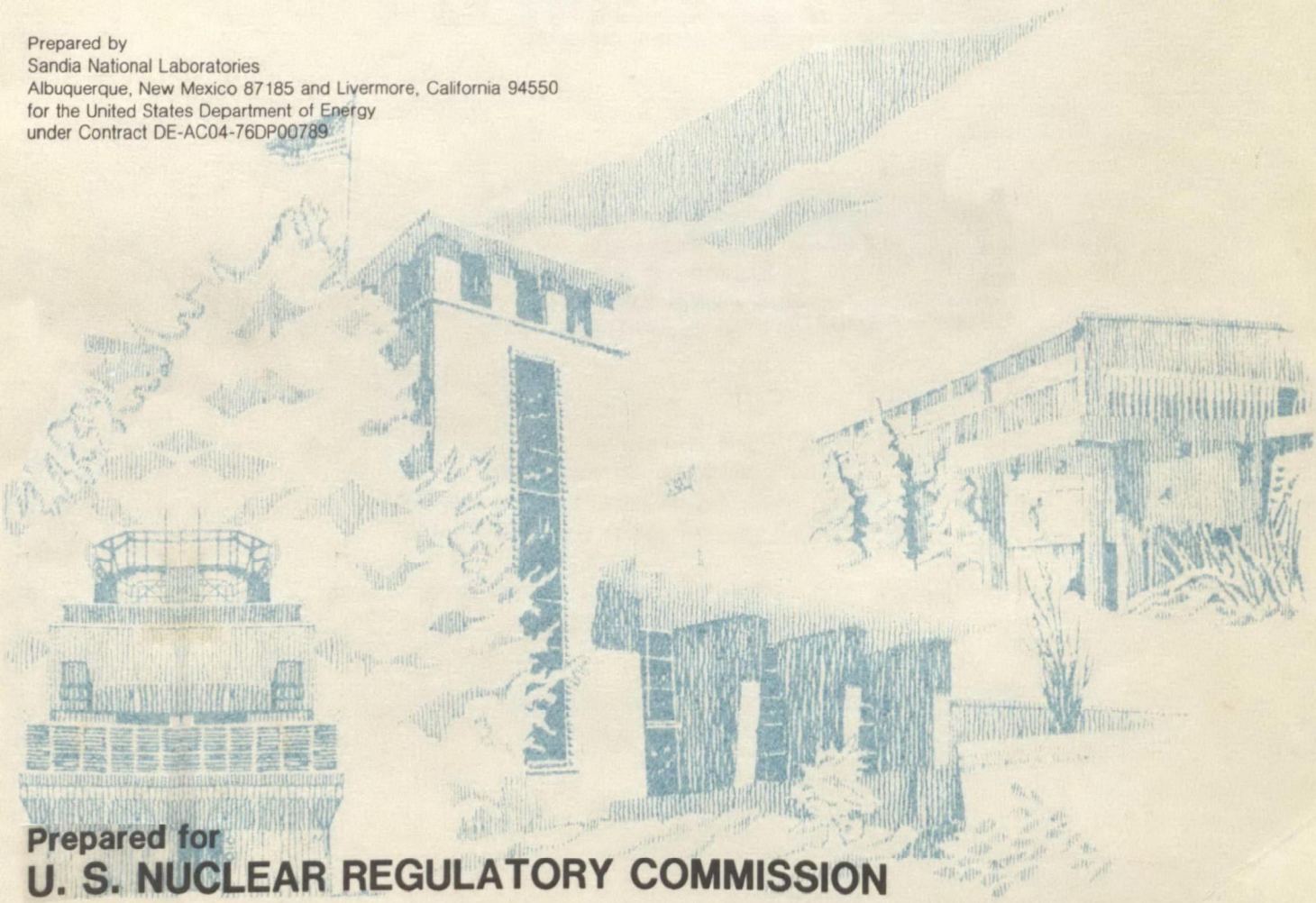
NUREG/CR-3624
SAND83-2365
RG
Printed March 1984

A FORTRAN 77 Program and User's Guide for the Generation of Latin Hypercube and Random Samples for Use With Computer Models

Ronald L. Iman, Michael J. Shortencarier

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789



Prepared for
U. S. NUCLEAR REGULATORY COMMISSION

SF2900Q (8-81)

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.

Available from
GPO Sales Program
Division of Technical Information and Document Control
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555

and

National Technical Information Service
Springfield, Virginia 22161

NUREG/CR-3624
SAND83-2365
RG

A FORTRAN 77 PROGRAM AND USER'S GUIDE FOR THE GENERATION OF LATIN HYPERCUBE AND RANDOM SAMPLES FOR USE WITH COMPUTER MODELS

Ronald L. Iman
and
Michael J. Shortencarier

March 1984

Sandia National Laboratories
Albuquerque, New Mexico 87185
Operated by
Sandia Corporation
for the
U.S. Department of Energy

Prepared for
Division of Risk Analysis
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555
Under Memorandum of Understanding DOE 40-550-75
NRC FIN No. A1339

ABSTRACT

This document has been designed for users of the computer program developed by the authors at Sandia National Laboratories for the generation of either Latin hypercube or random multivariate samples. The Latin hypercube technique employs a constrained sampling scheme, whereas random sampling corresponds to a simple Monte Carlo technique. The generation of these samples is based on information supplied to the program by the user describing the variables or parameters used as input to the computer model. The actual sampled values are used to form vectors of variables commonly used as input to computer models for purposes of sensitivity and uncertainty analysis studies. The present program replaces the previous Latin hypercube sampling program developed at Sandia National Laboratories (Iman, Davenport, and Zeigler, 1980). The present version is written using FORTRAN 77 and greatly extends the program while making the program portable and user friendly.

THIS PAGE INTENTIONALLY LEFT BLANK.

TABLE OF CONTENTS

Section		Page
Acknowledgement		vi
1.	Introduction to Latin Hypercube Sampling	1
2.	Input Parameters	9
	TITLE	11
	RANDOM SAMPLE	11
	NOBS	11
	NREPS	11
	RANDOM SEED	11
	CORRELATION MATRIX	11
	RANDOM PAIRING	12
	OUTPUT	12
	NORMAL	13
	LOGNORMAL	13
	UNIFORM	14
	LOGUNIFORM	14
	UNIFORM*	14
	LOGUNIFORM*	14
	TRIANGULAR	15
	BETA	15
	USER DISTRIBUTION	15
3.	Additional Information on Distributions Supplied by the Computer Program	15
	Normal Distribution	16
	Lognormal Distribution	17
	Uniform Distribution	18
	Loguniform Distribution	19
	Triangular Distribution	20
	Beta Distribution	23
4.	Examples of the Use of Subroutine USRDST to Generate Samples from a User Supplied Distribution	29
5.	Modifying the Computer Program	37
	Machine Constants	37
	Random Number Generator	37
	Redimensioning	38
	The Output File	39
References		40
Appendix - Example of Program Output		48

ACKNOWLEDGEMENT

The authors would like to thank Donald E. Amos of Sandia National Laboratories for his help in adapting subroutines in AMOSLIB for use in generating the beta distribution in the computer program. The authors would also like to acknowledge the contribution of James M. Davenport and Diane K. Zeigler in the development of an earlier version of the program.

1. INTRODUCTION TO LATIN HYPERCUBE SAMPLING

This report describes how to use the latest version of a computer program for the generation of multivariate samples either completely at random or by a constrained randomization termed Latin hypercube sampling (LHS). This program has been developed by the authors at Sandia National Laboratories and replaces the previous program described in Iman, Davenport, and Zeigler (1980). Every attempt has been made to make the present program portable and user-friendly while at the same time expanding the capability of the program to include additional sampling distributions. A complete listing of the computer code is given on microfiche inside the back cover. The first section of this report describes Latin hypercube sampling. The parameters needed by the program are described in Section 2 and are followed in Section 3 by a more detailed description of the various distributions built into the program. A subroutine that allows the user to sample from distributions (including empirical data) other than those built into the code is described in Section 4. Instructions for modifying the program are given in Section 5. The report concludes with examples of program output in an appendix.

The situation addressed by the computer program is the following. There is a variable of interest, Y, that is a function of other variables X_1, X_2, ..., X_K. This function may be quite complicated, for example, a computer model. A question to be investigated is: How does Y vary when the X's vary according to some assumed joint probability distribution? Related questions are: What is the expected value of Y? What is the 99th percentile of Y? etc.

A conventional approach to these questions is Monte Carlo. By sampling repeatedly from the assumed joint probability density function of the X's and evaluating Y for each sample, the distribution of Y, its mean, percentiles, etc., can be estimated. This is one option provided by the program for generating the X's. The program output, say for n Monte Carlo repetitions, is a set of k-dimensional vectors of input variables.

An alternative approach, which can yield more precise estimates, is to use a constrained sampling scheme. One such scheme, developed by McKay, Conover, and Beckman (1979), is Latin hypercube sampling (LHS). LHS selects n different values from each of k variables X_1, ..., X_k in the following manner. The range of each variable is divided into n nonoverlapping intervals on the basis of equal probability. One value from each interval is selected at random with respect to the probability density in the interval. The n values thus obtained for X_1 are paired in a random manner (equally likely combinations) with the n values of X_2. These n pairs are combined in a random manner with the n values of X_3 to form n triplets, and so on, until n k-tuplets are formed. This is the Latin hypercube sample. It is convenient to think of the LHS, or a random sample of size n, as forming an n x k matrix of input where the ith row contains specific values of each of the k input variables to be used on the ith run of the computer model.

The LHS technique has been applied to many different computer models since 1975. The results of an application of LHS to a large computer model can be found in Steck, Iman, and Dahlgren (1976). A more detailed description of LHS with application to sensitivity analysis techniques can be found in Iman, Helton, and Campbell (1981a, 1981b). A tutorial on LHS may be found in Iman and Conover (1982b). A comparison of LHS with other techniques is given in Iman and Helton (1983).

To help clarify how intervals are determined in the LHS, consider a simple example where it is desired to generate a LHS of size $n = 5$ with two input variables. Let us assume that the first random variable X_1 has a normal distribution concentrated on the range from A to B. In this program, the following interpretations (not subject to change by the user without modifying the code) are given to A and B for both the normal and lognormal distributions, namely

$$P(X_1 \leq A) = .001 \text{ and } P(X_1 \geq B) = .001 ,$$

where P(E) denotes the probability of event E. That is, A is defined as the .001 quantile and B is defined as the .999 quantile of the distribution of X_1. Thus, $P(A \leq X_1 \leq B) = .998$, so both the normal and lognormal distributions are truncated slightly in the program. That is, the sampling procedure excludes values outside the interval [A, B]. These definitions of A and B imply that the mean of the normal distribution is given by $\mu = (A + B)/2$ and since for a standardized normal variable Z,

$$P(Z \leq -3.09) = .001 ,$$

it follows that the standard deviation of the desired truncated normal distribution is given (to a close approximation) by

$$\sigma = (B - \mu)/3.09 = (B - A)/6.18 .$$

With the parameters μ and σ thus defined, the endpoints of the intervals are easily determined. The intervals for $n = 5$ are illustrated in Figure 1 in terms of both the density function and the more easily used cumulative distribution function (cdf). If the distribution were not truncated, then the intervals in Figure 1 would satisfy

$$P(A \leq X_1 \leq C) = P(F \leq X_1 \leq B) = .199$$
$$\cong P(C \leq X_1 \leq D) = P(D \leq X_1 \leq E) = P(E \leq X_1 \leq F) = .2 .$$

To account for truncation requires dividing these probabilities by .998. Thus, for all practical purposes, the five intervals correspond to 20% probability.

We shall assume in this example that the second random variable, X_2, has a uniform distribution on the interval from G to H. The corresponding intervals used in the LHS for X_2 are given in Figure 2 in terms of both the density function and the cdf.

The next step in obtaining the LHS is to pick specific values of X_1 and X_2 in each of their five respective intervals. This selection should be done in a random manner with respect to the density in each interval; that is, the selection should reflect the height of the density across the interval. For example, in the (A,C) interval for X_1, values close to C will have a higher probability of selection than will those values close to A. Next, the selected values of X_1 and X_2 are paired to form the required five input vectors. In the original concept of LHS as outlined in McKay, Conover, and Beckman (1979),

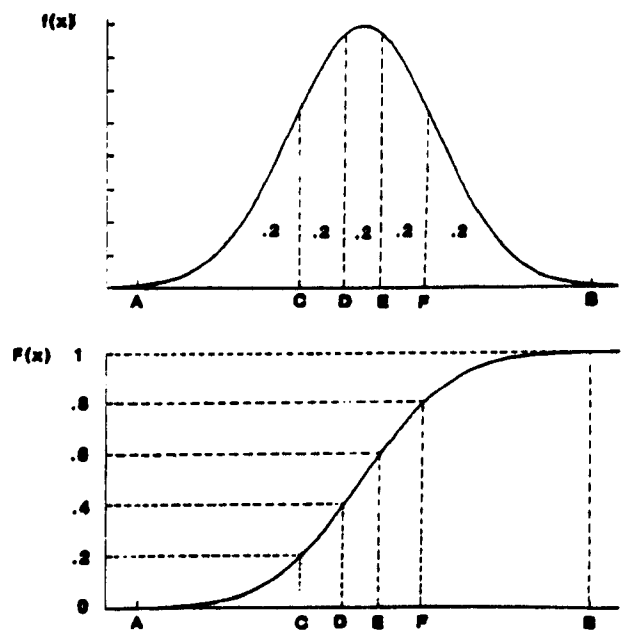


Figure 1. Intervals Used with a LHS of Size n = 5 in Terms of the Density Function and Cumulative Distribution Function for a Normal Random Variable

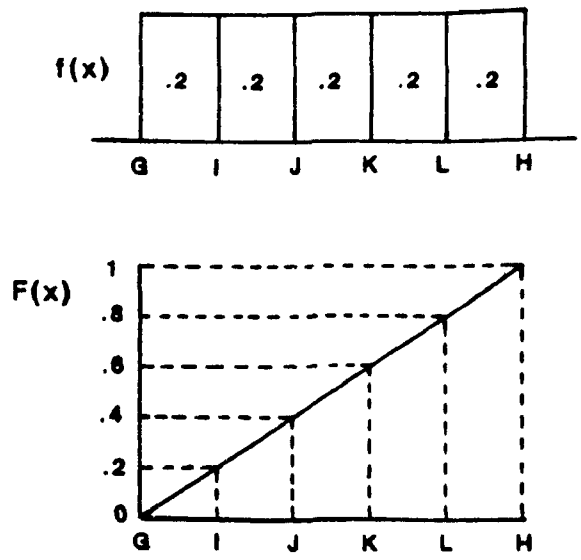


Figure 2. Intervals Used with a LHS of Size n = 5 in Terms of the Density Function and Cumulative Distribution Function for a Uniform Random Variable

the pairing was done by associating a random permutation of the first n integers with each input variable. For purposes of illustration, in the present example consider two random permutations of the integers (1, 2, 3, 4, 5) as follows:

Permutation Set No. 1: (3, 1, 5, 2, 4)

Permutation Set No. 2: (2, 4, 1, 3, 5)

By using the respective position within these permutation sets as interval numbers for X_1 (Set 1) and X_2 (Set 2), the following pairing of intervals would be formed.

Computer Run No.	Interval No. Used for X_1	Interval No. Used for X_2
1	3	2
2	1	4
3	5	1
4	2	3
5	4	5

Thus, on computer run number 1 the input vector is formed by selecting the specific value of X_1 from interval number 3 (D to E) and pairing this value with the specific value of X_2 selected from interval number 2 (I to J), etc. Once the specific values of each variable are obtained to form the five input vectors, a two-dimensional representation of the LHS can be made such as given in Figure 3.

Note in Figure 3 that all of the intervals for X_1 have been sampled, and the same is true of X_2. In general, a set of n LHS points in k-dimensional Euclidean space contains one point in each of the intervals for each of the k variables.

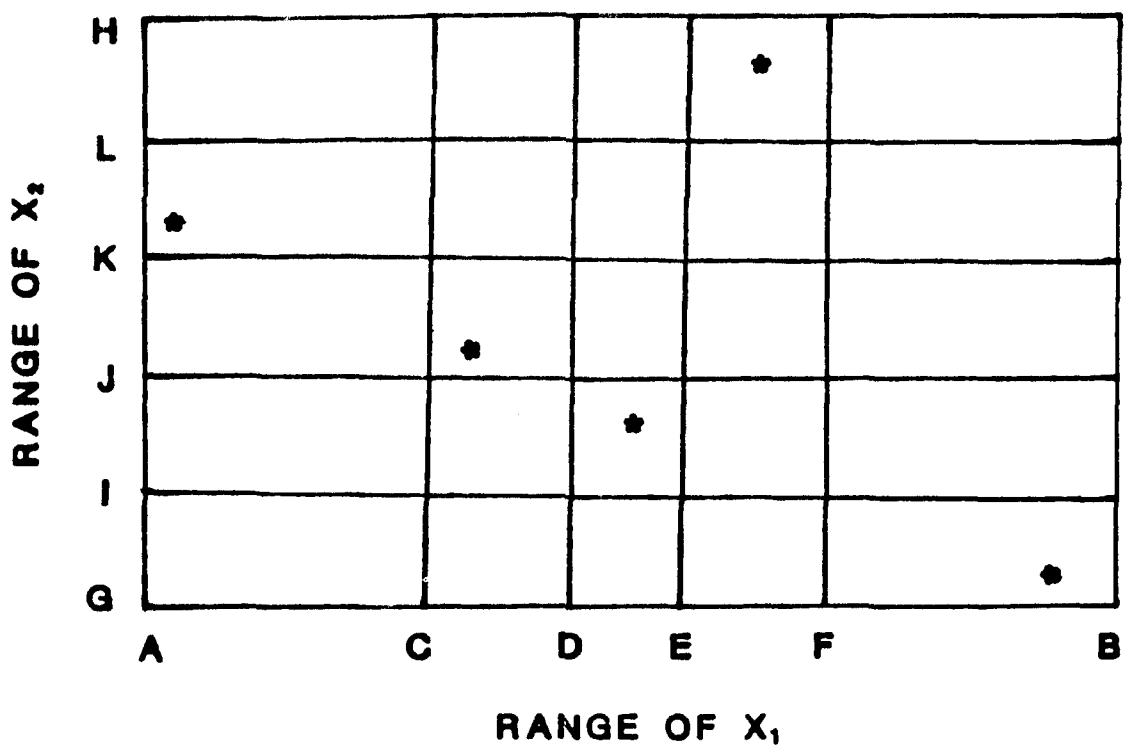


Figure 3. A Two-Dimensional Representation of One Possible LHS of Size 5 Utilizing X_1 and X_2

To illustrate how the specific values of a variable are obtained in a LHS, consider the following example. Suppose it is desired to obtain a LHS of size n = 5 from a normal distribution on the range from 0.0 to 10.0. Recall that these two limits are taken to represent the lower and upper .001 quantiles, respectively. Therefore, the random variable has a mean of five and a variance of 2.618 as indicated in Figure 4. These points together with the density characteristics of the normal distribution allow for the definition of the equal probability interval endpoints. These endpoints are shown in Figure 4 in terms of a density function. The next step is to randomly select an observation within each of the intervals. This selection is not done uniformly within the intervals shown in Figure 4, but rather it is done relative to the distribution being sampled (in this case, the normal distribution). This means that the sampling is done uniformly on the vertical axis of the cdf as shown in Figure 4.

Therefore, to get the specific values, n = 5 randomly selected uniform (0, 1) numbers (U_m, m = 1, 2, 3, 4, 5) are obtained to serve as probability levels. These probabilities are then scaled by

$$P_m = U_m(.2) + (m - 1)(.2) \qquad m = 1, 2, 3, 4, 5$$

This ensures that exactly one probability, P_m, will fall within each of the five intervals (0, .2), (.2, .4), (.4, .6), (.6, .8) and (.8, 1). The values P_m are used with the inverse normal distribution function to produce the specific values to be used in the LHS. Note that exactly one observation is taken from each interval shown in Figure 4. The entire process is shown in Table 1. Figure 4 makes it clear that when obtaining a LHS, it is easier to work with the cdf for each variable. This is the approach used in the computer program, rather than defining the endpoints of the intervals on the x-axis.

The above illustration shows how one input variable having a normal distribution is sampled with LHS. This procedure is repeated for each input variable, each time working with the corresponding cumulative distribution function. If a random sample is desired, then it is not necessary to divide the vertical axis into n intervals of equal width. Rather, n random numbers between 0 and 1 are obtained and each is mapped through the inverse distribution function to obtain the specific values. The final step in the sampling process involves pairing the selected values.

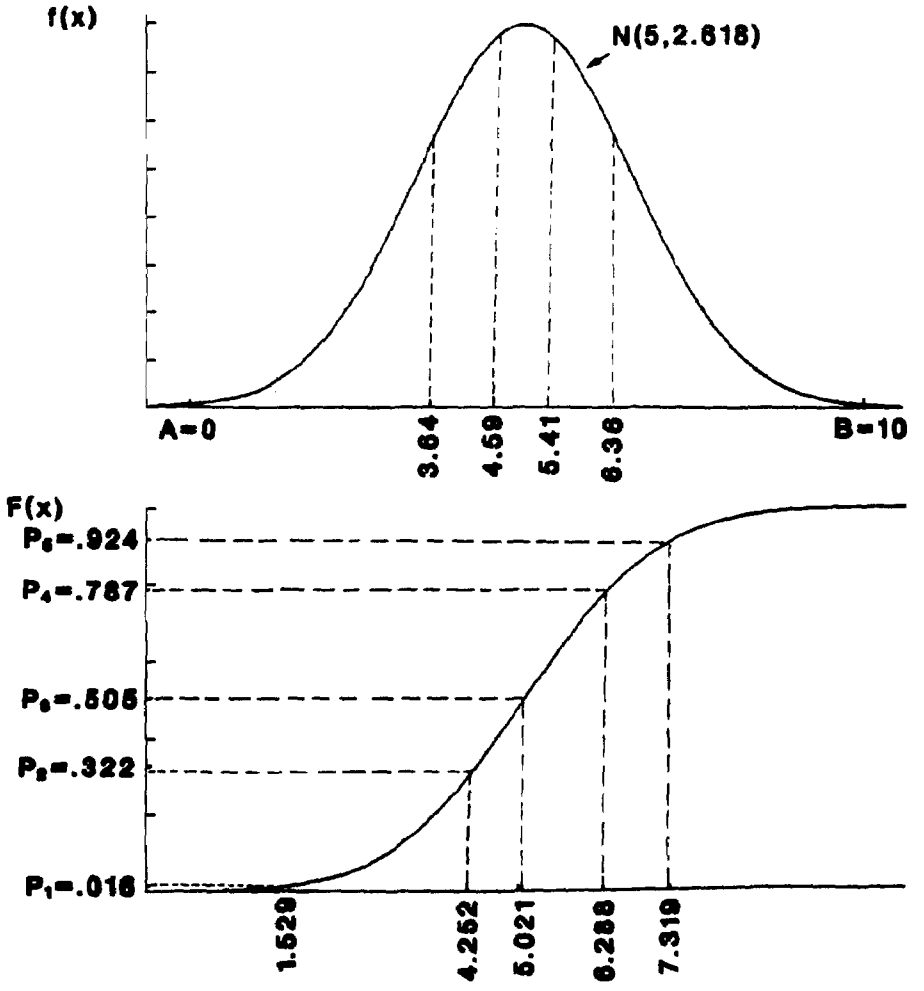


Figure 4. Interval Endpoints Used With a LHS of Size 5 (top) and Specific Values of X Selected Through the Inverse of the Distribution Function (bottom)

Table 1. One Possible Selection of Values for a LHS of Size 5 from a Normal Distribution on the Interval (0,10)

Interval Number m	Uniform (0,1) Random No. U_m	Scaled Probabilities With the Interval $P_m - U_m(.2)+(m-1)(.2)$	Corresponding Standard Normal Value (z-score) From the Inverse Distribution	Corresponding N(5,2.618) Observation Within the Intervals
1	.12718	.016	-2.144	1.529
2	.57689	.322	-0.462	4.252
3	.03652	.505	0.013	5.021
4	.29372	.787	0.796	6.288
5	.39332	.924	1.433	7.319

It should be noted that even though two variables are sampled independently and paired randomly, the sample correlation coefficient of the n pairs of variables in either a random sample or a LHS will, in general, not equal zero, just due to sampling fluctuations. In order to obtain a sample in which the sample correlations more nearly match the assumed, or intended, correlations, Iman and Conover (1982a) proposed a method for restricting the way in which the variables are paired. The effect of this restriction on the statistical properties of the estimated distribution of Y, its mean and percentiles, is not known but is felt to be small. The pairing of variables in the program can be done either randomly or by the restriction procedure through use of an input parameter, which is explained in the next section.

Additionally, the restricted pairing procedure of Iman and Conover can be used to induce a user-specified correlation among selected input variables through use of another input parameter explained in the next section. However, it should be pointed out that such induced correlations are based on the non-parametric technique known as rank correlation. Such a measure is used since it remains meaningful in the presence of nonnormal distributions on the input variables.

As a final note, if a correlation structure is <u>not</u> specified by the user, then the program computes a measure for detecting large pairwise correlations. This measure is known as the variance inflation factor (VIF) and is defined as the largest element on the diagonal of the inverse of the correlation matrix. As the VIF gets larger than 1, there may be some undesirably large pairwise correlations present. Marquardt and Snee (1975) deal with some very large VIFs ($> 2 \times 10^6$) and provide a very readable explanation on reasonable

sizes of VIFs. Marquardt (1970) indicates that there can be serious collinearity (i.e., large pairwise correlations present) for VIF > 10. Thus, there is certainly no problem as long as the VIF is close to 1. The VIF appears as part of the computer printout when the user requests the correlation matrix to be printed, given that no correlation structure has been specified by the user.

2. INPUT PARAMETERS

The LHS program requires certain parameters to be defined in order to generate one or more Latin hypercube or random samples. The program recognizes 17 keywords (no abbreviations allowed) which dictate the characteristics of the generated sample(s) such as type of sample (LHS or random), sample size, number of samples desired, correlation structure on input variables, and type of distribution specified on each variable. Other keywords are used to control the output from the printer. If the keyword requires accompanying numerical values, these values are input using list-directed read statements. In all cases, the generated sample(s) are automatically written on an output file in unformatted binary and it is up to the user to take the steps necessary to save the file. (More information on this point is given in Section 5.) The only restriction on the keywords is that there can be no leading blanks and at least one blank must follow each keyword.

There are a number of internal checks built into the program to ensure that the input parameters have been correctly specified. In the event an improper specification is detected, an appropriate message is printed and the execution of the program is terminated.

The role of each keyword will now be explained. For purposes of illustration, Table 2 gives an example setup that uses 16 of the 17 keywords to generate two random samples of size 20 each from nine input variables, some of which are correlated with one another.

Table 2. Parameter Setup for Generating
Two Random Samples of Size 20 Each.

```
 1.  TITLE - SETUP FOR GENERATION OF A RANDOM SAMPLE
 2.  RANDOM SAMPLE
 3.  NOBS 20
 4.  NREPS 2
 5.  RANDOM SEED - 898079140
 6.  NORMAL                OPTIONAL FIELD FOR NAMING INPUT VARIABLE 1
 7.  12  56
 8.  LOGNORMAL             OPTIONAL FIELD FOR NAMING INPUT VARIABLE 2
 9.  .01  2.13
10.  UNIFORM               OPTIONAL FIELD FOR NAMING INPUT VARIABLE 3
11.  1  3
12.  LOGUNIFORM            OPTIONAL FIELD FOR NAMING INPUT VARIABLE 4
13.  6.0E7  8.1E10
14.  UNIFORM*              OPTIONAL FIELD FOR NAMING INPUT VARIABLE 5
15.  3       5  6  9        1  2  3  4
16.  LOGUNIFORM*           OPTIONAL FIELD FOR NAMING INPUT VARIABLE 6
17.  5       3  3  4  4  6       5000  5500  6000  6500  7000  7500
18.  TRIANGULAR            OPTIONAL FIELD FOR NAMING INPUT VARIABLE 7
19.  10  15  30
20.  BETA                  OPTIONAL FIELD FOR NAMING INPUT VARIABLE 8
21.  10  45  .5  1.5
22.  USER DISTRIBUTION     OPTIONAL FIELD FOR NAMING INPUT VARIABLE 9
23.  4
24.  0  .2
25.  1  .3
26.  2  .4
27.  3  .1
28.  CORRELATION MATRIX
29.  3       1  2  .8       1  5  .7       2  5  .6
30.  OUTPUT CORR HIST DATA
```

TITLE

This keyword can be followed with alphanumeric data to help describe the application of the sample (see line 1 of Table 2). This information will be printed as a one-line header on each page of the output. This keyword is optional. If it is omitted, a blank header is generated at the top of each page.

RANDOM SAMPLE

If this keyword is present, a random sample(s) is generated. If it is omitted, a Latin hypercube sample(s) is generated.

NOBS *** This keyword is required. ***

This keyword must be followed by a positive integer that specifies the desired sample size. The maximum number of observations, currently 1,000, is easily changed (see the discussion elsewhere in this report).

NREPS

This keyword can be used to generate multiple samples. It is optional, but when it is present, it must be followed by a positive integer to specify the desired number of samples (each of size NOBS). If it is omitted, one sample is generated. If NREPS is followed by the positive integer m, then m complete samples (each of size NOBS) will be written back to back on the output file.

RANDOM SEED *** This keyword is required. ***

This keyword must be followed by an integer within the machine's range. For example, within plus or minus $2^{31} - 1$ on the VAX 11/780, within plus or minus $2^{48} - 1$ on the CDC 7600, and within plus or minus $2^{63} - 1$ on the CRAY 1. This number is used as a starting point for the random number generator and is printed at the beginning of each sample. If NREPS specifies a number greater than 1, the current value of the random seed is retrieved at the start of the generation of each new sample. This new value is printed at the beginning of the new sample so that any one desired sample can be regenerated by rerunning the program with the new seed and with the NREPS parameter omitted (or having NREPS 1).

CORRELATION MATRIX

This keyword is used when it is desired to induce a rank correlation structure among the input variables using the restricted pairing technique of Iman and Conover (1982a). It should be followed by one or more lines providing the desired rank correlations among those pairs of input variables having a rank correlation other than zero. The first value to be supplied is the number of pairwise rank correlations, m, followed by m ordered triples containing the numbers of the variables being correlated

along with the required rank correlation. Currently the number of pairwise rank correlations is limited to 50. See Section 5 for instructions on how to adjust this value. For example, line 29 of Table 2 first indicates that three pairs of variables are to be correlated. The next information indicates that variables 1 and 2 are to have a rank correlation of .8, then variables 1 and 5 are to have a rank correlation of .7, and finally variables 2 and 5 are to have a rank correlation of .6. If this keyword is omitted, all pairwise correlations are assumed to be zero. The user should note that the restricted pairing technique of Iman and Conover requires n > k. That is, the technique can only be applied directly if n > k; otherwise an error message is printed. For the example in Table 2, n = 20 and k = 9. One possibility for working around this restriction is to use the sampling technique in a piecewise fashion on a subset of the k variables where the number of variables used in each subset is less than n. The resulting subsets are then pieced together to form the n x k input matrix. Such a piecewise approach would ensure the desired correlations among variables within subsets, but there could exist undesired correlations between variables belonging to different subsets. In a case such as this, as well as in general, the resulting rank correlation matrix should be examined very carefully to make sure it satisfies the user's requirements. Additionally, in the case that the user does not specify a correlation matrix and does not have n > k, the program will generate the desired sampling through simple random mixing rather than restricted pairing. Since this approach brings up the possibility of unwanted correlations, the resulting correlation matrix should again be examined very carefully.

As a final note, if the input correlation structure is such that the rank correlation matrix is not a positive definite matrix, an iterative scheme (Iman and Davenport, 1982) built into the program will attempt to adjust the input rank correlation matrix to make it positive definite. In this case a message is printed out along with the adjusted matrix indicating that an adjustment has been made and requesting the user to examine the adjusted matrix to see if it still satisfies the correlation requirements.

RANDOM PAIRING

Use of this keyword allows the sampled values to be paired randomly as explained at the end of Section 1. If this keyword does not appear, then the restricted pairing technique of Iman and Conover (1982a) is used subject to the restrictions mentioned under the keyword CORRELATION MATRIX. In the event the user mistakenly includes both the keywords RANDOM PAIRING and CORRELATION MATRIX in the same run, the program will continue to execute; however, the former keyword is ignored with a message to that effect printed after the correlation matrix.

OUTPUT

This keyword is followed by one or more of three additional keywords. These additional keywords can appear in any order (separated by blanks).

Their purpose is to control the amount of printer output. These keywords function as follows.

CORR - Both the raw and rank correlation matrices associated with the actual sample generated are printed.

HIST - Histograms are generated for each variable in the sample based on the actual values of each variable in the sample.

DATA - If this option is specified, each complete sample (n observations on k variables) will be listed, followed by a complete listing of the ranks of each variable. Use of this option makes the individual sample input vectors available to the user.

The remaining keywords (at least one of which is required) allow the user to specify the distribution and corresponding parameters for each of the variables in the sample. Eight distributions providing a great deal of flexibility are supplied by the computer program. However, use of a ninth keyword allows for a user-supplied subroutine to be called in order to generate other types of distributions. This subroutine can also be used to obtain samples from empirical data by using the corresponding empirical distribution function. Such a user-supplied subroutine can easily be coded so that it can be called more than once in order to generate different distributions for different variables. Examples of the use of such a subroutine appear in Section 4. For each of the keywords given below additional information describing the variable can be placed as a comment immediately after the trailing blank at the end of the keyword (see the examples on lines 6, 8, 10, 12, 14, 16, 18, 20, and 22 of Table 2). Such information becomes part of the computer printout and is useful for reference. Each keyword must be followed by at least one additional line of information specifying the parameters of the distribution. The only possible exception is the keyword USER DISTRIBUTION where additional information is user-dependent. The number of times these keywords can be repeated is limited only by the dimensions in the program. The program is currently dimensioned to allow for 50 variables but is easily modified as indicated in Section 5.

NORMAL

The second line of information associated with this keyword supplies, in order, the .001 quantile and .999 quantile of the desired normal distribution. See line 7 of Table 2 where these quantiles are respectively specified as 12 and 56.

LOGNORMAL

The two parameters specified on the second line have the restriction that both must be positive. Again, these two parameters are defined as the .001 quantile and the .999 quantile. See line 9 of Table 2 where these quantiles are specified as .01 and 2.13 respectively.

UNIFORM

The second line of information provides, in order, the lower and upper endpoints of the interval that is to be sampled uniformly. See line 11 of Table 2 where a uniform distribution is defined on the interval from 1 to 3.

LOGUNIFORM

This distribution allows the variable to be sampled uniformly on the logarithms base 10 of the two positive parameters supplied on the second line of information. For example, on line 13 of Table 2, the values 6.0E7 and 8.1E10 are specified. The first step in the program is to find the base 10 logarithms of each of these values. These values are respectively 7.78 and 10.91. Next, a uniform distribution is generated on the interval 7.78 to 10.91. The last step is to find the antilogarithms of each of the latter values, i.e., 10^X. This scheme allows "uniform" sampling of variables on a logarithmic scale.

UNIFORM*

Use of this keyword allows for the samples from uniform distributions to be modified by changing the frequency with which uniform sampling is done within subintervals of the range of the variable. Thus, at least one additional line of information is required to allow different subintervals of a given interval to be sampled with frequencies other than what a strictly uniform distribution would provide. The first bit of information indicates the number of subintervals m, followed by m values indicating the frequency of sampling within each subinterval. Currently the maximum number of subintervals permitted is 50. See Section 5 for instructions of how to adjust this value. The sum of these frequencies must be equal to NOBS and each frequency must be greater than or equal to zero. The last information provides the endpoints of the subintervals in increasing order. For example, on line 15 of Table 2 rather than having 20 observa tions obtained uniformly on the interval from 1 to 4, the first number indicates that three subintervals are to be used. Next, the frequencies of sampling for these intervals are to be 5, 6, and 9. (Note that in general the sum of these frequencies is NOBS.) Finally, the endpoints of the subintervals are 1 and 2, 2 and 3, and 3 and 4. Thus, five observations are sampled according to a uniform distribution on the interval from 1 to 2, six observations are sampled according to a uniform distribution on the interval from 2 to 3, and finally, nine observations are sampled according to a uniform distribution on the interval from 3 to 4.

LOGUNIFORM*

This keyword applies to the loguniform distribution in exactly the same way as UNIFORM* does to the uniform distribution.

TRIANGULAR

This keyword requires an additional line of information containing three values, a, b, and c. The value b is the x-coordinate of the apex of the triangular distribution while a and c are the endpoints of the range. The program allows for $a < b < c$ or $a = b < c$ or $a < b = c$. In the case of $a = b$, the triangular distribution could be generated with the BETA keyword and $p = 1$ and $q = 2$. Also, for $b = c$, the triangular distribution can be generated with $p = 2$ and $q = 1$ from the beta distribution. Properties of the triangular distribution are given in the next section.

BETA

The second line of information accompanying this keyword contains two values A and B specifying the endpoints of the distribution followed by two shape parameters p and q. The shape parameters are described in detail in the next section along with figures illustrating the effect of different choices of p and q.

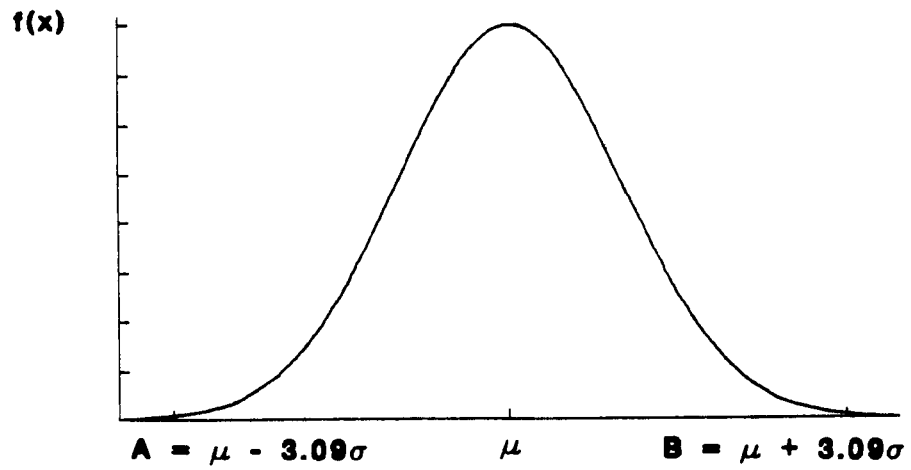
USER DISTRIBUTION

This keyword allows the user to modify a subroutine provided later in this report one or more times in order to generate samples from distributions other than those supplied by the program. Three examples are given in Section 4 to illustrate multiple uses of this option.

3. ADDITIONAL INFORMATION ON DISTRIBUTIONS SUPPLIED BY THE COMPUTER PROGRAM

In this section, a more detailed discussion is provided on each of the distributions that are built into the computer program. The next section provides additional information on the use of user-supplied distributions.

Normal Distribution



Density function: $f(x) = 1/(\sigma\sqrt{2\pi}) \exp\{-(x - \mu)^2/2\sigma^2\}$ $\quad -\infty < x < \infty$

Distribution function: $F(x) = \int_{-\infty}^{x} f(t)dt$ $\quad -\infty < x < \infty$

Expected value and variance: $E(X) = \mu$ and $V(X) = \sigma^2$

The user must specify A and B such that the following statements are satisfied:

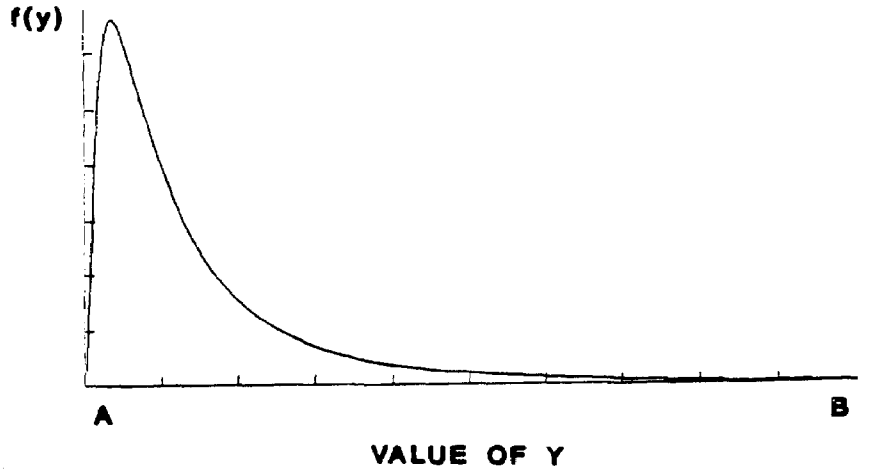
$$P(X < A) = .001 \text{ and } P(X > B) = .001$$

Thus, the normal distribution is truncated such that it is concentrated between A and B. The parameters of the truncated distribution can be expressed in terms of A and B as follows:

$$E(X) = \mu = (A + B)/2 \qquad V(X) = \sigma^2 = [(B - A)/6.18]^2$$

A truncated distribution is used since almost all applications of the program have historically involved a range from A to B. If the user has an application where truncation is not desired, then the normal distribution should be generated under the guise of subroutine USRDST which is explained in Section 4.

Lognormal Distribution



If $X \sim N(\mu, \sigma^2)$ and $Y = e^X$, then Y has a lognormal distribution.

$$f(y) = 1/(y\sigma\sqrt{2\pi}) \exp\{-(\ln y - \mu)^2/2\sigma^2\} \qquad y > 0$$

$$F(y) = \int_0^y f(t)dt \qquad y > 0$$

$$E(Y) = \exp(\mu + \sigma^2/2) \quad \text{and} \quad V(Y) = \exp(2\mu + \sigma^2)[\exp(\sigma^2) - 1]$$

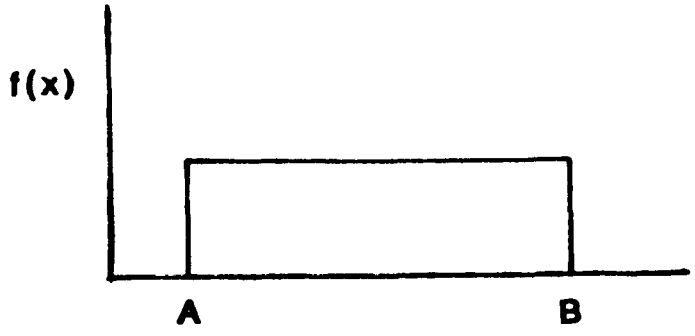
Median = e^{μ}

As with the normal distribution n, the user must specify A > 0 and B > 0 such that the following statements are satisfied:

$$P(Y < A) = .001 \text{ and } P(Y > B) = .001.$$

The program operates by first finding A* = ln A and B* = ln B and then generating a normal distribution with A* and B* playing the same roles as A and B with the normal distribution. Once samples from the normal distribution have been generated, each value of X is converted to a lognormal distribution sample value by $Y = e^X$.

Uniform Distribution



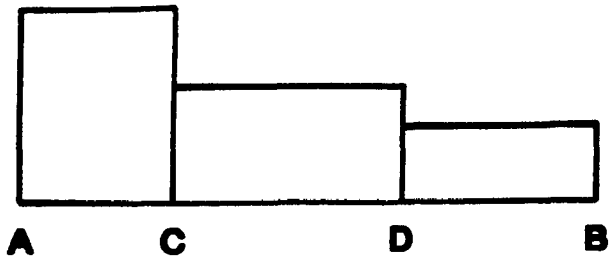
$f(x) = 1/(B - A) \qquad A \leq x \leq B$

$F(x) = (x - A)/(B - A) \qquad A \leq x \leq B$

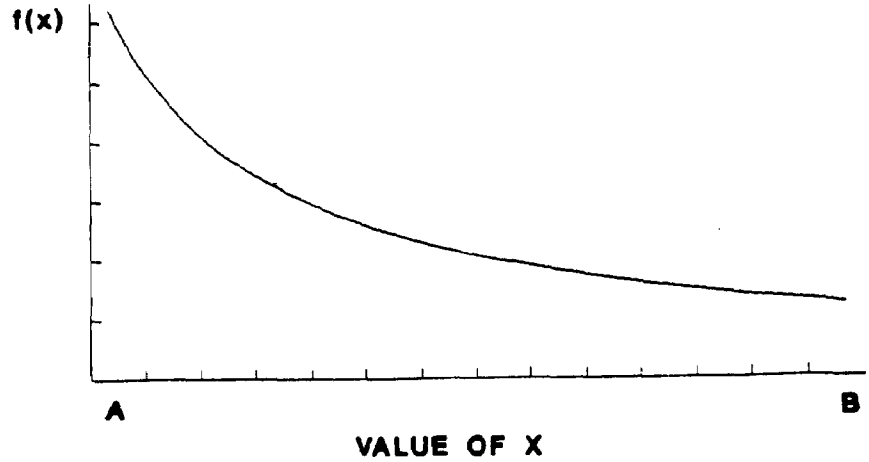
$E(X) = (A + B)/2$ and $V(X) = (B - A)^2/12$

The user must specify the endpoints A and B.

If UNIFORM* is specified, then a uniform distribution is sampled within each user-specified subinterval according to the user-specified frequency for each subinterval. Note that the subintervals do not have to be of equal width. This option is useful for generating a sample from a histogram. For example, the user could specify the interval endpoints as A < C < D < B with frequencies corresponding to the following histogram.



Loguniform Distribution



If X has a loguniform distribution on the interval from A to B where A > 0, then $Y = \log_{10} X$ has a uniform distribution on the interval from $\log_{10} A$ to $\log_{10} B$. Although the code uses base 10 logarithms, the following properties are stated in terms of natural logarithms in order to simplify the presentation.

$$f(x) = 1/x\ (\ln B - \ln A) \qquad A < x < B$$

$$F(x) = (\ln x - \ln A)/(\ln B - \ln A) \qquad A < x < B$$

$$E(X) = (B - A)/(\ln B - \ln A)$$

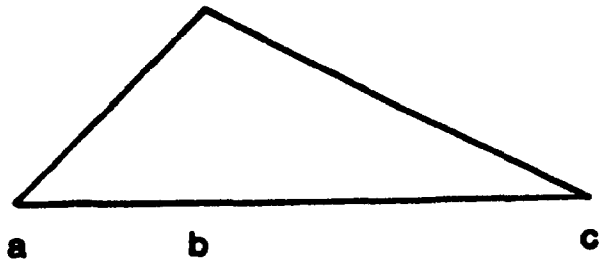
$$V(X) = (B - A)[(\ln B - \ln A)(B + A) - 2(B - A)]/[2(\ln B - \ln A)^2]$$

$$\text{Median} = \exp[(\ln B + \ln A)/2] = \sqrt{AB}$$

The user must specify A > 0 and B > 0. If it is desired to have A close to zero, then the user should be aware of how specific choices affect the sampling. For example, with $A = 10^{-3}$ and B = 10, 1/4 of the sample will be between 10^{-3} and 10^{-2}, another 1/4 will be between 10^{-2} and 10^{-1}, another 1/4 will be between 10^{-1} and 10^{0}, with the final 1/4 between 10^{0} and 10. If, however, A is selected to be 10^{-7}, then 1/2 will be between 10^{-7} and 10^{-3} and 1/2 between 10^{-3} and 10^{1} with only 1/8 between 10^{0} and 10. This distribution allows uniform sampling of variables on a logarithmic scale as each decade is sampled with the same frequency. The frequency of interval sampling is easily changed by using LOGUNIFORM* in the same manner as described with the uniform distribution.

Triangular Distribution

Case 1. $a < b < c$



$$f(x) = \frac{2(x - a)}{(c - a)(b - a)} \qquad a \leq x \leq b$$

$$= \frac{2(c - x)}{(c - a)(c - b)} \qquad b \leq x \leq c$$

$$F(x) = \frac{(x - a)^2}{(c - a)(b - a)} \qquad a \leq x \leq b$$

$$= \frac{b - a}{c - a} - \frac{(x + b - 2c)(x - b)}{(c - a)(c - b)} \qquad b \leq x \leq c$$

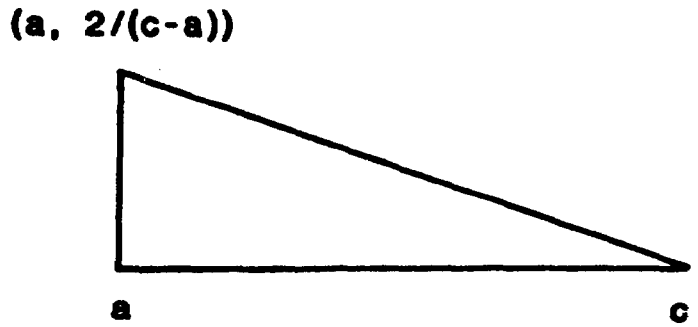
$$E(X) = \frac{a + b + c}{3} \qquad V(X) = \frac{a(a - b) + b(b - c) + c(c - a)}{18}$$

Median: $X_{.5} = a + \sqrt{(c - a)(b - a)/2} \qquad \text{if } b \geq \frac{a + c}{2}$

$$= c - \sqrt{(c - b)(c - a)/2} \qquad \text{if } b \leq \frac{a + c}{2}$$

Note with $a < b < c$, the user may want to consider using a beta distribution with a choice of $p = 2$, $q = 3$ or $p = 3$, $q = 2$ for example. See the figures provided in this section under the beta distribution.

Case 2. $a = b < c$

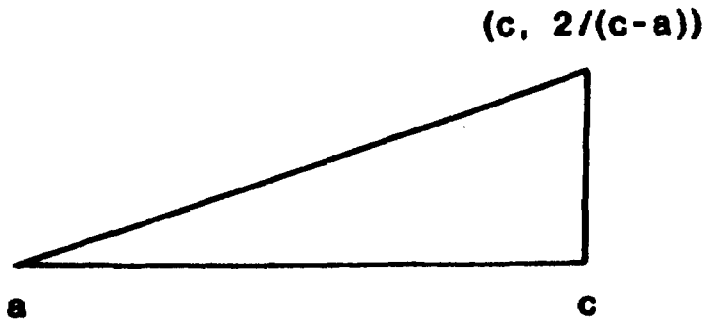


$$f(x) = \frac{2(c - x)}{(c - a)^2} \qquad F(x) = \frac{(x - a)(2c - x - a)}{(c - a)^2} \qquad a \leq x \leq c$$

$$E(X) = \frac{2a + c}{3} \qquad V(X) = \frac{(c - a)^2}{18} \qquad \text{Median: } X_{.5} = c - \frac{c - a}{\sqrt{2}}$$

Note that this distribution can also be generated using the beta distribution on the interval from a to c with $p = 1$ and $q = 2$.

Case 3. $a < b = c$



$$f(x) = \frac{2(x - a)}{(c - a)^2} \qquad F(x) = \frac{(x - a)^2}{(c - a)^2} \qquad a \leq x \leq c$$

$$E(X) = \frac{a + 2c}{3} \qquad V(X) = \frac{(c - a)^2}{18} \qquad \text{Median: } X_{.5} = a - \frac{c - a}{\sqrt{2}}$$

Note that this distribution can also be generated using the Beta distribution on the interval from a to c with $p = 2$ and $q = 1$.

Beta Distribution

Let Y be a random variable having a beta distribution on the interval (a,b) with parameters p and q. Further, let $X = (Y - a)/(b - a)$ so that $0 \leq X \leq 1$. Then X has a standard beta distribution with

$$E(X) = \frac{p}{p + q} \tag{1}$$

$$V(X) = \frac{pq}{(p + q)^2(p + q + 1)} . \tag{2}$$

Therefore,

$$E(Y) = (b - a)E(X) + a \tag{3a}$$

$$= \frac{aq + bp}{p + q} \tag{3b}$$

and

$$V(Y) = (b - a)^2 \, V(X) \tag{4a}$$

$$= \frac{(b - a)^2 pq}{(p + q)^2(p + q + 1)} . \tag{4b}$$

Equations (3b) and (4b) provide the user with the mean and variance of the random variable Y on a given interval from a to b for specific choices of p and q which act jointly to determine the shape of the underlying distribution. Figure 5 has been provided to help the user see the effect of various choices of q and p. As an example, this figure can be used to see the influence of changing p with q fixed at different values, or changing q with p fixed at different values, or letting p = q as both increase. In addition, equations (1) and (2) have been evaluated in Table 3 for the 16 choices of p and q in Figure 5. These values can easily be substituted into (3a) and (4a) to determine the mean and variance of Y for a given interval (a, b).

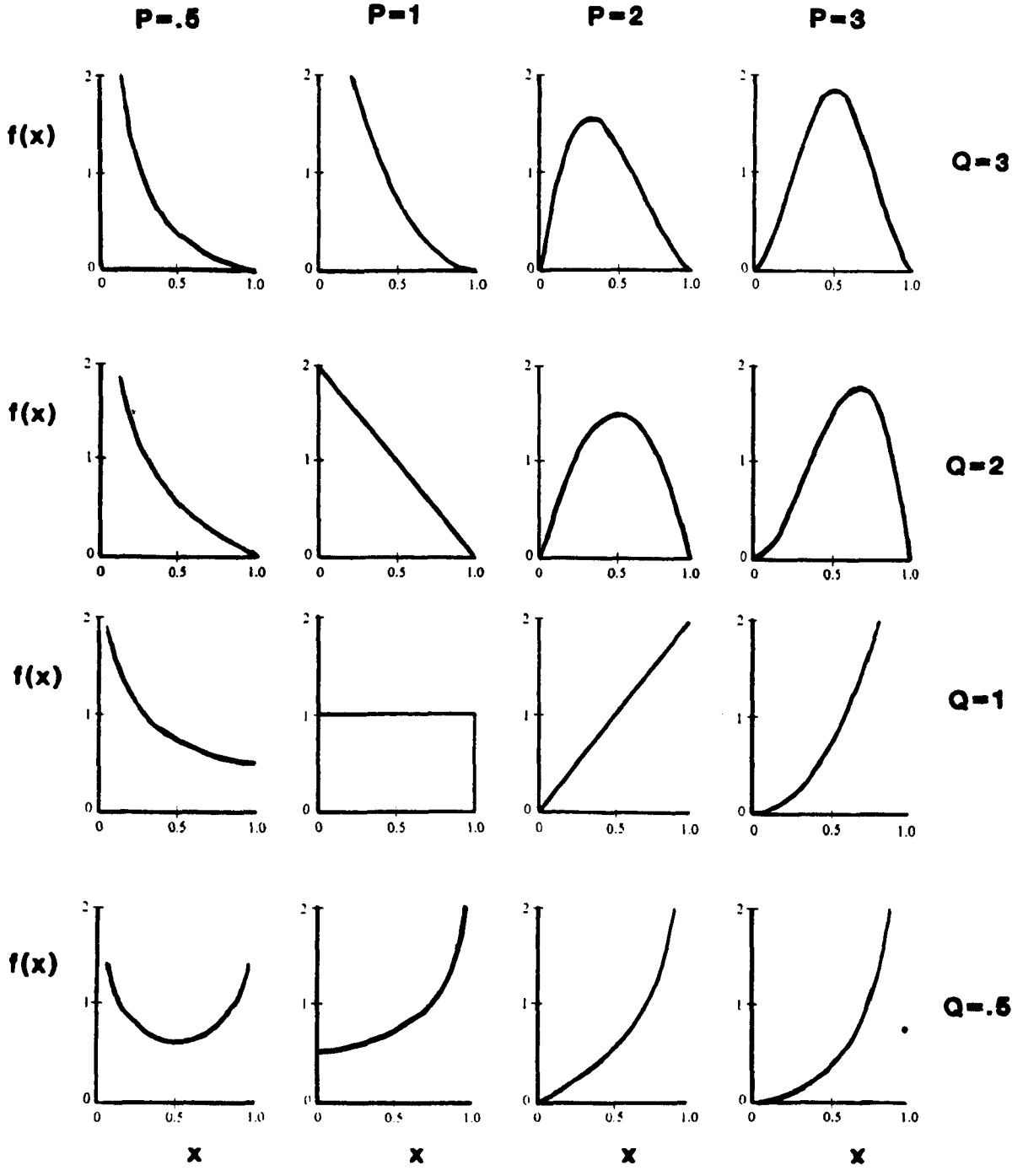


Figure 5. Beta Densities for Various Choices of the Parameters p and q.

Table 3. The Mean and Variance of a Standard Beta Distribution Corresponding to Choices of p and q Used in Figure 5

p	q	E(X)	V(X)	p	q	E(X)	V(X)
.5	.5	.500	.125	2	.5	.800	.046
.5	1	.333	.089	2	1	.667	.056
.5	2	.200	.046	2	2	.500	.050
.5	3	.143	.027	2	3	.400	.040
1	.5	.667	.089	3	.5	.857	.027
1	1	.500	.083	3	1	.750	.038
1	2	.333	.056	3	2	.600	.040
1	3	.250	.038	3	3	.500	.036

The final method used to illustrate the influence of p and q on the mean and variance is shown graphically in Figures 6 through 8. Figure 6 shows that the mean will increase as p increases for a fixed value of q. Figure 7 shows the mean will decrease as q increases for a fixed value of p. Figure 8 shows how p and q jointly influence the variance. It should be noted that the roles of p and q are interchangeable in Figure 8.

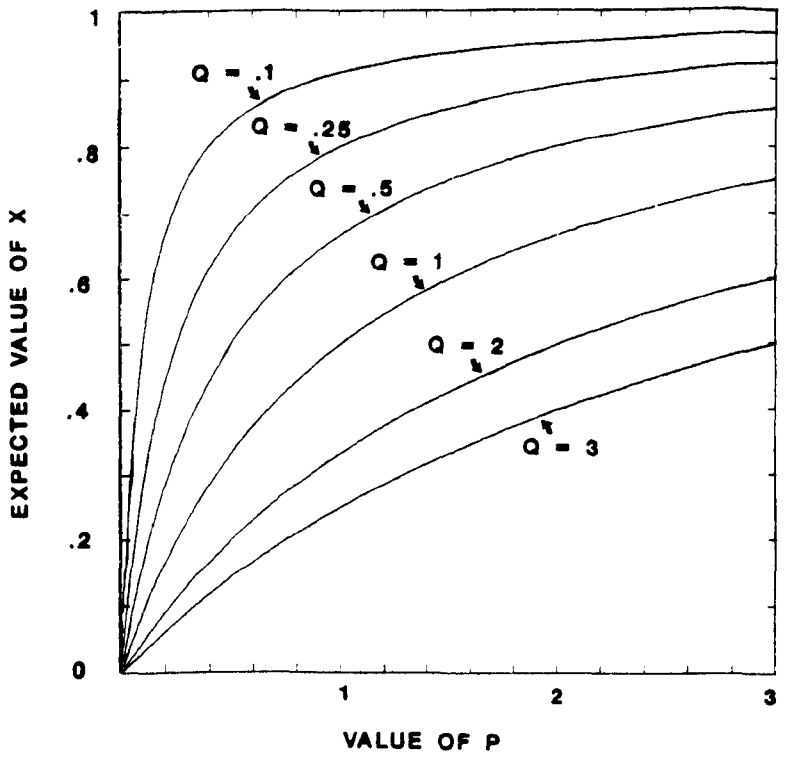


Figure 6. The Influence of Different Choices of P on the Expected Value of X for Various Choices of Q When X has a Standard Beta Distribution

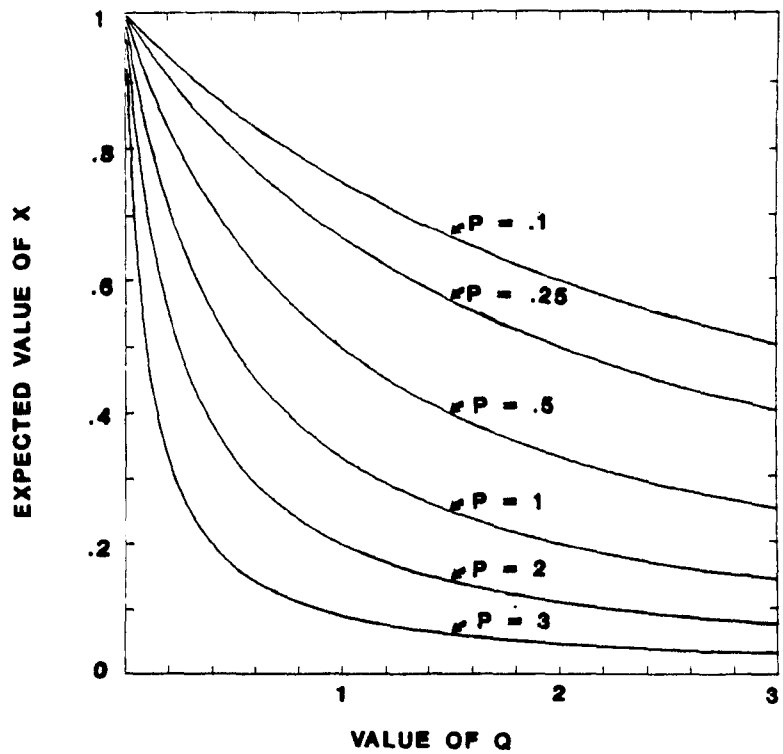


Figure 7. The Influence of Different Choices of Q on the Expected Value of X of X for Various Choices of P When X has a Standard Beta Distribution

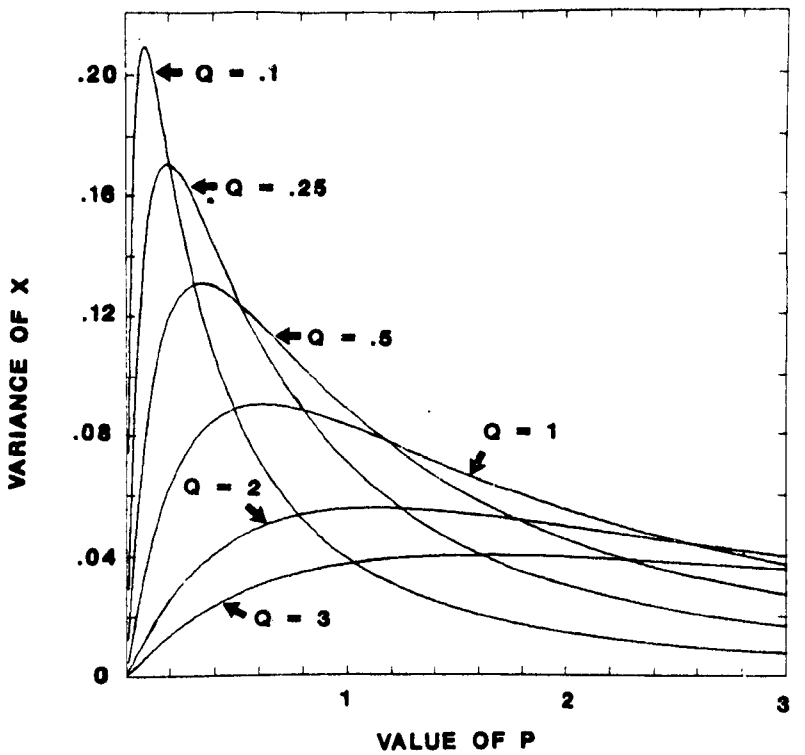


Figure 8. The Role of P and Q on Influencing the Variance of X Where X has a Standard Beta Distribution. The Roles of P and Q are Interchangeable in This Graph.

4. EXAMPLES OF THE USE OF SUBROUTINE USRDST TO GENERATE SAMPLES FROM A USER-SUPPLIED DISTRIBUTION

As mentioned in the previous section, the user may specify the keyword USER DISTRIBUTION whenever it is desired to obtain a sample from a random variable whose distribution is not built into the program. This is done through use of a user-supplied subroutine called USRDST which is given in Table 4. This subroutine is called each time the keyword USER DISTRIBUTION appears. Thus, if the user desires to supply distributions for more than one variable, then the keyword USER DISTRIBUTION must appear once for each such variable. In such a case the user must code subroutine USRDST to function in accordance with the number of the variables being processed. This communication link is established by the argument "J" in the calling list of the subroutine which contains the number of the current variable being sampled. Thus, if a user had 10 input variables and desired to invoke a user-supplied distribution on the 3rd and 7th variables, then the subroutine USRDST is automatically called twice from the main program. On the first call to the subroutine the value J = 3 is supplied by the main program, while on the second call J = 7. A FORTRAN "IF" statement within USRDST could be used to direct one action for the case with J = 3 and a different action for J = 7. Of course, this logic easily extends to more than two variables.

Three examples will now be given to show the use of the subroutine USRDST. The first example provides the setup for sampling from the following discrete probability distribution.

$$
\begin{aligned}
f(x) &= .2 \ , \ x = 0 \\
&= .3 \ , \ x = 1 \\
&= .4 \ , \ x = 2 \\
&= .1 \ , \ x = 3
\end{aligned}
$$

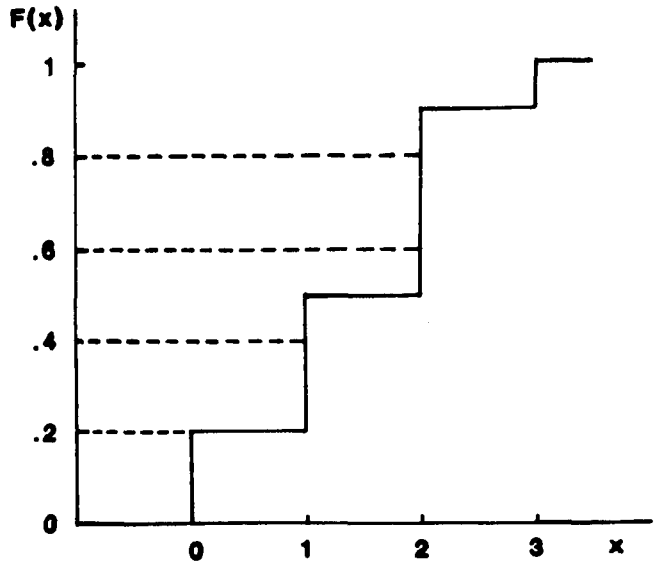


Figure 9. Distribution Function for a User-Supplied Discrete Probability Distribution

The corresponding distribution function appears in Figure 9. For this discrete variable to be sampled, it is necessary to input the probability distribution in the parameter list such as appeared in lines 22 to 27 of Table 2. Subroutine USRDST must be modified by the user to first read in this information so it can be used to construct the distribution function given in Figure 9. Next, the subroutine must use a "DO" loop to move up the vertical axis of the distribution function starting at zero and using n steps each of length PROBINC = 1/n. These intervals will be (0, 1/n), (1/n, 2/n), (2n, 3/n), ..., (n - 1/n, 1). A point R is selected at random in each of these intervals and mapped through the inverse of the distribution function to select the particular value of X. Thus, for a LHS with n = 5 (note the horizontal dashed lines added to Figure 9), the value x = 0 is selected once from the interval (0, .2); x = 1 is selected once from the interval (.2, .4); either x = 1 or x = 2 is selected from the interval (.4, .6) depending on the value that R takes on; the value x = 2 is selected from the interval (.6, .8); and finally either x = 2 or x = 3 is selected from the interval (.8, 1). If a random sample is desired, then each selection of X is made on the interval (0, 1). Once each value is selected, it is stored in the vector X using the LOC function that is defined in the subroutine for the ith observation on the jth variable. The FORTRAN setup for this example appears in Table 4.

Table 4. Subroutine USRDST For Sampling From a User-Supplied Discrete Probability Distribution (Continued on next page)

```
SUBROUTINE USRDST (J)
C
C THE FOLLOWING SIX LINES OF CODE ARE REQUIRED BY USRDST
C
      PARAMETER (NMAX=1000)
      PARAMETER (NVAR=50)
      PARAMETER (LENT=125)
      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
     1             ICORR,IDIST(NVAR)
      COMMON/SAMP/X(NMAX*NVAR)
C
C THE FOLLOWING LINE OF CODE IS SUPPLIED BY THE USER.
C XVAL AND FREQ MUST BE DIMENSIONED TO THE NUMBER OF UNIQUE VALUES
C THAT THE DISCRETE RANDOM VARIABLE TAKES ON AND CDF MUST BE
C DIMENSIONED TO THE NUMBER OF UNIQUE VALUES PLUS 1
C
      DIMENSION XVAL(4),FREQ(4),CDF(5)
C
C THE FOLLOWING FUNCTION DEFINITION IS REQUIRED BY USRDST
C
      LOC(I,J)=(J-1)*N+I
C
C READ IN THE VALUES FOR THE DISCRETE PROBABILITY FUNCTION.
C NP IS THE NUMBER OF UNIQUE VALUES OF THE RANDOM VARIABLE.
C XVAL(K) IS THE KTH UNIQUE VALUE OF THE RANDOM VARIABLE.
C FREQ(K) IS THE PROBABILITY ASSOCIATED WITH THE KTH UNIQUE VALUE.
C NOTE THAT THE READ STATEMENT MUST BE OF THE FORM READ(7,*)....
C
      READ(7,*)NP
      DO 1 K=1,NP
    1 READ(7,*)XVAL(K),FREQ(K)
C
C CONSTRUCT THE CUMULATIVE DISTRIBUTION FUNCTION
C
      CDF(1)=0.0
      DO 2 K=1,NP
    2 CDF(K+1)=CDF(K)+FREQ(K)
C
C SET THE STARTING POINT (STRTPT) EQUAL TO ZERO AND THE PROBABILITY
C INCREMENT (PROBINC) EQUAL TO 1/N FOR A LHS WHERE N IS THE SAMPLE SIZE
C
      STRTPT=0.0
      PROBINC=1.0
C
```

Table 4. Subroutine USRDST For Sampling From a User-Supplied Discrete Probability Distribution (Continued from preceding page)

```
C IF A RANDOM SAMPLE HAS BEEN SPECIFIED IN THE PARAMETER LIST THEN THE
C ARGUMENT IRS HAS BEEN SET EQUAL TO 1 IN THE MAIN PROGRAM, HENCE THE
C PROBABILITY INCREMENT IS SET EQUAL TO 1 SO THAT ALL OBSERVATIONS ARE
C SELECTED BY USING THE INTERVAL (0,1)
C
      IF(IRS.EQ.1)PROBINC=1.0
C
C THIS LOOP WILL OBTAIN THE N SAMPLE VALUES
C
      DO 4 I=1,N
C
C R IS A RANDOMLY SELECTED POINT IN THE CURRENT SUBINTERVAL OBTAINED
C BY USING THE RANDOM NUMBER GENERATOR RAN
C
      R=STRTPT+PROBINC*RAN(ISEED)
C
C THIS LOOP WILL SELECT THE SPECIFIC VALUE OF THE RANDOM VARIABLE
C CORRESPONDING TO R THROUGH THE INVERSE CUMULATIVE DISTRIBUTION
C FUNCTION.THESE VALUES ARE STORED IN THE VECTOR X THROUGH THE
C USE OF THE LOC FUNCTION
C
      DO 3 K=1,NP
         IF(R.GE.CDF(K).AND.R.LT.CDF(K+1))X(LOC(I,J))=XVAL(K)
    3 CONTINUE
C
C RESET THE STARTING POINT TO THE BEGINNING OF THE NEXT SUBINTERVAL
C UNLESS A RANDOM SAMPLE HAS BEEN SPECIFIED
C
      IF(IRS.NE.1)STRTPT=STRTPT+PROBINC
    4 CONTINUE
      RETURN
      END
```

The second example illustrates how to use subroutine USRDST to obtain a sample from an empirical distribution function (i.e., one generated from sample data). Suppose the information available on one of the input variables takes the form of sample data, i.e., the analyst wants to use the sample data with the computer model and not risk introducing additional uncertainty by replacing the sample data with an estimated distribution function. This is easily accomplished by sampling directly from the empirical distribution function formed by the sample data. As an example, suppose there are eight sample data points available as follows: 0.4, 0.9, 1.1, 1.4, 1.9, 2.2, 2.4, 2.7. To provide this information to the program, the following set up could be used.

```
USER DISTRIBUTION

    8    .4    .9    1.1    1.4    1.9    2.2    2.4    2.7
```

The first piece of information after the keyword is the sample size. Once the data points are read in, the subroutine USRDST will construct an empirical distribution function (edf) much as was done in Figure 9, only the stepheights will all be equal (in this case, the stepheight = 1/8). The graph of the edf appears in Figure 10. Thus, for a LHS of size 5 (note the horizontal dashed lines in Figure 6), either the value x = .4 or x = .9 is selected from the interval (0, .2); either x = .9, x = 1.1, or x = 1.4 is selected from the interval (.2, .4); either x = 1.4 or x = 1.9 is selected from the interval (.4, .6); either x = 1.9, x = 2.2, or x = 2.4 is selected from the interval (.6, .8); and finally either x = 2.4 or x = 2.7 is selected from the interval (.8, 1). Of course, a random sample would select each value from the interval (0, 1). The FORTRAN setup for this example appears in Table 5.

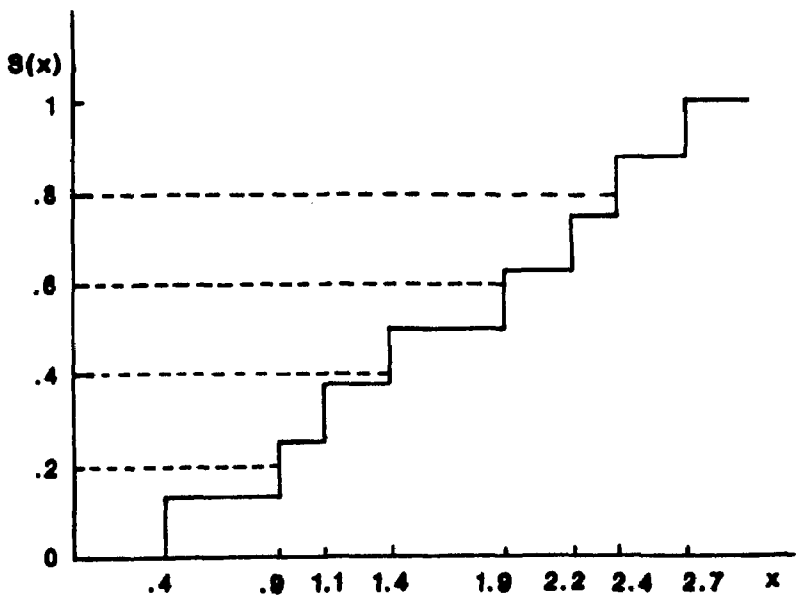


Figure 10. Empirical Distribution Function for User-Supplied Data

Table 5. Setup for Subroutine USRDST to Generate a Sample From An Empirical Distribution Function
(Continued on next page)

```
      SUBROUTINE USRDST (J)
C
C THE FOLLOWING SIX LINES OF CODE ARE REQUIRED BY USRDST
C
      PARAMETER (NMAX=1000)
      PARAMETER (NVAR=50)
      PARAMETER (LENT=125)
      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
     1             ICORR,IDIST(NVAR)
      COMMON/SAMP/X(NMAX*NVAR)
C
C THE FOLLOWING LINE OF CODE IS SUPPLIED BY THE USER.
C SVAL MUST BE DIMENSIONED TO THE NUMBER OF SAMPLE VALUES AND
C EDF MUST BE DIMENSIONED TO THE NUMBER OF SAMPLE VALUES PLUS 1
C
      DIMENSION SVAL(8),EDF(9)
C
C THE FOLLOWING FUNCTION DEFINITION IS REQUIRED BY USRDST
C
      LOC(I,J)=(J-1)*N+I
C
C READ IN THE SAMPLE SIZE NP AND THE SAMPLE VALUES.
C NOTE THAT THE READ STATEMENT MUST BE OF THE FORM READ(7,*)....
C
      READ(7,*)NP,(SVAL(K),K=1,NP)
C
C CONSTRUCT THE EMPIRICAL DISTRIBUTION FUNCTION
C
      STEP=1.0/FLOAT(NP)
      EDF(1)=0.0
      DO 6 K=1,NP
    6 EDF(K+1)=STEP*FLOAT(K)
C
C SET THE STARTING POINT (STRTPT) EQUAL TO ZERO AND THE PROBABILITY
C INCREMENT (PROBINC) EQUAL TO 1/N FOR A LHS WHERE N IS THE SAMPLE SIZE
C
      STRTPT=0.0
      PROBINC=1.0/FLOAT(N)
C
C IF A RANDOM SAMPLE HAS BEEN SPECIFIED IN THE PARAMETER LIST THEN THE
```

Table 5. Setup for Subroutine USRDST to Generate a Sample From An Empirical Distribution Function (Continued from preceding page)

```
C ARGUMENT IRS HAS BEEN SET EQUAL TO 1 IN THE MAIN PROGRAM, HENCE THE
C PROBABILITY INCREMENT IS SET EQUAL TO 1 SO THAT ALL OBSERVATIONS ARE
C SELECTED BY USING THE INTERVAL (0,1)
C
      IF(IRS.EQ.1)PROBINC=1.0
C
C THIS LOOP WILL OBTAIN THE N SAMPLE VALUES
C
      DO 8 I=1,N
C
C R IS A RANDOMLY SELECTED POINT IN THE CURRENT SUBINTERVAL OBTAINED
C BY USING THE RANDOM NUMBER GENERATOR RAN
C
      R=STRTPT+PROBINC*RAN(ISEED)
C
C THIS LOOP WILL SELECT THE SPECIFIC SAMPLE VALUE CORRESPONDING
C TO R THROUGH THE INVERSE EMPIRICAL DISTRIBUTION FUNCTION.
C THESE VALUES ARE STORED IN THE VECTOR X THROUGH THE USE OF THE
C LOC FUNCTION
C
      DO 7 K=1,NP
         IF(R.GE.EDF(K).AND.R.LT.EDF(K+1))X(LOC(I,J))=SVAL(K)
    7 CONTINUE
C
C RESET THE STARTING POINT TO THE BEGINNING OF THE NEXT SUBINTERVAL
C     UNLESS A RANDOM SAMPLE HAS BEEN SPECIFIED
C
      IF(IRS.NE.1)STRTPT=STRTPT+PROBINC
    8 CONTINUE
      RETURN
      END
```

The last example shows how to modify subroutine USRDST to generate samples from more than one user-supplied distribution. For this example, the user-supplied distributions in the two previous examples will both be generated in the same subroutine. It will be assumed that the discrete probability distribution is the third variable (J = 3) specified in the parameter list and that the empirical data correspond to the seventh (J = 7) variable in the parameter list. All that is needed is to pool the FORTRAN statements in Tables 4 and 5, along with the appropriate branching statements. The completed subroutine appears in Table 6.

Table 6. Subroutine USRDST for Sampling from Two Different User-Supplied Distributions

```
      SUBROUTINE USRDST(J)
C
C THE FOLLOWING SIX LINES OF CODE ARE REQUIRED BY USRDST
C
      PARAMETER (NMAX=1000)
      PARAMETER (NVAR=50)
      PARAMETER (LENT=125)
      COMMON/PARAM/TITLE(LENT),ISEED,N,NV,IRS,ICM,NREP,IDATA,IHIST,
     1             ICORR,IDIST(NVAR)
      COMMON/SAMP/X(NMAX*NVAR)
C
C     *****PUT LINES 11 to 17 FROM TABLE 4 HERE
C
C     *****PUT LINES 11 to 16 FROM TABLE 5 HERE
C
C THE FOLLOWING FUNCTION DEFINITION IS REQUIRED BY USRDST
C
      LOC(I,J)=(J-1)*N+I
C
C BRANCH TO STATEMENT 5 IF THE EMPIRICAL DISTRIBUTION IS REQUESTED
C
      IF(J.EQ.7)GO TO 5
C
C     *****PUT LINES 22 to 74 FROM TABLE 4 HERE (CDF CASE)
C
      RETURN
    5 CONTINUE
C
C     *****PUT LINES 21 to 69 FROM TABLE 5 HERE (EDF CASE)
C
      RETURN
      END
```

5. MODIFYING THE COMPUTER PROGRAM

The computer program has been written using the FORTRAN-77 language with the attempt to make the code as machine-independent (i.e., portable) as possible. This has been done by using the generic name for intrinsic functions whenever possible and also by avoiding the use of nonstandard syntax. However, certain items in the code are by their nature machine-dependent and must be adjusted before the code will run.

Machine Constants

The first items to be adjusted are the machine-dependent constants. These constants are set in functions I1MACH (integer machine constants) and R1MACH (floating-point machine constants). To alter these functions for a particular machine or environment, the desired set of constants should be activated by removing the "C" from column 1 in the program. (For a more detailed discussion, see the comments at the beginning of the functions I1MACH and R1MACH.)

Random Number Generator

In Section 2 of this report, the discussion of the keyword RANDOM SEED indicated that the possible range of values for this parameter was machine-dependent. In order to take advantage of the full machine range, the length (i.e., number of digits) allotted for the random seed must be adjusted. This length should be set to the number of digits in the largest integer representable by the machine plus one extra place to allow for the use of a sign. The current value for this length is based upon the VAX 11/780 which is $2^{31} - 1$ or 2147483653. By allowing an extra place for the sign (which is necessary since the VAX 11/780 permits the use of negative integers for a random seed), the length required is 11. This is the length used in the code. However, as an example, suppose the code was to run on a CDC 7600 machine. There the largest integer represented by the machine is $2^{48} - 1$ or 2.815×10^{14}. Including an extra place for the sign results in a length of 16. This new length must be inserted into the code in the following places:

1. SUBROUTINE BANNER -- in the FORMAT statement labeled 9001 the I11 should be changed to I16.

2. SUBROUTINE DATSQZ --in the PARAMETER statement, LENT = 11 should be changed to LENT = 16.

3. SUBROUTINE RDPAR -- in the PARAMETER statement, LENTC = 11 should be changed to LENTC = 16.

4. SUBROUTINE RDPAR -- in the FORMAT statement labeled 9003, the I11 should be changed to I16.

Different machines will also vary in the way the random number generator is accessed. The current version of the code uses the VAX 11/780 function RAN to obtain a pseudo-random number from the range (0, 1). The random seed is

placed in the variable ISEED and this variable is constantly being updated so that it contains the most recent value of the random seed. If the random number generator being used operates differently from this, the following adjustments will have to be made to the code:

1. If the random number generator needs to be preset, this should be done in subroutine RDPAR.

2. If the random number generator is called differently, these changes should be made in subroutines BETA, MIX, NORMAL, TRIANG, and UNIFRM.

3. If a separate call to the random number generator is required to retrieve the current value of the random seed, this should be done in subroutine BANNER and only after the first repetition.

Redimensioning

Section 2 indicated upper limits on the values of certain parameters. These were:

1. the number of observations, NMAX=1000

2. the number of variables, NVAR=50

3. the number of pairs of correlated variables, NCVAR=50

4. the number of subintervals in the UNIFORM* and LOGUNIFORM* distribution, NINTMX=50

These upper limits should be satisfactory for most situations. However, if any or all of these upper limits need to be adjusted, the new value must be replaced in every occurrence of the parameter. Table 7 shows every subroutine in which each of the above four parameters occurs. The PARAMETER statements are found at the beginning of each subroutine.

Table 7. List of Subroutines which Contain Parameters that may be Adjusted for Purposes of Redimensioning.

PARAMETER	SUBROUTINES
NMAX	LHS (main program), BETA, CORCAL, COROUT, DATOUT, FINDIT, HISTO, HPSRT, HSTOUT, MIX, NORMAL, OUTDAT, RANKER, RDPAR, TRIANG, UNIFRM, USRDST
NVAR	LHS (main program), BANNER, BETA, CHKDIM, CHKSTR, CHLSKY, CMCRD, CORCAL, COROUT, DATOUT, DMFSD, DSINV, FINDIT, HISTO, HPSRT, HSTOUT, MATINV, MIX, NORMAL, OUTCRD, OUTDAT, PMTRX, POSDEF, RANKER, RDPAR, SETDEF, TRIANG, UNIFRM, USRDST, VIF, WRTCRD, WRTPAR
NCVAR	CMCRD, RDPAR
NINTMX	CHKSTR, RDPAR

The Output File

The last item to be considered is the output file. The sample generated by the code is written to unit 1 in unformatted binary. Each record on the output file represents one input vector of variables and is of the form I, K, (X(J),J=1,K) where I is the number of the vector being written; K is the number of variables in the vector and X is the vector of values. There are n of these vectors written to unit 1 where n is the number of observations specified by the user through the use of the keyword NOBS. If more than one repetition has been specified (NREPS>1), then each complete sample will be written on unit 1 with no separators between repetitions. The definition of unit 1 for mass storage is the responsibility of the user as is the procedure for actually saving the output file.

REFERENCES

Iman, R. L. and Conover, W. J. (1982a). "A Distribution-Free Approach to Inducing Rank Correlation Among Input Variables," Communications in Statistics, B11(3), 311-334.

Iman, R. L. and Conover, W. J. (1982b). "Sensitivity Analysis Techniques: Self-Teaching Curriculum," Nuclear Regulatory Commission Report, NUREG/CR-2350, Technical Report SAND81-1978, Sandia National Laboratories, Albuquerque, New Mexico 87185.

Iman, R. L. and Davenport, J. M. (1982). "An Iterative Algorithm to Produce a Positive Definite Correlation Matrix from an "Approximate Correlation Matrix" (With a Program User's Guide)," Technical Report SAND81-1376, Sandia National Laboratories, Albuquerque, New Mexico 87185.

Iman, R. L., Davenport, J. M., and Zeigler, D. K. (1980). "Latin Hypercube Sampling (Program User's Guide)," Technical Report SAND79-1473, Sandia National Laboratories, Albuquerque, New Mexico 87185.

Iman, R. L. and Helton, J. C. (1983). "A Comparison of Uncertainty and Sensitivity Analysis Techniques for Computer Models." In preparation.

Iman, R. L., Helton, J. C., and Campbell, J. E. (1981a). "An Approach to Sensitivity Analysis of Computer Models, Part 1. Introduction, Input Variable Selection and Preliminary Variable Assessment," Journal of Quality Technology, 13(3), 174-183.

Iman, R. L., Helton, J. C., and Campbell, J. E. (1981b). "An Approach to Sensitivity Analysis of Computer Models, Part 2. Ranking of Input Variables, Response Surface Validation, Distribution Effect and Technique Synopsis," Journal of Quality Technology, 13(4), 232-240.

Marquardt, D. W. and Snee, R. D. (1975). "Ridge Regression in Practice," The American Statistician, 29(1), 3-20.

Marquardt, D. W. (1970). "Generalized Inverses, Ridge Regression, Biased Linear Estimation, and Nonlinear Estimation," Technometrics, 12(3), 591-612.

McKay, M. D., Conover, W. J., and Beckman. R. J. (1979). "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," Technometrics, 21, 239-245.

Steck, G. P., Iman, R. L., and Dahlgren, D. A. (1976). "Probabilistic Analysis of LOCA, Annual Report for FY1976," Technical Report SAND76-0535 (NUREG 766513), Sandia National Laboratories, Albuquerque, New Mexico 87185.

APPENDIX

Example of Program Output

This appendix presents sample output from two runs of the computer program. The first computer run is based on the list of parameters given below. This setup calls for the generation of one Latin hypercube sample consisting of 29 observations on 7 input variables. Two of these variables have user supplied distributions corresponding to the setup for subroutine USRDST in Table 6 of Section 4. Two of the remaining five variables use a normal distribution while the beta, lognormal, and loguniform distributions are each used once. The first two pages of output echo back the parameter setup. The DATA option with the OUTPUT parameter produces pages 3 and 4 of the output which contain the actual sample generated and the corresponding ranks. The raw and rank correlation matrices appear on pages 5 and 6 of the output. These matrices were requested through use of the CORR option with the OUTPUT parameter. The output option HIST is not exercised in this example.

```
TITLE - SETUP FOR LHS OUTPUT EXAMPLE 1
RANDOM SEED    -1692990931
NOBS   29
BETA       BETA DISTRIBUTION ON (10, 100)
     10     100          .5    .2
NORMAL    THIS WILL BE THE SECOND VARIABLE
      12      56
USER DISTRIBUTION       DISCRETE PROBABILITY DISTRIBUTION
    4
    0     .2
    1     .3
    2     .4
    3     .1
LOGNORMAL    EXAMPLE OF LOGNORMAL DISTRIBUTION
   .01      2.13
NORMAL    DISTRIBUTIONS MAY BE USED MORE THAN ONCE
   0      10
LOGUNIFORM
    6.E7    8.1E10
USER DISTRIBUTION      EMPIRICAL DATA
  8  .4   .9   1.1   1.4   1.9   2.2   2.4   2.7
OUTPUT    DATA    CORR
```

PAGE 1 OF COMPUTER OUTPUT FOR EXAMPLE 1 - ECHO OF INPUT PARAMETERS

```
TITLE - SETUP FOR LHS OUTPUT EXAMPLE 1
RANDOM SEED = -1692990931
NUMBER OF VARIABLES =   7
NUMBER OF OBSERVATIONS =   29
THE SAMPLE INPUT VECTORS WILL BE PRINTED ALONG WITH THEIR CORRESPONDING RANKS
THE CORRELATION MATRICES (RAW DATA AND RANK CORRELATIONS) WILL BE PRINTED
```

PAGE 2 OF COMPUTER OUTPUT FOR EXAMPLE 1 - ECHO OF INPUT DISTRIBUTIONS

```
TITLE - SETUP FOR LHS OUTPUT EXAMPLE 1

VARIABLE  DISTRIBUTION          RANGE              LABEL

    1     BETA            10.0     TO   100.       BETA DISTRIBUTION ON (10,100)
          WITH PARAMETERS  P =  0.50  Q =  2.00
          THIS CHOICE OF PARAMETERS GIVES A
          POPULATION MEAN OF   28.0      AND A
          POPULATION VARIANCE OF   370.

    2     NORMAL          12.0     TO   56.0       THIS WILL BE THE SECOND VARIABLE

    3     USER SUPPLIED DISTRIBUTION               DISCRETE PROBABILITY DISTRIBUTION

    4     LOGNORMAL     1.000E-02 TO   2.13        EXAMPLE OF LOGNORMAL DISTRIBUTION

    5     NORMAL        0.000E+00 TO   10.0        DISTRIBUTIONS MAY BE USED MORE THAN ONCE

    6     LOGUNIFORM    6.000E+07 TO  8.100E+10

    7     USER SUPPLIED DISTRIBUTION               EMPIRICAL DATA
```

PAGE 3 OF COMPUTER OUTPUT FOR EXAMPLE 1 - ACTUAL LHS SAMPLE GENERATED

TITLE - SETUP FOR LHS OUTPUT EXAMPLE 1

LATIN HYPERCUBE SAMPLE INPUT VECTORS

RUN NO.	X(1)	X(2)	X(3)	X(4)	X(5)	X(6)	X(7)
1	41.3	28.0	1.00	0.175	5.81	1.694E+10	0.400
2	11.2	40.9	2.00	7.095E-02	5.31	1.555E+09	1.10
3	34.3	34.5	1.00	0.194	5.49	7.505E+10	1.10
4	84.0	40.4	2.00	0.443	5.13	1.606E+08	1.90
5	10.1	26.5	2.00	0.212	3.51	1.855E+10	0.400
6	12.9	35.2	0.000E+00	0.161	4.71	3.533E+10	2.20
7	10.0	33.3	2.00	7.623E-02	6.41	2.400E+08	1.40
8	31.5	38.8	2.00	0.695	4.48	5.566E+10	2.40
9	69.3	37.3	3.00	0.115	4.64	2.973E+10	1.90
10	21.9	28.6	1.00	4.207E-02	5.56	3.959E+10	2.40
11	24.9	22.6	2.00	0.146	10.0	4.106E+08	1.40
12	19.7	25.6	2.00	3.918E-02	3.13	8.003E+07	2.70
13	17.7	35.6	0.000E+00	0.138	2.31	6.047E+08	0.900
14	16.9	29.4	1.00	0.930	4.91	1.473E+09	2.70
15	56.2	30.8	1.00	1.495E-02	6.01	3.000E+08	2.20
16	43.8	32.4	0.000E+00	0.102	4.13	2.468E+09	2.40
17	11.2	31.8	3.00	0.368	7.10	3.031E+09	2.40
18	61.5	29.8	0.000E+00	0.129	7.51	5.730E+09	1.10
19	10.3	44.1	1.00	0.302	4.01	3.805E+09	1.90
20	15.9	32.5	2.00	0.104	3.70	7.190E+09	2.20
21	20.4	20.1	1.00	0.379	5.75	1.090E+09	1.40
22	13.3	39.4	0.000E+00	5.923E-02	4.93	9.204E+09	0.900
23	29.1	46.4	1.00	0.258	6.90	2.073E+08	0.400
24	11.8	50.9	1.00	0.177	6.14	1.139E+08	2.70
25	10.6	37.7	2.00	9.040E-02	6.73	4.227E+09	0.900
26	48.1	34.1	2.00	0.241	2.85	6.475E+07	0.400
27	36.0	42.1	3.00	5.168E-02	4.35	1.[illegible]55E+10	2.20
28	14.4	23.9	1.00	0.286	1.89	8.762E+08	1.40
29	27.5	36.5	2.00	8.675E-02	3.24	4.670E+08	1.10

PAGE 4 OF COMPUTER OUTPUT FOR EXAMPLE 1 - RANKS OF SAMPLE VALUES BY VARIABLE

TITLE - SETUP FOR LHS OUTPUT EXAMPLE 1

RANKS OF LATIN HYPERCUBE SAMPLE INPUT VECTORS

RUN NO.	X(1)	X(2)	X(3)	X(4)	X(5)	X(6)	X(7)
1	23.	6.	11.	17.	21.	23.	3.
2	6.	25.	21.	6.	17.	14.	10.
3	21.	16.	11.	19.	18.	29.	10.
4	29.	24.	21.	27.	16.	4.	17.
5	2.	5.	21.	20.	6.	24.	3.
6	8.	17.	3.	16.	13.	26.	21.
7	1.	14.	21.	7.	24.	6.	14.
8	20.	22.	21.	28.	11.	28.	25.
9	28.	20.	28.	12.	12.	25.	17.
10	16.	7.	11.	3.	19.	27.	25.
11	17.	2.	21.	15.	29.	8.	14.
12	14.	4.	21.	2.	4.	2.	28.
13	13.	18.	3.	14.	2.	10.	6.
14	12.	8.	11.	29.	14.	13.	28.
15	26.	10.	11.	1.	22.	7.	21.
16	24.	12.	3.	10.	9.	15.	25.
17	5.	11.	28.	25.	27.	16.	25.
18	27.	9.	3.	13.	28.	19.	10.
19	3.	27.	11.	24.	8.	17.	17.
20	11.	13.	21.	11.	7.	20.	21.
21	15.	1.	11.	26.	20.	12.	14.
22	9.	23.	3.	5.	15.	21.	6.
23	19.	28.	11.	22.	26.	5.	3.
24	7.	29.	11.	18.	23.	3.	28.
25	4.	21.	21.	9.	25.	18.	6.
26	25.	15.	21.	21.	3.	1.	3.
27	22.	26.	28.	4.	10.	22.	21.
28	10.	3.	11.	23.	1.	11.	14.
29	18.	19.	21.	8.	5.	9.	10.

PAGE 5 OF COMPUTER OUTPUT FOR EXAMPLE 1 - RAW DATA CORRELATION MATRIX

```
TITLE - SETUP FOR LHS OUTPUT EXAMPLE 1
CORRELATIONS AMONG INPUT VARIABLES CREATED BY THE LATIN HYPERCUBE SAMPLE FOR RAW DATA
    1 1.0000
    2 0.0363 1.0000
    3 0.0791 0.0508 1.0000
    4 0.0010-0.0389 0.0323 1.0000
    5 0.0737-0.0266 0.0544-0.0343 1.0000
    6 0.0796 0.0222-0.0230 0.1003-0.0224 1.0000
    7-0.0138 0.0483 0.1114 0.2271-0.0203 0.0737 1.0000
            1      2      3      4      5      6      7
VARIABLES
THE VARIANCE INFLATION FACTOR FOR THIS MATRIX IS  1.07
```

PAGE 6 OF COMPUTER OUTPUT FOR EXAMPLE 1 - RANK CORRELATION MATRIX

```
TITLE - SETUP FOR LHS OUTPUT EXAMPLE 1
CORRELATIONS AMONG INPUT VARIABLES CREATED BY THE LATIN HYPERCUBE SAMPLE FOR RANK DATA
    1 1.0000
    2-0.0409 1.0000
    3-0.0306 0.0881 1.0000
    4-0.0232-0.0153-0.0301 1.0000
    5 0.0049-0.0030-0.0306-0.0089 1.0000
    6-0.0192-0.0054-0.0678-0.0187-0.0084 1.0000
    7-0.0109-0.0549 0.0902-0.0226-0.0223 0.0030 1.0000
            1      2      3      4      5      6      7
VARIABLES
THE VARIANCE INFLATION FACTOR FOR THIS MATRIX IS  1.02
```

The second computer run is based on the list of parameters given below. This setup differs from the first example in two respects. First, a rank correlation is specified involving variables X_1, X_2, and X_5. However, the specified rank correlations do not form a positive definite matrix. Hence, as explained in Section 2 under the keyword CORRELATION MATRIX, an iterative scheme (Iman and Davenport, 1982) built into the program adjusts the correlation matrix to make it positive definite. The output shows that the user specified correlations of .8, .7, and -.6 have been adjusted respectively to .5872, .4990, and -.4078 with the corresponding sample rank correlations being .5926, .5821, and -.3419. The second difference in this example is that the DATA option has been omitted from the OUTPUT list as the sample values are the same as before only paired differently to reflect the new correlation structure.

```
TITLE - SETUP FOR LHS OUTPUT EXAMPLE 2
RANDOM SEED     -1692990931
NOBS     29
BETA     BETA DISTRIBUTION ON (10, 100)
     10     100          .5     .2
NORMAL     THIS WILL BE THE SECOND VARIABLE
     12     56
USER DISTRIBUTION     DISCRETE PROBABILITY DISTRIBUTION
     4
     0     .2
     1     .3
     2     .4
     3     .1
LOGNORMAL     EXAMPLE OF LOGNORMAL DISTRIBUTION
     .01        2.13
NORMAL     DISTRIBUTIONS MAY BE USED MORE THAN ONCE
     0     10
LOGUNIFORM
     6.E7     8.1E10
USER DISTRIBUTION     EMPIRICAL DATA
     8    .4    .9    1.1    1.4    1.9    2.2    2.4    2.7
CORRELATION MATRIX
     3    1    2    .8    1    5    .7    2    5    -.6
OUTPUT     CORR
```

PAGE 1 OF COMPUTER OUTPUT FOR EXAMPLE 2 - ECHO OF INPUT PARAMETERS

```
TITLE - SETUP FOR LHS OUTPUT EXAMPLE 2
RANDOM SEED = -1692990931
NUMBER OF VARIABLES =   7
NUMBER OF OBSERVATIONS =   29
AN INPUT CORRELATION MATRIX HAS BEEN SPECIFIED
THE CORRELATION MATRICES (RAW DATA AND RANK CORRELATIONS) WILL BE PRINTED
```

PAGE 2 OF COMPUTER OUTPUT FOR EXAMPLE 2 - ECHO OF INPUT DISTRIBUTIONS

```
TITLE - SETUP FOR LHS OUTPUT EXAMPLE 2

VARIABLE  DISTRIBUTION          RANGE              LABEL

    1     BETA            10.0     TO   100.       BETA DISTRIBUTION ON (10,100)
          WITH PARAMETERS  P =  0.50  Q =  2.00
          THIS CHOICE OF PARAMETERS GIVES A
          POPULATION MEAN OF   28.0      AND A
          POPULATION VARIANCE OF   370.

    2     NORMAL          12.0     TO   56.0       THIS WILL BE THE SECOND VARIABLE

    3     USER SUPPLIED DISTRIBUTION               DISCRETE PROBABILITY DISTRIBUTION

    4     LOGNORMAL     1.000E-02 TO   2.13        EXAMPLE OF LOGNORMAL DISTRIBUTION

    5     NORMAL        0.000E+00 TO   10.0        DISTRIBUTIONS MAY BE USED MORE THAN ONCE

    6     LOGUNIFORM    6.000E+07 TO  8.100E+10

    7     USER SUPPLIED DISTRIBUTION               EMPIRICAL DATA
```

PAGE 3 OF COMPUTER OUTPUT FOR EXAMPLE 2 - ECHO OF USER INPUT RANK CORRELATION MATRIX WITH WARNING MESSAGE

```
TITLE - SETUP FOR LHS OUTPUT EXAMPLE 2

INPUT RANK CORRELATION MATRIX

    1 1.0000

    2 0.8000 1.0000

    5 0.7000-0.6000 1.0000

            1      2      5

VARIABLES

********* CAUTION USER PLEASE NOTE ********* CAUTION USER PLEASE NOTE ********* CAUTION USER PLEASE NOTE **********

THE INPUT RANK CORRELATION MATRIX IS NOT POSITIVE DEFINITE
AN ITERATIVE PROCEDURE HAS BEEN USED TO PRODUCE A SUBSTITUTE RANK CORRELATION MATRIX
THIS ADJUSTED RANK CORRELATION MATRIX APPEARS ON THE NEXT PAGE
THE USER SHOULD EXAMINE THIS MATRIX TO MAKE SURE THAT THE CORRELATION REQUIREMENTS ARE STILL SATISFIED

*****************************************************************************************************************
```

PAGE 4 OF COMPUTER OUTPUT FOR EXAMPLE 2 - ADJUSTED RANK CORRELATION MATRIX

```
TITLE - SETUP FOR LHS OUTPUT EXAMPLE 2

ADJUSTED RANK CORRELATION MATRIX

    1 1.0000

    2 0.5872 1.0000

    5 0.4990-0.4078 1.0000

            1      2      5

VARIABLES
```

PAGE 5 OF COMPUTER OUTPUT FOR EXAMPLE 2 - RAW DATA CORRELATION MATRIX

```
TITLE - SETUP FOR LHS OUTPUT EXAMPLE 2
CORRELATIONS AMONG INPUT VARIABLES CREATED BY THE LATIN HYPERCUBE SAMPLE FOR RAW DATA
    1 1.0000
    2 0.6441 1.0000
    3 0.0791 0.0646 1.0000
    4 0.0010 0.0189 0.0323 1.0000
    5 0.3411-0.3778-0.0729 0.0828 1.0000
    6 0.0796 0.1140-0.0230 0.1003-0.0040 1.0000
    7-0.0138 0.0285 0.1114 0.2271-0.0319 0.0737 1.0000
            1      2      3      4      5      6      7
VARIABLES
```

PAGE 6 OF COMPUTER OUTPUT FOR EXAMPLE 2 - RANK CORRELATION MATRIX

```
TITLE - SETUP FOR LHS OUTPUT EXAMPLE 2
CORRELATIONS AMONG INPUT VARIABLES CREATED BY THE LATIN HYPERCUBE SAMPLE FOR RANK DATA
    1 1.0000
    2 0.5926 1.0000
    3-0.0306 0.0249 1.0000
    4-0.0232-0.0089-0.0301 1.0000
    5 0.5281-0.3419-0.0925 0.0355 1.0000
    6-0.0192-0.0236-0.0678-0.0187-0.0202 1.0000
    7-0.0109-0.0459 0.0902-0.0226 0.0511 0.0030 1.0000
            1      2      3      4      5      6      7
VARIABLES
```

DISTRIBUTION: SAND83-2365, NUREG/CR-3624

US NRC Distribution Contractor (CDSI)
7300 Pearl Street
Bethesda, MD 20014
250 copies for RG

US NRC
Division of Risk Analysis
Washington, DC 20555
M. Cunningham (100)
J. Johnson (5)
L. Lancaster (5)
T. Margulies (5)
D. Rasmuson (5)

Los Alamos Scientific Laboratories (5)
Group S1, MS 606
Los Alamos, NM 87545
Attn: M. Bryson
R. J. Beckman
M. E. Johnson
M. D. McKay
R. A. Waller

Ioannis G. Bartzis
Greek Atomic Energy Commission
Nuclear Research Center Demokritos
Aghia Paraskevi
Attikis
GREECE

Paul Baybutt
Battelle Laboratories
505 King Avenue
Columbus, OH 43201

A. Bayer
INR-Kernforschungszentrum Karlsruhe
D-7500 Karlsruhe 1
Postfach 3640
WEST GERMANY

Carl A. Bennett
Human Affairs Research Center, Battelle
P. O. Box C-5395
Seattle, WA 98105

Nicholas Birkett
Department of Clinical Epidemiology
& Biostatistics
McMaster University
1200 Main Street
Hamilton, Ontario, L8N 3Z5
CANADA

Thomas A. Bishop
Battelle
505 King Avenue
Columbus, OH 43201

Carla Brofferio
Comitato Nazionale per l'Energia Nucleare
Viale Regina Margherita, 125
Casella Postale N. 2358
I-00100 Roma A.D.
ITALY

Robert Budnitz
Future Resources Associates
734 The Alemeda
Berkeley, CA 94707

Anthony R. Buhl
Technology for Energy Corp.
10770 Dutch Town Road
Knoxville, TN 37922

Klaus Burkart
Institut fur Neutronenphysik and
Reaktortechnik (INR)
Kernforschungszentrum Karlsruhe G.m.b.H.
Postfach 3640
D-7500 Karlsruhe 1
WEST GERMANY

Pietro Cagnetti
PAS
Comitato Nazionale per l'Energia Nucleare
Centro di Studi Nucleari della Casaccia
Via Anguillarese km 1+300
I-00060 Roma
ITALY

James E. Campbell (2)
Intera Environmental Consultants Inc.
11999 Katy Freeway
Houston, TX 77079

S. Chakraborty
Abtielung fur die Sicherheit der Kernanlagen
Eidgenossisches Amt fur Energiewirtschaft
Wurenlingen
SWITZERLAND

Alistair D. Christie
Deputy Director, Air Quality and
Inter-Environmental Research Branch
Environment Canada
Atmospheric Environment Service
4905 Dufferin Street
City of North York, Downsview
Ontario, M3H 5T4
CANADA

W. J. Conover
College of Business Administration
Texas Tech University
Lubbock, TX 79409

J. M. Davenport
Department of Mathematics
Texas Tech University
Lubbock, TX 79409

Pamela Doctor (2)
Battelle Northwest
P. O. Box 999
Richland, WA 99352

Darryl Downing
Computer Sciences
Building 2029, P. O. Box X
ORNL
Oak Ridge, TN 37830

Ove Edlund
Studsvik Energiteknik AB
Studsvik
Fack
S-611 82 Nykoping 1
SWEDEN

Bert Th. Eendebak
KEMA Laboratories
Utrechtseweg, 310
Postbus 9035
NL-6800 ET Arnhem
NETHERLANDS

Daniel Egan (2)
Office of Radiation Programs (ANR-460)
U.S. Environmental Protection Agency
Washington, DC 20460

Brenda Ganheart
Minerals Management Service
P. O. Box 7944
Metairie, LA 70010-7944

R. H. Gardner
Environmental Sciences Division
ORNL
Oak Ridge, TN 37830

Dave G. Gosslee (6)
P. O. Box Y, Bldg. 9704-1
ORNL
Oak Ridge, TN 37830

Richard Gunst (5)
Department of Statistics
Southern Methodist University
Dallas, TX 75275

Allan Gutjahr
Department of Mathematics
NMIMT
Socorro, NM 87801

William V. Harper (2)
Performance Analysis Department
Battelle
505 King Avenue
Columbus, OH 43201

Michael Haynes
United Kingdom Atomic Energy Authority
Safety & Reliability Directorate
Wigshaw Lane
Culcheth
Warrington WA3 4NE
UNITED KINGDOM

F. O. Hoffman
Health and Safety Research Division
ORNL
Oak Ridge, TN 37830

Stephen C. Hora
College of Business Administration
Texas Tech University
Lubbock, TX 79409

Frank W. Horsch
Kernforschungszentrum Karlsruhe G.m.b.H.
Postfach 3640
D-7500 Karlsruhe 1
WEST GERMANY

Toshinori Iijima
Division of Reactor Safety Evaluation
Reactor Safety Research Center
Japan Atomic Energy Research Institute
Tokai Research Establishment
Tokai-mura
Naka-gun
Ibaraki-ken 319-11
JAPAN

Janeen Judah
Department of Petroleum Engineering
Texas A&M University
College Station, TX 77843

Geoffrey D. Kaiser
Consulting Division
NUS Corporation
910 Clopper Road
Gaithersburg, MD 20878

Samuel C. Kao (2)
Applied Mathematics, 515
Brookhaven National Laboratory
Upton, NY 11973

Dean C. Kaul
Science Applications, Inc.
Suite 819
1701 East Woodfield Road
Schaumburg, IL 60195

G. Neale Kelly
National Radiological Protection Board
Chilton
Didcot
Oxon. OX11 ORQ
UNITED KINGDOM

K. E. Kemp
Department of Statistics
Kansas State University
Manhattan, KS 66502

Jan G. Kretzschmar
Studiecentrum voor Kernenergie (SCK/CEN)
Boeretang, 200
B-2400 Mol
BELGIUM

Daniel Manesse
Institut de Protection et de
Surete Nucleaire (IPSN)
Commissariat a l'Energie Atomique
Centre d'Etudes Nucleaires de
Fontenay-aux-Roses
Boite Postale 6
F-92260 Fontenay-aux-Roses
FRANCE

David S. Margoles (2)
Mathematics and Statistics Division
Lawrence Livermore Laboratory
P. O. Box 808 (L-316)
Livermore, CA 94550

Scott Mathews
EG&G Idaho
M.S. TSB
P. O. Box 1625
Idaho Falls, ID 83415

Marise Mikulis
Arthur D. Little, Inc.
Alcorn Park
Cambridge, MA 02140

Shan Nair
Research Division
Central Electricity Generating Board
Berkeley Nuclear Laboratories
Berkeley
Gloucestershire GL13 9PB
UNITED KINGDOM

William Nixon
United Kingdom Atomic Energy Authority
Safety & Reliability Directorate
Wigshaw Lane
Culcheth
Warrington WA3 4NE
UNITED KINGDOM

Don Paddleford
Westinghouse
P. O. Box 355
Pittsburgh, PA 15230

Norman C. Rasmussen
Department of Nuclear Engineering
Massachusetts Institute of Technology
77 Mass Avenue
Cambridge, MA 02139

Mark Reeves (3)
Intera Environmental Consultants
11511 Katy Freeway
Suite 630
Houston, TX 77079

Ilkka Savolainen
Technical Research Centre of Finland
Nuclear Engineering Laboratory
P. O. Box 169
SF-00181 Helsinki 18
FINLAND

Sebastiano Serra
ENEL-DCO
Ente Nazionale per l'Energia Elettrica
Via G.B. Martini, 3
Casella Postale N. 386
I-00186 Roma
ITALY

Juan Bagues Somonte
Junta de Energia Nuclear
Ciudad Universitaria
Avenida Complutense, 22
Madrid-3
SPAIN

David A. Stanners
Commission on European Communities
Joint Research Center
Ispra Establishment
21020 Ispra (Varese)
ITALY

John R. D. Stoute
Health Physics Division
Energieonderzoek Centrum Nederland (ECN)
Westerduinweg, 3
Postbus 1
NL-1755 Petten ZG
NETHERLANDS

Marian Stubna
Nuclear Power Plants
Research Institute
VUJE
919 31 Jaslovske Bohunice
orkes Trnava
CZECHOSLOVAKIA

Søren Thykier-Nielsen
Health Physics Department
Risø National Laboratory
Postbox 49
DK-4000 Roskilde
DENMARK

Ulf Tveten
Institute for Energy Technology
Postboks 40
N-2007 Kjeller
NORWAY

I. B. Wall
Electric Power Research Institute
3412 Hillview Avenue
P. O. Box 10412
Palo Alto, CA 94303

Keith Woodard
Pickard, Lowe & Garrick, Inc.
1200 18th Street, NW
Suite 612
Washington, DC 20036

Jon Young
Energy Incorporated
515 West Harrison
Suite 220
Kent, WA 98031

Eric R. Ziegel
Standard Oil Company (Indiana)
Amoco Research Center
P. O. Box 400
Naperville, IL 60566

Auguste Zurkinden
Abteilung fur die Sicherheit
der Kernanlagen
Bundesamt fur Energiewirtschaft
CH-5303 Wurenlingen
SWITZERLAND

Lai K. Chan
Department of Statistics
The University of Manitoba
Winnipeg, Manitoba
CANADA R3T 2N2

Hue McCoy
TRASANA
White Sands Missile Range
White Sands, NM 88002

Lissa Galbraith
Department of Industrial Engineering
and Operations Research
Virginia Polytechnic Institute
Blacksburg, VA 24061

Sandia National Laboratories Distribution

1642	D. E. Amos
2564	G. W. Smith
3141	C. M. Ostrander (5)
3151	W. L. Garner
6246	H. P. Stephens
6312	R. R. Peters
6400	A. W. Snyder
6410	J. W. Hickman
6411	A. S. Benjamin
6412	M. P. Bohn
6414	W. R. Cramond
6414	D. M. Erickson
6415	D. C. Aldrich
6415	D. J. Alpert
6415	D. E. Bennett
6415	R. P. Burke
6415	D. Chanin
6415	J. M. Griesmeyer
6415	J. C. Helton
6415	R. L. Iman (2)
6415	J. D. Johnson
6415	C. D. Leigh
6415	R. M. Ostmeyer
6415	L. T. Ritchie
6415	J. L. Sprung
6415	D. R. Strip
6415	A. R. Taig
6430	N. R. Ortiz
6431	M. S. Chu
6431	R. M. Cranwell
6432	L. D. Chapman
7200	J. M. Wiesen
7220	R. R. Prairie
7223	H. E. Anderson
7223	C. R. Clark
7223	K. V. Diegert
7223	R. G. Easterling
7223	E. L. Frost
7223	I. J. Hall
7223	B. P. Roudabush
7223	D. D. Sheldon
7223	M. J. Shortencarier
7223	F. W. Spencer
7223	D. A. Vopicka
8214	M. A. Pound

NRC FORM 335 (6-83) U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

1 REPORT NUMBER (Assigned by TIDC, add Vol. No., if any)
NUREG/CR-3624
SAND83-2365

2 Leave blank

3 TITLE AND SUBTITLE
A FORTRAN 77 PROGRAM AND USER'S GUIDE FOR THE GENERATION OF LATIN HYPERCUBE AND RANDOM SAMPLES FOR USE WITH COMPUTER MODELS

4 RECIPIENT'S ACCESSION NUMBER

5 DATE REPORT COMPLETED
MONTH: March YEAR: 1984

6 AUTHOR(S)
Ronald L. Iman and Michael J. Shortencarier

7 DATE REPORT ISSUED
MONTH YEAR

9 PROJECT/TASK/WORK UNIT NUMBER

8 PERFORMING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)
Sandia National Laboratories
Albuquerque, NM 87185

10 FIN NUMBER
A1339

11 SPONSORING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)
Division of Risk Analysis
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555

12a TYPE OF REPORT

12b PERIOD COVERED (Inclusive dates)

13 SUPPLEMENTARY NOTES

14 ABSTRACT (200 words or less)

This document has been designed for users of the computer program developed by the authors at Sandia National Laboratories for the generation of either Latin hypercube or random multivariate samples. The Latin hypercube technique employs a constrained sampling scheme, whereas random sampling corresponds to a simple Monte Carlo technique. The generaion of these samples is based on information supplied to the program by the user describing the variables or parameters used as input to the computer model. The actual sampled values are used to form vectors of variables commonly used as input to computer models for purposes of sensitivity and uncertainty analysis studies. The present program replaces the previous Latin hypercube sampling program developed at Sandia National Laboratories (Iman, Davenport, and Zeigler, 1980). The present version is written using FORTRAN 77 and greatly extends the program while making the program portable and user friendly.

15a KEY WORDS AND DOCUMENT ANALYSIS

15b DESCRIPTORS

16 AVAILABILITY STATEMENT
GPO Sales and NTIS

17 SECURITY CLASSIFICATION (This report)
Unclassified

18 NUMBER OF PAGES
67

19 SECURITY CLASSIFICATION (This page)
Unclassified

20 PRICE
$

☆U.S. GOVERNMENT PRINTING OFFICE: 1984—776-027 / 4301