CHAPTER 9

SIMULATING

WITH

NONSEASONAL MODELS

9.1 INTRODUCTION

Two of the most important uses of time series models in engineering are forecasting and simulation. In water resources applications, simulated sequences can assist designers in determining the appropriate size of a system and estimating the associated benefits and costs. Once in operation, reliable forecasts are required to ensure that the maximum benefits are obtained from operating the system. Chapter 8 of Part IV of the book deals with forecasting whereas the purpose of this chapter is to present flexible methods for simulating with ARMA and ARIMA models. Table 1.6.3 indicates the locations in the book where explanations are given about the theory and practice of both simulation and forecasting for different kinds of models.

The main objective of *forecasting* is to use the time series model fitted to a data set to obtain the most accurate estimate or prediction of future unknown observations. The goal of *simulation* is to employ the fitted model to generate a set of stochastically equivalent sequences of observations which could possibly occur in the future. These *simulated sequences* are often referred to as *synthetic data* by hydrologists because they are only possible realizations of what could take place. As a matter of fact, the overall science of fitting stochastic models to hydrologic data and using these models for simulation purposes is often called *synthetic hydrology*. Other titles for this field include *stochastic and operational hydrology*.

Simulation is now a widely accepted technique to aid in both the *design and operation* of water resources systems. Vogel and Stedinger (1988), for instance, demonstrate that using synthetic data generated by stochastic streamflow models can lead to improvements in the prediction of reservoir design capacity estimates. Besides the design and operation of large-scale engineering systems, another main use of simulation is to *investigate theoretical properties* of stochastic models. Often it is analytically impossible to derive certain theoretical characteristics of a given type of time series model. However, by using simulation one can determine these theoretical properties to any desired level of accuracy. In Chapter 10, simulation is used with ARMA models to study theoretical problems related to what is called the Hurst phenomenon.

When carrying out a simulation study, there are certain problems that should be avoided. For instance, many current simulation methods that are widely accepted, do not use correct initial values. Although the effect of *starting values* is transitory, it could cause systematic bias in a simulation study and, therefore, as pointed out by Moran (1959, Ch. 5) and Copas (1966), the choice of initial values is important. To attempt to overcome this problem, some researchers discard the first section of a synthetic time series to supposedly get rid of the effects of initial values. However, exactly how many values of the generated series should be rejected and how much computer time is wasted by generating data that is not used?

As an example of a conservative approach to the effect of starting values, consider the simulation study of Brown and Hardin (1973). These authors used deterministic starting values for an AR model of order two and then generated a series with a length of 30,000 values. The first 15,000 values of the synthetic trace were discarded to supposedly nullify the effects of using non-random initial values.

The simulation procedures given in this chapter and also by McLeod and Hipel (1978) do not require fixed starting values. They are designed in a manner such that random realizations of the underlying stochastic process are used as initial values. Therefore, the results of a simulation study are not significantly biased and it is not necessary to disregard any of the generated data.

Often it is required to generate k' time series of length k. Some researchers resort to producing a single synthetic series of length $k' \cdot k$ and then splitting this long series into k' series of length k. If any serial correlation is present, then the results of any simulation study will be biased by this rather crude procedure. To overcome this problem of *bias*, the authors recommend generating k' separate time series of length k. If the generating procedures given in this chapter are adopted, then each time another series of length k is obtained, new random realizations of the stochastic process are used as starting values.

Figure 9.1.1 displays the *overall approach* for using a time series model for simulating data. If necessary, the given time series can be transformed using the Box-Cox transformation in [3.4.30] in order to alleviate problems with non-normality and/or non-constant variance discovered in the original data or the residuals of a model fitted to the untransformed time series. Following the three stages of model construction presented in Part III and summarized in Figure III.1, one can then develop an ARMA or ARIMA model for describing the transformed series or the original series when a Box-Cox transformation is not needed. This fitted model can then be used for generating synthetic sequences. When simulating data using a time series model, procedures described in Section 9.2 can be employed for generating the uncorrelated a_t terms in the ARMA or ARIMA model. The generated a_t's are used in either the WASIM1 or WASIM2 procedures of Sections 9.3 and 9.4, respectively, in order to simulate unbiased sequences from ARMA models that use proper starting values. If one is using an ARIMA model for simulation purposes, then the methods of Section 9.5 must be used to produce the nonstationary sequences. Finally, if the simulation model was fitted to the transformed series, then one must take the inverse Box-Cox transformation of the simulated values to obtain synthetic values that have the same units as the original data, as explained in Section 9.6.

When the approach of Figure 9.1.1 is employed for obtaining simulated sequences, it is assumed that the parameters of the model producing the synthetic data are known exactly. However, in practice one must estimate the model parameters and the uncertainty contained in the parameter estimates is reflected by their standard errors. The WASIM3 procedure of Section 9.7 can be used for incorporating *parameter uncertainty* into a simulation study. Following this, three practical applications are given in Section 9.8 to portray the effectiveness of the aforementioned simulation techniques.

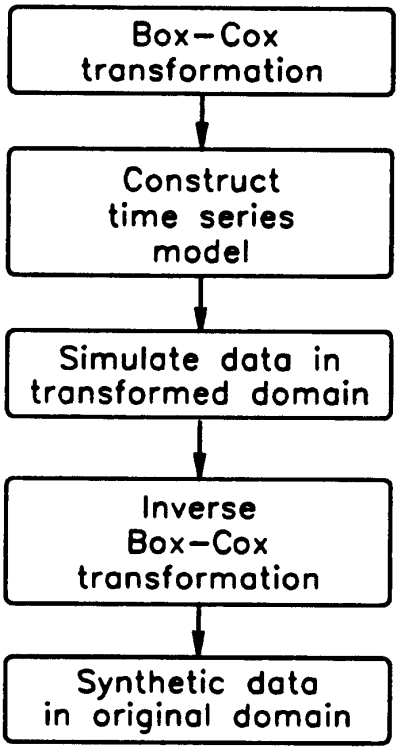


Figure 9.1.1. Overall procedure for simulating data.

9.2 GENERATING WHITE NOISE

9.2.1 Introduction

As shown in Figure 9.1.1, when developing a model for use in simulation, the first step is, if necessary, to transform the z_t series using the Box-Cox transformation in [3.4.30] to obtain the series $z_t^{(\lambda)}$. If the series is nonstationary, and, for example, the level is increasing over time, one can remove the nonstationarity using the differencing operator in [4.3.3] to obtain the stationary w_t series. As noted in Sections 2.4, 3.1, 4.1 and elsewhere in the book, often annual hydrological time series of moderate length are stationary. However, usually socio-economic series such as the water use, electricity consumption and Beveridge wheat price indices displayed in Figures 4.3.8, 4.3.10, and 4.3.15, respectively, are nonstationary. Whatever the case, one fits an ARMA model from [4.3.4] to the stationary w_t series which may be formed by first differencing the $z_t^{(\lambda)}$ series.

As an example as to how one would use an ARMA model for simulation, consider the cases of an AR(1) model written following [3.2.1] as

$$w_t = \phi_1 w_{t-1} + a_t \qquad [9.2.1]$$

where ϕ_1 is the first order AR parameter and a_t is the white noise sequence that is normal and independently distributed with mean zero and variance σ_a^2 (i.e. NID(0,σ_a^2)). For convenience, it is assumed that the mean of w_t in [9.2.1] is zero. If the w_t series were formed by differencing the w_t series, the mean would be zero. However, if the $z_t^{(\lambda)}$ series were stationary and hence $z_t^{(\lambda)} = w_t$, then the mean would probably be nonzero. In either case, one can use [9.2.1] to simulate the w_t series. If there is a nonzero mean, it can simply be added to the simulated values.

Suppose one wishes to employ [9.2.1] to simulate 10 values of the w_t series. Then

$$w_2 = \phi_1 w_1 + a_2$$

Given that ϕ_1 is estimated from the data or else known in advance, one would have to have a starting value, w_1, and a normal white noise term, a_2, to simulate w_2. To avoid bias caused by using a fixed starting value for w_1, one can employ WASIM1 or WASIM2 from Section 9.3 and 9.4, respectively to obtain w_1. The white noise term a_2 is generated using an approach described in this section. Next, one can obtain w_3 using

$$w_3 = \phi_1 w_2 + a_3$$

where w_2 is known from the previous step and a_3 is generated by the computer. In general,

$$w_j = \phi_1 w_{j-1} + a_j, \quad j = 2,3,\ldots,10$$

to simulate the ten values of w_t. If the w_t were differenced, one would have to use the algorithm of Section 9.5 to obtain the $z_t^{(\lambda)}$ series. Finally, the inverse Box-Cox transformation would have to be taken to get the simulated untransformed z_t's.

As shown in Figure 9.2.1, there are two main steps required to generate the a_t's which are NID(0,σ_a^2). The first stage is to use an appropriate random number generator to produce independent random variables that follow a uniform distribution on the interval from zero to one. Random number generators are discussed in Section 9.2.2. The next step is to employ a technique that transforms the uniformly distributed variables to ones that follow the required distribution such as a normal distribution. The approach for accomplishing this is presented in Section 9.2.3. A classical text on generating independently distributed random variables is the one by Knuth (1969) while the book of Yakowitz (1977) provides a well written and entertaining account of this and other topics in computational probability and simulation.

9.2.2 Random Number Generators

Overview

As indicated in Figure 9.2.1, the first step in generating independently distributed random variables that follow the same distribution is to produce independent variables that follow a uniform distribution on the interval (0,1). To obtain uniformly distributed variables one uses what is called a *random number generator*. Following the instructions encoded into a computer

program for the random number generator, a digital computer can produce the random numbers.

A digital computer follows a strictly deterministic process, since it exactly adheres to a program's precise instructions. Nonetheless, it is possible for a computer to generate a sequence of numbers $\{u_i\}$ which appear to be independent values distributed on the unit interval. These numbers are referred to as *pseudo-random numbers*. The *probability density function* (*pdf*), $f(u)$, and the *cumulative distribution function*, $F(u)$, for a uniformly or rectangular distributed random variable, u, on the interval (0,1), are shown in Figure 9.2.2.

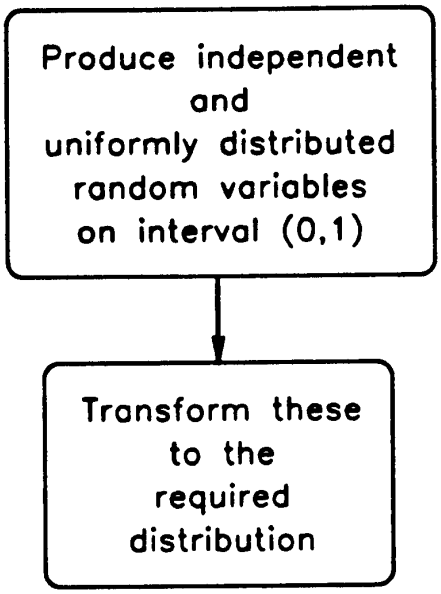
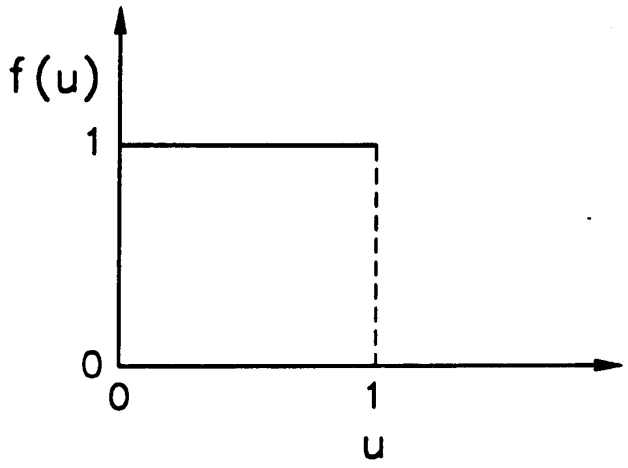


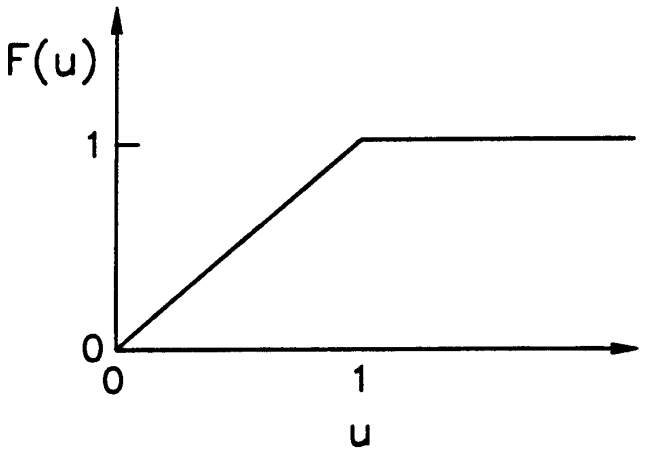
Figure 9.2.1. Generating identically and independently distributed random variables.

When employing a digital computer, some discrete approximation must be used in place of a continuous random variable such as one following the uniform distribution in Figure 9.2.2. Therefore, one may generate the first N terms of a decimal expansion of the value of a uniform variable. In an experiment for generating uniform variables, let D_j denote the jth decimal of the decimal expansion of an outcome. For each $k = 0,1,2,\ldots,9$, and $j = 1,2,\ldots,$ the event $D_j = k$ has a probability of $\frac{1}{10} = 0.1$.

The main idea behind a pseudo-random number generator is that successive numbers having the same length m are created in such a way that in the long run, each digit (i.e. $0,1,2,\ldots,9$) is expected to occur with probability 0.1 at each decimal place. Moreover, the occurrence of a given digit at a specific decimal place is independent of the digits occurring at other decimal places as well as previously generated random numbers.



(a) Probability density function



(b) Cumulative distribution function

Figure 9.2.2. Uniform probability density function and cumulative distribution function.

Over the years a range of pseudo-random number generators has been developed. One of the earliest random number generators is the *middle-square random number generator* of von Neumann (1951). One of the problems with this generator is that it has short cycles because after a certain length of time the uniform variables produced by the generator repeat themselves over relatively short time periods. Consequently, researchers developed generators that would have cycle lengths that are as long as possible.

Linear Congruential Random Number Generators

A particularly good family of pseudo-random number generator having maximum cycle length is the *linear congruential random number generator.* The linear congruential generators, originally suggested by Lehmer in 1948, form the most popular and highly studied class of methods for generating a sequence of pseudo-random numbers. These techniques are based upon the recurrence relationship

$$x_i = (ax_{i-1} + c) \ (mod\ m) \qquad [9.2.2]$$

where a *seed*, x_o, is an integer $0 \leq x_o < m$ that is required to start the generator; multiplier a is an integer $0 < a < m$; increment c is an integer $0 \leq c < m$. The symbol *mod m* stands for modulus m and, therefore, x_i is the remainder after $ax_{i-1} + c$ is divided by the integer m where $m > 0$. The sequence $\{x_i\}$ formed by allowing i to take on values $i = 1,2, \ldots,$ is often called a linear congruential sequence or a random number stream. In addition, when $c > 0$ or $c = 0$, the linear congruential generator is often referred to as a *mixed congruential generator* and a *multiplicative congruential generator,* (Wichmann and Hill, 1982), respectively.

The modulo arithmetic in [9.2.2] guarantees that each entry in $\{x_i\}$ is an integer falling in the interval $(0,m-1)$. Consequently, the set of numbers, $\{x_i/m\}$, forms a sequence of uniformly distributed random variables, $\{u_i\}$.

One must carefully select the parameters a, c, m, and x_o in [9.2.2] for the linear congruential method in order to obtain a sequence that follows the distributional properties of independent uniformly distributed random variables, has a sequence containing the maximum period length, and achieves computational efficiency. The *cycle length or period* is some integer p such that $u_i = u_{i+p}$ for all $i \geq 0$. Clearly, this cycle length cannot be greater than the modulus m. A large period length and, hence, large m is required for achieving apparent randomness of the entries in the sequence $\{\mu_i\}$ or, equivalently, $\{x_i\}$. Hull and Dobell (1962) show that the linear congruential sequence $\{u_i\}$ has period m if and only if the following restrictions are satisfied:

1. c is relatively prime to m (i.e. c and m have no common factor other than unity).
2. $a - 1$ is a multiple of the period p, for every prime p dividing m.
3. $a - 1$ is a multiple of 4 if m is a multiple of 4.

Rules for finding the period for any choices of a, c and m are presented by Marsaglia (1972). Further detailed information regarding the selection of the parameter values in [9.2.2] is given by Janson (1966), Knuth (1969), Dieter (1972), as well as many other authors.

In summary, the algorithm for the linear congruential random number generator is as follows:

A. **Input:** Carefully select the parameters a, c and m in [9.2.2], plus the starting value x_o and the length N of the sequence to be generated.

B. **Calculations:**

0. Set $i = 1$
1. $x_i = (ax_{i-1} + c) \ (mod\ m)$
2. $u_i = truncated\ decimal\ expansion\ of\ x_i/m$ [9.2.3]
3. $i = i + 1$
4. If $i < N$, go to 1.
5. Stop.

C. **Output:** Sequence $\{u_i\} = u_1, u_2, \ldots, u_N$, of independent uniformly distributed random variables on the interval (0,1).

Example:

To demonstrate how the linear congruential random generator works, consider a simple illustration for which $a = 3$, $c = 7$, $m = 16$ and $x_o = 2$. Hence, equation [9.2.2] becomes

$$x_i = (3x_{i-1} + 7) \ (mod\ 16)$$

Using this equation, one calculates the sequence for x_i as:

$x_o = 2$ (*starting value or seed*)

$x_1 = (3(2) + 7)(mod\ 16) = 13(mod\ 16) = 13$

$x_2 = (3(13) + 7)(mod\ 16) = 46(mod\ 16) = 14$

$x_3 = (3(14) + 7)(mod\ 16) = 49(mod\ 16) = 1$

$x_4 = (3(1) + 7)(mod\ 16) = 10(mod\ 16) = 10$

$x_5 = (3(10) + 7)(mod\ 16) = 37(mod\ 16) = 5$

$x_6 = (3(5) + 7)(mod\ 16) = 22(mod\ 16) = 6$

$x_7 = (3(6) + 7)(mod\ 16) = 25(mod\ 16) = 9$

$x_8 = (3(9) + 7)(mod\ 16) = 34(mod\ 16) = 2$

$x_9 = (3(2) + 7)(mod\ 16) = 13(mod\ 16) = 13$

.
.
.

Consequently, the sequence $\{x_i\}$ is:

$$\{x_i\} = 2,13,14,1,10,5,6,9,2,13,\cdots$$

By dividing each entry in the x_i set of numbers by the modulus, one obtains the u_i sequence according to [9.2.3] as:

$$\{u_i\} = \frac{2}{16},\frac{13}{16},\frac{14}{16},\frac{1}{16},\frac{10}{16},\frac{5}{16},\frac{6}{16},\frac{9}{16},\frac{2}{16},\frac{13}{16},\cdots$$

Notice that these two sequences repeat themselves after only eight terms. Because of this, the entries in $\{u_i\}$ would not be independent. In addition, since the calculated values can only take on discrete values that are integer multiples of $\frac{1}{16}$ and many of these values such as $\frac{3}{16}$ and $\frac{4}{16}$ are missing, the u_i's are not uniformly nor continuously distributed. As noted earlier, to mimic independence and a uniform distribution, the period has to be as large as possible and, hence, a, c, m and x_o must be wisely selected.

Testing Random Number Generators:

As pointed out by Yakowitz (1977) and others, all feasible random number generators are inherently faulty from a purely philosophical viewpoint. Nonetheless, when appropriate choices of input parameters are made for the linear congruential random number generator, the generated sequences can satisfy standard statistical tests and are adequate for use in engineering applications. Traditionally, one employs separate tests to ascertain if the sequence $\{u_i\}$ is independent and also uniformly distributed. When using a random number generator on a given computer facility for the first time, one should invoke a range of statistical tests to ensure that the independence and uniform distribution assumptions of the u_i's are satisfied. Furthermore, one wishes to employ a random number generator that is also computationally efficient.

9.2.3 Generation of Independent Random Variables

General Approach

As shown in Figure 9.2.1, sequences following distributions other than the rectangular or uniform distribution are obtained by transforming rectangularly distributed random variables to the required distribution. For almost all of the models discussed in this book, one assumes that the a_t innovations are normally independently distributed as NID$(0,\sigma_a^2)$. Therefore, when simulating using an ARMA or ARIMA model, one wishes to transform the uniformly distributed variables generated using the techniques from the previous section into a sequence which is Gaussian or normally distributed and has uncorrelated elements.

More specifically, suppose one wishes to transform a sequence $\{u_i\}$ of independent uniformly distributed random variables to some other distribution such as a normal distribution. Let the variable following the other distribution be denoted by w where $f(w)$ and $F(w)$ represent the

probability density function and cumulative distribution function, respectively. A *universal random variable generator* is available for transforming the uniform random variables to the required distribution. Following Yakowitz (1977, p. 41) the main steps in this algorithm are as follows:

A. **Inputs:** The sequence $\{u_i\}$ of independent uniformly distributed random variables and the formula $F(w)$ of the cumulative distribution for the distribution that one wishes the transformed variables to follow. Also, one should fix the length, N, of the w sequence to be generated.

B. **Calculations:**

0. Set $i = 1$.

1. For $u = u_i$, determine w such that

$$w = minimum \left\{ y : F(y) \geq u \right\} \qquad [9.2.4]$$

2. Assign $w_i = w$.

3. $i = i + 1$

4. If $i < N$, go to 1.

5. Stop.

C. **Output:** Sequence $\left\{ w_i \right\} = w_1, w_2, \ldots, w_N$, which are independent and follow the required distribution $F(w)$. It can be proven that if the u_i's are uniform, the w_i's will follow the distribution $F(w)$.

Simulating Independent Normal Sequences

The probability density function for a *standard normal random variable* having a mean of zero and standard deviation of unity is defined by the probability density function

$$f(y) = (2\pi)^{-1/2} \exp(-y^2/2), \quad -\infty < y < \infty \qquad [9.2.5]$$

In shortform notation, the distribution in [9.2.5] is written as $N(0,1)$. If the y's are also independently distributed, then they are normally independently distributed as NID(0,1). A normally distributed random variable, w, having a mean, μ_w, and standard derivation, σ_w, can be obtained from the standard normal variable, y, using the transformation

$$w = \sigma_w y + \mu_w \qquad [9.2.6]$$

The notation for the distribution of w is $N(\mu_w, \sigma_w^2)$. If the w's are normally independently distributed they are said to be NID(μ_w, σ_w^2). This *normal distribution* is often referred to as the *Gaussian distribution* in commemoration of the great German mathematician Karl Gauss.

A variety of approximate and exact algorithms are available for generating normally distributed random variables from uniformly distributed ones. Box and Muller (1958) provide an exact method for generating random variables. In particular, two random variables, u_1 and u_2,

that are independent and uniformly distributed on the interval (0,1) (see Section 9.2.2), are transformed to *NID*(0,1) random variables, y_1 and y_2, using the relationships

$$y_1 = (-2\ln u_1)^{1/2}\cos 2\pi u_2$$
$$y_2 = (-2\ln u_1)^{1/2}\sin 2\pi u_2 \qquad [9.2.7]$$

One then uses the transformation in [9.2.6] to obtain the random variables w_1 and w_2 that are distributed as NID(μ_w,σ_w^2). The accuracy of the Box-Muller algorithm is determined by the computer word length as well as the accuracy of the logarithmic and trigonometric routines that are available on the computer. The calculations required in [9.2.7] can make this algorithm less computationally efficient relative to other competing algorithms.

A highly recommended algorithm for exactly generating NID(0,1) observations from independent uniformly distributed random variables is the method of Marsaglia and Bray (1964). Marsaglia et al. (1964) and also Knuth (1969, pp. 105-112) provide detailed information about how to encode the Marsaglia-Bray algorithm. Knuth (1969) thinks that the Marsaglia-Bray algorithm is more efficient from a computational viewpoint than the other perfect generator of Box and Muller (1958). However, the Marsaglia-Bray algorithm is more complicated and, therefore, requires a greater programming effort.

Generating Other Distributions

As pointed out in the first part of Section 9.2.3, a general approach is available for transforming random variables from a random number generator to any required distribution. In the previous subsection, for example, techniques are discussed for generating random variables that are NID(0,1). Methods are also available for generating independent variables which follow distributions such as exponential, gamma or Poisson distributions.

One can generate some types of independently distributed variables directly from the NID(0,1) variables. For instance, to generate independent *log-normal random variables*, the procedure is as follows:

1. Generate NID(0,1) random variables y_i, $i = 1,2,\ldots,N$.
2. Use the transformation

$$v_i = \exp(\mu_w + \sigma_w y_i) \qquad [9.2.8]$$

to obtain v_i, $i = 1,2,\ldots,N$, that are log normally distributed having parameters μ_w and σ_w for the mean and standard deviation, respectively, of the NID(μ_w,σ_w^2) random variables.

Another example of generating a certain type of independent random variables directly from the NID(0,1) variables is the generation of independent *Pearson-type III variables*. The formula for the Pearson-type III distribution is

$$f(\nu) = \frac{1}{\Gamma(p)} e^{-\nu}\nu^{p-1} \qquad [9.2.9]$$

where $0 \le \nu < \infty, \ldots, p > 1$ and $\Gamma(p)$ is the gamma distribution. For generating independent *Pearson-type III random variables*, the procedure is as follows:

1. Generate NID(0,1) random variables y_i, $i = 1,2,\ldots,N$.

2. Use the transformation

$$\nu_i = p\left[1 - \frac{1}{9p} + \frac{y_i}{3\sqrt{p}}\right]^3 \qquad [9.2.10]$$

to obtain ν_i, $i = 1,2,\ldots,N$, that are independent Pearson-type III distributed random variables.

Another approach for generating synthetic data is to use the empirical distribution of the given observations or some transformation thereof to simulate possible future occurrences. In Section 9.8.3, a case study is employed for explaining how the empirical distribution of the estimated innovations of a fitted ARMA model can be used to simulate the innovations.

9.3 WATERLOO SIMULATION PROCEDURE 1

Following the titles used by McLeod and Hipel (1978), the simulation procedures of Sections 9.3 and 9.4 are referred as *WASIM1 (Waterloo Simulation Procedure 1)* and WASIM2 (Waterloo Simulation Procedure 2), respectively. Both of these techniques avoid the introduction of bias into a simulated sequence by producing starting values that are randomly generated from the underlying stochastic process. WASIM1 consists of using the random shock or MA form of an ARMA model to simulate the starting values and then using the original ARMA model to simulate the remaining synthetic data.

Let w_t be a stationary w_t series for time $t = 1,2,\ldots,n$, to which an ARMA model is fitted as in [4.3.4] to produce the model

$$\phi(B)w_t = \theta(B)a_t \qquad [9.3.1]$$

where

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$$

is the nonseasonal autoregressive (AR) operator or polynomial of order p such that the roots of the characteristic equation $\phi(B) = 0$ lie outside the unit circle for nonseasonal stationarity and the ϕ_i, $i = 1,2,\ldots,p$, are the nonseasonal AR parameters;

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q$$

is the nonseasonal moving average (MA) operator or polynomial of order q such that the roots of $\theta(B) = 0$ lie outside the unit circle for invertibility and θ_i, $i = 1,2,\ldots,q$, are the nonseasonal MA parameters; the a_t's are identically independently distributed innovations with mean 0 and variance σ_a^2 [IID$(0,\sigma_a^2)$] and often the disturbances are assumed to be normally independently distributed [NID$(0,\sigma_a^2)$].

For simulation purposes, the zero-mean stationary seasonal ARMA model of Chapter 12 can be considered as a natural extension of the nonseasonal process. Models with a non-zero mean (or any other type of deterministic component) are simulated by first generating the corresponding zero-mean process and then adding on the mean component.

Suppose that the w_t's are expanded in terms of a pure MA process as in [3.4.18]. This is termed the random shock form of an ARMA process and is written as

$$w_t = \frac{\theta(B)}{\phi(B)}a_t = \psi(B)a_t = (1 + \psi_1 B + \psi_2 B^2 + \cdots)a_t \qquad [9.3.2]$$

where $\psi_0 = 1$. A method for calculating the ψ_i parameters from the AR and MA parameters is presented in Section 3.4.3. If an AR operator is present, $\psi(B)$ forms an infinite series and therefore must be approximated by the finite series

$$\psi(B) \approx 1 + \psi_1 B + \psi_2 B^2 + \cdots + \psi_{q'} B^{q'} \qquad [9.3.3]$$

It is necessary to choose q' such that $\psi_{q'+1}, \psi_{q'+2}, \cdots$, are all negligible. Since the model is stationary, this can be accomplished by selecting q' sufficiently large such that the error given below is kept as small as desired.

$$\gamma_0 - \sum_{i=0}^{q'} \psi_i^2 < \textit{error} \qquad [9.3.4]$$

where γ_o is the theoretical variance of a given ARMA process with $\sigma_a^2 = 1$ and is calculated using the algorithm of McLeod (1975) presented in Appendix A3.2; *error* is the chosen error level (ex. *error* $= 10^{-5}$).

To obtain a synthetic series of k observations, first generate $k + q'$ white noise terms $a_{-q'+1}, a_{-q'+2}, \ldots, a_0, a_1, a_2, \ldots, a_k$. Next, calculate

$$w_t = a_t + \psi_1 a_{t-1} + \psi_2 a_{t-2} + \cdots + \psi_{q'} a_{t-q'} \qquad [9.3.5]$$

where $t = 1,2, \ldots, r$, and $r = \max(p,q)$. The remaining w_t are easily determined from [9.3.1] as

$$w_t = \phi_1 w_{t-1} + \phi_2 w_{t-2} + \cdots + \phi_p w_{t-p} + a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q} \qquad [9.3.6]$$

where $t = r+1, r+2, \ldots, k$.

The use of [9.3.6] avoids the truncation error present in [9.3.3]. Nevertheless if an AR operator is present (i.e., $p > 0$), there will be some systematic error in the simulated data due to the approximation involved in [9.3.5]. However, this bias can be kept to a tolerable level by selecting the "error" term in [9.3.4] to have a specified minimum value. Of course, if the model is a pure MA(q) model, then set $q' = q$ and [9.3.5] is exact and can be utilized to generate all of the synthetic data.

An inherent advantage of the WASIM1 simulation technique is that the only restriction on the white noise terms is that they are IID$(0,\sigma_a^2)$. Although in many situations it is often appropriate to employ NID$(0,\sigma_a^2)$ innovations, this simulation method does not preclude considering other types of distributions. For instance, after modelling a relatively long hydrological time series, the residuals from the historical data could be used to form an empirical distribution function for generating the white noise. This approach is illustrated in the application in Section 9.8.3. In other situations, it may be warranted to simulate the white noise by employing *Johnson variates* (Johnson, 1949; Hill, 1976; Hill et al., 1976) which have been applied to hydrological data by authors including Sangal and Biswas (1970), Stedinger (1980) and Kottegoda (1987). Atkinson and Pearce (1976) discuss the computer generation of Beta, Gamma and normal

random variables, while Delleur et al. (1976) suggest some distributions which can be employed in hydrology. Techniques for generating independent normal and other random variables are pointed out in Section 9.2.3.

9.4 WATERLOO SIMULATION PROCEDURE 2

9.4.1 WASIM2 Algorithm

WASIM2 (Waterloo Simulation Procedure 2) is based upon a knowledge of the theoretical autocovariance or autocorrelation function (ACF). Following McLeod (1975), a method for calculating the theoretical ACF for any ARMA process is presented in Section 3.4.2 and Appendix A3.2. Simulation approaches based upon knowing the theoretical ACF's of the underlying processes can also be used for simulating using other types of stochastic models. For example, in Section 10.4.6, the theoretical ACF of a fractional Gaussian noise model is used in the simulation technique for that model.

Suppose that it is required to generate k terms of an ARMA(p,q) model with innovations that are NID(0,σ_a^2). The following simulation procedure is exact to simulate $w_1, w_2, \ldots, w_k$, for all stationary ARMA(p,q) models.

1. Obtain the theoretical autocovariance function γ_j for $j = 0,1,\ldots,p-1$ by using the algorithm of McLeod (1975) with $\sigma_a^2 = 1$. (See Section 3.4.2 and Appendix A3.2.)

2. Following the approach of Section 3.4.3, determine the random shock coefficients ψ_j for $j = 1,2,\ldots,(q-1)$ in [9.3.3].

3. Form the covariance matrix $\Delta\sigma_a^2$ of $w_p, w_{p-1}, \ldots, w_1, a_p, a_{p-1}, \ldots, a_{p-q+1}$.

$$\Delta = \begin{bmatrix} \left(\gamma_{i-j}\right)_{p\times p} & \left(\psi_{j-i}\right)_{p\times q} \\ \left(\psi_{i-j}\right)_{q\times p} & \left(\delta_{i,j}\right)_{q\times q} \end{bmatrix}_{(p+q)\times(p+q)} \qquad [9.4.1]$$

   In the above equation, the (i,j) element and dimension of each partitioned matrix are indicated. The values of $\delta_{i,j}$ are 1 or 0 according to whether $i=j$ or $i\neq j$, respectively. When $i - j < 0$, then $\gamma_{i-j} = \gamma_{j-i}$ and $\psi_{i-j} = 0$.

4. Determine the lower triangular matrix **M** by Cholesky decomposition (see Ralston (1965, p. 410), Healy (1968), or Hornbeck (1975)) such that

$$\Delta = \mathbf{M}\,\mathbf{M}' \qquad [9.4.2]$$

5. Generate $e_1, e_2, \ldots, e_{p+q}$, and $a_{p+1}, a_{p+2}, \ldots, a_k$ where the e_t and a_t sequences are NID(0,σ_a^2).

6. Calculate $w_1, w_2, \ldots, w_p$, from

$$w_{p+1-t} = \sum_{j=1}^{t} m_{t,j} e_j\,, \quad t = 1,2,\ldots,p \qquad [9.4.3]$$

   where $m_{t,j}$ is the t,j entry in the matrix **M**.

7. Determine $a_{p-q+1}, a_{p-q+2}, \ldots, a_p$, from

$$a_{p+1-t} = \sum_{j=1}^{p+t} m_{t+p,j} e_j, \quad t = 1,2,\ldots,q \qquad [9.4.4]$$

8. Obtain $w_{p+1}, w_{p+2}, \ldots, w_k$, using

$$w_t = \phi_1 w_{t-1} + \phi_2 w_{t-2} + \cdots + \phi_p w_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q}$$
$$t = p+1, p+2, \ldots, k \qquad [9.4.5]$$

9. If another series of length k is required then return to step 5.

For a particular ARMA model, it is only necessary to calculate the matrix **M** once, no matter how many simulated series are synthesized. Therefore, WASIM2 is economical with respect to computer time required, especially when many time series of the same length are generated.

Often the white noise disturbances can be assumed to be NID$(0,\sigma_a^2)$ and it is desirable to have as much accuracy as possible in order to eliminate bias. For this situation, the authors recommend using WASIM2 for a pure AR model or an ARMA process. When simulating a pure MA process with innovations that are NID$(0,\sigma_a^2)$, the WASIM1 and WASIM2 procedures are identical.

9.4.2 Theoretical Basis of WASIM2

In Step 3 of the WASIM2 algorithm, one forms the covariance matrix $\Delta\sigma_a^2$ of $w_p, w_{p-1}, \ldots, w_1, a_p, a_{p-1}, \ldots, a_{p-q+1}$, which are contained in a vector **W**. Next, one determines the lower triangular matrix **M** for Δ in [9.4.2]. Following Steps 6 and 7, the starting values contained in the vector **W** can be generated using

$$\mathbf{W} = \mathbf{M}\,\mathbf{e} \qquad [9.4.6]$$

where the e_t's contained in the vector **e** are NID$(0,\sigma_a^2)$. In order to simulate exactly the starting values contained in **W**, the covariance matrix of **W** must be $\Delta\sigma_a^2$. This can be easily proven as follows:

$$\begin{aligned} Var[\mathbf{M}\,\mathbf{e}] &= E[\mathbf{M}\,\mathbf{e}(\mathbf{M}\,\mathbf{e})^T] \\ &= E[\mathbf{M}\,\mathbf{e}\,\mathbf{e}^T\mathbf{M}^T] = \mathbf{M}Var(\mathbf{e})\mathbf{M}^T \\ &= \sigma_a^2\mathbf{M}\,\mathbf{M}^T = \Delta\sigma_a^2 \end{aligned} \qquad [9.4.7]$$

where σ_a^2 is a diagonal matrix for which each diagonal entry is σ_a^2.

9.4.3 ARMA(1,1) Simulation Example

The purpose of this section is to show in detail for a specific model how calculations are made in the WASIM2 algorithm. Suppose one wishes to simulate 10 values with an ARMA(1,1) model using WASIM2. For convenience, assume that the mean of the w_t series in [9.3.1] is zero. If the mean were nonzero, it could be added to each of the simulated values.

The calculations for an ARMA(1,1) model using WASIM2 are as follows:

1. For an ARMA(1,1) model $p = 1$ and, hence, one only has to calculate γ_0 at step 1. From [3.4.16]

$$\gamma_0 = \frac{1 + \theta_1^2 - 2\phi_1\theta_1}{1 - \phi_1^2}\sigma_a^2$$

Letting $\sigma_a^2 = 1$

$$\gamma_0 = \frac{1 + \theta_1^2 - 2\phi_1\theta_2}{1 - \phi_1^2}$$

2. Using the identity in [3.4.21] one can calculate the random shock coefficients. Only $\psi_0 = 1$ is required for an ARMA(1,1) model.

3. The matrix Δ in [9.4.1] is

$$\Delta = \begin{bmatrix} \gamma_0 & \psi_0 \\ \psi_0 & \delta_{0,0} \end{bmatrix} = \begin{bmatrix} \gamma_0 & 1 \\ 1 & 1 \end{bmatrix}$$

Then $\Delta\sigma_a^2$ is the covariance matrix for (w_1, a_1).

4. In this step, one obtains the Cholesky decomposition matrix $\mathbf{M}$ for Δ such that

$$\Delta = \mathbf{M}\,\mathbf{M}^T$$

For the case of an ARMA(1,1) model

$$\Delta = \begin{bmatrix} \gamma_0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} m_{11} & 0 \\ m_{21} & m_{22} \end{bmatrix}\begin{bmatrix} m_{11} & m_{21} \\ 0 & m_{22} \end{bmatrix}$$

or

$$\begin{bmatrix} \gamma_0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} m_{11}^2 & m_{11}m_{21} \\ m_{21}m_{11} & m_{21}^2 + m_{22}^2 \end{bmatrix}$$

By equating (i,j) entries in the matrices on the left and right hand sides of the above equations, one can calculate m_{11}, m_{21} and m_{22}. In particular, for the (1,1) element:

$$m_{11}^2 = \gamma_0$$

Therefore,

$$m_{11} = \sqrt{\gamma_0}$$

For the (1,2) entry:

$$m_{11}m_{21} = 1$$

Therefore,

$$m_{21} = \frac{1}{m_{11}} = \frac{1}{\sqrt{\gamma_0}}$$

for the (2,2) element:

$$m_{21}^2 + m_{22}^2 = 1$$

Therefore,

$$m_{22}^2 = 1 - m_{21}^2 = 1 - \frac{1}{\gamma_0}$$

Therefore,

$$m_{22} = \sqrt{1 - \frac{1}{\gamma_0}}$$

Hence,

$$\mathbf{M} = \begin{bmatrix} \sqrt{\gamma_0} & 0 \\ \frac{1}{\sqrt{\gamma_0}} & \sqrt{1 - \frac{1}{\gamma_0}} \end{bmatrix}$$

5. The techniques of Section 9.2.3 can be used to generate (e_1,e_2) and $(a_2,a_3,\ldots,a_{10})$ where the sequences are NID$(0,\sigma_a^2)$.

6. The starting value w_1 is calculated using [9.4.3] as

$$w_1 = m_{11}e_1 = \sqrt{\gamma_0}e_1$$

7. The initial value for a_1 is found using [9.4.4] to be

$$a_1 = m_{21}e_1 + m_{22}e_2 = \frac{1}{\sqrt{\gamma_0}}e_1 + \left(\sqrt{1 - \frac{1}{\gamma_0}}\right)e_2$$

8. Use the given definition of the ARMA(1,1) model in [9.4.5] to get $w_2,w_3,\ldots,w_{10}$. More specifically,

$$w_2 = \phi_1 w_1 + a_2 - \theta_1 a_1$$

where the starting values for w_1 and a_1 are calculated in steps 6 and 7, respectively. The values for a_2 and also $a_3, a_4, \ldots, a_{10}$ are determined in step 5. Next

$$w_3 = \phi_1 w_2 + a_3 - \theta_1 a_2$$

where w_2 is calculated in the previous iteration and a_3 and a_2 are generated at step 5. By following the same procedure

$$w_4 = \phi_1 w_3 + a_4 - \theta_1 a_3$$

. .

. .

. .

$$w_{10} = \phi_1 w_9 + a_{10} - \theta_1 a_9$$

If w_t has a mean level, this can be added to the above generated values. Notice that the starting values w_1 and a_1 are randomly generated from the underlying ARMA(1,1) process. Therefore, the simulated sequence is not biased because of fixed starting values.

9.5 SIMULATION OF INTEGRATED MODELS

9.5.1 Introduction

As discussed in 2.4 and elsewhere in the book, for annual geophysical time series of a moderate length (perhaps a few hundred years), it is often reasonable to assume that a stationary model can adequately model the data. In Section 10.6 and also in Hipel and McLeod (1978), for example, stationary ARMA models are fitted to 23 time series which are measured from six different natural phenomena. Nevertheless, certain types of time series that are used in water resources could be nonstationary. The average annual cost of hydro-electric power and the total annual usage of water-related recreational facilities constitute two types of measurable processes which possess mean levels and variances that could change significantly over time. Other examples of nonstationary annual series are given in Section 4.3.3. In general, time series that reflect the socio-economic aspects of water resources planning may often be nonstationary, even over a short time span. Consequently, in certain situations it may be appropriate to incorporate a nonseasonal differencing operator into the nonseasonal model in order to account for the nonstationarity by following the approach of Section 4.3.

As explained in Chapter 12, if a seasonal ARIMA or SARIMA model is fit to seasonal data, usually both nonseasonal and seasonal differencing are required to account for the nonstationarity. Consider the case of average monthly observations. If the monthly mean and perhaps variance change from one year to the next for each specific month, then fitting a nonstationary SARIMA model to the data may prove to be reasonable. For example, the average monthly water demand for large cities tends to increase from year to year for each month. For the aforementioned situations, the simulation procedures for integrated models presented in this section could be useful.

As pointed out in Part VI, when considering seasonal hydrological data, such as average monthly riverflows, the individual monthly averages may have constant mean values but the means vary from month to month. Consequently, the time series of all the given data is by definition nonstationary but it still may not be appropriate to employ a nonstationary SARIMA model to describe the data. Rather, the given natural time series is firstly deseasonalized to produce a stationary nonseasonal data set and subsequently a nonseasonal model is fit to the deseasonalized data. For example, prior to fitting a nonseasonal ARMA model to the data, it is a common procedure to standardize average monthly riverflow time series to eliminate seasonality (see Chapter 13). The WASIM1 and WASIM2 simulation procedures of this chapter can be used with the deseasonalized models of Chapter 13 and the periodic models of Chapter 14. However,

for both of these types of seasonal models, one does not have to difference the data and, hence, the methods of Section 9.5 do not have to be used when using them for simulation.

Nonseasonal ARIMA and SARIMA models are presented in Chapters 4 and 12, respectively. For completeness of presentation of this simulation chapter, simulating with both types of models is discussed below.

9.5.2 Algorithms for Nonseasonal and Seasonal ARIMA Models

Although caution should be exercised when modelling nonstationary data, it is evident that situations may arise when it is suitable to invoke *differencing*. Any seasonal ARIMA model from [12.2.7] that contains a differencing operator is termed an *integrated* model. Suppose that it is required to simulate k values of $z_t^{(\lambda)}$ by using an integrated process. A stationary w_t series is related to the nonstationary $z_t^{(\lambda)}$ series by the equation

$$w_t = \nabla^d \nabla_s^D z_t^{(\lambda)}, \quad t = d'+1, d'+2, \ldots, k \qquad [9.5.1]$$

where s is the seasonal length ($s = 12$ for monthly data); $\nabla^d = (1-B)^d$ is the nonseasonal differencing operator of order d in [4.3.3] to produce nonseasonal stationarity of the dth differences and usually $d = 0$, 1 *or* 2; $\nabla_s^D = (1-B^s)^D$ is the seasonal differencing operator of order D in [12.2.3] to produce seasonal stationarity of the Dth differenced data and usually $d = 0$, 1, or 2, and for nonseasonal data $D = 0$; $d' = d + sD$.

Because of the differencing in [9.5.1], the d' initial values $w_1, w_2, \ldots, w_{d'}$, which determine the "current level" of the process, are assumed known. Given the d' initial values, the time series integration algorithm forms the integrated series $z_t^{(\lambda)}$ for $t = d'+1, d'+2, \ldots, k$. The integrated series is derived theoretically from the relationship

$$z_t^{(\lambda)} = S^d S_s^D w_t \qquad [9.5.2]$$

where $S = \nabla^{-1} = 1 + B + B^2 + \cdots$, is the nonseasonal summation operator; $S_s = \nabla_s^{-1} = 1 + B^s + B^{2s} + \cdots$, is the seasonal summation operator.

When employing [9.5.2] to obtain an integrated series, the methods of the previous sections are utilized to determine the w_t sequence. Then the integration algorithm that is developed presently in this section, is used to evaluate [9.5.2]. The situation where it is required to simulate data from a nonseasonal model containing a differencing operator, is first considered. This is followed by a discussion of the generation of synthetic data from a general seasonal model that possesses a seasonal differencing operator and perhaps also a nonseasonal differencing operator.

Nonseasonal Model:

The integration algorithm for a nonseasonal ARIMA model (i.e. $s = D = 0$) is as follows:

For $i = 1, 2, \ldots, d$:

1. Determine the starting value $\nabla^{d-i} z_t^{(\lambda)}$ by differencing the given initial values $z_1^{(\lambda)}, z_2^{(\lambda)}, \ldots, z_d^{(\lambda)}$.

2. Calculate $\nabla^{d-i} z_t^{(\lambda)}$ for $t = d+1, d+2, \ldots, k$, by employing the identity

$$\nabla^{d-i} z_t^{(\lambda)} = \nabla^{d+1-i} z_t^{(\lambda)} + \nabla^{d-i} z_{t-1}^{(\lambda)} \qquad [9.5.3]$$

Seasonal Model:

For a seasonal model, the integration algorithm is subdivided into two parts. The first stage consists of performing the nonseasonal integration.

For $i = 1,2, \ldots, d$:

1. Determine the starting value $\nabla^{d-i} \nabla_s^D z_{d'}'$, by differencing the given initial values $z_1^{(\lambda)}, z_2^{(\lambda)}, \ldots, z_{d'}^{(\lambda)}$.
2. Calculate $\nabla^{d-i} \nabla_s^D z_t^{(\lambda)}$ for $t = d'+1, d'+2, \ldots, k$, by using the equation

$$\nabla^{d-i} \nabla_s^D z_t^{(\lambda)} = \nabla^{d+i-i} \nabla_s^D z_t^{(\lambda)} - \nabla^{d-i} \nabla_s^D z_{t-1}^{(\lambda)} \qquad [9.5.4]$$

In the second stage the seasonal integration is performed. For $i = 1,2, \ldots, D$:

1. Determine the starting values $\nabla_s^D z_t^{(\lambda)}$ for $t = d', d'-1, \ldots, d' = s$, by differencing the given initial values $z_1^{(\lambda)}, z_2^{(\lambda)}, \ldots, z_{d'}^{(\lambda)}$.
2. Calculate $\nabla_s^{D-i} z_t^{(\lambda)}$ for $t = d'+1, d'+2, \ldots, k$ by using the equation

$$\nabla_s^{D-i} z_t^{(\lambda)} = \nabla_s^{D-i+1} z_t^{(\lambda)} + \nabla_s^{D-i} z_{t-s}^{(\lambda)} \qquad [9.5.5]$$

9.6 INVERSE BOX-COX TRANSFORMATION

As shown in Figure 9.1.1, before fitting a time series model to a given time series, one may wish to transform the data using the Box-Cox transformation in [3.4.30]. The main purposes of the Box-Cox transformation are to make the residuals of the fitted ARMA or ARIMA model (see Section 3.4.5) to be normally distributed and homoscedastic (have constant variance). Subsequent to simulating data sequences using the fitted model and following the techniques from the previous sections of this chapter, one must take the inverse Box-Cox transformation in order to obtain synthetic data that have the same units as the original time series.

From [3.4.30] the Box-Cox transformation of the original z_t series is

$$z_t^{(\lambda)} = \begin{cases} \lambda^{-1}[(z_t + c)^{\lambda} - 1], & \lambda \neq 0 \\ \ln(z_t + c), & \lambda = 0 \end{cases} \qquad [9.6.1]$$

where the constant c is chosen to be just large enough to cause all of the entries of the z_t series to be positive. When the data are nonstationary, one may also wish to difference the $z_t^{(\lambda)}$ series to obtain the w_t series in [4.3.3]. Of course, if the z_t or $z_t^{(\lambda)}$ series are already stationary, the w_t series is the same as $z_t^{(\lambda)}$. Whatever the case, after following the methods of the previous sections of Chapter 9 to obtain the simulated $z_t^{(\lambda)}$ sequences, one must take the inverse Box-Cox transformation to get the synthetic z_t data. The inverse Box-Cox transformation is written as

$$z_t = \begin{cases} [\lambda z_t^{(\lambda)} + 1]^{1/\lambda} - c, & \lambda \neq 0 \\ \exp[z_t^{(\lambda)}] - c, & \lambda = 0 \end{cases} \qquad [9.6.2]$$

where $z_t^{(\lambda)}$ stands for the simulated sequence obtained directly from the ARMA or ARIMA model. If the $z_t^{(\lambda)}$ sequence possesses a mean level, one can add the transformed mean to each of the generated z_t values calculated using [9.6.2].

9.7 WATERLOO SIMULATION PROCEDURE 3

9.7.1 Introduction

In simulation studies, one usually employs a calibrated ARMA model to simulate possible future sequences of the time series to which the model is fitted. One could, of course, assume a given theoretical model having specified parameter values and then use this model in a simulation study. However, when fitting the model to the series, one obtains maximum likelihood estimates (MLE's) and standard errors (SE's) for the model parameters (see Section 6.2 and Appendices A6.1 and A6.2). Because the time series used to calibrate the ARMA model is only one finite realization of the underlying stochastic process that generated this observed sequence, the population values of the model are not known. The MLE's of the model parameters constitute the best estimates of the population values given the available information. The uncertainties or variations of these estimates are reflected by their SE's. As explained in Appendix A6.2, the SE's are calculated as the square roots of the diagonal entries in the variance-covariance matrix for the estimated parameters.

The *WASIM3 (Waterloo Simulation Procedure 3) algorithm* can be used in simulation studies where it is required to incorporate *parameter uncertainty* into the analysis. Suppose that it is necessary to generate k' synthetic traces of length k. When generating each series of length k, different values of the model parameters are randomly selected if WASIM3 is employed. The WASIM3 procedure is explained in this book only for a nonseasonal ARMA model, since extension to the seasonal case is straightforward. In terms of Figure 9.1.1, the WASIM3 algorithm is used prior to taking the inverse Box-Cox transformation discussed in Section 9.6. Following the presentation of the WASIM3 algorithm in the next subsection, it is explained how parameter uncertainty is incorporated into reservoir design. Section 9.7.4 discusses how practitioners can deal with model uncertainty.

9.7.2 WASIM3 Algorithm

Suppose that the historical time series containing N values is modelled as an ARMA(p,q) model as in [9.3.1] that has an estimated mean level of $\hat{\mu}$. The Gaussian white noise residuals have an estimated variance denoted by $\hat{\sigma}_a^2$. Let the vector of the estimated ARMA parameters be given by

$$\hat{\beta} = (\hat{\phi}_1, \hat{\phi}_2, \ldots, \hat{\phi}_p, \hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_q) \qquad [9.7.1]$$

(see Section 6.2). The vector of the true model parameters is denoted by

$$\beta = (\phi_1, \phi_2, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q) \qquad [9.7.2]$$

The mean level of the true model is μ while the variance of the white noise is σ_a^2.

If a non-informative prior distribution is used for the model parameters, then β, μ and σ_a^2 are approximately independent with posterior distributions given by

$$\beta \approx N(\hat{\beta}, V_{\hat{\beta}}) \qquad [9.7.3]$$

where $V_{\hat{\beta}}$ is the estimated variance covariance matrix of β which is usually calculated at the estimation stage of model development (see Appendix A6.2) and N means normally distributed.

$$\mu \approx N\left(\hat{\mu}, \left(\frac{1 - \hat{\phi}_1 - \hat{\phi}_2 - \cdots - \hat{\phi}_p}{1 - \hat{\theta}_1 - \hat{\theta}_2 - \cdots - \hat{\theta}_q}\right)^{-2}, \frac{\hat{\sigma}_a^2}{N}\right) \qquad [9.7.4]$$

$$\sigma_a^2 \approx N\left(\hat{\sigma}_a^2, \frac{\hat{\sigma}_a^4}{2}\right) \qquad [9.7.5]$$

The findings given in [9.7.3] to [9.7.5] are based upon large sample theory. Nevertheless, these results can be used to obtain some idea of the importance, if any, of parameter uncertainty in a particular situation. It should be noted that if an informative prior distribution were used, the variances of the parameters would be less and hence the parameter uncertainty would also decrease.

The following algorithm for WASIM3 can be used to allow for parameter uncertainty when k' series of length k are to be generated from an ARMA(p,q) model.

1. Set $i = 1$.
2. Randomly generate values for β, μ and σ_a^2 using the posterior distributions given in [9.7.3], [9.7.4] and [9.7.5], respectively. Denote the generated parameter values as β_i, μ_i and $\sigma_{a,i}^2$. Refer to the book by Janson (1966) for a method to obtain random values from a multivariate normal distribution.
3. Use WASIM2 (or WASIM1) for an ARMA(p,q) process with parameters β_i, μ_i, and $\sigma_{a,i}^2$, to simulate a synthetic series of length k that is represented by $z_1^{(i)}, z_2^{(i)}, \ldots, z_k^{(i)}$. If the model contains a Box-Cox transformation, the inverse transformation in [9.6.2] is required.
4. Set $i = i + 1$. If $i \leq k'$ then repeat steps 2 and 3 to obtain another possible realization of the time series. When $i > k'$, the WASIM3 procedure is terminated.

9.7.3 Parameter Uncertainty in Reservoir Design

In this section, an algorithm is presented for estimating the expected utility of a *reservoir design* given the specified ARMA(p,q) model for the riverflow data and a posterior distribution $p(\beta, \mu, \sigma_a^2)$ for the parameters. For a given riverflow time series $z_1, z_2, \ldots, z_k$, and a particular reservoir design D, the (vector-valued) net benefit function is given by

$$NB = NB(z_1, \ldots, z_k; D) \qquad [9.7.6]$$

and the utility is

$$U = U(NB) \qquad [9.7.7]$$

The expected utility of D is then given by

$$u(D) = E\left\{U(NB(z_1,z_2, \ldots, z_k; D))\right\}$$

$$= \int_{z_1 z_2}\int \cdots \int_{z_k \beta_1 \beta_2}\int\int\int \cdots \int_{\beta_{p+q} \mu \sigma_a^2}\int\int\int U(NB(z_1,z_2, \ldots, z_k; D))$$

$$p(z_1,z_2, \ldots, z_k|\beta,\mu,\sigma_a^2)p(\beta,\mu,\sigma_a^2)$$

$$dz_1 dz_2 \cdots dz_k d\beta_1 d\beta_2 \cdots d\beta_{p+q} d\mu d\sigma_a^2 \qquad [9.7.8]$$

The best design, D_o, maximizes the value of $u(D)$.

After an ARMA model is fit to the given time series of historical riverflows, the following algorithm may be used to estimate $u(D)$ and a confidence interval (or Bayesian probability interval) for $u(D)$.

1. Set $i = 1, T_1 = 0, T_2 = 0$. Let k' be the number of series of length k that are to be generated. For example, k' may have a value of 10,000.
2. Generate a synthetic time series, $z_1^{(i)}, z_2^{(i)}, \ldots, z_k^{(i)}$, using the WASIM3 algorithm.
3. Calculate $u_i = U(NB(z_1^{(i)}, z_2^{(i)}, \ldots, z_k^{(i)}; D))$, set $T_1 = T_1 + u_i$, and set $T_2 = T_2 + u_i^2$.
4. Set $i = i + 1$ and go to step 2 if $i \leq k'$. Go to step 5 if $i > k'$.
5. Set

$$\bar{u} = \frac{1}{k'} T_1 \qquad [9.7.9]$$

and let

$$S_{\bar{u}} = \left[\left(\frac{1}{k'} T_2 - \bar{u}^2\right)/k'\right]^{1/2} \qquad [9.7.10]$$

The calculated $\bar{u}$ provides an estimate of $u(D)$ and a 95% confidence interval (or Bayesian probability interval) for $u(D)$ is given by $\bar{u} \pm 1.96 S_{\bar{u}}$. Although the aforesaid algorithm is explained for a nonseasonal ARMA(p,q) model, the same approach is valid for seasonal models. The number of generated synthetic traces (i.e. k') can be increased if more accuracy is required or decreased when less accuracy is needed.

9.7.4 Model Uncertainty

In the synthetic hydrology approach to reservoir design and operation, an ARMA model may be fit to a historical riverflow time series and then used to simulate other possible realizations of the riverflows. Two sources of possible error may arise. The model selected may be inappropriate or the estimated parameters may be inaccurate. The procedures of Part III emphasize techniques for selecting an appropriate model followed by efficient parameter estimation and diagnostic checking for possible model inadequacies. It is thus reasonable to suppose that the selected model is at least approximately valid and that the best possible estimates are obtained for the model parameters by using the method of maximum likelihood (see Section 6.2). On the other hand, if a possible inappropriate model is fit to the data and no checks of model adequacy are done, a seriously inadequate model may be chosen. It is demonstrated in Section 10.4.6, for example, that the use of fractional Gaussian noise models may give poor fits to annual riverflow time series when compared to ARMA models. If the methods of Part III are used with a hydrologic time series of at least 50 observations, the selection of an inappropriate model is not likely to occur. The reader may wish to refer to Section 5.2 for further discussions of modelling philosophies and different kinds of uncertainties.

Given that the best possible model is identified for fitting to a series and efficient parameter estimates are obtained, the WASIM3 algorithm can be used for simulation purposes in order to take parameter uncertainty into account. This parameter uncertainty is caused by the finite sample length of the time series to which the model is fitted.

9.8 APPLICATIONS

9.8.1 Introduction

Three applications are presented to illustrate the advantages and usefulness of the simulation procedures presented in this chapter. The first example demonstrates that the employment of WASIM2 in simulation studies avoids bias that is due to fixed starting values. In the second application, it is shown how the model residuals from the historical data can be used in conjunction with WASIM1 for generating synthetic data. Finally, the third example demonstrates how parameter uncertainty can be incorporated into a simulation study by using WASIM3.

9.8.2 Avoidance of Bias in Simulation Studies

The *rescaled adjusted range (RAR)* and the *Hurst coefficient K* defined in [10.2.9] and [10.3.4], respectively, are two statistics that are important in problems related to the Hurst phenomenon. In Chapter 10 the controversies surrounding the *Hurst phenomenon* are presented and it is demonstrated that ARMA processes are superior to fractional Gaussian noise models for explaining the Hurst phenomenon as well as modelling annual hydrological time series. In particular, it is shown in Section 10.6 that ARMA models statistically preserve the historical RAR or equivalently K. Accordingly, ARMA models are important tools for utilization in hydrological studies.

If the underlying process is an ARMA model, it can be shown theoretically that the RAR is a function only of the sample size and the AR and MA parameters (Hipel, 1975, Appendix B). In Section 10.6, it is demonstrated how to obtain the *empirical cumulative distribution function (ECDF)* for the RAR when the generating process is a specified stochastic model. In particular,

consider the ECDF for a Markov model (i.e. ARMA(1,0) process) with the AR parameter having a value of 0.7. When the WASIM2 technique is employed to generate 10,000 sequences of length 30, the value of the 0.95 quantile for the ECDF of the RAR is 12.15. The 95% confidence interval for this value is calculated to be from 12.09 to 12.19 (see Conover (1971, p. 111) for the method to calculate the confidence interval for a quantile).

If random realizations of the stochastic process are not utilized as starting values, systematic bias can be introduced into a simulation study such as the development of the ECDF for the RAR. For the Markov model with an AR parameter having a magnitude of 0.7, 10,000 sequences of length 30 were generated and for each sequence the mean value of zero was used as a starting value. In addition, exactly the same disturbances that were utilized in the simulation study using WASIM2, were employed for the biased study. The value of the 0.95 quantile for the biased ECDF of the RAR is 12.01. The 95% confidence interval for this quantile value is from 11.97 to 12.05. Notice that the confidence interval for the biased result does not intersect with the corresponding interval for the unbiased study. Consequently, fixed initial values should not be used in the development of the ECDF for a specified statistic and generating mechanism.

9.8.3 Simulation Studies Using the Historical Disturbances

When using WASIM1, it is not necessary to assume that the model residuals are NID(0,σ_a^2). In fact, it is not required to determine any theoretical distribution for the disturbances to follow. Rather, in certain situations it may be advantageous to use the residuals from the historical data to form an empirical distribution for generating the innovations. For example, when a relatively large sample is available, it may be desirable to use the empirical distribution of the residuals for simulation studies, no matter what theoretical distribution the empirical results may most closely resemble. In other instances, it may be difficult to determine which theoretical distribution to fit to the disturbances and, consequently, it may be profitable to employ the empirical distribution of the residuals. However, it should be pointed out that when the historical disturbances are employed, it is not possible to have a generated disturbance that is more extreme than any of the calculated residuals. Nevertheless, because of the form of the difference equation for an ARMA model in [9.3.1] that is fit to correlated data, it is possible that values of the generated data may be more extreme than those in the given time series.

A riverflow time series is considered to demonstrate how the empirical distribution for the residuals can be used in practice. The average annual flows of the Gota River in Sweden are available from 1807 to 1957 in a paper by Yevjevich (1963). A model is fit to this data by following the identification, estimation, and diagnostic check stages of model construction presented in detail in Part III of the book. The identification stage reveals that it may be appropriate to estimate an AR model of order two. The parameter estimates and corresponding SE's listed in Table 9.8.1 were calculated using the unconditional sum of squares method referred to in Section 6.2.3. At the estimation stage, the white noise residuals $\{\hat{a}_1,\hat{a}_2,\ldots,\hat{a}_{150}\}$ are determined using the backforecasting technique of Box and Jenkins (1976, Ch. 7). Diagnostic checks performed on the residuals confirm that the modelling assumptions are satisfied. In particular, by calculating confidence limits for the residual autocorrelation function using the technique of Section 7.3.2, the residuals are shown to be white noise.

Table 9.8.1. Parameter estimates for an ARMA(2,0) model fit to the Gota River data.

Parameters	MLE's	SE's
ϕ_1	0.591	0.079
ϕ_2	-0.274	0.078

To obtain synthetic data using the Gota model, the WASIM1 method is employed and the white noise terms are chosen by selecting at random an element of the set $\{\hat{a}_1,\hat{a}_2,\ldots,\hat{a}_{150}\}$. After one of the historical innovations is utilized, it is put back into the set of historical disturbances. Therefore, selection is done with replacement and this method is equivalent to using the empirical distribution of the residuals for the generation of the white noise terms.

As an example of a simulation study using the Gota model, consider the development of the ECDF for the Hurst coefficient K. The historical disturbances and the WASIM1 technique are used to generate 10,000 sequences, where each sequence contains 150 values. By calculating K for each of the 10,000 traces, the ECDF for K can be obtained as shown in Table 9.8.2 for a series length which is the same as the historical time series.

The historical value of K for the Gota River is calculated to be 0.689. Notice that the observed K value does not lie in the tails of the ECDF for K in Table 9.8.2. The probability that K for the Gota model is greater than the historical K is 0.281. In Section 10.6, this procedure is applied to 23 natural time series and by invoking a particular statistical test, it is demonstrated that ARMA models do statistically preserve the Hurst coefficient K or equivalently the RAR.

Table 9.8.2. ECDF of K for the Gota model.

Quantiles	Values of K for Empirical White Noise
0.025	0.556
0.050	0.571
0.100	0.590
0.200	0.613
0.300	0.630
0.400	0.645
0.500	0.658
0.600	0.671
0.700	0.686
0.800	0.703
0.900	0.725
0.950	0.744
0.975	0.757

9.8.4 Parameter Uncertainty in Simulation Experiments

An average annual riverflow series having a length of 96 years is modelled to show how parameter uncertainty can be brought into a practical simulation study. The yearly riverflows of the Mississippi River at St. Louis, are available from 1861 to 1957 in an article by Yevjevich (1963). By following the three stages of model development, the best process for modelling the Mississippi flows is found to be an ARMA(0,1) model. The MLE for the MA parameter θ_1 is -0.306 with a SE of 0.097.

By using WASIM1 (or equivalently WASIM2), the Mississippi model is employed to generate 10,000 series of length 96. The RAR is calculated for each of the 10,000 traces. The expected value or mean of the RAR for the 10,000 series is 13.439 with a standard deviation of 0.030.

The Mississippi model is used with WASIM3 to generate another 10,000 series of length 96. The innovations are different than those used for the simulation study with WASIM1. For each trace of length 96, the value of the MA parameter used in WASIM3 is determined by the equation

$$\theta_1 = -0.306 + 0.097\varepsilon_t \qquad [9.8.1]$$

where $t = 1,2,3,\ldots,10{,}000$; $\varepsilon_t \sim NID(0,1)$. Because the RAR is not a function of the mean level of the process or the variance of the model residuals, it is only necessary to randomly vary the MA parameter for this particular simulation study. The expected value of the RAR for the 10,000 synthetic data sets is 13.443 with a standard deviation of 0.031. A comparison of the results for the simulation experiment using a constant MA parameter with those utilizing a varying model parameter, reveals that there is no significant difference between the two expected values of the RAR. Hence, for this particular study, parameter uncertainty is not a crucial factor.

9.9 CONCLUSIONS

Improved simulation procedures are available for generating synthetic traces from ARMA and ARIMA models. Because random realizations of the underlying stochastic process are used as starting values, bias is not introduced into the simulated sequences. Furthermore, these techniques can be used in conjunction with models containing differencing operators or data that has been transformed by a Box-Cox transformation. The overall procedure for utilizing the simulation techniques are depicted in Figure 9.1.1 while detailed explanations are presented in Sections 9.2 to 9.7. Three representative applications of the simulation methods are given in Section 9.8.

If the WASIM1 method of Section 9.3 is utilized, it is not necessary that the distribution of the residuals be Gaussian. As shown by an example in Section 9.8.3, the empirical distribution of the residuals can be used for generation purposes. In addition, WASIM1 is exact for a pure MA process. On the other hand, the WASIM2 technique of Section 9.4 is an exact simulation procedure for any ARMA model. The only restriction with WASIM2 is that the residuals are NID$(0,\sigma_a^2)$.

When incorporating parameter uncertainty into a simulation study, the WASIM3 procedure of Section 9.7 is the proper method to implement. If it is deemed necessary to consider parameter uncertainty in reservoir design, one can employ the algorithm given in Section 9.7.3 for linking WASIM3 with the design problem. As discussed in Section 9.7.4, to circumvent difficulties

with model uncertainty, it is recommended that a proper ARMA model be fit to the given data set by following three stages of model development presented in Part III.

In Section 9.5, it is explained how the simulation techniques can be used with integrated models. The simulation methods can be extended for use with the three types of seasonal models of Part VI. For example, by writing the seasonal ARIMA model in the unfactored form shown in [12.2.11], one can directly employ the simulation methods of Chapter 9.

As explained in Section 9.7.3, simulation can be used in reservoir design. Simulation can also be employed for studying the theoretical properties of a given type of stochastic model. In Section 10.6, simulation is employed to demonstrate that ARMA models preserve statistically two important historical statistics called the Hurst coefficient and the rescaled adjusted range (also see Section 9.8). This forms the basis for the explanation of what is called the Hurst phenomenon. The Hurst phenomenon and related developments in long memory modelling are presented in Chapters 10 and 11, respectively, in Part V of the book.

PROBLEMS

9.1 Explain why it is not "completely correct" to employ statistical tests for checking the statistical properties of sequences generated by a random number generator.

9.2 Briefly outline at least two statistical tests for determining whether or not a random number generator produces independent observations. List the relative merits and disadvantages of these tests.

9.3 Briefly describe at least two statistical tests for checking if a random number generator produces uniformly distributed variables. Compare the relative advantages of these tests.

9.4 In Section 9.2.3, a universal random variable generator is presented for transforming independent uniform random variables to independent random variables following any required distribution. Prove that this algorithm is correct.

9.5

(a) Provide a numerical example to demonstrate how the mixed linear congruential random number generator works.

(b) Use a numerical illustration to explain how the steps of the multiplicative linear congruential random number generator are carried out.

9.6 Describe detailed guidelines regarding the choice of the coefficients a, c and m in the linear congruential random number generator in [9.2.2].

9.7 In Section 9.2.3, references are given for the Marsaglia-Bray algorithm which can be used to generate random variables that are NID(0,1). After referring to these references, describe the main steps in this algorithm. Discuss the main advantages and drawbacks of the Marsaglia-Bray algorithm.

9.8 Three algorithms for generating NID(0,1) random variables from independent uniformly distributed random variables are the central-limit algorithm, Teichroew method and Box-Muller generator (see, for example, Knuth (1969)). Briefly describe how each algorithm works and point out any overlap in the techniques. Compare their relative advantages and disadvantages from both theoretical and computational viewpoints.

9.9 Describe an approach for generating independent gamma random variables.

9.10 Draw a flow chart to outline how one can employ simulation in reservoir design.

9.11 Suppose that one wishes to simulate 10 values from an ARMA(2,2) model using WASIM1. Write down how each of these 10 values are calculated using the WASIM1 algorithm.

9.12

(a) Suppose that one wishes to use WASIM2 to simulate 10 values from an ARMA(2,1) model. Show all the calculations for generating this data.

(b) Prove that the random starting values in part (a) are from an ARMA(2,1) process.

9.13 Refer to appropriate references for explaining how to simulate from a multivariate normal distribution. Describe in detail how this is done within the WASIM3 algorithm.

9.14 Suppose that one wishes to simulate 10 values of a series to which an ARIMA(1,1,1) model was fitted to the square roots of the observations. Write down all of the detailed calculations for producing simulated values in the untransformed domain.

9.15

(a) Using annual time series from your field of interest, carry out a simulation experiment to demonstrate that ARMA models statistically preserve the historical autocorrelation function at lag 1.

(b) Incorporate parameter uncertainty into the simulation experiment executed in part (a).

REFERENCES

DATA SET

Yevjevich, V. M. (1963). Fluctuation of wet and dry years, 1, Research data assembly and mathematical models. Hydrology Paper no. 1, Colorado State University, Fort Collins, Colorado.

GENERATING INDEPENDENT RANDOM VARIABLES

Atkinson, A. C. and Pearce, M. C. (1976). The computer generation of beta, gamma and normal random variables. *Journal of the Royal Statistical Society, Series A*, 139(4):431-448.

Box, G. E. P. and Muller, M. E. (1958). A note on the generation of random normal deviates. *Annals of Mathematical Statistics*, 29:610-611.

Dieter, V. (1972). Statistical interdependence of pseudo-random numbers generated by the linear congruential method. In Zaremba, S., editor, *Applications of Number Theory to Numerical Analysis*, pages 289-318. Academic Press, New York.

Hill, I. D. (1976). Algorithm AS100, normal-Johnson and Johnson-normal transformations. *Applied Statistics*, 25(2):190-192.

Hill, I. D., Hill, R. and Holder, R. L. (1976). Algorithm AS99, fitting Johnson curves by moments. *Applied Statistics*, 25(2):180-189.

Hull, T. and Dobell, A. (1962). Random number generators. *SIAM Review*, 4:230-254.

Janson, B. (1966). *Random Number Generators*. Victor Pettersons, Bokindustri, Akiebolag, Stockholm.

Johnson, N. (1949). Systems of frequency curves generated by methods of translation. *Biometrika*, 36:149-176.

Knuth, D. E. (1969). *The Art of Computer Programming*. Addison-Wesley, Reading, Massachusetts.

Lehmer, D. H. (1949). Mathematical methods in large scale computing units. In *Proceedings of the Symposium on Large Scale Digital Calculating Machinery*, pages 141-146. Harvard University Press.

Marsaglia, G. (1972). The structure of linear congruential sequences. In Zaremba, S., editor, *Applications of Number Theory to Numerical Analysis*, pages 249-286. Academic Press, New York.

Marsaglia, G. and Bray, T. A. (1964). A convenient method for generating normal variables. *SIAM Review*, 6:260-264.

Marsaglia, G., MacLaren, G. M. M. and Bray, T. A. (1964). A fast procedure for generating normal variables. *Communications of the Association of Computing Machinery*, 7:4-10.

von Neumann, J. (1951). Various techniques used in connection with random digits. *NBS Applied Mathematics Series*, (12):36-38.

Weckmann, B. A. and Hill, I. D. (1982). An efficient and portable pseudo-random number generator. *Applied Statistics*, 31:188-190.

Yakowitz, S. J. (1977). *Computational Probability and Simulation*. Addison-Wesley, Reading, Massachusetts.

HYDROLOGICAL SIMULATION RESEARCH

Box, G. E. P. and Jenkins, G. M. (1976). *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco, revised edition.

Delleur, J. W., Tao, P. C., and Kavvas, M. L. (1976). An evaluation of the practicality and complexity of some rainfall and runoff time series models. *Water Resources Research*, 12(5):953-970.

Hipel, K. W. (1975). Contemporary Box-Jenkins Modelling in Water Resources. PhD thesis, University of Waterloo, Waterloo, Ontario.

Hipel, K. W. and McLeod, A. I. (1978). Preservation of the rescaled adjusted range, 2, simulation studies using Box-Jenkins models. *Water Resources Research*, 14(3):509-516.

Kottegoda, N. T. (1987). Fitting Johnson S_B curve by the method of maximum likelihood to annual maximum daily rainfalls. *Water Resources Research,* 23(4):728-732.

McLeod, A. I. and Hipel, K. W. (1978). Simulation procedures for Box-Jenkins models. *Water Resources Research,* 14(5):969-975.

Moran, P. A. P. (1959). *On the Theory of Storage.* Methuen, London.

Sangal, B. P. and Biswas, A. K. (1970). The three-parameter lognormal distribution and its application in hydrology. *Water Resources Research,* 6(2):505-515.

Stedinger, J. R. (1980). Fitting log normal distributions to hydrologic data. *Water Resources Research,* 16:481-490.

Vogel, R. M. and Stedinger, J. R. (1988). The value of stochastic streamflow models in overyear reservoir design applications. *Water Resources Research,* 24(9):1483-1490.

INITIAL VALUES

Brown, T. J. and Hardin, C. (1973). A note on Kendall's autoregressive series. *Journal of Applied Probability, 10:475-478.*

Copas, J. B. (1966). Monte Carlo results for estimation of a stable Markov time series. *Journal of the Royal Statistical Society, Series A,* 129:110-116.

STATISTICS AND NUMERICAL ANALYSIS

Conover, W. J. (1971). *Practical Nonparametric Statistics.* John Wiley, New York.

Healy, M. J. R. (1968). Algorithm AS6, triangular decomposition of a symmetric matrix. *Journal of the Royal Statistical Society,* Series C (Applied Statistics), 17:195-197.

Hornbeck, R. W. (1975). *Numerical Methods.* Quantum Publishers, New York.

Ralston, A. (1965). *A First Course in Numerical Analysis.* McGraw-Hill, New York.