measure	Nominal 90% CIs			Nominal 95% CIs		
		Skart	ASAP3	WASSP	Skart	ASAP3	WASSP
None	# replications	1,000	400	400	1,000	400	400
	CI coverage	92.5%	91%	92%	95.9%	94.3%	96.3%
	Avg. sample size	18,457	19,133	46,462	18,457	19,133	46,462
	Avg. CI half-length	0.821	0.9830	0.7840	1.0243	1.1920	0.9880
	Var. CI half-length	0.0339	0.1650	0.2260	0.0721	0.2520	0.3580
$\pm 15\%$	# replications	1,000	400	400	1,000	400	400
	CI coverage	90.6%	91.5%	92%	95.3%	96.8%	96.3%
	Avg. sample size	19,665	25,522	48,064	29,991	32,634	52,349
	Avg. CI half-length	0.7476	0.7550	0.6400	0.7844	0.7920	0.7050
	Var. CI half-length	0.0131	0.0370	0.0440	0.0125	0.0250	0.0360
$\pm 7.5\%$	# replications	1,000	400	400	1,000	400	400
	CI coverage	90.6%	91.3%	89%	95.5%	96.3%	96.5%
	Avg. sample size	57,743	58,844	82,680	77,332	77,211	124,368
	Avg. CI half-length	0.4311	0.4410	0.3920	0.4309	0.4470	0.4000
	Var. CI half-length	0.0014	0.0017	0.0050	0.0013	0.0012	0.0040

4.1.9. Two-State Discrete-Time Markov Chains

For the last series of test processes, we used a real-valued “reward” function defined on three irreducible aperiodic discrete-time Markov chains (DTMCs), all with relatively high positive correlation structure but with marginal distributions having different levels of skewness. In particular, we considered three two-state chains $\{Z_i(j) : i = 0, 1, \dots; j = 1, 2, 3\}$ on the state space $\{0, 1\}$ with the following one-step transition probability matrices:

$$\mathbf{P}(1) = \begin{pmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{pmatrix}, \quad \mathbf{P}(2) = \begin{pmatrix} 0.99 & 0.01 \\ 0.0625 & 0.9375 \end{pmatrix}, \quad \text{and} \quad \mathbf{P}(3) = \begin{pmatrix} 0.99 & 0.01 \\ 0.25 & 0.75 \end{pmatrix},$$

respectively. The associated output processes $\{X_i(j) = h[Z_i(j)] : i = 0, 1, \dots; j = 1, 2, 3\}$ with cost vector $[h(0), h(1)]^T = (5, 10)^T$, have steady-state means $\mu_{X(1)} = 7.5$, $\mu_{X(2)} = 5.6897$, and $\mu_{X(3)} = 5.1923$, respectively. The steady-state marginal distributions for these two-state DTMCs are given by

$$\boldsymbol{\pi}(1) = (0.5, 0.5), \quad \boldsymbol{\pi}(2) = (0.8621, 0.1379), \quad \text{and} \quad \boldsymbol{\pi}(3) = (0.9615, 0.0385),$$

respectively; and the initial conditions $Z_0(1), Z_0(2), Z_0(3)$ are respectively sampled from $\boldsymbol{\pi}(1), \boldsymbol{\pi}(2), \boldsymbol{\pi}(3)$ so that each process starts in steady-state operation. The steady-state marginal skewnesses for

each of these three processes are

$$\mathcal{B}(1) = 0.0, \quad \mathcal{B}(2) = 2.1, \quad \text{and} \quad \mathcal{B}(3) = 4.8,$$

respectively. We chose these three test problems in the performance evaluation to study highly correlated output processes under discrete steady-state distributions that are symmetric, skewed, or highly skewed.

Tables 4.12–4.14 summarize the results of applying Skart, ASAP3, and SBatch to these processes. For the first DTMC with skewness $\mathcal{B}(1) = 0.0$, the results in Table 4.12 reveal that Skart outperformed both ASAP3 and SBatch in sampling efficiency at all considered precision levels. All

Table 4.12: Performance of Skart, ASAP3, and SBatch in the two-state discrete-time Markov chain with marginal skewness $\mathcal{B}(1) = 0.0$ based on 1,000 replications

Prec. Req.	Performance Measure	Nominal 90% CIs			Nominal 95% CIs		
		Skart	ASAP3	SBatch	Skart	ASAP3	SBatch
None	CI coverage	93.40%	97.10%	93.20%	98.40%	99.10%	97.00%
	Avg. sample size	8,511	9,210	42,459	8,511	9,210	42,459
	Avg. CI half-length	0.5147	0.6248	0.2221	0.6194	0.7615	0.2652
	Var. CI half-length	0.0052	0.0076	0.0014	0.0071	0.0124	0.0020
$\pm 3.75\%$	CI coverage	92.10%	96.00%	93.30%	96.10%	97.80%	96.60%
	Avg. sample size	29,343	45,520	42,952	41,840	48,489	46,022
	Avg. CI half-length	0.2677	0.2610	0.2195	0.267	0.2715	0.2503
	Var. CI half-length	0.0003	0.0005	0.0010	0.0003	0.0002	0.0006
$\pm 1.875\%$	CI coverage	91.70%	91.90%	92.70%	95.50%	96.30%	97.20%
	Avg. sample size	101,833	104,325	110,499	137,104	142,924	158,003
	Avg. CI half-length	0.1359	0.1362	0.1352	0.1369	0.1362	0.1349
	Var. CI half-length	0	0	0	0	0	0
$\pm 0.9375\%$	CI coverage	89.90%	90.50%	93.10%	94.90%	95.00%	96.10%
	Avg. sample size	358,466	363,598	445,427	501,382	517,142	625,901
	Avg. CI half-length	0.0687	0.0684	0.0658	0.0689	0.0683	0.0652
	Var. CI half-length	0	0	0	0	0	0
$\pm 0.46875\%$	CI coverage	89.20%	90.50%	89.70%	95.50%	93.60%	95.70%
	Avg. sample size	1,406,197	1,440,692	1,725,495	1,991,597	2,043,950	2,454,197
	Avg. CI half-length	0.0344	0.0342	0.0320	0.0345	0.0342	0.0319
	Var. CI half-length	0	0	0	0	0	0

procedures also exhibited some overcoverage for the cases of no precision and $\pm 3.75\%$ relative precision.

DTMCs having marginal skewness $\mathcal{B}(2) = 2.1$ or $\mathcal{B}(3) = 4.8$ are much more difficult test problems compared with the DTMC having skewness $\mathcal{B}(1) = 0.0$, because the nonnormality issue is added to the high correlation problem of the simulation output. These severe complications make the

latter two DTMCs interesting test cases for evaluating the performance of Skart, ASAP3, and SBatch under extreme conditions.

The results in Tables 4.13–4.14 show that Skart and ASAP3 outperformed SBatch in sampling efficiency. SBatch performed very poorly in the DTMCs with marginal skewness $\mathcal{B}(2) = 2.1$ or $\mathcal{B}(3) = 4.8$, requiring excessively large sample sizes. In the DTMC with marginal skewness $\mathcal{B}(3) = 4.8$, which was the most difficult of all the DTMCs we tested, the results in Table 4.14 indicate that other than for the $\pm 0.46875\%$ precision level, Skart’s sampling efficiency was superior to that of ASAP3. For the DTMC with marginal skewness $\mathcal{B}(2) = 2.1$, the results in Table 4.13 demonstrate better sampling efficiency for Skart compared with that of ASAP3 for the no precision case and the $\pm 3.75\%$ precision level; but for smaller precision levels, ASAP3 outperformed Skart.

Table 4.13: Performance of Skart, ASAP3, and SBatch in the two-state discrete-time Markov chain with marginal skewness $\mathcal{B}(2) = 2.1$ based on 1,000 replications

Prec. Req.	Performance Measure	Nominal 90% CIs			Nominal 95% CIs		
		Skart	ASAP3	SBatch	Skart	ASAP3	SBatch
None	CI coverage	91.90%	89.50%	91.40%	95.60%	94.40%	95.50%
	Avg. sample size	8,711	21,991	111,570	8,711	21,991	111,570
	Avg. CI half-length	0.1891	0.1077	0.0465	0.2283	0.1286	0.0554
	Var. CI half-length	0.0012	0.0011	0.0001	0.0015	0.0016	0.0001
$\pm 3.75\%$	CI coverage	92.60%	89.70%	91.40%	97.60%	94.30%	95.50%
	Avg. sample size	9,097	22,047	111,570	10,726	22,129	111,570
	Avg. CI half-length	0.1829	0.1065	0.0465	0.1971	0.1261	0.0554
	Var. CI half-length	0.0008	0.0008	0.0001	0.0004	0.0009	0.0001
$\pm 1.875\%$	CI coverage	92.40%	90.10%	91.40%	96.90%	94.90%	95.50%
	Avg. sample size	25,112	24,809	111,570	34,850	30,274	111,577
	Avg. CI half-length	0.102	0.0956	0.0465	0.1017	0.1016	0.0554
	Var. CI half-length	0	0.0001	0.0001	0	0	0.0001
$\pm 0.9375\%$	CI coverage	92.20%	89.10%	91.40%	95.60%	95.80%	95.30%
	Avg. sample size	85,771	81,819	114,899	120,614	114,772	127,273
	Avg. CI half-length	0.0519	0.0516	0.0452	0.052	0.0517	0.0507
	Var. CI half-length	0	0	0	0	0	0
$\pm 0.46875\%$	CI coverage	89.50%	90.00%	92.00%	95.10%	95.40%	95.50%
	Avg. sample size	324,151	320,549	343,159	455,686	457,644	491,011
	Avg. CI half-length	0.026	0.0259	0.0259	0.0261	0.0259	0.0255
	Var. CI half-length	0	0	0	0	0	0

Table 4.14: Performance of Skart, ASAP3, and SBatch in the two-state discrete-time Markov chain with marginal skewness $\mathcal{B}(3) = 4.8$ based on 1,000 replications

Prec. Req.	Performance Measure	Nominal 90% CIs			Nominal 95% CIs		
		Skart	ASAP3	SBatch	Skart	ASAP3	SBatch
None	CI coverage	90.30%	91.60%	91.50%	95.90%	94.90%	95.60%
	Avg. sample size	21,705	28,522	165,394	21,705	28,522	165,394
	Avg. CI half-length	0.0404	0.0253	0.0105	0.0502	0.0302	0.0125
	Var. CI half-length	0.0009	0	0	0.0014	0	0
$\pm 3.75\%$	CI coverage	90.30%	91.60%	91.50%	95.90%	94.90%	95.60%
	Avg. sample size	21,705	28,522	165,394	21,705	28,522	165,394
	Avg. CI half-length	0.0404	0.0253	0.0105	0.0502	0.0302	0.0125
	Var. CI half-length	0.0009	0	0	0.0014	0	0
$\pm 1.875\%$	CI coverage	93.00%	91.60%	91.50%	96.00%	94.90%	95.60%
	Avg. sample size	21,860	28,522	165,394	22,106	28,522	165,394
	Avg. CI half-length	0.0377	0.0253	0.0105	0.0424	0.0302	0.0125
	Var. CI half-length	0.0005	0	0	0.0004	0	0
$\pm 0.9375\%$	CI coverage	92.90%	91.60%	91.50%	95.50%	94.90%	95.60%
	Avg. sample size	22,682	28,525	165,394	23,720	28,536	165,394
	Avg. CI half-length	0.0314	0.0253	0.0105	0.0358	0.0301	0.0125
	Var. CI half-length	0	0	0	0	0	0
$\pm 0.46875\%$	CI coverage	90.70%	90.90%	91.50%	95.50%	94.40%	95.60%
	Avg. sample size	34,672	33,111	165,394	48,518	43,408	165,394
	Avg. CI half-length	0.0232	0.0227	0.0105	0.0234	0.0235	0.0125
	Var. CI half-length	0	0	0	0	0	0

4.2. Performance Evaluation of N-Skart

To examine the performance of N-Skart with respect to coverage probability and the mean and variance of the half-length of its CIs, we applied N-Skart to some of the test problems discussed in the previous section. We used the following sample sizes in our experiments: 10,000; 20,000; 50,000; and 200,000. These particular values were singled out to evaluate the performance of N-Skart for what might be considered “small,” “medium,” and “large” sample sizes.

Table 4.15 shows the result of applying N-Skart to the queue waiting times for the $M/M/1$, $M/H_2/1$, $M/M/1/LIFO$, and $M/M/1/SIRO$ queueing systems as well as the AR(1) and ARTOP processes to construct nominal 90% and 95% CIs. The experimentation for each test problem included 1,000 independent replications of N-Skart.

The results in Table 4.15 indicate that the coverage probabilities provided by Skart for the given sample sizes were close to their nominal levels in almost all test problems, except for the queue-

Table 4.15: Performance of N-Skart for selected test problems based on 1,000 replications

Confidence Level	Performance Measure	M/M/1 90%				M/M/1 80%			
		10,000	20,000	50,000	200,000	10,000	20,000	50,000	200,000
90%	CI coverage	87.60%	88.40%	90.30%	90.00%	90.40%	90.70%	90.40%	89.60%
	Avg. rel. precision	33.32%	25.77%	17.66%	8.64%	20.57%	14.77%	9.23%	4.39%
	Avg. CI half-length	3.1309	2.4018	1.6195	0.7791	0.8395	0.5972	0.3702	0.1757
	Var. CI half-length	3.7402	2.1298	0.4448	0.0542	0.1038	0.0522	0.0115	0.0008
95%	CI coverage	92.20%	93.10%	94.00%	94.90%	94.90%	95.80%	94.20%	94.90%
	Avg. rel. precision	39.79%	30.93%	20.83%	10.46%	24.06%	17.91%	10.85%	5.28%
	Avg. CI half-length	3.7681	2.8757	1.8963	0.9461	0.9764	0.7231	0.4363	0.2114
	Var. CI half-length	4.847	2.2475	0.5176	0.0582	0.1076	0.0503	0.015	0.0012
Confidence Level	Performance Measure	M/H ₂ /1				AR(1)			
		10,000	20,000	50,000	200,000	10,000	20,000	50,000	200,000
90%	CI coverage	88.90%	91.20%	89.60%	91.60%	93.60%	93.20%	92.90%	92.50%
	Avg. rel. precision	33.76%	25.22%	16.33%	7.95%	4.02%	2.78%	1.71%	0.81%
	Avg. CI half-length	2.7817	2.0589	1.3233	0.6386	4.0009	2.7803	1.7146	0.8136
	Var. CI half-length	2.1832	0.7285	0.2497	0.0156	1.0087	0.2751	0.0494	0.0036
95%	CI coverage	93.00%	94.20%	94.90%	95.20%	96.40%	97.40%	98.50%	97.50%
	Avg. rel. precision	40.69%	31.16%	19.67%	9.56%	4.88%	3.37%	2.04%	0.97%
	Avg. CI half-length	3.3531	2.5868	1.5845	0.7673	4.8682	3.3709	2.0408	0.9704
	Var. CI half-length	3.1949	1.3782	0.3239	0.024	1.3579	0.4296	0.0689	0.005
Confidence Level	Performance Measure	ARTOP				M/M/1/LIFO			
		10,000	20,000	50,000	200,000	10,000	20,000	50,000	200,000
90%	CI coverage	83.10%	85.30%	87.40%	88.80%	84.50%	87.20%	88.50%	89.00%
	Avg. rel. precision	26.97%	22.03%	14.69%	9.17%	17.76%	13.54%	8.89%	4.40%
	Avg. CI half-length	0.5632	0.4505	0.2878	0.179	0.5757	0.4368	0.2855	0.1411
	Var. CI half-length	0.405	0.2802	0.0587	0.049	0.0534	0.0291	0.006	0.0008
95%	CI coverage	88.40%	90.60%	91.80%	93.60%	90.70%	92.70%	93.10%	95.10%
	Avg. rel. precision	33.15%	28.63%	23.08%	11.31%	21.87%	16.64%	10.84%	5.32%
	Avg. CI half-length	0.6715	0.5897	0.4816	0.2213	0.7094	0.5373	0.3481	0.1706
	Var. CI half-length	0.3698	0.9588	1.1747	0.1093	0.092	0.0526	0.0104	0.0013
Confidence Level	Performance Measure	M/M/1/SIRO							
		10,000	20,000	50,000	200,000				
90%	CI coverage	89.00%	89.30%	89.80%	89.50%				
	Avg. rel. precision	19.22%	14.53%	9.20%	4.39%				
	Avg. CI half-length	0.6264	0.4687	0.2965	0.1405				
	Var. CI half-length	0.0504	0.0215	0.0054	0.0005				
95%	CI coverage	94.10%	95.00%	95.90%	94.50%				
	Avg. rel. precision	23.20%	17.57%	11.10%	5.29%				
	Avg. CI half-length	0.756	0.5669	0.3577	0.1692				
	Var. CI half-length	0.0747	0.0325	0.0082	0.0008				

waiting-time process in the $M/M/1/LIFO$ queue and the ARTOP process, where Skart experienced

some minor undercoverage for the sample sizes 10,000 and 20,000. The pronounced level of non-normality and stochastic dependence exhibited by the $M/M/1/LIFO$ and ARTOP processes prevented N-Skart from working effectively with such unrealistically small sample sizes as 10,000 and 20,000. As mentioned in Section 2.4.3, Skart performs better when it is applied to processes with limited marginal skewness. In the cases of the $M/M/1/LIFO$ queueing system and the ARTOP process, when the sample size is small, the batch size cannot get sufficiently large to reduce the batch-means skewness to a reasonable level. It should be mentioned here that in all the experimentation reported in Table 4.15, we simply ignored the warning message issued by N-Skart for test problems in which the randomness test could not be passed due to insufficient data; and we requested that N-Skart deliver a CI on all 1,000 independent replications of each test problem.

In general, when we are working with N-Skart, a CI with abnormally large half-length or high relative precision should alert us regarding potential problems with the delivered CIs and a possible need for bigger sample size.

4.3. Efficiency Analysis

If the steady-state variance parameter γ_X is known for an output process $\{X_i : i = 1, 2, \dots, n\}$ that is stationary and satisfies (1.4), then the nominal $100(1 - \alpha)\%$ CI for the mean of this process μ_X ,

$$\bar{X}(n) \pm z_{1-\alpha/2} \sqrt{\frac{\gamma_X}{n}}, \quad (4.11)$$

is asymptotically valid in the sense that

$$\lim_{n \rightarrow \infty} \Pr \left\{ \mu_X \in \bar{X}(n) \pm z_{1-\alpha/2} \sqrt{\frac{\gamma_X}{n}} \right\} = 1 - \alpha. \quad (4.12)$$

For a relative precision level of r^* , Chow and Robbins [15] and Nádas [43] show that for constructing a CI under the ideal conditions described above, an efficient procedure would require an expected total sample size of

$$n^* = n^*(r^*) = \lceil \gamma_X z_{1-\alpha/2}^2 / (r^* \mu_X)^2 \rceil, \quad (4.13)$$

and as $r^* \rightarrow 0$, the coverage probability of the resulting CI will converge to the limiting value $1 - \alpha$.

In this section we present the results of performing an empirical efficiency analysis for Skart, ASAP3, WASSP, and SBatch. The optimal sample size n^* was first computed for some test problems

for which the steady-state variance parameter γ_X is analytically available or can at least be evaluated numerically to a high degree of accuracy. Next we computed \bar{n}/n^* , where \bar{n} denotes the average sample size reported by Skart, ASAP3, WASSP, or SBatch for each of the test problems. Progressively smaller values of the relative precision r^* were considered in our analysis so as to estimate the sampling efficiency of Skart, ASAP3, WASSP, and SBatch as $r^* \rightarrow 0$.

Tables 4.16–4.17 summarize the efficiency analysis results for applying Skart, ASAP3, WASSP, and SBatch to the $M/M/1$ and $M/H_2/1$ queue-waiting-time processes, the AR(1) process, and the selected DTMCs when constructing nominal 90% and 95% CIs. These tables display the values of n^* , \bar{n} , the ratio \bar{n}/n^* , and the coverage probability we observed by applying Skart, ASAP3, WASSP, and SBatch to these test problems. The problem specifications and numerical details for computing the steady-state variance parameter of these test problems are provided in [36].

From Table 4.16, we can see for progressively smaller values of r^* , the ratio \bar{n}/n^* consistently approached 1.0 for both Skart and ASAP3 in all the considered test problems. However for these two procedures, the rates of convergence did not appear to be the same. For ASAP3 we observed a faster convergence of \bar{n}/n^* to 1.0 for the $M/M/1$ and $M/H_2/1$ queueing systems, and DTMCs with $\mathcal{B}(2) = 2.1$ and $\mathcal{B}(3) = 4.8$ compared with the rate of convergence achieved by Skart for all these test processes. On the other hand, in the case of the AR(1) process and the DTMC with $\mathcal{B}(1) = 0.0$, this rate was faster for Skart. One advantage of Skart over ASAP3 is that for the highest (coarsest) relative precision values we considered in this efficiency analysis—which is $\pm 15\%$ in the case of the $M/M/1$ and $M/H_2/1$ queueing systems and $\pm 3.75\%$ in the case of the AR(1) process and the DTMCs—Skart almost always required smaller sample sizes compared with ASAP3.

Willink explained in [60] that when the distribution of a given process is normal, the CIs delivered by his modified Student's t -statistic are slightly wider than the standard CIs of (1.1), which are the optimal CIs for a normal distribution. Hence, as the precision level goes to zero and the distribution of the computed batch means gets closer to normal, Skart may require sample sizes that are slightly larger than the ones required by a standard CI of form (1.1). This explains why ASAP3's sampling efficiency was somewhat better than that of Skart at the smallest levels of precision reported in Tables 4.16 and 4.17.

Table 4.16: Asymptotic efficiency of Skart and ASAP3 in the sense of Chow and Robbins based on 1,000 replications of each test process

Procedure Type	Output Process	90% CI					95% CI				
		$r^*(\%)$	n^*	\bar{n}	\bar{n}/n^*	Cover.	$r^*(\%)$	n^*	\bar{n}	\bar{n}/n^*	Cover.
Skart	M/M/1 90% SU	15	53,306	70,473	1.32	87.50%	15	75,675	101,730	1.34	93.60%
		7.5	213,222	273,540	1.28	90.00%	7.5	302,670	401,816	1.33	95.30%
		3.75	852,886	1,057,080	1.24	90.50%	3.75	1,210,797	1,492,458	1.23	95.70%
	M/M/1 80% SU	15	14,853	21,006	1.41	89.10%	15	21,086	29,983	1.42	93.50%
		7.5	59,412	77,050	1.30	91.10%	7.5	84,344	112,206	1.33	95.40%
		3.75	237,650	286,566	1.21	93.60%	3.75	337,376	412,386	1.22	95.70%
	M/H ₂ /1 80% SU	15	45,486	62,754	1.38	89.40%	15	64,574	88,015	1.36	94.70%
		7.5	181,942	239,976	1.32	90.00%	7.5	258,293	336,319	1.30	95.40%
		3.75	727,765	875,718	1.20	91.20%	3.75	1,033,169	1,236,649	1.20	94.90%
	AR(1)	3.75	7,697	20,565	2.67	91.80%	3.75	10,927	21,781	1.99	94.90%
		1.875	30,789	48,077	1.56	93.20%	1.875	43,709	68,762	1.57	95.90%
		0.0938	123,154	168,190	1.37	93.40%	0.0938	174,835	230,460	1.32	97.00%
	DTMC B(1) = 0	3.75	21,167	29,343	1.39	92.10%	3.75	30,050	41,840	1.39	96.10%
		1.875	84,669	101,833	1.20	91.70%	1.875	120,199	137,104	1.14	95.50%
		0.9375	338,674	358,466	1.06	89.90%	0.09375	480,798	501,382	1.04	94.90%
	DTMC B(2) = 2.1	3.75	4,698	9,097	1.94	92.60%	3.75	6,669	10,726	1.61	97.60%
		1.875	18,791	25,112	1.34	92.40%	1.875	26,677	34,850	1.31	96.90%
		0.9375	75,164	85,771	1.14	92.20%	0.09375	106,707	120,614	1.13	95.60%
DTMC B(3) = 4.8	3.75	442	21,705	49.15	90.30%	3.75	627	21,705	34.62	95.90%	
	1.875	1,767	21,860	12.37	93.00%	1.875	2,508	22,106	8.81	96.00%	
	0.9375	7,066	22,682	3.21	92.90%	0.09375	10,031	23,720	2.36	95.50%	
ASAP3	M/M/1 90% SU	15	53,306	103,742	1.95	91.0%	15	75,675	140,052	1.85	95.5%
		7.5	213,222	287,568	1.35	89.5%	7.5	302,670	382,958	1.27	94.0%
		3.75	852,886	969,011	1.14	89.5%	3.75	1,210,797	1,341,522	1.11	93.5%
	M/M/1 80% SU	15	14,853	43,796	2.95	88.5%	15	21,086	46,106	2.19	93.0%
		7.5	59,412	72,060	1.21	86.8%	7.5	84,344	97,643	1.16	93.3%
		3.75	237,650	256,186	1.08	89.5%	3.75	337,376	365,353	1.08	93.0%
	M/H ₂ /1 80% SU	15	45,486	76,214	1.68	88.0%	15	64,574	96,706	1.50	93.3%
		7.5	181,942	228,482	1.26	90.0%	7.5	258,293	309,560	1.20	94.5%
		3.75	727,765	798,234	1.10	90.0%	3.75	1,033,169	1,115,986	1.08	94.7%
	AR(1)	3.75	7,697	41,076	5.34	95.5%	3.75	10,927	41,208	3.77	98.8%
		1.875	30,789	68,474	2.22	95.5%	1.875	43,709	101,526	2.32	99.3%
		0.0938	123,154	213,826	1.74	94.3%	0.0938	174,835	254,920	1.46	97.3%
	DTMC B(1) = 0	3.75	21,167	45,520	2.15	96.0%	3.75	30,050	48,489	1.61	97.8%
		1.875	84,669	104,325	1.23	91.9%	1.875	120,199	142,924	1.19	96.3%
		0.9375	338,674	363,598	1.07	90.5%	0.09375	480,798	517,142	1.08	95.0%
	DTMC B(2) = 2.1	3.75	4,698	22,047	4.69	89.7%	3.75	6,669	22,129	3.32	94.3%
		1.875	18,791	24,809	1.32	90.1%	1.875	26,677	30,274	1.13	94.9%
		0.9375	75,164	81,819	1.09	89.1%	0.09375	106,707	114,772	1.08	95.8%
DTMC B(3) = 4.8	3.75	442	28,522	64.58	91.6%	3.75	627	28,522	45.49	94.9%	
	1.875	1,767	28,522	16.15	91.6%	1.875	2,508	28,522	11.37	94.9%	
	0.9375	7,066	28,525	4.04	91.6%	0.09375	10,031	28,536	2.84	94.9%	
		0.46875	28,264	33,111	1.17	90.9%	0.046875	40,125	43,408	1.08	94.4%

Table 4.17: Asymptotic efficiency of WASSP and SBatch in the sense of Chow and Robbins based on 1,000 replications of each test process

Procedure Type	Output Process	90% CI					95% CI					
		$r^*(\%)$	n^*	\bar{n}	\bar{n}/n^*	Cover.	$r^*(\%)$	n^*	\bar{n}	\bar{n}/n^*	Cover.	
WASSP	90% SU	15	53,306	92,049	1.73	87.2%	15	75,675	143,920	1.90	93.0%	
		7.5	213,222	388,000	1.82	90.4%	7.5	302,670	598,020	1.98	97.0%	
		3.75	852,886	1,518,400	1.78	94.0%	3.75	1,210,797	2,361,300	1.95	97.7%	
	80% SU	15	14,853	42,497	2.86	89.0%	15	21,086	54,578	2.59	96.3%	
		7.5	59,412	117,540	1.98	88.5%	7.5	84,344	179,840	2.13	96.3%	
		3.75	237,650	465,160	1.96	94.0%	3.75	337,376	710,070	2.10	97.2%	
	80% SU	15	45,486	78,691	1.73	88.3%	15	64,574	138,960	2.15	94.5%	
		7.5	181,942	330,580	1.82	91.0%	7.5	258,293	519,990	2.01	95.7%	
		3.75	727,765	1,283,400	1.76	93.0%	3.75	1,033,169	2,006,800	1.94	98.0%	
	AR(1)	3.75	7,697	13,535	1.76	87.0%	3.75	10,927	21,099	1.93	95.0%	
		1.875	30,789	57,449	1.87	93.5%	1.875	43,709	90,371	2.07	97.7%	
			0.0938	123,154	229,730	1.87	94.0%	0.0938	174,835	333,050	1.90	98.0%
	SBatch	90% SU	15	53,306	66,719	1.25	86.6%	15	75,675	88,447	1.17	91.2%
			7.5	213,222	278,642	1.31	88.8%	7.5	302,670	403,844	1.33	94.0%
			3.75	852,886	1,151,178	1.35	89.8%	3.75	1,210,797	1,618,147	1.34	95.2%
80% SU		15	14,853	68,778	4.63	90.4%	15	21,086	70,225	3.33	95.3%	
		7.5	59,412	89,434	1.51	90.3%	7.5	84,344	115,037	1.36	94.2%	
		3.75	237,650	328,768	1.38	90.3%	3.75	337,376	474,520	1.41	95.2%	
80% SU		15	45,486	65,149	1.43	89.2%	15	64,574	84,363	1.31	93.3%	
		7.5	181,942	254,400	1.40	89.6%	7.5	258,293	364,154	1.41	94.7%	
		3.75	727,765	1,028,683	1.41	89.5%	3.75	1,033,169	1,396,922	1.35	94.7%	
AR(1)		3.75	7,697	29,831	3.88	91.5%	3.75	10,927	29,857	2.73	95.6%	
		1.875	30,789	42,182	1.37	91.2%	1.875	43,709	61,001	1.40	95.8%	
			0.0938	123,154	175,257	1.42	92.7%	0.0938	174,835	249,387	1.43	96.9%
DTMC $\mathcal{B}(1) = 0$		3.75	442	165,394	374.51	91.5%	3.75	627	165,394	263.80	95.6%	
		1.875	1,767	165,394	93.63	91.5%	1.875	2,508	165,394	65.95	95.6%	
		0.9375	7,066	165,394	23.41	91.5%	0.09375	10,031	165,394	16.49	95.6%	
	0.46875	28,264	165,394	5.85	91.5%	0.046875	40,125	165,394	4.12	95.6%		
DTMC $\mathcal{B}(2) = 2.1$	3.75	4,698	111,570	23.75	91.4%	3.75	6,669	111,570	16.73	95.5%		
	1.875	18,791	111,570	5.94	91.4%	1.875	26,677	111,577	4.18	95.5%		
	0.9375	75,164	114,899	1.53	91.4%	0.09375	106,707	127,273	1.19	95.3%		
	0.46875	300,657	343,159	1.14	92.0%	0.046875	426,826	491,011	1.15	95.5%		
DTMC $\mathcal{B}(3) = 4.8$	3.75	21,167	42,952	2.03	93.3%	3.75	30,050	46,022	1.53	96.6%		
	1.875	84,669	110,499	1.31	92.7%	1.875	120,199	158,003	1.31	97.2%		
	0.9375	338,674	445,427	1.32	93.1%	0.09375	480,798	625,901	1.30	96.1%		
	0.46875	1,354,696	1,725,495	1.27	89.7%	0.046875	1,923,190	2,454,197	1.28	95.7%		

All the test problems reported in Tables 4.16 and 4.17 were contaminated by initialization bias, nonnormality, and stochastic dependence; and therefore these test processes were considerably different from the stationary process assumed above in determining n^* . So it should not be surprising that Skart, ASAP3, WASSP, and SBatch all required larger average sample sizes \bar{n} to eliminate the initial transients in the data and to overcome the nonnormality and dependence issues.

Table 4.17 shows that WASSP and SBatch were less efficient asymptotically compared with ASAP3 and Skart. WASSP required almost twice the corresponding average sample sizes required by ASAP3 and Skart. SBatch's sampling efficiency was also inferior to that of ASAP3 and Skart for all the considered DTMCs. The ratio \bar{n}/n^* also exhibited much more erratic behavior for WASSP and SBatch as $r^* \rightarrow 0$. For example, see the values of \bar{n}/n^* for 90% CIs for the $M/M/1$ queueing system for both WASSP and SBatch.

Chapter 5

Conclusions and Future Research

5.1. Main Conclusions of the Research

A long-standing problem in the analysis of an output process generated by a steady-state simulation is the formulation of a robust and efficient procedure to construct a valid CI for the steady-state process mean. Three primary issues impede successful analysis. The first concerns the need to remove the influence of initial conditions on the data, the second concerns the proper handling of the correlation between successive observations of the target output process, and the third concerns the proper handling of the departures from normality in the simulation-generated responses. A good procedure requires dealing with these three issues to provide not only an accurate point estimator of the selected parameter but also a sufficiently stable estimator of the standard error of the point estimator so that we are then able to construct a meaningful and reliable CI.

In this research we have developed a new, completely automated nonoverlapping batch-means method, called Skart, for constructing a correlation- and skewness-adjusted CI for the steady-state mean of a simulation output process in either discrete time (i.e., observation-based statistics) or continuous time (i.e., time-persistent statistics). Skart incorporates some advantages of its predecessors ASAP3, WASSP, and SBatch (such as the sampling efficiency of ASAP3 and the ability of WASSP and SBatch to eliminate initialization bias effectively) while exploiting separate adjustments to the classical batch-means CI based on the corresponding effects of nonnormality and correlation of the delivered batch means.

N-Skart, the nonsequential version of Skart, has also been developed in this research for

handling the test problems in which the user supplies a single simulation-generated series of arbitrary length, and the user specifies the desired coverage probability for a CI based on that series.

From the experimental results presented in Chapter 4, it is evident that Skart outperformed both SBatch and WASSP with respect to CI coverage probability and sampling efficiency. Skart and ASAP3 produced comparable results in most of the test problems studied in Chapter 4, although in the problems involving the most extreme stress testing, Skart appeared to be a more robust procedure than ASAP3 overall. In particular, we found that ASAP3 did not perform well for relatively high (coarse) precision levels when it was applied to processes with an exceptionally high correlation structure, like the AR(1) process described in Section 4.1.4. The reason for Skart’s superior performance in test problems like this is that for highly correlated processes, Skart is able to control the excessive growth of the sample size in the randomness test of von Neumann by deflating the initial batch count k during the course of the randomness test and then by reinflating the new batch count k' after the randomness test is finally passed (see Section 3.3.2 for more details).

Also Skart’s sampling efficiency seems to be better than ASAP3 for relatively high (coarse) precision levels for processes which are markedly nonnormal, like the $M/M/1/LIFO$ process as explained in Section 4.1.6. The Shapiro-Wilk normality test in SBatch and WASSP is not efficient for highly skewed processes, because this test must be applied iteratively until the significance level of the test becomes extremely small ($\sim 10^{-30}$). This feature of the normality testing procedure in SBatch and WASSP has the potential to yield excessively large sample sizes simply to “pass” the normality test, without even considering any precision requirement that might also apply to the final CI that will be delivered. Skart handles the nonnormality issue differently by applying a skewness adjustment to the Student’s t -statistic, removing the need for a normality test.

The key advantages of Skart over other steady-state simulation output procedures can be summarized as follows:

- Skart is specifically designed to handle time-persistent statistics as well as observation-based statistics.
- Skart has a nonsequential version (N-Skart) in which the user merely supplies a single simulation-generated time series of an arbitrary fixed length and requests a CI with a specific coverage probability based on the available data.
- Skart usually requires a smaller initial sample size compared with other well-known simulation analysis procedures.

- Skart removes the need for the normality test, which can sometimes result in excessive sample sizes, by applying Willink's skewness adjustment to the classical NBM-based Student's t -statistic.
- Skart efficiently controls the sample size growth in highly correlated processes by the following techniques:
 - When testing the spaced batch means for randomness, Skart effectively applies an inflation factor to the batch size while also effectively applying deflation and reinflation factors to the batch count at critical points in the testing procedure so as to determine an adequate spacer size for use as the warm-up period, etc., without requiring an excessively large sample size for the randomness test alone.
 - When testing the computed CI for conformance to the precision requirement, Skart employs a reasonably efficient stopping rule featuring rapid convergence to a final sample size that is sufficient for satisfying the specified coverage-probability and precision requirements without being excessive.

The main theoretical contribution of this research can be summarized by the following result whose proof is given in appendix A. If for some sufficiently large batch size m the nonoverlapping batch means $\{Y_j(m)\}$ computed from a stationary process with mean μ_X and variance parameter $\gamma_X \in (0, \infty)$ constitute a stationary, invertible ARMA(p, q) process with finite autoregressive order $p \geq 0$ and finite moving-average order $q \geq 0$, then as $m \rightarrow \infty$ the batch means constitute a stationary, invertible ARMA(p, p) model in which the higher-order autoregressive (respectively, moving average) coefficients tend to zero faster than the first-order autoregressive (respectively, moving-average) coefficient tends to zero.

The immediate practical consequence of this result is that as the batch size $m \rightarrow \infty$, the batch means $\{Y_j(m)\}$ can be adequately modeled as an ARMA(1, 1) process. Moreover, in practice we have found that the simpler AR(1) model provides not only an adequate asymptotic model for the batch means as $m \rightarrow \infty$ but also the basis for an effective correction to the classical NBM CI for μ_X that accounts for any remaining correlation among the batch means.

5.2. Directions for Future Research

Several future research paths can be followed based on the results of this dissertation:

- It would be desirable to establish key asymptotic properties of the CIs delivered by Skart and prove rigorously that there is a nontrivial class of discrete-event stochastic systems for which Skart's CIs are asymptotically valid—that is, the CIs delivered by Skart have coverage probabilities equal to (or no less than) the user-specified nominal levels—as the user's absolute or relative precision specification tends to zero.
- It would be desirable to prove that in some nontrivial class of discrete-event stochastic systems, the average sample sizes required by Skart to deliver its CIs exhibit convergence in some sense to the corresponding minimal (efficient) sample sizes specified by Chow and Robbins [15] as the precision requirement (r^* or H^*) tends to zero.
- Explore multivariate extensions of Skart, where one might estimate the mean of several different performance measures, providing both point and confidence-region estimators.
- Another area of interest is to enhance Skart so that it can deliver high-quality point and CI estimates of the marginal variance and user-selected quantiles of the marginal output distribution of interest.
- The results from [60] show that if a given process has approximately normal marginals, then Willink's CIs are wider than those based on the classical Student's t -statistic, which are the optimal CIs for a normal distribution. Thus, it would be desirable to modify the precision requirement step of Skart, so that asymptotically as the overall run length and the batch size increase (possibly along with the batch count), an appropriate transition is made from Skart's correlation- and skewness-adjusted CI (1.8) to the classical CI (1.1) based on i.i.d. normal batch means.
- Develop a new Cornish-Fisher expansion that simultaneously considers both skewness and correlation adjustments to the classical Student's t -statistic. This would entail a significant theoretical extension of the approach of Johnson [29] and Willink [60], which are based on the assumption that the observations (basic data items) used in the analysis are i.i.d. but not necessarily normal.

Bibliography

- [1] Ahlfors, L.V. 1966. *Complex analysis: An introduction to the theory of analytic functions of one complex variable*. 2nd ed. New York: McGraw-Hill.
- [2] Alexopoulos, C., N.T. Argon, D. Goldsman, G. Tokol, and J.R. Wilson. 2007. Overlapping variance estimators for simulation. *Operations Research* 55 (6): 1090–1103.
- [3] Alexopoulos, C., N.T. Argon, N.M. Steiger, G. Tokol, and J.R. Wilson. 2007. Efficient computation of overlapping variance estimators for simulation. *INFORMS Journal on Computing* 19:314–327.
- [4] Amemiya, T., and R.Y. Wu. 1972. The effect of aggregation on prediction in the autoregressive model. *Journal of the American Statistical Association* 67 (339): 628–632.
- [5] Anderson, T.W. 1971. *The statistical analysis of time series*. New York: Wiley
- [6] Apostol, T.M. 1974. *Mathematical analysis*. 2nd ed. Reading, Massachusetts: Addison-Wesley.
- [7] Bickel, P.J., and K.A. Doksum. 1977. *Mathematical statistics: Basic ideas and selected topics*. San Francisco: Holden-Day.
- [8] Billingsley, P. 1968. *Convergence of probability measures*. New York: John Wiley & Sons.
- [9] Billingsley, P. 1995. *Probability and measure*. 3rd ed. New York: Wiley.
- [10] Birkhoff, G. and S. MacLane. 1965. *A survey of modern algebra*. 3rd ed. New York: The Macmillan Company.
- [11] Box, G.E.P., G.M. Jenkins, and G.C. Reinsel. 1994. *Time series analysis: Forecasting and control*. 3rd ed. Englewood Cliffs, NJ: Prentice Hall.
- [12] Box, G.E.P., G.M. Jenkins, and G.C. Reinsel. 2008. *Time series analysis: Forecasting and control*. 4th ed. Englewood Cliffs, NJ: Prentice Hall.
- [13] Brewer, K.R.W. 1973. Some consequences of temporal aggregation and systematic sampling for ARMA and ARMAX models. *Journal of Economics* 1:133–154.

- [14] Chatfield, C. 2004. *The analysis of time series: An introduction*. 6th ed. Boca Raton, Florida: Chapman & Hall/CRC.
- [15] Chow, Y.S., and H. Robbins. 1965. On the asymptotic theory of fixed-width sequential confidence intervals for the mean. *The Annals of Mathematical Statistics* 36:457–462.
- [16] Crane, M.A., and A.J. Lemoine. 1977. *An introduction to the regenerative method for simulation analysis*. New York: Springer-Verlag.
- [17] Damerdj, H. 1994. Strong consistency of the variance estimator in steady-state simulation output analysis. *Mathematics of Operations Research* 19:494–512.
- [18] Fishman, G.S. 1973. *Concepts and methods in discrete event digital simulation*. New York: Wiley.
- [19] Fishman, G.S., and L.S. Yarberr. 1997. An implementation of the batch means method. *INFORMS Journal on Computing* 9 (3): 296–310.
- [20] Fishman, G.S. 2001. *Discrete-event simulation: Modeling, programming, and analysis*. New York: Springer.
- [21] Goldsman, D., and L. Schruben. 1984. Asymptotic properties of some confidence interval estimators for simulation output. *Management Science* 30 (10): 1217–1225.
- [22] Griffiths, J., J. Flanders, and C. Sells. 2003. *Mastering Visual Studio .Net*. Paris: O’Reilly.
- [23] Hall, P. 1992. On the removal of skewness by transformation. *Journal of the Royal Statistical Society, Series B* 54 (1): 221–228.
- [24] Heidelberger, P., and P.D. Welch. 1981. A spectral method for confidence interval generation and run length control in simulations. *Communications of the ACM* 24 (4): 233–245.
- [25] Heidelberger, P., and P.D. Welch. 1981. Adaptive spectral methods for simulation output analysis. *IBM Journal of Research and Development* 25:860–876.
- [26] Heidelberger, P., and P.D. Welch. 1983. Simulation run length control in the presence of an initial transient. *Operations Research* 31:1109–1144.
- [27] Iglehart, D.L. 1975. Simulating stable stochastic systems, V: Comparison of ratio estimators. *Naval Research Logistics Quarterly* 22:553–565.
- [28] Iglehart, D.L. 1978. The regenerative method for simulation analysis, in *Current trends in programming methodology, Vol. III: Software modeling*. K.M. Chandy and R.T. Yeh, ed., pp. 52–71. New Jersey: Prentice-Hall, Inc.
- [29] Johnson, N.J. 1978. Modified t tests and confidence intervals for asymmetrical populations. *Journal of the American Statistical Association* 73 (363): 536–544.

- [30] Johnson, N.L., Kotz, S., and N. Balakrishnan. 1994. *Continuous univariate distributions, Vol. 1* 2nd ed. New York: John Wiley & Sons.
- [31] Kang, K., and B.W. Schmeiser. 1987. Properties of batch means from stationary ARMA time series. *Oper. Res. Lett.* 6(1): 19–24.
- [32] Kelton, W.D., Sadowski, R.P., and D.T. Sturrock. 2004. *Simulation with Arena*. 3rd ed. Boston: McGraw-Hill.
- [33] Kleijnen, J.P.C., Kloppenborg, G.L.J., and F.L. Meeuwssen. 1986. Testing the mean of an asymmetric population: Johnson's modified t test revisited. *Communications in Statistics: Theory and Methods* 15 (3): 715–732.
- [34] Knopp, K. 1952. *Elements of the theory of functions*. New York: Dover.
- [35] Koopmans, L.H. 1974. *The spectral analysis of time series*. New York: Academic Press.
- [36] Lada, E.K., N.M. Steiger, and J.R. Wilson. 2006. Performance evaluation of recent procedures for steady-state simulation analysis. *IIE Transactions* 38:711–727.
- [37] Lada, E.K., and J.R. Wilson. 2006. A wavelet-based spectral procedure for steady-state simulation analysis. *European Journal of Operational Research* 174 (3): 1769–1801.
- [38] Lada, E.K., J.R. Wilson, N.M. Steiger, and J.A. Joines. 2007. Performance of a wavelet-based spectral procedure for steady-state simulation analysis. *INFORMS Journal on Computing* 19 (2): 150–160.
- [39] Lada, E.K., N.M. Steiger, and J.R. Wilson. 2008. SBatch: A spaced batch means procedure for steady-state simulation analysis. *Journal of Simulation* forthcoming. Available online via <http://www.ise.ncsu.edu/jwilson/files/lada08jos.pdf> [accessed June 20, 2008].
- [40] Law, A.M., and J.S. Carson. 1979. A sequential procedure for determining the length of a steady-state simulation. *Operations Research* 27 (5): 1011–1025.
- [41] Law, A.M., and W.D. Kelton. 2000. *Simulation modeling and analysis*. 3rd ed. New York: McGraw-Hill.
- [42] Maplesoft. (2003). *Maple 9 Learning Guide*. Toronto: Waterloo Maple Inc.
- [43] Nádas, A. 1969. An extension of a theorem of Chow and Robbins on sequential confidence intervals for the mean. *The Annals of Mathematical Statistics* 40 (2): 667–671.
- [44] *Oxford English Dictionary*. 1989. 2nd ed. Oxford: Oxford University Press.
- [45] Priestley, M.B. 1981. *Spectral analysis and time series analysis*. London: Academic Press.
- [46] Rissanen, J. 1986. Order estimation of accumulated prediction errors, in *Essays in Time Series and*

- Allied Processes*, ed. J. Gani and M.B. Priestley, *Journal of Applied Probability*, 23 (A): 55–61.
- [47] Ross, S.M. 1997. *Introduction to probability models*. 6th ed. San Diego: Academic Press.
- [48] Schruben, L.W. 1983. Confidence interval estimation using standardized time series. *Operations Research* 31:1090–1108.
- [49] Shapiro, S.S., and M.B. Wilk. 1965. An analysis of variance test for normality (complete samples). *Biometrika* 52 (3/4): 591–611.
- [50] Silvestrini, A., and D. Veredas. 2008. Temporal aggregation of univariate and multivariate time series models: A survey. *Journal of Economic Surveys* 22 (3): 458–497.
- [51] Steiger, N.M., and J.R. Wilson. 2001. Convergence properties of the batch means method for simulation output analysis. *INFORMS Journal on Computing* 13 (4): 277–293.
- [52] Steiger, N.M., and J.R. Wilson. 2002. An improved batch means procedure for simulation output analysis. *Management Science* 48 (12): 1569–1586.
- [53] Steiger, N.M., E.K. Lada, J.R. Wilson, C. Alexopoulos, D. Goldsman., and F. Zouaoui. 2002. ASAP2: An improved batch means procedure for simulation output analysis, in *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J.L. Snowdon, and J.M. Charnes, pp. 336–344. Institute of Electrical and Electronics Engineers, Piscataway, NJ. Available online via www.informs-sim.org/wsc02papers/043.pdf (accessed June 20, 2008).
- [54] Steiger, N.M., E.K. Lada, J.R. Wilson, J.A. Joines, C. Alexopoulos, and D. Goldsman. 2005. ASAP3: A batch means procedure for steady-state simulation analysis. *ACM Transactions on Modeling and Computer Simulation* 15 (1): 39–73.
- [55] Stuart, A., and J.K. Ord. 1994. *Kendall's Advanced Theory of Statistics, Volume 1: Distribution Theory*. 6th ed. London: Edward Arnold.
- [56] Tesler, L.G. 1967. Discrete samples and moving sums in stationary stochastic processes. *Journal of the American Statistical Association* 62 (318): 484–499.
- [57] Tiao, G.C. 1972. Asymptotic behaviour of temporal aggregates of time series. *Biometrika* 59 (3): 525–531.
- [58] Wallace, D.L. 1958. Asymptotic approximations to distributions. *The Annals of Mathematical Statistics* 29 (3): 635–654.
- [59] Welch, P.D. 1983. The statistical analysis of simulation results, in *The computer performance modeling handbook*, ed. S.S. Lavenberg, pp. 268–328. New York: Academic Press.
- [60] Willink, R. 2005. A confidence interval and test for the mean of an asymmetric distribution. *Com-*

munications in Statistics—Theory and Methods 34:753–766.

- [61] von Neumann, J. 1941. Distribution of the ratio of the mean square successive difference to the variance. *The Annals of Mathematical Statistics* 12:367–395.
- [62] Yuan, M. and B.L. Nelson. 1994. Autoregressive-output-analysis methods revisited. *Annals of Operations Research* 53:391–418.

Appendices

Appendix A

Asymptotic Properties of Batch Means from Stationary, Invertible ARMA Processes with Increasing Batch Size

A.1 Motivation, Assumptions, and Basic Results about Batch-Means Processes

If $\{X_i\}$ is the output of a simulation model in steady-state operation, then under fairly general conditions the corresponding sequence of batch means $\{Y_j(m) = m^{-1} \sum_{i=(j-1)m+1}^{jm} X_i : j = 1, 2, \dots\}$ is asymptotically Gaussian as the batch size $m \rightarrow \infty$; see, for example, Steiger and Wilson [51]. Moreover, in practice many stationary Gaussian time series can be adequately modeled by a mixed autoregressive–moving average (ARMA) process of order (p, q) , provided the autoregressive order p and the moving-average order q are chosen properly; see, for example, p. 47 of Chatfield [14] and p. 140 of Priestley [45].

If the original (unbatched) process $\{X_i\}$ is a stationary and invertible ARMA(p^*, q^*) process with $0 \leq p^*, q^* < \infty$, then Tiao [57] shows that as the batch size $m \rightarrow \infty$, the batch means $\{Y_j(m)\}$ are asymptotically uncorrelated. In the following theorem we provide a simpler alternative proof of this result under what is arguably the most broadly applicable set of hypotheses.

THEOREM 1. *If the process $\{X_i\}$ is strictly stationary with mean μ_X and finite variance parameter γ_X ,*

then for the associated nonoverlapping batch means $\{Y_j(m)\}$ with batch size m , the lag- ℓ autocorrelation function,

$$\rho_\ell(m) = \text{Corr}[Y_j(m), Y_{j+\ell}(m)] \text{ for } \ell = 0, \pm 1, \pm 2, \dots, \quad (\text{A.1})$$

satisfies

$$\lim_{m \rightarrow \infty} \rho_\ell(m) = 0 \text{ for } \ell = \pm 1, \pm 2, \dots. \quad (\text{A.2})$$

Moreover if the process $\{X_i\}$ is ϕ -mixing with mixing coefficients $\{\phi_i\}$ satisfying

$$\sum_{i=0}^{\infty} \sqrt{\phi_i} < \infty, \quad (\text{A.3})$$

then

$$|\rho_{-\ell}(m)| = |\rho_\ell(m)| \leq 2\sqrt{\phi_{(\ell-1)m+1}} \xrightarrow{m \rightarrow \infty} 0 \text{ for } \ell = 2, 3, \dots; \quad (\text{A.4})$$

and for a fixed batch count k , the standardized nonoverlapping batch means $\{\sqrt{m/\gamma_X}[Y_j(m) - \mu_X] : j = 1, \dots, k\}$ are asymptotically i.i.d. standard normal random variables as the batch size $m \rightarrow \infty$:

$$\sqrt{m/\gamma_X}[Y_1(m) - \mu_X, \dots, Y_k(m) - \mu_X] \xrightarrow[m \rightarrow \infty]{D} N(\mathbf{0}_k, \mathbf{I}_k), \quad (\text{A.5})$$

where $\mathbf{0}_k$ denotes the $k \times 1$ null vector and \mathbf{I}_k denotes the $k \times k$ identity matrix.

Proof. The proof of (A.2) is by induction on ℓ for positive values of ℓ . From the well-known relation

$$\text{Var}[\bar{Y}(m, \ell)] = \frac{\text{Var}[Y(m)]}{\ell} \left[1 + 2 \sum_{u=1}^{\ell-1} \left(1 - \frac{u}{\ell}\right) \rho_u(m) \right] \text{ for } \ell = 2, 3, \dots, \quad (\text{A.6})$$

it follows that for $\ell = 2$ we have

$$\frac{\text{Var}[\bar{Y}(m, 2)]}{\text{Var}[Y(m)]/2} - 1 = \rho_1(m). \quad (\text{A.7})$$

From

$$\gamma_X = \lim_{m \rightarrow \infty} \ell m \text{Var}[\bar{Y}(m, \ell)] = \lim_{m \rightarrow \infty} m \text{Var}[Y(m)] \text{ for } \ell = 1, 2, \dots, \quad (\text{A.8})$$

we see that

$$\begin{aligned} \lim_{m \rightarrow \infty} \rho_1(m) &= \lim_{m \rightarrow \infty} \frac{\text{Var}[\bar{Y}(m, 2)]}{\text{Var}[Y(m)]/2} - 1 = \lim_{m \rightarrow \infty} \frac{2m \text{Var}[\bar{Y}(m, 2)]}{m \text{Var}[Y(m)]} - 1 \\ &= (\gamma_X/\gamma_X) - 1 = 0. \end{aligned} \quad (\text{A.9})$$

Now suppose that we have

$$\lim_{m \rightarrow \infty} \rho_\ell(m) = 0 \text{ for } \ell = 1, \dots, L, \quad (\text{A.10})$$

where $L \geq 1$. From (A.6) we have

$$\frac{\text{Var}[\bar{Y}(m, L+2)]}{\text{Var}[Y(m)]/(L+2)} - \frac{\text{Var}[\bar{Y}(m, L+1)]}{\text{Var}[Y(m)]/(L+1)} = \left[\frac{2}{(L+1)(L+2)} \sum_{u=1}^L u \rho_u(m) \right] - \frac{2}{L+2} \rho_{L+1}(m). \quad (\text{A.11})$$

Letting $m \rightarrow \infty$ in (A.11), we see from (A.8) that the left-hand side of (A.11) vanishes asymptotically,

$$\begin{aligned} & \lim_{m \rightarrow \infty} \left\{ \frac{\text{Var}[\bar{Y}(m, L+2)]}{\text{Var}[Y(m)]/(L+2)} - \frac{\text{Var}[\bar{Y}(m, L+1)]}{\text{Var}[Y(m)]/(L+1)} \right\} \\ &= \lim_{m \rightarrow \infty} \frac{(L+2)m \text{Var}[\bar{Y}(m, L+2)]}{m \text{Var}[Y(m)]} - \lim_{m \rightarrow \infty} \frac{(L+1)m \text{Var}[\bar{Y}(m, L+1)]}{m \text{Var}[Y(m)]} \\ &= (\gamma_X/\gamma_X) - (\gamma_X/\gamma_X) = 0. \end{aligned} \quad (\text{A.12})$$

From the induction hypothesis (A.10), we see that the first term on the right-hand side of (A.11) tends to zero as $m \rightarrow \infty$,

$$\lim_{m \rightarrow \infty} \frac{2}{(L+1)(L+2)} \sum_{u=1}^L u \rho_u(m) = \frac{2}{(L+1)(L+2)} \sum_{u=1}^L u \cdot \left[\lim_{m \rightarrow \infty} \rho_u(m) \right] = 0; \quad (\text{A.13})$$

and combining (A.11), (A.12), and (A.13), we finally obtain

$$\lim_{m \rightarrow \infty} \rho_{L+1}(m) = 0. \quad (\text{A.14})$$

The conclusion (A.2) follows by induction.

Now let \mathcal{M}_t^u denote the σ -field (or Borel-field) generated by the random variables $\{X_i : i = t, t+1, \dots, u\}$; and let \mathcal{M}_t^∞ denote the σ -field generated by the random variables $\{X_i : i = t, t+1, \dots\}$; see, for example, Billingsley [8, 9]. Since the $\{Y_j(m)\}$ are computed from nonoverlapping batches of size m , we see that $Y_j(m)$ is measurable with respect to \mathcal{M}_1^{jm} ; and for $\ell \geq 1$, the lagged batch mean $Y_{j+\ell}(m)$ is measurable with respect to $\mathcal{M}_{(j+\ell-1)m+1}^\infty$. It follows immediately from Equation (20.23) of Billingsley [8] that

$$|\rho_{-\ell}(m)| = |\rho_\ell(m)| \leq 2\sqrt{\phi_{(\ell-1)m+1}} \quad \text{for } \ell = 1, 2, \dots. \quad (\text{A.15})$$

The hypothesis (A.3) ensures that

$$\lim_{i \rightarrow \infty} \sqrt{\phi_i} = 0; \quad (\text{A.16})$$

and thus (A.4) follows immediately from (A.15). Finally, the limiting result (A.5) is the main conclusion of Theorem 1 of Steiger and Wilson [51]. ■

REMARK 1. It might seem plausible that the Central Limit Theorem (1.4) specialized to each of the standardized batch means $\{\sqrt{m/\gamma_X}[Y_j(m) - \mu_X] : j = 1, \dots, k\}$ and coupled with the result (A.2) should be sufficient to yield the multivariate limiting property (A.5). Unfortunately, a k -dimensional random vector with normal marginals and all pairwise correlations equal to zero does not necessarily possess a k -dimensional normal distribution; see, for example, Exercise 15.20 of Stuart and Ord [55]. Some additional hypothesis is needed to ensure that the dependence between the lagged batch means $Y_j(m)$ and $Y_{j+\ell}(m)$ falls off sufficiently fast as $|\ell| \rightarrow \infty$. An alternative to the assumption (A.3) is the assumption FCLT (2.10) that underlies the method of standardized time series; unfortunately neither of these assumptions is easily checked in practice.

REMARK 2. Equation (A.2) will play a key role in the rest of the development, and it is similar to Equation (2.8) of [57]. However, the only assumption required to obtain (A.2) is that $\{X_i\}$ is stationary with a finite variance parameter. By contrast, the derivation of Equation (2.8) of [57] is arguably more complicated and requires the stronger assumption that $\{X_i\}$ is a stationary and invertible ARMA(p^*, q^*) process for $0 \leq p^*, q^* < \infty$.

All the foregoing considerations form the basis for our fundamental assumption that the underlying process $\{X_i\}$ is such that with a sufficiently large batch size, the stochastic behavior of the associated batch means can be accurately approximated by an ARMA(p, q) process for finite nonnegative values of p and q . Specifically, we assume that for any batch size $m \geq m_0$ (where m_0 is a “base” batch size taken sufficiently large), the batch means $\{Y_j(m) : j = 1, 2, \dots\}$ can be adequately modeled by an ARMA(p, q) process

$$Y_j(m) = \mu_X + \sum_{\ell=1}^p \varphi_\ell(m)[Y_{j-\ell}(m) - \mu_X] + \varepsilon_j(m) - \sum_{\ell=1}^q \theta_\ell(m)\varepsilon_{j-\ell}(m), \quad (\text{A.17})$$

where: p and q depend on m_0 ; we have $0 \leq p, q < \infty$; the random “shocks” $\{\varepsilon_j(m) : j = 1, 2, \dots\}$ driving the process are randomly sampled from $N[0, \sigma_{\varepsilon(m)}^2]$; and the ARMA model (A.17) satisfies the following conditions—

- **Stationarity:** The roots of the degree- p autoregressive polynomial

$$\Phi_{p,m}(z) \equiv \begin{cases} 1, & \text{if } p = 0, \\ 1 - \sum_{\ell=1}^p \varphi_\ell(m)z^\ell, & \text{if } p \geq 1, \end{cases} \quad (\text{A.18})$$

must lie outside the unit circle in the complex plane

$$\mathbb{C} \equiv \{z = x + y\sqrt{-1} : x, y \in \mathbb{R}\}. \quad (\text{A.19})$$

- **Invertibility:** The roots of the degree- q moving-average polynomial

$$\Theta_{q,m}(z) \equiv \begin{cases} 1, & \text{if } q = 0, \\ 1 - \sum_{\ell=1}^q \theta_{\ell}(m)z^{\ell}, & \text{if } q \geq 1, \end{cases} \quad (\text{A.20})$$

must lie outside the unit circle in \mathbb{C} .

- **Unicity:** The polynomials $\Phi_{p,m}(z)$ and $\Theta_{q,m}(z)$ have no common roots and no multiple roots. Moreover any two roots of $\Phi_{p,m}(z)$ that are not complex conjugates of each other must have distinct absolute values. A similar condition applies to the roots of the moving-average polynomial $\Theta_{q,m}(z)$.

Recall that each complex number

$$z = x + y\sqrt{-1} \in \mathbb{C} \quad (\text{A.21})$$

has complex conjugate

$$\bar{z} \equiv x - y\sqrt{-1} \quad (\text{A.22})$$

and absolute value

$$|z| \equiv \sqrt{z\bar{z}} = \sqrt{x^2 + y^2}. \quad (\text{A.23})$$

In the subsequent discussion, we let

$$\mathbb{D}(R) \equiv \{z \in \mathbb{C} : |z| \leq R\} \quad (\text{A.24})$$

denote the disk of radius $R \geq 0$ centered at the origin in \mathbb{C} . Thus $\mathbb{D}(1)$ is the unit circle; and the stationarity condition is that

$$\Phi_{p,m}(z) \neq 0 \text{ for every } z \in \mathbb{D}(1). \quad (\text{A.25})$$

The invertibility condition is that

$$\Theta_{q,m}(z) \neq 0 \text{ for every } z \in \mathbb{D}(1). \quad (\text{A.26})$$

In addition to requiring that $\Phi_{p,m}(z)$ and $\Theta_{q,m}(z)$ have no common roots and no multiple roots, the unicity condition requires that

$$\text{If } \left\{ \begin{array}{l} \Phi_{p,m}(z_j) = 0 \text{ for } j = 1, 2 \\ \text{or} \\ \Theta_{q,m}(z_j) = 0 \text{ for } j = 1, 2 \end{array} \right\} \text{ and } z_2 \neq \bar{z}_1, \text{ then } |z_1| \neq |z_2|. \quad (\text{A.27})$$

REMARK 3. Because both $\Phi_{p,m}(z)$ and $\Theta_{q,m}(z)$ have real coefficients, their roots are either real numbers or conjugate pairs of complex numbers; see p. 104 of Birkhoff and MacLane [10]. In the unicity condition, the requirement that each polynomial has no multiple roots implies that for each polynomial, its real roots must all have distinct values. The requirement (A.27) is similar, but it stipulates that nonconjugate roots must have distinct absolute values. The unicity assumption does not appear to be restrictive in practice, but it is necessary for the analysis that follows. See also p. 236 of Anderson [5] and p. 58 of Box, Jenkins, and Reinsel [12].

If $p > 0$, then for $m \geq m_0$ we let $\{r_i(m) \in \mathbb{C} : i = 1, \dots, p\}$ denote the roots of $\Phi_{p,m}(z)$ so that we can write

$$\Phi_{p,m}(z) = \prod_{i=1}^p [1 - \delta_i(m)z] \text{ for all } z \in \mathbb{C}, \quad (\text{A.28})$$

where

$$\delta_i(m) = [r_i(m)]^{-1} \in \mathbb{C} \text{ and } |\delta_i(m)| < 1 \text{ for } i = 1, \dots, p \quad (\text{A.29})$$

by the stationarity condition. Similarly if $q > 0$, then for $m \geq m_0$ we let $\{u_j(m) \in \mathbb{C} : j = 1, \dots, q\}$ denote the roots of $\Theta_{q,m}(z)$ so that we can write

$$\Theta_{q,m}(z) = \prod_{j=1}^q [1 - \omega_j(m)z] \text{ for all } z \in \mathbb{C}, \quad (\text{A.30})$$

where

$$\omega_j(m) = [u_j(m)]^{-1} \in \mathbb{C} \text{ and } |\omega_j(m)| < 1 \text{ for } j = 1, \dots, q \quad (\text{A.31})$$

by the invertibility condition.

In view of Theorem 1, we make the following continuity assumption about the autoregressive and moving-average coefficients in (A.17):

$$\left. \begin{aligned} \lim_{m \rightarrow \infty} \varphi_i(m) &= 0 \text{ for } i = 1, \dots, p \text{ (provided } p > 0), \\ \lim_{m \rightarrow \infty} \theta_j(m) &= 0 \text{ for } j = 1, \dots, q \text{ (provided } q > 0). \end{aligned} \right\} \quad (\text{A.32})$$

A.2 Proof of (A.17) and (A.32) When the Original (Unbatched) Process Is ARMA

Although we cannot prove that (A.17) and (A.32) hold for every stationary stochastic process with a finite variance parameter, we can establish (A.17) and (A.32) in a large class of stationary

stochastic processes that at least make (A.17) and (A.32) plausible assumptions about any batch-means process $\{Y_j(m)\}$ obtained by aggregating (batching) a stationary process, provided the batch size m is sufficiently large.

THEOREM 2. *If the original (unbatched) process $\{X_i\}$ is a stationary and invertible ARMA(p^*, q^*) process,*

$$X_i = \mu_X + \sum_{\ell=1}^{p^*} \varphi_\ell^* (X_{i-\ell} - \mu_X) + \varepsilon_i - \sum_{\ell=1}^{q^*} \theta_\ell^* \varepsilon_{i-\ell}, \quad (\text{A.33})$$

where $0 \leq p^*, q^* < \infty$, then the batch-means process $\{Y_j(m) : j = 1, 2, \dots\}$ is a stationary and invertible ARMA(p, q) process (A.17) with

$$p = p^* \text{ and } q = \begin{cases} p^*, & \text{if } q^* < p^* + 1, \\ p^* + 1, & \text{if } q^* \geq p^* + 1, \end{cases} \text{ for } m \geq m_0 = |p^* - q^*| + 1. \quad (\text{A.34})$$

Moreover, the autoregressive and moving-average coefficients in the ARMA(p, q) model describing the batch-means process $\{Y_j(m)\}$ satisfy the continuity condition (A.32):

$$\left. \begin{aligned} \lim_{m \rightarrow \infty} \varphi_i(m) &= 0 \text{ for } i = 1, \dots, p \text{ (provided } p > 0), \\ \lim_{m \rightarrow \infty} \theta_j(m) &= 0 \text{ for } j = 1, \dots, q \text{ (provided } q > 0). \end{aligned} \right\} \quad (\text{A.35})$$

Proof. To avoid trivial cases, throughout the rest of this proof we assume that $p^* \geq 1$ and $q^* \geq 1$. It is straightforward to adapt the following argument to handle situations in which $p^* = 0$ or $q^* = 0$ (or both). The result (A.34) was first established in [13, p. 145]; see also Result 1 of [31] and Result 2 of [50]. Corresponding to the ARMA(p^*, q^*) representation (A.33) of the original (unbatched) process $\{X_i\}$, we let $\{\delta_\nu : \nu = 1, \dots, p^*\}$ denote the inverse roots of the associated autoregressive polynomial

$$\Phi_{p^*}(z) \equiv \begin{cases} 1, & \text{if } p^* = 0, \\ 1 - \sum_{\ell=1}^{p^*} \varphi_\ell^* z^\ell, & \text{if } p^* \geq 1; \end{cases} \quad (\text{A.36})$$

and we let $\{\omega_\nu : \nu = 1, \dots, q^*\}$ denote the inverse roots of the associated moving-average polynomial

$$\Theta_{q^*}(z) \equiv \begin{cases} 1, & \text{if } q^* = 0, \\ 1 - \sum_{\ell=1}^{q^*} \theta_\ell^* z^\ell, & \text{if } q^* \geq 1. \end{cases} \quad (\text{A.37})$$

Because (A.33) is stationary and invertible, we have

$$\left. \begin{aligned} |\delta_\nu| &< 1 \text{ for } \nu = 1, \dots, p^* \text{ (provided } p^* > 0) \\ |\omega_\nu| &< 1 \text{ for } \nu = 1, \dots, q^* \text{ (provided } q^* > 0) \end{aligned} \right\}. \quad (\text{A.38})$$

Notice that the development in Equations (A.36), (A.37), and (A.38) for the original (unbatched) process (A.33) is a special case for batch size $m = 1$ of the development in Equations (A.28)–(A.31) for the batch-means process (A.17).

If the original (unbatched) process has the ARMA(p^* , q^*) representation (A.33), then from the first (unnumbered) equation on p. 467 of [50] we see that in the ARMA(p , q) representation of the batch-means process $\{Y_j(m)\}$ with batch size

$$m \geq m_0 = |p^* - q^*| + 1 \quad (\text{A.39})$$

and autoregressive and moving-average orders p and q given by (A.34), the autoregressive parameters are given by

$$\varphi_1(m) = \sum_{v=1}^p \delta_v^m, \quad (\text{A.40})$$

$$\varphi_2(m) = - \sum_{v=1}^{p-1} \sum_{j=v+1}^p \delta_v^m \delta_j^m, \quad (\text{A.41})$$

$$\vdots$$

$$\varphi_p(m) = (-1)^{p+1} \prod_{v=1}^p \delta_v^m, \quad (\text{A.42})$$

where in general $\varphi_\ell(m)$ equals $(-1)^{\ell+1}$ times the sum of products of the m th powers of individual elements from $\{\delta_v : v = 1, \dots, p\}$ taken ℓ at a time for $\ell = 1, \dots, p$. The desired continuity condition (A.35) for the autoregressive coefficients $\{\varphi_v(m) : v = 1, \dots, p\}$ then follows immediately from (A.38) and (A.40)–(A.42).

A less straightforward approach is required to prove that the moving-average coefficients $\{\theta_v(m) : v = 1, \dots, q\}$ also satisfy the continuity condition (A.35). Computing batch means from the moving-average component of (A.33), we obtain the auxiliary process

$$U_j(m) = \frac{1}{m} \sum_{i=(j-1)m+1}^{jm} \left(\varepsilon_i - \sum_{\ell=1}^{q^*} \theta_\ell^* \varepsilon_{i-\ell} \right). \quad (\text{A.43})$$

From the analysis concerning the first (unnumbered) equation on p. 467 of [50], we see that (A.43) defines an invertible MA(q) process whose autocorrelation function,

$$\varrho_\ell(m) = \text{Corr}[U_j(m), U_{j+\ell}(m)] \text{ for } \ell = 0, \pm 1, \pm 2, \dots, \quad (\text{A.44})$$

satisfies

$$\left. \begin{aligned} \varrho_\ell(m) &\neq 0, & \text{if } \ell = \pm q, \\ \varrho_\ell(m) &= 0, & \text{if } \ell < -q \text{ or } \ell > q. \end{aligned} \right\} \quad (\text{A.45})$$

Our approach is based on the analysis given in §5.7.1 of [5], which we will adapt to show that given the marginal variance $\text{Var}[U(m)]$ and the associated correlations $\varrho_1(m), \dots, \varrho_q(m)$, the invertible $\text{MA}(q)$ representation of the auxiliary process $\{U_j(m)\}$ defined by (A.43) has moving-average coefficients $\{\theta_\ell(m) : \ell = 1, \dots, q\}$ that converge to zero as $m \rightarrow \infty$. We consider the roots of the degree- $(2q)$ polynomial

$$\mathcal{C}_m(z) = \sum_{\ell=0}^{2q} \varrho_{\ell-q}(m) z^\ell. \quad (\text{A.46})$$

As shown in §5.7.1 of [5], the roots of $\mathcal{C}_m(z) = 0$ can be divided into two sets $\{\varpi_\ell(m) : \ell = 1, \dots, q\}$ and $\{\varpi_{q+\ell}(m) : \ell = 1, \dots, q\}$ such that

$$\text{For } \ell = 1, \dots, q, \text{ we have } |\varpi_\ell(m)| \leq 1, \text{ and } \left\{ \begin{array}{l} \text{if } |\varpi_\ell(m)| < 1, \text{ then } \varpi_{q+\ell}(m) = 1/\varpi_\ell(m) \\ \text{if } |\varpi_\ell(m)| = 1, \text{ then } \varpi_{q+\ell}(m) = \varpi_\ell(m) \end{array} \right\}. \quad (\text{A.47})$$

Moreover, from the analysis in §5.7.1 of [5], we see that moving-average polynomial corresponding to the $\text{MA}(q)$ representation of (A.43) has the form

$$\Theta_{q,m}(z) \equiv 1 - \sum_{\ell=1}^q \theta_\ell(m) z^\ell = \prod_{\ell=1}^q [1 - \varpi_\ell(m) z] \quad (\text{A.48})$$

because

$$\mathcal{C}_m(z) = \begin{cases} \varrho_{-q}(m), & \text{if } z = 0, \\ z^q \Theta_{q,m}(z) \Theta_{q,m}(1/z), & \text{if } z \neq 0; \end{cases} \quad (\text{A.49})$$

and from (A.48), we see immediately that

$$\theta_1(m) = \sum_{v=1}^q \varpi_v(m), \quad (\text{A.50})$$

$$\theta_2(m) = - \sum_{v=1}^{q-1} \sum_{j=v+1}^q \varpi_v(m) \varpi_j(m), \quad (\text{A.51})$$

\vdots

$$\theta_q(m) = (-1)^{q+1} \prod_{v=1}^q \varpi_v(m), \quad (\text{A.52})$$

where in general $\theta_\ell(m)$ equals $(-1)^{\ell+1}$ times the sum of products of individual elements from $\{\theta_\nu(m) : \nu = 1, \dots, q\}$ taken ℓ at a time for $\ell = 1, \dots, q$. The desired continuity condition (A.35) for the moving-average coefficients $\{\theta_\nu(m) : \nu = 1, \dots, q\}$ will follow immediately if we can show that

$$\lim_{m \rightarrow \infty} |\varpi_\ell(m)| = 0 \text{ for } \ell = 1, \dots, q. \quad (\text{A.53})$$

To establish (A.53), we will prove that the quantity

$$b(m) \equiv \min\{|\varpi_{q+\ell}(m)| : \ell = 1, \dots, q\} \quad (\text{A.54})$$

satisfies

$$\lim_{m \rightarrow \infty} b(m) = +\infty. \quad (\text{A.55})$$

Suppose on the contrary that

$$\liminf_{m \rightarrow \infty} b(m) = b^* < \infty. \quad (\text{A.56})$$

Choose $\eta > 0$ arbitrarily. From (A.56) it follows immediately that we can find at least one integer $\nu \in \{1, \dots, q\}$ together with a strictly increasing subsequence $\{m(\ell) : \ell = 1, 2, \dots\}$ of batch sizes such that

$$\lim_{\ell \rightarrow \infty} m(\ell) = +\infty \text{ and } 1 \leq |\varpi_{q+\nu}[m(\ell)]| \leq b^* + \eta = b^{**} \text{ for } \ell = 1, 2, \dots. \quad (\text{A.57})$$

(Throughout the rest of the proof of Theorem 2, the integer ν and the associated subsequence $\{m(\ell) : \ell = 1, 2, \dots\}$ satisfying (A.57) will remain fixed.) From (A.57) we have

$$\{\varpi_{q+\nu}[m(\ell)] : \ell = 1, 2, \dots\} \subset \mathbb{A}(b^{**}), \quad (\text{A.58})$$

where

$$\mathbb{A}(b^{**}) \equiv \{z \in \mathbb{C} : 1 \leq |z| \leq b^{**}\} \quad (\text{A.59})$$

is the annulus centered at the origin with inner radius 1 and outer radius b^{**} . Since $\mathbb{A}(b^{**})$ is closed and bounded (and hence compact), the Bolzano-Weierstrass Theorem [34] implies that $\{\varpi_{q+\nu}[m(\ell)] : \ell = 1, 2, \dots\}$ has a limit point $\zeta \in \mathbb{A}(b^{**})$. This means that there exists a subsubsequence $\{\varpi_{q+\nu}[m(\ell_w)] : w = 1, 2, \dots\}$ of the subsequence $\{\varpi_{q+\nu}[m(\ell)] : \ell = 1, 2, \dots\}$ such that

$$\lim_{w \rightarrow \infty} \varpi_{q+\nu}[m(\ell_w)] = \zeta \in \mathbb{A}(b^{**}). \quad (\text{A.60})$$

We have

$$\mathcal{C}_{m(\ell_w)}\{\varpi_{q+\nu}[m(\ell_w)]\} = 0 \text{ for } w = 1, 2, \dots, \quad (\text{A.61})$$

since for each $m \geq m_0$, recall that $\varpi_{q+\nu}(m)$ is a root of $\mathcal{C}_m(z) = 0$. On the other hand, we must have

$$\begin{aligned} \lim_{w \rightarrow \infty} \mathcal{C}_{m(\ell_w)} \{ \varpi_{q+\nu}[m(\ell_w)] \} &= \lim_{w \rightarrow \infty} \sum_{\ell=0}^{2q} \varrho_{\ell-q}[m(\ell_w)] \{ \varpi_{q+\nu}[m(\ell_w)] \}^\ell \\ &= \sum_{\ell=0}^{2q} \left\{ \lim_{w \rightarrow \infty} \varrho_{\ell-q}[m(\ell_w)] \right\} \times \left(\lim_{w \rightarrow \infty} \{ \varpi_{q+\nu}[m(\ell_w)] \}^\ell \right) \\ &= \zeta^q \end{aligned} \quad (\text{A.62})$$

because $\varrho_0(m) \equiv 1$ and $\lim_{m \rightarrow \infty} \varrho_\ell(m) = 0$ for $\ell = \pm 1, \pm 2, \dots, \pm q$ by Equation (A.2) of Theorem 1. Since we must have $|\zeta| \geq 1$ by the definition (A.59), Equation (A.62) contradicts (A.61); and thus the assumption (A.56) must be false for $b^* < \infty$. It follows that

$$\infty = \liminf_{m \rightarrow \infty} b(m) \leq \limsup_{m \rightarrow \infty} b(m) \quad (\text{A.63})$$

so that we have

$$\lim_{m \rightarrow \infty} b(m) = +\infty; \quad (\text{A.64})$$

and thus the desired conclusion of Theorem 2 follows. \blacksquare

A.3 Consequences of Assumption (A.32) When the Batch-Means Process Is ARMA

LEMMA 1. *If $\{Y_j(m)\}$ is a stationary and invertible ARMA(p, q) process (A.17) for $m \geq m_0$ and the continuity condition (A.32) holds, then*

$$\left. \begin{aligned} \lim_{m \rightarrow \infty} |\delta_i(m)| &= 0 \text{ for } i = 1, \dots, p \text{ (provided } p > 0), \\ \lim_{m \rightarrow \infty} |\omega_j(m)| &= 0 \text{ for } j = 1, \dots, q \text{ (provided } q > 0). \end{aligned} \right\} \quad (\text{A.65})$$

Proof. The proof of Lemma 1 is similar in spirit to that of Theorem 2, but the details are somewhat different. To avoid trivial cases, throughout the rest of this proof we assume that $p \geq 1$ and $q \geq 1$. It is straightforward to adapt the following argument to handle situations in which $p = 0$ or $q = 0$ (or both). We will prove that the quantities

$$\left. \begin{aligned} a_1(m) &\equiv \min\{|r_i(m)| : i = 1, \dots, p\} \\ a_2(m) &\equiv \min\{|\omega_j(m)| : j = 1, \dots, q\} \end{aligned} \right\} \quad (\text{A.66})$$

satisfy

$$\lim_{m \rightarrow \infty} a_j(m) = +\infty \text{ for } j = 1, 2; \quad (\text{A.67})$$

and from (A.29) and (A.31), we have

$$\left. \begin{array}{l} |\delta_i(m)| \leq [a_1(m)]^{-1} \quad \text{for } i = 1, \dots, p \\ |\omega_j(m)| \leq [a_2(m)]^{-1} \quad \text{for } j = 1, \dots, q \end{array} \right\} \text{ for all } m \geq m_0; \quad (\text{A.68})$$

thus the desired conclusion (A.65) will follow immediately from (A.67).

Suppose that on the contrary

$$\liminf_{m \rightarrow \infty} a_1(m) = a_1^* < \infty. \quad (\text{A.69})$$

Choose $\eta > 0$ arbitrarily. From (A.69) it follows that we can find at least one index $i \in \{1, \dots, p\}$ together with a strictly increasing subsequence $\{m(\ell) : \ell = 1, 2, \dots\}$ such that

$$\lim_{\ell \rightarrow \infty} m(\ell) = +\infty \text{ and } |r_i[m(\ell)]| \leq a_1^* + \eta = a_1^{**} \text{ for } \ell = 1, 2, \dots. \quad (\text{A.70})$$

(Throughout the rest of the proof of Lemma 1, the index i and the associated subsequence $\{m(\ell) : \ell = 1, 2, \dots\}$ are fixed.) From (A.70) we have

$$\{r_i[m(\ell)] : \ell = 1, 2, \dots\} \subset \mathbb{D}(a_1^{**}); \quad (\text{A.71})$$

and since $\mathbb{D}(a_1^{**})$ is closed and bounded (and hence compact), the Bolzano-Weierstrass Theorem [34] implies that $\{r_i[m(\ell)] : \ell = 1, 2, \dots\}$ has a limit point $\zeta \in \mathbb{D}(a_1^{**})$. This means that there exists a subsubsequence $\{r_i[m(\ell_w)] : w = 1, 2, \dots\}$ of the subsequence $\{r_i[m(\ell)] : \ell = 1, 2, \dots\}$ such that

$$\lim_{w \rightarrow \infty} r_i[m(\ell_w)] = \zeta \in \mathbb{D}(a_1^{**}). \quad (\text{A.72})$$

We have

$$\Phi_{p,m(\ell_w)}\{r_i[m(\ell_w)]\} = 0 \text{ for } w = 1, 2, \dots, \quad (\text{A.73})$$

since by definition $r_i(m)$ is a root of $\Phi_{p,m}(z) = 0$ for each $m \geq m_0$. On the other hand, we must have

$$\begin{aligned} \lim_{w \rightarrow \infty} \Phi_{p,m(\ell_w)}\{r_i[m(\ell_w)]\} &= 1 - \sum_{v=1}^p \left\{ \lim_{w \rightarrow \infty} \varphi_v[m(\ell_w)] \right\} \left\{ \lim_{w \rightarrow \infty} r_i[m(\ell_w)] \right\}^v \\ &= 1 - \sum_{v=1}^p \{0\} \{\zeta\}^v \\ &= 1. \end{aligned} \quad (\text{A.74})$$

Clearly (A.74) contradicts (A.73); and thus the assumption (A.69) must be false for $a_1^* < \infty$. It follows that

$$\infty = \liminf_{m \rightarrow \infty} a_1(m) \leq \limsup_{m \rightarrow \infty} a_1(m) \quad (\text{A.75})$$

so that we have

$$\lim_{m \rightarrow \infty} a_1(m) = +\infty. \quad (\text{A.76})$$

A similar argument yields the remaining part of (A.67), and the desired conclusion of Lemma 1 follows.

■

A.4 Proof of the Main Result When the Batch-Means Process Is ARMA

To complete the analysis of the batch-means process $\{Y_\ell(m) : \ell = 1, 2, \dots\}$ for sufficiently large values of the batch size m , we impose the following regularity condition on the way in which the inverse autoregressive roots $\{\delta_i(m) : i = 1, \dots, p\}$ and the inverse moving-average roots $\{\omega_j(m) : j = 1, \dots, q\}$ tend to zero in absolute value as $m \rightarrow \infty$:

$$\limsup_{m \rightarrow \infty} \frac{|\delta_i(m)|}{|\sum_{\ell=1}^p \delta_\ell(m)|} < \infty \text{ for } i = 1, \dots, p \text{ (provided } p > 0) \quad (\text{A.77})$$

and

$$\limsup_{m \rightarrow \infty} \frac{|\omega_j(m)|}{|\sum_{\ell=1}^q \omega_\ell(m)|} < \infty \text{ for } j = 1, \dots, q \text{ (provided } q > 0). \quad (\text{A.78})$$

REMARK 4. Equation (A.65) in Lemma 1 ensures that both the numerators and denominators in (A.77) and (A.78) tend to zero as $m \rightarrow \infty$. The condition (A.77) merely requires that the absolute value of the sum of inverse autoregressive roots should tend to zero as $m \rightarrow \infty$ at a rate which does not differ too much from the rates at which the absolute values of each of the individual inverse autoregressive roots tend to zero. Condition (A.78) imposes a similar requirement on the rates at which the absolute values of the inverse moving-average roots tend to zero as $m \rightarrow \infty$. In particular, autoregressive (respectively, moving-average) roots that decline in absolute value according to an inverse-power law in the batch size m or exponentially in the batch size as $m \rightarrow \infty$ and that satisfy the unicity condition can be easily shown to satisfy condition (A.77) (respectively, (A.78)).

THEOREM 3. *If for all $m \geq m_0$ the batch-means process $\{Y_j(m) : j = 1, 2, \dots\}$ is an ARMA(p, q) process (A.17) that satisfies the stationarity, invertibility, and unicity conditions as well as the continuity condition (A.32) and the regularity conditions (A.77) and (A.78), then as $m \rightarrow \infty$ we have*

$$\varphi_\nu(m) = o[\varphi_1(m)] \text{ for } \nu = 2, \dots, p \text{ (provided } p > 1), \quad (\text{A.79})$$

$$\theta_\nu(m) = o[\theta_1(m)] \text{ for } \nu = 2, \dots, q \text{ (provided } q > 1), \quad (\text{A.80})$$

and

$$\text{Var}[\bar{Y}(m, k')] \sim \frac{\text{Var}[Y(m)]}{k'} \left[\frac{1 - \varphi_1^2(m)}{1 - 2\varphi_1(m)\theta_1(m) + \theta_1^2(m)} \right] \left[\frac{1 - \theta_1(m)}{1 - \varphi_1(m)} \right]^2 \quad (\text{A.81})$$

$$\sim \frac{\text{Var}[Y(m)]}{k'} \left[\frac{1 + \varphi_1(m)}{1 - \varphi_1(m)} \right]. \quad (\text{A.82})$$

Proof. To avoid trivial cases, throughout the rest of this proof we assume that $p \geq 2$ and $q \geq 2$. It is straightforward to adapt the following argument to handle situations in which $p \leq 1$ or $q \leq 1$ (or both). Corresponding to the ARMA(p, q) model of the batched means $\{Y_j(m)\}$ for $m \geq m_0$, we see that the autoregressive polynomial (A.28) expressed in terms of its roots (A.29) is given by

$$\Phi_{p,m}(z) = \prod_{v=1}^p [1 - \delta_v(m)z] \text{ for } z \in \mathbb{C}. \quad (\text{A.83})$$

Expanding (A.83) and equating the result to the formal definition

$$\Phi_{p,m}(z) \equiv 1 - \sum_{v=1}^p \varphi_v(m)z^v, \quad (\text{A.84})$$

we see that

$$\varphi_1(m) = \sum_{v=1}^p \delta_v(m), \quad (\text{A.85})$$

$$\varphi_2(m) = - \sum_{v=1}^{p-1} \sum_{j=v+1}^p \delta_v(m)\delta_j(m), \quad (\text{A.86})$$

\vdots

$$\varphi_p(m) = (-1)^{p+1} \prod_{v=1}^p \delta_v(m), \quad (\text{A.87})$$

where in general $\varphi_\ell(m)$ equals $(-1)^{\ell+1}$ times the sum of products of roots taken ℓ at a time for $\ell = 1, \dots, p$. Similarly, the moving-average polynomial (A.30) expressed in terms of its roots (A.31) is given by

$$\Theta_{q,m}(z) = \prod_{v=1}^q [1 - \omega_v(m)z] \text{ for } z \in \mathbb{C}; \quad (\text{A.88})$$

Expanding (A.88) and equating the result to the formal definition

$$\Theta_{q,m}(z) \equiv 1 - \sum_{v=1}^q \theta_v(m)z^v, \quad (\text{A.89})$$

we obtain

$$\theta_1(m) = \sum_{\nu=1}^q \omega_\nu(m), \quad (\text{A.90})$$

$$\theta_2(m) = - \sum_{\nu=1}^{q-1} \sum_{j=\nu+1}^q \omega_\nu(m) \omega_j(m), \quad (\text{A.91})$$

⋮

$$\theta_q(m) = (-1)^{q+1} \prod_{\nu=1}^q \omega_\nu(m), \quad (\text{A.92})$$

where in general $\theta_\ell(m)$ equals $(-1)^{\ell+1}$ times the sum of products of individual elements from $\{\theta_\nu(m) : \nu = 1, \dots, q\}$ taken ℓ at a time for $\ell = 1, \dots, q$.

We will prove (A.80) in detail; the proof of (A.79) is similar. Let

$$\nu_m^* \equiv \arg \max \{|\omega_\nu(m)| : \nu = 1, \dots, q\}. \quad (\text{A.93})$$

We have

$$\begin{aligned} |\theta_2(m)/\theta_1(m)| &= \left| \omega_{\nu_m^*}(m) \left[\frac{\omega_{\nu_m^*}(m)}{\sum_{\nu=1}^q \omega_\nu(m)} \right] \left\{ \sum_{\nu=1}^q \left[\frac{\omega_\nu(m)}{\omega_{\nu_m^*}(m)} \right]^2 \right. \right. \\ &\quad \left. \left. + \sum_{\nu=1}^{q-1} \sum_{j=\nu+1}^q \left[\frac{\omega_\nu(m)}{\omega_{\nu_m^*}(m)} \right] \cdot \left[\frac{\omega_j(m)}{\omega_{\nu_m^*}(m)} \right] \right\} \right| \\ &\leq |\omega_{\nu_m^*}(m)| \cdot \left\{ \frac{|\omega_{\nu_m^*}(m)|}{|\sum_{\nu=1}^q \omega_\nu(m)|} \right\} \\ &\quad \times \left\{ \sum_{\nu=1}^q \left| \frac{\omega_\nu(m)}{\omega_{\nu_m^*}(m)} \right|^2 + \sum_{\nu=1}^{q-1} \sum_{j=\nu+1}^q \left| \frac{\omega_\nu(m)}{\omega_{\nu_m^*}(m)} \right| \cdot \left| \frac{\omega_j(m)}{\omega_{\nu_m^*}(m)} \right| \right\}. \end{aligned} \quad (\text{A.94})$$

Applying Lemma 1 to the $\{\omega_\nu(m)\}$, we see that

$$\lim_{m \rightarrow \infty} |\omega_{\nu_m^*}(m)| \leq \lim_{m \rightarrow \infty} \sum_{\nu=1}^q |\omega_\nu(m)| = \sum_{\nu=1}^q \left[\lim_{m \rightarrow \infty} |\omega_\nu(m)| \right] = 0. \quad (\text{A.95})$$

Applying the regularity property (A.78) to the $\{\omega_\nu(m)\}$, we see that

$$\begin{aligned} \limsup_{m \rightarrow \infty} \frac{|\omega_{\nu_m^*}(m)|}{|\sum_{\nu=1}^q \omega_\nu(m)|} &\leq \limsup_{m \rightarrow \infty} \sum_{i=1}^q \left[\frac{|\omega_i(m)|}{|\sum_{\nu=1}^q \omega_\nu(m)|} \right] \\ &\leq \sum_{i=1}^q \left[\limsup_{m \rightarrow \infty} \frac{|\omega_i(m)|}{|\sum_{\nu=1}^q \omega_\nu(m)|} \right] < \infty. \end{aligned} \quad (\text{A.96})$$

The third term in (A.94) is bounded above by

$$\begin{aligned} \sum_{v=1}^q \left| \frac{\omega_v(m)}{\omega_{v_m^*}(m)} \right|^2 + \sum_{v=1}^{q-1} \sum_{j=v+1}^q \left| \frac{\omega_v(m)}{\omega_{v_m^*}(m)} \right| \cdot \left| \frac{\omega_j(m)}{\omega_{v_m^*}(m)} \right| \\ \leq \sum_{v=1}^q 1 + \sum_{v=1}^{q-1} \sum_{j=v+1}^q 1 = q + \frac{(q-1)q}{2} \text{ for every } m \geq m_0. \end{aligned} \quad (\text{A.97})$$

Combining (A.94), (A.95), (A.96), and (A.97), we have

$$\lim_{m \rightarrow \infty} |\theta_2(m)/\theta_1(m)| = 0. \quad (\text{A.98})$$

A similar analysis shows that

$$\lim_{m \rightarrow \infty} |\theta_v(m)/\theta_1(m)| = 0 \text{ for } v \neq 1. \quad (\text{A.99})$$

Thus as $m \rightarrow \infty$, we see that $\{Y_j(m)\}$ is approximately an ARMA(1, 1) process with variance parameter

$$\gamma_{Y(m)} = \text{Var}[Y(m)] \left[\frac{1 - \varphi_1^2(m)}{1 - 2\varphi_1(m)\theta_1(m) + \theta_1^2(m)} \right] \left[\frac{1 - \theta_1(m)}{1 - \varphi_1(m)} \right]^2; \quad (\text{A.100})$$

see, for example, §3.4.3 of Box, Jenkins, and Reinsel [12]. We have

$$\begin{aligned} 1 - 2\varphi_1(m)\theta_1(m) + \theta_1^2(m) &= [1 - \theta_1(m)]^2 + 2\theta_1(m) - 2\varphi_1(m)\theta_1(m) \\ &\sim [1 - \theta_1(m)]^2 \end{aligned} \quad (\text{A.101})$$

since the terms $\theta_1(m)$ and $-2\varphi_1(m)\theta_1(m)$ are negligible in comparison with $[1 - \theta_1(m)]^2$. Thus we have

$$\begin{aligned} \gamma_{Y(m)} &\sim \text{Var}[Y(m)] \left[\frac{1 - \varphi_1^2(m)}{[1 - \theta_1(m)]^2} \right] \left[\frac{1 - \theta_1(m)}{1 - \varphi_1(m)} \right]^2 \\ &= \text{Var}[Y(m)] \left[\frac{1 + \varphi_1(m)}{1 - \varphi_1(m)} \right] \text{ as } m \rightarrow \infty. \end{aligned} \quad (\text{A.102})$$

From the usual relation

$$\text{Var}[Y(m, k')] \sim \frac{\gamma_{Y(m)}}{k'} \text{ as } m \rightarrow \infty \quad (\text{A.103})$$

and (A.102), we finally obtain the desired result (A.82). ■

Remark 5. This development suggests a minor revision in the computation of the correlation adjustment A that could be applied to the estimation $S_{m,k'}^2/k'$ of the variance of the grand mean $\bar{Y}(m, k')$ of the batch means, where we are now assuming that we have $m \geq m_0$ for some “minimally” acceptable

batch size m_0 . From the truncated, nonspaced batch means, we could calculate the lag- ℓ correlations for $\ell = 1, 2$ as follows:

$$\hat{\rho}_{Y(m)}(\ell) = \frac{1}{k' - \ell} \sum_{j=1}^{k' - \ell} \frac{[Y_j(m) - \bar{Y}(m, k')][Y_{j+\ell}(m) - \bar{Y}(m, k')]}{S_{m, k'}^2}. \quad (\text{A.104})$$

Then we could take

$$\hat{\varphi}_1(m) = \frac{\hat{\rho}_{Y(m)}(2)}{\hat{\rho}_{Y(m)}(1)}, \quad (\text{A.105})$$

the usual moment-matching estimator of the autoregressive parameter if the $\{Y_j(m)\}$ are an ARMA(1, 1) process. In practice, however, we have found that at the batch sizes delivered by Skart, the values of $\hat{\rho}_{Y(m)}(1)$ and $\hat{\rho}_{Y(m)}(2)$ are quite small and sometimes yield the anomalous result

$$\hat{\varphi}_1(m) = \frac{\hat{\rho}_{Y(m)}(2)}{\hat{\rho}_{Y(m)}(1)} > 1, \quad (\text{A.106})$$

which is outside the region of stationarity and invertibility for an ARMA(1, 1) process; see, for example, Figure 3.10 of [12]. This suggested using the reduced AR(1) model instead, which amounts to taking

$$\hat{\theta}_1(m) = 0 \text{ and } \hat{\varphi}_1(m) = \hat{\rho}_{Y(m)}(1) \quad (\text{A.107})$$

in Equations (A.100) and (A.102).

It should be noted that our use of the AR(1) approximation (A.107) to the final batch means delivered by Skart is consistent with the use of an AR(1) model for the final batch means delivered by ASAP3 and SBatch. In fact on p. 51 of Steiger *et al.* [54], we find the following statement:

On the i th iteration of step [3] of ASAP3, the use of the AR(1) time series model (4) to represent the behavior of the truncated batch means is based on all our previous computational experience with ASAP and ASAP2. Although the original ASAP algorithm was equipped to fit more general autoregressive–moving average time series models to the batch means, for *every* application of ASAP in which the batch means had first passed the Shapiro-Wilk test for ... normality, the simple AR(1) model fitted to the batch means exhibited no significant lack of fit as well as the best fit of all the time series models tested.

(The emphasis in this statement has been added.) As documented in Steiger and Wilson [52], the original ASAP procedure sought to identify and estimate the best-fitting ARMA time-series model for the final sequence of truncated batch means from the following list of candidate models:

AR(1); AR(2); MA(1); MA(2); and ARMA(1, 1),

where standard IMSL routines were used for all parameter estimation and goodness-of-fit testing.

The main consequence of Theorem 3 is that this list of candidate models for the truncated batch means delivered by Skart can be limited to AR(1), MA(1), and ARMA(1,1) models; and all our computational experience with Skart and its predecessors ASAP, ASAP2, ASAP3, and SBatch support our use of the correlation adjustment (1.11)–(1.12) based on an AR(1) model of the truncated batch means delivered by Skart.

Appendix B

Distribution of First Regular $M/M/1$ Queue Waiting Time X_1 given $N(0) = c$ Initial Nonregular Customers

In this appendix we first show that in an $M/M/1$ queue with arrival rate λ and service rate μ , the conditional moment generating function and c.d.f. of the waiting time X_1 for the first regular customer to arrive after time zero, given c initial nonregular customers in the system at time zero is,

$$\begin{aligned}
 M_{X_1}(t) &= \sum_{j=0}^{c-1} (1-\psi)\psi^j \cdot \frac{1}{(1-t/\mu)^{c-j}} + \psi^c \\
 &= \frac{1-\psi}{(1-t/\mu)^c} \cdot \sum_{j=0}^{c-1} [\psi(1-t/\mu)]^j + \psi^c \\
 &= \frac{1-\psi}{(1-t/\mu)^c} \cdot \frac{1-[\psi(1-t/\mu)]^c}{1-\psi(1-t/\mu)} + \psi^c,
 \end{aligned} \tag{B.1}$$

and

$$F_{X_1}(w) = \sum_{j=0}^c (1-\psi)\psi^j \cdot F_{\text{Er}}(w; c-j, \mu) + \psi^{c+1}F_0(w), \tag{B.2}$$

where $\psi = \mu/(\lambda + \mu)$, and $F_{\text{Er}}(\cdot)$ is the c.d.f. for an Erlang distribution.

Regular customers enter the $M/M/1$ queue with an exponential rate λ , so the probability that the first regular customer observes $c - j$ ($0 \leq j \leq c$) nonregular customers in the system at arrival

time is

$$\left(\frac{\lambda}{\mu + \lambda}\right)\left(\frac{\mu}{\mu + \lambda}\right)^j = (1 - \psi)\psi^j. \quad (\text{B.3})$$

We also know that the total service time S for $c - j$ nonregular customers in the $M/M/1$ queue, or in other words the waiting time for the first regular customer knowing that $c - j$ nonregular customers remained in the system, has an Erlang distribution with mean

$$E_{\text{Er}}(S; c - j, \mu) = \frac{c - j}{\mu}, \quad (\text{B.4})$$

and a moment generating function

$$M_S(t) = \frac{1}{(1 - t/\mu)^{c-j}} \text{ for } t < \mu. \quad (\text{B.5})$$

Thus, in view of (B.3) and (B.5), we can easily derive the conditional moment generating function (B.1) and c.d.f (B.2) of the waiting time X_1 .

In view of (B.2) and (B.4), the following equation for the mean of the waiting time X_1 can also be easily derived:

$$E[X_1] = \sum_{j=0}^c \psi^j (1 - \psi) \cdot \frac{c - j}{\mu}. \quad (\text{B.6})$$

We can simplify Equation (B.6) as following:

$$\begin{aligned} E[X_1] &= \frac{1 - \psi}{\mu} \sum_{j=0}^c (c - j)\psi^j \\ &= \frac{1 - \psi}{\mu} \left\{ c \cdot \frac{1 - \psi^{c+1}}{1 - \psi} - \psi \cdot \sum_{j=1}^c j \psi^{j-1} \right\} \\ &= \frac{1 - \psi}{\mu} \left\{ c \cdot \frac{1 - \psi^{c+1}}{1 - \psi} - \psi \cdot \frac{d}{d\psi} \sum_{j=0}^c \psi^j \right\} \\ &= \frac{1 - \psi}{\mu} \left\{ c \cdot \frac{1 - \psi^{c+1}}{1 - \psi} - \psi \cdot \frac{d}{d\psi} \left(\frac{1 - \psi^{c+1}}{1 - \psi} \right) \right\} \\ &= \frac{1 - \psi}{\mu} \left\{ c \cdot \frac{1 - \psi^{c+1}}{1 - \psi} - \psi \left[\frac{-(c + 1)\psi^c}{1 - \psi} + \frac{1 - \psi^{c+1}}{(1 - \psi)^2} \right] \right\} \\ &= \frac{1}{\mu} \left\{ c - c\psi^{c+1} + (c + 1)\psi^{c+1} - \frac{\psi - \psi^{c+2}}{1 - \psi} \right\} \end{aligned} \quad (\text{B.7})$$

$$\begin{aligned}
&= \frac{1}{\mu} \left\{ c + \psi^{c+1} - \frac{\psi - \psi^{c+2}}{1 - \psi} \right\} \\
&= \frac{1}{\mu(1 - \psi)} \{c(1 - \psi) - \psi + \psi^{c+1}\} \\
&= \frac{c}{\psi} - \frac{\psi}{\mu} \cdot \left(\frac{1 - \psi^c}{1 - \psi} \right). \tag{B.8}
\end{aligned}$$

Hence, using Equation (B.8), the conditional mean of the the waiting time in an $M/M/1$ queue with arrival rate $\lambda = 0.9$ and service rate $\mu = 1.0$ for the first regular customer to arrive after time zero, given $c = 113$ initial customers in the system at time zero is $E[X_1] = 111.889$.

The Maple code for evaluating the expression $(d^2/dt^2)M_{X_1}(0)$ in Equation (4.3) and then computing the conditional variance of the waiting time in an $M/M/1$ queue with arrival rate $\lambda = 0.9$ and service rate $\mu = 1.0$ for the first regular customer to arrive after time zero, given $c = 113$ initial customers in the system at time zero is given in the following:

```

lambda:=0.9;
mu:=1.0;
psi:=mu/(lambda+mu);
c:=113;
s:=t/mu;
R:=psi*(1-s);
m:=((1-psi)/((1-s)^c))*((1-R^c)/(1-R)) + psi^c;
dm := diff(normal(m),t);
d2m := normal(diff(normal(m),t,t));
Dm := unapply(dm,t);
D2m := unapply(d2m,t);
EX := Dm(0);
VarX := D2m(0)-(EX^2);
sqrt(VarX);

```

Appendix C

Skart and N-Skart User's Manual

C.1 Downloading Skart and N-Skart

Both Skart and N-Skart procedures were built in the Visual Studio.Net 2003 programming platform [22] and coded in the Visual Basic.Net programming language. The file SkartFiles.zip contains two folders for the Skart and N-Skart procedures and can be downloaded from the following hyperlink:

`<http://www.ise.ncsu.edu/jwilson/files/SkartFiles.zip>.`

All the subroutines and functions of the Skart (respectively, N-Skart) procedure are saved under the “Skart” (respectively, “N-Skart”) module in the Skart (respectively, N-Skart) project. The visual Basic.Net code for each of these subroutines and functions is given at the end of this appendix. In order to run either of Skart or N-Skart procedure, the Visual Studio.Net 2003 or a newer version of this software is required to be installed on the user's computer.

C.2 Running Skart in Visual Basic

When you invoke the Skart procedure, the code written in the *Main* subroutine of the Skart project, which is a call to the *SkarProcedure* subroutine, is initially executed. Thus, to select the desired simulation model and the CI specifications, the user has to set the arguments of the *SkarProcedure* subroutine before invoking the Skart procedure. The list of the arguments required by the *SkarProcedure* subroutine and a description of their acceptable values is given in Table C.1.

Table C.1: Input arguments required for the SkarProcedure subroutine

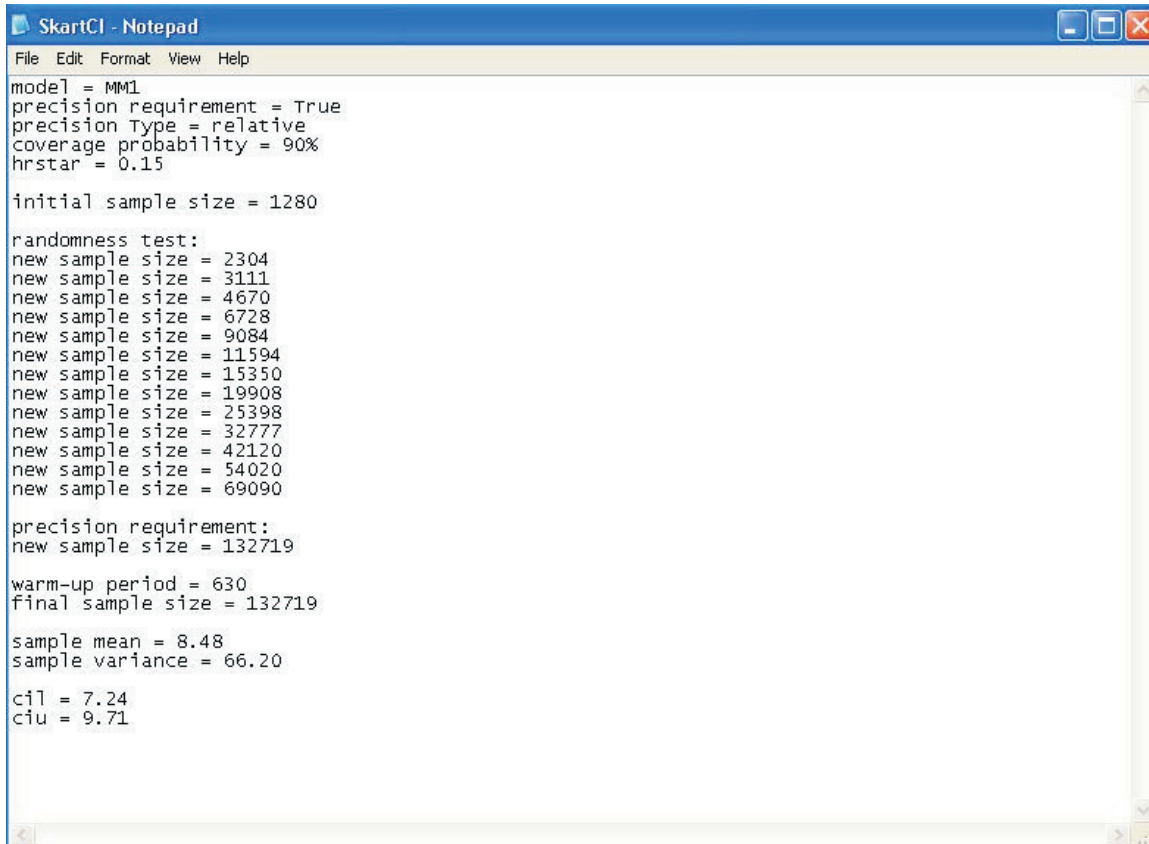
Argument	Description
model	The argument <i>model</i> sets the name of the simulation model to be used to generate the data.
precReq	The argument <i>precReq</i> is set to “True” if a relative or absolute precision requirement is desired for the final CI, otherwise <i>precReq</i> is set to “False”.
precisionType	The argument <i>precisionType</i> specifies the type of precision requirement and can either take the value “absolute” or “relative”. If <i>precReq</i> is set to “False”, then the value of the argument <i>precisionType</i> is irrelevant.
alpha	The argument <i>alpha</i> defines the desired confidence level for the final CI, $100(1-\alpha)\%$, where $0 < \alpha < 1$.
hrstar	The argument <i>hrstar</i> is used to set the desired value of the final CI half-length. <i>hrstar</i> would take the value h^* for the absolute precision requirement and r^* for the relative precision requirement as described in (3.16).

The output of the Skart procedure is saved in a Notepad file. The Notepad file is designed to demonstrate the following details to the user of the Skart procedure:

- The simulation model name, precision requirement, precision type, confidence level, and precision requirement.
- The initial sample size and the sequence of sample sizes required in the randomness test and precision requirement step.
- The final length of the warm-up period and the final sample size.
- The process sample mean and variance.
- The upper and lower confidence limits.

Figure C.1 shows a screen-shot of the output file of the Skart procedure when it is applied to an $M/M/1$ queue-waiting-time process with an empty-and-idle initial condition, an interarrival rate of $\lambda = 0.9$ customers per-time-unit, and a service rate of $\mu = 1.0$ customers per-time-unit, asking for a 90% CI with a relative precision requirement of $\pm 15\%$. In this system the steady-state expected

waiting time is $\mu_X = 9.0$ time units. Figure C.1 provides all above-mentioned details to the user for the considered $M/M/1$ queue-waiting-time process.



```

SkartCI - Notepad
File Edit Format View Help
model = MM1
precision requirement = True
precision Type = relative
coverage probability = 90%
hrstar = 0.15

initial sample size = 1280

randomness test:
new sample size = 2304
new sample size = 3111
new sample size = 4670
new sample size = 6728
new sample size = 9084
new sample size = 11594
new sample size = 15350
new sample size = 19908
new sample size = 25398
new sample size = 32777
new sample size = 42120
new sample size = 54020
new sample size = 69090

precision requirement:
new sample size = 132719

warm-up period = 630
final sample size = 132719

sample mean = 8.48
sample variance = 66.20

cil = 7.24
ciu = 9.71

```

Figure C.1: Screen-shot of the Notepad output window showing the result of applying Skart procedure to an $M/M/1$ queue-waiting-time process

C.3 Running N-Skart in Visual Basic

When you invoke the N-Skart procedure, the code written in the *Main* subroutine of the N-Skart project, which is a call to the *NSkarProcedure* subroutine, is initially executed. Thus, to select the simulation model, confidence level, and total available sample size, the user has to set the arguments of the *NSkarProcedure* subroutine before invoking the N-Skart procedure. The list of the arguments required by the *NSkarProcedure* subroutine and a description of their acceptable values is given in Table C.2.

Table C.2: Input arguments required for the NSkarProcedure subroutine

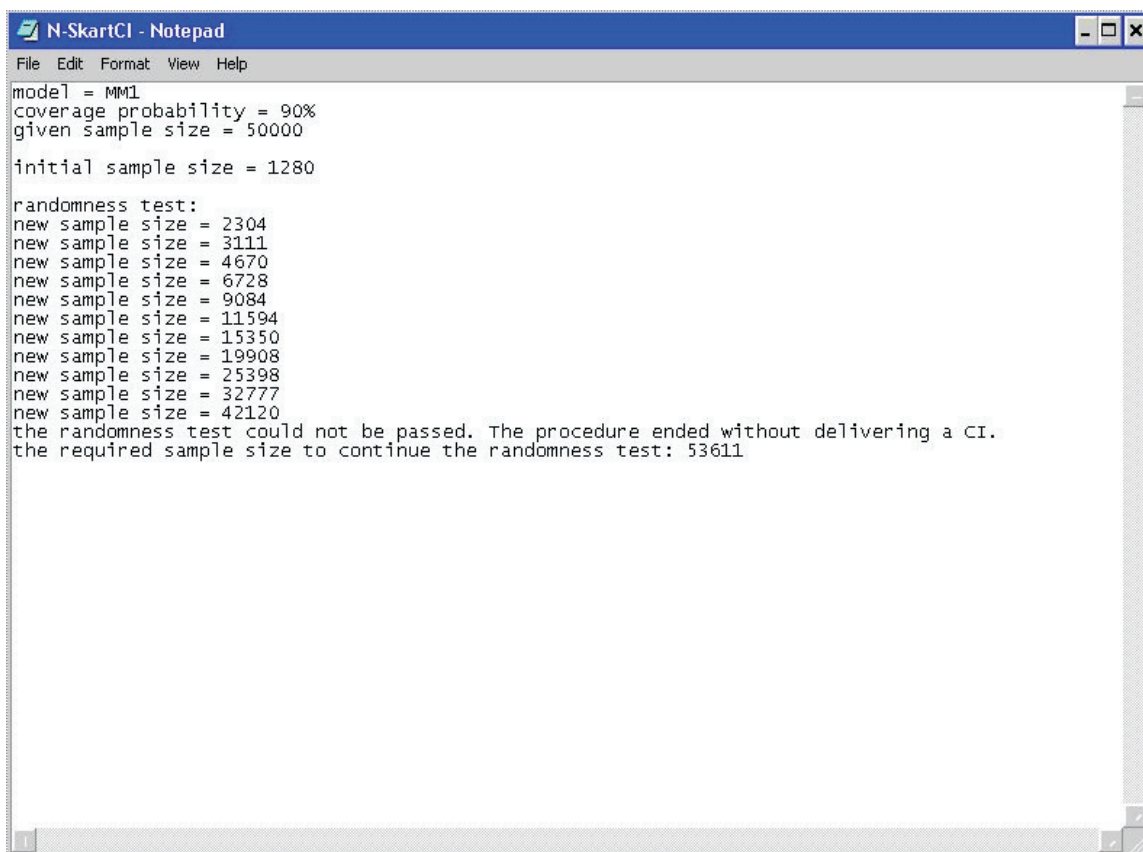
Argument	Description
model	The argument <i>model</i> sets the name of the simulation model to be used to generate the data.
alpha	The argument <i>alpha</i> defines the desired confidence level for the final CI, $100(1-\alpha)\%$, where $0 < \alpha < 1$.
sampleSize	The argument <i>sampleSize</i> specifies the total available sample size.

The output of the N-Skart procedure is saved in a Notepad file. The Notepad file is designed to demonstrate the following details to the user of the N-Skart procedure:

- The simulation model name, confidence level, and total available sample size.
- The initial sample size and the sequence of sample sizes required in the randomness test.
- If the randomness test could not be passed due to insufficient data, a warning message is issued which provides the user with the total sample size required to continue the randomness test.
- If the user chooses to quit the procedure, the following message is generated: “The randomness test could not be passed. The procedure ended without delivering a CI”.
- If the user chooses to continue with constructing a CI, a message is generated warning the user that the delivered CI might not provide the target confidence level.
- The final length of the warm-up period and the final sample size.
- The process sample mean and variance.
- The upper and lower confidence limits.
- The relative precision level of the delivered CI.
- The following message is generated if the user neglected the warning for insufficient data: “A CI with abnormally high relative precision should alert you regarding potential problems with the delivered CI and a possible need for bigger sample size”.

Figures C.2)–(C.3 show screen-shots of the output file of the N-Skart procedure asking for a 90% CI on 50,000 simulation-generated observations of an $M/M/1$ queue-waiting-time process with an empty-and-idle initial condition, an interarrival rate of $\lambda = 0.9$ customers per-time-unit, and a service rate of $\mu = 1.0$ customers per-time-unit. In this system the steady-state expected waiting time is $\mu_X = 9.0$ time units. Figure C.4 shows the result of applying the N-Skart procedure to the same process explained above only when the available sample size is 200,000.

Figure C.2 demonstrates the result window when the user decides to quit the N-Skart procedure due to insufficient data. In order to continue with the randomness test the user would have to supply 53,611 observations. Figure C.3 shows the result window when the user chooses to continue with delivering a CI knowing that the randomness test could not be passed. In this case, the delivered CI might not provide the target confidence level. Finally, Figure C.4 demonstrates the result window when there is enough observation available to deliver a CI.



```
N-SkartCI - Notepad
File Edit Format View Help
model = MM1
coverage probability = 90%
given sample size = 50000
initial sample size = 1280

randomness test:
new sample size = 2304
new sample size = 3111
new sample size = 4670
new sample size = 6728
new sample size = 9084
new sample size = 11594
new sample size = 15350
new sample size = 19908
new sample size = 25398
new sample size = 32777
new sample size = 42120
the randomness test could not be passed. The procedure ended without delivering a CI.
the required sample size to continue the randomness test: 53611
```

Figure C.2: Screen-shot of the Notepad output window showing the result of applying N-Skart procedure to an $M/M/1$ queue-waiting-time process when the user decides to quit the procedure due to insufficient data


```

N-SkartCI - Notepad
File Edit Format View Help
model = MM1
coverage probability = 90%
given sample size = 50000

initial sample size = 1280

randomness test:
new sample size = 2304
new sample size = 3111
new sample size = 4670
new sample size = 6728
new sample size = 9084
new sample size = 11594
new sample size = 15350
new sample size = 19908
new sample size = 25398
new sample size = 32777
new sample size = 42120
the randomness test could not be passed due to insufficient data (to continue required: 53611)
the user choosed to continue with constructing a CI.
the delivered CI might not provide the target confidence level!!!

warm-up period = 608
final sample size = 49392

sample mean = 8.16
sample variance = 48.84

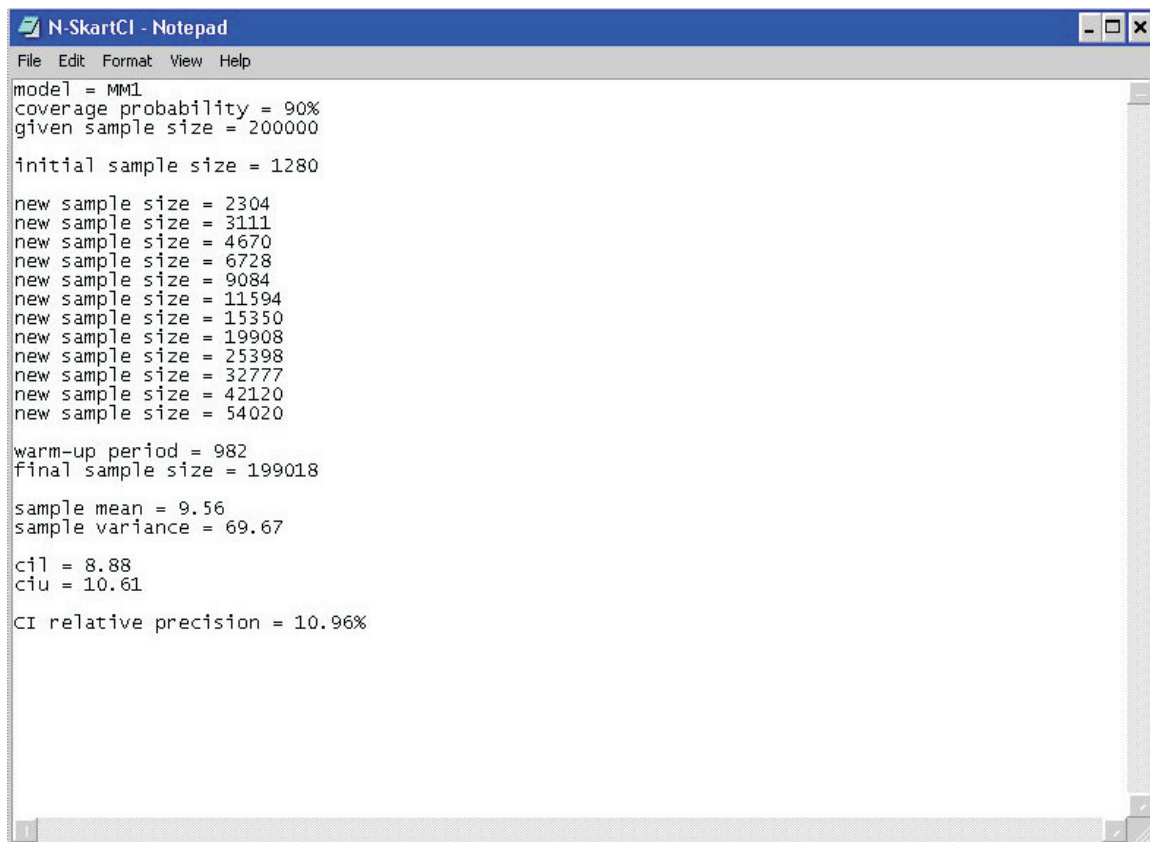
c1l = 7.23
ciu = 9.44

CI relative precision = 15.77%

-----
A CI with abnormally high relative precision should alert you regarding potential problems
with the delivered CI and a possible need for bigger sample size!!!!!!

```

Figure C.3: Screen-shot of the Notepad output window showing the result of applying N-Skart procedure to an $M/M/1$ queue-waiting-time process when the user chooses to continue with delivering a CI knowing that the randomness test could not be passed



```
model = MM1
coverage probability = 90%
given sample size = 200000

initial sample size = 1280

new sample size = 2304
new sample size = 3111
new sample size = 4670
new sample size = 6728
new sample size = 9084
new sample size = 11594
new sample size = 15350
new sample size = 19908
new sample size = 25398
new sample size = 32777
new sample size = 42120
new sample size = 54020

warm-up period = 982
final sample size = 199018

sample mean = 9.56
sample variance = 69.67

c1l = 8.88
c1u = 10.61

CI relative precision = 10.96%
```

Figure C.4: Screen-shot of the Notepad output window showing the result of applying N-Skart procedure to an $M/M/1$ queue-waiting-time process when there is enough observation available to deliver a CI

C.4 Visual Basic Codes for Skart and N-Skart Functions and Subroutines

```

Public Sub SkartProcedure(ByVal model As String, ByVal precReq As Boolean, ByVal precisionType As _
    precType, ByVal alpha As Double, ByVal hrstar As Double)

    Dim i, j As Integer
    Dim k As Integer = 1280           'initial batch count
    Dim kPrime As Integer           'current batch count
    Dim m As Integer                'batch size
    Dim alphaRan As Double = 0.2    'randomness test size
    Dim dStar As Integer            'maximum number of batches allowed in a spacer
    Dim d As Integer                'current number of batches in a spacer
    Dim w As Integer                'spacer size
    Dim b As Integer                'number of times the batch count has been deflated
    Dim nonspacedbatch(k - 1), spacedBatch(k - 1) As Double
    Dim sampleMean, sampleVar, standardError As Double
    Dim Skewness, halfLength, halfLengthStar As Double
    Dim A As Double                 'correlation adjustment
    Dim G1, G2 As Double
    xlsResultApp = CType(CreateObject("Excel.Application"), Excel.Application)
    reportFile = New StreamWriter("SkartCI.txt")

    reportFile.WriteLine("model = " & model )
    reportFile.WriteLine("precision requirement = " & precReq)
    If precisionType = precType.absolute Then
        reportFile.WriteLine("precision Type = absolute")
    Else
        reportFile.WriteLine("precision Type = relative")
    End If
    reportFile.WriteLine("coverage probability = " & (1 - alpha) * 100 & "%")
    reportFile.WriteLine("hrstar = " & hrstar * 100 & "%")
    reportFile.WriteLine()

    'generates initial sample of size 1,280
    Dim initialData() As Double = {}
    data = runSimulation(model, initialData, 1, k)

    'computes the sample skewness of the last 1024 observations in the sample: 1,024=1,280-256
    Skewness = SkewnessFun(data, 256)

    'sets the initial value of the batch size based on the computed sample skewness
    If Skewness <= 4 Then
        m = 1
        reportFile.WriteLine("initial sample size = " & m * k)
    Else
        m = 16
        'collects data: 20,480=1,280*16
        data = runSimulation(model, data, m, k)
        reportFile.WriteLine("initial sample size = " & m * k)
    End If
    reportFile.WriteLine()

    'computes the batch means
    For j = 0 To k - 1
        For i = 0 To m - 1
            nonspacedbatch(j) += data(j * m + i) / m
        Next
    Next

    reportFile.WriteLine("randomness test:")
    'Applies the von Neumann test for randomness
    Dim randomnessStatus As Boolean
    While randomnessStatus = False
        'sets the maximum number of batches allowed in a spacer
        Skewness = SkewnessFun(nonspacedbatch, k * 0.2)
        dStar = 10
        If Math.Abs(Skewness) > 0.5 Then
            dStar = 3
        End If

        randomnessStatus = vonNuemannTest(nonspacedbatch, alphaRan)
        If randomnessStatus = False Then
            While (randomnessStatus = False)
                d += 1
            End While
        End If
    End While

```

```

        kPrime = 0
        ReDim spacedBatch(Math.Floor(k / (d + 1)) - 1)
        For j = d To k - 1 Step d + 1
            spacedBatch(kPrime) = nonspacedbatch(j)
            kPrime += 1
        Next
        randomnessStatus = vonNuemannTest(spacedBatch, alphaRan)
        If (d = dStar) Then Exit While
    End While
    If (randomnessStatus = False And d = dStar) Then
        m = Math.Ceiling(Math.Sqrt(2) * m)
        k = Math.Ceiling(0.9 * k)
        data = runSimulation(model, data, m, k)
        reportFile.WriteLine("new sample size = " & m * k)
        d = 0
        b += 1
        For j = 0 To k - 1
            nonspacedbatch(j) = 0
            For i = 0 To m - 1
                nonspacedbatch(j) += data(j * m + i) / m
            Next
        Next
    End If
Else
    kPrime = k
End If
End While

'skips the warm-up period
w = d * m
data = runSimulation(model, data, m, kPrime, w)

kPrime = Math.Ceiling(kPrime * (1 / 0.9) ^ b)
m = Math.Max(m, Math.Floor(data.Length / kPrime))
ReDim nonspacedbatch(kPrime - 1)
If m * kPrime > data.Length Then
    data = runSimulation(model, data, m, kPrime)
    reportFile.WriteLine("new sample size = " & m * kPrime)
End If

'computes the current set of truncated, nonspaced batch means
For j = 0 To kPrime - 1
    For i = 0 To m - 1
        nonspacedbatch(j) += data(j * m + i) / m
    Next
Next

'computes the sample mean and variance
For j = 0 To kPrime - 1
    sampleMean += nonspacedbatch(j) / kPrime
Next
For j = 0 To kPrime - 1
    sampleVar += (nonspacedbatch(j) - sampleMean) ^ 2 / (kPrime - 1)
Next

'computes the correlation adjustment and the CI's standard error
Dim sigma, phiHat As Double
For j = 0 To kPrime - 2
    sigma += (nonspacedbatch(j) - sampleMean) * (nonspacedbatch(j + 1) - sampleMean)
Next
phiHat = sigma / (sampleVar * (kPrime - 1))
A = (1 + phiHat) / (1 - phiHat)
standardError = Math.Sqrt(A * sampleVar / kPrime)
If precReq = True And precisionType = precType.relative Then
    halfLengthStar = hrstar * sampleMean
Else
    halfLengthStar = hrstar
End If
halfLength = SkewnessAdj(nonspacedbatch, kPrime, m, alpha, _
    sampleMean, sampleVar, standardError, w, G1, G2)

```

```

Dim kPrimeNew, maxVal As Double
reportFile.WriteLine()
If precReq = True Then reportFile.WriteLine("precision requirement:")
While (halfLength > halfLengthStar And precReq = True)
    kPrimeNew = Math.Ceiling(((halfLength / halfLengthStar) ^ 2) * kPrime)
    If kPrime = 1024 Then
        maxVal = Math.Max(1.02, ((halfLength / halfLengthStar) ^ 2))
        m = Math.Ceiling(Math.Min(maxVal, 2) * m)
    End If
    kPrime = Math.Min(kPrimeNew, 1024)
    ReDim nonspacedbatch(kPrime - 1)
    data = runSimulation(model, data, m, kPrime)
    reportFile.WriteLine("new sample size = " & m * kPrime)
    For j = 0 To kPrime - 1
        nonspacedbatch(j) = 0
        For i = 0 To m - 1
            nonspacedbatch(j) += data(j * m + i) / m
        Next
    Next
    sampleVar = 0
    sampleMean = 0
    'computes the sample mean and variance
    For j = 0 To kPrime - 1
        sampleMean += nonspacedbatch(j) / kPrime
    Next
    For j = 0 To kPrime - 1
        sampleVar += (nonspacedbatch(j) - sampleMean) ^ 2 / (kPrime - 1)
    Next

    'computes the correlation adjustment and the CI's standard error
    sigma = 0
    For j = 0 To kPrime - 2
        sigma += (nonspacedbatch(j) - sampleMean) * (nonspacedbatch(j + 1) - sampleMean)
    Next
    phiHat = sigma / (sampleVar * (kPrime - 1))
    A = (1 + phiHat) / (1 - phiHat)
    standardError = Math.Sqrt(A * sampleVar / kPrime)
    If precReq = True And precisionType = precType.relative Then
        halfLengthStar = hrstar * sampleMean
    Else
        halfLengthStar = hrstar
    End If
    halfLength = SkewnessAdj(nonspacedbatch, kPrime, m, alpha, _
        sampleMean, sampleVar, standardError, w, G1, G2)
End While

Dim CIlb, CIub As Double
If precReq = True Then
    CIlb = sampleMean - halfLength
    CIub = sampleMean + halfLength
Else 'no precision requirement
    CIlb = sampleMean - G1 * standardError
    CIub = sampleMean - G2 * standardError
End If

reportFile.WriteLine()
reportFile.WriteLine("warm-up period = " & w)
reportFile.WriteLine("final sample size = " & kPrime * m)
reportFile.WriteLine()
reportFile.WriteLine("sample mean = " & Format(sampleMean, "0.00"))
reportFile.WriteLine("sample variance = " & Format(sampleVar, "0.00"))
reportFile.WriteLine()
reportFile.WriteLine("cil = " & Format(CIlb, "0.00"))
reportFile.WriteLine("ciu = " & Format(CIub, "0.00"))
reportFile.Close()

End Sub

```

```

Public Sub NSkartProcedure(ByVal model As String, ByVal alpha As Double, ByVal sampleSize As _
    Integer)

    Dim i, j As Integer
    Dim k As Integer = 1280           'initial batch count
    Dim kPrime As Integer             'current batch count
    Dim m As Integer                 'batch size
    Dim alphaRan As Double = 0.2     'randomness test size
    Dim dStar As Integer             'maximum number of batches allowed in a spacer
    Dim d As Integer                 'current number of batches in a spacer
    Dim w As Integer                 'spacer size
    Dim b As Integer                 'number of times the batch count has been deflated
    Dim nonspacedbatch(k - 1), spacedBatch(k - 1) As Double
    Dim sampleMean, sampleVar, standardError As Double
    Dim Skewness, halfLength As Double
    Dim A As Double                  'correlation adjustment
    Dim G1, G2 As Double
    Dim CIrisk As Boolean
    xlsResultApp = CType(CreateObject("Excel.Application"), Excel.Application)
    reportFile = New StreamWriter("N-SkartCI.txt")

    reportFile.WriteLine("model = " & model)
    reportFile.WriteLine("coverage probability = " & (1 - alpha) * 100 & "%")
    reportFile.WriteLine("given sample size = " & sampleSize)
    reportFile.WriteLine()

    'generates sample of given fix size
    Dim initialData() As Double = {0}
    data = runSimulation(model, initialData, 1, sampleSize)

    'computes the sample skewness of the last 80% of the given sample
    Skewness = SkewnessFun(data, sampleSize * 0.2)

    'sets the initial value of the batch size based on the computed sample skewness
    If Skewness <= 4 Then
        m = 1
        reportFile.WriteLine("initial sample size = " & m * k)
    Else
        m = Math.Floor(Math.Min(16, sampleSize / 1280))
        reportFile.WriteLine("initial sample size = " & m * k)
    End If
    reportFile.WriteLine()

    'computes the batch means
    For j = 0 To k - 1
        For i = 0 To m - 1
            nonspacedbatch(j) += data(j * m + i) / m
        Next
    Next

    reportFile.WriteLine("randomness test:")
    'Applies the von Neumann test for randomness
    Dim randomnessStatus As Boolean
    While randomnessStatus = False
        'sets the maximum number of batches allowed in a spacer
        Skewness = SkewnessFun(nonspacedbatch, k * 0.2)
        dStar = 10
        If Math.Abs(Skewness) > 0.5 Then
            dStar = 3
        End If

        randomnessStatus = vonNuemannTest(nonspacedbatch, alphaRan)
    If randomnessStatus = False Then
        While (randomnessStatus = False)
            d += 1
            kPrime = 0
            ReDim spacedBatch(Math.Floor(k / (d + 1)) - 1)
            For j = d To k - 1 Step d + 1
                spacedBatch(kPrime) = nonspacedbatch(j)
                kPrime += 1
            Next
        End While
    End If

```

```

        randomnessStatus = vonNuemannTest(spacedBatch, alphaRan)
    If (d = dStar) Then Exit While
End While
If (randomnessStatus = False And d = dStar) Then
    'check if the updates would cause the required sample size to get higher than _
    'available(data)
    If Math.Ceiling(Math.Sqrt(2) * m * 0.9 * k) > sampleSize Then
        'issue a warning that the randomness test could not be passed
        If MessageBox.Show("The randomness test could not be passed due to _
        insufficient data. Do you want to continue?", _
        "N-Skart Procedure", MessageBoxButtons.YesNo, MessageBoxIcon.Question) _
        = DialogResult.Yes Then
            randomnessStatus = True
            reportFile.WriteLine("the randomness test could not be passed due to _
            insufficient data (to continue required: " & _
            Math.Ceiling(Math.Sqrt(2) * m * 0.9 * k) & ")")
            reportFile.WriteLine("the user choosed to continue with constructing _
            a CI.")
            reportFile.WriteLine("the delivered CI might not provide the target _
            confidence level!!!")
            reportFile.WriteLine()
            CIrisk = True
        Else
            reportFile.WriteLine("the randomness test could not be passed. The _
            procedure ended without delivering a CI.")
            reportFile.WriteLine("the required sample size to continue the _
            randomness test: " & Math.Ceiling(Math.Sqrt(2) * m * 0.9 * k))
            reportFile.Close()
            Exit Sub
        End If
    End If

Else
    m = Math.Ceiling(Math.Sqrt(2) * m)
    k = Math.Ceiling(0.9 * k)
    reportFile.WriteLine("new sample size = " & m * k)
    d = 0
    b += 1
    For j = 0 To k - 1
        nonspacedbatch(j) = 0
        For i = 0 To m - 1
            nonspacedbatch(j) += data(j * m + i) / m
        Next
    Next
End If
End If
Else
    kPrime = k
End If
End While

'initial warm-up period
w = d * m

'reinflate the batch count
kPrime = Math.Min(Math.Ceiling(kPrime * (1 / 0.9) ^ b), k)

'increase the batch count and batch size to use all the given data
Dim f As Double
f = Math.Sqrt((sampleSize - w) / (kPrime * m))
kPrime = Math.Min(Math.Floor(f * kPrime), 1024)
If kPrime < 1024 Then
    m = Math.Floor(f * m)
Else
    m = Math.Floor((sampleSize - w) / 1024)
End If

'recompute the warm-up period
w = sampleSize - m * kPrime

ReDim nonspacedbatch(kPrime - 1)

```



```

'Throw away the warm-up period
data = runSimulation(model, data, m, kPrime, w)

'computes the current set of truncated, nonspaced batch means
For j = 0 To kPrime - 1
    For i = 0 To m - 1
        nonspacedbatch(j) += data(j * m + i) / m
    Next
Next

'computes the sample mean and variance
For j = 0 To kPrime - 1
    sampleMean += nonspacedbatch(j) / kPrime
Next
For j = 0 To kPrime - 1
    sampleVar += (nonspacedbatch(j) - sampleMean) ^ 2 / (kPrime - 1)
Next

'computes the correlation adjustment and the CI's standard error
Dim sigma, phiHat As Double
For j = 0 To kPrime - 2
    sigma += (nonspacedbatch(j) - sampleMean) * (nonspacedbatch(j + 1) - sampleMean)
Next
phiHat = sigma / (sampleVar * (kPrime - 1))
A = (1 + phiHat) / (1 - phiHat)
standardError = Math.Sqrt(A * sampleVar / kPrime)
halfLength = SkewnessAdj(nonspacedbatch, kPrime, m, alpha, _
    sampleMean, sampleVar, standardError, w, G1, G2)

Dim CIlb, CIub As Double
'no precision requirement
CIlb = sampleMean - G1 * standardError
CIub = sampleMean - G2 * standardError

reportFile.WriteLine("warm-up period = " & w)
reportFile.WriteLine("final sample size = " & kPrime * m)
reportFile.WriteLine()
reportFile.WriteLine("sample mean = " & Format(sampleMean, "0.00"))
reportFile.WriteLine("sample variance = " & Format(sampleVar, "0.00"))
reportFile.WriteLine()
reportFile.WriteLine("cil = " & Format(CIlb, "0.00"))
reportFile.WriteLine("ciu = " & Format(CIub, "0.00"))
reportFile.WriteLine()
reportFile.WriteLine("CI relative precision = " & Format(Math.Max(sampleMean - CIlb, _
CIub - sampleMean) * 100 / sampleMean, "0.00") & "%")
If CIrisk = True Then
    reportFile.WriteLine()
    reportFile.WriteLine("-----")
    reportFile.WriteLine("A CI with abnormally high relative precision should alert _
you regarding potential problems")
    reportFile.WriteLine("with the delivered CI and a possible need for bigger _
sample size!!!!!!")
End If
reportFile.Close()
End Sub

```

```
'main subroutine for N-Skart procedure
Sub Main()
    NSkartProcedure("MM1", 0.1, 50000)
End Sub
```

```
'main subroutine for Skart procedure
Sub Main()
    SkartProcedure("MM1", True, precType.relative, 0.1, 0.15)
End Sub
```

```
Public Function SkewnessFun(ByVal myData() As Double, Optional ByVal cutOff As Integer = 0) As _
Double
    Dim j As Integer
    Dim skewness, mean, mom2, mom3, mom4 As Double
    Dim dataLength As Integer = myData.Length()

    For j = cutOff To dataLength - 1
        mean += myData(j) / (dataLength - cutOff)
    Next
    For j = cutOff To dataLength - 1
        mom2 += (myData(j) - mean) ^ 2
        mom3 += (myData(j) - mean) ^ 3
    Next
    SkewnessFun = ((dataLength - cutOff) ^ 2.5 * mom3) / (mom2 ^ 1.5 * ((dataLength - cutOff) - 1)_)
        * ((dataLength - cutOff) - 2))
End Function
```

```
Public Function vonNuemannTest(ByVal myData() As Double, ByVal alpha As Double) As Boolean

    Dim j As Integer
    Dim dataSize As Integer = myData.Length

    'Compute the von Neumann test statistic
    Dim sampleMean, sampleVar As Double
    For j = 0 To dataSize - 1
        sampleMean += myData(j) / dataSize
    Next
    For j = 0 To dataSize - 1
        sampleVar += (myData(j) - sampleMean) ^ 2
    Next
    'Compute the mean square successive difference
    Dim succDiff As Double
    For j = 0 To dataSize - 2
        succDiff += (myData(j) - myData(j + 1)) ^ 2
    Next
    Dim ranTestStat As Double = 1 - (succDiff / (2 * sampleVar))

    'take alpha=0.2 for the 2-sided test
    Dim P As Double
    Dim test As Double

    P = xlsResultApp.Application.WorksheetFunction.NormSInv(1 - alpha / 2)
    If Math.Abs(ranTestStat) > P * Math.Sqrt((dataSize - 2) / (dataSize ^ 2 - 1)) Then
        vonNuemannTest = False
    Else
        vonNuemannTest = True
    End If
End Function
```

```

Public Function SkewnessAdj(ByVal nonspacedbatch() As Double, ByVal kPrime As Double, _
    ByVal m As Double, ByVal alpha As Double, _
    ByVal sampleMean As Double, _
    ByVal sampleVar As Double, ByVal standardError As Double, _
    ByVal w As Integer, ByRef G1 As Double, ByRef G2 As Double) As Double

    Dim j As Integer
    Dim spacedBatch(kPrime - 1) As Double

    'computes spaced batch means to calculate the skewness adjustment
    Dim dPrime As Integer = Math.Ceiling(w / m) 'number of batches in a spacer
    Dim kDoublePrime As Integer
    ReDim spacedBatch(Math.Floor(kPrime / (dPrime + 1)) - 1)
    For j = dPrime To kPrime - 1 Step dPrime + 1
        spacedBatch(kDoublePrime) = nonspacedbatch(j)
        kDoublePrime += 1
    Next

    Dim spacedSampleMean, spacedSampleVar, spacedMom3 As Double
    For j = 0 To kDoublePrime - 1
        spacedSampleMean += spacedBatch(j) / kDoublePrime
    Next
    For j = 0 To kDoublePrime - 1
        spacedSampleVar += (spacedBatch(j) - spacedSampleMean) ^ 2 / (kDoublePrime - 1)
        spacedMom3 += (kDoublePrime * (spacedBatch(j) - spacedSampleMean) ^ 3) / _
            ((kDoublePrime - 1) * (kDoublePrime - 2))
    Next

    Dim beta As Double = (spacedMom3 / spacedSampleVar ^ 1.5) / (6 * Math.Sqrt(kDoublePrime))
    Dim t1 As Double = xlsResultApp.Application.WorksheetFunction.TInv(alpha, _
        kDoublePrime - 1)
    Dim t2 As Double = -t1
    If (1 + 6 * beta * (t1 - beta)) < 0 Then
        G1 = ((2 * beta) ^ (-1)) * (-Math.Abs(1 + 6 * beta * (t1 - beta)) ^ (1 / 3)) - 1)
    Else
        G1 = ((2 * beta) ^ (-1)) * (((1 + 6 * beta * (t1 - beta)) ^ (1 / 3)) - 1)
    End If
    If (1 + 6 * beta * (t2 - beta)) < 0 Then
        G2 = ((2 * beta) ^ (-1)) * (-Math.Abs(1 + 6 * beta * (t2 - beta)) ^ (1 / 3)) - 1)
    Else
        G2 = ((2 * beta) ^ (-1)) * (((1 + 6 * beta * (t2 - beta)) ^ (1 / 3)) - 1)
    End If
    SkewnessAdj = Math.Max(standardError * Math.Abs(G1), standardError * Math.Abs(G2))
End Function

```

```

Public Function runSimulation(ByVal model As String, ByVal givenData() As Double, _
    ByVal batchsize As Integer, ByVal batchcount As Integer, _
    Optional ByVal spacerLength As Integer = 0) As Double()

    If model = "MM1" Then
        runSimulation = GenerateMM1(givenData, batchsize, batchcount, spacerLength)
    ElseIf model = "MH2" Then
        runSimulation = GenerateMH2(givenData, batchsize, batchcount, spacerLength)
    ElseIf model = "AR" Then
        runSimulation = GenerateAR1(givenData, batchsize, batchcount, spacerLength)
    ElseIf model = "ARTOP" Then
        runSimulation = GenerateARTOP(givenData, batchsize, batchcount, spacerLength)
    End If
End Function

```
