AN EXPERIMENTAL INVESTIGATION

OF THE TRANSIENT PHASE

IN DIGITAL SIMULATION

by

SUNIL VASANT KIRTANE

B.S., Ranchi University, India, 1964

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1971

Approved by

L. E. Grosh

Major Professor

Table of Contents

ACKNOWLEDGEMENTS

I dedicate this thesis to my parents Dr. and Mrs. Vasant B. Kirtane whose happiness means to me more than any success. I express my deep appreciation to Dr. L. E. Grosh for his help and valuable guidance throughout this work. I thank Mrs. Marie Jirak for her consistently good typing work. I am thankful to Miss Suanne Singer for her help in preparing this thesis.

CHAPTER 1

INTRODUCTION

1.1 Introduction

The technique of simulation is used in developing and evaluating a model of some real world system. It is a numerical technique based on a mathematical model of the system. In the present work results of simulation experiments obtained by a digital computer are discrete approximations of mathematical expressions.

Simulation experiments are useful in solving multivariable problems that are too complex to be represented by mathematical expressions in predicting the behavior of a proposed system. Simulation experiments differ from Monte Carlo Methods in that Monte Carlo analysis is used to solve deterministic problems whereas simulation techniques are applied to dynamic problems for which no closed mathematical expression can be constructed. (9)

Steady state conditions in a simulation experiment are reached when successive observations of an output statistic give no extra information regarding the future behavior of the system. At this stage the variable becomes statistically stable, i.e. there is no change in its expected value and its variance.

"The term steady state is used in reference to parameter stability. In its usual context this term refers to the stabilizing of the probabilities of the various system states. It is felt that the parameters being estimated are reasonably good indicators of this stabilizing process and the aforementioned definition is made." (26)

An experimenter usually starts a simulation run with atypical values which do not correspond to the steady state conditions. Consequently there is a build-up in the system during which the expected value and the variance of output statistics are unstable. This period in a simulation run is called the transient phase. Figure (1.1)

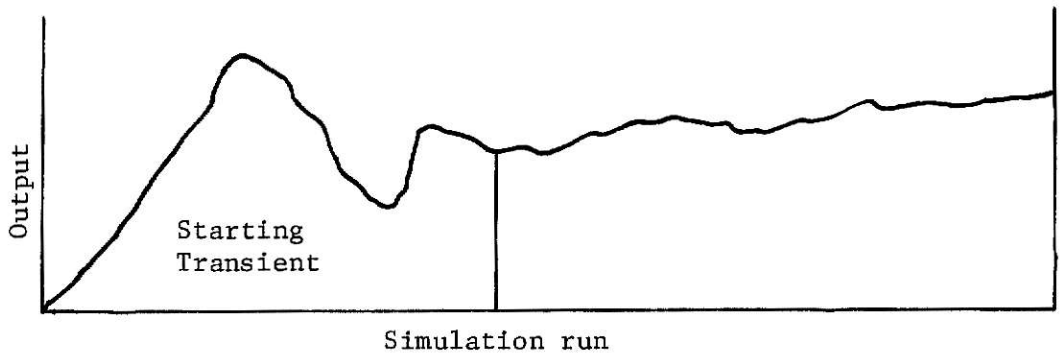


Figure 1.1

To the analyst, the information gathered during the transient phase is of no use because he is basically interested in information concerning a statistic only when it has reached steady state conditions. "Transient period data adds nothing to the final solution and should be discarded whenever possible. Its only real use is in indicating the termination of the transient period." (26) The transient phase, therefore, becomes an unwanted region in a simulation run.

Research workers have suggested several techniques to reduce the variability of statistics during the transient phase so as to attain steady state conditions earlier in simulation experiments. (13) The techniques are of a physical or a statistical nature. A typical example of a physical trial and error technique is the several types of preloading procedures in which the experimenter loads a system to a certain value at the commencement of the simulation run so that the initial build-up may be reduced. In some predictive experiments where no information is available concerning a statistic, it is only a guess as to what the level of preloading should be. In such cases any physical technique to control the build-up would call for a hit and miss method which may result in increased computer cost.

1.2 Literature Review

1.2.1 The Reduction of the Length of the Transient Period

A simulation experiment conducted with straightforward sampling will usually go through a long transient period before steady state conditions are attained. This involves high computer costs. Several experimental and statistical procedures have been suggested to reduce the variability of the output statistics, consequently reducing the transient phenomenon in simulation experiments. Gaver (13) has summarized some of these procedures.

Conway (3) suggested an approach which consisted of making two or more replications of a simulation experiment, each time using a different random sequence to generate the random variates. This procedure is experimental and not statistical. It reduces the variability of the output statistic but does not prove economical by the way of computer cost.

Some attempts were made by experimenters to use various preloading methods where some prespecified values are assigned to the system parameters at the commencement of the run in order to reduce the initial build-up in the system. The major objection to this procedure is that it cannot be used in many predictive models.

Hammersley and Morton (18) introduced the concept of antithetic transformations and Hammersley and Handscomb (17) extended this idea to form antithetic variates for the Monte Carlo techniques and the simulation experiments. They showed that the pairs of random variates formed by using random numbers and the complements of these random numbers are negatively correlated. A mathematical proof of this is given by Page. (24) Page also shows that if a 2-tuple is formed by using random numbers r_i and s_i and if another 2-tuple is formed by using random numbers $r_i' = (1-r_i)$ and $s_i' = (1-s_i)$, then the two 2-tuples are negatively correlated. In the present work an attempt is made to reduce the variability of an output statistic by using the antithetic variate approach.

Stratification as explained by Gaver (13) is essentially an extension of the antithetic variate approach, with the difference that it uses three random numbers r, (r+1/3) and (r+2/3) in the range (0,1) to form negatively correlated random variates.

The random variates in the simulation experiments are formed by using random numbers generated by one or more pseudorandom generators. Before starting a simulation experiment it is necessary to test the generated random numbers to ascertain that they are uncorrelated.

Coveyou and MacPherson (4) showed that random numbers generated by the multiplicative congruence method satisfy most statistical tests for independency. Some of these tests are described by Kendall and Smith (20), Naylor et. al. (22), and Good (15).

In the present study a frequency test and a spectral analysis (1) were performed to test the generated random numbers.

1.2.2 The Determination of the End of the Transient Period

In simulation experiments it is very important to correctly locate the end of the transient phase. A failure to do this leads to erroneouse results in estimating the system parameters. It is accepted by research workers that the system parameters are good indicators of the stability of any system.

Beuno (2) suggested that the stability of the sample mean should be used as the indicator of system stability. He presented a technique for comparing means of successive samples from time series by using a sequential t-test. The shortcoming of Beuno's method is that it uses the stabilization of the sample mean as the one decision criterion and does not take into account the stability of the variance of this mean. Also, Beuno assumes the data to be independent which is not true for a time series. Fishman (10) showed that successive observations in time series are highly autocorrelated. This is also supported by Reese (26), Gaver (14), and Newell (23). Fishman suggested a method for comparing sample means of autocorrelated data taking into consideration the autocovariance between the observations. Reese (26) combined this concept of autocorrelation with Beuno's dynamic procedure and suggested a technique to compare successive sample means using a sequential t-test.

In 1969 Fishman (12) developed a procedure for determining the sample size necessary to estimate the sample mean of the autocorrelated data. This procedure is used in the present study on successive samples from simulation outputs and the means of these samples are compared to study the stability of the process.

1.3 The Antithetic Variate Approach

The antithetic variate approach is a statistical technique recommended for reducing the variability of an output statistic in a simulation run. A statistical explanation of the antithetic variate approach is as follows:

Let a statistic $Z(t)$ which is a function of time t be represented by

$$Z(t) = X(t) + Y(t)$$

Then,

$$V[Z(t)] = V[X(t)] + V[Y(t)] + Cov[X(t), Y(t)]$$

It is desired to reduce the variability of $Z(t)$.

Since $V[X(t)]$ and $V[Y(t)]$ are positive quantities, $V[Z(t)]$ is reduced by assigning a negative value to $Cov[X(t), Y(t)]$. In other words, $X(t)$ and $Y(t)$ should be chosen such that they produce a negative correlation. The variables produced by purposely introducing negative correlation between pairs are known as antithetic variates. (17)

If $X(t)$ is a function of a random number r, where r is uniformly distributed over $(0,1)$ and $Y(t)$, a function of $(1-r)$, it can be shown that $X(t)$ and $Y(t)$ are negatively correlated. (24) If $X(t)$ and $Y(t)$ are antithetic variates, $Cov[X(t), Y(t)]$ will tend to reduce the variability of $Z(t)$. Figure (1.2)

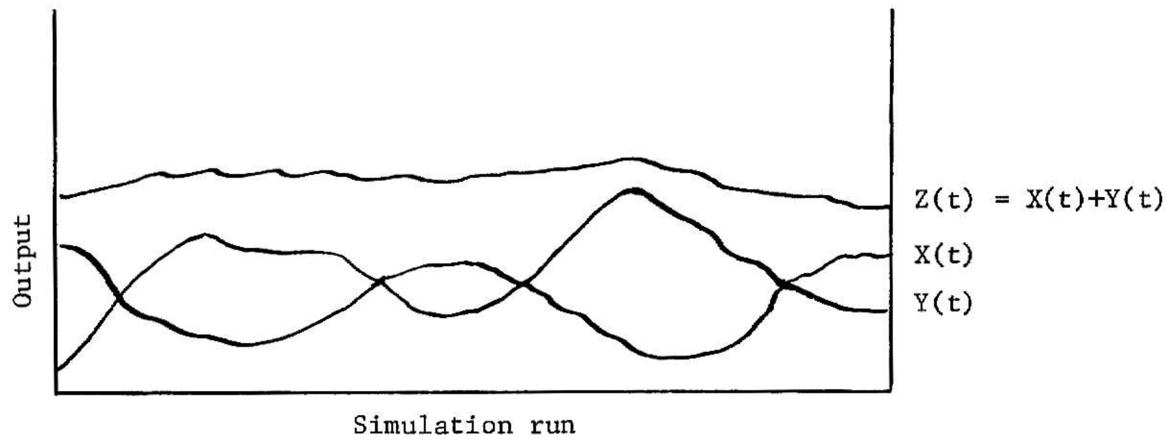


Figure 1.2

In the present work an attempt is made to study the extent to which the antithetic variate approach affects the variability of an output statistic in a simulation experiment. A single server queueing system is simulated with a facility to change the utilization factor of the process. The statistic gathered is the time averaged number of units in the system. Two seed values are used for random number generation; one for generating interarrival time and the other for service time. Two simulations are conducted simultaneously in a single run, one using the generated random numbers and the other using their complements, so that the output statistics of the two simulations will form antithetic variates. The mean of the realizations from the two simulations is expected to have less variability than that observed in each of the two individual simulations.

The effectiveness of the antithetic variate approach is measured by the simulation efficiency. The efficiency of a variance reduction technique can be measured in terms of the necessary sample size or computer time used to attain steady state conditions.

$$\text{Simulation efficiency} = \eta_{sim} = \frac{100(T_S - T_I)}{T_S}\ \%$$

where, T_S = Computer time required by straightforward sampling

T_I = Computer time required by variance reduction technique.

Usually, by using a variance reduction technique, the sample size is reduced. (13), (8)

The output statistics of a simulation experiment are usually autocorrelated. Consider, for example, the waiting time of a unit as an output statistic. The waiting time of the nth unit will depend to a certain extent on that of the (n-1)th unit and to a certain extent on the (n-2)th unit and so on. When the value of an observation depends upon its predecessors such data is called autocorrelated data. (10)

1.4 Autocorrelated Data in Time Series

For an independent data sample of size N, the mean is estimated as

$$\bar{X} = 1/N \sum_{i=1}^{N} X_i \qquad i = 1, 2, \ldots, N$$

and the variance of the mean is estimated by

$$s^2(\bar{X}) = 1/N \sum_{i=1}^{N} (X_i-\bar{X})^2/N$$

But this is not applicable to autocorrelated data where the autocovariance between observations at all significant lags must also be considered.

The estimated autocovariance of data at any lag is given by

$$C_{N,\tau} = E\ [(X_t-\bar{X})(X_{t-\tau}-\bar{X})]$$

where τ = lag

X_t = observed value in the time series at time t

$\bar{X}$ = estimated mean of the sample

Because of this autocorrelated structure of the data, an estimated variance of the sample mean is given by

$$V(\bar{X}) = \frac{1}{T}\left\{\hat{R}(0) + 2\sum_{\tau=1}^{M} (1-\tau/M)\hat{R}(\tau)\right\}$$

where T = Time length of the sample

$\hat{R}(0)$ = Estimate of the variance of X

M = Order of the scheme

$\hat{R}(\tau)$ = Estimated autocovariance coefficient at lag τ (15)

Data from a simulation experiment is often autocorrelated. The experimenter has no prior knowledge of the autoregressive coefficients and the order of the autoregressive scheme. Some technique has to be used, therefore, to estimate the parameters of the linear autoregressive scheme that will represent the autocorrelated data.

1.5 The Fishman Technique (12)

G.S. Fishman has suggested a technique that determines the sample size of autocorrelated data so that its mean can be estimated with a prespecified level of confidence. In the present work an attempt is made to locate the end of the transient phase in simulation experiments by using the Fishman technique in a recursive manner. The Fishman technique estimates the sample mean such that the variance of this sample mean is within prespecified limits. The decision criterion in a Fishman technique, therefore, is the limiting value of the variance of the sample mean.

A good indication of the steady state conditions in a system is the stabilization of the parameters, namely the mean and the variance of different output statistics. If the Fishman technique is applied to successive samples from a time series, the value of the sample mean will tend to stabilize as the process becomes stationary. Thus, if we can obtain the conditions in a time series where the sample mean and the variance of this mean become stable, it will indicate that the system has reached steady state.

In order to study the stability of the sample mean, a limit is imposed on the algebraic difference between successive sample means. As we proceed to take successive samples from a time series we expect that this difference will diminish until it is less than the specified limit. At this stage we can say that the system has reached stable conditions. Figure (1.3)

THIS BOOK CONTAINS NUMEROUS PAGES WITH THE ORIGINAL PRINTING BEING SKEWED DIFFERENTLY FROM THE TOP OF THE PAGE TO THE BOTTOM.

THIS IS AS RECEIVED FROM THE CUSTOMER.

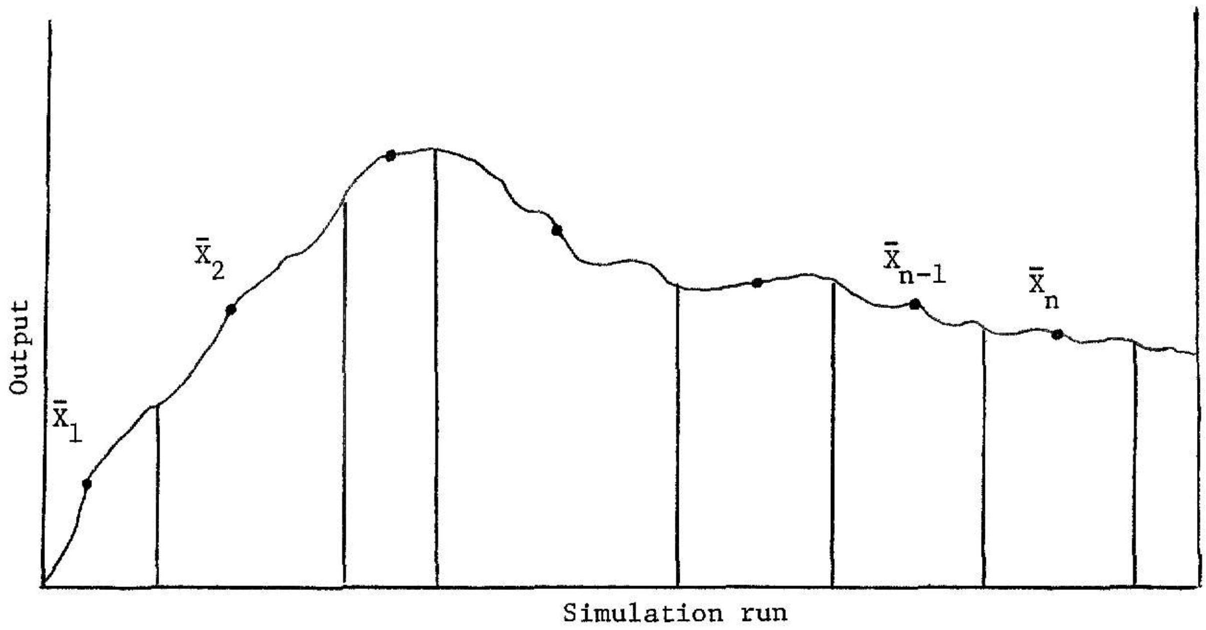


Figure 1.3

In the transient phase the size of the successive samples and the order of the autoregressive scheme as determined by the Fishman technique fluctuate because the process is non-stationary. However, after the process reaches steady state conditions the successive samples are expected to have the same relative size and the same order for the autoregressive scheme.

By using the Fishman technique successively on the simulation data, we expect to locate the end of the transient phase by studying the stability of the sample mean, the variance of this mean, and the sample size and the order of the autoregressive scheme.

1.6 Scope of Work

The work proposed to be carried out on this thesis is summarized as follows.

1. To conduct simulation experiments with the antithetic variate approach for a single server queueing system with utilization factors of 0.5, 0.6, 0.7, 0.8 and 0.9.
2. To study the effect of this antithetic variate approach on simulation efficiency.
3. To study the characteristics of a typical output statistic from the simulation experiments.
4. To apply the Fishman technique on successive samples of the data to locate the end of the transient phase.
5. To study the effect of utilization factor on the length of the transient phase.

CHAPTER 2

THE SIMULATION

2.1 GASP II A

Simulation experiments were conducted using GASP II A. (General Activity Simulation Program). GASP II A uses various subroutines written in FORTRAN IV which perform most operations required for simulation experiments. Readers are referred to the book "Simulation with GASP II A" by Pritsker A.A.B. and Kiviat P.J. (25), where a detailed description of these subroutines is included. A simulation program using GASP II A is directed by programmer-written subroutines which control the flow of the simulation.

A general flow diagram showing the operation of a GASP II A system is given in figure (2.1). (25)

2.2 System to be Simulated

A single server queueing system was simulated for the present study. The utilization factor for a single server queue system is given by,

$$\rho = \frac{\lambda}{\mu}$$

where,

λ = Mean arrival rate

μ = Mean service rate

Experiments were conducted with utilization factors of 0.5, 0.6, 0.7, 0.8 and 0.9. The arrival rate and the service rate were assumed to follow exponential distributions with means λ and μ respectively.

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

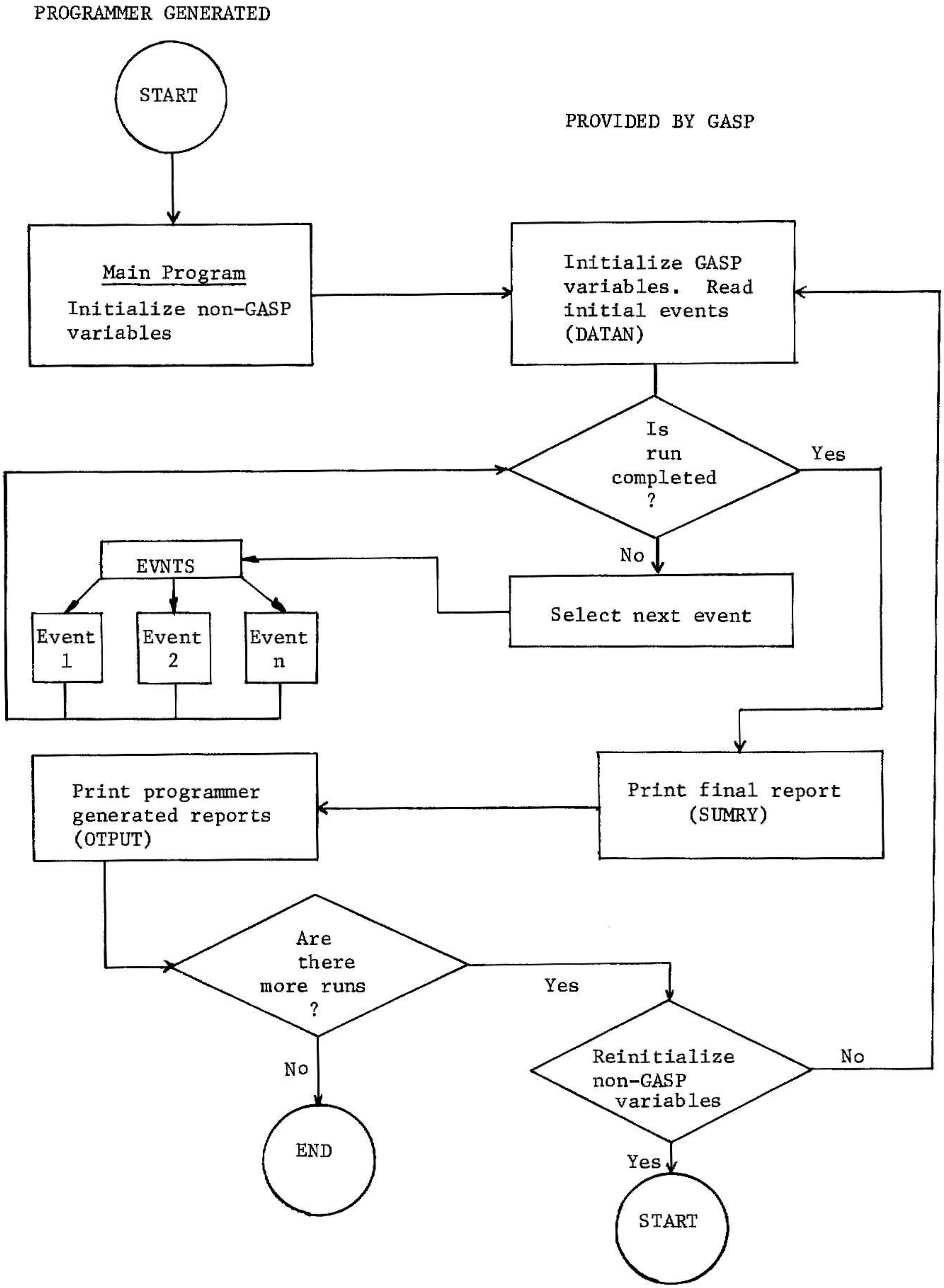


Figure 2.1 A Typical GASP Program

A schematic diagram of this model is given in figure (2.2).

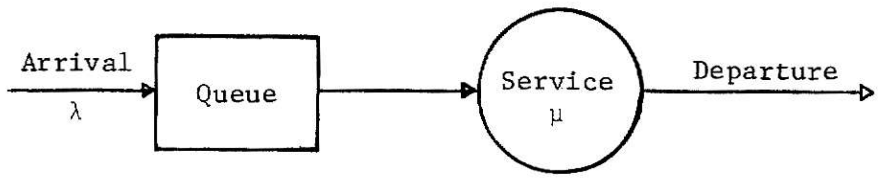


Figure 2.2

2.3 The Antithetic Variate Approach

Two independent simulations were carried out simultaneously in each run. This was done by using the generated random numbers for one simulation (simulation A) and the complements of these numbers for the second simulation (simulation B). The two simulations were kept separate by assigning different values to a coding attribute to the entries in each simulation. The output statistic gathered by these simulations are negatively correlated.

2.4 Files

Three files were maintained to store the information pertaining to the entries.

1. File number 1 was the event file to store the attributes of an entry associated with any event in the simulation.
2. File number 2 was the queue file for storing the attributes associated with an element in the queue, belonging to simulation 1.
3. File number 3 was similar to file number 2 except that it was used for the elements belonging to simulation 2.

2.5 Attributes

The following attributes were used to give full information concerning an entry in a file. The indexing or coding attributes which were integers, were represented by JTRIB(I). All floating point attributes were represented by ATRIB(I). This is a special facility offered by GASP II A.

ATRIB(1) Denotes the time that an event is to occur. This is the ranking attribute for file 1 and its value determines the service priority of an entry in the file.

ATRIB(2) Denotes the time when a unit enters service.

JTRIB(1) Denotes the event code of an entry. The value of JTRIB(1) indicates which event is to take place.

JTRIB(2) Denotes to which of the two simulations an entry belongs. If JTRIB(2) = 1, the entry belongs to simulation A and if JTRIB(2) = 2, the entry belongs to simulation B. This keeps the two simulations separate from each other.

2.6 Priority Rule for Service

The priority rule for service was First In First Out. (FIFO). The ranking attribute was the time of arrival which was denoted by ATRIB(1). Elements were picked from the queue files on the 'Low Value First (LVF)' basis.

2.7 Output Statistic

The statistic gathered was the time averaged number of units in the system. From GASP subroutine TMST, the time averaged number in the system is given by SSUMA(N,2), where N is a code associated with an entry. This SSUMA(N,2) is a variable in the subroutine TMST. It calculates the product of the instaneous number of units in the system and the current simulation time.

SSUMA(N,2) was reset after each report time. This gave only the value of the area between two successive report times. (Figure (2.3)). The value of the statistic was multiplied by a scaling factor SKALE to account for the decimal points and the final output was taken in an integer form.

The output was collected in the form of punched cards so that the data could be used for subsequent operations and testing. Six fields were reserved on a card for each observation. The program stored twelve observations at a time, six for each simulation. After taking the twelfth observation all twelve values were punched on a card and then the storage was cleared for the next observation. A facility was provided to sequence the cards for reference.

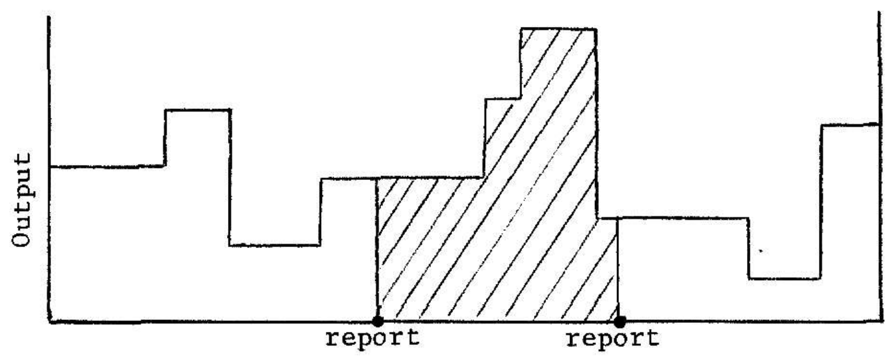


Figure 2.3

2.8 Main Program

A main program was written to initialize all non-GASP variables. The variables were made available to all programmer written subroutines through common statements.

The non-GASP variables used are given below:

XISYS - Instantaneous number of units in the system.

XL - Mean arrival rate λ.

XMU - Mean service rate μ.

TAU - The time interval at which reporting is done.

2.9 Subroutine EVNTS

Subroutine EVNTS assigned codes to different events occurring in a simulation experiment. This subroutine is called by GASP each time an event is scheduled to take place. The event codes assigned were 1 for arrival, 2 for service and 3 for reporting.

2.10 Subroutine ARRVL

Subroutine ARRVL is called by subroutine EVNTS whenever an arrival is scheduled to take place. First it schedules the next arrival by drawing a random variate from an exponential distribution with mean λ, and files it. It then checks the status of the service facility. If the server is busy the arriving unit is placed in the queue file and processed later. If the server is free the arriving unit is served. This is done by drawing a random variate from an exponential distribution with mean μ, and filing an end of service event. After this the control is turned back to GASP through subroutine EVNTS. A flow diagram of the arrival event is given in figure (2.4).

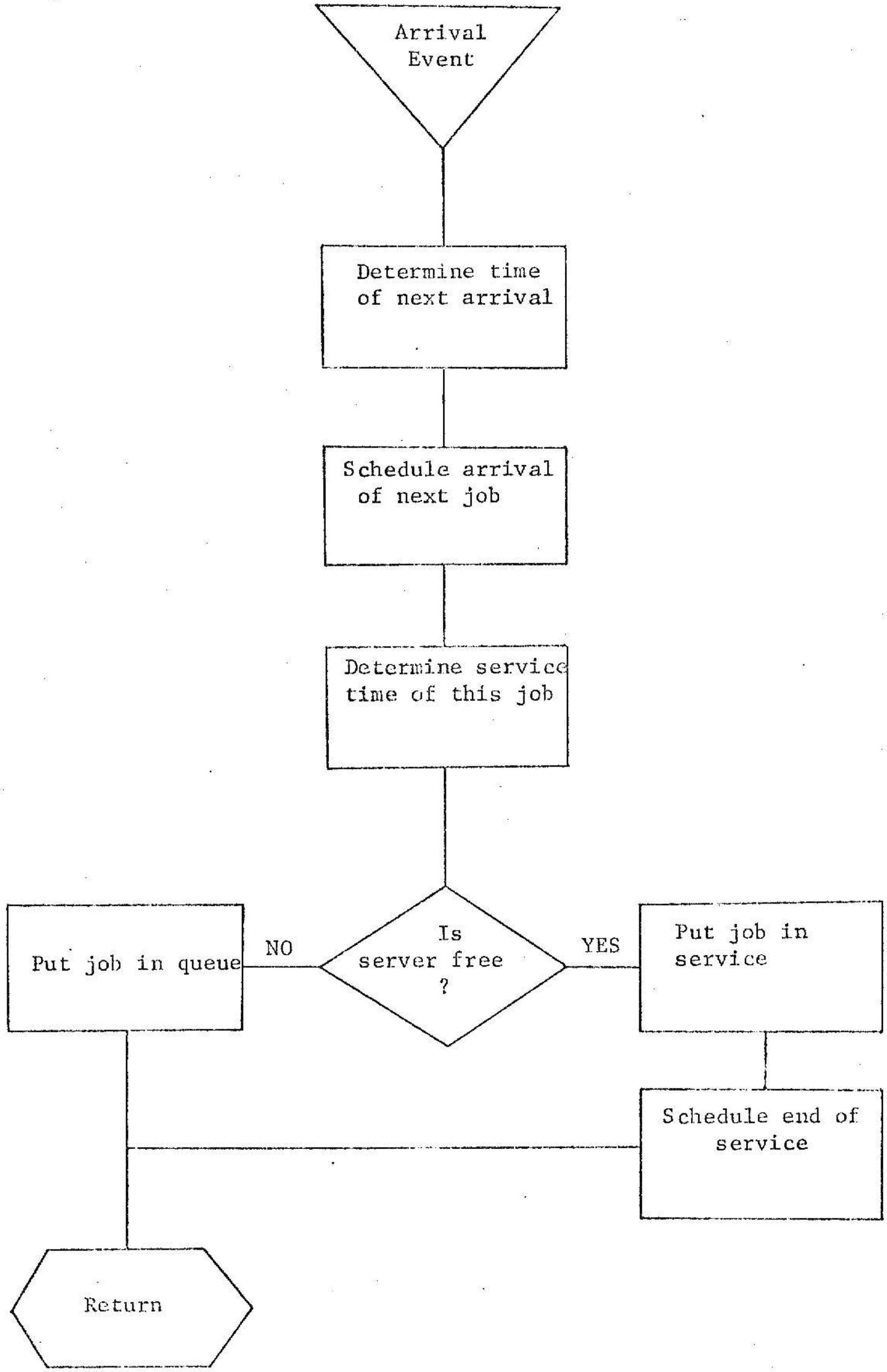


Figure 2.4 Arrival Event

2.11 Subroutine ENDSV

Subroutine ENDSV is called by subroutine EVNTS when an end of service occurs. If there are units waiting in the queue, it serves one unit from the queue according to the assigned priority rule. This is done by drawing a random variate from an exponential distribution with mean μ, and filing an end of service event. If there are no elements in the system the server remains idle until the arrival of the next unit. A flow diagram of the end of service event is given in figure (2.5).

2.12 Subroutine REPORT

Subroutine REPORT is called by subroutine EVNTS whenever the programmer desires to have a report on the statistic being collected. In the present work the subroutine REPORT is called after every TAU time units.

Subroutine REPORT computes the required statistic with the help of the GASP subroutines TMST and COLCT, and then returns control to GASP through subroutine EVNTS.

In the present work, each time the subroutine REPORT was called, it gave a pair of output observations. The first belonged to simulation 1 and the second belonged to simulation 2.

A FORTRAN listing of all programmer written subroutines is given in Appendix B.

2.13 The Random Generator

2.13.1 Random Number Generation

In the present work the arrival and end of service events were created by using random variates generated from a pseudorandom generator. The random numbers were generated by the multiplicative congruence method.

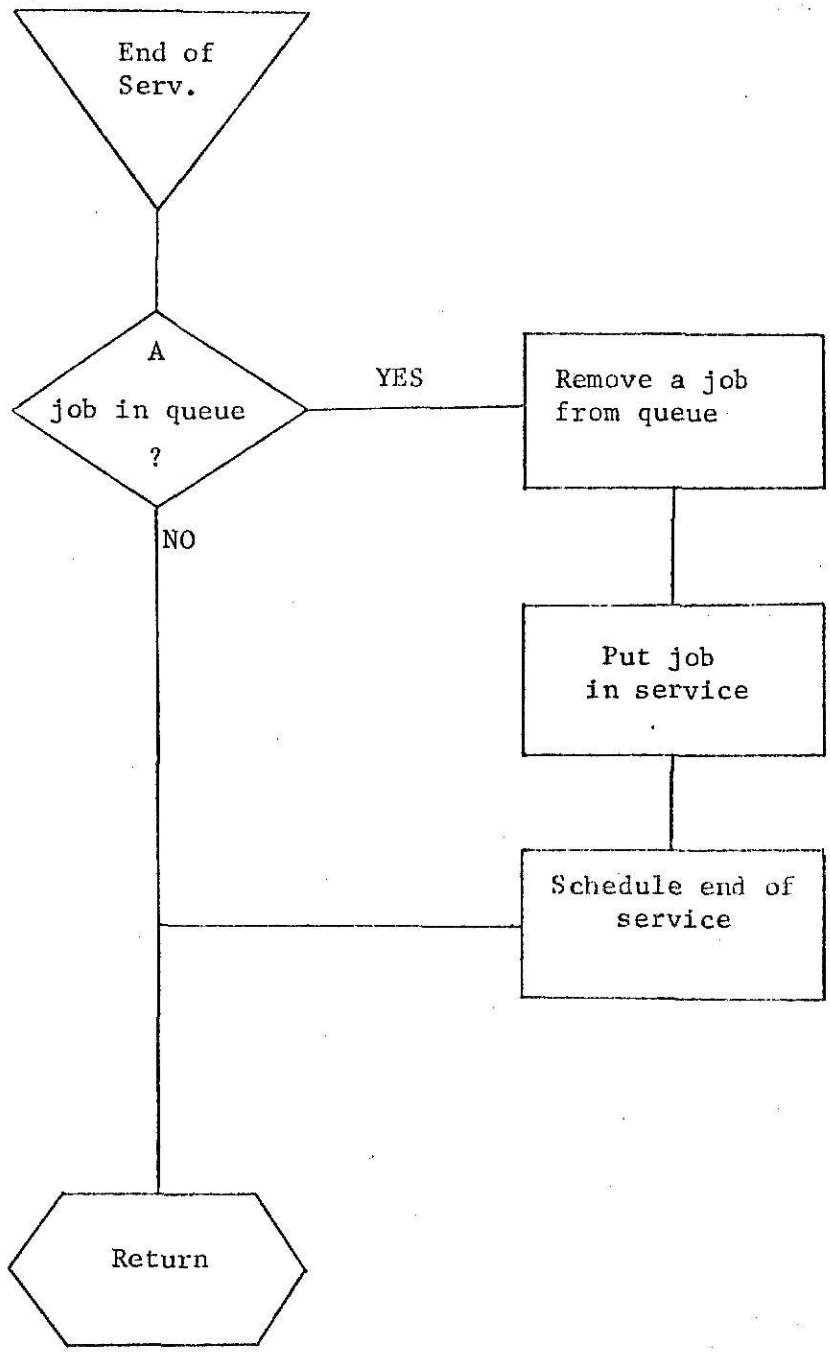


Figure 2.5 End of Service Event

"Coveyou and MacPherson, who offer a unified theory of the statistical behavior of n-tuples of pseudorandom generators conclude that currently there is no better method of generating n-tuples than by the simple multiplicative congruence method." (11).

A mathematical representation of the multiplicative congruence method is

$$C_{i+1} = C_i \, A \pmod{P}$$

where, $C_{i+1} = (i+1)^{th}$ random number

$C_i = i^{th}$ random number

A and P are constants.

As seen from the above expression the i^{th} random number is used for generating the $(i+1)^{th}$ random number. In order to start the sequence an initial value $C = C_o$ is required. This is called the seed for random number generation.

A truly random sequence of numbers is produced by some random natural processes; but it is preferred to generate random numbers using a computer because the generation is fast and reproducible. However such a sequence is not truly random. "The random generators based on mathematical relations are not truly random since the sequence is completely deterministic." (9). Such a sequence is called a sequence of pseudorandom numbers. A shortcoming of this sequence is that it is of a repetitive nature; but a proper choice of A, C_o, and P will give a long sequence before it starts repeating itself.

It was necessary to test the generated random numbers for their independency before the generator could be accepted for using in simulation experiments. The procedures used for testing the generator are described below.

2.13.2 Testing of Random Numbers

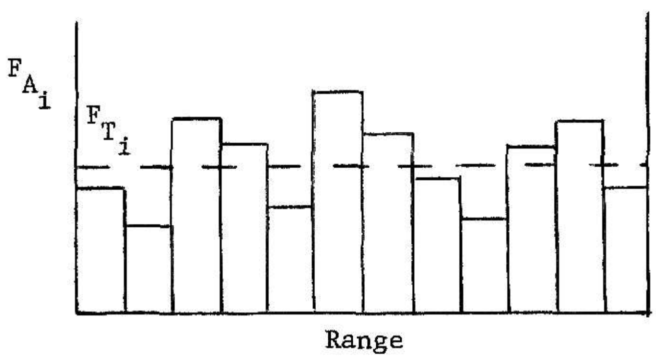
Random numbers generated on a computer are expected to be uniformly distributed. This means if we generate R random numbers over (0,1) and tabulate relative frequencies in N equal intervals, the probability of a number falling in one particular interval is 1/R. (Figure 2.6)

The hypothesis that the generated random numbers come from a uniform distribution can be verified by performing a chi-square test. (9).

The variable

$$\chi^2_{N-1} = \sum_{i=1}^{N} \frac{(F_{A_i} - F_{T_i})^2}{F_{T_i}}$$

is approximately distributed as chi-square with N-1 degrees of freedom.



where, N = number of intervals

F_{A_i} = actual frequency of random numbers in the i^{th} interval

F_{T_i} = theoretical frequency of random numbers in the i^{th} interval

Figure 2.6

The calculated value is compared with the theoretical chi-square value with (N-1) degrees of freedom at a specified confidence level. If the calculated value is less than the theoretical value, the hypothesis that the random numbers came from a uniform distribution is accepted.

A FORTRAN listing of the chi-square test is given in Appendix C.

The serial test suggested by I.J. Good (14) is an extension of the frequency test. This procedure tests the autocorrelation between random numbers only for a lag of 1. This is not conclusive in testing for independence between random numbers for lags greater than one. Plots of the autocorrelation function and the power spectrum were obtained and studied to detect any probable autocorrelation and cyclic behavior for lags greater than 1. A description of the plotting and the interpretation of the autocorrelation data and the power spectrum is given in the references. (1), (21) The plots showed that the random numbers were fairly independent and acceptable for the present study. The plots are given in Appendix C.

The pseudorandom generator was accepted, therefore, to generate the random variates for the simulation experiments.

2.13.3 Function RANI(I)

The function subprogram RANI(I) was written to generate random numbers r_i and their complements $(1-r_i)$. The ith random number put out by the subprogram is the generated random number or its complement depending upon the seed index (I) which is used as the argument of the subprogram.

Mathematically, the ith random number is given by

$$r_{i,(2k-1)} = r_{(i-1),(2k-1)} A \pmod{P}$$

$$r_{i,(2k)} = r_{(i-1),(2k)} A \pmod{P} \qquad k = 1, 2, \ldots$$

where, (2k-1) and (2k) = Seed number indices

A = Multiplying constant

In other words, if the seed index is odd the subprogram gives the generated random numbers and if the seed index is even it gives the complements of the generated random numbers. This switching operation is done by the subroutine MOD(I,J).

A FORTRAN listing of RANI(I) is given in Appendix B.

2.14 Testing the Simulation Output

Box and Jenkins Procedure

The simulation output was subjected to a testing procedure suggested by Box and Jenkins (1) to study the statistical behavior of the data. (21) This procedure calculates and plots the sample autocorrelation coefficients and the partial autocorrelation coefficients of the given data. These plots are shown in Appendix D. The nature of these plots assists in deciding whether the given data should be represented by an autoregressive scheme, a moving average scheme or a combination of the two. The findings of these tests are discussed in Chapter 4.

The data was tested for autocorrelation up to a lag of 50.

CHAPTER 3

FISHMAN TECHNIQUE

3.1 The Fishman Procedure

For autocorrelated data the Fishman technique (12) determines the sample size required to estimate the sample mean with a specified level of confidence. It is designed to be built into a simulation experiment. In the present work the Fishman technique is used as a subroutine (FISHMN). A step by step description of the Fishman technique is as follows.

1. Subroutine FISHMN draws an initial sample of size IM. The mean of the sample and the autocovariance for lags 0 to IRP are calculated. IRP is the highest order of autoregressive scheme for which the data is tested. Usually an order of scheme between 5 and 10 is adequate. (12)

Autocovariances at all lags τ are calculated by the formula

$$C_{N,\tau} = \frac{1}{N} \sum_{t=1}^{N-\tau} (X_t-\bar{X})(X_{t+\tau}-\bar{X}) \qquad \tau = 0, \ldots \text{IRP} \tag{3.1}$$

where N = sample size.

$\bar{X}$ = sample mean.

2. The next step is to estimate the autoregressive coefficients $[b_{r,s}(r=0,\text{IRP};s=0,r)]$. Here r corresponds to the order of the scheme and s corresponds to lags. For s = 0, $\hat{b}_{r,0} = 1$, r = 0, 1, ..., IRP

Using the expressions presented by Durbin (7) and Whittle (27)

$$v_r = \sum_{s=0}^{r} \hat{b}_{r,s}\, C_s \qquad r = 0, \ldots, (\text{IRP}-1) \tag{3.2}$$

$$w_r = \sum_{s=0}^{r} \hat{b}_{r,s} C_{r+s-1} \tag{3.3}$$

$$\hat{b}_{r+1,r+1} = - w_r/v_r \tag{3.4}$$

$$\hat{b}_{r+1,s} = \hat{b}_{r,s} + \hat{b}_{r+1,r+1} + \hat{b}_{r,r-s+1} \qquad s = 1, \ldots, r \tag{3.5}$$

3. Using the estimates of autoregressive coefficients, an estimate of the sample residual variance of the autoregressive scheme at all lags is found by

$$\hat{\sigma}^2_{r+1} = \frac{1}{N-r} \sum_{t=r+1}^{N} [\sum_{s=0}^{r} b_{r+1,s} (X_{t-s} - \bar{X})]^2 \qquad r = 0, 1, \ldots, (IRP-1) \tag{3.6}$$

It may be noted that for a lag of zero, the variance is equal to the variance of the independent data.

4. The order of the autoregressive scheme depends upon the significance of the autocorrelation coefficients. The order of the scheme is the biggest lag for which the autocorrelation coefficient is significant. A confidence band is placed on each of the diagonal elements of the autoregression coefficient matrix.

According to the results shown by Whittle (27), the square of the standard deviation or the unexplained variability is given by

$$\omega_r = 1 - \hat{b}^2_{r,r} \tag{3.7}$$

and the confidence band for $\hat{b}_{r,r}$ is given by

$$\hat{b}_{r,r} \pm P(\omega_r/N)^{1/2} \tag{3.8}$$

where, N = sample size

α = specified significance level

P = normal point corresponding to α.

Placing this confidence interval on $\hat{b}_{r,r}$, the order of the last significant autocorrelation coefficient is determined. This is the order of the autoregressive scheme.

5. The sum of the autocovariances $\sum_{\tau\to\infty} C_\tau = m$ is a finite quantity for non-cyclic data because the effect of the autocorrelation decreases as the number of intervening events increases.

If the order of the scheme is p, an estimate of m is given by

$$\hat{m} = \frac{\sigma_p^2}{(\sum_{s=0}^{p} b_{p,s})^2} \tag{3.9}$$

If the order of the scheme is zero it means that the data is not autocorrelated and $\hat{m} = C_o$.

6. Having obtained $\hat{m}$ for a sample of size N, the variance of the sample mean is given by $\hat{m}/N$.

The reliability of the sample mean is expressed by

$$P_r[|\bar{X}-\mu| < Q\sqrt{\hat{m}/N}\ ] \leq (1-\beta) \tag{3.10}$$

where,

μ = population mean

β = specified significance level

Q = normal point corresponding to β

Suppose we wish to determine a sample size N^* such that the variance of the resulting sample mean is less than or equal to V with a probability of $(1-\beta)$ or,

$$P_r[(\bar{X}_{N*} - Q\sqrt{V}) < \mu < (\bar{X}_{N*} + Q\sqrt{V})] \leq (1-\beta) \tag{3.11}$$

Then $\quad V = \hat{m}/N^*$

or, $\quad N^* = \hat{m}/V \qquad (3.12)$

Now, if c is a specified constant, we may write

$$P_r[(\bar{X}_{N*}-c) < \mu < (\bar{X}_{N*}+c)] \leq (1-\beta) \tag{3.13}$$

Then from (3.11) and (3.13)

$$c = Q\sqrt{V}$$

or,

$$V = (c/Q)^2 \tag{3.14}$$

Using expressions (3.9), (3.12), and (3.14), the value of N^* is calculated.

If the size of the sample drawn, N, is less than N^*, the new sample size is calculated by adding a fraction γ of the difference (N^*-N) to N. Such a scaling increases the number of iterations and the convergence is slower but it prevents the drawing of an excessive sample from the simulation experiment. The procedure is repeated from Step (2) onwards after updating the sample size and drawing extra observations.

8. When the actual sample size is greater than or equal to the required sample size N^*, the program computes the sample mean and variance.

A block diagram of the Fishman technique is given in Appendix E.

3.2 Main Program

It was decided to use the Fishman technique to locate the end of the transient phase in the simulation experiments. This was done by checking the stability of the sample mean. The algebraic difference between successive sample means was used as the criterion for stability of the sample mean. If the difference between successive sample means is less than a specified value δ, then it is considered that the sample mean has attained a steady value. This was accomplished by a main program coupled with the subroutine FISHMN.

A step by step description of the main program is as follows.

A. The parameters read into the main program are listed below.

IM	=	Initial sample size
IRP	=	Proposed (maximum) order of the scheme
P	=	Normal point corresponding to a significance level α.
Q	=	Normal point corresponding to a significance level β.
CONF	=	A constant for placing a confidence interval on the sample mean.
DELTA	=	A constant for comparing successive sample means.
GAMMA	=	A scaling factor for drawing additional sample.

B. Elements of the time series are read in and stored. These realizations are obtained from the punched output of the simulation experiments conducted separately.

C. A reporting time interval is selected which is a multiple of the original time interval used in the simulation experiments. TAU represents the original time interval and TINT represents the desired time interval.

D. Control is passed to subroutine FISHMN to obtain a sample mean.

E. The successive values of the sample means are obtained from subroutine FISHMN and are compared. If the difference between two successive samples is less than the specified value δ, the main program prints a comment accordingly.

F. A fresh sample is fed to subroutine FISHMN and the procedure is repeated.

3.3 Parameters used in Subroutine FISHMN

ARGUMENT

X: These are the elements of the time series passed as an array from the main program to subroutine FISHMN.

COMMON

IM, IRP, P, Q, GMA, CONF: These values are read in by the main program and passed on to subroutine FISHMN. Their significance is explained earlier.

NEXT: The value of the sample size when it exceeds the required sample size. The value of NEXT is passed from subroutine FISHMN to the main program.

NBASE: The point in the time series where testing by the Fishman technique is concluded. A fresh test commences from the observation X(NBASE+1). NBASE is updated by adding to it the value of NEXT after the completion of each Fishman test.

XBARR: The value of the sample mean calculated by the subroutine FISHMN and passed to the main program. It is stored in the main program as an array and is used for comparison of successive means.

NOBS: The value of the total number of observations available. It is computed in the main program and passed to the subroutine FISHMN. If the size of the sample to be drawn by the subroutine FISHMN exceeds NOBS, the program stops, thus preventing a waste of computer time.

3.4 Values of Parameters Selected for the Experiments

IM = 50

IRP = 10

α = 0.05

P = 2.24

β = 0.1

Q = 1.65

CONF = 0.33

DELTA = 10.00

GAMMA = 0.3333

CHAPTER 4

DISCUSSION OF RESULTS

4.1 Testing of Random Numbers

The generated random numbers were expected to come from a uniform distribution. A frequency test was performed on 4000 generated random numbers divided into 100 equal intervals over (0,1). The one tail chi-square test performed at the 95% confidence level was accepted. A FORTRAN listing of this test and the results are given in Appendix C. The exponential random variates required for the simulation experiments are developed from the uniform random sequence given by this generator.

Having accepted the generated random numbers as uniformly distributed, they were tested for the presence of any autocorrelation and any cyclic component.

The autocorrelation test was performed by plotting the autocorrelation coefficients for 2000 random numbers for a maximum lag of 100. For a large sample N, assuming that the theoretical autocorrelation coefficients are equal to zero, the standard deviation of the autocorrelation coefficients is given by $\sigma = 1/\sqrt{N}$. A confidence interval of $\pm 2\sigma$ was placed on the plot of the autocorrelation coefficients. This plot and the calculations for the confidence interval are given in Appendix C. Most of the data points fell well within this confidence interval; so it was safe to assume that there is no significant autocorrelation between the random numbers.

A plot of the power spectrum for 2000 random numbers was obtained to study the presence of any cyclic component in the data. The plot of the power spectrum and the calculations for the confidence interval are given in Appendix C. No peak with excessive power was found in the power spectrum, hence no significant cyclic component was identified.

The frequency test, the autocorrelation test and the spectrum test indicate that the generated sequence is random and independent. Hence, the random number generator was accepted for the simulation experiments.

4.2 The Simulation Efficiency

The simulation efficiency shows that the antithetic variate simulation approximately halves the transient phase. The results of the efficiency measurements for the utilization factors of 0.5, 0.6 and 0.7 are given in Appendix A. It was not possible to obtain these measurements for the utilization factors of 0.8 and 0.9 because the simulations did not converge at these utilization factors. A sample plot of the cumulative means for the two simulations and for the mean value simulation is given in Appendix B. The plot is for the utilization factor of 0.5.

4.3 Box and Jenkins Procedure

4.3.1 The Autoregressive Nature of the Data

A test suggested by Box and Jenkins was applied to study the nature of the simulation output. The test says that if the autocorrelation function of the data appears in the form of damped exponentials and/or damped sine waves and if the partial autocorrelation function cuts off then the process is autoregressive. Autocorrelation coefficients were plotted for a maximum lag of 50. A plot for the utilization factor of 0.5 and a reporting time interval of 25 time units is given in Appendix D.

The plot showed that the autocorrelation coefficients were in the form of a damped sine wave, whereas the partial autocorrelation coefficients showed a cutoff point after which they tailed off. It was inferred from these results that the simulation data can be represented by an autoregressive process. The Fishman technique which is designed to test autocorrelated data can, therefore, be applied to this simulation output.

4.3.2 Effect of the Utilization Factor on the Autocorrelated Data

The plots of the autocorrelation coefficients for the utilization factors of 0.5, 0.6, 0.7, 0.8 and 0.9 and the reporting time interval of 25 time units are given in Appendix D. From these plots it is seen that the autocorrelation between observations as well as the order of the scheme increase as the utilization factor of the process is increased. Thus as the congestion in a queueing system increases the realizations of a statistic in the resulting time series become more and more dependent upon the preceding realizations of the statistic. The plots show a very high autocorrelation between observations for the utilization factors of 0.8 and 0.9.

4.3.3 Effect of the Reporting Time Interval on the Autocorrelated Data

The plots of the autocorrelation coefficients for the reporting time intervals of 10, 25, 50, 100 and 200 time units at a utilization factor of 0.5 are given in Appendix D. These plots show that the autocorrelation between observations decreases as the reporting time interval is increased. This is in accordance with the intuitive concept that the dependence of an element upon a preceding element in a time series will become less as the two elements are farther apart in time.

A very important phenomenon observed in these tests was that as the reporting time interval was increased, the sample autocorrelation coefficients started to show a cut-off point and the partial autocorrelation coefficients damped out. This indicates that as the reporting time interval is increased, the nature of the process changes from an autoregressive scheme to a moving average scheme. The Fishman technique, therefore, was not expected to give meaningful results for the data corresponding to high reporting time intervals.

4.4 The Fishman Tests

Initially the cumulative means of the data points obtained from simulation experiments were used as input data for the Fishman tests; but this data was highly autocorrelated and consequently very large samples were demanded by the Fishman tests. This approach was abandoned, therefore, and the tests were conducted using the instantaneous values obtained from simulation experiments.

The results obtained from successive applications of the Fishman technique on the simulation output for the utilization factors of 0.5, 0.6 and 0.7 are given in Appendix A. It was possible to obtain fair results for the utilization factor 0.5, but for the utilization factors of 0.6 and 0.7 the tests demand a very high sample size. The values of the sample mean and the variance of this sample are fairly consistent, but the order of the autoregressive scheme and the size of the sample fluctuate considerably from test to test. The plot in Appendix E shows the inconsistent fluctuation of the autocorrelation coefficients for some successive tests. These values correspond to a utilization factor of 0.5 and a reporting time interval of 10 time units.

The plots in Appendix E show the effect of the utilization factor on the autocorrelation between observations. It is clear from the plots that the autocorrelation between observations increases as the utilization factor of the process is increased. The plots correspond to the utilization factors of 0.5, 0.6 and 0.7 and a reporting time interval of 25 time units.

Next it was desired to study the effect of the reporting time interval on the results of successive Fishman tests. Plots in Appendix E show the change in the autocorrelation due to the change in the reporting time interval. The tests were conducted with reporting time intervals of 10, 25, 50, 100 and 200 time units and the utilization factor of the process was 0.5.

It is obvious from the plots that the correlation between observations decreases as the reporting time interval is increased, since the autocorrelation between observations drops as the number of intervening events increases. The Fishman tests also show that the required sample size increases considerably as the reporting time interval is made larger. Data obtained at higher reporting time intervals cannot be represented by an autoregressive scheme and consequently the Fishman technique which is built essentially for autocorrelated data, fails to give meaningful results. There is a possibility that the order of the autoregressive scheme is higher than 10, but a maximum order of scheme of 10 was chosen for our experiments because, according to Fishman, an order of scheme of 10 should be more than adequate to represent the autoregressive scheme.

The Fishman tests were found inconclusive for locating the end of the transient phase for the type of data obtained from our simulation experiments. The technique appears to be extremely sensitive to the autocorrelation between observations and the data to be tested by this technique will have to be chosen carefully so as to have some consistent autoregressive representations.

4.5 Conclusions

1. The pseudorandom generator, using the multiplicative congruence method, is acceptable for simulation experiments.

2. The simulation conducted with the antithetic variate approach approximately halves the length of the transient period. It does not appear to be an effective technique for reducing the computer cost involved in simulation experiments.

3. The reporting time interval of the simulation experiments has a significant effect on the nature of the simulation output. As the reporting time interval is increased the representation of the time series changes from an autoregressive scheme to a moving average scheme.

4. The Fishman technique is sensitive to the autocorrelation between the observations of the data sample. Successive applications of the Fishman test on the simulation output do not give conclusive information regarding the termination of the transient phase.

5. The Fishman technique is applicable strictly to autocorrelated data and will not give any meaningful results for any other type of data.

APPENDIX A

The Results

1. The Simulation Efficiency.

2. The Results of the Fishman Test.

A.1 The Simulation Efficiency

The simulation efficiencies are calculated from the plots of the cumulative means for simulation A, simulation B, and the mean value simulation.

TABLE I

Utilization Factor	Simulation Efficiency
0.5	58.63%
0.6	58.47%
0.7	54.94%

No values are available for utilization factors of 0.8 and 0.9 because the output statistic did not converge to a steady value.

A.2 The Results of the Fishman Test

TABLE I

Results of the Fishman tests for different reporting time intervals.

Utilization factor = 0.5

The results are for the mean value simulation.

TINT = 10

Test No.	$T^{\dagger}$	T	$\bar{X}$ Mean	$\sigma_{\bar{X}}$ Std. Dev. of Sample Mean	Ord. of Scheme	No. of Iterations
1	250	500	83.46	12.86	2	1
2	340	500	83.46	15.18	1	1
3	200	500	83.46	11.58	2	1
4	480	620	92.20	16.00	7	2
5	530	590	85.98	17.36	1	2
6	1310	1910	100.47	15.08	2	9
7	3340	3360	93.56	18.13	2	8
8	760	870	89.17	17.08	1	7
9	420	500	83.46	16.80	2	1
10	780	4280	98.69	7.77	2	2
11	910	1240	94.35	15.61	1	2
12	640	690	90.55	17.61	1	4
13	760	970	88.81	16.16	2	2
14	770	3080	96.04	9.14	2	3
15	680	700	89.62	18.00	1	3
16	90	500	83.46	18.11	10	1
17	450	500	83.46	17.28	1	1
18	100	500	83.46	8.31	5	1
19	2400	3890	96.44	14.30	2	5
20	550	670	91.43	16.56	5	3

$T^{\dagger}$ is the time corresponding to the required sample size.

T is the time corresponding to the sample size fed in.

* Indicates incomplete tests due to insufficient data points.

TABLE I (Continued)

TINT = 25

Test No.	$T^{\dagger}$	T	$\bar{X}$ Mean	$\sigma_{\bar{X}}$ Std. Dev. of Sample Mean	Ord. of Scheme	No. of Iterations
1	1425	1425	250.94	18.24	0	6
2	1725	1725	262.59	18.22	0	9
3	5925	5925	279.28	18.18	10	7
4	6225	6225	282.03	18.20	1	6
5	6425	6475	276.43	18.14	1	6
6	17225	41075	262.35	11.78	8	7
7	2075	3275	238.17	14.48	4	4
8	4225	5550	268.43	15.86	2	3
9	13150	13175	260.87	18.17	8	11
* 10	46700	16400+	-	-	1	1

TINT = 50

Test No.	$T^{\dagger}$	T	$\bar{X}$ Mean	$\sigma_{\bar{X}}$ Std. Dev. of Sample Mean	Ord. of Scheme	No. of Iterations
1	62950	63200	522.62	18.15	2	11
* 2	47000	37350+	-	-	1	10

TINT = 100

Test No.	$T^{\dagger}$	T	$\bar{X}$ Mean	$\sigma_{\bar{X}}$ Std. Dev. of Sample Mean	Ord. of Scheme	No. of Iterations
*	247800	97200+	-	-	1	3

TINT = 200

Test No.	$T^{\dagger}$	T	$\bar{X}$ Mean	$\sigma_{\bar{X}}$ Std. Dev. of Sample Mean	Ord. of Scheme	No. of Iterations
*	359800	126600+	-	-	1	1

A.2 The Results of the Fishman Test (Continued)

TABLE II

Results of the Fishman test for different utilization factors, for the independent simulations and the mean value simulation.

Reportive time interval TINT = 25

Utilization factor = 0.5

Results for the mean value simulation are given in TABLE I.

A-simulation

Test No.	$T^{\dagger}$	T	$\bar{X}$ Mean	$\sigma_{\bar{X}}$ Std. Dev. of Sample Mean	Ord. of Scheme	No. of Iterations
1	14625	14625	226.53	18.20	1	8
2	48350	53125	263.07	17.35	4	10
3	12450	16575	229.74	15.77	4	8
* 4	238450	1250	-	-	1	1

B-simulation

Test No.	$T^{\dagger}$	T	$\bar{X}$ Mean	$\sigma_{\bar{X}}$ Std. Dev. of Sample Mean	Ord. of Scheme	No. of Iterations
1	22100	38425	258.50	13.79	2	6
2	24875	51525	260.16	12.64	2	3
3	3800	3800	238.50	18.19	0	8
* 4	171750	68750	-	-	1	3

A.2 The Results of the Fishman Test (Continued)

TABLE II (Continued)

Utilization factor = 0.6

Mean Value Simulation

Test No.	$T^{\dagger}$	T	$\bar{X}$ Mean	$\sigma_{\bar{X}}$ Std. Dev. of Sample Mean	Ord. of Scheme	No. of Iterations
1	19025	19025	370.61	18.19	2	13
* 2	54725	54550	-	-	3	9
A-simulation						
* 1	135525	98975	-	-	9	4
B-simulation						
* 1	74000	72900	-	-	3	8

Utilization factor = 0.7

Mean Value Simulation

	177050	1250	-	-	7	1
A-simulation						
	548400	82775	-	-	3	3
B-simulation						
	362600	91300	-		3	2

APPENDIX B

The Simulation

1. FORTRAN Listing of the Simulation Program.

2. Sample Plot of the Cumulative Means.

ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

B.1 FORTRAN Listing of the Simulation Program

Main Program

```
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
     1NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
     2TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
     3ATRIB(10),ENQ(14),INN(14),JCELS(10,30),KRANK(14),MAXNQ(14),MFE(14)
     4,MLC(14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
     50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14),
     6XL,XMU,XISYS(2),TLD(2),TAU,SKALE,ISIM,IND,NCA,NEXP,JAREA(12)
      DIMENSION NSET(500),QSET(500)
      NCRDR = 1
      NPRNT = 3
      IND = 0
      NCA = 0
      XISYS(1) = 0.0
      XISYS(2) = 0.0
      TLD(1) = 0.0
      TLD(2) = 0.0
      READ 10,XL,XMU,TAU,SKALE,NEXP
      UFACT = XL/XMU
   10 FORMAT(4F10.5,I5)
   20 FORMAT('OMEAN INTERARRIVAL TIME  =',T50,F10.5/' MEAN SERVICE TIME
     1  =',T50,F10.5/' UTILIZATION FACTOR  =',T50,F10.5/' SCALE  =',T50,
     2F10.5/' REPORTING INTERVAL  =',T50,F10.5/' END OF SIMULATION AT  '
     3,T50,F10.2/' NUMBER OF EXPERIMENT  =',T50,I5)
      CALL GASP(NSET,QSET)
      WRITE(3,20)XL,XMU,UTFACT,SKALE,TAU,TFIN,NEXP
      CALL EXIT
      END
```

Subroutine EVNTS

```
      SUBROUTINE EVNTS(IXX,NSET,QSET)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
     1NOQ,NORPT,NCT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
     2TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
     3ATRIB(10),ENQ(14),INN(14),JCELS(10,30),KRANK(14),MAXNQ(14),MFE(14)
     4,MLC(14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
     50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14),
     6XL,XMU,XISYS(2),TLD(2),TAU,SKALE,ISIM,IND,NCA,NEXP,JAREA(12)
      DIMENSION NSET(1),QSET(1)
      ISIM = JTRIB(2)
      GO TO (1,2,3),IXX
    1 CALL ARRVL(NSET,QSET)
      RETURN
    2 CALL ENDSV(NSET,QSET)
      RETURN
    3 CALL REPRT(NSET,QSET)
      RETURN
      END
```

Subroutine ENDSV

```
      SUBROUTINE ENDSV(NSET,QSET)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
     1NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
     2TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
     3ATRIB(10),ENQ(14),INN(14),JCELS(10,30),KRANK(14),MAXNQ(14),MFE(14)
     4,MLC(14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
     50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14),
     6XL,XMU,XISYS(2),TLD(2),TAU,SKALE,ISIM,IND,NCA,NEXP,JAREA(12)
      DIMENSION NSET(1),QSET(1)
      Z = XISYS(ISIM)
      CALL TMST(Z,TNOW,ISIM,NSET,QSET)
      TLD(ISIM) = TNOW
      XISYS(ISIM) = XISYS(ISIM) -1.0
      IF(NQ(ISIM + 1))7,8,9
    7 CALL ERROR(41,NSET,QSET)
    8 RETURN
    9 MFEZ=MFE(ISIM+1)
      CALL RMOVE(MFEZ,(ISIM+1),NSET,QSET)
      ATRIB(1) = TNOW -XMU*ALOG(RANI(ISIM +2))
      JTRIB(1) = 2
      JTRIB(2) = ISIM
      CALL FILEM(1,NSET,QSET)
      RETURN
      END
```

Subroutine OTPUT

```
      SUBROUTINE OTPUT(NSET,QSET)
      COMMON IO,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
     1NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
     2TBEG,TFIN,MXX,NPRNT,NCROR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
     3ATRIB(10),ENQ(14),INN(14),JCELS(10,30),KRANK(14),MAXNQ(14),MFE(14)
     4,MLC(14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
     50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14),
     6XL,XMU,XISYS(2),TLD(2),TAU,SKALE,ISIM,IND,NCA,NEXP,JAREA(12)
      DIMENSION NSET(1),QSET(1)
      RETURN
      END
```

B.1 FORTRAN Listing of the Simulation Program (Continued)

Subroutine REPRT

```
      SUBROUTINE REPRT(NSET,QSET)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
     1NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
     2TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
     3ATRIB(10),ENQ(14),INN(14),JCELS(10,30),KRANK(14),MAXNQ(14),MFE(14)
     4,MLC(14)  ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
     50,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14),
     6XL,XMU,XISYS(2),ILD(2),TAU,SKALE,ISIM,IND,NCA,NEXP,JAREA(12)
      DIMENSION NSET(1),QSET(1)
      DO 100 JSIM = 1,2
      ISIM = JSIM
      XAS=XISYS(ISIM)
      CALL TMST(XAS,TNOW,ISIM,NSET,QSET)
      IND = IND + 1
      AREA = SSUMA(ISIM,2)
      JAREA(IND) = AREA*SKALE
  100 CONTINUE
      IF (IND - 12)7,6,6
    6 NCA = NCA + 1
      PUNCH 99,(JAREA(IND),IND = 1,12),NCA,NEXP
      IND = 0
    7 ATRIB(1) = TNOW + TAU
      CALL FILEM(1,NSET,QSET)
      SSUMA(1,2) = 0.0
      SSUMA(2,2) = 0.0
   99 FORMAT(12I6,2I4)
      RETURN
      END
```

Function RANI

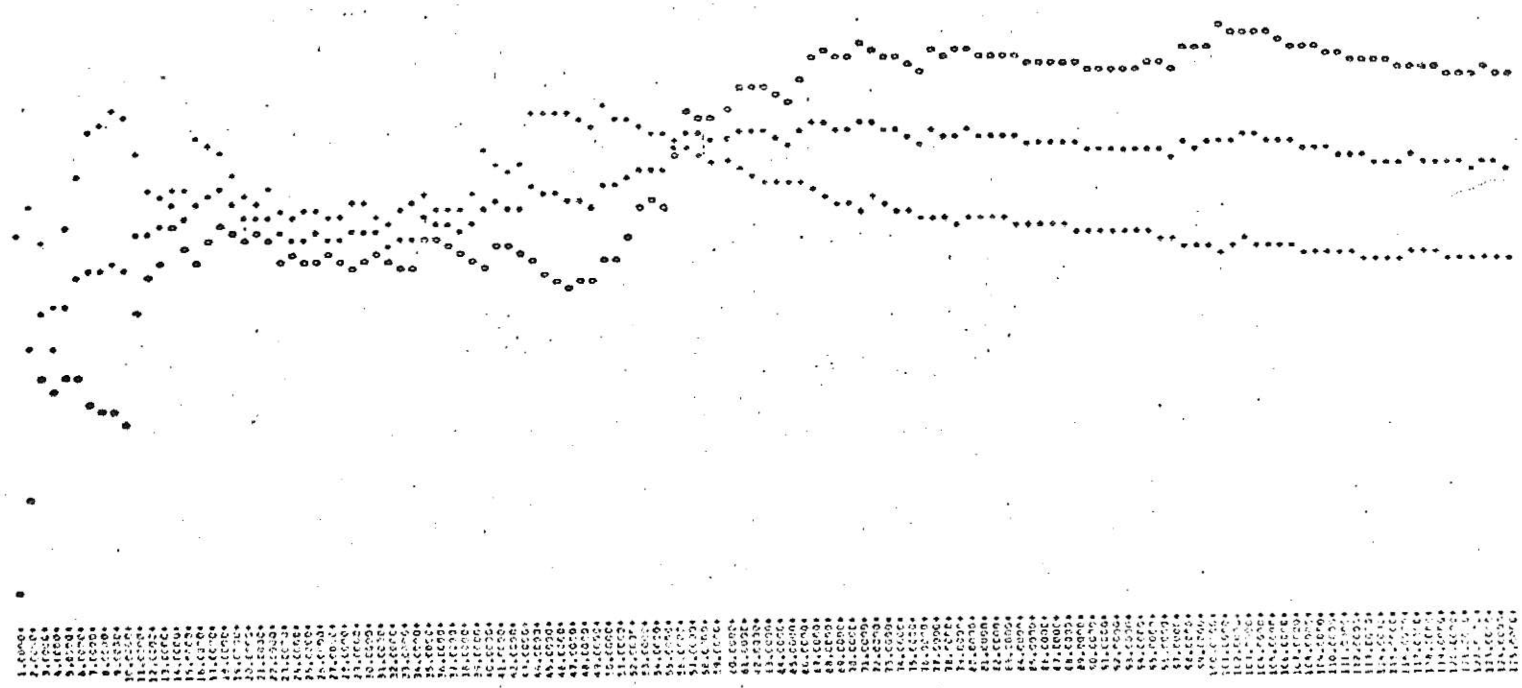
```
      FUNCTION RANI(I)
      COMMON ID,IM,INIT,JEVNT,JMNIT,MFA,MSTOP,MX,MXC,NCLCT,NHIST,
     1NOQ,NORPT,NOT,NPRMS,NRUN,NRUNS,NSTAT,OUT,TNOW,
     2TBEG,TFIN,MXX,NPRNT,NCRDR,NEP,VNQ(14),IMM,MAXQS,MAXNS,
     3ATRIB(10),ENQ(14),INN(14),JCELS(10,30),KRANK(14),MAXNQ(14),MFE(14)
     4,MLC(14) ,NCELS(5),NQ(14),PARAM(20,4),QTIME(14),SSUMA(10,5),SUMA(1
     5C,5),NAME(6),NPROJ,MON,NDAY,NYR,JCLR,JTRIB(12),IX(8),MLE(14),
     6XL,XMU,XISYS(2),TLD(2),TAU,SKALE,ISIM,INU,NCA,NEXP,JAREA(12)
      DIMENSION NSET(1),QSET(1)
      IX(I) = IX(I)*65539
      IF(IX(I).LT.0) IX(I) = IX(I)+2147483647+1
      RANI  = IX(I)*0.4656613D-9
      IF(MOD(I,2).EQ.0) RANI=1.C -RANI
      RETURN
      END
```

0000 A-Simulation

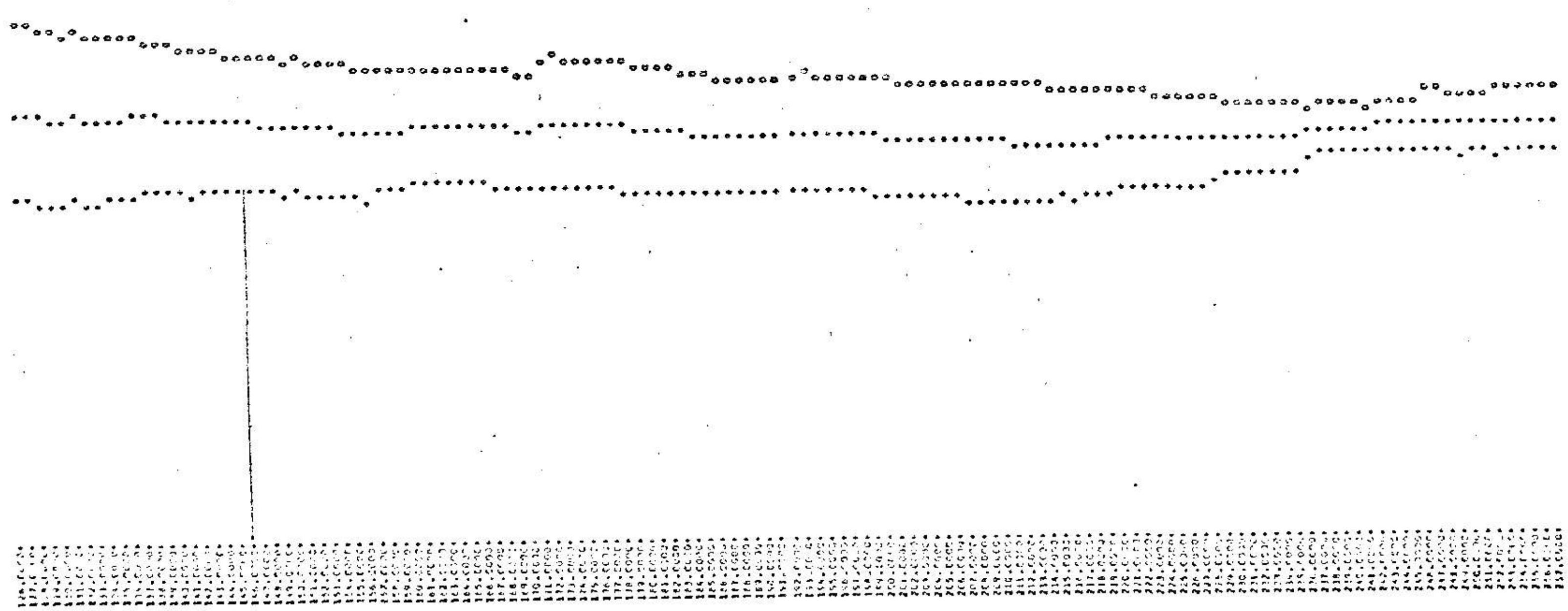
++++ B-Simulation

**** Mean Value Simulation

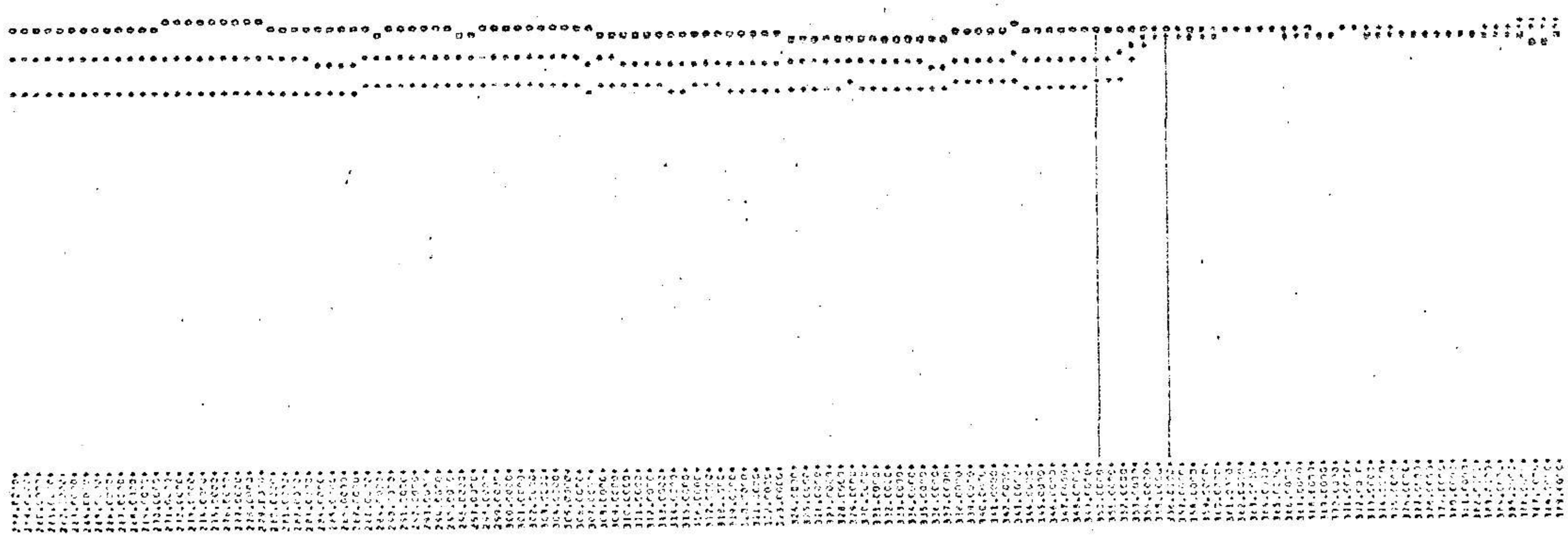
B.2 Sample Plot of the Cumulative Means



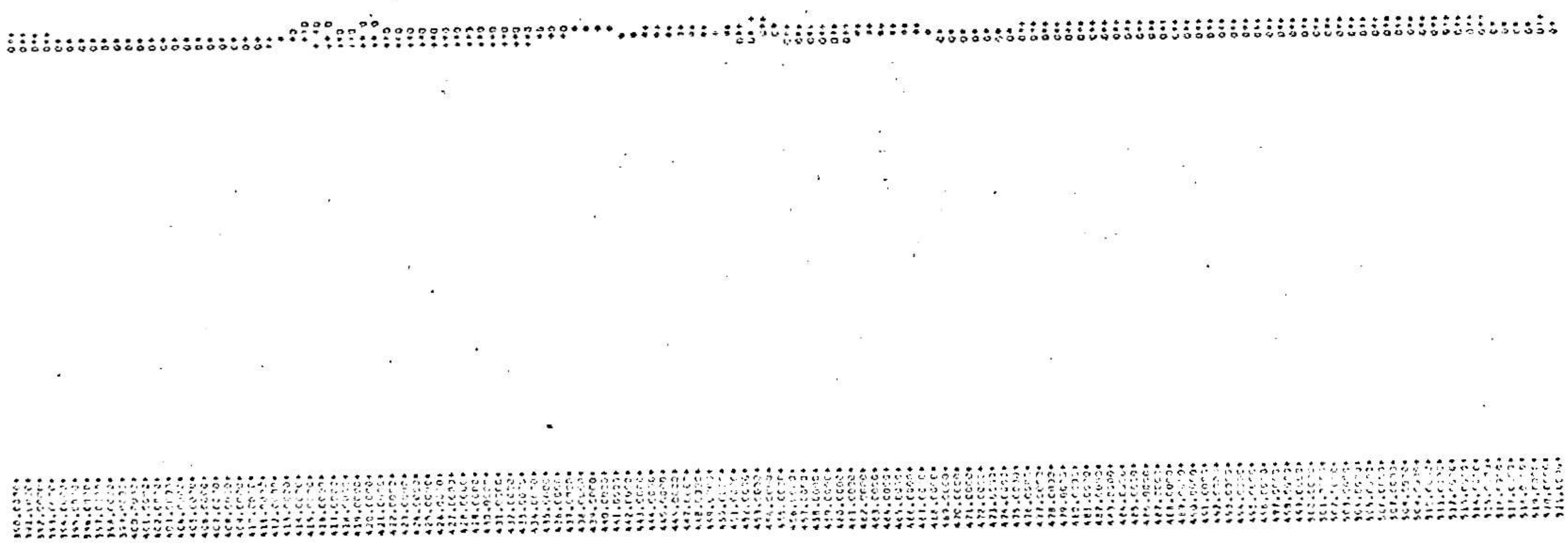
B.2 Sample Plot of Cumulative Means (Continued)



B.2 Sample Plot of Cumulative Means (Continued)



B.2 Sample Plot of Cumulative Means (Continued)



APPENDIX C

Testing of Random Numbers

1. FORTRAN Listing of the Chi-square Test for Generated Random Numbers and the Results of this Test.
2. Confidence Interval for the Plot of the Autocorrelation Function.
3. Plot of the Autocorrelation Function.
4. Confidence Interval for the Power Spectrum.
5. Plot of the Spectrum.

```
IV G LEVEL. 18                  MAIN                 DATE = 71159            09/44/04

          DIMENSION IRANGE(100)
          READ 999,IG
      999 FORMAT(I10)
          READ 101,NMBRS,NORNGE,TBLCH1
      101 FORMAT(2I5,F10.3)
          DO 125  I=1,NORNGE
      125 IRANGE(I)=0
          DO 102  I=1,NMBRS
          IY=IG*65539
          IF(IY) 108,103,103
      108 IY= IY + 2147483647 + 1
      103 YFL=IY
          YFL=YFL * 0.4656613D-09
          L=YFL*NORNGE+1
          IRANGE(L)=IRANGE(L)+1
      102 IG=IY
          SUMCH1=0.0
          AN1=NMBRS
          AN2=NORNGE
          Y=AN1/AN2
          DO 201  I=1,NORNGE
      201 SUMCH1=SUMCH1+((IRANGE(I)-Y)**2)/Y
          WRITE(3,303) NMBRS,NORNGE
      303 FORMAT('           DISTRIBUTION OF',I6,'  UNIFORM RANDOM NUMBERS IN
         1',I6,' INTERVALS',//)
          N=NORNGE/10-1
          DO 304 I=1,10
          L=10*(I-1)+1
          M=L+N
      304 WRITE(3,305)(IRANGE(K),K=L,M)
      305 FORMAT(/,10X,10I6)
          WRITE(3,306)SUMCH1,TBLCH1
      306 FORMAT(//,'            OBS. CHI-SQ. VALUE=',F8.3,'   CHI-SQ.VALUE FR
         1OM TABLES = 'F8.3,//)
          IF(SUMCH1-TBLCH1)307,307,308
      307 WRITE(3,309)
      309 FORMAT('       OBS. CHI-SQ. VALUE BEING SMALLER THAN TABLE VALUE TE
         1ST FOR UNIFORM RANDOMNESS IS SUCCESSFUL')
          GO TO 318
      308 WRITE(3,310)
      310 FORMAT('       OBS. CHI-SQ. VALUE BEING GREATER THAN TABLE VALUE TE
         1ST FOR UNIFORM RANDOMNESS IS UNSUCCESSFUL')
      318 STOP
          END
```

C.1 FORTRAN Listing of the Chi-square Test for Generated Random Numbers and the Results of this Test (Continued).

```
DISTRIBUTION OF  4000  UNIFORM RANDOM NUMBERS IN   100 INTERVALS

   36   39   41   31   24   35   37   36   37   48
   37   41   29   44   42   37   42   34   34   63
   44   43   44   52   36   52   36   26   41   33
   31   50   34   47   40   46   30   42   36   37
   48   36   49   50   37   47   35   31   44   34
   47   52   34   44   39   40   45   49   37   40
   43   51   39   26   39   45   37   39   29   50
   39   41   42   40   36   34   37   38   40   51
   39   41   44   33   35   37   45   36   25   33
   30   44   35   48   43   41   51   52   41   46

OBS. CHI-SQ. VALUE= 120.700   CHI-SQ.VALUE FROM TABLES =  123.000

OBS. CHI-SQ. VALUE BEING SMALLER THAN TABLE VALUE TEST FOR UNIFORM RANDOMNESS IS SUCCESSFUL
```

C.2 Confidence Interval for the Autocorrelation Function

For a large sample the standard deviation $\sigma_p = \frac{1}{\sqrt{N}}$

where, N is the sample size

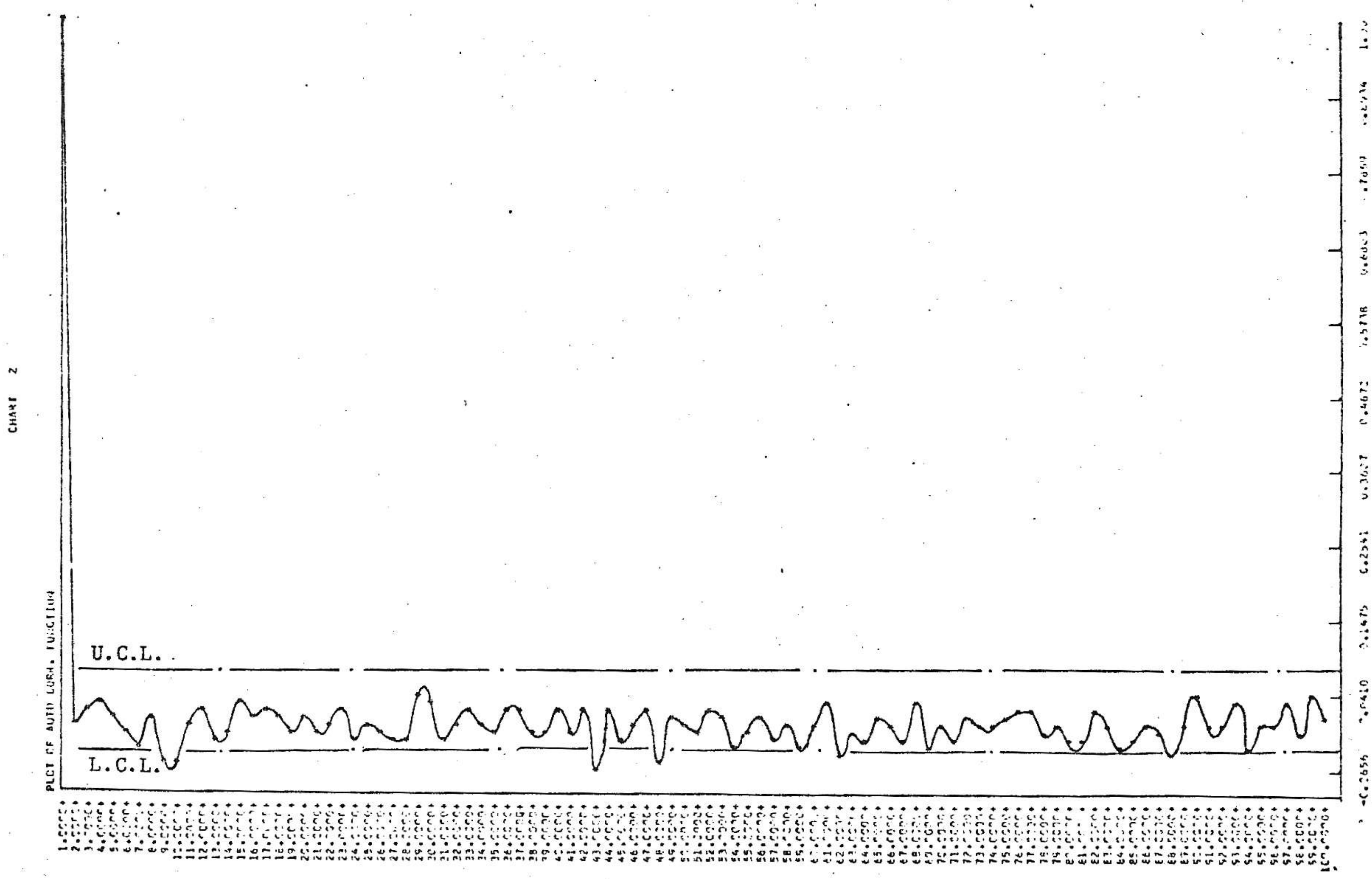
The standard deviation for the data at lag 0 $= \frac{1}{\sqrt{2000}} = 0.02235$

The confidence interval $\pm\ 2\ \sigma = \pm\ 0.0447$

The standard deviation for the data at lag 100 $= \frac{1}{\sqrt{1900}} = 0.0229$

The confidence interval $\pm\ 2\ \sigma = \pm\ 0.0458$

C.3 Plot of the Autocorrelation Function



C.4 Confidence Interval for the Power Spectrum

The degrees of freedom

$\nu = 2\ b\ (N/M)$

where,

N = Number of data points

M = Number of lags

b = Constant for the type of window

For the Tukey window $b = 4/3$

$$\nu = 2 \times \frac{4}{3} \times \frac{2000}{100} = 53.3 \approx 53$$

Using a 99% confidence level,

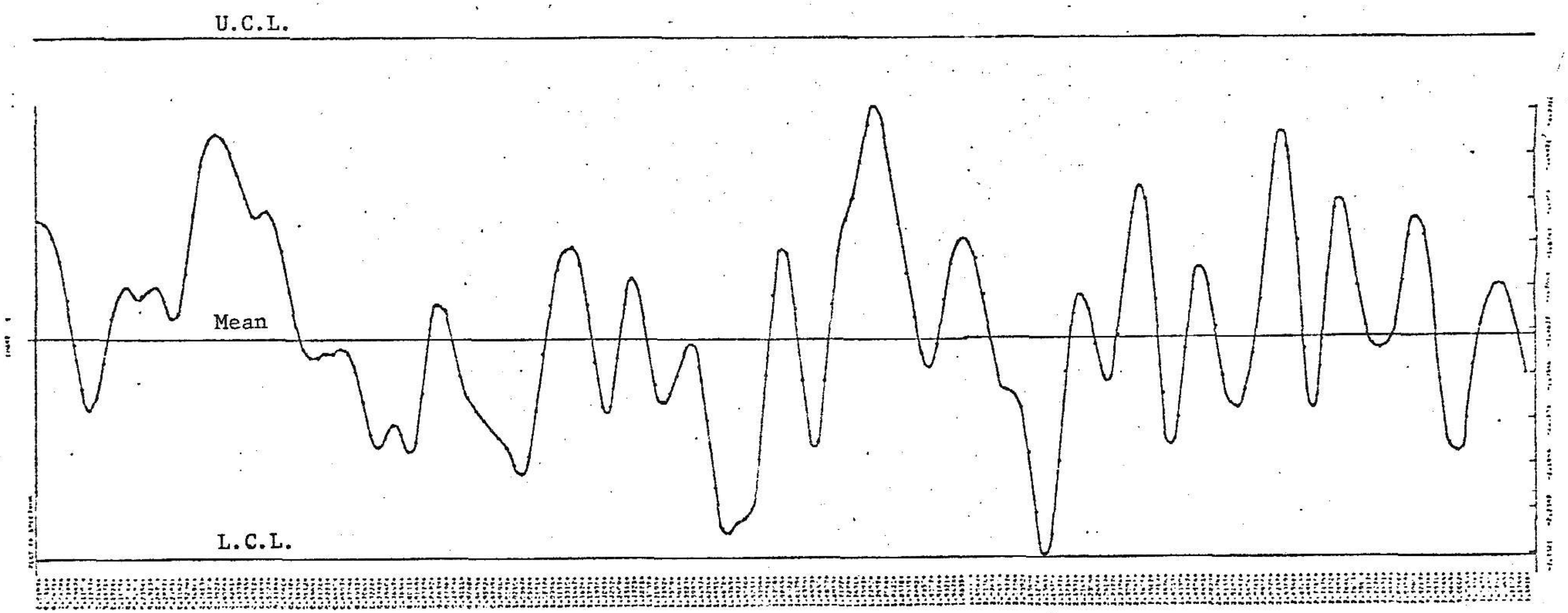
$$\chi^2_{0.995}\ (53) = 82.25 \qquad \ln(53/82.25) = \ln(0.664) = -0.440$$

$$\chi^2_{0.005}\ (53) = 30.25 \qquad \ln(53/30.25) = \ln(1.750) = 0.560$$

The lower control limit = $-1.870 - 0.440 = -\ 2.31$

The upper control limit = $-1.870 + 0.560 = -\ 1.31$

C.5 Plot of the Spectrum.

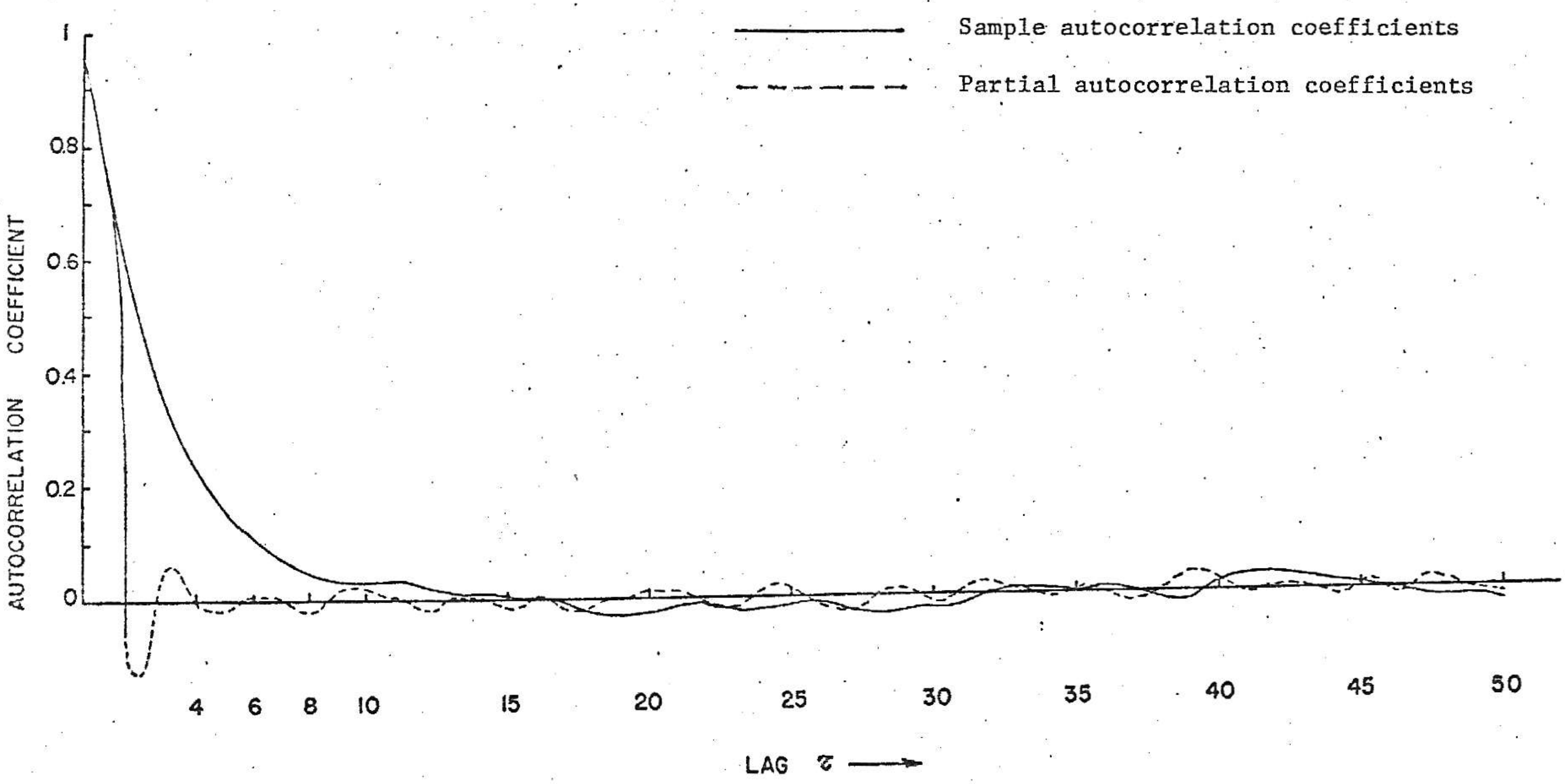


APPENDIX D

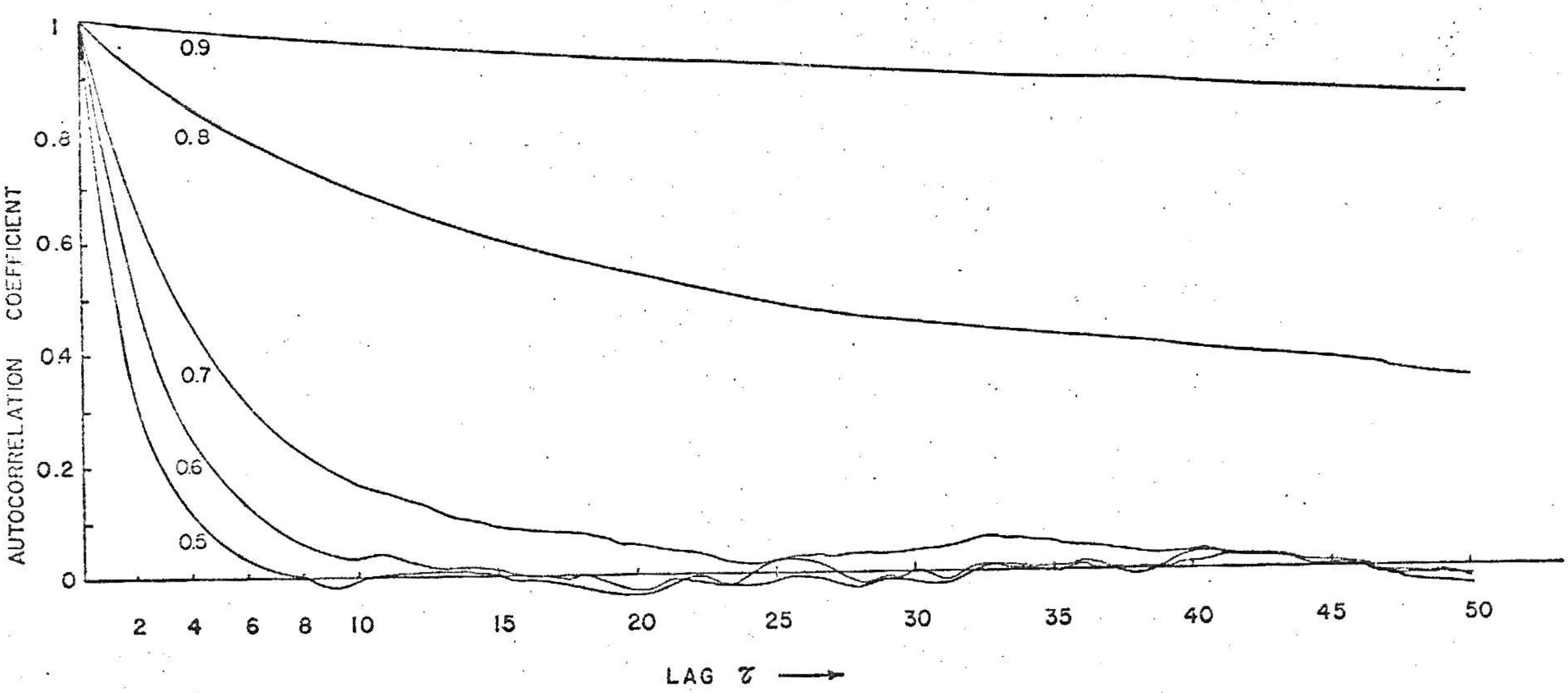
Box and Jenkins Procedure

1. Plot of the Autocorrelation Coefficients and the Partial Autocorrelation Coefficients.
2. Effect of the Utilization Factor on the Autocorrelation Coefficients.
3. Effect of the Reporting Time Interval on the Autocorrelation Coefficients.
4. Effect of the Reporting Time Interval on the Partial Autocorrelation Coefficients.

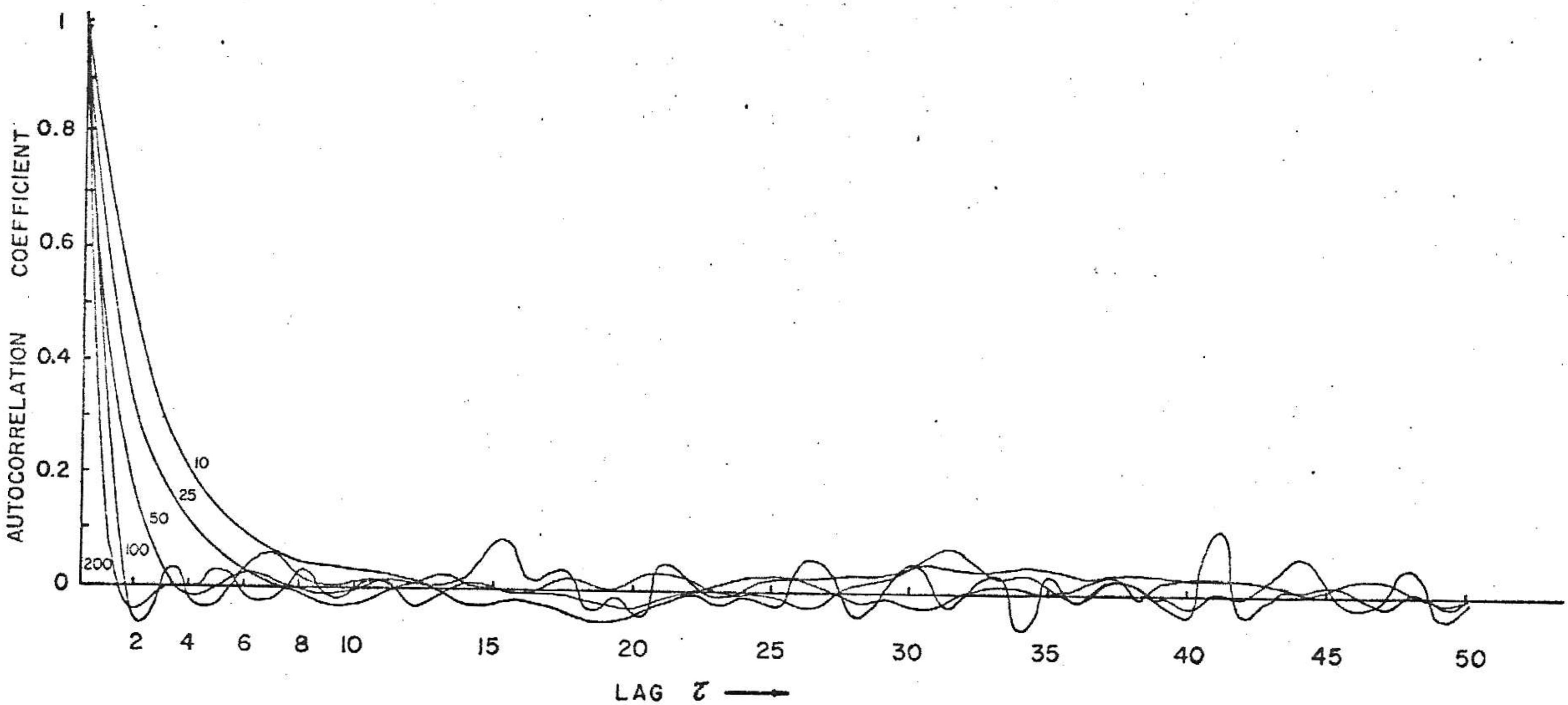
D.1 Plot of the Autocorrelation Coefficients and Partial Autocorrelation Coefficients



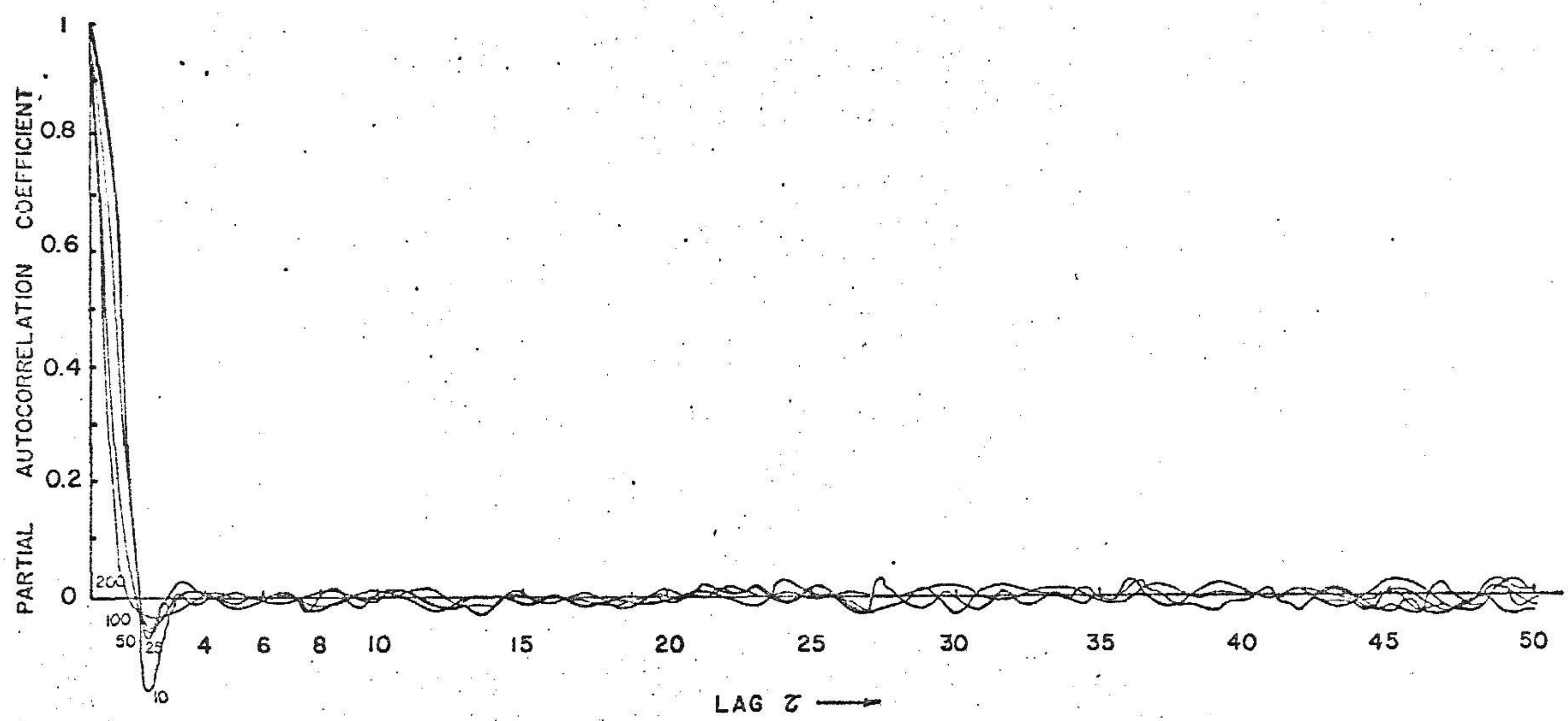
D.2 Effect of the Utilization Factor on the Autocorrelation Coefficients



D.3 Effect of the Reporting Time Interval on the Autocorrelation Coefficients



D.4 Effect of the Reporting Time Interval on the Partial Autocorrelation Coefficients



APPENDIX E

The Fishman Technique

1. FORTRAN Listing of the Fishman Technique.
2. Plot of the Autocorrelation Coefficients for Successive Fishman Tests.
3. Effect of the Utilization Factor on the Autocorrelation Coefficients.
4. Effect of the Reporting Time Interval on the Autocorrelation Coefficients.
5. Block Diagram of the Fishman Technique.

E.1 FORTRAN Listing for the Fishman Technique

```
      DIMENSION IX(20),XMEAN(20),X(5000),A(5000),B(5000)
      DIMENSION AX(5000),BX(5000),CX(5000)
      COMMON IM,IRP,P,C,GMA,CONF,NEXT,NBASE,XBARR,NOBS
      READ (5,150)IM,KARDS,IRP,P,C,GMA,CONF,DELTA,TAU,TINT
  150 FORMAT(I5,I10,I5,7F10.3)
  101 FORMAT(12I6)
      I=0
      DO 50 J=1,KARDS
      READ(5,101)(IX(N),N=1,12)
      DO 60 M=1,12,2
      I=I+1
      L=M+1
      A(I)=IX(M)
      B(I)=IX(L)
   60 CONTINUE
   50 CONTINUE
      NADD=TINT/TAU
      NOBS=KARDS*6./NADD
      I=0
      DO 10 J=1,NOBS
      ASUM=0.0
      BSUM=0.0
      DO 20 K=1,NADD
      I=I+1
      ASUM=ASUM+A(I)
      BSUM=BSUM+B(I)
   20 CONTINUE
      AX(J)=ASUM
      BX(J)=BSUM
      CX(J)=(AX(J)+BX(J))/2.0
      X(J)=CX(J)
   10 CONTINUE
      NBASE = 0
      NUM=6*KARDS
      DO 154 J=1,20
      NEW = NUM - NBASE
      WRITE(6,153) J
  153 FORMAT('1** TEST NUMBER',2X,I3,'   **')
      CALL FISHMN(X)
      NBASE = NBASE + NEXT
      XMEAN(J) = XBARR
      IF(J.EQ.1) GO TO 154
      IF(ABS(XMEAN(J)-XMEAN(J-1)).GT.DELTA) GO TO 154
      WRITE(6,155) XMEAN(J)
  155 FORMAT(////' MEAN VALUE OF THE STATISTICS',/' FOUND BY RECURRENCE
     1TEST IS',T40,F10.2)
  154 CONTINUE
      STOP
      END
```

E.1 FORTRAN Listing for the Fishman Technique (Continued)

```
      SUBROUTINE FISHMN(X)
      DIMENSION X(1),XP(5000),C(20),B(20,20),VAR(20),OMEGA(20)
      COMMON IM,IRP,P,Q,GMA,CONF,NEXT,NBASE,XBARR,NOBS
      IRC = IRP + 1
      K = 0
      KOUNT = 0
      N=IM
    1 TOT = 0.0
      KOUNT = KOUNT + 1
      WRITE(6,103) KOUNT
  103 FORMAT(////' ** ITERATION NUMBER',2X,I3,'  **')
      LT = K+ 1
      WRITE(6,1100)LT,N,NBASE
 1100 FORMAT(3I10)
      NX = LT + NBASE
      NY = N + NBASE
      IF(NY.GT.NOBS) STOP
      DO 10 L=NX,NY
      TOT=TOT+X(L)
   10 CONTINUE
      XBAR = TOT/(N-K)
      DO 20 J=NX,NY
      XP(J)=X(J)-XBAR
   20 CONTINUE
      DO 30 J = 1,IRC
      SUMA = 0.0
      JJ = J -1
      NZ=NY-JJ
      DO 40 JK=NX,NZ
      SUMA = SUMA + XP(JK)*XP(JK + JJ)
   40 CONTINUE
      C(J) = SUMA/(N-K)
      B(J,1) =1.0
   30 CONTINUE
      WRITE(6,5100)TOT,XBAR,NX,NY
 5100 FORMAT(' TOT='T20,F20.2,/' XBAR =',T20,F10.2,/' NX =',T20,I5,/' NY
     1 =',T20,I5)
      DO 2000 LL = 1,IRC
      WRITE(6,900)C(LL)
  900 FORMAT(20X,F20.2)
 2000 CONTINUE
      DO 50 IR =1,IRP
      SV = 0.0
      SW = 0.0
      DO 60 IS = 1,IR
      SV = SV + B(IR,IS)*C(IS)
      SW = SW + B(IR,IS)*C(IR-IS+2)
   60 CONTINUE
      B(IR+1,IR+1) = -SW/SV
      IF(IR.EQ.1) GO TO 50
      DO 70 JS=2,IR
      B(IR+1,JS)=B(IR,JS)+B(IR+1,IR+1)*B(IR,IR-JS+2)
   70 CONTINUE
   50 CONTINUE
      DO 1000 II=1,IRC
      WRITE(6,910)(B(II,JJ),JJ=1,II)
  910 FORMAT(10F10.3)
 1000 CONTINUE
      VAR(1)=0.0
      OMEGA(1)=0.0
```

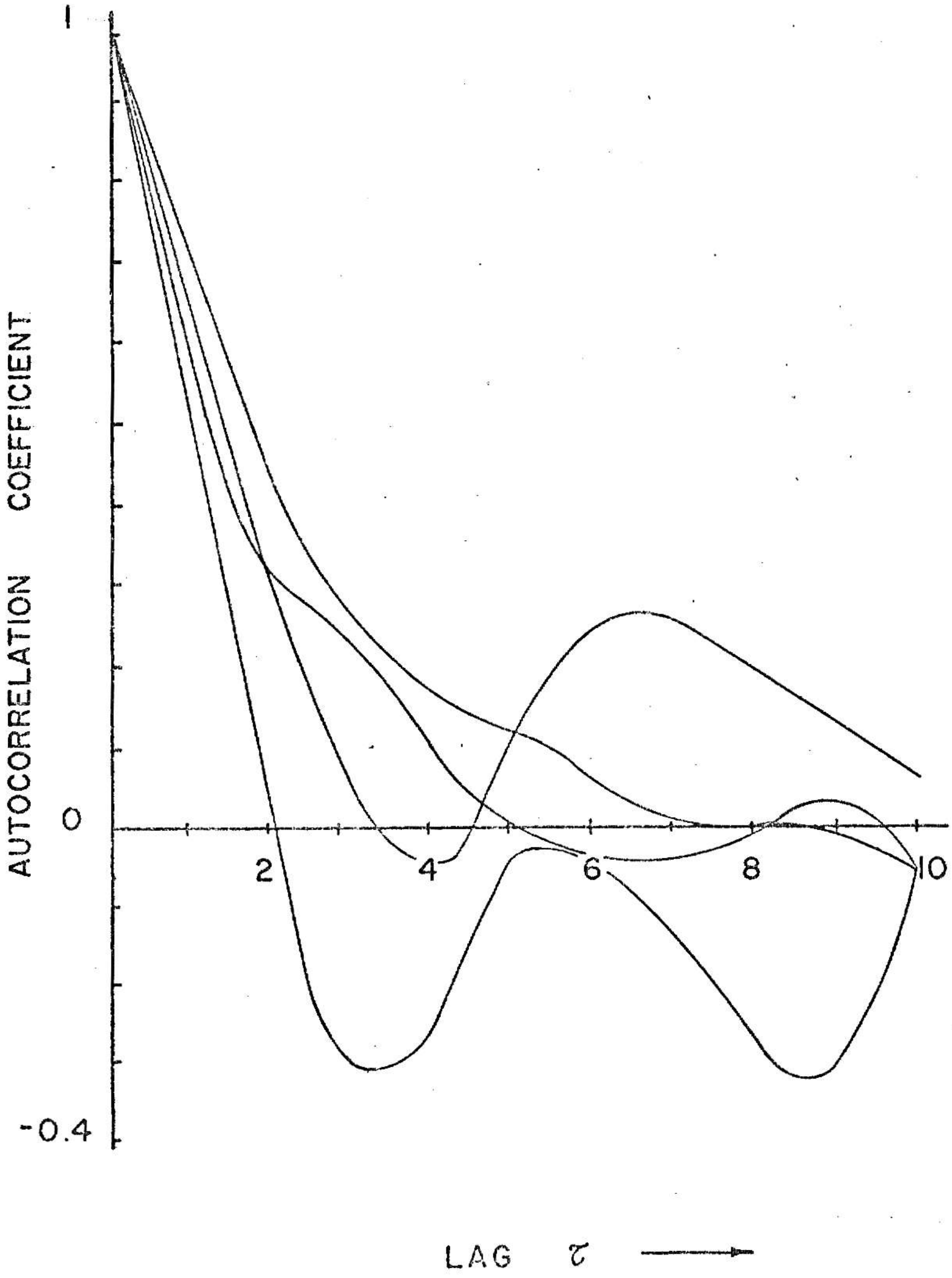
E.1 FORTRAN Listing for the Fishman Technique (Continued)

```
      DO 300 I=2,IRC
      IK=I-1
      NT=N+1
      SUMB=0.0
      DO 310 KT=I,NT
      SUMC=0.0
      DO 320 KS=1,IK
      SUMC=SUMC + B(I,KS)*(X(KT-KS+NBASE) - XBAR)
  320 CONTINUE
      SUMB=SUMB + SUMC**2
  310 CONTINUE
      VAR(I) = SUMB/(N-K)
      OMEGA(I)=1.0 - B(I,I)**2
  300 CONTINUE
      DO 3000 I=1,IRC
 3000 WRITE(6,4000) VAR(I),OMEGA(I)
 4000 FORMAT(20X,F10.3,20X,F10.3)
      IP=1
      JP=0
      LR=1
  400 LR=LR + 1
      TEST=P*(OMEGA(LR)/(N-K))**0.5
      A1=B(LR,LR) + TEST
      A2=B(LR,LR) - TEST
      IF((A1.GT.0.).AND.(A2.LT.0.)) GO TO 410
      IP=LR
  410 IF(LR.LT.IRC) GO TO 400
      JP = IP-1
      WRITE(6,5000)JP
 5000 FORMAT(' ORDER OF THE SCHEME IS  ',T50,I3)
      IF(IP.EQ.1) GO TO 500
      BP=0.0
      DO 600 I=1,IP
      BP=BP + B(IP,I)
  600 CONTINUE
      EM=VAR(IP)/(BP**2)
C     K=EM/C(1)
      WRITE (6,1500) K
 1500 FORMAT(' THE BIAS ADJUSTMENT IS  ',T50,I5)
      GO TO 610
  500 EM=C(1)
      K=0
  610 VE=(CONF/Q)**2
      NSTAR=EM/VE
      WRITE(6,2500) NSTAR,N
 2500 FORMAT(' REQUIRED SAMPLE SIZE IS  ',T50,I10,/' ACTUAL SAMPLE SIZE
     1IS  ',T50,I10)
      IF (NSTAR.LE.(N-K)) GO TO 710
      M=GMA*(NSTAR-N+K) + 1.
      N=GMA*(NSTAR+K) + (1.0-GMA)*N + 1.
      WRITE(6,2100)N
 2100 FORMAT(//' DEFECIENT SAMPLE SIZE =',T50,I10)
      GO TO 1
  710 SUMD=0.0
      MK=K+1
      DO 800 L=MK,N
      SUMD=SUMD + X(L)
  800 CONTINUE
      XBAR=SUMD/(N-K)
      VXBAR=EM/(N-K)
```

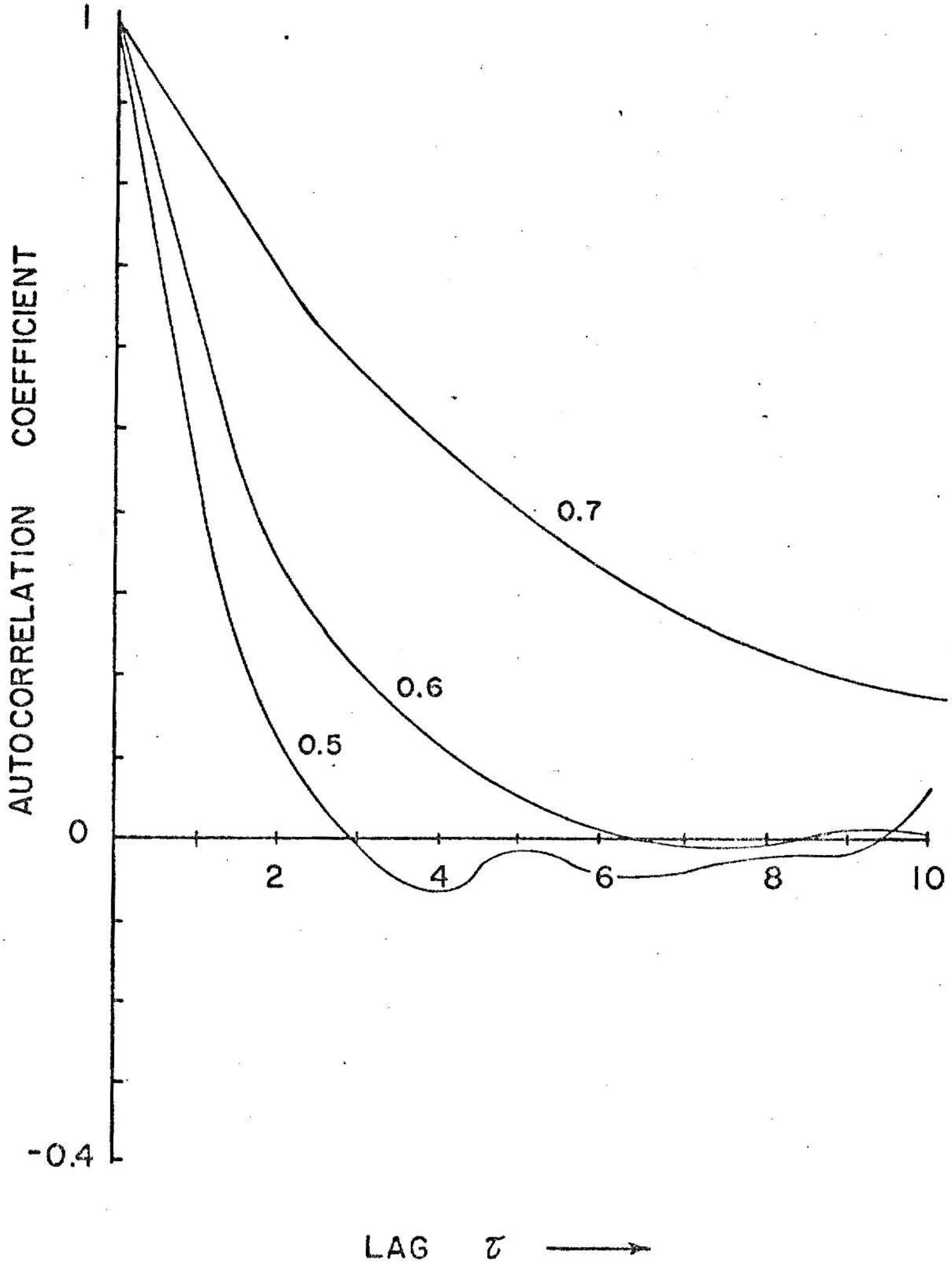
E.1 FORTRAN Listing for the Fishman Technique (Continued)

```
      STDEV = VXBAR**0.5
      WRITE(6,6000)XBAR,VXBAR,STDEV
 6000 FORMAT(' MEAN OF THE SAMPLE IS  ',T50,F10.2,/' VARIANCE OF THE SAM
     1PLE MEAN IS  ',T50,F10.2,/' STANDARD DEVIATION IS',T50,F10.2)
      NEXT = N
      XBARR = XBAR
      RETURN
      END
```

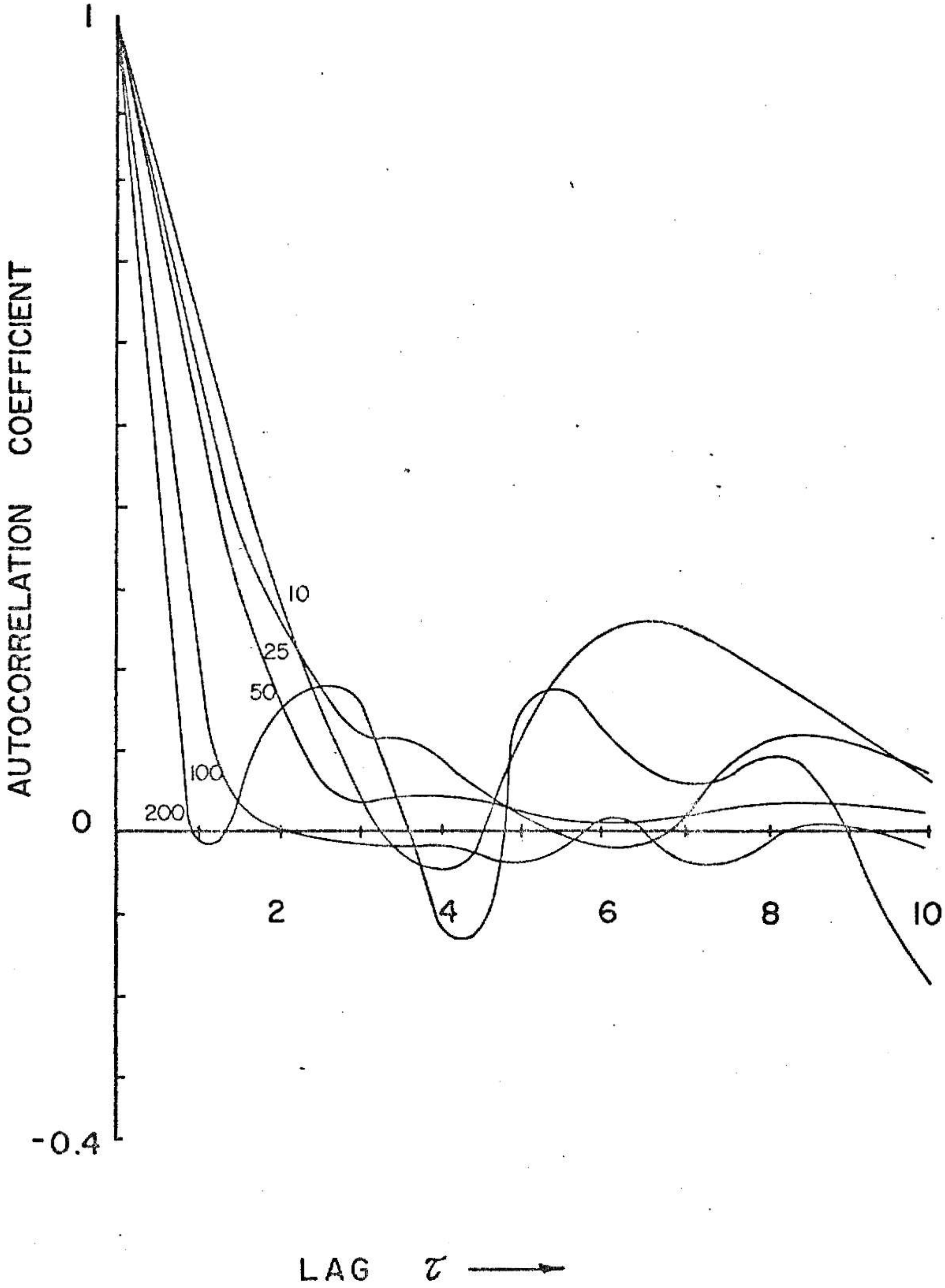
E.2 Plot of the Autocorrelation Coefficients for Successive Fishman Tests



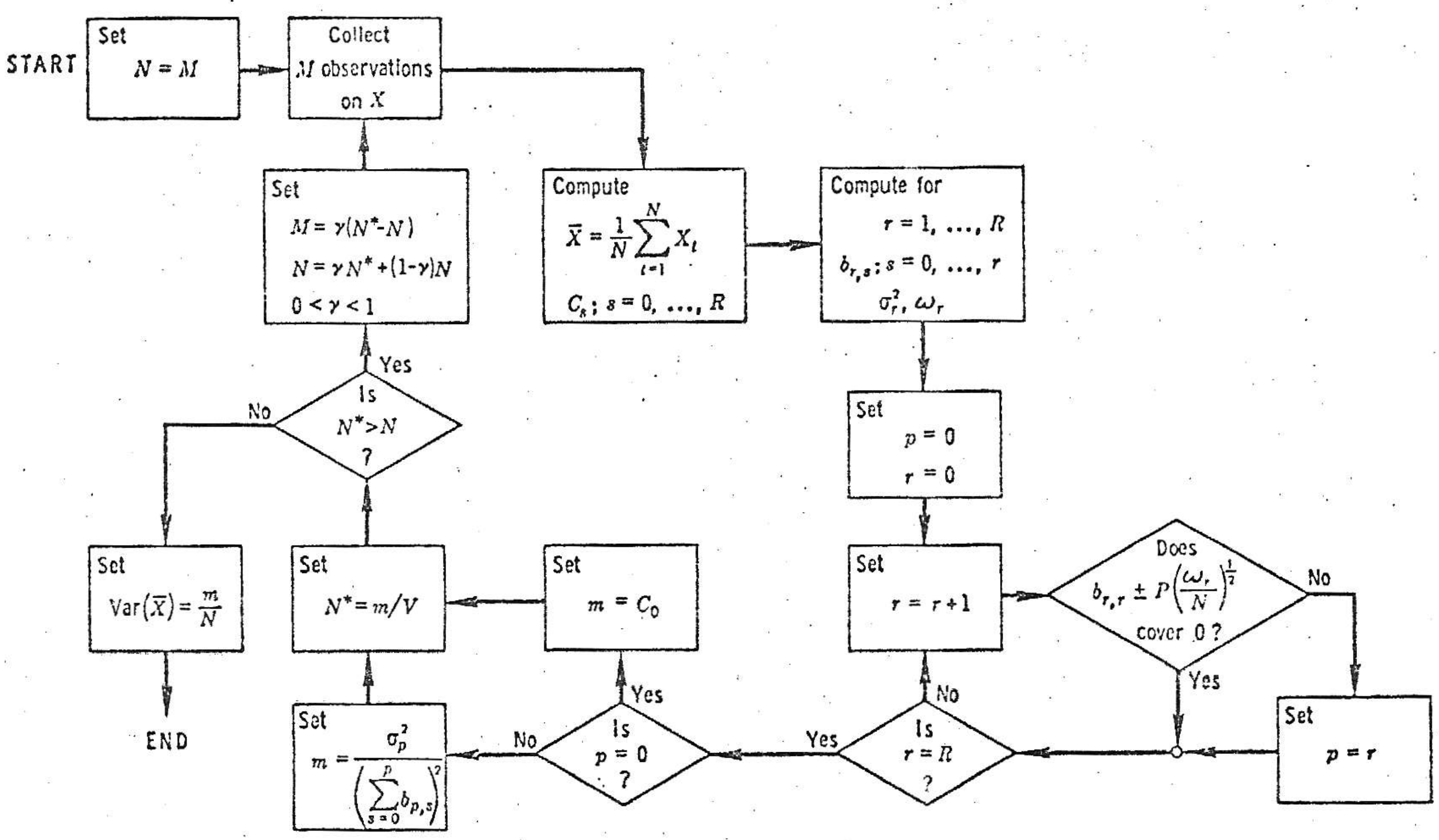
E.3 Effect of the Utilization Factor on the Autocorrelation Coefficients



E.4 Effect of the Reporting Time Interval on the Autocorrelation Coefficients



E.5 Block Diagram of the Fishman Technique



BIBLIOGRAPHY

1. Box, G.E. and Jenkins, G.M. Time Series Analysis, Holden-Day Inc., San Francisco, California, 1970.

2. Beuno, R.J. "Some Practical Solutions to Two Statistical Problems in Simulation", Master of Science Thesis in Engineering, University of California, Los Angeles, 1966.

3. Conway, R.W. "Some Tactical Problems in Digital Simulation", Management Science, Vol 10, No. 1, 1963.

4. Coveyou, R.R. and MacPherson, R.D. "Fourier Analysis of Uniform Random Number Generators", J.A.C.M., Vol. 14, No. 1, January, 1967.

5. Cox, W.R. and Smith, W.L. Queues, London Methuen and Co., Ltd., 1961.

6. Diananda, P.H. "Some Probability Limit Theorems with Statistical Applications", Proc. Camb. Phil. Soc. XLIV, 1953.

7. Durbin, J. "The Fitting of Some Time Series Models", Review Inst. Int. Stat., Vol 28:3, 1960.

8. Ehernfeld, S. and Ben Tuvia, S. "The Efficiency of Statistical Simulation Procedure", Technometrics, Vol 2, May, 1962.

9. Emshoff, J.R. and Sisson, R.L. Computer Simulation Models, The MacMillan Company, London, 1970.

10. Fishman, G.S. "Problems in the Statistical Analysis of Simulation Experiments: The Comparison of Means and the Length of Sample Records", The Rand Corporation, RM 4880-PR, February, 1966.

11. Fishman, G.S. and Kiviat, P.J. "Digital Computer Simulation: Statistical Considerations", The Rand Corporation, RM 5387-PR, November, 1967.

12. Fishman, G.S. "Digital Computer Simulation: Statistical Considerations", The Rand Corporation, RM 5866-PR, August, 1969.

13. Gaver, D.P. "Statistical Methods for Improving Simulation Efficiency", Third Conference on Applications of Simulation, December, 1969.

14. Gaver, D.P. "Diffusion Approximations and Methods for Certain Congestion Problems", J. of Applied Probability, 5, 1968.

15. Good, I.J. "The Serial Test for Sampling Numbers and Other Tests for Randomness", Proc. Camb. Phil. Soc., Vol. XLIX, 1953.

16. Gordon, G. System Simulation, Prentice-Hall Inc., 1969.

17. Hammersley, J.M. and Handscomb, D.C. Monte Carlo Methods, London Methuen and Co. Ltd., 1964.

18. Hammersley, J.M. and Morton, K.W. "A New Monte Carlo Technique: Antithetic Variates", Proc. Camb. Phil. Soc., 52, 1956.

19. Harris, F.R. "A Study of the Transient Stage Problems in Simulations", Master of Science Thesis in Industrial Engineering, Kansas State University, Manhattan, Kansas, 1971.

20. Kendall, M.G. and Smith, B.B. "Randomness and Random Sampling Numbers", J. of Roy. Soc., Statistic Section, 101, 1938.

21. Ko, S.S. "Application of General Autoregressive Moving Average Stochastic Models to Time Series and Simulation Problem", Master of Science Thesis in Industrial Engineering, Kansas State University, Manhattan, Kansas, 1971.

22. Naylor, T.N., Wertz, K. and Wonacott, T.N. "Methods for Analyzing Data from Computer Simulation Experiments", Comm. A.C.M., Vol. X, No. 1, November, 1967.

23. Newell, G.F. "Queues with Time Dependent Arrival Rates", J. of Applied Probability, 5, 1968.

24. Page, E.S. "On Monte Carlo Methods in Congestion Problems", Operations Research, Vol. 13, No. 2, March-April, 1965.

25. Pritsker, A.A.B. and Kiviat, P.J. "Simulation with GASP II A", Prentice-Hall Inc., 1969.

26. Reese, P.A. "Steady State Parameter Estimation in Computer", Arizona State University, Bulletin No. 5, January, 1969.

27. Whittle, P. Prediction and Regulation by Linear Least Square Methods, The English Universities Press, London, 1963.

AN EXPERIMENTAL INVESTIGATION

OF THE TRANSIENT PHASE

IN DIGITAL SIMULATION

by

SUNIL VASANT KIRTANE

B.S., Ranchi University, India, 1964

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY

Manhattan, Kansas

1971

ABSTRACT

A single server queueing system was simulated on the computer using GASP II A. The random generator used for the simulation experiments was tested for independency. A subroutine with this generator was used for generating antithetic variates for the simulation. Utilization factors of 0.5, 0.6, 0.7, 0.8 and 0.9, and reporting time intervals of 10, 25, 50, 100 and 200 time units were tried.

The nature of the output was studied by using the Box and Jenkins procedure. It showed that the process is of an autoregressive nature for small reporting time intervals and of a moving average nature for larger intervals. It also showed that the autocorrelation between observations increased as the utilization factor of the process was increased.

The Fishman technique was applied to the successive samples from simulation outputs. The Fishman technique was found very sensitive to the nature of the data and the autocorrelation between observations. The successive tests did not give any conclusive results regarding the end of the transient phase in simulation experiments.