LARGE-SCALE COMPUTER-AIDED STATISTICAL MATHEMATICS

Peter A. W. Lewis

Naval Postgraduate School
Monterey, California

November 1972

NPS55LW72111A

NAVAL POSTGRADUATE SCHOOL

Monterey, California

LARGE-SCALE COMPUTER-AIDED

STATISTICAL MATHEMATICS

by

Peter A. W. Lewis

November 1972

30

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral M. B. Freeman, USN
Superintendent

M. U. Clauser
Provost

ABSTRACT:

Some thoughts on large-scale computer-aided statistical mathematics (primarily simulation) which were presented at the 6th Annual Conference on the Computer Science/Statistics Interface conference are presented. Comments of participants and panelists (D. F. Andrews, J. N. Arvesen, D. P. Gaver, and G. Marsaglia) have been added to the original text.

Prepared by:

Peter A. W. Lewis
Department of Operations Research
and Administrative Sciences

Approved by:

J. R. Borsting, Chairman
Department of Operations Research
and Administrative Sciences

Released by:

J. M. Wozencraft
Dean of Research

NPS55LW72111A

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Large-Scale Computer-Aided Statistical Mathematics

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical Report

5. AUTHOR(S) (First name, middle initial, last name)

Peter A. W. Lewis

6. REPORT DATE November 1972	7a. TOTAL NO. OF PAGES 20	7b. NO. OF REFS 54
8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) NPS55LW72111A 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

Some thoughts on large-scale computer-aided statistical mathematics (primarily simulation) which were presented at the 6th Annual Conference on the Computer Science/Statistics Interface conference are presented. Comments of participants and panelists (D. F. Andrews, J. N. Arvesen, D. P. Gaver, and G. Marsaglia) have been added to the original text.

DD Form 1473 (PAGE 1)

S/N 0101-207-0811

Security Classification

A-31409

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computers						
Simulation						
Quantiles						
Jackknife						
Variance reduction techniques						
Sorting						
Statistical mathematics						
Multiprogrammed computers						
Percentiles						
Random number generation						
Multiplicative congruential generators						
Ordering						
Bias reduction						

LARGE-SCALE COMPUTER-AIDED STATISTICAL MATHEMATICS

Peter A. W. Lewis
Naval Postgraduate School
Monterey, California

Abstract

Some thoughts on large-scale computer-aided statistical mathematics (primarily simulation) which were presented at the 6th Annual Conference on the Computer Science/Statistics Interface conference are presented. Comments of participants and panelists (D. F. Andrews, J. N. Arvesen, D. P. Gaver, and G. Marsaglia) have been added to the original text.

1. INTRODUCTION

The aim of this paper is to stimulate discussion on large-scale computer-aided statistical mathematics (primarily simulation) at this conference. There has been a lot of discussion of computers and statistics (Hartley, 1972; Milton and Nelder, 1969; Chambers, 1970), but little of large-scale use of computers in simulation experiments to solve open distributional problems. This has perhaps been because of the unavailability of large computers and large amounts of computer time to research workers and statisticians. I think this will change rapidly over the next ten years as internal computation speed and the size of random access memories go up. The talk given by Dr. A. G. Anderson at this conference has amply illustrated this trend.

To take advantage of this availability, and to use the internal time sharing of central processing units inherent in multiprogramming, new statistical techniques which are computation-oriented will have to be developed. There is already growing impetus in this direction and this new technology coupled to the computers will make an enormous impact on statistics. I should note, too, that large-scale simulations are commonplace in industry and development laboratories, but the inefficiency of most of these computations is appalling.

There are recent surveys of several aspects of statistical computing (Hemmerle, 1967; Halton, 1970; Chambers, 1970; Freiberger and Grenander, 1971), most notably that by Tukey (1972b) who has been responsible for many of the new ideas in statistical computation. Consequently, I will only describe here the evolution of a computer program called COMPSTAT which was developed to try to use the IBM 360/91 computer at the IBM Research Center as efficiently as possible. The problems encountered in developing this program, some solved but others open, are more than enough for one paper.

*Sponsored by the Office of Naval Research through Contract NR 042-288 and the Foundation Research Program at the Naval Postgraduate School.

My interest in the problem of large-scale statistical computation grew from the frustration of trying to deal with non-normal time series, in particular, point processes (Cox and Lewis, 1966), and of having to write a book around the many gaps in the distribution theory. The first problem I tackled was the distribution of product-moment statistics (Lewis and Goodman, 1970), since one can, in principle, find recursion relationships to generate the distribution for successive sample sizes n. It took six months to verify the mathematics, six months to program it, and even then I wasn't sure enough of the programming to publish the results. I then turned to simulation, and quickly ran into several equally frustrating problems:

(1) Many procedures, notably variance reduction techniques, were very particular to the problem at hand and difficult to generalize. For example, a technique which works in estimating the mean of a distribution may not work when one is also interested in estimating the variance or higher moments.

(2) Most published statistical estimation (point and interval) techniques were "valid" asymptotically, and were prohibitively expensive in terms of number of operations (addition and multiplication) and memory cells required.

(3) Most "canned" routines were slow and generally unreliable.

(4) "Tooling up" took an excessive amount of time, and storing results, tabulating results and manipulating results was difficult.

It was therefore decided to look into the procedures and algorithms available, program them efficiently if they were useful, develop new techniques which were fast and economical of storage where necessary, and put them into a standard program which could be used for large scale simulations.

Several guidelines were set:

a) All procedures were to be computationally simple, use as little memory as possible and to be as broadly applicable as possible. In particular, this meant they should use as little information as possible about the statistic, say S, to be simulated. For example, one might not want to use the specific information that S was positive.

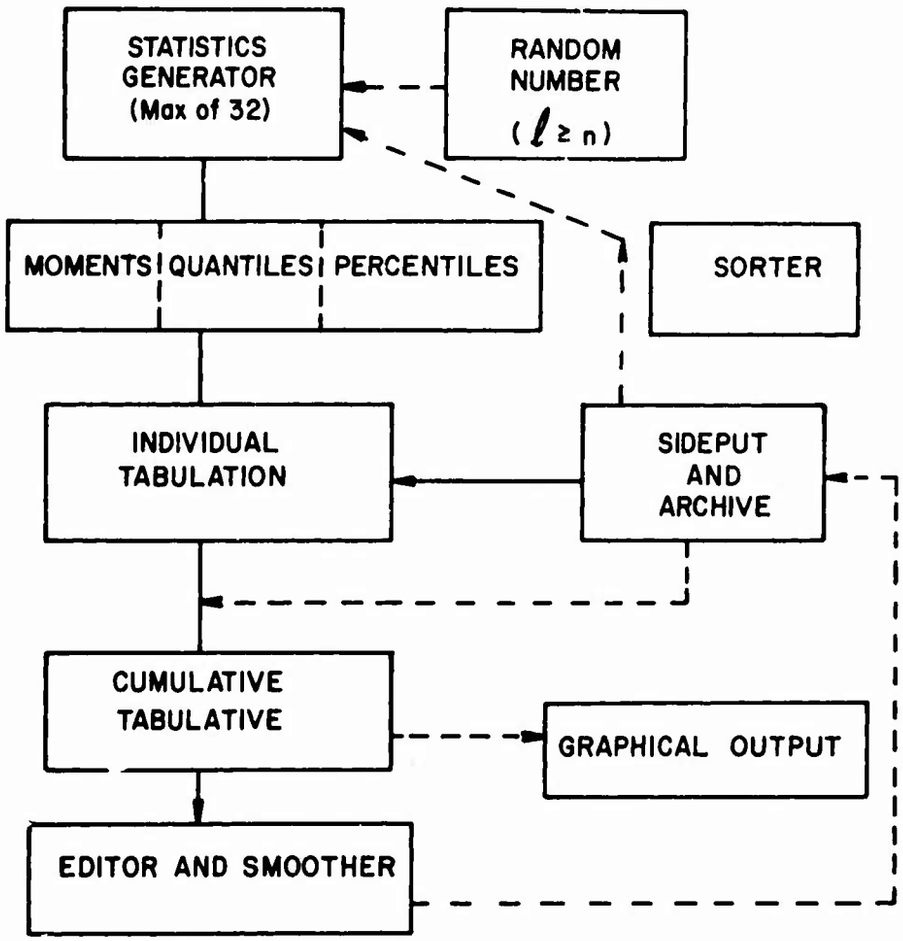
b) To utilize the speed of the computers, the best way seemed to be to compute the distributions of as many statistics as possible simultaneously.

c) Memory requirements should be kept fixed and relatively small in order to use excess CPU (Central Processing Unit) time by running in a lowest priority partition in a multiprogrammed invironment.

A block diagram of the overall program, COMPSTAT, which was developed is shown in Figure 1. We discuss this program generally before going into details of implementation and unsolved problems in later sections.

Referring to Figure 1, the symbol n is used to refer to sample size in statistical simulations, so that the statistic S might be the average of the observations in a random sample of size n. Any value of n may be used in the program, though it is written so as to repeat simulations on successive values of n if required. A number m of replications is specified by the user, with the option of splitting m into r blocks of size m' each ($m = rm'$). This is done to obtain estimates of the variance of estimates and also to allow for checkpoints to be taken.

On each replication the STATISTICS GENERATOR can call for $\ell \geq n$ random numbers, unsorted or sorted by magnitude. The user writes the STATISTICS GENERATOR, specifying up to 32 statistics (functions of the ℓ random variates). This has proved to be a very flexible arrangement; the statistics could be, for example,



FLOWGRAPH OF COMPSTAT PROGRAM

Figure 1.

a) the sample serial correlations of lags 1 to 32 in a series of random variables of length n;

b) the waiting times of 32 successive customers in a simulated queue;

c) an estimate of a parameter in a distribution, the jackknifed estimate of the parameter, the jackknifed variance, and the pseudo-values;

d) 32 points in the simulated spectrum of a time series of length n.

There are many other possibilities. For each of these statistics the user can specify that he wants estimates of the first four moments of S, 16 quantiles of the distribution of S, and 16 percentiles of S (or any combination of these three). Quantiles here is used to mean the solution x_α of the equation $\alpha = F_S(x_\alpha)$, where α is given and $F_S(x)$ is the distribution of S. A percentile is just $F(x)$ for given x. We assume the quantile exists.

	Mean	Stand. Dev	Lower Quantiles							
	$\tilde{\mu}$	$\tilde{\sigma}$	$\tilde{x}_{0.001}$	$\tilde{x}_{0.002}$	$\tilde{x}_{0.005}$	$\tilde{x}_{0.010}$	$\tilde{x}_{0.020}$	$\tilde{x}_{0.025}$	$\tilde{x}_{0.050}$	$\tilde{x}_{0.100}$
Normal (Exact)	11.982	2.562	6.989	7.197	7.512	7.789	8.113	8.229	8.642	9.165
Exponential	11.824	2.827	6.133	6.378	6.746	7.068	7.445	7.580	8.063	8.668
	(0.001)	(0.001)	(0.005)	(0.003)	(0.003)	(0.002)	(0.002)	(0.002)	(0.002)	(0.001)
1/2 Weibull	213.828	85.678	67.580	72.483	80.052	87.068	95.410	98.523	109.707	124.384
	(0.011)	(0.033)	(0.067)	(0.080)	(0.082)	(0.059)	(0.032)	(0.030)	(0.036)	(0.032)

	Skewness	Kurtosis	Upper Quantiles							
	$\tilde{\gamma}_1$	$\tilde{\gamma}_2$	$\tilde{x}_{0.900}$	$\tilde{x}_{0.950}$	$\tilde{x}_{0.975}$	$\tilde{x}_{0.980}$	$\tilde{x}_{0.990}$	$\tilde{x}_{0.995}$	$\tilde{x}_{0.998}$	$\tilde{x}_{0.999}$
Normal (Exact)	1.442		15.324	16.764	18.176	18.627	20.024	21.415	23.251	24.638
Exponential	1.061	2.120	15.514	17.064	18.532	19.061	20.552	22.045	24.055	25.557
	(0.003)	(0.023)	(0.004)	(0.005)	(0.005)	(0.006)	(0.009)	(0.009)	(0.024)	(0.033)
1/2 Weibull	1.557	5.082	323.012	373.912	425.816	442.749	496.882	553.934	634.068	700.534
	(0.003)	(0.035)	(0.084)	(0.059)	(0.136)	(0.172)	(0.182)	(0.328)	(0.472)	(1.696)

Table 1

Table 1 shows the form chosen to tabulate the results (moments and quantiles) of a simulation involving m replications for each n. These results are averages and sample standard deviations of the results of the r blocks of m' replications, all of this being stored in an archive which Tukey has aptly called the SIDEPUT. The estimated standard deviations of the estimates are given in brackets just below the estimates; below them we give (not shown) the estimated quantiles after subtraction of the estimated mean $\tilde{\mu}$ and division by the estimated standard deviation $\tilde{\sigma}$. This allows the experimenter to judge whether the statistic is approximately normally distributed.

The last blocks in Figure 1 allow for CUMULATIVE TABULATION on n, EDITING and SMOOTHING of the results (including rounding and printing tables for publication), and GRAPHICAL OUTPUT as shown in Figures 2 & 3. It is easy to see in the figures that this statistic is not normally distributed and is converging very slowly with n to the asymptotic ($n \to \infty$) distribution. The positive skewness of the distribution is also evident.

An original, rather inefficient, COMPSTAT program was used to implement a study of tests of independence in point processes. Twenty statistics were computed simultaneously on an IBM 360/91 in a 120K partition. Some of these results have been published (Lewis, 1972) and are partially reproduced here (Figures 2 and 3); others will appear later.

A study on a similar scale of robust estimates of location was undertaken at Princeton (Andrews, et al, 1972); they had the advantage over me of both manpower and expertise.

It is hoped to rewrite the COMPSTAT program at some later time in order to incorporate all of the recent advances in statistical computing technology described below.

2. DETAILS

We discuss now the details of the implementation of a program such as COMPSTAT. At its inception in 1966 we quickly ran up against the lack of real

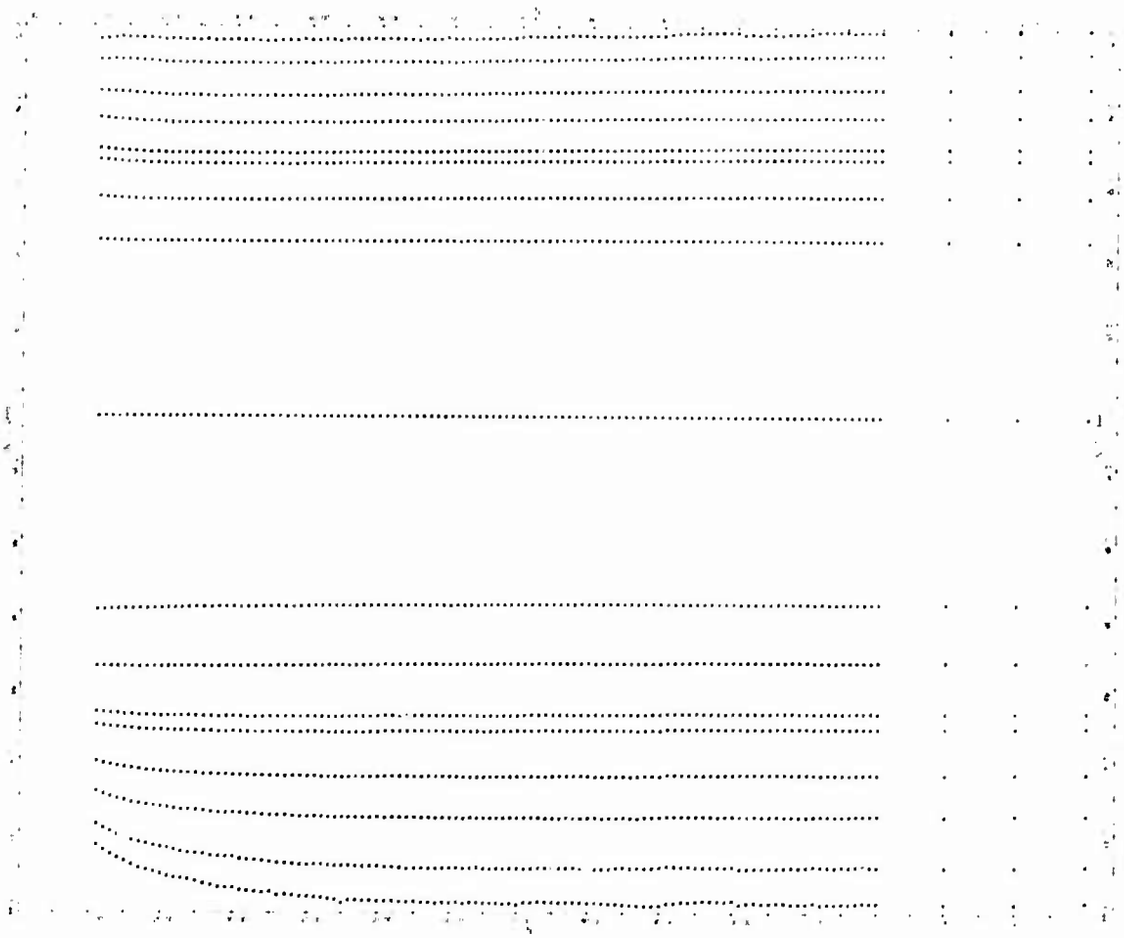


Figure 2

computational considerations in many standard statistical procedures. The situation is better at present, with books such as Hemmerle (1967) and Knuth (1969) now available. Knuth (1969) in particular is invaluable. There are still many problems, however, particularly relating to large-scale computations.

a) Random Number Generation

Clearly the statistical quality of the random numbers available for large-scale simulations will be the limiting factor in how far one can go in utilizing large-scale computers in simulations. In 1966 the main generator in use was RANDU in the IBM SSP package. It is still widely used today, by default, even though it is known to knowledgeable users to have poor statistical properties. There are no published test results on RANDU, except one brought to my attention at the conference (Bates and Zirkle, 1971) but there are papers published on problems which have been encountered with its use. Moreover, as a statistical consultant one comes up against many cases in which strange results in simulations are remedied by replacing RANDU by another random number generator.

In this respect it might be noted that if statisticians are guilty of ignoring computational aspects of their procedures, computer scientists are equally guilty of ignoring the statistical aspects of algorithms. There are hundreds of clever random number algorithms in the literature

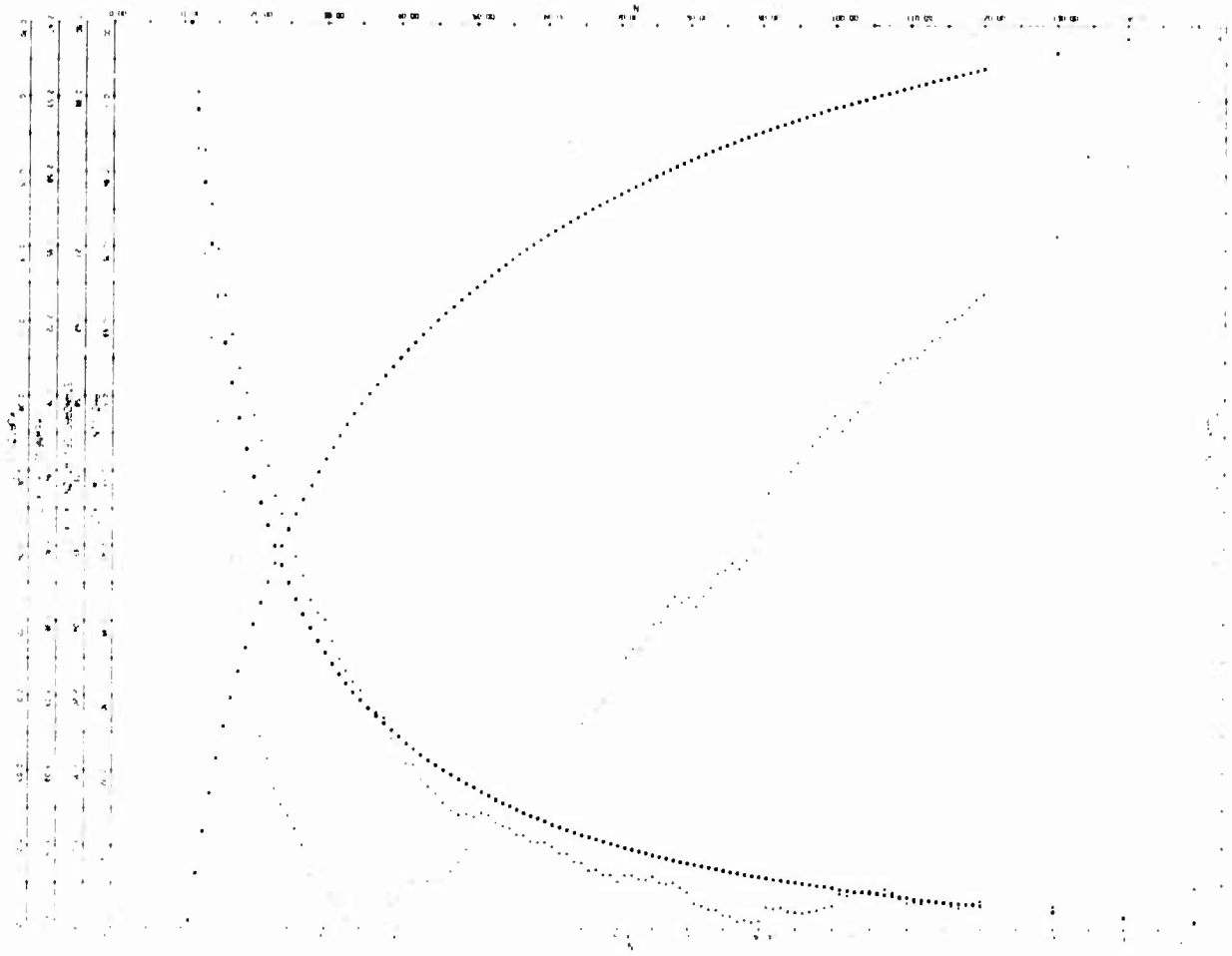


Figure 3

[see the bibliography by Nance and Overstreet, 1972] but there are virtually no accompanying test results published. Moreover, it is generally difficult to do so in a computer journal.

In 1966 we started to investigate the problem of random number generation for large-scale computation, and after extensive testing we developed a 31-bit pseudo-random number generator for the System 360. This is a multiplicative congruential generator of the form (Lewis, Goodman, and Miller, 1969)

$$x_{i+1} \equiv Ax_i \pmod{p},$$

where p, a prime, is $2^{31} - 1$ and $A = 7^5$ is a positive primitive root of p, thus guaranteeing a cycle of length p for the generator. Another advantage of using a positive primitive root for the multplier is that low order bits are also random. Beside the assembly language version given in the paper, a very fast version of this generator which generates arrays of integer or floating point numbers is available in the new IBM SL/MATH package. We will refer to this as the GGL generator; it generates a random number in 1.40 μsecs on a 360/91 and in 16.00 μsecs on a 360/67. A version has been written for the 360/67 at the Naval Postgraduate School using a division simulation algorithm due to Lehmer (see Payne, Rabung, Bagyo, 1969; Liniger, 1961).

Test results for GGL are given in Lewis, Goodman, and Miller (1969) and extensive subsequent use turned up no obvious problems, though constant care was exercised. For example, in Table 1 the maximum periodogram values with 1/2-Weibull distributed variates (squares of exponentially distributed variates) are very large. In this simulation $r = 6$, $m' = 750{,}000$ and $m = 4{,}500{,}000$. As a check, normal deviates were used in a very large simulation and no discrepancy from the exact distribution (shown in Table 1) even at 0.999 quantiles was found. This GGL random number generator was used in the Princeton study (Andrews, et al, 1972) and is used in APL.

Nevertheless, valid doubts continue to be expressed about the use of the GGL generator in large-scale computations, particularly in light of Marsaglia's results (Marsaglia, 1968, 1972) on the structure of sequences of numbers from congruential generators. These results have shed a lot of light on problems which can be encountered with congruential generators, but I don't believe they tell the whole story. There are also new types of generators being advocated. Some of these are too cumbersome for consideration, but others are popular, in particular the Tausworthe or shift register generators (Tausworthe, 1965). Some doubt has been cast on the statistical properties of these shift register pseudo-random number generators recently; my own preference is, for speed and simplicity, to go to shuffled congruential generators (Marsaglia and Bray, 1968). Tukey (1972) ascribes this idea to Gentlemen but it seems quite old and was put forward by Marsaglia in the early 1960's. I have found no documentation of the statistical properties of shuffled generators, although they are intuitively appealing.

We have undertaken further statistical tests of some of the above generators at the Naval Postgraduate School. In particular, we have been interested in correlating test results with results of Couveyou and MacPherson, 1967; (see also Knuth, 1969, pp. 82-100.) and Marsaglia, 1972. It seems to me that the Couveyou-MacPherson Fourier analysis is the best analytical tool for predicting performance of random number generators that has appeared. Marsaglia's results [1972] on the lattice structure of the congruential generators are also useful.

The tests referred to started with the GGL random number generator, RANDU and a TAUSWORTHE generator (Tausworthe, 1965) and the runs test. As in Lewis, Goodman, and Miller (1969), runs of length eight or longer are pooled and a Chi-square statistic computed. Nominally, this has a Chi-square distribution with 7 degrees of freedom and we denote it as χ^2_7. Table 2 gives summary statistics on 20 runs of 2^{16} numbers each for the three generators. The Tausworthe generator is now analytically known to have poor runs performance (Toothill, Robinson, and Adams, 1971). The runs test rejects neither GGL nor RANDU if the test statistic is assumed to be distributed as a Chi-square variate with 7 degrees of freedom. As mentioned before, RANDU is known to be poor; this is shown in Table 3 giving the Couveyou-MacPherson wave numbers for GGL and RANDU for dimensions up to 7. It is in higher dimensions that RANDU is particularly poor*

Table 2. Runs test; Chi-square statistic.

	χ^2_7	$\hat{\sigma}_{\chi^2_7}$
GGL	6.846	4.084
RANDU	7.939	3.502
TAUSWORTHE	9.972	10.428

Table 3. Wave numbers for two generators.*

Dimension	GGL	RANDU
2	16,807	23,172
3	638.9	10.86
4	146.25	10.77
5	67.21	10.77
6	29.92	
7	16.55	

* I am indebted to Dr. L. R. Turner, NASA Lewis Research Center, Cleveland, Ohio, for these numbers.

A comparison of the three samples of size 20 using a two-sample Kolmogorov test rejects the hypothesis that any of them are from the same distribution! This is a sad result for large-scale simulation, particularly if one were trying to simulate the distribution of the Chi-square summary statistic, χ^2_7, for the runs test.

The three generators have been shuffled and the Chi-square statistics of 100 tests for samples of size 2^{16} numbers from each generator were found to be distributionally commensurate. The shuffled Tausworthe generator was still suspect, however.

Results of this testing will be given elsewhere; other statistical tests are being evaluated. The conclusions so far are interesting. There is mild evidence that shuffling helps. The main conclusion seems to be, however, that the runs test is virtually useless. (Note that Bates and Zirkle accept RANDU, partly on the basis of runs tests.) And although recent books (Newman and Odell, 1971; Maisel and Gnugnoli, 1972) tout the runs test as amongst the best, I have been unable to find any documentation for this. It seems to be an example of a stochastic rumor to which I too have contributed (Lewis, Goodman, and Miller, 1969). Perhaps some readers can guide me to work on the power of the runs test; Lehman (1959, p. 155) points out that a modified runs test has certain optimum properties in testing for independence in a binary sequence against 1st-order Markov alternatives. Even results on the power of the runs test relative to the serial correlation test for first order normal autoregressive schemes would be of interest.

There is clearly much work to be done in random number generation. I have also not mentioned the need for efficiently generated, reliable normally distributed random variates. These, and perhaps several other kinds of random deviates should be provided as primitives, just as random numbers are provided as primitives in APL. A package to generate normally and exponentially distributed random variables is available from Marsaglia at McGill University. It uses some of Marsaglia's own methods and is very fast. A survey of some of these methods is given by Ahrens and Dieter (1972).

b) Ordering.

Ordering (sorting) of quantities, or obtaining ranks, is a basic operation in statistical computation; a survey is given by Martin (1971). The main use we had for it initially was in quantile estimation, and here it was a bottleneck since, in general, ordering of n quantities takes a number of operations proportional to $n(\ell n\ n)$ and memory capacity proportional to n. The quantile estimation problem is discussed below; the use of ordering is now used mainly in COMPSTAT in generating statistics such as the median. There are several points to be made here.

(i) Uniformly distributed random variates can be ordered by address modification schemes (Isaac and Singleton, 1956) in time proportional to n, although for large computations $3n$ memory positions are needed to avoid overflows. An algorithm for this type of sorting is provided in COMPSTAT.

(ii) It is clear that by using pilot estimates of a non-uniform distribution, address modification schemes can be used on any data. These take, asymptotically, n operations, but for reasonable sample sizes the procedure is slow and cumbersome programming-wise. The scheme is due to Floyd at Stanford. There is renewed interest in this area and Chambers (1971) has a scheme for partial sorting which is more efficient than an $n(\ell n\ n)$ sort. Andrews (personal communication) also has a scheme for obtaining the median; it uses a pilot estimation scheme and is subject to overflows which could be a problem in large-scale simulations.

(iii) Schemes using the Markov property of the gaps (differences between successive order statistics) (see David, 1971, p. 17) are available for producing ordered uniform variates (Schucany, 1972; Lurie and Hartley, 1972).

We had tried in COMPSTAT an equivalent scheme based on the independence of the gap statistics for exponentially distributed variates. These schemes for moderate n are more time consuming than the $n(\log n)$ schemes, but are more efficient in use of memory space. Their primary use would seem to be when only a few of the low or high order statistics are needed. Two points should be made here:

1. It is computationally easier to generate high order statistics, rather than the low order statistics advocated by Lurie and Hartley (1972) and Shucany (1972). Denoting the uniform variates by U_i and the ordered uniform variates by $U_{(i)}$, we have

$$\text{prob}\left\{U_{(i)} \le u_{(i)} \mid U_{i+1} = u_{(i+1)}\right\} = \left\{\frac{u_{(i)}}{u_{(i+1)}}\right\}^i$$

$$(i=1,2,\ldots n;\ u_{(i)} \le u_{(i+1)},\ u_{n+1} = 1)$$

If only low order uniform order statistics are required, they are generated as $U'_{(i)} = 1 - U_{(n+1-i)}$.

2. The time consuming operation in the above is to take the $(1/i)$th power. This is done, usually, using logarithms and the scheme is then equivalent to generating order statistics from a unit exponential distribution. However, since it is much faster to generate exponential variates using some of Marsaglia's sampling procedures than it is to generate them by taking the logarithm of a uniform variate, it is faster to generate ordered uniform random numbers by starting with exponential variates.

The basis for this is that if $E_{(i)}$, $i = 1, 2, \ldots n$, denotes ordered unit exponential variates from a sample of size n, and we let $E_{(0)} = 0$, the gap statistics (Cox and Lewis, p. 26-27)

$$D_{(i)} = E_{(i)} - E_{(i+1)} \qquad (i = 1, --, n)$$

are independent exponentials with mean $E(D_{(i)}) = (n+1-i)^{-1}$. Thus if we have n unit exponentials, generated say by one of Marsaglia's schemes, we generate

$$E_{(i)} = \sum_{j=1}^{i} (n+1-j)\, E_j \qquad (i = 1, \ldots n)$$

and

$$U_{(i)} = 1 - \exp\{E_{(i)}\} \qquad (i = 1, \ldots n).$$

An $n(\log n)$ sorting and ranking scheme is also provided in COMPSTAT, for sorting and ordering within the STATISTICS GENERATOR.

c) Quantiles and Percentiles.

Estimating quantiles was the second biggest bottleneck in implementing COMPSTAT. Quantiles are more basic in characterizing distributions than percentiles, although, for example, one is interested in percentiles when evaluating by simulation the power of a test based on a statistic S. Thus, given the α-quantile x_α of S under a null hypothesis, one wants the percentile corresponding to S and x_α under a different hypothesis.

Percentile estimation as a binomial process is essentially straightforward and ideal by our criterion of simplicity and economy of computation and memory requirements. It is also unbiased. However, it appears that greater efficiency should be obtained by coupling estimates at different x_α's, although I haven't been able to do so. Most schemes appear to require assumptions about boundedness of the probability density function. Somerville (1970) has some results in this area; it appears to be an area for further research.

Quantile estimation based on order statistics is advocated in most texts (see David, 1971). For large-scale computation the sorting time required and the memory capacity is prohibitive. Stochastic approximation schemes (Robbins and Monro, 1951; Hodges and Lehman, 1956) were then tried but found to converge at an impossibly slow rate for large quantiles. These two quantile estimation schemes are prime examples of statistical procedures which are not attuned to computing realities, and whose asymptotic properties are deceptive as far as practical applications are concerned.

A solution was finally found (Goodman, Lewis, and Robbins, 1972) which combined the stochastic approximation with a data transformation. Typically

if the α-quantile was required ($\alpha>0.5$), the maxima of successive groups of size v of realizations of S are found. The problem is then one, if $\alpha' = \alpha^v$, of finding the $x_{\alpha'}$ quantile, which is equal to x_α, in a distribution which is the v^{th} power of the distribution S, $F_S(x)$. By taking v large enough to make $\alpha' \approx 1/2$ the problem becomes one of estimating a median, although other values of v can be used. Stochastic approximations work well with medians, but as the bias is apparently of order $m^{-1/2}$, jackknifing is required to reduce the bias.

The present scheme (Goodman, Lewis, and Robbins, 1972) based on the maximum transformation and stochastic approximation solves the basic problems of quantile estimation, but research is continuing to improve it. Computationally it is very good since finding a maximum requires only two memory cells and computation time is linear in m, the number of realizations of S which are generated. It is also simple to compute in parallel the quantiles for several levels, e.g. α = 0.990, 0.995, 0.999.

D. Salsburg has raised the question as to whether one wouldn't want to order the data anyway to do, for instance, a normal probability plot of the simulated distribution. This may be true for samples of size m equal to about 500; beyond that the sorting in a large-scale simulation becomes onerous, time-wise and memory-wise, and I feel a plot using 16 quantiles, as in Figure 2, plus the moments in Figure 3, is as good as or better than a full probability plot.

d) Bias and Bias Reduction.

It is essential for sensible and interpretable simulation results to have estimates of the variances of the simulated quantities. However, sectioning the m replications in a large-scale simulation into r sections of m' replications to estimate the variance of estimates (see Mosteller and Tukey, 1968) brings in problems of bias. This is because one wants r to be about 10 to get reliable estimates of the variance, but the resulting m' may be too small to reduce the bias in the simulated quantity to acceptable levels.

This problem seems to be well in hand because of the jackknife technique for bias reduction which was developed by Quenouille (1956), pushed by Tukey (1958) and generalized by Schucany, Gray, and Owen (1971) and Gray and Schucany (1972). A similar technique was used by Gaver and Hoel (1970) in examining small-sample Poisson probability estimates. Some price may be paid in inflation of the variance of the estimator (Miller, 1964; also Goodman, Lewis, and Robbins, 1972, for a specific case).

In COMPSTAT the jackknife is quite simply incorporated into the STATISTICS GENERATOR.

e) Variance Estimation.

The problem of bias appears to have been alleviated directly by the jackknife, and indirectly because of a suggestion by Tukey (1958) that the sample standard deviation based on the pseudo-values in the jackknife procedure be used to estimate the variance of the jackknifed estimate. There is some evidence that this procedure is broadly applicable, although Miller (1968) pointed out cases where it can give poor results. In general, n-fold jackknifing in a small sample of size n can give an estimate with a very inflated variance, though this problem disappears as $n \to \infty$. Relevant references are Arvesen (1969), a review by Arvesen and Salsburg (1972), and Mosteller and Tukey (1968).

The jackknifing procedure will probably be most useful when available computation time is too short for sectioning. For a description of variance estimation techniques based on sectioning, see Mosteller and Tukey (1968).

f) Variance Reduction Techniques.

I have not discussed variance reduction techniques so far. An excellent review is given by Gaver (1969); see also Hammersley and Handscomb (1964). These variance reduction techniques can be implemented in COMPSTAT but there seem to be several drawbacks, mainly that the methods are particular to the problems at hand. Thus, a large amount of time can be spend deriving, say, an antithetic

variate for a particular problem and this may, when large computers are available, be an inefficient way to use statisticians.

The most important drawback to most methods, however, is that a method that reduces the variance of an estimate of the mean of a statistic S will often inflate the variance of an estimate of the variance of S. This is clearly true for many antithetic variate techniques (Hammersley and Mauldon, 1956) and would be worse when quantiles or percentiles are also required. This may be all right in nearly normal situations, but not in others.

Much more research is required on variance reduction techniques that are applicable to all aspects of the characterization of a distribution, and are easily derived. Control variable techniques (Fieller and Hartley, 1959) seem to me the best candidate for this role.

An empirical control variable technique can be implemented with COMPSTAT when exploration is required around a null situation. This may, for instance, be a test of hypothesis in which power against small deviations is of interest. Again, small variations in scheduling algorithms in complex queues might be of interest to see what improvement they make to, say, throughput time.

One might then do a very precise simulation of the characterizations of the statistic under the null hypothesis. Fix m', the number of replications per section, and let r, the number of sections, be large and denote by $\bar{\theta}_0(r)$ the estimated quantity under the null hypothesis. This will be the average of the estimates of θ_0 from the r sections. Results for the sections are kept in the SIDEPUT, together with the seed for the random number generator which initiates each section of the simulation. The quantity is estimated under alternative conditions using the same random numbers using only r' sections, where $r' << r$. Call this quantity $\bar{\theta}_\varepsilon(r')$. If $\bar{\theta}_0(r')$ is the null (average) estimate from the first r' sections, $\bar{\theta}_0(r-r')$ the null (average) estimate from the last $r - r'$ sections, then the control variable estimate is

$$\tilde{\theta}_\varepsilon(r') = \bar{\theta}_\varepsilon(r') - \bar{\theta}_0(r') + \bar{\theta}_0(r)$$

$$= \bar{\theta}_\varepsilon(r') - \left(\frac{r-r'}{r}\right)\bar{\theta}_0(r') + \left(\frac{r-r'}{r}\right)\bar{\theta}_0(r-r').$$

Then

$$\mathrm{var}[\tilde{\theta}_\varepsilon(r')] = \mathrm{var}[\bar{\theta}_\varepsilon(r')] + \left(\frac{r-r'}{r}\right)^2 \mathrm{var}[\bar{\theta}_0(r')] -$$

$$\left(\frac{r-r'}{r}\right)\mathrm{cov}[\bar{\theta}_\varepsilon(r')\,\bar{\theta}_0(r')] + \left(\frac{r-r'}{r}\right)\mathrm{var}[\bar{\theta}_0(r-r')].$$

The common random numbers used to generate the estimates should make the estimates $\bar{\theta}_\varepsilon(r')$ and $\bar{\theta}_0(r')$ highly correlated, and the above equation is the variance in the usual control variable situation except for the last term. If r is large relative to r' this last term should be small relative to the other terms.

It is possible to use subsequent sections of size r' in the original simulation of θ_0 to explore other alternatives, say θ_{ε_1}, θ_{ε_2}, ... There are interesting design and analysis problems in this scheme which will be explored elsewhere.

One final point should be made here about control variables. Let $\bar{\theta}$ be the uncontrolled estimate and $\tilde{\theta}$ the controlled estimate (generated from the same random numbers). It is not often realized that even with a regression adjusted control (see Gaver, 1969) the maximum attainable variance reduction is

$$\frac{\mathrm{var}(\tilde{\theta})}{\mathrm{var}(\bar{\theta})} = 1 - \rho^2,$$

where ρ is the correlation between $\tilde{\theta}$ and $\bar{\theta}$. It can be very difficult and time-consuming, especially for the inexperienced practitioner, to find a control which gives a high enough ρ to justify the pratitioners time. And in many cases equivalent speed ups can be achieved by using more efficient random number generators, ordering routines, etc.

The time factor to achieve a high ρ is one reason for putting forward the empirical scheme above.

g) Planning Simulation Experiments.

The empirical control variable suggestion in the previous section brings up the whole question of

the design of simulation experiments. Thus, it would be reasonable to use the empirical scheme, or plan an experiment around the null value ξ_0? This would be appropriate if the range of parameters of interest were known in advance. The empirical control variable technique seems attractive as an on-line, interactive procedure, especially when estimates of the variances of the estimates are available, as in COMPSTAT. Some formal analysis is still needed and this could be formidable.

In general, it would seem that the output of large-scale simulation would be a fertile field for application of techniques of analysis of variance and experimental design. I am not familiar with much by way of specific applications; several recent books, including that by Mihram (1972), which I have not examined carefully, do treat analysis of simulation experiments. The tendency, however, does seem to be to just regurgitate the old theory without specifically worrying about particular problems of simulation experiments.

A simple case occurs when an experimenter has two variance reduction techniques available, say two control variables, and a fixed number of replications m he can perform. He wants to choose the control variable which minimizes the variance of the final estimate of a parameter, say θ, which could be the mean of a statistic S. If m' is large enough so that the estimates in each of the r sections ($rm'=m$) are unbiased and normally distributed, this is a classical two arm bandit problem.

I know of no one, however, who has actually done this, probably because the benefit of reduced variance doesn't outweigh the extra cost of tooling up for two estimates of θ. It could be feasible with COMPSTAT. Once more than one parameter is involved, say the mean and variance of S, the problem is much more complicated. In general I think, however, that as computers develop simulation will make many statistical practices developed in vacuo widely useful.

Some of many other open design problems can be seen by considering Figure 2, where estimated quantiles of a distribution are plotted. One would generally want to smooth these plots or fit some regression function to assess the rate of convergence to the asymptotic normal distribution. There are problems in that the number of simulations, m, was fixed in advance and thus, the variances at each n vary. Moreover, one would want to couple the smoothing or regression analysis of the various quantiles. These are both functionally and statistically correlated for each n across quantiles and with n for each quantile.

Detailed analysis of such graphic output needs much more work; it is possible that the work of Efron and Morris (1972) may be relevant to this problem.

Besides the smoothing, any program such as COMPSTAT should provide facility for direct plotting of output tables of rounded and perhaps smoothed data. This is one facility computer scientists can provide us with.

3. MISCELLANEOUS PROBLEMS AND OPEN QUESTIONS.

I have not touched on many questions in large-scale simulation. A few are discussed here to emphasize that there are many problems that do not even start to fit on present or future computers. Thus, simulation, especially without some analytic support, is not always a possible way out of problems, although some people feel simulation is the last resort. Other questions discussed below indicate that there are simple problems we cannot handle.

a) Conditional distributions.

Conditioning poses problems in simulations which I do not know how to handle efficiently. Thus, in fitting exponential polynomials to data from a nonhomogeneous Poisson process (Lewis, 1972) observed for a time t_0, one wants to condition on the number, n, of events observed in $(0,t_0)$. The times to events t_i are then order statistics from a uniform random sample of size n. In testing for a second order term in the polynomial one wants the conditional distribution of Σt_i^2, given n and Σt_i. Conceptually this is simple to see,

as Σt_i^2 is the distance from the origin to the $n-1$ dimensional hyperplane defined by fixing Σt_i. The joint asymptotic normality of Σt_i^2 and Σt_i give the result that for large n, $\Sigma t_i^2/n$ has a conditional normal distribution with mean (Lewis, 1972).

$$\mu = E(\Sigma t_i^2 | u; \Sigma t_i) = t_0^2 \frac{\Sigma t_i}{nt_0} - \frac{1}{6}$$

and standard deviation

$$\sigma = \frac{1}{4} \frac{t_0}{(12n)^{1/2}} .$$

How does one simulate this problem for small n and assess the rate of convergence to n? This must be a very common problem.

b) Multivariate problems.

I have not mentioned simulation of multivariate statistics S. An immediate problem here is that quantiles and percentiles are not uniquely defined, so one has to use joint moments, which could be estimated in COMPSTAT, or rely on probability density functions. I have not discussed density estimation here at all. Multivariate problems, of course, also bring in new aspects of graphical and tabular output which are non-trivial.

c) Simulated maximum likelihood.

As a last stab, I would like to mention another area which interests me. In complicated time series we now have computationally feasible tools such as spectral analysis to help in defining and delineating models. Once this is done, however, there are often no reasonable ways of estimating parameters of the model, especially since likelihoods cannot be derived, even though the model is structurally simple. It would be useful to simulate the joint density of the observations at the observed data point as a function of the parameters so as to find the maximum likelihood estimates of the parameters. I assume this is worth the cost to the experimenter. One then has a more complicated case of a), closely related to response surface designs. The solution seems to be far away.

The reader is referred to the papers by Tukey (1972 a, b) for further problems.

4. ACKNOWLEDGEMENTS

I wish to thank A. S. Goodman of the IBM Yorktown Heights Research Center for his long involvement with the COMPSTAT program, G. P. Learmonth of the Naval Postgraduate School for his co-work on testing random number generators, and D. P. Gaver and G. S. Shedler for stimulating my interest in these problems. Also A. G. Anderson for support at IBM on the COMPSTAT program.

5. REFERENCES

1) Ahrens, J. H., and Dieter, U. (1972). Computer Methods for sampling from the exponential and normal distributions. Communications of the ACM, 15, 873-882.

2) Andrews, D. F., Bickel, P. J., Hampel, F. R., Huber, P. J., Rogers, W. H. and Tukey, J. W. (1972). Robust Estimates of Location: Survey and Advances. Princeton University Press: Princeton, New Jersey.

3) Arvesen, J. N. (1969). Jackknifing U-statistics. Annals of Mathematical Statistics, 40, 2076-2100.

4) Arvesen, J. N., and Salsburg, D. S. (1972). Approximate tests and confidence intervals using the jackknife. To appear in Perspectives in Biometry, R. Elashoff (ed.). Academic Press, New York.

5) Bates, C. B., and Zirkle, J. A. (1971). Analysis of random numbers from four uniform random number generators. Report TR4-71, U. S. Army Combat Development Command, Fort Belvoir, Virginia. (August, 1971).

6) Chambers, J. (1970). Computers in Statistical Research: Simulation and Computer-Aided Mathematics. Technometrics, 12, 1-16.

7) Chambers, J. (1971). Algorithm 410, Partial sorting [M1]. Comm. ACM, 14, 357-8.

8) Couveyou, R. L., and MacPherson, R. D. (1969). Fourier analysis of uniform random number generations. Journal of ACM, 14, 100-119.

9) Cox, D. R., and Lewis, P. A. W. (1966). The statistical analysis of series of events. Methuen: London and Barnes and Noble: New York.

10) David, H. A. (1971). Order Statistics. Wiley: New York.

11) Efron, B., and Morris, C. (1972). Limiting risk of Bayes and Empirical Bayes Estimations - Part II. Journal Am. Statist. Assoc., 67, 130-139.

12) Fieller, E. C., and Hartley, H. O. (1959). Sampling with control variables. Biometrika, 41, 494.

13) Freiberger, W., and Grenander, U. (1971). A short course in computational probability and statistics. Springer-Verlag: Berlin.

14) Gaver, D. P., and Hoel, D. G. (1970). Comparison of certain small-sample Poisson probability estimates. Technometrics, 12, 835-850.

15) Gaver, D. P. (1969). Statistical methods for improving simulation efficiency. Proc. 3rd Annual Conference on Applications of Simulation: Los Angeles, Dec. 1969.

16) Goodman, A. S., Lewis, P. A. W., and Robbins, H. E. (1972). Simultaneous estimation of large numbers of extreme quantiles in simulation experiments. Submitted to J. Amer. Statist. Assoc.

17) Gray, H. L., and Shucany, W. R. (1972). The generalized jackknife statistic. Marcel Dekker: New York.

18) Halton, J. H. (1970). A retrospective and prospective survey of the Monte Carlo method. SIAM Review, 12, 1-63.

19) Hammersley, J., and Handscomb, D. C. (1964). Monte Carlo Methods. Methuen: London.

20) Hammersley, J. M. and Mauldon, J. G. (1956). General principles of antithetic variates. Proceedings of the Cambridge Philosophical Society, 52, 476-481.

21) Hartley, H. O. (1972). The impact of computers on statistics. Proc. of Computer Science and Statistics: 5th Annual Symposium on the Interface. M. O. Locks (ed.). Western Periodical Co.: Hollywood, California.

22) Hemmerle, W. J. (1967). Statistical computations on a digital computer. Blaisdell: Waltham, Massachusetts.

23) Hodges, J. L., and Lehman, E. L. (1956). Two approximations to the Robbins-Monro process. Proc. 3rd Berkeley Symposium on Mathematical Statistics and Probability, 1, 95-104.

24) Isaac, E. J. and Singleton, R. C. (1956). Sorting by address calculation. Journal of ACM, 3, 169-174.

25) Knuth, D. E. (1969). Fundamental Algorithms. Addison-Wesley: Reading, Massachusetts.

26) Lehman, E. L. (1959). Testing Statistical Hypotheses. Wiley: New York.

27) Lewis, P. A. W. (1972). Recent results in the statistical analysis of univariate point processes. In Stochastic Point Processes, P. A. W. Lewis (ed.). Wiley: New York.

28) Lewis, P. A. W., Goodman, A. S., and Miller, J. M. (1969). A pseudo-random number generator for the System/360. IBM Systems Journal, 8, 136-146.

29) Lewis, P. A. W., and Goodman, A. S. (1970). The null distribution of the first three product-moment statistics for exponential, half-Gamma, and normal scores. In Selected Tables in Mathematical Statistics, Harter, H. L., and Owen, D. B. (eds.). Markham Publishing Co.: Chicago, Illinois.

30) Liniger, W. (1961). On a method by D. H. Lehmer for the generation of pseudo-random numbers. Numerische Mathematik, 3, 265-270.

31) Lurie, D., and Hartley, H. O. (1972). Machine-generation of order statistics for Monte Carlo computations. The American Statistician, 26, 26-29.

32) Maisel, H., and Gnugnoli, G. (1972). Simulation of discrete stochastic systems. Science Research Associates, Inc.: Chicago.

33) Marsaglia, G., and Bray, T. A. (1968). One-line random number generators and their use in combinations. Comm. ACM, 11, 757-759.

34) Marsaglia, G. (1968). Random numbers fall mainly in the planes. Proc. National Academy of Sciences, 61, 25-28.

35) Marsaglia, G. (1972). The structure of linear congruential sequences. In Applications of Number Theory in Numerical Analysis, S. K. Zaremba (ed.). Academic Press: New York.

36) Martin, W. A. (1971). Sorting. Computing Surveys, 3, 147-74.

37) Mihram, G. A. (1972). Simulation: Statistical Foundations and Methodology. Academic Press: New York.

38) Miller, R. G., Jr. (1964). A trustworthy jackknife. Annals of Mathematical Statistics 35, 1594-1605.

39) Miller, R. G., Jr. (1968). Jackknifing variances. Annals of Mathematical Statistics, 39, 567-582.

40) Milton, R. C., and Nelder, J. A. (1969). Statistical Computation. Academic Press: New York.

41) Mosteller, F., and Tukey, J. W. (1968). Data analysis, including statistics. Handbook of Social Psychology, 2nd ed , V. 2. G. Lindzey and E. Aronson (eds.). Addison-Wesley: Reading, Mass. 80-203.

42) Nance, R. E., and Overstreet, C. (1972). A bibliography on random number generation. Computing Reviews (Bibliography, 29) 13, 495-508.

43) Newman, T. G., and Odell, P. L. (1971). The generation of random deviates. Griffin: London and Hafner: New York.

44) Payne, W. H., Rabung, J. R., and Bagyo, T. P. (1969). Coding the Lehman pseudo-random number generator. Comm. of ACM, 12, 85-86.

45) Quenouille, M. H. (1956). Notes on bias in estimation. Biometrika, 43, 353-360.

46) Robbins, H. E., and Monro, S. (1951). A stochastic approximation method. Annals of Mathematical Statistics, 22, 400-407.

47) Schucany, W. R. (1972). Order statistics in simulation. J. Statist. Comput. Simul., 1, 281-286.

48) Schucany, W. R., Gray, H. L., and Owen, D. B. (1971). On bias reduction in estimation. Journal Amer. Statist. Assoc., 66, 524-533.

49) Somerville, P. N. (1970). A technique for the computation of percentage points of a statistic. Technometrics, 12, 373-382.

50) Tausworthe, R. C. (1965). Random numbers generated by linear recurrence modulo two. Math Comp., 19, 201-209.

51) Toothill, J. P. R., Robinson, W. D., and Adams, A. G. (1971). The runs up-and-down performance of Tausworthe pseudo-random number generators. Journal of ACM, 18, 381-399.

52) Tukey, J. W. (1958). Bias and confidence in not-quite large samples. Annals of Mathematical Statistics, (abstract), 29, 614.

53) Tukey, J. W. (1972a). Lags in statistical technology. Canadian Journal of Statistics.

54) Tukey, J. W. (1972b). How computing and statistics affect each other. To appear.

Note added in proof: I am indebted to Dr. J. P. C. Kleijnen for pointing out some references on the design and analysis of computer simulation experiments.

Naylor, T. H. (ed.) (1969). The design of computer simulation experiments. Duke University Press. Durham, N. C.

Kleijnen, J. P. C., Naylor, T. H., and Seaks, T. G. (1972). The use of multiple ranking procedures to analyze simulations of management systems: a tutorial. MANAGEMENT SCIENCE, APPLICATION SERIES, 18, 245-257.

Kleijnen, J. P. C. (1972). The statistical design and analysis of computer simulation experiments: a survey. MANAGEMENT INFORMATICS, 1, 57-66.