A STUDY OF THE TRANSIENT STAGE PROBLEM IN SIMULATIONS

by

FRED RODNEY HARRIS

B.A., Southwestern College, 1968

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

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.

TABLE OF CONTENTS

INTRODUCTION

Operations Research is a body of knowledge used to assist in the decisions necessary for the efficient operation of organized systems. The field is concerned with the problems of how to conduct or coordinate the activities of large and usually complex organizations. The approach used to solve the problems encountered is the scientific approach. There are many descriptions of the scientific approach. Hillier and Lieberman [11] give a ~~the~~ simple~~st~~:

> 1. The structuring of the real life situation into a mathematical model ... so that a solution relevant to the decision-maker's objectives can be sought.
>
> 2. Exploring the structure of such solutions and developing systematic procedures for obtaining them.
>
> 3. Developing a solution that yields an optimal value of the (decision-maker's) objective.

The approach is three-fold. It requires the development of a model, evaluation of the model, and determining the best possible course of action from the model. It may not be necessary to proceed through the entire sequence of this approach for the solution to the problem may be found in any of the three steps.

Simulation is a technique used both in developing a model and evaluating the model. Simulation is a numerical technique for performing experiments upon a mathematical model of the system. The three general cases which call for the use of simulation are (1) analysis of a system which has a very complex mathematical and statisical nature, (2) analysis of systems not yet in existence, and (3) evaluation of possible alterations of existing systems.

The system under consideration is subdivided into a number of subsystems, each of which can be easily described mathematically or statistically. Programming these subsystems causes the results to be discrete. For this reason, it is not possible to obtain a continuous mathematical expression of any of the processes under consideration. Since all continuous processes must be approximated numerically, a true optimal solution cannot be found using simulation.

By creating a number of subsystems and allowing them to interact, the investigator is able to study a large complex system. The investigator must select measures of performance which he will observe. It is important that measures of performance be selected which are pertinent to the decision-maker's objective. It is possible to observe the effect on the entire system of a change in any of the subsystems. In this way, possible alternatives are studied. It is also possible to create subsystems which do not yet exist in the real world or may never exist.

Simulation is a useful technique; therefore, studies of the problems encountered when using this technique will be valuable to operations researchers. One of the problems encountered is the transient stage. The transient stage of a simulation is the initial period of the simulation when the mean and variance of the measures of performance are erratic and not true to the real-life system. The transient stage problem is the subject of this paper.

STATEMENT OF THE PROBLEM

In the past it has been observed that simulations initially go through a "warming-up" period. During this period of time, the system's measures of performance are quite unstable and erratic in behavior. Once this warm-up or transient stage is past, the simulation is said to be in equilibrium.

The problem to be considered is the estimation of the length of the transient stage. Unfortunately, due to the experimental nature of simulations, it is not possible to statistically forecast the length of the transient stage. The problem becomes one of determining when the simulation has reached equilibrium. We shall state the problem as the estimation of the termination of the transient stage.

Since the transient stage is not stable, the output during this stage is not usually desirable. It is to the experimenter's advantage to know when the simulation has passed through the transient stage. It is possible to get a rough idea of when the simulation has reached equilibrium within a reasonable degree of accuracy by doing the simulation for a long period of time and examining the output. This can be quite expensive in terms of time and computer costs.

The problem will be constrained to require that the solution be dynamic. A dynamic solution is one which will indicate whether or not equilibrium has been reached while the simulation is still running on the computer.

A second constraint comes from the definition of equilibrium. Equilibrium, according to Cox and Mize [5], is achieved "when the system's measured performance varies only within an acceptable and predictable range". This definition implies that the simulation must be stable in both mean and variance. It is easy to see that a constant mean may be associated with an erratic variance. We must therefore find a method of estimating termination of the transient stage which considers both mean and variance.

It is suggested that a possible approach to the problem can be found through spectral analysis. Spectral analysis does not adjust for autocorrelation, rather, it decomposes the variance so the effect of autocorrelation can be studied. Autocorrelation is a measure of the relationship between X_t and $X_{t+\tau}$. The suggestion is to evaluate spectral analysis as a possible technique in determining when the simulation has attained equilibrium.

It is proposed that by comparing spectra from successive segments of a simulation-generated time series of parameter estimates, it can be determined whether or not equilibrium has been attained. When there is no longer a significant difference between spectra of adjacent segments, it can be said that there is no longer a significant difference in the segments' variance or that equilibrium has been achieved.

LITERATURE SURVEY

Simulation is a well-established method of evaluating a system. The common problem of determining the length of the transient stage has been encountered by all who have used the technique. Statistical methods of solving the problem have only recently been suggested.

Previously the experimenter generally followed the advice given by Conway [3] who recommended making several pilot runs and examining the results. Upon examining the results, the experimenter may either pre-load the system at the expected value of the parameters, or determine the average length of the transient stage and begin data collection at that point in the next simulation.

Conway's approach requires repetitive simulations, which increases computer costs, and does not satisfy the need for a well-defined solution of the problem. Bueno [2] realized the need for more study of the problems involved in the stabilization of computer simulation. His preliminary work indicated that much could be learned from such studies and the work was indeed necessary. He suggested approaching the problem by comparing sample means with a grand mean through the student's t-test. Fishman [6,7] also became interested in the problem and began work regarding the autocorrelation nature of simulations. He first showed that simulation-generated time series are highly autocorrelated [7] and established an application of spectral analysis to such time series. He indicated in the second paper [6] how the problem of obtaining independent observations might be resolved and how the changing nature of the autocorrelation affects the simulation.

Reese [20] combined the ideas of Bueno and Fishman by introducing a dynamic approach to the problem. He suggested a sequential t-test of estimated parameter means in searching for stability. This method is dynamic, thus eliminating the necessity of repetitive simulations. It is important to have a dynamic solution to the problem; the question of whether or not a simulation has achieved stability should be answered in regard to some analysis of the sample variance, as well as the sample mean. It is possible for repetitive samples having a stable mean to be erratic in variance. The simulation cannot be thought of as being in equilibrium according to the definition presented above.

Let us first review the work done by Bueno, Fishman, and Reese.

SOME PRACTICAL SOLUTIONS TO TWO STATISTICAL PROBLEMS IN SIMULATION

Ramon J. Bueno

The purpose of Bueno's paper is to propose a solution for two problems encountered in simulation work. The problems dealt with are: (1) estimating the length of the transient period and (2) estimation of the length of run required to produce a desired precision in the statistic.

The procedure chosen for estimating the transient period varies according to the type of simulation. The three types of simulations considered are: (1) systems with endogenously independent periods, (2) systems with exogenously independent periods, and (3) systems with continuous and inter-dependent periods. By endogenously independent, Bueno means the system repeatedly reverts back to an initial stage independent of the previous period. The length of simulation time between stages is not a function of time, but depends on the nature of the system. For example, an inventory system which reverts back to a maximum stock level when stock reaches some minimum stock level. The exogenously independent system is one which reverts back to an initial stage periodically, such as retail sales systems in which cash-on-hand reverts back to some level at the beginning of each day. The simulation time between stages is a function of time. The continuous system is one which operates without reverting back to an initial stage. For example, a continuously operating production line is a continuous system.

For simulations of types (1) and (2) , Bueno proposes using the student's t-test to determine the periods biased by transient behavior. The simulation is divided into n time periods. The statistic's mean within the i^{th} period, $\bar{x}_i$, is compared to the statistic's grand mean for periods i+1 through the n^{th} period, $\bar{\bar{x}}_{i+1}$. Bueno suggests discarding the first period as it is known to contain transient bias.

Application of the student's t-test requires three criterion:

1. The observations, the $\bar{x}_i$'s, must be independent.
2. The observations must be normally distributed.
3. The known values, the $\bar{\bar{x}}_{i+1}$'s, must be normally distributed.

Bueno states that since the $\bar{x}_i$'s and $\bar{\bar{x}}_{i+1}$'s are calculated from several single observations, they are normally distributed by virtue of the Central Limit Theorem. It should be mentioned that if the several single observations are autocorrelated, the assumption of independence is not justified.

In applying the student's test to simulations of the third type, Bueno suggests that by observing the change in the sample variance from period to period, a correction factor, δ_i , can be applied to the mean. This correction factor is based on the sum of the correlation of period i, P_i , with all remaining periods. Bueno asserts that by correcting for the autocorrelation of the system, the three criterion of the student's t-test are met.

Fishman [6] has pointed out that successive statistic means of simulation-generated time series are highly autocorrelated. Reese [20] also found that observations of simulation statistics are highly autocorrelated. Similar results were encountered in this study, as will be shown later in this paper.

In dealing with autocorrelation, one must consider the source of the autocorrelation. It may be that the system being simulated is autocorrelated, that the simulation may induce autocorrelation, or that both may be true. It is possible that the time interval between observations is short and the system does not change between observations. This would induce autocorrelation that is not in the real system. Studies by Fishman [7] have shown that the length of the transient stage is related to the amount of autocorrelation in the simulation.

THE ANALYSIS OF SIMULATION-GENERATED TIME SERIES

George S. Fishman and Philip J. Kiviat

This paper presents the first published application of spectral analysis to simulation-generated time series. The application of spectral analysis is used for comparison of simulations of queueing systems differing only in queue management. Fishman and Kiviat first present a discussion of why spectral analysis can and should be applied to simulation-generated time series. They then give a brief development of spectral theory, and its application to single-channel queueing systems. They conclude that the differences in the statistical properties of queueing systems can be easily identified by using spectral analysis.

The relevance of their work to this paper is in Fishman and Kiviat's justification of the use of spectral analysis in studying simulation output and their interpretation of the simulation in the light of spectral analysis.

Fishman and Kiviat have two reasons for using spectral analysis in studying simulation-generated time series. First, as previously mentioned, simulation data are autocorrelated; thus, the investigator cannot apply commonly used statistical tools. They point out that some investigators attempt to adjust for autocorrelation. The criticism of this is twofold. One, they doubt that methods used to remove autocorrelation actually accomplish the purpose; and, two, doing away with autocorrelation removes information about the system with which the investigator should be concerned. This leads to their second reason for advocating spectral analysis. The purpose of simulating a system is to study a stochastic process. The investigator is interested in the average level of activity, deviations from this level, and the length of time the deviation lasts. Since spectral analysis is concerned with estimating the occurrence of cyclic elements and the period of these elements, spectral analysis is particularly suited for the investigator's needs.

In their analysis of the simulation-generated time series, Fishman and Kiviat first plot both the autocorrelation function and the spectral density function. By doing this for two simulations, a visual comparison of the graphs indicates to the investigator where statistical analysis of the differences would be most fruitful. For example, they choose to compare the queue length of two simulated queueing systems. One system operated with FIFO queue management and the other operated under SHOPN queue management. By using the sample mean and variance from a segment of the time series after equilibrium had been obtained, they place confidence limits on the mean queue size. By assuming that the spectral function was normal $\log[f(\lambda)] \sim N\{\log[f(\lambda)], \psi(M,T)\}$, the confidence interval $P[e^{-\Phi} \leq \hat{f}(\lambda)/f(\lambda) \leq e^{-\Phi}] = 1-\alpha$ was derived. The estimated spectrum at frequency λ is $f(\lambda)$. $\psi(M,T) = 2M/3T$, where M = the number of lags in the estimate and T = total simulation time, and $\Phi = P(\alpha/2)[\psi(M,T)]$
The confidence interval was used to determine whether or not $f(\lambda)$ may be considered to have been drawn from a process with $f(\lambda)$ as the spectral density function.

By observing both the autocorrelation function and the spectral density functions, the investigator can estimate what cyclic elements are present in each of the simulations. By comparing these, he can gain some insight as to the similarities or differences between the processes being studied. By using the null hypothesis and the confidence intervals mentioned, he will be able to test for a statistical difference between the two simulations. This is clearly of aid to one interested in comparing two simulations. The application of these methods toward finding the end of the transient stage remains to be seen. It is noted that Fishman and Kiviat did not use these methods in determining the end of the transient stage in their examples; however, they were not interested in this problem. It would be interesting to apply cross-spectral analysis to the problem of comparing simulation-generated time series from two simulations.

PROBLEMS IN THE STATISTICAL ANALYSIS OF SIMULATION EXPERIMENTS: THE COMPARISON OF MEANS AND THE LENGTH OF SAMPLE RECORDS

George S. Fishman

This paper presents a technique for comparing the means of two simulations. The paper is of importance to this work because of Fishman's study of the spectra of simulation-generated time series and the discussion concerning autocorrelation.

Fishman points out that simulation-generated time series are highly autocorrelated. He has found that the variance of the sample mean for autocorrelated data is inversely proportional to a fraction of the number of observations. This fraction is related to the autocorrelated properties of the simulation. The fractional number of observations can be regarded as the equivalent independent observations.

The number of equivalent independent observations is related to the strength of autocorrelation over τ^* units. Fishman terms τ^* the correlation time, which is a measure of the interval of time over which the time series is autocorrelated and outside of which the time series is not correlated with itself. Correlation time τ^* is a measure of the strength of autocorrelation relative to time. The importance of correlation time is that simulation-generated time series are autocorrelated and the autocorrelation is not constant throughout the simulation. The amount of autocorrelation changes as the simulation progresses and it seems to decrease, as was the case in this study.

The correlation time is defined as $\tau^*=\pi f(0)/2$, where $f(0)$ is the spectral density function. Since correlation time is a measure of the longest effect of autocorrelation, the zero frequency is used. This value is not known and in order to estimate it, we use spectral analysis to find $G(0)$, the spectral distribution function. Since $G(\lambda) = R_o f(\lambda)$ and R_o is the covariance, we can estimate $\hat{\tau}^* = \pi G(0)/2R_o$.

Fishman also points out that two simulations of the same system will likely have different values of τ^*. He suggests that each simulation be evaluated in the light of its unique τ^*. The importance of this can be illustrated by considering two simulations of the same system, both having the same variance for a particular statistic but with differing τ^*. The simulation with a higher τ^* will have fewer changes during any given length of time and fewer equivalent independent observations. We could not expect as good an estimate of the parameter mean due to the larger value of τ^*. The simulation with a lower value of τ^* will have a greater change over the given period of time and more equivalent independent observations, thus a better estimate of the parameter mean.

The important points, from this view, are that each simulation has a unique autocorrelation function and that the autocorrelation function for the simulation changes with time.

STEADY STATE PARAMETER ESTIMATION IN COMPUTER SIMULATED SYSTEMS

Peter A. Reese

This work suggests a method for determining the termination of the transient stage of a simulation. The technique is the application of sequential t-testing to observations made at m intervals. The technique was evaluated using a queueing model and the theoretical value of the waiting time.

Reese relies on (though does not mention) Fishman's concept of correlation time. He suggests that by selecting a lag time of length m, which corresponds to the longest autocorrelation interval, and sampling only observations separated by m time units, the samples are independent and normally distributed. This is consistent with Fishman's development of equivalent independent observations.

The sequential t-test is designed for making a decision regarding a population parameter by taking repetitive samples. A simple hypothesis is set up, H_o: $\theta=\theta_o$ against H_1: $\theta\neq\theta_o$. Decisions can be made continuously in time using the sequential t-test. The decision criterion is

$B < \frac{\theta_o}{\theta_1} f(x:\theta) < A$ where $f(x:\theta)$ is the probability density function of which θ is a parameter and $A = \alpha/1-\beta$, $B = \beta/1-\alpha$. If the sample statistic is less than B, accept H_o; and if the sample statistic is greater than A, reject H_o. If the sample statistic is between A and B, no decision can be made.

Reese develops the sequential t-test for a normally distributed sample mean. If the sample statistic is λ_n, the test is $B < \lambda_n < A$

where

$$\lambda_n = \frac{1}{2}\left\{\left(\frac{n+t_o^2}{n+t_1^2}\right)^{\frac{1}{2(n+1)}} + \left(\frac{n+t_o^2}{n+t_2^2}\right)^{\frac{1}{2(n+1)}}\right\}$$

$$t_o = \frac{\bar{x}-\bar{\bar{x}}}{S_{\bar{x}}}, \quad t_1 = \frac{\bar{x}-\bar{\bar{x}}-\delta}{S_{\bar{x}}}, \quad t_2 = \frac{\bar{x}-\bar{\bar{x}}+\delta}{S_{\bar{x}}}$$ and δ is the longest correlated interval.

This corresponds to Fishman's correlation time.

The procedure being Accept H_o if $\lambda_n < B$; H_o: $\mu = E[\bar{x}-\bar{\bar{x}}] = 0$

Reject H_o if $\lambda_n > A$; H_1: $\mu \neq 0$

The important aspects of the application of the sequential t-test to determining the termination of the transient stage are:

1) the test is dynamic

2) the test deals with the sample mean

One of the most difficult problems in estimating the length of the transient stage has been in selecting or constructing a technique which allows the experimenter to determine the end of the transient stage while the simulation is still in the computer. To meet this requirement, any selected technique must be programmable and may not interupt the continuous flow of the simulation. The sequential t-test seems to this writer to be the first method suggested to deal with the transient stage problem which fully meets this dynamic requirement.

The second aspect is of interest to this writer. Reese assumes that once a sample mean does not differ from the grand mean of a statistic, the statistic has reached stability. This implies that stability is a function of the first moment. It is suggested that by the above criterion, $\hat{\mu}_i$ and $\hat{\mu}_{i+1}$ may be statistically equal, while $\hat{\sigma}_i$ and $\hat{\sigma}_{i+1}$ or $\hat{R}_i$ and $\hat{R}_{i+1}$ are very unequal. It is reasonable that stability should be sought in both the mean and the variance.

APPROACH TO THE PROBLEM

The proposal to be evaluated is that of comparing successive segments of a simulation-generated time series in order to determine whether or not equilibrium has been attained. It is contended that when there is no longer a significant difference between the spectra of adjacent segments, it can be said that equilibrium has been achieved.

It is our contention that initially a simulation-generated time series can be described as some trend function plus a series of cyclic components plus an error term,

$$X_t = g[X(t)] + \sum_{j=1}^{k} a_j[\sin(\omega_j t) + i\cos(\omega_j t)] + \varepsilon_t$$

An analog to this is the color spectrum. When we see a color (i.e., blue) we know that actually the entire color spectrum is present with the shorter blue light frequencies dominating.

The same is true of simulation statistics. The parameter estimates are actually some trend function and a number of cyclic elements or frequencies. As the simulation progresses, some of these frequencies, $\sum_{j=1}^{k} a_j[\sin(\omega_j t) + i\cos(\omega_j t)]$, degenerate. These terms do not completely degenerate, rather some portions of it become equivalent to zero, while others remain. Once the parameter has achieved stability, it can be expressed such that

$$X_t = g[X(t)] + \sum_{j=1}^{k} [a_j(\sin(jt) + i\cos(jt))] + \varepsilon_t$$

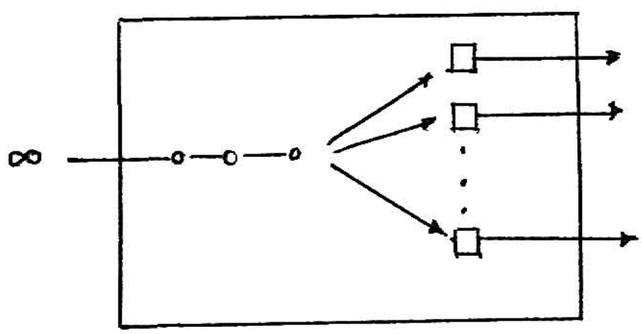
where a_j is the strength of the frequency j. It is not known whether or not it is possible to forecast how fast or which frequencies will degenerate before the simulation is run.

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.

For testing the above hypothesis, it is suggested that a simulation be studied to see if the spectrum of the time series does change with time. Such a procedure cannot prove the contention, but it will give evidence as to whether or not the hypothesis is worth entertaining. Since our primary interest is with a system that when simulated, will have a definite transient stage and a measurable state of equilibrium, a queueing system with multiple serving stations was selected. The system will have a definite transient stage and the expected values [1] of the system can be found.

The system simulated for this study was a simple queueing system with multiple serving stations. The time between arrivals was generated from an exponential distribution. The arrivals entered the system singly and were served on a first-in, first-out basis. The queue was not restricted in size. The service stations were parallel, each serving a single unit at a time. The underlying service time was exponentially distributed.



Turning to queueing analysis as presented in Chapter 6 of Sasieni, Yaspan, and Freidman's Operations Research: Methods and Problems [21], we find the average number of units in the system,

$$L_s = \frac{\lambda}{\mu} + \frac{(\lambda/\mu)^r \lambda\mu}{(r-1)! \ (r\mu-\lambda)^2} P_o$$

the average number of units in the queue,

$$L_q = L_s - \frac{\lambda}{\mu}$$

and the average utilization of service stations which is

$$E(u) = \frac{E[\text{stations occupied}]}{\text{Total stations}}$$

$$E(u) = \frac{1}{r} P_1 + \frac{2}{r} P_2 + \frac{3}{r} P_3 + \dots + \frac{r-1}{r} P_{r-1} + P_r \left[1 - \sum_{n=1}^{r-1} P_n\right]$$

where

λ = average arrivals per hour

μ = average service per hour

r = number of serving stations

$$P_o = \frac{(\lambda/\mu)^n}{n!} + \frac{(\lambda/\mu)^r}{r!} (1-\lambda/r\mu)^{-1} \quad , \quad P_n = \frac{(\lambda/\mu)^n}{n!} P_o, \quad 0 < n \leq r$$

By observing the number of units in the system every hour, one can construct a simulation-generated time series. The variance of the time series can be analyzed by using spectral analysis. Spectral analysis decomposes the variance into a series of cyclic elements. The set of these cyclic components is termed the spectrum of the time series. In order to compare the spectrum at time t with the spectrum at time t + τ, we use cross-spectral analysis. Cross-spectral analysis will give the investigator a measure of the correlation between corresponding cyclic elements. This measure is termed coherence.

It is possible to determine whether or not the spectrum of a simulation-generated time series changes with time by utilizing cross-spectral analysis. If successive segments of the time series are significantly different, then the simulation is considered to be in the transient stage. If the successive segments do not change, the simulation is considered to be in the steady state, or equilibrium. We must also be concerned with the lag between the actual end of the transient and the first indication that the simulation has passed out of the transient stage.

SPECTRAL ANALYSIS

Much of the output of simulations can be considered simulation-generated time series. Let us first discuss time series in general and a method of analyzing such data. A time series is a continuous or discrete series of values which are related to time, though the relationship need not be a causal one. Examples of time series are the daily temperature reading in Manhattan, Kansas from 1801 to the present, the Dow-Jones stock market averages for the past seven years, the weekly national sales of Zappo Peanut Butter, or the number of items in a simulated system observed every hour for 2000 simulated hours. The daily temperature readings are a good example of a continuous time series, for at no time is there no temperature. The stock market averages are an example of discrete time series, for the series is not continuous with time. Since stock cannot be bought or sold in the market on Sunday afternoon, there is some time interval when stock market averages do not exist.

Time series are spoken of as having a trend. The stock market seems to have had a downward trend lately. This trend is the line about which the values of the series seem to fall. Time series also are often cyclic; for example, daily temperature readings vary with the season. Zappo Peanut Butter may have yearly cycles as well as bi-monthly or quarterly cycles. If we were to look at only six months of Zappo Peanut Butter sales, the yearly cycle may appear to be a trend in the data. Extending this idea, a time series can be considered a series of cyclic elements, some longer than the sample length, some cycles shorter than the sample length. When considering a time series with constant trend, x, the variance around x can be thought of as the result of combining the cyclic elements with lengths, or periods, shorter than our sample record. The amount of variance contributed by each cyclic element will be related to its period and amplitude. One way to estimate the length of these cyclic elements is to observe the autocorrelation function of the time series. If the autocorrelation has equidistant peaks or valleys, the existence of cyclic elements should be strongly suspected.

Spectral analysis is a statistical technique designed to estimate the cyclic elements, i.e., spectral components, of time series' variance. This technique is uniquely applicable to simulation-generated time series for it utilizes the time series' autocorrelation function in estimating the spectral components. (We have previously mentioned that simulation-generated time series are autocorrelated.) The application of spectral analysis for examination of simulation output was first suggested by Fishman [7], but not in conjunction with the problem being considered in this thesis. We shall give a brief development of spectral analysis.

Consider a series of finite length

$$\{X_t: t = 1, 2, \ldots, n\}$$

As has been mentioned, the time series will have a trend of some form. This trend may change with time or be constant. Let us consider $\alpha(t)$ a constant with time as the trend of X_t. We can express X_t

$$X_t = \alpha(t) + \varepsilon(t)$$

where $\varepsilon(t)$ is the amount of error at time t.

The time series can also be thought of as a series of cyclic elements, oscillating about, $\alpha = a$.

$$X_t = \sum_{j=1}^{K} a_j(\cos \omega_j t + i \sin \omega_j t)$$

Where ω_j, j=1, 2, ..., n is a set of real numbers $|\omega_j| \leq k$ and a_j, $j = 1, \ldots, k$ is a set of independent complex variables, with $E[a_j] = 0$, all j, $E[a_j\bar{a}_j] = \sigma_j^2$, $E[a_j\bar{a}_k] = 0$, $j \neq k$. This can be thought of as $X_t = \sum_{j=1}^{k} a_j e^{i\omega_j t}$; however, by using the trigonometric equivalent, the cyclic concept is more apparent.

The cyclic terms of this generating process will have a period of $\frac{2\pi}{\omega_j}$. The cyclic elements in the time series arise from the $(\cos\omega_j t + i \sin\omega_j t)$ terms. If we allow $E[X_t] = 0$ then,

$$\mu_k = E[X_t X_{t-k}]$$

$$= \sum_{j=1}^{k} \sigma_j^2 (\cos \omega_j t + i \sin \omega_j t)$$

The autocorrelation is a function which increases by the amount of variance in the interval (0,t), and as ω_j moves from 1 to k. In a continuous form this is expressed,

$$\mu_t = \int_{-\pi}^{\pi} (\cos \omega_j t + i \sin \omega_j t) dF(\omega)$$

a linear cyclic function.

If we can isolate the σ_j^2 we can know at which ω_j the most of the variance is contributed. This is termed spectral estimation at ω_j. Once the important ω_j are located, the periods can be identified.

Since we are concerned with real processes of constant mean, $dF(\omega)=dF(-\omega)$ is true. The above expression of μ_t can be written , $\mu_t = \cos(t\omega) dF(\omega)$, giving the following representation of the spectrum,

$$F(\omega) = \frac{1}{2} [\mu_o + \sum_{j=1}^{\infty} \mu_j \cos(\omega_j)]$$

To put this in a computational form

$$f(\omega) = \frac{1}{2} [C_o + 2 \sum_{j=1}^{m-1} C_j \cos(\omega_j)]$$

where $C_k = \frac{1}{N} \sum_{t=1}^{N-k} X(t)X(t+k)$, k=0, ..., N-1 is the estimate of μ_j.

In order to get a better estimate of the frequency, a weighting factor is necessary. The estimate becomes

$$\hat{f}(\omega) = \frac{1}{2\pi} \left\{C_o\lambda_o(\omega) + 2 \sum_{j=1}^{m-1} \lambda_j(\omega)C_j \cos(\omega j)\right\}$$

The weighting factor has had much discussion and many have been suggested. For the scope of this work, only two will be mentioned.

The Tukey-Hanning Window

$$\lambda_k = \frac{1}{2} \left(1 + \cos \frac{\pi k}{m}\right)$$ where m is an integer usually chosen m n/3

The Parzen Window

$$\lambda_k = \begin{cases} 1 - \frac{6k^2}{m^2} (1 - \frac{k}{m}), & 0 < k \leq m/2 \\ 2[1 - \frac{k}{m}]^3, & m/2 \leq k < m \end{cases}$$

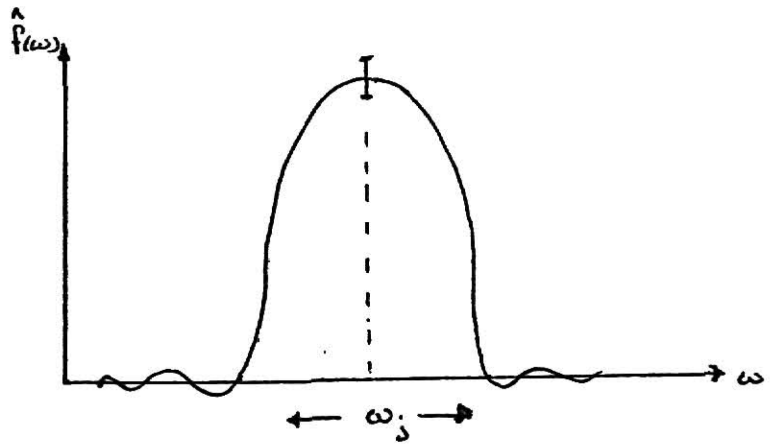
For further discussion of weighting factors, see Spectral Analysis of Economic Time Series by C.W.J. Granger [9], Power Spectrum from the View of Communications Engineers by Blackman and Tukey [11], or Spectral Analysis and Its Applications by Jenkins and Watts [14].

The final form of the spectral estimate is

$$\hat{f}(\omega) = \frac{1}{2\pi} \sum_{j=1}^{m} \lambda(\omega) C_j \cos(\omega j)$$

Experience (Granger [9] pages 59-62) has shown that in using this estimate, a sample of at least 200 points is desirable; for with fewer sample points, it is possible that cyclic elements would go unnoticed.

The estimate of the spectrum is presented graphically.



There are two areas in which error can arise in the estimate. First, there will be some error in the height of the estimate. This error is related to the window used. Second, there is error associated with cell width. Both errors involve the number of lags, m. A problem arises at this point. If one set m at n/3, the bandwidth is wide, thus giving an accurate estimate of the power at the frequency, but a less accurate estimate of the frequency. If one sets m low, n/6, the reverse is true. We have a good estimate of the frequency, but a poor estimate of the power at that frequency. Parzen [17] suggests doing the analysis three times with m = n/3, n/6, n/12.

The estimates have been programmed in FORTRAN and computed for simulation-generated time series on an IBM 360/50. A listing of this program is shown in Appendix A. A sample spectrum from a simulation-generated time series is also shown.

CROSS-SPECTRAL ANALYSIS

In order to study the transient stage of a simulation, it was decided to examine the spectrum of a sample time series taken from the transient stage of a simulation. The spectrum gave an indication of the cyclic structure of a specific interval of the simulation-generated time series. By comparing the spectra of two successive samples, one can determine whether or not the spectral components have changed. The hypothesis of this study is that a change in successive sample spectra will indicate degeneration of the transient properties of the simulation.

From the theory of stationary processes we know that the power at a frequency, ω_j , is independent of the power at all other frequencies [9]. It can also be shown that the power at ω_j is independent of all other frequencies of another variable except the component at ω_j .

If we are given two time series $X(t)$ and $Y(t)$ which are not independent and we obtain the spectra of the two series, we can determine the squared correlation of $X_{\omega_j}(t)$ and $Y_{\omega_j}(t)$. $X_{\omega_j}(t)$ is defined as the spectral power contributed to the variance of $X(t)$ at ω_j . Since the power at each frequency is independent of the power at all other frequencies, the squared correlation between all $X_{\omega_j}(t)$ and $Y_{\omega_j}(t)$ pairs need not be constant. A smoothed plot of the squared correlation against the frequency over all frequencies is termed the coherence diagram. By observing the coherence diagram of $[X_{\omega_j}(t),Y_{\omega_j}(t)]$ the similarity of the two spectra can be measured. A coherence diagram of two homogeneous time series will be in the neighborhood of one for all frequencies. Similarly, the coherence diagram of two completely unlike series will approach zero for all frequencies.

Before disucssing how cross-spectral analysis will be used to study the transient stage problem, we shall first consider a method of estimating the coherence at each frequency.

In order to estimate the cross-spectrum of two time series, X(t), Y(t) we first compute their autocorrelation functions, $C_{XX}(k)$, $C_{YY}(k)$ as described above. We then compute their cross-correlation functions,

$$C_{XY}(k) = \frac{1}{N}\sum_{t=0}^{N-k} X(t+k)Y(t) \qquad \text{and} \qquad C_{YX}(k) = \frac{1}{N}\sum_{t=0}^{N-k} X(t+k)Y(t), \text{ where}$$

$k = 0, 1, \ldots, m$. Cross-correlation can be interpreted to mean the correlation of X at time t with Y at time t + k, as k moves from 1 to N-1.

Recall that the power spectrum is

$$f_X(\omega) = \frac{1}{2\pi}\left[\mu_X(0) + 2\sum_{\tau=1}^{\infty} \mu_X(\tau) \cos \tau\mu\right]$$

which is estimated by

$$\hat{f}_X(\omega) = \frac{1}{2\pi}\left[C_{XX}(0)\lambda_o(\omega) + 2\sum_{j=1}^{m-1} \lambda_j(\omega)C_{XX}(j) \cos(\omega j)\right]$$

We can obtain $\hat{f}_X(\omega)$ and $\hat{f}_Y(\omega)$. The cross-spectrum is a linear combination of the in-phase cyclic elements of X(t), Y(t), and the out-of-phase cyclic elements of the two series. The spectrum of the in-phase elements is termed the co-spectrum, $c(\omega)$. The out-of-phase spectrum is termed the quadrature spectrum, $q(\omega)$, and is an imaginary number. These spectra are estimated in a similar fashion as the power spectrum.

The co-spectrum is estimated

$$\hat{c}(\omega_j) = \frac{\lambda_o}{4\pi}\left[C_{XY}(0) + C_{YX}(0)\right] + \frac{1}{2\pi}\sum_{k=1}^{m} \lambda_k\left[C_{XY}(k) + C_{YX}(k)\right] \cos\omega_j k$$

The quadrature spectrum is estimated

$$\hat{q}(\omega_j) = \frac{1}{2\pi} \sum_{k=1}^{m} \lambda_k \, [C_{XY}(k) - C_{YX}(k)] \, \sin\omega_j k$$

Since the co- and quadrature are spectra, they are the summation of the cross-product of the cyclic elements and the associated cross-correlations. This is analogous to the power spectrum being the cross-product of auto-correlation and cyclic elements.

As mentioned above, the cross-spectrum is the linear combination of the co- and quadrature spectra. This is expressed $Cr(\omega_j) = c(\omega_j) + i\ q(\omega_j)$ and is a real number.

Granger states, without proof, that

$$c^2(\omega_j) + q^2(\omega_j) \leq f_X(\omega_j)\ f_Y(\omega_j).$$

This relationship is the coherence-inequality. From the coherence-inequality, we obtain the coherence

$$C(\omega_j) = [c^2(\omega_j) + q^2(\omega_j)]/[f_X(\omega_j)\ f_Y(\omega_j)].$$

Clearly, $C(\omega_j)$ will range from zero to one, and is always positive. Jenkins points out the analogy between $C(\omega_j)$ and the correlation coefficient. He also mentions that the points (of the coherence diagram) will be scattered about a straight line with a scatter which is large if the coherency is low and small if the coherence is high [12].

The co-spectral and quadrature estimates were found for simulation-generated time series. The program used is found in Appendix B as well as a sample coherence diagram. A short discussion of the program is found in Appendix B.

Let us now turn to how cross-spectral analysis will aid in the study of the transient stage. The hypothesis is that when a state of equilibrium has been achieved, the simulation statistics will not change in mean or variance. Since the approach to a time series is that X(t) is the summation of all cyclic elements with a constant mean, any change in the mean will result in a disproportionate contribution to the spectrum at low frequencies. A changing mean will be observed in the successive power spectra of two successive samples from the simulation statistic. By using cross-spectral analysis, one can determine if the variance of the successive samples are statistically similar. It is believed that in both mean and variance, successive samples from a simulation statistic will be different during the transient stage. During the steady state, both the mean and the variance of a simulation statistic will be similar for successive samples. This will be reflected in the coherence diagram. If one can show that the coherence changes as the simulation passes into equilibrium, then a basis exists for the hypothesis that by comparing spectra from successive segments of a simulation-generated time series, it can be determined whether or not equilibrium has been attained.

THE SIMULATION

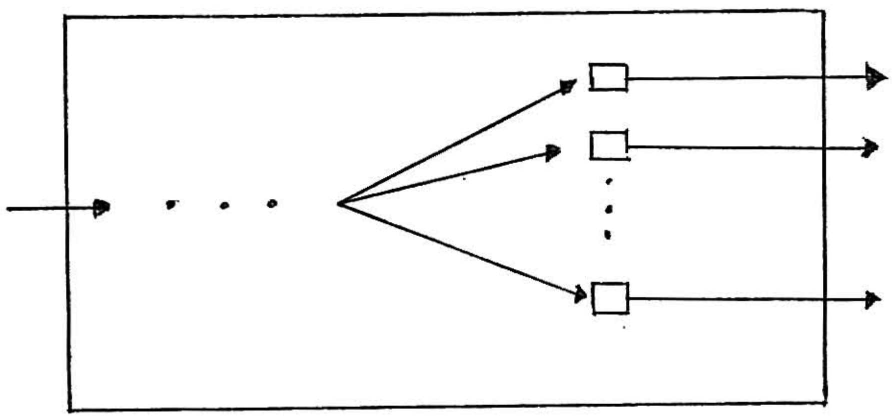
We have given a statement of the problem, a survey of the literature, and a discussion of the techniques to be used to study the transient stage. Before the presentation of the results of the study, a short discussion of the simulation will be presented to aid in understanding the results and conclusions.

The simulation was done in GASP II. GASP stands for General Activity Simulation Program. It is a collection of subroutines which accomplish most of the tasks common to simulation experiments. GASP is written in FORTRAN, which allows it to be used on almost all computer systems. Since the language is FORTRAN, the output can easily be input into special analysis programs such as the spectral analysis programs used in this study. Other simulation systems, such as GPSS and SIMSCRIPT, do not have this feature at this time.

The GASP subroutines are designed for "next event" simulation. A "next event" simulation is one in which time does not move forward in equal increments, such as hours, but in unequal increments from one event to the next. In order to accomplish this, an event calendar is constructed. This calendar keeps track of the coming sequence of events and amount of simulation time between events. The investigator is required to write event subroutines which compute what type of event next occurs at what time.

GASP is a collection of subroutines. The activities described above are accomplished in the subroutines GASP, SET, FILEM, REMOVE, and FIND. The GASP routine provides two subroutines for gathering statistics, COLECT and TMSTAT. The subroutine HISTOG is included to collect histograms of at most five distributions in which the investigator may be interested. The subroutines PRINTQ, SUMARY, and OUTPUT, are provided to report statistical computations and any special information the investigator may desire.

In explaining how GASP II operates, let us consider the system being studied in this paper. A diagram of the system is seen below.



To initiate the system, we must have an initial arrival into the queue. This is done with input read by subroutine DATAIN. Upon the arrival of an item, the simulation must check to find if a service station is free to process the new item. If no station is free, the item is placed in the queue by FILEM. Notice that all arrivals wait in a single queue; if it were desired, the GASP routine could provide separate queues for each station and either FIFO or LIFO queue management. The simulation must generate the next arrival and the inter-arrival time. The inter-arrival time is obtained from an exponential deviate generator. GASP provides several random deviate generators as well as a random number generator. The exponential generator used in this simulation was from Computer Simulation Techniques by Naylor, Baintfy, Burdick, and Chu [15]. The technique is straightforward. Since the exponential is $f(x) = \alpha e^{-\alpha x}$, the cumulative distribution is $F(x)=1-e^{-\alpha x}$

It is known that random numbers are uniformly distributed; if $r = e^{-\alpha x}$, then the exponential deviate x is equal to $\alpha \log r$. From this scheme, the inter-arrival time for the next arrival is obtained. The next arrival can then be stored in the GASP calendar.

If the current arrival is placed in the queue, the GASP routine will consult the calendar to determine what event should occur next and when it should occur. Let us assume the item was immediately assigned to a station. The number of free stations would be decremented by one and a completion of service must be determined. The completion would be stored in the GASP calendar with the time of its occurrence.

When a completion of service occurs, the simulation determines whether or not an item is waiting in the queue. If one is waiting, it is assigned to the free station and the completion time is computed. The simulation then checks the calendar for the next event. If there is no item in the queue, the simulation immediately checks the calendar for the next event.

As the simulation is progressing, the experimenter will be interested in gathering statistics concerning various aspects of the system. The statistics gathered in this simulation were the total items in the system, the items in the queue, and the utilization of the system. The distributions of the inter-arrival time and the service time were also gathered. Both were anticipated to be exponentially distributed. Statistics related to time, such as the number of items in the system and the number of items in the queue, were collected in TMSTAT. Statistics not related to time, such as inter-arrival time and service times, both related to the exponential random number generator, were collected in COLECT. The subroutine SUMARY will print the mean, standard deviation, minimum and maximum of all statistics gathered. The subroutine OUTPUT will print or punch any special output the investigator may desire. OUTPUT is used to punch the three series, utilization, queue size, and system size, as input for special analysis programs. A REPORT subroutine was constructed by the investigator in order to gather statistics every DELTA hours and print these statistics after every NOBS observation. The general logic diagram, Fig. 1.1, of this simulation can be found in Appendix C. The program and a sample of the output are also located there.

There was a problem encountered in the simulation experiments. As mentioned above, a multiple service station queueing system was studied in order to easily determine the steady state stage of the simulation. This was to be done by using theoretical values of the average utilization, the average size of the queue, and the average size of the system. The first attempts to simulate the system resulted in values far divergent from the expected values. A table of these values and the expected values can be found below. The REPORT subroutine was revised after the first experiment to give samples of the inter-arrival distribution and the service time distribution. A histogram of the inter-arrival times (Fig. 1.2 in Appendix C) indicates that the distribution is not exponential as was expected.

The random number generator was examined to determine if it was the cause of the unexpected distribution. The random numbers used in the exponential generator were again generated in the same order as in the simulation. This was possible by the fact that random numbers generated by digital methods are not truly random, but pseudo-random. Although a Chi-Square test would show the numbers to be random, they can be generated again in the same order. The reason for this is that the numbers are not based on the conventional base ten number system. They are based on a modulo system of base 2^{36-2}. A modulo number system is described by the relationship

$$A_{i+1} = A_i b (\text{mod } M),$$

where A and b are integers and M is the base of the modulo system. In digital simulation, b is referred to as the seed and A as the random number. The following is an example of a modulo number system, base 11 and using the seed 7.

$$A_{i+1} = A_i * 7 \ (\text{mod } 11)$$

A_{i+1}	A_i*7	$A_i*7(\text{mod } 11)$
7	1*7	7
5	7*7	49 = 4(11) + 5
2	5*7	35 = 3(11) + 2
3	2*7	14 = 11 + 3
.	.	.
.	.	.
.	.	.

Continuing the example would show the system to be cyclic. The random numbers used in the simulation were examined to determine if they were cyclic and thus not random. First, a Chi-Square test was done. The results revealed within 95% confidence that it was not possible to show that the values were not random. The spectrum of the random number series was obtained by maintaining the order of the values occurrences and considering the series a simulation-generated time series. The result was that no significant frequencies were found.

While these tests were being done, a classroom project done by Ruben Kacinhoff showed that the exponential distribution could be approximated by allowing a simulation to run for over two million simulated hours. In light of this information, the transient stage of this simulation was compared to the transient stage of other simulations. Examination of the transient stage of the simulation indicated that the behavior of the simulations measures of performance are typical of simulations performed by investigators in the past (see Job Shop Simulation by Santosh Dipchand Vaswani [22]). It is also found that the length of the transient stage could be found by the methods suggested by Conway. Since the end of the transient stage could be found, there was no reason to discard the simulation in favor of redesigning the random number generator or running it again for over a million simulated hours.

In the literature survey, it was mentioned that simulation-generated output is highly autocorrelated. This was found to be true. Shown in Appendix C are ten graphs showing the autocorrelation functions of twenty segments of the simulation-generated time series associated with the number of items in the system. Each segment of time represents 100 simulated hours with an observation taken every half hour or 200 observations. It can be seen that the autocorrelation is quite strong throughout the simulation (1100 hours). This means that it cannot be assumed that the observations of the time series are independent. As the reader will recall, this was Fishman and Kiviat's criticism of most simulation analysis; it does not account for autocorrelated output. It was for this reason that they suggested spectral analysis as an approach to analysis of simulation output.

It can be observed from Fig. 1.3 in Appendix C that the autocorrelation function changes as the simulation progresses. It is asserted that the autocorrelation found is the sum of the actual autocorrelation of the system and the autocorrelation induced by the simulation. If it were known how to separate these two autocorrelation functions, a better analysis of the simulation could be accomplished. The changing autocorrelation reflects the changing variance of the time series. The changing variance is indicative of the transient stage of a simulation. Fishman and Kiviat suggest the use of spectral analysis in comparing variance if the data is known to be autocorrelated. Since a definition of equilibrium in simulation statistics is that the variance and mean vary within acceptable ranges, it is reasonable that comparing the variance of successive segments of simulation-generated time series would indicate whether or not the simulation is in equilibrium. Spectral analysis would be an appropriate approach since it is known that the data is autocorrelated.

To illustrate the long range effects of the transient stage, a simulation of a simple FIFO, single-channel queue was allowed to run for 135,000 simulated hours. The average number of arrivals per hour was 100, while the average number of departures per hour was 95. Both arrivals and departures were generated from an exponential distribution. An observation of the number of units in the system was made every 100 hours. The cumulative average of this statistic was given every 1000 hours.

It can be seen in Fig. 1.4 in Appendix C that the statistic is clearly in the transient stage for 25,000 hours. The statistic does not seem to become more or less constant, as would be expected of the cumulative average number of units in the system, until beyond 90,000 hours. The statistic seemed to pass out of the transient stage somewhere between 25,000 hours and 70,000 hours; yet the effect of having been in the transient stage can be seen for another 20,000 hours.

Let us now turn to a method of determining the end of the transient stage. We will present Conway's method. We will then compare the methods suggested by Bueno and Reese. Upon discussing those results, a discussion of the application of cross-spectral analysis will be presented.

COMPARISON OF THE METHODS SUGGESTED BY BUENO AND REESE

The first experiment was a simulation of a multiple station queueing system. The simulation was begun with no units in the system and allowed to operate for 1100 hours. The end of the transient stage was determined by the two methods mentioned above.

As stated previously, the criterion of a good solution is that it be dynamic and consider both mean and variance. It was found that the solution needs another constraint. The method used to evaluate the end of the transient stage must be practical. If the method is so sensitive to changes in the test statistic that after a reasonable length of time it has not reached a decision, the method is impractical. One reason for determining the length of the transient stage is to conserve computer time. A method which is wasteful of computer time due to oversensitivity is not of practical value.

We shall now discuss the determination of the length of the transient stage by Conway's method [4]. The system was simulated and the simulation statistics were plotted. Conway suggests that various parameter estimates should be examined in search of limiting. Vaswani [22] followed Conway's advice and plotted average inter-arrival times along with average inter-departure times. As these values converge, the simulation is thought to be in equilibrium. This is intuitively reasonable, for if the inter-arrival of units is equal to the inter-departure of units, the number of units in the system is neither increasing nor decreasing. If the size of the system is stable, it is reasonable to assume that the stochastic processes within the system are not varying except within acceptable and predictable ranges.

The average inter-arrival and average inter-departure times seem to converge after approximately 700 hours as shown in Fig. 2.1, Appendix D. In order to gain more insight into the length of the transient stage, a few more parameter estimates were plotted (Fig. 2.2 through Fig. 2.6). These indicated that the size of the system seemed to stabilize between 625 and 750 hours. One should be warned against plotting the number of units in the system. This will include a large amount of variance and is difficult to interpret meaningfully (see Fig. 2.3). The moving average of the number of units in the system is more meaningful (Fig. 2.2). The average service time was plotted. This plot brings up an interesting point.

Bueno mentioned that the individual stochastic processes of a simulation seem to stabilize before the entire system reaches equilibrium. The average service time begins quite large and quickly levels off in the neighborhood of 55 hours. It indicates equilibrium after only 100 hours. This lends credit to Bueno's belief.

The average change in the size of the system was plotted for 700 hours, after which it approached zero. This statistic (Fig. 2.5) indicated that a steady state condition exists after approximately 600 to 700 hours. The average time in the queue was plotted. This was less helpful than the others. The statistic (Fig. 2.6) seemed to reach a plateau between 625 and 750 hours; however, a sharp increase in the last three observations made it difficult to interpret. In the light of the previous material, this was likely to have been a random fluctuation.

It can be seen that the length of the transient stage for this simulation is between 600 and 700 hours. This seems to be the best method for estimating the length of the transient stage. Although it does not meet the criterion of being dynamic, it does provide a good estimate against which other methods were compared. Since the solution to the problem of estimating the length of the transient stage has no definite answer, the above was considered the best estimate.

Let us now consider Bueno's suggested method [2]. Bueno suggested setting an upper and a lower limit of successive sample means of a particular simulation statistic. For this experiment, the total items in the system were chosen as the simulation statistic. Bueno's criterion is that when two successive sample means fall within the limits, we can consider the simulation in equilibrium. The length of the transient stage is from time zero through the period preceeding the two sample means falling within the limits. This procedure is the application of control charts to the transient stage problem.

In order to employ the technique we must first calculate the grand mean, the variance of the sample means, and the upper and lower bounds. Let N be the number of samples taken. Bueno suggests discarding the first sample as it is known to be biased.

1. The Grand Mean

$$\bar{\bar{X}} = \frac{1}{N-1} \sum_{i=2}^{N} \bar{X}_i$$

2. The Variance of Sample Means

$$S^2 = \frac{1}{N-2} \sum_{i=3}^{N} (\bar{X}_i - \bar{\bar{X}})^2$$

3. The Upper and Lower Bounds

$$\text{Limit} = \bar{\bar{X}} \pm 2S/\sqrt{N-1}$$

Once this is done, the sample means are plotted against time. The upper and lower limits are drawn. Starting from the first sample period, we locate the first two sample periods, m and m + 1, such that both lie within the area bounded by the control limits. Periods one through m - 1 are considered the transient stage.

This has been done for the simulation in question (see Fig. 3.1). We observe that the length of the transient stage, according to Bueno, is 300 hours. This is not in accord with the conclusions drawn from the method proposed by Conway. The reason is that Bueno has set the limits such that .95 of the variance is included within the limits. This causes the lower limit to be closer to zero, thus allowing the statistic to enter the acceptance interval sooner. If the limits would include only .90 of the variance, the transient stage would be considered from zero to 600 hours.

It is observed that the sample means fall outside the limits at the tenth sample period, or the 1000th hour sample. While this event is not inconsistent with the concept of control charts, it does leave the investigator in doubt as to the length of the transient stage. Bueno makes no provision for the event.

In order to employ Bueno's method, we require the use of the grand mean. It in turn requires that the simulation be completed before the output can be analyzed. The method is not dynamic.

USE OF THE BUENO METHOD TO
FIND THE END OF THE TRANSIENT STAGE

1. Calculate the grand mean.

$$\bar{\bar{X}} = \frac{\sum_{i=2}^{N} \bar{X}_i}{N-1} = 51.26$$

where $\bar{x}_i$ = sample mean for period i

N = number of periods considered

2. Calculate the variance of sample means.

$$S^2 = \frac{\sum_{i=3}^{N} (\bar{X}_i - \bar{\bar{X}})^2}{N-2} = 84.89, \; S = 9.21$$

3. Calculate the upper and lower control limits.

$$UCL = \bar{\bar{X}} + 2S/\sqrt{N-1} = 57.09$$

$$LCL = \bar{\bar{X}} - 2S/\sqrt{N-1} = 45.43$$

4. Plot sample means against time periods.

5. Draw in UCL and LCL.

6. Starting from $\bar{X}_o$ find the first $\bar{X}_m$ such that both $\bar{X}_m$ and $\bar{X}_{m+1}$ lie within the area bounded by UCL and LCL.

7. Consider periods 1 through m - 1 as the transient period.

We shall now consider Reese's proposals. As mentioned above, this technique is based on the sequential t-test. As long as the test statistic lies between B and A, no decision can be made. This can be taken to mean that the simulation is still in the transient stage. Again, the number of units in the system is the simulation statistic under consideration.

In order to use the sequential t-test, we must set our sample size, n, and the lag associated with the process, m. Reese states, "It is suggested that a large value of m should be used." and "The values of m should be smaller than n." The value of n was set at 100, so the value of m was set at 60. With each sample, we compute the following:

1. The grand mean, up to and including the current sample

$$\bar{\bar{X}} = \frac{1}{N} \sum_{i=1}^{N} X_i$$ where N is the total observations up to the present

2. The sample mean

$$\bar{X} = \frac{1}{n} \sum_{i=N+m+1}^{N+m+n} X_i$$

3. The sample variance

$$S_X^2 = \frac{1}{n-1} \sum_{j=N+m+1}^{N+m+n} (X_j - \bar{X})^2$$

4. The sample autocorrelation

$$R_\tau = \frac{1}{n} \sum_{j=1}^{n} (X_i - \bar{X})(X_{i+\tau} - \bar{X})$$

5. The variance, adjusted for autocorrelation

$$S_{\bar{X}}^2 = \frac{1}{n} [S_X^2 + 2 \sum_{\tau=1}^{m} R_\tau]$$

It is at this point that Reese considers the autocorrelation of the time series. He computes the autocorrelation for each sample in recognition of the changing nature of autocorrelation.

With the above information, Reese has developed the following test. In testing:

H_o: $\mu = E[\bar{X} - \bar{\bar{X}}] = 0$, the steady state hypothesis

H_1: $\mu \neq 0$

we will accept H_o if λ_n is less than B, and we will reject H_o if λ_n is greater than A. If $B < \lambda_n < A$, then we can make no decision, and go on to the next sample. In testing for the end of the transient stage, a no-decision situation implies that the simulation is still in the transient stage.

From Wald [23] we find that

$$\lambda_n = \frac{1}{2} \left\{ \frac{g_n(t;\mu_o+\delta_1) + g_n(t;\mu_o-\delta_2)}{g_n(t;\mu_o)} \right\}, \quad \text{in this case } \delta_1 = \delta_2 = \delta$$

where $g_n(\cdot)$ is the density function of a t distribution with n degrees of freedom. Reese has derived

$$\lambda_n = \frac{1}{2}\{[(n+t^2)/(n+t_1^2)]^{1/2(n+1)} + [(n+t^2)/(n+t_2^2)]^{1/2(n+1)}\}$$

where $t = \dfrac{\bar{X}-\bar{\bar{X}}}{S_{\bar{X}}}$, $t_1 = \dfrac{\bar{X}-\bar{\bar{X}}-\delta}{S_{\bar{X}}}$, $t_2 = \dfrac{\bar{X}-\bar{\bar{X}}+\delta}{S_{\bar{X}}}$, and $A = \dfrac{1-\beta}{\alpha}$, $B = \dfrac{\beta}{1-\alpha}$

This test was programmed for an IBM 1620. (A listing can be found in Appendix D). It was found that after 900 hours, we could make no decision. According to the test, the transient stage has not terminated. Reese points out that the method is quite conservative and forces the accumulation of a large number of samples. The reason given is that the grand mean is biased, and the bias decreases only after the transient stage has passed. As can be seen, while Reese's method is dynamic and does consider the variance as well as the sample mean of the simulation statistic, it is overly conservative. The method eliminates too many of the early periods of the simulation to be practical. It is also noted that Reese adjusts the sample variance for autocorrelation. Removing the information provided by the autocorrelation is a loss of knowledge concerning the system being studied.

Reese does suggest that this method might be improved if an upper limit of N could be established or if the test was derived using Chebychev's Inequality. These suggestions are beyond the scope of this paper but are worthy of further consideration.

THE REESE METHOD

Given

Size of sample, n = 100

$$\bar{\bar{X}} = \frac{1}{N}\sum_{i=1}^{N} X_i; \quad \bar{X} = \frac{1}{n}\sum_{j=N+m+1}^{N+m+n} X_j$$

$$S_X^2 = \frac{1}{n-1}\sum_{i=N+m+1}^{N+m+n} (X_i-\bar{X})^2, \quad S_{\bar{X}}^2 = \frac{1}{n}\{S_X^2 + 2\sum_{\tau=1}^{m} R_\tau\}$$

$$R_\tau = \frac{1}{n}\sum_{i=1}^{n} (X_i-\bar{X})(X_{i+\tau}-\bar{X})$$

We also have the following information concerning the sequential t-test:

$$\lambda_n = \frac{1}{2}\{[(n+t^2)/(n+t_1^2)]^{1/2(n+1)} + [(n+t^2)/(n+t_2^2)]^{1/2(n+1)}\}$$

$$t = \frac{\bar{X}-\bar{\bar{X}}}{S_{\bar{X}}}, \quad t_1 = \frac{\bar{X}-\bar{\bar{X}}-\delta}{S_{\bar{X}}}, \quad t_2 = \frac{\bar{X}-\bar{\bar{X}}+\delta}{S_{\bar{X}}}, \quad A = \frac{1-\beta}{\alpha}, \quad B = \frac{\beta}{1-\alpha}$$

$$B < \lambda_n < A$$

This λ_n may be used to test the hypothesis

$$H_o;\ \mu = E[\bar{X}-\bar{\bar{X}}] = 0$$

$$H_1;\ \mu \neq 0$$

When λ_n less than B, accept the steady state hypothesis.

When λ_n greater than A, reject the steady state hypothesis.

When B less than λ_n less than A, no decision can be made.

The experimenter should move to the next sample.

EXPERIMENTS

The interest of the investigator during this study was to determine how the spectrum of a simulation-generated time series changes during the transient stage of the simulation. As mentioned above, part of the output of the simulation was three time series, the utilization of service stations, the size of the queue, and the size of the system. Graphical analysis indicated that the size of the system would be the most interesting to study.

Since the variance during the transient stage is known to be erratic and the spectrum can be thought of as a decomposition of the variance, we can expect that the spectra of two segments of a time series will be dissimilar if both segments are within the transient stage of the simulation. As mentioned above, the coherence diagram is a measure of the correlation of two spectra at each frequency of the two spectra. In short, if we select two segments of a time series, both within the transient stage and obtain their spectra, the changing variance of the transient stage will be reflected in the coherence diagrams.

A coherence diagram is composed of M values, one for each frequency to be considered by the investigator. Each value of the coherence diagram represents the coherence, or squared correlation, of the two associated spectral power estimates at that frequency. The mean of the coherence diagram is considered the average coherence of the two spectra. This value is a point estimate of the similarity of the two spectra.

We now have a measure of the similarity of two segments of a time series. By comparing this value for the first two segments of a time series, then the second pair of segments, and so forth, we can find out whether or not the variance is becoming more or less similar. We would expect the simulation-generated time series to remain quite dissimilar throughout the transient stage of the simulation. We would expect the variance to become quite similar during the steady state period of the simulation. It is expected that a plot of the average coherence of successive pairs of segments will be rather low initially (in the neighborhood of 0.00 to 0.20) then, upon entering the steady state period, to rise sharply. To summarize:

1. Select a time series for consideration.
2. Subdivide the series in segments. The segments may or may not be disjoint. All segments must be of equal length and should consist of at least 200 observations of the process.
3. Compute the spectra of each segment.
4. Compare successive spectra by computing the coherence diagram.
5. Determine the average coherence of each pair of segments.

Since a sharp increase in the average coherence is expected as the simulation passes into equilibrium, we use this increase to indicate the end of the transient stage. There will be a lag between the actual end of the trasient stage and the increase of average coherence. The lag is due to the need to have two segments in equilibrium before the average coherence begins to rise.

The next problem to be encountered is how far will the coherence rise before we can conclude the simulation is in equilibrium. Studies by Goodman [8] have shown that coherence values are distriubted in a bivariate normal distribution. Granger [9] gives the following distribution function:

$$F(u) = 1 - (1-u^2)^{n/m-1}$$

where n is sample size and m is frequency bands. In this experiment, n = 200 and m = 30 for all spectra. From the above example, if the true C(j) = 0, where j = 1, k, then the mean coherence would be less than .325. We now have an upper bound for average coherence values. When the average coherence is below the upper bound, we can conclude that the true coherence is not different from zero. When the sample average coherence is greater than the upper bound, we can conclude that the true coherence is rising and that the sample average coherence is the best estimate of the true coherence.

Another problem was encountered in this study. Upon determining the transient stage in the previous section, it was found that only seven segments of 200 observations could be studied before the simulation passed into equilibrium. It was also found that 200 simulation hours would pass before the average coherence values would indicate equilibrium, due to the time lag problem mentioned before. These two problems were resolved by overlapping segments. It was decided to overlap the segments by 50 simulation hours, or 100 observations of the process. This was done for the utilization statistic, the size of the queue series, and the size of the system series. The overlapping segments were zero to 100 hours, 50 to 150 hours and so forth. This not only increased the number of segments during the transient stage to fourteen, but also cut down the time lag to 100 simulation hours.

The first time series considered was the number of items in the system. Observations were made every half hour of simulation time. The first analysis of this series was illustrative of the general nature of the change in the average coherence values as the simulation progressed. Fig. 4.1 in Appendix E shows the change in the average coherence values as the simulation passes through the transient stage. Recall the length of the transient stage was found to be 700 hours for this simulation. It can be seen that during the transient stage, the average coherence values remain quite low, below .20. When both segments are beyond the end of the transient stage, the average coherence values move abruptly upward. An upper control limit of 95% confidence level has been drawn in to show there has been a definite change in the process. Unfortunately, not all of the trend was removed from these spectra. As a result of this, the average coherence values obtained during the steady state period do not exceed the .325 value mentioned earlier.

A line was fitted to the trend of each segment of the series under consideration using least squares regression. The removal of the trend from the data did not seem to affect the average coherence values from within the transient stage period as much as it seemed to affect the average coherence values associated with steady state period. As can be seen in Fig. 4.2, upon passing into equilibrium, the average coherence values are greater than the required value of .325, and they continue to move upward. We can conclude that the variance of the size of the system has stabilized after 700 hours.

The next time series was the size of the queue statistic. This series was obtained by gathering half hour observations of the queue size. The series brings up an interesting point. From Fig. 4.3, it can be seen that the queue size is not in equilibrium. Although the trend was removed in the same way as in the size of the system statistic, the average coherence values remain too low to conclude that the series has reached a steady state. This is a clear example of equilibrium not occurring simultaneously with all statistics of a simulation. An upper control limit of 95% confidence level has been drawn onto Fig. 4.3. It can be seen that after the size of the system passed into equilibrium, it was reflected in the queue size statistic. If more computer time had been available, the same pattern would have occurred as seen in Fig. 4.2.

A yet unresolved problem with the use of cross-spectral analysis in determining the end of the transient stage was encountered in the utilization statistic. Each observation of this statistic was of the percentage of service stations in use at the time of the observation. The problem is how to handle a segment of a time series which has no variance. The average coherence values are based on spectra which are decompositions of the variance. When there is no variance, there are no average coherence values. In this experiment, a constant value of 100% utilization was observed for 200 simulated hours, or one segment of the time series. The rule adopted by this investigator was that if 95% of the values in the following segment are the same as the constant value in the segment with no variance, conclude that the series is in equilibrium. If this is not the case, continue waiting for equilibrium. This problem will relate to the nature of the simulation or the investigator's knowledge of the system.

The spectra from the three series were interpreted to determine if the cyclic elements identified from the spectra could be identified as having a particular physical cause. The most frequently occurring cyclic elements in the size of the system statistic, shown on Table 4.5, are 13.1 items per cycle and 10.5 items per cycle. The only relationship between the cyclic elements in this statistic and the parameters of the simulation is between the second element and the average time between arrivals in the system. The average time between arrivals was 1.05 hours, which is one-tenth of the second element. The two most frequent elements in the queue size statistic are 10.5 items per cycle and 13.1 items per cycle. These are the same elements found in the size of the system statistic. The most frequently occurring cyclic element of the utilization statistic was 10.0 items per cycle. This is very close to the 10.5 items per cycle element found in the other two statistics. No relationship, other than the one mentioned, could be found between the cyclic elements and the simulation. It is interesting to notice in Tables 4.5, 4.6, and 4.7 that only the two most frequently observed cyclic elements are present in the spectra after the statistics have reached equilibrium. All other elements disappear from the spectra after the transient stage. This is indicative of variance which is changing, as is the case in the transient stage of a simulation.

CONCLUSIONS

The problem which was considered is the estimation of the end of the transient stage of a simulation. Since the steady state of a simulation is characterized by stable variance and the transient stage of a simulation is characterized by unstable variance, the end of the transient stage can be detected by a change in the spectrum. If the variance of two successive segments of a simulation statistic are similar, the statistic is in the steady state. It was found that cross-spectral analysis can be used to solve the problem. It has been shown that by comparing two successive segments of a simulation statistic, it is possible to detect a change in variance.

An important requirement of the solution is that it be dynamic. The solution must be programmable and capable of detecting the end of the transient stage while the simulation is in progress. The cross-spectral technique is dynamic. The investigator need only determine the size of the segments of the statistic and the number of lags to be considered (M). The spectrum of each segment can be computed as the statistic is generated and retained in memory until the next spectrum has been computed. The cross-spectrum can then be computed and the coherence diagram found. Since the size of the segments are fixed, as is M, the average coherence can always be compared to a fixed critical value as described above. Once the average coherence value exceeds the critical value, it is known that the statistic is in equilibrium and the simulation can be constructed to stop testing for the end of the transient stage.

This solution has some limitations. At the present time some simulation languages, such as GPSS, require special equipment to be compatable with FORTRAN. The solution requires a FORTRAN subroutine to compute the spectrum of each segment. Computer time might be a limitation, depending upon the nature of the simulation. At no time was more than four minutes of computer time required to compute a spectrum.

Further research is encouraged in this area. In trying to resolve the problem, practical considerations such as number of observations required or complexity of computer programming were not taken into account. The major consideration was to resolve the problem in a way which did not disregard the effect of autocorrelation and could solve the problem in a dynamic fashion. It may be possible to achieve a dynamic solution which accounts for autocorrelation without using spectral analysis, perhaps by some comparison of the autocorrelation functions of successive segments of the simulation statistic. Another interesting problem which needs to be studied with regard to this solution is the effect of changing the time between observations and the effect of using disjoint segments rather than overlapping segments as was done in this study.

APPENDIX A:

1) Spectral Analysis Program and listing of input variables

2) Sample output of Spectral Analysis Program

3) Sample Spectrum

LISTING OF INPUT VARIABLES FOR SPECTRAL ANALYSIS PROGRAM

N - length of time series to be considered

M - lag to be used

NQ - number of plot points in spectrum

NVAR - number of different time series to be considered

X(I) - the individual observations

THE FOLLOWING DOCUMENT HAS PRINTING THAT EXTENDS INTO THE BINDING.

THIS IS AS RECEIVED FROM CUSTOMER.

ILLEGIBLE

THE FOLLOWING DOCUMENT (S) IS ILLEGIBLE DUE TO THE PRINTING ON THE ORIGINAL BEING CUT OFF

ILLEGIBLE

```
C        THIS PROGRAM IS USED TO ARRANGE THE INPUT DATA FOR THE PARZEN  FRH 1
C      SUBROUTINES WHICH FOLLCW.                                        FRH 1
C                                                                       FRH 1
C                                                                       FRH 1
C        PR....PROBLEM NUMBER (MAY BE ALPHAMERIC)                       PLRG1
C        PR1...PROBLEM NUMBER (CONTINUED)                               PLRG1
C        N.....NUMBER OF OBSERVATIONS                                   PLRG1
C        NPLOT.OPTION CODE FOR PLOTTING                                 PLRG1
C              0  IF PLOT IS NOT DESIRED.                               PLRG1
C              1  IF PLOT IS DESIRED.                                   PLRG1
C     M.....LAG USED IN PARZN1 AND PARZN2
C     NQ.....NUMBER OF POINTS PLOTTED
C     IW.OPTION CODE FOR SPECTRAL WINDOW USED
C     0      IF TUKEY-HANNING WINDOW USED
C     1      IF PARZEN WINDOW USED
C     IPROB.....NUMBER OF PRCBLEMS TO BE PROCESSED
C     MDEG.....HIGHEST DEGREE POLYNOMIAL SPECIFIED
C                                                                       PLRG1
      DIMENSION XL(800),A(1500),X(800)                                   FRH
      REAL XR(800)/800*0.0/                                             FRH 1
    1 FORMAT(5I3)                                                       FRH 1
    2 FORMAT(15F5.0,5X)                                                 FRH 1
    3 FORMAT('                      SAMPLE    '/(15F8.0))               FRH 1
    5 FORMAT(A4,A2,I2,I1)                                               FRH 1
    6 FORMAT(27H1POLYNOMIAL REGRESSION.....,A4,A2/)                     FRH 1
    7 FORMAT(' N=',I4,' M=',I3,' NQ=',I3,' IW=',I3,' MDEG=',I3,' NPLOT='FRH 1
     1,I3)                                                              FRH 1
    9 FORMAT('                    ****DATA SET',I2,'**** ')             FRH 1
      READ(1,1)N,M,NQ,IW,IPRCB                                          FRH 1
      DO 8 K=1,IPROB                                                    FRH 1
      READ(1,5) PR,PR1,MDEG,NPLOT                                       FRH 1
      WRITE(3,9) K                                                      FRH 1
      WRITE(3,6) PR,PR1
      WRITE(3,7) N,M,NQ,IW,MDEG,NPLOT                                   FRH 1
      READ(1,2)(X(I),I=1,N)                                             FRH 1
      WRITE(3,3)(X(I),I=1,N)                                            FRH 1
      L=N*MDEG
      DO 110 I=1,N
      A(L+I)=X(I)
  110 A(I)=I-151
      DO 4 J=1,N                                                        FRH 1
      XL(J)=J                                                           FRH 1
    4 XL(J+N)=X(J)                                                      FRH 1
      CALL POLRG(N,MDEG,NPLOT,A,XR)
      DO 100 M=1,100,25
  100 CALL PARZN1(XR,N,NQ,M,IW)
    8 CONTINUE                                                          FRH 1
      END                                                               FRH 1
```

```
      SUBROUTINE POLRG(N,M,NPLOT,X,XR)
C                                                                       PLRG
C     A REVISIJN OF THE SAMPLE PROGRAM, POLRG, FROM THE IBM
C     SCIENTIFIC SUBROUTINE PACKAGE, VERSION III, PAGES 408-412.
C                                                                       PLRG
C        SAMPLE MAIN PROGRAM FOR POLYNOMIAL REGRESSION - POLRG          PLRG
C                                                                       PLRG
C        PURPOSE                                                        PLRG
C           (1) READ THE PROBLEM PARAMETER CARD FOR A POLYNOMIAL REGRES-PLRG
C           SIJN, (2) CALL SUBROUTINES TO PERFORM THE ANALYSIS, (3)     PLRG
C           PRINT THE REGRESSION COEFFICIENTS AND ANALYSIS OF VARIANCE  PLRG
C           TABLE FOR POLYNOMIALS OF SUCCESSIVELY INCREASING DEGREES,   PLRG
C           AND (4) OPTIONALLY PRINT THE TABLE OF RESIDUALS AND A PLOT  PLRG
C           OF Y VALUES AND Y ESTIMATES.                                PLRG
C                                                                       PLRG
C        REMARKS                                                        PLRG
C           THE NUMBER OF OBSERVATIONS, N, MUST BE GREATER THAN M+1,    PLRG
C           WHERE M IS THE HIGHEST DEGREE POLYNOMIAL SPECIFIED.         PLRG
C           IF THERE IS NO REDUCTION IN THE RESIDUAL SUM OF SQUARES     PLRG
C           BETWEEN TWO SUCCESSIVE DEGREES OF THE POLYNOMIALS, THE      PLRG
C           PROGRAM TERMINATES THE PROBLEM BEFORE COMPLETING THE ANALY- PLRG
C           SIS FOR THE HIGHEST DEGREE POLYNOMIAL SPECIFIED.            PLRG
C                                                                       PLRG
C        METHOD                                                         PLRG
C           REFER TO B. OSTLE, 'STATISTICS IN RESEARCH', THE IOWA STATE PLRG
C           COLLEGE PRESS', 1954, CHAPTER 6.                            PLRG
C                                                                       PLRG
C     ..................................................................PLRG
C                                                                       PLRG
C     THE FOLLOWING DIMENSION MUST BE GREATER THAN OR EQUAL TO THE      PLRG
C     PRODUCT OF N*(M+1), WHERE N IS THE NUMBER OF OBSERVATIONS AND M   PLRG
01)  IEY033I COMMENTS  DELETED
      DIMENSION X(1)                                                    FRH 8
C                                                                       PLRG
C     THE FOLLOWING DIMENSION MUST BE GREATER THAN OR EQUAL TO THE      PLRG
C     PRODUCT OF M*M..                                                  PLRG
C                                                                       PLRG
      DIMENSION DI(100)                                                 PLRG
C                                                                       PLRG
C     THE FOLLJWING DIMENSION MUST BE GREATER THAN OR EQUAL TO          PLRG
C     (M+2)*(M+1)/2..                                                   PLRG
C                                                                       PLRG
      DIMENSION D(66)                                                   PLRG
C                                                                       PLRG
C     THE FOLLOWING DIMENSIONS MUST BE GREATER THAN OR EQUAL TO M..     PLRG
C                                                                       PLRG
      DIMENSION B(10),E(10),SB(10),T(10)                                PLRG
C                                                                       PLRG
C     THE FOLLOWING DIMENSIONS MUST BE GREATER THAN OR EQUAL TO (M+1).. PLRG
C                                                                       PLRG
      DIMENSION XBAR(11),STD(11),COE(11),SUMSQ(11),ISAVE(11)            PLRG
C                                                                       PLRG
C     THE FOLLJWING DIMENSION MUST BE GREATER THAN OR EQUAL TO 10..     PLRG
C                                                                       PLRG
      DIMENSION ANS(10)                                                 PLRG
C                                                                       PLRG
C     THE FOLLOWING DIMENSION WILL BE USED IF THE PLOT OF OBSERVED DATA PLRG
C     AND ESTIMATES IS DESIRED. THE SIZE OF THE DIMENSION, IN THIS      PLRG
```

```
C     CASE, MUST BE GREATER THAN OR EQUAL TO N*3.  OTHERWISE, THE SIZE  PLRG
C     OF DIMENSION MAY BE SET TO 1.                                     PLRG
C                                                                       PLRG
         DIMENSION P(900)                                               FRH 8
C                                                                       PLRG
C     THE FOLLOWING DIMENSION WILL BE USED FOR PLOTTING THE RESIDUALS
      DIMENSION XR(1)
C
C     ..................................................................PLRG
C                                                                       PLRG
C        IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE  PLRG
C        C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION      PLRG
C        STATEMENT WHICH FOLLOWS.                                       PLRG
C                                                                       PLRG
C     DOUBLE PRECISION X,XBAR,STD,D,SUMSQ,DI,E,B,SB,T,ANS,DET,COE       PLRG
C                                                                       PLRG
C        THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS    PLRG
C        APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS      PLRG
C        ROUTINE.                                                       PLRG
C                                                                       PLRG
C                                                                       PLRG
    5 FORMAT(32HOPOLYNOMIAL REGRESSION OF DEGREE,I3)                    PLRG
    6 FORMAT(12HO  INTERCEPT,E20.7)                                     PLRG
    7 FORMAT(26HO  REGRESSION COEFFICIENTS/(6E20.7))                    PLRG
    8 FORMAT(1HO/24X,24HANALYSIS OF VARIANCE FOR,I4,19H  DEGREE POLYNOMIPLRG
     1AL/)                                                              PLRG
    9 FORMAT(1HO,5X,19HSOURCE OF VARIATION,7X,9HDEGREE OF,7X,6HSUM OF,9XPLRG
     1,4HMEAN,10X,1HF,9X,20HIMPROVEMENT IN TERMS/33X,7HFREEDOM,8X,7HSQUAPLRG
     2RES,7X,6HSQUARE,7X,5HVALUE,8X,17HOF SUM OF SQUARES)               PLRG
   10 FORMAT(20HO  DUE TO REGRESSION,12X,I6,F17.5,F14.5,F13.5,F20.5)    PLRG
   11 FORMAT(32H   DEVIATION ABOUT REGRESSION   ,I6,F17.5,F14.5)        PLRG
   12 FORMAT(8X,5HTOTAL,19X,I6,F17.5///)                                PLRG
   13 FORMAT(17HO  NO IMPROVEMENT)                                      PLRG1
   14 FORMAT(1HO//27X,18HTABLE OF RESIDUALS//16H OBSERVATION NO.,5X,7HX PLRG1
     1VALUE,7X,7HY VALUE,7X,10HY ESTIMATE,7X,8HRESIDUAL/)               PLRG1
   15 FORMAT(1HO,3X,I6,F18.5,F14.5,F17.5,F15.5)                         PLRG1
C                                                                       PLRG1
C     ..................................................................PLRG1
C                                                                       PLRG1
C                                                                       PLRG1
      CALL GDATA (N,M,X,XBAR,STD,D,SUMSQ)                               PLRG1
C                                                                       PLRG1
      MM=M+1                                                            PLRG1
      SUM=0.0                                                           PLRG1
      NT=N-1                                                            PLRG1
C                                                                       PLRG1
      DO 200 I=1,M                                                      PLRG1
      ISAVE(I)=I                                                        PLRG1
C                                                                       PLRG1
C     FORM SUBSET OF CORRELATION COEFFICIENT MATRIX                     PLRG1
C                                                                       PLRG1
      CALL ORDER (MM,D,MM,I,ISAVE,DI,E)                                 PLRG1
C                                                                       PLRG1
C     INVERT THE SUBMATRIX OF CORRELATION COEFFICIENTS                  PLRG1
C                                                                       PLRG1
      CALL MINV (DI,I,DET,B,T)                                          PLRG1
C                                                                       PLRG1
      CALL MULTR (N,I,XBAR,STD,SUMSQ,DI,E,ISAVE,B,SB,T,ANS)             PLRG1
```

```
      C                                                              PLRG1
      C     PRINT THE RESULT OF CALCULATION                          PLRG1
      C                                                              PLRG1
            WRITE (3,5) I                                            PLRG1
            IF(ANS(7)) 140,130,130                                   PLRG1
        130 SUMIP=ANS(4)-SUM                                         PLRG1
            IF(SUMIP) 140, 140, 150                                  PLRG1
        140 WRITE (3,13)                                             PLRG1
            GO TO 210                                                PLRG1
        150 WRITE (3,6) ANS(1)                                       PLRG1
            WRITE (3,7) (B(J),J=1,I)                                 PLRG1
            WRITE (3,8) I                                            PLRG1
            WRITE (3,9)                                              PLRG1
            SUM=ANS(4)                                               PLRG1
            WRITE (3,10) I,ANS(4),ANS(6),ANS(10),SUMIP               PLRG1
            NI=ANS(8)                                                PLRG1
            WRITE (3,11) NI,ANS(7),ANS(9)                            PLRG1
            WRITE (3,12) NT,SUMSQ(MM)                                PLRG1
      C                                                              PLRG1
      C     SAVE COEFFICIENTS FOR CALCULATION OF Y ESTIMATES         PLRG1
      C                                                              PLRG1
            COE(1)=ANS(1)                                            PLRG1
            DO 160 J=1,I                                             PLRG1
        160 COE(J+1)=B(J)                                            PLRG1
            LA=I                                                     PLRG1
        200 CONTINUE                                                 PLRG1
      C                                                              PLRG1
      C     TEST WHETHER PLOT IS DESIRED                             PLRG1
      C                                                              PLRG1
        210 IF(NPLOT) 300,300,220
      C                                                              PLRG1
      C        CALCULATE ESTIMATES                                   PLRG1
      C                                                              PLRG1
        220 NP3=N+N                                                  PLRG1
            DO 230 I=1,N                                             PLRG1
            NP3=NP3+1                                                PLRG1
            P(NP3)=COE(1)                                            PLRG1
            L=I                                                      PLRG1
            DO 230 J=1,LA                                            PLRG1
            P(NP3)=P(NP3)+X(L)*COE(J+1)                              PLRG1
        230 L=L+N                                                    PLRG1
      C                                                              PLRG1
      C        COPY OBSERVED DATA                                    PLRG1
      C                                                              PLRG1
            N2=N                                                     PLRG1
            L=N*M                                                    PLRG1
            DO 240 I=1,N                                             PLRG1
            P(I)=X(I)                                                PLRG2
            N2=N2+1                                                  PLRG2
            L=L+1                                                    PLRG2
        240 P(N2)=X(L)                                               PLRG2
      C                                                              PLRG2
      C     PRINT TABLE OF RESIDUALS                                 PLRG2
      C                                                              PLRG2
            WRITE (3,5) LA                                           PLRG2
            WRITE (3,14)                                             PLRG2
            NP2=N                                                    PLRG2
            NP3=N+N                                                  PLRG2
```

```
      DO 250 I=1,N                                          PLRG2
      NP2=NP2+1                                             PLRG2
      NP3=NP3+1                                             PLRG2
      XR(I)=I
      XR(I+N)=P(NP2)-P(NP3)
  250 WRITE(3,15) I,P(I),P(NP2),P(NP3),XR(I+N)
C                                                           PLRG2
      CALL PLOT(4,XR,N,2,N,0)
C                                                           PLRG2
  300 CONTINUE
      RETURN
      END                                                   PLRG2
```

```
      SUBROUTINE PARZN1(X,N,NQ,M,IW)                                    FRH 1
C                                                                       FRH 1
C     THIS PROCEDURE COMPUTES THE AUTO CORRELATION FUNCTION, R1, FOR    FRH 1
C     I=1,2,....M+1.  THE FUNCTION AT LAG M IS STORED AT I=M+1.  THE    FRH 1
C     TIME SERIES ARE OF LENGTH, N, AND IS STORED IN THE ARRAY Y,       FRH 1
C     BEGINNING AT L1.  THE AUTO CORRELATION FUNCTION IS NORMALIZED     FRH 1
C     TO HAVE A VALUE 1 AT THE ORIGIN.  THE NORMALIZING FACTOR IS D1.   FRH 1
C     THE FUNCTIONS ARE ADDED INTO THE ARRAY R1.                        FRH 1
C                                                                       FRH 1
    1 FORMAT(2I4)                                                       FRH 1
    2 FORMAT(4X,F5.0,4X,F5.0)                                           FRH 1
    3 FORMAT(F12.5,12X,F12.5)                                           FRH 1
      DIMENSION X(1)                                                    FRH 1
      REAL R1(400)/400*0.0/,CI(400)/400*0.0/                            FRH 1
      DATA D1,D3/0.0,0.0/                                               FRH 1
      MM=0                                                              FRH 1
      DO 5 I=1,N                                                        FRH 1
      D1=D1+X(I)**2                                                     FRH 1
    5 CONTINUE                                                          FRH 1
      MM=M+1                                                            FRH 1
      DO 7 KK=1,MM                                                      FRH 1
      KK1=KK-1                                                          FRH 1
      NM=N+KK-1                                                         FRH 1
      NK=N-KK+1                                                         FRH 1
      SUM1=0.0                                                          FRH 1
      DO 6 JL=1,NK                                                      FRH 1
      SUM1=SUM1+X(JL)*X(JL+KK1)                                         FRH 1
    6 CONTINUE                                                          FRH 1
      R1(KK)=SUM1/D1                                                    FRH 1
    7 CONTINUE                                                          FRH 1
      CALL SPLOT(M,R1,2)                                                FRH 1
      CALL PARZN2(R1,NQ,M,IW)                                           FRH 1
      RETURN                                                            FRH 1
      END                                                               FRH 1
```

```
C                                                                     FRH 1
      SUBROUTINE PARZN2(R1,NQ,M,IW)                                   FRH 1
C                                                                     FRH 1
C     THIS PROCEDURE COMPUTES N+1 POINTS OF THE ESTIMATED SPECTRAL    FRH 1
C     DENSITY FUNCTION FROM R(I) WHICH IS THE AUTO CORRELATION FUNCTION FRH 1
C     THE TRUNCATION POINT IS M, THE WEIGHTING KERNEL USED IS OMEGA.  FRH 1
C     THE SINES AND COSINES NEEDED ARE COMPUTED RECURSIVELY           FRH 1
C                                                                     FRH 1
C                                                                     FRH 1
      DIMENSION R1(400),RE(400)                                       FRH 1
      REAL OMEGA(500)/500*0.0/,T1(500)/500*0.0/,T2(500)/500*0.0/,TE(500)FRH 1
     1/500*0.0/,TO(500)/500*0.0/,F1(500)/500*0.0/,CO(500)/500*0.0/    FRH 1
      DATA PI/3.1416/,PIV/0.3183/                                     FRH 1
    1 FORMAT(4F10.5)                                                  FRH 1
    2 FORMAT('  RESULTS OF PARZEN TRANSFORM'//'         F1         F2 FRH 1
     1  CO         QU'/ (F10.5,10X,F10.5))                            FRH 1
      IF(IW) 32,30,32                                                 FRH 1
C                         PARZEN WINDOW                               FRH 1
   32 M1=M/2                                                          FRH 1
      M2=M1+1                                                         FRH 1
      DO 33 K=1,M1                                                    FRH 1
   33 OMEGA(K)=1.-((6.*(K**2))/(M**2))*(1.-((1.*K)/M))                FRH 1
      DO 34 K=M2,M                                                    FRH 1
   34 OMEGA(K)=2.*(1.-(1.*K)/M)                                       FRH 1
      GO TO 31                                                        FRH 1
C                      TUKEY HANNING WINDOW                           FRH 1
   30 DO 3 I=1,M                                                      FRH 1
    3 OMEGA(I)=.5*(1.0+COS(I*PI/M))                                   FRH 1
   31 C=NQ                                                            FRH 1
      C1=COS(PI/C)                                                    FRH 1
      C3=C1                                                           FRH 1
      D1=C1                                                           FRH 1
      C2=SIN(PI/C)                                                    FRH 1
      D4=C2                                                           FRH 1
      D6=2.0*C1                                                       FRH 1
      P1=0.5*R1(1)                                                    FRH 1
      K=M+1                                                           FRH 1
      DO 4 I=2,K                                                      FRH 1
      A=OMEGA(I)                                                      FRH 2
      P1=P1+R1(I)*A                                                   FRH 2
    4 CONTINUE                                                        FRH 2
      F1(1)=P1*PIV                                                    FRH 2
      DO 6 I=1,NQ                                                     FRH 2
      U11=0.0                                                         FRH 2
      U12=U11                                                         FRH 2
      U13=U11                                                         FRH 2
      U14=U11                                                         FRH 2
      U21=U11                                                         FRH 2
      U22=U11                                                         FRH 2
      DO 5 J=2,K                                                      FRH 2
      JJ=M+3-J                                                        FRH 2
      A=OMEGA(JJ)                                                     FRH 2
      U31=D6*U21-U11+R1(JJ)*A                                         FRH 2
      U11=U21                                                         FRH 2
      U21=U31                                                         FRH 2
      U12=U22                                                         FRH 2
    5 CONTINUE                                                        FRH 2
      F1(I+1)=(D1*U21-U11+R1(1)*0.5)*PIV                              FRH 2
```

```
      D1=(C1*C3)-(C2*D4)                                      FRH 2
      D4=(D4*C1)+(C3*C2)                                      FRH 2
      C3=D1'                                                  FRH 2
      D6=2.0*D1                                               FRH 2
    6 CONTINUE                                                FRH 2
      CALL SPLOT(NQ,F1,3)                                     FRH 2
      RETURN                                                  FRH 2
      END                                                     FRH 2
```

```
      C                                                                  FRH 2
      C                                                                  FRH 2
            SUBROUTINE SPLOT(N,X,NO)                                     FRH 2
      C                                                                  FRH 2
      C        THIS ROUTINE IS USED TO MODIFY THE RESULTS FROM THE PARZEN SUBRFRH 2
      C        SO THAT THEY CAN BE GRAPHED BY SUBROUTINE PLOT.           FRH 2
      C                                                                  FRH 2
            DIMENSION X(800),OUT(101),YPR(11),ANG(9),PLT(1600),Y(800)    FRH 2
        102 FORMAT(' PLOT VALUES '/(8F14.5))                             FRH 2
      C                                                                  FRH 2
      C     INITIALIZE VALUES                                            FRH 2
            PI=3.1415927                                                 FRH 2
            K=0                                                          FRH 2
            M=2                                                          FRH 2
            IF(NO.NE.3)     GO TO 10                                     FRH 2
            DO 106 I=1,N                                                 FRH 2
            IF(X(I))100,100,99                                           FRH 2
         99 X(I)=ALOG(X(I))                                              FRH 2
            GO TO 106                                                    FRH 2
        100 X(I)=-10.0                                                   FRH 2
        106 CONTINUE                                                     FRH 2
            DO 8 I=1,N                                                   FRH 2
            PLT(I)=I*PI/N                                                FRH 2
            PLT(I+N)=X(I)                                                FRH 2
          8 CONTINUE                                                     FRH 2
            GO TO 20                                                     FRH 2
         10 DO 11 I=1,N                                                  FRH 2
            PLT(I)=I                                                     FRH 2
            PLT(I+N)=X(I)                                                FRH 2
         11 CONTINUE                                                     FRH 2
         20 CALL PLOT(NO,PLT,N,M,N,0)                                    FRH 2
            WRITE(3,102) (X(I),I=1,N)                                    FRH 2
            RETURN                                                       FRH 2
            END                                                          FRH 2
```

```
C                                                                       FRH 2
      SUBROUTINE PLOT(NO,A,N,M,NL,NS)                                   FRH 2
C                                                                       FRH 2
      DIMENSION OUT(101),YPR(11),ANG(9),A(1)                            FRH 2
C                                                                       FRH 2
    1 FORMAT(1H1,60X,7H CHART ,I3,//)                                   FRH 2
    2 FORMAT(1H ,F11.4,5H+    ,101A1)                                   FRH 2
    3 FORMAT(1H )                                                       FRH 2
    4 FORMAT(10H *+0156789)                                             FRH 2
    5 FORMAT( 10A1)                                                     FRH 2
    7 FORMAT(1H ,16X,101H.         .         .         .         .      FRH 2
     1    .         .         .         .         .         .)          FRH 2
    8 FORMAT(1H0,9X,11F10.4//)                                          FRH 2
    9 FORMAT(1H ,16X,101A1)                                             FRH 2
  200 FORMAT( 10X,'      PLOT OF DATA')                                 FRH 2
  201 FORMAT( 10X,'     PLOT OF AUTO CORR. FUNCTION')                   FRH 2
  202 FORMAT( 10X,'      PLOT OF SPECTRUM')                             FRH 2
  203 FORMAT( 10X,'           PLOT OF RESIDUALS')                       FRH 2
C                                                                       FRH 2
C     ..................................................................FRH 2
C                                                                       FRH 2
      NLL=NL                                                            FRH 2
C                                                                       FRH 2
      IF(NS)16,16,10                                                    FRH 2
C                                                                       FRH 2
C        SORT BASE VARIABLE IN ASCENDING ORDER                          FRH 2
C                                                                       FRH 2
   10 DO 15 I=1,N                                                       FRH 2
      DO 14 J=1,N                                                       FRH 2
      IF(A(I)-A(J))14,14,11                                             FRH 2
   11 L=I-N                                                             FRH 2
      LL=J-N                                                            FRH 2
      DO 12 K=1,M                                                       FRH 2
      L=L+N                                                             FRH 2
      LL=LL+N                                                           FRH 2
      F=A(L)                                                            FRH 2
      A(L)=A(LL)                                                        FRH 2
   12 A(LL)=F                                                           FRH 2
   14 CONTINUE                                                          FRH 2
   15 CONTINUE                                                          FRH 3
C                                                                       FRH 3
C        TEST NLL                                                       FRH 3
C                                                                       FRH 3
   16 IF(NLL)20,18,20                                                   FRH 3
   18 NLL=50                                                            FRH 3
C                                                                       FRH 3
C        PRINT TITLE                                                    FRH 3
C                                                                       FRH 3
   20 WRITE(3,1)NO                                                      FRH 3
      GO TO (91,92,93,94),NO                                            FRH 3
   91 WRITE(3,200)                                                      FRH 3
      GO TO 21                                                          FRH 3
   92 WRITE(3,201)                                                      FRH 3
      GO TO 21                                                          FRH 3
   93 WRITE(3,202)                                                      FRH 3
      GO TO 21                                                          FRH 3
   94 WRITE(3,203)                                                      FRH 3
   21 CONTINUE                                                          FRH 3
```

```
C
C          DEVELOP BLANKS AND DIGITS FOR PRINTING
C
         REWIND 10
         WRITE(10,4)
         REWIND 10
         READ(10,5)BLANK,(ANG(I),I=1,9)
         REWIND 10
C
C          FIND SCALE FOR BASE VARIABLE
C
         XSCAL=(A(N)-A(1))/(FLOAT(NLL-1))
C
C          FIND SCALE FOR CROSS VARIABLES
C
         M1=N+1
         YMIN=A(M1)
         YMAX=YMIN
         M2=M*N
         DO 40 J=M1,M2
         IF(A(J)-YMIN)28,28,26
      26 IF(A(J)-YMAX)40,40,30
      28 YMIN=A(J)
         GO TO 40
      30 YMAX=A(J)
      40 CONTINUE
         YSCAL=(YMAX-YMIN)/100.0
C
C          FIND BASE VARIABLE PRINT POSITION
C
         XB=A(1)
         L=1
         MY=M-1
         I=1
      45 F=I-1
         XPR=XB+F*XSCAL
C        IF(A(L)-XPR)51,51,70        THIS CARD HAS BEEN REMOVED
C
C          FIND CROSS VARIABLES
C
      51 DO 55 IX=1,101
      55 OUT(IX)=BLANK
      57 DO 60 J=1,MY
         LL=L+J*N
         JP=((A(LL)-YMIN)/YSCAL)+1.0
         OUT(JP)=ANG(J)
      60 CONTINUE
C
C          PRINT LINE AND CLEAR, OR SKIP
C
         WRITE(3,2)XPR,(OUT(IZ),IZ=1,101)
         L=L+1
         GO TO 80
      70 WRITE(3,3)
      80 I=I+1
         IF(I-NLL)45,84,86
      84 XPR=A(N)
         GO TO 51
```

```
      C                                                                FRH 3
      C          PRINT CROSS VARIABLES NUMBERS                         FRH 3
      C                                                                FRH 3
         86 WRITE(3,7)                                                 FRH 3
            YPR(1)=YMIN                                                FRH 3
            DO 90 KN=1,9                                               FRH 3
         90 YPR(KN+1)=YPR(KN)+YSCAL*10.0                               FRH 3
            YPR(11)=YMAX                                               FRH 3
            WRITE(3,8)(YPR(IR),IR=1,11)                                FRH 3
            RETURN                                                     FRH 3
            END                                                        FRH 3
```

SAMPLE OUTPUT

The time series used for this sample was a series of random numbers generated from the random number generator used in this study. The ourput is organized such that the input variables are shown first, then the auto-correlation and cross-correlations are given, and finally the results of the Parzen transformation, which is the spectrum.

ILLEGIBLE DOCUMENT

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

THIS IS THE BEST COPY AVAILABLE

N= 500 M=150 NQ=100 NVAR= 1

THE INPUT

X	Y					
26.00000	47.00000	39.00000	89.00000	59.00000	16.00000	16.00000
63.00000	21.00000	50.00000	6.00000	38.00000	6.00000	60.00000
64.00000	37.00000	62.00000	93.00000	34.00000	31.00000	67.00000
78.00000	5.00000	71.00000	18.00000	13.00000	47.00000	24.00000
26.00000	5.00000	48.00000	82.00000	37.00000	16.00000	56.00000
52.00000	5.00000	43.00000	46.00000	91.00000	2.00000	81.00000
13.00000	97.00000	5.00000	89.00000	23.00000	19.00000	50.00000
18.00000	10.00000	3.00000	88.00000	17.00000	53.00000	26.00000
31.00000	71.00000	33.00000	45.00000	23.00000	4.00000	86.00000
7.00000	29.00000	73.00000	64.00000	48.00000	73.00000	13.00000
21.00000	30.00000	20.00000	49.00000	28.00000	47.00000	32.00000
23.00000	37.00000	32.00000	89.00000	74.00000	2.00000	14.00000
16.00000	4.00000	69.00000	58.00000	94.00000	73.00000	58.00000
18.00000	2.00000	52.00000	30.00000	55.00000	77.00000	82.00000
72.00000	43.00000	17.00000	67.00000	95.00000	14.00000	34.00000
46.00000	72.00000	60.00000	77.00000	80.00000	20.00000	62.00000
24.00000	80.00000	40.00000	39.00000	74.00000	37.00000	25.00000
4.00000	34.00000	67.00000	61.00000	18.00000	20.00000	25.00000
15.00000	15.00000	94.00000	19.00000	32.00000	19.00000	30.00000
56.00000	60.00000	85.00000	42.00000	77.00000	4.00000	19.00000
67.00000	91.00000	23.00000	5.00000	35.00000	69.00000	92.00000
64.00000	24.00000	24.00000	31.00000	14.00000	17.00000	13.00000
20.00000	35.00000	73.00000	10.00000	8.00000	93.00000	29.00000
71.00000	28.00000	67.00000	42.00000	46.00000	45.00000	0.00000
77.00000	57.00000	20.00000	20.00000	65.00000	99.00000	33.00000
70.00000	41.00000	86.00000	64.00000	87.00000	65.00000	86.00000
36.00000	52.00000	47.00000	84.00000	51.00000	70.00000	93.00000
5.00000	17.00000	3.00000	71.00000	41.00000	87.00000	93.00000
20.00000	14.00000	16.00000	29.00000	11.00000	16.00000	53.00000
70.00000	74.00000	60.00000	77.00000	81.00000	99.00000	87.00000
99.00000	20.00000	31.00000	58.00000	24.00000	93.00000	63.00000
66.00000	4.00000	40.00000	50.00000	24.00000	87.00000	74.00000
43.00000	42.00000	69.00000	14.00000	68.00000	51.00000	77.00000
53.00000	55.00000	95.00000	74.00000	90.00000	33.00000	42.00000
18.00000	72.00000	3.00000	44.00000	78.00000	87.00000	97.00000
86.00000	19.00000	24.00000	46.00000	57.00000	71.00000	69.00000
54.00000	90.00000	99.00000	46.00000	10.00000	0.00000	74.00000
60.00000	20.00000	59.00000	70.00000	66.00000	55.00000	49.00000
93.00000	13.00000	18.00000	26.00000	58.00000	90.00000	83.00000
33.00000	19.00000	23.00000	74.00000	28.00000	80.00000	45.00000
9.00000	66.00000	79.00000	97.00000	49.00000	41.00000	74.00000
74.00000	71.00000	54.00000	39.00000	74.00000	84.00000	54.00000
74.00000	75.00000	95.00000	45.00000	56.00000	85.00000	43.00000
6.00000	33.00000	95.00000	9.00000	64.00000	77.00000	77.00000
10.00000	57.00000	5.00000	71.00000	7.00000	18.00000	92.00000
77.00000	18.00000	25.00000	69.00000	23.00000	63.00000	54.00000
38.00000	10.00000	43.00000	63.00000	5.00000	98.00000	98.00000
61.00000	46.00000	64.00000	61.00000	86.00000	21.00000	58.00000
86.00000	29.00000	45.00000	11.00000	91.00000	56.00000	30.00000
25.00000	29.00000	47.00000	14.00000	73.00000	82.00000	6.00000
78.00000	72.00000	63.00000	64.00000	83.00000	36.00000	77.00000
26.00000	17.00000	68.00000	22.00000	90.00000	84.00000	20.00000
60.00000	86.00000	34.00000	65.00000	60.00000	53.00000	54.00000
99.00000	41.00000	21.00000	28.00000	45.00000	6.00000	2.00000
84.00000	61.00000	97.00000	57.00000	34.00000	11.00000	81.00000
61.00000	23.00000	32.00000	63.00000	95.00000	2.00000	27.00000

91.00000	46.00000	75.00000	3.00000	40.00000	75.00000	64.00000
28.00000	95.00000	83.00000	24.00000	19.00000	11.00000	98.00000
35.00000	98.00000	25.00000	42.00000	60.00000	83.00000	51.00000
90.00000	61.00000	57.00000	35.00000	70.00000	97.00000	33.00000
75.00000	42.00000	22.00000	39.00000	28.00000	99.00000	6.00000
66.00000	18.00000	28.00000	76.00000	20.00000	46.00000	56.00000
0.00000	11.00000	22.00000	45.00000	26.00000	17.00000	33.00000
34.00000	77.00000	63.00000	54.00000	9.00000	34.00000	47.00000
25.00000	6.00000	3.00000	61.00000	87.00000	97.00000	84.00000
29.00000	12.00000	98.00000	57.00000	18.00000	25.00000	35.00000
96.00000	39.00000	32.00000	99.00000	18.00000	99.00000	79.00000
42.00000	91.00000	16.00000	28.00000	33.00000	2.00000	55.00000
97.00000	27.00000	5.00000	14.00000	45.00000	76.00000	55.00000
98.00000	36.00000	23.00000	51.00000	12.00000	83.00000	19.00000
88.00000	32.00000	91.00000	87.00000	37.00000	62.00000	3.00000
91.00000	69.00000	36.00000	72.00000	35.00000	27.00000	98.00000
70.00000	23.00000	62.00000	89.00000	30.00000	40.00000	54.00000
90.00000	62.00000	64.00000	13.00000	51.00000	31.00000	71.00000
89.00000	48.00000	19.00000	51.00000	93.00000	96.00000	88.00000
88.00000	50.00000	77.00000	44.00000	26.00000	84.00000	87.00000
32.00000	79.00000	67.00000	81.00000	54.00000	99.00000	21.00000
8.00000	42.00000	58.00000	1.00000	19.00000	83.00000	71.00000
76.00000	44.00000	21.00000	3.00000	0.00000	10.00000	35.00000
97.00000	5.00000	10.00000	22.00000	53.00000	20.00000	84.00000
81.00000	7.00000	30.00000	30.00000	23.00000	43.00000	85.00000
93.00000	14.00000	68.00000	79.00000	69.00000	59.00000	2.00000
2.00000	59.00000	51.00000	6.00000	61.00000	10.00000	90.00000
95.00000	27.00000	83.00000	83.00000	71.00000	96.00000	85.00000
37.00000	11.00000	75.00000	99.00000	75.00000	53.00000	11.00000
55.00000	51.00000	9.00000				

THE AUTO-CORRELATION FUNCTIONS ,R1(),R2() AND
THE CROSS-CORRELATION FUNCTIONS,CI(),CT()

R1	R2	CI	CT
1.00000	1.00000	0.75291	0.75291
0.74963	0.75807	0.73606	0.74324
0.74203	0.73838	0.74773	0.73387
0.73582	0.73603	0.73038	0.73330
0.74241	0.73823	0.74363	0.73085
0.73106	0.75864	0.73336	0.74091
0.75076	0.74851	0.75069	0.72847
0.75791	0.76433	0.73262	0.73373
0.74328	0.72657	0.75931	0.74193
0.76075	0.72933	0.70512	0.72868
0.72584	0.73615	0.72746	0.73809
0.74489	0.73936	0.74487	0.73449
0.72164	0.73646	0.72594	0.71855
0.75765	0.73754	0.75203	0.73617
0.74136	0.73377	0.71690	0.73198
0.72377	0.72926	0.72761	0.72590
0.71033	0.72251	0.73574	0.73898
0.72710	0.70112	0.73131	0.71186
0.71953	0.71475	0.71439	0.71690
0.72568	0.70994	0.71675	0.69272
0.73652	0.71617	0.72895	0.72021
0.72401	0.70723	0.71771	0.72341
0.72737	0.70911	0.73138	0.72187
0.71445	0.70228	0.70153	0.72309
0.71245	0.70738	0.73901	0.72242
0.70665	0.71715	0.71835	0.73160
0.71107	0.71679	0.70978	0.70912

0.72905	0.69490	0.68688	0.71203
0.73964	0.70465	0.71403	0.70048
0.70556	0.70959	0.71301	0.71071
0.70060	0.70641	0.69169	0.68984
0.71228	0.67649	0.71110	0.68992
0.70070	0.70159	0.70194	0.71167
0.71098	0.70034	0.70089	0.69490
0.72110	0.68467	0.70472	0.70878
0.69818	0.69431	0.70517	0.70616
0.71545	0.69423	0.69793	0.70361
0.70040	0.69257	0.69150	0.69484
0.69931	0.68242	0.66938	0.70263
0.69526	0.68126	0.70047	0.67080
0.70388	0.70097	0.68396	0.68745
0.70644	0.67841	0.66624	0.69371
0.69385	0.69384	0.66888	0.68159
0.68075	0.68244	0.67292	0.67878
0.69104	0.67211	0.67798	0.68703
0.70423	0.66882	0.67984	0.68636
0.65124	0.69080	0.66467	0.69260
0.68752	0.68938	0.66209	0.67691
0.70052	0.67850	0.66669	0.68533
0.69141	0.67993	0.66687	0.69459
0.67970	0.69110	0.68025	0.67661
0.68424	0.67312	0.66187	0.69290
0.64893	0.66377	0.66146	0.69692
0.66876	0.68751	0.65715	0.70191
0.67367	0.67624	0.67373	0.70486
0.65622	0.68436	0.63455	0.68443
0.67759	0.66275	0.64525	0.67783
0.68309	0.67259	0.67178	0.67899
0.65885	0.67329	0.67280	0.68985
0.66271	0.64705	0.66550	0.67789
0.66879	0.67515	0.64516	0.68754
0.65785	0.65058	0.66649	0.67636
0.66954	0.65251	0.63791	0.67927
0.65573	0.65938	0.64925	0.68654
0.67364	0.65369	0.67072	0.67713
0.66104	0.67348	0.65431	0.68405
0.67039	0.63958	0.64930	0.67534
0.66886	0.64158	0.65778	0.65565
0.65825	0.65213	0.64751	0.66690
0.66059	0.65511	0.63640	0.66280
0.66537	0.66031	0.63871	0.65357
0.65752	0.63570	0.64994	0.67453
0.65540	0.63848	0.65014	0.66911
0.64823	0.64022	0.65762	0.66570
0.65806	0.64031	0.62665	0.63591
0.66477	0.62919	0.62486	0.65458
0.66170	0.64549	0.63436	0.65286
0.64636	0.63584	0.63258	0.65422
0.65281	0.62717	0.62977	0.66603
0.64879	0.63118	0.62551	0.64543
0.63312	0.63131	0.61896	0.64657
0.64599	0.63526	0.61659	0.63243
0.64859	0.62126	0.61462	0.64590
0.64366	0.63981	0.61296	0.64077
0.62592	0.61095	0.62580	0.64126
0.65142	0.61887	0.61377	0.63432
0.64126	0.62351	0.61490	0.64619

0.62508	0.60871	0.60849	0.65412
0.63538	0.61279	0.60938	0.64878
0.64136	0.59525	0.60233	0.63075
0.63454	0.61986	0.59881	0.62769
0.61361	0.59255	0.60408	0.64437
0.62406	0.62111	0.61462	0.64225
0.63472	0.62120	0.62124	0.62404
0.62617	0.60158	0.61079	0.62618
0.62235	0.60250	0.58509	0.61855
0.61312	0.58235	0.59417	0.62927
0.62516	0.58312	0.59359	0.62686
0.64143	0.58437	0.59085	0.62191
0.63267	0.59144	0.61296	0.62204
0.59605	0.61815	0.60706	0.60134
0.61091	0.58474	0.60834	0.62495
0.63039	0.60562	0.59376	0.63410
0.59308	0.59290	0.58617	0.62845
0.61233	0.57875	0.59832	0.59156
0.60703	0.59999	0.59902	0.61342
0.60937	0.57837	0.61307	0.59486
0.62776	0.59010	0.59722	0.59973
0.61943	0.57140	0.57768	0.61623
0.59435	0.56814	0.57068	0.59648
0.59920	0.57850	0.57410	0.59736
0.59815	0.56651	0.57871	0.57804
0.58551	0.58594	0.57457	0.59167
0.60724	0.57445	0.56672	0.59218
0.59670	0.56234	0.58827	0.58789
0.59937	0.57676	0.57396	0.59544
0.58167	0.58303	0.54421	0.57844
0.60206	0.56633	0.55456	0.58007
0.59974	0.57290	0.56630	0.58817
0.60890	0.55384	0.56871	0.56943
0.60143	0.56218	0.55684	0.59915
0.59575	0.55129	0.58382	0.57956
0.57807	0.57232	0.56731	0.57462
0.58293	0.55476	0.53628	0.57808
0.57367	0.55163	0.55649	0.57576
0.58264	0.54573	0.52735	0.56997
0.60919	0.55886	0.54920	0.58372
0.58047	0.56298	0.54851	0.59514
0.58247	0.56619	0.56121	0.59395
0.57131	0.55544	0.54481	0.56753
0.59022	0.53996	0.54393	0.57330
0.56971	0.53155	0.53849	0.56080
0.57074	0.54541	0.55257	0.56193
0.56118	0.54495	0.55942	0.56133
0.56731	0.53733	0.55467	0.57756
0.55893	0.53201	0.53765	0.56993
0.57330	0.52777	0.53350	0.57122
0.57103	0.51672	0.54395	0.57123
0.57257	0.51796	0.51677	0.55449
0.54393	0.53321	0.53475	0.56467
0.55034	0.52842	0.55134	0.56158
0.55845	0.51687	0.55534	0.56383
0.56783	0.53677	0.53770	0.56625
0.56396	0.51671	0.51567	0.56707
0.54672	0.51765	0.52497	0.56527
0.55536	0.51108	0.48700	0.55415
0.56123	0.52671	0.51547	0.55622

0.55534	0.51728	0.52855	0.54973
0.53930	0.52208	0.51766	0.54832
0.54208	0.51236	0.50366	0.54474

RESULTS OF PARZEN TRANSFORM

F1	F2	CO	QU
16.33110	16.10693	16.02914	0.00000
3.56782	3.60489	3.58390	9.76773
0.17293	0.15027	0.15037	3.77343
0.04193	0.10956	0.01778	2.57851
0.04458	0.11454	-0.00602	1.96408
0.08612	0.08143	-0.01205	1.52824
0.06908	0.04629	0.03656	1.21040
0.02508	0.05147	0.00028	1.07123
0.03208	0.05450	0.00274	0.95681
0.03434	0.02637	-0.00159	0.83736
0.02683	0.01835	-0.00027	0.73861
0.02227	0.02421	0.00993	0.65663
0.02982	0.03442	0.01419	0.62882
0.03022	0.02752	0.00387	0.56387
0.02609	0.04324	-0.01397	0.50424
0.03412	0.06212	-0.01332	0.44523
0.04531	0.04936	0.00782	0.44984
0.04645	0.04659	0.02261	0.43967
0.04103	0.03322	0.00446	0.42193
0.04086	0.02734	-0.00389	0.41678
0.05464	0.02825	-0.01230	0.38261
0.07267	0.03120	-0.01508	0.32562
0.04441	0.03419	-0.01379	0.23855
0.02941	0.02091	-0.00997	0.30166
0.05218	0.03802	0.01465	0.29817
0.03716	0.03989	0.00104	0.26505
0.01917	0.04823	-0.00027	0.27682
0.03650	0.02545	0.01009	0.26012
0.08643	0.06295	0.03744	0.26476
0.10395	0.07291	-0.03045	0.25643
0.07305	0.07307	0.01567	0.23762
0.04503	0.05387	0.02286	0.24411
0.05005	0.06449	0.00889	0.22674
0.02917	0.05368	0.00710	0.17900
0.02688	0.07649	0.02707	0.19804
0.03365	0.04561	0.01389	0.19406
0.04192	0.03412	0.01898	0.16786
0.04667	0.05135	-0.00839	0.17174
0.01318	0.05499	-0.01066	0.17003
0.01837	0.05068	-0.00913	0.17284
0.02546	0.06482	-0.00059	0.17044
0.03749	0.05588	0.02127	0.15574
0.03763	0.03618	-0.01619	0.17433
0.05586	0.04676	-0.05133	0.13995
0.04948	0.03745	-0.02221	0.14438
0.07804	0.01034	0.00034	0.14346
0.03954	0.02096	-0.01585	0.13615
0.02411	0.03747	-0.00188	0.11307
0.01895	0.04247	0.01094	0.11237
0.06651	0.05031	0.03424	0.13491
0.04067	0.02932	0.03108	0.14390
0.03826	0.01894	0.00545	0.13789
0.01309	0.04349	0.00710	0.10351
0.03606	0.03393	-0.01530	0.10895

0.04515	0.03273	-0.02959	0.10324
0.01989	0.07086	0.00306	0.09172
0.03666	0.07683	0.04251	0.09494
0.04011	0.03251	0.00690	0.09003
0.06368	0.01236	-0.00779	0.09444
0.08485	0.01304	0.00317	0.11517
0.04946	0.06011	0.00170	0.09725
0.03074	0.05266	-0.02000	0.07925
0.01842	0.04548	0.00193	0.08334
0.07129	0.04457	0.01358	0.08155
0.04102	0.03295	-0.01246	0.06998
0.02490	0.03862	-0.01086	0.09269
0.03596	0.03105	0.00022	0.06233
0.06200	0.03092	-0.03102	0.06433
0.02533	0.03413	-0.03111	0.04856
0.01187	0.03291	-0.01883	0.04735
0.03218	0.03115	-0.00948	0.06637
0.08618	0.02250	0.02658	0.07165
0.06188	0.01796	0.02052	0.02953
0.03096	0.03377	0.00095	0.02037
0.02292	0.03983	0.00932	0.03498
0.03598	0.02945	0.02900	0.05647
0.01892	0.05154	0.03104	0.03973
0.02983	0.03316	0.03045	0.05141
0.03688	0.02224	0.00958	0.04439
0.02342	0.03924	0.00221	0.04866
0.02311	0.08775	-0.02019	0.06442
0.02713	0.02700	0.01051	0.04379
0.02472	0.03637	0.00719	0.04503
0.02248	0.03150	-0.01416	0.05830
0.04373	0.04914	-0.01567	0.04233
0.03403	0.02413	-0.00930	0.01369
0.02319	0.07097	-0.03971	0.01711
0.03403	0.06215	-0.04443	-0.00081
0.03884	0.02697	-0.01945	0.01695
0.06911	0.04664	-0.00774	0.01675
0.07144	0.04754	-0.00254	0.04297
0.05792	0.05056	0.03109	0.05510
0.04951	0.03670	0.04881	0.02265
0.06453	0.02597	0.03379	0.00754
0.08257	0.03260	-0.00407	0.00921
0.03489	0.02719	0.00174	-0.00043
0.01959	0.01631	0.01322	0.01248
0.03054	0.00909	0.01357	0.00467
0.02729	0.02255	0.02523	-0.00673
0.02372	0.03193	0.02746	-0.01127
0.02234	0.05034	0.02954	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000

CHART 3

```
         PLOT OF SPECTRUM
0.0314+
0.0628+
0.1257+                                          *
0.1571+                      *
0.1885+                       *
0.2199+                                *
0.2513+                             *
0.2827+               *
0.3142+                  *
0.3456+                   *
0.3770+                *
0.4084+             *
0.4398+                 *
0.4712+                 *
0.5027+               *
0.5341+                   *
0.5655+                       *
0.5969+                       *
0.6283+                      *
0.6597+                      *
0.6911+                          *
0.7226+                              *
0.7540+                       *
0.7854+                 *
0.8168+                         *
0.8482+                    *
0.8796+           *
0.9111+                    *
0.9425+                                *
0.9739+                                   *
1.0053+                              *
1.0367+                       *
1.0681+                        *
1.0996+                 *
1.1310+                *
1.1624+                   *
1.1938+                      *
1.2252+                       *
1.2566+      *
1.2881+           *
1.3195+               *
1.3509+                    *
1.3823+                    *
1.4137+                          *
1.4451+                        *
1.4765+                               *
1.5080+                     *
1.5394+              *
1.5708+           *
1.6022+                            *
1.6336+                      *
1.6650+                     *
```

APPENDIX B:

1) Cross-Spectral Analysis Program

2) Sample of Cross-Spectral Output

3) Sample Coherence Diagram and Method of Calculating Average Coherence

CROSS-SPECTRAL ANALYSIS

PROGRAM

The program is designed to accept two time series, each of N observations. The trends of the two series are removed in the main program using least squares. In most cases, the only remaining components are the variance and random error. If there should be some trend remaining, it will show an unusually high amount of power in the lower frequencies of the spectra.

The de-trended, or whitened, series are then passed to PARZN1. This subroutine calculates the auto- and cross-correlation functions of the two series. The co- and quadature spectra are calculated in PARZN2. The coherence diagram is found in PARZN2, and passed into a plotting routine.

This is the program which was used for all experimental work in this paper.

```
C        THIS PROGRAM IS USED TO ARRANGE THE INPUT DATA FOR THE PARZEN SFRH 1000
C        TIMES WHICH FOLLOW.                                             FRH 1010
C                                                                        FRH 1020
      DIMENSION X(2000),XL(400)
    1 FORMAT(6I5)                                                        FRH 1040
    2 FORMAT(25F3.0,5X)                                                  FRH 1050
    3 FORMAT('              THE INPUT')                                  FRH 1060
    7 FORMAT(' N=',I4,' M=',I3,' NQ=',I3,'  NVAR=',I3,'  IGO=',I3,' NPROFRH 1070
     1B=',I3)                                                            FRH 1071
    9 FORMAT('0',20X,'****DATA SET',I2,'**** TIME SERIES (',I6,',',I6,')FRH 1080
     1,(',I6,',',I6,')')                                                 FRH 1081
   11 FORMAT('           VARIABLE',I3)                                   FRH 1090
   12 FORMAT(5X,15F5.1)                                                  FRH 1171
   13 FORMAT(5X,'XTOT',6X,'YTOT',6X,'ZTOT',6X,'XZ',8X,'YZ',8X,'X2',8X,'Y
     12',8X,'Z2'/8F10.1)
   14 FORMAT (10X,'LEAST SQUARE REGRESSION',I2,'  Z=',F8.3,'+',F8.3,'X')
      READ (1,1) N,M,NQ,NVAR,IGO,NPROB                                   FRH 1110
      WRITE(3,7) N,M,NQ,NVAR,IGO,NPROB                                   FRH 1120
      LAP=100
      IKONT=0
      DO 4 J=1,NVAR
    4 READ(1,2) (X(I+N*(J-1)),I=1,N)                                     FRH 1130
      READ(1,1) ID1,ID2,ID3,ID4                                          FRH 1122
      DO 10 L=1,NPROB                                                    FRH 1121
      WRITE(3,9) L,ID1,ID2,ID3,ID4                                       FRH 1123
      WRITE(3,3)                                                         FRH 1140
      IK=100+(IKONT*N)
      JK=IK+N
      DO 6 J=1,2
      WRITE(3,11) J                                                      FRH 1160
    6 WRITE(3,12) (X(I+LAP*(J-1)),I=IK,JK)
C     FIND AND SUBTRACT THE MEAN FOR EACH VARIABLE                       FRH 1180
      A=N
      B=N+1
      XZ=0.0
      YZ=0.0
      X2=0.0
      Y2=0.0
      Z2=0.0
      XTOT=0.0                                                           FRH 1190
      YTOT=0.0                                                           FRH 1200
      ZTOT=0.0
      DO 8 K=IK,JK
      X2=X2+(X(K)**2)
      LK=K-IK+1
      Y2=Y2+(X(K+LAP)**2)
      Z2=Z2+(LK**2)
      XZ=XZ+(X(K)*LK)
      YZ=YZ+(X(K+LAP)*LK)
      ZTOT=ZTOT+LK
      XTOT=XTOT+X(K)                                                     FRH 1220
    8 YTOT=YTOT+X(K+LAP)
      XBAR=XTOT/N                                                        FRH 1240
      YBAR=YTOT/N                                                        FRH 1250
      ZBAR=ZTOT/N
      C=(ZBAR*ZTOT)-Z2
      XSLOPE=((XBAR*ZTOT)-XZ)/C
      YSLOPE=((YBAR*ZTOT)-YZ)/C
```

```
      X0=XBAR-(ZBAR*XSLOPE)
      Y0=YBAR-(ZBAR*YSLOPE)
      WRITE(3,13) XTOT,YTOT,ZTOT,XZ,YZ,X2,Y2,Z2
      KNT=1
      WRITE (3,14) KNT,X0,XSLOPE
      KNT=2
      WRITE (3,14) KNT,Y0,YSLOPE
      DO 5 J=1,N
      NL=J+IK-1
      ZJ=X0+(XSLOPE*X(NL))
      ZI=Y0+(YSLOPE*X(NL+LAP))
      XL(J)=X(NL)-ZJ
    5 XL(J+N)=X(NL+LAP)-ZI
      CALL PARZN1(XL,N,NQ,M,IGO)                                        FRH 1290
      ID1=ID1+100
      ID2=ID2+100
      ID3=ID3+100
      ID4=ID4+100
      IKONT=IKONT+1
   10 CONTINUE                                                          FRH 1291
      STOP                                                              FRH 1300
      END                                                               FRH 1310
```

```
      SUBROUTINE PARZN1 (X,N,NG,M,IGO)                                  FRH 1320
C                                                                       FRH 1330
C        THIS PROCEDURE COMPUTES THE AUTO AND CROSS CORRELATION FUNCTIONFRH 1340
C        R2(),CI(),AND CT(), FOR I=1,2,...M+1.THE FUNCTION AT LAG M IS SFRH 1350
C        AT I=M+1. THE TIME SERIES ARE OF EQUAL LENGTH, N , AND BOTH AREFRH 1360
C        STORED IN THE ARRAY Y(), ONE BEGINNING AT L1, AND THE OTHER AT FRH 1370
C        THE AUTO CORRELATION FUNCTIONS ARE NORMALIZED TO HAVE A VALUE IFRH 1380
C        ORIGIN, AND THE CROSS CORRELATION FUNCTIONS ARE ALSO CONSISTENTFRH 1390
C        NORMALIZED. THE NORMALIZING FACTORS ARE D1,D2, AND D3. THE FUNCFRH 1400
C        ARE ADDED INTO THE ARRAYS R1(),R2(),CI(),AND CT(), TO ALLOW POOFRH 1410
C        OF COVARIANCES.                                                FRH 1420
C                                                                       FRH 1430
    1 FORMAT('      THE AUTO-CORRELATION FUNCTIONS ,R1(),R2() AND'/'    FRH 1440
     1THE CROSS-CORRELATION FUNCTIONS,CI(),CT()'/'           R1        RFRH 1450
     22          CI              CT')                                   FRH 1460
    3 FORMAT( 4F12.5)                                                   FRH 1470
      DIMENSION X(1)                                                    FRH 1480
      DIMENSION R1(250),R2(250),CI(250),CT(250)
      D1=0.0
      D2=0.0
      D3=0.0
      DO 2 I=1,N
      R1(I)=0.0
      R2(I)=0.0
      CI(I)=0.0
    2 CT(I)=0.0
      DO 5 I=1,N                                                        FRH 1520
      D1=D1+X(I)**2                                                     FRH 1530
      D2=D2+X(I+N)**2                                                   FRH 1540
    5 CONTINUE                                                          FRH 1550
      D3=SQRT(D1*D2)                                                    FRH 1560
      MM=M+1                                                            FRH 1570
      DO 7 KK=1,MM                                                      FRH 1580
      KK1=KK-1                                                          FRH 1590
      NM=N+KK-1                                                         FRH 1600
      NK=N-KK+1                                                         FRH 1610
      SUM1=0.0                                                          FRH 1620
      SUM2=0.0                                                          FRH 1630
      SUM3=0.0                                                          FRH 1640
      SUM4=0.0                                                          FRH 1650
      DO 6 JL=1,NK                                                      FRH 1660
      SUM1=SUM1+X(JL)*X(JL+KK1)                                         FRH 1670
      SUM2=SUM2+X(N+JL)*X(NM+JL)                                        FRH 1680
      SUM3=SUM3+X(JL)*X(NM+JL)                                          FRH 1690
    6 SUM4=SUM4+X(N+JL)*X(KK1+JL)                                       FRH 1700
      R1(KK)=SUM1/D1                                                    FRH 1710
      R2(KK)=SUM2/D2                                                    FRH 1720
      CI(KK)=SUM3/D3                                                    FRH 1730
    7 CT(KK)=SUM4/D3                                                    FRH 1740
      WRITE(3,1)                                                        FRH 1750
      WRITE(3,3)(R1(KK),R2(KK),CI(KK),CT(KK), KK=1,M)                   FRH 1760
      CALL PARZN2 (R1,R2,CI,CT,NG,M,IGO)                                FRH 1770
      RETURN                                                            FRH 1780
      END                                                               FRH 1790
```

```
C                                                                       FRH 1800
      SUBROUTINE PARZN2 (R1,R2,CI,CT,NQ,M,IGO)                          FRH 1810
C                                                                       FRH 1820
C        THIS PROCEDURE COMPUTES N+1 POINTS OF TWO ESTIMATED SPECTRAL DEFRH 1830
C        FUNCTIONS AND OF THE CO-SPECTRUM AND QUADRATURE SPECTRUM FROM RFRH 1840
C        AND R2(I), WHICH ARE THE AUTO-CORRELATION FUNCTIONS AND RE(I) AFRH 1850
C        WHICH ARE THE EVEN AND ODD PARTS OF THE CROSS-CORRELATION FUNCTFRH 1860
C        TWO TIME SERIES. THE TRUNCATION POINT IS M, THE THE WEIGHTING KFRH 1870
C        USED IS OMEGA(I). THE SINES AND COSINES NEEDED ARE COMPUTED RECFRH 1880
C                                                                       FRH 1890
C     IGO IS AN INDICATOR REFERING TO THE WINDOW DESIRED                FRH 1900
C          IGO=1, THE PARZEN WINDOW IS USED                             FRH 1910
C             =0, THE TUKEY-HANNING WINDOW IS USED                      FRH 1920
C                                                                       FRH 1930
C                                                                       FRH 1940
      DIMENSION R1(1),R2(1),C(500),P(500),AA(500),B(500),CT(1),CI(1)    FRH 1950
      DIMENSION F1(500),F2(500),CO(500),QU(500),OMEGA(500),RE(500),RO(50
     10)
      DATA PI/3.1416/,PIV/0.3183/                                       FRH 1980
    1 FORMAT(4F10.5)                                                    FRH 1990
    2 FORMAT('  RESULTS OF PARZEN TRANSFORM'//'       F1         F2     FRH 2000
     1  CO        QU        COH.      PHASE'/(6F10.5))                  FRH 2010
      DO 15 I=1,500
      F1(I)=0.0
      F2(I)=0.0
      CO(I)=0.0
      QU(I)=0.0
      OMEGA(I)=0.0
      RE(I)=0.0
   15 RO(I)=0.0
      IF(IGO)8,9,8                                                      FRH 2020
    8 CONTINUE                                                          FRH 2030
C                                                                       FRH 2040
C     THE PARZEN WINDOW                                                 FRH 2050
C                                                                       FRH 2060
      PASS=M/2                                                          FRH 2070
      DO 10 K=1,M                                                       FRH 2080
      IF(K.GT.PASS)          GO TO 11                                   FRH 2090
      OMEGA(K)=1.-((6.*(K**2))/(M**2))*(1.-((1.*K)/M))                  FRH 2100
      GO TO 10                                                          FRH 2110
   11 OMEGA(K)=2.*(1.-(1.*K)/M)**3                                      FRH 2120
   10 CONTINUE                                                          FRH 2130
      GO TO 13                                                          FRH 2140
C                                                                       FRH 2150
C     THE TUKEY-HANNING WINDOW                                          FRH 2160
C                                                                       FRH 2170
    9 DO 3 I=1,M                                                        FRH 2180
    3 OMEGA(I)=.5*(1.0+COS((PI*I)/M))                                   FRH 2190
   13 CONTINUE                                                          FRH 2200
C                                                                       FRH 2210
C     CALCULATE THE SUMS AND DIFFERENCES OF THE CROSS-CORRELATIONS      FRH 2220
C                                                                       FRH 2230
      DO 14 I=1,M                                                       FRH 2240
      RE(I)=(CI(I)+CT(I))/2.0                                           FRH 2250
      RO(I)=(CI(I)-CT(I))/2.0                                           FRH 2260
   14 CONTINUE                                                          FRH 2270
      C1=COS(PI/NQ)                                                     FRH 2280
      C3=C1                                                             FRH 2290
```

```
      D1=C1
      C2=SIN(PI/NQ)
      D4=C2
      D6=2.0*C1
      P1=0.5*R1(1)
      P2=0.5*RE(1)
      P4=0.5*R2(1)
      K=M+1
      DO 4 I=2,K
      A=OMEGA(I)
      P1=P1+R1(I)*A
      P2=P2+RE(I)*A
    4 P4=P4+R2(I)*A
      F1(1)=P1*PIV
      F2(1)=P4*PIV
      CO(1)=P2*PIV
      QU(1)=0.0
      DO 6 I=1,NQ
      U11=0.0
      U12=0.0
      U13=0.0
      U14=0.0
      U21=0.0
      U22=0.0
      U23=0.0
      U24=0.0
      DO 5 J=2,K
      JJ=M+3-J
      A=OMEGA(JJ)
      U31=D6*U21-U11+R1(JJ)*A
      U32=D6*U22-U12+RE(JJ)*A
      U33=D6*U23-U13+RO(JJ)*A
      U34=D6*U24-U14+R2(JJ)*A
      U11=U21
      U21=U31
      U12=U22
      U22=U32
      U13=U23
      U23=U33
      U14=U24
      U24=U34
    5 CONTINUE
      F1(I+1)=(D1*U21-U11+R1(1)*0.5)*PIV
      F2(I+1)=(D1*U24-U14+R2(1)*0.5)*PIV
      CO(I+1)=(D1*U22-U12+RE(1)*0.5)*PIV
      QU(I+1)=D4*U23*PIV
      D1=(C1*C3)-(C2*D4)
      D4=(D4*C1)+(C3*C2)
      C3=D1
      D6=2.0*D1
    6 CONTINUE
C     CALCULATE THE COHERENCE AND PHASE DIAGRAM
      DO 7 I=1,M
      C(I)=(CO(I)**2+QU(I)**2)/(F1(I)*F2(I))
    7 P(I)=ATAN(QU(I)/CO(I))
C
C     WRITE BOTH SPECTRAL POWER FUNCTIONS,THE CO-SPECTRUM FUNCTION,
C     THE QUADURATURE FUNCTION,THE COHERENCE AND THE PHASE DIAGRAM.
```

```
C
      WRITE(3,2)(F1(I),F2(I),CO(I),QU(I),C(I),P(I),I=1,M)
C
C
C     PLOT BOTH POWER SPECTRA,THE COHERENCE, AND PHASE DIAGRAM
C
      CALL SPLOT(NQ,F1,3)
      CALL SPLOT(NQ,F2,3)
      DO 12 I=1,M
      AA(I)=I
      AA(I+M+1)=C(I)
      B(I)=I
   12 B(I+M)=P(I)
C     A DUMMY COHERENCE VALUE IS ADDED TO THE COHERENCE VECTOR
      AA(M+1)=M+1
      AA(M+M+2)=1.000
      NU=M+1
      CALL PLOT(1,AA,NU,2,NU,0)
      CALL PLOT(2,B,M,2,M,0)
      RETURN
      END
```

```
C
      SUBROUTINE SPLOT(N,X,NC)
C
C        THIS ROUTINE IS USED TO MODIFY THE RESULTS FROM THE PARZEN SUB
C        SO THAT THEY CAN BE GRAPHED BY SUBROUTINE PLOT.
C
      DIMENSION X(400),OUT(101),YPR(11),ANG(9),PLT(400),Y(400)
    2 FORMAT( 47X,F6.0,9X,F4.0)
  101 FORMAT(I4,3X,F10.5,5X,I4,3X,F10.5,5X,I4,3X,F10.5)
  102 FORMAT(' PLOT VALUES '/)
C
C     INITIALIZE VALUES
      PI=3.1415927
      K=0
      M=2
      WRITE(3,102)
C
      DO 10 I=1,N
      PLT(I)=0.
   10 Y(I)=0.
      DO 106 I=1,N
      IF(X(I))100,100,99
   99 X(I)=ALOG(X(I))
      GO TO 106
  100 X(I)=3.9999
  106 CONTINUE
      NCOL=N/3
      DO 6 I=1,NCOL
      II=I+NCOL
      III=I+(2*NCOL)
    6 WRITE(3,101) I,X(I),II,X(II),III,X(III)
      DO 8 I=1,N
      PLT(I)=I*PI/N
      PLT(I+N)=X(I)
    8 CONTINUE
      CALL PLOT(NC,PLT,N,M,N,0)
      RETURN
      END
```

```
C
      SUBROUTINE PLOT(NO,A,N,M,NL,NS)
C
      DIMENSION OUT(101),YPR(11),ANG(9),A(1)
C
    1 FORMAT(1H1,60X,7H CHART ,I3,//)
    2 FORMAT(1H ,F11.4,5H+    ,101A1)
    3 FORMAT(1H )
    4 FORMAT(10H *+0156789)
    5 FORMAT( 10A1)
    7 FORMAT(1H ,16X,101H.         .         .         .         .
     1    .         .         .         .         .         .)
    8 FORMAT(1H0,9X,11F10.4//)
    9 FORMAT(1H ,16X,101A1)
  200 FORMAT( 10X,'    PLOT OF COHERENCE')
  201 FORMAT( 10X,'    PLOT OF PHASE ')
  202 FORMAT( 10X,'      PLOT OF SPECTRUM')
C
C     ....................................................................
C
      NLL=NL
C
      IF(NS)16,16,10
C
C        SORT BASE VARIABLE IN ASCENDING ORDER
C
   10 DO 15 I=1,N
      DO 14 J=1,N
      IF(A(I)-A(J))14,14,11
   11 L=I-N
      LL=J-N
      DO 12 K=1,M
      L=L+N
      LL=LL+N
      F=A(L)
      A(L)=A(LL)
   12 A(LL)=F
   14 CONTINUE
   15 CONTINUE
C
C        TEST NLL
C
   16 IF(NLL)20,18,20
   18 NLL=50
C
C        PRINT TITLE
C
   20 WRITE(3,1)NO
      GO TO (91,92,93),NO
   91 WRITE(3,200)
      GO TO 21
   92 WRITE(3,201)
      GO TO 21
   93 WRITE(3,202)
   21 CONTINUE
C
C        DEVELOP BLANKS AND DIGITS FOR PRINTING
C
```

```
      REWIND 10                                                      FRH 4010
      WRITE(10,4)                                                    FRH 402
      REWIND 10                                                      FRH 4030
      READ (10,5)BLANK,(ANG(I),I=1,9)                                FRH 403
      REWIND 10                                                      FRH 4032
C
C        FIND SCALE FOR BASE VARIABLE
C
      XSCAL=(A(N)-A(1))/(FLOAT(NLL-1))
C
C        FIND SCALE FOR CROSS VARIABLES
C
      M1=N+1
      YMIN=A(M1)
      YMAX=YMIN
      M2=M*N
      DO 40 J=M1,M2
      IF(A(J)-YMIN)28,28,26
   26 IF(A(J)-YMAX)40,40,30
   28 YMIN=A(J)
      GO TO 40
   30 YMAX=A(J)
   40 CONTINUE
      YSCAL=(YMAX-YMIN)/100.0
C
C        FIND BASE VARIABLE PRINT POSITION
C
      XB=A(1)
      L=1
      MY=M-1
      I=1
   45 F=I-1
      XPR=XB+F*XSCAL
      IF(A(L)-XPR)51,51,70
C
C        FIND CROSS VARIABLES
C
   51 DO 55 IX=1,101
   55 OUT(IX)=BLANK
   57 DO 60 J=1,MY
      LL=L+J*N
      JP=((A(LL)-YMIN)/YSCAL)+1.0
      OUT(JP)=ANG(J)
   60 CONTINUE
C
C        PRINT LINE AND CLEAR, OR SKIP
C
      WRITE(3,2)XPR,(OUT(IZ),IZ=1,101)
      L=L+1
      GO TO 80
   70 WRITE(3,3)
   80 I=I+1
      IF(I-NLL)45,84,86
   84 XPR=A(N)
      GO TO 51
C
C        PRINT CROSS VARIABLES NUMBERS
C
```

```
   86 WRITE(3,7)
      YPR(1)=YMIN
      DO 90 KN=1,9
   90 YPR(KN+1)=YPR(KN)+YSCAL*10.0
      YPR(11)=YMAX
      WRITE(3,8)(YPR(IR),IR=1,11)
      RETURN
      END
```

Sample of Cross-Spectral Output

```
          ****DATA SET 2**** TIME SERIES (    100, 20100),( 10100, 20100)
  THE INPUT
 VARIABLE  1
 2.0  3.0  0.0  0.0  2.0  3.0  2.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  1.0
 1.0  1.0  1.0  1.0  3.0  4.0  1.0  2.0  2.0  2.0  3.0  3.0  4.0  6.0  6.0
 5.0  5.0  5.0  5.0  5.0  5.0  5.0  5.0  6.0  7.0  6.0  6.0  4.0  3.0  4.0
 2.0  0.0  0.0  2.0  1.0  0.0  1.0  0.0  0.0  5.0  8.0  8.0  7.0  6.0  6.0
 6.0  6.0  5.0  1.0  0.0  1.0  1.0  0.0  0.0  1.0  1.0  0.0  0.0  0.0  0.0
 1.0  1.0  0.0  0.0  0.0  0.0  2.0  2.0  0.0  0.0  0.0  3.0  3.0  2.0  5.0
 5.0  6.0  6.0  7.0  7.0  9.0  8.0  8.0 10.0  9.0  8.0  7.0  7.0  6.0  8.0
 9.0  9.0  5.0  7.0  6.0  7.0  7.0  6.0  6.0  4.0  4.0  3.0  3.0  5.0  5.0
 5.0  4.0  3.0  5.0  6.0  6.0  6.0  6.0  4.0  3.0  3.0  6.0  6.0  4.0  5.0
 4.0  4.0  5.0  2.0  3.0  3.0  4.0  4.0  4.0  5.0  6.0  7.0  8.0 10.0 10.0
11.0 13.0 11.0 10.0 10.0  9.0 11.0 11.0 12.0 10.0 10.0 12.0 12.0 12.0 13.0
15.0 12.0 10.0  9.0 10.0  9.0  9.0  9.0  9.0 10.0 10.0 10.0 13.0 13.0 11.0
 9.0  9.0  8.0  9.0  9.0  9.0 10.0  9.0  8.0  8.0  8.0  9.0  7.0  7.0  7.0
 6.0  5.0  5.0  7.0  6.0  6.0
 VARIABLE  2
 8.0  7.0  7.0  6.0  8.0  9.0  9.0  5.0  7.0  6.0  7.0  7.0  6.0  6.0  4.0
 4.0  3.0  3.0  5.0  5.0  5.0  4.0  3.0  5.0  6.0  6.0  6.0  6.0  4.0  3.0


 3.0  6.0  6.0  4.0  5.0  4.0  4.0  5.0  2.0  3.0  3.0  4.0  4.0  4.0  5.0
 6.0  7.0  8.0 10.0 10.0 11.0 13.0 11.0 10.0 10.0  9.0 11.0 11.0 12.0 10.0
10.0 12.0 12.0 12.0 13.0 15.0 12.0 10.0  9.0 10.0  9.0  9.0  9.0  9.0 10.0
10.0 10.0 13.0 13.0 11.0  9.0  9.0  8.0  9.0  9.0  9.0 10.0  9.0  8.0  8.0
 8.0  9.0  7.0  7.0  7.0  6.0  5.0  5.0  7.0  6.0  6.0  7.0  8.0  8.0  7.0
 5.0  4.0  5.0  3.0  3.0  3.0  3.0  4.0  6.0  6.0 10.0 11.0 12.0 12.0 11.0
 8.0  8.0  7.0  6.0  6.0  7.0  7.0  8.0  4.0  4.0  1.0  3.0  3.0  4.0  5.0
 5.0  4.0  4.0  4.0  3.0  2.0  3.0  4.0  3.0  3.0  4.0  2.0  2.0  2.0  3.0
 4.0  3.0  2.0  1.0  1.0  3.0  1.0  3.0  3.0  3.0  5.0  6.0  6.0  6.0  6.0
 5.0  4.0  5.0  3.0  2.0  0.0  0.0  0.0  3.0  4.0  2.0  3.0  3.0  2.0  3.0
 3.0  5.0  4.0  3.0  3.0  3.0  3.0  3.0  2.0  4.0  4.0  6.0  4.0  0.0  2.0
 2.0  3.0  3.0  3.0  3.0  3.0
```

XTOT	YTOT	ZTOT	XZ	YZ	X2	Y2	Z2
1052.0	1175.0	20301.0	135265.0	101699.0	8166.0	8887.0	2727101.0

LEAST SQUARE REGRESSION 1 Z= 0.922+ 0.043X

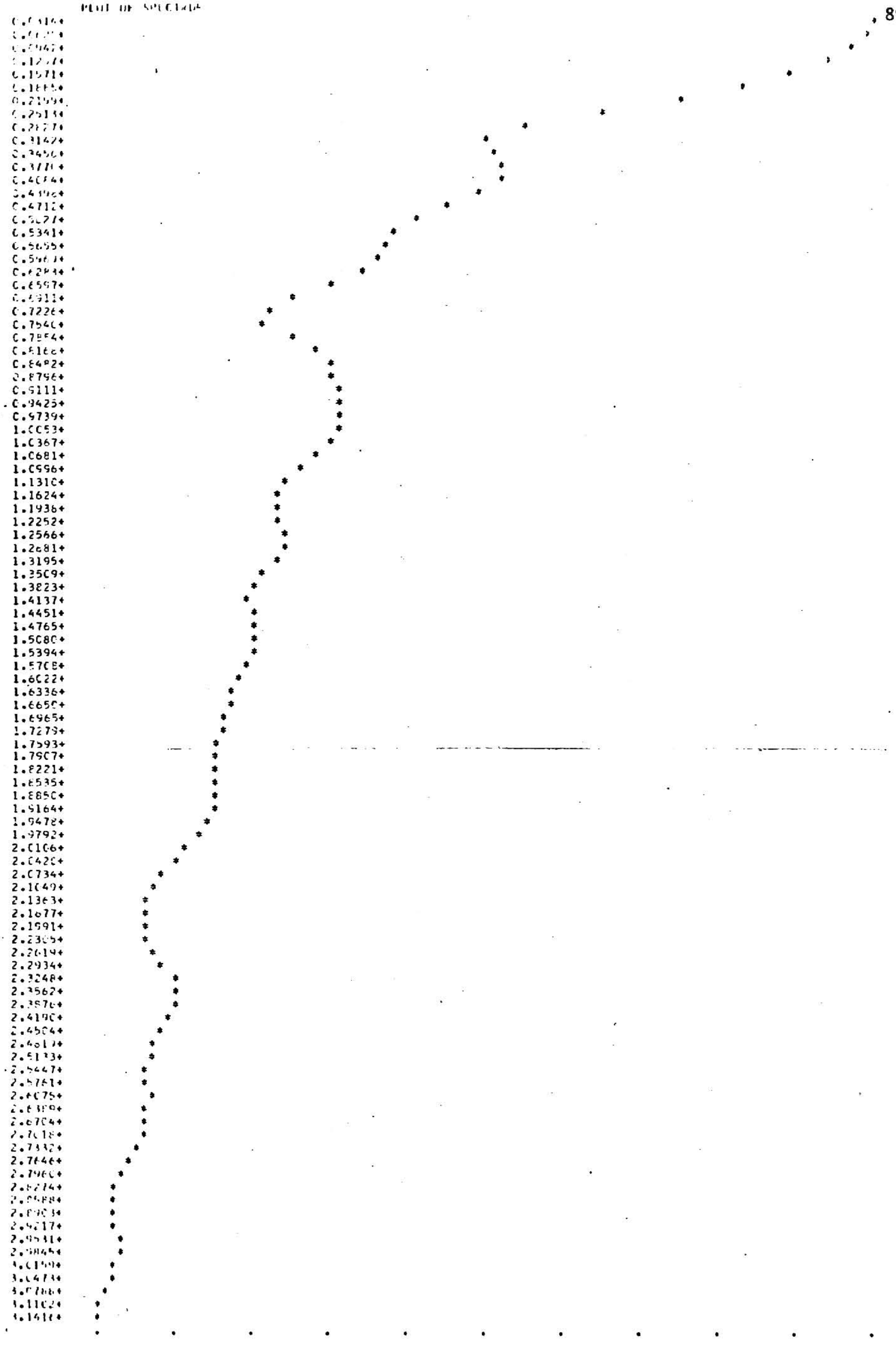
LEAST SQUARE REGRESSION 2 Z= 8.551+ -0.026X

THE AUTO-CORRELATION FUNCTIONS ,R1(),R2() AND
THE CROSS-CORRELATION FUNCTIONS,CI(),CT()

R1	R2	CI	CT
1.00000	1.00000	-0.73743	-0.73743
0.97155	0.94159	-0.73750	-0.73333
0.94183	0.89019	-0.72974	-0.72826
0.92324	0.83941	-0.71924	-0.71654
0.90640	0.79812	-0.71073	-0.70350
0.89906	0.76529	-0.70637	-0.70027
0.86822	0.73410	-0.69884	-0.69475
0.84771	0.70643	-0.68356	-0.68131
0.82514	0.68804	-0.67090	-0.67374
0.80367	0.66913	-0.65767	-0.67446
0.78856	0.65425	-0.65116	-0.67295
0.77630	0.63885	-0.64044	-0.67853
0.76674	0.63779	-0.62927	-0.67589
0.75972	0.63425	-0.61781	-0.67481
0.74698	0.62545	-0.60716	-0.66731
0.72963	0.60562	-0.59567	-0.66571
0.71276	0.59197	-0.57688	-0.66234
0.69462	0.56802	-0.55719	-0.65366
0.67728	0.54323	-0.54754	-0.65185
0.66866	0.52123	-0.53851	-0.64473
0.65781	0.49934	-0.52305	-0.62881
0.64472	0.48518	-0.50611	-0.61591
0.62497	0.47081	-0.49212	-0.59724
0.60711	0.46202	-0.47634	-0.57496
0.59960	0.45745	-0.47521	-0.54759
0.59192	0.43957	-0.46744	-0.52951
0.58582	0.42995	-0.45812	-0.51426
0.57703	0.40475	-0.46039	-0.48681
0.56681	0.37770	-0.46222	-0.45483
0.55816	0.35354	-0.45850	-0.42315

RESULTS OF PARZEN TRANSFORM

F1	F2	CO	QU	COH.	PHASE
3.70475	3.21174	-3.00337	0.0	0.75808	0.0
3.53647	3.07493	-2.85992	0.02767	0.75222	-0.00968
3.07115	2.69582	-2.46370	0.04737	0.73341	-0.01922
2.41384	2.15781	-1.90531	0.05379	0.69752	-0.02822
1.70125	1.57010	-1.30272	0.04600	0.63612	-0.03530
1.06059	1.03538	-0.76543	0.02746	0.53422	-0.03585
0.57612	0.62338	-0.36570	0.00448	0.37243	-0.01226
0.27423	0.35848	-0.12539	-0.01608	0.16257	0.12757
0.13018	0.22401	-0.02184	-0.02921	0.04561	0.92882
0.08975	0.17864	-0.00670	-0.03302	0.07080	1.37064
0.09522	0.17633	-0.02867	-0.02882	0.09841	0.78798
0.10483	0.18199	-0.05058	-0.02002	0.15512	0.37686
0.10012	0.17802	-0.05614	-0.01051	0.18305	0.18505
0.08186	0.16218	-0.04645	-0.00320	0.16332	0.06888
0.05989	0.14036	-0.03105	0.00069	0.11472	-0.02221
0.04321	0.11919	-0.01896	0.00159	0.07029	-0.08384
0.03481	0.10190	-0.01385	0.00087	0.05430	-0.06265
0.03203	0.08798	-0.01396	-0.00003	0.06919	0.00244
0.03013	0.07546	-0.01516	-0.00021	0.10111	0.01378
0.02601	0.06324	-0.01429	0.00051	0.12434	-0.03544
0.01973	0.05204	-0.01083	0.00172	0.11706	-0.15726
0.01360	0.04360	-0.00636	0.00284	0.08181	-0.42044
0.01006	0.03920	-0.00294	0.00346	0.05217	-0.86624
0.01005	0.03861	-0.00163	0.00344	0.03730	-1.12935
0.01262	0.04009	-0.00207	0.00293	0.02545	-0.95576
0.01594	0.04144	-0.00315	0.00216	0.02224	-0.60534
0.01849	0.04112	-0.00393	0.00139	0.02283	-0.34138
0.01979	0.03888	-0.00420	0.00067	0.02346	-0.15823
0.02024	0.03551	-0.00439	0.00001	0.02676	-0.00286
0.02040	0.03212	-0.00499	-0.00059	0.03852	0.11730


PLOT OF SPECTRA

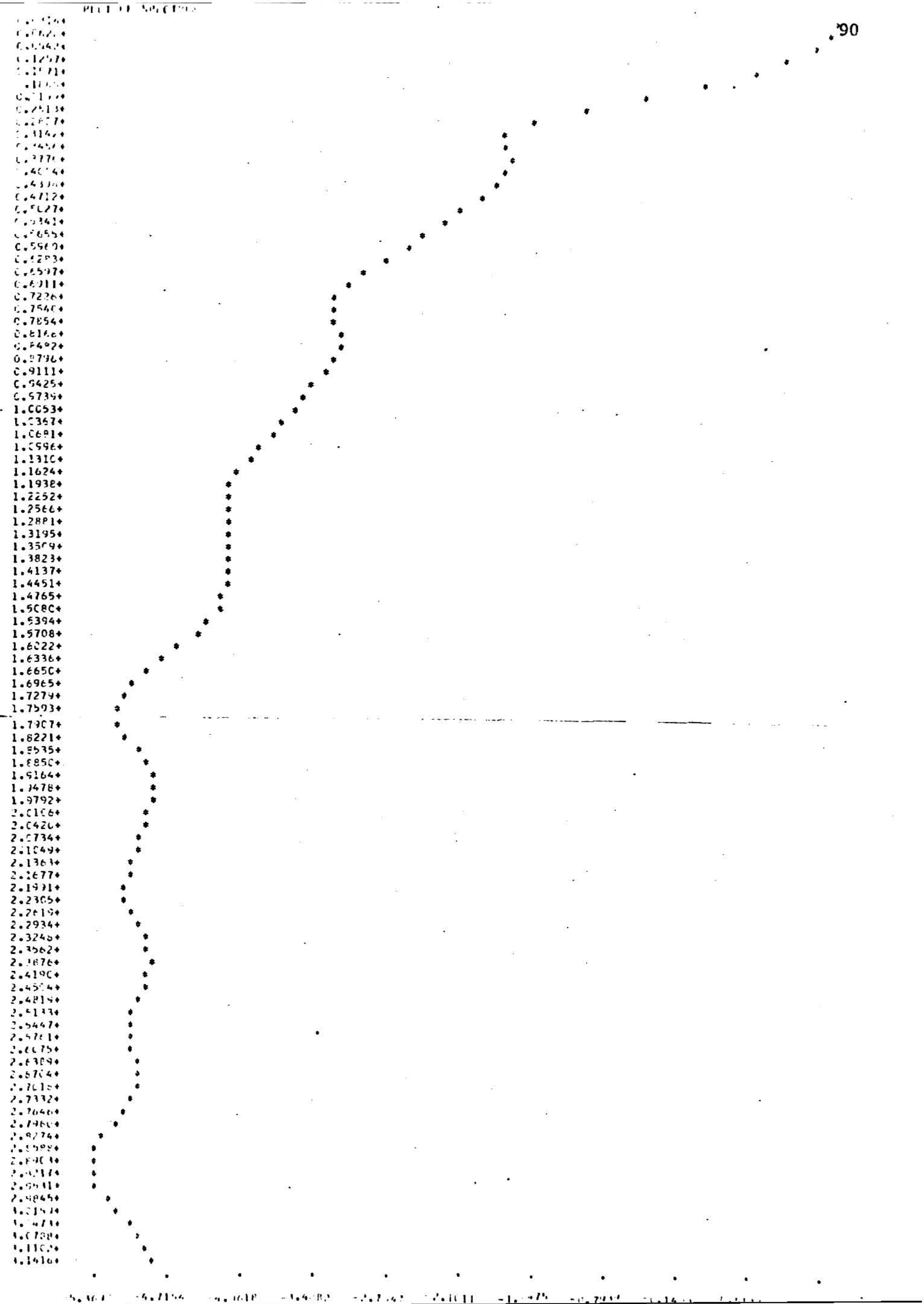
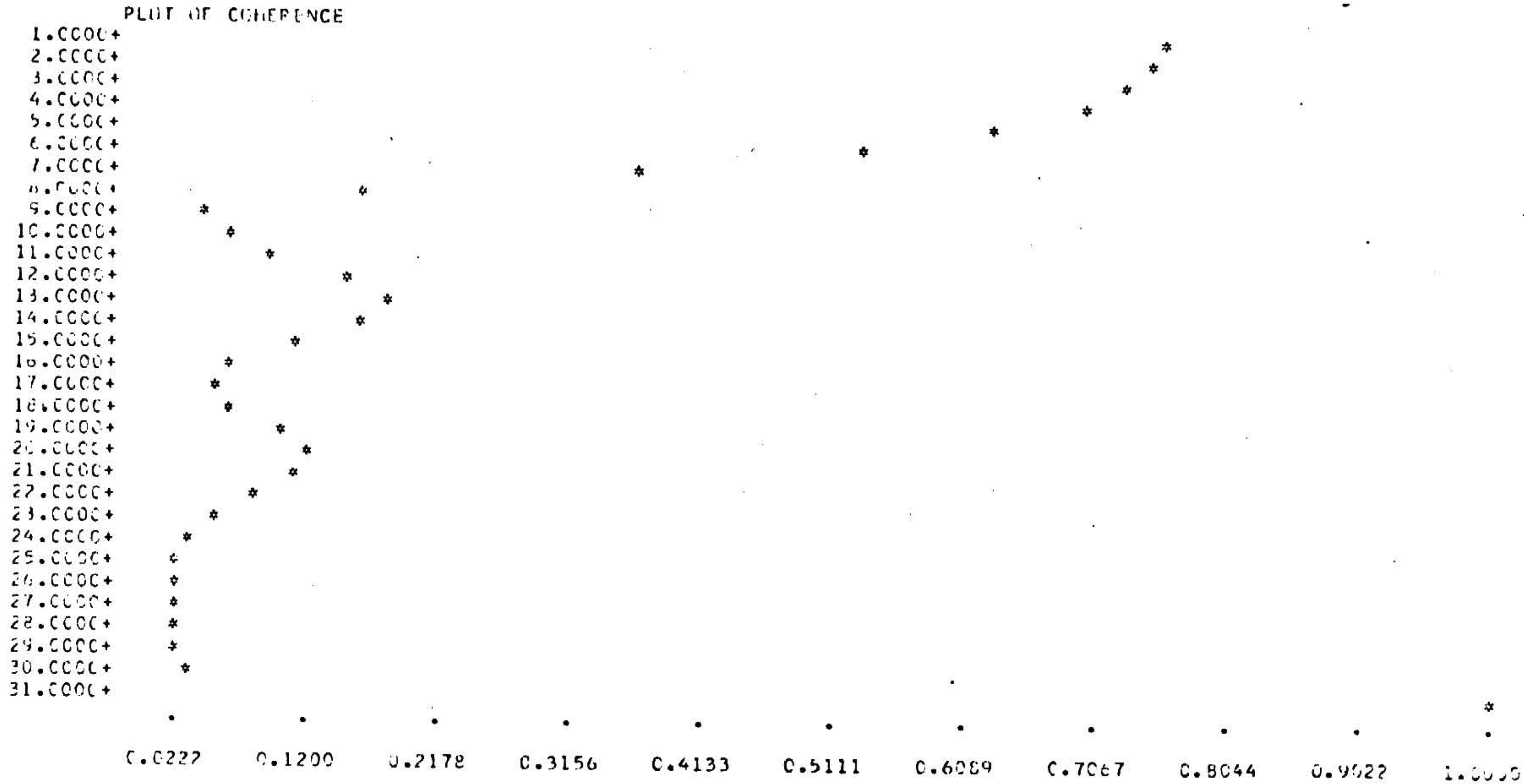


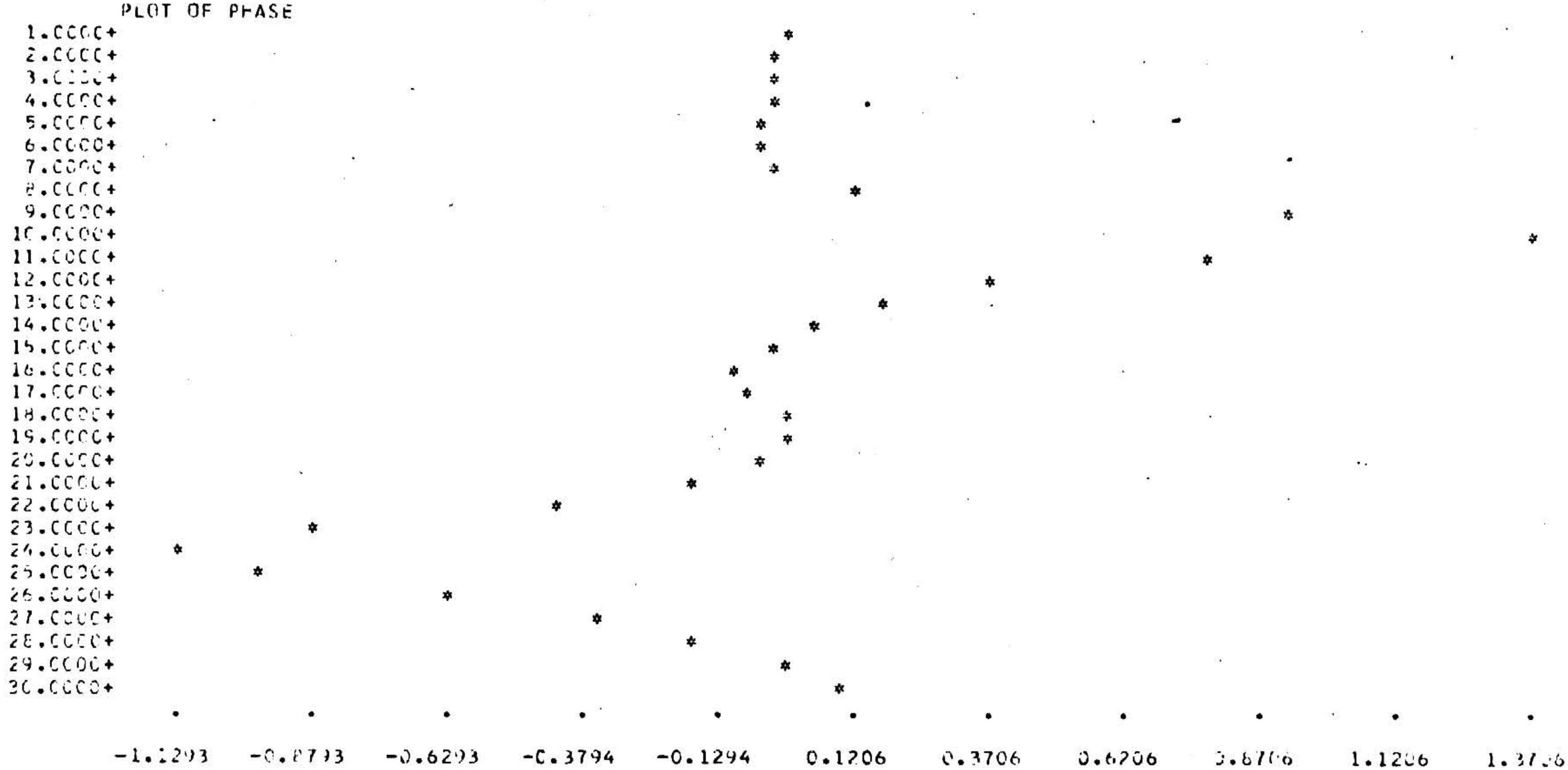
CHART 1


PLOT OF COHERENCE
1.0000+
2.0000+
3.0000+
4.0000+
5.0000+
6.0000+
7.0000+
9.0000+
10.0000+
11.0000+
12.0000+
13.0000+
14.0000+
15.0000+
16.0000+
17.0000+
18.0000+
19.0000+
21.0000+
22.0000+
23.0000+
24.0000+
25.0000+
26.0000+
27.0000+
28.0000+
29.0000+
30.0000+
31.0000+
0.0222
0.1200
0.2178
0.3156
0.4133
0.5111
0.6089
0.7067
0.8044
0.9022

91

CHART 2

PLOT OF PHASE



92

AVERAGE COHERENCE

The average coherence was found by

$$\frac{1}{M}\sum_{i=1}^{M} C_i \ .$$

For example, using the sample output, average coherence, x,

$x = (1/30)(.7580 + .7522 + \ldots + .0385)$

$= .1944$

APPENDIX C:

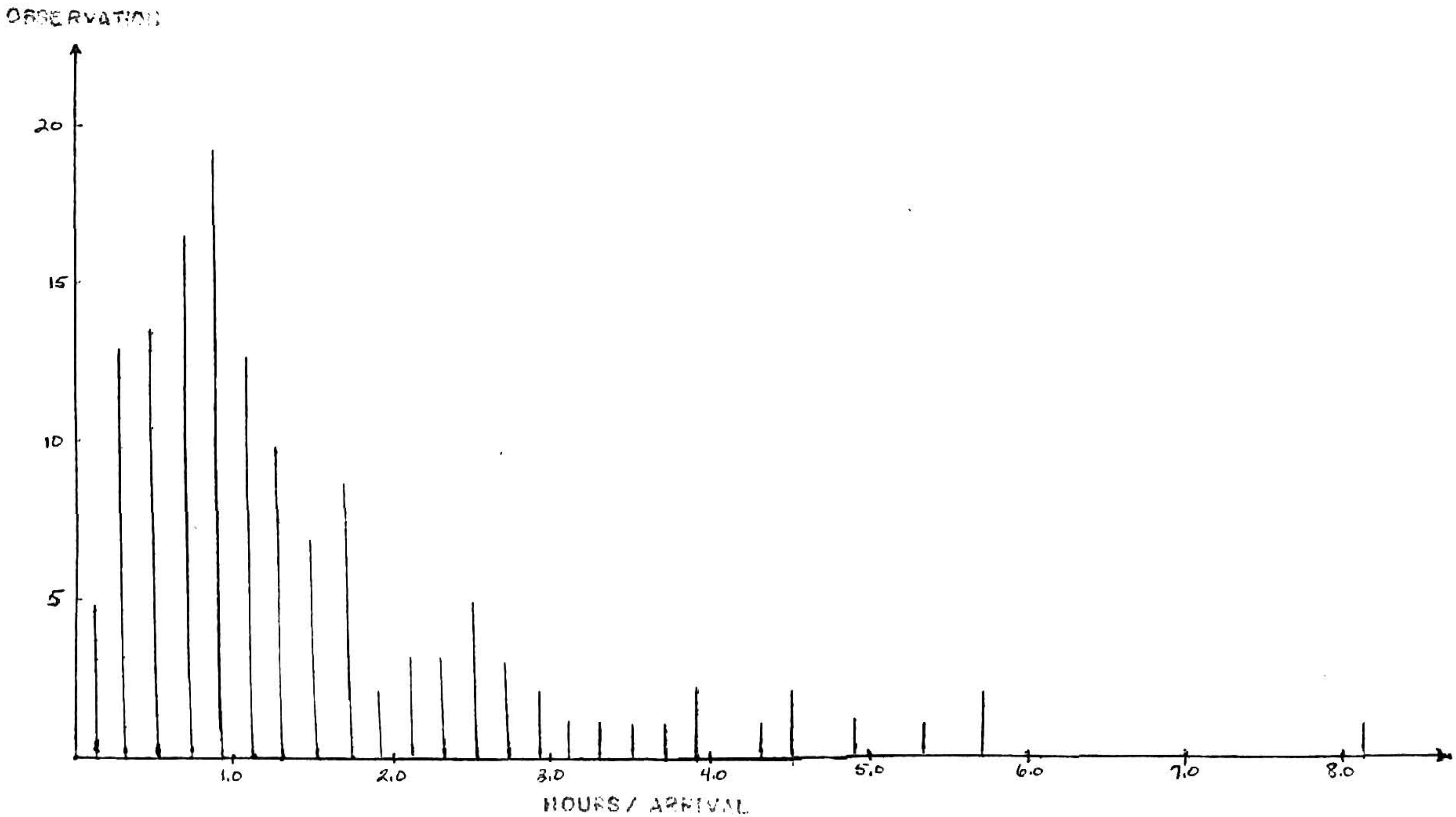
1) Histograph of Inter-arrival Distribution
   Figure 1.1

2) GASP Simulation Program

3) General Logic Diagram of GASP Simulation

4) Autocorrelation functions of successive segments of the Number of Units in the System Statistic
   Figures 1.3.1, 1.3.2, 1.3.3, 1.3.4, 1.3.5

5) Average Number of Units in the System Statistic
   Figure 1.4



Histograph of Inter-arrival Distribution

Figure 1.1

GASP Simulation Program

Shown here are the seven subroutines written for the simulation of this paper. The subroutines are:

MAIN - used to initiate the simulation

EVENTS - used to determine which event is to be called

ARRVL - used to simulate the arrival of an item into the system

SERVE - used to begin the service of an item

DONE - used to complete service of an item and remove it from the system

REPORT - used to observe the various statistics at regular intervals

OUTPUT - a special subroutine which punched the output in a form compatable with the spectral program

The five standard GASP subroutines were used with this simulation. They have not been shown and may be found in GASP II: A FORTRAN Based Simulation Language by A.B. Pritsker and P.J. Kiviat.

```
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,
     1NCOLCT,NHISTC,NOQ,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,
     2OUT,SCALE,NSEED,TNOW,TSTART,TSTOP,MXX
      COMMON ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),
     1MAXNQ(4),MFE(4),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),
     2QTIME(4),SSUMA(10,5),SUMA(10,5),IX(8)
      COMMON XSR,IKONT,XQUE,XSYS,II,ISERVE,P,EX,DELTA,NOBS,NREP,TLAST
      COMMON UTIL(500),JQUE(500),JSYS(500),XX(500),TIME(500)
      DIMENSION NSET(6,3000)
    1 FORMAT(F5.2,2I5,2F10.4)
    2 FORMAT(F5.2,'-DELTA',I5,'-NOBS',I5,'-ISERVE',F10.4,'-P',F10.4,'-EX
     1')
    3 FORMAT(3F10.4)
    4 FORMAT(F6.1,'-XSR',F10.3,'-XQUE',F10.3,'-XSYS')
      READ(1,1) DELTA,NOBS,ISERVE,P,EX
      WRITE(3,2) DELTA,NOBS,ISERVE,P,EX
      READ(1,3) XSR,XQUE,XSYS
      WRITE(3,4) XSR,XQUE,XSYS
      IKONT=0
      II=0
      NREP=0
      TLAST=0.0
      CALL CASP(NSET)
      STOP
      END
```

```
      SUBROUTINE EVENTS(K,NSET)
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,
     1NCOLCT,NHISTO,NOQ,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,
     2OUT,SCALE,ISEED,TNOW,TSTART,TSTOP,MXX
      COMMON ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),
     1MAXNQ(4),MFE(4),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),
     2QTIME(4),SSUMA(10,5),SUMA(10,5),IX(8)
      COMMON XSR,IKONT,XQUE,XSYS,II,ISERVE,P,EX,DELTA,NOBS,NREP,TLAST
      COMMON UTIL(500),JQUE(500),JSYS(500),XX(500),TIME(500)
      DIMENSION NSET(6,1)
      GO TO (1,2,3,100),K
    1 CALL ARRVL(NSET)
      RETURN
    2 CALL DONE(NSET)
      RETURN
    3 CALL REPORT(NSET)
      RETURN
  100 CALL MONTR(NSET)
      RETURN
      END
```

```
      SUBROUTINE ARRVL(NSET)
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,
     1NCOLCT,NHISTO,NOQ,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,
     2OUT,SCALE,NSEED,TNOW,TSTART,TSTOP,MXX
      COMMON ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),
     1MAXNQ(4),MFE(4),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),
     2QTIME(4),SSUMA(10,5),SUMA(10,5),IX(8)
      COMMON XSR,IKONT,XQUE,XSYS,II,ISERVE,P,EX,DELTA,NOBS,NREP,TLAST
      COMMON UTIL(500),JQUE(500),JSYS(500),XX(500),TIME(500)
      DIMENSION NSET(6,1)
   11 FORMAT(' ERROR  -  XSYS LESS THAN ZERO *****')
      XSYS=NQ(2)+XSR
      IF(XSYS.GE.0.0)     GO TO 10
      WRITE(3,11)
      CALL ERROR(1,NSET)
   10 XQUE=NQ(2)
      CALL TMSTAT(XQUE,TNOW,2,NSET)
      CALL TMSTAT(XSYS,TNOW,3,NSET)
      CALL FILEM(2,NSET)
      ATTRIB(1)=TNOW+EXPO(P)
      AT=ATTRIB(1)-TNOW
      ATTRIB(3)=AT
      ATTRIB(2)=1.0
      ATTRIB(4)=ATTRIB(1)
      CALL FILEM(1,NSET)
      CALL HISTOG(AT,1.0,2.0,1)
      CALL COLECT(AT,1,NSET)
      IF(XSR.EQ.ISERVE)                 RETURN
      CALL SERVE(NSET)
      RETURN
      END
```

```
      SUBROUTINE SERVE(NSET)
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,
     1NCOLCT,NHISTO,NOQ,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,
     2OUT,SCALE,NSEED,TNOW,TSTART,TSTOP,MXX
      COMMON ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),
     1MAXNQ(4),MFE(4),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),
     2QTIME(4),SSUMA(10,5),SUMA(10,5),IX(8)
      COMMON XSR,IKONT,XQUE,XSYS,II,ISERVE,P,EX,DELTA,NOBS,NPEP,TLAST
      COMMON UTIL(500),JQUE(500),JSYS(500),XX(500),TIME(500)
      DIMENSION NSET(6,1)
      XQUE=NQ(2)
      CALL TMSTAT(XSR,TNOW,1,NSET)
      CALL TMSTAT(XQUE,TNOW,2,NSET)
      CALL TMSTAT(XSYS,TNOW,3,NSET)
      XSR=XSR+1
      MFE2=MFE(2)
      CALL REMOVE(MFE2,2,NSET)
      ST=EXPO(EX)
      CALL HISTOG(ST,1.0,2.0,2)
      CALL COLECT(ST,2,NSET)
      ATTRIB(1)=TNOW+ST
      ATTRIB(2)=2.0
      CALL FILEM(1,NSET)
      TSYS=ATTRIB(1)-ATTRIB(4)
      CALL HISTOG(TSYS,0.0,3.0,4)
      CALL COLECT(TSYS,4,NSET)
      TQUE=TNOW-ATTRIB(4)
      CALL HISTOG(TQUE,0.0,3.0,3)
      CALL COLECT(TQUE,3,NSET)
      RETURN
      END
```

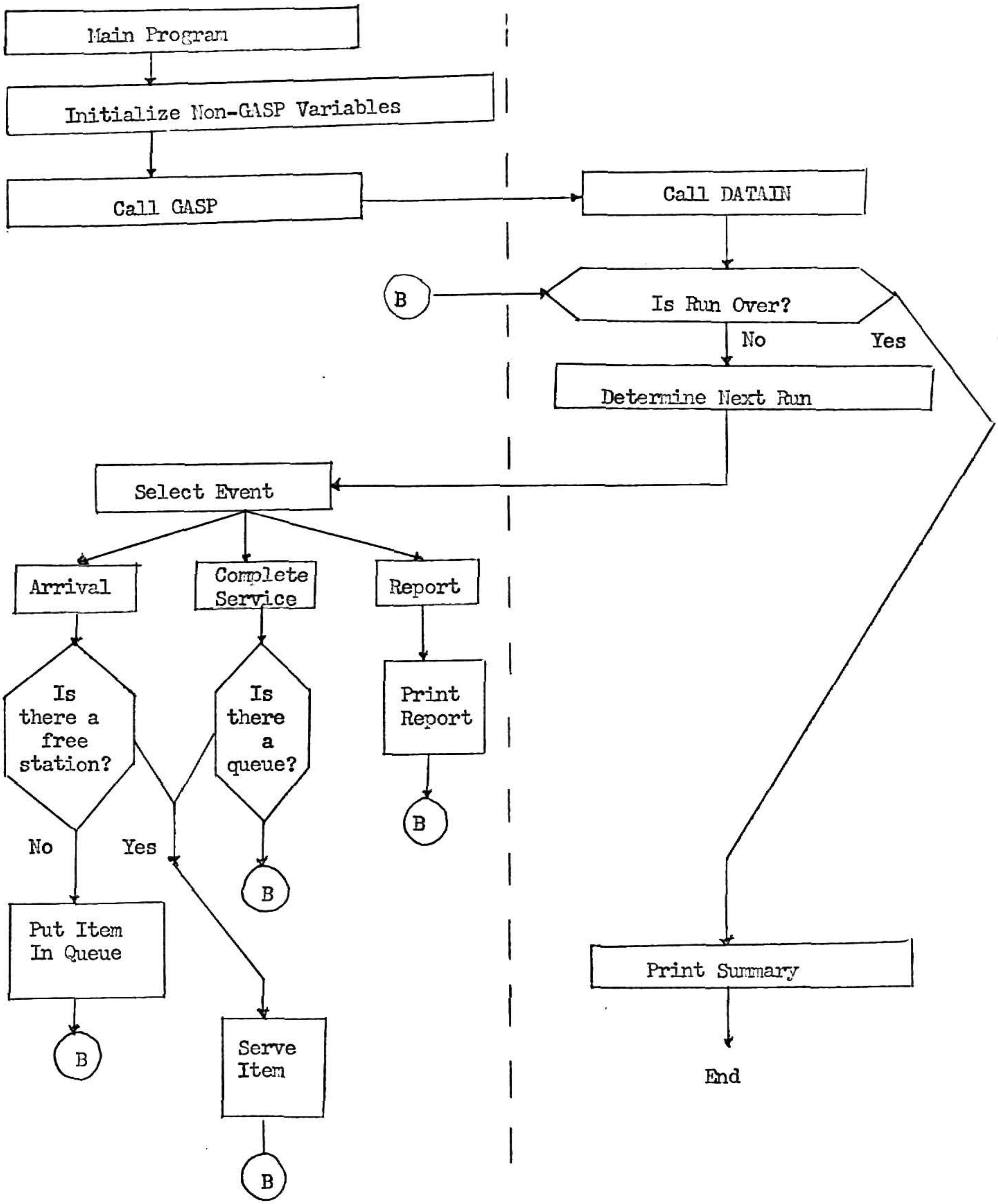
```
      SUBROUTINE DONE(NSET)
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,
     1NCOLCT,NHISTO,NOQ,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,
     2OUT,SCALE,NSEED,TNOW,TSTART,TSTOP,MXX
      COMMON ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),
     1MAXNQ(4),MFE(4),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),
     2QTIME(4),SSUMA(10,5),SUMA(10,5),IX(8)
      COMMON XSR,IKONT,XQUE,XSYS,II,ISERVE,P,EX,DELTA,NOBS,NREP,TLAST
      COMMON UTIL(500),JQUE(500),JSYS(500),XX(500),TIME(500)
      DIMENSION NSET(6,1)
      CALL TMSTAT(XSR,TNOW,1,NSET)
      CALL TMSTAT(XSYS,TNOW,3,NSET)
      XSR=XSR-1
      XSYS=NQ(2)+XSR
C     RECORD TIME SINCE LAST DEPARTURE
      DEP=TNOW-TLAST
      TLAST=TNOW
      CALL COLECT(DEP,5,NSET)
      IF(NQ(2))1,1,2
    2 CALL SERVE(NSET)
    1 RETURN
      END
```

```
      SUBROUTINE REPORT(NSET)
      COMMON ID,IM,INIT,JEVENT,JMONIT,MFA,MSTOP,MX,MXC,
     1NCOLCT,NHISTO,NOQ,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,
     2OUT,SCALE,NSEED,TNOW,TSTART,TSTOP,MXX
      COMMON ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),
     1MAXNQ(4),MFE(4),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),
     2QTIME(4),SSUMA(10,5),SUMA(10,5),IX(8)
      COMMON XSR,IKONT,XQUE,XSYS,II,ISERVE,P,EX,DELTA,NOBS,NREP,TLAST
      COMMON UTIL(500),JQUE(500),JSYS(500),XX(500),TIME(500)
      DIMENSION AR(500),DEP(500),QUET(500),SYST(500),SER(500)
      DIMENSION NSET(6,1)
    6 FORMAT('-',14X,'UTILIZATION',32X,'QUEUE',27X,'NO. UNITS IN SYSTEM'
     1/6X,'MEAN',6X,'STD.DEV.',6X,'MAX',10X,'MEAN',5X,' STD.DEV.',6X,'MA
     2X',10X,'MEAN',6X,'STD.DEV.',6X,'MAX')
    7 FORMAT(4X,F8.6,4X,F8.6,4X,F8.6,4X,F8.3,4X,F8.3,4X,F8.3,3X,F9.3,3X,
     1F9.3,3X,F9.3)
    8 FORMAT('1',10X,'  SYSTEM STATISTICS',5X,I5,' OBSERVATIONS AT SIMUL
     1ATION TIME',F10.2)
   11 FORMAT('-',40X,'*************************')
   12 FORMAT(8X,'TIME',8X,'UTIL',7X,'QUEUE',7X,'SYSTEM',3X,'AVE ARRIVAL-
     1AVE DEPART','  QUEUE TIME SYSTEM TIME SERVICE TIME')
   13 FORMAT(2F12.4,2I12,5F12.3)
      IKONT=IKONT+1
      ATTRIB(1)=TNOW+DELTA
      ATTRIB(2)=3.0
      CALL FILEM(1,NSET)
      II=II+1
      SERVE=ISERVE
      UTIL(II)=XSR/SERVE
      JQUE(II)=NQ(2)
      JSYS(II)=NQ(2)+XSR
      TIME(II)=TNOW
      AR(II)=SUMA(1,1)/SUMA(1,3)
      IF(SUMA(5,3).GT.0)          GO TO 1
      DEP(II)=0.0
      QUET(II)=0.0
      SYST(II)=0.0
      SER(II)=0.0
      RETURN
    1 DEP(II)=SUMA(5,1)/SUMA(5,3)
      QUET(II)=SUMA(3,1)/SUMA(3,3)
      SYST(II)=SUMA(4,1)/SUMA(4,3)
      SER(II)=SUMA(2,1)/SUMA(2,3)
      IF(IKONT.LT.NOBS)                  RETURN
      CALL OUTPUT(NSET)
      IKONT=0
      NREP=NREP+1
C     COMPUTING MEAN, STD.DEV., AND MAX
      DO 5 I=1,II
    5 XX(I)=UTIL(I)
      DO 4 J=1,3
      FMAX=XX(1)
      DO 14 I=2,II
   14 IF(FMAX.LT.XX(I))       FMAX=XX(I)
      XS=0.0
      XSS=0.0
      DO 21 I=1,II
      XS=XS+XX(I)
```

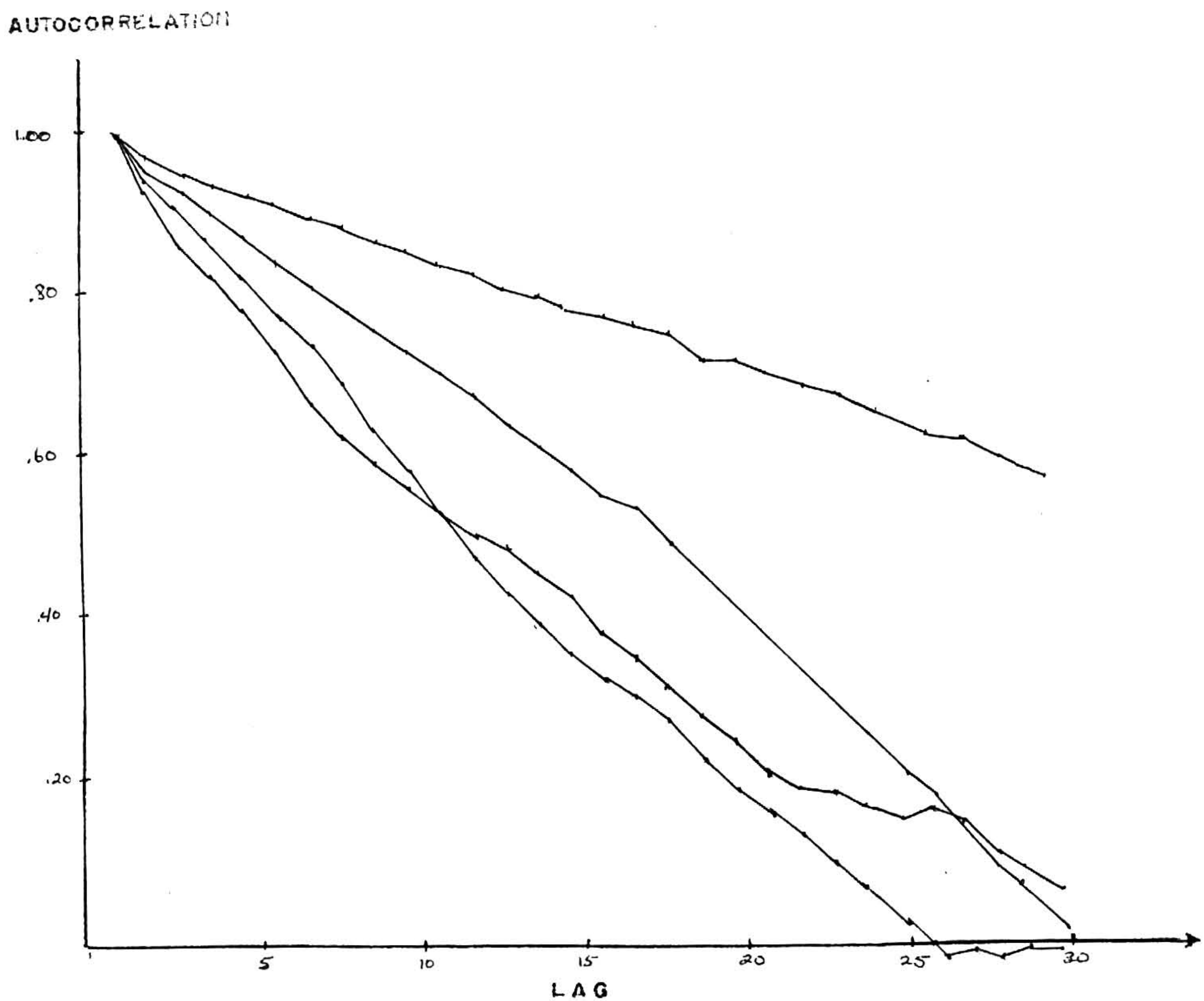
```
   21 XSS=XSS+(XX(I)*XX(I))
      AVE=XS/II
      STD=SQRT(XSS/II-AVE*AVE)
      IF(J.GT.1)       GO TO 19
      DO 20 N=1,II
   20 XX(N)=JQUE(N)
      X1=AVE
      X2=STD
      X3=FMAX
      GO TO 4
   19 IF(J.GT.2)      GO TO 4
      DO 22 N=1,II
   22 XX(N)=JSYS(N)
      X4=AVE
      X5=STD
      X6=FMAX
    4 CONTINUE
      X7=AVE
      X8=STD
      X9=FMAX
C     PRINT OUT STATISTICS
      WRITE(3,8)
      WRITE(3,6)
      WRITE(3,7) X1,X2,X3,X4,X5,X6,X7,X8,X9
      WRITE(3,11)
      WRITE(3,12)
      DO 15 I=1,II
      WRITE(3,13) TIME(I),UTIL(I),JQUE(I),JSYS(I),AR(I),DEP(I),QUET(I),S
     1YST(I),SER(I)
   15 CONTINUE
      II=0
      RETURN
      END
```

```
      SUBROUTINE CUTPUT(NSET)
      COMMON ID,IM,INIT,JEVENT,JMCNIT,MFA,MSTOP,MX,MXC,
     1NCOLCT,NHISTO,NOQ,NORPT,NOT,NPRAMS,NRUN,NRUNS,NSTAT,
     2CUT,SCALE,NSEED,TNOW,TSTART,TSTOP,MXX
      COMMON ATTRIB(4),ENQ(4),INN(4),JCELLS(5,22),KRANK(4),
     1MAXNQ(4),MFE(4),MLC(4),MLE(4),NCELLS(5),NQ(4),PARAMS(20,4),
     2QTIME(4),SSUMA(10,5),SUMA(10,5),IX(8)
      COMMON XSR,IKONT,XQUE,XSYS,II,ISERVE,P,EX,DELTA,NOBS,NREP,TLAST
      COMMON UTIL(500),JQUE(500),JSYS(500),XX(500),TIME(500)
      DIMENSION NSET(6,1)
      DIMENSION IU(500)
    4 FORMAT(4I5,F10.3,10X,I5)
    5 FORMAT(4I5)
    6 FORMAT(4F10.4)
   11 FORMAT(I10,4F10.4)
   12 FORMAT(4Z16)
   13 FORMAT(25I3,'UTL',I2)
   14 FORMAT(25I3,'QUE',I2)
   16 FORMAT(25I3,'SYS',I2)
   21 FORMAT(5F15.5)
      N=II/25
      ITEST=N*25
      IF(ITEST.NE.II)      GO TO 17
      DO 1 I=1,N
      DO 19 JY=1,25
   19 IU(JY+25*(I-1))=UTIL(JY+25*(I-1))*100
    1 WRITE(2,13) (IU(J+25*(I-1)),J=1,25),I
      DO 2 I=1,N
    2 WRITE(2,14) (JQUE(J+25*(I-1)),J=1,25),I
      DO 15 I=1,N
   15 WRITE(2,16) (JSYS(J+25*(I-1)),J=1,25),I
C     PUNCH INPUT FOR NEXT RUN
      WRITE(2,6) XSR,XQUE,XSYS
      WRITE(2,4) MSTOP,JCLEAR,NORPT,NEP,TNOW,NSEED
      WRITE(2,12)IX(1),IX(2)
      WRITE(2,5) (INN(I),I=1,NOQ)
      WRITE(2,5) (KRANK(I),I=1,NOQ)
      WRITE(2,5) (NCELLS(I),I=1,NHISTO)
      WRITE(2,6) (PARAMS(1,J),J=1,4)
C     PUNCH THE CALENDAR
      DO 7 JQ=1,NOQ
      LINE=MFE(JQ)
      IF(LINE-1) 8,10,10
   10 DO 9 I=1,IM
      ATTRIB(I)=NSET(I,LINE)
    9 ATTRIB(I)=ATTRIB(I)/SCALE
      WRITE(2,11) JQ,(ATTRIB(I),I=1,IM)
      LINE=NSET(MX,LINE)
      IF(LINE-7777)  10,8,8
    8 CONTINUE
    7 CONTINUE
C     PUNCH CURRENT STATISTICS
      DO 20 I=1,NCOLCT
   20 WRITE(2,21) (SUMA(I,J),J=1,5)
      DO 22 I=1,NSTAT
   22 WRITE(2,21) (SSUMA(I,J),J=1,5)
      RETURN
   17 WRITE(3,18)
```

```
   18 FORMAT(40X,'***** ERROR IN OUTPUT')
      STOP
      END
```

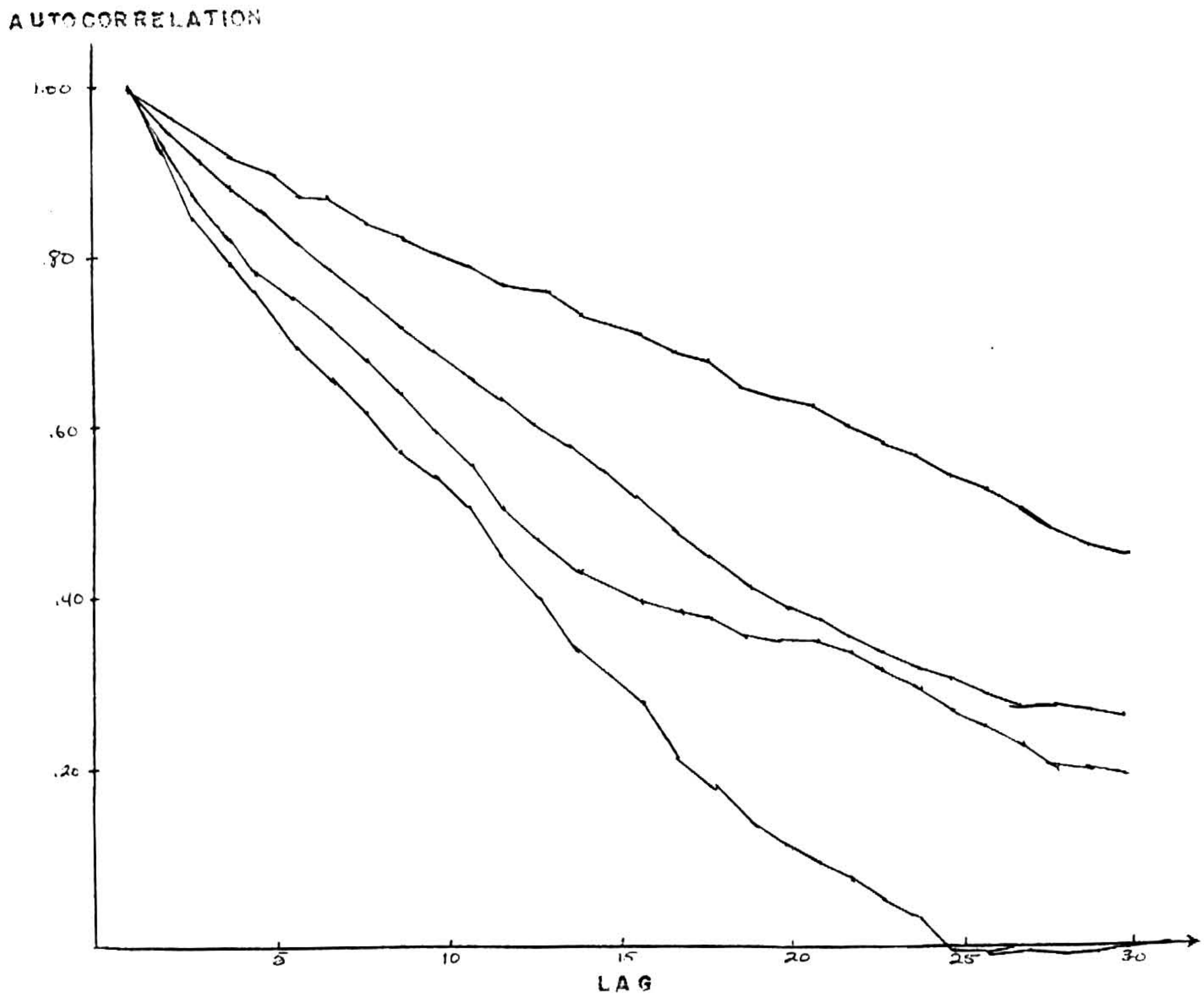


General Logic Diagram
Figure 1.2



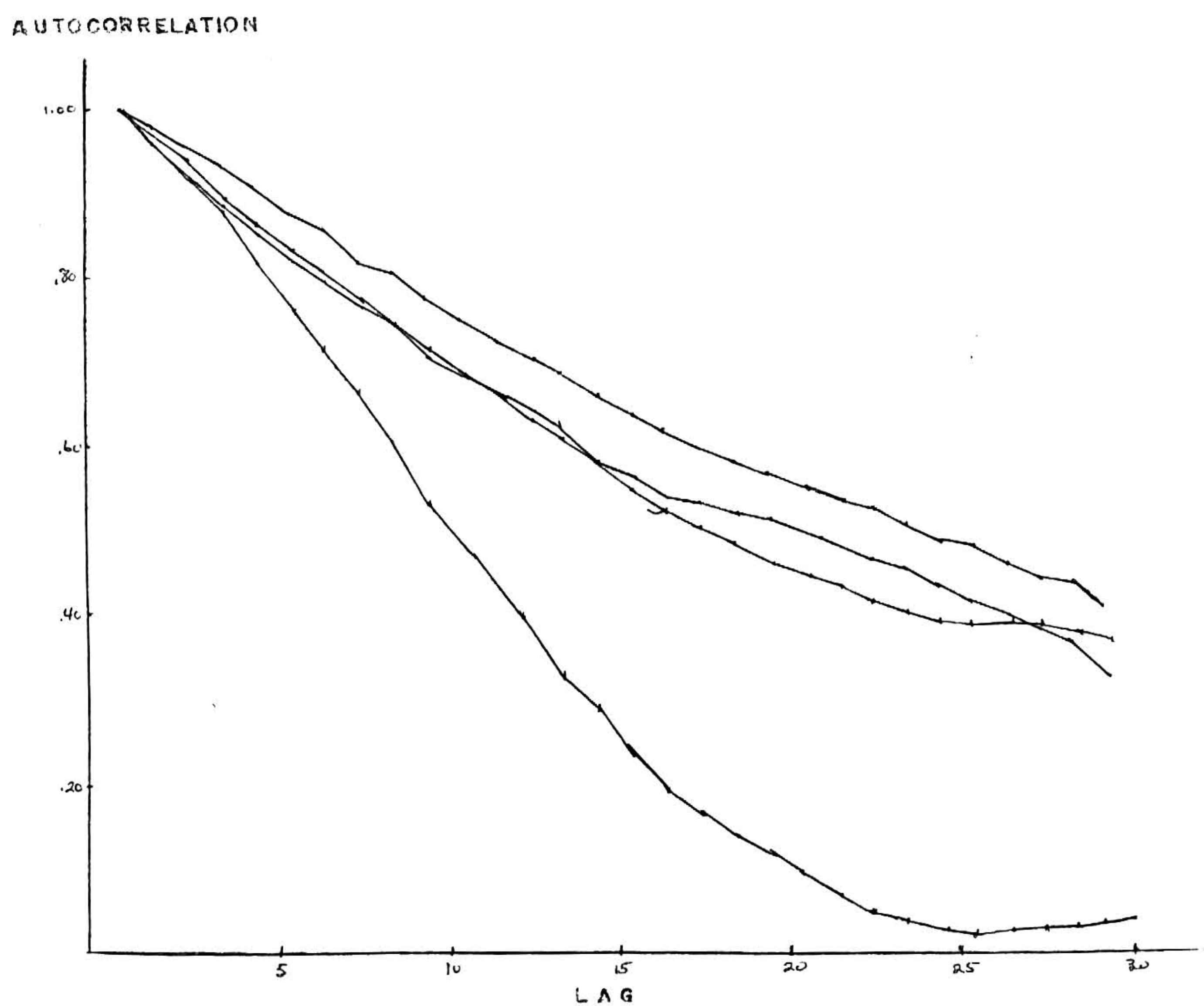
Autocorrelation of Number of Units in the System Statistic

Figure 1.3.1



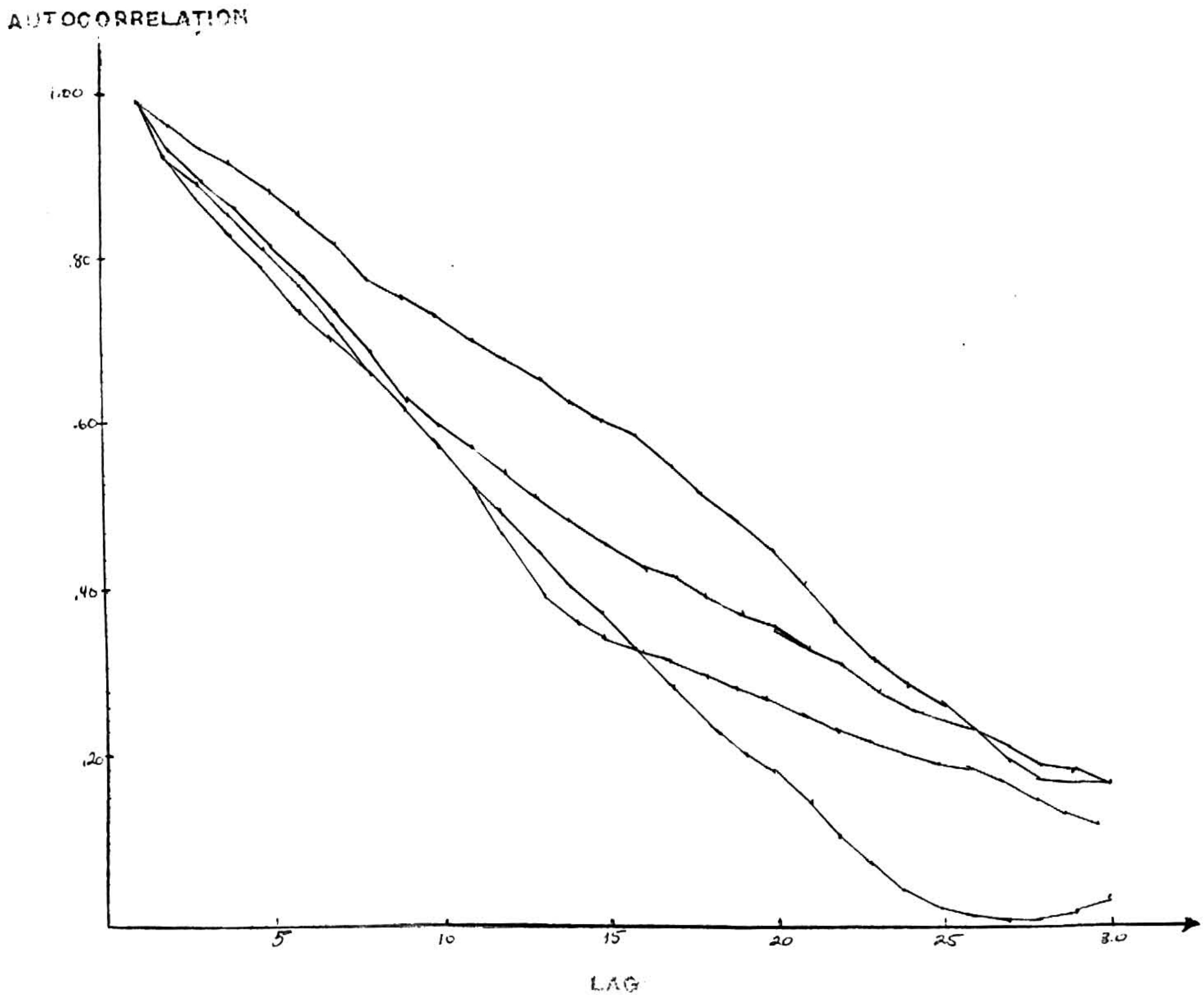
Autocorrelation of the Number of Units in the System Statistic

Figure 1.3.2



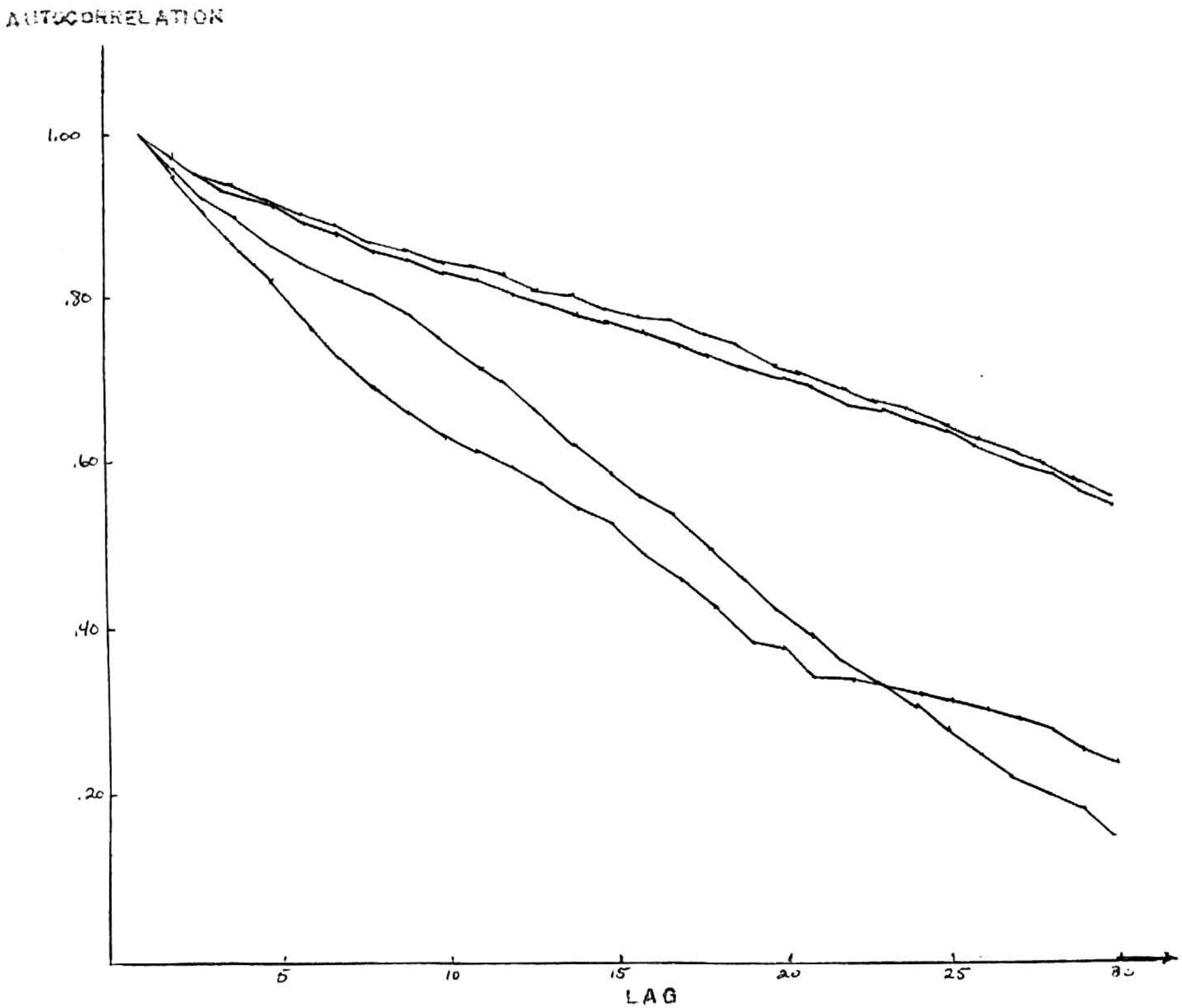
Autocorrelation of the Number of Units in the System Statistic

Figure 1.3.3



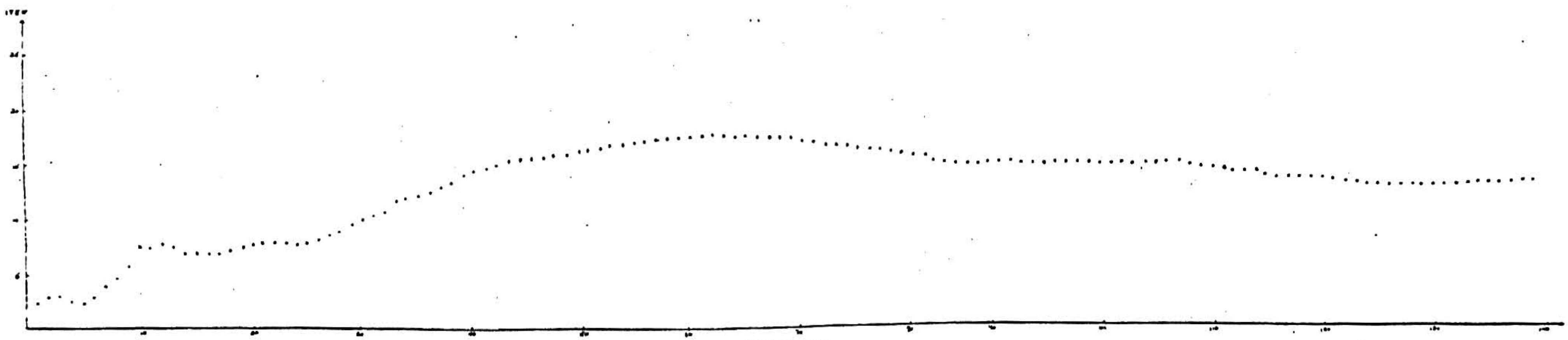
Autocorrelation of the Number of Units in the System Statistic

Figure 1.3.4



Autocorrelation of the Number of Units in the System Statistic

Figure 1.3.5



Average Number of Units in the System Statistic

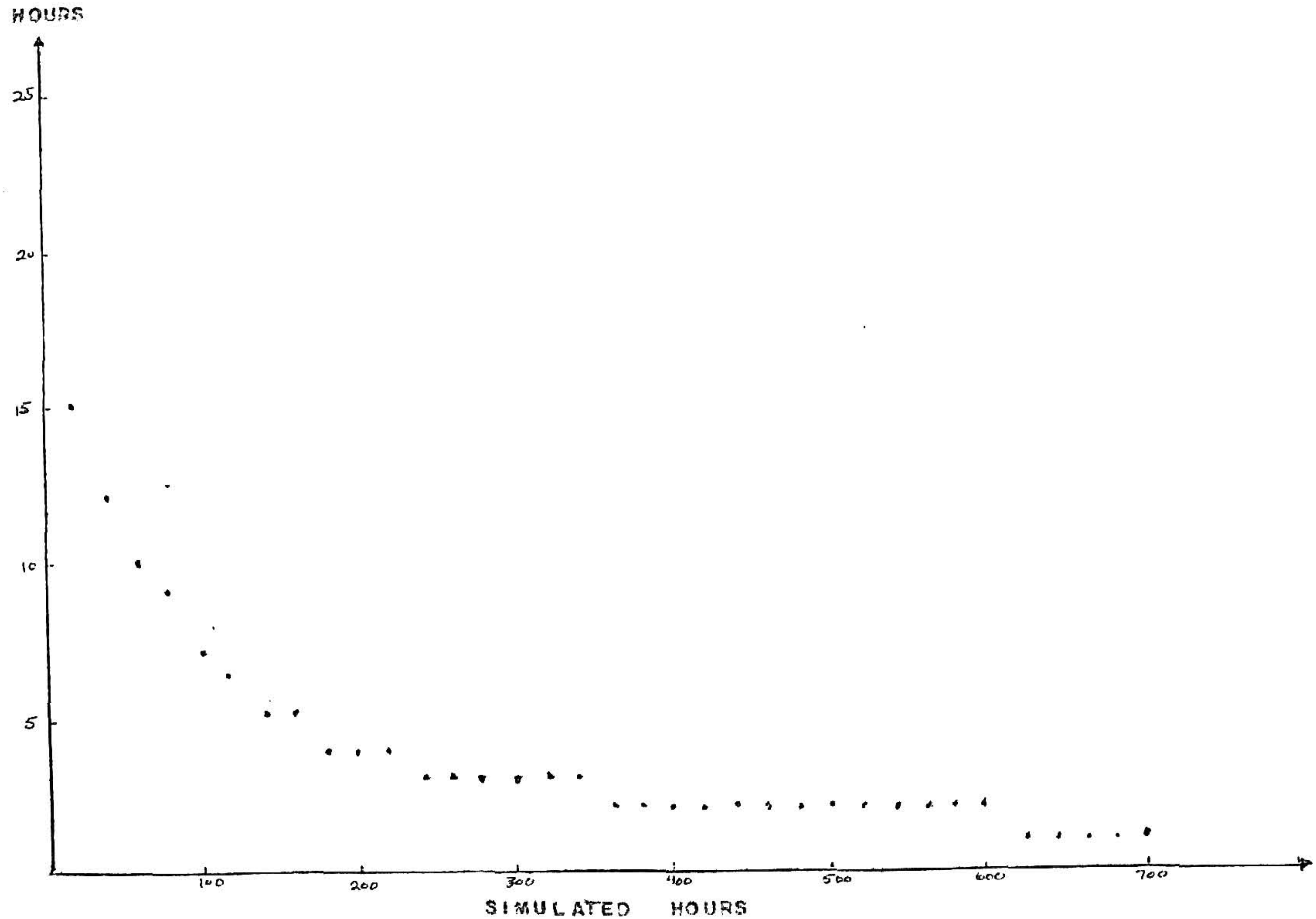
Figure 1.4

APPENDIX D:

1) Documentation of Conway's Method
   Figures 2.1, 2.2, 2.3, 2.4, 2.5, 2.6

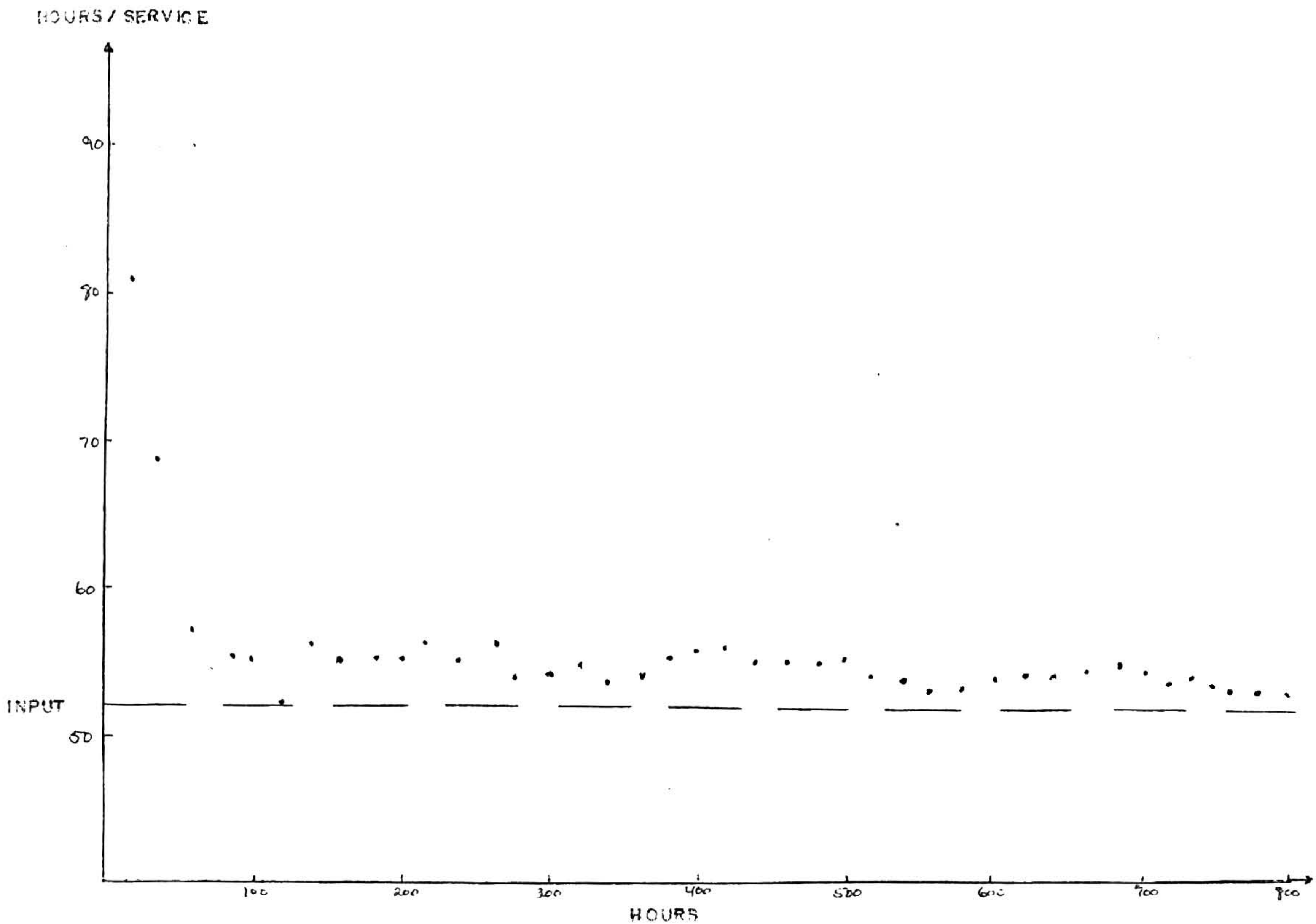
2) Documentation of Bueno's Method
   Figure 3.1

3) Documentation of Reese's Method
   Listing of Reese's Method Program and Results



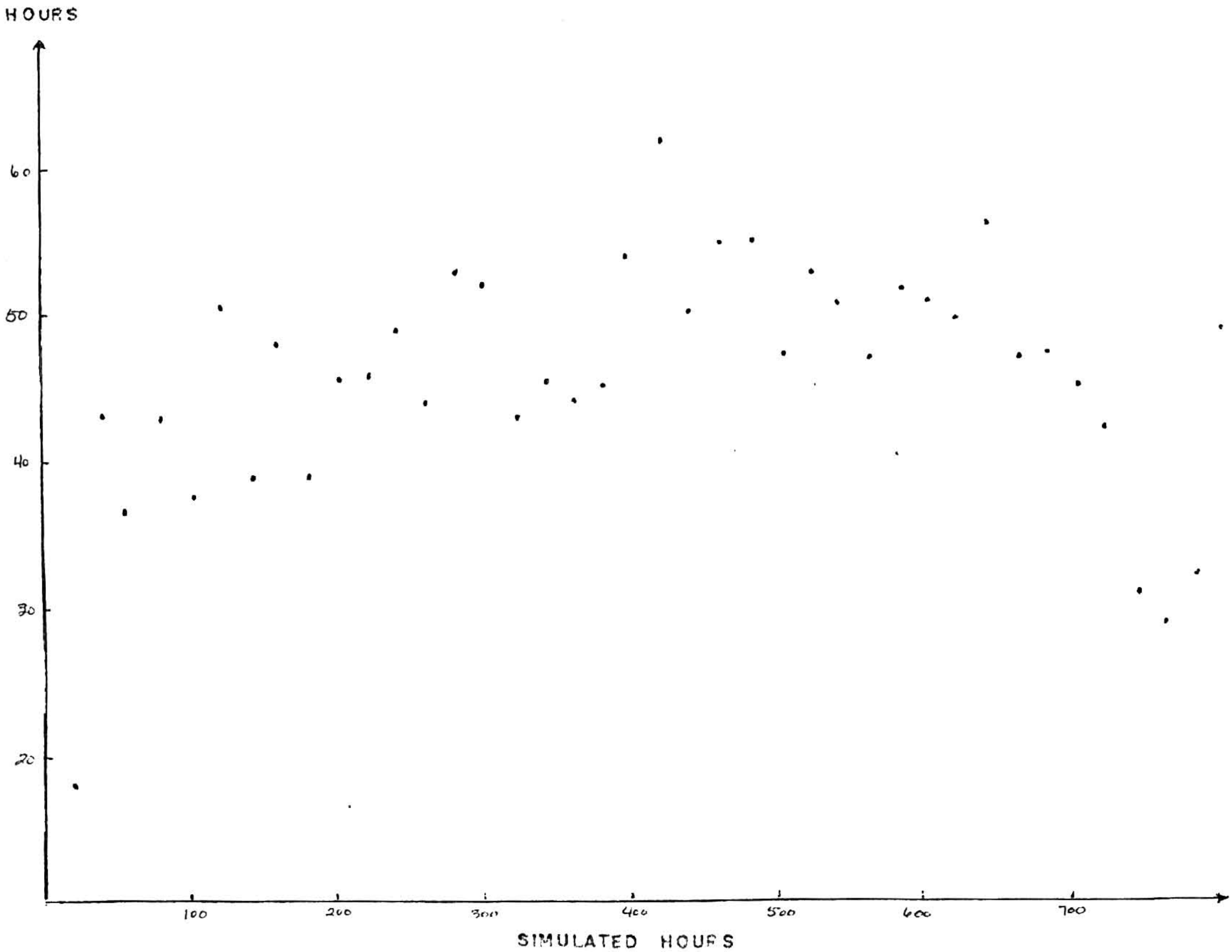
Average Increase in Number of Units in the System

Figure 2.1



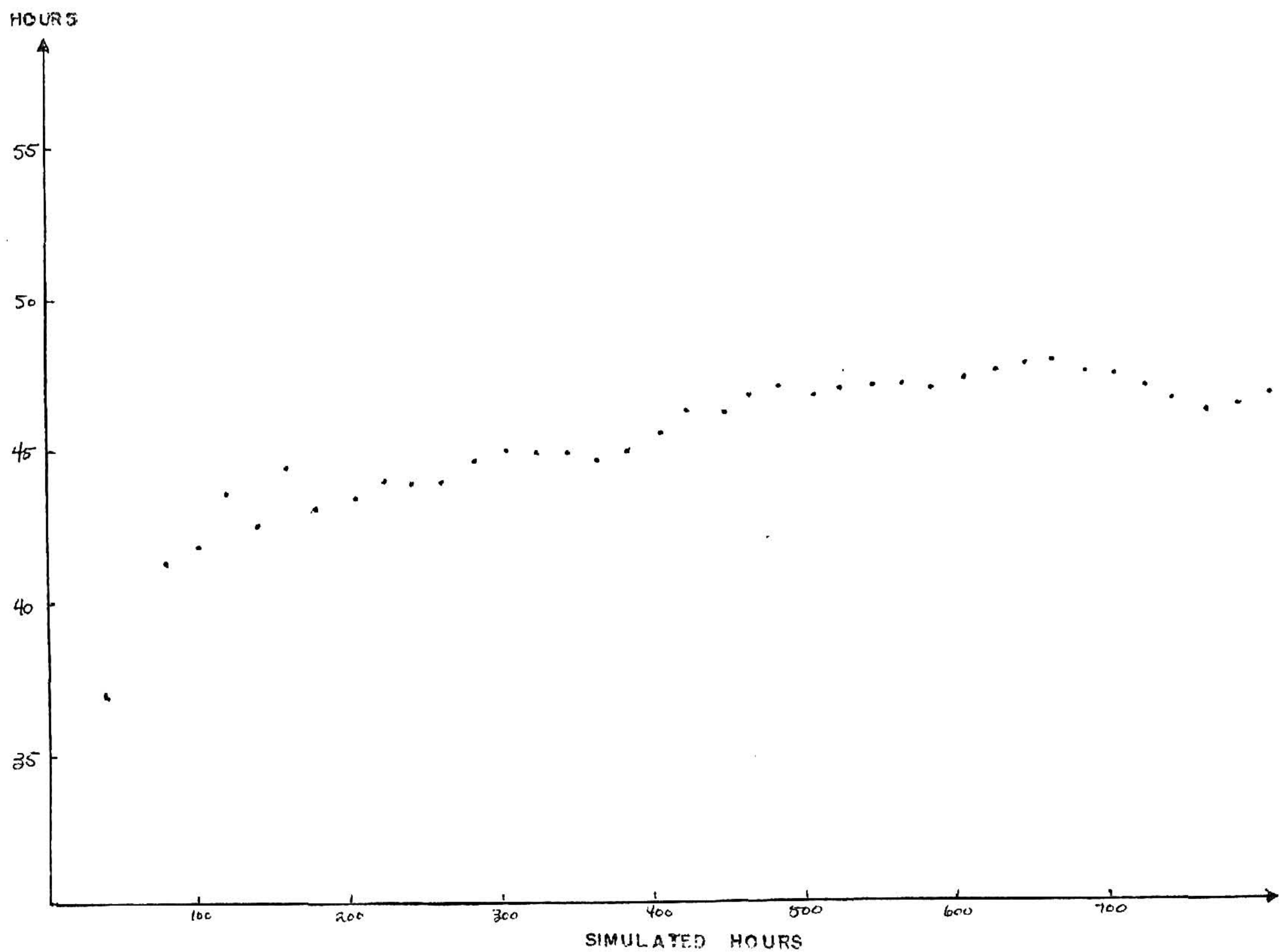
Average Service Time

Figure 2.2



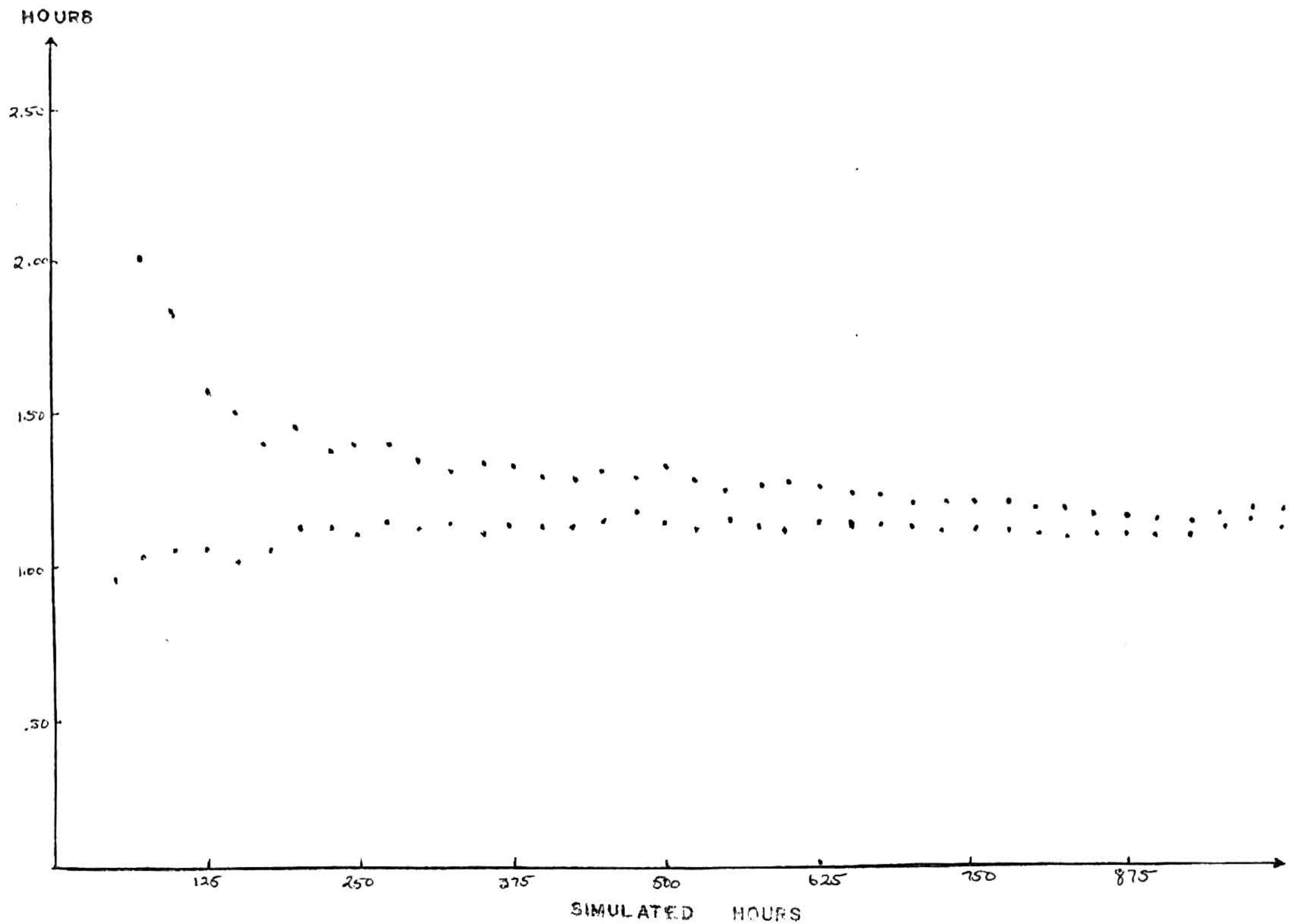
Number of Units in the System

Figure 2.3



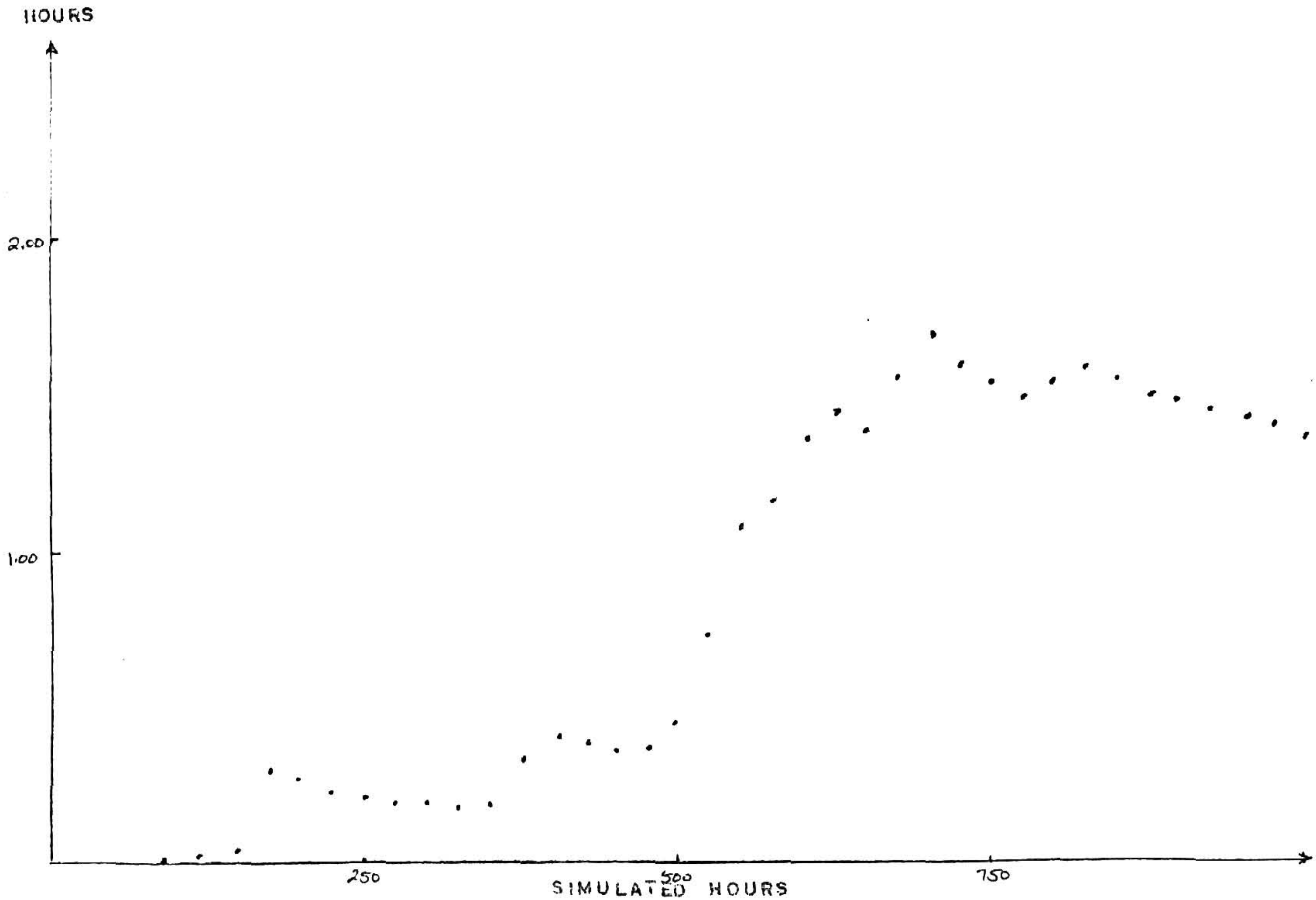
Average Number of Units in the System

Figure 2.4



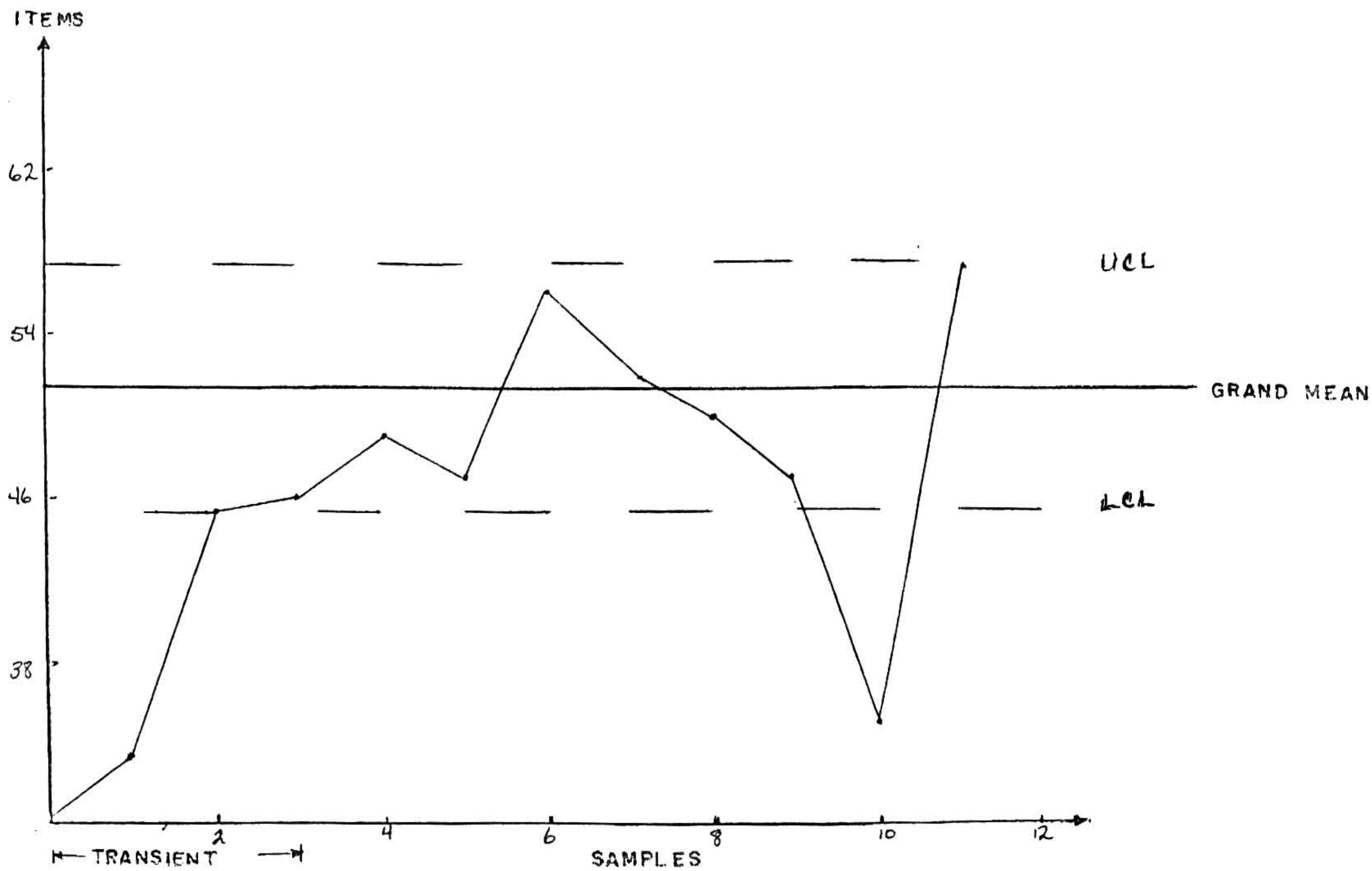
Average Inter-arrivals and Average Interdeparture Times

Figure 2.5



Average Time in the Queue

Figure 2.6



Sample Mean Values

Figure 3.1

RESULTS OF THE REESE METHOD

(Including sample computations)

TRIAL	TIME	N	B		A
1	75	150	.1875	.99	4.25
2	100	200	.1875	.99	4.25
3	125	250	.1875	.99	4.25
4	150	300	.1875	.99	4.25
5	175	350	.1875	.99	4.25
6	200	400	.1875	.99	4.25
7	225	450	.1875	.99	4.25
8	250	500	.1875	.99	4.25
9	275	550	.1875	.99	4.25
10	300	600	.1875	.99	4.25
11	325	650	.1875	.99	4.25
20	550	1100	.1875	.99	4.25
21	575	1150	.1875	.99	4.25
22	600	1200	.1875	.99	4.25
23	625	1250	.1875	.99	4.25
24	650	1300	.1875	.99	4.25
25	675	1350	.1875	.99	4.25
26	700	1400	.1875	.99	4.25

```
$JOB           RGP,KP=26,TIME=5,PAGES=100
C  C  A PROGRAM FOR REESE METHOD
      DIMENSION X(1400)
   10 FORMAT(3I5<
   11 FORMAT(25F3.0<
   12 FORMAT(10HGRAND MEAN,F15.5,10X,11HSAMPLE MEAN,F15.5<
   13 FORMAT(8HVARIANCE,F10.5,10X,10HSAMPLE VAR,F10.5<
      READ 10,M,IN,N
      READ 11,(X(I),I=1,1400)
      PRINT 10,M,IN,N
C     CALCULATE GRAND MEAN
      X2=0.0
      Z=N
      DO 1 I=1,N
    1 X2=X2+X(I)
      X2=X2/Z
C     CALCULATE SAMPLE MEAN
      J=M+N
      K=J+IN
      X1=0.0
      Y=IN
      DO 2 I=J,K
    2 X1=X1+X(I)
      X1=X1/Y
C     CALCULATE VARIANCE
      SS=0.0
      DO 3 I=J,K
      DIF=X(I)-X1
    3 SS=SS+(DIF*DIF)
      SS=SS/(Y-1.0)
C     CALCULATE VARIANCE OF SAMPLE
      R=0.0
      DO 4 IT=1,M
      A=0.0
      DO 5 I=1,IN
      IQ=I+IT
    5 A=A+((X(I)-X1)*(X( IQ )-X1))
      A=A/Y
    4 R=R+A
      S2=(SS+(2.0*R))/Y
      PRINT 12,X2,X1
      PRINT 13,SS,S2
      SX=SQRT(S2)
      B=(X1-X2)/SX
      DELTA=(.20*X1)/SX
      T0=B**2
      T1=(B-DELTA)**2
      T2=(B+DELTA)**2
C     PRINT OUT T0,T1,T2
      PRINT 102,SX,T0,T1,T2
  102 FORMAT(3HSX#,F10.5,10X,3HT0#,F10.5,10X,3HT1#,F10.5,3HT2#,F10.5<
      STOP
      END
```

APPENDIX E:

1) Documentation of Experiments
   Tables 4.1, 4.2, 4.3, 4.4
   Figures 4.1, 4.2, 4.3, 4.4

2) The Relationship of Spectra and Simulation Input
   Tables 4.5, 4.6, 4.7

Table 4.1

Size of System Statistic
Average, Maximum, and Minimum Coherence
M=30, Not Pre-Whitened

Sample	Time	Mean	Maximum	Minimum	
1	(0.100) (50.150)	.1681	1.0000	.0019	
2	(50.150) (100.200)	.0224	.2119	.0197	
3		.0798	.2555	.0007	
4		.0339	.2952	.0370	
5		.0572	.1270	.0176	
6		.0117	.1085	.0000	
7		.0879	.6924	.0012	
8		.1590	.4507	.0004	
9		.0310	.5180	.0004	
10		.0557	.1972	.0003	
11		.0624	.2125	.0092	
12		.0980	.1715	.0017	
*13		.1407	.4716	.0123	
*14		.1595	.4361	.0016	
*15		.0347	.0830	.0000	
16		.0814	.2559	.0003	
17		.2949	.6377	.0250	
18		.1786	.5092	.0010	
19		.1212	.2709	.0006	
20	(950.1050) (1000.1100)	.4260	.6938	.1159	Above U.C.L.

*The transient stage terminates within these samples.

Table 4.2

Size of System Statistic
Average, Maximum, and Minimum Coherence
M=30, Least Squares Used to Pre-Whiten

Sample	Time	Mean	Maximum	Minimum
1	(0.100) (50.150)	.1641	.8945	.0013
2	(50.150) (100.200)	.1295	.3021	.0195
3		.0822	.2486	.0034
4		.1213	.2682	.0354
5		.0526	.1226	.0088
6		.0528	.1244	.0005
7		.1835	.6273	.0022
8		.0955	.5791	.0012
9		.1761	1.0000	.0006
10		.0521	.1728	.0046
11		.0795	.1823	.0146
12		.0950	.1949	.0011
*13		.1132	.3836	.0018
*14		.1061	.2070	.0193
*15		.0476	.0579	.0002
16		.0934	.3627	.0026
17		.3973	.8130	.1210
18		.3026	.7447	.0238
19		.3344	.8640	.1504
20	(950.1050) (1000.1100)	.4916	.8298	.0874

*The transient stage terminates within these samples.

Table 4.3

Queue Size Statistic
Average, Maximum, and Minimum Coherence
M=30

Sample	Time	Mean	Maximum	Minimum
1	(0.100) (50.150)	.1127	.5481	.0018
2	(50.150) (100.200)	.1471	.3083	.0021
3		.1082	.3222	.0019
4		.1297	.3176	.0207
5		.0615	.1357	.0104
6		.0683	.2110	.0032
7		.1460	.6199	.0043
8		.1311	.4046	.0020
9		.0781	.2269	.0039
10		.0595	.3087	.0006
11		.0288	.0880	.0002
12		.0221	.0488	.0020
*13		.0542	.2612	.0000
*14		.0665	.2292	.0010
*15		.1058	.3114	.0022
16		.0817	.2748	.0066
17		.2791	.6239	.0893
18		.2130	.5246	.0037
19		.0560	.1635	.0016
20	(950.1050) (1000.1100)	.0976	.6579	.0130

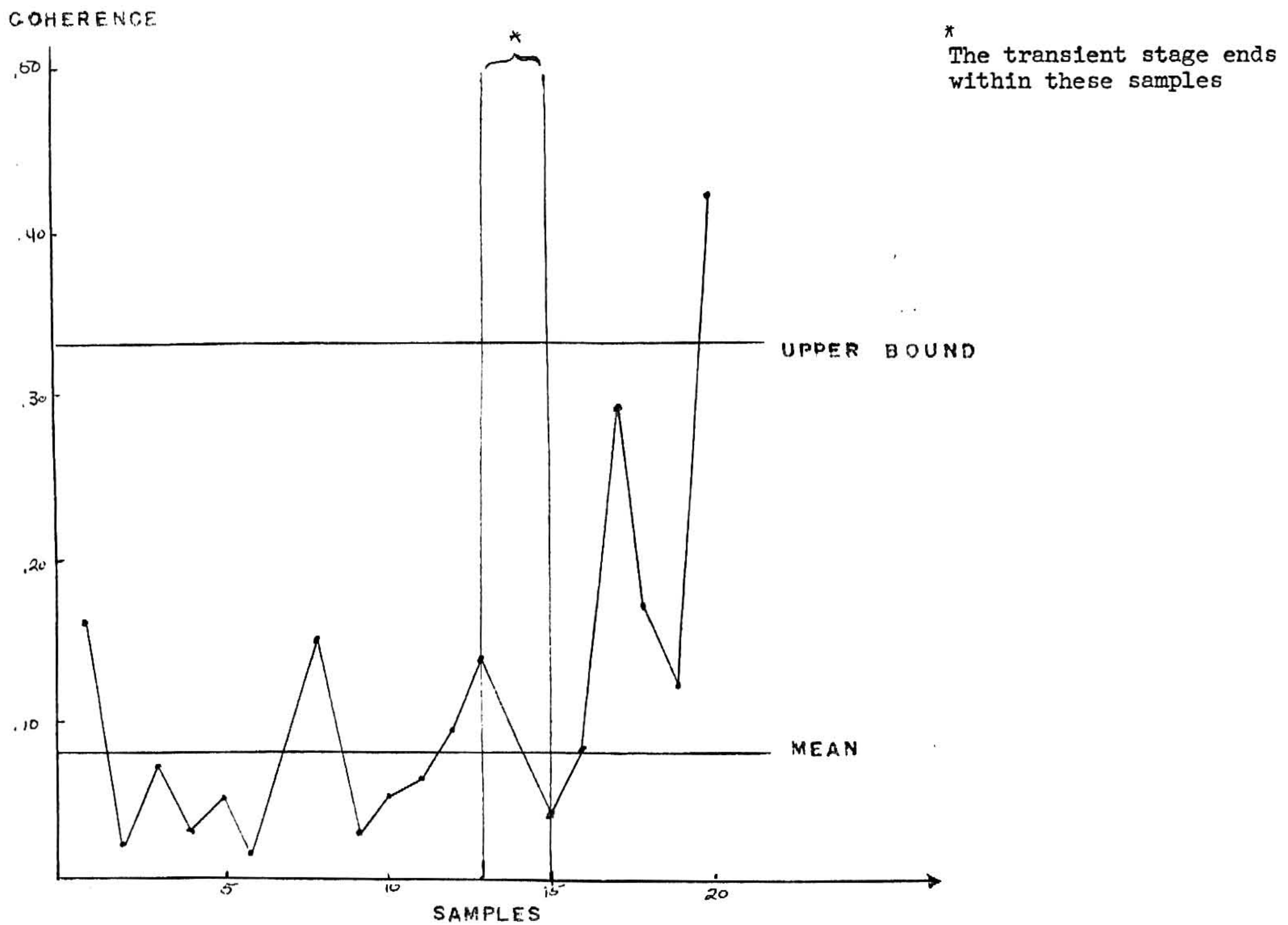
*The transient stage terminates somewhere within these samples.

Table 4.4

Utilization Statistic
Average, Maximum, and Minimum Coherence
M=30

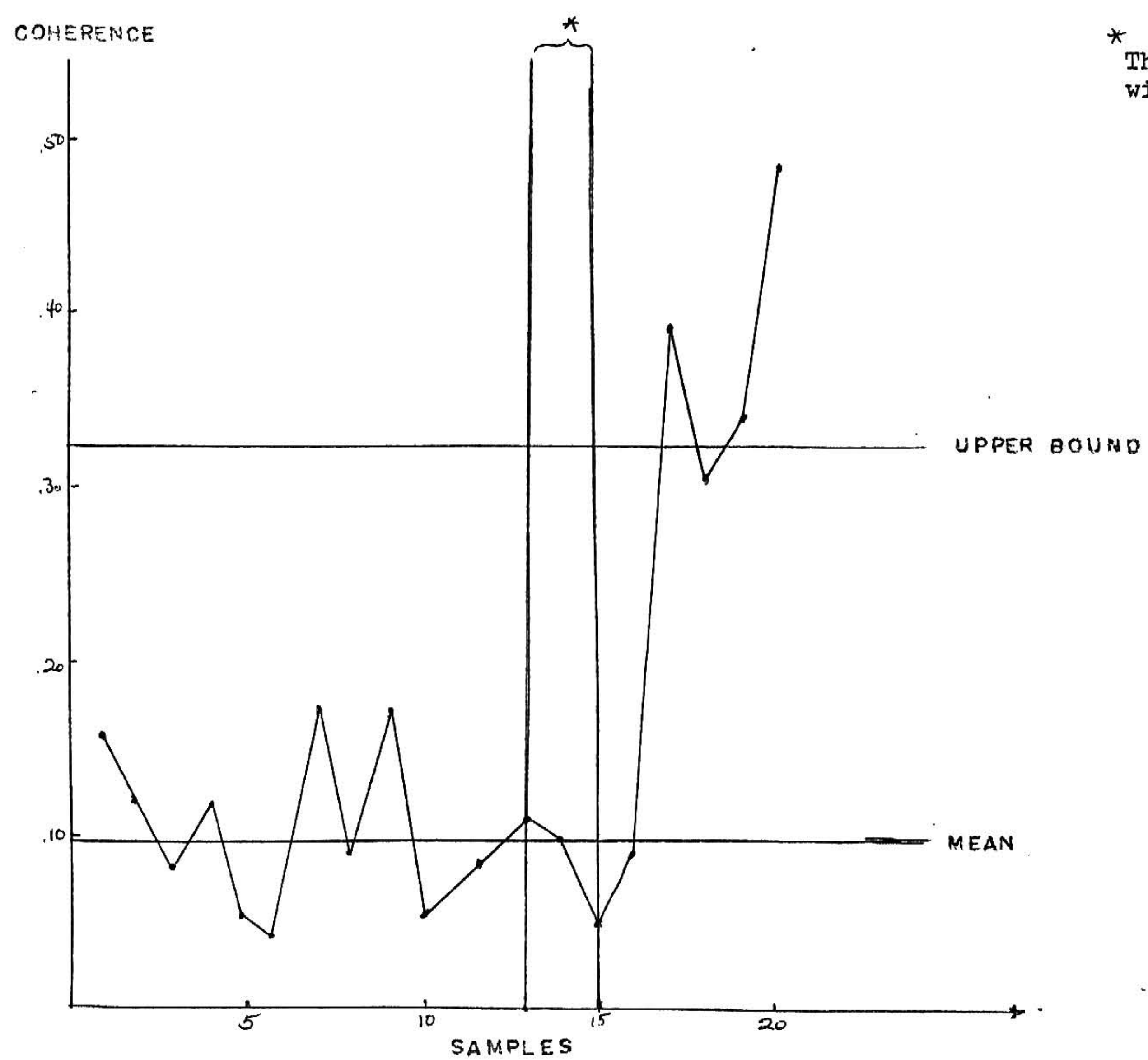
Sample	Time	Mean	Maximum	Minimum
1	(0,100) (50,150)	.0022	.0192	.0000
2	(50,150) (100,200)	.0047	.0170	.0006
3		.0078	.0397	.0000
4		.0706	.3998	.0002
5		.0822	.2771	.0027
6		.0193	.0784	.0004
7		.1702	.8086	.0003
8		.2159	.6290	.0252
9		.1671	.7620	.0138
10		.1555	.4855	.0026
11		.1242	.2668	.0053
12		.1656	.4429	.0053
13		.1321	.5221	.0082
14		.2553	.6714	.0549
15		.0376	.1081	.0003
16		.0631	.1741	.0010
*17		-----	-----	-----
*18		-----	-----	-----
*19		-----	-----	-----
*20		-----	-----	-----

*These samples had no variance, thus no coherence exists.



Mean Coherence Values for Number of Units in the System Statistic

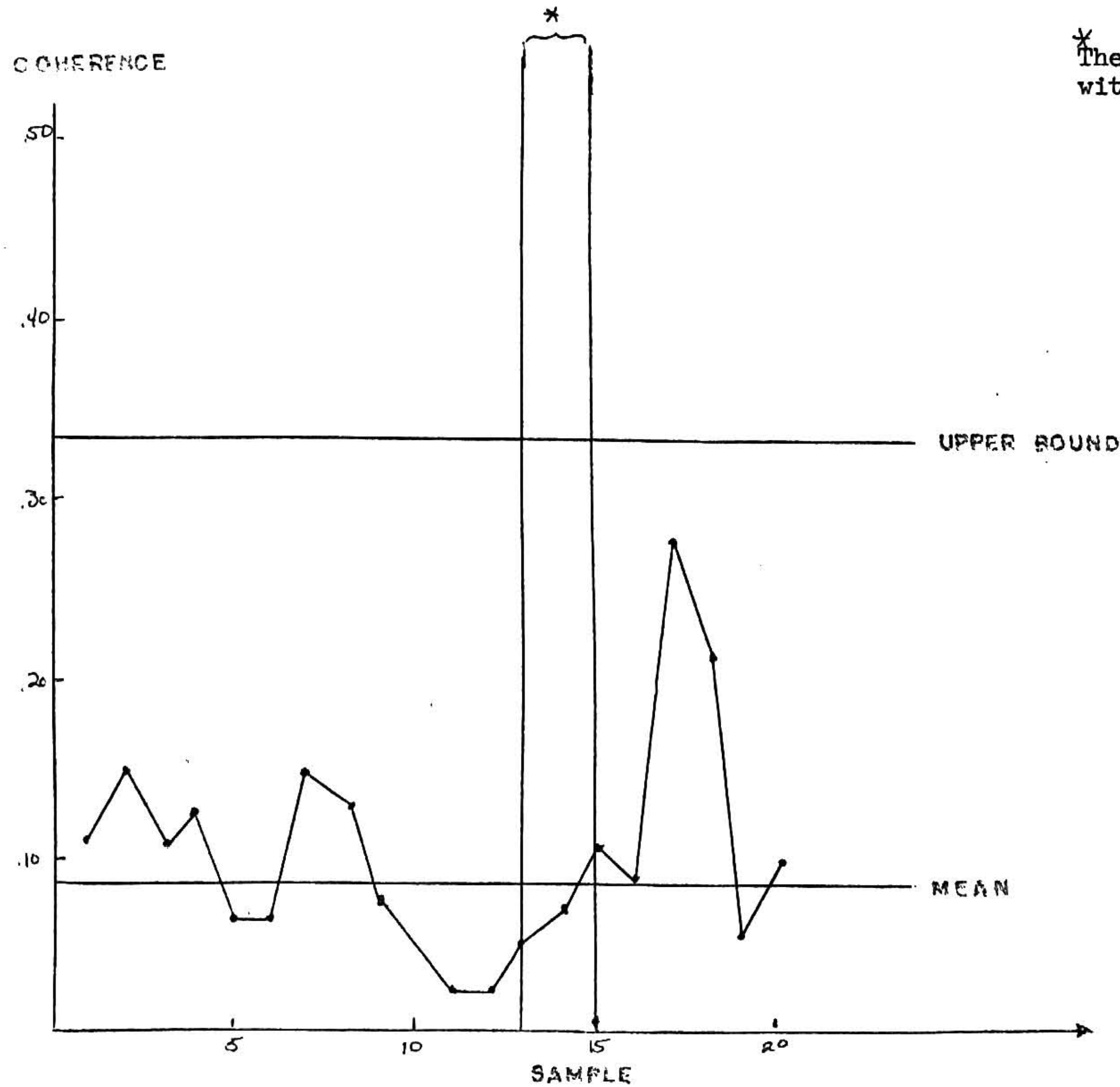
Figure 4.1



* The transient stage ends within these samples

Mean Coherence Values for Number of Units in the System Statistic

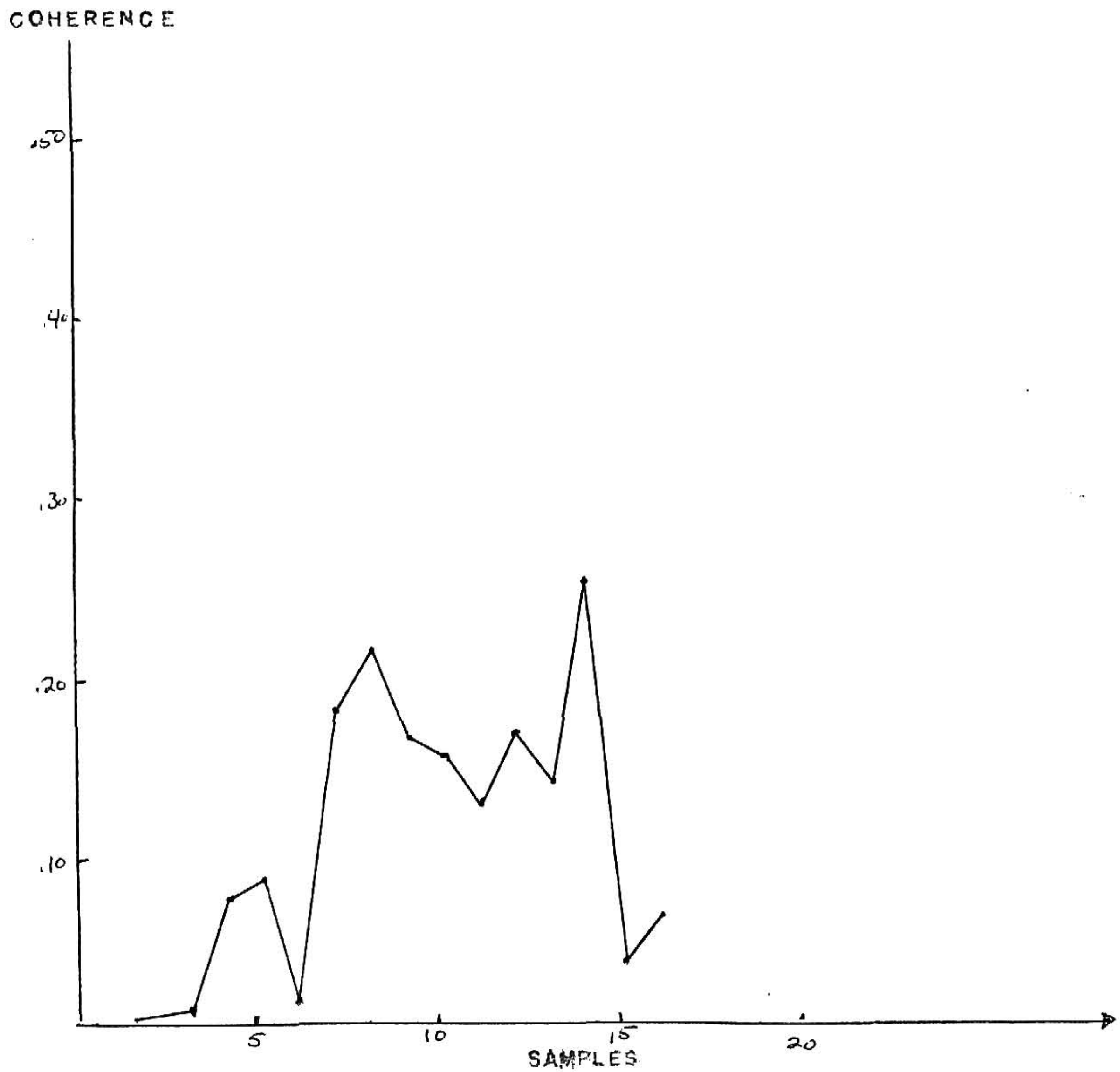
Figure 4.2



*The transient stage ends within these samples.

Mean Coherence Values for Queue Size

Figure 4.3



Mean Coherence Values for Utilization Statistic

Figure 4.4

Table 4.5

Interpretation of Cyclic Elements in
Size Of System Statistic

Segment	Time	Elements	(Items/cycle)	
1	(0,100)	13.1	10.5	7.7
2	(50,150)	13.1		
3	(100,200)	13.1		
4	(150,250)	13.1	8.3	
5	(200,300)	10.0		
6	(250,350)	12.5	7.7	
7	(300,400)	14.0	8.3	
8	(350,450)	13.5	11.0	7.7
9	(400,500)	13.5	11.0	
10	(450,550)	16.7	10.5	
11	(500,600)	10.0		
12	(550,650)	13.1	10.5	
13	(600,700)	None found		
14	(650,750)	13.1	10.5	
15	(700,800)	None found		
16	(750,850)	8.3		
17	(800,900)	13.1	10.5	8.0
18	(850,950)	13.1	10.5	
19	(900,1000)	None found		
20	(950,1050)	13.1	10.5	

Table 4.6

Queue Size Statistic

Segment	Time	Elements	(Items/cycle)
1	(0,100)	13.1	10.5
2	(50,150)	12.5	63.5
3	(100,200)	13.1	
4	(150,250)	8.3	
5	(200,300)	8.3	
6	(250,350)	12.5	
7	(300,400)	7.2	
8	(350,450)	13.1	10.5
9	(400,500)	10.5	
10	(450,550)	10.5	8.0
11	(500,600)	None	
12	(500,650)	10.5	
13	(600,700)	10.5	
14	(650,750)	13.3	
15	(700,800)	13.3	
16	(750,850)	8.3	
17	(800,900)	13.1	
18	(850,950)	13.1	10.5
19	(900,1000)	12.5	
20	(950,1050)	10.5	

Table 4.7

Utilization Statistic

Segment	Time	Elements	(Items/cycle)
1	(0,100)	48.0	
2	(50,150)	48.0	
3	(100,200)	55.6	
4	(150,250)	None	
5	(200,300)	1.00	
6	(250,350)	1.00	
7	(300,400)	13.1	1.00
8	(350,450)	13.1	1.00
9	(400,500)	13.1	1.00
10	(450,550)	16.7	1.00
11	(500,600)	10.5	
12	(550,650)	16.7	
13	(600,700)	None	
14	(650,750)	16.7	
15	(700,800)	None	
16	(750,850)	9.1	
17	(800,900)	1.00	
*18	(850,950)	-----	
*19	(900,1000)	-----	
*20	(950,1050)	-----	

*It was not possible to compute the spectra for these samples.

BIBLIOGRAPHY

1. Blackman, R. B. and Tukey, J. W. Power Spectrum from the View of Communications Engineers, Dover Paperback Press, 1958.

2. Bueno, Ramon 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. Conway, R. W. "Some Tactical Problems in Simulation Method", Rand Corporation Memorandum RM-3244-BR, October, 1962.

5. Cox, Grady J. and Mize, Joe H. Essentials of Simulation, Prentice-Hall, Inc., 1968.

6. Fishman, George S. "Problems in the Statistical Analysis of Simulation Experiments: The Comparison of Means and the Length of Sample Records", Communications of the ACM, Vol. 10, No. 2, 1967.

7. Fishman, George S. and Kiviat, P. J. "The Analysis of Simulation-Generated Time Series", Management Science, Vol. 13, No. 7, March, 1967.

8. Goodman, N. R. "Spectral Analysis of Multiple Stationary Time Series", Proceedings, Symposium on Time Series Analysis, John Wiley and Sons, 1963.

9. Granger, C. W. J. Spectral Analysis of Economic Time Series, Princeton University Press, 1964.

10. Hamming, R. W. Numerical Methods for Scientists and Engineers, McGraw-Hill Book Co., Inc., 1962.

11. Hillier, Frederick S. and Lieberman, Gerald J. Introduction to Operations Research, Holden-Day, Inc., 1967.

12. Jenkins, G. M. "Cross-Spectral Analysis and the Estimation of Linear Open Loop Transfer Functions", Proceedings, Symposium of Time Series Analysis, John Wiley and Sons, 1963.

13. Jenkins, G. M. "General Considerations in the Analysis of Spectra", Technometrics, Vol. 3, No. 2, May, 1961.

14. Jenkins, G. M. and Watts, D. G. Spectral Analysis and Its Applications, Holden-Day, Inc., 1968.

15. Naylor, Balintfy, Burdick, and Chu Computer Simulation Techniques, John Wiley and Sons, 1966.

16. Parzen, Emanuel "An Approach to Empirical Time Series Analysis", Radio Science Journal of Research, Vol. 68D, No. 9, September, 1964.

17. Parzen, Emanuel "Mathematical Considerations in the Estimation of Spectra", Technometrics, Vol. 3, No. 2, May, 1961.

18. Parzen, Emanuel Stochastic Processes, Holden-Day, Inc., 1962.

19. Reese, Peter A. "Steady State Parameter Estimation in Computer Simulated Systems", Arizona State University Bulletin, January, 1969.

20. Rivett, Patrick An Introduction to Operations Research, Basic Books, Inc., 1968.

21. Sasieni, M., Yaspan, A., and Freidman, L. Operations Research: Methods and Problems, John Wiley and Sons, Inc., 1966.

22. Vaswani, S. D. Job Shop Scheduling, Master of Science Thesis in Industrial Engineering, Kansas State University, Manhattan, Kansas, 1969.

23. Wald, A. Sequential Analysis, Wiley & Sons, 1947.

A STUDY OF THE TRANSIENT STAGE PROBLEM IN SIMULATIONS

by

FRED RODNEY HARRIS

B.A., Southwestern College, 1968

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

The purpose of this paper is to consider the problem of estimating the termination of the transient stage of a simulation. The transient stage of a simulation is that period of simulated time in which the statistics of the simulation are erratic and unstable. In the past, knowledge of a simulation's transient stage was gained by making pilot runs of the simulation, or by examining the statistics after the simulation was completed. These techniques can be wasteful of computer time and may cause the experimenter to mistakenly interpret transient stage statistics as representative of a truely stable system. There is clearly a need for a technique which estimates the end of the transient stage as the simulation is in progress.

Past research in this area has been presented. The initial work by Conway suggested making several pilot runs of a simulation and determining the transient stage by graphing the various statistics. The use of control charts was suggested by Bueno. A sequential t-test was suggested by Reese. Fishman first introduced the use of spectral analysis in examining the simulation generated output. He did not consider the transient stage problem.

This paper presents the concept of using spectral analysis to resolve the problem. The approach is based on the fact that two spectra from a statistic with stable variance will be strongly correlated. The squared correlation, the coherence, of two successive spectra of a simulation statistic was found. An upper bound was determined. If the coherence of the two segments was greater than the bound, the two segments can be considered part of the stabilized simulation. If

the coherence was less than the bound, the two segments can be considered part of the transient stage. This technique was applied successfully to three simulation statistics. The technique is presented as a solution to the problem.