INTERACTIVE ANALYSIS OF OUTPUT FROM GPSS-BASED SIMULATIONS

Richard W. Andrews
Thomas J. Schriber

ABSTRACT

An interactive FORTRAN subroutine is presented which can be used via HELP Blocks in conjunction with ongoing, GPSS-based simulations to determine and collect the sample size needed to estimate the mean of a process with a specified level of statistical precision. The subroutine assumes an autoregressive representation for the random variable of interest, and estimates the variance of the sample mean by estimating the order and parameters of the autoregressive process. The statistical methodology followed by the subroutine is that developed and reported by Fishman [1,2]. Under interactive user control, the subroutine goes back to the GPSS model as often as may be necessary to extend sample size to the point that the user-specified level of statistical precision is achieved. An example illustrating use of the subroutine with a GPSS model is presented, and experience gained in using the technique is discussed. Source statements for the FORTRAN subroutine are provided, facilitating the relatively easy use of this autoregressive technique for the community of GPSS modelers.

I. INTRODUCTION

A frequent objective of simulation is to construct a confidence interval on the mean value of a process. For example, in a manufacturing context it may be of interest to construct a 95% confidence interval on the mean value of the random variable defined as elapsed time between order receipt and order shipment. When observations taken on the random variable of interest are independent, straightforward statistical techniques can be used to construct a confidence interval for the mean. It is typically the case in simulation modeling, however, that observed values of the random variable are not independent of each other but tend to be autocorrelated. For instance, the time required to complete the j-th order in the above example might influence the time required to complete the j+1-st order, etc. The presence of autocorrelation considerably complicates the process of confidence interval construction.

The complications of autocorrelation are so formidable, in fact, that simulation practitioners frequently resort to techniques (such as the method of replications, the method of batch means, and the regenerative method) which attempt to use the autocorrelated observations in an approximately independent manner. By their very nature, however, these techniques fail to exploit all of the information available in a series of observations on a random variable. As a result, larger sample sizes are usually needed to achieve a stated level of statistical precision than would be the case if the issue of autocorrelation were confronted directly. In consequence, simulation-supplied information is often considerably more expensive than it need be.

Methods for dealing directly with autocorrelated output from simulation models have been proposed and are reported in the literature. When these articles come to the attention of the typical simulation practitioner, however, they may appear to involve forbidding statistical methodology, and/or the time required to become familiar with and apply the resulting techniques may be judged to be prohibitively large. It is the purpose of the present paper, then, to make readily available to the typical GPSS model builder (and, for that matter, to users of other simulation languages as well), in a concise but nontechnical manner, a relatively sophisticated methodology for analysis of simulation output. This methodology uses autocorrelation information to construct a confidence interval on the mean value of a process. Statistical details are introduced only to the extent necessary to provide insight into the nature of the analysis, and set the stage for intelligent use of the methodology.

The paper presents an implementation of the methodology in the form of an interactive, user-friendly FORTRAN subroutine. Use of this subroutine is demonstrated in the context of a specific problem which has been modeled in GPSS. By studying these materials, then, the GPSS modeler should be able to understand and subsequently use the subroutine. Furthermore, the subroutine does not contain any features specific to GPSS models as such; this means it can also be directly used by those who model in FORTRAN, or in other languages which can call FORTRAN-based subroutines.

The specific statistical technique embodied in the subroutine is an outgrowth of the work first reported in [1], and later expanded and refined in [2]. This methodology assumes an autoregressive representation for the output from a simulation, and then builds a confidence interval for the mean of a process variable of interest, based on the estimated autoregressive model.

Section II of the paper explains the autoregressive model for simulated data in non-technical terms, and reviews the highlights in [1] and [2]. It also briefly discusses why a sophisticated model, such as an autoregressive model, may be necessary to correctly analyze simulated output data.

Section III then discusses the interactive approach followed by the subroutine (named AUTOR, for AUTO-Regressive), and indicates features of the program which make it easily usable. Then a specific problem is presented in Section IV and used to demonstrate how AUTOR can be coupled with a GPSS model addressing the problem. Because AUTOR is interactive and the user has to provide dynamic input settings for proper application, these settings and their influence on the performance of AUTOR are then discussed in Section V. An acknowledgement, references, and appendices then conclude the paper.

II. THE AUTOREGRESSIVE MODEL

The autoregressive model is given by

$$X(t) = B(1)X(t-1) + B(2)X(t-2) + \cdots + B(p)X(t-p) + k + E(t)$$

where X(i) is the i-th value observed for the random variable (i=t,t-1,t-2,...,t-p); p, B(1), B(2),...,B(p), and k are constants; and E(t) is a random disturbance or error term in effect for the t-th observation. Note that X(t) depends directly on the most recent p observations. The equation as written is called a p-th order autoregressive model because it includes p lagged terms in the variable X.

In the problem context which is to follow, the random variable of interest, X, will be a measure of cost, expressed in dollars per hour. For convenience in the discussion, X will consequently be referred to in these terms. And, because observations will be taken on X at the end of each hour, it can then be said that X(t) is the cost per hour observed during the t-th hour, X(t-1) is the cost per hour observed during the (t-1)-st hour, etc.

For purposes of demonstration, let us assume an autoregressive model with p = 2, B(1) = 0.8, B(2) = -0.2, and k = 2. (Of course, in an actual situation values for p, the B's, and k will not be known, but will have to be estimated from values outputted by a simulation model.) In addition, assume that the random disturbance term, E(t), is normally distributed with a mean zero and variance V^2, and that the disturbances are independent of each other from observation to observation. The assumption that E(t) is normally distributed is required for the hypothesis testing and confidence interval construction to be discussed later. However, the conclusions will still be accurate even if the random error term is only approximately normally distributed.

Using this assumed model, and letting $V^2 = 1$, we will generate a short sequence of values. Table 1 illustrates such a sequence with X(-1) = 5.00 and X(0) = 5.00. The entries in Table 1 show that the value of X(t) directly depends on the values X(t-1) and X(t-2). For example, the value in period 6 is directly dependent on the values in periods 5 and 4, plus a random disturbance whose value is specific for hour 6. Since the cost in hour 4 depends directly on hours 3 and 2, and so on, it is clear that the current value of the hourly cost random variable depends to some extent on all past values. The magnitude of this dependence does, however, decline with time. For example, if E(1) in Table 1 is changed from 0.91 to 1.91, the value of X(1) is affected considerably (changing from 5.91 to 6.91), but the value of X(6) is only affected slightly (changing from 5.19 to 5.20, with all of these changes shown in parentheses in the table).

It is important to emphasize that p, the B's, and k are assumed to be constant for a given process, and that the random error terms are independent and identically distributed from a normal distribution. These restrictions imply that the autoregressive representation being used here does not apply to observations taken from a model which is operating under transient conditions. In other words, the methodology implemented here applies only to observations taken from a steady state simulation. In statistical terms, this means that AUTOR is appropriate for autoregressive representations of stationary processes. The user of AUTOR must consequently take steps to guarantee that steady state conditions are in effect when AUTOR is brought into play. AUTOR checks a condition necessary but not sufficient for stationarity. This means that while AUTOR cannot guarantee stationarity, it can detect certain instances of non-stationarity. If and when non-stationarity is detected, AUTOR writes an appropriate message to the user, and returns a code to the simulation model signaling that the simulation is to be terminated.

Now consider the matter of representing the variable X as an autoregressive process. Such representation requires determination of p + 3 parameters [namely, the values of p, k, V^2, and B(1) through B(p)]. Three steps are involved in determining values for these parameters:

(1) Conduct one or more hypothesis tests to estimate the order of the autoregressive process.

(2) Estimate B(1) through B(p) and k.

(3) Estimate V^2.

The methodology underlying these steps is discussed in detail in [1] and [2]. Here, we limit ourselves to a brief review of the hypothesis testing used to estimate the autoregressive order of the process. For r = 0,1,2,...,pmax, the null hypothesis, H_0, is that the autoregressive order is r, whereas the alternative hypothesis, H_a, is that the autoregressive order is greater than r, but no greater than pmax + 1. The hypothesis testing procedure examines candidate autoregressive orders equal to 0, 1, 2, ..., possibly up through pmax, until H_0 is accepted (if at all). For example, if pmax is set at 10 and H_0 is rejected for r = 0 and 1 but accepted for r = 2, then the order of the autoregressive representation will be taken as 2 (i.e., p = 2). AUTOR

Table 1

Values Generated from the Autoregressive Model

$X(t) = 0.8X(t-1) - 0.2X(t-2) + 2 + E(t)$

with E(t) Normal(0,1), X(-1) = 5, and X(0) = 5

Time t	E(t)	X(t-2)	X(t-1)	X(t)
-1				5.00
0			5.00	5.00
1	0.91 (1.91)		5.00	5.91 (6.91)
2	-0.11	5.00	5.91 (6.91)	5.62 (6.42)
3	-1.72	5.91 (6.91)	5.62 (6.42)	3.59 (4.03)
4	1.42	5.62 (6.42)	3.59 (4.03)	5.17 (5.36)
5	-0.45	3.59 (4.03)	5.17 (5.36)	4.97 (5.03)
6	0.25	5.17 (5.36)	4.97 (5.03)	5.19 (5.20)
7	-1.71	4.97 (5.03)	5.19 (5.20)	3.45
8	0.62	5.19	3.45	4.34
9	1.45	3.45	4.34	6.23
10	1.18	4.34	6.23	7.30

queries the user for the value of pmax which is to be used in the hypothesis testing. With this one exception, the user has no ability to potentially influence the determination of the p + 3 parameters in the autoregressive representation.

Guidelines to the user for specifying a value of pmax are spelled out in Section V of this paper.

Incidentally, the hypothesis testing outlined above is conducted at a reliability level of 95%. This simply means that the probability of rejecting the null hypothesis H_0 when it is true is 0.05.

With p determined and values for the other parameters in the autoregressive representation subsequently computed by AUTOR, the next and last step is to build a confidence interval for the mean of the variable X. The average of the observed values (call it xbar) is the point estimate of the mean. Of course, building a confidence interval requires that the standard deviation of xbar be estimated. Under the (often invalid, and therefore naive) assumption of independent, identically distributed observations, the variance of xbar is simply estimated as the variance of the process divided by sample size (i.e., V^2/n), where the sample variance is used as an estimate of the unknown variance of the process. Applying this naive approach to the sample of size 10 shown in Table 1, and working with the process variance assumed there of 1 ($V^2 = 1$), the standard deviation of xbar would be computed as 0.316 [= SQRT(1/10)]. (Of course, in an actual situation we would have to estimate V^2; however, this example will provide a relative comparison between the naive approach, and the autoregressive approach.)

In the autoregressive approach, the variance of xbar is computed as $V^2/(n[1-B(1)-B(2)-\dots-B(p)])$, and so the resulting standard deviation for the Table 1 example is 0.50 [= SQRT(1/10[1-0.8+0.2])]. Now, comparing the autoregressive-based standard deviation of 0.50 against the naively computed standard deviation of 0.316, it is clear that the confidence interval constructed using the autoregressive approach will be wider than the one computed using the naive approach. This illustrates that if the naive approach is taken when the autoregressive approach is appropriate, a confidence interval can and usually will be reported that is more precise than warranted.

A little reflection shows that the naive approach is actually a special case of the autoregressive approach. If the first hypothesis tested (the hypothesis that r = 0) is accepted, then the autoregressive order is determined to be zero, meaning that the variables X(1), X(2), ..., X(t) are independent and identically distributed. In effect, this means that when AUTOR is used, the case of independent, identically distributed observations, if justified, will be handled properly. In conclusion, AUTOR handles iid (independent, identically distributed) observations if warranted, and at the same time eliminates the need on the part of the user to make a potentially damaging naive assumption in analyzing his or her simulation output.

III. THE SUBROUTINE AUTOR

The subroutine AUTOR, listed in Appendix A, is designed to carry on a running dialog both with a simulation model, and with an interactive user. The dynamics and some of the options involved in using the subroutine will now be sketched out.

The action starts with the simulation model, which can be thought of as a main program. (The main program discussed in the next section is a GPSS model but it could, of course, take other forms, such as that of a conventional FORTRAN main program, or of another FORTRAN subroutine). In general, the simulation model must move itself first of all through transient conditions and into a steady state of operation (with the simulated time required to bring this about, if any, having been determined by the modeler through earlier experimentation).

The main program then calls AUTOR which, detecting that this is the first call on it, requests that the user input values for (1) the confidence level which is to apply to the confidence interval on the mean; (2) the maximum autoregressive order which might potentially be examined; and (3) the number of observations to be taken initially on the random variable of interest. Control is then passed back and forth between AUTOR and the main program the number of times needed to collect this initial sample size. (Each time the main program has control returned to it by AUTOR, it proceeds with the simulation until one additional observation has been collected on the random variable, and then passes this value to AUTOR. Experience shows that this approach makes it relatively easy to convert a simulation model not originally developed for AUTOR use to one that can use AUTOR quite naturally.)

AUTOR then performs the autoregressive analysis on this initial sample, and reports relevant results, including a confidence interval, to the user. If the user is not satisfied with this confidence interval, (s)he can indicate what width interval is desired. AUTOR then reports to the user what the estimated additional sample size required to achieve this confidence interval would be. The user is then given three options: (1) terminate the simulation; (2) specify the amount by which the existing sample is to be supplemented by the taking of additional observations; or (3) provide a new specification on the desired width of the confidence interval, in response to which AUTOR will report the estimated additional sample size required to achieve this newly specified size of confidence interval.

If the user selects option 2, AUTOR then interacts accordingly with the main program to extend the total number of observations as requested, analyzes and reports out the statistics associated with the total sample, and again gives the user the three options indicated above. This back and forth process typically continues until the user chooses to terminate the simulation, or until AUTOR terminates the simulation because the total sample size has reached 15,000. (AUTOR cannot handle samples consisting of more than 15,000 observations, although this restriction could easily be relaxed by introducing modest changes in AUTOR.)

Other stopping conditions which might come about include the possibility that AUTOR detects non-stationarity in the sample, and the possibility that no autoregressive order within the range specified by the user is found to be acceptable. In this latter case, the user is given the chance to supply new values for the largest autoregressive order subject to testing (up to a maximum potential order of 25), and for the initial sample size in what will then be the first in a new series of simulations.

Whenever the simulation is eventually terminated, AUTOR passes control back to the main program, indicating that the simulation is not to be continued.

In terms of its underlying statistical methodology, AUTOR is identical to a subroutine presented in [2]. That is, AUTOR incorporates the same hypothesis testing procedure (for estimating the order of the autoregressive process), and the same computational techniques (to compute estimated values for the other autoregressive parameters). The subroutine in [2] is, however, set up for batch mode use. This means that its user is necessarily denied the opportunity to participate interactively in a sequential process whereby the characteristics of the random variable of interest (estimate of its mean; estimate of the variance of the sampling distribution of the mean; and estimate of the sample size required to construct a confidence interval with a given level of precision) are explored in ongoing fashion. The benefits of interactive exploration are self-evident. Perhaps of equal importance for GPSS users is that [2] does not indicate how a subroutine like AUTOR can be coupled with an ongoing GPSS simulation, as is done in this paper. Finally, AUTOR, as displayed in Appendix A, contains a comprehensive variable dictionary and has been documented liberally with comments.

It is judged that the preceding brief description of AUTOR, combined with the commentary contained in the source listing and further discussion of its use in the following section, will prove more than adequate in making it possible for interested persons to use AUTOR themselves.

IV. AN APPLICATION OF AUTOR

The following problem, which will be used to illustrate the use of AUTOR in context, is a modified version of a problem in [4].

"A certain materials-handling unit is used to transport goods between producing centers in a job shop. Calls for the materials-handling unit to move a load come essentially at random (i.e., according to a Poisson input process) at a mean rate of two per hour. The total time required to move a load has an exponential distribution with an expected time of 15 minutes. The total equivalent uniform hourly cost (capital recovery cost, plus operating cost) for the materials-handling unit is $20. The estimated cost of idle goods (waiting to be moved, or in transit) because of increased in-process inventory is $10 per load per hour. Furthermore, the scheduling of the work at the producing center allows for just 1 hour from the completion of a load at one center to the arrival of that load at the next center. Therefore an additional $5 per load per hour of delay (including transit time) after the first hour is to be charged for lost production. Using simulation, estimate the mean cost per hour and report a 95% confidence interval for this mean hourly cost."

A key decision which must be made in modeling the materials-handling system concerns the approach to follow in making observations on the cost-per-hour random variable of interest. Suppose it is agreed to make an observation at the end of each simulated hour. Two choices are then available for computing the cost incurred during the hour: (1) the cost could be based only on those jobs which have left the system during the hour; or (2) the cost could be based on all jobs which have been (and perhaps still are) in the system at any time during the course of the hour. Of course, total costs incurred in the long run will turn out to be the same either way. The alternative cost-accounting choices outlined above differ only in the timing with respect to which cost information is accumulated. Should a job in the system be viewed as contributing continuously to the value of the cost-per-hour random variable (choice 2 above), or should its total cost contribution be taken into account in one lump sum, at the time the finished job leaves the system (choice 1 above)?

Following either choice, the cost-per-hour random variable has one and the same expected value. Does it then matter which choice is made in deciding how to take readings on the random variable? Well, if one observational method has a smaller variance associated with it than the other, then it is the better method to follow. The reason is that the smaller variance leads to construction of a narrower confidence interval for a given sample size, other things being equal. Now, it is perhaps intuitively clear that costs accumulated continuously (method 2) will show less variation on an hour-to-hour basis than costs which are based only on jobs completed during the hour. For example, if a given job is resident in the system part of hour 1, all of hour 2, and part of hour 3, then by method 2 part of its delay cost will be accounted for in hour 1, another part in hour 2, and the remaining part in hour 3, whereas by method 1, instead of spreading out its delay cost across hours 1, 2, and 3, all of its delay cost will be taken into account in one lump sum in hour 3. This means that hourly costs observed under method 2 will be "smoother" (fluctuate less) than those observed under method 1. Although it would not be incorrect to follow method 1, method 2 will be adopted here, simply because its statistical characteristics are superior to those of method 1.

Appendix B shows a GPSS model which simulates the fundamental one-line, one-server queuing system described in the problem statement, which follows method 2 to take observations on the hourly cost random variable, and which uses the HELPB form of the HELP Block to call on AUTOR for analysis of output. (For an excellent, comprehensive discussion of HELP Block use in GPSS, see [3].) The model is liberally annotated, and should be readily understandable to the experienced GPSS user. The following comments might, however, be in order: (1) Cost accumulation per method 2 above is accomplished by using Savevalues which are updated and maintained in Model Segment 3 [lines 96-112]. Those unfamiliar with the approach used should consult section 5.11 in [5]. (2) The first call on AUTOR is made at the line 46 HELP Block. This call doesn't result in passing an observed value to AUTOR; it simply gives AUTOR the opportunity to obtain input values from the user. (3) The simulation proceeds 40 hours (24,000 time units, with the implicit time unit set at 0.1 minutes) to move through transient conditions and into steady state, then proceeds through one more hour before passing the first observation to AUTOR [see lines 113-129]. The value being passed to AUTOR is placed in the fullword, integer-type Savevalue HRCST, with the name of this Savevalue then being used in the B Operand position on the pertinent HELPB Block [line 125]. The Savevalue HRCST consequently is the "calling variable," and the corresponding "dummy variable" in AUTOR is the fullword, integer-type variable OBSER [see line 1, Appendix A]. (4) The GPSS model is written in GPSS/H, a GPSS implementation consistent with but superior to GPSS V [3]. Expressions have been used for convenience in Block Operand positions in lines 66, 74, and 77 in the model, a technique which is permissible in GPSS/H, but not in GPSS V. (To run the model under GPSS V, the only changes needed are to replace the expressions at lines 66, 74, and 77 with Arithmetic Variables.)

The beginning portion of a run made using the GPSS model coupled with AUTOR is shown in Appendix C. Information typed in by the user has been underlined (after the fact) in Appendix C to make it readily distinguishable from information typed out by AUTOR. The following features of the run can be followed by referring to Appendix C:

(1) The user specifies a confidence level of 95%, a maximum potential autoregressive order of 25, and an initial sample size of 100.

(2) AUTOR reports back a confidence interval on the mean hourly cost of [$21.09,$44.86], stating that a 2-nd order autoregressive representation has been fitted to the 100 observations. The autoregressive equation is:

$$X(t) = 1.23X(t-1) - 0.488X(t-2) + 8.34$$

(3) Dissatisfied with the wide confidence interval, the user then indicates that a confidence interval whose half-width is $1 is desired. AUTOR responds that this will require an estimated additional 14,020 observations to be taken.

(4) Backing off, the user then indicates that a confidence interval with a half-width of $2 would be of interest. AUTOR responds that this will require an estimated additional 3,430 observations to be taken.

(5) The user decides, conservatively, simply to have an additional 400 observations taken. The underlying thinking is that the initial sample size may have been too small to provide a very accurate estimate of the population variance (and this variance, in turn, is being used by AUTOR to estimate the number of additional observations needed to shrink the width of the confidence interval to a range which satisfies the user.)

(6) Based on what is now a total sample of 500 observations, AUTOR reports out a confidence interval of [$26.97,$32.45], with an associated 2-nd order autoregressive representation

$$X(t) = 0.838X(t-1) - 0.201X(t-2) + 10.78$$

In subsequent steps (not shown in Appendix C), the following resulted:

(7) The user specified a confidence interval half-width of $2. AUTOR recommended an additional sample size of 438.

(8) The user decided to have another 438 observa-

tions taken. Having taken these additional observations (bringing the total observations to 938), AUTOR reported out a confidence interval of [$28.20, $32.87], failing to satisfy the $2 half-width requirement.

(9) The user again specified a half-width of $2. This time, AUTOR recommended that an additional sample of size 343 be taken.

(10) The user decided to have another 343 observations taken. Having taken these additional observations (bringing the total observations to 1,281), AUTOR reported out a confidence interval of [$28.36, $31.80], and an associated autoregressive representation

$$X(t) = 0.839X(t-1) - 0.182X(t-2) + 10.31$$

Satisfied with this confidence interval, the user then told AUTOR to terminate the session.

The available analytic solution for the problem at hand indicates that the expected hourly cost is $30.68. The confidence interval reported in (10) above does cover the mean, as it would be expected to 95% of the time.

Of course, since AUTOR is interactive, there are many routes through the program, of which the above is only one example. For a given simulation model, the user is encouraged to experiment with a variety of initial settings and a variety of dynamic input values. The next section provides guidelines for the user in specifying such settings.

In the above sample use of AUTOR, it is instructive to compare the autoregressive confidence interval with that resulting from the naive approach (the approach which assumes independent, identically distributed observations). In the output from AUTOR corresponding to (10) above, the sample variance was reported to be 236.6 (not shown in Appendix C). Under iid assumptions, this value would have been used as an estimate of the population variance which, in turn, would have been used to construct a confidence interval of [$29.24,$30.93]. This iid-generated confidence interval is much more precise than warranted (given that the observations are positively autocorrelated, as they are), with corresponding undesirable implications in the broad context of decision making.

V. GUIDELINES FOR SETTING AUTOR PARAMETERS

As is evident from the Section IV discussion, there are five distinct parameters which AUTOR requests the user to provide:

(1) percentage confidence level applying to the confidence interval which is to be constructed;

(2) maximum order potentially to be considered for the autoregressive process;

(3) initial sample size;

(4) half-width for the confidence interval which the user sees him/herself as tentatively requiring;

(5) amount by which the current set of observations is to be extended in attempting to refine the current confidence interval.

Guidelines and restrictions for each of these parameters will now be discussed, per the above numbers.

(1) AUTOR only accepts confidence levels of 90%, 95%, and 99%. A 95% confidence level is frequently the standard chosen in statistical work. In the present context, a 95% confidence level implies that approximately 95 out of 100 confidence intervals constructed for the mean will cover (i.e., include) the true mean. Of course, other things being equal, a 99% confidence interval is wider than a 95% interval, and a 90% interval is narrower than a 95% interval.

(2) The maximum autoregressive order potentially to be considered (maxp) can be as small as 1, and as large as 25. In our experience, AUTOR has reported autoregressive orders as high as 9; however, in such cases the coefficients of the terms that are lagged by more than 2 [i.e., B(3), B(4), ...] have been very small.

If maxp is set too small, it may turn out that no autoregressive order will be accepted. Recall from earlier discussion that in such a case, AUTOR gives the user a chance to specify a new value for maxp (unless the maximum value which AUTOR can accommodate for maxp, 25, has already been specified by the user), and proceeds with a new series of simulations. It is suggested that nothing smaller than 5 be used for maxp.

What if maxp has been set to 25, and AUTOR still indicates that no autoregressive order is acceptable? In this case, it might be well to extend the sample size. If it continues to be true that no autoregressive order up to 25 is acceptable, then this simply means what it says: autoregressive representations through order 25 are not adequate to model the output from the simulation. In such a case, the modeler might choose either to experiment with higher potential autoregressive orders (which could be done, for example, by making some changes in AUTOR), or might switch to one of the alternative methods for confidence interval construction mentioned in Section I.

(3) AUTOR will accept any initial sample size up to and including 15,000. Initial sample size should be small enough to avoid unduly high simulation costs, and yet should be large enough to provide a fairly good initial estimate of the variance of the process. These two issues clearly can only be resolved in the context of the specific simulation model at hand. In any event, initial sample size probably should not be smaller than 50, because (a) the order of the autoregressive representation itself may be quite sensitive to sample size in the region of small sample sizes, and (b) for sample sizes that are too small, the estimate of the process variance can be poor to the point of leading to unrealistically large estimates of the additional sample size needed to achieve reasonable levels of statistical precision.

(4) As for the ultimate level of statistical precision which is to be achieved (i.e., width of the confidence interval), the user must determine, in context, what is reasonable and/or required in this regard. The cost of taking additional observations

must be balanced against the potential cost of having the information produced by the experimentation be of inadequate quality. Knowing the particulars of the problem itself, and of the context in which the problem is set, the user must exercise good judgment to determine what constitutes a satisfactory stopping point in the experimentation.

(5) The number of additional observations to be taken is limited only by the fact that the number of observations made in total cannot exceed 15,000. As indicated earlier, AUTOR estimates the number of additional observations needed to achieve the user-specified size of confidence interval. This "recommended sample increment" should be viewed by the user as an upper bound, especially if the total sample size on which the recommendation is based is small (meaning the estimate of the process variance may be poor).

One approach is to specify additional sampling amounting to only one half or even one third of the recommended sample increment [1]. An alternative approach is simply to extend the sample size repeatedly by a fixed amount, such as 100 or 500, until an acceptable level of statistical precision has been reached.

Experience leads to the recommendation that a mixture of these two strategies be used. Early on, sample size might simply be extended by relatively small, fixed amounts. Then, after the estimate of the process variance seems to have stabilized, something like 50% of the recommended sample size increments might be taken until a satisfactory stopping point has been reached.

No narrative as short as this can cover in fine detail all of the issues at hand. These guidelines should only be considered as a starting point. Use of AUTOR and its underlying methodology is urged, and comments on the resulting experiences are invited.

VI. ACKNOWLEDGEMENT

Eric L. Kintzer, MBA '78, The University of Michigan, programmed an early version of AUTOR as part of a seminar project. His work was taken as the starting point for our work.

VII. REFERENCES

[1] Fishman, George S., "Estimating Sample Size in Computing Simulation Experiments," Management Science, Vol. 18, No. 1 (1971), pp. 21-38.
[2] Fishman, George S., Concepts and Methods in Discrete Event Digital Simulation, Wiley-Interscience, 1973.
[3] Henriksen, James O., GPSS/H Users Manual, Wolverine Software Associates, Falls Church, VA 1978.
[4] Hillier, Frederick S., and Gerald J. Lieberman, Introduction to Operations Research, 2nd Edition, Holden-Day, 1974.
[5] Schriber, Thomas J., Simulation Using GPSS, John Wiley & Sons, Inc., 1974.

APPENDIX A: THE AUTOR SUBROUTINE

(Because the article would otherwise be too long, only the beginning and end of the AUTOR source listing can be shown here. A complete listing and source deck for AUTOR can be obtained from Professor Thomas J. Schriber, Graduate School of Business, The University of Michigan, Ann Arbor, MI 48109.)

```
      SUBROUTINE AUTOR(OBSER)
C    ------------------------------------------------------------
C    ---DICTIONARY OF PROGRAM VARIABLES---
C
C    B       - SAMPLE AUTOREGRESSIVE COEFFICIENTS;  B(R,S)
C              IS THE S-TH COEFFICIENT FOR THE R-TH ORDER.
C    BSUM    - SUM OF THE B'S FOR THE ORDER BEING TESTED.
C    CHI     - CHISQUARED CRITICAL VALUES, WITH ALPHA = .05
C              AND DEGREES OF FREEDOM = 1,2,3,...,25.
C    CHICRT  - THE SPECIFIC CRITICAL CHISQUARED VALUE BEING USED IN
C              THE HYPOTHESIS TEST TO DETERMINE THE A-R ORDER.
C    CHISTA  - COMPUTED CHISQUARED TEST STATISTIC.
C    CLPCT   - PERCENT CONFIDENCE LEVEL (REAL TYPE).
C    CON     - ESTIMATED CONSTANT IN AUTOREGRESSIVE EQUATION.
C    CONLEV  - CONFIDENCE LEVEL ASSOCIATED WITH CONFIDENCE INTERVAL.
C    CPL     - LOWER POINT OF CONFIDENCE INTERVAL.
C    CPU     - UPPER POINT OF CONFIDENCE INTERVAL.
C    DFCHI   - COMPUTED DEGREES OF FREEDOM FOR HYPOTHESIS TEST
C              ON AUTOREGRESSIVE ORDER.
C    DFT     - COMPUTED DEGREES OF FREEDOM FOR CONFIDENCE
C              INTERVAL T-STATISTIC.
C    HWGOAL  - USER SPECIFIED HALF-WIDTH OF CONFIDENCE INTERVAL.
C    HWNOW   - HALF-WIDTH OF CURRENT CONFIDENCE INTERVAL.
C    ICLPCT  - PERCENT CONFIDENCE LEVEL (INTEGER TYPE).
C    MAXP    - HIGHEST ORDER OF AUTOREGRESSIVE PROCESS CONSIDERED.
C    MAXPP1  - MAXP + 1. (MAXP = 1 => 2 ORDERS (0 AND 1) ARE TESTED,
C              ETC.)
C    MAXPP2  - MAXP + 2.
C    N       - INITIAL NUMBER OF OBSERVATIONS SPECIFIED BY USER.
C              LATER, THE TOTAL NUMBER OF OBSERVATIONS TO BE TAKEN.
C    NDIFF   - ESTIMATED NUMBER OF ADDITIONAL OBSERVATIONSREQUIRED
C              TO PRODUCE REQUESTED CONFIDENCE INTERVAL WIDTH.
C    NEGB    - NEGATIVE OF B VALUES.
C    NINC    - INCREMENTAL NUMBER OF OBSERVATIONS REQUESTED.
C    NOBS    - NUMBER OF OBSERVATIONS TAKEN TO DATE.
C    NREAL   - INITIAL NUMBER OF OBSERVATIONS REQUESTED.
C    NSTAR   - ESTIMATED NUMBER OF OBSERVATIONS REQUIRED TO
C              PRODUCE REQUESTED CONFIDENCE INTERVAL WIDTH.
C    NTOTAL  - TOTAL NUMBER OF OBSERVATIONS REQUESTED BY USER.
C    OBSER   - OBSERVATION PASSED FROM SIMULATION MODEL
C              TO AUTOR SUBROUTINE.  ALSO, CODE RETURNED TO SIMULATION
C              MODEL FROM AUTOR SUBROUTINE.
C    P       - ORDER OF A-R PROCESS DETERMINED BY HYPOTHESIS TESTING.
C    R       - SAMPLE AUTOVARIANCES R(0)...R(MAXQ).
C    RMAXP   - REAL-TYPE EQUIVALENT OF MAXP.
C    SIGNL1  - SIGNAL FROM USER TO AUTOR SUBROUTINE...
C              1.0 => SATISFACTION WITH CURRENT CONFIDENCE INTERVAL
C              0.0 => DISSATISFACTION WITH CURRENT CONFIDENCE INTERVAL
C    SIGNL2  - SIGNAL FROM USER TO AUTOR SUBROUTINE...
C              1.0 => TERMINATE THE RUN
C              2.0 => TAKE MORE OBSERVATIONS
C              3.0 => GET NEW USER-SPECIFIED HALF-WIDTH
C    SIGSQR  - SAMPLE RESIDUAL VARIANCES.
C    TAPX    - TERMS USED IN COMPUTING CRITICAL VALUE OF T-STATISTIC.
C    TSUM    - INTERMEDIATE VARIABLE USED IN COMPUTING DFT.
C    TVALUE  - COMPUTED T STATISTIC.
C    V       - INTERMEDIATE VARIABLES USED IN COMPUTATION OF B'S.
C    VXBAR   - ESTIMATED VARIANCE OF XBAR.
C    W       - INTERMEDIATE VARIABLES USED IN COMPUTATION OF B'S.
C    X       - VALUES OF THE STOCHASTIC SEQUENCE FOR WHICH A
```

```
      C            CONFIDENCE INTERVAL ON THE MEAN IS DESIRED.
      C   XBAR   - SAMPLE MEAN OF THE OBSERVED STOCHASTIC SEQUENCE.
      C   Z      - Z VALUE FOR SPECIFIC CONFIDENCE LEVEL REQUESTED.
      C   ZFLAG  - 0 => FIRST CALL ON SUBROUTINE; 1 => A SUBSEQUENT CALL.
      C   ZSQ    - Z SQUARED.
      C   ZVALS  - Z VALUES FOR CONFIDENCE LEVELS .99, .95, AND .90.
      C   ----------------------------------------------------------------
           INTEGER DFCHI , DFT    , I      , ICLPCT, IM1    , J      ,
          1        K      , MAXP   , MAXPP1, MAXPP2, N      , NDIFF  ,
          2        NMIP1  , NOBS   , NSTAR , NTOTAL, OBSER  , P      ,
          3        ZFLAG
      C   ----------------------------------------------------------------
           REAL B      , BSUM   , CHI    , CHICRT, CHISTA, CLPCT  ,
          1     CON    , CONLEV, CPL    , CPU    , HWGOAL, HWNOW  ,
          2     NEGB   , NINC   , NREAL  , R      , RMAXP  , SIGNL1,
          3     SIGNL2, SIGSQR, TAPX   , TSUM   , TVALUE, V      ,
          4     VXBAR  , W      , X      , XBAR   , Z      , ZSQ    ,
          5     ZVALS
      C   ----------------------------------------------------------------
           DIMENSION B(27,27), CHI(26), R(27)    , SIGSQR(27), TAPX(4),
          1          V(27)    , W(27)   , X(15000), ZVALS(4)
      C   ----------------------------------------------------------------
           DATA ZVALS/2.576,1.96,0.0,1.645/
      C   ----------------------------------------------------------------
           DATA CHI/3.8415, 5.9915, 7.8147, 9.4877,
          1        11.0705,12.5916,14.0671,15.5073,
          2        16.9190,18.3070,19.6751,21.0261,
          3        22.3621,23.6848,24.9958,26.2962,
          4        27.5871,28.8693,30.1435,31.4104,
          5        32.6705,33.9244,35.1725,36.4151,
          6        37.6525,38.8852/
      C   ----------------------------------------------------------------
           DATA ZFLAG/0/
      C   ----------------------------------------------------------------
      C
      C   ---TEST ZFLAG TO DETERMINE NATURE OF CALL ON SUBROUTINE---
      C
           IF (ZFLAG .NE. 0) GO TO 75
           ZFLAG = 1
      C
      C   ---COLLECT PARAMETERS FROM USER---
      C
           WRITE(6,10)
       10  FORMAT('0***ALWAYS USE A DECIMAL POINT WHEN'/
          1       '     ENTERING VALUES FOR THIS PROGRAM***'/
          2       '0ENTER THE PERCENT CONFIDENCE LEVEL AS'/
          3       ' 99., 95., OR 90.'/
          4       ' ?')
```

```
      C
      C   ---NO AUTOREGRESSIVE ORDER FOUND---
      C
       460 WRITE(6,470)
       470 FORMAT('0***NO A-R ORDERS FOR YOUR SETTING OF MAXP AND'/
          1       '     TOTAL SAMPLE SIZE WERE ACCEPTED AS ADEQUATELY'/
          2       '     REPRESENTING THE DATA.  NEW INPUT VALUES FOR'/
          3       '     MAXP AND INITIAL SAMPLE SIZE WILL NOW BE'/
          4       '     REQUESTED OF YOU***')
           GO TO 25
      C
      C   ---TERMINATE THE SIMULATION---
      C
       480 WRITE(6,490)
       490 FORMAT('0***SIMULATION TERMINATED***')
           OBSER = 1
           RETURN
           END
END OF FILE
```

APPENDIX B: A GPSS MODEL WHICH USES THE AUTOR SUBROUTINE

```
          SIMULATE
*
*       THE FOLLOWING STATEMENT PREVENTS THE OPERATING SYSTEM
*       FROM RE-LOADING A FRESH COPY OF THE AUTOR SUBROUTINE
*       EACH TIME IT IS CALLED.  THIS SAVES CPU TIME, AND ALSO
*       MEANS THE SUBROUTINE RETAINS DATA FROM CALL TO CALL.
*
*       SET INITIAL MULTIPLIER VALUES FOR RN1 AND RN2
*
          RMULT      12345,56789
*
*       DEFINE FUNCTIONS TO SUPPORT SIMULATING POISSON ARRIVALS
*       (EXPO1) AND EXPONENTIAL SERVICE TIMES (EXPO2)
*
 EXPO1 FUNCTION   RN1,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/.75,1.38
.8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2
.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
*
 EXPO2 FUNCTION   RN2,C24
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/.75,1.38
.8,1.6/.84,1.83/.88,2.12/.9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2
.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/.999,7/.9998,8
*
*     DEFINE VARIABLES TO...
*
*         ...CONTROL THE MOVEMENT OF A WAITING JOB (BV$DOUBL)
*         ...SAMPLE FROM THE SERVICE TIME DISTRIBUTION (V$STIME)
*         ...COMPUTE THE HOURLY DELAY COST (V$HRCST)
*
 DOUBL BVARIABLE  FNU$EQP+M1'E'600
*
 STIME FVARIABLE  150*FN$EXPO2
*
 HRCST VARIABLE   (10*X$NSUM+15*X$XSUM)/600+20
*
*     DEFINE A TABLE FOR TABULATION WITHIN THE GPSS MODEL ITSELF OF
*     THE RANDOM VARIABLE OF INTEREST (HOURLY DELAY COST)
*
 HRCST TABLE      X$HRCST,20,20,15
*
*     MODEL SEGMENT 1 (CALL ON SUBROUTINE AUTOR TO OBTAIN INITIAL
*                      VALUES FROM THE USER)
*
          GENERATE   ,,,1,3           BRING IN 1 TRANSACTION
          HELPB      AUTOR,HRCST      CALL AUTOR (THE ARGUMENT X$HRCST
*                                     HAS NO ROLE TO PLAY IN THIS CALL)
          TERMINATE                   TASK ACCOMPLISHED
*
*      MODEL SEGMENT 2 (HANDLING OF JOBS BY EQUIPMENT)
*
          GENERATE   300,FN$EXPO1,,,2 JOB (MATERIAL) ENTERS THE SYSTEM
          SAVEVALUE  NORML+,1         UPDATE COUNT OF JOBS IN
*                                     NORMAL-DELAY STATUS
          SPLIT      1,JUMPC          SEND SHADOW TO TRIGGER A SYSTEM
*                                     UPDATE IN 1 HOUR (IF JOB IS STILL
*                                     IN SYSTEM, ITS DELAY STATUS WILL
*                                     BE UPDATED AT THAT TIME)
          TEST E     BV$DOUBL,1       JOB WAITS UNTIL IT CAN GO INTO
*                                     SERVICE OR UNTIL ITS DELAY STATUS
*                                     GOES FROM NORMAL TO EXCESS
          GATE NU    EQP,JUMPB        SEIZE IF SERVICE CAN BEGIN NOW!
*                                     ELSE, GO UPDATE JOB DELAY STATUS
          SEIZE      EQP              CAPTURE THE EQUIPMENT
          ASSIGN     1,V$STIME        SET P1 = SERVICE TIME REQUIREMENT
```

```
          TEST GE      600-M1,P1,JUMPA   MOVE SEQUENTIALLY IF JOB STATUS
*                                        WILL REMAIN NORMAL WHILE JOB'S
*                                        ENTIRE SERVICE TIME ELAPSES
          ADVANCE      P1                ENTIRE SERVICE TIME ELAPSES
          RELEASE      EQP               FREE UP THE EQUIPMENT
          SAVEVALUE    NORML-,1          UPDATE NORMAL-DELAY JOB-COUNT
          TERMINATE                      FINISHED JOB LEAVES THE SYSTEM
*
    JUMPA ASSIGN       2,P1-(600-M1)     SET P2 = PORTION OF SERVICE TIME
*                                        WHEN JOB  WILL BE IN EXCESS-DELAY
*                                        STATUS
          ADVANCE      600-M1            NORMAL DELAY SERVICE TIME ELAPSES
          SAVEVALUE    NORML-,1          UPDATE NORMAL-DELAY JOB-COUNT
          SAVEVALUE    XCESS+,1          UPDATE EXCESS-DELAY JOB-COUNT
          ADVANCE      P2                EXCESS DELAY SERVICE TIME ELAPSES
          RELEASE      EQP               FREE UP THE EQUIPMENT
          SAVEVALUE    XCESS-,1          UPDATE EXCESS-DELAY JOB-COUNT
          TERMINATE                      FINISHED JOB LEAVES THE SYSTEM
*
    JUMPB SAVEVALUE    NORML-,1          UPDATE NORMAL-DELAY JOB-COUNT
          SAVEVALUE    XCESS+,1          UPDATE EXCESS-DELAY JOB-COUNT
          SEIZE        EQP               CAPTURE EQUIPMENT ASAP
          ADVANCE      V$STIME           SERVICE TIME ELAPSES
          RELEASE      EQP               FREE UP THE EQUIPMENT
          SAVEVALUE    XCESS-,1          UPDATE EXCESS-DELAY JOB-COUNT
          TERMINATE                      FINISHED JOB LEAVES THE SYSTEM
*
    JUMPC ADVANCE      600               JOB-SHADOW WAITS 1 HOUR
          TERMINATE                      JOB-SHADOW LEAVES SYSTEM, HAVING
*                                        TRIGGERED A SYSTEM UPDATE
*
*    MODEL SEGMENT 3 (INTEGRATES JOBS-IN-SYSTEM VS. TIME, AS A
*                        FUNCTION OF JOB DELAY STATUS)
*
          GENERATE     ,,24000,1,1,,F    SEED THE SEGMENT WHEN STEADY-
    LOOPA ASSIGN       1,X$NORML         SET P1 = NORMAL-DELAY JOB-COUNT
*                                        IN EFFECT AT END OF CURRENT
*                                        SYSTEM UPDATE
          ASSIGN       2,X$XCESS         SET P2 = EXCESS-DELAY JOB-COUNT
          MARK         3                 SET P3 = TIME OF CURRENT SYSTEM
*                                        UPDATE
          TEST G       MP3,0             WAIT FOR NEXT CLOCK DVANCE AND
*                                        ENSUING SYSTEM UPDATE
          SAVEVALUE    NSUM+,P1*MP3      UPDATE NORMAL DELAY INTEGRAL
          SAVEVALUE    XSUM+,P2*MP3      UPDATE EXCESS DELAY INTEGRAL
          TRANSFER     ,LOOPA            GO RECORD CURRENT DELAY STATES
*                                        PRIOR TO NEXT CLOCK ADVANCE
*
*    MODEL SEGMENT 4 (COMPUTES DELAY COST EXPERIENCED DURING HOUR
*                        JUST ELAPSED, AND PASSES THIS VALUE TO THE
*                        SUBROUTINE AUTOR)
*
          GENERATE     ,,24000,1         SEED THE SEGMENT WHEN STEADY
*                                        STATE BEGINS AFTER 40 HOURS
   LOOPB  SAVEVALUE    NSUM,0            SET NORMAL DELAY INTEGRAL = 0
          SAVEVALUE    XSUM,0            SET EXCESS DELAY INTEGRAL = 0
          ADVANCE      600               WAIT WHILE 1 HOUR ELAPSES
          SAVEVALUE    HRCST,V$HRCST     SET X$HRCST = HOUR'S DELAY COST
          TABULATE     HRCST             TABULATE THE HOUR'S DELAY COST
          HELPB        AUTOR,HRCST       PASS HOUR'S DELAY COST TO AUTOR
          TEST E       X$HRCST,1,LOOPB   LOOP IF AUTOR RETURN-CODE ISN'T 1
          PRINT        HRCST,HRCST,T     ELSE, PRINT GPSS-TABLED
*                                        DELAY-COST OBSERVATIONS
          TERMINATE    1                 STOP THE SIMULATION
*
*         CONTROL CARDS
*
          START        1,NP              START THE SIMULATION; SUPPRESS
*                                        DEFAULT PRINTOUT WHEN FINISHED
          END                            RETURN CONTROL TO OP SYSTEM
END OF FILE
```

APPENDIX C

INITIAL PORTION OF AN INTERACTIVE SESSION USING THE GPSS MODEL COUPLED WITH THE AUTOR SUBROUTINE

```
#EXECUTION BEGINS

***ALWAYS USE A DECIMAL POINT WHEN
   ENTERING VALUES FOR THIS PROGRAM***

ENTER THE PERCENT CONFIDENCE LEVEL AS
99., 95., OR 90.
?
95.

ENTER "MAXP", THE MAXIMUM
POTENTIAL AUTOREGRESSIVE ORDER
(1. <= ENTRY <= 25.)
?
25.

ENTER THE INITIAL SAMPLE SIZE
(50. <= ENTRY <= 15000.)
?
100.

   AUTOREGRESSIVE ANALYSIS PROGRAM INITIALIZED
   ===========================================

                     CONFIDENCE LEVEL =    95%
 MAXIMUM POSSIBLE AUTOREGRESSIVE ORDER =    25
        INITIAL NUMBER OF OBSERVATIONS =   100

            STATISTICAL RESULTS OF SIMULATION
            =================================

                          SAMPLE MEAN =       32.9800

                LOWER CONFIDENCE POINT =       21.0975

                UPPER CONFIDENCE POINT =       44.8625

                      SAMPLE VARIANCE =    0.7876E+03

     ESTIMATED VARIANCE OF SAMPLE MEAN =    0.2915E+02

                          SAMPLE SIZE =           100

         EQUIVALENT DEGREES OF FREEDOM =            11

              COMPUTED CRITICAL T VALUE =          2.20

                AUTOREGRESSIVE ORDER =             2

             SAMPLE RESIDUAL VARIANCE =    0.1864E+03


    ESTIMATED AUTOREGRESSIVE EQUATION...


       X(T) = 0.83415012E+01

            + 0.12356567E+01 X(T -  1)

            + -.48858285E+00 X(T -  2)
```

```
IS THIS CONFIDENCE INTERVAL SATISFACTORY?
IF YES, TYPE 1.0;  IF NO, TYPE 0.0
?
0.

ENTER THE DESIRED HALF-WIDTH
OF THE CONFIDENCE INTERVAL
?
1.

YOUR  CONFIDENCE INTERVAL HALF-WIDTH REQUIRES
APPROXIMATELY      14020 ADDITIONAL OBSERVATIONS.

AT THIS STAGE YOU HAVE THREE OPTIONS:
  (1) TYPE 1.0 TO STOP THE SIMULATION; OR
  (2) TYPE 2.0 TO TAKE MORE OBSERVATIONS; OR
  (3) TYPE 3.0 TO SPECIFY A (NEW) HALF-WIDTH
?
3.

ENTER THE DESIRED HALF-WIDTH
OF THE CONFIDENCE INTERVAL
?
2.

YOUR  CONFIDENCE INTERVAL HALF-WIDTH REQUIRES
APPROXIMATELY       3430 ADDITIONAL OBSERVATIONS.

AT THIS STAGE YOU HAVE THREE OPTIONS:
  (1) TYPE 1.0 TO STOP THE SIMULATION; OR
  (2) TYPE 2.0 TO TAKE MORE OBSERVATIONS; OR
  (3) TYPE 3.0 TO SPECIFY A (NEW) HALF-WIDTH
?
2.

ENTER NUMBER OF ADDITIONAL OBSERVATIONS
?
400.
           STATISTICAL RESULTS OF SIMULATION
           =================================

                         SAMPLE MEAN =      29.7120

               LOWER CONFIDENCE POINT =      26.9727

               UPPER CONFIDENCE POINT =      32.4512

                      SAMPLE VARIANCE =   0.2493E+03

     ESTIMATED VARIANCE OF SAMPLE MEAN =   0.1862E+01

                          SAMPLE SIZE =          500

         EQUIVALENT DEGREES OF FREEDOM =           51

             COMPUTED CRITICAL T VALUE =         2.01

                 AUTOREGRESSIVE ORDER =            2

            SAMPLE RESIDUAL VARIANCE =   0.1227E+03

   ESTIMATED AUTOREGRESSIVE EQUATION...

      X(T) = 0.10788321E+02

           + 0.83823466E+00 X(T -  1)

           + -.20133120E+00 X(T -  2)
```