Abstract

LADA, EMILY KATE. A Wavelet-Based Procedure for Steady-State Simulation Output Analysis. (Under the direction of Dr. James R. Wilson.)

The objective of this research is to develop an automated sequential procedure by which an asymptotically valid confidence interval is constructed for the steady-state mean of a simulation output process. This procedure, called **WASSP**, determines a batch size and a warm-up period beyond which the computed batch means constitute an approximately stationary Gaussian process. **WASSP** then uses wavelets to approximate the log of the smoothed periodogram of the batch means process, from which an estimate of the steady-state variance constant (SSVC) of the original (unbatched) process is obtained. Together with a sample mean that has been suitably truncated to eliminate initialization bias, the SSVC estimator is used to construct a reliable confidence-interval estimator of the steady-state mean that satisfies a user-specified absolute or relative precision requirement. An extensive performance evaluation includes testing **WASSP** on a suite of processes that include extreme examples of the warm-up and correlation problems. The results indicate that **WASSP** is successful in detecting and eliminating initialization bias as well as in constructing an approximately stationary process so that an asymptotically valid confidence interval for the steady-state mean can be generated even if the original process is highly

correlated and has a pronounced initial transient period. Furthermore, the performance evaluation also includes a comparison of *WASSP* to other methods for steady-state output analysis. The results indicate that *WASSP* is in general a more robust procedure than the other methods, and we believe that *WASSP* represents an advance in spectral methods for steady-state output analysis.

A WAVELET-BASED PROCEDURE FOR STEADY-STATE SIMULATION OUTPUT ANALYSIS

by

EMILY KATE LADA

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

OPERATIONS RESEARCH

Raleigh

2003

APPROVED BY:

Chair of Advisory Committee

Put all your soul into it, play the way you feel!

—Frédéric Chopin

Biography

Emily Lada was born December 18, 1974, in Durham, North Carolina. She graduated from Athens Drive High School in 1993. In the fall of 1993 she entered the University of North Carolina at Chapel Hill. She graduated with highest distinction in May 1997, receiving a Bachelor of Arts degree in Mathematics with a minor in Music.

In the fall of 1997 she began graduate study at North Carolina State University. She received her Master of Science in Operations Research in 2000. While at NC State, she participated in the Preparing the Professoriate Program. She also received a North Carolina State University Dean of Engineering Fellowship, a General Electric Faculty for the Future Fellowship, and a Selected Professions Fellowship from the American Association of University Women (AAUW). After completing her Ph.D., Emily hopes to find a faculty position at a university where she can teach Operations Research courses and continue her research with wavelets and large-scale simulation modeling and analysis.

Acknowledgments

One does not complete a dissertation alone. I would first like to thank Dr. James R. Wilson for serving as my advisor. I was shocked the first time Dr. Wilson sent me an email at 4:00 A.M. I have since come to realize that Dr. Wilson is completely dedicated to helping students succeed, even if it means only getting a few hours of sleep. Dr. Wilson has always kept my best interests in mind, and for that I sincerely thank him.

I would also like to thank Dr. Steve Roberts for serving on my committee. I truly value and appreciate all the advice he has given me over the years. I thank Dr. Charlie Smith for also serving on my committee and for all the enlightening conversations we have had (and of course, for the very handy erasers I found in my mailbox every holiday). Thank you also to Dr. Elmor Peterson for serving on my committee. I have tremendous respect for Dr. Peterson and the contributions he has made to the field of Operations Research. I sincerely appreciate the guidance he has given me since I have been at NC State.

I thank Dr. David Goldsman and Dr. Brani Vidakovic from Georgia Tech for all their insight and advice. I have enjoyed working with Dr. Goldsman (he always finds a way to keep you smiling), and I hope we will be able to work together again in the future. Dr. Vidakovic's expertise in the field of wavelets was a tremendous help to me, and I hope I will be able to collaborate with him as well on future research projects. I also thank Dr. Natalie Steiger for being my friend and mentor. Her work on ASAP was fundamental to the development of this dissertation.

To my family—I cannot thank you enough for your never-ending love and support. Thank you mom and dad for always having high aspirations for me. I am who I am today because of

you. Dad, your math genes have come in handy too! Thank you Michael for always supporting me through all the struggles and victories.

I finally would like to acknowledge the following sources of financial support: NSF grant DMI-9900164, the American Association of University Women Selected Professions Fellowship, and the General Electric Faculty for the Future Teaching Fellowship.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Simulation is one of the most popular tools for studying the behavior of complex systems. Its popularity is due in part to its applicability in a wide variety of fields, from manufacturing to healthcare. Discrete-event stochastic simulation is a technique for modeling the operation of a target system over time by a numerical (usually computer-based) representation—that is, a simulation model— in which changes in the state of the model occur only at a countable set of distinct points in time [39]. These state changes generally represent complex interactions among several sequences of random variables that constitute the probabilistic input processes driving the operation of the simulation model; and the modeling objective is to mimic the operation of the target system with a degree of accuracy that is sufficient for the purpose of the study at hand.

A nonterminating simulation is one in which we are interested in long-run (steady-state) average performance measures. Let $\{X_i : i = 1, 2, \ldots\}$ denote a stochastic process representing the sequence of outputs generated by a single run of a nonterminating probabilistic simulation. For example, X_i might represent the time in the system (cycle time) for the ith workpiece to complete all its processing in a simulation of a production factility. If the simulation is in steady-state operation, then the random variables $\{X_i\}$ will have the same steady-state cumulative distribution function (c.d.f.), $F_X(x) = \Pr\{X_i \leq x\}$ for all real x and for $i = 1, 2, \ldots$.

Usually in a nonterminating probabilistic simulation, we are interested in constructing point and confidence interval estimators for some parameter, or characteristic, of the steady-state c.d.f. $F_X(x)$. In this research, we are primarily interested in estimating the steady-state mean, $\mu_X = \mathrm{E}[X] = \int_{-\infty}^{\infty} x \, dF_X(x)$; and we limit the discussion to output processes for which $\mathrm{E}[X_i^2] < \infty$ so that the process mean μ_X and process variance $\sigma_X^2 = \mathrm{Var}[X_i] = \mathrm{E}[(X_i - \mu_X)^2]$ are well defined. We let n denote the length of the time series $\{X_i\}$ of outputs generated by a single, long run of the simulation; and we often refer to n as the *run length*.

The sample mean,

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i, \tag{1.1}$$

is an intuitively appealing point estimate of μ_X. One might, however, question the amount of uncertainty associated with this point estimate; and this consideration naturally leads to the idea of building a confidence interval for μ_X that is centered on $\overline{X}$ and has a half-length which appropriately accounts for the variability (variance) of $\overline{X}$. An estimate of the process variance σ_X^2 is the sample variance,

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

If the X_i's are independent and identically distributed (i.i.d.) normal random variables and if $0 < \beta < 1$, then a standard $100(1-\beta)\%$ confidence interval for μ_X is

$$\overline{X} \pm t_{1-\beta/2;\, n-1}\frac{S}{\sqrt{n}}, \tag{1.2}$$

where $t_{1-\beta/2;\, n-1}$ is the $1-\beta/2$ quantile of Student's t-distribution with $n-1$ degrees of freedom. Moreover, if the X_i's are i.i.d. but not necessarily normal, then an *asymptotically valid* $100(1-\beta)\%$ confidence interval for μ_X is

$$\overline{X} \pm z_{1-\beta/2}\frac{S}{\sqrt{n}}, \tag{1.3}$$

where $z_{1-\beta/2}$ is the $1-\beta/2$ quantile of the standard normal distribution; and this means that the corresponding *coverage probability* satisfies

$$\lim_{n\to\infty} \Pr\left\{\mu_X \in \ \overline{X} \pm z_{1-\beta/2}\frac{S}{\sqrt{n}}\right\} = 1-\beta. \tag{1.4}$$

There are two fundamental problems associated with analyzing stochastic output from a nonterminating simulation. The first problem is that it is usually impossible to start a simulation in steady-state operation, thereby making it necessary to decide how long the "warm-up" period should be so that for each simulation output X_l generated after the end of the warm-up period, the corresponding expected value $\mathrm{E}[X_l]$ is sufficiently close to the steady-state mean μ_X. If observations generated prior to the end of the warm-up period are included in the analysis, then the resulting point estimator $\overline{X}$ will be biased; and such bias in the point estimator (1.1) can severely degrade the coverage probability of the corresponding confidence intervals (1.2) and (1.3) [39, 70, 71]. This is known as the start-up or initialization bias problem.

The second problem associated with analyzing output from a nonterminating probabilistic

simulation is that the observations $\{X_i\}$ are typically correlated; and this phenomenon can also invalidate the confidence intervals given by (1.2) and (1.3). In particular, it is assumed in (1.2) and (1.3) that

$$\mathrm{Var}[\,\overline{X}\,] = \frac{\sigma_X^2}{n} \tag{1.5}$$

so that we can use $S/\sqrt{n}$ as an estimator of $\sqrt{\mathrm{Var}[\overline{X}]}$, the standard error of $\overline{X}$, in the confidence intervals (1.2) and (1.3) for μ_X. If the observations are correlated, however, then equation (1.5) for the variance of $\overline{X}$ is not generally true.

Before we can establish what the variance of $\overline{X}$ is when the $\{X_i\}$ are correlated, we first need to state some basic definitions. The stochastic process $\{X_i\}$ is said to be *weakly stationary* if it has finite second moments and a constant mean,

$$\mathrm{E}[X_i^2] < \infty \;\text{ and }\; \mathrm{E}[X_i] = \mu_X \quad \text{for all } i,$$

and if the covariance between two observations X_i and X_{i+l},

$$\gamma_X(l) = \mathrm{Cov}[X_i, X_{i+l}] = \mathrm{E}[(X_i - \mu_X)(X_{i+l} - \mu_X)] \text{ for all } i \text{ and } l = 0, \pm 1, \pm 2, \ldots, \tag{1.6}$$

depends only on the absolute value $|l|$ of the time difference (or lag) $(i + l) - i = l$. Thus in a weakly stationary process $\{X_i\}$, we have a constant variance $\mathrm{Var}[X_i] = \gamma_X(0) = \sigma_X^2$ for all i; moreover, $\gamma_X(l) = \gamma_X(-l)$ for $l = 1, 2, \ldots$. The autocorrelation function at lag l is defined as

$$\rho_X(l) = \frac{\gamma_X(l)}{\gamma_X(0)} = \frac{\gamma_X(l)}{\sigma_X^2} \quad \text{for} \quad l = 0, \pm 1, \pm 2, \ldots\,,$$

which is a measure of the linear association between two random variables, X_i and X_{i+l}, for all i.

If the process $\{X_i : i = 1, \ldots n\}$ is weakly stationary, then

$$\begin{aligned}
\mathrm{Var}[\overline{X}] &= \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathrm{Cov}[X_i, X_j] \\
&= \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \gamma_X(|j - i|) \\
&= \frac{1}{n^2} \left[n\gamma_X(0) + 2 \sum_{l=1}^{n-1} (n - l)\gamma_X(l) \right] \\
&= \frac{\sigma_X^2}{n} \left[1 + 2 \sum_{l=1}^{n-1} \left(1 - \frac{l}{n} \right) \rho_X(l) \right],
\end{aligned} \tag{1.7}$$

which clearly is not equal to the result obtained for i.i.d. $\{X_i\}$ in equation (1.5). However, if in equation (1.7) we have

$$\rho_X(l) = 0 \quad \text{for} \quad l = 1, 2, \ldots, \tag{1.8}$$

then (1.5) holds and under some weak-dependence conditions on the process $\{X_i\}$ that are detailed below, the confidence interval (1.3) is asymptotically valid in the sense of (1.4) even when the marginal c.d.f. $F_X(x)$ is not normal. Unfortunately most output processes generated by steady-state simulations do not statisfy (1.8); and thus we must seek a generalization of (1.3) in which the term $S/\sqrt{n}$ is replaced by a suitable estimator of $\sqrt{\text{Var}[\overline{X}]}$.

Random variables like the $\{X_i\}$ that are derived from a steady-state simulation are often *weakly dependent* in the sense of ϕ-mixing [6]. This means roughly that X_i's far from each other in the sequence are almost independent, implying that the lag-l correlation $\rho_X(l) \to 0$ as $|l| \to \infty$. In order to construct a confidence interval for the steady-state mean of the output process $\{X_i\}$, we need an estimate of the variance of $\overline{X}$. By applying an appropriate central limit theorem for dependent processes (for example, Theorem 20.1 of [6]), we see that under fairly general conditions on the process $\{X_i\}$, the following limit results hold:

- We have the absolute summability property

$$\sum_{l=-\infty}^{\infty} |\gamma_X(l)| < \infty \tag{1.9}$$

  so that γ_X, the *steady-state variance constant* (SSVC) for the process $\{X_i\}$ is well defined and given by

$$\gamma_X \equiv \lim_{n\to\infty} n\text{Var}[\overline{X}] = \sum_{l=-\infty}^{\infty} \gamma_X(l) = \gamma_X(0) + 2\sum_{l=1}^{\infty} \gamma_X(l); \tag{1.10}$$

  moreover, we have $\gamma_X > 0$.

- We have the key weak-convergence property

$$\sqrt{n}\,[\overline{X} - \mu_X] \xrightarrow{D} N(0, \gamma_X) \quad \text{as} \quad n \to \infty. \tag{1.11}$$

The limit results (1.10) and (1.11) provide a basis for constructing an asymptotically valid confidence interval for μ_X. From (1.10) we see that the variance of $\overline{X}$ is given asymptotically by

$$\text{Var}[\overline{X}] = \frac{\gamma_X}{n} + o\left(\frac{1}{n}\right) \quad \text{as} \quad n \to \infty. \tag{1.12}$$

Combining (1.11) and (1.12), we see that in general an asymptotically valid $100(1-\beta)\%$ confidence interval for μ_X is given by

$$\overline{X} \pm z_{1-\beta/2}\sqrt{\frac{\gamma_X}{n}}. \tag{1.13}$$

Therefore, if it is possible to obtain a suitably behaved estimate of the SSVC γ_X as well as a version of the sample mean $\overline{X}$ that has been suitably truncated to eliminate the effects of initialization bias, then a variant of display (1.13) might be used to construct asymptotically valid confidence intervals for the steady-state mean of a simulation-generated output process with correlated observations.

1.1. Scope and Objectives of Research

The primary objective of this research is to develop a completely automated sequential procedure by which an asymptotically valid confidence interval is constructed for the steady-state mean μ_X of a simulation output process $\{X_i\}$. This procedure, called **WASSP**, will address both the initialization bias and correlation problems and deliver an approximate $100(1-\beta)\%$ confidence interval of the form

$$\overline{X} \pm t_{1-\beta/2,\,\nu}\sqrt{\frac{\widehat{\gamma}_X}{n}}, \tag{1.14}$$

where $t_{1-\beta/2,\,\nu}$ is the $(1-\beta/2)$-quantile of the Student's t-distribution with ν degrees of freedom.

The performance evaluation phase of the research will focus primarily on processes for which the theoretical steady-state parameters μ_X, σ_X^2, and γ_X are known, such as the sequence $\{X_i\}$ of queueing times in the $M/M/1$ queue. In this case, X_i is the waiting time for the ith customer in a single-server queueing system with the following characteristics that induce highly significant effects on the $\{X_i\}$ due to initialization bias and correlation among successive outputs: i.i.d. exponential interarrival times; i.i.d. exponential service times; a steady-state server utilization that is sufficiently high (say, 90%); and an initial condition that is sufficiently atypical of steady-state operation (say, an empty-and-idle initial condition so that $X_1 = 0$).

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

(a) Develop a sequential independence test that can be used to—

   i) construct a set of spaced batch means such that the interbatch spacer preceding each batch is sufficiently large to ensure all computed batch means are approximately independent and identically distributed so that the batch means can subsequently be tested for normality; and

ii) determine an appropriate warm-up period—that is, the "interbatch" spacer preceding the first batch—beyond which all computed batch means are approximately independent of the simulation model's initial conditions.

(b) Develop a sequential normality test for determining an appropriate batch size so that if we also use the spacer size determined in item (a).i) above, then the resulting set of spaced batch means forms an approximately i.i.d. Gaussian process.

(c) Develop an automated procedure to compute a wavelet-based estimator of the power spectrum of the adjacent (nonspaced) batch means that are computed beyond the end of the warm-up period determined in item (a).ii) above using the batch size determined in item (b) above. The wavelet-based spectrum estimator can then be used to compute an estimator of the SSVC γ_X.

(d) Formulate a sequential confidence-interval estimation procedure that delivers an approximate $100(1-\beta)\%$ confidence interval of the form (1.14) and that satisfies a user-specified absolute or relative precision requirement. This procedure will incorporate

- the truncated sample mean $\overline{X}$ that is obtained by averaging all the adjacent (nonspaced) batch means beyond the end of the warm-up period identified in item (a).ii) above; and
- the wavelet-based SSVC estimator $\widehat{\gamma}_X$ developed in item (c) above.

(e) Conduct an extensive performance evaluation of the overall procedure for confidence interval estimation (that is, $\mathcal{WASSP}$), including a comparison of $\mathcal{WASSP}$ to other methods for steady-state output analysis. The performance evaluation phase will consist of testing $\mathcal{WASSP}$ on a suite of problems that include extreme examples of the warm-up and correlation problems. Many of the test problems will also exhibit the same characteristics as a variety of practical simulation applications.

(f) Develop, verify (debug), and document user-friendly, computationally efficient software that will automatically apply $\mathcal{WASSP}$ to a given set of observations and construct a confidence interval for the mean. The software will be designed so that it can be used in conjunction with popular simulation packages like Arena [36].

1.2. Organization of the Dissertation

The rest of this dissertation is organized as follows. The first part of Chapter 2 reviews existing nonspectral methods for steady-state simulation output analysis, followed by a discussion of Fourier methods and wavelet theory. Chapter 2 concludes with a discussion of existing spectral

methods for steady-state simulation output analysis. Chapter 3 presents a detailed algorithmic statement of WASSP together with a complete justification of the statistical and numerical results on which WASSP is based. In Chapter 4, we present the results of applying WASSP and its competitors to a suite of five test processes that were designed specifically to explore the robustness of the selected procedures against the statistical anomalies commonly encountered in the analysis of outputs generated by large-scale steady-state simulation experiments. Finally, conclusions and recommendations for future research are detailed in Chapter 5.

The appendices provide complete technical details on the formulation, evaluation, and practical application of WASSP. Appendices A and B contain derivations of some statistical properties of the log of the smoothed periodogram that form the basis for the wavelet-based spectrum estimator used in WASSP. Appendices C and D contain derivations of numerical methods that we implemented to evaluate key steady-state characteristics of the test problems used in the experimental performance evaluation of WASSP. Finally Appendix E provides a user's manual for WASSP, including instructions on downloading, installing, and running the WASSP software.

Chapter 2

Literature Review

In this chapter we will first give a brief overview of several nonspectral methods for steady-state simulation output analysis. We will then present standard results from Fourier and wavelet analysis; and we introduce the power spectrum of a stationary stochastic process, explaining how it can be estimated using wavelets. Finally, we will review spectral methods for steady-state simulation output analysis.

2.1. Nonspectral Methods For Steady-State Output Analysis

A number of methods have already been developed for steady-state simulation output analysis. The following subsections provide a brief overview of several nonspectral methods.

2.1.1 *Replication/Deletion*

Suppose we want to estimate the steady-state mean $\mathrm{E}[X] = \mu_X$ of the output process $X_1, \ldots, X_n$. Using the replication/deletion approach [39], we make k replications of the simulation, each of length n, and delete the first l observations from each replication. Let $X_{j,i}$ denote the ith observation from the jth replication and let

$$\overline{X}_j = \frac{\sum_{i=l+1}^{n} X_{j,i}}{n-l}$$

denote the truncated sample mean of the jth replication, $j = 1, \ldots, k$. Note that the initial bias problem is addressed by using only those observations $X_{j,l+1}, \ldots, X_{j,n}$ that correspond to a (nearly) steady-state situation. By deleting those observations at the beginning of the simulation run, we eliminate the initial bias due to the choice of initial system conditions [39]. Also note that the $\overline{X}_j$'s are independent and identically distributed random variables with $\mathrm{E}[\overline{X}_j] \doteq \mu_X$; and if n is sufficiently large, the Central Limit Theorem implies $\overline{X}_j \mathrel{\dot\sim} N(\mu_X, \mathrm{Var}(\overline{X}_j))$, where

the symbol $\dot{\sim}$ means "is approximately distributed as" and [39]

$$\begin{aligned}\mathrm{Var}(\overline{X}_j) &= \frac{1}{n-l}\left\{\gamma_X(0) + 2\sum_{u=0}^{n-l}\left(1-\frac{u}{n-l}\right)\gamma_X(u)\right\} \\ &\approx \frac{\gamma_X}{n-l} \quad \text{as} \quad n-l \to \infty.\end{aligned}$$

Taking the grand mean computed over all truncated replications,

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

as a point estimate for μ_X and

$$\widehat{\mathrm{Var}}[\overline{\overline{X}}] = \frac{\sum_{j=1}^{k}\left[\overline{X}_j - \overline{\overline{X}}\right]^2}{k(k-1)}$$

as an estimate for the variance of $\overline{\overline{X}}$, for $0 < \beta < 1$, we can construct an asymptotically valid $100(1-\beta)\%$ confidence interval for μ_X in the following way:

$$\overline{\overline{X}} \pm t_{1-\beta/2,k-1}\sqrt{\widehat{\mathrm{Var}}[\overline{\overline{X}}]}.$$

In order to determine the length of the warm-up period l, Welch [68] proposes a graphical method that requires making k' initial pilot runs, each of length n'. Let

$$\widehat{\mu}_i = \frac{1}{k'}\sum_{j=1}^{k'} X_{j,i}$$

be an estimate of the transient mean function $\mu_i = \mathrm{E}[X_i]$ for the ith observation X_i, $i = 1, \ldots, n'$ and let $\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$ denote the set of integers. Compute the corresponding moving average of w observations centered at the ith observation

$$\widehat{\mu}_i(w) = \begin{cases} \frac{1}{2i-1}\displaystyle\sum_{l=-(i-1)}^{i-1}\widehat{\mu}_{i+l}, & i = 1, \ldots, \lfloor\frac{w-1}{2}\rfloor \\ \frac{1}{2\lfloor\frac{w-1}{2}\rfloor+1}\displaystyle\sum_{l=-\lfloor\frac{w-1}{2}\rfloor}^{\lfloor\frac{w-1}{2}\rfloor}\widehat{\mu}_{i+l}, & i = \lfloor\frac{w-1}{2}\rfloor+1, \ldots, n-\lfloor\frac{w-1}{2}\rfloor \\ \frac{1}{2(n-i)+1}\displaystyle\sum_{l=-(n-i)}^{n-i}\widehat{\mu}_{i+l}, & i = n-\lfloor\frac{w-1}{2}\rfloor+1, \ldots, n, \end{cases} \tag{2.1}$$

where $w \in \mathbb{Z}, w > 0$ is the window size for the moving average. Choose l to be the value of i beyond which the plot of the $\{\widehat{\mu}_i(w) : i = 1, 2, \ldots\}$ appears to converge.

The popularity of the replication/deletion approach is due in part to its simplicity. The method is easy to understand and implement, making it especially attractive to those who do not have the statistical background needed to implement some of the other techniques for steady-state output analysis. Furthermore, we are able to achieve truly independent observations $\{\overline{X}_j : 1 \leq j \leq k\}$. The replication/deletion method is computationally inefficient, however, in the sense that it requires the deletion of a total of lk observations. In addition, the method can potentially require an excessive sample size n to achieve approximate normality of the replicate means, $\{\overline{X}_j\}$. Finally, there is no definitive method for determining how large l should be. Since Welch's graphical procedure is only approximate, it is possible that some observations close to the deletion point l may still contain initialization bias, thereby resulting in a biased estimate of μ_X.

2.1.2 *Regenerative Method*

The main premise behind the regenerative method is the fact that many stochastic systems "start afresh probabilistically" at certain points in time, called regeneration times, resulting in independent and identically distributed blocks of data [14]. The sequence of observations between two regeneration times is referred to as a regeneration cycle. To illustrate the idea of regeneration cycles, consider the system sojourn times for an $M/M/1$ queue with traffic intensity less than one. We can define a regeneration cycle for this system to be a busy period followed by an idle period. Each new cycle begins when a customer arrives to the system and finds the server idle, since in the associated customer-arrival event, all the stochastic processes driving the simulation must be *independently* resampled when that event occurs—namely, (a) the delay to the next customer arrival must be randomly sampled from the exponential distribution of interarrival times; and (b) the delay in service for the current customer must be randomly sampled from the exponential distribution of service times. Thus at each point in the sequence of customer sojourn times where the current arrival finds the server idle, we know that the current and future customer sojourn times are independent of the customer sojourn times up to that point. The same probabilistic structure as in previous cycles is followed, thereby resulting in random-length blocks of independent and identically distributed observations. One class of problems for which it is easy to identify regeneration times is the class of generalized semi-Markov processes (GSMPs) with single states. For a discussion on how to apply the regenerative method to these types of systems, see Henderson and Glynn [29].

Let $T_1, T_2, \ldots$ denote the random sequence of regeneration times for the simulation-generated process $\{X_i\}$, and let $\{X_k : T_j \leq k < T_{j+1}\}$ denote the jth regeneration cycle. As an example, for the system sojourn time process $\{X_i : i \geq 1\}$ in the $M/M/1$ queue, T_1 is the customer

number of the first customer with no wait and T_j is the customer number of the jth customer with no wait. We also define the "length" of the jth regenerative cycle j in the following way,

$$\tau_j \equiv T_{j+1} - T_j, \quad \text{for} \quad j \geq 1.$$

Thus in a regenerative simulation of $M/M/1$ sojourn times, τ_j is the number of customers served in the jth regenerative cycle. In order to apply the regenerative method to a process, the following conditions must hold: $\Pr(T_1 < \infty) = 1; \Pr(0 < \tau_j < \infty) = 1$ for $j \geq 1$; the initial cycle $\{X_k : 0 \leq k < T_1\}$ is independent of $\{X_k : k \geq T_1\}$; and the jth cycle $\{X_k : T_j \leq k < T_{j+1}\}$ is an independent, identically distributed replicate of the first cycle $\{X_k : T_1 \leq k < T_2\}$ [31].

Defining

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

as the accumulated "reward" on the jth regenerative cycle, we see that the sequence $\{(Y_j, \tau_j), j \geq 1\}$ consists of i.i.d. random vectors. For example in a regenerative simulation of $M/M/1$ sojourn times, Y_j represents the total customer-hours ("manhours") in the system that are accumulated over the jth cycle. If $\mathrm{E}[|Y_1|] < \infty$ and $\mathrm{E}[\tau_1] < \infty$ then the steady-state mean of the process $\{X_i : i = 1, \ldots, n\}$ is

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

by the renewal-reward theorem [50]. To construct an asymptotically valid confidence interval for μ_X using a regenerative approach, we simulate m regeneration cycles using one long run of length $n = n(m)$. Taking

$$\overline{Y} = \frac{1}{m} \sum_{j=1}^{m} Y_j$$

and

$$\overline{\tau} = \frac{1}{m} \sum_{j=1}^{m} \tau_j,$$

we see that a point estimate for μ_X is

$$\widehat{\mu}_X = \frac{\overline{Y}}{\overline{\tau}}. \tag{2.2}$$

Now, to obtain an estimate of the variance of $\widehat{\mu}_X$, we define

$$V_j = Y_j - \mu_X \tau_j, \quad \text{for} \quad j = 1, \ldots, m,$$

and

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

Notice that $\mathrm{E}[V_j] = 0$ and

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

Since the V_j's are functions of i.i.d. random variables, they are also i.i.d.; and if $0 < \sigma_V^2 < \infty$, then the Central Limit Theorem holds, implying

$$\frac{\sqrt{m}\,\overline{V}}{\sigma_V} \xrightarrow{D} N(0,1) \quad \text{as} \quad m \to \infty, \tag{2.3}$$

where $\xrightarrow{D}$ denotes convergence in distribution. To estimate σ_V^2, we compute the following sample estimates,

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

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

and

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

Using these sample estimates and equation (2.2), we obtain a strongly consistent estimate for σ_V^2,

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

Now, substituting into equation (2.3) and applying Slutsky's theorem [5], we have

$$\frac{\sqrt{m}(\widehat{\mu}_X - \mu_X)}{\widehat{\sigma}_V / \overline{\tau}} \xrightarrow{D} N(0,1) \quad \text{as} \quad m \to \infty;$$

and for m sufficiently large, an asymptotically valid $100(1-\beta)\%$ confidence interval for the steady-state mean μ_X is

$$\widehat{\mu}_X \pm z_{1-\beta/2} \frac{\widehat{\sigma}_V}{\sqrt{m}\,\overline{\tau}},$$

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

What makes the regenerative method appealing in practice is that it is relatively simple to implement, the initialization bias problem is not an issue, and the random vectors $\{(Y_j, \tau_j) : j = 1, \ldots, m\}$ are truly i.i.d. Like the replication/deletion method, however, a regenerative approach could possibly require an enormous run length $n(m)$ in order to obtain a sufficiently large number of regenerative cycles m (generally we need $m \geq 100$ [30]). Furthermore, there

are a number of simulation problems of interest where the expected time between successive regeneration times is extremely long and therefore do not admit a regenerative analysis–type approach. The $M/M/1$ queue with traffic intensity close to one is one example of such a problem.

2.1.3 *Method of Batch Means*

Assume the output process $X_1, X_2, \ldots$, is stationary so that the joint distribution of the X_i's is insensitive to time shifts and $\mathrm{E}[X_i] = \mu_X$. Also assume the process is weakly dependent in the sense that X_i's far from each other in the sequence are almost independent. That is, the lag-l correlation $\rho_X(l) \to 0$ as $|l| \to \infty$. In the classical nonoverlapping batch means (NOBM) method, we make one long simulation run of length n and divide the observations $X_1, \ldots, X_n$ into k adjacent nonoverlapping batches, each of size m (so that $n = km$).

Let

$$\overline{X}_j = \frac{1}{m} \sum_{i=m(j-1)+1}^{mj} X_i \quad \text{for} \quad j = 1, \ldots, k$$

denote the sample mean of the jth batch and let the grand mean

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

be a point estimate of the steady-state mean μ_X. If the batch size m is chosen large enough, then we can treat the batch means $\{\overline{X}_j : 1 \le j \le k\}$ as approximately independent and normally distributed random variables with $\mathrm{E}[\overline{X}_j] = \mu_X$ and

$$\begin{aligned} \mathrm{Var}[\overline{X}_j] &= \frac{1}{m} \left[\gamma_X(0) + 2 \sum_{l=1}^{m-1} \left(1 - \frac{l}{m}\right) \gamma_X(l) \right] \\ &\approx \frac{\gamma_X}{m} \quad \text{as} \quad m \to \infty, \end{aligned}$$

where γ_X is the SSVC of the process and $\gamma_X(l)$ is as defined in (1.6). The sample variance of the k batch means,

$$S^2 = \frac{1}{k-1} \sum_{j=1}^{k} \left(\overline{X}_j - \overline{\overline{X}} \right)^2, \tag{2.4}$$

is used as an estimate for $\mathrm{Var}[\overline{X}_j]$. With $m \to \infty$ and k fixed so that $n \to \infty$, we can construct an asymptotically valid $100(1-\beta)\%$ confidence interval for the steady-state mean μ_X in the

following way,

$$\overline{\overline{X}} \pm t_{1-\beta/2,k-1}\frac{S}{\sqrt{k}}. \tag{2.5}$$

The main difficulty with any NOBM procedure is the determination of an adequate batch size m so that the batch means $\{\overline{X}_j\}$ are approximately uncorrelated and normal. In Steiger and Wilson [57], an improvement to the classical NOBM confidence interval (2.5) is developed. Their method is based on the observation that the batch means $\{\overline{X}_j\}$ often achieve approximate joint multivariate normality at a batch size m that is substantially smaller than the batch size required to ensure approximate independence of the batch means. Their method for steady-state simulation output analysis, called ASAP (Automated Simulation Analysis Procedure), addresses both the initial bias problem and the correlation problem. Furthermore, ASAP is equipped with a scheme for controlling the simulation run length in order to deliver a confidence interval satisfying a user-specified precision requirement. The precision requirement can either be specified as a maximum absolute half-length h^* or a maximum fraction r^* of the magnitude of the final estimate of the mean.

Once the user has supplied a simulation-generated sequence $\{X_i : i = 1, \ldots, n\}$, a confidence coefficient $1-\beta$ specifying that the desired confidence interval coverage probability is $1-\beta$, and an absolute or relative precision requirement, ASAP delivers a nominal $100(1-\beta)\%$ confidence interval for μ_X that satisfies the specified absolute or relative precision requirement, provided no additional simulation is required; or otherwise, ASAP requires a new, larger sample size n to be supplied to the algorithm. Iteration i of ASAP executes the following steps:

[1] Divide the total simulation output $\{X_j : j = 1, \ldots, n_i\}$ required so far into k_i batches of size m_i (where $k_1 = 96$ and $m_1 = 16$ so that $n_1 = k_1 m_1 = 1,536$ by default) and compute the corresponding batch means $\{\overline{X}_1, \ldots, \overline{X}_{k_i}\}$.

[2] Discard the first two batches (to address the warm-up problem).

[3] If the remaining $k_i' = k_i - 2$ batch means have already passed the independence test, then compute the updated confidence interval (2.5)–(2.6) and go to step **[6]**; otherwise, test the remaining $k_i' = k_i - 2$ batch means for independence using von Neumann's ratio of the sample mean square successive difference to the sample variance [20], using the level of significance $\alpha_{\text{ind}} = 0.2$ for the two-sided independence test. If the batch means $\{\overline{X}_3, \ldots, \overline{X}_{k_i}\}$ pass the test for independence, then the classical NOBM confidence interval (2.5) is constructed with midpoint $\overline{\overline{X}}$ and half-length

$$H = t_{1-\beta/2,k_i'-1}\frac{S}{\sqrt{k_i'}}, \tag{2.6}$$

where S is as defined in (2.4); and control passes to step **[6]**. If the batch means $\{\overline{X}_3, \ldots, \overline{X}_{k_i}\}$ fail the test for independence, then go to step **[4]**.

[4] If the remaining batch means have already passed the multivariate normality test, then compute the updated confidence interval (2.8)–(2.9) and go to step **[6]**; otherwise, the batch means are tested for multivariate normality. Sixteen four-dimensional vectors are constructed by inserting a *spacer* consisting of two adjacent batch means after each group of four adjacent batch means to obtain approximately independent 4×1 vectors $\{\boldsymbol{y}_l = [\overline{X}_{6l-3}, \overline{X}_{6l-2}, \overline{X}_{6l-1}, \overline{X}_{6l}] : l = 1, \ldots, 16\}$. The Shapiro-Wilk test for multivariate normality [40] is applied to the data set $\{\boldsymbol{y}_l : l = 1, \ldots, 16\}$ with level of significance $\alpha_{\text{mvn}} = 0.10$. If the batch means fail the test for multivariate normality, then

$$\begin{aligned} k_{i+1} &\leftarrow 96, k'_{i+1} \leftarrow k_{i+1} - 2, \\ m_{i+1} &\leftarrow \lfloor \sqrt{2} m_i \rfloor, \\ n_{i+1} &\leftarrow k_{i+1} m_{i+1}, \\ i &\leftarrow i + 1; \end{aligned}$$

and control returns to step **[1]**.

[5] If the 4×1 vectors of batch means $\{\boldsymbol{y}_l : l = 1, \ldots, 16\}$ pass the test for normality, then an adjustment is made to the classical NOBM confidence interval (2.5) by taking into account the deviation of the distribution of the NOBM t-ratio,

$$t = \frac{\overline{\overline{X}} - \mu_X}{\sqrt{S^2 / k'_i}} \tag{2.7}$$

from the usual Student's t-distribution with $k'_i - 1$ degrees of freedom. The adjustment to the critical value $t_{1-\beta/2, k'_i - 1}$ of Student's t-distribution is based on an inverted Cornish-Fisher expansion for (2.7) that involves the first four cumulants of (2.7). This type of adjustment will take into account the dependence between the batch means. An estimate of the variance of the grand mean, $\overline{\overline{X}}$, is then computed by first fitting an autoregressive-moving average (ARMA) time series model [8] to the sequence of k'_i batch means $\{\overline{X}_3, \ldots, \overline{X}_{k_i}\}$. The estimator $\widehat{\text{Var}}[\overline{\overline{X}}]$ is then derived from the maximum likelihood estimators of the parameters of the fitted ARMA process. A confidence interval of the form

$$\overline{\overline{X}} \pm H, \tag{2.8}$$

where

$$H = \left[\left(1 + \frac{\widehat{\kappa}_2 - 1}{2} - \frac{\widehat{\kappa}_4}{8}\right) z_{1-\beta/2} + \frac{\widehat{\kappa}_4}{24}\ z^3_{1-\beta/2}\right] \sqrt{\frac{\widehat{\mathrm{Var}}(\overline{\overline{X}})}{k_i'}}, \tag{2.9}$$

is computed. In (2.9), $\widehat{\kappa}_2$ and $\widehat{\kappa}_4$ are estimates of the second and fourth cumulants, respectively, of the NOBM t-ratio (2.7).

[6] Determine if the current confidence interval satisfies the user's precision requirement,

$$H \leq H^*, \tag{2.10}$$

where

$$H^* = \begin{cases} \infty, & \text{for no user-specified precision level;} \\ r^*|\overline{\overline{X}}|, & \text{for a user-specified relative precision level } \ r^*; \\ h^*, & \text{for user-specified absolute precision level } \ h^*. \end{cases}$$

If (2.10) is satisfied, then ASAP terminates, returning the current confidence interval. If the precision requirement (2.10) is not satisfied, then ASAP computes the additional number of batches k_i^+ with batch size m_i that are required to satisfy (2.10),

$$k_i^+ = \left\lceil \left(\frac{H}{H^*}\right)^2 k_i' \right\rceil - k_i'.$$

If for iteration $i+1$ the total number of batches $k_i + k_i^+ > 1502$, then a new batch size is calculated,

$$m_{i+1} = \left\lfloor \sqrt{2} m_i \right\rfloor$$

and the batch count remains fixed so that

$$k_{i+1} \leftarrow k_i.$$

If $k_i + k_i^+ \leq 1502$, then a new batch count is computed,

$$k_{i+1} = k_i + k_i^+$$

and the batch size remains fixed,

$$m_{i+1} \leftarrow m_i.$$

The iteration index is updated,

$$i \leftarrow i + 1,$$

and control passes to step **[1]**.

Notice that in general, iteration $i+1$ of ASAP begins by deleting the first two batches of size m_{i+1} from the data set $\{X_i : i = 1, \ldots, n_{i+1}\}$, where $n_{i+1} = m_{i+1}k_{i+1}$. Batch means are computed for the remaining $k'_{i+1} = k_{i+1} - 2$ batches. Iterations of ASAP that are executed to fulfill the precision requirement (3.55) do not involve retesting the batch means for independence or normality. If a correlation-adjusted confidence interval was constructed on iteration i, then an updated ARMA model is fit to the current set of k'_{i+1} batch means. A new estimate of $\mathrm{Var}[\overline{\overline{X}}]$ is computed from the ARMA model and the updated confidence interval (2.8) is computed. If the classical NOBM confidence iterval (2.5) was used on iteration i, then (2.5) is recomputed using the current set of batch means. If the precision requirement (3.55) is satisfied on iteration $i+1$, then ASAP terminates. If the precision requirement is not satisfied, then ASAP computes a new total number of batches k_{i+2} and a new batch size m_{i+2} as described above and procedes to iteration $i+2$.

One advantage of ASAP is it is completely automated and requires no user intervention. Experimental results indicate the method performs well for a variety of problems. However, there are certain stochastic systems for which significant departures from normality of the batch means are observed even for batch sizes sufficiently large to ensure negligible dependence between the batch means. That is, if the batch means pass the test for independence, then the classical confidence interval (2.5) is constructed, even though it is possible the batch means may exhibit significant departures from normality. In order to avoid the anomalous behavior of ASAP in such cases, an improved variant of ASAP, ASAP2, was developed [58]. Like ASAP, ASAP2 is a sequential procedure for steady-state simulation output analysis where a user-specified precision requirement is used to control the run-length of the simulation. The main difference between ASAP and ASAP2, however, is that ASAP2 does not require a test for independence. Step **[3]** in the above algorithm for ASAP is eliminated in ASAP2, and the batch means are only tested for multivariate normality. Upon acceptance of the hypothesis of joint multivariate normality of the batch means, the correlation-adjusted confidence interval (2.8) is constructed. Another difference between ASAP2 and ASAP is the first four batches are removed in ASAP2 to account for the warm-up period and the spacer used to separate the batch means used in the multivariate normality test consists of four batches, as opposed to the two batches used in ASAP. Furthermore, instead of having a fixed level of significance α_{mvn} in the multivariate normality test like the ASAP algorthim has, ASAP2 reduces the level of significance on iteration i of the normality test after each failed test according to,

$$\alpha_{\mathrm{mvn}}(i) = \alpha_{\mathrm{mvn}}(1)\exp\left[-\tau(i-1)^2\right] \quad \text{for} \quad i = 1, 2, \ldots, \tag{2.11}$$

where $\alpha_{\mathrm{mvn}}(1) = 0.10$ and $\tau = 0.18421$. This scheme for reducing the level of significance on each iteration of the normality test is designed so that ASAP2 avoids the excessive variability in the final sample size and confidence-interval half-length that is sometimes observed with ASAP.

An extensive performance evaluation indicates that ASAP2 frequently outperforms ASAP in terms of confidence interval coverage, the mean and variance of the half-length of its confidence intervals, and its total sample size. It should be recognized, however, that ASAP2 was designed for use with a user-specified precision requirement; and in the absence of a precision requirement, ASAP2-generated confidence intervals can be highly variable in their half-lengths.

2.1.4 *Standardized Time Series Analysis*

Let $X_{j,i}$ denote the ith observation from the jth replication of a steady-state simulation, $1 \leq i \leq m$; let

$$\overline{X}_j = \frac{1}{m} \sum_{i=1}^{m} X_{j,i}$$

be the sample mean for the jth replication, $j = 1, \ldots, k$ (or alternatively, $X_{j,i}$ may be interpreted as the ith observation within the jth batch of size m for a single, prolonged simulation run so that $\overline{X}_j$ is the jth batch mean for batches of size m in a run of length $n = km$); and finally let

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

be the grand mean over all replications (or batches). We will assume that the output process from the jth replication (or batch) $\{X_{j,i} : 1 \leq i \leq m\}$ is stationary and $X_{j,i}$ and $X_{j,i+l}$ are asymptotically independent as $l \to \infty$ in the sense of ϕ-mixing [6].

Define the standardized process

$$T_j(t) = \frac{1}{\sqrt{m\gamma_X}} \sum_{i=1}^{\lfloor mt \rfloor} \left(\overline{X}_j - X_{j,i} \right) \quad \text{for} \;\; 0 \leq t \leq 1, \tag{2.12}$$

where $\lfloor \cdot \rfloor$ denotes the greatest integer function and γ_X is the SSVC. It has been proved that as $m \to \infty$, the function $T_j(\cdot)$ converges in distribution to a Brownian bridge process [22]. Furthermore, we define

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

and

$$A = \frac{12}{m^3 - m} \sum_{j=1}^{k} A_j^2 + m \sum_{j=1}^{k} \left(\overline{X}_j - \overline{\overline{X}} \right)^2. \tag{2.13}$$

Since $T_j(\cdot)$ converges to a Brownian bridge process as $m \to \infty$, it can be shown that:

$$\frac{12}{m^3 - m} \sum_{j=1}^{k} A_j^2 \xrightarrow{D} \gamma_X \chi^2(k) \quad \text{as} \quad m \to \infty, \tag{2.14}$$

where $\chi^2(k)$ denotes a chi-square random variable with k degrees of freedom;

$$m \sum_{j=1}^{k} \left(\overline{X}_j - \overline{\overline{X}} \right)^2 \xrightarrow{D} \gamma_X \chi^2(k-1) \quad \text{as} \quad m \to \infty; \tag{2.15}$$

and finally that (2.14) and (2.15) are independent. These results imply that

$$\frac{A}{\gamma_X (2k-1)} \xrightarrow{D} \frac{\chi^2(2k-1)}{2k-1} \quad \text{as} \quad m \to \infty.$$

If m is sufficiently large so that the sample means $\{\overline{X}_j : 1 \leq j \leq k\}$ are (approximately) independent, identically distributed normal random variables, then

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

are stochastically independent.

We are now in a position to construct a Student t-ratio in the following way. Since

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

we have

$$\frac{\frac{\overline{\overline{X}} - \mu_X}{\sqrt{\frac{\gamma_X}{km}}}}{\sqrt{\frac{A}{\gamma_X (2k-1)}}} = \frac{\overline{\overline{X}} - \mu_X}{\sqrt{\frac{A}{km(2k-1)}}} \sim t_{2k-1} \quad \text{as} \quad m \to \infty, \tag{2.16}$$

where t_{2k-1} denotes a random variable having Student's t-distribution with $2k-1$ degrees of freedom. This implies that for $0 < \beta < 1$, we have

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

Therefore, a $100(1-\beta)\%$ asymptotically valid confidence interval for the steady-state mean μ_X

is

$$\overline{\overline{X}} \pm t_{1-\beta/2,2k-1}\sqrt{\frac{A}{km(2k-1)}}.$$

Notice that in equation (2.16) in the above analysis, the SSVC γ_X cancels out, thereby eliminating the problem of having to estimate it. This is one of the appealing aspects of a standardized time series approach. There are several disadvantages to this method, however. First, the entire method is based on the fact that as $m \to \infty$, the standardized process $T_j(\cdot)$ converges to a Brownian bridge process. It can take extremely large values of m for such a convergence to take place, however. Second, the method is complicated, making it more difficult to implement in practice than other methods for steady-state simulation output analysis.

2.2. Fourier Methods

Fourier analysis plays a key role in the development of spectral methods for steady-state output analysis, as well as wavelet theory. Therefore, we will first introduce some basic Fourier results before discussing spectral methods for simulation output analysis. Let $\mathbb{R}$ denote the real numbers. Any periodic function $f(t), t \in \mathbb{R}$, with period 2π satisfying

$$\|f\|_2^2 \equiv \int_0^{2\pi} |f(t)|^2 dt < \infty \tag{2.17}$$

is said to belong to the space $L_2[-\pi, \pi]$ of square-integrable, real-valued functions on the interval $[-\pi, \pi]$; and any $f(t)$ staisfying (2.17) can be expressed as a linear combination of sine and cosine functions in the following way,

$$f(t) = \frac{1}{2}a_0 + \sum_{n\in\mathbb{N}} a_n \cos(nt) + \sum_{n\in\mathbb{N}} b_n \sin(nt) \quad \text{almost everywhere,} \tag{2.18}$$

where $\mathbb{N} = \{1, 2, \ldots\}$ denotes the set of natural numbers [10]. Equation (2.18) is called the *Fourier series* for $f(t)$; and the *Fourier coefficients* a_n and b_n are evaluated as follows,

$$\begin{aligned} a_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(u)\, \cos(nu)\, du, \quad n \in \mathbb{N} \cup \{0\}, \\ b_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} f(u)\, \sin(nu)\, du, \quad n \in \mathbb{N}. \end{aligned}$$

Equivalently, the Fourier series for $f(t)$ can be written

$$f(t) = \sum_{n\in\mathbb{Z}} \hat{f}(n)\, \exp(nt\sqrt{-1}) \quad \text{for almost all } t,$$

where the Fourier coefficients $\hat{f}(n)$ are computed as follows,

$$\hat{f}(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(u) \exp(-nu\sqrt{-1})\ du, \quad n \in \mathbb{Z}$$

[3]. The term corresponding to $n = 1$ in (2.18) is called the fundamental, and represents a cosine/sine wave whose period is exactly that of the function $f(t)$. The next term, corresponding to $n = 2$, is called the first harmonic and represents a cosine/sine wave whose period is exactly half that of $f(t)$. The terms corresponding to $n = 3, 4, 5, \ldots$ are called the second, third, fourth, etc., harmonics [47].

Suppose $f(t), t \in \mathbb{R}$, is now any function (that is not necessarily periodic) satisfying

$$\|f\|_2^2 \equiv \int_{-\infty}^{\infty} |f(t)|^2 dt < \infty,$$

so that $f \in L_2(\mathbb{R})$, the space of square-integrable, real-valued functions on the real line $\mathbb{R}$. We define the *Fourier transform* of $f(t)$ as follows,

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) \exp(-\omega t\sqrt{-1})\ dt \ \text{ for all real } \omega. \tag{2.19}$$

We can recover the original function $f(t)$ by using the inverse Fourier transform,

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) \exp(\omega t\sqrt{-1})\ d\omega \ \text{ for almost all real } t. \tag{2.20}$$

Alternatively, (2.19) and (2.20) may be written as

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) \ \{\cos(\omega t) - \sqrt{-1}\sin(\omega t)\}\ dt$$

and

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) \ \{\cos(\omega t) + \sqrt{-1}\sin(\omega t)\}\ d\omega, \tag{2.21}$$

respectively. If $f(t) = f(-t)$ for all $t \in \mathbb{R}$ (that is, f is an even function), then from (2.21) we get the cosine transform,

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) \cos(\omega t)\ d\omega \ \text{ for almost all real } t. \tag{2.22}$$

Similarly, if $f(t) = -f(-t)$ for all $t \in \mathbb{R}$ (that is, f is an odd function), we have the sine

transform,

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) \sin(\omega t)\ d\omega \quad \text{for almost all } t \in \mathbb{R} \tag{2.23}$$

[47].

The Fourier transform and its inverse are used to reproduce a function $f(t)$ as a sum of the complex exponentials, $\exp(nt\sqrt{-1}), n \in \mathbb{Z}$ [35]. However, if $f(t)$ has a sharp spike, then reproducing $f(t)$ by summing a series of complex exponentials is inefficient. In particular, the narrower the spike, the more high-frequency complex exponentials must be used in the construction of $f(t)$ using equation (2.20). Since these complex exponentials repeat the same oscillatory behavior all along $\mathbb{R}$, more complex exponentials will be required in order to cancel the previous ones before and after the spike occurs. In this way, the Fourier transform is not an efficient method for reconstructing functions with local behavior, such as a sharp spike.

A more efficient method for estimating a function $f(t)$ with local behavior is the windowed Fourier transform (WFT). The goal of the WFT method is to localize a function f in the time domain and the frequency domain simultaneously. Only the values $f(u)$ for $u \leq t$ can be used to compute the frequency distribution of f at time t. Furthermore, there is a time lag $T > 0$ such that only the values $f(u)$ for $t - T \leq u \leq t$ can influence the frequency at time t. This information can be used to construct a weight function, or window, that will be used to localize a function in time. Let $g(u)$ be a real-valued function that vanishes outside the interval $-T \leq u \leq 0$. For every $t \in \mathbb{R}$, define

$$f_t(u) \equiv g(u - t)\ f(u),$$

where f_t is a localized version of f that depends only on the values $f(u)$ for $t - T \leq u \leq t$. We define the windowed Fourier transform of f as the Fourier transform of f_t,

$$\begin{aligned} \hat{f}_t(\omega) &= \int_{-\infty}^{\infty} f_t(u) \exp(-\omega u\sqrt{-1})\ du \\ &= \int_{-\infty}^{\infty} g(u - t)\ f(u) \exp(-\omega u\sqrt{-1})\ du. \end{aligned} \tag{2.24}$$

To reconstruct f from $\hat{f}_t$, we use the following reconstruction formula,

$$f(u) = \frac{1}{2\pi \|g\|_2^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(\omega u\sqrt{-1}) g(u - t)\ \hat{f}_t(\omega)\ d\omega\ dt, \tag{2.25}$$

which holds for almost all $u \in \mathbb{R}$, provided $f, g \in L_2(\mathbb{R})$ and $\|g\|_2^2 > 0$ [35].

While the WFT method is more efficient in reproducing functions with local behavior than the Fourier transform method, the WFT can be inefficient when the local behavior occurs in

a much shorter time interval than the truncation point T for the window. A similar problem occurs when very long and smooth features of the function are to be reproduced by the WFT (that is, features of the function f characterized by time intervals much greater than the truncation point T). In summary, the WFT introduces a *scale* into the analysis and reconstruction of functions equal to the magnitude of the truncation point T. The reproduction of features of f much shorter or much longer than this scale will require the combination of many complex exponentials.

2.3. Wavelet Theory

The goal now becomes to find a method for analyzing and reproducing functions with local behavior that does not require a fixed scale. Wavelet analysis is such a method. Consider the wavelet function $\psi(t) \in L_2(\mathbb{R})$. Using the terminology of the WFT method, we can think of $\psi(t)$ as a window function. However, to avoid the problems associated with using a fixed scale, we will use all possible scalings of $\psi(t)$. In particular, let $\psi_{a,b}(t), a \in \mathbb{R}\backslash\{0\}, b \in \mathbb{R}$, be a family of functions created via translations and rescalings (or dilations) of the wavelet function $\psi(t)$,

$$\psi_{a,b}(t) = |a|^{-p} \ \psi\left(\frac{t-b}{a}\right) \quad \text{for all real } t, \tag{2.26}$$

where $p \geq 0$ is fixed and a is the scaling parameter. For general $a > 1$, $\psi_{a,b}(t)$ is obtained by stretching $\psi(t)$ by the factor a in the horizontal direction. Similarly, if $0 < a < 1$, then $\psi_{a,b}(t)$ is obtained by compressing $\psi(t)$ in the horizontal direction. If $a < 0$, then $\psi_{a,b}(t)$ is obtained by reflecting $\psi(t)$ with respect to the vertical axis.

The wavelet function $\psi(t)$ is assumed to satisfy the following admissibility condition,

$$C_\psi = \int_{-\infty}^{\infty} \frac{|\widehat{\psi}(\omega)|^2}{|\omega|} \ d\omega \quad < \infty, \tag{2.27}$$

where $\widehat{\psi}(\omega)$ is the Fourier transformation of $\psi(t)$. The admissibility condition (2.27) implies that $\widehat{\psi}(\omega) \to 0$ as $\omega \to 0$; and it follows from Lebesgue's dominated convergence theorem that

$$\widehat{\psi}(0) = 0 = \int_{-\infty}^{\infty} \psi(t) \ dt.$$

Furthermore, it is assumed that $\psi(t)$ has unit norm,

$$\|\psi\|_2^2 = \int_{-\infty}^{\infty} \psi^2(t) \ dt = 1.$$

The continuous wavelet transform (CWT) of a function $f(t)$ is defined as

$$f_{\mathrm{CWT}}(a,b) = \int_{-\infty}^{\infty} f(t)\ \psi_{a,b}(t)\ dt. \tag{2.28}$$

Note that the scale parameter a and the translation parameter b vary continuously over $\mathbb{R}\backslash\{0\}\times\mathbb{R}$. When the admissibility condition (2.27) is satisfied, it is possible to find the inverse continuous wavelet transformation using the resolution of identity,

$$f(t) = \frac{1}{C_\psi}\int_{-\infty}^{\infty}\left[\int_{-\infty}^{\infty} f_{\mathrm{CWT}}(a,b)\ \psi_{a,b}(t)\ |a|^{2p-3}\ da\right] db \ \text{ for almost all } t\in\mathbb{R},$$

provided $f\in L_2(\mathbb{R})$ and $C_\psi > 0$ [62].

Since all scales $a\neq 0$ are used in the CWT (2.28), the reconstruction is highly redundant. Ideally, the entire frequency spectrum should be covered by discrete scalings of $\psi(t)$ [35]. Let

$$a = 2^{-j}, b = k2^{-j}, j,k\in\mathbb{Z}. \tag{2.29}$$

If we set $p = 1/2$ and substitute the values of a and b given in (2.29) into (2.26), we have a family of functions $\{\psi_{j,k}(t) : j,k\in\mathbb{Z}\}$ created via discrete translations and rescalings of the wavelet function $\psi(t)$, where

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k),\ \ j,k\in\mathbb{Z}.$$

Using the above values of a and b, we have a discrete transformation that is invertible and furthermore, all information about the decomposed function will be preserved. In the next section, discrete wavelet transformations for the values of a and b given in (2.29) are described.

2.4. Multiresolution Analysis

Multiresolution analysis was formulated in 1986 by S. Mallat and Y. Meyer [41]. Multiresolution analysis is used for performing discrete wavelet analysis, as well as the construction of wavelet functions. A multiresolution analysis is a sequence of closed subspaces $\{V_j : j\in\mathbb{Z}\}$ in $L_2(\mathbb{R})$. This sequence of "successive approximation spaces" satisfies the following conditions:

(i) $\{V_j : j\in\mathbb{Z}\}$ is an increasing sequence of subspaces so that

$$\cdots\subset V_{-2}\subset V_{-1}\subset V_0\subset V_1\subset V_2\cdots; \tag{2.30}$$

(ii) the subspace defined by the limit of $\{V_j : j\in\mathbb{Z}\}$ is dense in $L_2(\mathbb{R})$ so that the closure of

$\bigcup_{j\in\mathbb{Z}} V_j$,

$$\mathrm{cl}\left(\bigcup_{j\in\mathbb{Z}} V_j\right) = L_2(\mathbb{R}); \tag{2.31}$$

and

(iii) the null function (that is, the function $g_0(x) = 0$ for almost all $x \in \mathbb{R}$) is the only common element of all the $\{V_j\}$,

$$\bigcap_{j\in\mathbb{Z}} V_j = \{0\} \tag{2.32}$$

[41]. There are several additional requirements the sequence of subspaces $\{V_j\}$ must satisfy to form a multiresolution analysis. First, the V-spaces must be self-similar,

$$f(2^j t) \in V_j \ \text{ if and only if } \ f(t) \in V_0 \ \text{ for all } j \in \mathbb{Z}. \tag{2.33}$$

That is, all the V-spaces are scaled versions of the central space V_0. Second, V_0 must be invariant under integer translations,

$$f(t) \in V_0 \ \text{ implies } \ f(t-k) \in V_0 \ \text{ for all } \ k \in \mathbb{Z}. \tag{2.34}$$

Finally, there must exist a *scaling function* $\phi(t) \in V_0$ so that

$$\{\phi_{0,k}(t) : k \in \mathbb{Z}\} \tag{2.35}$$

is an orthonormal basis for V_0, where

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k), \quad j,k \in \mathbb{Z}.$$

From (2.33) and (2.35), we see that

$$\{\phi_{j,k}(t) : j,k \in \mathbb{Z}\}$$

is an orthonormal basis for V_j [16]. We will also assume

$$\int_{\mathbb{R}} \phi(t)\ dt \neq 0.$$

Whenever a sequence of subspaces satisfies the properties of a multiresolution analysis, then there exists an orthonormal basis for $L_2(\mathbb{R})$,

$$\{\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k), j, k \in \mathbb{Z}\},$$

where $\psi(t)$ is the mother wavelet function obtained from the scaling function $\phi(t)$:

$$\psi(t) = \sqrt{2}\sum_{k\in\mathbb{Z}} c_{1-k}(-1)^k \phi(t-k), \tag{2.36}$$

where

$$c_k = \sqrt{2}\int_{-\infty}^{\infty} \phi(t)\phi(2t-k)\ dt,\ k \in \mathbb{Z}, \tag{2.37}$$

are the (generalized) Fourier coefficients of $\phi(t)$ in V_1 with respect to the orthonormal basis $\{\sqrt{2}\phi(2t-k) : k \in \mathbb{Z}\}$ for V_1 [16]. If we fix some $j_0 \in \mathbb{Z}$, then any target function $f(t) \in L_2(\mathbb{R})$ can be expressed as

$$f(t) = \sum_{k\in\mathbb{Z}} c_{j_0,k}\phi_{j_0,k}(t) + \sum_{j=j_0}^{\infty}\sum_{k\in\mathbb{Z}} d_{j,k}\psi_{j,k}(t) \quad \text{for almost all } t, \tag{2.38}$$

where the scaling coefficients

$$c_{j_0,k} = \int_{\mathbb{R}} f(t)\phi_{j_0,k}(t)\,dt \ \text{ for } k \in \mathbb{Z},$$

and the wavelet coefficients

$$d_{j,k} = \int_{\mathbb{R}} f(t)\psi_{j,k}(t)\,dt \ \text{ for } k \in \mathbb{Z} \ \text{ and } \ j \geq j_0$$

are defined as the inner product of $f(t)$ with the basis functions $\phi_{j_0,k}(t)$ and $\psi_{j,k}(t)$, respectively [62]. The representation (2.38) is analogous to the Fourier series representation of a square integrable function on the interval $[0, 2\pi)$ in terms of trigonometric basis functions.

The discrete wavelet transform (DWT) maps data from the time domain to the wavelet domain [62]. Wavelet transforms are linear and can be defined by the matrix operation

$$\boldsymbol{W} = \boldsymbol{\Theta}\boldsymbol{X}, \tag{2.39}$$

where $\boldsymbol{X} = (X_1, \ldots, X_n)^{\mathrm{T}}$ is the original data vector, $\boldsymbol{\Theta}$ is the $n\times n$ matrix that defines the DWT associated with the particular scaling functions and wavelet functions used in the representation (2.38), and $\boldsymbol{W}$ is the $n \times 1$ vector of estimated scaling and wavelet coefficients. Extending the

analogy between wavelet analysis and Fourier analysis, we see that transforming a data set via the DWT closely resembles the process of computing the Fast Fourier Transformation (FFT) of that data set. Because of the orthogonality of $\boldsymbol{\Theta}$, the inverse DWT is given by

$$\boldsymbol{X} = \boldsymbol{\Theta}^{\mathrm{T}} \boldsymbol{W}, \tag{2.40}$$

where $\boldsymbol{\Theta}^{\mathrm{T}}$ denotes the transpose of $\boldsymbol{\Theta}$. Mallat [41] developed an efficient algorithm to compute the DWT and the inverse DWT if the total sample size $n = 2^J$ for some positive integer J. For more details on this algorithm, see [46]. In this research, we will assume the sample size n is a power of 2 and Mallat's algorithm will be used to compute the DWT and the inverse DWT.

In general, after applying the DWT to a data set $\{X_i : i = 1, \ldots, n\}$, where the sample size has the form $n = 2^J$, we get the following approximation formula for the function $f(t)$,

$$f(t) \approx \sum_{k=0}^{2^{j_0}-1} \widehat{c}_{j_0,k} \phi_{j_0,k}(t) + \sum_{j=j_0}^{J-1} \sum_{k=0}^{2^j-1} \widehat{d}_{j,k} \psi_{j,k}(t), \tag{2.41}$$

where the estimated scaling coefficients $\{\widehat{c}_{j_0,k} : k = 0, 1, \ldots, 2^{j_0} - 1\}$ and the estimated wavelet coefficients $\{\widehat{d}_{j,k} : k = 0, 1, \ldots, 2^j - 1\}$ for the jth level of resolution ($j = j_0, j_0 + 1, \ldots, J - 1$) all appear as corresponding elements of the vector $\boldsymbol{W}$ defined by (2.39). In general, at the jth level of resolution, there will be 2^j coefficients, beginning with the coarsest level $j = j_0$. In the remainder of this thesis, any general empirical coefficient in the vector $\boldsymbol{W}$ will be denoted by $\widehat{w}_{j,k}$. An estimated coefficient will be denoted by $\widehat{w}_{j,k}$ when it is not necessary to distinguish it as a scaling coefficient or a wavelet coefficient.

2.5. Thresholding Wavelet Coefficients

There are many wavelet model selection procedures in the literature that are based on the idea of selecting "important" wavelet coefficients and setting to zero the "unimportant" coefficients. These methods attempt to find an optimal number of coefficients to accurately represent the data, thereby leading to a simplified and smoother (less noisy) data model. In many situations, it is often the case that the goal is to estimate a function $f(t)$ from some set of noisy data $\{x_i : i = 1, \ldots, n\}$. Suppose the data x_i are given by

$$x_i = f_i + \varepsilon_i, \tag{2.42}$$

where the errors $\{\varepsilon_i\}$ are i.i.d. with $\varepsilon_i \sim N(0, \sigma_\varepsilon^2)$ and $f_i = f(i)$ for $i = 1, \ldots, n$. When we take the DWT of the noisy data vector

$$\boldsymbol{X} = (x_1, \ldots, x_n)^{\mathrm{T}} = (f_1 + \varepsilon_1, \ldots, f_n + \varepsilon_n)^{\mathrm{T}},$$

we have

$$\widehat{w}_{j,k} = \widetilde{f}_{j,k} + \widetilde{\varepsilon}_{j,k}, \tag{2.43}$$

where: $\widehat{w}_{j,k}$ denotes a particular element in the vector $\boldsymbol{W}$ of scaling and wavelet coefficients in (2.39); and $\widetilde{f}_{j,k}$ and $\widetilde{\varepsilon}_{j,k}$ are the corresponding coefficients at the jth level of resolution for the "signal" component $(f_1, \ldots, f_n)^{\mathrm{T}}$ and "noise" component $(\varepsilon_1, \ldots, \varepsilon_n)^{\mathrm{T}}$ of $\boldsymbol{X}$, respectively. From (2.43), we see that each coefficient, $\widehat{w}_{j,k}$, generated by the DWT contributes noise with variance $\sigma^2_{\widetilde{\varepsilon}}$. By shrinking the small coefficients to zero, this noise can be removed, while at the same time preserving the original features of the function f.

One method often used to fit data using wavelets is to compute a set of multiresolution approximations [42, 9]. This method involves first constructing an approximation to the data using the coarsest-level signal and then adding increasingly finer levels of resolution. As more levels of resolution are used, the approximation to the target data set improves. Figures 2.1–2.3 below depict multiresolution approximations to a data set of size $n = 128$ that we obtained from a rapid thermal chemical vapor deposition (RTCVD) process. For more information on the RTCVD process, see [38]. Notice in particular the following aspects of Figures 2.1–2.3:

(i) In the top panel of Figure 2.1, the solid curve represents the wavelet approximation to the data set based on taking the level index $j_0 = 2$ in equation (2.38) and then estimating only the coefficients $\{c_{2,k} : k = 0, 1, 2, 3\}$ of all relevant shifts (translations) of the j_0-level scaling functions $\{\phi_{2,k}(t) : k \in \mathbb{Z}\}$ while setting all other coefficients on the right-hand side of (2.38) to zero. Thus the top panel of Figure 2.1 depicts the accuracy of the wavelet approximation to the target data set based on four coefficients at the coarsest level.

(ii) In the bottom panel of Figure 2.1, the solid curve represents the wavelet approximation to the data set based on estimating the eight coefficients at the top two coarsest levels—namely, the coefficients $\{c_{2,k} : k = 0, 1, 2, 3\}$ of all relevant shifts of the j_0-level scaling functions $\{\phi_{2,k}(t) : k \in \mathbb{Z}\}$ and the coefficients $\{d_{2,k} : k = 0, 1, 2, 3\}$ of the corresponding shifts of the j_0-level wavelet functions $\{\psi_{2,k}(t) : k \in \mathbb{Z}\}$—while setting all other coefficients at the finer levels to zero.

(iii) In the top panel of Figure 2.2, the solid curve represents the wavelet approximation to the data set based on estimating sixteen coefficients—namely, all the j_0-level coefficients mentioned in the previous item **(ii)** and the coefficients $\{d_{3,k} : k = 0, 1, \ldots, 7\}$ of all relevant shifts of the $(j_0 + 1)$-level wavelet functions $\{\psi_{3,k}(t) : k \in \mathbb{Z}\}$—while setting all other coefficients at the finer levels to zero.

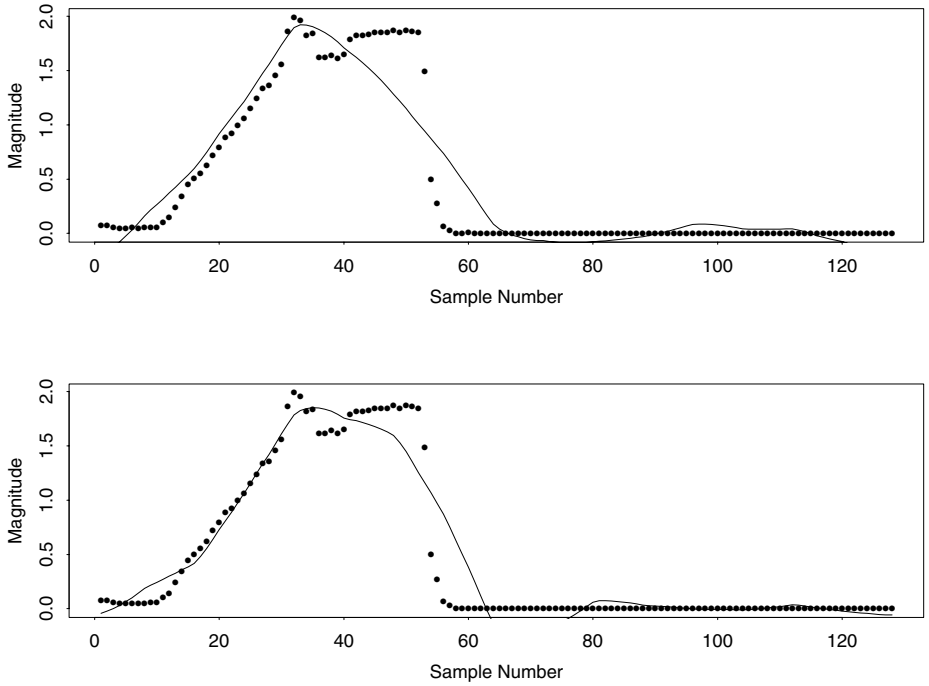

Figure 2.1: Multiresolution approximation to a run of the RTCVD process using four coefficients (top panel) and eight coefficients (bottom panel).

(iv) Similarly in the remaining panels of Figures 2.2 and 2.3, the solid curve represents the wavelet approximation to the data set based on estimating the indicated number of coefficients up to and including all relevant shifts of the next-higher-level wavelet functions while setting all other coefficients at the finer levels to zero. At each level j in this example, there are 2^j nonzero coefficients $\{d_{j,k} : k = 0, 1, \ldots, 2^j - 1\}$ associated with the relevant shifts of the corresponding wavelet functions $\{\psi_{j,k}(t) : k \in \mathbb{Z}\}$.

Notice the improvement in the approximation as more levels of detail are added by including coefficients at the finer levels hierarchically. At the finest level of resolution with index $j = j_0 + 4 = 6$, the total number of estimated wavelet coefficients equals the size of the data set ($n = 128$) so that the data set is exactly reconstructed as shown in the bottom panel of Figure 2.3.

While easy to use, this type of "linear" multiresolution approximation tends to oversmooth the data. For example, in the bottom panel of Figure 2.1, the eight coefficients in the two coarsest-resolution levels are unable to represent the dip around sample number 40. Of course, if such a dip in the data is considered to be less important or data noise, then it is reasonable to filter out that dip. However, if that dip is considered to be an important characteristic of the

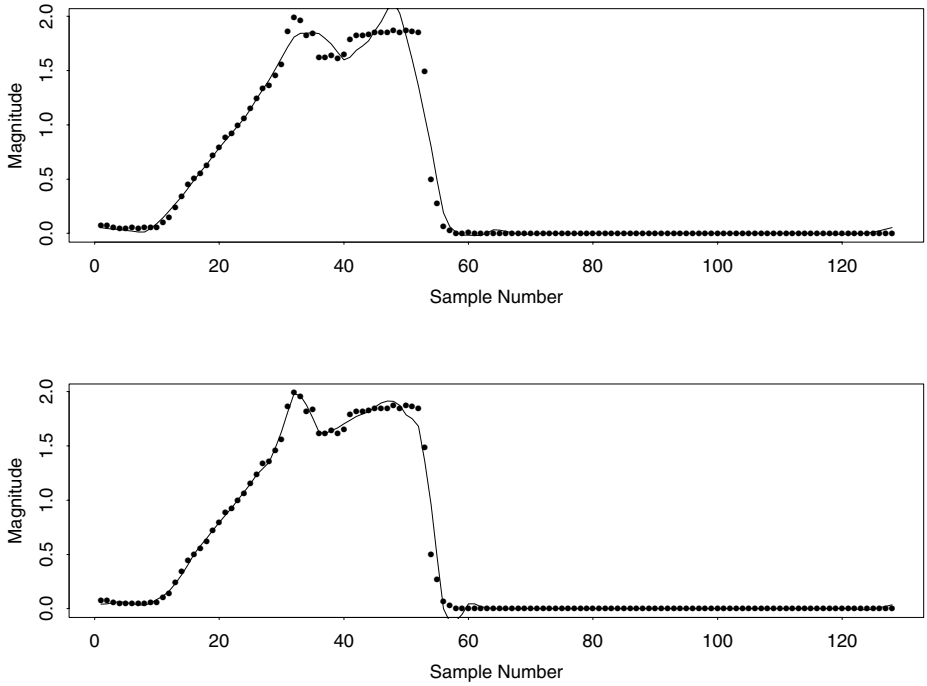

Figure 2.2: Multiresolution approximation to a run of the RTCVD process using sixteen coefficients (top panel) and thirty-two coefficients (bottom panel).

target data set, then we should use "nonlinear" approximation methods. In particular, nonlinear methods that select "important" wavelet coefficients (usually the largest in magnitude) and set to zero the "unimportant" coefficients (usually those representing noise) are effective in accurately representing small jumps or dips in the data with typically fewer coefficients than an approach based on a straightforward multiresolution approximation. One simple nonlinear approximation method is to select a certain number of coefficients with the largest magnitude, regardless of the level of resolution, and set the remaining coefficients to zero. Figure 2.4 shows a reconstruction of the same data set depicted in Figures 2.1–2.3 using the eight estimated wavelet coefficients having the largest magnitudes. Even though this reconstruction is not perfect, the dip around sample number 40 is represented. Figure 2.5 shows the same data set reconstructed with the nineteen estimated coefficients having the largest magnitudes. From this plot, it is clear that it is possible to achieve an excellent approximation to the data using as few as nineteen coefficients.

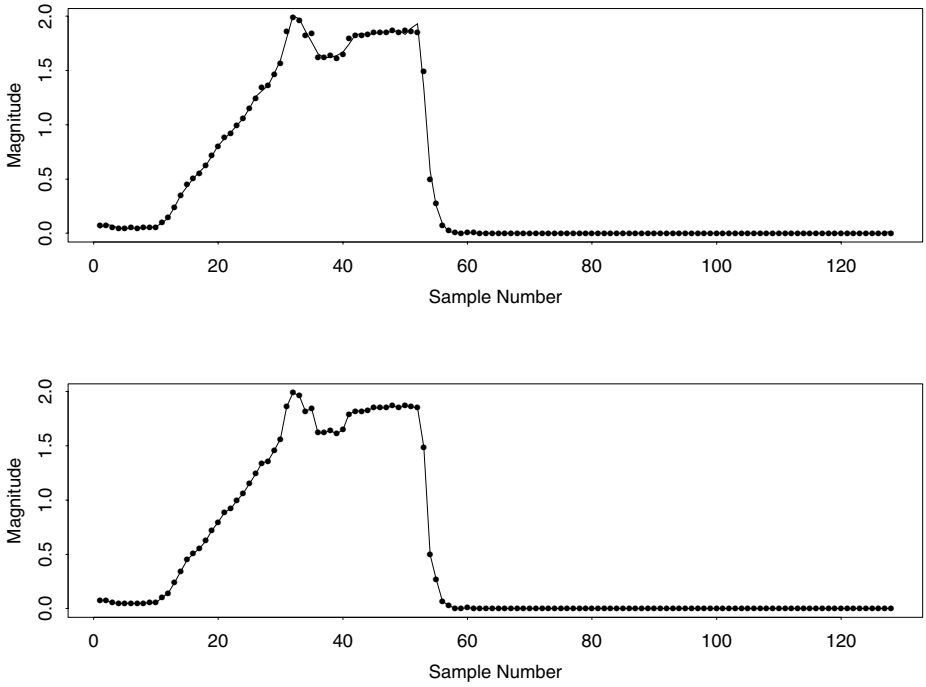

Figure 2.3: Multiresolution approximation to a run of the RTCVD process using sixty-four coefficients (top panel) and 128 coefficients (bottom panel).

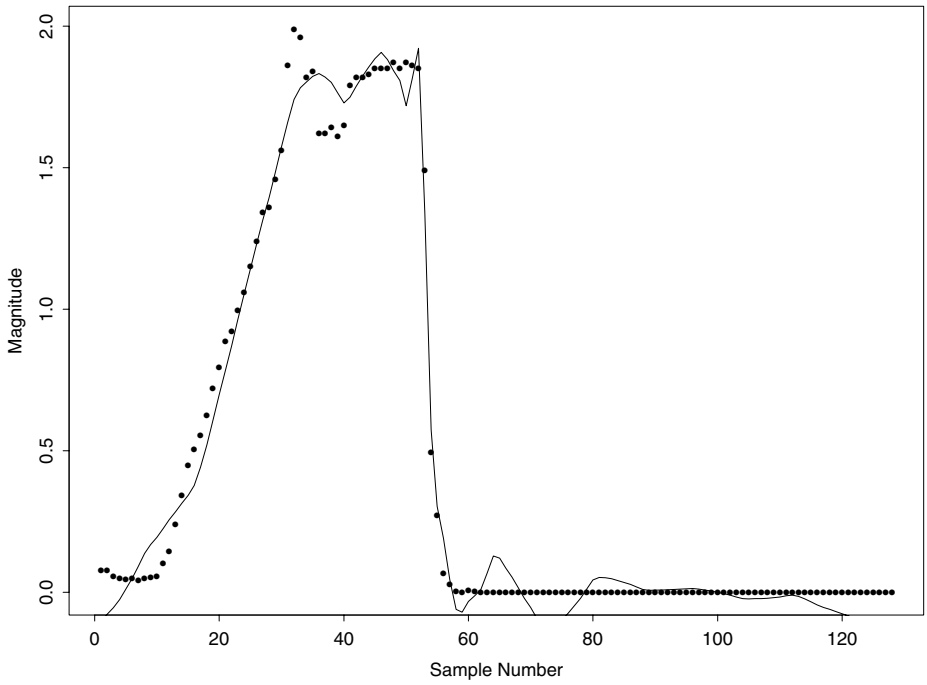

Figure 2.4: Reconstruction of the RTCVD data set using the eight largest-magnitude coefficients.

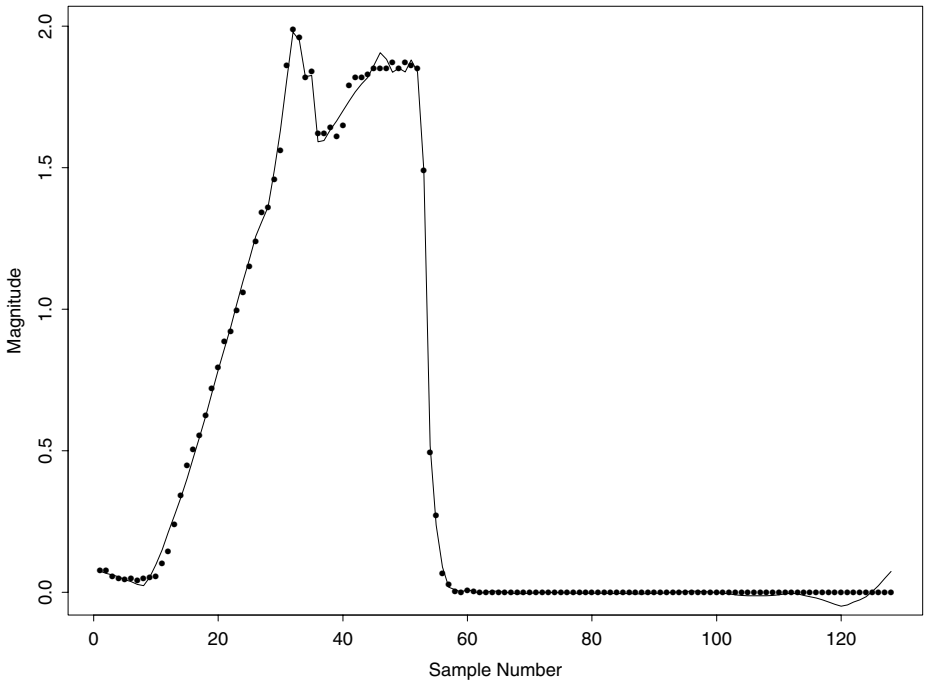

Figure 2.5: Reconstruction of the RTCVD data set using the nineteen largest-magnitude coefficients.

Another nonlinear approximation method for the selection of significant wavelet coefficients was proposed by Donoho and Johnstone [18]. Their *thresholding* method is based on the idea that wavelet coefficients are set to zero if their absolute value is below a certain threshold level $\lambda \geq 0$. The two most common thresholding rules are *hard* and *soft*. Using either the hard or soft thresholding rule, we obtain the thresholded scaling coefficients,

$$\widehat{c}^*_{j_0,k} = \vartheta_\lambda(\widehat{c}_{j_0,k}); \tag{2.44}$$

where

$$\vartheta_\lambda(y) = \begin{cases} y & \text{if } |y| > \lambda \\ 0 & \text{if } |y| \leq \lambda, \end{cases} \tag{2.45}$$

for hard thresholding and

$$\vartheta_\lambda(y) = \text{sgn}\,(y)\max(0, |y| - \lambda), \tag{2.46}$$

for soft thresholding. Similarly, we obtain the thresholded wavelet coefficients

$$\widehat{d}^*_{j,k} = \vartheta_\lambda(\widehat{d}_{j,k}),$$

where ϑ_λ is as defined in (2.45) or (2.46). Hard thresholding is a "keep or kill" rule, whereas soft thresholding is a "shrink or kill" rule. Once the thresholded coefficients are obtained (using either the hard or soft rule), the original function f can be reconstructed in the following way,

$$f(t) \approx \sum_{k=0}^{2^{j_0}-1} \widehat{c}^*_{j_0,k}\phi_{j_0,k}(t) + \sum_{j=j_0}^{J-1}\sum_{k=0}^{2^j-1} \widehat{d}^*_{j,k}\psi_{j,k}(t).$$

The choice of an appropriate threshold λ is crucial for the effectiveness of the thresholding method for removing noise. If the threshold is too large, certain local features of the target function may not be preserved. On the other hand, if the threshold is too small, there will be noise in the reconstructed function. There are many techniques for selecting a threshold in the literature. Donoho and Johnstone developed the universal threshold [17],

$$\lambda = \frac{\sigma_\varepsilon}{\sqrt{n}}\sqrt{2\ln(n)}, \tag{2.47}$$

where

$$\sigma^2_\varepsilon = \text{Var}(\varepsilon_i) = \sigma^2_{\widetilde{\varepsilon}} = \text{Var}(\widetilde{\varepsilon}_{j,k}) \quad \text{for} \quad i = 1, \ldots, n. \tag{2.48}$$

Equation (2.48) holds since the wavelet transform is orthogonal. The universal threshold is appealing because of its simplicity. In practice, however, it tends to oversmooth. That is, the only feature of the data used to compute the universal threshold is σ_ε. For large samples, it has been shown that the universal threshold will remove all the noise in the reconstruction with high probability, but features of the original, underlying function may be lost as well [1]. As an alternative to the universal threshold, Donoho and Johnstone proposed an *adaptive* threshold. Their procedure, called *SureShrink*, removes noise by assigning a threshold to each multiresolution level by minimizing the Stein unbiased estimate of risk (SURE) for threshold estimates. For more details about the SureShrink procedure, see [18]. There are a number of other thresholding schemes in the literature, including methods that utilize Bayesian approaches and the statistical technique of cross-validation to select a threshold. For a comprehensive overview of these methods, as well as others, see [45] or [62].

When selecting a threshold, there are no specific rules. In most cases it depends on the application. The top plot of Figure 2.6 below shows a noisy doppler signal that was created by sampling $n = 1024$ equally spaced points on the interval $[0, 1]$ of a doppler signal with random noise added to it. The middle plot shows the reconstruction of the doppler signal after applying a soft threshold to the wavelet coefficients using Donoho and Johnstone's SureShrink method. The bottom plot of Figure 2.6 shows the same noisy doppler signal reconstructed after applying a soft threshold to the wavelet coefficients using the universal threshold (2.47). The sample standard deviation of the finest scale detail coefficients $\{\widehat{d}_{J-1,k} : k = 0, \ldots, 2^{J-1}\}$ was used to estimate the standard deviation, σ_ε, of the noise term ε_i in (2.42). This is the default method for estimating σ_ε in the statistical software package S-Plus [9]. For the doppler signal in Figure 2.6, it appears that the universal threshold results in a slightly smoother, less noisy signal than the SureShrink method. For a comprehensive overview of proposals for threshold selection, see [45].

2.6. Wavelet-Based Estimation of the Spectrum

The analysis of a time series $\{X_i\}$ can be conducted in either the time domain or the frequency domain. In the time domain, the relationship between observations at different points in time is studied. In the frequency domain, the cyclical movements of the data are studied [21]. These two methods for analyzing time series data are complementary and each provides a different insight into the behavior of a particular time series.

One important tool for exploring the frequency behavior of a time series is estimation of the spectral density (or power spectrum). The spectral density function $p_X(\omega)$ of the process

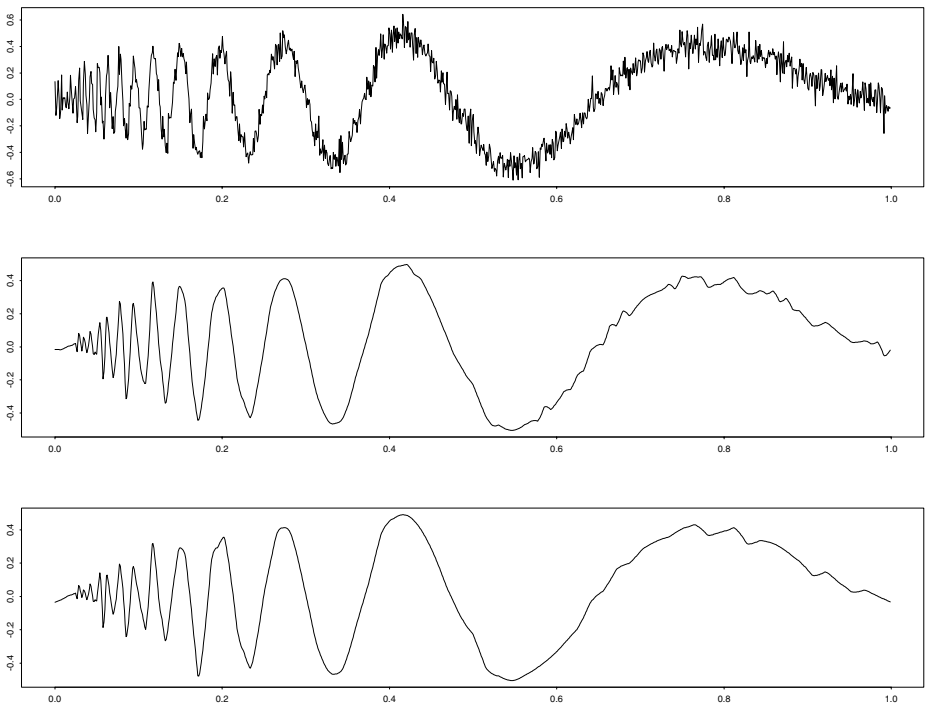

Figure 2.6: Doppler signal with random noise (top panel). Reconstruction of the doppler signal after applying the SureShrink method to the wavelet coefficients (middle panel). Reconstruction of the doppler signal after applying the universal threshold to the wavelet coefficients (bottom panel).

$\{X_i : i = 1, \ldots, n\}$ is defined as follows,

$$p_X(\omega) = \sum_{l=-\infty}^{\infty} \gamma_X(l)\cos(2\pi\omega l) \quad \text{for} \quad -\frac{1}{2} \leq \omega \leq \frac{1}{2}, \tag{2.49}$$

where $\gamma_X(l)$ is the autocovariance function, as defined in (1.6). From (2.22), we see that the spectral density is the cosine transform of the autocovariance function. An asymptotically unbiased estimate of the spectrum $p_X(\omega)$ is the periodogram $\{I(\frac{l}{n}),\ \text{for } 0 \leq l \leq n-1\}$, where

$$\begin{aligned} I\left(\frac{l}{n}\right) &= \frac{1}{n}\left\{\left[\sum_{j=1}^{n} X_j \cos\left(\frac{2\pi(j-1)l}{n}\right)\right]^2 + \left[\sum_{j=1}^{n} X_j \sin\left(\frac{2\pi(j-1)l}{n}\right)\right]^2\right\} \\ &= \frac{|a(l)|^2}{n}, \end{aligned} \tag{2.50}$$

where $a(l)$ is the discrete Fourier transform of the process $X_1, X_2, \ldots, X_n$. The periodogram

has the following asymptotic properties for large n:

$$\begin{aligned}
&\mathrm{E}\Big[I\Big(\tfrac{l}{n}\Big)\Big] \approx p_X\Big(\tfrac{l}{n}\Big), && 0 < l < \tfrac{n}{2},\\
&\mathrm{Var}\Big[I\Big(\tfrac{l}{n}\Big)\Big] \approx p_X^2\Big(\tfrac{l}{n}\Big), && 0 < l < \tfrac{n}{2},\\
&\mathrm{Cov}\Big[I\Big(\tfrac{l}{n}\Big), I\Big(\tfrac{j}{n}\Big)\Big] \approx 0, && 0 < l \neq j < \tfrac{n}{2}, \quad \text{and}\\
&I\Big(\tfrac{l}{n}\Big) \;\dot{\sim}\; p_X\Big(\tfrac{l}{n}\Big)\,\chi^2(2)/2, && 0 < l < \tfrac{n}{2},
\end{aligned} \tag{2.51}$$

where $\chi^2(2)$ denotes a chi-square random variable with two degrees of freedom so that $\chi^2(2)/2$ is an exponential random variable with mean one. From the above properties, the advantage of working in the frequency domain instead of the time domain becomes apparent. While the original output process $\{X_i\}$ may be highly correlated, the periodogram $\{I(\frac{l}{n}) : l = 1, \ldots, \frac{n}{2}-1\}$ is approximately uncorrelated.

In practice, the periodogram is not a good estimate of the spectrum since it does not have a constant variance. To obtain an estimate that has constant variance, the natural logarithm of the peridogram is used to estimate the log-spectrum. Several wavelet-based techniques for estimating the log-spectrum from the log-periodogram have been developed. Both Moulin [43] and Gao [21] have proposed wavelet-based methods for estimating the log-spectrum, denoted by

$$\zeta(\omega) = \ln\{p_X(\omega)\},$$

using the thresholding ideas presented in the previous section. In general, both methods follow the basic steps below:

(i) Expand $\mathcal{L}_i = \ln\{I\left(\frac{i}{n}\right)\}$ for $i = 1, 2, \ldots, n-1$ as a wavelet series using the DWT,

$$\boldsymbol{W} = \boldsymbol{\Theta}\boldsymbol{\mathcal{L}}, \tag{2.52}$$

where $\boldsymbol{\mathcal{L}}$ is given by

$$\begin{aligned}
\boldsymbol{\mathcal{L}} &= [\mathcal{L}_1, \ldots, \mathcal{L}_{n-1}]^{\mathrm{T}}\\
&= \left[\ln\left\{I\left(\frac{1}{n}\right)\right\}, \ldots, \ln\left\{I\left(\frac{n-1}{n}\right)\right\}\right]^{\mathrm{T}}.
\end{aligned} \tag{2.53}$$

(ii) Threshold the resulting coefficients $\widehat{c}_{j_0,k}$ and $\widehat{d}_{j,k}$ to obtain the coefficients $\widehat{c}^*_{j_0,k}$ and $\widehat{d}^*_{j,k}$.

(iii) Perform the inverse transform

$$\boldsymbol{\mathcal{L}}^* = \boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{W}^*,$$

where $\boldsymbol{W}^*$ is the vector containing the thresholded coefficients $\widehat{c}^*_{j_0,k}$ and $\widehat{d}^*_{j,k}$, to obtain

the corresponding thresholded wavelet approximation $\mathfrak{L}^*$ to the vector (2.53) of log-periodogram values.

The most important step in the above algorithm for wavelet-based estimation of the log-spectrum is the selection of an appropriate threshold. When we use wavelets to estimate the log-spectrum, we represent the log-periodogram as a signal plus noise, as in (2.42), where the signal is the true log-spectrum. However, because the noise associated with the log-periodogram is not normally distributed, we need to make a modification to equation (2.42). Wahba [63] proposed the following non-Gaussian model for the log-periodogram using the fourth property of the periodogram in (2.51),

$$\mathfrak{L}_i - \mathrm{E}\left[\ln\left(\frac{\chi^2(2)}{2}\right)\right] = \zeta\left(\frac{i}{n}\right) + \delta_i, \quad i = 1, 2, \ldots, n-1, \tag{2.54}$$

where $\zeta\left(\frac{i}{n}\right)$ is the log-spectrum at frequency i/n and δ_i is the noise, or error term, with

$$\delta_i = \ln\left(\frac{\chi^2(2)}{2}\right) - \mathrm{E}\left[\ln\left(\frac{\chi^2(2)}{2}\right)\right],$$

so that the δ_i have zero mean and are i.i.d. Furthermore, by calculating the first two derivatives of the moment generating function of δ_i at zero, we see that

$$\mathrm{E}\left[\ln\left(\frac{\chi^2(2)}{2}\right)\right] = -\gamma, \tag{2.55}$$

where $\gamma \approx 0.57721$ is the Euler-Mascheroni constant and

$$\mathrm{E}[\delta_i] = 0 \quad \text{and} \quad \mathrm{Var}[\delta_i] = \frac{\pi^2}{6}, \ i = 1, 2, \ldots, n-1. \tag{2.56}$$

Using (2.55), we can rewrite (2.54) as follows,

$$\mathfrak{L}_i = \zeta\left(\frac{i}{n}\right) - \gamma + \delta_i. \tag{2.57}$$

When we take the DWT of the log-periodogram, we have from (2.53) that

$$\boldsymbol{\mathfrak{L}} = \left[\zeta\left(\frac{1}{n}\right) - \gamma + \delta_1, \ldots, \zeta\left(\frac{n-1}{n}\right) - \gamma + \delta_{n-1}\right]^{\mathrm{T}}; \tag{2.58}$$

and thus the components of $\boldsymbol{W}$ defined by (2.52) have the form

$$\widehat{w}_{j,k} = \widetilde{\zeta}_{j,k} + \widetilde{\delta}_{j,k}, \tag{2.59}$$

where: $\widehat{w}_{j,k}$ represents a particular coefficient for the log-periodogram at resolution level j; and $\widetilde{\zeta}_{j,k}$ and $\widetilde{\delta}_{j,k}$ are the corresponding estimates at the jth level of resolution for the translated log-spectrum $\left[\zeta\left(\frac{1}{n}\right)-\gamma,\ldots,\zeta\left(\frac{n-1}{n}\right)-\gamma\right]^{\mathrm{T}}$ and the error component $(\delta_1,\ldots,\delta_{n-1})^{\mathrm{T}}$, respectively. That is, each coefficient $\widehat{w}_{j,k}$ has a certain amount of noise, $\widetilde{\delta}_{j,k}$, corrupting it. Furthermore, the distribution of the coefficient $\widetilde{\delta}_{j,k}$ is independent of the translation index k, but depends on the resolution level j.

Since the error term δ_i in (2.54) is log-χ^2 distributed, thresholding techniques based on normally distributed noise, like those in the previous section, cannot be used to estimate the log-spectrum. To address this problem, both Gao and Moulin developed thresholds for wavelet-based estimation of the spectrum that depend on the multiresolution level. Gao's soft thresholding algorithm for the empirical wavelet coefficients $\{\widehat{d}_{j,k}: k=0,1,\ldots,2^j-1\}$ at a particular resolution level j is given below,

$$\lambda_j = \max\left[\frac{\pi}{\sqrt{6n}}\sqrt{2\ln(n)},\ 2^{-(J-j-1)/4}\ln(2n)\right] \quad \text{for} \quad j=j_0,j_0+1,\ldots,J-1. \tag{2.60}$$

Because the empirical scaling coefficients $\{\widehat{c}_{j_0,k}: k=0,1,\ldots,2^{j_0}-1\}$ contain important information about the underlying coarse features of the signal, they are not thresholded.

The practicality of Gao's thresholding scheme can be justified as follows. Since $\boldsymbol{\Theta}$ is an $(n-1)\times(n-1)$ orthogonal matrix [46], it follows that the variance-covariance matrix of the $(n-1)\times 1$ wavelet coefficient vector $\boldsymbol{W}$ in (2.52) is given by

$$\begin{aligned}
\mathrm{Cov}(\boldsymbol{W}) &= \boldsymbol{\Theta}\,\mathrm{Cov}(\boldsymbol{\mathcal{L}})\,\boldsymbol{\Theta}^{\mathrm{T}} \\
&\approx \boldsymbol{\Theta}\left[\frac{\pi^2}{6}\boldsymbol{I}_{(n-1)\times(n-1)}\right]\boldsymbol{\Theta}^{\mathrm{T}} \\
&= \frac{\pi^2}{6}\boldsymbol{\Theta}\,\boldsymbol{\Theta}^{\mathrm{T}} \\
&= \frac{\pi^2}{6}\boldsymbol{I}_{(n-1)\times(n-1)},
\end{aligned} \tag{2.61}$$

where $\boldsymbol{I}_{(n-1)\times(n-1)}$ denotes the $(n-1)\times(n-1)$ identity matrix; and it follows from (2.61) that

$$\mathrm{Var}(\widehat{c}_{j_0,k}) = \mathrm{Var}(\widehat{d}_{j,k}) = \frac{\pi^2}{6} \quad \text{for} \quad k=0,1,\ldots,2^j-1 \quad \text{and} \quad j=j_0,j_0+1,\ldots,J-1. \tag{2.62}$$

Moreover, if the dependence structure of the time series $\{X_i: i=1,\ldots\}$ dies off sufficiently fast, then the empirical coefficients $\widehat{d}_{j,k}$ will be asymptotically normal,

$$\widehat{d}_{j,k} - \mathrm{E}\left[\widehat{d}_{j,k}\right] \sim N(0,\pi^2/6), \quad \text{as} \quad n\to\infty \quad \text{and} \quad j\to j_0 \tag{2.63}$$

[21]. Typically, the longer the memory of the time series, the weaker the asymptotic normality

When computing the DWT of the log-periodogram as in (2.52), we calculated each coefficient as a linear combination of terms in the log-periodogram. The number of terms in the linear combination for a particular coefficient at resolution level j increases exponentially as $j \to j_0$. As a result, for the coarser levels of coefficients (j close to j_0), there are a sufficient number of terms in the linear combination used to compute the DWT for the Central Limit Theorem to take effect, leading to (2.63). Therefore, when n is large, the universal threshold (2.47) can be used for the coarse levels of resolution since Gaussian approximation applies. Notice that the first term on the right hand side of (2.60) is in fact the universal threshold (2.47) with $\sigma_\varepsilon = \pi/\sqrt{6}$.

For the finer levels of resolution ($j \gg j_0$), the normality approximation breaks down since the coefficients at a fixed resolution level j, $\{\widehat{d}_{j,k} : k = 0, \ldots, 2^j - 1\}$, are linear combinations of a small number of points in the log-periodogram. That is, there are not enough terms in the linear combination used to compute a particular coefficient $\widehat{d}_{j,k}$ for the Central Limit Theorem to take effect. Gao developed the threshold given in the second part of the right-hand side of equation (2.60) for the finer levels of resolution where the Gaussian approximation cannot be applied. This threshold will in general be larger than the universal threshold since a stricter, or larger threshold is needed at the finer levels of resolution where the departure from normality is the greatest. For details on the derivation and justification of this threshold, see [21].

As an alternative to Gao's threshold for estimation of the spectrum, Moulin's wavelet-based method for estimating $\zeta(\omega) = \ln\{p_X(\omega)\}$ includes a hard thresholding scheme based on a saddlepoint approximation to the distribution of the sample wavelet coefficients of $\mathfrak{L}$. The goal of Moulin's thresholding method is to shrink the small coefficients towards zero, since those coefficients are likely due to noise. This is done by testing the null hypothesis

$$\mathrm{H}_0 : d_{j,k} = 0$$

against the alternative

$$\mathrm{H}_a : d_{j,k} \neq 0,$$

for each wavelet coefficient $d_{j,k}$. The null hypothesis is accepted when the empirical coefficient $\widehat{d}_{j,k}$ falls in the acceptance region $[\lambda_j^-, \lambda_j^+]$ at significance level α ($0 < \alpha < 1$), where $\lambda_j^- < 0$ and $\lambda_j^+ > 0$. That is,

$$\Pr[\lambda_j^- < \widehat{d}_{j,k} < \lambda_j^+ | \mathrm{H}_0] = \int_{\lambda_j^-}^{\lambda_j^+} dG_{\widetilde{\delta}_{j,k}}(u) = 1 - \alpha, \tag{2.64}$$

where $G_{\widetilde{\delta}_{j,k}}(u)$ is the cumulative distribution function (c.d.f.) of the error term $\widetilde{\delta}_{j,k}$ in (2.59). Since the distribution of $\widetilde{\delta}_{j,k}$ is asymmetric with respect to the origin, the values of λ_j^- and λ_j^+

can be different. This hypothesis test is equivalent to applying the following hard threshold to the empirical wavelet coefficients,

$$\widehat{d}^{*}_{j,k} = \begin{cases} 0 & \text{if } \lambda_j^- < \widehat{d}_{j,k} < \lambda_j^+ \\ \widehat{d}_{j,k} & \text{otherwise.} \end{cases} \tag{2.65}$$

If we define the tail probabilities for a particular resolution level j as follows,

$$Q_{\widetilde{\delta}_{j,k}}(x) = \begin{cases} \int_x^{\infty} dG_{\widetilde{\delta}_{j,k}}(u) & x \geq 0 \\ \int_{-\infty}^{x} dG_{\widetilde{\delta}_{j,k}}(u) & x < 0, \end{cases}$$

we see from (2.64) that

$$Q_{\widetilde{\delta}_{j,k}}(\lambda_j^-) + Q_{\widetilde{\delta}_{j,k}}(\lambda_j^+) = \alpha, \tag{2.66}$$

and hence the magnitudes of the thresholds λ_j^- and λ_j^+ decrease with α. Let the probability of failure P_F be the least upper bound on the probability that the largest-magnitude noise wavelet coefficient is smaller than λ_j^- or larger than λ_j^+. That is, P_F is the least upper bound on the probability that a particular noisy coefficient will fall outside the interval $[\lambda_j^-, \lambda_j^+]$ and ultimately will not be thresholded correctly using the hard thresholding rule (2.65), so that we have

$$P_F \equiv \max_{j_0 < j < J} \left\{ \Pr \left[\max_{\substack{d_{j,k}=0 \\ 0 \leq k < 2^j}} \left(|\widehat{d}_{j,k}| \right) > \max \left(|\lambda_j^-|, |\lambda_j^+| \right) \right] \right\}. \tag{2.67}$$

From (2.66), Moulin shows that P_F is bounded from above by

$$P_F \leq 1 - (1-\alpha)^{n[2^{-1} - 2^{-(J-j_0)}]}.$$

The objective is to compute thresholds λ_j^- and λ_j^+ that approximate a specified probability of failure, P_F. This may be done using a saddle point approximation to the tail probability $Q_{\widetilde{\delta}_{j,k}}(.)$. The saddle point approximation is the first term in an asymptotic series expansion for $Q_{\widetilde{\delta}_{j,k}}(.)$. One simple solution to (2.66) is

$$Q_{\widetilde{\delta}_{j,k}}(\lambda_j^-) = Q_{\widetilde{\delta}_{j,k}}(\lambda_j^+) = \frac{\alpha}{2}, \tag{2.68}$$

with

$$\alpha = 1 - (1 - \widehat{P}_F)^{1/\left\{n[2^{-1} - 2^{-(J-j_0)}]\right\}}. \tag{2.69}$$

For a given probability of failure P_F and a given resolution level j, the thresholds λ_j^- and λ_j^+ can be computed from the distribution of the noise coefficients $\widetilde{\delta}_{j,k}$ using (2.68) and (2.69). Moulin states that this thresholding scheme should only be applied to the empirical wavelet coefficients $\{\widehat{d}_{j,k} : k = 0, 1, \ldots, 2^j - 1\}$ at resolution levels $j = j_0 + 1, \ldots, J - 1$. He presumes that the empirical scaling coefficients $\{\widehat{c}_{j_0,k} : k = 0, 1, \ldots, 2^{j_0} - 1\}$ and wavelet coefficients $\{\widehat{d}_{j_0,k} : k = 0, 1, \ldots, 2^{j_0} - 1\}$ at resolution j_0 contain information about the coarse features of the signal and should not be thresholded.

As an alternative to both Gao and Moulin's methods that use the periodogram as a spectrum estimator, Percival and Walden have proposed a wavelet-based method for estimating the power spectrum that uses a multitaper spectrum estimator. One drawback of their method is that the multitaper estimates are not approximately uncorrelated, and therefore an estimate of the correlation must be computed. For more details on this multitaper method, see [46] or [65].

2.7. Spectral Methods for Simulation Output Analysis

Instead of working in the time domain with the original output process $\{X_i : i = 1, \ldots, n\}$, a spectral analysis approach to steady-state output analysis works in the frequency domain, under the assumption that the process $X_1, X_2, \ldots$ is stationary with $\mathrm{E}[X_i] = \mu_X$. For frequency $\omega = 0$ we have

$$p_X(0) = \gamma_X = \sum_{l=-\infty}^{\infty} \gamma_X(l), \tag{2.70}$$

where $p_X(\omega)$ is the spectral density function as defined in (2.49). Provided $\sum_{l=-\infty}^{\infty} |\gamma_X(l)| < \infty$ and n is sufficiently large, the variance of $\overline{X}$ can be approximated by

$$\widehat{\mathrm{Var}}[\overline{X}] \approx \frac{p_X(0)}{n}, \tag{2.71}$$

where $\overline{X}$ is the usual point estimate of μ_X,

$$\overline{X} = \frac{1}{n} \sum_{i=1}^{n} X_i.$$

Therefore, to construct an asymptotically valid confidence interval for the steady-state mean μ_X, we need to estimate the spectral density at zero frequency, $p_X(0)$.

Classical methods for spectral estimation are based on the usual estimator for $\gamma_X(l)$,

$$\widehat{\gamma}_X(l) = \frac{1}{n-l} \sum_{j=1}^{n-l} (X_j - \overline{X})(X_{j+l} - \overline{X}), \; l = 0, \pm 1, \pm 2, \ldots, \pm(n-1). \tag{2.72}$$

Given n observations from the original output process, we can estimate at most $n-1$ covariance functions $\gamma_X(l)$ using equation (2.72). Hence, an estimate of the SSVC γ_X is

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

Summations over $\widehat{\gamma}_X(l)$ for a wide range of l tend to be highly variable, however. To compensate for this, it was found that if we use the truncated estimator,

$$\widehat{p}_X(0) = \widehat{\gamma}_X = \sum_{l=-M}^{M} \widehat{\gamma}_X(l), \tag{2.73}$$

where the truncation point M is much less than n, then the variance of the estimator $\widehat{p}_X(0)$ can be reduced, but at the cost of increasing the bias. To further improve the estimator in (2.73), we can weight the $\widehat{\gamma}_X(l)$'s in the following way, resulting in a less biased estimate $\widehat{p}_X(0)$,

$$\widehat{p}_X(0) = \widehat{\gamma}_X = \sum_{l=-M}^{M} w(l)\widehat{\gamma}_X(l), \tag{2.74}$$

where $w(l)$, the lag window, is a positive, even function with $w(0) = 1$ and $w(l) = 0$ for $|l| > M$. If the spectral density function $p_X(\omega)$ is approximately linear near 0, then $\widehat{p}_X(0)$ in (2.74) is an unbiased estimate of $p_X(0)$. That is, $\mathrm{E}[\widehat{p}_X(0)] \approx p_X(0)$. Using (2.74) and (2.71), we see that an asymptotically valid $100(1-\beta)\%$ confidence interval for μ_X is

$$\overline{X} \pm z_{1-\beta/2}\sqrt{\frac{\widehat{p}_X(0)}{n}}.$$

The two main issues that arise when using the classical approach to spectral estimation are the determination of the truncation point M and the lag window $w(l)$. It can be very difficult to decide on appropriate values for both M and $w(l)$. Furthermore, a more fundamental problem with the classical approach is the assumption that the spectral density function $p_X(\omega)$ is approximately linear near 0 [26]. Since $p_X(\omega)$ is an even function, it is symmetric about 0, implying that at 0 it either has a peak, it has a valley, or it is flat. Therefore, $p_X(\omega)$ is not approximately linear near 0 (unless it is flat); and the estimate for $p_X(0)$ given in equation (2.74) will be biased.

2.7.1 *Heidelberger and Welch's Spectral Method*

Heidelberger and Welch [26] developed an alternative approach to the classical methods for spectral analysis. The goal of their spectral analysis method is to estimate $p_X(0)$ from the

values of the periodogram. From (2.50), notice that $I(0) = n\overline{X}^2$; and hence the zero value of the periodogram depends only on the average value of the process and contains no information about $p_X(0)$. Heidelberger and Welch use regression techniques to obtain an estimate of $p_X(0)$ from the values of the periodogram in the region near 0. The classical approach described earlier averages over the periodogram, but it only gives an unbiased estimate of $p_X(0)$ when the function $p_X(\omega)$ is flat near 0. Heidelberger and Welch's approach assumes $p_X(\omega)$ is smooth near 0, thereby allowing the use of regression analysis and resulting in an estimate $\widehat{p}_X(0)$ equal to the y-intercept of the fitted regression function that does not have the bias problems that the classical estimator (2.74) has.

The steps of Heidelberger and Welch's procedure for the output process $\{X_i : i = 1, \ldots, n\}$ are as follows:

(i) Compute the periodogram $I(\frac{i}{n})$ for $i = 1, \ldots, n-1$ using (2.50) for the data $\{X_i : i = 1, \ldots, n\}$. Average over adajacent periodogram values to obtain a smoother, less erratic function

$$\widetilde{I}(\omega_j) = \frac{1}{2}\left[I\left(\frac{2j-1}{n}\right) + I\left(\frac{2j}{n}\right)\right],$$

where

$$\omega_j = \frac{1}{2}\left(\frac{2j-1}{n} + \frac{2j}{n}\right) = \frac{4j-1}{2n} \quad \text{for} \quad j = 1, \ldots, \frac{n}{4}.$$

Take the logarithm of this smoothed function to obtain a function

$$\widetilde{\mathcal{L}}(\omega_j) = \ln\{\widetilde{I}(\omega_j)\} \quad \text{for} \quad j = 1, \ldots, \frac{n}{4}$$

that has constant variance, so that $\widetilde{\mathcal{L}}(\omega_1)$ is the log of the average of $I(\frac{1}{n})$ and $I(\frac{2}{n})$, $\widetilde{\mathcal{L}}(\omega_2)$ is the log of the average of $I(\frac{3}{n})$ and $I(\frac{4}{n})$, and so on.

(ii) Using ordinary least squares, fit a polynomial of degree q,

$$g(\omega_j) = \sum_{h=0}^{q} a_h \omega_j^h,$$

to

$$\widetilde{\mathcal{L}}(\omega_j) + 0.270 \tag{2.75}$$

for $j = 1, \ldots, K$, where K is the number of points in the log of the smoothed periodogram $\widetilde{\mathcal{L}}(\omega_j)$ that are used to obtain the polynomial fit. The value 0.270 in equation (2.75) is the correction for the bias introduced by taking a logarithmic transformation.

(iii) The least squares estimate of a_0, $\widehat{a}_0$, is used as the estimator for $\zeta(0) = \ln[p_X(0)]$. An

estimate for $p_X(0)$ is obtained as follows,

$$\widehat{p}_X(0) = C_1 \mathrm{e}^{\widehat{a}_0},$$

where $C_1 = C_1(K, q)$ is a constant chosen to make $\widehat{p}_X(0)$ approximately unbiased.

(iv) A $100(1-\beta)\%$ confidence interval for the steady-state mean μ_X is given by

$$\overline{X} \pm t_{1-\beta/2,\nu} \sqrt{\frac{\widehat{p}_X(0)}{n}},$$

where the computation of the degrees of freedom $\nu = \nu(K, q)$ is as described in [26].

Heidelberger and Welch determined empirically that fitting a quadratic to the log of the smoothed periodogram (that is, setting $q = 2$) provides an optimal balance between small sample bias and large sample stability. If q is chosen large, then complexities in the shape of the spectrum are more easily captured. The larger q is, however, the larger the variance of the estimate of the spectrum at zero frequency. They also found empirically that setting $K = 25$ gives better small sample results, while setting $K = 50$ gives better large sample results.

Heidelberger and Welch's spectral method can also be applied to batched data. Suppose the output process $\{X_i : i = 1, \ldots, n\}$ is divided into k batches of size m. Let

$$\overline{X}_j = \overline{X}_j(m) = \frac{1}{m} \sum_{i=m(j-1)+1}^{mj} X_i \quad \text{for} \quad j = 1, \ldots, k$$

denote the jth batch mean,

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

denote the grand mean of the k batch means $\overline{X}_1, \ldots, \overline{X}_k$, and

$$p_{\bar{X}(m)}(\omega) = \sum_{l=-\infty}^{\infty} \gamma_{\bar{X}(m)}(l) \cos(2\pi\omega l), \quad -\frac{1}{2} \le \omega \le \frac{1}{2}$$

be the spectral density function of the batch-means process $\{\overline{X}_1(m), \ldots, \overline{X}_k(m)\}$ for batches of size m. Using the following result,

$$p_{\bar{X}(m)}(0) = \frac{p_X(0)}{m}, \tag{2.76}$$

we see that a confidence interval for the steady-state mean μ_X can be constructed by following the steps (i)–(iii) above to estimate $p_{\bar{X}(m)}(0)$. A $100(1-\beta)\%$ confidence interval for μ_X is then

given by,

$$\overline{\overline{X}} \pm t_{1-\beta/2,\nu}\sqrt{\frac{\widehat{p}_{\bar{X}(m)}(0)}{k}}, \tag{2.77}$$

where $\widehat{p}_{\bar{X}(m)}(0)$ is the Heidelberger-Welch estimator of the power spectrum at zero frequency for batch means with batch size m, where a quadratic polynomial ($q = 2$) is fitted to K points on the bias-corrected log of the smoothed periodogram (2.75) based on k batch means, and $\nu = C_1(K, 2)$ is taken from Table I of [26].

2.7.2 *A Sequential Spectral Method*

The spectral method described in Section 2.7.1 applies to a sample of fixed size. Heidelberger and Welch also developed a sequential method for constructing confidence intervals that satisfy a given accuracy requirement, specified as a maximum relative fraction r^* of the magnitude of the final grand mean $\overline{\overline{X}}$. In addition to the accuracy requirement, a maximum limit on the length of the simulation run is also specified. A confidence interval is generated once either the accuracy requirement is satisfied or the maximum run length is reached. Heidelberger and Welch's sequential procedure for confidence interval estimation is given as follows. First, the values of the following parameters are specified:

(i) the point t_1 at which the first confidence interval is to be constructed;

(ii) the maximum run length $t_{\max}$;

(iii) the required relative half-width r^*; and

(iv) a multiplier τ by which the run length is increased if the precision requirement is not met.

Beginning at time t_1, a set of batch means is computed (as described in [26]) and a confidence interval of the form $\overline{\overline{X}} \pm H$ is generated using equation (2.77). If

$$H \leq r^*|\overline{\overline{X}}|, \tag{2.78}$$

then the simulation is terminated. Otherwise, the simulation is continued and additional observations are generated until time

$$t_2 = \min(\tau \times t_1, t_{\max}). \tag{2.79}$$

A confidence interval is generated at time t_2. If the precision requirement (2.78) is met or if $t_2 = t_{\max}$, the simulation is terminated. Otherwise, additional observations are generated until

time

$$t_3 = \min(\tau \times t_2, t_{\max}). \tag{2.80}$$

A confidence interval is generated at time t_3. This process of generating data and then constructing a confidence interval is continued until either the precision requirement is met or the maximum simulation time $t_{\max}$ is reached.

In an attempt to improve both the small and large sample behavior of their spectral method, Heidelberger and Welch proposed a more flexible, adaptive method which selects the degree of the polynomial fit to the log of the smoothed periodogram according to the shape of the periodogram [27]. Previously, Heidelberber and Welch found a quadratic was necessary to obtain an unbiased estimate of the spectrum at zero frequency for small samples. As the batch size is increased, however (that is, as the sample becomes larger), the correlation between the batch means goes to zero and the spectrum becomes flatter. Therefore for large samples, fitting a linear function to the log of the smoothed periodogram may be more appropriate. In [27], Heidelberger and Welch propose several adaptive spectral methods for confidence interval estimation that use the following smoothing techniques: polynomial smoothing with the degree selected sequentially using standard regression statistics, polynomial smoothing with the degree selected by cross validation, and smoothing splines with the amount of smoothing determined by cross validation. Heidelberger and Welch tested their adaptive method on both samples of a fixed size and in the context of the sequential procedure described at the beginning of this section. They found that the performance of the adaptive methods when using a fixed sample size did not improve the small sample behavior, and only marginally improved the large sample behavior. They also found that the adaptive methods performed poorly when used in a sequential procedure. Therefore, Heidelberger and Welch ultimately recommend using the quadratic method described in [26].

Heidelberger and Welch also propose a variation of their sequential spectral method that incorporates a procedure for detecting and eliminating initialization bias [28]. To identify an appropriate warm-up period, Heidelberger and Welch apply a stationarity test to a sequence of batch means. The stationarity test involves the computation of a Brownian Bridge statistic whose asymptotic distribution under the assumption of stationarity is known. Beginning at time t_1, a set of batch means is computed and the stationarity test is applied to the sequence of batch means $\{\overline{X}_1, \ldots, \overline{X}_{t_1}\}$ to determine if there is a point t^* such that $\{\overline{X}_{t^*+1}, \ldots, \overline{X}_{t_1}\}$ is a sample from a covariance stationary process. If the sequence of batch means $\{\overline{X}_1, \ldots, \overline{X}_{t_1}\}$ passes the stationarity test, then a confidence interval of the form (2.77) is generated from the truncated set of batch means $\{\overline{X}_{t^*+1}, \ldots, \overline{X}_{t_1}\}$. If the precision requirement (2.78) is satisfied, then the simulation is terminated. Otherwise, we proceed to the next checkpoint t_2 as defined in (2.79).

If the sequence of batch means $\{\overline{X}_1, \ldots, \overline{X}_{t_1}\}$ fails the stationarity test (that is, there is no value of t^* for which the sequence of batch means $\{\overline{X}_{t^*+1}, \ldots, \overline{X}_{t_1}\}$ is a sample from a covariance stationary process), then we proceed immediately to the next checkpoint t_2 as defined by (2.79). At time t_2, the test for stationarity is applied to the sequence of batch means $\{\overline{X}_1, \ldots, \overline{X}_{t_2}\}$. This process of generating data, testing for stationarity, and then (possibly) constructing a confidence interval continues until either the precision requirement is met or the maximum simulation time $t_{\max}$ is reached. If the simulation reaches time $t_{\max}$, then the test for stationarity is performed on the sequence of batch means $\{\overline{X}_1, \ldots, \overline{X}_{t_{\max}}\}$. If the test is failed, then a confidence interval is not returned. If the stationarity test is passed, then a confidence interval is returned that may or may not satisfy the precision requirement.

Heidelberger and Welch's experimental results indicate that for processes with a pronounced initial transient period, the inclusion of a test for stationarity in their sequential spectral method results in point and confidence interval estimators with lower bias, smaller average confidence interval half-lengths, and shorter run lengths than when no test for stationarity is performed. However, for processes with no initial transient period, the inclusion of a stationarity test actually causes a slight degradation in the performance of their sequential spectral method. Furthermore, Heidelberger and Welch found that if the simulation run length is short relative to the length of the initial transient period, then the proposed stationarity test may not be powerful enough to detect initialization bias. Finally, Heidelberger and Welch found that their method had difficulty detecting an appropriate data-truncation point when the initial checkpoint t_1 was actually within the warm-up period. Therefore, Heidelberger and Welch's method for detecting and eliminating initialization bias requires the user to specify an initial time t_1 that is not in the warm-up period, implying the user must have some general indication of the length of the warm-up period for the method to work effectively.

Since the fast Fourier transform can be used to compute the periodogram, the spectral method can be more computationally efficient than the replication/deletion and regenerative methods. Furthermore, a spectral approach involves working with uncorrelated periodogram values, rather than with a highly correlated output sequence. However, the periodogram can have highly erratic behavior, making implementation of the method difficult. The experimental results obtained by Heidelberger and Welch for their quadratic spectral method indicate that restricting to polynomial fits to the periodogram is not flexible enough to obtain stable estimates of the spectrum at zero frequency. In the next chapter, we will describe a wavelet-based spectral method for steady-state output analysis that is an extension of Heidelberger and Welch's spectral method. This wavelet-based spectral method more adequately addresses the erratic behavior of the periodogram so that a stable estimate of the spectrum at zero frequency can be obtained.

Chapter 3

A Wavelet-Based Spectral Method

In the previous chapter, both nonspectral and spectral methods for steady-state output analysis were reviewed. From the experimental results for the existing spectral approaches, it is clear that additional follow-up work is needed to improve the performance of spectral methods in practice. In [26], Heidelberger and Welch fit a quadratic polynomial to the natural log of the smoothed periodogram of a simulation output process that may be raw (unbatched) or aggregated into a time series of nonoverlapping batch means. They apply this method with data sets of a fixed size as well as with a stopping rule. From Heidelberger and Welch's computational results, it is clear their spectral method does not produce sufficiently reliable confidence intervals (CIs) in practice. In an effort to improve the performance of their method, Heidelberger and Welch propose in [27] an adaptive spectral method that fits higher order polynomials to the log of the smoothed periodogram. Heidelberger and Welch find, however, that this adaptive spectral method does not perform any better than their original spectral method.

There are several explanations for the unreliable performance of Heidelberger and Welch's spectral method in terms of confidence interval coverage. First, to smooth the periodogram, Heidelberger and Welch average adjacent periodogram values before applying the logarithmic transformation. Based on our computational experience in estimating the log-spectrum of batched or unbatched simulation output processes from the log of the smoothed periodogram of the corresponding time series of batch means or individual (unbatched) observations, we have found that averaging adjacent periodogram values does not provide enough smoothing and that this approach will ultimately lead to an excessively noisy estimate of the associated log-spectrum. Least-squares estimation of a polynomial in the presence of excessive noise is in general very erratic, and Heidelberger and Welch's experimental results reflect this fact.

Another possible explanation for the unreliable performance of Heidelberger and Welch's spectral method is that a simple polynomial function of degree up to, say, three may not be sufficiently flexible to approximate adequately the behavior of the underlying log-spectrum even

when the log of the smoothed periodogram yields a reasonably accurate estimator of the log-spectrum. Furthermore, even after smoothing the periodogram, we have found that the resulting log-smoothed-periodogram has a certain degree of noise associated with it. Heidelberger and Welch attempt to denoise the log-smoothed-periodogram by applying standard least-squares regression techniques. These methods can be thought of as thresholding schemes in the sense that terms in the polynomial fitted to the log-periodogram are discarded until the desired fit is achieved. The problem with this type of thresholding is that going from a cubic to a quadratic polynomial, for example, makes a huge difference in the adequacy of the fit; and more than noise may be eliminated in the process—that is, important features of the log-smoothed-periodogram may be lost by simply going from a cubic to a quadratic fit.

In this chapter, we describe **WASSP**, a sequential, wavelet-based spectral method for constructing a valid confidence interval with user-specified levels of precision and coverage probability as an estimator of the mean of a steady-state simulation output process. This method addresses the problems associated with previous spectral methods and can be viewed as a natural extension of Heidelberger and Welch's method. **WASSP** determines a batch size and a warm-up period (statistics clearing time, data-truncation point) beyond which the computed batch means are not necessarily independent but constitute an approximately stationary Gaussian process; and then **WASSP** uses wavelets to approximate the log of the (appropriately) smoothed periodogram of the batch means, thereby yielding what is hoped to be a more flexible and accurate estimate of the log-spectrum of the batch means in the neighborhood of zero frequency. Moreover, the associated estimate of the steady-state variance constant of the original (unbatched) process can be combined with the grand average of all the batch means computed beyond the warm-up period to yield a reliable confidence-interval estimator of the steady-state mean response, where the delivered confidence interval satisfies a user-specified absolute or relative precision requirement.

In general, wavelets are an excellent tool for estimating functions that exhibit certain types of local behavior, such as sharp spikes or dips. The complex behavior often exhibited by the power spectra of batched or unbatched simulation outputs can be easily captured using wavelets. Furthermore, wavelets are an excellent tool for denoising log-spectrum approximations based on the log of the smoothed periodogram of batched or unbatched time series (see Chapter 3 of [66]). The wavelet coefficients can be carefully thresholded to filter out noise, while at the same time ensuring that important features of the log-smoothed-periodogram are retained.

The rest of this chapter is organized as follows. Section 3.1 provides a general overview of **WASSP** and how it operates. Section 3.2 contains a formal algorithmic statement of **WASSP**. A detailed description of the steps comprising **WASSP** is given in Section 3.3.

3.1. Overview of WASSP

Figure 3.1 depicts a high-level flowchart of the operation of WASSP. The algorithm begins by dividing the initial, simulation-generated output process into a fixed number of batches of uniform size. Batch means are computed for all batches, and an independence test is applied to the set of batch means. The independence test serves two purposes:

- It is used to construct a set of spaced batch means such that the interbatch spacer preceding each batch is sufficiently large to ensure all computed batch means are approximately independent and identically distributed (i.i.d.) so that subsequently the batch means can be tested for normality.
- It is used to determine an appropriate data-truncation point (statistics clearing time, warm-up period)—that is, the "interbatch" spacer preceding the first batch—beyond which all computed batch means are approximately independent of the simulation model's initial conditions.

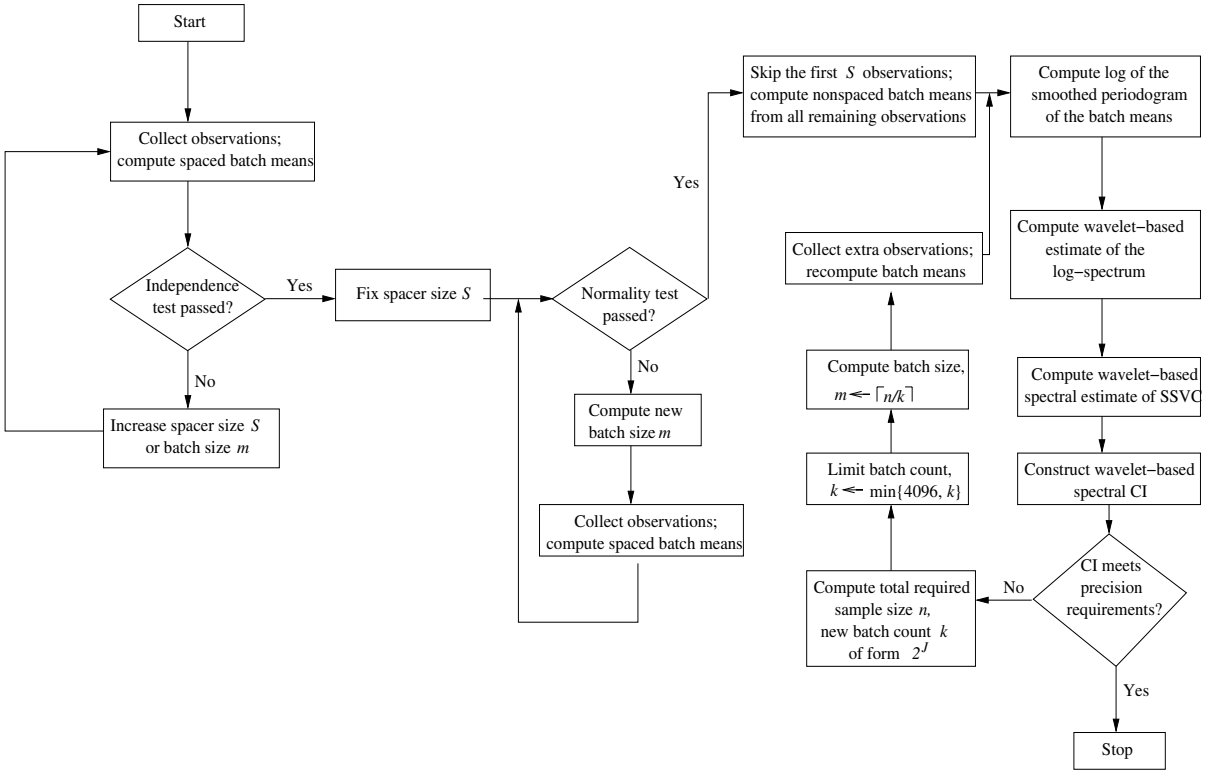

Figure 3.1: High-Level Flowchart of WASSP.

Once the independence test is passed, the set of approximately i.i.d. spaced batch means is tested for normality. Each time the normality test is failed, the following steps are executed:

- the batch size is increased;
- a new set of spaced batch means is computed using the final spacer size determined by the independence test; and
- the normality test is repeated for the new set of spaced batch means.

Once the normality test is passed, all simulation-generated data beyond the warm-up period is used to compute adajacent (nonspaced) batch means of the previously determined batch size; then the periodogram of the approximately normal batch means is computed. In general, the periodogram is more well behaved for data that is approximately normal. Furthermore, the periodogram approaches its asymptotic properties (as listed in equation (2.51)) more quickly if the data is normal.

After the periodogram for the approximately normal adjacent (nonspaced) batch means is computed, the periodogram is smoothed by taking a moving average of width A points. $\mathcal{WASSP}$ allows the user to specify the value of A in the set $\{5, 7, 9, 11\}$, with the default taken as $A = 7$. In contrast to $\mathcal{WASSP}$'s approach for smoothing the periodogram, with Heidelberger and Welch's approach the periodogram is smoothed by taking a moving average with a fixed width of 2 points.

To obtain an estimate of the SSVC of the original (unbatched) process, we compute a wavelet-based estimate of the log-spectrum of the batch means over the frequency range $(-\frac{1}{2}, \frac{1}{2})$ (expressed in cycles per unit time) from the log of the smoothed periodogram of the batch means. The estimated wavelet coefficients are thresholded using a variant of Gao's thresholding algorithm, as described in Section 2.6.

From the thresholded wavelet approximation to the log-spectrum of the batch means, we compute an estimate of the spectrum of the original (unbatched) process at zero frequency (that is, the SSVC); and finally we compute a confidence interval of the form (1.14), where the midpoint of the confidence interval is the average of all the adjacent (nonspaced) batch means that are computed after skipping the initial spacer.

The confidence interval is then tested to determine if it meets a user-specified absolute or relative precision requirement. If the precision requirement is satisfied, then $\mathcal{WASSP}$ delivers the latest confidence interval and terminates. Otherwise, the following steps are executed:

(a) The total required sample size is estimated; and on the assumption that the current batch size is maintained, the estimated batch count is expressed as the largest power of two yielding a total delivered sample size not exceeding the required sample size.

(b) If the estimated batch count exceeds 4,096, then the batch count is reduced to 4,096 and the batch size is adjusted (increased) so that the total delivered sample size is not less than the total required sample size.

(c) The required additional observations are obtained (by restarting the simulation if necessary); and the batch means are recomputed using the latest batch size after skipping the initial spacer.

(d) The log of the smoothed periodogram for the new set of batch means is computed.

(e) A new estimate of the SSVC is obtained from a wavelet-based estimate of the log of the smoothed periodogram of the latest set of batch means.

(f) A confidence interval is computed and the precision requirement is retested.

Note that if the confidence interval in step (f) above fails to meet the precision requirement, then it is not necessary to repeat the independence test or the normality test; instead steps (a)–(f) are repeated until the precision requirement is satisfied.

3.2. Formal Algorithmic Statement of WASSP

WASSP requires the following user-supplied inputs:

(a) a simulation-generated output process $\{X_i : i = 1, 2, \ldots, n\}$ from which the steady-state expected response μ_X is to be estimated;

(b) the desired confidence interval coverage probability $1 - \beta$, where $0 < \beta < 1$; and

(c) an absolute or relative precision requirement specifying the final confidence interval half-length in terms of

- a maximum acceptable half-length h^* (for an absolute precision requirement); or
- a maximum acceptable fraction r^* of the magnitude of the confidence interval midpoint (for a relative precision requirement).

WASSP delivers the following outputs:

(a) a nominal $100(1 - \beta)\%$ confidence interval for μ_X that satisfies the specified absolute or relative precision requirement, provided no additional data are required; or

(b) a new, larger sample size n to be supplied to WASSP when it is executed again.

If additional observations of the target process must be generated by the user's simulation model before a confidence interval with the required precision can be delivered, then WASSP must be executed again with all the observations accumulated so far; and this cycle of simulation followed by automated wavelet-based spectral output analysis may be repeated several times before WASSP finally delivers a confidence interval.

A formal algorithmic statement of WASSP is given on the succeeding pages. In Section 3.3 we describe the steps in the algorithm in more detail.

WASSP Algorithm

[1] Divide the initial sample of size $n \leftarrow 4096$ into $k \leftarrow 256$ batches of size $m \leftarrow 16$. Compute the batch means $\{\overline{X}_j(m) : j = 1, \ldots, k\}$ as in (3.7). Set the initial spacer size $\mathcal{S} \leftarrow 0$. Set the independence test size $\alpha_{\text{ind}} \leftarrow 0.20$. Set the initial normality test size $\alpha_{\text{nor}}(1) \leftarrow 0.05$ and the iteration counter for the normality test $i \leftarrow 0$. Set the smoothing parameter $a \leftarrow 3$ to compute a moving average of width $A \leftarrow 2a + 1$ points for the log-periodogram of the batch means.

[2] Test the k batch means for independence using the von Neumann test (3.9)–(3.11) with level of significance α_{ind}.

[2.1] If the k adjacent batch means pass the independence test, then set $k' \leftarrow k$ and go to **[4]**.

[2.2] Insert a *spacer* of one ignored batch between the $k' \leftarrow k/2$ remaining batch means and update the spacer size to the current batch size, $\mathcal{S} \leftarrow m$.

[2.3] Test the k' spaced batch means for independence using the von Neumann test (3.9)–(3.11) with level of significance α_{ind}. If independence test is passed, then go to **[4]**.

[2.4] Insert another ignored batch into each spacer; update the spacer size and the batch count,

$$\mathcal{S} \leftarrow \mathcal{S} + m \text{ and } k' \leftarrow \left\lfloor \frac{n}{m + \mathcal{S}} \right\rfloor .$$

[2.5] If $k' \geq 25$ then go to **[2.3]**; else go to **[3]**.

[3] Increase the batch size m and update the sample size n according to

$$m \leftarrow \left\lfloor \sqrt{2}m \right\rfloor \quad \text{and} \quad n \leftarrow km;$$

collect additional observations; recompute the adjacent (nonspaced) batch means $\{\overline{X}_j(m) : j = 1, \ldots, k\}$; and go to **[2]**.

[4] Update the iteration counter for the normality test, $i \leftarrow i + 1$. Test the k' approximately i.i.d. batch means for normality using the Shapiro-Wilk test (3.12)–(3.13) with significance level $\alpha_{\text{nor}}(i)$ given by (3.14). If the spaced batch means pass the test for normality, then compute the total accumulated sample size $n \leftarrow k'(\mathcal{S} + m)$ and go to **[6]**.

[5] Increase the batch size m according to

$$m \leftarrow \left\lfloor \sqrt{2}m \right\rfloor ;$$

recompute the spaced batch means using current spacer size $\mathcal{S}$; and go to **[4]**.

WASSP Algorithm (Continued)

[6] Skip the first $\mathcal{S}$ observations in the total accumulated sample of size n. Divide the remaining $n' \leftarrow n - \mathcal{S}$ observations into batches of size m, where the number of batches k is

$$k \leftarrow 2^{\lfloor \log_2(n'/m) \rfloor}.$$

Compute the adjacent (nonspaced) batch means $\{\overline{X}_j(m) : j = 1, \ldots, k\}$.

[7] Compute the periodogram (3.17) of the k batch means; and take

$$I_{\bar{X}(m)}(0) \leftarrow \frac{1}{a}\sum_{l=1}^{a} I_{\bar{X}(m)}\left(\frac{l}{k}\right), \qquad I_{\bar{X}(m)}\left(\frac{1}{2}\right) \leftarrow \frac{1}{a}\sum_{l=1}^{a} I_{\bar{X}(m)}\left(\frac{\frac{k}{2}-l}{k}\right),$$

and

$$I_{\bar{X}(m)}\left(-\frac{l}{k}\right) \leftarrow I_{\bar{X}(m)}\left(\frac{l}{k}\right) \quad \text{for} \quad l = 1, 2, \ldots, \tfrac{k}{2} - 1.$$

Compute the smoothed periodogram $\left\{\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right) : l = 0, \pm 1, \ldots, \pm\left(\frac{k}{2} - 1\right), \frac{k}{2}\right\}$ according to (3.22) based on a moving average of width $A = 2a + 1$ points. Compute the log-smoothed-periodogram,

$$\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) \leftarrow \ln\left[\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \quad \text{for} \quad l = 0, \pm 1, \ldots, \pm\left(\tfrac{k}{2} - 1\right), \tfrac{k}{2}. \tag{3.1}$$

[8] Correct for the bias induced by the log transformation (3.1):

$$\begin{aligned}
\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) &\leftarrow \widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) - [\Psi(a) - \ln(a)] \quad \text{for} \quad l = 0 \quad \text{and} \quad l = \tfrac{k}{2}, \\
\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) &\leftarrow \widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) - \left[\Psi\left(\frac{\nu_{|l|}}{2}\right) - \ln\left(\frac{\nu_{|l|}}{2}\right)\right] \quad \text{for} \quad 1 \le |l| \le a, \\
\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) &\leftarrow \widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) - [\Psi(A) - \ln(A)] \quad \text{for} \quad a < |l| < \tfrac{k}{2} - a, \\
\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) &\leftarrow \widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) - \left[\Psi\left(\frac{\nu_{k/2-|l|}}{2}\right) - \ln\left(\frac{\nu_{k/2-|l|}}{2}\right)\right] \\
&\qquad\qquad \text{for} \quad \tfrac{k}{2} - a \le |l| \le \tfrac{k}{2} - 1,
\end{aligned}$$

where $\Psi(\cdot)$ is the digamma function defined by (3.28); and for each positive integer j with $1 \le j \le a$, the quantity ν_j is defined by (3.41).

WASSP Algorithm (Continued)

[9] Compute the wavelet-based estimate $\widehat{\zeta}_{\bar{X}(m)}(\omega)$ of the batch means log-spectrum $\zeta_{\bar{X}(m)}(\omega)$.

[9.1] Take $J \leftarrow \log_2(k)$; set the number of resolution levels

$$L \leftarrow \left\lfloor \frac{J}{2} \right\rfloor$$

for the wavelet-based estimate $\{\widehat{\zeta}_{\bar{X}(m)}\left(\frac{l}{k}\right) : l = 0, \pm 1, \ldots, \pm\left(\frac{k}{2} - 1\right), \frac{k}{2}\}$ of the log-spectrum $\{\zeta_{\bar{X}(m)}\left(\frac{l}{k}\right)\}$ of the batch means process $\{\overline{X}_j(m) : j = 1, \ldots, k\}$; and set the coarsest level of resolution

$$j_0 \leftarrow J - L.$$

[9.2] Compute $\{\widehat{\zeta}_{\bar{X}(m)}\left(\frac{l}{k}\right)\}$ from the DWT (3.44) of $\{\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\}$ and threshold the resulting wavelet coefficients $\{\widehat{d}_{j,l} : j = j_0, \ldots, J-1; l = 0, 1, \ldots, 2^j - 1\}$ using Gao's thresholding scheme (2.60) to obtain the thresholded coefficients $\{\widehat{d}^*_{j,l}\}$.

[9.3] Perform the inverse transform (3.47) to obtain the thresholded wavelet approximation $\{\widehat{\zeta}_{\bar{X}(m)}\left(\frac{l}{k}\right)\}$ to the log-spectrum of the batch means process.

[10] Compute the wavelet-based estimate of the spectrum of the batch means process at zero frequency,

$$\widehat{p}_{\bar{X}(m)}(0) \leftarrow \exp\left[\widehat{\zeta}_{\bar{X}(m)}(0)\right], \tag{3.2}$$

and the wavelet-based estimate of the SSVC for the original (unbatched) process,

$$\widehat{\gamma}_X \leftarrow m\, \widehat{p}_{\bar{X}(m)}(0).$$

[11] Compute the approximate $100(1-\beta)\%$ confidence interval for μ_X,

$$\overline{\overline{X}}(m,k) \pm t_{1-\beta/2,\, 2a} \sqrt{\frac{\widehat{\gamma}_X}{n'}}, \tag{3.3}$$

where $n' \leftarrow mk$ and $\overline{\overline{X}}(m,k)$ is the grand average of the k batch means $\{\overline{X}_1(m), \ldots, \overline{X}_k(m)\}$ that are computed as in (3.8) after skipping the warm-up period consisting of the first $\mathcal{S}$ observations in the total accumulated sample of size $n = mk + \mathcal{S} = n' + \mathcal{S}$.

WASSP Algorithm (Continued)

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

[12.1] If the half-length

$$H \leftarrow t_{1-\beta/2,\,2a}\sqrt{\frac{\widehat{\gamma}_X}{n'}} = t_{1-\beta/2,\,2a}\sqrt{\frac{\widehat{p}_{\bar{X}(m)}(0)}{k}}$$

of the confidence interval (3.3) satisfies the user-specified precision requirement

$$H \leq H^*, \tag{3.4}$$

where

$$H^* \leftarrow \begin{cases} \infty, & \text{for no user-specified precision level,} \\ r^*|\overline{\overline{X}}(m,k)|, & \text{for a user-specified relative precision level } r^*, \\ h^*, & \text{for user-specified absolute precision level } h^*, \end{cases} \tag{3.5}$$

then deliver the confidence interval (3.3) and stop.

[12.2] Set

$$k^* \leftarrow \left\lceil \left(\frac{H}{H^*}\right)^2 k \right\rceil.$$

[12.3] Update the number of batch means k, the batch size m, and the total sample size n as follows:

$$\begin{aligned} k &\leftarrow \min\left\{2^{\lfloor \log_2(k^*) \rfloor}, 4096\right\}, \\ m &\leftarrow \left\lceil \left(\frac{k^*}{k}\right) m \right\rceil, \\ n &\leftarrow km + \mathcal{S}. \end{aligned} \tag{3.6}$$

[12.4] Obtain the additional simulation-generated observations (by restarting the simulation if necessary), and go to [**6**].

3.3. Detailed Description of Steps in WASSP

3.3.1 *Test for Independence of the Batch Means*

The WASSP algorithm begins by dividing the initial sample $\{X_i : i = 1, \ldots, n\}$ into $k = 256$ batches of size $m = 16$. The minimum number of data points required by WASSP is 4,096, which is the same initial sample size required by ASAP2. Based on our experimental results using WASSP as well as on our results using ASAP2, we have found that this initial sample size works well in practice. Furthermore, for any large-scale simulation study, it is relatively inexpensive to generate a sample of size 4,096. A batch mean for each of the batches is computed. Let

$$\overline{X}_j = \overline{X}_j(m) = \frac{1}{m} \sum_{i=m(j-1)+1}^{mj} X_i \tag{3.7}$$

denote the jth batch mean for $j = 1, \ldots, k$; and let

$$\overline{\overline{X}}(m,k) = \frac{1}{k} \sum_{j=1}^{k} \overline{X}_j(m) \tag{3.8}$$

denote the grand average of the k batch means.

The batch means $\{\overline{X}_1(m), \ldots, \overline{X}_k(m)\}$ are tested for independence using the von Neumann test. The hypothesis of independent, identically distributed batch means,

$$\mathcal{H}_{\text{ind}} : \{\overline{X}_j(m) : j = 1, \ldots, k\} \text{ are i.i.d.}, \tag{3.9}$$

is tested by computing the test statistic,

$$C_k = 1 - \frac{\sum_{j=1}^{k-1} \left[\overline{X}_j(m) - \overline{X}_{j+1}(m)\right]^2}{2\sum_{i=1}^{k} \left[\overline{X}_i(m) - \overline{\overline{X}}(m,k)\right]^2}, \tag{3.10}$$

where C_k is a relocated and rescaled version of the ratio of the mean square successive difference to the variance of the batch means. If $\{\overline{X}_1(m), \ldots, \overline{X}_k(m)\}$ are normal, then under H_0, the statistic C_k has zero mean, variance $(k-2)/(k^2-1)$, and a distribution that is close to normal for $k \geq 8$. If the batch means are nonnormal but i.i.d., then the distribution of $C_k/\sqrt{(k-2)/(k^2-1)}$ can be approximated by a normal distribution with mean 0 and variance 1 for $k \geq 20$ [72]. Since WASSP's test for independence always involves at least 25 batch means,

the hypothesis (3.9) can be tested as follows. If

$$|C_k| \leq z_{1-\alpha_{\text{ind}}/2}\sqrt{\frac{k-2}{k^2-1}}, \tag{3.11}$$

where $z_{1-\alpha_{\text{ind}}/2}$ is the $1-\alpha_{\text{ind}}/2$ quantile of the standard normal distribution, then the hypothesis (3.9) is accepted; otherwise the hypothesis (3.9) is rejected so that **WASSP** must increase the spacer size before retesting (3.9).

Each time the test for independence is repeated, there is a certain probability of making a type II error (that is, falsely declaring the batch means to be i.i.d. when they are not). Therefore, it is necessary to carefully select α_{ind} so that the probability of making a type II error is small, while at the same time the probability of a type I error (falsely declaring the batch means to be dependent or nonstationary when they are in fact i.i.d.) is also at a reasonable level. If α_{ind} is too small, then the probability of making a type II error becomes high. Since **WASSP** uses the results of the independence test to determine the length of the warm-up period, this could lead to an inaccurate identification of the warm-up period, ultimately leading to biased point and confidence-interval estimators of μ_X. On the other hand if α_{ind} is too large, then the risk of making a type I error becomes high and it is possible that extremely large sample sizes will be required to pass the independence test. Through extensive experimentation, we found that setting $\alpha_{\text{ind}} = 0.2$ works well in practice and provides an effective balance between type I and type II errors.

If the $k = 256$ adjacent batch means defined by (3.7) pass the independence test (3.9)–(3.11) at the level of significance α_{ind}, then we set k', the number of batch means retained for the normality test, equal to 256; and we proceed to perform the normality test as detailed in Subsection 3.3.2 below. On the other hand, if the $k = 256$ batch means fail the test for independence, then a *spacer* consisting of one ignored batch is inserted between the $k' = 128$ remaining batch means that are to be retested for independence. That is, every other batch mean, beginning with the second batch mean, is retained and the alternate batch means are discarded as depicted in Figure 3.2.

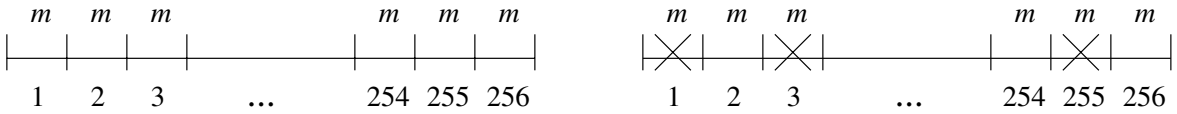

Figure 3.2: Left-hand figure depicts the original 256 batches of size m. Right-hand figure depicts $k' = 128$ batches of size m with interleaved spacers, each consisting of 1 ignored batch so that the spacer size $\mathcal{S} = m$. The X-marks denote batches that compose each spacer. The spaced batch means are computed from the unmarked batches.

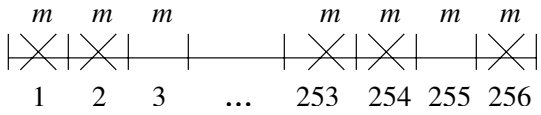

Figure 3.3: Depiction of $k' = 85$ batches of size m with spacers consisting of 2 batches so that $\mathcal{S} = 2m$. The X-marks denote adjacent batches that compose each spacer. The spaced batch means are computed from the unmarked batches.

As illustrated in the right-hand part of Figure 3.2, the $k' = 128$ remaining batch means $\{\overline{X}_2(m), \overline{X}_4(m), \ldots, \overline{X}_{256}(m)\}$ are retested for independence. If the test is failed, then another ignored batch is added to the spacer (that is, every third batch mean is retained) and the $k' = 85$ remaining batch means $\{\overline{X}_3(m), \overline{X}_6(m), \ldots, \overline{X}_{255}(m)\}$ are retested for independence as illustrated in Figure 3.3. This process is continued until either the independence test is passed or the number of batch means reaches the lower limit $k' = 25$ (that is, the point has been reached where every tenth batch mean from the original $k = 256$ batch means has been retained). In order for the independence test to work properly (that is, in order to use the standard normal distribution as an approximation to the distribution of the test statistic $C_k/\sqrt{(k-2)/(k^2-1)}$ under the null hypothesis (3.9)), a minimum of about 20 batch means is required. If the number of retained batch means drops to $k' = 25$ and the batch means still fail the independence test, then the batch size m is increased by a factor of $\sqrt{2}$,

$$m \longleftarrow \left\lfloor \sqrt{2}m \right\rfloor;$$

the initial sample $\{X_i : i = 1, \ldots, n\}$ is rebatched into $k = 256$ batches of size m; and the k batch means are recomputed. If $n < km$, then more data must be collected and n must be updated before the batch means can be recomputed and retested for independence.

The process of testing for independence is repeated, starting with the $k = 256$ recomputed batch means. If the independence test is failed, then additional batches are added iteratively to the spacers lying between the batches from which we compute the set of k' retained batch means (as described earlier) until either the independence test is passed or the number of retained batch means reaches $k' = 25$. If the independence test is failed for $k' = 25$, then the batch size m is increased by a factor of $\sqrt{2}$ again; the original sample $\{X_i : i = 1, \ldots, n\}$ is rebatched into $k = 256$ batches of size m; additional data is collected if necessary; and the test for independence is repeated. We found in practice that it is not necessary to double the total sample size every time the independence test reaches the point that it becomes necessary to increase the batch size (that is, when $k' = 25$ and the independence test is failed). We found that it is appropriate to double the sample size every other time the batch size needs to be increased, implying that the batch size should be increased by a factor of $\sqrt{2}$ on each iteration

of this step in $\mathcal{WASSP}$.

Once the independence test is passed, there will be a set of k' approximately independent batch means, where $25 \leq k' \leq 256$. At this point, the *spacer length* $\mathcal{S}$ is computed. The spacer length is the number of *observations* in each spacer lying between the batches from which we compute the set of k' retained batch means. The spacer length is computed as follows,

$$\mathcal{S} = ms,$$

where s is the number of batches in each spacer. For example, if the final number of retained batch means is $k' = 85$, then there are $s = 2$ batches in the spacer preceding each retained batch. The first $\mathcal{S}$ observations $\{X_1, \ldots, X_{\mathcal{S}}\}$ will comprise the warm-up period. We justify this formulation of the warm-up period as follows. The von Neumann test checks the batch means for randomness. Those batch means computed from observations comprising the warm-up period will exhibit a significant deterministic trend or a significant degree of stochastic dependence. Once the simulation reaches steady-state operation, the batch means will no longer exhibit a trend; and provided the batch means are approximately independent, they will appear to be more nearly random. If the batch means exhibit a trend, then it is likely that the trend will be detected by the von Neumann test and the hypothesis (3.9) will be rejected. Each time the test for independence is failed, another batch is added to the spacer preceding each retained batch.

Once the spaced batch means pass the independence test (3.9)–(3.11), two conclusions can be deduced.

- First, the observations $\{X_1, \ldots, X_{\mathcal{S}}\}$ used to compute the batch means $\{\overline{X}_1(m), \overline{X}_2(m), \ldots, \overline{X}_s(m)\}$ comprising the first spacer can be regarded as the warm-up period since the retained batch means following the first spacer do not exhibit a deterministic trend or any type of stochastic dependence on the simulation's initial conditions.

- Second, the retained batch means after the warm-up period are randomly sampled from a common distribution. That is, batch means computed from batches separated by a spacer consisting of $\mathcal{S}$ successive observations are approximately i.i.d.

Figure 3.4 depicts a typical transient mean function for a general, simulation-generated output process, along with the batch means computed from a possible realization of this output process. The trend (nonstationarity) of the batch means $\overline{X}_1(m), \ldots, \overline{X}_s(m)$ computed from observations in the warm-up period can be clearly seen. Figure 3.4 also shows the first three retained batch means $\overline{X}_{s+1}(m), \overline{X}_{2s+2}(m)$, and $\overline{X}_{3s+3}(m)$, each preceded by a spacer consisting of s batch means.

One of the advantages of $\mathcal{WASSP}$ over Heildelberger and Welch's spectral method is that $\mathcal{WASSP}$ provides an automatic scheme for effectively identifying an appropriate warm-up period,

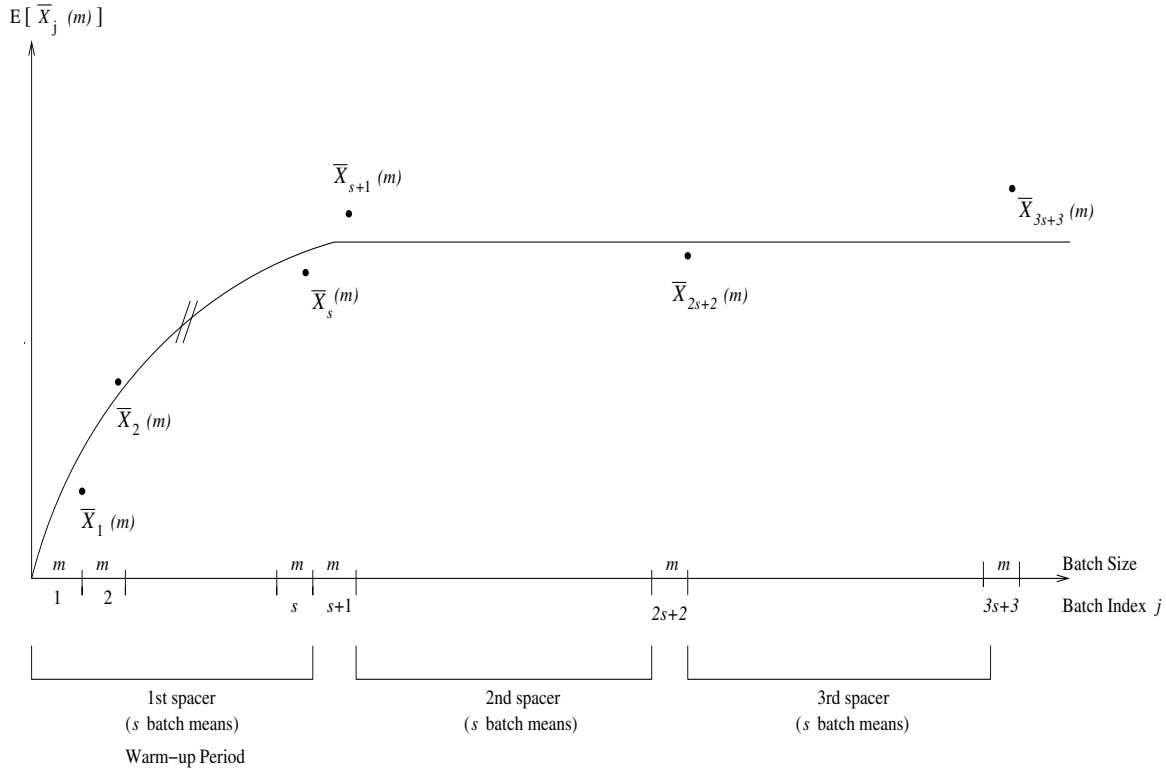

Figure 3.4: Depiction of the transient mean function $\left\{\mathrm{E}\left[\overline{X}_j(m)\right] : j = 1, 2, \ldots\right\}$ for a general output process and the batch means computed from one realization of the output process.

thereby ensuring the delivered confidence interval is not affected by system warm-up bias. Chapter 4 provides substantial experimental evidence of the effectiveness of WASSP's scheme for eliminating the effects of initialization bias.

3.3.2 *Test for Normality of the Batch Means*

Because the Shapiro-Wilk test for normality [55] requires a data set consisting of i.i.d. observations, we apply this test to the k' batch means that were retained after passing the preceding test for independence. To assess the normality of the sample $\{\overline{X}_1(m), \ldots, \overline{X}_{k'}(m)\}$, we start by sorting the observations in ascending order to obtain the order statistics

$$\overline{X}_{(1)}(m) \leq \overline{X}_{(2)}(m) \leq \cdots \leq \overline{X}_{(k')}(m).$$

The Shapiro-Wilk test statistic is then computed as follows,

$$W = \frac{\left\{ \sum_{l=1}^{\lfloor k'/2 \rfloor} \delta_{k'-l+1} \left[\overline{X}_{(k'-l+1)}(m) - \overline{X}_{(l)}(m) \right] \right\}^2}{\sum_{l=1}^{k'} \left[\overline{X}_l(m) - \overline{\overline{X}}(m,k) \right]^2}, \tag{3.12}$$

where the coefficients $\{\delta_{k'-l+1} : l = 1, \ldots, \lfloor k'/2 \rfloor\}$ are tabulated in [55] and [19]. The test statistic W is then compared to the appropriate lower $100\alpha_{\text{nor}}\%$ critical value $w_{\alpha_{\text{nor}}}$ of the distribution of W under the null hypothesis of i.i.d. normal batch means,

$$\mathcal{H}_{\text{nor}} : \{\overline{X}_j(m) : j = 1, \ldots, k'\} \overset{\text{i.i.d.}}{\sim} N(\mu_X, \sigma^2_{\overline{X}(m)}). \tag{3.13}$$

If $W \leq w_{\alpha_{\text{nor}}}$, then at the level of significance α_{nor} we reject the hypothesis $\mathcal{H}_{\text{nor}}$ that the retained batch means $\{\overline{X}_j(m) : j = 1, \ldots, k'\}$ are normal.

For the first iteration of the normality test, the iteration counter is set to $i \leftarrow 1$ and the level of significance for the Shapiro-Wilk test is $\alpha_{\text{nor}}(1) = 0.05$. In general if on the ith iteration of the normality test (3.12)–(3.13) the hypothesis (3.13) is rejected at the level of significance $\alpha_{\text{nor}}(i)$, then the following steps are taken:

(a) The iteration counter i is increased,

$$i \leftarrow i + 1,$$

and the batch size m is increased by a factor of $\sqrt{2}$,

$$m \leftarrow \left\lfloor \sqrt{2} m \right\rfloor.$$

(b) The overall data set $\{X_1, \ldots, X_n\}$ is redivided into k' batches of size m so that each batch of size m is separated by $\mathcal{S}$ observations, where the spacer size $\mathcal{S}$ was fixed in the preceding test for independence; and if necessary, additional simulation-generated data is collected to allow computation of k' spaced batch means with the new batch size m and the fixed spacer size $\mathcal{S}$.

(c) The spaced batch means $\{\overline{X}_j(m) : j = 1, \ldots, k'\}$ are computed.

(d) The level of significance $\alpha_{\text{nor}}(i)$ for the current iteration i of the Shapiro-Wilk test is set according to

$$\alpha_{\text{nor}}(i) \leftarrow \alpha_{\text{nor}}(1) \exp[-\tau(i-1)^2] \quad \text{for} \quad i = 1, 2, \ldots, \tag{3.14}$$

where $\tau = 0.184206$. Table 3.1 lists the values of $\alpha_{\text{nor}}(i)$ for $i = 1, \ldots, 6$. Note that for $i = 7, 8, \ldots$, the significance level $\alpha_{\text{nor}}(i)$ is decreased by at least an order of magnitude on each additional iteration of the normality test.

(e) The k' spaced batch means $\{\overline{X}_j(m) : j = 1, \ldots, k'\}$ are tested for normality at the level of significance $\alpha_{\text{nor}}(i)$.

If the normality hypothesis (3.13) is rejected in step (e), then steps (a)–(e) above are repeated until the hypothesis is accepted.

Table 3.1: Level of significance $\alpha_{\text{nor}}(i)$ for the ith iteration of the Shapiro-Wilk normality test (3.12)–(3.13).

i	$\alpha_{\text{nor}}(i)$
1	0.050
2	0.042
3	0.024
4	0.0095
5	0.0026
6	0.0005
≥ 7	$< \alpha_{\text{nor}}(i-1)/10$

The scheme in step (d) above for setting $\alpha_{\text{nor}}(i)$ on each iteration of the normality test is specifically designed so that **WASSP** avoids excessive variability in the final sample size. We have seen through extensive experimentation that beginning with $\alpha_{\text{nor}}(1) = 0.05$ and decreasing $\alpha_{\text{nor}}(i)$ on each iteration of the normality test according to (3.14) greatly reduces the final sample size required by **WASSP**, especially in those cases where the data are highly nonnormal. Furthermore, we have found that **WASSP** performs well even when there is a mild departure from normality in the resulting batch means. Therefore, decreasing the level of significance (3.14) of the normality test (3.12)–(3.13) on each iteration i effectively controls the required sample size without adversely affecting the performance of **WASSP**.

One of the main differences in the **WASSP** spectral method and Heidelberger and Welch's spectral method is that **WASSP** always requires normalization of the output process via batching (aggregation) with a batch size sufficient to induce approximate marginal normality of the batch means before the periodogram of the batch means is constructed. One advantage of performing this extra step is that the periodogram is in general more well behaved for normal data [47], thereby facilitating the process of obtaining an estimate of the log-spectrum of the batch means in a neighborhood of zero frequency. Furthermore, convergence of the periodogram to its asymptotic properties (listed in equation (2.51)) is faster for Gaussian processes [47]. Quick

convergence of the periodogram of the batch means to its asymptotic properties is important because those properties will be used to compute an estimate of the bias that is introduced when we apply the logarithmic transformation to the smoothed periodogram of the batch means. Ultimately this bias term will be subtracted from the log-smoothed-periodogram of the batch means to yield an approximately unbiased estimator of the log-spectrum of the batch means. The computation of this bias term will be described in the next section.

3.3.3 *Estimation of the Log-Spectrum of the Batch Means*

Once the normality test is passed, independence of the batch means is no longer required. Therefore, the first spacer consisting of the observations $\{X_1, X_2, \ldots, X_{\mathcal{S}}\}$ is deleted (to account for the warm-up period), and the remaining $n' = n - \mathcal{S}$ observations are rebatched into k adjacent (nonspaced) batches of size m. To construct the wavelet-based estimate of the log-spectrum of the batch means in a neighborhood of zero frequency, we see that the number of points in the log-periodogram (that is, the number of batch means k) must be a power of two. Therefore, k is set to the largest power of two less than or equal to n'/m,

$$k = 2^{\lfloor \log_2(n'/m) \rfloor}, \tag{3.15}$$

where m is the final batch size required for the batch means to pass the normality test. For $j = 1, \ldots, k$, the jth batch mean $\overline{X}_j(m)$ is computed. The next step in **WASSP** is to obtain an estimate of the log-spectrum of the batch means in a neighborhood of zero frequency by computing a wavelet-based estimate of the batch means log-spectrum from the log of the smoothed periodogram of the batch means process $\{\overline{X}_1(m), \ldots, \overline{X}_k(m)\}$.

The periodogram of the batch means process is computed by taking the fast Fourier transform of $\boldsymbol{X} = \{\overline{X}_1(m), \ldots, \overline{X}_k(m)\}$,

$$(\mathcal{F}\boldsymbol{X})_l = \sum_{j=1}^{k} \overline{X}_j(m) \exp\Big[-2\pi(\sqrt{-1})(j-1)l/k\Big] \quad \text{for} \quad l = 1, 2, \ldots, k-1. \tag{3.16}$$

Since we will be interested in obtaining an estimate of the log-spectrum of the batch means in a neighborhood of zero frequency using the values of the log-smoothed-periodogram of the batch means in a neighborhood of zero, we will use a full period of the periodogram. The periodogram is symmetric, so that for $l = 1, 2, \ldots, \frac{k}{2} - 1$,

$$\begin{aligned} I_{\bar{X}(m)}\Big(\tfrac{l}{k}\Big) &= I_{\bar{X}(m)}\Big(-\tfrac{l}{k}\Big) = \tfrac{1}{k}|(\mathcal{F}\boldsymbol{X})_l|^2 \qquad (3.17) \\ &= \frac{1}{k}\left\{\left[\sum_{j=1}^{k} \overline{X}_j(m) \cos\Big(\frac{2\pi(j-1)l}{k}\Big)\right]^2 + \left[\sum_{j=1}^{k} \overline{X}_j(m) \sin\Big(\frac{2\pi(j-1)l}{k}\Big)\right]^2\right\}. \end{aligned}$$

To compute the smoothed periodogram of the batch means based on a moving average of $A = 2a+1$ periodogram values, we first must determine appropriate values for the periodogram at $l = 0$ and $l = \frac{k}{2}$. Using the definition (3.17), we see that the value of the periodogram at $l = 0$ is simply a scaled sum of the batch means and provides no information about the spectrum of the batch means at zero frequency. As an alternative, we take the value of the periodogram at $l = 0$ as follows,

$$I_{\bar{X}(m)}(0) \equiv \frac{1}{a}\sum_{l=1}^{a} I_{\bar{X}(m)}\left(\frac{l}{k}\right). \tag{3.18}$$

We will assume that for $l \neq 0$ and a sufficiently small relative to k, the periodogram values $\{I_{\bar{X}(m)}(\frac{l+u}{k}) : u = 0, \pm 1, \ldots, \pm a\}$ have expected values approximately equal to $p_{\bar{X}(m)}\left(\frac{l}{k}\right)$. In particular for $l = 0$, we assume that the periodogram values $\{I_{\bar{X}(m)}(\frac{l+u}{k}) : u = \pm 1, \ldots, \pm a\}$ have expected values approximately equal to $p_{\bar{X}(m)}(0)$. That is,

$$\mathrm{E}\left[I_{\bar{X}(m)}\left(\frac{l+u}{k}\right)\right] \approx p_{\bar{X}(m)}\left(\frac{l}{k}\right), \text{ where } u = \begin{cases} 0, \pm 1, \pm 2, \ldots, \pm a & \text{for } l = 1, \ldots, \frac{k}{2} - 1 \\ \pm 1, \pm 2, \ldots, \pm a & \text{for } l = 0. \end{cases} \tag{3.19}$$

Setting $l = 0$ and using (3.19), we have

$$\mathrm{E}\left[I_{\bar{X}(m)}(0)\right] = \frac{1}{a}\sum_{l=1}^{a} \mathrm{E}\left[I_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \approx \frac{1}{a}\sum_{l=1}^{a} p_{\bar{X}(m)}(0) = p_{\bar{X}(m)}(0);$$

and therefore we see (3.18) is an approximately unbiased estimate of the spectrum of the batch means at zero frequency, provided a is not too large relative to k.

On the basis of considerations paralleling those leading to the revised definition (3.18) of the periodogram at zero frequency, we make the following definition of the periodogram at frequency $\frac{1}{2}$:

$$I_{\bar{X}(m)}\left(\frac{1}{2}\right) \equiv \frac{1}{a}\sum_{l=1}^{a} I_{\bar{X}(m)}\left(\frac{\frac{k}{2} - l}{k}\right). \tag{3.20}$$

We define the periodogram at frequency $\frac{1}{2}$ for the sole purpose of facilitating wavelet-based estimation of the log-spectrum of the batch means, as described in Section 3.3.4.

Since the spectrum of the batch means is symmetric about zero on the interval $[-\frac{1}{2}, \frac{1}{2}]$ with period 1, we see that

$$I_{\bar{X}(m)}\left(\frac{k/2 + l}{k}\right) = I_{\bar{X}(m)}\left(\frac{k/2 - l}{k}\right) \quad \text{for } l = 1, 2, \ldots, (\tfrac{k}{2} - 1); \tag{3.21}$$

and therefore the smoothed periodogram of the batch means $\{\widetilde{I}_{\bar{X}(m)}(\frac{l}{k}) : l = -(\frac{k}{2}-1), \ldots, 0, \ldots,$

$\frac{k}{2}-1, \frac{k}{2}\}$ can be computed using a moving average of $A = 2a + 1$ points according to,

$$\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right) = \frac{1}{A}\sum_{u=-a}^{a} I_{\bar{X}(m)}\left(\frac{l+u}{k}\right) \quad \text{for} \quad l = 0, \pm 1, \pm 2, \ldots, \pm\left(\tfrac{k}{2}-1\right), \tfrac{k}{2}. \tag{3.22}$$

WASSP allows the user to select the value of the smoothing parameter A from the set of values $\{5, 7, 9, 11\}$. The default value of A is 7. If the smoothing parameter is set too high, then the log-smoothed-periodogram of the batch means will be oversmoothed, resulting in an estimate that is flatter in the neighborhood of zero frequency than the true log-spectrum of the batch means process. If A is too small, then the log-smoothed-periodogram of the batch means will not be smoothed enough, resulting in an excessively noisy estimate of the log-spectrum of the batch means. We found through extensive experimentation that setting $A = 7$ works well for a variety of problems. There are instances, however, where it may be necessary to set A smaller or larger than 7. For example, certain simulation-generated output processes have a power spectrum with a very sharp peak in the neighborhood of zero frequency. In order to approximate adequately the sharpness of the peak, it may be necessary to set the smoothing parameter A to 5.

Another justification for setting the default smoothing parameter $A = 7$ is the following: at zero frequency, the log-smoothed-periodogram has $2a$ degrees of freedom (this result is derived in equation (3.25)). Therefore, if $A = 7$ then $a = 3$, implying we have 6 degrees of freedom at zero frequency. This is comparable to what Heidelberger and Welch found in [26]. For their spectral output analysis method, they found that their best-performing estimate of the log-spectrum at zero frequency has 7 degrees of freedom. Presumably, based on Heidelberger and Welch's results, any improved spectral estimator should have approximately 7 degrees of freedom. Therefore, selecting $A = 7$ as the default smoothing parameter in **WASSP** seems to provide an acceptable balance between the degrees of freedom associated with the final estimate of the SSVC of the original (unbatched) process and the amount of smoothing required to capture the important features of the underlying log-spectrum of the batch means.

It is also worth noting at this point that the smoothed periodogram cannot be computed using a moving average of $A = 2a$ periodogram values. If the smoothing parameter A is even, then the smoothed periodogram $\widetilde{I}_{\bar{X}(m)}(\frac{l}{k})$ will be associated with frequencies for which the index $l = \pm\frac{1}{2}, \pm\frac{3}{2}, \pm\frac{5}{2}, \ldots, \pm\frac{k-3}{2}$. If we then add a point at zero frequency using (3.18), however, we will no longer have equally spaced function values on both sides of zero frequency. Since we will ultimately be computing the discrete wavelet transform of the log of the smoothed periodogram of the batch means (as described in Section 3.3.4), the points in the log of the smoothed periodogram of the batch means must be equally spaced. Such a condition is satisfied only when the smoothing parameter A is odd.

The natural log of the smoothed periodogram of the batch means, $\{\widetilde{\mathcal{L}}_{\bar{X}(m)}(\frac{l}{k}) : l = 0, \pm 1, \ldots,$

$\pm(\frac{k}{2}-1), \frac{k}{2}\}$, is used as an estimator of the log-spectrum $\{\zeta_{\bar{X}(m)}(\frac{l}{k}) : l = 0, \pm 1, \ldots, \pm(\frac{k}{2}-1), \frac{k}{2}\}$ of the corresponding batch means process. However, upon applying the properties of the periodogram in (2.51), we find that the expected value of our estimator is

$$\begin{aligned} \mathrm{E}\left\{\ln\left[\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]\right\} &\approx \mathrm{E}\left\{\ln\left[\sum_{u=-a}^{a} p_{\bar{X}(m)}\left(\frac{l}{k}\right) w_u \frac{\chi_u^2(2)}{2}\right]\right\} \\ &= \mathrm{E}\left\{\ln\left[p_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] + \ln\left[\frac{1}{2}\sum_{u=-a}^{a} w_u \chi_u^2(2)\right]\right\} \\ &= \zeta_{\bar{X}(m)}\left(\frac{l}{k}\right) + \mathrm{E}\left\{\ln\left[\frac{1}{2}\sum_{u=-a}^{a} w_u \chi_u^2(2)\right]\right\}, \end{aligned} \tag{3.23}$$

where $\{\chi_u^2(2) : u = 0, \pm 1, \ldots, \pm a\}$ are i.i.d. chi-square random variables with 2 degrees of freedom and the $\{w_u : u = 0, \pm 1, \ldots, \pm a\}$ are nonnegative deterministic weights such that $\sum_{u=-a}^{a} w_u = 1$. Therefore, when we take the log of the smoothed periodogram, bias is introduced and must be removed.

Specifically, at zero frequency so that $l = 0$ and with smoothing parameter $A = 2a + 1$, we have

$$\widetilde{I}_{\bar{X}(m)}(0) = \frac{1}{A}\sum_{u=-a}^{a} I_{\bar{X}(m)}\left(\frac{u}{k}\right) = \frac{1}{a}\sum_{u=1}^{a} I_{\bar{X}(m)}\left(\frac{u}{k}\right), \tag{3.24}$$

since the sequence $\{I_{\bar{X}(m)}\left(\frac{l}{k}\right) : l = -(\frac{k}{2}-1), \ldots, 0, \ldots, \frac{k}{2}-1\}$ is symmetric with respect to $l = 0$ and $I_{\bar{X}(m)}(0)$ is as given in (3.18). Applying the properties of the periodogram in (2.51), we have

$$\begin{aligned} \widetilde{I}_{\bar{X}(m)}(0) &\dot{\sim} \frac{1}{a}\sum_{u=1}^{a} p_{\bar{X}(m)}(0)\,\frac{\chi_u^2(2)}{2} \\ &\sim p_{\bar{X}(m)}(0)\,\frac{\chi^2(2a)}{2a}, \end{aligned} \tag{3.25}$$

where $\chi^2(2a)$ denotes a chi-square random variable with $2a$ degrees of freedom. Taking the natural log of (3.25), we have

$$\begin{aligned} \widetilde{\mathcal{L}}_{\bar{X}(m)}(0) &= \ln[\widetilde{I}_{\bar{X}(m)}(0)] \\ &\dot{\sim} \ln\left[p_{\bar{X}(m)}(0)\,\frac{\chi^2(2a)}{2a}\right] \\ &= \ln[p_{\bar{X}(m)}(0)] + \ln\left[\frac{\chi^2(2a)}{2a}\right]. \end{aligned} \tag{3.26}$$

Therefore, our estimate of the log-spectrum of the batch means at zero frequency has the

following expected value,

$$\begin{aligned} \mathrm{E}\Big[\widetilde{\mathcal{L}}_{\bar{X}(m)}(0)\Big] &= \mathrm{E}\Big\{\ln\Big[p_{\bar{X}(m)}(0)\Big]\Big\} + \mathrm{E}\left[\ln\left(\frac{\chi^2(2a)}{2a}\right)\right] \\ &= \zeta_{\bar{X}(m)}(0) + \Psi(a) - \ln(a); \end{aligned} \tag{3.27}$$

where given the gamma function,

$$\Gamma(z) \equiv \int_0^\infty t^{z-1}\, e^{-t}\, dt \;\; \text{for all } z \text{ with } \mathrm{Re}(z) > 0,$$

the digamma function is defined as follows,

$$\Psi(z) \equiv \frac{d}{dz}\, \ln\left[\,\Gamma(z)\,\right] = \frac{\Gamma'(z)}{\Gamma(z)} \tag{3.28}$$

[2]. The term

$$\Psi(a) - \ln(a) \tag{3.29}$$

is the bias at zero frequency that is introduced by taking the log of the smoothed periodogram of the batch means. The result in (3.27) correponds to taking $w_u = 1/a$ for $u = 1, 2, \ldots, a$ and $w_u = 0$ for $u = -a, -a+1, \ldots, 0$ in equation (3.23).

A similar analysis at the frequency $\frac{1}{2}$ so that $l = \frac{k}{2}$ yields

$$\begin{aligned} \mathrm{E}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{1}{2}\right)\right] &= \mathrm{E}\left\{\ln\left[p_{\bar{X}(m)}\left(\frac{1}{2}\right)\right]\right\} + \mathrm{E}\left[\ln\left(\frac{\chi^2(2a)}{2a}\right)\right] \\ &= \zeta_{\bar{X}(m)}\left(\frac{1}{2}\right) + \Psi(a) - \ln(a); \end{aligned} \tag{3.30}$$

and therefore the bias term at the frequency $\frac{1}{2}$ is the same as the bias term for zero frequency (as given in display (3.29)). A complete derivation of the bias term (3.29) is given in Section A.1 of Appendix A.

Similarly, at frequency $\frac{l}{k}$ for $a < |l| < \frac{k}{2} - a$, we have

$$\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) = \ln\left[\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \;\dot{\sim}\; \ln\left[p_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] + \ln\left[\frac{\chi^2(2A)}{2A}\right], \tag{3.31}$$

and

$$\mathrm{E}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \approx \mathrm{E}\left\{\ln\left[p_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]\right\} + \mathrm{E}\left[\ln\left(\frac{\chi^2(2A)}{2A}\right)\right]$$

$$= \quad \zeta_{\bar{X}(m)}\left(\frac{l}{k}\right) + \Psi(A) - \ln(A). \tag{3.32}$$

Therefore, the bias term for frequency $\frac{l}{k}$ where $a < |l| < \frac{k}{2} - a$ is

$$\Psi(A) - \ln(A). \tag{3.33}$$

The result in (3.32) corresponds to setting $w_u = 1/A$ for $u = 0, \pm 1, \ldots, \pm a$ in (3.23). A complete derivation of the bias term (3.33) is given in Section A.2 of Appendix A.

At frequency $\frac{l}{k}$ for $1 \leq |l| \leq a$, we see that $\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)$ is in general a weighted average of independent chi-square random variables in which the weights are all positive constants that are not necessarily equal in value. The results of Satterthwaite [53] and Welch [67] ensure that an excellent approximation to the distribution of $\widetilde{I}_{\bar{X}(m)}(\frac{l}{k})$ is given by

$$\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right) \;\dot{\sim}\; \frac{\chi^2(\nu_{|l|})}{\nu_{|l|}}\, p_{\bar{X}(m)}\left(\frac{l}{k}\right), \tag{3.34}$$

where $\nu_{|l|}$, the "effective" degrees of freedom, is as computed in equation (A.14) of Appendix A; see also display (3.41) below. The log of the smoothed periodogram of the batch means can then be written as follows,

$$\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) = \ln\left[\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \;\dot{\sim}\; \ln\left[p_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] + \ln\left[\frac{\chi^2(\nu_{|l|})}{\nu_{|l|}}\right]. \tag{3.35}$$

Furthermore,

$$\begin{aligned} \mathrm{E}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] &\approx \mathrm{E}\left\{\ln\left[p_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]\right\} + \mathrm{E}\left[\ln\left(\frac{\chi^2(\nu_{|l|})}{\nu_{|l|}}\right)\right] \\ &= \zeta_{\bar{X}(m)}\left(\frac{l}{k}\right) + \Psi\left(\frac{\nu_{|l|}}{2}\right) - \ln\left(\frac{\nu_{|l|}}{2}\right), \end{aligned} \tag{3.36}$$

and the bias term for frequency $\frac{l}{k}$ where $1 \leq |l| \leq a$ is

$$\Psi\left(\frac{\nu_{|l|}}{2}\right) - \ln\left(\frac{\nu_{|l|}}{2}\right). \tag{3.37}$$

Section A.3 of Appendix A has the complete derivation of the bias term (3.37).

Finally, at frequency $\frac{l}{k}$ for $\frac{k}{2} - a \leq |l| \leq \frac{k}{2} - 1$, we see that as in the previous case, $\widetilde{I}_{\bar{X}(m)}(\frac{l}{k})$ is in general a weighted average of independent chi-square random variables. As in (3.34), we assert that

$$\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right) \;\dot{\sim}\; \frac{\chi^2(\nu_{k/2-|l|})}{\nu_{k/2-|l|}}\, p_{\bar{X}(m)}\left(\frac{l}{k}\right), \tag{3.38}$$

where $\nu_{k/2-|l|}$, the effective degrees of freedom, is as computed in equation (A.14) of Appendix A; see also display (3.41) below. Paralleling the computation of the bias term at frequency $\frac{l}{k}$ for $1 \le |l| \le a$, we find that the bias term for $\frac{k}{2} - a \le |l| \le \frac{k}{2} - 1$ is

$$\Psi\left(\frac{\nu_{k/2-|l|}}{2}\right) - \ln\left(\frac{\nu_{k/2-|l|}}{2}\right). \tag{3.39}$$

Section A.4 of Appendix A has the complete derivation of the bias term (3.39). Table 3.2 shows the values of the bias terms (3.29), (3.33), (3.37), and (3.39) for smoothing parameter values $A = \{5, 7, 9, 11\}$ and frequency $\frac{l}{k}$ where $l = 0, \pm 1, \ldots, \pm(\frac{k}{2} - 1), \frac{k}{2}$.

Table 3.2: Bias $\mathrm{E}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] - \zeta_{\bar{X}(m)}\left(\frac{l}{k}\right)$ for smoothing parameter $A = 5$, 7, 9, and 11 and for frequency $\frac{l}{k}$, where $l = 0, \pm 1, \ldots, \pm\left(\frac{k}{2} - 1\right), \frac{k}{2}$.

l	$A = 5$	$A = 7$	$A = 9$	$A = 11$
0	-0.2704	-0.1758	-0.1302	-0.1033
± 1	-0.2131	-0.1496	-0.1152	-0.0937
± 2	-0.1496	-0.1302	-0.1033	-0.0856
± 3		-0.0937	-0.0856	-0.0731
± 4			-0.0681	-0.0638
± 5				-0.0536
$a < \lvert l\rvert < \frac{k}{2} - a$	-0.1033	-0.0731	-0.0566	-0.0461
$\pm(\frac{k}{2}-)$				-0.0536
$\pm(\frac{k}{2} - 4)$			-0.0681	-0.0638
$\pm(\frac{k}{2} - 3)$		-0.0937	-0.0856	-0.0731
$\pm(\frac{k}{2} - 2)$	-0.1496	-0.1302	-0.1033	-0.0856
$\pm(\frac{k}{2} - 1)$	-0.2131	-0.1496	-0.1152	-0.0937
$\frac{k}{2}$	-0.2704	-0.1758	-0.1302	-0.1033

To obtain an estimate of the log-spectrum $\{\zeta_{\bar{X}(m)}(\frac{l}{k}) : l = -(\frac{k}{2}-1), \ldots, 0, \ldots, \frac{k}{2}-1, \frac{k}{2}\}$ of the batch means process, we have described in this section a procedure that first involves computing the periodogram of the batch means $\{I_{\bar{X}(m)}(\frac{l}{k}) : l = -(\frac{k}{2} - 1), \ldots, 0, \ldots, \frac{k}{2} - 1, \frac{k}{2}\}$. We then compute the smoothed periodogram of the batch means $\{\widetilde{I}_{\bar{X}(m)}(\frac{l}{k}) : l = -(\frac{k}{2}-1), \ldots, 0, \ldots, \frac{k}{2}-1, \frac{k}{2}\}$, and take $\widetilde{\mathcal{L}}_{\bar{X}(m)}(\frac{l}{k}) = \ln[\widetilde{I}_{\bar{X}(m)}(\frac{l}{k})]$ as our estimate of $\zeta_{\bar{X}(m)}(\frac{l}{k})$. An alternative estimate of

$\zeta_{\bar{X}(m)}(\frac{l}{k})$ may be obtained by computing the log-periodogram, $\ln[I_{\bar{X}(m)}(\frac{l}{k})]$, and then smoothing the result by taking a moving average of log-periodogram points; and we let $\overline{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)$ denote the resulting smoothed-log-periodogram,

$$\overline{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) = \frac{1}{A}\sum_{u=-a}^{a} \ln\left[I_{\bar{X}(m)}\left(\frac{l+u}{k}\right)\right] \text{ for } l = 0, \pm 1, \ldots, \pm\left(\tfrac{k}{2}-1\right), \tfrac{k}{2}. \tag{3.40}$$

In Appendix B we have derived expressions for the variances of these two different estimates of $\zeta_{\bar{X}(m)}(\frac{l}{k})$ for frequencies $l = 0, \pm 1, \pm 2, \ldots, \pm(\frac{k}{2}-1), \frac{k}{2}$; and the results are displayed in Table 3.3. For each integer j where $1 \le j \le a$, the quantities $\nu_j^{\#}$ and ν_j in Table 3.3 are defined as follows:

$$\nu_j^{\#} = \frac{2aA^2}{4a^2 - 2aj + 4a - 2j + 1} \quad \text{and} \quad \nu_j = \left\lfloor \nu_j^{\#} \right\rfloor. \tag{3.41}$$

Table 3.3: Comparison of the variances of the estimators $\widetilde{\mathcal{L}}_{\bar{X}(m)}(\frac{l}{k})$ and $\overline{\mathcal{L}}_{\bar{X}(m)}(\frac{l}{k})$ of the log-spectrum of the batch means at frequency $\frac{l}{k}$, where $l = 0, \pm 1, \ldots, \pm(\frac{k}{2}-1), \frac{k}{2}$.

Frequency Index l	$\text{Var}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}(\frac{l}{k})\right]$	$\text{Var}\left[\overline{\mathcal{L}}_{\bar{X}(m)}(\frac{l}{k})\right]$
$l = 0, \frac{k}{2}$	$\Psi'(a)$	$\dfrac{\Psi'(1)}{a}$
$1 \le \lvert l\rvert \le a$	$\Psi'\left(\dfrac{\nu_{\lvert l\rvert}}{2}\right)$	$\dfrac{\Psi'(1)}{\nu_{\lvert l\rvert}^{\#}/2}$
$a < \lvert l\rvert < \frac{k}{2} - a$	$\Psi'(A)$	$\dfrac{\Psi'(1)}{A}$
$\frac{k}{2} - a \le \lvert l\rvert \le \frac{k}{2} - 1$	$\Psi'\left(\dfrac{\nu_{k/2-\lvert l\rvert}}{2}\right)$	$\dfrac{\Psi'(1)}{\nu_{k/2-\lvert l\rvert}^{\#}/2}$

In Table 3.4, we show the values of $\text{Var}[\widetilde{\mathcal{L}}_{\bar{X}(m)}(0)]$ and $\text{Var}[\overline{\mathcal{L}}_{\bar{X}(m)}(0)]$ for smoothing parameter values $A = 5$, 7, 9, and 11. From this table we see that at zero frequency, smoothing the periodogram of the batch means and then taking the natural log results in an estimate of $\zeta_{\bar{X}(m)}(0)$ with smaller variance than the estimate obtained by computing the natural log of the periodogram of the batch means and then smoothing by taking a moving average of log-periodogram values. Furthermore, we prove in Appendix B the following general variance

reduction results,

$$\Psi'(j) < \frac{\Psi'(1)}{j} \quad \text{for} \quad j = 2, 3, \ldots \tag{3.42}$$

and

$$\Psi'\left(\frac{\lfloor x \rfloor}{2}\right) < \frac{\Psi'(1)}{x/2} \quad \text{for all real} \quad x \geq 3. \tag{3.43}$$

Applying (3.42) and (3.43) to the variances in Table 3.3, we can conclude that for all frequencies $\frac{l}{k}$ where $l = 0, \pm 1, \ldots, \pm(\frac{k}{2} - 1), \frac{k}{2}$ and for any value of the smoothing parameter A, smoothing the periodogram of the batch means and then taking the logarithm results in an estimate of $\zeta_{\bar{X}(m)}(\frac{l}{k})$ with smaller variance than the estimate of $\zeta_{\bar{X}(m)}(\frac{l}{k})$ obtained by smoothing the log-periodogram of the batch means.

Table 3.4: Comparison of the variance of $\widetilde{\mathcal{L}}_{\bar{X}(m)}(0)$, the log of the smoothed periodogram of the batch means (3.1) at zero frequency, to the variance of $\overline{\mathcal{L}}_{\bar{X}(m)}(0)$, the smoothed log-periodogram of the batch means (3.40) at zero frequency.

Smoothing Parameter	$\text{Var}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}(0)\right]$	$\text{Var}\left[\overline{\mathcal{L}}_{\bar{X}(m)}(0)\right]$
$A = 5$ $(a = 2)$	0.6449	0.8225
$A = 7$ $(a = 3)$	0.3949	0.5483
$A = 9$ $(a = 4)$	0.2838	0.4112
$A = 11$ $(a = 5)$	0.2213	0.3290

3.3.4 *Wavelet-Based Estimation of the Spectrum of the Batch Means*

In the previous section, we computed the log of the smoothed periodogram of the batch means $\{\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) : l = 0, \pm 1, \ldots, \pm(\frac{k}{2} - 1), \frac{k}{2}\}$. The next step in **WASSP** is to expand $\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)$ as a wavelet series to obtain a wavelet-based estimate of the log-spectrum of the batch means $\zeta_{\bar{X}(m)}\left(\frac{l}{k}\right)$. When the batch means periodogram $\{I_{\bar{X}(m)}\left(\frac{l}{k}\right) : l = 1, \ldots, \frac{k}{2} - 1\}$ was computed in the previous section, the number of points k in the batch means process $\{\overline{X}_1(m), \ldots, \overline{X}_k(m)\}$ was a power of two. After adding to the batch means periodogram the point $I_{\bar{X}(m)}(0)$ defined by (3.18) together with the points $\{I_{\bar{X}(m)}\left(\frac{l}{k}\right) : l = -(\frac{k}{2} - 1), \ldots, -1\}$ in the frequency range $\left(-\frac{1}{2}, 0\right)$, we see that the number of points in the extended batch means periodogram (and consequently in the log of the smoothed batch means periodogram) on the frequency range $\left(-\frac{1}{2}, \frac{1}{2}\right)$ becomes $2(\frac{k}{2} - 1) + 1 = k - 1$, an odd number. To compute the DWT of the log of the

smoothed batch means periodogram, $\{\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\}$, we must have a power of two for the total number points in the log-smoothed-periodogram of the batch means. To make the number of points in the log of the smoothed periodogram of the batch means a power of two, we add the end point $\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{k/2}{k}\right) = \widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{1}{2}\right)$, as defined in (3.20).

To compute the DWT of the log-smoothed-periodogram $\{\widetilde{\mathcal{L}}_{\bar{X}(m)}(\frac{l}{k})\}$ of the batch means process using k data points, we first correct for the bias in each of the components of $\widetilde{\mathcal{L}}_{\bar{X}(m)}(\frac{l}{k})$ so that we have the DWT

$$\widetilde{\boldsymbol{W}} = \boldsymbol{\Theta}\,\widetilde{\boldsymbol{\mathcal{L}}}, \tag{3.44}$$

where $\widetilde{\boldsymbol{\mathcal{L}}}$ is the bias-corrected log of the smoothed periodogram based on the bias terms (3.29), (3.33), (3.37) and (3.39); and $\boldsymbol{\Theta}$ is the $k \times k$ matrix that defines the DWT associated with the `s8` symmlet [9]. In Figure 3.5, the `s8` scaling function $\phi(t)$ and wavelet function $\psi(t)$ are shown. The `s8` symmlet is an excellent overall choice for representing many functions since it is orthogonal, smooth, nearly symmetric, and nonzero on a relatively short interval [9].

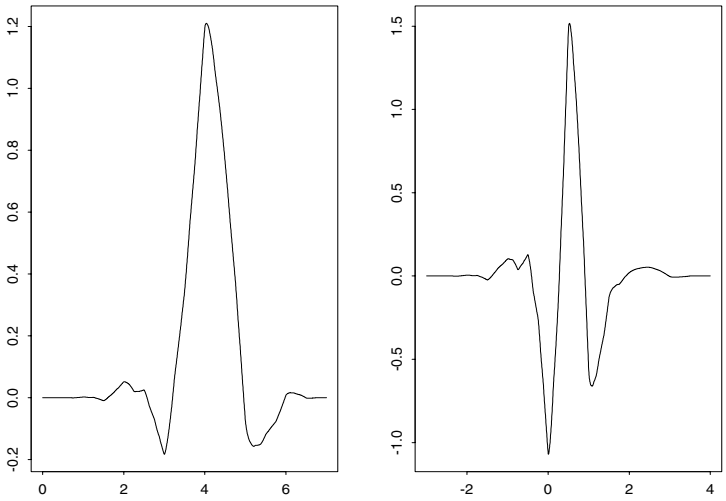

Figure 3.5: The `s8` symmlet scaling function $\phi(t)$ (left) and the `s8` symmlet wavelet function $\psi(t)$ (right).

Since the total number of points in the bias-corrected log of the smoothed periodogram of the batch means has the form $k = 2^J$, the number of resolution levels L used in the wavelet

decomposition of $\{\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\}$ is defined as follows,

$$L \equiv \left\lfloor \frac{J}{2} \right\rfloor, \tag{3.45}$$

and the coarsest level of resolution j_0 is then given by

$$j_0 = J - L. \tag{3.46}$$

This will give a total of 2^{j_0} coefficients at the coarsest level of resolution. Table 3.5 lists the values of L and j_0, as well as the total number of coefficients at level j_0, for various values of k.

Table 3.5: Number of levels of resolution L obtained by computing the DWT of a data set of size k. Also shown is the coarsest level of resolution j_0, the range of values for the resolution level j, and the number of coefficients at the coarsest level j_0.

k	J	L	j_0	j	# coefficients at level j_0
32	5	2	3	3,4	8
64	6	3	3	3,4,5	8
128	7	3	4	4 5,6	16
256	8	4	4	4,5,6,7	16
512	9	4	5	5,6,7,8	32
4096	12	6	6	6,7,8,9,10,11	64

The justification of (3.45) is as follows. The energy $\mathcal{E}_{\boldsymbol{f}}$ of a signal

$$\boldsymbol{f} = [f_1, f_2, \ldots, f_k]$$

of length k is defined as

$$\mathcal{E}_{\boldsymbol{f}} \equiv f_1^2 + f_2^2 + \ldots, + f_k^2.$$

Any orthogonal discrete wavelet transform of the signal $\boldsymbol{f}$ conserves the energy of $\boldsymbol{f}$ [66]. That is, if we let

$$\mathcal{E}_{j_0}^c \equiv \sum_{l=0}^{2^{j_0}-1} \widehat{c}_{j_0,l}^2$$

be the total energy of the scaling coefficients at level j_0 and

$$\mathcal{E}_j^d \equiv \sum_{l=0}^{2^j-1} \widehat{d}_{j,l}^2$$

be the total energy of the jth level wavelet coefficients for $j = j_0, \ldots, J-1$, then

$$\mathcal{E}_{\boldsymbol{f}} = \mathcal{E}^c_{j_0} + \sum_{j=j_0}^{J-1} \mathcal{E}^d_j.$$

In general, the DWT redistributes the energy of $\boldsymbol{f}$ by compressing most of the energy into the coarsest-level scaling coefficients. Consider the example shown in Figures 3.6 and 3.7. The top plot of Figure 3.6 shows the plot of a Doppler signal sampled at $k = 2^{10}$ discrete time points. The bottom plot shows the energy plot for the 1-level DWT of the Doppler signal so that $L = 1$ in this case. Taking $L = 1$ implies that $j_0 = J - L = 9$ so that the DWT yields the scaling coefficients $\{\widehat{c}_{9,l} : l = 0, 1, \ldots, 2^9 - 1\}$ and the wavelet coefficients $\{\widehat{d}_{9,l} : l = 0, 1, \ldots, 2^9 - 1\}$. The top line represents the total energy contained in the original Doppler signal. The middle line represents the total energy contained in the first level of wavelet coefficients $\mathcal{E}^d_9$. The last line represents the total energy contained in the scaling coefficients $\mathcal{E}^c_9$. From this plot, we can see that nearly 100% of the energy of the Doppler signal is contained in the coarsest-level scaling coefficients. That is, the energy of the Doppler signal has been compressed into a signal that is 1/2 the length of the Doppler signal.

The top plot of Figure 3.7 shows the energy plot for the case $L = 2$ so that we compute the 2-level DWT of the Doppler signal. In this case, $j_0 = 8$. The second line from the top represents the energy $\mathcal{E}^d_9$. The next line represents $\mathcal{E}^d_8$ and the bottom line represents $\mathcal{E}^c_8$. From this plot, we can see that about 98% of the energy of the Doppler signal is still contained in the scaling coefficients, and that the energy of the Doppler signal has been compressed into a signal that is 1/4 the length of the original signal.

The bottom plot of Figure 3.7 shows the energy plot for the case $L = 6$ so that we compute the 6-level DWT of the Doppler signal taking $j_0 = J - L = 4$. The second line from the top represents $\mathcal{E}^d_9$; the following lines respectively represent $\mathcal{E}^d_8$, $\mathcal{E}^d_7$, $\mathcal{E}^d_6$, $\mathcal{E}^d_5$, $\mathcal{E}^d_4$; and the bottom line represents $\mathcal{E}^c_4$. From this plot, we see that less than 80% of the energy of the Doppler signal is *localized* in the signal

$$\widehat{\boldsymbol{c}}_4 \equiv [\widehat{c}_{4,0}, \widehat{c}_{4,1}, \ldots, \widehat{c}_{4,15}]$$

of length $2^{j_0} = 16$ consisting of the coarsest-level scaling coefficients. As $j_0 \to 0$ (that is, as the number of scaling coefficients decreases), the amount of energy localized in the scaling coefficients must necessarily decrease. In the quantum theory of physics, Heisenberg's Uncertainty Principle states roughly that it is impossible to localize a fixed amount of energy into an arbitrarily small time interval. One wavelet-transform analog of Heisenberg's Uncertainty Principle (Theorem 2.4.1 of [62]) states roughly that for signal $\boldsymbol{f}$ of length k, it is impossible to localize the energy $\mathcal{E}_{\boldsymbol{f}}$ of the signal $\boldsymbol{f}$ in a subsignal consisting of scaling and/or wavelet coefficients extracted from the DWT of $\boldsymbol{f}$, where the length of the subsignal is very much less than the

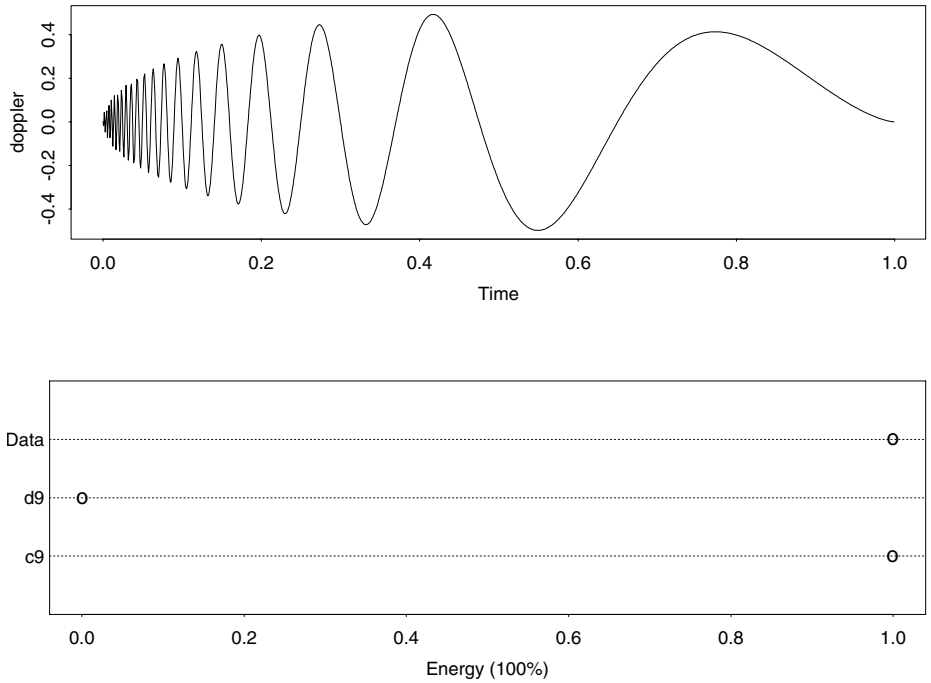

Figure 3.6: Plot of a Doppler signal sampled at $n = 2^{10}$ discrete time points (top); and energy plot for the 1-level DWT of the Doppler signal.

length of the original signal; see pp. 8–9 of [66].

In equation (3.45), the number of resolution levels L is set to ensure the number of scaling coefficients is large enough so that a high percentage of the energy of the bias-corrected log of the smoothed periodogram of the batch means is contained in the scaling coefficients (which are not thresholded), while at the same time the number of scaling coefficients is not so large that many of the scaling coefficients represent noise. If $L = 1$ (that is, we have the maximum possible number of scaling coefficients, $k/2$), then nearly 100% of the energy of the bias-corrected log of the smoothed periodogram of the batch means will be contained in the scaling coefficients. In this situation, however, it is also likely that some of the scaling coefficients will represent noise. Since we are not thresholding the scaling coefficients, the ultimate result will be a less-smooth estimate of the log-spectrum of the batch means. On the other hand, if $L = J$ so that there is only one scaling coefficient, then it is highly likely that the single scaling coefficient does not represent noise. However, the estimate of the log-spectrum of the batch means will not be as accurate since all coefficients but one will be thresholded, including smaller-magnitude coefficients that do not represent noise.

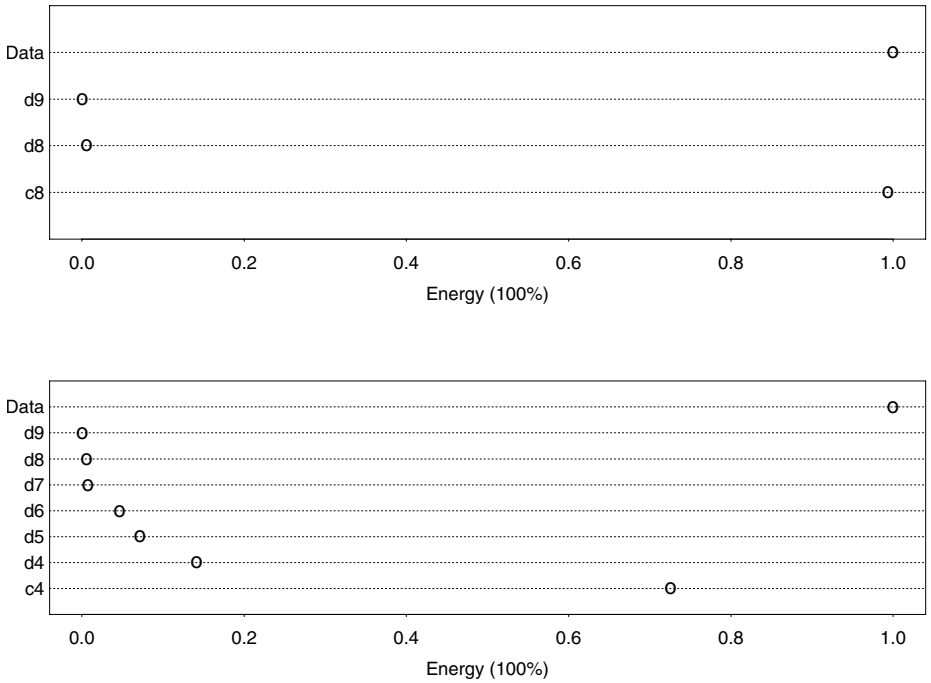

Figure 3.7: 2-level DWT of the Doppler signal shown in Figure 3.6 (top); and energy plot for the 6-level DWT of the same Doppler signal (bottom).

After computing the DWT of the bias-corrected log of the smoothed periodogram of the batch means $\widetilde{\mathfrak{L}}$ as given by (3.44), we threshold the resulting wavelet coefficients $\{\widehat{d}_{j,l} : j = j_0, j_0+1, \ldots, J-1; l = 0, 1, \ldots, 2^j - 1\}$ using Gao's thresholding scheme (2.60) to obtain the coefficients $\{\widehat{d}^*_{j,l} : j = j_0, j_0+1, \ldots, J-1; l = 0, 1, \ldots, 2^j - 1\}$. The empirical scaling coefficients $\{\widehat{c}_{j_0,l} : l = 0, 1, \ldots, 2^{j_0} - 1\}$ are not thresholded since it is presumed they contain information about the coarse features of the log-spectrum of the batch means. The inverse transform

$$\widetilde{\boldsymbol{\mathfrak{L}}}^* = \boldsymbol{\Theta}^{\mathrm{T}} \widetilde{\boldsymbol{W}}^* \tag{3.47}$$

is computed, where $\widetilde{\boldsymbol{W}}^*$ is the vector containing the coefficients $\{\widehat{c}_{j_0,l} : l = 0, 1, \ldots, 2^{j_0} - 1\}$ and $\{\widehat{d}^*_{j,l} : j = j_0, \ldots, J-1; l = 0, 1, \ldots, 2^j - 1\}$, to obtain the corresponding thresholded wavelet approximation

$$\widetilde{\boldsymbol{\mathfrak{L}}}^* = \left[\widetilde{\mathfrak{L}}^*_1, \widetilde{\mathfrak{L}}^*_2, \ldots, \widetilde{\mathfrak{L}}^*_k\right]^{\mathrm{T}} = \left[\widetilde{\mathfrak{L}}^*_{\bar{X}(m)}\left(\frac{-(\frac{k}{2}-1)}{k}\right), \ldots, \widetilde{\mathfrak{L}}^*_{\bar{X}(m)}(0), \ldots, \widetilde{\mathfrak{L}}^*_{\bar{X}(m)}\left(\tfrac{1}{2}\right)\right]^{\mathrm{T}}$$

to the vector $\widetilde{\mathfrak{L}}$ of the bias-corrected log of the smoothed periodogram of the batch means. Therefore, our wavelet-based estimate of the log-spectrum of the batch means $\zeta_{\bar{X}(m)}\left(\frac{l}{k}\right)$ is given as follows,

$$\widehat{\zeta}_{\bar{X}(m)}\left(\frac{l}{k}\right) = \widetilde{\mathfrak{L}}^{*}_{\bar{X}(m)}\left(\frac{l}{k}\right), \quad \text{for} \quad l = 0, \pm 1, \ldots, \pm\left(\tfrac{k}{2} - 1\right), \tfrac{k}{2}.$$

Figures 3.8–3.10 show plots of the bias-corrected log of the smoothed periodogram of the batch means and the corresponding wavelet estimate of the bias-corrected log-smoothed periodogram for the batch means of the waiting times in an $M/M/1$ queueing system with 90% traffic intensity and an empty-and-idle initial condition. Each figure was generated by first collecting n =32,768 waiting time observations from a simulation replication of the $M/M/1$ queueing system. The independence test and the normality test were conducted to obtain a batch means process $\{\overline{X}_1(m), \ldots, \overline{X}_k(m)\}$, where we have $k = 32$ for replication 1; we have $k = 64$ for replication 2; and we have $k = 128$ for replication 3. Finally, the bias-corrected log of the smoothed periodogram of the batch means and the corresponding wavelet estimate were computed for each replication. From these plots, it is clear that there is a significant amount of variability in the shape of the bias-corrected log of the smoothed periodogram of the batch means from replication to replication. It is also evident that transforming the bias-corrected log of the smoothed periodogram using the thresholded DWT successfully removes the noise, as can be seen clearly in Figures 3.8–3.10.

Tables 3.6–3.8 show the total amount of energy contained in each resolution level of both the thresholded and nonthresholded versions of the wavelet estimate of the bias-corrected log-smoothed periodogram of the batch means for replications 1–3. From these tables, we see that before thresholding, the total energy of the bias-corrected log-smoothed periodogram of the batch means is conserved after computing the DWT; and furthermore, the majority of this energy is compressed into the thesholded coarsest-level scaling coefficients.

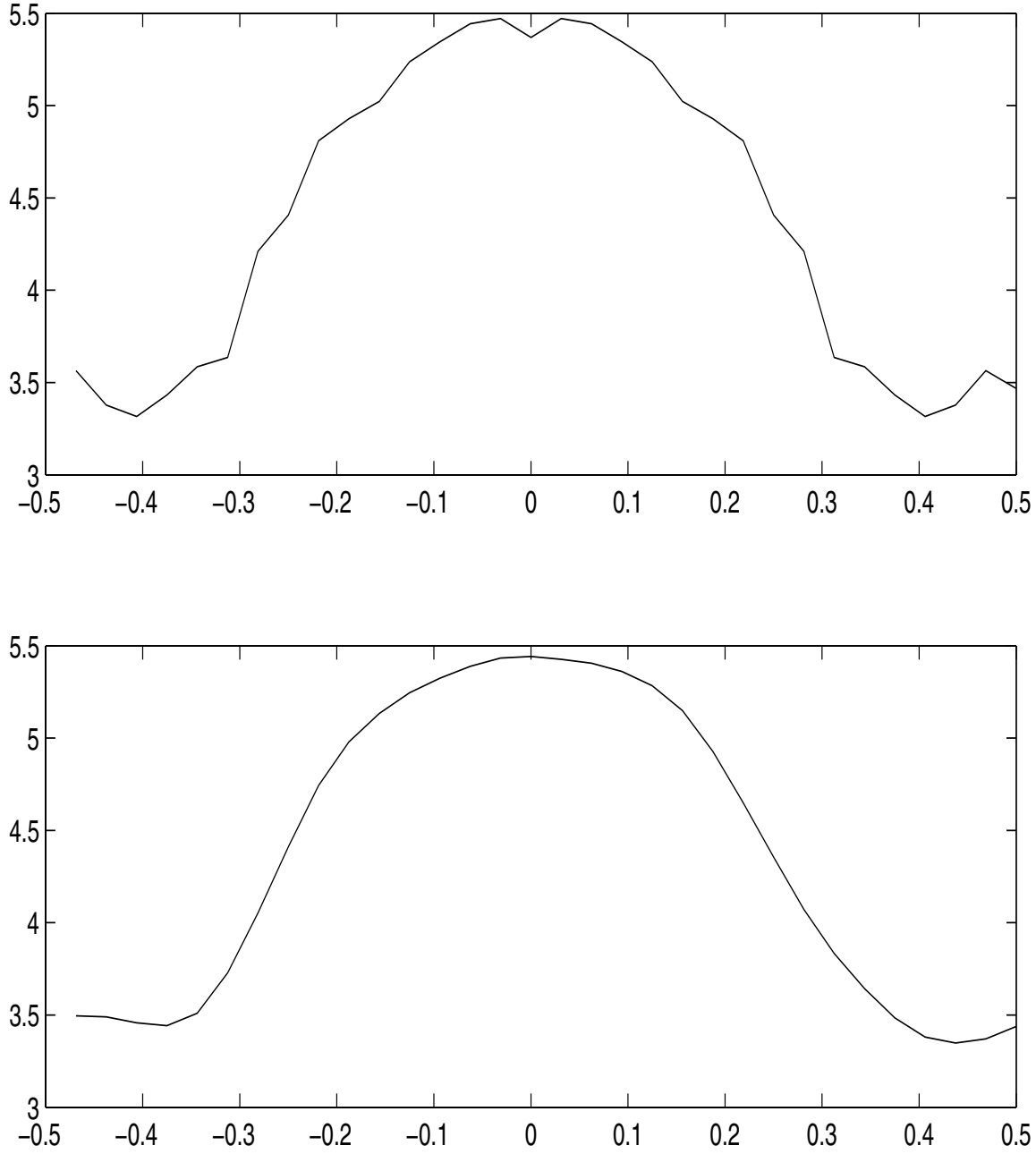

Figure 3.8: The bias-corrected smoothed log-periodogram of the batch means (top) and the corresponding thresholded wavelet estimate (bottom) for $k = 32$ batch means computed from the waiting times for an $M/M/1$ queueing system with 90% traffic intensity, replication 1.

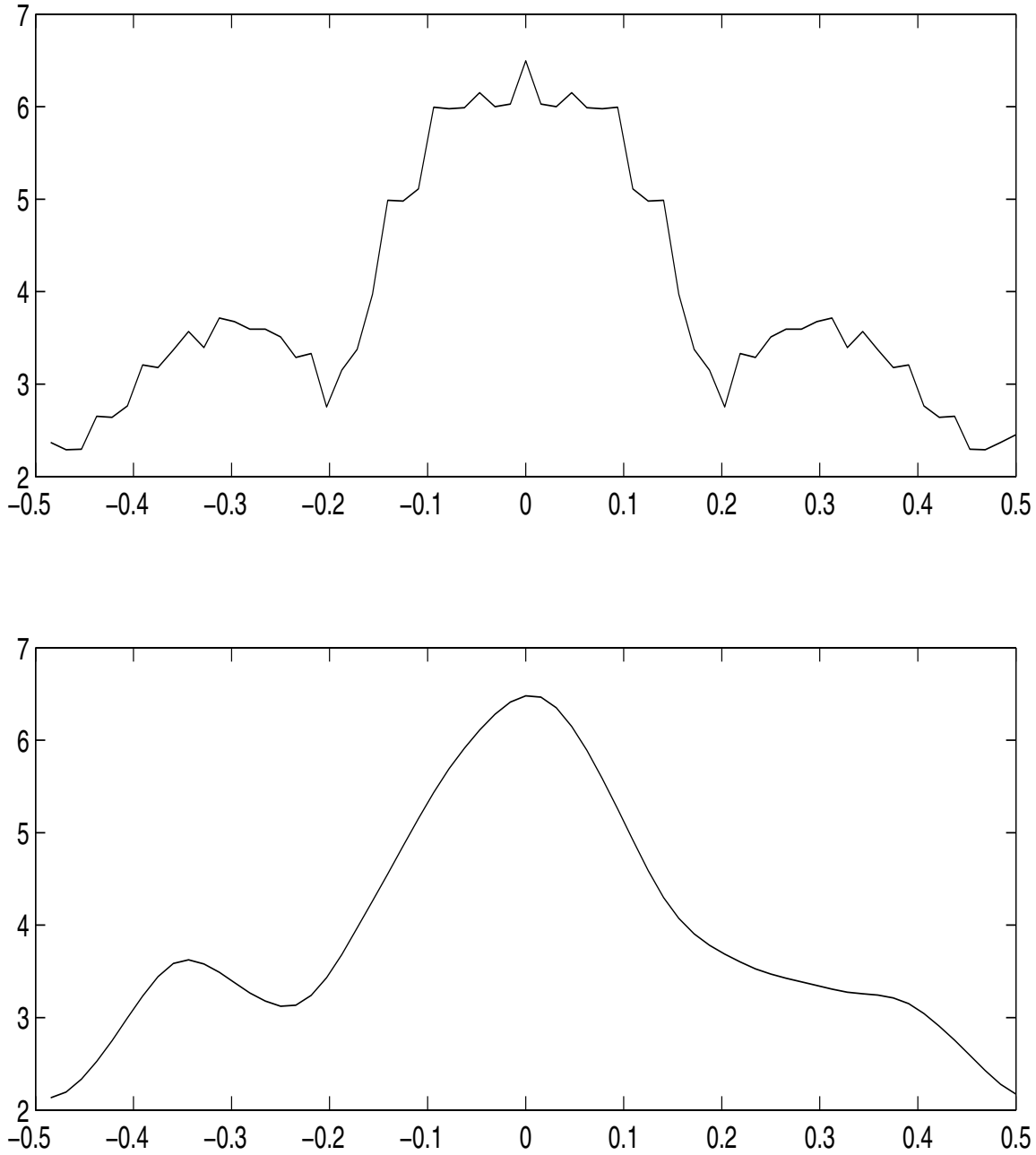

Figure 3.9: The bias-corrected smoothed log-periodogram of the batch means (top) and the corresponding thresholded wavelet estimate (bottom) for $k = 64$ batch means computed from the waiting times for an $M/M/1$ queueing system with 90% traffic intensity, replication 2.

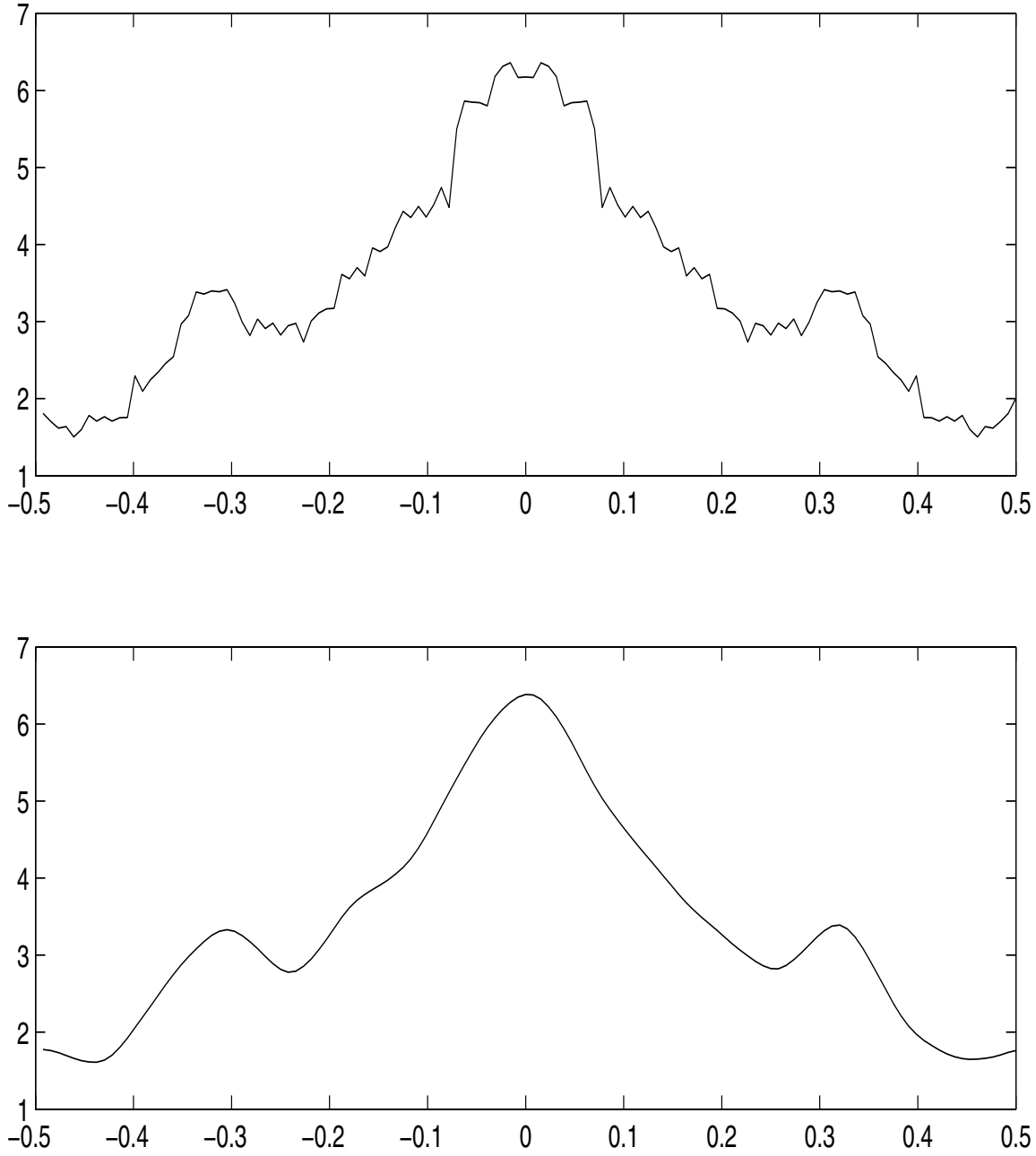

Figure 3.10: The bias-corrected smoothed log-periodogram of the batch means (top) and the corresponding thresholded wavelet estimate (bottom) for $k = 128$ batch means computed from the waiting times for an $M/M/1$ queueing system with 90% traffic intensity, replication 3.

Table 3.6: Energy decomposition by resolution level of the wavelet estimate of the bias-corrected log-smoothed periodogram of the batch means for replication 1 of the $M/M/1$ queue waiting time process shown in Figure 3.8 with total energy $\mathcal{E}_{\widetilde{\mathcal{L}}} = 458.5368$.

	Before Thresholding		After Thresholding	
	Energy	% of $\mathcal{E}_{\widetilde{\mathcal{L}}}$	Energy	% of $\mathcal{E}_{\widetilde{\mathcal{L}}}$
$\mathcal{E}_3^c$	457.8729	99.86	457.8729	99.86
$\mathcal{E}_3^d$	0.2731	0.06	0	0
$\mathcal{E}_4^d$	0.3908	0.08	0	0

Table 3.7: Energy decomposition by resolution level of the wavelet estimate of the bias-corrected log-smoothed periodogram of the batch means for replication 2 of the $M/M/1$ queue waiting time process shown in Figure 3.9 with total energy $\mathcal{E}_{\widetilde{\mathcal{L}}} = 1074.8$.

	Before Thresholding		After Thresholding	
	Energy	% of $\mathcal{E}_{\widetilde{\mathcal{L}}}$	Energy	% of $\mathcal{E}_{\widetilde{\mathcal{L}}}$
$\mathcal{E}_3^c$	1065.6	99.14	1065.6	99.14
$\mathcal{E}_3^d$	6.2289	0.58	0	0
$\mathcal{E}_4^d$	0.7250	0.07	0	0
$\mathcal{E}_5^d$	2.2165	0.21	0	0

Table 3.8: Energy decomposition by resolution level of the wavelet estimate of the bias-corrected log-smoothed periodogram of the batch means for replication 3 of the $M/M/1$ queue waiting time process shown in Figure 3.10 with total energy $\mathcal{E}_{\widetilde{\mathcal{L}}} = 1{,}741.1$.

	Before Thresholding		After Thresholding	
	Energy	% of $\mathcal{E}_{\widetilde{\mathcal{L}}}$	Energy	% of $\mathcal{E}_{\widetilde{\mathcal{L}}}$
$\mathcal{E}_4^c$	1,737.5	99.79	1,737.5	99.79
$\mathcal{E}_4^d$	1.5979	0.09	0	0
$\mathcal{E}_5^d$	0.9495	0.05	0	0
$\mathcal{E}_6^d$	1.0202	0.06	0	0

The wavelet-based estimate of the spectrum of the batch means process can now be computed as follows,

$$\widehat{p}_{\bar{X}(m)}\left(\frac{l}{k}\right) = \exp\left[\widehat{\zeta}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \quad \text{for} \quad l = 0, \pm 1, \ldots, \pm\left(\tfrac{k}{2} - 1\right), \tfrac{k}{2}.$$

It follows that a wavelet-based estimate of the SSVC for the original (unbatched) process is

given by

$$\widehat{\gamma}_X = m \cdot \widehat{p}_{\bar{X}(m)}(0), \tag{3.48}$$

and an approximate $100(1-\beta)\%$ confidence interval for μ_X is

$$\overline{\overline{X}}(m,k) \pm t_{1-\beta/2,2a} \sqrt{\frac{\widehat{\gamma}_X}{n'}}, \tag{3.49}$$

where $n' = mk$ and $\overline{\overline{X}}(m,k)$ is the grand average of the k batch means $\{\overline{X}_1(m), \ldots, \overline{X}_k(m)\}$.

A proof of the result displayed in (3.48) is given as follows. For the unbatched steady-state process $\{X_i : i = \mathcal{S}+1, \mathcal{S}+2, \ldots, \mathcal{S}+n'\}$ consisting of the observations generated beyond $\mathcal{S}$, the end of the warm-up period, the SSVC γ_X is defined as

$$\gamma_X \equiv \lim_{n' \to \infty} n' \operatorname{Var}\Big[\overline{X}(n')\Big], \tag{3.50}$$

where

$$\overline{X}(n') = \frac{1}{n'} \sum_{i=1+\mathcal{S}}^{\mathcal{S}+n'} X_i. \tag{3.51}$$

If we batch the process $\{X_i : i = 1+\mathcal{S}, 2+\mathcal{S}, \ldots, \mathcal{S}+n'\}$ into k batches of size m, then we have a new process consisting of the batch means $\{\overline{X}_i(m) : i = 1, 2, \ldots, k\}$, where

$$\overline{X}_i(m) = \frac{1}{k} \sum_{j=\mathcal{S}+(i-1)m+1}^{im} X_j.$$

Furthermore, the grand mean of the process $\{\overline{X}_i(m) : i = 1, 2, \ldots, k\}$ is given by

$$\overline{\overline{X}}(m,k) = \frac{1}{k} \sum_{l=1}^{k} \overline{X}_l(m). \tag{3.52}$$

Recognizing that

$$\overline{\overline{X}}(m,k) = \overline{X}(n')$$

and

$$n' = mk,$$

we see that equation (3.50) can be rewritten as

$$\gamma_X \quad = \quad \lim_{k \to \infty} mk \operatorname{Var}\Big[\overline{\overline{X}}(m,k)\Big]$$

$$= \quad m \lim_{k\to\infty} k \operatorname{Var}\left[\overline{\overline{X}}(m,k)\right]. \tag{3.53}$$

Now, the SSVC of the batch means process $\{\overline{X}_i(m) : i = 1, 2, \ldots, k\}$ is defined as

$$\gamma_{\bar{X}(m)} \equiv \lim_{k\to\infty} k \operatorname{Var}\left[\overline{\overline{X}}(m,k)\right]. \tag{3.54}$$

Substituting (3.54) into (3.53), we have the result

$$\gamma_{\bar{X}(m)} = \frac{\gamma_X}{m}.$$

3.3.5 *Fulfilling the Precision Requirement*

WASSP is a sequential method for constructing confidence intervals that satisfy a given accuracy requirement, specified as either a maximum relative fraction r^* of the magnitude of the final grand mean $\overline{\overline{X}}(m,k)$, or as a maximum absolute half-length h^*. A confidence interval is delivered once the accuracy requirement is satisfied. The user is also allowed the option of not specifying a precision level, in which case a confidence interval is delivered after one iteration of **WASSP**.

The half-length of the confidence interval (3.49) is given by

$$H = t_{1-\beta/2,2a}\sqrt{\frac{\widehat{\gamma}_X}{n'}} = t_{1-\beta/2,2a}\sqrt{\frac{\widehat{p}_{\bar{X}(m)}(0)}{k}}.$$

If the confidence interval (3.49) satisfies the precision requirement,

$$H \le H^* \tag{3.55}$$

where H^* is as given in (3.5), then **WASSP** terminates and the confidence interval (3.49) is delivered.

If the precision requirement (3.55) is not satisfied, then the total number of batches required to satisfy the precision requirement is computed as follows,

$$k^* = \left\lceil \left(\frac{H}{H^*}\right)^2 k \right\rceil;$$

and thus the total sample size required to meet the precision requirement is k^*m. However, since the number of batches must be a power of two, the batch count k is set for the next iteration of **WASSP** as follows:

$$k \longleftarrow \min\left\{2^{\lfloor \log_2(k^*)\rfloor}, 4096\right\}, \tag{3.56}$$

where 4,096 is the upper bound on the number of batch means used in **WASSP**.

Setting the batch size for the next iteration of **WASSP** requires some care. If the current batch size m is used on the next iteration of **WASSP** together with the new batch count k given by (3.56), then the new total sample size including the warm-up period is

$$n \leftarrow km + \mathcal{S}.$$

If $k < k^*$, then the new total sample size $km + \mathcal{S}$ will be less than the total sample size $k^*m + \mathcal{S}$ required to meet the precision requirement. In this case, the new batch size m^* is computed from

$$km^* + \mathcal{S} = k^*m + \mathcal{S}, \tag{3.57}$$

where k is given by (3.56). Eliminating $\mathcal{S}$ from (3.57) and then dividing both sides of (3.57) by k, we see that we must take

$$m^* = \frac{k^*}{k}m.$$

Therefore, if $k < k^*$, then the new batch size m for the next iteration of **WASSP** is assigned according to

$$m \leftarrow \left\lceil \left(\frac{k^*}{k} \right) m \right\rceil, \tag{3.58}$$

so that the total sample size n is increased approximately by a factor of $\left(\frac{H}{H^*}\right)^2$.

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

$$n \leftarrow km + \mathcal{S},$$

where the corresponding batch count k and batch size m are given by (3.56) and (3.58), respectively. The additional simulation-generated observations are obtained by restarting the simulation or by retrieving the extra data from storage; and then the next iteration of **WASSP** is performed.

The next chapter summarizes the results of a comprehensive experimental performance evaluation of **WASSP** that is specifically designed to reveal both the strengths and weaknesses of **WASSP**.

Chapter 4

Performance Evaluation of WASSP

We applied WASSP to a variety of particularly difficult test processes that were designed specifically to explore the robustness of the procedure against the statistical anomalies commonly encountered in the analysis of outputs generated by large-scale steady-state simulation experiments. The test processes used in the performance evaluation of WASSP are as follows:

(i) the $M/M/1$ queue waiting time process with server utilization equal to 0.90;

(ii) the first-order autoregressive process (AR(1)) with lag-one correlation equal to 0.995;

(iii) the "AR(1)-to-Pareto" (ARTOP) process that has marginals given by a Pareto distribution with lower limit and shape parameter equal to 1 and 2.1, respectively (implying the marginal mean and variance are both finite while the marginal skewness and kurtosis are both infinite), and that is obtained by applying to process (iii) the composite of the inverse of the specified Pareto c.d.f. and the standard normal c.d.f;

(iv) the "AR(1)-to-Johnson" (ARTOJ) process that has marginals given by a Johnson S_U distribution with mean, variance, skewness, and kurtosis equal to 1, 1, 100, and 900, respectively, and that is obtained by applying to process (iii) above the inverse of the normalizing translation defining the specified Johnson S_U distribution; and

(v) the $M/M/1$/LIFO queue waiting time process with server utilization equal to 0.80.

Subsections 4.1–4.6 describe each of these processes in more detail.

For each of the above test processes, the parameters used to evaluate the performance of WASSP are the coverage probability of its confidence intervals, the mean and half-length of its confidence intervals, and the total required sample size. We performed independent replications of WASSP to construct nominal 90% and 95% confidence intervals that satisfy a specified precision requirement. The following three precision requirements were used:

(a) no precision—that is, we set $h^* = \infty$ in (3.4) and (3.5) so $\mathcal{WASSP}$ delivers the confidence interval (3.3) using the batch count and batch size required to pass the independence and normality tests;

(b) $\pm$ 15% precision—that is, $\mathcal{WASSP}$ delivers the confidence interval (3.3) satisfying the relative precision requirement given by (3.4) and (3.5) with $r^* = 0.15$; and

(c) $\pm$ 7.5% precision—that is, $\mathcal{WASSP}$ delivers the confidence interval (3.3) satisfying the relative precision requirement given by (3.4) and (3.5) with $r^* = 0.075$.

For each test process listed above, the theoretical steady-state mean response is available analytically (that is, a numerical method can be implemented to rapidly compute the exact value of the steady-state mean response to several significant figures); and therefore we were able to evaluate the performance of $\mathcal{WASSP}$ in terms of actual versus nominal coverage probabilities for the confidence intervals delivered by the procedure. For the sake of comparison, we also applied ASAP2 and Heidelberger and Welch's spectral method to the test processes listed above. The next section describes how we implemented Heidelberger and Welch's spectral procedure.

4.1. Implementation of Heidelberger and Welch's Spectral Method

Comparing the performance of $\mathcal{WASSP}$ with that of Heidelberger and Welch's procedure is complicated since the latter procedure requires the user to specify an upper limit $t_{\max}$ on the allowable simulation run length; and in many practical applications, the user may seek to use a steady-state simulation analysis procedure as a means of estimating a reasonable simulation run length *based on the ultimate objectives of the simulation study.* To make a fair comparison between the performance of $\mathcal{WASSP}$ and the performance of Heidelberger and Welch's sequential spectral method (as described in Section 2.7.2), we first applied $\mathcal{WASSP}$ to a realization of a particular process so as to obtain not only the corresponding $\mathcal{WASSP}$-based confidence interval but also a complete data set to which we may apply the (partially) sequential version of the Heidelberger-Welch procedure. In particular on each replication of $\mathcal{WASSP}$ and Heidelberger and Welch's procedure for the $\pm$15% and the $\pm$7.5% precision cases, we set the maximum run length $t_{\max}$ for Heidelberger and Welch's method to the final sample size required by $\mathcal{WASSP}$ for that replication. However, since Heidelberger and Welch's method for detecting and eliminating initialization bias in their sequential spectral method [28] is not fully developed and since there is no precise and complete algorithmic statement of the procedure that we can use to compare to $\mathcal{WASSP}$, we decided to apply their sequential method without the test for initialization bias. Therefore, Heidelberger and Welch's method is applied to the entire data set, including those observations comprising the warm-up period. For the no precision case, we applied the

Heidelberger and Welch method to a fixed sample of size 4,096 (as described in Section 2.7.1). That is, we applied Heidelberger and Welch's method to the first 4,096 observations of the data set used by WASSP for each replication.

Heidelberger and Welch describe a method for batching data in [26]. We employed this same batching procedure in our implementation of Heidelberger and Welch's method. In particular, we set the maximum number of possible batches $k^* = 200$. Within one run (replication) of a test process, let t_i denote the time (sample size) at which the ith confidence interval is computed. The values of t_i are computed as follows,

$$t_1 = \lceil 0.15\, t_{\max} \rceil \tag{4.1}$$

and

$$t_i = \min\{\lceil 1.5\, t_{i-1} \rceil, t_{\max}\}, \quad i = 2, 3, \ldots . \tag{4.2}$$

The factors 0.15 in (4.1) and 1.5 in (4.2) are based on Heidelberger and Welch's recommendations for implementing their procedure in practice [26]. The batch size m_i and the number of batches k_i at time (sample size) t_i can then be computed according to

$$m_i = 2^{\lfloor \log_2\{(t_i - 1)/k^*\} \rfloor},$$

and

$$k_i = \left\lfloor \frac{t_i}{m_i} \right\rfloor,$$

respectively, where $t_i \geq k^*$. Finally, to be consistent with Heidelberger and Welch's experimental results in [26], we estimated the log-spectrum by fitting a polynomial of degree $q = 2$ to the first $K = 25$ points of the log of the smoothed periodogram. Setting the parameters $K = 25$ and $q = 2$ implies that the bias-correction constant $C_1(K, q) = 0.882$ and the "effective" degrees of freedom $\nu(K, q) = 7$. Note that C_1 and ν are associated with the Heidelberger-Welch spectral estimator $\widehat{p}_X(0)$ of the SSVC. A confidence interval of the form (2.77) is computed at each time (sample size) t_i for $i = 1, 2, \ldots$, until either the precision requirement is satisfied, or the maximum run length is reached (that is, $t_i = t_{\max}$). For the no precision case, a confidence interval of the form (2.77) is computed using a total of 4,096 observations.

4.2. The $M/M/1$ Queue Waiting Time Process

The performance of WASSP for the $M/M/1$ queue waiting time process is discussed in this section. For this system, X_i is the waiting time for the ith customer, $i = 1, 2, \ldots$, in a single-server queueing system with i.i.d. exponential interarrival times having mean 10/9, i.i.d. exponential

service times having mean 1, a steady-state server utilization of 90%, and an empty-and-idle initial condition (so that $X_1 = 0$). The theoretical mean for this waiting time process is $\mu_X = 9.0$.

The $M/M/1$ queue waiting time process with 90% server utilization is a particularly difficult test process for several reasons. First, because the system is starting empty and idle, the duration of the initial transient period is long. If the end of the warm-up period is not identified correctly and observations prior to the end of the warm-up period are used in the analysis, then the resulting point estimate $\overline{\overline{X}}$ will be biased. A second aspect of this process which makes it a difficult test case is that once the system has reached steady-state operation, the autocorrelation function decays very slowly with increasing lags. Consequently, this process will allow thorough evaluation of the robustness of **WASSP**'s independence test and the ability of **WASSP** to

(a) identify a point beyond which all computed batch means are approximately independent of the simulation model's initial conditions; and

(b) construct a set of spaced batch means such that the interbatch spacer preceding each batch is sufficiently large to ensure all computed batch means are approximately i.i.d.

Finally, this process is a difficult test case because the marginal distribution of waiting times is markedly nonnormal, having an exponential tail. Hence, this process will also allow us to evaluate the effectiveness of **WASSP**'s normality test and its ability to determine an appropriate batch size beyond which computed batch means are approximately normal.

For an $M/M/1$ queue with traffic intensity ϱ and mean interarrival time $\mathrm{E}[B_i]$, where B_i is the interarrival time between the $(i-1)$st and ith customers for $i = 1, 2, \ldots$, the power spectrum of the steady-state waiting time process $\{X_i\}$ for $\omega \in [-1/2, 1/2]$ is

$$p_X(\omega) = \frac{\varrho^3(2-\varrho)}{(1-\varrho)^2}\,\mathrm{E}^2[B_i] + \frac{1-\varrho^2}{\pi}\,\mathrm{E}^2[B_i] \int_0^r \frac{t^{5/2}(r-t)^{1/2}[\cos(2\pi\omega) - t]}{(1-t)^3[1 - 2t\cos(2\pi\omega) + t^2]}\,dt, \tag{4.3}$$

where $r = 4\varrho/(1+\varrho)^2$. Since it appears that the literature lacks available computing formulas for $p_X(\omega)$, the result (4.3) is derived in Appendix D.

For the selected steady-state $M/M/1$ queue waiting time process with arrival rate 10/9 and service rate 1, Figure 4.1 displays plots of the spectrum $p_X(\omega)$ and the log-spectrum $\ln[p_X(\omega)]$ for $\omega \in [-1/2, 1/2]$. Clearly both functions are sharply peaked at zero frequency. In the **WASSP** algorithm, however, we are actually estimating the log-spectrum of the batch means process $\ln[p_{\bar{X}(m)}(\omega)]$. From the plot in Figure 4.1 we can get a general idea of what the spectrum of the batch means process will look like since the peakedness of $p_{\bar{X}(m)}(\omega)$ depends on the peakedness of the spectrum of the underlying process. While the power spectrum of the batch means will be less peaked than $p_X(\omega)$, it will still be sharply peaked; and this property will enable us to

gauge the robustness of WASSP's wavelet-based technique for estimating the spectrum at zero frequency.

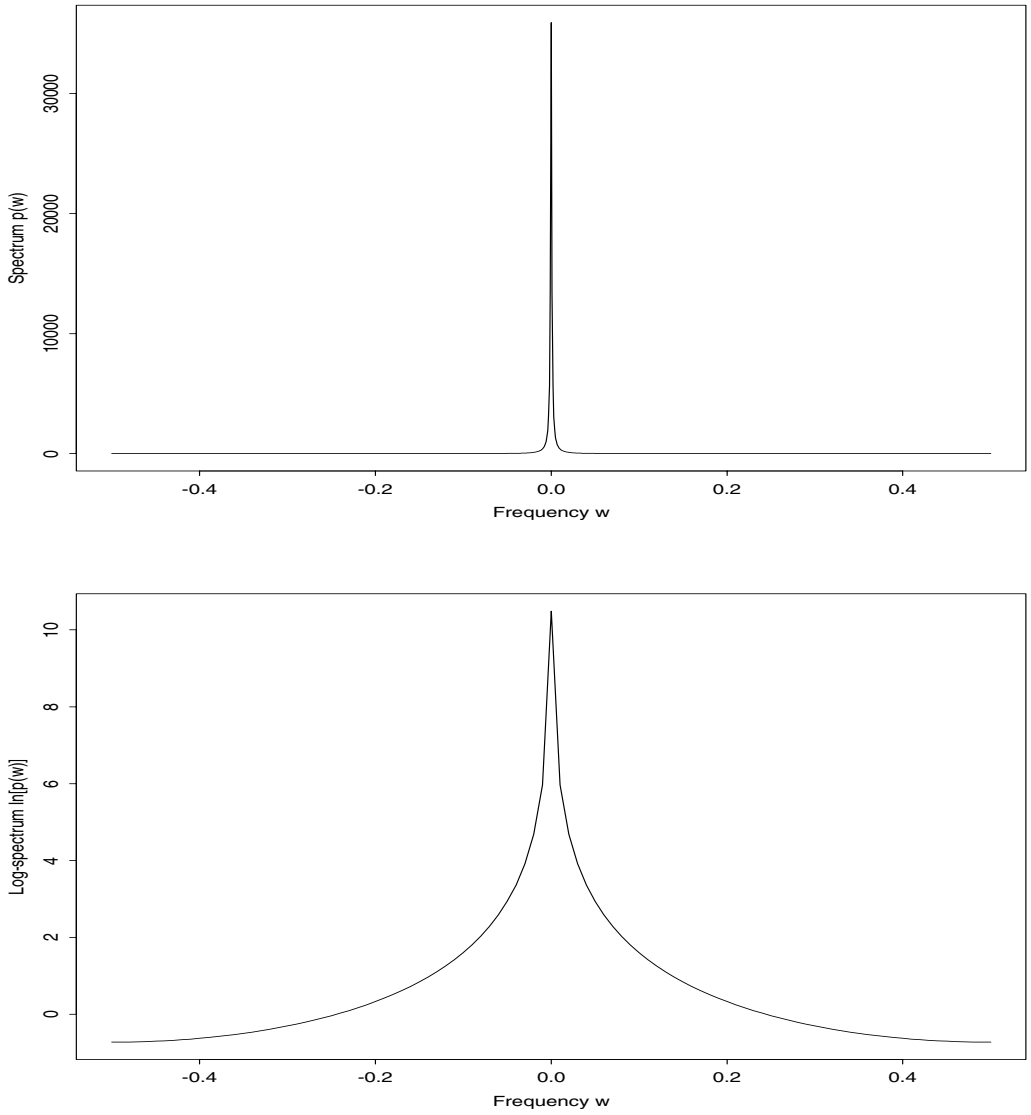

Figure 4.1: Plots for frequency $\omega \in [-\frac{1}{2}, \frac{1}{2}]$ of the spectrum $p_X(\omega)$ (top panel) and the log-spectrum $\ln[p_X(\omega)]$ (bottom panel) of the steady-state $M/M/1$ queue waiting time process $\{X_i\}$ with 90% server utilization (arrival rate 10/9 and service rate 1).

Table 4.1 shows the performance of WASSP for the $M/M/1$ queue waiting time process using the smoothing parameter values $A = 5, 7, 9,$ and 11. The results are based on 1,000 independent replications of nominal 90% confidence intervals (CIs). Table 4.2 shows the corresponding results for nominal 95% CIs. The standard error is less than 1% for each coverage estimator in Tables 4.1 and 4.2. From these tables, it is evident that the coverage probability decreases in general as the smoothing parameter increases. This is due to the target process having a power spectrum with a sharp peak in the neighborhood of zero frequency. As the smoothing parameter A is increased, WASSP's estimate of the power spectrum near zero frequency becomes flatter, resulting in an estimate of the SSVC that is biased low. For the no precision case, clearly $A = 5$ yields the best results in terms of coverage probability. If one were only interested in generating an initial, or pilot, CI for the steady-state mean of this process without imposing a precision requirement, then it might be desirable to change the default smoothing parameter from $A = 7$ to $A = 5$. Similarly, for the $\pm 15\%$ precision case, the results for $A = 5$ appear to be better than those for $A = 7$. However, for the $\pm 7.5\%$ case, there is significant overcoverage for $A = 5$; and it appears that asymptotically, the default smoothing parameter $A = 7$ produces better results than $A = 5$. For $A = 9$ and $A = 11$, the coverage probabilities for the $\pm 7.5\%$ precision case are also excellent. However, the small sample results for $A = 9$ and $A = 11$ are not as good as those for $A = 5$ and $A = 7$.

In summary, it is evident from Tables 4.1 and 4.2 that while there may be slight differences in the results for the allowable values of A, setting $A = 5, 7, 9,$ or 11 yields acceptable results for this system and WASSP appears to be robust in terms of the smoothing parameter A.

Table 4.1: Performance of **WASSP** using different values of A for the $M/M/1$ queue waiting time process with 90% server utilization and empty-and-idle initial condition. Results are based on 1,000 independent replications of nominal 90% CIs.

Precision Requirement	Performance Measure	Smoothing Parameter			
		$A = 5$	$A = 7$	$A = 9$	$A = 11$
None	CI coverage	87%	83%	82%	80%
	Avg. sample size	13,171	12,956	13,964	13,020
	Max. sample size	128,142	123,102	159,408	123,102
	Avg. CI half-length	3.5545	3.1776	2.8900	2.8387
	Var. CI half-length	3.6149	2.5342	1.8898	1.6919
±15%	CI coverage	89%	84%	83%	82%
	Avg. sample size	112,400	88,782	73,443	66,793
	Max. sample size	931,840	819,248	665,600	396,288
	Avg. CI half-length	1.1047	1.1060	1.1231	1.1390
	Var. CI half-length	0.0414	0.0368	0.0358	0.0342
±7.5%	CI coverage	93%	91%	90%	90%
	Avg. sample size	458,550	371,380	340,070	314,330
	Max. sample size	2,609,152	1,871,936	1,986,560	3,260,416
	Avg. CI half-length	0.5866	0.5914	0.5925	0.5957
	Var. CI half-length	0.0069	0.0066	0.0066	0.0056

Table 4.2: Performance of **WASSP** using different values of A for the $M/M/1$ queue waiting time process with 90% server utilization and empty-and-idle initial condition. Results are based on 1,000 independent replications of nominal 95% CIs.

Precision Requirement	Performance Measure	Smoothing Parameter			
		$A = 5$	$A = 7$	$A = 9$	$A = 11$
None	CI coverage	90%	90%	89%	87%
	Avg. sample size	13,233	13,895	13,454	13,912
	Max. sample size	114,580	159,408	135,232	159,408
	Avg. CI half-length	4.4831	3.8909	3.6634	3.3978
	Var. CI half-length	5.6563	3.6705	2.9545	2.4103
±15%	CI coverage	95%	93%	91%	90%
	Avg. sample size	198,210	140,790	122,300	108,580
	Max. sample size	3,747,840	917,584	587,776	452,608
	Avg. CI half-length	1.1352	1.1436	1.1433	1.1538
	Var. CI half-length	0.0352	0.0324	0.0300	0.0280
±7.5%	CI coverage	98%	97%	95%	94%
	Avg. sample size	793,220	599,070	532,240	473,350
	Max. sample size	5,599,232	3,477,536	2,752,512	1,945,600
	Avg. CI half-length	0.5950	0.5953	0.5970	0.5967
	Var. CI half-length	0.0061	0.0060	0.0055	0.0052

4.2.1 *Validation of Student's t-Ratio Assumptions for the $M/M/1$ Queue Waiting Time Process*

When a confidence interval of the form (3.49) is delivered by $\mathcal{WASSP}$, its validity depends on the auxiliary random variable

$$\mathcal{T} = \frac{\overline{\overline{X}}(m,k) - \mu_X}{\sqrt{\frac{\widehat{\gamma}_X}{n}}} = \frac{\left(\overline{\overline{X}} - \mu_X\right) \Big/ \sqrt{\frac{\gamma_X}{n}}}{\sqrt{\left(\frac{2a\widehat{\gamma}_X}{\gamma_X}\right) \Big/ 2a}} = \frac{\mathcal{Z}}{\sqrt{\mathcal{Q}/2a}} \tag{4.4}$$

having Student's t-distribution with $2a$ degrees of freedom; and in general this distributional requirement is met if the following assumptions hold—

Assumption $\mathbf{A_1}$: The numerator of (4.4) satisfies

$$\mathcal{Z} = \frac{\overline{\overline{X}}(m,k) - \mu_X}{\sqrt{\frac{\gamma_X}{n}}} \sim N(0,1). \tag{4.5}$$

Assumption $\mathbf{A_2}$: The squared denominator of (4.4) satisfies

$$\mathcal{Q} = \frac{2a\widehat{\gamma}_X}{\gamma_X} \sim \chi^2(2a). \tag{4.6}$$

Assumption $\mathbf{A_3}$: The numerator $\mathcal{Z}$ and the squared denominator $\mathcal{Q}$ are independent.

In the following experimental study of the stochastic behavior of $\mathcal{T}$, $\mathcal{Z}$, and $\mathcal{Q}$ when $\mathcal{WASSP}$ is applied to the waiting time process in the $M/M/1$ queue, we took $a = 3$ so that the smoothing parameter $A = 7$ and $\widehat{\gamma}_X$, the $\mathcal{WASSP}$-based estimator of γ_X, has $2a = 6$ degrees of freedom. To determine if the $\mathcal{T}$-values generated by $\mathcal{WASSP}$ do in fact follow a t-distribution with $2a = 6$ degrees of freedom, we generated the following plots, each based on 400 independent replications of $\mathcal{WASSP}$ with the confidence-interval specifications of no precision, $\pm 15\%$ precision, and $\pm 7.5\%$ precision and with nominal confidence levels of 90% and 95%:

(i) A plot showing the histogram of $\mathcal{T}$-values superimposed on the probability density function (p.d.f.) of the t-distribution with 6 degrees of freedom, together with a plot showing the empirical c.d.f. of the $\mathcal{T}$-values superimposed on the c.d.f. of the t-distribution with 6 degrees of freedom.

(ii) A plot showing the histogram of $\mathcal{Z}$-values superimposed on the p.d.f. of the $N(0,1)$ distribution, together with a plot showing the empirical c.d.f. of the $\mathcal{Z}$-values superimposed on the c.d.f. of the $N(0,1)$ distribution.

(iii) A plot showing the histogram of $\mathcal{Q}$-values superimposed on the p.d.f. of the $\chi^2(6)$ distribution, together with a plot showing the empirical c.d.f. of the $\mathcal{Q}$-values superimposed on the c.d.f. of the $\chi^2(6)$ distribution.

Figures 4.2–4.7 display the plots described in items (i)–(iii) above. To generate these plots, first we had to compute the theoretical SSVC γ_X of the $M/M/1$ queue waiting time process. From [15] we have in general that

$$\gamma_X = \frac{\varrho^3(\varrho^3 - 4\varrho^2 + 5\varrho + 2)}{(1-\varrho)^4}\,\mathrm{E}^2[B_i], \tag{4.7}$$

where ϱ is the traffic intensity and B_i is the interarrival time between the $(i-1)$st and ith customers. Substituting $\varrho = 0.9$ and $\mathrm{E}[B_i] = 10/9$, we have $\gamma_X = 35{,}901$.

Several observations can be made about the plots in Figures 4.2–4.7. First, from the middle row of plots in Figures 4.2 and 4.5 where the empirical distributions of $\mathcal{Z}$ are compared to the $N(0,1)$ distribution, we see that the distribution of $\mathcal{Z}$-values is slightly skewed. This indicates that for the no precision case, the point estimate of the mean, $\overline{\overline{X}}(m,k)$, is biased. This bias is likely the result of a combination of two different factors. First, the mean $\overline{\overline{X}}(m,k)$ is affected by initialization bias. For the no precision case, it appears that **WASSP** has not been able to eliminate entirely the effects of system warm-up on the estimate of the mean. Second, the simulation run length n is random. This implies that

$$\overline{\overline{X}}(m,k) = \overline{X}(n) = \frac{\sum_{i=1}^{n} X_i}{n} \tag{4.8}$$

is a ratio of two random variables; and for such a ratio estimator, in general we have

$$\mathrm{E}\Big[\,\overline{X}(n)\,\Big] \neq \frac{\mathrm{E}[\sum_{i=1}^{n} X_i]}{\mathrm{E}[n]} \tag{4.9}$$

(see Sections 6.3 and 6.8 of [12]). It is clear that when **WASSP** is applied to the process $\{X_i : i = 1, 2, \ldots\}$, the resulting final sample size n is a *stopping time* for the process ([49], p. 229); and thus, for example, in the special case that the $\{X_i\}$ are i.i.d., then Wald's equation ([49], Corollary 7.2.3) implies that

$$\mathrm{E}\left[\sum_{i=1}^{n} X_i\right] = \mathrm{E}[n]\,\mu_X \tag{4.10}$$

so that in this situation we have

$$\mathrm{E}\Big[\,\overline{\overline{X}}(m,k)\,\Big] \neq \mu_X. \tag{4.11}$$

From the preceding discussion it follows that WASSP's final point estimator $\overline{\overline{X}}(m,k)$ is generally biased. Asymptotically as the precision requirement $r^* \to 0$, the sample size $n \to \infty$, and the mean $\overline{X}(n)$ converges with probability one to μ_X, and the bias due to the randomness of the simulation run length n goes to zero. Furthermore, as the simulation run length increases, the effect of system warm-up on the estimate of the mean decreases. This can be seen clearly in the middle row of plots in Figures 4.4 and 4.7. At the $\pm 7.5\%$ precision level, a large amount of the bias affecting the mean has been eliminated.

A second observation that can be made about the plots in Figures 4.2–4.7 is that for the most part, the $\mathcal{Q}$-values do closely follow the $\chi^2(6)$ distribution. In some cases, however (as in Figure 4.7), the $\mathcal{Q}$-values do not appear to track the $\chi^2(6)$ distribution as closely as in others. We can expect the behavior of the $\mathcal{Q}$-values to be erratic at times since $\mathcal{Q}$ is a random variable based on the sample covariance structure (that is, second-degree sample moments) of the simulation-generated time series $\{X_i : i = 1, 2, \ldots, n\}$.

A final observation is that overall the $\mathcal{T}$-values closely follow the t-distribution with 6 degrees of freedom; and asymptotically, the Assumptions A_1 and A_2 required for the random variable $\mathcal{T}$ to have a t-distribution with 6 degrees of freedom appear to hold (at least to a reasonable approximation) for the $M/M/1$ waiting time process.

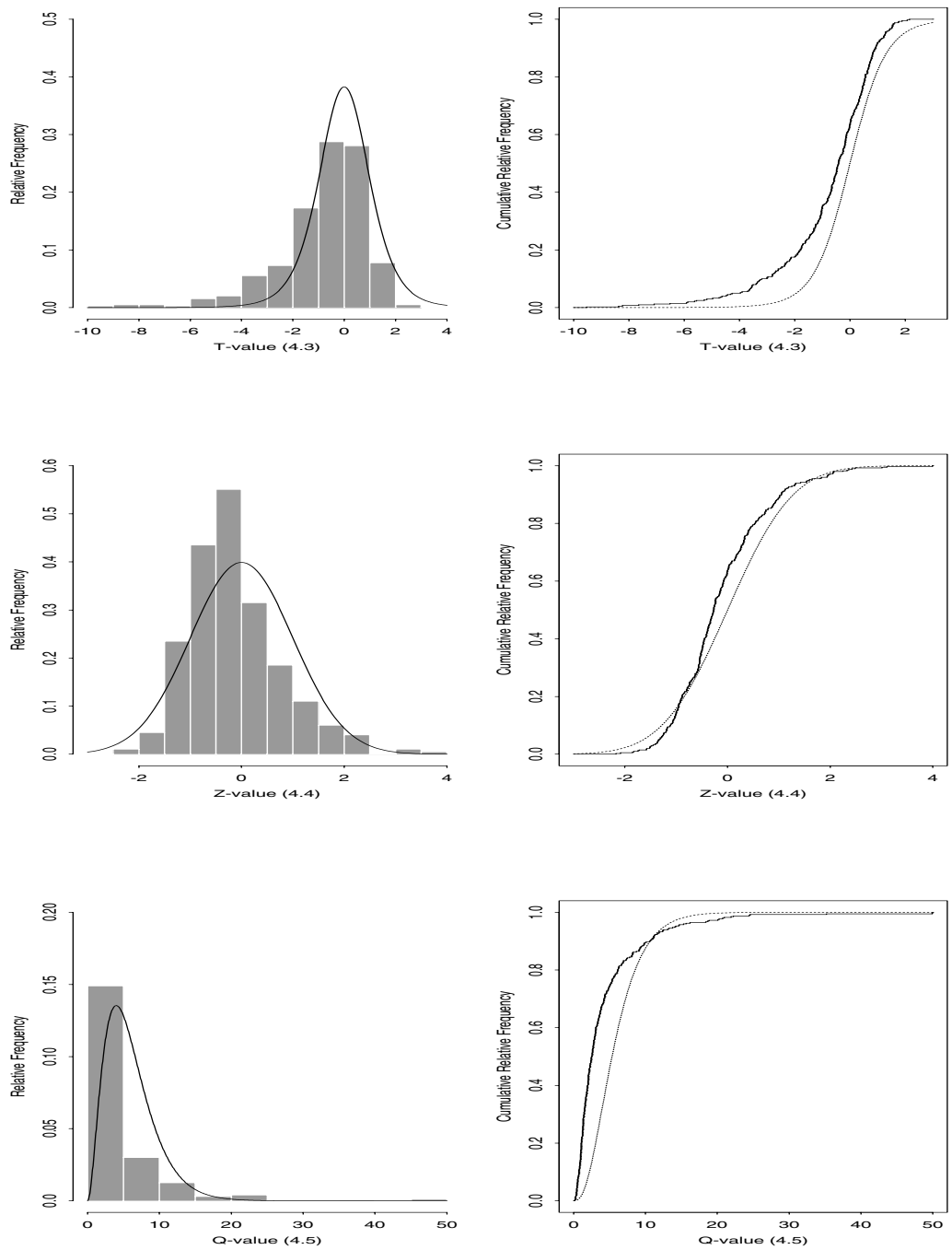

Figure 4.2: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 6 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(6)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** to estimate mean $M/M/1$ waiting time using a 90% CI with no precision requirement.

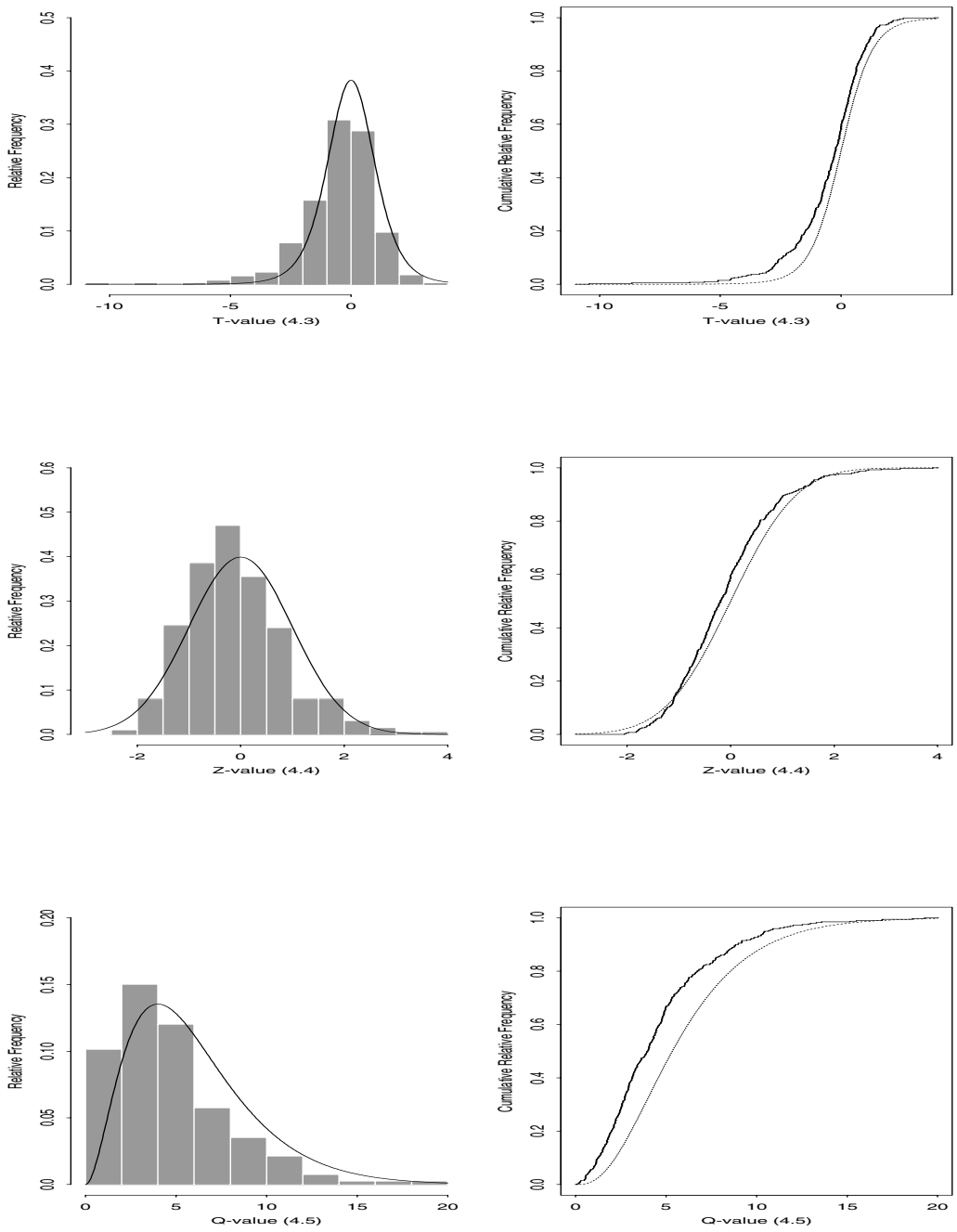

Figure 4.3: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 6 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(6)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** to estimate mean $M/M/1$ waiting time using a 90% CI with $\pm 15\%$ precision requirement.

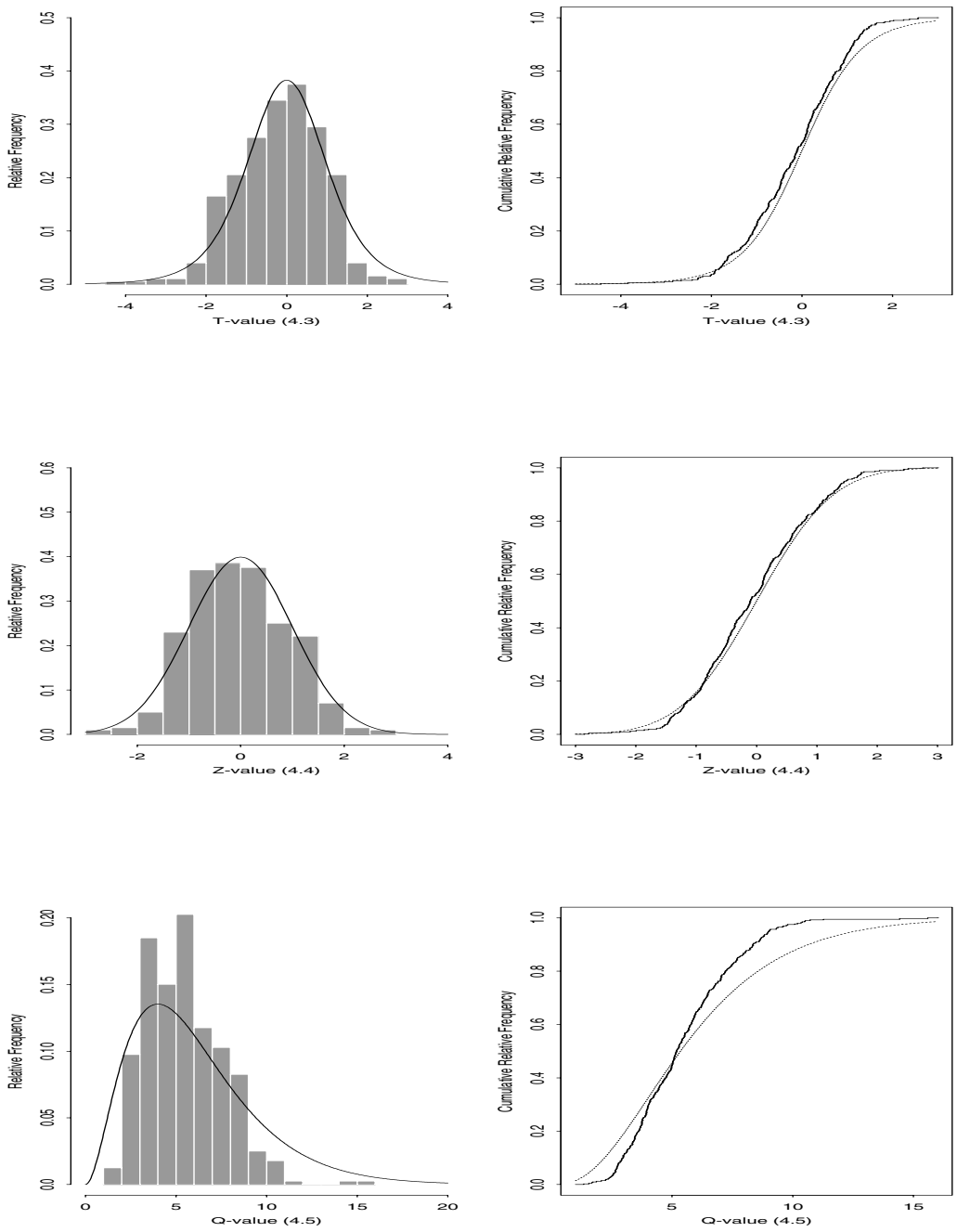

Figure 4.4: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 6 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(6)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** to estimate mean $M/M/1$ waiting time using a 90% CI with $\pm 7.5\%$ precision requirement.

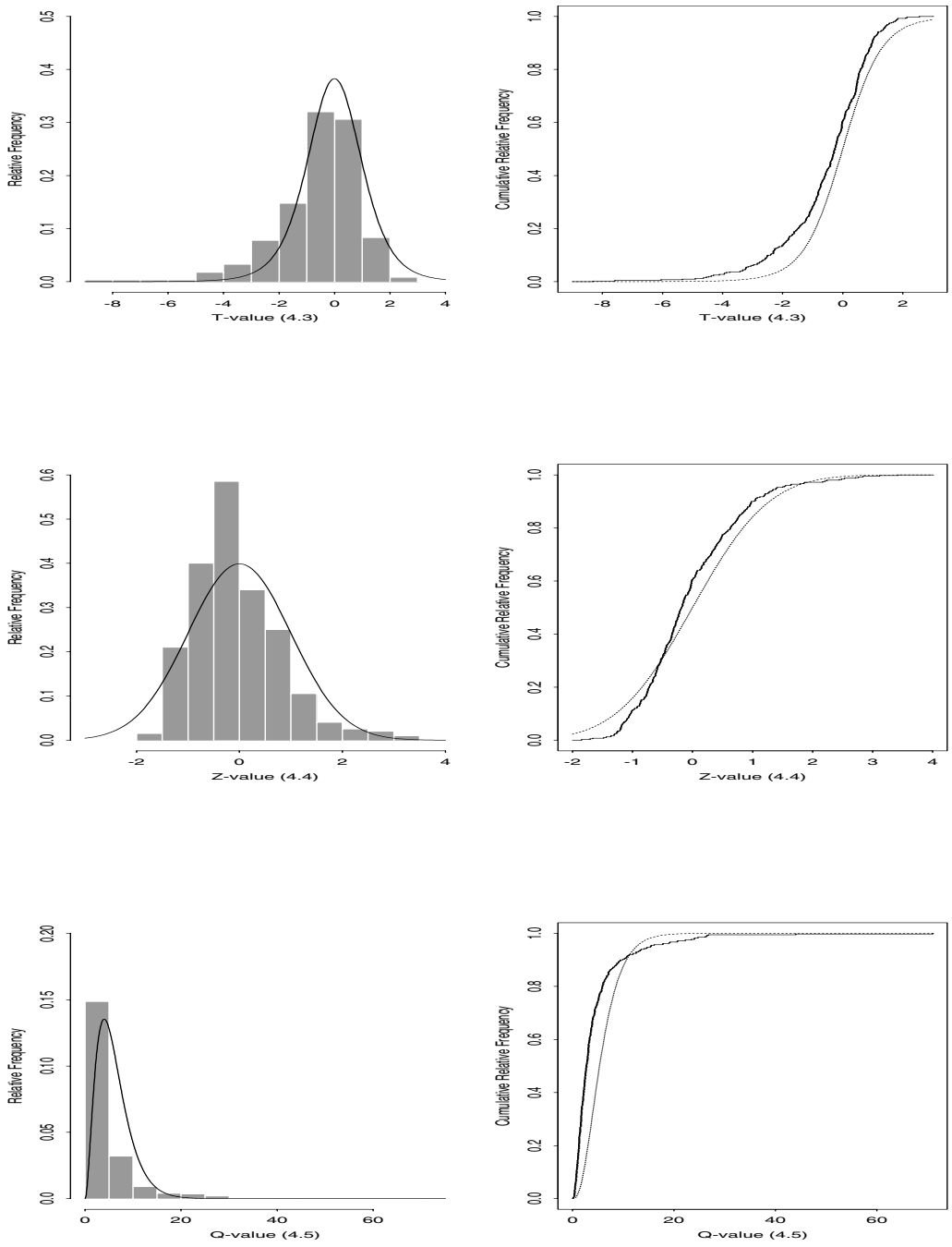

Figure 4.5: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 6 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(6)$ distribution, respectively. Results obtained from 400 i.i.d. runs of $\mathcal{WASSP}$ to estimate $M/M/1$ waiting time using a 95% CI with no precision requirement.

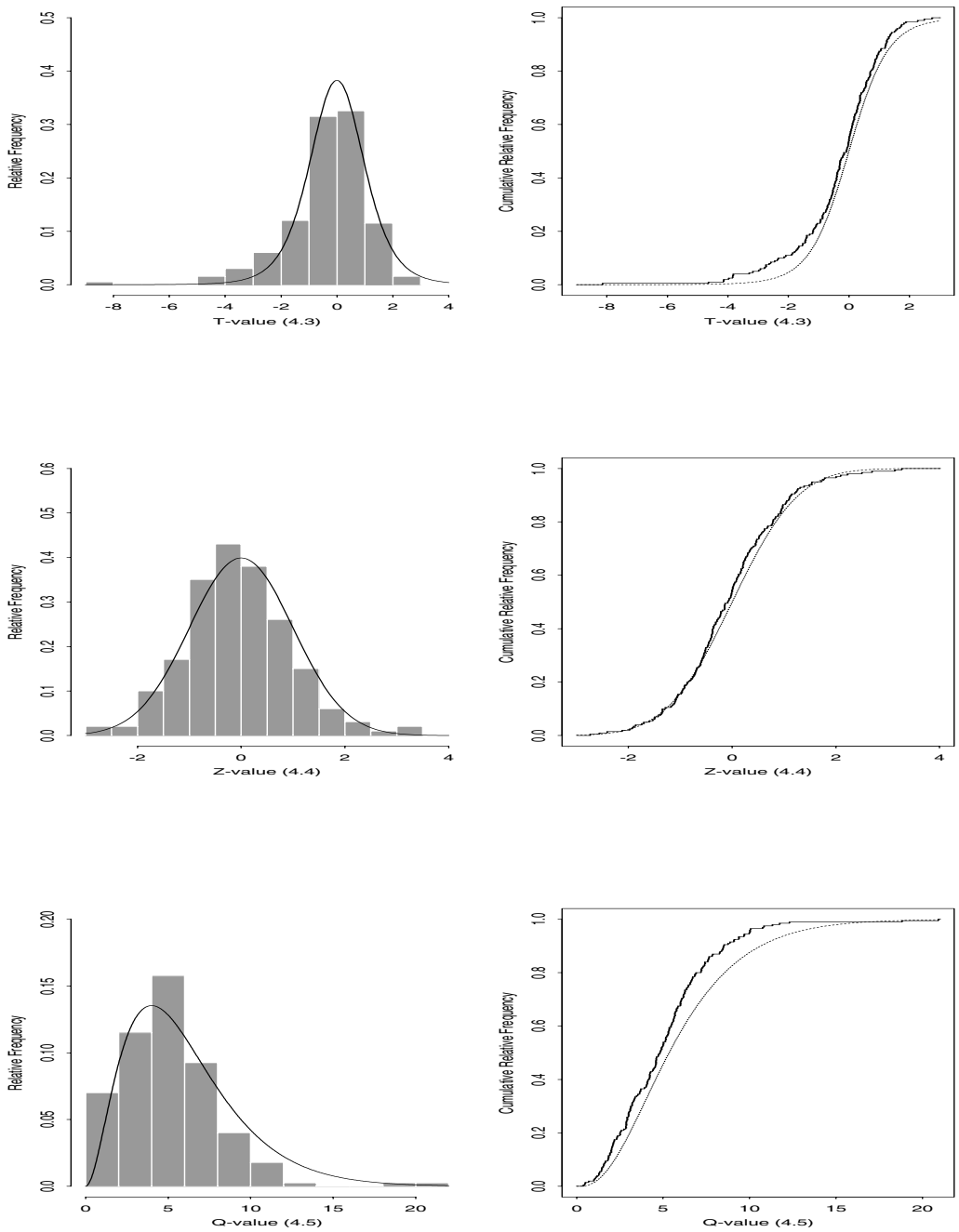

Figure 4.6: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 6 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(6)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** to estimate mean $M/M/1$ waiting time using a 95% CI with $\pm 15\%$ precision requirement.

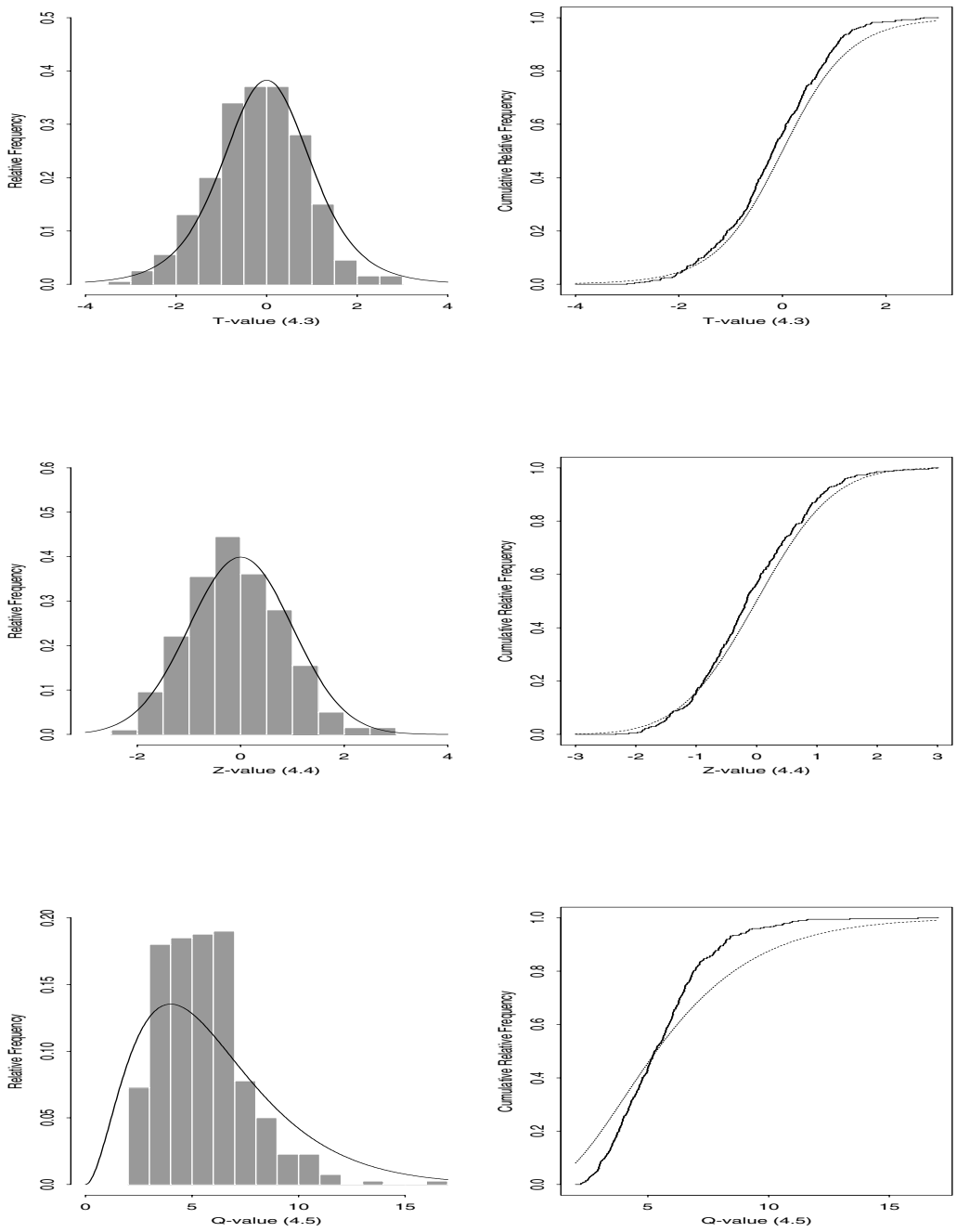

Figure 4.7: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 6 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(6)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** to estimate mean $M/M/1$ waiting time using a 95% CI with $\pm 7.5\%$ precision requirement.

To verify Assumption A_3 required for the random variable $\mathcal{T}$ to have Student's t-distribution with $2a$ degrees of freedom, we estimated the correlation between $\overline{\overline{X}}(m,k)$ and $\widehat{\gamma}_X$ for 400 replications of $\mathcal{WASSP}$ with confidence levels of 90% and 95% and with the smoothing parameter $A = 7$. In Tables 4.3 and 4.4, we see that there is significant correlation between $\overline{\overline{X}}(m,k)$ and $\widehat{\gamma}_X$, strongly suggesting that in this application of $\mathcal{WASSP}$, Assumption A_3 does not hold so $\mathcal{Z}$ and $\mathcal{Q}$ are not independent. As the precision requirement $r^* \to 0$, the correlation between $\overline{\overline{X}}(m,k)$ and $\widehat{\gamma}_X$ decreases very slowly. This may be a partial explanation for the undercoverage seen in some of the small sample cases (that is, for the no precision and the $\pm 15\%$ precision cases) for the $M/M/1$ system.

Table 4.3: Correlation between $\overline{\overline{X}}(m,k)$ and $\widehat{\gamma}_X$ for 400 independent replications of $\mathcal{WASSP}$ to estimate mean $M/M/1$ waiting time using a 90% CI.

Precision Requirement	Corr $\left[\overline{\overline{X}}(m,k), \widehat{\gamma}_X\right]$
None	0.5829
$\pm 15\%$	0.5778
$\pm 7.5\%$	0.4922

Table 4.4: Correlation between $\overline{\overline{X}}(m,k)$ and $\widehat{\gamma}_X$ for 400 independent replications of $\mathcal{WASSP}$ to estimate mean $M/M/1$ waiting time using a 95% CI.

Precision Requirement	Corr $\left[\overline{\overline{X}}(m,k), \widehat{\gamma}_X\right]$
None	0.6540
$\pm 15\%$	0.5605
$\pm 7.5\%$	0.4531

4.2.2 *Comparison of WASSP and ASAP2 for the $M/M/1$ Queue Waiting Time Process*

Tables 4.5 and 4.6 show a comparison of the performance of $\mathcal{WASSP}$ (using $A = 7$) and ASAP2 for the $M/M/1$ queue waiting time process. The coverage probabilities for ASAP2 have a standard error of approximately 1.5% for nominal 90% confidence intervals and a standard error of approximately 1% for nominal 95% confidence intervals since only 400 replications of ASAP2 were performed. The coverage probabilities for $\mathcal{WASSP}$ have a standard error of 0.95%

for nominal 90% confidence intervals and a standard error of 0.69% for nominal 95% confidence intervals since we performed 1000 replications of **WASSP**. From these two tables, it is clear that in the no precision case ASAP2 has a higher coverage probability than **WASSP**. However, the variance of the confidence interval half-length is markedly higher for ASAP2 than for **WASSP**. This suggests that for the no precision case, **WASSP** produces much more stable confidence intervals than ASAP2. Furthermore, in the no precision case the confidence level has no effect on the average and maximum sample size. Therefore, these two values should be the same for nominal 90% and 95% CIs if the procedure being tested is applied to the same set of data for both confidence levels. We see from Tables 4.5 and 4.6 that this is the case for ASAP2. However, the average and maximum sample sizes are different for nominal 90% and 95% CIs generated by **WASSP**. This is because common random numbers were not used to generate the data sets used to test **WASSP** and therefore the algorithm was applied to different sets of data for nominal 90% CIs and nominal 95% CIs.

For the $\pm 15\%$ precision case, ASAP2 appears to perform slightly better in terms of confidence interval coverage probability than **WASSP**. However, from Tables 4.1 and 4.2 we see that for smoothing parameter $A = 5$, **WASSP** produces results that are comparable to those for ASAP2 when the relative precision requirement $r^* = 0.15$ is imposed. Clearly, once a precision requirement is imposed, ASAP2 produces confidence intervals that exhibit the same stability as the confidence intervals produced by **WASSP**, as can be seen from the variance of the confidence interval half-length.

For the case of $\pm 7.5\%$ precision, Tables 4.5 and 4.6 indicate that **WASSP** and ASAP2 perform essentially the same, suggesting that as the relative precision requirement $r^* \to 0$, **WASSP** and ASAP2 produce comparable results for this test process in terms of coverage probability, average confidence interval half-length, and variance of the confidence interval half-length. The average sample size is significantly higher for **WASSP** than for ASAP2, however.

Table 4.5: Performance of **WASSP** (using $A = 7$) and ASAP2 for the $M/M/1$ queue waiting time process with 90% server utilization and empty-and-idle initial condition. Results are based on independent replications of nominal 90% CIs.

Precision Requirement	Performance Measure	Procedure	
		WASSP	ASAP2
None	# replications	1,000	400
	CI coverage	83%	88%
	Avg. sample size	12,956	22,554
	Max. sample size	123,102	131,072
	Avg. CI half-length	3.1776	6.44
	Var. CI half-length	2.5342	167.0
±15%	# replications	1,000	400
	CI coverage	84%	90%
	Avg. sample size	88,782	93,374
	Max. sample size	819,248	260,624
	Avg. CI half-length	1.1060	1.18
	Var. CI half-length	0.0368	0.025
±7.5%	# replications	1,000	400
	CI coverage	91%	92%
	Avg. sample size	371,380	281,022
	Max. sample size	1,871,936	796,076
	Avg. CI half-length	0.5914	0.630
	Var. CI half-length	0.0060	0.002

Table 4.6: Performance of **WASSP** (using $A = 7$) and ASAP2 for the $M/M/1$ queue waiting time process with 90% server utilization and empty-and-idle initial condition. Results are based on independent replications of nominal 95% CIs.

Precision Requirement	Performance Measure	Procedure	
		WASSP	ASAP2
None	# replications	1,000	400
	CI coverage	90%	90%
	Avg. sample size	13,895	22,554
	Max. sample size	159,408	131,072
	Avg. CI half-length	3.8909	8.3
	Var. CI half-length	3.6705	350.0
±15%	# replications	1,000	400
	CI coverage	93%	95%
	Avg. sample size	140,790	126,839
	Max. sample size	917,584	364,672
	Avg. CI half-length	1.1436	1.204
	Var. CI half-length	0.0324	0.02
±7.5%	# replications	1,000	400
	CI coverage	97%	96%
	Avg. sample size	599,070	382,040
	Max. sample size	3,477,536	856,256
	Avg. CI half-length	0.5953	0.633
	Var. CI half-length	0.0060	0.002

4.2.3 *Comparison of* WASSP *and Heidelberger and Welch's Spectral Method for the* $M/M/1$ *Queue Waiting Time Process*

Tables 4.7 and 4.8 show the results for **WASSP** and the spectral method of Heidelberger and Welch (H&W) for nominal 90% and 95% confidence intervals, respectively. Since it is possible that the Heidelberger and Welch algorithm could run out of data before the precision requirement is satisfied, we included in Tables 4.7 and 4.8 the overall coverage (for all 1000 replications, whether or not the precision requirement was met) as well as the coverage for those CIs that satisfied the precision requirement. Furthermore, Tables 4.7 and 4.8 also report the estimated mean square error of the grand mean,

$$\widehat{\mathrm{MSE}}\Big[\,\overline{\overline{X}}(m,k)\,\Big] = \frac{\displaystyle\sum_{l=1}^{1000}\Big[\,\overline{\overline{X}}_l(m_l,k_l) - \mu_X\,\Big]^2}{1000}, \tag{4.12}$$

where on replication l of **WASSP** or the H&W procedure, $\overline{\overline{X}}_l(m_l,k_l)$ denotes the delivered grand mean based on k_l batches of size m_l for $l = 1,\ldots,1000$; and the corresponding estimated standard error of (4.12),

$$\widehat{\mathrm{SE}}\Big\{\widehat{\mathrm{MSE}}\Big[\,\overline{\overline{X}}(m,k)\,\Big]\Big\} = \left(\frac{\displaystyle\sum_{l=1}^{1000}\left\{\Big[\,\overline{\overline{X}}_l(m_l,k_l) - \mu_X\,\Big]^2 - \widehat{\mathrm{MSE}}\Big[\,\overline{\overline{X}}(m,k)\,\Big]\right\}^2}{1000\times 999}\right)^{1/2}. \tag{4.13}$$

The statistics (4.12) and (4.13) provide an indication of the amount of bias associated with the final point estimator $\overline{\overline{X}}(m,k)$ delivered by each procedure.

For the no precision case, $\widehat{\mathrm{MSE}}[\overline{\overline{X}}(m,k)]$ for Heidelberger and Welch's method is nearly double that for **WASSP**. This can be attributed to the fact that the Heidelberger and Welch algorithm does not address the warm-up problem at all; moreover, in general the randomness of the final sample size used by the H&W procedure can introduce an additional bias into $\overline{\overline{X}}(m,k)$ for the same reasons elaborated in the discussion leading up to display (4.11) above. Once a precision requirement is imposed and the sample size begins to increase, $\widehat{\mathrm{MSE}}[\overline{\overline{X}}(m,k)]$ for Heidelberger and Welch's method begins to decrease with the decreasing effect of initialization bias and ratio-estimator bias on $\overline{\overline{X}}(m,k)$. From the values of $\widehat{\mathrm{MSE}}[\overline{\overline{X}}(m,k)]$ for **WASSP**, it can be seen (especially in the no precision case) that $\overline{\overline{X}}(m,k)$ is slightly affected by initialization bias as well as ratio-estimator bias. As the precision requirement goes to zero, the effects of these two types of bias on **WASSP**'s estimate of the mean decrease, as reflected in the values of $\widehat{\mathrm{MSE}}[\overline{\overline{X}}(m,k)]$ for the precision levels of $\pm 15\%$ and $\pm 7.5\%$.

From Tables 4.7 and 4.8, we also see that Heidelberger and Welch's method consistently

requires smaller average sample sizes than WASSP. However, it is difficult to compare the average sample sizes for the precision levels of $\pm 15\%$ and $\pm 7.5\%$ since not all 1000 CIs delivered by the Heidelberger and Welch method satisfied the precision requirement. It is also evident from Tables 4.7 and 4.8 that the coverage of the confidence intervals delivered by Heidelberger and Welch's method is consistently much less than the coverage of the confidence intervals generated by WASSP.

Table 4.7: Performance of $\mathcal{WASSP}$ (using $A = 7$) and Heidelberger and Welch's spectral method for the $M/M/1$ queue waiting time process with 90% server utilization and empty-and-idle initial condition. Results are based on 1,000 independent replications of nominal 90% CIs.

Precision Requirement	Performance Measure	Procedure	
		$\mathcal{WASSP}$	H&W
None	# replications.	1,000	1,000
	Overall CI coverage	83.2%	78.6%
	Avg. sample size	12,956	4,096
	Max. sample size	123,102	4,096
	Avg. CI half-length	3.1776	4.0415
	Var. CI half-length	2.5342	4.6892
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	3.6379	7.0497
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.2815	0.6062
	# replications satisfying precision requirement	1,000	1,000
	Coverage for CIs satisfying precision requirement	83.2%	78.6%
±15%	# replications	1,000	1,000
	Overall CI coverage	83.6%	79.6%
	Avg. sample size	88,782	65,282
	Max. sample size	819,248	314,464
	Avg. CI half-length	1.1060	1.3154
	Var. CI half-length	0.0368	0.3765
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	0.6575	0.8940
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.0368	0.0442
	# replications satisfying precision requirement	1,000	767
	Coverage for CIs satisfying precision requirement	83.6%	75.0%
±7.5%	# replications	1,000	1,000
	Overall CI coverage	90.8%	84.10%
	Avg. sample size	371,380	298,860
	Max. sample size	1,871,936	1,216,420
	Avg. CI half-length	0.5914	0.6852
	Var. CI half-length	0.0060	0.0599
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	0.1199	0.2186
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.0066	0.0117
	# replications satisfying precision requirement	1,000	673
	Coverage for CIs satisfying precision requirement	90.8%	79.35%

Table 4.8: Performance of $\mathcal{WASSP}$ (using $A = 7$) and Heidelberger and Welch's spectral method for the $M/M/1$ queue waiting time process with 90% server utilization and empty-and-idle initial condition. Results are based on 1,000 independent replications of nominal 95% CIs.

Precision Requirement	Performance Measure	Procedure	
		$\mathcal{WASSP}$	H&W
None	# replications	1,000	1,000
	Overall CI coverage	90.0%	84.8%
	Avg. sample size	13,895	4,096
	Max. sample size	159,408	4,096
	Avg. CI half-length	3.8909	5.1046
	Var. CI half-length	3.6705	7.3659
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	3.3902	7.3704
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.2118	0.8204
	# replications satisfying precision requirement	1,000	1,000
	Coverage for CIs satisfying precision requirement	90.0%	84.8%
±15%	# replications	1,000	1,000
	Overall CI coverage	92.8%	87.0%
	Avg. sample size	140,790	104,290
	Max. sample size	917,584	482,306
	Avg. CI half-length	1.1436	1.3521
	Var. CI half-length	0.0324	0.3791
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	0.3861	0.6325
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.0237	0.0313
	# replications satisfying precision requirement	1,000	717
	Coverage for CIs satisfying precision requirement	92.8%	83.68%
±7.5%	# replications	1,000	1,000
	Overall CI coverage	96.5%	92.6%
	Avg. sample size	599,070	458,310
	Max. sample size	3,477,536	1,841,421
	Avg. CI half-length	0.5953	0.6910
	Var. CI half-length	0.0060	0.0523
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	0.0743	0.1378
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}\|[\overline{\overline{X}}(m,k)]\}$	0.0040	0.0077
	# replications satisfying precision requirement	1,000	673
	Coverage for CIs satisfying precision requirement	96.5%	89.6%

4.3. The First-Order Autoregressive Process

The next test process used in the performance evaluation of **WASSP** is the first-order autoregressive process. Let $\{\delta_i : i = 1, 2, \ldots\}$ be a white noise process that is randomly sampled from $N(0, \sigma_\delta^2)$. We define an autoregressive process of order one as follows,

$$X_i = \mu_X + \rho(X_{i-1} - \mu_X) + \delta_i, \quad \text{for} \quad i = 1, 2, \ldots, \tag{4.14}$$

where μ_X is the steady-state mean of the process and ρ is the lag-one correlation of the process in steady-state operation. To generate the AR(1) process $\{X_i : i = 1, \ldots, n\}$ from (4.14), we first set $X_0 = 0$, corresponding to an empty-and-idle initial condition. We then set the autoregressive parameter $\rho = 0.995$, the mean $\mu_X = 100$, and the variance of the white noise process $\sigma_\delta^2 = 1$. One of the most difficult aspects of this test proces is the exceptionally long initial transient period.

The spectrum of the AR(1) process (4.14) in steady-state operations is

$$p_X(\omega) = \frac{\sigma_\delta^2}{1 - 2\rho\cos(2\pi\omega) + \rho^2} \quad \text{for} \quad \omega \in [-\frac{1}{2}, \frac{1}{2}]; \tag{4.15}$$

see equation (6.2.20) of [33]. For the selected AR(1) process (4.14) with $\mu_X = 100$, $\rho = 0.995$, $\sigma_\delta^2 = 1$, $\sigma_X^2 = \sigma_\delta^2/(1 - \rho^2) = 100.25$, and steady-state initial condition $X_0 \sim N(\mu_X, \sigma_X^2)$, Figure 4.8 depicts plots of the spectrum $p_X(\omega)$ and the log-spectrum $\ln[p_X(\omega)]$ for $\omega \in [-1/2, 1/2]$. Figure 4.8 reveals that both the spectrum and log-spectrum of the selected AR(1) process exhibit peakedness at zero frequency similar to that exhibited by their counterparts for the $M/M/1$ waiting time process. Thus the given AR(1) process provides another severe test of **WASSP**'s wavelet-based technique for estimation of the spectrum at zero frequency.

Tables 4.9 and 4.10 show the performance of **WASSP** for the AR(1) process described above using the smoothing parameter values A=5, 7, 9, and 11. The results are based on 1,000 independent replications of nominal 90% and 95% CIs. From these tables, it is clear that as the value of A increases, the coverage probabilities decrease slightly. As previously observed for the $M/M/1$ process, the extreme peakedness of the log-spectrum of the selected AR(1) process in the neighborhood of zero frequency results in progressively greater underestimation of the SSVC with increasing values of **WASSP**'s smoothing parameter A. However, as with the $M/M/1$ queue waiting time process, Tables 4.9 and 4.10 also show the robustness of **WASSP** in terms of the value of the smoothing parameter A since setting A=5, 7, 9, or 11 yields acceptable results for this process.

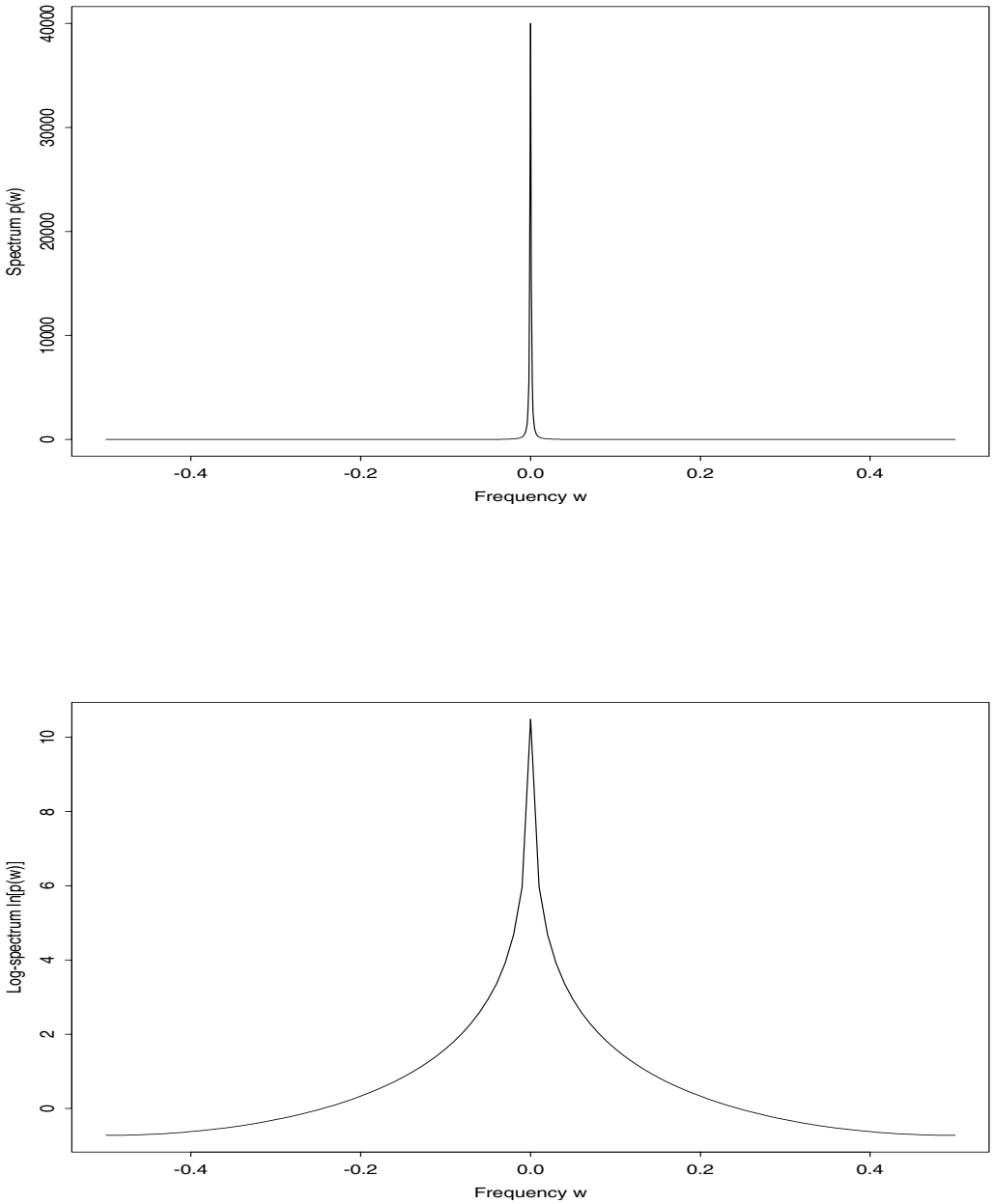

Figure 4.8: Plots for frequency $\omega \in [-\frac{1}{2}, \frac{1}{2}]$ of the spectrum $p_X(\omega)$ (top panel) and the log-spectrum $\ln[p_X(\omega)]$ (bottom panel) for the steady-state AR(1) process (4.14) with $\mu_X = 100$, $\rho = 0.995$, and $\sigma_\delta^2 = 1$.

Table 4.9: Performance of **WASSP** using different values of A for the AR(1) process (4.14) with $\mu_X = 100$, $X_0 = 0$, $\rho = 0.995$, and $\sigma_\delta^2 = 1$. Results are based on 1,000 independent replications of nominal 90% CIs.

Precision	Performance	Smoothing Parameter			
Requirememt	Measure	$A = 5$	$A = 7$	$A = 9$	$A = 11$
None	CI coverage	94%	90%	87%	85%
	Avg. sample size	9,885	9,703	9,960	9,715
	Max. sample size	21,504	22,120	27,468	29,736
	Avg. CI half-length	6.0418	5.3424	4.9107	4.6365
	Var. CI half-length	2.2874	1.7498	1.2413	1.0990
±15%	CI coverage	93%	92%	87%	84%
	Avg. sample size	9,715	9,739	9,769	9,752
	Max. sample size	29,736	29,736	21,504	29,736
	Avg. CI half-length	6.1614	5.3423	4.8754	4.5442
	Var. CI half-length	2.4859	1.5663	1.3592	1.0379
±7.5%	CI coverage	93%	92%	89%	85%
	Avg. sample size	9,756	9,899	9,885	9,960
	Max. sample size	29,736	30,090	21,504	27,468
	Avg. CI half-length	5.7021	5.3037	4.8869	4.5663
	Var. CI half-length	1.3305	1.3122	1.2303	0.9708
±3.75%	CI coverage	91%	86%	87%	86%
	Avg. sample size	17,269	13,470	11,897	11,080
	Max. sample size	52,224	49,152	32,512	29,736
	Avg. CI half-length	3.1739	3.2235	3.2890	3.3300
	Var. CI half-length	0.1924	0.1453	0.1230	0.1094

Table 4.10: Performance of **WASSP** using different values of A for the AR(1) process (4.14) with $\mu_X = 100$, $X_0 = 0$, $\rho = 0.995$, and $\sigma_\delta^2 = 1$. Results are based on 1,000 independent replications of nominal 95% CIs.

Precision Requirement	Performance Measure	Smoothing Parameter			
		$A = 5$	$A = 7$	$A = 9$	$A = 11$
None	CI coverage	97%	95%	95%	93%
	Avg. sample size	9,769	9,834	9,781	9,809
	Max. sample size	21,504	29,736	24,174	29,736
	Avg. CI half-length	7.8435	6.7748	6.1069	5.5963
	Var. CI half-length	4.1196	2.7061	2.0884	1.5721
±15%	CI coverage	98%	95%	94%	91%
	Avg. sample size	9,885	9,814	9,715	9,752
	Max. sample size	21,504	24,120	29,736	29,736
	Avg. CI half-length	7.8686	6.7411	6.1871	5.5864
	Var. CI half-length	3.8799	2.6902	2.1548	1.5686
±7.5%	CI coverage	96%	94%	93%	92%
	Avg. sample size	10,378	9,941	9,777	9,585
	Max. sample size	27,468	30,208	21,504	22,592
	Avg. CI half-length	6.2788	6.1240	5.7928	5.5829
	Var. CI half-length	0.8404	1.0529	1.2285	1.2269
±3.75%	CI coverage	98%	95%	94%	92%
	Avg. sample size	28,893	20,868	17,288	14,990
	Max. sample size	88,064	68,608	65,536	52,096
	Avg. CI half-length	3.2704	3.2908	3.3212	3.3841
	Var. CI half-length	0.1936	0.1393	0.1255	0.0850

4.3.1 *Validation of Student's t-Ratio Assumptions for the AR(1) Process*

To validate the assumptions required to ensure the auxiliary random variable $\mathcal{T}$ defined in (4.4) will possess Student's t-distribution with $2a = 6$ degrees of freedom when we set $A = 7$ and apply **WASSP** to the AR(1) process, we generated the same types of plots as for the $M/M/1$ queue waiting time process. As detailed below, when **WASSP** was applied to the AR(1) process (4.14) with $\mu_X = 100$, $X_0 = 0$, $\rho = 0.995$, and $\sigma_\delta^2 = 1$, the procedure required nearly the same average sample sizes and generated not only confidence intervals but also $\mathcal{T}$-, $\mathcal{Z}$-, and $\mathcal{Q}$-values with nearly the same performance characteristics at all three previously selected levels of precision—namely, no precision, ±15% precision, and ±7.5% precision. To provide a more comprehensive experimental validation of Assumptions A_1–A_3 when **WASSP** is applied to an AR(1) process, we also included the case of ±3.75% precision; and the resulting plots for all four precision levels are displayed in Figures 4.9–4.16. To generate these plots, we first computed the theoretical SSVC for the AR(1) system as follows [8],

$$\gamma_X = \frac{\sigma_\delta^2}{(1-\rho)^2} = \frac{1}{(1-0.995)^2} = 40{,}000.$$

From these plots, it appears that Assumptions A_1 and A_2 are satisfied. It is worth noting, however, that it is still possible to see some skewness in the distribution of $\mathcal{Z}$-values, as in the middle row of plots in Figure 4.9. As with the $M/M/1$ system, this bias is likely a combination of the effects of system warm-up and the randomness of the simulation run length n.

In general for the plots in Figures 4.9–4.11 and Figures 4.13–4.15, only slight differences can be detected between the cases defined by the three associated precision levels (no precision, ±15% precision, and ±7.5% precision). This can be attributed to the fact that the average sample size for each of the three precision levels is approximately the same, as can be seen in the column for $A = 7$ in Tables 4.9 and 4.10. From Table 4.9, the average CI half-length is 5.4728 for the no precision case, indicating that with no precision requirement for nominal 90% CIs, **WASSP** is delivering CIs that are already within about ±6% of the mean. From Table 4.10, the average CI half-length is 6.8943 for the no precision case, indicating that without a precision requirement **WASSP** is delivering nominal 95% CIs that are already within about ±7% of the mean. Therefore, there is going to be little difference in the results for the three previously selected precision levels.

Finally, to validate Assumption A_3 required for $\mathcal{T}$ to have a t-distribution with 6 degrees of freedom—namely that $\mathcal{Z}$ and $\mathcal{Q}$ are independent, we computed the sample correlation between $\overline{\overline{X}}(m,k)$ and $\widehat{\gamma}_X$ for 400 independent replications of **WASSP** applied to the AR(1) process. As shown in Tables 4.11 and 4.12, the correlations between $\overline{\overline{X}}(m,k)$ and $\widehat{\gamma}_X$ are small, indicating approximate validity of Assumption A_3.

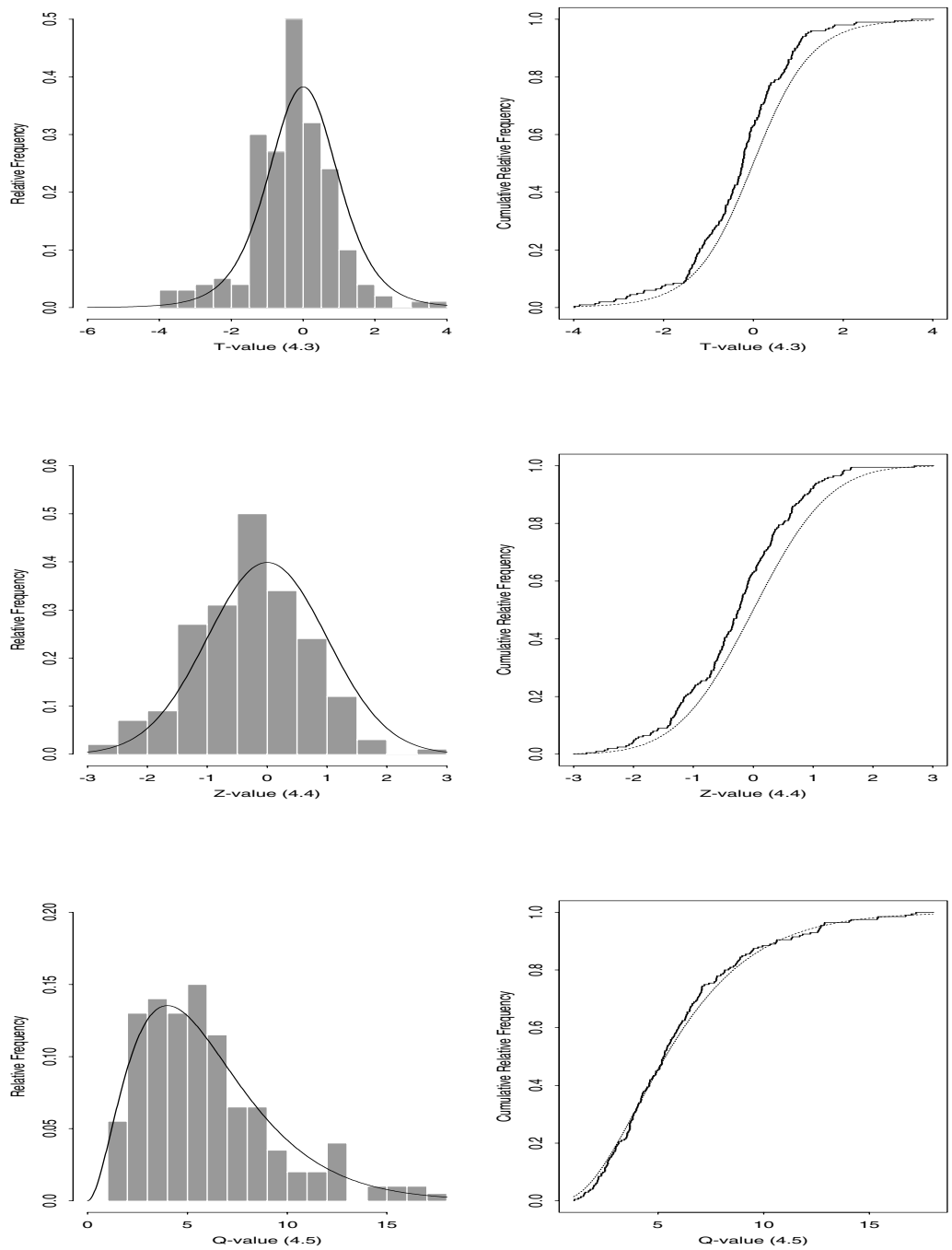

Figure 4.9: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 6 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(6)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** applied to the AR(1) process (4.14) using a 90% CI with no precision requirement.

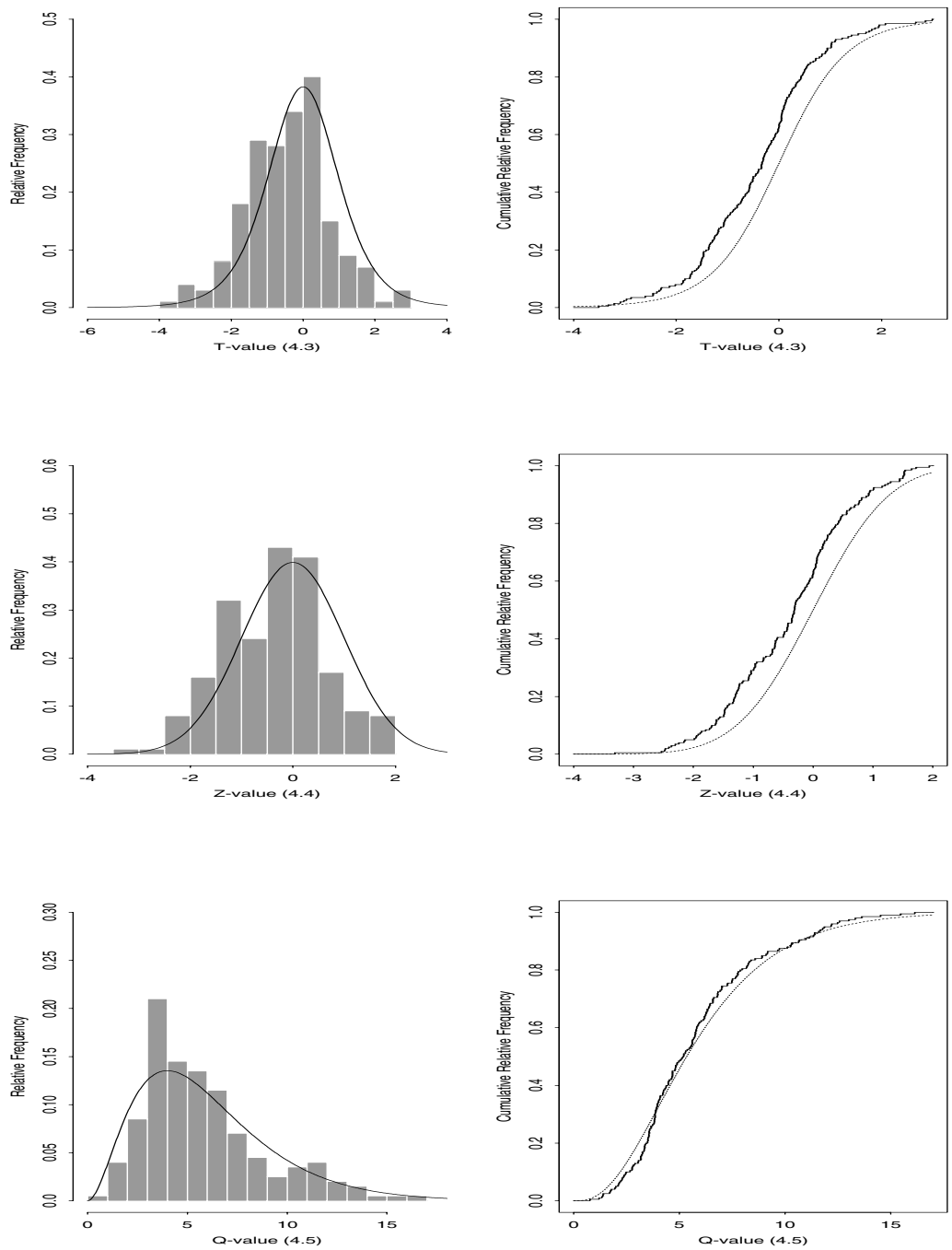

Figure 4.10: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 6 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(6)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** applied to the AR(1) process (4.14) using a 90% CI with $\pm 15\%$ precision requirement.

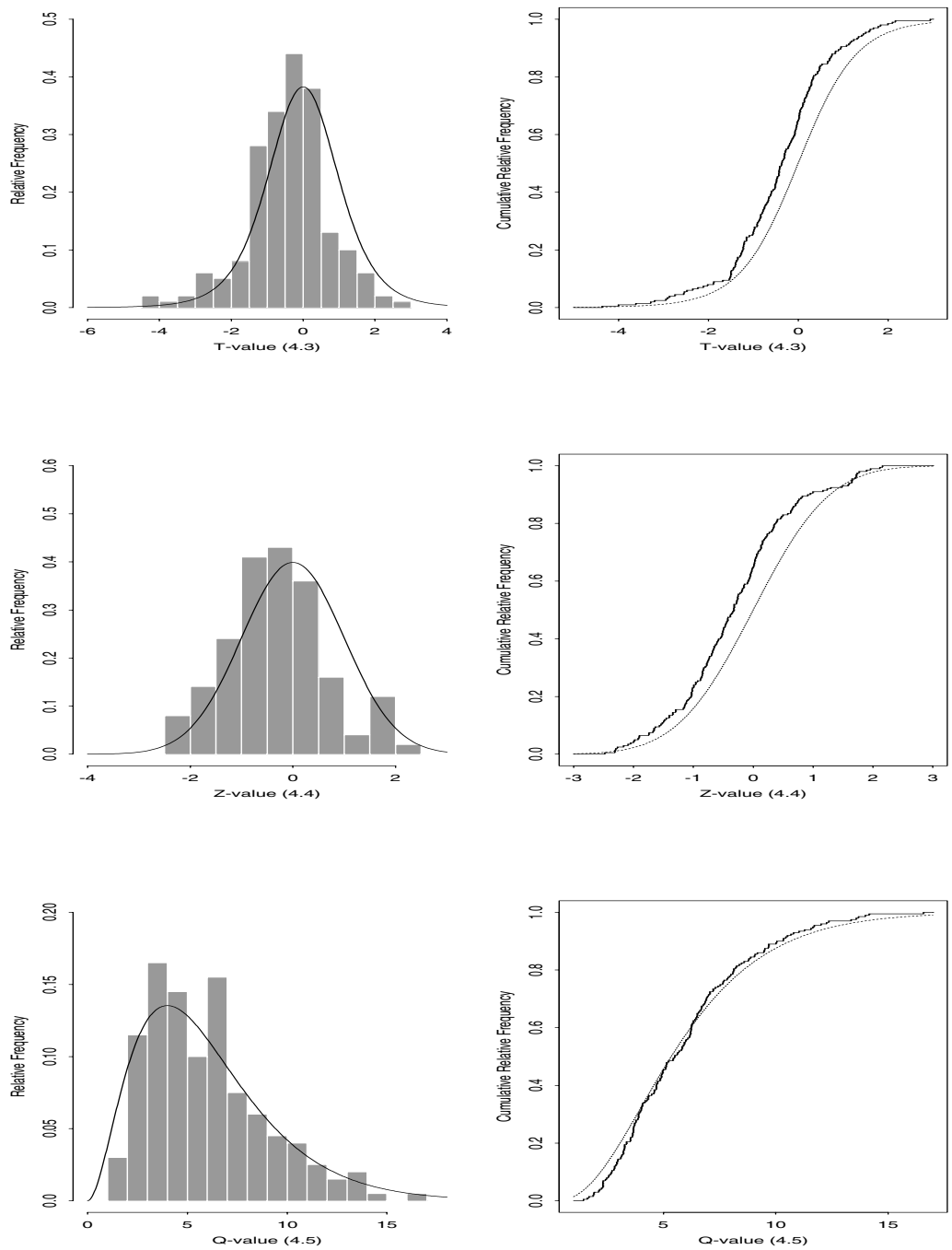

Figure 4.11: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 6 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(6)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** applied to the AR(1) process (4.14) using a 90% CI with $\pm 7.5\%$ precision requirement.

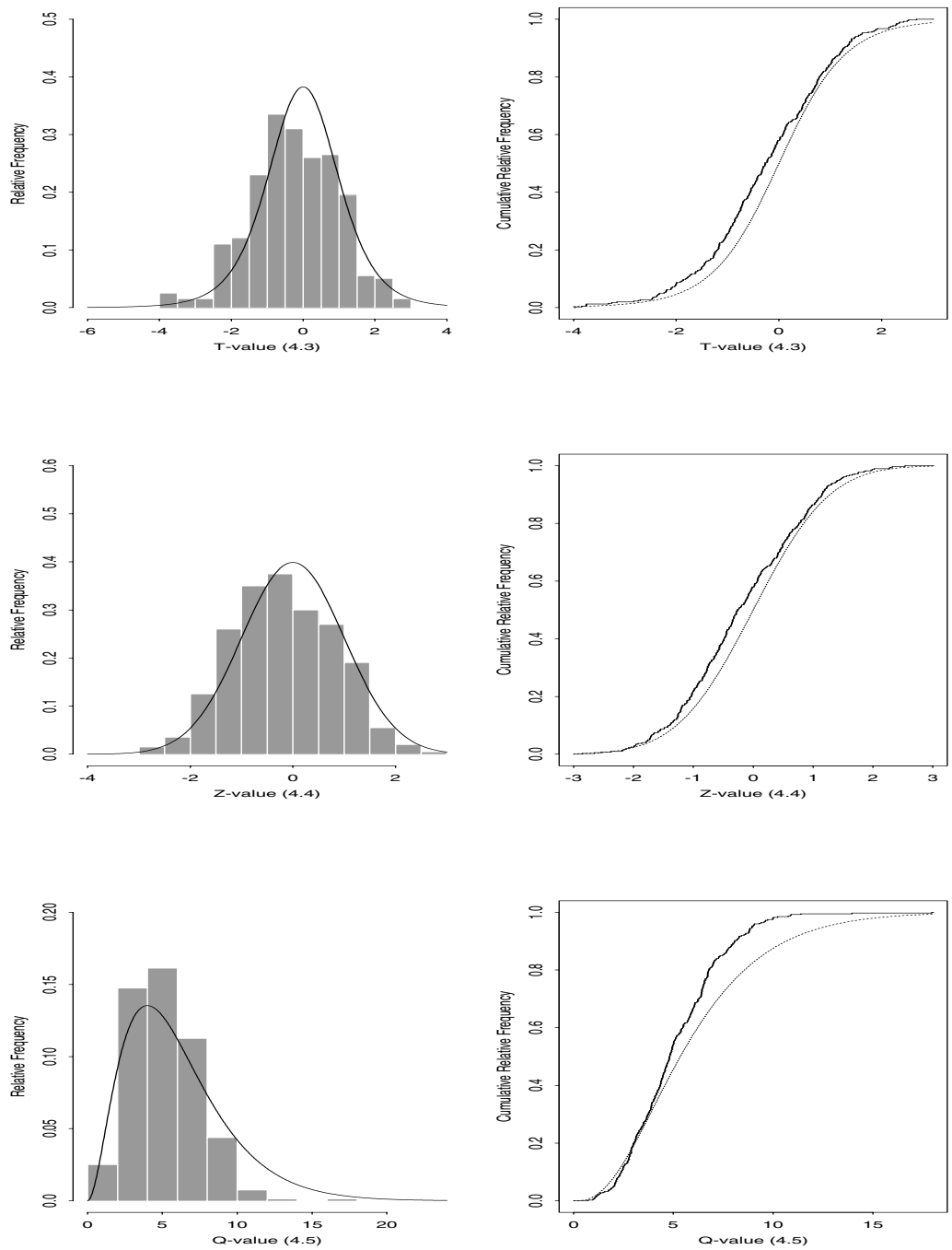

Figure 4.12: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 6 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(6)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** applied to the AR(1) process (4.14) using a 90% CI with $\pm 3.75\%$ precision requirement.

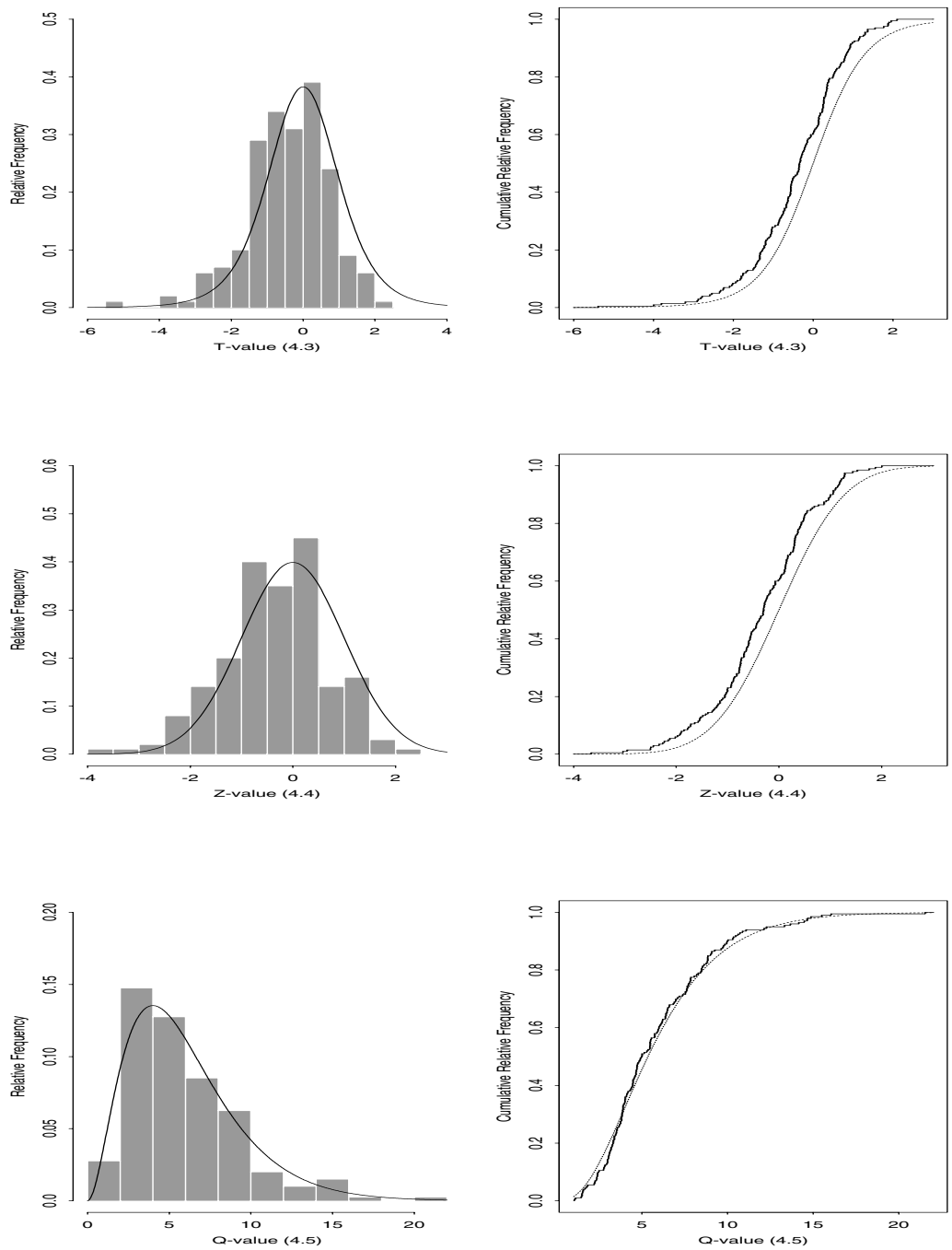

Figure 4.13: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 6 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(6)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** applied to the AR(1) process (4.14) using a 95% CI with no precision requirement.

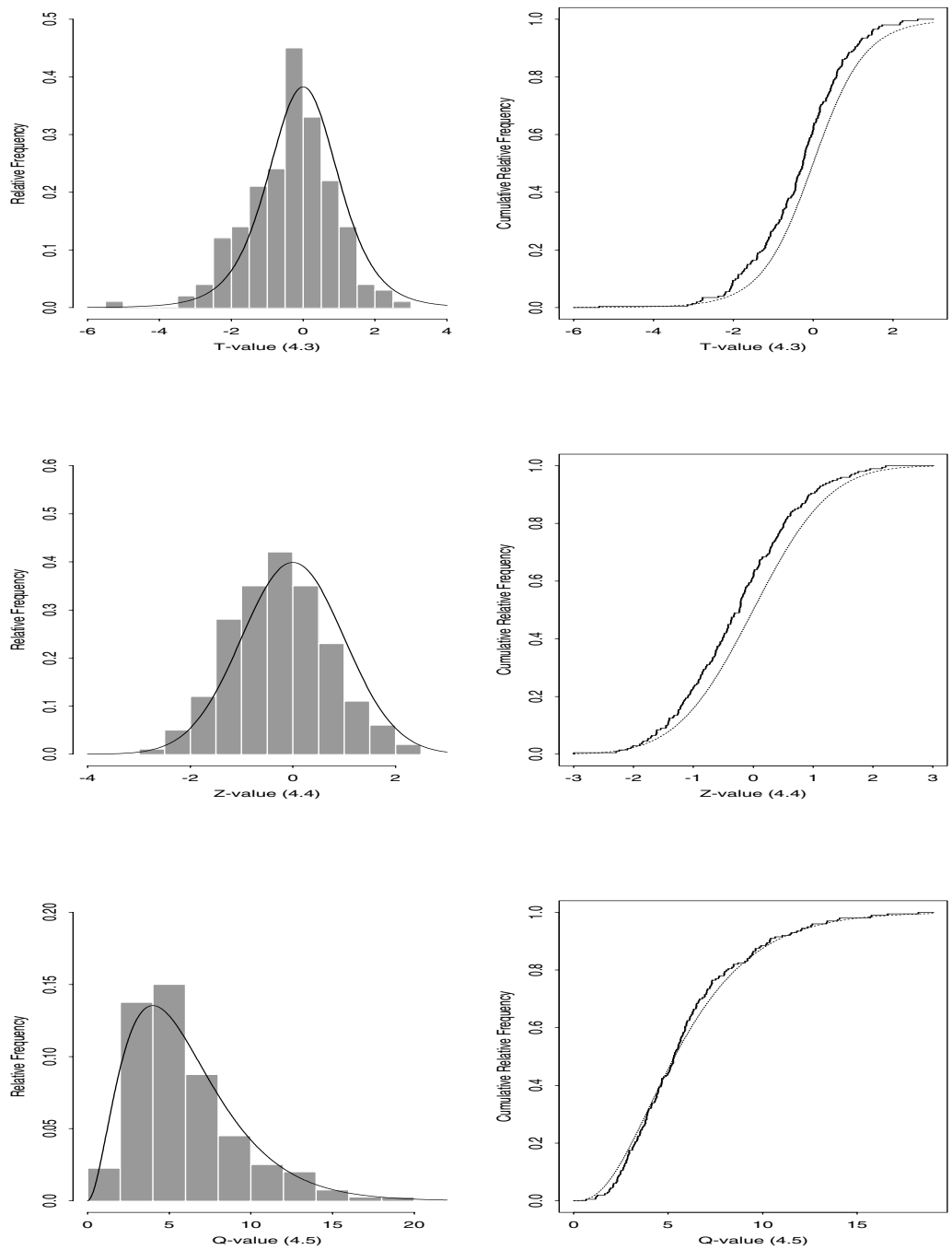

Figure 4.14: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 6 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(6)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** applied to the AR(1) process (4.14) using a 95% CI with $\pm 15\%$ precision requirement.

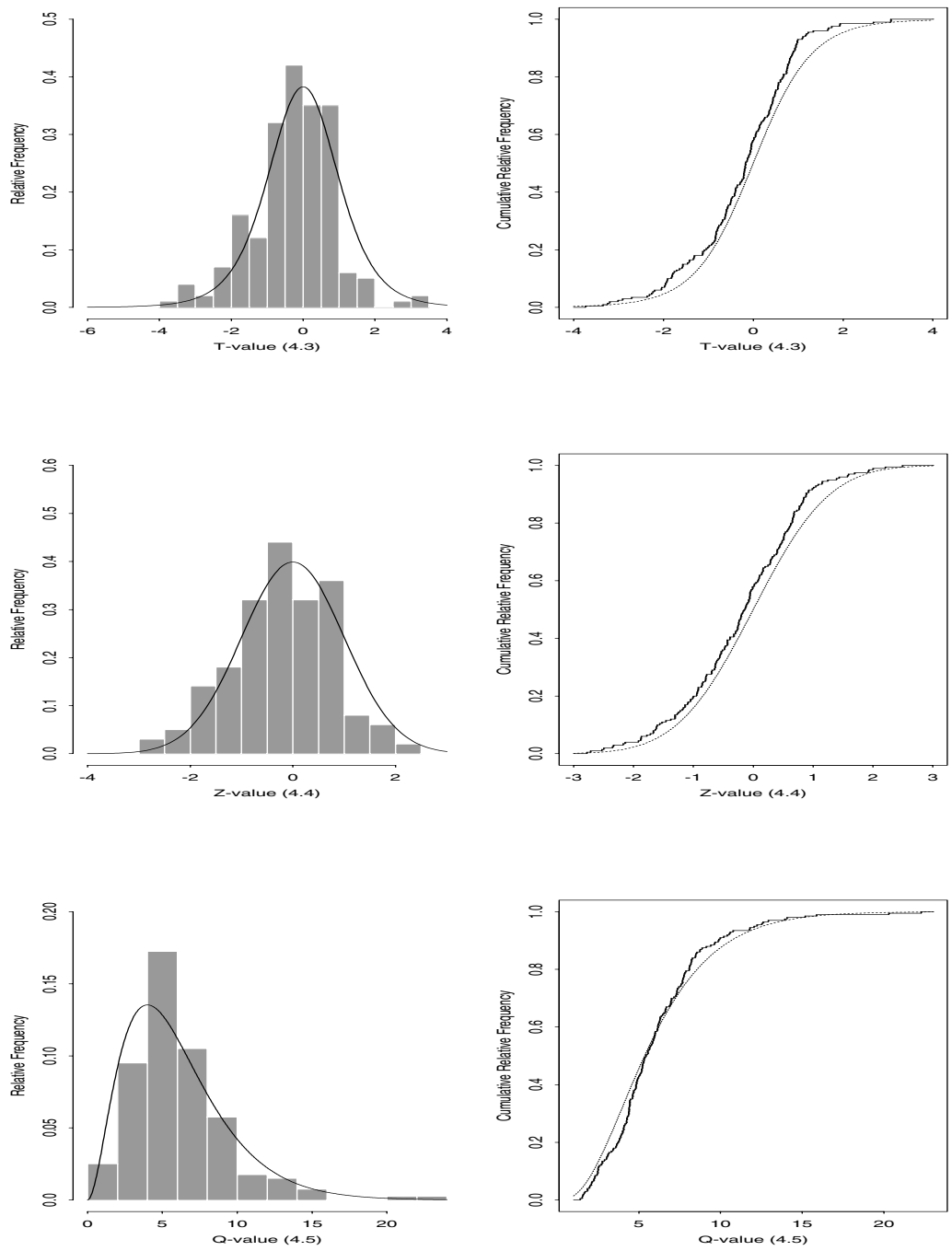

Figure 4.15: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 6 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(6)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** applied to the AR(1) process (4.14) using a 95% CI with $\pm 7.5\%$ precision requirement.

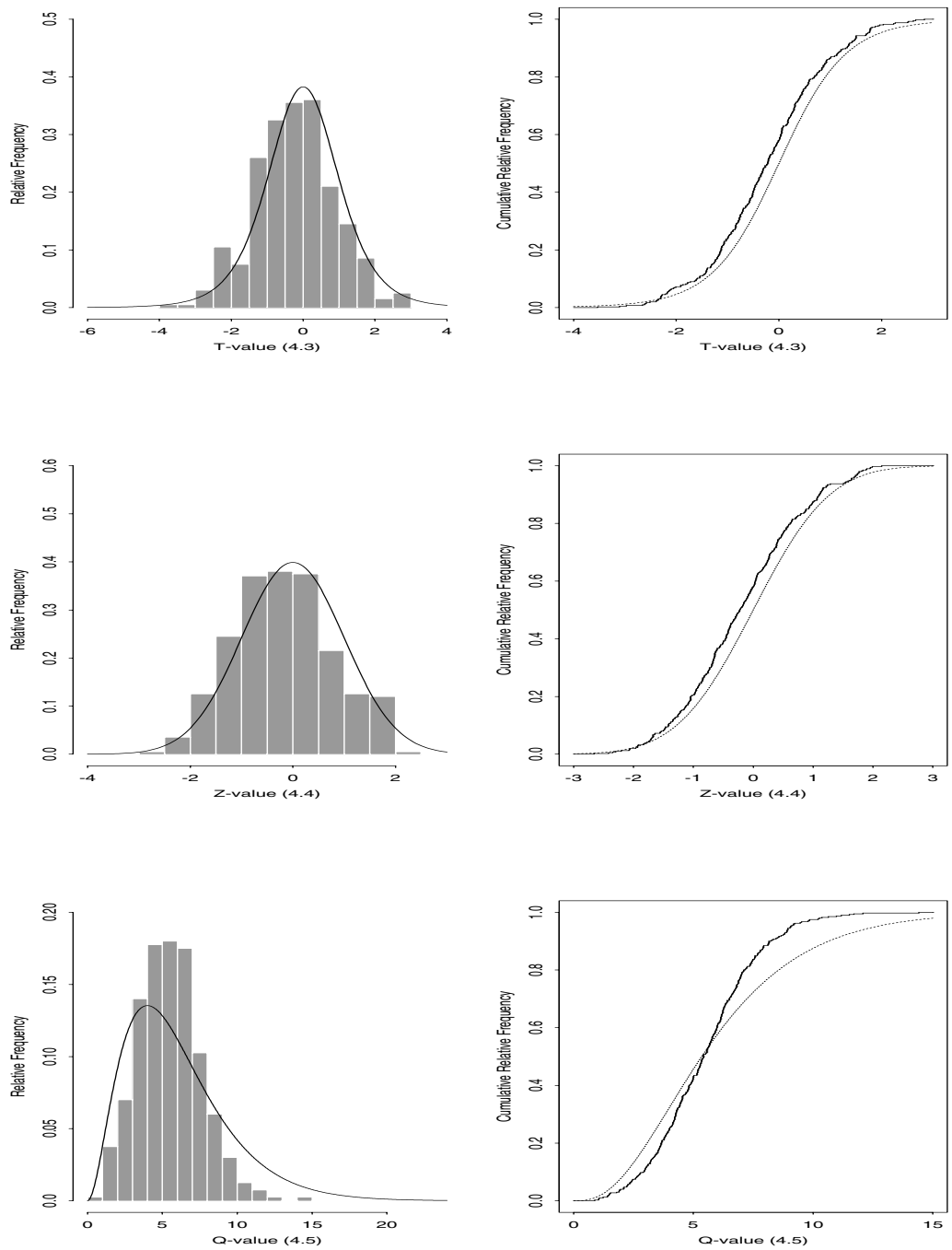

Figure 4.16: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 6 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(6)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** applied to the AR(1) process (4.14) using a 95% CI with $\pm 3.75\%$ precision requirement.

Table 4.11: Correlation between $\overline{\overline{X}}(m,k)$ and $\widehat{\gamma}_X$ for 400 independent replications of **WASSP** applied to the AR(1) process (4.14) using a 90% CI.

Precision Requirement	Corr $\left[\overline{\overline{X}}(m,k), \widehat{\gamma}_X\right]$
None	-0.0041
±15%	0.0209
±7.5%	0.2768
±3.75%	0.1006

Table 4.12: Correlation between $\overline{\overline{X}}(m,k)$ and $\widehat{\gamma}_X$ for 400 independent replications of **WASSP** applied to the AR(1) process (4.14) using a 95% CI.

Precision Requirement	Corr $\left[\overline{\overline{X}}(m,k), \widehat{\gamma}_X\right]$
None	0.1115
±15%	0.0189
±7.5%	0.0844
±3.75%	0.0647

4.3.2 *Comparison of WASSP and ASAP2 for the AR(1) Process*

Tables 4.13 and 4.14 show a comparison of the performance of **WASSP** (using $A = 7$) and ASAP2 for the AR(1) process (4.14) with $\mu_X = 100$, $X_0 = 0$, $\rho = 0.995$, and $\sigma_\delta^2 = 1$ so that there is a pronounced initial transient in each time series of simulation-generated observations from this process. For the no precision case with nominal 90% and 95% CIs, it is clear that ASAP2 completely breaks down. The CIs delivered by ASAP2 for the no precision case are so wide that they have no practical meaning. Furthermore, the catastrophically high variance of the CI half-length indicates that ASAP2 does not produce stable, reliable CIs when no precision requirement is specified.

When a precision requirement is imposed, the tremendous variance of the confidence interval half-length on the first iteration of ASAP2 causes the predicted sample sizes for succeeding iterations to be too high. The sample sizes for the ±15%, ±7.5%, and ±3.75% precision cases are about ten times higher for ASAP2 than for **WASSP**. Furthermore, the actual precision of the confidence intervals delivered by ASAP2 is significantly lower than the requested level. For example in Table 4.13, the average confidence interval half-length for the ±7.5% precision case

is 1.9787. This indicates that at the ±7.5% precision level, ASAP2 is delivering confidence intervals that are within about ±2% of the mean. This can also be attributed to the fact that the variance of the confidence interval half-length is extremely high on the first iteration of the ASAP2 algorithm, and therefore the predicted sample sizes needed to achieve the desired precision are much higher than necessary.

The breakdown of ASAP2 for this process is likely due to the fact that this particular AR(1) process has an extremely high lag-one correlation. When an ARMA time series model is fit to the sequence of batch means for this AR(1) process, the result is a maximum likelihood estimator for the autoregressive parameter that is very close to 1; and such a result leads to unstable estimates of the first four cumulants of the NOBM t-ratio (2.7) which are ultimately used to construct the correlation-adjusted confidence interval (2.8).

4.3.3 *Comparison of WASSP and Heidelberger and Welch's Spectral Method for the AR(1) Process*

Tables 4.15 and 4.16 show the results of applying **WASSP** and Heidelberger and Welch's spectral method to the AR(1) process. The results for Heidelberger and Welch's spectral method were obtained in the same manner as described in Section 4.1. From both Tables 4.15 and 4.16, examination of the statistic $\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$ reveals significant bias in the estimate of the mean for Heidelberger and Welch's method. This bias is likely due for the most part to system warm-up. It is also worth noting the highly erratic results for nominal 90% CIs produced by the Heidelberger and Welch method. For the no precision and the ±7.5% precision cases, H&W-based nominal 90% confidence intervals exhibit significant overcoverage; however, for the ±15% precision case the confidence intervals exhibit catastrophic undercoverage. For nominal 95% CIs delivered by the H&W method in the no precision and ±7.5% precision cases, the resulting coverage probabilities are also significantly higher than the nominal level. Overall, **WASSP** clearly outperforms Heidelberger and Welch's spectral method for the AR(1) process.

Table 4.13: Performance of **WASSP** (using $A = 7$) and ASAP2 for the AR(1) process (4.14) with $\mu_X = 100$, $X_0 = 0$, $\rho = 0.995$, and $\sigma_\delta^2 = 1$. Results are based on independent replications of nominal 90% CIs.

Precision Requirement	Performance Measure	Procedure	
		WASSP	ASAP2
None	# replications	1,000	400
	CI coverage	90%	100%
	Avg. sample size	9,703	10,305
	Max. sample size	22,120	23,040
	Avg. CI half-length	5.3424	90.5508
	Var. CI half-length	1.7498	5,159.18
±15%	# replications	1,000	400
	CI coverage	92%	96.25%
	Avg. sample size	9,739	51,908
	Max. sample size	29,736	201,472
	Avg. CI half-length	5.3423	3.3881
	Var. CI half-length	1.5663	9.8356
±7.5%	# replications	1,000	400
	CI coverage	92%	92.75%
	Avg. sample size	9,899	101,331
	Max. sample size	30,090	403,200
	Avg. CI half-length	5.3037	1.9438
	Var. CI half-length	1.3122	2.6427
±3.75%	# replications	1,000	400
	CI coverage	86%	92.25%
	Avg. sample size	13,470	197,683
	Max. sample size	49,152	806,144
	Avg. CI half-length	3.2235	1.2124
	Var. CI half-length	0.1453	0.8517

Table 4.14: Performance of **WASSP** (using $A = 7$) and ASAP2 for the AR(1) process (4.14) with $\mu_X = 100$, $X_0 = 0$, $\rho = 0.995$, and $\sigma_\delta^2 = 1$. Results are based on independent replications of nominal 95% CIs.

Precision Requirement	Performance Measure	Procedure	
		WASSP	ASAP2
None	# replications	1,000	400
	CI coverage	95%	100%
	Avg. sample size	9,834	10,305
	Max. sample size	29,736	23,040
	Avg. CI half-length	6.7748	136.594
	Var. CI half-length	2.7061	14,915.191
±15%	# replications	1,000	400
	CI coverage	95%	97.5%
	Avg. sample size	9,814	75,677
	Max. sample size	24,120	378,368
	Avg. CI half-length	6.7411	3.3191
	Var. CI half-length	2.6902	11.0498
±7.5%	# replications	1,000	400
	CI coverage	94%	96.5%
	Avg. sample size	9,941	147,110
	Max. sample size	30,208	755,968
	Avg. CI half-length	6.1240	1.9787
	Var. CI half-length	1.0529	3.0829
±3.75%	# replications	1,000	400
	CI coverage	95%	96.5%
	Avg. sample size	20,868	291,709
	Max. sample size	68,608	1,512,448
	Avg. CI half-length	3.2908	1.2139
	Var. CI half-length	0.1393	0.8906

Table 4.15: Performance of **WASSP** (using $A = 7$) and Heidelberger and Welch's spectral method for the AR(1) process (4.14) with $\mu_X = 100$, $X_0 = 0$, $\rho = 0.995$, and $\sigma_\delta^2 = 1$. Results are based on 1,000 independent replications of nominal 90% CIs.

Precision Requirement	Performance Measure	Procedure	
		WASSP	H&W
None	# replications	1,000	1,000
	Overall CI coverage	90.5%	93.1%
	Avg. sample size	9,703	4,096
	Max. sample size	22,120	4,096
	Avg. CI half-length	5.3424	9.9476
	Var. CI half-length	1.7498	2.0874
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	8.6182	32.070
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.3941	0.9890
	# replications satisfying precision requirement	1,000	1,000
	Coverage for CIs satisfying precision requirement	90.5%	93.1%
±15%	# replications	1,000	1,000
	Overall CI coverage	92.2%	72.2%
	Avg. sample size	9,739	4,220
	Max. sample size	29,736	6,371
	Avg. CI half-length	5.3423	12.1689
	Var. CI half-length	1.5663	1.3045
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	8.2502	106.4578
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.4012	3.1019
	# replications satisfying precision requirement	1,000	1,000
	Coverage for CIs satisfying precision requirement	92.2%	72.2%
±7.5%	# replications	1,000	1,000
	Overall CI coverage	92%	94%
	Avg. sample size	9,899	8,061
	Max. sample size	30,090	22,854
	Avg. CI half-length	5.3037	6.5924
	Var. CI half-length	1.3122	1.1403
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	8.4103	11.0716
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.3811	0.4663
	# replications satisfying precision requirement	1,000	845
	Coverage for CIs satisfying precision requirement	92%	93.37%

Table 4.16: Performance of $\mathcal{WASSP}$ (using $A = 7$) and Heidelberger and Welch's spectral method for the AR(1) process (4.14) with $\mu_X = 100$, $X_0 = 0$, $\rho = 0.995$, and $\sigma_\delta^2 = 1$. Results are based on 1,000 independent replications of nominal 95% CIs.

Precision Requirement	Performance Measure	Procedure	
		$\mathcal{WASSP}$	H&W
None	# replications	1,000	1,000
	Overall CI coverage	95.1%	97.1%
	Avg. sample size	9,834	4,096
	Max. sample size	29,736	4,096
	Avg. CI half-length	6.7748	12.3665
	Var. CI half-length	2.7061	3.2718
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	8.7998	33.2424
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.4151	1.0399
	# replications satisfying precision requirement	1,000	1,000
	Coverage for CIs satisfying precision requirement	95.1%	97.1%
±15%	# replications	1,000	1,000
	Overall CI coverage	94.8%	94.8%
	Avg. sample size	9,814	4,732
	Max. sample size	24,120	11,483
	Avg. CI half-length	6.7411	12.4977
	Var. CI half-length	2.6902	1.5553
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	8.4478	42.9346
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.4127	1.6567
	# replications satisfying precision requirement	1,000	1,000
	Coverage for CIs satisfying precision requirement	94.8%	94.8%
±7.5%	# replications	1,000	1,000
	Overall CI coverage	94.2%	97.7%
	Avg. sample size	9,941	9,315
	Max. sample size	30,208	21,504
	Avg. CI half-length	6.1240	7.6182
	Var. CI half-length	1.0529	2.4289
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	8.0383	9.0171
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.3755	0.4124
	# replications satisfying precision requirement	1,000	512
	Coverage for CIs satisfying precision requirement	94.2%	96.48%

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

The next system used to test the performance of **WASSP** is the "AR(1)-to-Parteo," or ARTOP process. Let $\{Z_j : j = 1, 2, \ldots\}$ be a stationary AR(1) process with $N(0,1)$ marginals and lag-one correlation ρ. The process $\{Z_j : j = 1, 2, \ldots\}$ can be generated as follows:

$$Z_j = \rho Z_{j-1} + \delta_j, \tag{4.16}$$

where $Z_0 \sim N(0,1)$ and $\{\delta_j : j = 1, 2, \ldots\} \overset{\text{i.i.d.}}{\sim} N(0, \sigma_\delta^2)$ is a white noise process with variance

$$\sigma_\delta^2 = \sigma_Z^2(1 - \rho^2) = 1 - \rho^2. \tag{4.17}$$

If $\{X_j : j = 1, 2, \ldots\}$ is an ARTOP process with marginal c.d.f.

$$F_X(x) \equiv \Pr\{X \le x\} = \begin{cases} 1 - \left(\frac{\xi}{x}\right)^\vartheta, & x \ge \xi, \\ 0, & x < \xi, \end{cases} \tag{4.18}$$

where $\xi > 0$ is a location parameter and $\vartheta > 0$ is a shape parameter, then $\{X_j\}$ is generated from the "base process" (4.16) as follows. First, the base process $\{Z_j : j = 1, 2, \ldots\}$ is fed into the standard normal c.d.f. to get a sequence of correlated, uniform(0,1) random variables $\{U_j : j = 1, 2, \ldots\}$; that is, $U_j = \Phi(Z_j)$ for $j = 1, 2, \ldots$, where

$$\Phi(z) = \int_{-\infty}^{z} \varphi(w)\, dw \quad \text{and} \quad \varphi(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2} \tag{4.19}$$

respectively denote the $N(0,1)$ c.d.f. and p.d.f. for all real z. Finally, the process $\{U_j : j = 1, 2, \ldots\}$ is fed into the inverse of the Pareto c.d.f. (4.18) to generate the process $\{X_j : j = 1, 2, \ldots\}$ as follows,

$$X_j = F_X^{-1}[U_j] = F_X^{-1}[\Phi(Z_j)] = \frac{\xi}{[1 - \Phi(Z_j)]^{1/\vartheta}}, \quad j = 1, 2, \ldots. \tag{4.20}$$

The mean and the variance of the ARTOP process (4.20) are given by [34]

$$\mu_X = \mathrm{E}[X_j] = \vartheta\xi(\vartheta - 1)^{-1}, \quad \text{for} \quad \vartheta > 1, \tag{4.21}$$

and

$$\sigma_X^2 = \xi^2\vartheta(\vartheta - 1)^{-2}(\vartheta - 2)^{-1}, \quad \text{for} \quad \vartheta > 2, \tag{4.22}$$

respectively.

We decided to set the parameters of the Pareto distribution (4.18) according to $\vartheta = 2.1$ and $\xi = 1$; and we set the lag-one correlation in the base process (4.16) to $\rho = 0.995$. This yields an ARTOJ process $\{X_j : j = 1, 2, \ldots\}$ whose marginal distribution has mean, variance, skewness, and kurtosis respectively given by

$$\mu_X = 1.9091,\ \sigma_X^2 = 17.3554,\ \mathrm{E}\left[\left(\frac{X_j - \mu_X}{\sigma_X}\right)^3\right] = \infty,\ \text{ and }\ \mathrm{E}\left[\left(\frac{X_j - \mu_X}{\sigma_X}\right)^4\right] = \infty. \quad (4.23)$$

The most difficult aspect of this system is that the marginals are highly nonnormal, and their distribution has a very heavy tail. We sampled Z_0 from the $N(0, 1)$ distribution when generating the process $\{X_j\}$ so that the process was started in steady-state operation. Therefore, there is no warm-up problem for this process.

Tables 4.17 and 4.18 show the performance of **WASSP** for the ARTOP process (4.20) described above using the smoothing parameter values $A = 5, 7$, and 9. The results are based on 400 independent replications of nominal 90% and 95% CIs. From these two tables, we see that the coverage decreases in general as the smoothing parameter increases. For nominal 90% CIs with $A = 7$ and $A = 9$, the resulting coverage probabilities are unacceptable at all three precision levels. For nominal 95% CIs, the coverage probabilities for $A = 7$ and $A = 9$ are unacceptable for the no precision and the $\pm 15\%$ precision levels. In this application of **WASSP**, the smoothing parameter value $A = 5$ appears to yield the best results for both nominal 90% and 95% confidence intervals, especially in the $\pm 7.5\%$ precision case. While it is the case that for $A = 5$ there is significant undercoverage in the small-sample cases, clearly as the sample size increases the coverage probabilities approach the nominal level. It is unclear at this point why $A = 5$ produces the best results for this process. Nonetheless, it is recommended that the default smoothing parameter be changed from $A = 7$ to $A = 5$ for this ARTOP process.

Table 4.17: Performance of **WASSP** using different values of A for the ARTOP process (4.20) with $\xi = 1.0$, $\vartheta = 2.1$, $\rho = 0.995$ and $Z_0 \sim N(0,1)$. Results are based on 400 independent replications of nominal 90% CIs.

Precision Requirement	Performance Measure	Smoothing Parameter		
		$A = 5$	$A = 7$	$A = 9$
None	CI coverage	80.5%	76.5%	74.0%
	Avg. sample size	15,276	15,350	17,021
	Max. sample size	122,176	321,470	901,340
	Avg. CI half-length	0.5740	0.4524	0.4447
	Var. CI half-length	0.2943	0.0991	0.5885
±15%	CI coverage	78.25%	72.0%	69.25%
	Avg. sample size	81,636	54,077	44,250
	Max. sample size	2,554,011	735,232	1,909,760
	Avg. CI half-length	0.2201	0.2214	0.2247
	Var. CI half-length	0.0022	0.0022	0.0018
±7.5%	CI coverage	89%	81%	83%
	Avg. sample size	377,140	303,840	281,890
	Max. sample size	5,066,880	3,829,760	3,946,496
	Avg. CI half-length	0.1187	0.1164	0.1183
	Var. CI half-length	0.0004	0.0004	0.0004

Table 4.18: Performance of **WASSP** using different values of A for the ARTOP process (4.20) with $\xi = 1.0$, $\vartheta = 2.1$, $\rho = 0.995$, and $Z_0 \sim N(0,1)$. Results are based on 400 independent replications of nominal 95% CIs.

Precision Requirement	Performance Measure	Smoothing Parameter		
		$A = 5$	$A = 7$	$A = 9$
None	CI coverage	88.5%	81.25%	78%
	Avg. sample size	15,584	15,350	15,214
	Max. sample size	146,421	321,470	321,470
	Avg. CI half-length	0.6863	0.5697	0.5207
	Var. CI half-length	0.1837	0.1572	0.1309
±15%	CI coverage	85.75%	86%	76.5%
	Avg. sample size	131,410	111,010	70,355
	Max. sample size	1,439,854	2,871,296	459,776
	Avg. CI half-length	0.2241	0.2255	0.2254
	Var. CI half-length	0.002	0.002	0.0016
±7.5%	CI coverage	95%	91%	95%
	Avg. sample size	782,638	532,315	474,795
	Max. sample size	6,156,442	6,033,408	6,877,184
	Avg. CI half-length	0.1198	0.1197	0.1186
	Var. CI half-length	0.0005	0.0004	0.0005

4.4.1 *Validation of Student's t-Ratio Assumptions for the ARTOP Process*

To validate the assumptions required for the random variable $\mathcal{T}$ in (4.4) to have Student's t-distribution with $2a$ degrees of freedom when we set $A = 5$ and apply $\mathcal{WASSP}$ to the ARTOP process (4.20), we generated the same types of histogram and c.d.f. plots as for the $M/M/1$ process and the AR(1) process. To generate these plots, we first used the method detailed in Appendix C to compute the theoretical SSVC for the ARTOP process. From Equation (C.9), we have the following estimate of γ_X with maximum relative error $\varepsilon_{\text{rel}} = 10^{-6}$,

$$\gamma_X \approx 1612.7791494. \tag{4.24}$$

Figures 4.17–4.22 show the histogram and c.d.f. plots for the ARTOP process (4.20) using $A = 5$. In the middle row of plots in Figures 4.17 and 4.20, the distribution of $\mathcal{Z}$-values is skewed, implying the mean $\overline{\overline{X}}(m, k)$ is biased. Since this ARTOP process is started in steady-state operation, there is no bias in the mean due to system warm-up. Therefore, the bias must be caused entirely by the randomness of the simulation run length n. We can see in the middle row of plots in Figures 4.19 and 4.22 that as the sample size n increases, the bias due to the randomness of the simulation run length decreases. From the remainder of the plots, it is clear that asymptotically Assumptions A_1 and A_2 are satisfied at least to a reasonable approximation.

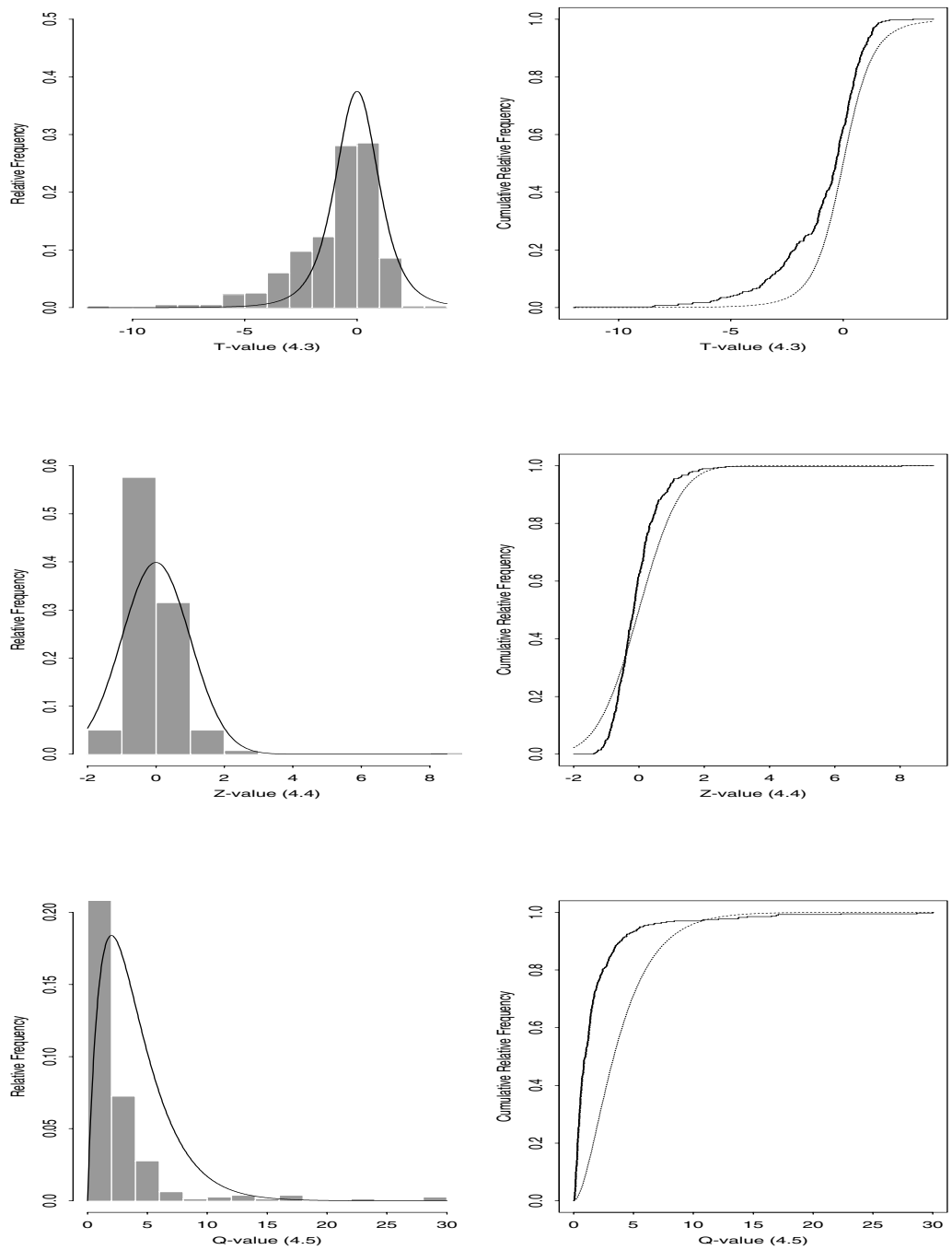

Figure 4.17: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 4 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(4)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** applied to the ARTOP process (4.20) using a 90% CI with no precision requirement.

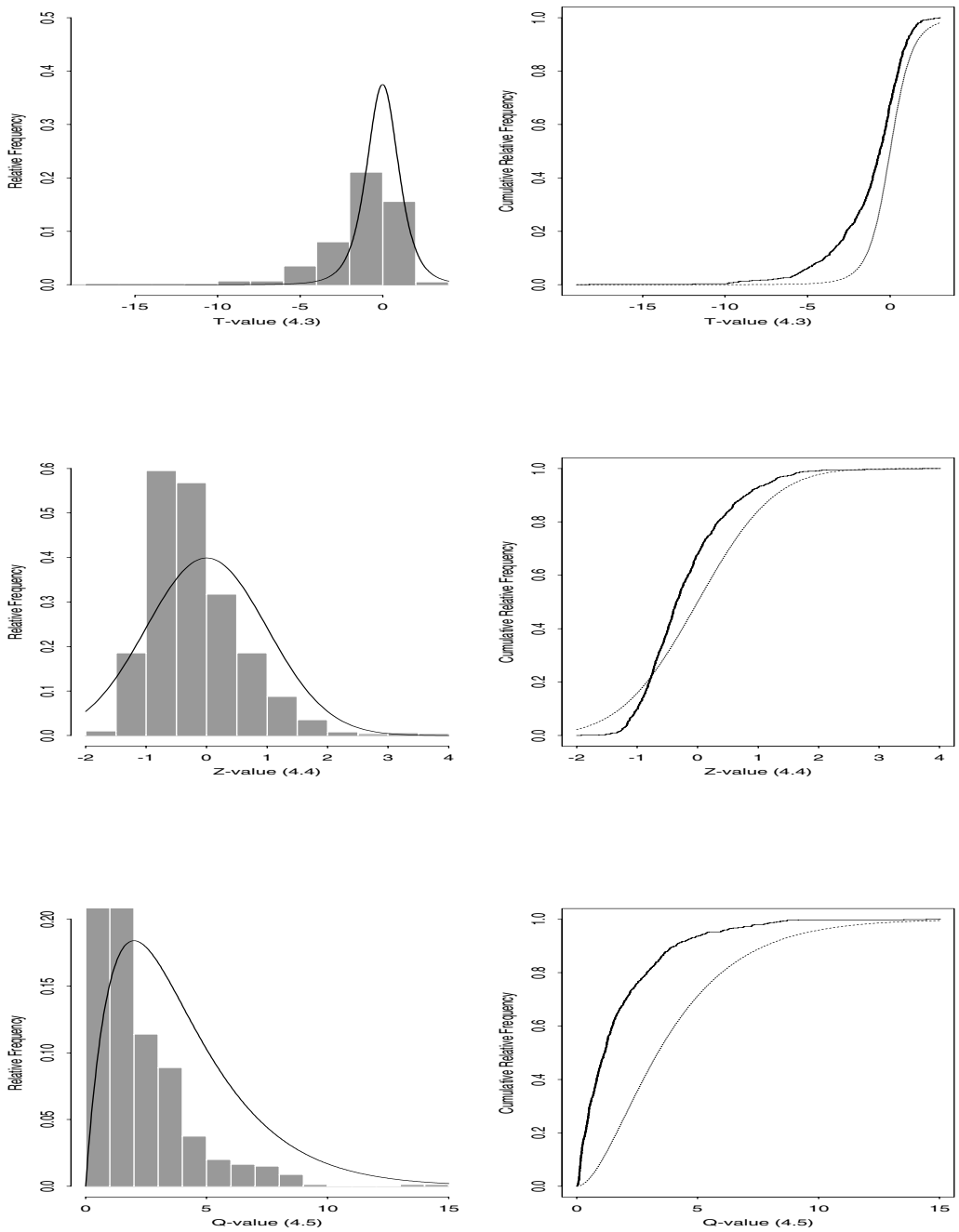

Figure 4.18: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth cuves)—namely, the t-distribution with 4 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(4)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** applied to the ARTOP process (4.20) using a 90% CI with $\pm 15\%$ precision requirement.

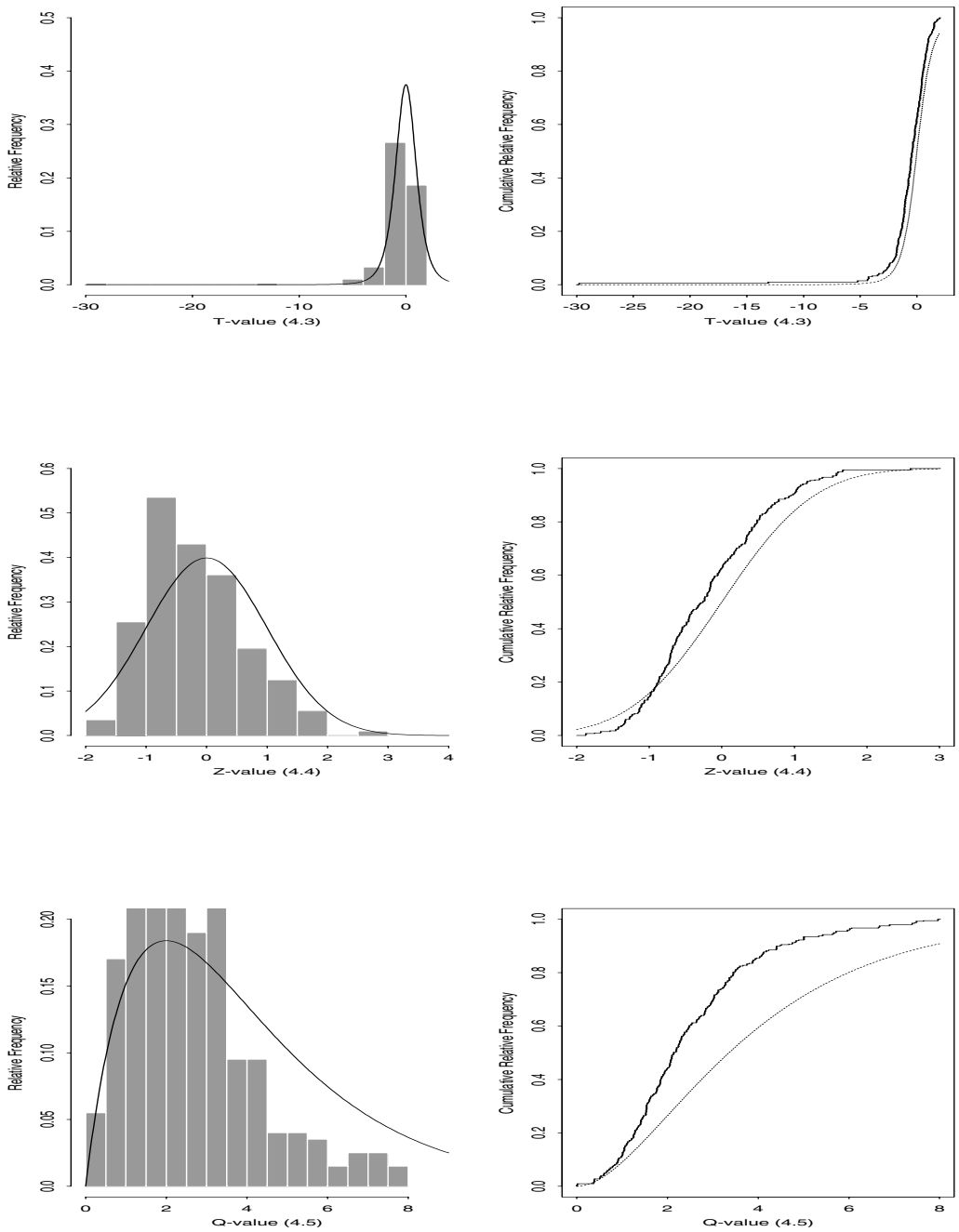

Figure 4.19: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 4 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(4)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** applied to the ARTOP process (4.20) using a 90% CI with $\pm 7.5\%$ precision requirement.

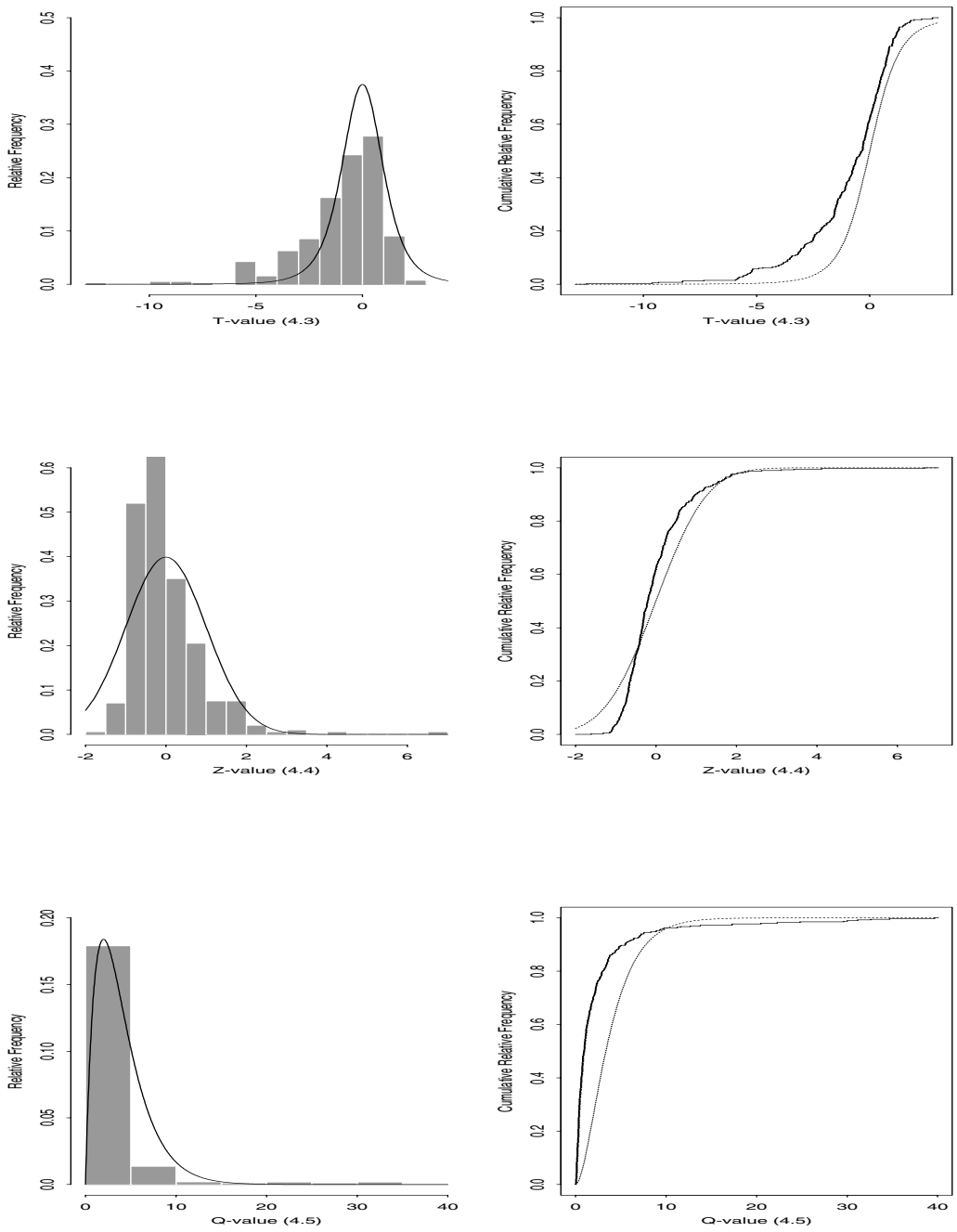

Figure 4.20: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 4 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(4)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** applied to the ARTOP process (4.20) using a 95% CI with no precision requirement.

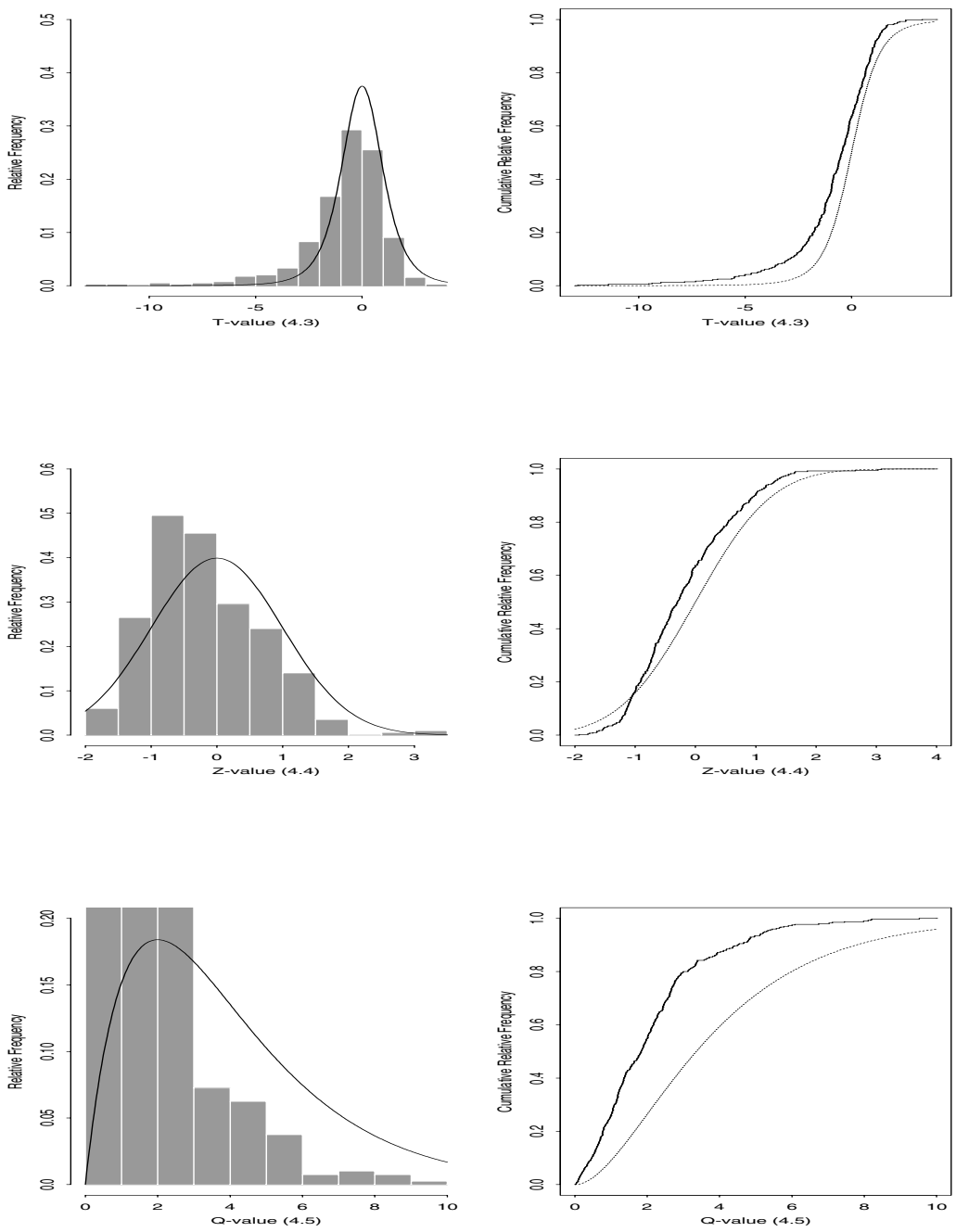

Figure 4.21: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 4 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(4)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** applied to the ARTOP process (4.20) using a 95% CI with $\pm 15\%$ precision requirement.

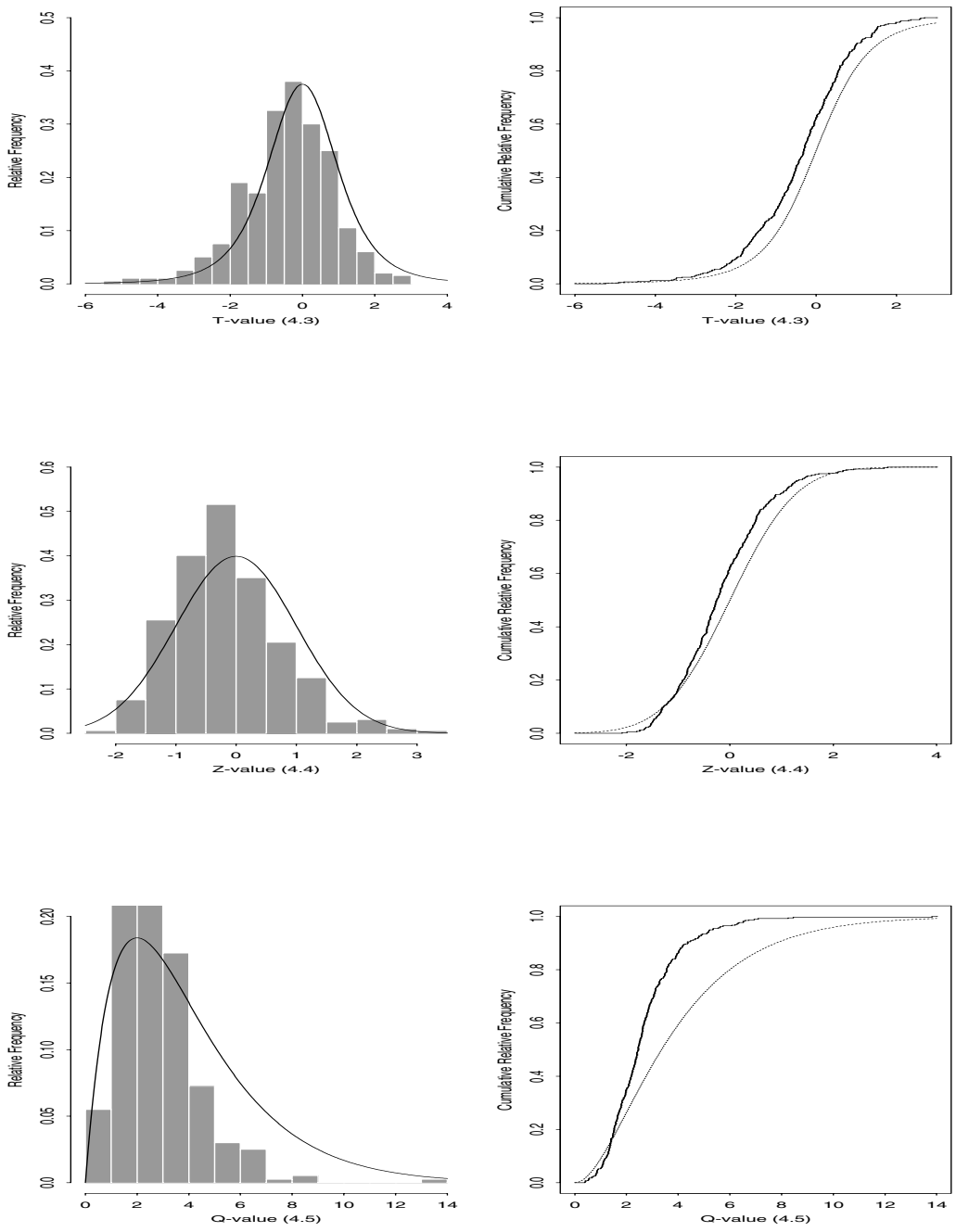

Figure 4.22: Comparison of the empirical distributions (step functions) of 400 $\mathcal{T}$-values (top row), $\mathcal{Z}$-values (middle row), and $\mathcal{Q}$-values (bottom row) with their corresponding assumed theoretical distributions (smooth curves)—namely, the t-distribution with 4 degrees of freedom, the $N(0,1)$ distribution, and the $\chi^2(4)$ distribution, respectively. Results obtained from 400 i.i.d. runs of **WASSP** applied to the ARTOP process (4.20) using a 95% CI with $\pm 7.5\%$ precision requirement.

To validate Assumption A_3, we computed the correlation between $\overline{\overline{X}}(m,k)$ and $\widehat{\gamma}_X$ for 400 independent replications of **WASSP** applied to the ARTOP process (4.20). From Tables 4.19 and 4.20, we see that the correlation between $\overline{\overline{X}}(m,k)$ and $\widehat{\gamma}_X$ is significant at all three levels of precision. This implies that $\mathcal{Z}$ and $\mathcal{Q}$ are not independent, and this may partially explain the undercoverage seen in the no precision and $\pm 15\%$ precision cases in Tables 4.17 and 4.18.

Table 4.19: Correlation between $\overline{\overline{X}}(m,k)$ and $\widehat{\gamma}_X$ for 400 independent replications of **WASSP** applied to the ARTOP process (4.20) using $A = 5$ and a 90% CI.

Precision Requirement	Corr $\left[\overline{\overline{X}}(m,k), \widehat{\gamma}_X\right]$
None	0.5906
$\pm 15\%$	0.5587
$\pm 7.5\%$	0.5462

Table 4.20: Correlation between $\overline{\overline{X}}(m,k)$ and $\widehat{\gamma}_X$ for 400 independent replications of **WASSP** applied to the ARTOP process (4.20) using $A = 5$ and a 95% CI.

Precision Requirement	Corr $\left[\overline{\overline{X}}(m,k), \widehat{\gamma}_X\right]$
None	0.6203
$\pm 15\%$	0.5882
$\pm 7.5\%$	0.5034

4.4.2 *Comparison of WASSP and ASAP2 for the ARTOP Process*

Tables 4.21 and 4.22 show a comparison of the performance of **WASSP** (using $A = 5$) and ASAP2 for the ARTOP process (4.20). For nominal 90% and 95% CIs with no precision requirement, ASAP2 yields a higher coverage probability than **WASSP** does. However, ASAP2 is also requiring significantly larger sample sizes than **WASSP** is for the no precision case. For nominal 90% CIs in the $\pm 15\%$ precision case, ASAP2 clearly outperforms **WASSP**; and the two methods produce comparable results for nominal 95% CIs with $\pm 15\%$ precision. Asymptotically, **WASSP** appears to outperform ASAP2 in the ARTOP process, however. For nominal 90% and 95% confidence intervals with $\pm 7.5\%$ precision, the coverage probability for **WASSP** is right at the nominal level, while the coverage probability for ASAP2 is significantly below the nominal

level. In fact, the coverage probabilites for ASAP2 are about the same for both nominal 90% and nominal 95% confidence intervals at all three levels of precision.

Table 4.21: Performance of **WASSP** (using $A = 5$) and ASAP2 for the ARTOP process (4.20) with $\xi = 1.0$, $\vartheta = 2.1$, $\rho = 0.995$, and $Z_0 \sim N(0,1)$. Results are based on 400 independent replications of nominal 90% CIs.

Precision Requirement	Performance Measure	Procedure	
		WASSP	ASAP2
None	# replications	400	400
	CI coverage	80.5%	85.75%
	Avg. sample size	15,276	113,336
	Max. sample size	122,176	524,288
	Avg. CI half-length	0.5740	0.1788
	Var. CI half-length	0.2943	0.0123
±15%	# replications	400	400
	CI coverage	78.25%	85.5%
	Avg. sample size	81,636	117,883
	Max. sample size	2,554,011	524,288
	Avg. CI half-length	0.2201	0.1645
	Var. CI half-length	0.0022	0.0025
±7.5%	# replications	400	400
	CI coverage	89%	84.75%
	Avg. sample size	377,140	183,534
	Max. sample size	5,066,880	1,650,924
	Avg. CI half-length	0.1187	0.1276
	Var. CI half-length	0.0004	0.0002

4.4.3 *Comparison of WASSP and Heidelberger and Welch's Spectral Method for the ARTOP Process*

Tables 4.23 and 4.24 show the results of comparing the performance of **WASSP** (using $A = 5$) with Heidelberger and Welch's sequential spectral method. We first notice from these two tables that the values $\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$ for **WASSP** and Heidelberger and Welch's method are comparable. Since this ARTOP process is started in steady-state, $\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$ for Heidelberger and Welch's method will not be inflated by system warm-up bias, as we have seen in some of the other test processes. We also notice from these two tables that **WASSP**-generated CIs satisfying the precision requirement have much better coverage probabilities than the H&W-generated CIs. Even though **WASSP** produces significant undercoverage in the small-sample case (especially for nominal 90% CIs), asymptotically it produces coverages that are at the nominal

Table 4.22: Performance of **WASSP** (using $A = 5$) and ASAP2 for the ARTOP process (4.20) with $\xi = 1.0$, $\vartheta = 2.1$, $\rho = 0.995$, and $Z_0 \sim N(0, 1)$. Results are based on 400 independent replications of nominal 95% CIs.

Precision Requirement	Performance Measure	Procedure	
		WASSP	ASAP2
None	# replications	400	400
	CI coverage	88.5%	90.25%
	Avg. sample size	15,584	113,336
	Max. sample size	146,421	524,288
	Avg. CI half-length	0.6863	0.2138
	Var. CI half-length	0.1837	0.0177
±15%	# replications	400	400
	CI coverage	85.75%	90.5%
	Avg. sample size	131,410	121,015
	Max. sample size	1,439,854	524,288
	Avg. CI half-length	0.2241	0.1906
	Var. CI half-length	0.0020	0.0025
±7.5%	# replications	400	400
	CI coverage	95%	90.25%
	Avg. sample size	782,638	252,741
	Max. sample size	6,156,442	1,953,339
	Avg. CI half-length	0.1198	0.1315
	Var. CI half-length	0.0005	0.0001

level. Even at the ±7.5% precision level, the coverage for Heidelberger and Welch's method is significantly below the nominal level.

Table 4.23: Performance of $\mathcal{WASSP}$ (using $A = 5$) and Heidelberger and Welch's spectral method for the ARTOP process (4.20) with $\xi = 1.0$, $\vartheta = 2.1$, $\rho = 0.995$, and $Z_0 \sim N(0,1)$. Results are based on 400 independent replications of nominal 90% CIs.

Precision Requirement	Performance Measure	Procedure	
		$\mathcal{WASSP}$	H&W
None	# replications	400	400
	Overall CI coverage	80.5%	80%
	Avg. sample size	15,276	4,096
	Max. sample size	122,176	4,096
	Avg. CI half-length	0.5740	0.7553
	Var. CI half-length	0.2943	1.0580
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	0.1311	0.3028
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.0131	0.0554
	# replications satisfying precision requirement	400	400
	Coverage for CIs satisfying precision requirement	80.5%	80%
±15%	# replications	400	400
	Overall CI coverage	78.25%	73%
	Avg. sample size	81,636	43,981
	Max. sample size	2,554,011	637,500
	Avg. CI half-length	0.2201	0.2522
	Var. CI half-length	0.0022	0.0108
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	0.0397	0.0537
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.0021	0.0025
	# replications satisfying precision requirement	400	354
	Coverage for CIs satisfying precision requirement	78.25%	70.06%
±7.5%	# replications	400	400
	Overall CI coverage	89%	78%
	Avg. sample size	377,140	216,715
	Max. sample size	5,066,880	1,309,545
	Avg. CI half-length	0.1187	0.1348
	Var. CI half-length	0.0004	0.0031
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	0.0086	0.0135
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.0017	0.0018
	# replications satisfying precision requirement	400	321
	Coverage for CIs satisfying precision requirement	89%	74.43%

Table 4.24: Performance of $\mathcal{WASSP}$ (using $A = 5$) and Heidelberger and Welch's spectral method for the ARTOP process (4.20) with $\xi = 1.0$, $\vartheta = 2.1$, $\rho = 0.995$, and $Z_0 \sim N(0,1)$. Results are based on 400 independent replications of nominal 95% CIs.

Precision Requirement	Performance Measure	Procedure	
		$\mathcal{WASSP}$	H&W
None	# replications	400	400
	Overall CI coverage	88.5%	82.5%
	Avg. sample size	15,584	4,096
	Max. sample size	146,421	4,096
	Avg. CI half-length	0.6863	0.9283
	Var. CI half-length	0.1837	2.2767
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	0.0927	0.3003
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.0053	0.0627
	# replications satisfying precision requirement	400	400
	Coverage for CIs satisfying precision requirement	88.5%	82.5%
±15%	# replications	400	400
	Overall CI coverage	85.75%	82.5%
	Avg. sample size	131,410	71,059
	Max. sample size	1,439,854	671,948
	Avg. CI half-length	0.2241	0.2586
	Var. CI half-length	0.0020	0.0130
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	0.0261	0.0335
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.0018	0.0019
	# replications satisfying precision requirement	400	349
	Coverage for CIs satisfying precision requirement	85.75%	80.52%
±7.5%	# replications	400	400
	Overall CI coverage	95.3%	92.7%
	Avg. sample size	782,638	432,664
	Max. sample size	6,156,442	2,517,168
	Avg. CI half-length	0.1198	0.1355
	Var. CI half-length	0.0005	0.0018
	$\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]$	0.0034	0.0065
	$\widehat{\text{SE}}\{\widehat{\text{MSE}}[\overline{\overline{X}}(m,k)]\}$	0.0009	0.0016
	# replications satisfying precision requirement	400	334
	Coverage for CIs satisfying precision requirement	95.3%	90.6%

4.5. The AR(1)-to-Johnson (ARTOJ) Process

Let $\{X_j : j = 1, 2, \ldots\}$ be an AR(1)-to-Johnson (ARTOJ) process whose marginals have a Johnson S_U distribution with shape parameters ϑ_1 and ϑ_2, scale parameter λ, and location paramter ξ. Starting from a "base process" $\{Z_j : j = 1, 2, \ldots\}$ as specified in (4.16)–(4.17), we generate the process $\{X_j : j = 1, 2, \ldots\}$ as follows,

$$X_j = \xi + \lambda \sinh\left(\frac{Z_j - \vartheta_1}{\vartheta_2}\right) \quad \text{for} \quad j = 1, 2, \ldots, \tag{4.25}$$

where the hyperbolic sine function is given by

$$\sinh(z) = \frac{1}{2}\left(e^z - e^{-z}\right) \quad \text{for all real } z.$$

To evaluate the performance of **WASSP**, we set the parameters of the ARTOJ process as in Table 4.25. With these parameters, the marginal distribution of the process $\{X_j\}$ is a Johnson S_U distribution with mean, variance, skewness, and kurtosis respectively given by

$$\mu_X = 1, \quad \sigma_X^2 = 1, \quad \mathrm{E}\left[\left(\frac{X_j - \mu_X}{\sigma_X}\right)^2\right] = 100, \quad \text{and} \quad \mathrm{E}\left[\left(\frac{X_j - \mu_X}{\sigma_X}\right)^4\right] = 900. \tag{4.26}$$

Since the distribution of the marginals of the process $\{X_j\}$ has such large values of skewness and kurtosis, it is highly nonnormal and it will exhibit a heavy tail to the right. We will start the ARTOJ process in steady-state operation by sampling Z_0 from the $N(0, 1)$ distribution.

Table 4.25: Parameter values for the ARTOJ process (4.25).

Parameter	Value
ϑ_1	-0.3082812288
ϑ_2	0.7563372077
λ	0.2193791025
ξ	0.7797155628
ρ	0.995

Tables 4.26 and 4.27 show the performance of **WASSP** for the ARTOJ process using the smoothing parameter values $A = 7$ and $A = 9$. The results are based on 400 independent replications of nominal 90% and 95% CIs. From these two tables, we see that the results for the no precision and the $\pm 15\%$ precision levels with $A = 7$ and $A = 9$ are both excellent. However, at the $\pm 7.5\%$ precision level, the results for $A = 7$ indicate that both nominal 90% and 95% confidence intervals have significant overcoverage.

Table 4.26: Performance of **WASSP** and ASAP2 for the ARTOJ process (4.25). Results are based on 400 independent replications of nominal 90% CIs.

Precision Requirement	Performance Measure	Procedure		
		WASSP($A = 7$)	**WASSP**($A = 9$)	ASAP2
None	CI coverage	85.25%	88.25%	91.75%
	Avg. sample size	9,874	9,904	40,160.6
	Max. sample size	122,176	75,712	185,344
	Avg. CI half-length	0.3102	0.2968	0.1194
	Var. CI half-length	0.0194	0.0139	0.0027
±15%	CI coverage	87.25%	85.5%	92%
	Avg. sample size	56,912	48,383	43,359.8
	Max. sample size	462,848	679,936	185,344
	Avg. CI half-length	0.1263	0.1283	0.1070
	Var. CI half-length	0.0005	0.0005	0.0007
±7.5%	CI coverage	93.5%	91.0%	92.5%
	Avg. sample size	258,020	239,230	90,533.2
	Max. sample size	1,880,064	1,757,184	197,775
	Avg. CI half-length	0.0652	0.0650	0.0704
	Var. CI half-length	0.00009	0.00009	0.00003

Along with the performance of **WASSP** for the smoothing parameter values $A = 7$ and $A = 9$, Tables 4.26 and 4.27 also show the performance of ASAP2 for the ARTOJ process. For nominal 90% and 95% CIs, there is little difference between the coverage probabilities at all three precision levels for ASAP2. Another interesting point about the results is that in the ±7.5% precision case with nominal 90% CIs, the coverage for **WASSP** using $A = 7$ and $A = 9$ is about the same as the coverage for ASAP2. However, the average sample sizes required by **WASSP** are significantly larger than those required by ASAP2.

4.6. The $M/M/1$/LIFO Queue Waiting Time Process

Finally, we applied **WASSP** to the waiting time process in the $M/M/1$/LIFO queue with 80% server utilization and an empty-and-idle initial condition. For this process, X_i is the waiting time for the ith customer, $i = 1, 2, \ldots$, in a single-server queueing system with i.i.d. exponential interarrival times having mean 1, i.i.d. exponential service times having mean 0.8, and a last-in-first-out queueing discipline. The theoretical mean for this waiting time process is $\mu_X = 3.2$. This process is a difficult test case since the marginal distribution of waiting times has been observed to possess large values of skewness and kurtosis and therefore is highly nonnormal.

Table 4.28 shows a comparison of the performance of **WASSP** (using $A = 7$) with ASAP2

Table 4.27: Performance of $\mathcal{WASSP}$ and ASAP2 for the ARTOJ process (4.25). Results are based on 400 independent replications of nominal 95% CIs.

Precision Requirement	Performance Measure	Procedure		
		$\mathcal{WASSP}(A=7)$	$\mathcal{WASSP}(A=9)$	ASAP2
None	CI coverage	93.25%	90%	95.5%
	Avg. sample size	10,319	9,919	40,160.6
	Max. sample size	246,336	60,336	185,344
	Avg. CI half-length	0.3986	0.3628	0.1431
	Var. CI half-length	0.0258	0.0225	0.0039
±15%	CI coverage	95.75%	93%	96%
	Avg. sample size	92,107	81,235	46,819.4
	Max. sample size	664,576	512,000	185,344
	Avg. CI half-length	0.1288	0.1300	0.1194
	Var. CI half-length	0.0004	0.0004	0.0005
±7.5%	CI coverage	97.75%	97.5%	96.8%
	Avg. sample size	420,790	377,064	126,220.4
	Max. sample size	3,170,304	7,618,560	260,866
	Avg. CI half-length	0.0652	0.0658	0.0708
	Var. CI half-length	0.0001	0.00009	0.00002

when both procedures are applied to the $M/M/1$/LIFO queue waiting time process. The results in Table 4.28 are based on 400 independent replications of nominal 90% CIs. At all three precision levels, $\mathcal{WASSP}$ performs slightly better than ASAP2 in terms of confidence interval coverage probability. However, the sample sizes required by $\mathcal{WASSP}$ are significantly larger than those required by ASAP2. In Table 4.29, the performance of $\mathcal{WASSP}$ and ASAP2 is compared for independent replications of nominal 95% CIs. From this table we see that, as in Table 4.28, $\mathcal{WASSP}$ performs slightly better than ASAP2 in terms of coverage; however $\mathcal{WASSP}$ requires larger sample sizes than ASAP2 does.

One possible explanation for the difference in performance between $\mathcal{WASSP}$ and ASAP2 in this process is that the ASAP2 algorithm relies much more heavily on the normality of the batch means than $\mathcal{WASSP}$ does. As discussed earlier, the normalization step in the $\mathcal{WASSP}$ algorithm is used only to expedite the convergence of the periodogram to its asymptotic properties. The normalization step in the ASAP2 algorithm, however, plays a more central role. In particular, joint multivariate normality of the batch means is required to derive expressions for the first four cumulants of the NOBM t-ratio (2.7), from which the adjusted confidence interval (2.8) is then computed. The final average sample sizes in Tables 4.28 and 4.29 suggest that ASAP2 may not be taking sample sizes large enough to ensure joint multivariate normality of the batch means.

Table 4.28: Performance of **WASSP** (using $A = 7$) and ASAP2 for the $M/M/1$/LIFO queue waiting time process with 80% server utilization and empty-and-idle initial condition. Results are based on 400 independent replications of nominal 90% CIs.

Precision Requirement	Performance Measure	Procedure	
		WASSP	ASAP2
None	# replications	400	400
	CI coverage	90%	87%
	Avg. sample size	97,990	53,957
	Max. sample size	954,112	185,344
	Avg. CI half-length	0.3359	0.261
	Var. CI half-length	0.0397	0.005
±15%	# replications	400	400
	CI coverage	91%	86%
	Avg. sample size	105,060	54,017
	Max. sample size	954,112	185,344
	Avg. CI half-length	0.2922	0.260
	Var. CI half-length	0.0131	0.004
±7.5%	# replications	400	400
	CI coverage	91%	87%
	Avg. sample size	162,503	68,313
	Max. sample size	1,905,920	236,800
	Avg. CI half-length	0.1862	0.219
	Var. CI half-length	0.00182	0.0005

Table 4.29: Performance of **WASSP** (using $A = 7$) and ASAP2 for the $M/M/1$/LIFO queue waiting time process with 80% server utilization and empty-and-idle initial condition. Results are based on 400 independent replications of nominal 95% CIs.

Precision Requirement	Performance Measure	Procedure	
		WASSP	ASAP2
None	# replications	400	400
	CI coverage	92%	93%
	Avg. sample size	111,874	53,957
	Max. sample size	1,905,920	185,344
	Avg. CI half-length	0.4093	0.312
	Var. CI half-length	0.0621	0.008
±15%	# replications	400	400
	CI coverage	95%	93%
	Avg. sample size	126,433	54,255
	Max. sample size	1,905,920	185,344
	Avg. CI half-length	0.3142	0.3076
	Var. CI half-length	0.0117	0.0053
±7.5%	# replications	400	400
	CI coverage	95%	93%
	Avg. sample size	209,182	91,432
	Max. sample size	1,905,920	337,152
	Avg. CI half-length	0.19274	0.226
	Var. CI half-length	0.0013	0.001

Chapter 5

Conclusions and Future Research

5.1. Main Conclusions of the Research

We have proposed a wavelet-based spectral procedure for constructing an asymptotically valid confidence interval for the steady-state mean of a simulation output process. This procedure, called *WASSP*, addresses two fundamental problems associated with analyzing stochastic output from a nonterminating simulation—the initialization bias and correlation problems.

The primary advantage of a spectral method for steady-state output analysis is that it enables one to work with approximately uncorrelated periodogram values rather than with a highly correlated output sequence. The proposed method uses wavelets to approximate the log of the smoothed periodogram of the associated batch means process, from which we obtain an estimate of the steady-state variance constant of the original (unbatched) process. Together with a sample mean that has been suitably truncated to eliminate initialization bias, the SSVC estimator is used to construct a reliable confidence-interval estimator of the steady-state mean response that satisfies a user-specified absolute or relative precision requirement.

There are several key differences between *WASSP* and previous spectral methods (in particular, Heidelberger and Welch's spectral method) for steady-state output analysis. First, to smooth the periodogram Heidelberger and Welch average adjacent periodogram values. *WASSP*, on the other hand, allows the user to select a moving average of width 5, 7, 9, or 11 points. The main advantage to using a larger number of points in the moving average is that a less noisy estimate of the power spectrum can be obtained. Furthermore, having a range of values for the moving average width gives the user flexibility in selecting an appropriate value of the smoothing parameter.

A second key difference between *WASSP* and Heidelberger and Welch's method is that the latter procedure uses standard least-squares regression techniques to approximate the log of the smoothed periodogram by means of a simple quadratic polynomial. *WASSP*, however, uses

wavelets to approximate the log of the smoothed periodogram. The main premise behind using wavelets is that this approach can yield a more flexible and accurate estimator of the power spectrum than standard regression techniques can provide, especially in the neighborhood of zero frequency.

A third key difference between WASSP and Heidelberger and Welch's method is that WASSP always requires normalization via batching (aggregation) of the output process before the periodogram is constructed. One advantage of this normalizing step is that the periodogram is in general more well-behaved for normal data, and this effect facilitates estimation of the spectrum in a neighborhood of zero frequency. Finally, WASSP provides an automatic scheme for effectively identifying an appropriate warm-up period beyond which all relevant statistics are to be accumulated, thereby ensuring that the final delivered confidence interval is not affected by system warm-up bias.

In the experimental performance evaluation summarized in Chapter 4, we presented five test processes that were specifically designed to explore the robustness of WASSP and its competitors against the statistical anomalies commonly encountered in the analysis of outputs generated from large-scale, steady-state simulation experiments. We also used the same five test processes to compare the performance of Heidelberger and Welch's spectral method and ASAP2 with the performance of WASSP. From the experimental results presented in Chapter 4, it is evident that WASSP outperforms Heidelberger and Welch's method; and we believe WASSP represents an advance in spectral methods for simulation output analysis. Furthermore, we can conclude that while WASSP and ASAP2 produce comparable results in some cases, WASSP is in general a more robust procedure than ASAP2. In particular, we found that ASAP2 does not perform well when applied to processes with an exceptionally high lag-one correlation, like the AR(1) process described in Section 4.3. However, it is not entirely fair to compare WASSP and ASAP2 as we have done in Chapter 4 since ASAP2 is not designed to deliver an estimate of the SSVC. Both WASSP and ASAP2 are designed to deliver point and confidence-interval estimators for the steady-state mean of a simulation output process. WASSP, however, also provides an estimator of the SSVC with reasonably stable behavior.

5.2. Directions for Future Research

The experimental results detailed in Chapter 4 provide substantial evidence of WASSP's ability to deliver approximately valid confidence intervals for the steady-state mean of a simulation-generated output process with relative precision levels and nominal coverage probabilities that often arise in practical applications. However, it would be desirable to prove rigorously that for a nontrivial class of discrete-event stochastic systems, the confidence intervals delivered by WASSP are asymptotically valid—that is, they 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.

More generally, a careful theoretical examination of $\mathcal{WASSP}$'s asymptotic properties should address the question of *efficiency* in the sense of [11] and [44]. For example, if we are estimating μ_X by a $100(1-\beta)\%$ confidence interval ($0 < \beta < 1$) with maximum acceptable half-length h^* as in (3.5), then the "ideal" confidence interval (1.13) based on exact knowledge of γ_X will require the "best case" run length

$$n^*(h^*) = \left\lceil \frac{z^2_{1-\beta/2}\gamma_X}{(h^*)^2} \right\rceil .$$

If $n_{\mathcal{W}}(h^*)$ denotes the corresponding (random) run length required by $\mathcal{WASSP}$, then an analysis of $\mathcal{WASSP}$'s asymptotic efficiency in this context should focus on the behavior of

$$\frac{n_{\mathcal{W}}(h^*)}{n^*(h^*)} \quad \text{and} \quad \frac{\mathrm{E}[n_{\mathcal{W}}(h^*)]}{n^*(h^*)} \quad \text{as} \quad h^* \to 0. \tag{5.1}$$

A similar asymptotic efficiency analysis is required when $\mathcal{WASSP}$ delivers confidence intervals for μ_X with a user-specified relative precision level r^* as in (3.5). In this situation the "best case" run length based on exact knowledge of all the relevant parameters of the target output process is

$$n^*(r^*) = \left\lceil \frac{z^2_{1-\beta/2}\gamma_X}{(r^*\,\mu_X)^2} \right\rceil .$$

If $n_{\mathcal{W}}(r^*)$ denotes the corresponding (random) run length required by $\mathcal{WASSP}$, then the relevant analysis of $\mathcal{WASSP}$'s asymptotic efficiency should be based on the behavior of

$$\frac{n_{\mathcal{W}}(r^*)}{n^*(r^*)} \quad \text{and} \quad \frac{\mathrm{E}[n_{\mathcal{W}}(r^*)]}{n^*(r^*)} \quad \text{as} \quad r^* \to 0. \tag{5.2}$$

As an essential complement to future theoretical developments on $\mathcal{WASSP}$, we should continue the experimental work that has been designed to explore the robustness of $\mathcal{WASSP}$. This work will require identifying classes of simulation output processes beyond those described in Chapter 4 that will provide challenging test cases for $\mathcal{WASSP}$ and other state-of-the-art simulation analysis techniques. This extended performance evaluation should include applying $\mathcal{WASSP}$ to processes with long-range dependence [60].

Another direction of future research is to modify $\mathcal{WASSP}$ so that the value of the smoothing parameter A is automatically determined within the procedure based on the observed characteristics of the target output process as well as the user's specification of a confidence coefficient and precision requirement for the final confidence interval. Currently, the user selects the value of the smoothing parameter, with the default being $A = 7$. We saw in Chapter 4 that for some

cases, like the ARTOP process in Section 4.4, **WASSP** performs significantly better if we set the smoothing parameter $A = 5$ rather than using the default value. By automating the selection of the value of A, we relieve the user of the responsibility for setting an appropriate value of the smoothing parameter. The first step toward automating the selection of A would be to deduce a relationship between the shape of the spectrum $p_X(\omega)$ and the appropriate value of the smoothing parameter for each of the five test processes described in Chapter 4. To do this, we need to develop a computationally efficient method for calculating the spectrum $p_X(\omega)$ of an "AR(1)-to-Anything" process. Furthermore, as the batch size $m \to \infty$, the spectrum $p_{\bar{X}(m)}(\omega)$ of the associated batch means process becomes flatter and converges uniformly to zero for all $\omega \in [-\frac{1}{2}, \frac{1}{2}]$. Therefore, it would be desirable to allow the smoothing parameter $A \to \infty$ as $m \to \infty$; and such a modification might also improve the efficiency of **WASSP** as measured by (5.1) and (5.2).

In the future we would also like to determine an appropriate method for eliminating the bias that is introduced when in step **[10]** of **WASSP** we exponentiate $\widehat{\zeta}_{\bar{X}(m)}(0)$, the wavelet-based estimate of the log-spectrum of the batch means at zero frequency. When we compute the estimate of the spectrum of the batch means at zero frequency via equation (3.2), we make the following assumption,

$$\mathrm{E}\Big\{\exp\Big[\widehat{\zeta}_{\bar{X}(m)}(0)\Big]\Big\} = \exp\Big\{\mathrm{E}\Big[\widehat{\zeta}_{\bar{X}(m)}(0)\Big]\Big\}. \tag{5.3}$$

However, since the exponential function is convex, Jensen's inequality [49] implies that

$$\mathrm{E}\Big\{\exp\Big[\widehat{\zeta}_{\bar{X}(m)}(0)\Big]\Big\} \geq \exp\Big\{\mathrm{E}\Big[\widehat{\zeta}_{\bar{X}(m)}(0)\Big]\Big\}, \tag{5.4}$$

with strict inequality in (5.4) if $\mathrm{Var}[\widehat{\zeta}_{\bar{X}(m)}(0)] > 0$, a condition that holds in every nontrivial application of **WASSP**. Moreover, since we have seen that at zero frequency the **WASSP**-based batch means spectrum estimator yields asymptotically

$$\exp\Big\{\mathrm{E}\Big[\widehat{\zeta}_{\bar{X}(m)}(0)\Big]\Big\} \approx \exp\Big[\zeta_{\bar{X}(m)}(0)\Big] = p_{\bar{X}(m)}(0) = \gamma_{\bar{X}(m)} = \frac{\gamma_X}{m}, \tag{5.5}$$

it follows that on the average, we are actually overestimating the SSVC γ_X by assuming (5.3). Asymptotically, making the assumption (5.3) will have little effect on the performance of **WASSP** since the bias introduced by exponentiating $\widehat{\zeta}_{\bar{X}(m)}(0)$ goes to zero as the batch size m gets larger. However, assuming that (5.3) holds may affect **WASSP**'s performance in small-sample cases; and therefore it would be desirable to determine a method for correcting for the bias that is introduced when we exponentiate $\widehat{\zeta}_{\bar{X}(m)}(0)$. This would involve estimating $\mathrm{Var}[\widehat{\zeta}_{\bar{X}(m)}(0)]$, the variance of the wavelet-based estimate of the log-spectrum of the batch means at zero frequency.

Another direction of future research is to determine the effects of using wavelet basis func-

tions other than the s8 symmlet on the performance of **WASSP**. Furthermore, to obtain an estimate of the SSVC of the original (unbatched) process, **WASSP** generates a wavelet-based estimate of the batch means power spectrum $p_{\bar{X}(m)}(\omega)$ for every frequency $\omega \in [-\frac{1}{2}, \frac{1}{2}]$. However, since we are primarily interested in obtaining an estimate of the batch means power spectrum at zero frequency, a possible direction for future research is to devise a method for estimating $p_{\bar{X}(m)}(\omega)$ in a neighborhood of $\omega = 0$ that becomes progressively smaller as the batch size m increases and has the form

$$[-b(m), b(m)] \quad \text{where} \quad 0 < b(m+1) \leq b(m) \quad \text{for all } m \quad \text{and} \quad \lim_{m \to \infty} b(m) = 0.$$

Another possible avenue of future research involves the implementation of **WASSP**. We implemented the **WASSP** algorithm in MATLAB (as described in Appendix E). The data must first be loaded into MATLAB and then a function call is made to execute **WASSP**. It would be desirable, however, to produce a version of **WASSP** that can be executed from the simulation software package Arena [36]. This would allow the user to restart the simulation and easily collect additional data if necessary. We would also like to investigate how **WASSP** could be incorporated into other popular simulation packages, such as Automod [48], ProModel [25], and Extend [37].

In the future, it would also be interesting to develop a nonspectral wavelet-based method for steady-state simulation output analysis and see how it compares with **WASSP**. Such a method could possibly use the maximal overlap discrete wavelet transform (MODWT) [64] to obtain a nonspectral estimator of the SSVC γ_X. This MODWT-based method might yield estimators $\overline{\overline{X}}^{(\mathrm{M})}$ and $\widehat{\gamma}_X^{(\mathrm{M})}$ for μ_X and γ_X, respectively, with the following properties:

$$\widehat{\gamma}_X^{(\mathrm{M})} \longrightarrow \gamma_X \quad \text{with probability 1} \quad \text{as} \quad r^* \to 0 \quad \text{or} \quad h^* \to 0; \tag{5.6}$$

$$\frac{\overline{\overline{X}}^{(\mathrm{M})} - \mu_X}{\sqrt{\widehat{\gamma}_X^{(\mathrm{M})}/n}} \xrightarrow{D} N(0,1) \quad \text{as} \quad r^* \to 0 \quad \text{or} \quad h^* \to 0; \tag{5.7}$$

and

$$\begin{aligned} \lim_{r^* \to 0} \Pr\left\{ \mu_X \in \overline{\overline{X}}^{(\mathrm{M})} \pm z_{1-\beta/2} \sqrt{\frac{\widehat{\gamma}_X^{(\mathrm{M})}}{n}} \right\} &= \lim_{h^* \to 0} \Pr\left\{ \mu_X \in \overline{\overline{X}}^{(\mathrm{M})} \pm z_{1-\beta/2} \sqrt{\frac{\widehat{\gamma}_X^{(\mathrm{M})}}{n}} \right\} \\ &= 1 - \beta, \end{aligned} \tag{5.8}$$

where $\overline{\overline{X}}^{(\mathrm{M})} \pm z_{1-\beta/2}\sqrt{\widehat{\gamma}_X^{(\mathrm{M})}/n}$ is the final confidence interval delivered by the MODWT-based procedure.

For the **WASSP**-based estimators $\overline{\overline{X}}$ and $\widehat{\gamma}_X$, we have good empirical evidence (as exhibited in Chapter 4) showing that to a reasonable approximation,

$$\widehat{\gamma}_X \xrightarrow{D} \gamma_X \frac{\chi^2(2a)}{(2a)} \quad \text{as} \quad r^* \to 0 \quad \text{or} \quad h^* \to 0; \tag{5.9}$$

and thus we see that $\widehat{\gamma}_X$ does not converge to γ_X with probability one under any circumstances. Furthermore, all our experimentation indicates that **WASSP** yields an excellent approximation to the limit property

$$\frac{\overline{\overline{X}} - \mu_X}{\sqrt{\widehat{\gamma}_X / n}} \xrightarrow{D} t_{2a} \quad \text{as} \quad r^* \to 0 \quad \text{or} \quad h^* \to 0; \tag{5.10}$$

and thus an MODWT-based nonspectral method for steady-state output analysis with properties (5.6)–(5.8) could outperform **WASSP** at least asymptotically as the precision requirement tends to zero. It may be possible, however, to develop a variant of **WASSP** in which the smoothing parameter $a \to \infty$ as the batch size $m \to \infty$. If we allow the degrees of freedom in **WASSP**'s estimator $\widehat{\zeta}_{\bar{X}(m)}(\omega)$ of the log-spectrum $\zeta_{\bar{X}(m)}(\omega)$ of the batch means to go to infinity as the batch size $m \to \infty$, then it might be possible to obtain **WASSP**-based estimators $\overline{\overline{X}}$ and $\widehat{\gamma}_X$ with properties analogous to (5.6)–(5.8).

Finally, we hope in the future to apply wavelet techniques to simulation-related problems other than steady-state output analysis. For example, because of their flexibility, we believe wavelets could be used to estimate the density of a process used as an input to a large-scale simulation model. Furthermore, we believe wavelets could be used for estimation, analysis, and optimization of simulation metamodels—that is, regression models of simulation-generated responses as those responses depend on the simulation's input parameters or system design variables.

Bibliography

[1] Abramovich, F., T. C. Bailey, and T. Sapatinas. 2000. Wavelet analysis and its statistical applications. *The Statistician* 49 (1): 1–29.

[2] Abramowitz, M., and I. A. Stegun, ed. 1972. *Handbook of mathematical functions, with formulas, graphs, and mathematical tables*. New York: Dover Publications.

[3] Bachman, G., Narici, L., and E. Beckenstein. 2000. *Fourier and wavelet analysis*. New York: Springer.

[4] Bartlett, M. S., and D. G. Kendall. 1946. The statistical analysis of variance-heterogeneity and the logarithmic transformation. *Supplement to the Journal of the Royal Statistical Society* 8 (1): 128–138.

[5] Bickel, P. J., and K. A. Doksum. 1977. *Mathematical statistics: Basic ideas and selected topics*. San Francisco: Holden-Day.

[6] Billingsley, P. 1968. *Convergence of probability measures*. New York: Wiley.

[7] Box, G. E. P., and D. R. Cox. 1964. An analysis of transformations. *Journal of the Royal Statistical Society. Series B, Statistical Methodology* 26 (2): 211–243.

[8] Box, G. E. P., and G. M. Jenkins. 1994. *Time series analysis: Forecasting and control*. 3d ed. Englewood Cliffs, New Jersey: Prentice Hall.

[9] Bruce, A., and H.-Y. Gao. 1996. *Applied wavelet analysis with S-PLUS*. New York: Springer.

[10] Carleson, L. 1966. On the convergence and growth of partial sums of Fourier series. *Acta Mathematica* 116:135–137.

[11] 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.

[12] Cochran, W. G. 1977. *Sampling techniques*. 3d ed. New York: Wiley.

[13] Cramér, H. 1946. *Mathematical methods of statistics*. Princeton, New Jersey: Princeton University Press.

[14] Crane, M. A., and A. J. Lemoine. 1977. *An introduction to the regenerative method for simulation analysis*. New York: Springer-Verlag.

[15] Daley, D. J. 1968. The serial correlation coefficients of waiting times in a stationary single server queue. *Journal of the Australian Mathematical Society* 8:683–699.

[16] Daubechies, I. 1992. *Ten lectures on wavelets*. Philadelphia: Society for Industrial and Applied Mathematics.

[17] Donoho, D. L., and I. M. Johnstone. 1994. Ideal spatial adaptation by wavelet shrinkage. *Biometrika* 81 (3): 425–455.

[18] Donoho, D. L., and I. M. Johnstone. 1995. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association* 90 (432): 1200–1224.

[19] Fishman, G. S. 1978. *Principles of discrete event simulation*. New York: Wiley.

[20] Fishman, G. S. 1978. Grouping observations in digital simulation. *Management Science* 24 (5): 510–521.

[21] Gao, H.-Y. 1997. Choice of thresholds for wavelet shrinkage estimate of the spectrum. *Journal of Time Series Analysis* 18 (3): 231–251.

[22] Goldsman, D., and L. Schruben. 1984. Asymptotic properties of some confidence interval estimators for simulation output. *Management Science* 30 (10): 1217–1225.

[23] Gradshteyn, I. S., and I. M. Ryzhik. 1994. *Table of integrals, series, and products*. Edited by Alan Jeffrey and translated from the Russian by Scripta Technica, Inc. 5th ed. San Diego: Academic Press.

[24] Hanselman, D., and B. Littlefield. 1998. *Mastering* MATLAB 5: *A comprehensive tutorial and reference*. New Jersey: Prentic Hall.

[25] Harrell, C. R., and R. N. Price. 2002. Simulation modeling using PROMODEL technology. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 192–198. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Available online via <http://www.informs-cs.org/wsc02papers/024.PDF>.

[26] 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.

[27] Heidelberger, P., and P. D. Welch. 1981. Adaptive spectral methods for simulation output analysis. *IBM Journal of Research and Development* 25 (6): 860–876.

[28] Heidelberger, P., and P. D. Welch. 1983. Simulation run length control in the presence of an initial transient. *Operations Research* 31 (6): 1109–1144.

[29] Henderson, S. G., and P. W. Glynn. 2001. Regenerative steady-state simulation of discrete-event systems. *ACM Transactions on Modeling and Computer Simulation* 11 (4): 313–345.

[30] Iglehart, D. L. 1975. Simulating stable stochastic systems, V: Comparison of ratio estimators. *Naval Research Logistics Quarterly* 22:553–565.

[31] Iglehart, D. L. 1978. The regenerative method for simulation analysis. In *Current trends in programming methodology, Vol. III: Software modeling*, ed. K. M. Chandy and R. T. Yeh, 52–71. Englewood Cliffs, New Jersey: Prentice-Hall.

[32] Irizarry, M. A., M. E. Kuhl, E. K. Lada, S. Subramanian, and J. R. Wilson. 2003. Analyzing transformation-based simulation metamodels. *IIE Transactions* to appear.

[33] Jenkins, G. M., and D. G. Watts. 1968. *Spectral analysis and its applications*. San Francisco: Holden-Day.

[34] Johnson, N. L., S. Kotz, and N. Balakrishnan. 1994. *Continuous univariate distributions, Vol. 1*. 2d ed. New York: Wiley.

[35] Kaiser, G. 1994. *A friendly guide to wavelets*. Boston: Birkhäuser.

[36] Kelton, W. D., R. P. Sadowski, and D. A. Sadowski. 2002. *Simulation with Arena*. 2d ed. Boston: McGraw-Hill.

[37] Krahl, D. 2002. The Extend simulation environment. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 205–213. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Available online via <http://www.informs-cs.org/wsc02papers/026.PDF>.

[38] Lada, E. K., J. C. Lu, and J. R. Wilson. 2001. A wavelet-based procedure for process fault detection. *IEEE Transactions on Semiconductor Manufacturing* 15 (1): 79–90.

[39] Law, A. M., and W. D. Kelton. 2000. *Simulation modeling and analysis*. 3d ed. Boston: McGraw-Hill.

[40] Malkovich, J. F., and A. A. Afifi. 1973. On tests for multivariate normality. *Journal of the American Statistical Association* 68 (341): 176–179.

[41] Mallat, S. G. 1989. Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbb{R})$. *Transactions of the American Mathematical Society* 315 (1): 69–87.

[42] Mallat, S. G. 1998. *A wavelet tour of signal processing*. San Diego: Academic Press.

[43] Moulin, P. 1994. Wavelet thresholding techniques for power spectrum estimation. *IEEE Transactions on Signal Processing* 42 (11): 3126–3136.

[44] 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.

[45] Nason, G. P. 1995. Choice of the threshold parameter in wavelet function estimation. In *Wavelets and statistics*, ed. A. Antoniadis and G. Oppenheim, 261–280. New York: Springer-Verlag.

[46] Percival, D. B., and A. T. Walden. 2000. *Wavelet methods for time series analysis*. Cambridge: Cambridge University Press.

[47] Priestley, M. B. 1981. *Spectral analysis and time series*. London: Academic Press.

[48] Rohrer, M. W., and I. W. McGregor. 2002. Simulating reality using AutoMod. 2002. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 173–181. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Available online via <http://www.informs-cs.org/wsc02papers/021.PDF>.

[49] Ross, S. M. 1983. *Stochastic processes*. New York: Wiley.

[50] Ross, S. M. 1997. *Introduction to probability models*. 6th ed. San Diego: Academic Press.

[51] Royston, J. P. 1982. An extension of Shapiro and Wilk's W test for normality to large samples. *Applied Statistics* 31 (2): 115–124.

[52] Saito, N. 1994. Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion. In *Wavelets in geophysics*, ed. E. Foufoula-Georgiou and P. Kumar, 299–324. New York: Academic Press.

[53] Satterthwaite, F. E. 1946. An approximate distribution of estimates of variance components. *Biometrics Bulletin* 2 (6): 110–114.

[54] Serroukh, A., and A. T. Walden. 2000. Wavelet scale analysis of bivariate time series I: Motivation and estimation. *Journal of Nonparametric Statistics* 13 (1): 1–36.

[55] Shapiro, S. S., and M. B. Wilk. 1965. An analysis of variance test for normality (complete samples). *Biometrika* 52 (3–4): 591–611.

[56] Song, W. T., and B. W. Schmeiser. 1995. Optimal mean-squared-error batch sizes. *Management Science* 41 (1): 110–123.

[57] Steiger, N. M., and J. R. Wilson. 2002. An improved batch means procedure for simulation output analysis. *Management Science* 48 (12): 1569–1586.

[58] Steiger, N. M., C. Alexopoulos, D. Goldsman, E. K. Lada, J. R. Wilson, 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, 336–344. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Available online via <http://www.informs-cs.org/wsc02papers/043.PDF> [accessed December 16, 2002].

[59] Stuart, A., and J. K. Ord. 1994. *Kendall's advanced theory of statistics, Vol. 1: Distribution theory*. 6th ed. London: Edward Arnold.

[60] Suárez-González, A., J. C. López-Ardao, C. López-García, M. Rodríguez-Pérez, M. Fernández-Veiga, and M. E. Sousa-Vieira. 2002. A batch means procedure for mean value estimation of processes exhibiting long range dependence. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 456–464. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Avail-

able online via <http://www.informs-cs.org/wsc02papers/057.PDF>.

[61] Swain, J. J., Venkatraman, S., and J. R. Wilson. 1988. Least-squares estimation of distribution functions in Johnson's translation system. *Journal of Statistical Computation and Simulation* 29:271–297.

[62] Vidakovic, B. 1999. *Statistical modeling by wavelets*. New York: Wiley.

[63] Wahba, G. 1980. Automatic smoothing of the log periodogram. *Journal of the American Statistical Association* 75 (369): 122–132.

[64] Walden, A. T. 2001. Wavelet analysis of discrete time series. In *Proceedings of the European Congress of Mathematics: Barcelona, July 10–14, 2000*, ed. C. Casacuberta, R. M. Miró-Roig, J. Verdera, and S. Xambó, 627–641. Basel: Birkhäuser Verlag. Available online via <http://mat.uab.es/art3ecm/walden.pdf>.

[65] Walden, A. T., D. B. Percival, and E. J. McCoy. 1998. Spectrum estimation by wavelet thresholding of multitaper estimators. *IEEE Transactions on Signal Processing* 46 (12): 3153–3165.

[66] Walker, J. S. 1999. *A primer on wavelets and their scientific applications*. Boca Raton: CRC Press.

[67] Welch, B. L. 1956. On linear combinations of several variances. *Journal of the American Statistical Association* 51 (273): 132–148.

[68] Welch, P. D. 1983. The statistical analysis of simulation results. In *Computer performance modeling handbook*, ed. S. S. Lavenberg, 268–329. New York: Academic Press.

[69] Wilson, J. R. 2003. Personal communication.

[70] Wilson, J. R., and A. A. B. Pritsker. 1978a. A survey of research on the simulation startup problem. *Simulation* 31 (2): 55–58.

[71] Wilson, J. R., and A. A. B. Pritsker. 1978b. Evaluation of startup policies in simulation experiments. *Simulation* 31 (3): 79–89.

[72] Young, L. C. 1941. On randomness in ordered sequences. *The Annals of Mathematical Statistics* 12:293–300.

Appendix A

Bias Adjustment in Estimating the Log-Spectrum of the Batch Means

In this appendix, we compute the mean and variance of $\widetilde{\mathcal{L}}_{\bar{X}(m)}(\frac{l}{k})$, the log-smoothed-periodogram of the batch means at the Fourier frequency $\frac{l}{k}$, for $l = 0, \pm 1, \ldots, \pm(\frac{k}{2} - 1), \frac{k}{2}$ so as to obtain a bias adjustment to $\widetilde{\mathcal{L}}_{\bar{X}(m)}(\frac{l}{k})$ that will yield an unbiased estimator of $\zeta_{\bar{X}(m)}(\frac{l}{k})$, the log-spectrum of the batch means. Throughout this appendix, $\{\chi^2_u(2) : u = 1, 2, \ldots\}$ denotes a set of i.i.d. chi-square variates, each with 2 degrees of freedom.

A.1. Bias Adjustment at Frequency $\frac{l}{k}$ for $l = 0$ and $l = \frac{k}{2}$

For $l = 0$, smoothing parameter $A = 2a + 1$, and $I_{\bar{X}(m)}(0)$ as defined in (3.18), we see from (3.22) that $\widetilde{I}_{\bar{X}(m)}(0)$, the smoothed periodogram of the batch means at zero frequency, is given by,

$$\begin{aligned}
\widetilde{I}_{\bar{X}(m)}(0) &= \frac{1}{A}\sum_{u=-a}^{a} I_{\bar{X}(m)}\left(\frac{u}{k}\right) \\
&= \frac{1}{A}\sum_{u=-a}^{-1} I_{\bar{X}(m)}\left(\frac{u}{k}\right) + \frac{1}{A} I_{\bar{X}(m)}(0) + \frac{1}{A}\sum_{u=1}^{a} I_{\bar{X}(m)}\left(\frac{u}{k}\right) \\
&= \frac{1}{A}\sum_{u=1}^{a} I_{\bar{X}(m)}\left(\frac{u}{k}\right) + \frac{1}{Aa}\sum_{u=1}^{a} I_{\bar{X}(m)}\left(\frac{u}{k}\right) + \frac{1}{A}\sum_{u=1}^{a} I_{\bar{X}(m)}\left(\frac{u}{k}\right) \\
&= \frac{2a+1}{aA}\sum_{u=1}^{a} I_{\bar{X}(m)}\left(\frac{u}{k}\right) \\
&= \frac{1}{a}\sum_{u=1}^{a} I_{\bar{X}(m)}\left(\frac{u}{k}\right)
\end{aligned}$$

$$
\begin{aligned}
&\dot{\sim} \quad \frac{1}{a}\sum_{u=1}^{a} p_{\bar{X}(m)}(0)\,\frac{\chi_u^2(2)}{2} \\
&\sim \quad p_{\bar{X}(m)}(0)\,\frac{\chi^2(2a)}{2a},
\end{aligned}
$$

since in general the sum of n i.i.d. chi-square random variables with ν degrees of freedom is a chi-square random variable with $n\nu$ degrees of freedom. Taking the natural log of the smoothed periodogram of the batch means at zero frequency, we have

$$
\begin{aligned}
\widetilde{\mathcal{L}}_{\bar{X}(m)}(0) &= \ln[\widetilde{I}_{\bar{X}(m)}(0)] \\
&\dot{\sim} \quad \ln\left[p_{\bar{X}(m)}(0)\,\frac{\chi^2(2a)}{2a}\right] \\
&= \ln[p_{\bar{X}(m)}(0)] + \ln\left[\frac{\chi^2(2a)}{2a}\right].
\end{aligned}
$$

Therefore, the log of the smoothed periodogram of the batch means at zero frequency has the following expected value,

$$
\begin{aligned}
\mathrm{E}\Big[\widetilde{\mathcal{L}}_{\bar{X}(m)}(0)\Big] &\approx \mathrm{E}\Big\{\ln\Big[p_{\bar{X}(m)}(0)\Big]\Big\} + \mathrm{E}\left[\ln\left(\frac{\chi^2(2a)}{2a}\right)\right] \\
&= \zeta_{\bar{X}(m)}(0) + \mathrm{E}\left[\ln\left(\frac{\chi^2(2a)}{2a}\right)\right]. \qquad \text{(A.1)}
\end{aligned}
$$

Since we are using $\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)$, the log of the smoothed periodogram of the batch means, to estimate $\zeta_{\bar{X}(m)}\left(\frac{l}{k}\right)$, the log-spectrum of the batch means, the term

$$
\mathrm{E}\left\{\ln\left[\frac{\chi^2(2a)}{2a}\right]\right\}
$$

is the bias introduced at frequency zero after taking the logarithm of $\widetilde{I}_{\bar{X}(m)}(0)$; and we need to derive a computational formula for this bias term to obtain an unbiased estimator of $\zeta_{\bar{X}(m)}(0)$ based on $\widetilde{\mathcal{L}}_{\bar{X}(m)}(0)$. In general, the expected value of a random variable of the form

$$
B = \ln\left[\frac{\chi^2(\nu)}{\nu}\right] \qquad \text{(A.2)}
$$

can be computed as follows. The moment generating function of B is given by

$$
M_B(t) \equiv \mathrm{E}\Big[e^{tB}\Big] = \int_0^\infty \exp\left[t\ln\left(\frac{x}{\nu}\right)\right]\frac{1}{2^{\nu/2}\,\Gamma(\nu/2)}\,x^{\nu/2-1}\,e^{-x/2}\,dx \qquad \text{(A.3)}
$$

$$
\begin{aligned}
&= \frac{1}{2^{\nu/2}\,\Gamma(\nu/2)} \int_0^\infty \left(\frac{x}{\nu}\right)^t x^{\nu/2-1}\, e^{-x/2}\, dx \\
&= \frac{1}{2^{\nu/2}\,\Gamma(\nu/2)\,\nu^t} \int_0^\infty x^{(t+\nu/2)-1}\, e^{-x/2}\, dx \\
&= \frac{2^{t+\nu/2}\,\Gamma(t+\nu/2)}{2^{\nu/2}\,\Gamma(\nu/2)\,\nu^t} \int_0^\infty \frac{x^{t+\nu/2-1}\, e^{-x/2}}{2^{t+\nu/2}\,\Gamma(t+\nu/2)}\, dx \\
&= \frac{\Gamma(t+\nu/2)}{\Gamma(\nu/2)(\nu/2)^t},
\end{aligned}
$$

since the density function of a chi-square random variable with $2t+\nu$ degrees of freedom integrates to 1. That is,

$$
\int_0^\infty \frac{x^{t+\nu/2-1}\, e^{-x/2}}{2^{t+\nu/2}\,\Gamma(t+\nu/2)}\, dx = 1.
$$

The cumulant generating function of B is defined as follows [13],

$$
\begin{aligned}
K_B(t) &\equiv \ln\left[M_B(t)\right] \\
&= \ln\left[\frac{\Gamma(t+\nu/2)}{\Gamma(\nu/2)\,(\nu/2)^t}\right] \\
&= \ln\left[\Gamma(t+\nu/2)\right] - \ln\left[\Gamma(\nu/2)\right] - t\,\ln(\nu/2). \qquad \text{(A.4)}
\end{aligned}
$$

Taking the first derivative of $K_B(t)$ gives

$$
K_B'(t) = \frac{\Gamma'(t+\nu/2)}{\Gamma(t+\nu/2)} - \ln(\nu/2).
$$

Since the expected value of B is the first derivative of $K_B(t)$ evaluated at $t=0$, we have

$$
\begin{aligned}
\mathrm{E}\Big\{\ln\left[\chi^2(\nu)/\nu\right]\Big\} &= \mathrm{E}[B] \\
&= K_B'(t)|_{t=0} \\
&= \frac{\Gamma'(\nu/2)}{\Gamma(\nu/2)} - \ln(\nu/2) \\
&= \Psi(\nu/2) - \ln(\nu/2), \qquad \text{(A.5)}
\end{aligned}
$$

where $\Psi(\cdot)$ is the digamma function [23].

Using the result (A.5) with $\nu = 2a$, we see that the expected value of the log of the smoothed

periodogram of the batch means at zero frequency as given by (A.1) is

$$\mathrm{E}\Big[\widetilde{\mathcal{L}}_{\bar{X}(m)}(0)\Big] \approx \zeta_{\bar{X}(m)}(0) + \Psi(a) - \ln(a).$$

Conducting a similar analysis for the frequency $\frac{l}{k} = \frac{1}{2}$, we find that the expected value of the log of the smoothed periodogram of the batch means at frequency $l = \frac{1}{2}$ is

$$\mathrm{E}\Big[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\tfrac{1}{2}\right)\Big] \approx \zeta_{\bar{X}(m)}\left(\tfrac{1}{2}\right) + \Psi(a) - \ln(a).$$

A.2. Bias Adjustment at Frequency $\frac{l}{k}$ for $a < |l| < \frac{k}{2} - a$

At the frequency $\frac{l}{k}$ for $a < |l| < \frac{k}{2} - a$, the smoothed periodogram of the batch means is given by

$$\begin{aligned}
\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right) &= \frac{1}{A}\sum_{u=-a}^{a} I_{\bar{X}(m)}\left(\frac{l+u}{k}\right) \\
&\dot{\sim} \frac{1}{A}\sum_{u=1}^{A} p_{\bar{X}(m)}\left(\frac{l}{k}\right)\frac{\chi^2_u(2)}{2} \\
&\sim p_{\bar{X}(m)}\left(\frac{l}{k}\right)\frac{\chi^2(2A)}{2A}.
\end{aligned}$$

Taking the natural log, we have

$$\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) = \ln\left[\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \;\dot{\sim}\; \ln\left[p_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] + \ln\left[\frac{\chi^2(2A)}{2A}\right].$$

The expected value of the log of the smoothed periodogram of the batch means is

$$\begin{aligned}
\mathrm{E}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] &\approx \mathrm{E}\left\{\ln\left[p_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]\right\} + \mathrm{E}\left[\ln\left(\frac{\chi^2(2A)}{2A}\right)\right] \\
&= \zeta_{\bar{X}(m)}\left(\frac{l}{k}\right) + \mathrm{E}\left[\ln\left(\frac{\chi^2(2A)}{2A}\right)\right]. \qquad \text{(A.6)}
\end{aligned}$$

Using the result (A.5) with $\nu = 2A$, we see that the expected value of the bias term in equation (A.6) at the frequency $\frac{l}{k}$ for $a < |l| < \frac{k}{2} - a$ is

$$\mathrm{E}\left\{\ln\left[\frac{\chi^2(2A)}{2A}\right]\right\} \approx \Psi(A) - \ln(A).$$

A.3. Bias Adjustment at Frequency $\frac{l}{k}$ for $1 \leq |l| \leq a$

At the frequency $\frac{l}{k}$ for $1 \leq |l| \leq a$, we observe that $\widetilde{I}_{\bar{X}(m)}(\frac{l}{k})$ is a weighted average of independent scaled chi-square variates in which the weights are all positive constants that are not necessarily equal in value; and thus the results of Satterthwaite [53] and Welch [67] ensure that an excellent approximation to the distribution of $\widetilde{I}_{\bar{X}(m)}(\frac{l}{k})$ is given by

$$\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right) \;\dot{\sim}\; \frac{\chi^2(\nu_{|l|})}{\nu_{|l|}}\, p_{\bar{X}(m)}\left(\frac{l}{k}\right),$$

where $\nu_{|l|}$ is the "effective" degrees of freedom for $\widetilde{I}_{\bar{X}(m)}(\frac{l}{k})$ as formulated by Satterthwaite. The log of the smoothed periodogram of the batch means can then be written as follows,

$$\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) = \ln\left[\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \;\dot{\sim}\; \ln\left[p_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] + \ln\left[\frac{\chi^2(\nu_{|l|})}{\nu_{|l|}}\right].$$

Furthermore,

$$\begin{aligned} \mathrm{E}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] &\approx \mathrm{E}\left\{\ln\left[p_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]\right\} + \mathrm{E}\left[\ln\left(\frac{\chi^2(\nu_{|l|})}{\nu_{|l|}}\right)\right] \\ &= \zeta_{\bar{X}(m)}\left(\frac{l}{k}\right) + \mathrm{E}\left[\ln\left(\frac{\chi^2(\nu_{|l|})}{\nu_{|l|}}\right)\right]. \end{aligned} \tag{A.7}$$

Before (A.7) can be simplified further, the effective degrees of freedom ν_l must be computed. In general, the effective degrees of freedom ν_{eff} for $\widetilde{I}_{\bar{X}(m)}(\frac{l}{k})$ is defined as follows,

$$\nu_{\text{eff}} = \frac{2}{\mathrm{CV}^2\left[\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]}, \tag{A.8}$$

where $\mathrm{CV}[\widetilde{I}_{\bar{X}(m)}(\frac{l}{k})]$ is the coefficient of variation of $\widetilde{I}_{\bar{X}(m)}(\frac{l}{k})$,

$$\mathrm{CV}^2\left[\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \equiv \frac{\mathrm{Var}\left[\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]}{\mathrm{E}^2\left[\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]}. \tag{A.9}$$

For the frequency $\frac{l}{k}$ with $1 \leq |l| \leq a$, the expected value of the smoothed periodogram of the batch means is computed as follows,

$$\mathrm{E}\left[\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] = \mathrm{E}\left\{\frac{1}{A}\left[\sum_{u=1}^{a-l} 2I_{\bar{X}(m)}\left(\frac{u}{k}\right) + I_{\bar{X}(m)}(0) + \sum_{u=a-l+1}^{l+a} I_{\bar{X}(m)}\left(\frac{u}{k}\right)\right]\right\}$$

$$
\begin{aligned}
&= \frac{1}{A}\left\{2\sum_{u=1}^{a-l}\mathrm{E}\Big[I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\Big]+\frac{1}{a}\sum_{u=1}^{a}\mathrm{E}\Big[I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\Big]\right.\\
&\qquad\left.+\sum_{u=a-l+1}^{l+a}\mathrm{E}\Big[I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\Big]\right\}\\
&\approx \frac{1}{A}\Big[2(a-l)\,p_{\bar{X}(m)}\Big(\frac{l}{k}\Big)+p_{\bar{X}(m)}\Big(\frac{l}{k}\Big)+2\,l\,p_{\bar{X}(m)}\Big(\frac{l}{k}\Big)\Big]\\
&= \frac{1}{A}\Big[(2a+1)\,p_{\bar{X}(m)}\Big(\frac{l}{k}\Big)\Big]\\
&= p_{\bar{X}(m)}\Big(\frac{l}{k}\Big). \qquad \text{(A.10)}
\end{aligned}
$$

Before computing the variance of the smoothed periodogram of the batch means, we first rewrite the smoothed periodogram as follows,

$$
\begin{aligned}
\widetilde{I}_{\bar{X}(m)}\Big(\frac{l}{k}\Big) &= \frac{1}{A}\left[\sum_{u=1}^{a-l}2I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)+\frac{1}{a}\sum_{u=1}^{a}I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)+\sum_{u=a-l+1}^{l+a}I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\right]\\
&= \frac{1}{A}\left[\sum_{u=1}^{a-l}\Big(2I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)+\frac{1}{a}I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\Big)+\frac{1}{a}\sum_{u=a-l+1}^{a}I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\right.\\
&\qquad\left.+\sum_{u=a-l+1}^{l+a}I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\right]\\
&= \frac{1}{A}\left[\Big(\frac{2a+1}{a}\Big)\sum_{u=1}^{a-l}I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)+\Big(\frac{a+1}{a}\Big)\sum_{u=a-l+1}^{a}I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\right.\\
&\qquad\left.+\sum_{u=a+1}^{l+a}I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\right]. \qquad \text{(A.11)}
\end{aligned}
$$

Now, the variance of the smoothed periodogram can be computed from (A.11):

$$
\begin{aligned}
\mathrm{Var}\Big[\widetilde{I}_{\bar{X}(m)}\Big(\frac{l}{k}\Big)\Big] &\approx \Big(\frac{1}{2a+1}\Big)^2\left\{\Big(\frac{2a+1}{a}\Big)^2\sum_{u=1}^{a-l}\mathrm{Var}\Big[I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\Big]\right.\\
&\qquad\left.+\Big(\frac{a+1}{a}\Big)^2\sum_{u=a-l+1}^{a}\mathrm{Var}\Big[I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\Big]\right.
\end{aligned}
$$

$$+ \sum_{u=a+1}^{l+a} \mathrm{Var}\left[I_{\bar{X}(m)}\left(\frac{u}{k}\right)\right]\Bigg\}$$

$$= \left(\frac{1}{2a+1}\right)^2 p^2_{\bar{X}(m)}\left(\frac{l}{k}\right)\Bigg\{\left(\frac{2a+1}{a}\right)^2 \sum_{u=1}^{a-l} \mathrm{Var}\left[\frac{\chi^2_u(2)}{2}\right]$$

$$+ \left(\frac{a+1}{a}\right)^2 \sum_{u=a-l+1}^{a} \mathrm{Var}\left[\frac{\chi^2_u(2)}{2}\right]$$

$$+ \sum_{u=a+1}^{l+a} \mathrm{Var}\left[\frac{\chi^2_u(2)}{2}\right]\Bigg\}$$

$$= \left(\frac{1}{2a+1}\right)^2 p^2_{\bar{X}(m)}\left(\frac{l}{k}\right)\Bigg[\left(\frac{2a+1}{a}\right)^2 \sum_{u=1}^{a-l} 1 + \left(\frac{a+1}{a}\right)^2 \sum_{u=a-l+1}^{a} 1$$

$$+ \sum_{u=a+1}^{l+a} 1\Bigg]$$

$$= \left(\frac{1}{2a+1}\right)^2 p^2_{\bar{X}(m)}\left(\frac{l}{k}\right)\left[\left(\frac{2a+1}{a}\right)^2 (a-l) + \left(\frac{a+1}{a}\right)^2 l + l\right]$$

$$= p^2_{\bar{X}(m)}\left(\frac{l}{k}\right) \frac{4a^2 - 2al + 4a - 2l + 1}{a(2a+1)^2}. \tag{A.12}$$

Substituting (A.10) and (A.12) into (A.9), we have for $1 \leq |l| \leq a$

$$\mathrm{CV}^2\left[\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] = \frac{4a^2 - 2al + 4a - 2l + 1}{a(2a+1)^2} = \frac{4a^2 - 2al + 4a - 2l + 1}{aA^2}. \tag{A.13}$$

If for each $j \in \{1, 2, \ldots, a\}$ we let

$$\nu_j^{\#} = \frac{2aA^2}{4a^2 - 2aj + 4a - 2j + 1} \quad \text{and} \quad \nu_j = \left\lfloor \nu_j^{\#} \right\rfloor, \tag{A.14}$$

then we have from (A.8) that the effective degrees of freedom $\nu_{\mathrm{eff}} = \nu_{|l|}$ at frequency $\frac{l}{k}$ for $1 \leq |l| \leq a$. Finally, using the result (A.5) with $\nu = \nu_{|l|}$, we see that the expected value of the log of the smoothed periodogram of the batch means in equation (A.7) for frequency $\frac{l}{k}$ where $1 \leq |l| \leq a$ is approximately given by

$$\mathrm{E}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \approx \zeta_{\bar{X}(m)}\left(\frac{l}{k}\right) + \Psi\left(\frac{\nu_{|l|}}{2}\right) - \ln\left(\frac{\nu_{|l|}}{2}\right).$$

A.4. Bias Adjustment at Frequency $\frac{l}{k}$ for $\frac{k}{2} - a \leq |l| \leq \frac{k}{2} - 1$

At frequency $\frac{l}{k}$ for $\frac{k}{2} - a \leq |l| \leq \frac{k}{2} - 1$, the derivation of the bias term is the same as in the case for $1 \leq |l| \leq a$ since the periodogram is symmetric with respect to frequency $\frac{1}{2}$ in the same way that it is symmetric with respect to frequency zero. In particular, if we use the following approximation to the distribution of $\widetilde{I}_{\bar{X}(m)}(\frac{l}{k})$,

$$\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right) \;\dot{\sim}\; \frac{\chi^2(\nu_{k/2-|l|})}{\nu_{k/2-|l|}}\, p_{\bar{X}(m)}\left(\frac{l}{k}\right),$$

where $\nu_{k/2-|l|}$ is the "effective" degrees of freedom for $\widetilde{I}_{\bar{X}(m)}(\frac{l}{k})$ as formulated by Satterthwaite [53], then the log of the smoothed periodogram of the batch means can be written as follows,

$$\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right) = \ln\left[\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \;\dot{\sim}\; \ln\left[p_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] + \ln\left[\frac{\chi^2(\nu_{k/2-|l|})}{\nu_{k/2-|l|}}\right]$$

and

$$\mathrm{E}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \approx \zeta_{\bar{X}(m)}\left(\frac{l}{k}\right) + \mathrm{E}\left[\ln\left(\frac{\chi^2(\nu_{k/2-|l|})}{\nu_{k/2-|l|}}\right)\right]. \tag{A.15}$$

Furthermore, by an argument that parallels the argument leading to (A.10), we see that the expected value of the smoothed periodogram of the batch means at the frequency $\frac{l}{k}$ for $\frac{k}{2} - a \leq |l| \leq \frac{k}{2} - 1$ is

$$\begin{aligned}
\mathrm{E}\left[\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] &= \mathrm{E}\left\{\frac{1}{A}\left[\sum_{u=k-l-a}^{\frac{k}{2}-1} 2I_{\bar{X}(m)}\left(\frac{u}{k}\right) + I_{\bar{X}(m)}\left(\frac{1}{2}\right) + \sum_{u=l-a}^{k-l-a-1} I_{\bar{X}(m)}\left(\frac{u}{k}\right)\right]\right\} \\
&= p_{\bar{X}(m)}\left(\frac{l}{k}\right).
\end{aligned} \tag{A.16}$$

If we rewrite the smoothed periodogram as follows,

$$\begin{aligned}
\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right) &= \frac{1}{A}\left[\left(\frac{2a+1}{a}\right)\sum_{u=k-l-a}^{\frac{k}{2}-1} I_{\bar{X}(m)}\left(\frac{u}{k}\right) + \left(\frac{a+1}{a}\right)\sum_{u=\frac{k}{2}-a}^{k-l-a-1} I_{\bar{X}(m)}\left(\frac{u}{k}\right)\right. \\
&\qquad \left. + \sum_{u=l-a}^{\frac{k}{2}-(a+1)} I_{\bar{X}(m)}\left(\frac{u}{k}\right)\right],
\end{aligned} \tag{A.17}$$

then the result analogous to (A.12) at frequency $\frac{l}{k}$ for $\frac{k}{2} - a \leq |l| \leq \frac{k}{2} - l$ is

$$\mathrm{Var}\left[\widetilde{I}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \approx p^2_{\bar{X}(m)}\left(\frac{l}{k}\right) \frac{4a^2 - 2a(\frac{k}{2} - l) + 4a - 2(\frac{k}{2} - l) + 1}{a(2a+1)^2}. \tag{A.18}$$

Therefore from (A.9) and (A.8), we have the effective degrees of freedom $\nu_{\text{eff}} = \nu_{k/2-|l|}$ for $\widetilde{I}_{\bar{X}(m)}(\frac{l}{k})$ at frequency $\frac{l}{k}$ for $\frac{k}{2} - a \leq |l| \leq \frac{k}{2} - 1$, where ν_j is as defined in (A.14).

Finally, using the result (A.5) with $\nu = \nu_{k/2-|l|}$, we see that the expected value of the log of the smoothed periodogram of the batch means in equation (A.15) at frequency $\frac{l}{k}$ for $\frac{k}{2} - a \leq |l| \leq \frac{k}{2} - 1$ is approximately given by

$$\mathrm{E}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \approx \zeta_{\bar{X}(m)}\left(\frac{l}{k}\right) + \Psi\left(\frac{\nu_{k/2-|l|}}{2}\right) - \ln\left(\frac{\nu_{k/2-|l|}}{2}\right).$$

Appendix B

Variance Reduction Analysis for Estimating the Log-Spectrum of the Batch Means Process

In Chapter 3 we proposed an estimator $\widetilde{\mathcal{L}}_{\bar{X}(m)}(\frac{l}{k})$ of the log-spectrum $\zeta_{\bar{X}(m)}(\frac{l}{k})$ for the batch means process. The estimator $\widetilde{\mathcal{L}}_{\bar{X}(m)}(\frac{l}{k})$ is obtained by computing the natural log of the smoothed periodogram of the batch means. We also proposed in Chapter 3 an alternative estimator of the log-spectrum of the batch means, namely $\overline{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)$, that can be obtained by computing the natural log of the periodogram of the batch means and then smoothing the result by taking a moving average of log-periodogram values. In this Appendix we will derive expressions for the variances of these two estimators of $\zeta_{\bar{X}(m)}(\frac{l}{k})$ at frequencies $l = 0, \pm 1, \pm 2, \ldots, \pm(\frac{k}{2} - 1), \pm\frac{k}{2}$. Furthermore, we will prove that a variance reduction is achieved in estimating the log-spectrum of the batch means by working with $\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)$ rather than with $\overline{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)$. Throughout this appendix, $\{\chi^2_u(2) : u = 1, 2, \ldots\}$ denotes a set of i.i.d. chi-square variates, each with 2 degrees of freedom.

To derive expressions for the variances of $\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)$ and $\overline{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)$, we will need to compute the variance of a random variable of the form

$$B = \ln\left[\frac{\chi^2(\nu)}{\nu}\right]. \tag{B.1}$$

Using the definition of the cumulant generating function of B in (A.4), we have

$$\begin{aligned} \mathrm{Var}[B] &= K''_B(t)|_{t=0} \\ &= \frac{\Gamma''(\frac{\nu}{2})\ \Gamma(\frac{\nu}{2}) - \Gamma'(\frac{\nu}{2})}{\left[\Gamma(\frac{\nu}{2})\right]^2} \end{aligned}$$

$$= \quad \Psi'\left(\frac{\nu}{2}\right), \tag{B.2}$$

where

$$\Psi'(z) = \frac{d}{dz}\,\Psi(z) \text{ for all } z \text{ with } \mathrm{Re}(z) > 0$$

is the trigamma function [23].

To establish that we achieve a variance reduction in estimating the log-spectrum of the batch means by working with $\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)$, the log-smoothed-periodogram of the batch means, rather than $\overline{\mathcal{L}}_{\bar{X}(m)}(\frac{l}{k})$, the smoothed-log-periodogram of the batch means, we will need the following two propositions.

Proposition 1 *If j is an integer and $j \geq 2$, then*

$$\Psi'(j) < \frac{\Psi'(1)}{j}. \tag{B.3}$$

Proof: From equation (6.4.3) of [2], we have

$$\Psi'(j) = -\left[-\zeta(2) + \sum_{l=1}^{j-1} \frac{1}{l^2}\right] \quad \text{for} \quad j = 2, 3, \ldots, \tag{B.4}$$

where

$$\zeta(s) \equiv \sum_{l=1}^{\infty} \frac{1}{l^s} \quad \text{for} \quad \mathrm{Re}(s) > 1 \tag{B.5}$$

is the Riemann zeta function. From equations (6.4.2) and (23.2.24) of [2], we see that

$$\Psi'(1) = \zeta(2) = \frac{\pi^2}{6}. \tag{B.6}$$

Combining (B.4) and (B.6), we have

$$\Psi'(j) = \Psi'(1) - \sum_{l=1}^{j-1} \frac{1}{l^2} \quad \text{for} \quad j = 2, 3, \ldots. \tag{B.7}$$

Now the desired conclusion (B.3) holds if and only if we have

$$\Psi'(1) - \sum_{l=1}^{j-1} \frac{1}{l^2} < \frac{\Psi'(1)}{j} \quad \text{for} \quad j = 2, 3, \ldots; \tag{B.8}$$

and (B.8) in turn holds if and only if we have

$$\Psi'(1) < \frac{j}{j-1} \sum_{l=1}^{j-1} \frac{1}{l^2} \quad \text{for} \quad j = 2, 3, \ldots . \tag{B.9}$$

To establish (B.9), we examine the properties of the function

$$\Upsilon(j) \equiv \frac{j}{j-1} \sum_{l=1}^{j-1} \frac{1}{l^2} \quad \text{for} \quad j = 2, 3, \ldots . \tag{B.10}$$

From the definition (B.10) of the function $\Upsilon(j)$, we have

$$\begin{aligned} \Upsilon(j+1) &= \frac{j+1}{j} \left[\frac{j-1}{j} \Upsilon(j) + \frac{1}{j^2} \right] \\ &= \Upsilon(j) - \left[\frac{\Upsilon(j)}{j^2} - \frac{j+1}{j^3} \right] \quad \text{for} \quad j = 2, 3, \ldots . \end{aligned} \tag{B.11}$$

Since

$$\frac{j^2-1}{j^2} < 1 < \sum_{l=1}^{j-1} \frac{1}{l^2} \quad \text{for} \quad j = 2, 3, \ldots , \tag{B.12}$$

it follows immediately from (B.10) and (B.12) that

$$\frac{\Upsilon(j)}{j^2} - \frac{j+1}{j^3} > 0 \quad \text{for} \quad j = 2, 3, \ldots ; \tag{B.13}$$

and combining (B.13) and (B.11), we see that

$$\Upsilon(j+1) < \Upsilon(j) \quad \text{for} \quad j = 2, 3, \ldots , \tag{B.14}$$

so that $\{\Upsilon(j) : j = 2, 3, \ldots\}$ is a strictly monotone decreasing sequence with lower limit

$$\lim_{j \to \infty} \Upsilon(j) = \sum_{l=1}^{\infty} \frac{1}{l^2} = \zeta(2) = \Psi'(1). \tag{B.15}$$

It follows from (B.14) and (B.15) that (B.9) holds; and thus (B.8) holds so that the desired conclusion (B.3) is true.

Proposition 2 *If x is a real number and $x \geq 3$, then*

$$\Psi' \left(\frac{\lfloor x \rfloor}{2} \right) < \frac{\Psi'(1)}{x/2}. \tag{B.16}$$

Proof: For $x = 3$, we have

$$\Psi'\left(\frac{3}{2}\right) = \frac{\pi^2}{2} - 4 < \frac{\Psi'(1)}{3/2} = \frac{\pi^2}{9}; \tag{B.17}$$

and for $x = 4$, we have

$$\Psi'(2) = \frac{\pi^2}{6} - 1 < \frac{\Psi'(1)}{2} = \frac{\pi^2}{12} \tag{B.18}$$

by equations **8.366** 8 and **8.366** 12 of [23]. Since the functions

$$g_1(x) \equiv \frac{\Psi'(1)}{x/2} \quad \text{and} \quad g_2(x) \equiv \Psi'\left(\frac{\lfloor x \rfloor}{2}\right) \quad \text{for all real} \quad x \geq 3 \tag{B.19}$$

have the derivatives

$$g_1'(x) < 0 \quad \text{and} \quad g_2'(x) = 0 \quad \text{for} \quad x \in (3, 4), \tag{B.20}$$

displays (B.17)–(B.20) imply that the desired conclusion (B.16) holds for all $x \in [3, 4]$. To complete the proof, we establish the desired result on arbitrary adjacent intervals of the form $[2j, 2j + 1]$ and $[2j + 1, 2j + 2]$ for $j = 2, 3, \ldots$.

For $x = 2j$, we have

$$g_2(2j) = \Psi'(j) < \frac{\Psi'(1)}{j} = g_1(2j) \tag{B.21}$$

by Proposition 1. Next we establish that

$$g_2(2j) = \Psi'(j) < \frac{\Psi'(1)}{j + \frac{1}{2}} = g_1(2j + 1) \quad \text{for} \quad j = 2, 3, \ldots. \tag{B.22}$$

Notice that (B.22) holds if and only if

$$\Psi'(1) > (j + \tfrac{1}{2})\, \Psi'(j) = (j + \tfrac{1}{2}) \left[\Psi'(1) - \sum_{l=1}^{j-1} \frac{1}{l^2}\right] \quad \text{for} \quad j = 2, 3, \ldots, \tag{B.23}$$

(see (B.7)); and (B.23) in turn holds if and only if

$$\Psi'(1) < \frac{2j + 1}{2j - 1} \sum_{l=1}^{j-1} \frac{1}{l^2} \quad \text{for} \quad j = 2, 3, \ldots. \tag{B.24}$$

To prove (B.24), we examine the properties of the function

$$\Theta(j) \equiv \frac{2j+1}{2j-1} \sum_{l=1}^{j-1} \frac{1}{l^2} \quad \text{for} \quad j = 2, 3, \ldots . \tag{B.25}$$

As shown in Table B.1 below, it is easy to verify by direct numerical evaluation that

$$\Theta(j) > \Psi'(1) \quad \text{for} \quad j = 2, 3, 4, 5. \tag{B.26}$$

Table B.1: Results of evaluating $\Theta(j)$ as defined by (B.25) for $j = 2, \ldots, 5$.

j	$\Theta(j)$
2	1.666667
3	1.75
4	1.75
5	1.739969

Moreover, we see that

$$\Theta(j) - \Theta(j+1) = \frac{4}{(2j+1)(2j-1)} \sum_{l=1}^{j-1} \frac{1}{l^2} - \frac{(2j+3)}{(2j+1)j^2}; \tag{B.27}$$

and we assert that

$$\Theta(j) - \Theta(j+1) > 0 \quad \text{for} \quad j = 5, 6, \ldots . \tag{B.28}$$

In view of (B.27), we see that (B.28) holds if and only if

$$\sum_{l=1}^{j-1} \frac{1}{l^2} > 1 + \frac{1}{j} - \frac{3}{4j^2} \quad \text{for} \quad j = 5, 6, \ldots . \tag{B.29}$$

Direct numerical evaluation of (B.29) for $j = 5$ yields the value 1.4263611 on the left-hand side and the value 1.17 on the right-hand side. Moreover, it is clear that for $j = 5, 6, \ldots$, the sequence of values on the left-hand side of (B.29) is strictly monotone increasing to the limit $\frac{\pi^2}{6} > 1$. On the other hand, it is easy to verify that the sequence of values on the right-hand side of (B.29) is strictly monotone decreasing to the limit 1. It follows that (B.29) and (B.28) hold.

Finally we observe that

$$\lim_{j\to\infty} \Theta(j) = \sum_{l=1}^{\infty} \frac{1}{l^2} = \Psi'(1). \tag{B.30}$$

Combining (B.25), (B.26), (B.28), and (B.30), we see that (B.24) holds; and thus (B.22) holds. For the interval $(2j, 2j+1)$, we observe the following analogue of (B.20),

$$g_1'(x) < 0 \ \text{ and } \ g_2'(x) = 0 \ \text{ for } \ x \in (2j, 2j+1) \ \text{ and for } \ j = 2, 3, \ldots. \tag{B.31}$$

Combining (B.21), (B.22), and (B.31), we finally obtain the desired conclusion (B.16) for all $x \in [2j, 2j+1]$ and for $j = 2, 3, \ldots$.

To complete the proof, we need to establish that (B.16) holds for all $x \in [2j+1, 2j+2]$ and for $j = 2, 3, \ldots$. From equation **8.363** 8 of [23], we see that

$$\begin{aligned} g_2(2j+1) &= \Psi'(j+\tfrac{1}{2}) \\ &= \sum_{l=0}^{\infty} \frac{1}{(j+\frac{1}{2}+l)^2} \\ &< \sum_{l=0}^{\infty} \frac{1}{(j+l)^2} \\ &= \Psi'(j) \\ &= g_2(2j) \ \text{ for } \ j = 2, 3, \ldots; \end{aligned} \tag{B.32}$$

and in view of (B.22), we have

$$g_1(2j+1) > g_2(2j+1) \ \text{ for } \ j = 2, 3, \ldots. \tag{B.33}$$

Moreover, we have the following analogue of (B.31),

$$g_1'(x) < 0 \ \text{ and } \ g_2(x) = 0 \ \text{ for } \ x \in (2j+1, 2j+2) \ \text{ and for } \ j = 2, 3, \ldots. \tag{B.34}$$

Finally we have

$$g_2(2j+2) = \Psi'(j+1) < \frac{\Psi'(1)}{j+1} = g_1(2j+2) \ \text{ for } \ j = 2, 3, \ldots. \tag{B.35}$$

Combining (B.33), (B.34), and (B.35), we see that (B.16) holds for all $x \in [2j+1, 2j+2]$ and for $j = 2, 3, \ldots$. Putting all the results together, we have

$$\Psi'\left(\frac{\lfloor x \rfloor}{2}\right) < \frac{\Psi'(1)}{x/2} \ \text{ for } \ x \in [3, \infty); \tag{B.36}$$

and this completes the proof of Proposition 2.

B.1. Variance Computation at Frequency $\frac{l}{k}$ for $l = 0$ and $l = \frac{1}{2}$

In this section, we will compute the variance of the log of the smoothed periodogram of the batch means and the variance of the smoothed log-periodogram of the batch means at frequencies 0 and $\frac{1}{2}$. From (3.26), we have

$$\mathrm{Var}\Big[\widetilde{\mathcal{L}}_{\bar{X}(m)}(0)\Big] \approx \mathrm{Var}\Big\{\ln\Big[p_{\bar{X}(m)}(0)\Big]\Big\} + \mathrm{Var}\left\{\ln\left[\frac{\chi^2(2a)}{2a}\right]\right\}. \tag{B.37}$$

Taking $\nu = 2a$ and substituting into (B.2), we have

$$\mathrm{Var}\Big[\widetilde{\mathcal{L}}_{\bar{X}(m)}(0)\Big] = \Psi'(a). \tag{B.38}$$

Now for the smoothed log-periodogram of the batch means, $\overline{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)$, we first notice

$$\begin{aligned}
\mathrm{Var}\Big[\overline{\mathcal{L}}_{\bar{X}(m)}(0)\Big] &= \mathrm{Var}\left\{\frac{1}{a}\sum_{l=1}^{a}\ln\left[I_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]\right\} \\
&= \frac{1}{a^2}\sum_{l=1}^{a}\mathrm{Var}\left\{\ln\left[I_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]\right\} \\
&\approx \frac{1}{a}\mathrm{Var}\Big\{\ln\Big[I_{\bar{X}(m)}(0)\Big]\Big\},
\end{aligned} \tag{B.39}$$

assuming that for $l = 1, \ldots, a$, we have $\mathrm{Var}\{\ln[I_{\bar{X}(m)}(\frac{l}{k})]\} \approx \mathrm{Var}\{\ln[I_{\bar{X}(m)}(0)]\}$. From the general properties of the periodogram, we have

$$\begin{aligned}
\ln\Big[I_{\bar{X}(m)}(0)\Big] &\dot{\sim} \ln\left[p_{\bar{X}(m)}(0)\,\frac{\chi^2(2)}{2}\right] \\
&= \ln\Big[p_{\bar{X}(m)}(0)\Big] + \ln\left[\frac{\chi^2(2)}{2}\right].
\end{aligned} \tag{B.40}$$

Taking $\nu = 2$ and plugging into (B.2), we have

$$\mathrm{Var}\Big\{\ln\Big[I_{\bar{X}(m)}(0)\Big]\Big\} = \mathrm{Var}\left\{\ln\left[\frac{\chi^2(2)}{2}\right]\right\} = \Psi'(1), \tag{B.41}$$

and therefore the variance of the smoothed log-periodogram of the batch means at zero fre-

quency is given by

$$\mathrm{Var}\Big[\overline{\mathcal{L}}_{\bar{X}(m)}(0)\Big] = \frac{1}{a}\,\mathrm{Var}\Big\{\ln\Big[I_{\bar{X}(m)}(0)\Big]\Big\} = \frac{\Psi'(1)}{a}. \tag{B.42}$$

A similar analysis for $l = \frac{1}{2}$ yields

$$\mathrm{Var}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{1}{2}\right)\right] = \Psi'(a) \tag{B.43}$$

and

$$\mathrm{Var}\left[\overline{\mathcal{L}}_{\bar{X}(m)}\left(\frac{1}{2}\right)\right] = \frac{1}{a}\,\mathrm{Var}\left\{\ln\left[I_{\bar{X}(m)}\left(\frac{1}{2}\right)\right]\right\} = \frac{\Psi'(1)}{a}. \tag{B.44}$$

Using result (B.3), we have

$$\Psi'(a) < \frac{\Psi'(1)}{a} \quad \text{for} \quad a = 2, 3, \ldots; \tag{B.45}$$

and therefore a reduction in variance is achieved at frequencies 0 and $\frac{1}{2}$ by working with the log-smoothed-periodogram of the batch means rather than the smoothed-log-periodogram of the batch means.

B.2. Variance Computation at Frequency $\frac{l}{k}$ for $a < |l| < \frac{k}{2} - a$

From equation (3.31), we have the following expression for the variance of the log of the smoothed periodogram of the batch means at the frequency $\frac{l}{k}$ for $a < |l| < \frac{k}{2} - a$,

$$\mathrm{Var}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \approx \mathrm{Var}\left\{\ln\left[p_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]\right\} + \mathrm{Var}\left\{\ln\left[\frac{\chi^2(2A)}{2A}\right]\right\}. \tag{B.46}$$

Taking $\nu = 2A$ and substituting into (B.2), we have

$$\mathrm{Var}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] = \Psi'(A). \tag{B.47}$$

Now, to compute the variance of the smoothed log-periodogram of the batch means, we first notice that

$$\begin{aligned}
\mathrm{Var}\left[\overline{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] &= \mathrm{Var}\left\{\frac{1}{A}\sum_{u=-a}^{a}\ln\left[I_{\bar{X}(m)}\left(\frac{l+u}{k}\right)\right]\right\} \\
&= \frac{1}{A^2}\sum_{u=-a}^{a}\mathrm{Var}\left\{\ln\left[I_{\bar{X}(m)}\left(\frac{l+u}{k}\right)\right]\right\}
\end{aligned}$$

$$\approx \quad \frac{1}{A}\mathrm{Var}\left\{\ln\left[I_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]\right\}. \tag{B.48}$$

From the general properties of the periodogram, we have

$$\begin{aligned}\ln\left[I_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] &\dot{\sim} \ln\left[p_{\bar{X}(m)}\left(\frac{l}{k}\right)\frac{\chi^2(2)}{2}\right] \\ &= \ln\left[p_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] + \ln\left[\frac{\chi^2(2)}{2}\right].\end{aligned} \tag{B.49}$$

Taking $\nu = 2$ and plugging into (B.2), we have

$$\mathrm{Var}\left\{\ln\left[I_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]\right\} = \mathrm{Var}\left\{\ln\left[\frac{\chi^2(2)}{2}\right]\right\} = \Psi'(1), \tag{B.50}$$

and therefore the variance of the smoothed log-periodogram of the batch means is given by

$$\mathrm{Var}\left[\overline{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] = \frac{1}{A}\,\mathrm{Var}\left\{\ln\left[I_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]\right\} = \frac{\Psi'(1)}{A}. \tag{B.51}$$

Using result (B.3), we have

$$\Psi'(A) < \frac{\Psi'(1)}{A} \quad \text{for} \quad A = 3, 4, \ldots; \tag{B.52}$$

and therefore a reduction in variance is achieved at the frequency $\frac{l}{k}$ for $a < |l| < \frac{k}{2} - a$ by working with the log-smoothed-periodogram of the batch means rather than the smoothed-log-periodogram of the batch means.

B.3. Variance Computation at Frequency $\frac{l}{k}$ for $1 \leq |l| \leq a$

From equation (3.35), we have the following expression for the variance of the log of the smoothed periodogram of the batch means at frequency $\frac{l}{k}$ for $1 \leq |l| \leq a$,

$$\mathrm{Var}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \approx \mathrm{Var}\left\{\ln\left[p_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]\right\} + \mathrm{Var}\left\{\ln\left[\frac{\chi^2(\nu_{|l|})}{\nu_{|l|}}\right]\right\}. \tag{B.53}$$

Taking $\nu = \nu_{|l|}$ and substituting into (B.2), we have

$$\mathrm{Var}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] = \Psi'\left(\frac{\nu_{|l|}}{2}\right). \tag{B.54}$$

The analogue of (A.11) for the smoothed-log-periodogram of the batch means requires calculation of the following weighted average of the log-periodogram values $\{\ln[I_{\bar{X}(m)}(\frac{u}{m})] : u =$

$1, 2, \ldots, l+a\}$,

$$\begin{aligned}\overline{\mathcal{L}}_{\bar{X}(m)}\Big(\frac{l}{k}\Big) &= \Big(\frac{1}{2a+1}\Big)\Bigg\{\Big(\frac{2a+1}{a}\Big)\sum_{u=1}^{a-l}\ln\Big[I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\Big] \\ &\qquad + \Big(\frac{a+1}{a}\Big)\sum_{u=a-l+1}^{a}\ln\Big[I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\Big] \\ &\qquad + \sum_{u=a+1}^{l+a}\ln\Big[I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\Big]\Bigg\}. \qquad \text{(B.55)}\end{aligned}$$

It follows immediately from (B.55) that we have the following expression for the variance of $\overline{\mathcal{L}}_{\bar{X}(m)}\Big(\frac{l}{k}\Big)$,

$$\begin{aligned}\mathrm{Var}\Big[\overline{\mathcal{L}}_{\bar{X}(m)}\Big(\frac{l}{k}\Big)\Big] &= \Big(\frac{1}{2a+1}\Big)^2\Bigg\{\Big(\frac{2a+1}{a}\Big)^2\sum_{u=1}^{a-l}\mathrm{Var}\Big\{\ln\Big[I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\Big]\Big\} \\ &\qquad + \Big(\frac{a+1}{a}\Big)^2\sum_{u=a-l+1}^{a}\mathrm{Var}\Big\{\ln\Big[I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\Big]\Big\} \\ &\qquad + \sum_{u=a+1}^{l+a}\mathrm{Var}\Big\{\ln\Big[I_{\bar{X}(m)}\Big(\frac{u}{k}\Big)\Big]\Big\}\Bigg\} \\ &= \Big(\frac{1}{2a+1}\Big)^2\Bigg\{\Big(\frac{2a+1}{a}\Big)^2\sum_{u=1}^{a-l}\mathrm{Var}\Big\{\ln\Big[p_{\bar{X}(m)}\Big(\frac{l}{k}\Big)\Big]+\ln\Big[\frac{\chi^2_u(2)}{2}\Big]\Big\} \\ &\qquad + \Big(\frac{a+1}{a}\Big)^2\sum_{u=a-l+1}^{a}\mathrm{Var}\Big\{\ln\Big[p_{\bar{X}(m)}\Big(\frac{l}{k}\Big)\Big]+\ln\Big[\frac{\chi^2_u(2)}{2}\Big]\Big\} \\ &\qquad + \sum_{u=a+1}^{l+a}\mathrm{Var}\Big\{\ln\Big[p_{\bar{X}(m)}\Big(\frac{l}{k}\Big)\Big]+\ln\Big[\frac{\chi^2_u(2)}{2}\Big]\Big\}\Bigg\} \\ &= \Big(\frac{1}{2a+1}\Big)^2\Bigg[\Big(\frac{2a+1}{a}\Big)^2\sum_{u=1}^{a-l}\Psi'(1) + \Big(\frac{a+1}{a}\Big)^2\sum_{u=a-l+1}^{a}\Psi'(1) \\ &\qquad + \sum_{u=a+1}^{l+a}\Psi'(1)\Bigg] \\ &= \Big(\frac{1}{2a+1}\Big)^2\Bigg[\Big(\frac{2a+1}{a}\Big)^2(a-l)\Psi'(1) + \Big(\frac{a+1}{a}\Big)^2 l\,\Psi'(1) + l\,\Psi'(1)\Bigg]\end{aligned}$$

$$= \quad \Psi'(1)\,\frac{4a^2 - 2al + 4a - 2l + 1}{a(2a+1)^2}. \tag{B.56}$$

Combining (B.54), (B.56), and (A.14), we see that

$$\mathrm{Var}\left[\overline{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] = \frac{\Psi'(1)}{\nu^{\#}_{|l|}/2}, \tag{B.57}$$

and

$$\mathrm{Var}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] = \Psi'\left(\frac{\lfloor \nu^{\#}_{|l|} \rfloor}{2}\right) \quad \text{for} \quad 1 \le |l| \le a. \tag{B.58}$$

Since we always take $2 \le a \le 5$ in **WASSP**, we see from (A.14) that

$$\nu^{\#}_{|l|} \ge 5.26 \quad \text{for} \quad 2 \le a \le 5 \quad \text{and} \quad 1 \le |l| \le a. \tag{B.59}$$

Using result (B.16), we have for frequency $\frac{l}{k}$ where $1 \le |l| \le a$ the result,

$$\Psi'\left(\frac{\lfloor \nu^{\#}_{|l|} \rfloor}{2}\right) < \frac{\Psi'(1)}{\nu^{\#}_{|l|}/2}; \tag{B.60}$$

and therefore a reduction in variance is achieved by working with the log-smoothed-periodogram rather than the smoothed-log-periodogram.

B.4. Variance Computation at Frequency $\frac{l}{k}$ for $\frac{k}{2} - a \le |l| \le \frac{k}{2} - 1$

At frequency $\frac{l}{k}$ for $\frac{k}{2} - a \le |l| \le \frac{k}{2} - 1$, the computation of the variance for the log of the smoothed periodogram of the batch means is the same as for frequency $\frac{l}{k}$ where $1 \le |l| \le a$. That is, for $\frac{k}{2} - a \le |l| \le \frac{k}{2} - 1$, we have

$$\mathrm{Var}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] \approx \mathrm{Var}\left\{\ln\left[p_{\bar{X}(m)}\left(\frac{l}{k}\right)\right]\right\} + \mathrm{Var}\left\{\ln\left[\frac{\chi^2(\nu_{k/2-|l|})}{\nu_{k/2-|l|}}\right]\right\}. \tag{B.61}$$

Taking $\nu = \nu_{k/2-|l|}$ and substituting into (B.2), we have

$$\mathrm{Var}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] = \Psi'\left(\frac{\nu_{k/2-|l|}}{2}\right). \tag{B.62}$$

By an analysis that parallels the derivation of (B.56), we have the following expression for

the variance of $\overline{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)$ at frequency $\frac{l}{k}$ for $\frac{k}{2}-a \leq |l| \leq \frac{k}{2}-1$:

$$\begin{aligned}\mathrm{Var}\left[\overline{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] &= \left(\frac{1}{2a+1}\right)^2 \Bigg\{ \left(\frac{2a+1}{a}\right)^2 \sum_{u=k-l-a}^{\frac{k}{2}-1} \mathrm{Var}\left\{\ln\left[I_{\bar{X}(m)}\left(\frac{u}{k}\right)\right]\right\} \\ &\quad + \left(\frac{a+1}{a}\right)^2 \sum_{u=\frac{k}{2}-a}^{k-l-a-1} \mathrm{Var}\left\{\ln\left[I_{\bar{X}(m)}\left(\frac{u}{k}\right)\right]\right\} \\ &\quad + \sum_{u=l-a}^{\frac{k}{2}-(a+1)} \mathrm{Var}\left\{\ln\left[I_{\bar{X}(m)}\left(\frac{u}{k}\right)\right]\right\} \Bigg\}. \end{aligned} \tag{B.63}$$

Combining (B.62), (B.63), and (A.14), we see that

$$\mathrm{Var}\left[\overline{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] = \frac{\Psi'(1)}{\nu^{\#}_{k/2-|l|}/2}, \tag{B.64}$$

and

$$\mathrm{Var}\left[\widetilde{\mathcal{L}}_{\bar{X}(m)}\left(\frac{l}{k}\right)\right] = \Psi'\left(\frac{\lfloor \nu^{\#}_{k/2-|l|} \rfloor}{2}\right) \quad \text{for} \quad \tfrac{k}{2}-a \leq |l| \leq \tfrac{k}{2}-1. \tag{B.65}$$

Since we always take $2 \leq a \leq 5$ in $\mathcal{WASSP}$, we see from (A.14) that

$$\nu^{\#}_{k/2-|l|} \geq 5.26 \quad \text{for} \quad 2 \leq a \leq 5 \quad \text{and} \quad \tfrac{k}{2}-a \leq |l| \leq \tfrac{k}{2}-1. \tag{B.66}$$

Using result (B.16), we have for frequency $\frac{l}{k}$ where $\frac{k}{2}-a \leq |l| \leq \frac{k}{2}-1$ the result,

$$\Psi'\left(\frac{\lfloor \nu^{\#}_{k/2-|l|} \rfloor}{2}\right) < \frac{\Psi'(1)}{\nu^{\#}_{k/2-|l|}/2}; \tag{B.67}$$

and therefore a reduction in variance is achieved by working with the log-smoothed-periodogram rather than the smoothed-log-periodogram.

Appendix C

Computation of the SSVC for an "AR(1)-to-Anything" Process

The following computation of the SSVC for an "AR(1)-to-anything" process is based on [69]. Let $\{Z_j : j = 1, 2, \ldots\}$ denote a stationary AR(1) process with $N(0, 1)$ marginals and lag-one correlation ρ as specified by (4.16)–(4.17); and let $\{X_j : j = 1, 2, \ldots\}$ denote the corresponding "AR(1)-to-anything" process with marginal c.d.f. $F_X(x) \equiv \Pr[X_j \leq x]$ for all x. Then $\{X_j\}$ can be generated from the base process (4.16) according to

$$X_j = F_X^{-1}\left[\Phi(Z_j)\right], \quad j = 1, 2, \ldots. \tag{C.1}$$

We seek to calculate the SSVC for the process $\{X_j\}$,

$$\gamma_X \equiv \sum_{l=-\infty}^{+\infty} \mathrm{Cov}(X_j, X_{j+l}) = \sum_{l=-\infty}^{+\infty} \gamma_X(l). \tag{C.2}$$

Notice that

$$\begin{aligned} \gamma_X(l) &= \mathrm{Cov}\left\{F_X^{-1}\left[\Phi(Z_j)\right], F_X^{-1}\left[\Phi(Z_{j+l})\right]\right\} \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \left\{F_X^{-1}\left[\Phi(z_1)\right] - \mu_X\right\}\left\{F_X^{-1}\left[\Phi(z_2)\right] - \mu_X\right\} \varphi_2\left(z_1, z_2; \rho^{|l|}\right)\, dz_1\, dz_2, \end{aligned}$$

where

$$\varphi_2(z_1, z_2; r) \equiv \frac{\exp\left[-\frac{z_1^2 - 2r z_1 z_2 + z_2^2}{2(1-r^2)}\right]}{2\pi\sqrt{1-r^2}}$$

is the bivariate standard normal p.d.f. with correlation $r \in (-1, +1)$. To simplify the notation, we let

$$D(z_i) \equiv F_X^{-1}\left[\Phi(z_i)\right] - \mu_X \quad \text{for} \quad i = 1, 2, \tag{C.3}$$

so that we have

$$\gamma_X(l) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} D(z_1)\, D(z_2)\, \varphi_2\Big(z_1, z_2; \rho^{|l|}\Big)\, dz_1\, dz_2. \tag{C.4}$$

It follows from equation (12.6.8) of [13] that for $r \in (-1, +1)$, we have

$$\varphi_2(z_1, z_2; r) = \sum_{u=0}^{\infty} \frac{1}{u!} H_u(z_1)\, H_u(z_2)\, \varphi(z_1)\, \varphi(z_2)\, r^u, \tag{C.5}$$

where for $u = 0, 1, 2, \ldots$, the uth Hermite polynomial is defined by

$$\frac{d^u}{dz^u} e^{-z^2/2} = (-1)^u H_u(z)\, e^{-z^2/2}.$$

Inserting (C.5) into (C.4) and then inserting the result into (C.2), we have

$$\begin{aligned}
\gamma_X &= \sum_{l=-\infty}^{+\infty} \int_{-\infty}^{+\infty} \prod_{i=1}^{2} D(z_i) \left[\sum_{u=0}^{\infty} \frac{\rho^{|l|u}}{u!} \prod_{j=1}^{2} H_u(z_j)\, \varphi(z_j)\right] dz_1\, dz_2 \\
&= \sigma_X^2 + 2 \sum_{l=1}^{\infty} \sum_{u=0}^{\infty} \frac{\rho^{ul}}{u!} \prod_{j=1}^{2} \left[\int_{-\infty}^{+\infty} D(z_j)\, H_u(z_j)\, \varphi(z_j)\, dz_j\right] \\
&= \sigma_X^2 + 2 \sum_{l=1}^{\infty} \sum_{u=0}^{\infty} \frac{\rho^{ul}}{u!} \left\{\int_{-\infty}^{+\infty} \left\{F_X^{-1}\left[\Phi(z)\right] - \mu_X\right\} H_u(z)\, \varphi(z)\, dz\right\}^2.
\end{aligned} \tag{C.6}$$

If we let

$$\Xi_u \equiv \int_{-\infty}^{+\infty} \left\{F_X^{-1}\left[\Phi(z)\right] - \mu_X\right\} H_u(z)\, \varphi(z)\, dz \tag{C.7}$$

for $u = 0, 1, \ldots$, then we have

$$\Xi_0 = 0 \quad \text{and} \quad \Xi_u = \int_{-\infty}^{\infty} F_X^{-1}\left[\Phi(z)\right]\, H_u(z)\, \phi(z)\, dz \quad \text{for} \quad u = 1, 2, \ldots,$$

so that (C.7) becomes

$$\gamma_X = \sigma_X^2 + 2 \sum_{l=1}^{\infty} \sum_{u=1}^{\infty} \frac{1}{u!} \rho^{lu}\, \Xi_u^2$$

$$
\begin{aligned}
&= \sigma_X^2 + 2\sum_{u=1}^{\infty} \frac{1}{u!}\Xi_u^2 \left[\sum_{l=1}^{\infty} \rho^{lu}\right] \\
&= \sigma_X^2 + 2\sum_{u=1}^{\infty} \frac{\Xi_u^2\, \rho^u}{u!(1-\rho^u)}. \qquad \text{(C.8)}
\end{aligned}
$$

Display (C.8) suggests a numerical method for estimating γ_X. To estimate γ_X with maximum relative error ε_{rel}, compute the partial sum

$$
Q_v = \sigma_X^2 + 2\sum_{u=1}^{v} \frac{\Xi_u^2\, \rho^u}{u!(1-\rho^u)}, \qquad \text{(C.9)}
$$

where $Q_0 \equiv \sigma_X^2$ and for $u \geq 1$, we evaluate Ξ_u numerically as

$$
\Xi_u \approx \int_{-8}^{8} F_X^{-1}\left[\Phi(z)\right]\, H_u(z)\, \varphi(z)\, dz. \qquad \text{(C.10)}
$$

We stop evaluating Ξ_u via (C.10) when

$$
\left|\frac{Q_v - Q_{v-1}}{Q_{v-1}}\right| \leq \varepsilon_{\text{rel}}; \qquad \text{(C.11)}
$$

and then we deliver Q_v as the estimate of γ_X with maximum relative error ε_{rel}.

Appendix D

Computation of the Power Spectrum for the $M/M/1$ Queue Waiting Time Process

If $\{X_i : i = 1, 2, \ldots\}$ is the waiting time process for an $M/M/1$ queue in steady-state operation with arrival rate λ, service rate μ, and server utilization $\varrho = \lambda/\mu < 1$, then we have [15]

$$\mu_X = \frac{\varrho}{\mu(1-\varrho)} = \frac{\varrho^2}{\lambda(1-\varrho)}, \tag{D.1}$$

$$\sigma_X^2 = \frac{\varrho(2-\varrho)}{\mu^2(1-\varrho)^2} = \frac{\varrho^3(2-\varrho)}{\lambda^2(1-\varrho)^2}, \tag{D.2}$$

and

$$\gamma_X = \frac{(\varrho^3 - 4\varrho^2 + 5\varrho + 2)\,\varrho^3}{\lambda^2(1-\varrho)^4}. \tag{D.3}$$

We seek to calculate and plot the spectrum of $\{X_i\}$,

$$\begin{aligned} p_X(\omega) &= \sum_{l=-\infty}^{+\infty} \gamma_X(l)\cos(2\pi\omega l) \\ &= \sigma_X^2 \left\{ 1 + 2\sum_{l=1}^{\infty} \rho_X(l)\cos(2\pi\omega l) \right\}, \end{aligned} \tag{D.4}$$

where $-1/2 \leq \omega \leq 1/2$ and

$$\rho_X(l) \equiv \mathrm{Corr}(X_i, X_{i+l}) = \frac{\gamma_X(l)}{\sigma_X^2} \quad \text{for} \quad l = 1, 2, \ldots. \tag{D.5}$$

Let $q_X(\omega)$ denote the cosine transform of the autocorrelation function (D.5),

$$q_X(\omega) = 1 + 2\sum_{l=1}^{\infty} \rho_X(l) \cos(2\pi\omega l), \quad \text{for} \quad \omega \in \left[-\tfrac{1}{2}, \tfrac{1}{2}\right]. \tag{D.6}$$

From equation (34) of [15] as corrected on p. 117 of [56], we have

$$\rho_X(l) = \frac{(1-\varrho)^3(1+\varrho)}{2\pi\,\varrho^3(2-\varrho)} \int_0^r t^{|l|}\, \frac{t^{3/2}(r-t)^{1/2}}{(1-t)^3}\, dt \quad \text{for} \quad l = \pm 1, \pm 2, \ldots, \tag{D.7}$$

and hence

$$q_X(\omega) = 1 + \frac{(1-\varrho)^3(1+\varrho)}{\pi\,\varrho^3(2-\varrho)} \int_0^r \left[\sum_{l=1}^{\infty} t^l \cos(2\pi\omega l)\right] \frac{t^{3/2}(r-t)^{1/2}}{(1-t)^3}\, dt, \tag{D.8}$$

where

$$r \equiv \frac{4\varrho}{(1+\varrho)^2} \in (0,1) \quad \text{since} \quad \varrho \in (0,1). \tag{D.9}$$

It follows from (D.9) that $|t| < 1$ in the integrand of (D.8); and from formula 1.447(2) on p. 39 of [23], we deduce that

$$\sum_{l=1}^{\infty} t^l \cos(2\pi\omega l) = \frac{t\cos(2\pi\omega) - t^2}{1 - 2t\cos(2\pi\omega) + t^2} = \frac{t[\cos(2\pi\omega) - t]}{1 - 2t\cos(2\pi\omega) + t^2}. \tag{D.10}$$

Inserting (D.10) into (D.8), we have

$$q_X(\omega) = 1 + \frac{(1-\varrho)^3(1+\varrho)}{\pi\,\varrho^3(2-\varrho)} \int_0^r \left[\frac{\cos(2\pi\omega) - t}{1 - 2t\cos(2\pi\omega) + t^2}\right] \frac{t^{5/2}(r-t)^{1/2}}{(1-t)^3}\, dt \tag{D.11}$$

from which we can recover a formula for the power spectrum,

$$p_X(\omega) = \sigma_X^2\, q_X(\omega), \quad -\tfrac{1}{2} \leq \omega \leq \tfrac{1}{2}. \tag{D.12}$$

Appendix E

WASSP User's Manual

E.1. Installation of WASSP

We coded the WASSP algorithm in MATLAB [24] and it can be invoked using version 5.3 or 6.1 of MATLAB in either the Unix or Windows environment. The following steps must be completed to properly install WASSP:

(a) Download the file `wassp.zip` from the website

`<ftp.ncsu.edu/pub/eos/pub/jwilson/wassp.zip>`.

The following MATLAB functions are contained in the file `wassp.zip`: `wassp.m`, `batch3.m`, `wavest.m`, `smooth2.m`, `indep2.m`, and `wilks.f`. The MATLAB code for each of these functions is given at the end of this appendix. The file `wassp.zip` also contains the MATLAB data set `L.mat`.

(b) Download WaveLab from the website

`<http://www-stat.stanford.edu/~wavelab>`.

Complete instructions for installing WaveLab are given at the above website. WaveLab is a software package designed to run in conjunction with MATLAB, and WaveLab provides the functions necessary to compute the discrete wavelet transform and the inverse discrete wavelet transform.

(c) Compile the Fortran function `wilks.f` in MATLAB. This can be done by typing the command

```
mex wilks.f
```

at the MATLAB command prompt. A compiled version of `wilks.f` will be created. The resulting file will have a platform-dependent extension, namely `.mexsol` for sol2 or SunOS 5.x and `.dll` for Windows.

Once steps (a)–(c) above have been completed, *WASSP* is ready to use.

E.2. Invoking *WASSP* in MATLAB

Before *WASSP* can be invoked, the data must be loaded into MATLAB using the `load` command. The data should either be a MATLAB data file (`.mat` file) or an ASCII data file with one observation per line. *WASSP* can then be invoked using the MATLAB function call

```
[cil,ciu,extradata]=wassp(data,PrecReq,RP,hrstar,beta,A)
```

at the command prompt. Table E.1 lists the arguments required by the `wassp` function, as well as a description of their acceptable values. Table E.2 lists the arguments returned by the `wassp` function.

In Figure E.1, a screen shot of the MATLAB command window is shown. On the first command line, the MATLAB data set `L.mat` is loaded. On the second command line, the function `wassp` is invoked using the data set `L`. The parameter `PrecReq` is set to `'F'`, indicating that no precision requirement is specified. The parameters `RP` and `hrstar` are set to the values `'F'` and 0.30, respectively. However, since the parameter `PrecReq` is set to `'F'`, the values of `RP` and `hrstar` are irrelevant and will have no effect on the delivered confidence interval. Finally, the parameter `beta` is set to 0.10, implying a 90% confidence interval is desired and the smoothing parameter `A` is set to the default value of 7.

In Figure E.2 another screen shot of the MATLAB command window is shown. In this example, the data set `L.mat` has already been loaded, and the `wassp` function is called with a relative precision requirement of ±15%. In order to construct the requested 90% confidence interval that satisfies a relative precision requirement of ±15%, the user would have to supply an additional 66,128 observations. Once the additional observations have been collected, the new appended data set must be loaded into MATLAB and the function `wassp` can then be called again.

Table E.1: Input arguments required to run the function `wassp` in MATLAB.

Argument	Definition
`data`	The name of the data set that has already been loaded into MATLAB.
`PrecReq`	The argument `PrecReq` is set to `'T'` if an absolute or relative precision requirement is specified; and `PrecReq` is set to `'F'` if no precision requirement is desired.
`RP`	The argument `RP` is set to `'T'` if a relative precision requirement is specified; and `RP` is set to `'F'` if an absolute precision requirement is specified. If the value of `PrecReq` is `'F'`, then `RP` may have either the value `'T'` or the value `'F'`.
`hrstar`	The argument `hrstar` is defined by the precision requirement (3.5) specifying the final confidence interval half-length. For an absolute precision requirement, `hrstar` should be set equal to h^*, the maximum acceptable confidence-interval half-length as in (3.5). For a relative precision requirement, `hrstar` should be set equal to r^*, the maximum acceptable fraction of the magnitude of the confidence-interval midpoint as in (3.5).
`beta`	For the final confidence interval (3.3) delivered by *WASSP*, the desired coverage probability is given by 1-`beta`, where $0 <$ `beta` < 1.
`A`	The argument `A` is the the value of the smoothing parameter A in equation (3.22), where `A`=5, 7, 9, or 11.

Table E.2: Output arguments returned by the function `wassp` in MATLAB.

Argument	Definition
`cil`	The lower limit of the final confidence interval (3.3). If more data is required, then `cil=[]`.
`ciu`	The upper limit of the final confidence interval (3.3). If more data is required, then `ciu=[]`.
`extradata`	The number of additional observations required to compute the desired confidence interval.

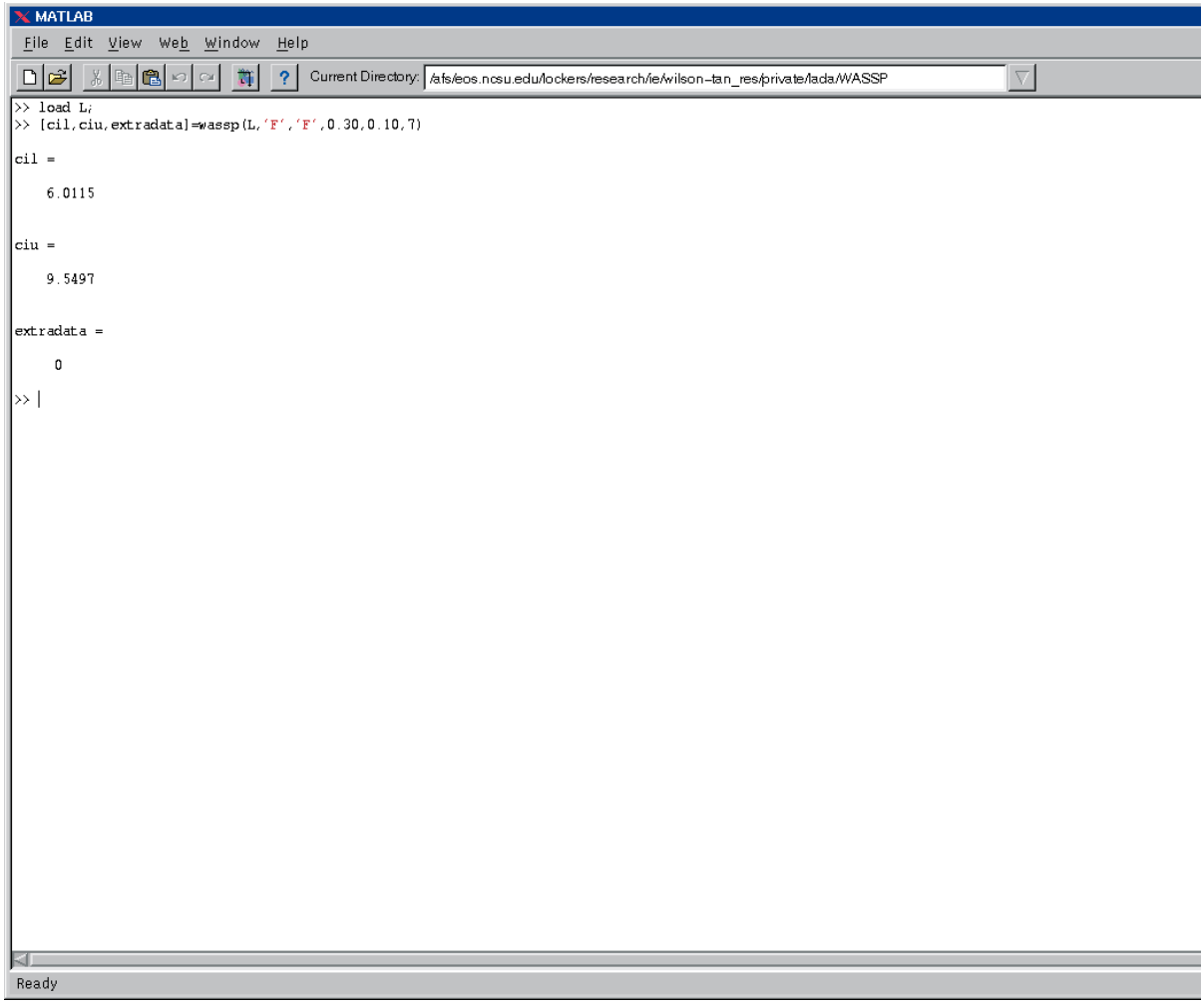

Figure E.1: Display of the MATLAB command window showing how to invoke the function `wassp` with no precision requirement. In this example, `wassp` returned the upper and lower limits of the requested confidence interval.

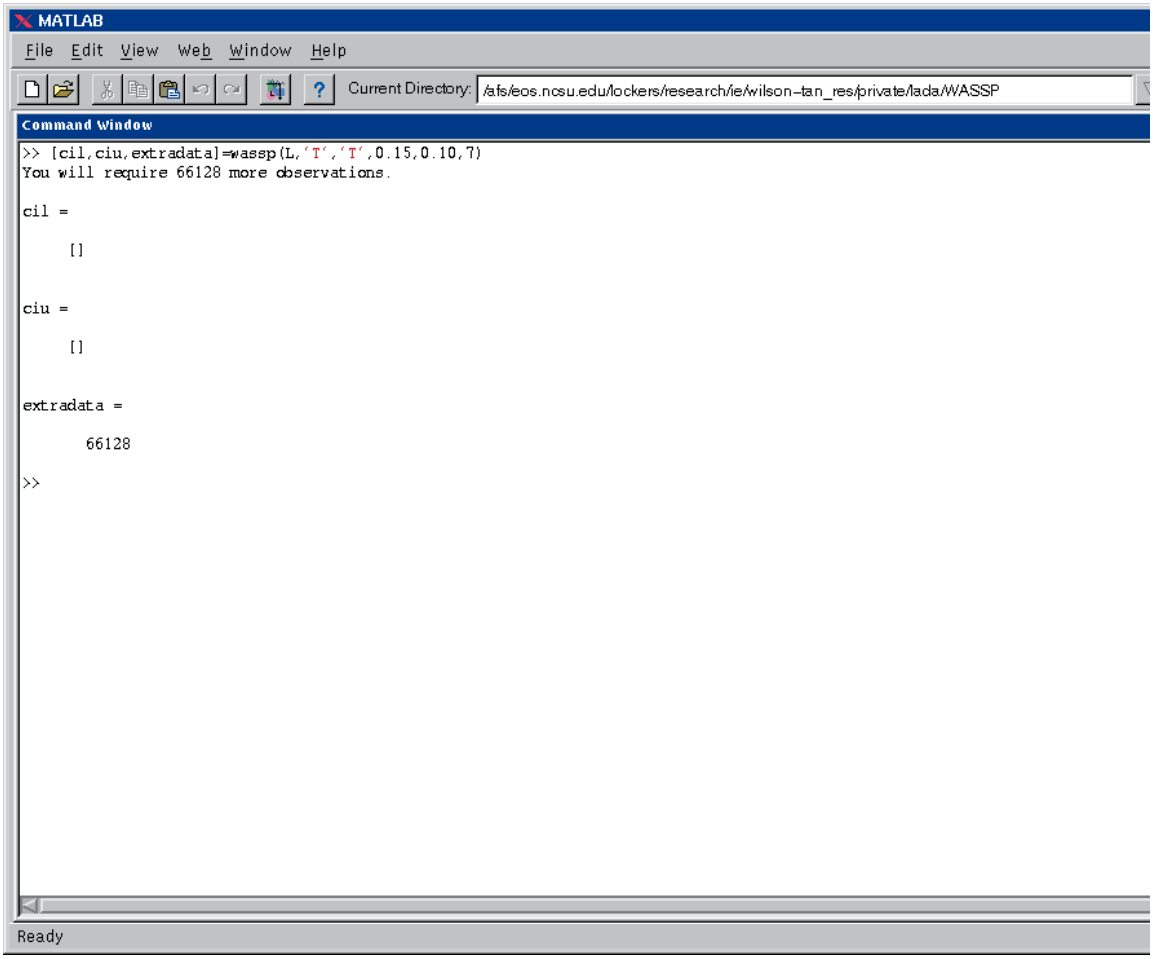

Figure E.2: Display of the MATLAB command window showing how to invoke the function `wassp` with a specified precision requirement. In this example, `wassp` returned the amount of extra data required to compute the requested confidence interval.

E.3. Listing of MATLAB Code for *Wassp*

```
% This program computes a (1-beta)100% confidence interval given a specified
% precision requirement hrstar using the WASSP algorithm.  If RP='T', then a rel
ative precision requirement
% has been specified.
% The parameter A is the value of the smoothing parameter, where A=5,7,9, or 11.

function [cil,ciu,extradata]=wassp(data,PrecReq,RP,hrstar,beta,A)

% Function requires six input arguments
if nargin < 6
   error('Not enough input arguments.');
   return
end

if (beta >= 1) | (beta <= 0)
disp(['The parameter beta must be between 0 and 1'])
return
end

if ~((A==5)|(A==7)|(A==9)|(A==11))
disp(['The parameter A must be equal to 5,7,9,or 11'])
return
end

if ~((PrecReq=='T')|(PrecReq=='F'))
disp(['The parameter PrecReq must be either T or F'])
return
end

if ~((RP=='T')|(RP=='F'))
disp(['The parameter RP must be either T or F'])
return
end

if ((length(data)) < 4096)
disp(['WASSP requires a minimum of 4096 observations'])
return
end

ind='F';
normpass='F';
batchsize=16;
numbatches=256;
spacerlen=1;

% alpha value for normality test
tol=0.05;
% inc is the number of iterations of the normality test completed
inc=1;
L=data;
datamspacer=[];
tempdatalength=numbatches*batchsize;

if RP=='T'
        rstar=hrstar;
end

[datamspacer,B,moredata,batchsize,datasize,ind,normpass,numbatches,spacerlen,ext
radata,tol,inc,tempdatalength]=batch3(datamspacer,L,ind,normpass,batchsize,numba
tches,spacerlen,tol,inc,tempdatalength);

         if moredata=='T'

               % generate 'extradata' more observations
               disp(['You will require ', num2str(extradata), 'more observations
'])
               cil=[];
               ciu=[];
               return
```



```
         end

         wavecoef=wavest(B,A);
         len=length(wavecoef);
         mid=exp((wavecoef((len/2))));
         sd=sqrt(mid/(length(B)));
         xbar=mean(B);

         t1=tinv(1-beta/2,A-1);
         hw=t1*sd;
         cil=xbar-hw;
         ciu=xbar+hw;
         extradata=0;
         k=length(B);
         maxdatasize=datasize;

 if PrecReq =='T'

        if RP=='T'
                hrstar=rstar*abs(xbar);
         end

          while (hw > hrstar)

                kplus=ceil((hw/hrstar)^2*numbatches);
                kprime=floor(log2(kplus));
                numbatches=min(2^kprime,4096);
                oldbatchsize=batchsize;
                batchsize=ceil((kplus/numbatches)*oldbatchsize);

                if ((numbatches*batchsize) > length(datamspacer))
                        %generate more data
                        extradata=(numbatches*batchsize)-length(datamspacer);
                        disp(['You will require ' num2str(extradata) ' more obs
ervations.'])

                        ciu=[];
                        cil=[];

                        return
                end
                %rebatch the data
                start=1;
                for j=1:numbatches
                        i=start:1:(start+batchsize-1);
                        B(j)=sum(datamspacer(i))./batchsize;
                        start=start+batchsize;
                end

                wavecoef=wavest(B,A);
                len=length(wavecoef);
                mid=exp((wavecoef((len/2))));
                sd=sqrt(mid/(length(B)));
                xbar=mean(B);

                t1=tinv(1-beta/2,A-1);
                hw=t1*sd;
                cil=xbar-hw;
                ciu=xbar+hw;
                extradata=0;

                if RP=='T'
                        hrstar=rstar*abs(xbar);
                end

        end
end
```

```
% This program batches the observations

% Start initially with  a sample of size 4096
% This gives k=256 batches of size m=16 initially

% The dataset passes into this program should be of size 4096 at least.

function [datamspacer,B,moredata,batchsize,datasize,ind,normpass,numbatches,spac
erlen,extradata,tol,inc,tempdatalength]=batch3(datamspacer,data,ind,normpass,bat
chsize,numbatches,spacerlen,tol,inc,tempdatalength)

% start initially by taking only the first 4096 observations
%len=4096;
m=batchsize;

% construct the initial 256 batch means
k=256;
if ind=='F'
        start=1;
        for j=1:k
                i=start:1:(start+m-1);
                tempB(j)=sum(data(i))./m;
                start=start+m;
        end
end

if ind=='T'
        start=spacerlen+1;
        for j=1:numbatches
                i=start:1:(start+m-1);
                indB(j)=sum(data(i))./m;
                start=start+spacerlen;
        end
        length(indB);
end



%'fact'is the factor by which we increase the batchsize m
fact=sqrt(2);
while (ind=='F')
     pass=indep2(tempB);

     spacer=1;
     if pass=='T'
             ind='T';
             indB=tempB;
     else
             % if initial test for independence fails, then throw out every othe
r
             % batch mean, giving a total of 128 batch means.
             while ((pass=='F') & (spacer ~= 10))
                     spacer=spacer+1;
                     i=spacer:spacer:length(tempB);
                     temp2B=tempB(i);
                     pass=indep2(temp2B);
             end
             if ((spacer <= 10) & (pass=='T'))
                     indB=temp2B;
                     spacer;

                     %exit loop and test for normality
                     ind='T';
             else
                     % reset batch size
                     m=floor(fact*m);

                     i=1:1:(m*k);
                     if length(data)<(m*k)
                             moredata='T';
                             B=[];
                             batchsize=16;
                             datasize=length(data);
                             ind='F';
                             normpass='F';
                             numbatches=256;
                             spacerlen=1;
                             extradata=(m*k)-length(data);

                             return
                     else
                     samp=data(i);

                     %recompute the batch means
                     start=1;
                     for j=1:k
                        i=start:1:(start+m-1);
                        tempB(j)=sum(samp(i))./m;
                        start=start+m;
                     end

                     end
             end



     end
     spacerlen=(spacer-1)*m;
end

% test the independent batch means, stored in indB, for normality

if normpass=='F'

[xbar,var,w,pw] = wilks(indB,length(indB));
clear mex

% want to keep the # of observations between batches
% and the number of batches constant.

tempdatalength=(m*numbatches)+(spacerlen*numbatches);

numbatches=length(indB);


while (pw < tol)
     m=floor(fact*m);

     inc=inc+1;
     tol=0.05*exp(-0.184206*(inc-1)^2);

     i=1:1:((m*numbatches)+(spacerlen*(numbatches)));
     if length(data)<((m*numbatches)+(spacerlen*(numbatches)))
        pw=1;
        moredata='T';
        B=[];
        batchsize=m;
        datasize=length(data);
        ind='T';
        normpass='F';
        extradata=((m*numbatches)+(spacerlen*(numbatches)))-length(data);
        return
     else
     samp=data(i);
     tempdatalength=length(samp);
     start=spacerlen+1;
     for j=1:numbatches
        i=start:1:(start+m-1);
```

```
        indB(j)=sum(samp(i))./m;
        start=start+spacerlen;
     end
     [xbar,var,w,pw] = wilks(indB,length(indB));
     clear mex

     end
end
tempdatalength=(m*numbatches)+(spacerlen*numbatches);

normpass='T';
% rebatch so don't throw away any observations except the first spacerlength

% need to make sure we have a final number of batches that is a power of 2

dataavail=tempdatalength-spacerlen ;
k=floor(log2(dataavail/m));
numbatches=2^k;


minbatches=32;
if numbatches < minbatches
        numbatches=minbatches;
end

end



if length(data)<((numbatches*m)+spacerlen)
        moredata='T';
        B=[];
        batchsize=m;
        datasize=length(data);
        ind='T';
        normpass='T';
        extradata=((numbatches*m)+spacerlen)-length(data);
        return
else
        start=1;

        %throw out the first spacer
        datamspacer=data((spacerlen+1):end);

        length(datamspacer);

        for j=1:numbatches
        i=start:1:(start+m-1);
        finalB(j)=sum(datamspacer(i))./m;
        start=start+m;
        end

        length(finalB);
        B=finalB;
        moredata='F';
        batchsize=m;
        datasize=max((batchsize*length(finalB)+spacerlen),tempdatalength);
        extradata=0;
end
```

```
% This program computes a wavelet estimate of the log of the smoothed periodogra
m.
% WaveLab is required to do the wavelet computations.

function wavecoef=wavest(data,A)


%The following constants are values of the Psi function

PsiA=[0 1.5061 1.8728 2.1406 2.3518];
Psia=[0 0.4228 0.9228 1.2561 1.5061];
dof=[0 0 0 0 0;5/2 7/2 0 0 0;7/2 4 11/2 0 0;9/2 5 6 15/2 0;11/2 6 7 8 19/2];
Psi=[0 0 0 0 0;0.7032 1.1032 0 0 0;1.1032 1.2561 1.6111 0 0;1.3889 1.5061 1.7061
 1.9468 0;1.6111 1.7061 1.8728 2.0156 2.1977];

% Compute the FFT of the data

Y=fft(data);

N = length(Y)

a=(A-1)/2;
% remove the first component of Y which is the sum of the data
Y(1) = [];

%Compute the periodogram
powertemp = (abs(Y(1:N/2)).^2)/N;

%powertemp is the vector containing the values of the periodogram, not including
%zero frequency.

%Compute the value of the periodogram at 0 frequency by averaging the first 3
%values in the vector powertemp.

power0=(1/a)*(sum(powertemp(1:a)));
powertemp(end)=(1/a)*(sum(powertemp(length(powertemp)-a:length(powertemp)-1)));

%add the value at 0 frequency to the front of the vector containing the
%values of the periodogram.

power=cat(2,power0,powertemp);

%The following commands are needed to construct a plot of the periodogram
n=length(power)-1;
for j=1:n
    freq2temp(j)=(j/n)*1/2;
end
freqzero=0;
freq2=cat(2,freqzero,freq2temp);

power1=cat(2,powertemp(a:-1:1),power);
power2=cat(2,power1,powertemp(length(powertemp)-1:-1:length(powertemp)-a));

% smooth the periodogram by taking a moving average
power_ma=smooth2(power2,A);

Ylog=log(power_ma);

clear powertemp;
clear power;
clear power1;
clear power2;
clear power_ma;


%include the bias correction

Ylog(1)=Ylog(1)-[Psia(a)-log(a)];
Ylog(end)=Ylog(end)-[Psia(a)-log(a)];
```



```
j=1:1:a;
Ylog(1+j)=Ylog(1+j)-[Psi(a,j)-log(dof(a,j))];

i=(a+2):(length(Ylog)-a-1);
Ylog(i)=Ylog(i)-[PsiA(a)-log(A)];

i=(length(Ylog)-1):-1:(length(Ylog)-a);
Ylog(i)=Ylog(i)-[Psi(a,j)-log(dof(a,j))];

Ylog=cat(2,fliplr(Ylog(2:end)),Ylog);
Ylog(1)=[];

len=length(Ylog);

% M is the number of levels used in the wavelet decomposition

M=floor(log2(sqrt(len)));

[n2,J]=dyadlength(Ylog);

% l is the coarsest level of coefficients
l=J-M;

qmf=MakeONFilter('Symmlet',8);

% C is the vector of scaling and wavelet coefficients
C=FWT_PO(Ylog,l,qmf);

% implementation of Gao's thresholding algorithm
% M is the number of levels of coefficients
% Threshold from FINEST level to coarsest level, not including the scaling coeff
icients.
flipC=fliplr(C);
start=1;
test(1)=(pi/(sqrt(3)))*sqrt(log(len));
for j=1:M
     % s is the number of coefficients at each level
     s=2^(J-j);
     test(2)=(2^((-j+1)/4))*(log(len));
     t=max(test);
     for k=start:(s+start -1)
          if (abs(flipC(k))-t) > 0
               flipC(k)=(sign(flipC(k)))*(abs(flipC(k))-t);
          else
               flipC(k)=0;
          end
     end
     start=start+s;
end

Ct=fliplr(flipC);
 a1=IWT_PO(Ct,l,qmf);

%the commands below are necessary if a plot of the wavelet estimate is desired
temp=fliplr(freq2temp);
temp=(-1)*temp;
temp(1)=[];
freq4=cat(2,temp,freq2);
subplot(2,1,1)
plot(freq4,Ylog)
title('Log of the Smoothed Periodogram')
subplot(2,1,2)
plot(freq4,a1)
title('Wavelet Estimate')

wavecoef=a1;
```

```
% von Neumann test for independence
function pass=indep2(Z)


L3=Z;

% k is the sample size used in the von Neumann test for independence

k=length(L3);


% Compute the mean-square successive difference

d=0;
for j=1:(k-1)
     d=d+(L3(j)-L3(j+1))^2 ;
end


% Compute the von Neumann test statistic

m=mean(L3);

v=0;
for j=1:k
     v=v+(L3(j)-m)^2;
end

C=1-(d/(2*v));

%take alpha=0.2 for the 2-sided test
z=0.9;

p=norminv(z,0,1);
test=p*sqrt((k-2)/(k^2-1));


if abs(C) > test
        pass='F';
else
        pass='T';
end
```

```
% This program computes a moving average of width nperiod for the data ts.

function tsmav=smooth2(ts,nperiod)

a=(nperiod-1)/2;
n=length(ts);

for j=(a+1):n-a
%Determine the window length
    x(1)=j-1;
    x(2)=floor(nperiod/2);
    x(3)=n-j;
    w=min(x);
    tsmav(j)=1/((2*w)+1)*sum(ts(j-w:j+w));
end
tsmav(1:a)=[];
```

```
c This program computes the Shapiro-Wilk test statistic and the P-value associat
ed with
c the test statistic value.  This program is designed to be called from MatLab
c ***************************************************************
c ---------------------------------------------------------------
c                    The gateway routine
c ---------------------------------------------------------------
c ***************************************************************
c   This program must first be compiled.  At the Matlab prompt, type
c   mex wilks.f
c   This will create the file wilks.mexsol (if using a Sun workstation).
c   wilks.mexsol is the compiled version of this program.
c
c   This function will be called in MATLAB as:
c
c   [xbar,var,w,pw] = wilks(x,n);
c
c   where x is the dataset and n is the size of x
c
c ---------------------------------------------------------------

      subroutine mexFunction(nlhs, plhs, nrhs, prhs)

      implicit double precision (a-h,o-z)

c ---------------------------------------------------------------
c  MATLAB uses double precision -- the fortran code used in
c  sub wilks needs other data types. The variables brought into
c  this gateway sub are converted before calling wilks and
c  then upon return are converted back to double before
c  sending the outputs back to MATLAB
c
c  The fortran needs these data types:
c     x        real array
c     n        integer
c     xbar     double
c     var      real
c     w        real
c     pw       real
c ---------------------------------------------------------------
c

      dimension dx(2100)
      real x(2100)
      integer nlhs, nrhs
      integer plhs(*), prhs(*)
      integer mxGetPr, mxCreateFull
      integer n
      double precision xbar,dvar,dn,dw,dpw
      real var, w, pw

c      print*,'           '
c      print*,'*** inside mex routine ***'

c     MATLAB sees an "m x n" matrix  (m-rows, n-cols)
c     mxGetN gets the number of cols
c     mxGetM gets the number of rows


      mysize = 2100

c  Create the output array.
      plhs(1) = mxCreateFull(1,1,0)
      plhs(2) = mxCreateFull(1,1,0)
      plhs(3) = mxCreateFull(1,1,0)
      plhs(4) = mxCreateFull(1,1,0)

c  ----------Load the data into Fortran arrays------------
```



```
c      print*,'about to do mxCopyPtr functions'

c  get the x vector
c  note: mxCopyPtrToReal8 is really a double precision function!
      call mxCopyPtrToReal8(mxGetPr(prhs(1)), dx, mysize)

c  get the size of the vector
      call mxCopyPtrToReal8(mxGetPr(prhs(2)),dn,1)
      n = INT(dn)

c      print*, '*** finished mxCopyPtr functions ***'
c      print*,' '

      do 10 i = 1, n
         x(i) = dx(i)
10    continue




c     Call the fortran computational subroutine
      call myswtest(x,n,xbar,var,w,pw)

c      print*, '                                  '
c      print*, '  back from fortran  '

c      print*, 'xbar = ', xbar
c      print*, 'var = ', var
c      print*, 'w = ', w
c      print*, 'pw = ', pw


c  Now send the returns back to MATLAB
c
c   This will be called in MATLAB as:
c
c   [xbar,var,w,pw] = wilks(x,n);
c
c

c   --- FOR I/O TESTING ONLY ---
c      xbar = 1.25
c      var = 1.30
c      w = 1.35
c      pw = 1.40
c  ------------------------------
      dvar = var
      dw = w
      dpw = pw


c     Load the outputs into a MATLAB array.
c      print*,'about copy gateway to matlab'
      call mxCopyReal8ToPtr(xbar,mxGetPr(plhs(1)),1)
      call mxCopyReal8ToPtr(dvar,mxGetPr(plhs(2)),1)
      call mxCopyReal8ToPtr(dw,mxGetPr(plhs(3)),1)
      call mxCopyReal8ToPtr(dpw,mxGetPr(plhs(4)),1)

      return
      end


c     Bruce Schmeiser and Jim Wilson       July 1988
c     Exercise 7:  Building simulation metamodels

      subroutine myswtest(x,n,xbar,var,w,pw)
      dimension x(n)
      integer n
```

```
      double precision xbar
      real var, w, pw
      dimension a(1500), icol(2)
c      dimension xdata(2000,2), range(4)
c      dimension x(2000)
c      character*40 ititle,ixlabl,iylabl
c      character*1 ichar,iplus
c      character*10 filnam
c      character*80 objfmt
      character symbol*2, q*1, c*1, r*1
c      character iopt
      double precision ssqx, diff, xn
      data ixlabl/'   i-th order statistic x(i)'/
      data iylabl/' std nor quantile '/
      data iplus/'+'/
      data q/'q'/, c/'c'/, r/'r'/
      data nxdim/2000/, idat/1/, icdim/2/, ixrdim/2000/
      data symbol/'*'/
      data eps/.01/, icol/1,2/

c      print *,' '
c      print *,' '
c      print *,' '
c      print *,' '
c      print *,' '
c      print *,'       ******************************************'
c      print *,'       *                                        *'
c      print *,'       * Extended Shapiro-Wilk Test for Normality *'
c      print *,'       *                                        *'
c      print *,'       ******************************************'
c      print *,' '
c      print *,' '
c      print *,' '
c      print *,' '
c10    print *,' Enter the filename for the data set to be examined'
c      write (6,12)
c12    format (' >'$)
c      read (*,14) filnam
c14    format (a10)
c      call finit(idat,filnam)
c
c*** read in the data
c
c
c*** Number of observations
c
c      read (idat,*,end=999) n
      if (n .gt. nxdim) then
         print *,' Maximum sample sample size exceeded'
c         stop
      end if
c
c**** Dataset title
c
c      read (idat,'(A)',end=999) ititle
c
c**** Dataset format
c
c      read (idat,'(A)',end=999) objfmt
c
c**** Data values
c
20    if (n .eq. 0) then
         print *,' Data set empty'
c         stop
      end if
c      read (idat,objfmt,err=999,end=999) (x(i),i=1,n)
c
c**** Sort the data in ascending order
```



```
c
      call vsrta(x,n)
c
c*** perform the shapiro-wilk test for normality
c
      n2 = n/2
c   double-precision computation of sum of squared deviations
c   about the sample mean
c
      xbar = 0.0
      ssqx = 0.0
      do 200 i = 1, n
      xbar = xbar + dble(x(i))
200   continue
      xn = n
      xbar = xbar/xn
      do 210 i=1,n
      diff = dble(x(i)) - xbar
      ssqx = ssqx + diff*diff
210   continue
c
c   convert the final result to single precision
c
      ssq = sngl( ssqx )
c
c   set up the a-coefficients for the shapiro-wilk test
c
      call wcoef(a,n,n2,eps,ifault)
      if (ifault .ne. 0) then
         print *,' Fault indication ',ifault,' in computing a(.) '
c         stop
      end if
c
c   perform extended shapiro-wilk test
c
      call wext(x,n,ssq,a,n2,eps,w,pw,ifault)
      if (ifault .ne. 0) then
         print *,' Fault indication ',ifault,' in normality test'
c         stop
      end if
c      print *, ' '
c      print *, ' '
c      print *, ' '
c      write (6,'(A)') ititle
c      print *, ' '
c      print*, ' Sample mean = ',sngl(xbar)
      var = ssq/float(n-1)
c      print *, ' '
c      print*, ' Sample variance = ',var
c      print *, ' '
c      print *, ' '
c      print *, ' The extended Shapiro-Wilk test for normality:'
c      print *, ' '
c      print*, ' The computed test statistic W = ',w
c      print*, '    has significance probability ',pw
c      print *,' '
c      print *,' Enter c to continue, q to quit '
c      write (6,12)
c      read (5,'(a)') iopt
c      if (iopt .eq. q) then
c         go to 999
c      endif
c
c***  now generate the normal probability plot for this sample
c
c
c*** set up the y-values corresponding to the n(0,1) distribution
c
c      rn= n
```

```
c      npp = min0(20,n)
c      ispac1 = n/npp
c      ispac2 = ispac1 + 1
c      igap1 = n - npp*ispac1
c      igap2 = npp*ispac2 - n
c      if (igap1 .le. igap2) then
c         ispac = ispac1
c      else
c         ispac = ispac2
c      endif
c      ihspac = ispac/2
c      do 30 ii = 1, npp
c      i = ihspac + ((ii-1)*ispac)
c      if (i .lt. 1) then
c         i = 1
c      elseif (i .gt. n) then
c         i = n
c      endif
c      ri = i
c      qt = (ri - 0.5)/rn
c      xdata(ii,1) = x(i)
c      call mdnris(qt,xdata(ii,2),ier)
c      if (ier .gt. 0) then
c         print *, ' Error number ',ier,' in inverse normal cdf'
c         stop
c      end if
c30    continue
c
c*** generate the normal probability plot
c
c      range(1) = xdata(1,1)
c      range(2) = xdata(npp,1)
c      range(3) = xdata(1,2)
c      range(4) = xdata(npp,2)
c      icol(1) = 1
c      icol(2) = 2
c      call sctp(npp,icdim,xdata,ixrdim,icol,range,symbol,
c     $          ixlabl,iylabl,ititle)
c      print *, ' Enter r to repeat, q to quit'
c      write (6,12)
c      read (5,'(a)') iopt
c      if (iopt .eq. r) then
c         go to 10
c      endif
c  999   stop


       end


       subroutine wext(x,n,ssq,a,n2,eps,w,pw,ifault)
c
c  algorithm as 181 applied statistics (1982) volume 31, no. 2
c
c  calculates shapiro-wilk w statistic and its significance level
c
      real x(n),a(n2),lamda,wa(3),wb(4),wc(4),wd(6),we(6),wf(7),
     $  c1(5,3),c2(5,3),c(5),unl(3),unh(3)
      integer nc1(3),nc2(3)
      logical upper
      data wa(1),wa(2),wa(3)
     $  /0.118898, 0.133414, 0.327907/,
     $   wb(1),wb(2),wb(3),wb(4)
     $  /-0.37542, -0.492145, -1.124332, -0.199422/,
     $   wc(1),wc(2),wc(3),wc(4)
     $  /-3.15805, 0.729399, 3.01855, 1.558776/,
     $   wd(1),wd(2),wd(3),wd(4),wd(5),wd(6)
     $  /0.480385, 0.318828, 0.0, -0.0241665, 0.00879701, 0.002989646/,
     $  we(1),we(2),we(3),we(4),we(5),we(6)
```



```
     $  /-1.91487, -1.37888, -0.04183209, 0.1066339, -0.03513666,
     $   -0.01504614/,
     $   wf(1),wf(2),wf(3),wf(4),wf(5),wf(6),wf(7)
     $  /-3.73538, -1.015807, -0.331885, 0.1773538, -0.01638782,
     $   -0.03215018, 0.003852646/
      data c1(1,1), c1(2,1), c1(3,1), c1(4,1), c1(5,1),
     $     c1(1,2), c1(2,2), c1(3,2), c1(4,2), c1(5,2),
     $     c1(1,3), c1(2,3), c1(3,3), c1(4,3), c1(5,3) /
     $     -1.26233, 1.87969, 0.0649583, -0.0475604, -0.0139682,
     $     -2.28135, 2.26186, 0.0, 0.0, -0.00865763,
     $     -3.30623, 2.76287, -0.83484, 1.20857, -0.507590/
      data c2(1,1), c2(2,1), c2(3,1), c2(4,1), c2(5,1),
     $     c2(1,2), c2(2,2), c2(3,2), c2(4,2), c2(5,2),
     $     c2(1,3), c2(2,3), c2(3,3), c2(4,3), c2(5,3) /
     $     -0.287696, 1.78953, -0.180114, 0.0, 0.0,
     $     -1.63638, 5.60924, -3.63738, 1.08439, 0.0,
     $     -5.991908, 21.04575, -24.58061, 13.78661, -2.835295/
      data unl(1), unl(2), unl(3)/-3.8, -3.0, -1.0/,
     $     unh(1), unh(2), unh(3)/8.6, 5.8, 5.4/
      data nc1(1), nc1(2), nc1(3)/5,5,5/,
     $     nc2(1), nc2(2), nc2(3)/3, 4, 5/
      data pi6/1.90985932/, stqr/1.04719755/, upper/.true./,
     $     zero/0.0/, tqr/0.75/, one/1.0/, onept4/1.4/, three/3.0/,
     $     five/5.0/
      ifault = 1
      pw = one
      w = one
      if (n.le. 2) return
      ifault = 3
      if (n/2 .ne. n2) return
      ifault = 2
      if (n .gt. 2000) return
c
c       calculate w
c
      ifault = 0
      w = zero
      an = n
      i = n
      do 10 j=1,n2
      w = w + a(j)*(x(i) - x(j))
      i = i - 1
10    continue
      w = w*w/ssq
      if (w .lt. one) go to 20
      w = one
      return
c
c      get significance level of w
c
20    if (n .le.6) go to 100
c
c       n between 7 and 2000 ... transform w to y, get mean and standard
c       deviation, standardize, and get significance level
c
      if (n .gt. 20) go to 30
      al = alog(an) - three
      lamda = poly(wa,3,al)
      ybar = exp(poly(wb,4,al))
      sdy = exp(poly(wc,4,al))
      go to 40
30    al = alog(an) - five
      lamda = poly(wd,6,al)
      ybar = exp(poly(we,6,al))
      sdy = exp(poly(wf,7,al))
40    y = (one-w)**lamda
      z = (y-ybar)/sdy
      pw = alnorm(z,upper)
      return
```

```
c
c
c      deal with n less than 7 (exact significance level for n = 3)
c
100   if (w .le. eps) go to 160
      ww = w
      if (n .le. 3) go to 150
      un = alog((w-eps)/(one-w))
      n3 = n - 3
      if (un .lt. unl(n3)) go to 160
      if (un .ge. onept4) go to 120
      nc = nc1(n3)
      do 110 i = 1, nc
110   c(i) = c1(i,n3)
      eu3 = exp(poly(c,nc,un))
      go to 140
120   if (un .gt. unh(n3)) return
      nc = nc2(n3)
      do 130 i = 1, nc
130   c(i) = c2(i,n3)
      un = alog(un)
      eu3 = exp(exp(poly(c,nc,un)))
140   ww = (eu3 + tqr)/(one + eu3)
150   pw = pi6 *(atan(sqrt(ww/(1.0-ww))) - stqr)
      return
160   pw = zero
      return
      end
c



      subroutine wcoef(a,n,n2,eps,ifault)
c
c      algorithm as 181.1 applied statistics (1982) vol. 31, no. 2
c
c      obtain array a of weights for calculating w
c
      real a(n2), c4(2), c5(2), c6(3)
      data c4(1),c4(2)/0.6869,0.1678/,c5(1),c5(2)/0.6647,0.2412/,
     $  c6(1),c6(2),c6(3)/0.6431,0.2806,0.0875/
      data rsqrt2/0.70710678/, zero/0.0/, half/0.5/, one/1.0/,
     $ two/2.0/, six/6.0/, seven/7.0/, eight/8.0/, thirt/13.0/
      ifault = 1
      if (n .le. 2) return
      ifault = 3
      if (n/2 .ne. n2) return
      ifault = 2
      if (n .gt. 2000) return
      ifault = 0
      if (n .le. 6) go to 30
c
c
c      n .gt. 6 calculate rankits using approximate routine
c      nscor2 (as 177)
c
      call nscor2(a,n,n2,ifault)
      sastar = zero
      do 10 j=2,n2
10    sastar = sastar + a(j)*a(j)
      sastar = sastar *eight
      nn = n
      if (n .le. 20) nn = nn - 1
      an = nn
      a1sq = exp(alog(six*an + seven) - alog(six*an + thirt)
     $  + half*(one +(an-two)*alog(an+one) - (an - one)
     $  *alog(an+two)))
      a1star = sastar /(one/a1sq - two)
      sastar = sqrt(sastar + two *a1star)
      a(1) = sqrt(a1star)/sastar
```



```
      do 20 j = 2, n2
20    a(j) = two*a(j)/sastar
      go to 70
c
c      n .le. 6 use exact values for weights
c
30    a(1) = rsqrt2
      if (n .eq. 3) go to 70
      n3 = n - 3
      go to (40, 50, 60), n3
40    do 45 j=1,2
45    a(j) = c4(j)
      go to 70
50    do 55 j=1,2
55    a(j) = c5(j)
      go to 70
60    do 65 j=1, 3
65    a(j) = c6(j)
c
c      calculate the minimum possible value of w
c
70    eps = a(1)*a(1)/(one - one/float(n))
      return
      end
c


      function poly(c,nord,x)
c
c      algorithm as 181.2 applied statistics (1982) vol. 31, no. 2.
c
c      calculates the algebraic polynomial of order nord-1 with array
c      of coefficients c.  zero order coefficient is c(1).
c
      real c(nord)
      poly = c(1)
      if (nord .eq. 1) return
      p = x*c(nord)
      if (nord .eq. 2) go to 20
      n2 = nord- 2
      j = n2 + 1
      do 10 i = 1, n2
      p = (p + c(j))*x
      j = j - 1
10    continue
20    poly = poly + p
      return
      end



      subroutine nscor1(s,n,n2,work,ifault)
c
c      algorithm as 177  applied statistics (1982) vol. 31, no. 2
c
c      exact calculation of normal scores
c
      real s(n2), work(4,721)
      real zero, one, c1, d, c, scor, ai1, ani, an, h, alnfac
      data one/1.0e0/, zero/0.0e0/, h/0.025e0/, nstep/721/
      ifault = 3
      if (n2 .ne. n/2) return
      ifault = 1
      if (n .le. 1) return
      ifault = 0
      if (n .gt. 2000) ifault = 2
      an = n
c
c      calculate natural log of factorial(n)
c
      c1 = alnfac(n)
```

```
      d = c1 - alog(an)
c
c      accumulate ordinates for calculation of integral for rankits
c
      do 20 i=1,n2
      i1 = i - 1
      ni = n - i
      ai1 = i1
      ani = ni
      c = c1 - d
      scor = zero
      do 10 j = 1, nstep
10    scor = scor + exp(work(2,j) + ai1*work(3,j) + ani*work(4,j) + c)*
     $          work(1,j)
      s(i) = scor*h
      d = d + alog((ai1 + one)/ani)
20    continue
      return
      end




      subroutine init(work)
c
c      algorithm as 177.1 applied statistics (1982) vol. 31, no. 2
c
      real work(4,721)
      real xstart, h, pi2, half, xx, alnorm
      data xstart/-9.0e0/, h/0.025e0/, pi2/-0.918938533e0/,
     $     half/0.5e0/, nstep/721/
      xx = xstart
c
c      set up arrays for calculation of integral
c
      do 10 i=1, nstep
      work(1,i) = xx
      work(2,i) = pi2 - xx*xx*half
      work(3,i) = alog(alnorm(xx,.true.))
      work(4,i) = alog(alnorm(xx,.false.))
      xx = xstart + float(i)*h
10    continue
      return
      end


      real function alnfac(j)
c
c      algorithm 177.2 applied statistics (1982) vol. 31, no. 2
c
c      natural logarithm of factorial for nonnegative argument
c
      real r(7), one, half, a0, three, four, fourtn, fortty,
     $     fivfty, w, z
      data r(1), r(2), r(3), r(4), r(5), r(6), r(7)/0.0e0, 0.0e0,
     $ 0.69314718056e0, 1.79175946923e0, 3.17805383035e0,
     $ 4.78749174278e0, 6.57925121101e0/
      data one, half, a0, three, four, fourtn, fortty, fivfty/
     $ 1.e0, 0.5e0, 0.918938533205e0, 3.0e0,4.0e0,14.0e0, 420.0e0,
     $ 5040.0e0/
      if (j .ge. 0) go to 10
      alnfac = one
      return
10    if (j .ge. 7) go to 20
      alnfac = r(j+1)
      return
20    w = j + 1
      z = one/(w*w)
```



```
      alnfac = (w-half)*alog(w) - w + a0 + (((four - three*z)
     $  *z - fourtn)*z + fortty)/(fivfty*w)
      return
      end




      subroutine nscor2(s,n,n2,ifault)
c
c      algorithm as 177.3  applied statistics (1982) vol. 31, no. 2
c
c      approxmiation for rankits
c
      real s(n2),eps(4),dl1(4),dl2(4),gam(4),lam(4),bb,d,b1,an,
     $ ai,e1,e2,l1,correc,ppnd
      data eps(1), eps(2), eps(3), eps(4)
     $ /0.419885e0, 0.450536e0, 0.456936e0, 0.468488e0/,
     $ dl1(1), dl1(2), dl1(3), dl1(4)
     $ /0.112063e0, 0.121770e0, 0.239299e0, 0.215159e0/,
     $ dl2(1), dl2(2), dl2(3), dl2(4)
     $ /0.080122e0, 0.111348e0, -0.211867e0, -0.115049e0/,
     $ gam(1), gam(2), gam(3), gam(4)
     $ /0.474798e0, 0.469051e0, 0.208597e0, 0.259784e0/,
     $ lam(1), lam(2), lam(3), lam(4)
     $ /0.282765e0, 0.304856e0, 0.407708e0, 0.414093e0/
     $ bb/-0.283833e0/, d/-0.106136e0/, b1/0.5641896e0/
      ifault = 3
      if (n2 .ne. n/2) return
      ifault = 1
      if (n .le. 1) return
      ifault = 0
      if (n .gt. 2000) ifault = 2
      s(1) = b1
      if (n .eq. 2) return
c
c      calculate normal areas for 3 largest rankits
c
      an = n
      k = 3
      if (n2 .lt. k) k = n2
      do 5 i = 1, k
      ai = i
      e1 = (ai - eps(i))/(an + gam(i))
      e2 = e1**lam(i)
      s(i) = e1 + e2*(dl1(i) + e2*dl2(i))/an - correc(i,n)
5     continue
      if (n2 .eq. k) go to 20
c
c      calculate normal areas for remaining rankits
c
      do 10 i = 4, n2
      ai = i
      l1 = lam(4) + bb/(ai + d)
      e1 = (ai - eps(4))/(an + gam(4))
      e2 = e1**l1
      s(i) = e1 + e2*(dl1(4) + e2*dl2(4))/an - correc(i,n)
10    continue
c
c      convert normal tail areas to normal deviates
c
      ier = 0
20    do 30 i= 1, n2
30    s(i) = -ppnd(s(i),ier)
      if (ier .ne. 0) ifault = 4
      return
      end


      real function correc(i, n)
```

```
c
c      algorithm as 177.4 applied statistics (1982) vol. 31, no. 2
c
c      calculates correction for tail area of normal distribution
c      corresponding to the ith largest rankit in sample size n
c
      real c1(7), c2(7), c3(7), an, mic, c14
      data c1(1), c1(2), c1(3), c1(4), c1(5), c1(6), c1(7)
     $  /9.5e0, 28.7e0, 1.9e0, 0.0e0, -7.0e0, -6.2e0, -1.6e0/,
     $ c2(1), c2(2), c2(3), c2(4), c2(5), c2(6), c2(7)
     $ /-6.195e3, -9.569e3, -6.728e3, -17.614e3, -8.278e3, -3.570e3,
     $ 1.075e3/,
     $ c3(1), c3(2), c3(3), c3(4), c3(5), c3(6), c3(7)
     $ /9.338e4, 1.7516e5, 4.1040e5, 2.157e6, 2.376e6, 2.065e6,
     $ 2.065e6/,
     $ mic/1.e-6/, c14/1.9e-5/
      correc = c14
      if (i*n .eq. 4) return
      correc = 0.0
      if (i .lt. 1 .or. i .gt. 7) return
      if (i .ne. 4 .and. n .gt. 20) return
      if (i .eq. 4 .and. n .gt. 40) return
      an = n
      an = 1.0/(an*an)
      correc = (c1(i) + an*(c2(i) + an*c3(i)))*mic
      return
      end

      function alnorm(x,upper)
c
c      algorithm as 66  applied statistics (1973) vol. 22, no. 3
c
c      evaluates the tail area of the standardised normal curve
c      from x to infinity if upper is .true. or from minus
c      infinity to x if upper is .false.
c
c*** note: instead of algorithm as 66, we have substituted the imsl
c*** routine 'mdnor'.  both routines have machine-dependent parameters.
c
      logical upper
      z = x
      if (upper) z = -z
      call mdnor(z,tail)
      alnorm = tail
      return
      end


      real function ppnd(p, ifault)
c
c      algorithm as 111  applied statistics (1977), vol. 26, no. 1
c
c      produces normal deviate corresponding to lower tail area of p
c
c*** note: instead of algorithm as 111, we have substituted the imsl
c*** routine 'mdnris'.
c
      call mdnris(p,y,ier)
      ppnd = y
      ifault = 0
      if (ier .eq. 0) return
      ppnd = 0.0
      ifault = 1
      return
      end


      subroutine sctp(n,nvar,a,lda,icol,range,symbol,
     1                                  xtitle,ytitle,title)
```



```
c     bruce schmeiser    purdue university    july 1987
c     scatter plot
c     this routine is written to look like sctp in imsl's stat/library 1987.
c     nvar is restricted to the value 2.
      dimension a(2000,2), icol(nvar), range(4), symbol(nvar)
      character xtitle*40, ytitle*40, title*40
      dimension yprint(40)
      character ytitl2*40, yprint*1
      equivalence (ytitl2, yprint)
      dimension screen(24,80),icount(24,80)
      character screen*1, char*1
      dimension char(11)
      data char/' ','1','2','3','4','5','6','7','8','9','M'/
      data mnpltx/1/, mxpltx/68/, mnplty/1/, mxplty/18/
      ytitl2 = ytitle
      do 100 i=mnpltx,mxpltx
      do 100 j=mnplty,mxplty
  100 icount(j,i) = 0
      xmin = range(1)
      xmax = range(2)
      ymin = range(3)
      ymax = range(4)
      xwidth = (xmax - xmin) / (mxpltx - mnpltx + 1)
      ywidth = (ymax - ymin) / (mxplty - mnplty + 1)
      if (xwidth .le. 0.) xwidth = .00001
      if (ywidth .le. 0.) ywidth = .00001
c
c.....compute counts
c
      do 200 iobs=1,n
      x = a(iobs,1)
      y = a(iobs,2)
      i = (x - xmin)/xwidth + 1
      j = (y - ymin)/ywidth + 1
      if (i .lt. 1)  i = 1
      if (j .lt. 1)  j = 1
      if (i .gt. 80) i = 80
      if (j .gt. 24) j = 24
  200 icount(j,i) = icount(j,i) + 1
c
c.....create screen
c
      do 300 i=mnpltx,mxpltx
      do 300 j=mnplty,mxplty
      if (icount(j,i) .gt. 10) icount(j,i) = 10
  300 screen(j,i) = char(icount(j,i)+1)
c
c.....print screen
c
      write (6,1011) ymax
      do 400 j=mnplty,mxplty
      jline = mxplty + mnplty - j
      jtit = j - mnplty + 1
  400 write (6,1010) yprint(jtit), (screen(jline,i),i=mnpltx,mxpltx)
 1010 format (5x, a1, 4x, '+', 68a1)
      write (6,1011) ymin
 1011 format (f9.3, 1x, 6('+--------'),'+-------')
      write (6,1012) xmin, xtitle, xmax
 1012 format (f15.3, 8x, a40, 1x, f15.3)
      return
      end
*
*=====================================================================
*
      subroutine finit (iunit,filnam)
*
*=====================================================================
*
*     --- This routine performs the necessary checking, and then
```

```
*     --- initializes a file for use by the fitting program.
*
      character*10 filnam
      logical fex,fop
      inquire (file = filnam,exist = fex,opened = fop)
      if (.not. fop) then
         if (fex) then
            open (iunit,file = filnam,status = 'old')
         else
            open (iunit,file = filnam,status = 'new')
         end if
      end if
      rewind iunit
      return
      end
*
*=====================================================================
*
      subroutine vsrta  (a,la)
c                                  specifications for arguments
      integer             la
      real                a(la)
c                                  specifications for local variables
      integer             iu(21),il(21),i,m,j,k,ij,l
      real                t,tt,r
c                                  first executable statement
      m=1
      i=1
      j=la
      r=.375
      if (la.le.0) return
   10 if (i .eq. j) go to 55
   15 if (r .gt. .5898437) go to 20
      r=r+3.90625e-2
      go to 25
   20 r=r-.21875
   25 k=i
c                                  select a central element of the
c                                  array and save it in location t
      ij=i+(j-i)*r
      t=a(ij)
c                                  if first element of array is greater
c                                  than t, interchange with t
      if (a(i) .le. t) go to 30
      a(ij)=a(i)
      a(i)=t
      t=a(ij)
   30 l=j
c                                  if last element of array is less than
c                                  t, interchange with t
      if (a(j) .ge. t) go to 40
      a(ij)=a(j)
      a(j)=t
      t=a(ij)
c                                  if first element of array is greater
c                                  than t, interchange with t
      if (a(i) .le. t) go to 40
      a(ij)=a(i)
      a(i)=t
      t=a(ij)
      go to 40
   35 if(a(l).eq.a(k)) go to 40
      tt=a(l)
      a(l)=a(k)
      a(k)=tt
c                                  find an element in the second half of
c                                  the array which is smaller than t
   40 l=l-1
      if (a(l) .gt. t) go to 40
```



```
c                                  find an element in the first half of
c                                  the array which is greater than t
   45 k=k+1
      if (a(k) .lt. t) go to 45
c                                  interchange these elements
      if (k .le. l) go to 35
c                                  save upper and lower subscripts of
c                                  the array yet to be sorted
      if (l-i .le. j-k) go to 50
      il(m)=i
      iu(m)=l
      i=k
      m=m+1
      go to 60
   50 il(m)=k
      iu(m)=j
      j=l
      m=m+1
      go to 60
c                                  begin again on another portion of
c                                  the unsorted array
   55 m=m-1
      if (m .eq. 0) return
      i=il(m)
      j=iu(m)
   60 if (j-i .ge. 11) go to 25
      if (i .eq. 1) go to 10
      i=i-1
   65 i=i+1
      if (i .eq. j) go to 55
      t=a(i+1)
      if (a(i) .le. t) go to 65
      k=i
   70 a(k+1)=a(k)
      k=k-1
      if (t .lt. a(k)) go to 70
      a(k+1)=t
      go to 65
      end
*
*=====================================================================
*
      subroutine mdnris (p,y,ier)
c                                  specifications for arguments
      real                p,y
      integer             ier
c                                  specifications for local variables
      real                eps,g0,g1,g2,g3,h0,h1,h2,a,w,wi,sn,sd
      real                sigma,sqrt2,x,xinf
      data                xinf/1.7014e+38/
      data                sqrt2/1.414214/
      data                eps/.59605e-07/
      data                g0/.1851159e-3/,g1/-.2028152e-2/
      data                g2/-.1498384/,g3/.1078639e-1/
      data                h0/.9952975e-1/,h1/.5211733/
      data                h2/-.6888301e-1/
c                                  first executable statement
      ier = 0
      if (p .gt. 0.0 .and. p .lt. 1.0) go to 5
      ier = 129
      sigma = sign(1.0,p)
      y = sigma * xinf
      go to 9000
    5 if(p.le.eps) go to 10
c     bug fix ? from imsl
c     x = 1.0 -(p + p)
      x = 1.0 - p
      x = x - p
      call merfi (x,y,ier)
```

```
      y = -sqrt2 * y
      go to 9005
c                                  p too small, compute y directly
   10 a = p+p
      w = sqrt(-alog(a+(a-a*a)))
c                                  use a rational function in 1./w
      wi = 1./w
      sn = ((g3*wi+g2)*wi+g1)*wi
      sd = ((wi+h2)*wi+h1)*wi+h0
      y = w + w*(g0+sn/sd)
      y = -y*sqrt2
      go to 9005
 9000 continue
c     call uertst(ier,'mdnris')
 9005 return
      end
*
*====================================================================
*
      subroutine mdnor  (y,p)
c                                  specifications for arguments
      real               p,y
c                                  specifications for local variables
      real               sqr1d2
      data               sqr1d2/.7071068/
c                                  first executable statement
      p = .5 * erfc(-y*sqr1d2)
      return
      end
*
*====================================================================
*
      subroutine merfi (p,y,ier)
c                                  specifications for arguments
      real               p,y
      integer            ier
c                                  specifications for local variables
      real               a,b,x,z,w,wi,sn,sd,f,z2,rinfm,a1,a2,a3,b0,b1,
     *                   b2,b3,c0,c1,c2,c3,d0,d1,d2,e0,e1,e2,e3,f0,f1,
     *                   f2,g0,g1,g2,g3,h0,h1,h2,sigma
      data               a1/-.5751703/,a2/-1.896513/,a3/-.5496261e-1/
      data               b0/-.1137730/,b1/-3.293474/,b2/-2.374996/
      data               b3/-1.187515/
      data               c0/-.1146666/,c1/-.1314774/,c2/-.2368201/
      data               c3/.5073975e-1/
      data               d0/-44.27977/,d1/21.98546/,d2/-7.586103/
      data               e0/-.5668422e-1/,e1/.3937021/,e2/-.3166501/
      data               e3/.6208963e-1/
      data               f0/-6.266786/,f1/4.666263/,f2/-2.962883/
      data               g0/.1851159e-3/,g1/-.2028152e-2/
      data               g2/-.1498384/,g3/.1078639e-1/
      data               h0/.9952975e-1/,h1/.5211733/
      data               h2/-.6888301e-1/
      data               rinfm/1.7014e+38/
c                                  first executable statement
      ier = 0
      x = p
      sigma = sign(1.0,x)
c                                  test for invalid argument
      if (.not.(x.gt.-1. .and. x.lt.1.)) go to 30
      z = abs(x)
      if (z.le. .85) go to 20
      a = 1.-z
      b = z
c                                  reduced argument is in (.85,1.),
c                                     obtain the transformed variable
    5 w = sqrt(-alog(a+a*b))
      if (w.lt.2.5) go to 15
      if (w.lt.4.) go to 10
```



```
c                                  w greater than 4., approx. f by a
c                                     rational function in 1./w
      wi = 1./w
      sn = ((g3*wi+g2)*wi+g1)*wi
      sd = ((wi+h2)*wi+h1)*wi+h0
      f = w + w*(g0+sn/sd)
      go to 25
c                                  w between 2.5 and 4., approx. f
c                                     by a rational function in w
   10 sn = ((e3*w+e2)*w+e1)*w
      sd = ((w+f2)*w+f1)*w+f0
      f = w + w*(e0+sn/sd)
      go to 25
c                                  w between 1.13222 and 2.5, approx.
c                                     f by a rational function in w
   15 sn = ((c3*w+c2)*w+c1)*w
      sd = ((w+d2)*w+d1)*w+d0
      f = w + w*(c0+sn/sd)
      go to 25
c                                  z between 0. and .85, approx. f
c                                     by a rational function in z
   20 z2 = z*z
      f = z+z*(b0+a1*z2/(b1+z2+a2/(b2+z2+a3/(b3+z2))))
c                                  form the solution by mult. f by
c                                     the proper sign
   25 y = sigma*f
      ier = 0
      go to 9005
c                                  error exit. set solution to plus
c                                     (or minus) infinity
   30 ier = 129
      y = sigma * rinfm
 9000 continue
c     call uertst(ier,'merfi ')
 9005 return
      end
*
*====================================================================
*
      real function erfc(y)
c                                  specifications for arguments
      real               y
c                                  specifications for local variables
      integer            isw,i
      dimension          p(3),q(2),p1(5),q1(4),p2(3),q2(2)
      real               p,q,p1,q1,p2,q2,xmin,xlarge,sqrpi,x,
     *                   res,xsq,xnum,xden,xi,xbig
c                                  coefficients for 0.0 .le. y .lt.
c                                  .477
      data               p(1)/.3166529/,p(2)/1.722276/,
     *                   p(3)/21.38533/
      data               q(1)/7.843746/,q(2)/18.95226/
c                                  coefficients for .477 .le. y
c                                  .le. 4.0
      data               p1(1)/.5631696/,p1(2)/3.031799/,
     *                   p1(3)/6.865018/,p1(4)/7.373888/,
     *                   p1(5)/4.318779e-5/
      data               q1(1)/5.354217/,q1(2)/12.79553/,
     *                   q1(3)/15.18491/,q1(4)/7.373961/
c                                  coefficients for 4.0 .lt. y
      data               p2(1)/-5.168823e-2/,p2(2)/-.1960690/,
     *                   p2(3)/-4.257996e-2/
      data               q2(1)/.9214524/,q2(2)/.1509421/
c                                  constants
      data               xmin/1.0e-5/,xlarge/4.1875e0/
c                                  erfc(xbig) .approx. setap
      data               xbig/9.0/
      data               sqrpi/.5641896/
c                                  first executable statement
```

```
      x = y
      isw = 1
      if (x.ge.0.0e0) go to 5
      isw = -1
      x = -x
    5 if (x.lt..477e0) go to 10
      if (x.le.4.0e0) go to 25
      if (isw .gt. 0) go to 35
      if (x.lt.xlarge) go to 40
      res = 2.0e0
      go to 55
c                                  abs(y) .lt. .477, evaluate
c                                  approximation for erfc
   10 if (x.lt.xmin) go to 15
      xsq = x*x
      xnum = (p(1)*xsq+p(2))*xsq+p(3)
      xden = (xsq+q(1))*xsq+q(2)
      res = x*xnum/xden
      go to 20
   15 res = x*p(3)/q(2)
   20 if (isw.eq.-1) res = -res
      res = 1.0e0-res
      go to 55
c                                  .477 .le. abs(y) .le. 4.0
c                                  evaluate approximation for erfc
   25 xsq = x*x
      xnum = p1(5)*x+p1(1)
      xden = x+q1(1)
      do 30 i=2,4
         xnum = xnum*x+p1(i)
         xden = xden*x+q1(i)
   30 continue
      res = xnum/xden
      go to 45
c                                  4.0 .lt. abs(y), evaluate
c                                  minimax approximation for erfc
   35 if (x.gt.xbig) go to 50
   40 xsq = x*x
      xi = 1.0e0/xsq
      xnum = (p2(1)*xi+p2(2))*xi+p2(3)
      xden = (xi+q2(1))*xi+q2(2)
      res = (sqrpi+xi*xnum/xden)/x
   45 res = res*exp(-xsq)
      if (isw.eq.-1) res = 2.0e0-res
      go to 55
   50 res = 0.0e0
   55 erfc = res
      return
      end
```