SANDIA REPORT

SAND98–0210 • UC–505
Unlimited Release
Printed February 1998

A User's Guide to LHS: Sandia's Latin Hypercube Sampling Software

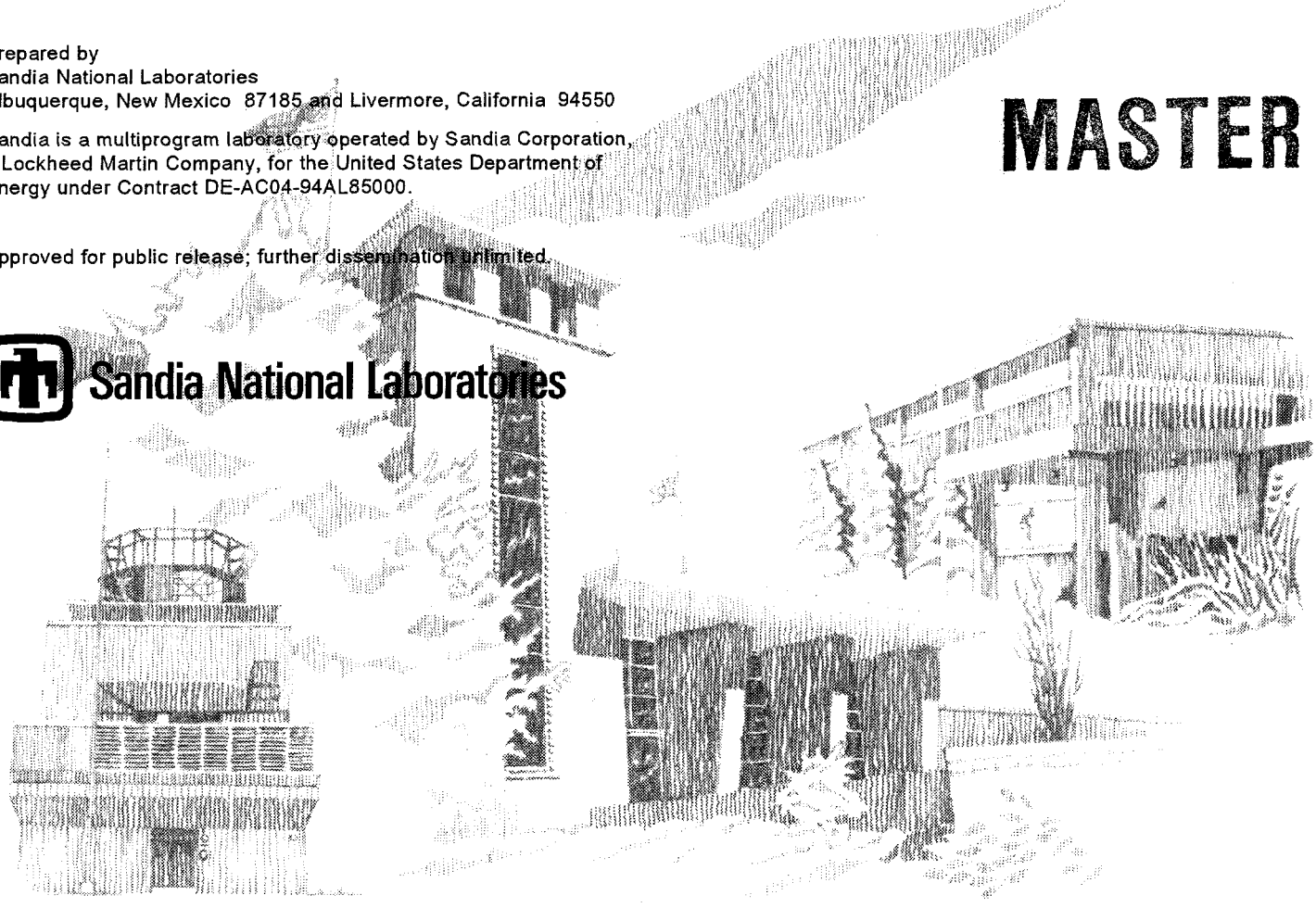
Gregory D. Wyss, Kelly H. Jorgensen

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories



Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A07
Microfiche copy: A01

DISCLAIMER

Portions of this document may be illegible electronic image products. Images are produced from the best available original document.

SAND98-0210
Unlimited Release
Printed February 1998

Distribution
Category UC-505

A User's Guide to LHS: Sandia's Latin Hypercube Sampling Software

Gregory D. Wyss and Kelly H. Jorgensen
Risk Assessment and Systems Modeling Department
Sandia National Laboratories
PO Box 5800
Albuquerque, NM 87185-0747

Abstract

This document is a reference guide for LHS, Sandia's Latin Hypercube Sampling Software. This software has been developed to generate 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 present program replaces the previous Latin hypercube sampling program developed at Sandia National Laboratories (SAND83-2365). This manual covers the theory behind stratified sampling as well as use of the LHS code both with the Windows graphical user interface and in the stand-alone mode.

Acknowledgments

This report documents a major upgrade to the original LHS package that was originally developed by Ronald L. Iman, Michael J. Shortencarier, J. M. Davenport, and D. K. Ziegler during the late 1970s and early 1980s. The authors are indebted to them for their pioneering work in the area of Latin hypercube sampling.

A number of people at Sandia National Laboratories have made significant contributions to this document. Many thanks go to Sharon Daniel, who generated many of the graphics used throughout this manual and served as a reviewer. Dennis Anderson generously donated his time to the referee of this manual. Dave Carlson and Kevin Maloney of Sandia's Assessment Technologies Department and Allen Camp of Sandia's Risk Assessment and Systems Modeling Department provided technical and managerial support for this project. The authors are grateful for the help received from all of these colleagues.

Contents

Figures

Tables

1 Introduction

For more than fifteen years, the Latin hypercube sampling (LHS) program has been successfully used to generate multivariate samples of statistical distributions. Its ability to use either Latin hypercube sampling or pure Monte Carlo sampling with both random and restricted pairing methods has made it an important part of uncertainty analyses in areas ranging from probabilistic risk assessment (PRA) to complex simulation modeling. LHS has received widespread use, but its utility has been severely limited because it supported relatively few distribution types and contained a Fortran code that was specific to Digital Equipment's VAX computer systems. As part of a major PRA code modernization effort at Sandia National Laboratories, the LHS software has been extensively revised, extended, and upgraded to meet current needs. The latest upgrade has been incorporated into ARRAMIS™, Sandia National Laboratories' risk and reliability analysis software suite. This report documents the capabilities and use of the revised LHS code.[1]

The revisions that have been incorporated into this version of the LHS software have been strongly driven by comments and requests from users of the original code. The enhanced features of the code include:

- Tailoring of the code for an IBM-compatible personal computer platform while maintaining a high degree of portability to other platforms

- Removal of virtually all machine-dependent code through the use of Fortran 90 intrinsic functions

- Addition of a truly portable random number generator that has been demonstrated to produce random sequences that are identical to several significant figures using many different compilers, operating systems, and computational platforms

- Dynamic array allocation that enables the analyst to modify program limits interactively without recompiling the software

- Addition of more than 25 new distributions, including several discrete distributions and five ways to enter and sample arbitrary distributions

- Addition of a new input file format that is compatible with other Sandia PRA codes within the ARRAMIS™ software suite while maintaining backward compatibility with the previous input format

- Using names to identify the random variables to be sampled and correlated – in both the code input and output (column numbers in the output file were used in the previous format)

[1] The original version of the LHS software was documented in SAND83-2365 (Iman and Shortencarier, 1984) and the revised code presented here constitutes a major revision.

- Development of an LHS postprocessor that allows the user to perform mathematical functions on sets of random variables, including multiplication, addition, exponentiation, minimum/ maximum operations, and zero/one sampling

- Development of the capability to sample the Dirichlet distribution (a multivariate relative of the beta distribution)

- Automatic inclusion of point-estimate values calculated from the sampled random variables in the output file (mean, median, or user-specified)

- Greater speed and robustness when calculating beta distributions

- Enhanced error checking and reporting

- Development of a Microsoft Windows-based user interface to assist the analyst with input preparation

- Development of a graphical output system that supports the plotting of distributions generated by LHS

- Support for both command line and Windows-interfaced executable versions of the software to ensure portability to non-PC platforms

- Support for Windows dynamic link library versions of the software that can be easily incorporated into other software packages that require uncertainty analysis capabilities

This manual is divided into six chapters with six appendices. The next chapter is an introduction that describes the concepts involved in the Latin hypercube sampling and restricted pairing methods employed by the software. Chapter 3 describes how the analyst interacts with the Windows user interface for the LHS software, while Chapter 4 contains detailed descriptions of the input format and the distributions supported by LHS. Chapter 5 describes how the CRAVE graphics package can be used to graph the distributions generated by LHS, while Chapter 6 is a summary of the report. Appendix A contains a detailed description of the LHS keyword file format (required by users of the non-PC version of LHS). Appendix B contains a detailed description of the various output files produced by LHS. A detailed sample problem, complete with input and output files, can be found in Appendix C. Details on backward compatibility with previous versions of LHS are included in Appendix D, while information for modifying the array sizes used within LHS can be found in Appendix E. Finally, Appendix F contains a "cheat sheet" summarizing the distributions used within LHS.

2 Latin Hypercube Sampling Theory

Latin hypercube sampling was developed to address the need for uncertainty assessment for a particular class of problems. Consider a variable Y that is a function of other variables $X_1, X_2, \ldots, X_k$. This function may be very complicated, for example, a computer model. A question to be investigated is "How does Y vary when the Xs vary according to some assumed joint probability distribution?" Related questions are "What is the expected value of Y?" and "What is the 99th percentile of Y?"

A conventional approach to these questions is to apply Monte Carlo sampling. By sampling repeatedly from the assumed joint probability density function of the Xs and evaluating Y for each sample, the distribution of Y, along with its mean and other characteristics, can be estimated. The LHS software supports this approach for generating samples of the Xs. The program output, for n Monte Carlo repetitions, is a set of n vectors of input variables (each such vector is k-dimensional). Each input vector can then be evaluated by the function or program to generate n values of the result Y (Y may be a scalar or a vector whose dimensionality is determined by the function or program of interest). This approach yields reasonable estimates for the distribution of Y if the value of n is quite large. However, since a large value of n requires a large number of computations from the function or program of interest, which is potentially a very large computational expense, additional methods of uncertainty estimation were sought.

2.1 Sampling

An alternative approach, which can yield more precise estimates, is to use a constrained Monte Carlo sampling scheme. One such scheme, developed by McKay, Conover, and Beckman (1979), is Latin hypercube sampling. Latin hypercube sampling selects n different values from each of k variables $X_1, \ldots X_k$ 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. These n k-tuplets are the same as the n k-dimensional input vectors described in the previous paragraph. This is the Latin hypercube sample. It is convenient to think of this sample (or any random sample of size n) as forming an $(n \times k)$ matrix of input where the ith row contains specific values of each of the k input variables to be used on the i^{th} run of the computer model.

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

To help clarify how intervals are determined in the Latin hypercube sample, consider a simple example where it is desired to generate a Latin hypercube sample of size $n = 5$ with two input variables. Let us assume that the first random variable X_1 has a normal distribution with a mean value of μ and a variance of σ^2. The endpoints of the intervals are easily determined based on the parameters μ and σ^2. The intervals for $n = 5$ are illustrated in Figure 2-1 in terms of both the density function and the more easily used cumulative distribution function (CDF). The intervals in Figure 2-1 satisfy

$$P(-\infty \leq X_1 \leq A) = P(A \leq X_1 \leq B) = P(B \leq X_1 \leq C)$$

$$= P(C \leq X_1 \leq D) = P(D \leq X_1 \leq \infty) = 0.2$$

Thus each of the five intervals corresponds to a 20% probability.

We will assume in this example that the second random variable, X_2, has a uniform distribution on the interval from G to L. The corresponding intervals used in the Latin hypercube sample for X_2 are given in Figure 2-2 in terms of both the density function and the CDF.

The next step in obtaining the Latin hypercube sample is to pick specific values of X_1 and X_2 in each of their five respective intervals. This selection must be done in a random manner with respect to the density in each interval; that is, the selection must reflect the height of the density across the interval. For example, in the $(-\infty, A)$ interval for X_1, values close to A will have a higher probability of selection than those values in the tail of the distribution that extends to $-\infty$. Next, the selected values of X_1 and X_2 are randomly paired to form the five required two-dimensional input vectors. In the original concept of Latin hypercube sampling as outlined in McKay, Conover, and Beckman (1979), the pairing was done by associating a random permutation of the first n integers with each input variable. For 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:

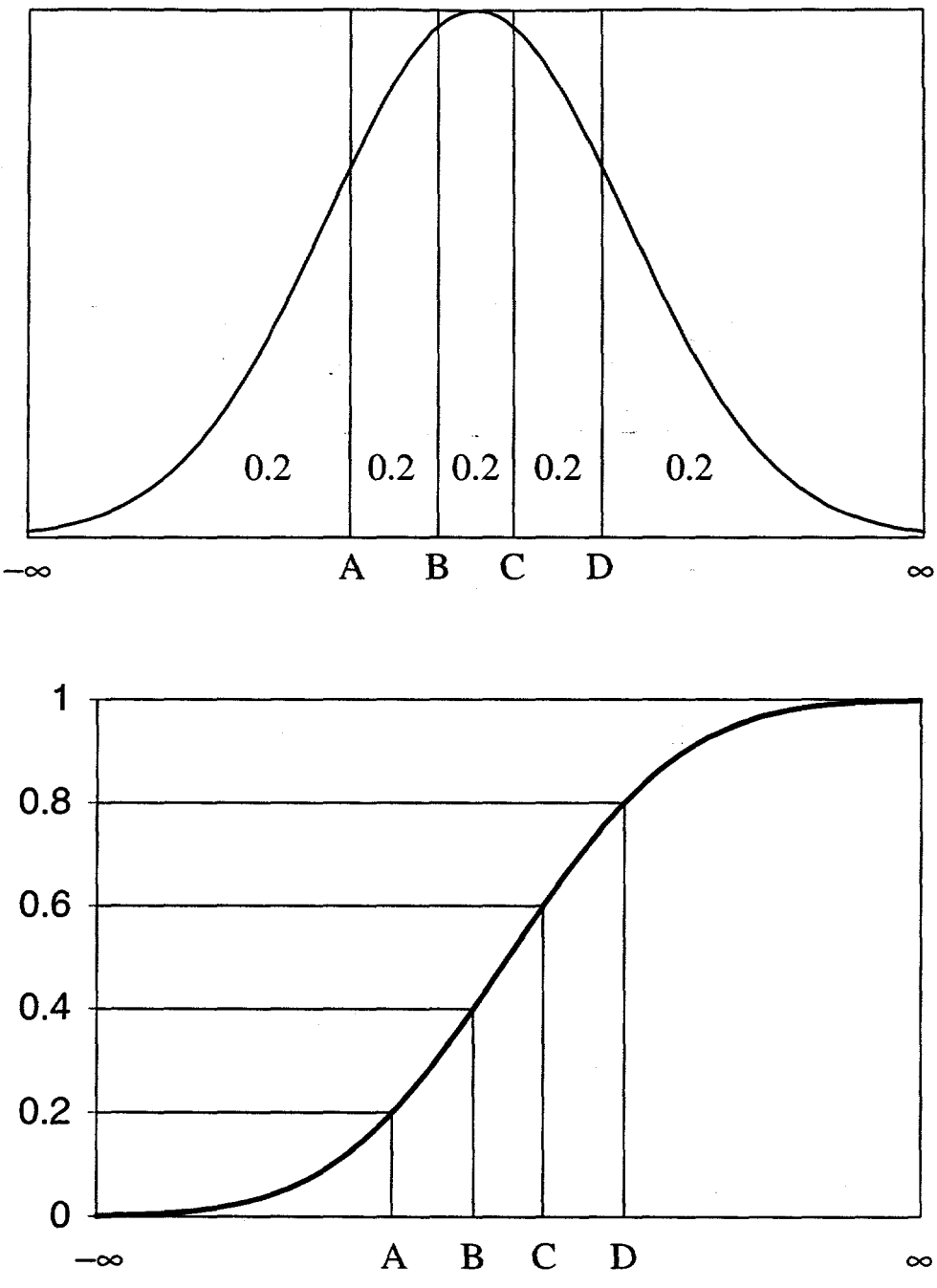


Figure 2-1: Intervals Used with a Latin Hypercube Sample of Size n = 5 in Terms of the Density Function and Cumulative Distribution Function for a Normal Random Variable

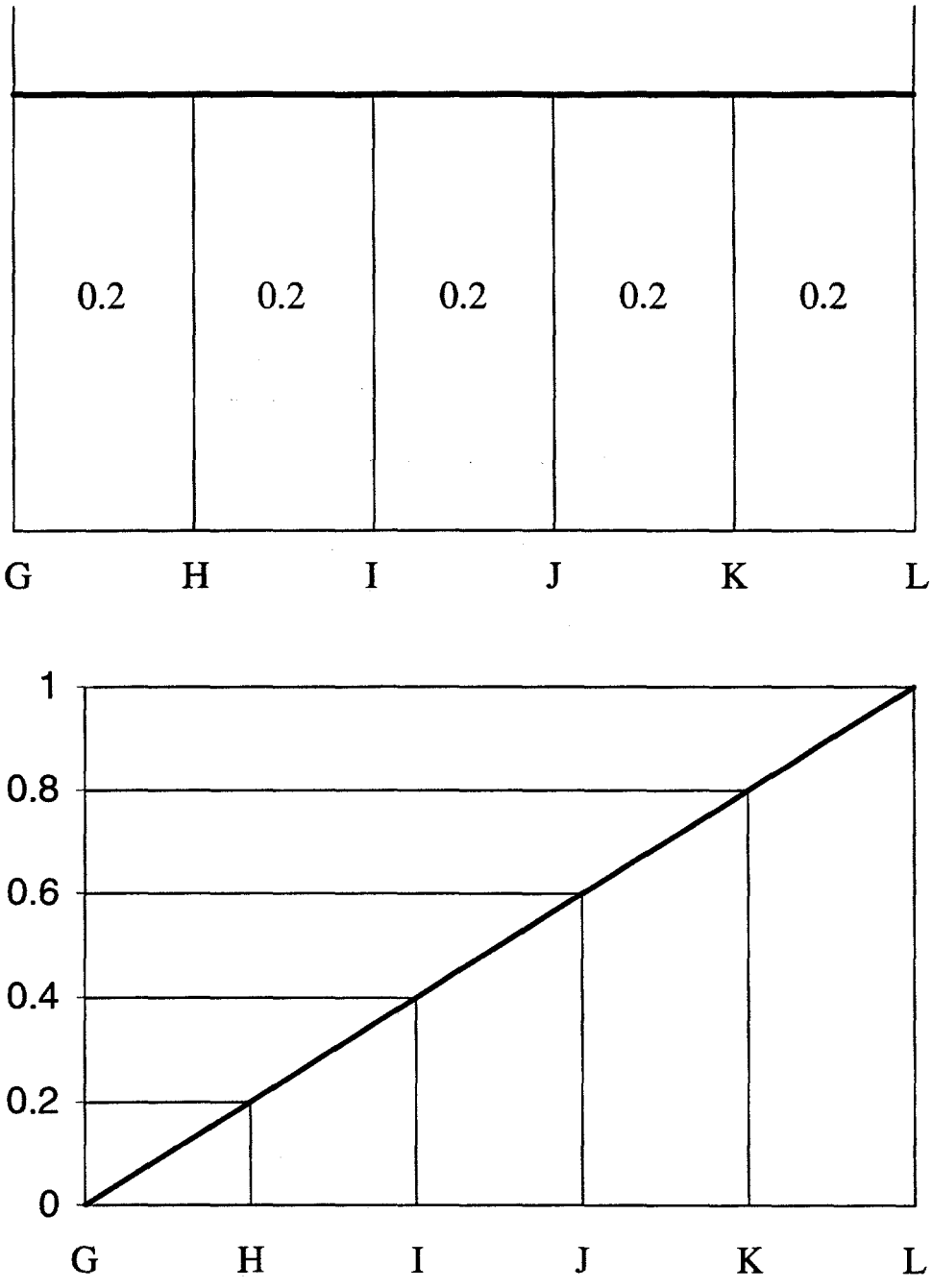


Figure 2-2: Intervals Used with a Latin Hypercube Sample of Size n = 5 in Terms of the Density Function and Cumulative Distribution Function for a Uniform Random Variable

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 the interval number 3 (B to C) and pairing this value with the specific value of X_2 selected from interval number 2 (H to I). The vectors for the second and subsequent runs are constructed in a similar manner.

Once the specific values of each variable are obtained to form the five input vectors, a two-dimensional representation of the Latin hypercube sample such as that given in Figure 2-3 can be made. Note in Figure 2-3 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 Latin hypercube sample points in k-dimensional Euclidean space contains one point in each of the intervals for each of the k variables.

To illustrate how the specific values of a variable are obtained in a Latin hypercube sample, consider the following example. Suppose it is desired to obtain a Latin hypercube sample of size n = 5 from a normal distribution with a mean of 5.0 and a variance of 2.618 as indicated in Figure 2-4. The density characteristics of the normal distribution allow for the definition of the equal probability intervals. These intervals are shown in Figure 2-4 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 2-4, but rather it is done relative to the probability density function distribution being sampled (in this case, the normal distribution). This is equivalent to uniformly sampling from the *quantiles* of the distribution (equivalent to sampling the vertical axis of the CDF) and then "inverting" the CDF to obtain the actual distribution values that those quantiles represent. This process is illustrated in Figure 2-4.

Therefore to get the specific values, n = 5 numbers are randomly selected from the standard uniform distribution (uniformly distributed between 0 and 1). Let these be denoted as U_m, where m = 1, 2, 3, 4, 5. These values will be used to select distribution values randomly from within each of the n = 5 intervals. To accomplish this, each of the random numbers U_m is scaled to obtain a corresponding cumulative probability, P_m, so that each P_m lies within the m^{th} interval. Thus, for this example with n = 5,

$$P_m = \left(\frac{1}{5}\right)U_m + \left(\frac{m-1}{5}\right)$$

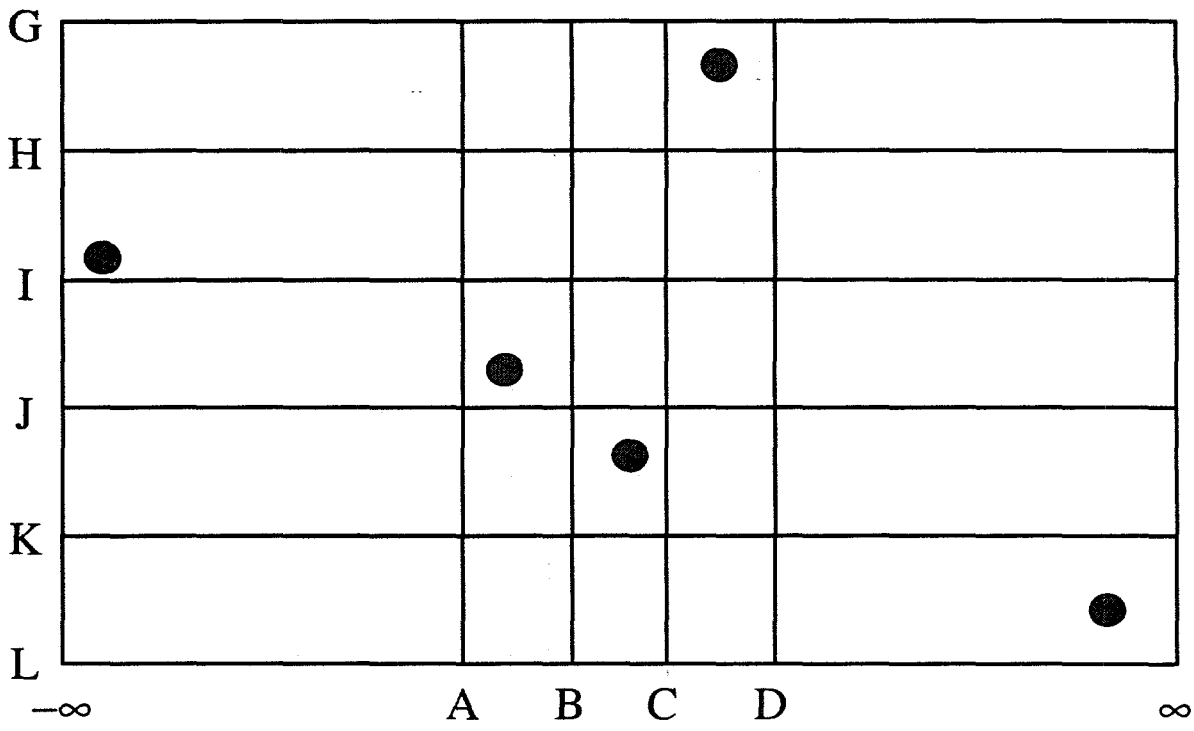


Figure 2-3: A Two-Dimensional Representation of One Possible Latin Hypercube Sample of Size 5 Utilizing X_1 and X_2

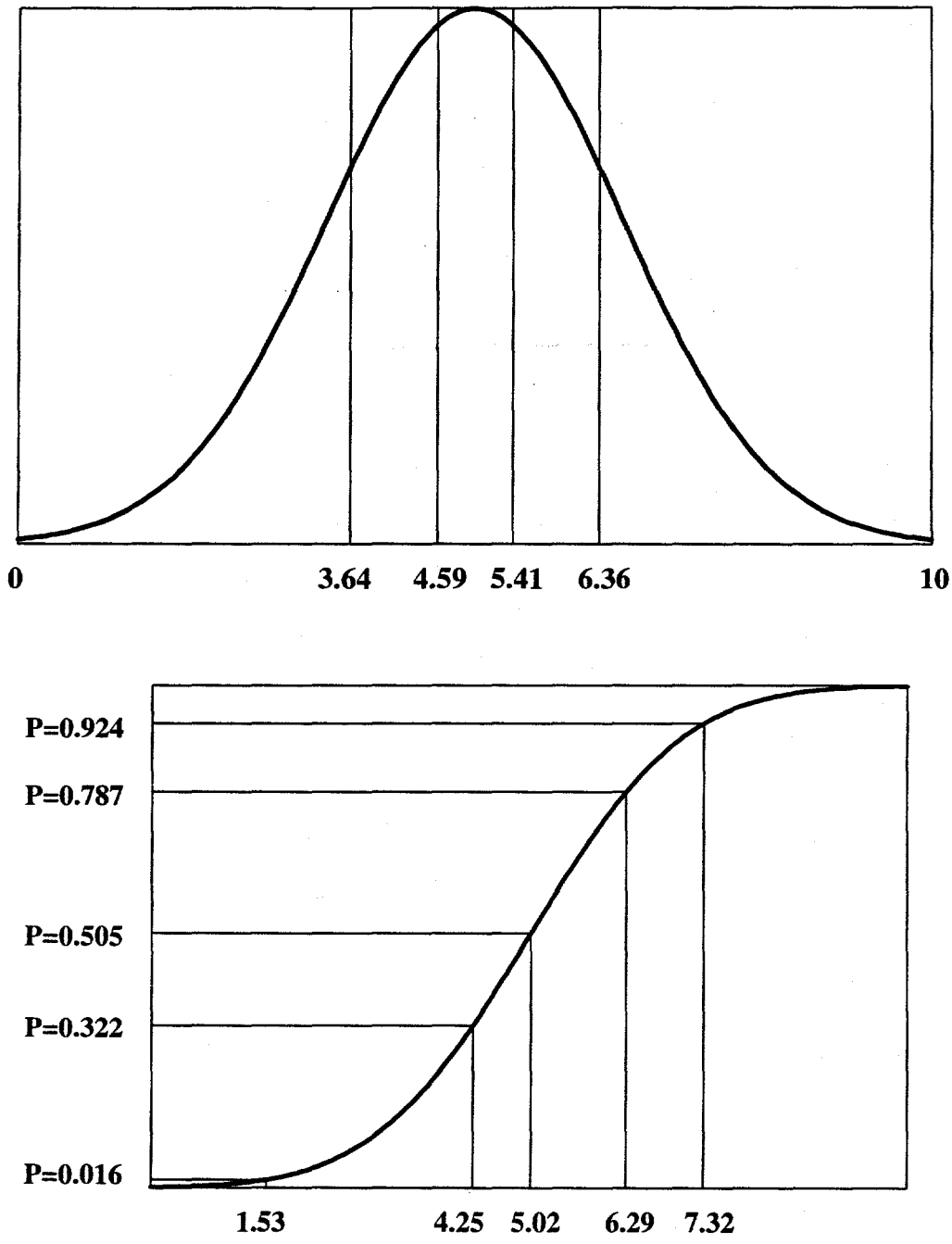


Figure 2-4: Interval Endpoints Used with a Latin hypercube sample of Size 5 (top) and Specific Values of X Selected Through the Inverse of the Distribution Function (bottom)

This ensures that exactly one probability, P_m, will fall within each of the five intervals (0, 0.2), (0.2, 0.4), (0.4, 0.6), (0.6, 0.8) and (0.8, 1). The values P_m are used with the inverse normal distribution function to produce the specific values to be used in the final Latin hypercube sample. Note that exactly one observation is taken from each interval shown in Figure 2-4. The entire process is shown in Table 2-1. Figure 2-4 makes it clear that when obtaining a Latin hypercube sample, 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.

Figure 2-4 shows how one input variable having a normal distribution is sampled with Latin hypercube sampling. 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 directly (i.e., without scaling) mapped through the inverse distribution function to obtain the specific values. The final step in the sampling process involves pairing the selected values.

Table 2-1: One Possible Selection of Values for a Latin Hypercube Sample of Size 5 from a Normal Distribution with a Mean of 5 and a Standard Deviation of 2.618

Interval Number m	Uniform (0,1) Random No. U_m	Scaled Probabilities Within the Interval $P_m = U_m(0.2) + (m-1)(0.2)$	Corresponding Standard Normal Value (Z-score) from the Inverse Distribution	Corresponding N(5, 2.618) Observation Within the Intervals
1	0.080	0.016	-2.144	1.529
2	0.610	0.322	-0.462	4.252
3	0.525	0.505	0.013	5.021
4	0.935	0.787	0.796	6.288
5	0.620	0.924	1.433	7.319

2.2 Pairing

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 Latin hypercube sample will, in general, not equal zero 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 believed to be small. The LHS software supports both the random pairing of variables described in the previous section and the restricted pairing of variables that is explained in this section.

While a full description of the restricted pairing methodology is beyond the scope of this report, a heuristic description is appropriate in order to give the prospective LHS software user a greater level of comfort with the software. Recall that the basic method for LHS is to select a set of n observations (random samples) for each random variable, and then to randomly "pair" those selected observations. This yields a set of n vectors, where each vector contains one value for each random variable. Thus, if there are k random variables, each of the n vectors is k-dimensional.

In the process of sampling a single distribution (generating n observations of values for that distribution), there is no significance to the order in which those observations are obtained. What matters for the purposes of the uncertainty analysis is that the entire *collection* of observations faithfully preserve the properties of the distribution from which it was sampled.

We can now view the pairing process as follows. Imagine writing the value of each observation for distribution X_1 on a separate slip of paper and placing those slips into a hat. There would now be n separate slips of paper in that hat. Now imagine following the same process for distribution X_2, and placing the slips in a second hat. The same process could be followed for distributions X_3 through X_k, so that at the end of the process one would have k hats, each containing n slips of paper with distribution values written on them.

To accomplish the random pairing process described previously, one would simply draw one slip of paper at random from each of the k hats, and the group of values written on those k slips of paper would form the first observation of the output data set. In the language of the previous section, these values would form the k-dimensional input vector for the first computer run, or the first row in the $(n \times k)$ matrix of sampling results. The second set of values drawn randomly from the k hats would form the second observation, k-dimensional input vector, or matrix row, and so forth until the last, or n^{th}, set of values was drawn to form the last such observation. Since exactly n values were generated for each distribution, all of the hats are now empty. Thus, all of the generated distribution values were used, so each column in the $(n \times k)$ matrix faithfully represents the distribution from which its values were drawn.

One could, however, visualize a different pairing process in which the slips of paper from the k hats were poured out to make k separate piles. The slips of paper from those piles could then be arranged in columns to form the $(n \times k)$ matrix described previously. Since there is no significance to the *order* in which the individual slips of paper are arranged in these columns, one might imagine a person examining the entire $(n \times k)$ matrix and *deliberately* moving slips of paper around within columns to achieve some goal for the overall matrix. If one were clever, one could order each column so that the correlation between its values and those of every other column in the matrix is as small as possible. When this process is completed, the $(n \times k)$ matrix still contains the same values in the same columns, but each column has been intentionally reordered. Each column in the matrix faithfully represents the distribution from which its values were drawn because, once again, all of the values generated for the distribution were used. Also, each row still represents the k-dimensional input vector for a computer run because it contains one value from each of the distributions X_1 through X_k. Thus the $(n \times k)$ matrix is in every sense equivalent to that generated by the random pairing process described earlier. In fact, there is some small probability that the random pairing algorithm would generate this matrix if exactly the right sequence of random draws

were to occur during the process. This intentional selection of the pairing of the selected observations is directly analogous to the restricted pairing procedure developed by Iman and Conover and implemented within the LHS software.

It is logical to ask then whether this restricted pairing technique can be used to induce specified *nonzero* correlations between random variables since it is already being used to cause these same correlations to tend toward zero (or, more properly, become as small as possible). The answer is yes, with the limitation that such induced correlations are based on the nonparametric technique known as rank correlation. Such a measure is used since it remains meaningful in the presence of non-normal distributions on the input variables. The LHS software provides the user with input parameters by which these pairwise correlations can be specified.

While correlation is defined and measured in terms of pairs of random variables, it is possible to consider multivariate correlations. One might, for example, have a group of five random variables that are all strongly correlated with each other. This situation can be specified to the LHS software by defining correlations between every possible combination (pair) of random variables within that five-variable group. For a group of five random variables, this would require 10 correlation specification statements (all the unique combinations of five random variables taken two at a time).

There are mathematical limitations to the ways that random variables can be correlated with one another. For example, the following three-way correlation is statistically impossible:

Correlation (A, B) = –0.95
Correlation (A, C) = –0.95
Correlation (B, C) = –0.95

The first two statements imply that large values of A tend to be paired with small values of both B and C, while the last statement implies that small values of B tend to be paired with large values of C. This condition cannot be realized by any real sampling scheme. An impossible correlation could also be inadvertently specified by a user if he or she were to attempt to define the five-variable correlation described above, but one pairwise combination of variables was omitted from the program input. This would cause the LHS software to attempt to correlate all five variables with each other and yet maintain one pair of variables within that group as statistically independent. This is clearly a statistical impossibility.

When the LHS software encounters such a condition, it generates a warning for the user and makes the smallest adjustments possible to the requested correlations so that it can generate its results. However, such conditions generally indicate that the user has either made a mistake in specifying input to the software or does not thoroughly understand the correlation conditions that he or she is attempting to model since contradictory correlation information has been entered. When the software detects these conditions, it is important to step back and reevaluate the software input to ensure that these contradictions are eliminated.

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 readable explanation on reasonable sizes of VIFs. Marquardt (1970) indicates that there can be serious colinearity (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 without specifying a correlation structure.

This page intentionally left blank

3 The LHS Interface

This chapter explains how the LHS interface is used to specify input for the Latin hypercube sampling engine. There are other ways to run the Latin hypercube sampling engine without using this interface. Information on preparing input to be read directly by LHS can be found in Chapter 4 and Appendix F.

3.1 What the LHS Interface Does

The LHS interface is designed to provide a friendly facility for stratified data sampling. It takes as its input ASCII-formatted files that contain sampling options and distributions specified for basic events in fault trees and/or event trees and processes those files to obtain sampled data. At the user's option, these data can be plotted with the CRAVE (CONTAIN and Risk Assessment Visualization Environment) package, entered into the SETAC code (Sandia Event Tree Analysis Code) for event frequencies, pulled into the SABLE package (Sandia Automated Boolean Logic Evaluation) (Hays et al. 1996) for basic event failure probabilities, or used by TEMAC3 software (Top Event Matrix Analysis Code, Version 3) for uncertainty analysis.[1]

LHS consists of two major parts: the Windows user interface and the Fortran dynamic link library (DLL). The interface is designed to provide a friendly environment for the user to specify sampling options; import basic events from fault trees, existing value blocks, cut sets, and event trees; specify distributions for these variables; and postprocess the distributions. The interface writes files that contain the sampling options and distributions specified by the user. The DLL uses these files to perform the stratified sampling operations specified by the user through the interface.

3.2 Main Screen

The Main screen, shown in Figure 3-1, is the first screen that appears when the LHS interface is invoked. This screen provides an overview of the currently selected options, which include the sampling options and distributions specified. Options cannot be changed on this screen. The Modify buttons must be pressed to bring up the detailed screens. This screen can be used to browse the summary information to determine whether the currently selected options will cause the LHS interface to sample all distributions desired. When you are satisfied with the options that you have selected, the Sample Data button is pressed to run LHS. The results can then be viewed by pressing the Plot button, which will invoke CRAVE. After sampling is completed, the Postprocess button can be pressed to mathematically manipulate the distributions to more accurately model the desired system.

[1] User documentation for CRAVE can be found in Chapter 5 of this report. User documentation for SETAC and TEMAC3 will be published at a later date.

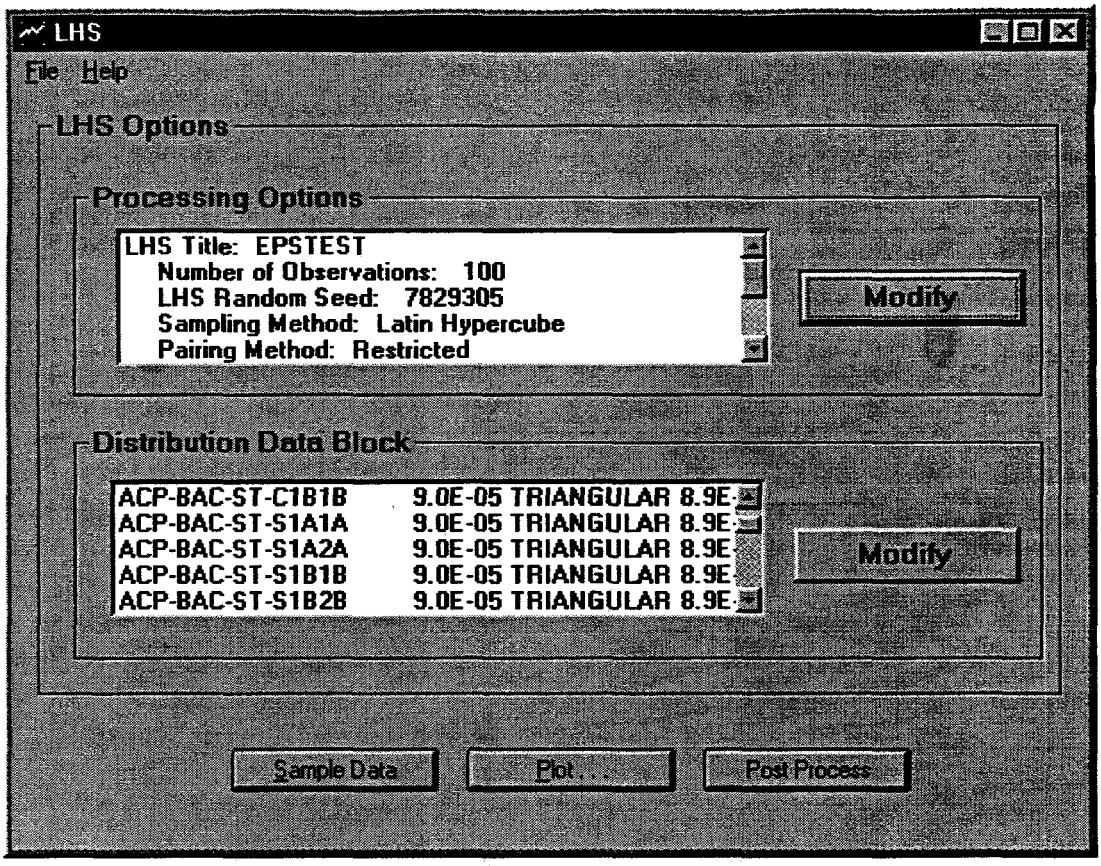


Figure 3-1: The Main Screen

3.2.1 The Sample Data Button

The Sample Data button is located at the bottom of the Main screen. When this button is pressed, the LHS interface performs the following operations:

- LHS checks the sampling options and the distribution names and parameters specified by the user to ensure that they are complete and consistent
- LHS writes the ASCII text files that contain instructions for the DLL
- LHS calls the DLL and waits in the background for it to finish (the DLL will print messages to the screen in a separate DOS-like window)
- LHS checks for run-time errors in the DLL

If a run-time error occurs, a diagnostic message is displayed. The LHS interface will display where the results and/or error files can be found for viewing and printing.

3.2.2 The Plot Button

The Plot button will invoke CRAVE, the probabilistic risk assessment graphing package. Instructions for using CRAVE are contained in Chapter 5. CRAVE can plot any or all of the sampled distributions as a cumulative distribution function, probability density function (PDF), and/or complementary cumulative distribution function (CCDF).

3.2.3 The Postprocess Button

The Postprocess button allows the user to mathematically manipulate the sampled distributions. The Postprocessing option provides a simple equation editor for building mathematical representations to more accurately model the desired physical system. The Postprocessing section of the LHS interface is discussed in detail in Section 3.5.

3.2.4 File Handling

The following options are available through the File menu on the Main screen.

Starting a New Session. When you first enter LHS, you are ready to start a new data sampling session. If you have already selected options and wish to start over with a new session, select File | New from the menu on the Main screen. A verification screen will appear; you may either continue with the existing session or start a new one.

Opening a Preexisting Session. To open a preexisting data sampling session, select File | Open from the menu on the Main screen. A standard Windows dialog will allow you to browse through drives, directories, and files until you select the file you wish to open. Files containing a LHS analysis usually have a *.LHI extension.

Saving a Session. To save an existing session, select File | Save from the menu on the Main screen. If you have previously named the session, the file will be saved automatically. If you have not previously named the session, a standard Windows dialog will allow you to specify the name of the session and its path.

If you wish to save a session under a different name, select File | Save As from the menu on the Main Screen. A standard Windows dialog will allow you to specify the name of the session and its path.

Importing an Event List. Many times there are existing files that contain the events you wish to sample. The LHS interface allows you to glean the primary event information from standard SABLE fault tree files (file extension *.SET), standard SABLE value block files (*.DAT), cut set files (*.SCO), and event tree definition files (*.DTI). To import this information, select File | Import from the menu on the Main screen. A standard Windows dialog will allow you to specify the name of the file to import and its path. You may select multiple files to import. If you've already started a data sampling session or you've already imported a list, you will be given the

option to either start a new session with the imported list, add the imported list to the current session, or cancel the operation.

3.3 Sampling Options

This section discusses how to select various sampling options. To modify sampling options, go to the area of the Main screen marked Processing Options and press the Modify button. Figure 3-2 shows the LHS Processing Options screen.

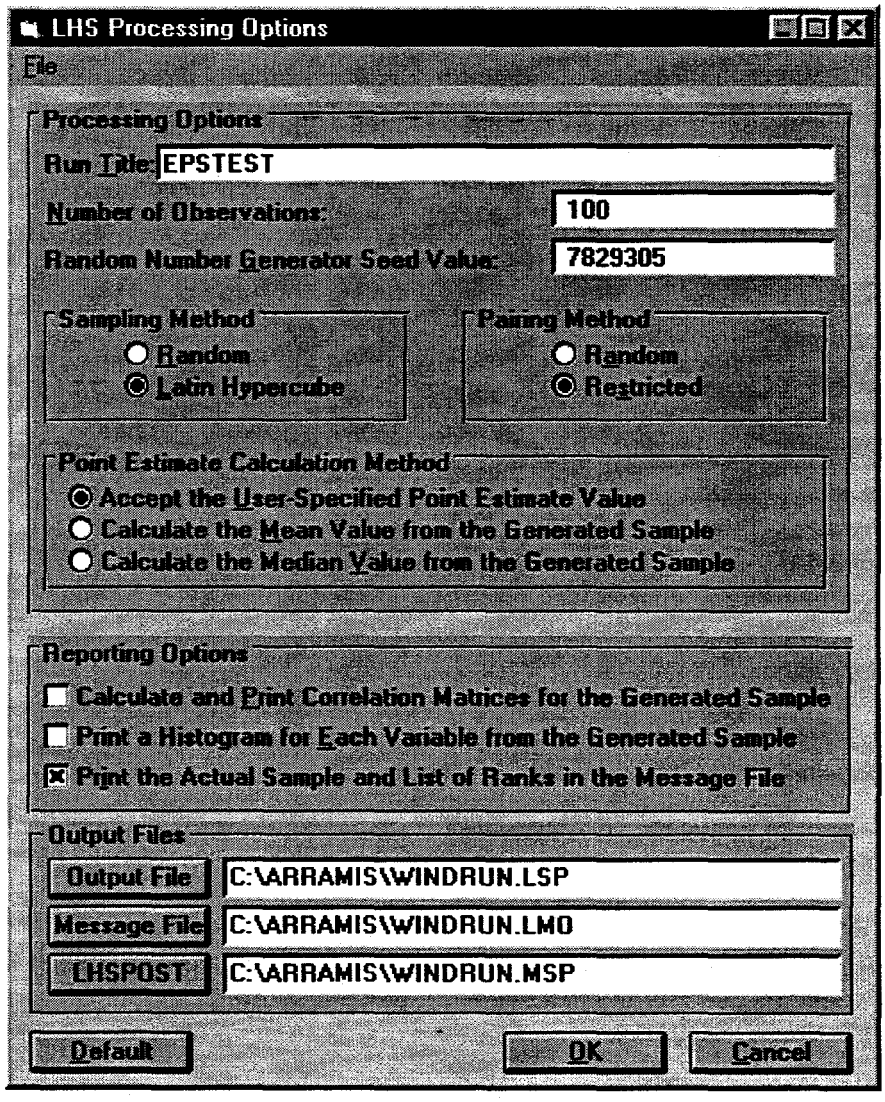


Figure 3-2: The LHS Processing Options Screen

3.3.1 Run Title

The Run Title is an alpha-numeric string of up to 70 characters. This string is available for uniquely identifying the input and output associated with this LHS run.

3.3.2 Number of Observations

The number of observations is the actual number of samples generated for each distribution specified. This number cannot exceed 1000 unless the program parameters are changed (see Appendix E).

3.3.3 Random Number Generator Seed Value

The LHS DLL contains its own random number generator to ensure portability and reproducibility of results. You must enter a positive integer seed value for the random number generator. This seed must be less than 2^{31}. The value of this seed should be greater than 1000 because certain smaller values can generate random sequences that compromise statistical randomness properties.

3.3.4 Sampling Method

The LHS interface allows two sampling modes; Latin hypercube and random. Random sampling will cause LHS to perform pure Monte Carlo sampling. Latin hypercube sampling will cause LHS to perform stratified sampling and generally allows for a smaller sampling set (smaller number of observations).

3.3.5 Pairing Method

LHS allows two pairing modes; random and restricted. With random pairing, all distributions are randomly paired without considering any correlations that may be induced during the pairing process. Any pairwise correlations that you specify (see Section 3.6) will be ignored. Restricted pairing will attempt a correlation coefficient of zero for all distributions unless you request induced correlations.

Restricted pairing cannot be accomplished if you have more distributions than observations. Therefore, if you select Restricted Pairing and more distributions than observations, the LHS sampling engine will revert to a modified form of Random Pairing. In this method, the LHS engine randomly pairs the distributions 25 times and then selects from these 25 candidate pairings the one that produces the smallest maximum pairwise correlation. Note that this is different from the process that is followed when you select Random Pairing in the LHS input. The modified random pairing algorithm requires more computational effort than either restricted pairing or traditional random pairing. In addition, since it selects the candidate pairing that produces the smallest maximum pairwise correlation, it ignores all pairwise correlations from the input.

3.3.6 Point Estimate Calculation Method

The LHS sampled data output file contains a point estimate value for each sampled distribution; there are three types. You may select the sample mean, the sample median, or a user-supplied point estimate. If you are importing the LHS distribution information and sampled data into TEMAC3, you must provide a point estimate value and select Accept the User-Specified Point Estimate Value.

3.3.7 Reporting Options

There are three available reporting options for the message file with the LHS interface. Selecting Calculate and Print Correlation Matrices for the Generated Sample will print the raw and rank correlation matrices associated with the actual sample generated. Selecting Print a Histogram for Each Variable from the Generated Sample will print one text histogram for each random variable. Selecting Print the Actual Sample and List of Ranks in the Message File will include the actual data samples in the message file. This may generate a ***large*** message file. Any combination of the above three options may be used.

3.3.8 Output Files

You may specify the names of the three output files; the Output file which contains the data samples (used for input to other programs); the Message file, which contains run information for the user; and the LHSPOST file, which will contain the sampled data from the postprocessing run (see Section 3.6). By default, each file will be named WINDRUN, with the appropriate file extension.

3.4 Specifying Distribution Data

This section describes how to specify distribution data and induce pairwise correlations. To specify distributions, go to the area of the Main screen marked Distribution Data Block and press the Modify button. This will invoke the LHS Distribution screen, shown in Figure 3-3. This screen contains two columns. The column on the left contains the names of the basic events and their specified distributions. LHS will sample every distribution specified in this column. The column on the right contains the names of all imported events (see Section 3.2.2 for instructions on importing data). The buttons between the columns allow you to specify distributions, remove distributions, or edit distributions.

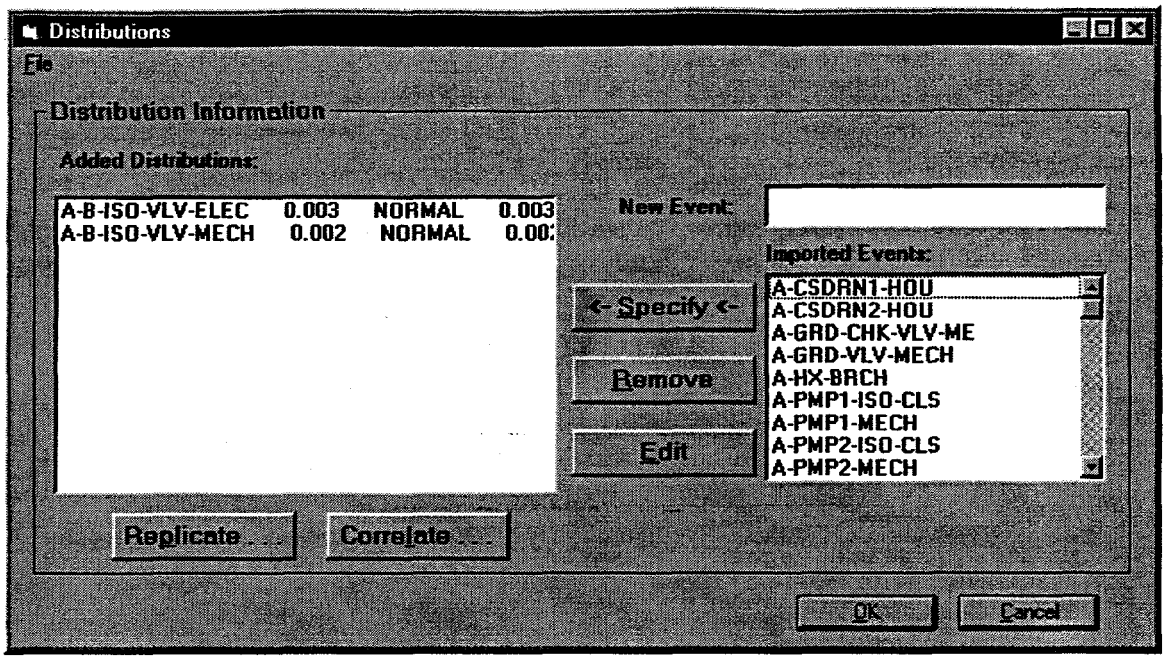


Figure 3-3: The LHS Distribution Screen

3.4.1 Specifying a Distribution

To specify a distribution, you may (1) select a variable from the Imported Events list and press the Specify button, (2) double click on a variable from the Imported Events list, (3) type the name of a new variable in the New Event text box and press the Specify button, or (4) type the name of a new variable in the New Event text box and press Return. Each of these actions will bring up the Parameters screen, shown in Figure 3-4. Once in this screen, you may edit the name of the variable, select a distribution from the drop-down selection list, and input the appropriate parameters. The parameters change dynamically based on which distribution you have selected. For instance, if you select a Normal distribution, you must enter a mean and standard deviation. If you select an Exponential distribution, you only need to input the time constant. The LHS interface will perform error checking on your parameters and notify you of any problems. Pressing OK will save the distribution information, add it to your specified distributions listed in the left column on the Distributions screen, and return you to the Distributions screen.

3.4.2 Removing a Distribution

To remove a distribution, either double click the name in the column on the left or select the name and press the Delete button. You will be asked for confirmation. If you remove a distribution, the corresponding event will be added back to the Imported Events list on the right.

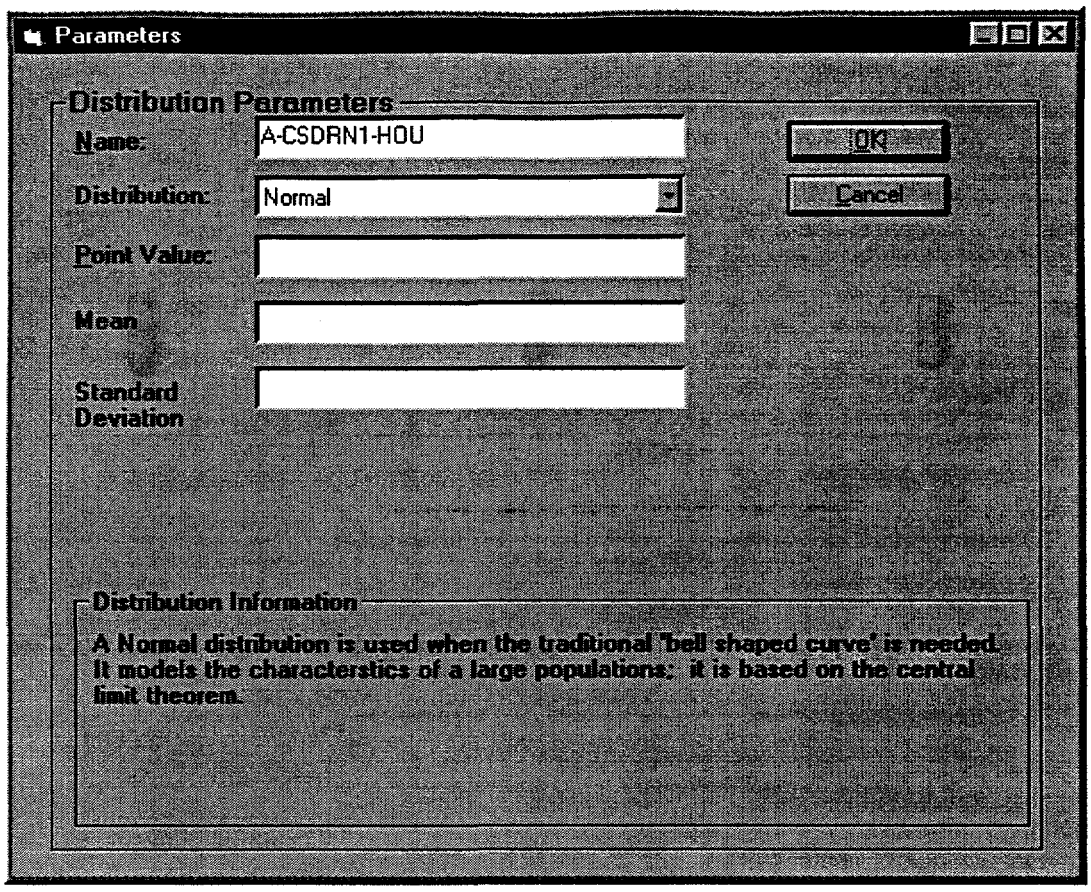


Figure 3-4: The Parameters Screen

3.4.3 Editing a Distribution

To edit a distribution that has already been specified, select the name in the column on the left and press the Edit button. The Parameters screen (Figure 3-4) will appear and you can change the distribution type and/or the associated parameters.

3.4.4 Replicating a Distribution

Since many components are similar, you may want to sample the same distribution for these components, or you may want to use identical samples for all of these components (total correlation). Both of these processes are accessed using the replicate distribution function. To replicate a distribution, select a distribution from the column on the left and press the Replicate button. The Replicate screen, shown in Figure 3-5, will appear.

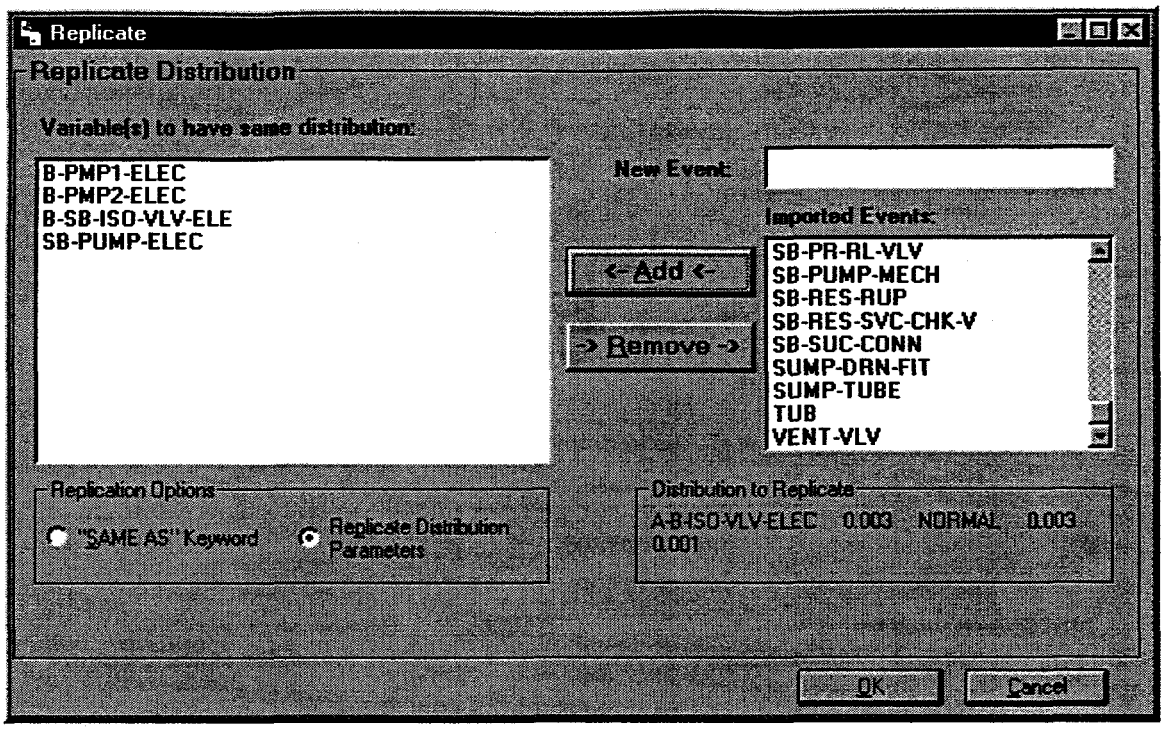


Figure 3-5: The Replicate Screen

This screen allows you to either copy the distribution and parameter list to other variables or use the SAME AS keyword to induce total correlation between the specified random variables.

The Replicate screen contains two columns. The column on the left contains the names of the variables that you have selected for replication. The column on the right contains the names of all remaining imported events. The buttons between the columns allow you to either add the variables to the column on the left or remove them from the column. A copy of the distribution that you are replicating is listed under the right column for reference. Under the column on the left, you may select where you wish to use the SAME AS keyword or simply copy the parameters of the selected distribution. For example, in Figure 3-5, the variables B-PMP1-ELEC, B-PMP2-ELEC, B-SB-ISO-VLV-ELE, and SB-PUMP-ELEC will each have a Normal distribution with a point value of 0.003, a mean of 0.003, and a standard deviation of 0.001. These parameters were copied from the existing variable, A-B-ISO-VLV-ELEC. Each of these distributions will be sampled individually.

Adding a Variable to the Replicating List. To add a variable to the replicating list, you may (1) select a variable from the Imported Events list and press the Add button, (2) double click on a variable from the Imported Events list, (3) type the name of a new variable in the New Event text box and press the Add button, or (4) type the name of a new variable in the New Event text box and press enter. Any of these actions will place the variable name in the column in the left. You may also choose multiple variable names by selecting them with your mouse and holding down the Control (Ctrl) key, or choose a range of names by selecting them while holding down the Shift key.

Removing a Variable from the Replicate List. To remove a variable from the replicating list, you may either double click the name in the column on the left or select the name and press the Remove button.

3.4.5 Correlating Distributions

While statistical distributions are often independent of one another, this is not always true. Often one distribution will demonstrate a significant tendency to vary in parallel with or in opposition to one or more other distributions. This property, known as correlation, must be modeled as the individual distributions are sampled if the sampling results are to be a faithful model of the actual conditions that are being modeled.

LHS is able to induce correlations between random variables if the Restricted Pairing option is selected. LHS assumes that all variables are independent (i.e., the correlation between it and all other random variables is zero) unless it is told otherwise through user input. The restricted pairing algorithm causes LHS to pair random variables in such a way as to match this or any other specified correlation conditions as closely as possible (a dependency between events).

The concept of correlation is defined only between *pairs* of random variables. LHS enables the user to specify the correlation between such pairs as follows: First, the user selects one random variable that is to be part of the desired correlation. The user can then select, one at a time, other random variables that are to be correlated with the selected random variable, and specify the degree of correlation that is to be induced. In the LHS interface, the first random variable is selected from the main screen in the same manner that is used to identify the distribution to be replicated (in a previous section). By pressing the Correlate button with an event name highlighted, the user is telling LHS that he or she wishes to specify one or more other random variables that are to be correlated with this random variable. This action will invoke the Correlate Distributions screen, shown in Figure 3-6.

This screen allows you to correlate the selected distribution with any of the other already-specified distributions. There are two columns on this screen. The column on the left contains all distributions that have been specified; they are eligible for correlation with the selected distribution. The column on the right lists the correlations that have already been defined. In this column you may select whether you want to display all correlations for this sampling session or just the correlations associated with the selected distribution.

To specify a variable to be correlated with the selected variable, select the desired variable from the column on the left by clicking and highlighting its name. Then enter a correlation coefficient in the edit bar above the column on the right. To add this correlation to the list, either hit Enter or press the Checkmark button. The correlation coefficient must be between −1.0 and 1.0. If you wish to edit a coefficient, select the desired correlation from the column on the right and the coefficient will appear in the edit box. You may change its value and, again, either hit Enter or press the Checkmark button to complete the correlation change.

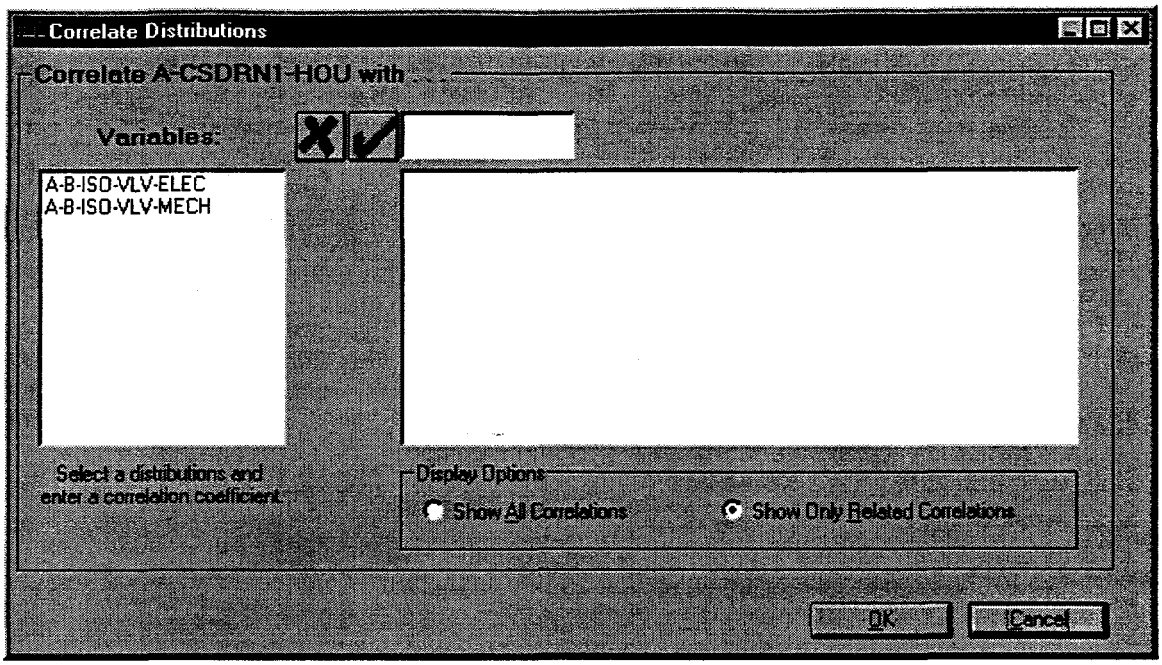


Figure 3-6: The Correlate Distributions Screen

3.5 Postprocessing Options

Once you have sampled your data (see Section 3.2.1), you may want to combine your distributions mathematically to more accurately represent physical situations. This is commonly done in event tree analysis. A simple equation editor has been provided in the postprocessing section of the LHS interface.

To postprocess your sampled data, press the Postprocess button on the Main screen. This will invoke the Postprocessing screen, shown in Figure 3-7. This screen contains two columns. The column on the left contains variables that were sampled in the data sampling session. The column on the right contains a list of postprocessing equations that have already been added.

To add an equation, simply type the equation in the edit box above the column on the right. You may include the variable names from the column on the right by selecting them; the variable name will be inserted into your equation where your cursor is. When entering an equation, you must enter a new, unique variable name, a point value if you selected Accept the User-Specified Point Estimate Value on the LHS Processing Options screen, an equals sign (=), and the equation itself. You may use any of the following mathematical operators as well as numeric constants.

+	Addition
-	Subtraction
*	Multiplication
/	Division

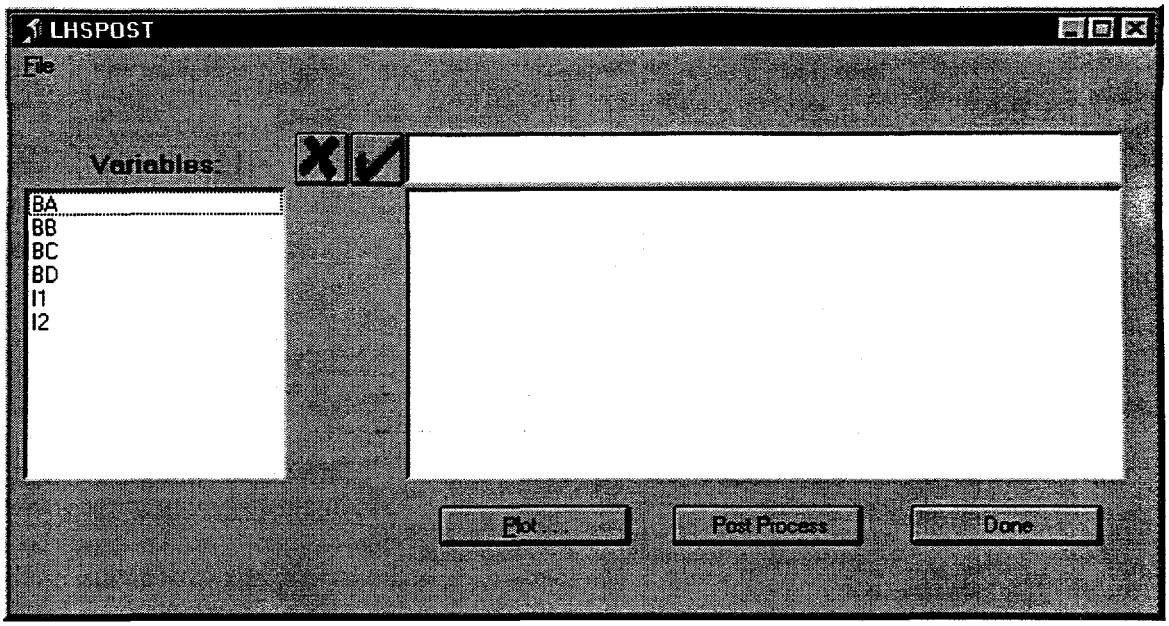


Figure 3-7: The Postprocessing Screen

^	Exponentiation
Min[ ]	Minimum of the enclosed values
Max[ ]	Maximum of the enclosed values
Neg( )	Negative of value
Log( )	Natural logarithm
Exp( )	Exponential of value (e^{value})
Sin( )	Sine
Cos( )	Cosine
Tan( )	Tangent
Int( )	Integer form (truncated decimal)
Abs( )	Absolute value
Sqrt( )	Square root
Sinh( )	Hyperbolic sine
Cosh( )	Hyperbolic cosine
Tanh( )	Hyperbolic tangent
Asin( )	Inverse sine
Acos( )	Inverse cosine
Atan( )	Inverse tangent
Nint( )	Integer form (rounding of decimal)
Log10( )	Base 10 logarithm
Zero1[ ]	Compares two values; if they are within 0.5 of each other, the value is 1.0, otherwise it is 0.0

There are several things to note about the mathematical expression:

- There is no scientific notation
- Square brackets can be used only with the Min, Max, and Zero1 function
- Values inside the square brackets must be separated by commas
- Nested square-bracket operators are not allowed
- You must place spaces before and after each operator, including hierarchical parentheses

Several valid equations are listed below.

Var1 0.003 = Var2 * Sin(Var3) + Var4
PumpFails = Pump1 * Pump2 * Pump3
BestOfFour = Min[Pump1 , Pump2 , Pump3 , Pump4]

Invalid equations are listed below.

Equation	Invalid Because . . .
Var1 0.003 = Var2 * (Sin(Var3) + Var4	Unclosed parentheses
Pumps = Pump1 * [Pump2 * Pump3 + Pump5]	Can only use square brackets with Min, Max, and Zero1 functions
BestOfThree = Min[Pump1,Pump2,Pump3]	No spaces between commas and square brackets

Once you have entered an equation, pressing the Checkmark button will add it to the list in the column on the right. Pressing the X button will cancel the operation and erase the equation. If you wish to edit the equation, select it from the column on the right and it will appear in the edit box.

Once you have added all of your equations, you can perform the calculations by pressing the Postprocess button. This will call up the LHSPOST Fortran DLL, which will perform all calculations. Once these are complete, you can either plot them with CRAVE by pressing the Plot button or press the Done button, which will take you to the Main screen in the LHS interface.

3.6 Additional Topics

3.6.1 Function of OK and Cancel

Pressing the OK button on any screen will save the information on the current screen and summarize it on the main screen. The Cancel button will revert to the "old" information before alterations were made.

3.6.2 Keyboard and Mouse Operations

For those who prefer using a keyboard to a mouse, several options are built into this program to allow flexibility between the two input devices.

If a caption on an option (such as a button or a menu option) has a letter that is underlined, pressing that letter and the Alt key simultaneously is equivalent to clicking the mouse on that option. For example, pressing Alt - S is the same as clicking Sample Data on the Main screen with the mouse.

The Tab key will move you forward around the screen to all valid options. The combination of the Shift and Tab keys will move you backward around the screen. Pressing the Enter key when positioned on a button will press the button. Pressing the space bar when positioned on a check box will check the box.

3.6.3 Getting Help

Four help options are available in the LHS interface, three of which are available through the Help menu on the Main screen: To learn how to use help files in a Windows environment, select *Help* | How to Use Help from the menu on the Main screen. To get a listing of all help topics for the LHS interface, select Help | Contents from the menu on the Main screen. To get help by searching for a specific keyword relative to the LHS interface, select Help | Search for Help On from the menu on the Main screen.

If you need help on a current screen or control, pressing F1 will activate the help files and display the portion related to the current screen or control.

3.6.4 Leaving Suggestions

If you have suggestions for this program, select Help | Suggestion Box from the menu. This will activate a screen (Figure 3-8), allowing you to enter your comments. Pressing Place in Suggestion Box will save your comments in a text file.

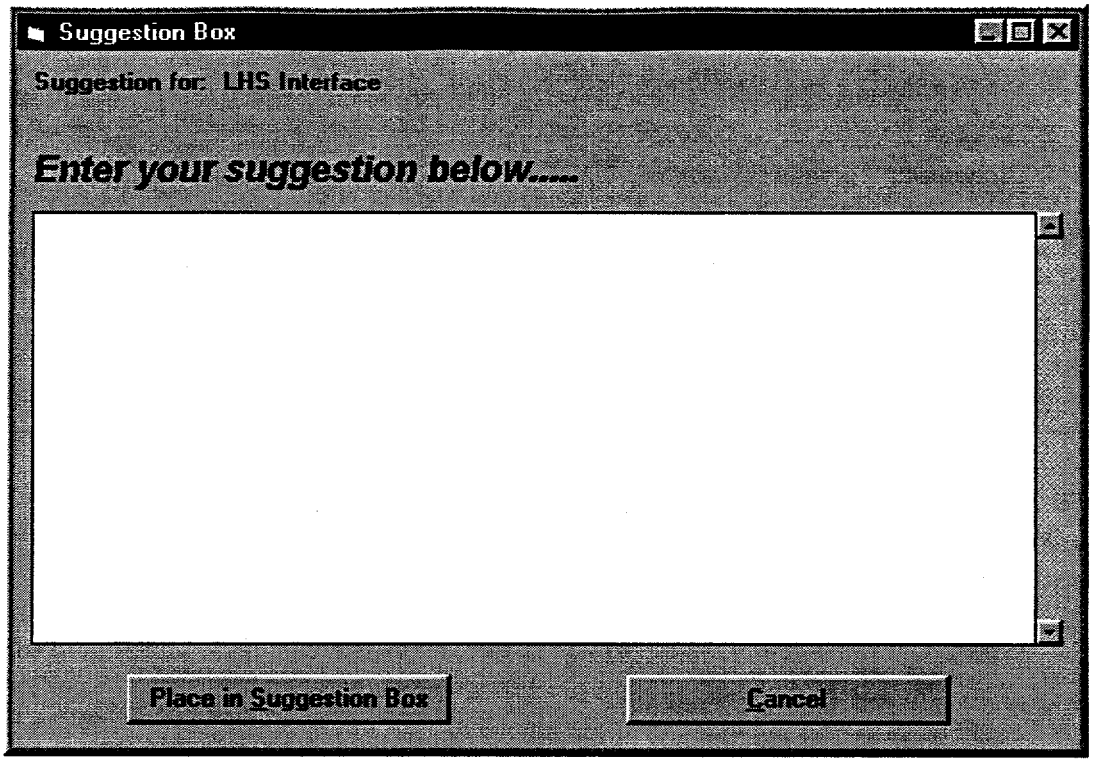


Figure 3-8: The Suggestion Screen

This page intentionally left blank

4 Distribution Input

The LHS program requires certain parameters to be defined in order to perform its sampling calculations. The program recognizes 33 types of keywords that dictate the type of distribution specified for each variable. Keywords that are used to control other aspects of the program input and output are generally not needed by the user because they are generated by automated user interfaces. However, for the sake of completeness, they are described in Appendix A.

Two principal methods are used to specify distributions to the LHS software: the Windows-based LHS user interface (available to the users of ARRAMIS™), and the model definition file (required when performing a SETAC event tree analysis). For users of the Windows-based interface, this chapter simply provides detailed descriptions of the distributions supported by LHS and the definitions of the parameters required to invoke them. For these users, Sections 4.1 and 4.3 are provided for informational purposes only. However, users of the model definition file method will find that Section 4.1 defines the format required by LHS for the model definition file, and the remainder of the chapter provides not only the distribution definitions, but also the format required by LHS in order to properly specify them. Examples are provided for all keywords described in this chapter to assist the analyst in the proper use of the software.

4.1 Input Characteristics

4.1.1 File Characteristics

All files read by LHS are based on a common file format that provides the user with a great deal of latitude in generating easily read, self-documenting input. Each LHS input file is a space-delimited free-format ASCII text file. As the file is read, its contents are treated as case-insensitive. The format supports continuation lines as well as whole-line and trailing comments.

LHS assumes that each input line will contain no more than 80 ASCII characters. If longer input lines are used, the code will ignore all input that is not contained within the first 80 columns. Each line is assumed to contain either a whole-line comment or a command (possibly followed by a trailing comment). Each command must be fully contained on a single line unless the line ends with a continuation character (either a "#" or a "%", preceded by one or more blanks). When a command is placed on more than one line, each line must end with a continuation character unless it terminates the command. On personal computers, when concatenated with all of its continuation lines, a line may contain up to 32,767 characters (not including redundant blank spaces and intervening comment lines).

The LHS input file format provides for both full-line and trailing comments. Under this format, all blank lines and all text following "$" characters (dollar sign) are considered comments. A full-line comment is any line that contains only blank spaces or a dollar sign as its first nonblank character. A trailing comment consists of any text that follows a dollar sign on any line in the file. In a trailing

comment, the dollar sign must be preceded by one or more blank spaces. Since LHS ignores all text that follows a dollar sign, it is not possible to place comments between items on a single line. Trailing comments may, however, be placed after a continuation character on a continued line, and full-line comments may be placed between a continued line and its continuation or between consecutive continuation lines.

A command may begin and end at any point on the line (an arbitrary number of leading and trailing spaces may be used as desired). If a command keyword is composed of more than one word, those words must be separated by *one and only one blank space*. However, the user may include as many blank spaces as desired between a keyword and any required alphabetic or numeric values. For example,

```
Data:   Roger    Bounded Normal  0.5  0.3   0.0 1.0
```

is a legal input line because each keyword or value (Data:, Roger, Bounded Normal, 0.5, 0.3, 0.0 and 1.0) may be preceded and followed by an arbitrary number of blank spaces, but only a single blank space is permitted within the keyword Bounded Normal. It should be noted that LHS treats commas (",") and tab characters as being fully equivalent to and interchangeable with blank space characters.

The alphanumeric input to LHS has been designed to allow maximum user flexibility. All character input is case-insensitive, so a user can provide any combination of upper- and lower-case characters for LHS input. Thus, the keywords Normal, NORMAL, normal and noRMaL are all considered equivalent by LHS.

All numeric input is read in Fortran list-directed (free-format) input. Thus, a user may specify numeric input in any Fortran-standard format, including integer (146), floating point (15.643), and exponential notation (1.426E-3).

4.1.2 Names of Random Variables

Since results from LHS are often used as input by other programs, LHS provides a way for those programs to properly identify the individual components of the sampled data in the LHS output file. LHS does this by requiring each specified distribution to be identified by a unique name. Each name must meet the following criteria:

- It may contain no more than 16 characters
- The "$" (dollar sign), "#" (number sign), "%" (percent sign), "," (comma), " " (blank space), and tab characters are not valid within names
- A name must not form a valid representation of a real number for an ANSII-standard Fortran compiler

Examples of valid and invalid distribution names are found in the following table.

Table 4-1: Valid and Invalid Distribution Names

Valid Distribution Names	Invalid Distribution Names	Reason Name is Invalid
Valve-Fails	Amount of Hydrogen	Includes spaces
Spark	%Hydrogen	Includes continuation character
PulseWidth	Repair$	Includes comment character
Hydrogen-Percent	1234567	A number
acceleration	Loss%	Includes continuation character
	Valve#1-Fail	Includes continuation character
	Failure,Valve1	Includes a comma
	TheValveFailsWideOpen	Too many characters

Any name that is longer than 16 characters will be truncated to 16 characters. These names appear in the LHS output file and are used to identify which distribution corresponds to which set of observations in that file.

4.1.3 Distribution File Structure

LHS was designed to allow the generation of integrated input files when used with other parts of ARRAMIS™. The objective is to allow a user to combine modeling information and the associated uncertainty analysis information in a single input file that can be read by all codes. This capability has been demonstrated within ARRAMIS™ in the SETAC event tree analysis code, as well as the SABLE fault tree analysis software and the TEMAC3 cut set uncertainty analysis module. In order to accomplish this goal, LHS has been designed to ignore much of the input file and select as its own input only those lines that meet strict selection criteria. LHS will take as its input any line from the file on which the first keyword is Data:. This allows a user to sprinkle LHS input throughout the input file without adversely affecting the input to the modeling code. With the exception of the Dataset block (described later), LHS ignores all lines in the input file that do not begin with Data:.

LHS also recognizes an option where its input is concentrated in a block at the bottom of a file. This block must begin with the keyword Dataset: on a line by itself. All lines that follow this statement are assumed to belong to LHS input regardless of whether they begin with the keyword Data:. Note that continuation and comment lines are permitted for both types of distribution input statements.

The keywords that can be used to describe distribution input are described later in this chapter. However, they all follow a common format:

name point_value dist-name dist-parm dist-parm .....

In this format, *name* is the variable name the user assigns to this distribution (described in the previous section). For each named distribution, LHS will generate not only sampled data, but also a

point estimate value. This value can be used by software models that cannot accept or effectively use the sampled results provided by LHS. At the user's discretion, this point value can be either the sample mean, the sample median, or a user-specified value. If the analyst wishes to use the user-specified value option, then he or she must enter that value using the *point_value* parameter. This field is required only when the analyst specifies that LHS is to use the "user-specified value" for the point estimate instead of computing the sample mean or median for this purpose. In all other instances, the *point_value* parameter is optional and can be omitted.[1] More details on how LHS generates point estimate values can be found in Chapter 3 and Appendices A and B.

The remaining parameters in this format are used to specify which distribution LHS is to sample for this random variable, and what that distribution's characteristics are to be. The *dist-name* parameter specifies which type of distribution is to be sampled (e.g., normal, uniform, hypergeometric), and the *dist-parm* values are the distribution parameters required by LHS to describe the particular distribution. Note that each distribution type requires its own unique set of parameters, and that the number of parameters required can vary from one to four (for standard distribution types) or more (for user-defined distribution types). The user is free to enter the required parameters in any format that he or she finds meaningful as long as it remains consistent with the format requirements described in this section. Note that this distribution definition statement must either be preceded by a Data: qualifier or be part of the Dataset: block at the end of the input file if the LHS software is to recognize it.

The following is an example of a valid LHS input file that contains a number of distribution and correlation statements:

```
Data:    MOV-1A-FTC  1.0E-3  Lognormal  1.0E-3   3
$  Here is a full line comment

$  This distribution is entered using several continuation
$  lines to aid the analyst in documenting the data
Data:  MOV-2B-FTRO   1.8E-4   % $ Point value from data
                   Lognormal  %
                     1.0E-4   % $ Mean, From ref. xxxx
                       3      % $ Error Factor

Data:  Recover-Time   Uniform  38  51
                     $ Bounds from ref. wxyz
```

[1] The analyst *must* specify that LHS use the user-specified point value if he or she intends that this LHS *input* file be used by the TEMAC3 cut set uncertainty analysis code.

```
Dataset:

Velocity  Bounded Normal  950  74  500  1500
Test-Results  Binomial  0.01 847

Fire Temperature  Continuous Linear   #
        7                 # $ 7 pairs of data
        300  0.0          # $ minimum value
        500  0.1          # $ 10 percentile
        650  0.25         # $ 25 percentile
        800  0.5          # $ median
        1000 0.75         # $ 75 percentile
        1400 0.95         # $ 95 percentile
        1900 1.0          # $ distribution maximum

Correlate  MOV-1A-FTC  MOV-2B-FTRO  0.4
```

4.2 Continuous Distributions Recognized by LHS

The keywords described in this section are used to specify which distributions are to be sampled by the LHS program. Users may select from among 21 traditional continuous distribution functions and 5 empirical continuous distribution functions in order to achieve the best possible representation of their data. Discrete distribution functions are described in the next section. The 21 continuous distributions are contained in 5 major groups: normal distributions, lognormal distributions, uniform and loguniform distributions, user-specified continuous distributions, and miscellaneous continuous distributions. The parameters for the various distributions are summarized in a brief tabular reference in Appendix F.

4.2.1 Normal Distributions

The LHS software provides the user with four different methods for sampling from the normal distribution. The normal distribution is defined by the density function

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left(\frac{(x-\mu)^2}{2\sigma^2}\right)} \quad -\infty < x < \infty,$$

where the distribution mean and variance are μ and σ^2, respectively. The standard deviation of the distribution, which is required by LHS as an input parameter for several normal distribution sampling methods, is denoted by σ. The first sampling method for the normal distribution samples over all quantiles. The remaining methods sample normal distributions that have been truncated or bounded.

NORMAL ***mean standard-deviation***

The keyword NORMAL specifies that a normal distribution is to be performed. The function is sampled over all quantiles. The normal distribution is defined by two input parameters: the mean (μ) and the standard deviation (σ). The mean may be any real value; however, the standard deviation must be positive.

Example: `Item1 0.0 NORMAL 0.0 1.0`

This example defines a normal distribution named "Item1" with a point value of 0.0, a mean of 0.0, and a standard deviation of 1.0. Figure 4-1 shows the PDFs of the three normal distributions with varying μ and σ.

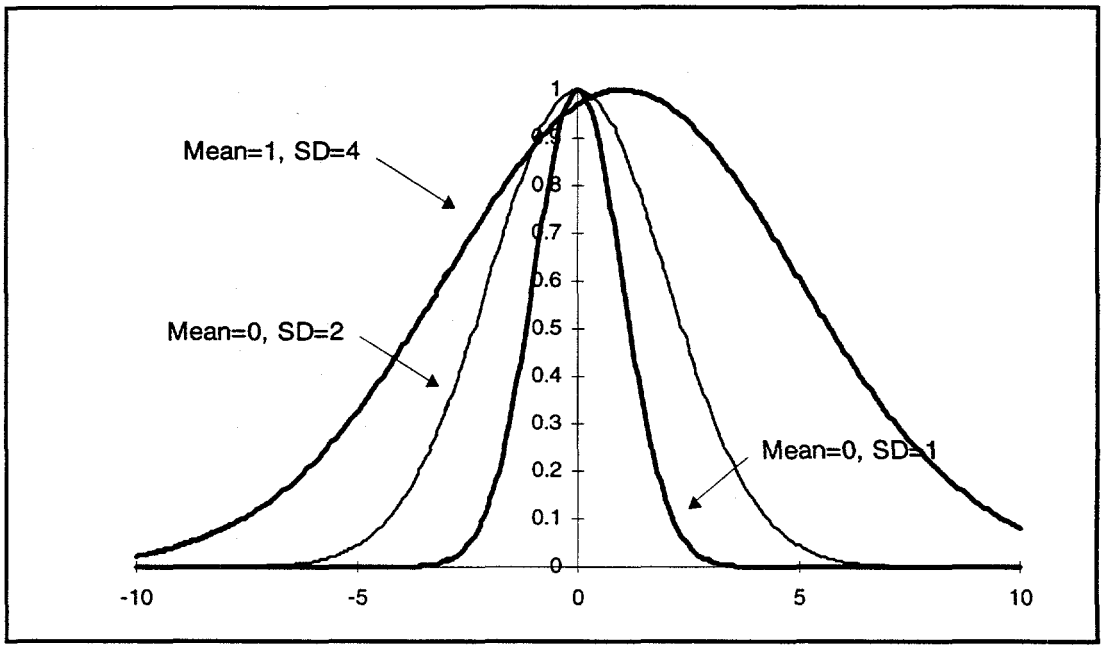


Figure 4-1: Normal Distribution PDF

TRUNCATED NORMAL ***mean standard_deviation lower upper***

A truncated normal distribution is a normal distribution that is only sampled between two user-specified quantiles. The TRUNCATED NORMAL distribution is defined by four input parameters: the mean and standard deviation of the normal distribution, the lower quantile beyond which sampling is not to occur, and the upper quantile beyond which sampling is not to occur. As above, the standard deviation must be positive. In addition, the lower and upper quantile values must satisfy the condition $0.0 \leq lower < upper \leq 1.0$.

Example: `Item2 0.5 TRUNCATED NORMAL 3.0 1.0 0.1 0.8`

This example defines a distribution with the name "Item2" and a point value of 0.5. The normal distribution with a mean of 3.0 and a standard deviation of 1.0 is to be sampled only between the 0.1 and 0.8 quantiles (the PDF for this distribution is shown in Figure 4-2).

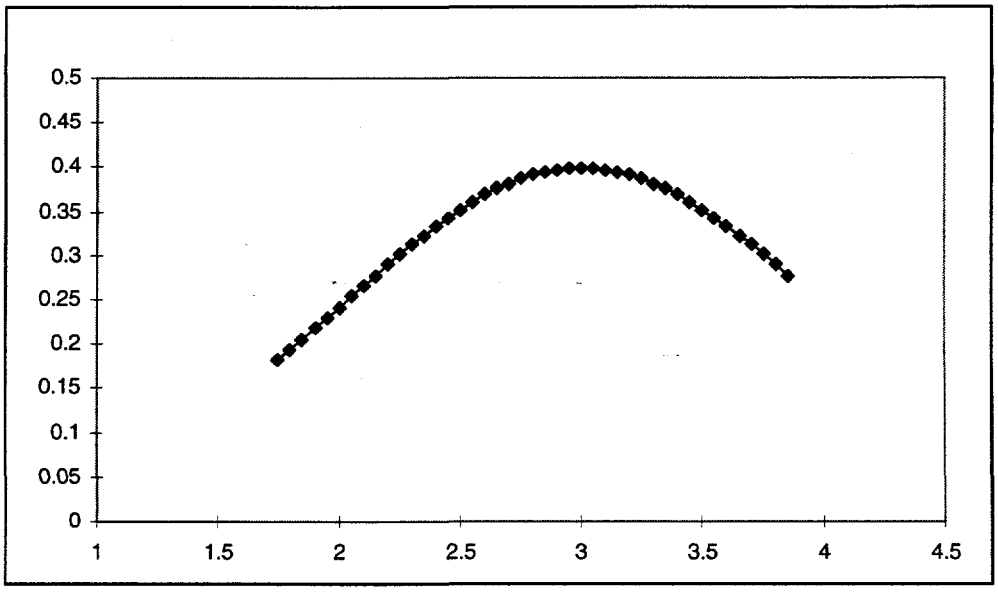


Figure 4-2: Truncated Normal Distribution PDF ($\mu = 3$, $\sigma = 1$, *lower* = 0.1, *upper* = 0.8)

BOUNDED NORMAL ***mean standard_deviation lower_bound upper_bound***

The bounded normal distribution is similar to the truncated normal distribution except that the user specifies the distribution values between which sampling is to occur, instead of the quantiles between which sampling is to occur. Four parameters are required: the mean and standard deviation of the sampled normal distribution, the lower function value beyond which sampling is not to occur, and the upper function value beyond which sampling is not to occur. The *lower_bound* parameter must be less than the *upper_bound.*

Example: `Item4 3.5 BOUNDED NORMAL 2.9 2.0 1.0 6.0`

This example defines a distribution named "Item4" with a mean of 2.9 and a standard deviation of 2.0 that is to be sampled only between the values 1.0 and 6.0. The point value of 3.5 is an optional input parameter if the user allows LHS to automatically compute a point value as the mean or median of the generated sample. Figure 4-3 shows the PDF of the normal distribution between 1.0 and 6.0.

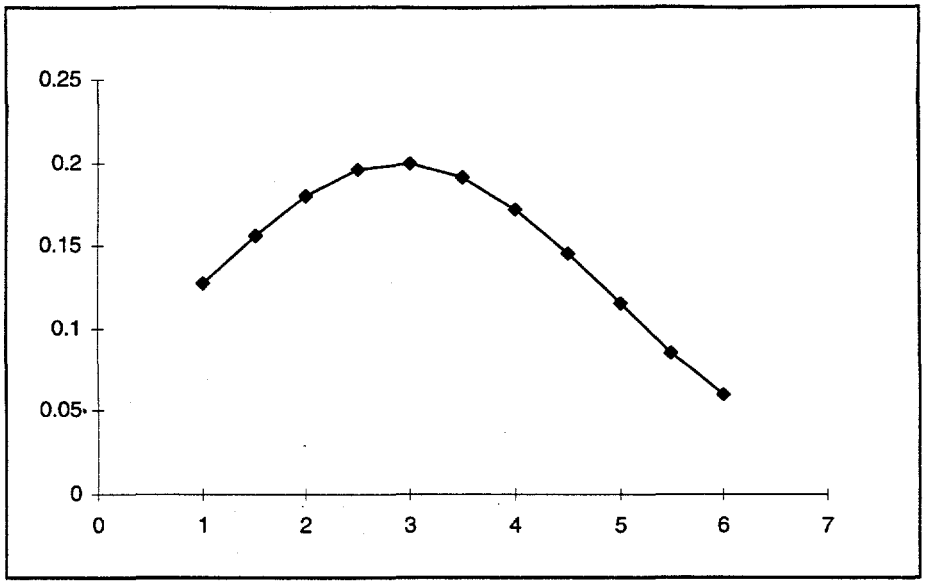


Figure 4-3: Bounded Normal Distribution PDF ($\mu = 2.9$, $\sigma = 2.0$, *lower_bound* = 1.0, *upper_bound* = 6.0)

NORMAL-B ***value_at_0.001 value_at_0.999***

This method of sampling a bounded normal distribution is provided mainly for backward compatibility with previous versions of LHS. It samples a normal distribution between the quantiles of 0.001 and 0.999. However, instead of specifying the mean and standard deviation of the normal distribution along with the bounds or quantiles to be sampled between, the user specifies only the range that is to be sampled. LHS *assumes* that the endpoints of this range are the values of the distribution at the 0.001 and 0.999 quantiles, respectively. These parameters can be related to the mean and the standard deviation of the distribution as follows:

$$V_{0.001} = \mu - 3.09023\sigma$$

$$V_{0.999} = \mu + 3.09023\sigma$$

Example: `Item3  5.85  NORMAL-B  2.0  9.7`

This example defines a normal distribution named "Item3" that has a range from 2.0 to 9.7. This range translates to a normal distribution with a mean of 5.85 and a standard deviation of 1.25 that is sampled only between the values of 2.0 and 9.7, or equivalently, between the quantiles of 0.001 and 0.999. In this example, the optional point value has been set to the mean value of the distribution. Figure 4-4 shows the normal-B distribution.

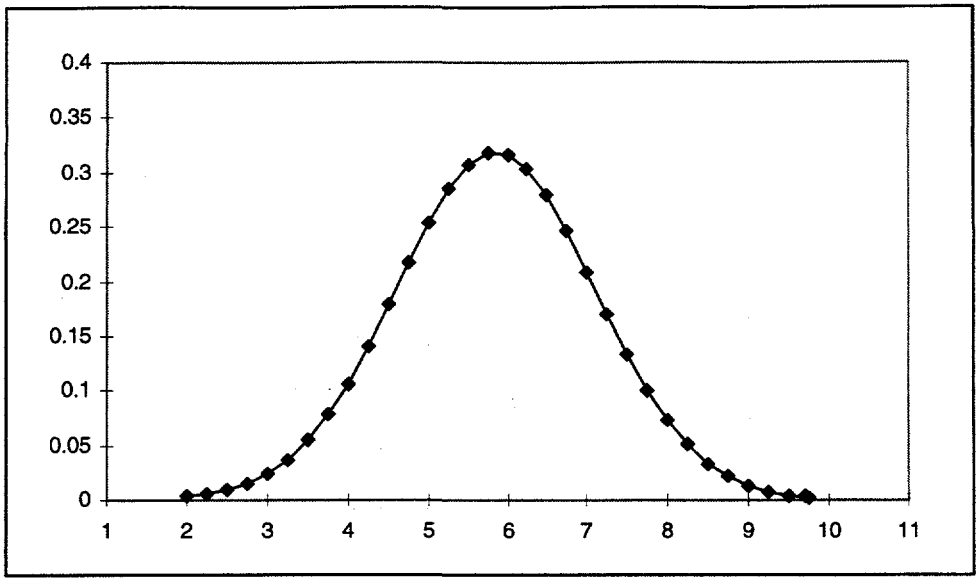


Figure 4-4: Normal-B Distribution PDF (*value_at_0.001* = 2.0, *value_at_0.999* = 9.7)

4.2.2 Lognormal Distributions

A lognormal distribution is simply a distribution whose logarithm is described by a normal distribution. A lognormal distribution is defined by a density function of

$$f(y) = \frac{1}{y\sigma\sqrt{2\pi}} \exp\left[\frac{-(\ln\ y - \mu)^2}{2\sigma^2}\right], \qquad y > 0$$

where μ and σ are the mean and standard deviations of the underlying normal distribution. The mean, variance, and median of the lognormal distribution can be computed from μ and σ using the following formulas:

$$E(y) = e^{\left(\mu + \frac{\sigma^2}{2}\right)}$$

$$V(y) = e^{\left(2\mu+\sigma^2\right)} \times \left(e^{\sigma^2} - 1\right)$$

$$\text{median} = e^{\mu}.$$

Lognormal distributions are typically specified in one of two ways throughout the literature. One way is to specify the mean and standard deviation of the underlying normal distribution (μ and σ) as described above. The other way is to specify the distribution using the mean of the lognormal distribution itself and a term called the "error factor." The error factor for a lognormal distribution is defined as the ratio of the 95th percentile to the median, or, equivalently, the ratio of the median to the 5th percentile. Physically, its square represents the width of a 90% confidence interval with

respect to the median. The mathematical relationship between the input mean and error factor, and the parameters of the underlying normal distribution (μ and σ) is shown by the following relations:

$$\sigma = \frac{\ln(\text{error factor})}{1.645}$$

$$\mu = \ln(\text{input mean}) - \frac{1}{2}\sigma^2 .$$

When the mean and error factor are used as input for the lognormal distribution, both input parameters must be positive, and the error factor must be greater than one. If μ and σ are specified, there is no restriction on μ, but σ must be positive.

LOGNORMAL *mean error_factor*

The keyword LOGNORMAL is used to specify a lognormal distribution that is sampled over all quantiles using the mean and the error factor as input parameters.

```
Example: Item7 0.32 LOGNORMAL    0.01    3.0
```

This example represents a lognormal distribution with a mean of 0.1 and an error factor of 3.0. This translates to an underlying normal distribution with a mean of –4.82818 and a standard deviation of 0.667849. The 90% confidence interval for this distribution is a factor of 9.0. Figure 4-5 shows the PDF of a lognormal distribution.

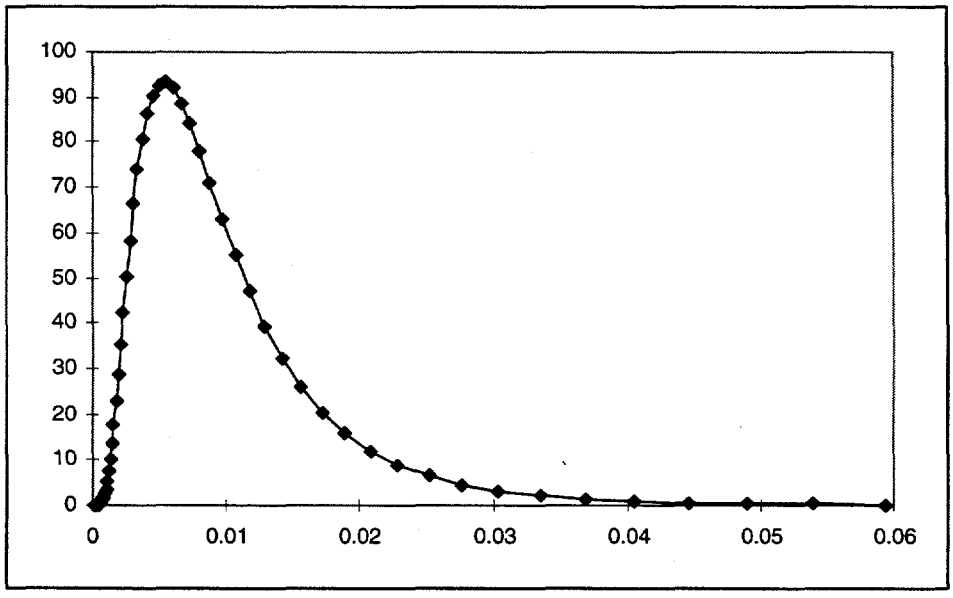


Figure 4-5: Lognormal Distribution PDF (*mean* = 0.01, *error_factor* = 3.0)

LOGNORMAL-N ***mean standard_deviation***

Use of the keyword LOGNORMAL-N will also produce a lognormal distribution sampled over all quantiles. With this keyword, however, the user must enter the distribution parameters as the mean and standard deviation of the underlying normal distribution (μ and σ).

Example: `Item6  0.2  LOGNORMAL-N  -4.82818  0.667849`

This example produces a lognormal distribution based on an underlying normal distribution with a mean of –4.82818 and a standard deviation of 0.667849. This lognormal distribution has a mean of 0.01 and an error factor of 3.0. This example is equivalent to the example shown for the lognormal distribution type above.

TRUNCATED LOGNORMAL ***mean error_factor lower upper***

The keyword TRUNCATED LOGNORMAL is used to describe a lognormal distribution that is sampled only between two user-specified quantiles using the mean and the error factor as input parameters. The *lower* and *upper* parameters represent the lower and upper quantiles beyond which sampling is not to occur. The lower and upper quantile values must satisfy the condition $0.0 \leq lower < upper \leq 1.0$.

Example: `Item10 TRUNCATED LOGNORMAL  2.0  2.0  0.13  0.86`

This example yields a lognormal distribution named "Item10" with a mean value of 2.0 and an error factor of 2.0. The distribution will only be sampled between the 13th and 86th percentiles (the 0.13 and 0.86 quantiles). In this example, the optional point value has been omitted.

TRUNCATED LOGNORMAL-N ***mean standard_deviation lower upper***

The keyword TRUNCATED LOGNORMAL-N is equivalent in every way to the keyword TRUNCATED LOGNORMAL except that in this case the user must enter the lognormal distribution parameters as the mean and standard deviation of the underlying normal distribution (μ and σ) as was done for the LOGNORMAL-N keyword described previously. This distribution will be sampled only between the quantiles specified by the *lower* and *upper* input parameters. The lower and upper quantile values must satisfy the condition $0.0 \leq lower < upper \leq 1.0$.

Example: `Item11 TRUNCATED LOGNORMAL-N  2.0  3.0  0.13  0.86`

This example produces a lognormal distribution whose underlying normal distribution has a mean of 2.0 and a standard deviation of 3.0, and is only sampled between the 13th and 86th quantiles. The equivalent mean and error factor are 665.1 and 139.1. In this example, the optional point value has been omitted.

BOUNDED LOGNORMAL ***mean error_factor lower_bound upper_bound***

A bounded lognormal distribution is similar to the truncated lognormal distribution described previously except that the user specifies the distribution values between which sampling is to occur, instead of the quantiles between which sampling is to occur. The keyword BOUNDED LOGNORMAL must be accompanied by the mean, error factor, the lower function value beyond which sampling should not occur and the upper function value beyond which sampling should not occur. The *lower_bound* parameter must be less than the *upper_bound*, and both values must be positive.

Example: `Item7 0.22 BOUNDED LOGNORMAL 0.34 2.0 0.1 2.2`

This example produces a lognormal distribution named "Item7" with a mean of 0.34, an error factor of 2.0, and a point estimate value of 0.22. The lognormal distribution will not be sampled for values less than 0.1 nor for values greater than 2.2.

BOUNDED LOGNORMAL-N ***mean standard_deviation lower_bound upper_bound***

The keyword BOUNDED LOGNORMAL-N is equivalent in every way to the keyword BOUNDED LOGNORMAL except that in this case the user must enter the lognormal distribution parameters as the mean and standard deviation of the underlying normal distribution (μ and σ) as was done for the LOGNORMAL-N keyword described previously. This distribution will be sampled only between the lognormal distribution values specified by the *lower_bound* and *upper_bound* input parameters. The *lower_bound* parameter must be less than the *upper_bound*, and both values must be positive.

Example: `Item8 1.0 BOUNDED LOGNORMAL-N 0.1 0.2 1.0 2.0`

This example would produce a lognormal distribution based on an underlying normal distribution with the parameters $\mu = 0.1$ and $\sigma = 0.2$. The distribution will be sampled only between the lognormal distribution values 1.0 and 2.0.

LOGNORMAL-B ***value_at_0.001 value_at_0.999***

This method of sampling a bounded lognormal distribution is provided mainly for backward compatibility with previous versions of LHS. It samples a lognormal distribution between the quantiles of 0.001 and 0.999. However, instead of specifying the mean and error factor of the lognormal distribution along with the bounds or quantiles to be sampled between, the user specifies only the range that is to be sampled. LHS *assumes* that the endpoints of this range are the values of the lognormal distribution at the 0.001 and 0.999 quantiles, respectively. These parameters must be positive, and *value_at_0.001* must be less than *value_at_0.999*. These parameters can be related to the mean and the standard deviation of the underlying normal distribution, as well as the lognormal mean and error factor as follows:

$$V_{0.001} = e^{(\mu - 3.09023\sigma)} = e^{\left(\ln(M) - \frac{1}{2}\left(\frac{\ln(E)}{1.645}\right)^2 - 3.09023\left(\frac{\ln(E)}{1.645}\right)\right)}$$

$$V_{0.999} = e^{(\mu + 3.09023\sigma)} = e^{\left(\ln(M) - \frac{1}{2}\left(\frac{\ln(E)}{1.645}\right)^2 + 3.09023\left(\frac{\ln(E)}{1.645}\right)\right)} ,$$

where M and E are the mean and error factor of the desired lognormal distribution, and μ and σ are the mean and standard deviation of the underlying normal distribution.

Example: `Item12  4.55  LOGNORMAL-B  2.0  9.7`

This example defines a lognormal distribution named "Item3" that has a range from 2.0 to 9.7. This range translates to an underlying normal distribution with a mean of $\mu = 1.48$ and a standard deviation of $\sigma = 0.255$, or equivalently, a lognormal mean of $M = 4.55$ and an error factor of $E = 1.52$. The distribution is sampled only between the values of 2.0 and 9.7, or, equivalently, between the quantiles of 0.001 and 0.999. In this example, the optional point value has been omitted.

4.2.3 Uniform and Loguniform Distributions

A uniform distribution, specified over a particular bounded interval (A, B), has the property that all points within that interval are equally likely. A loguniform distribution is simply a distribution whose logarithm is described by a uniform distribution. A uniform distribution is defined by the following probability density and cumulative distribution functions:

$$f(x) = \frac{1}{B-A}, \quad A \le x \le B$$

$$F(x) = \frac{x-A}{B-A}, \quad A \le x \le B .$$

The mean $E(x)$, median $M(x)$, and variance $V(x)$ are

$$E(x) = M(x) = \frac{A + B}{2}$$

$$V(x) = \frac{(B-A)^2}{12}$$

Similarly, a loguniform distribution is defined by the following probability density and cumulative distribution functions:

$$f(x) = \frac{1}{x}(\ln B - \ln A) \quad A < x < B$$

$$F(x) = \frac{\ln x - \ln A}{\ln B - \ln A}, \quad A < x < B .$$

The mean $E(x)$, median $M(x)$, and variance $V(x)$ are

$$E(x) = \frac{B - A}{\ln B - \ln A}$$

$$M(x) = \exp\left(\frac{\ln B + \ln A}{2}\right) = \sqrt{AB}$$

$$V(x) = (B - A)\,\frac{(\ln B - \ln A)(B + A) - 2(B - A)}{2(\ln B - \ln A)^2} .$$

UNIFORM ***A*** ***B***

A uniform distribution samples values uniformly between two specified intervals. The keyword UNIFORM, followed with two endpoints, A and B, will sample a uniform distribution.

Example: `Item13 0.5 UNIFORM 0.0 1.0`

This example samples a uniform distribution named "Item13" over the interval from 0.0 and 1.0. Figure 4-6 shows the PDF of a uniform distribution.

LOGUNIFORM ***A*** ***B***

The keyword LOGUNIFORM allows the logarithm of the variable to be sampled uniformly over the logarithm of the range specified by the two endpoints supplied. The endpoints A and B must be positive.

An example may help the uninitiated user understand how a loguniform distribution is sampled in LHS. Consider an example distribution with $A = 10^{-3}$ and $B = 10$. The logarithm (base 10 is used here for convenience) of this distribution, which has distribution endpoints $A_{\log 10} = -3$ and $B_{\log 10} = 1$, is sampled as a uniform distribution. The individual sample results are then converted back to the original distribution by a base 10 exponentiation process. Thus, one-quarter of the samples will

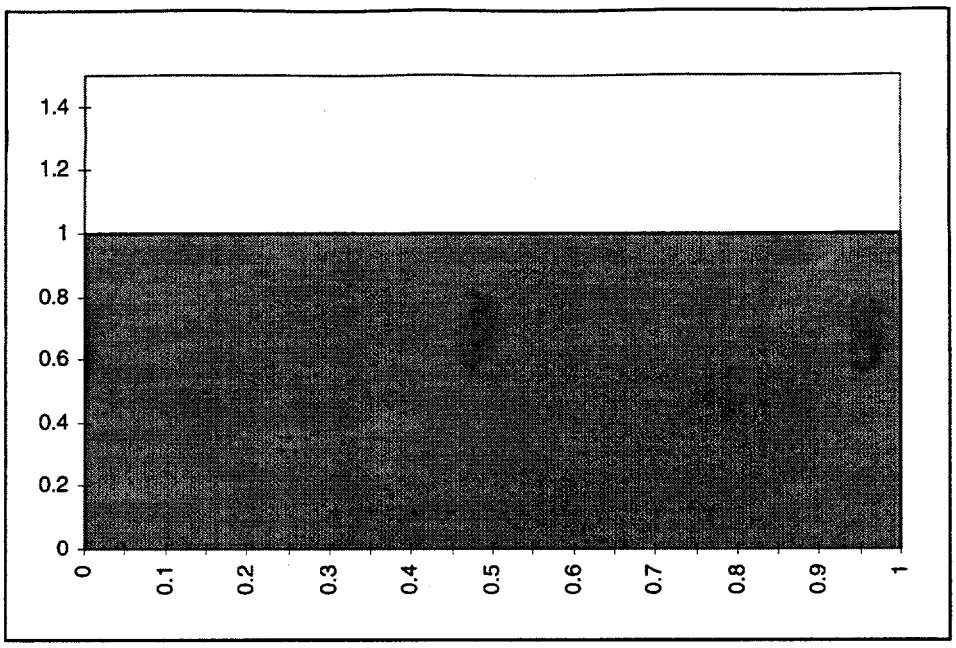


Figure 4-6: Uniform Distribution PDF ($A = 0.0$, $B = 1.0$)

be drawn between –3 and –2 for the underlying uniform distribution, another one-quarter will be drawn between –2 and –1, another one-quarter will be drawn between –1 and 0, and the final one-quarter will be drawn between 0 and 1. The lognormal variable will show one-quarter of its samples between 10^{-3} and 10^{-2}, another one-quarter between 10^{-2} and 10^{-1}, another one-quarter between 10^{-1} and 10^{0} (or 1) and the one-quarter between 1 and 10. Thus the variable is sampled uniformly on a logarithmic scale, and each decade is sampled with the same frequency.

Example: `Item14 0.375 LOGUNIFORM 0.001 10`

In this example, the variable named "Item14" with a point value of 0.375 will be sampled as a loguniform distribution over a range from 0.001 to 10. This input string implements the example distribution described in the previous paragraph. Figure 4-7 shows a Loguniform PDF.

4.2.4 User-Defined Continuous Distributions

The user-defined continuous distributions implemented within LHS represent various generalizations of the uniform and loguniform distributions described in the previous section. In each case, the user enters data that will be interpreted by LHS as the points for a multisection distribution that is, at the user's discretion, either piecewise-uniform or piecewise-loguniform. LHS provides five different methods by which these distributions can be entered. Three of these methods result in piecewise-uniform distributions (linear interpolation of the input data), while the other two result in piecewise-loguniform distributions (logarithmic interpolation of the input data).

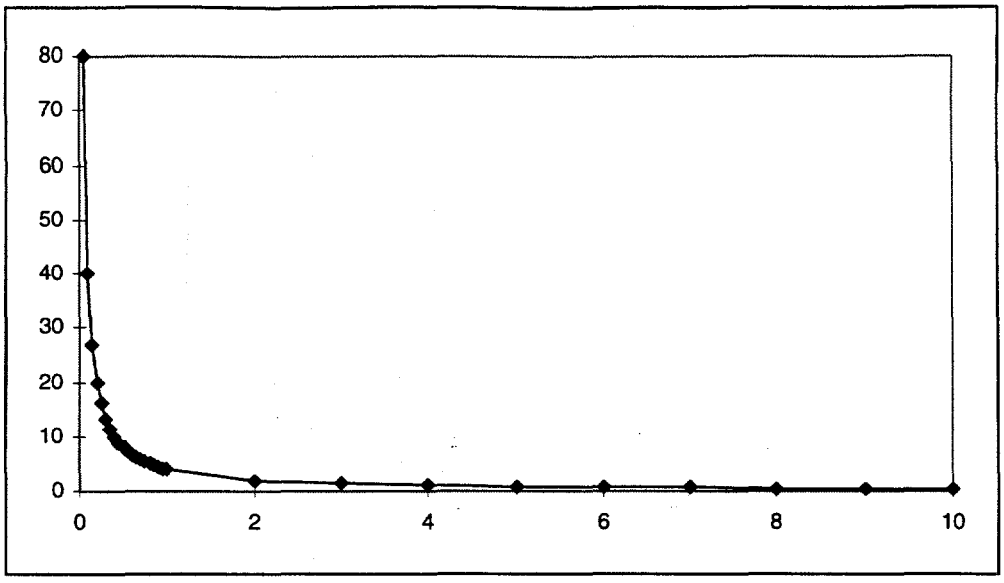


Figure 4-7: Loguniform PDF ($A = 0.001$, $B = 10$)

CONTINUOUS LINEAR ***n ordered_pair$_1$ ordered_pair$_2$ ... ordered_pair$_n$***

A continuous linear distribution is used to approximate an arbitrary distribution (either an empirical distribution or some other continuous distribution form not implemented within LHS) in terms of a set of piecewise uniform distributions. The user enters the CDF for this arbitrary distribution as a series of ordered pairs. The user must first specify *n*, an integer number of ordered pairs that will be used to represent the CDF ($n > 1$). The *n* ordered pairs that represent the CDF follow, forming the remainder of the input for this distribution. Within each ordered pair, the first number is the value of the distribution; the second number is the cumulative probability (CDF value) associated with this distribution value. *The probabilities in the ordered pairs (the second entry in the ordered pair) must increase monotonically starting with 0.0 and ending with 1.0.* The distribution values (the first entry in the ordered pair) must also increase monotonically. LHS uses these points to generate a piecewise-uniform distribution with the quantiles specified by the user. In other words, linear interpolation is used between the user-specified quantiles to define the CDF for this distribution. If $n = 2$, a uniform distribution is generated between the two points.

```
Example:  Item19    4.5     CONTINUOUS LINEAR    3   #
                    5.0    0.0    #
                    7.0    0.72   #
                   10.0    1.0
```

This example constructs an empirical CDF named "Item19" with a minimum at $y = 5.0$, a maximum at $y = 10.0$, and its 72nd percentile at $y = 7.0$. Note the use of the continuation characters (#) to place the input requirements on more than one line to facilitate user interpretation. This distribution will be sampled as a piecewise uniform distribution with 72% of the samples drawn uniformly over the interval (5.0, 7.0) and 28% of the samples drawn uniformly over the interval

(7.0, 10.0). Figure 4-8 shows the PDF of a continuous linear distribution corresponding to the example with "Item19".

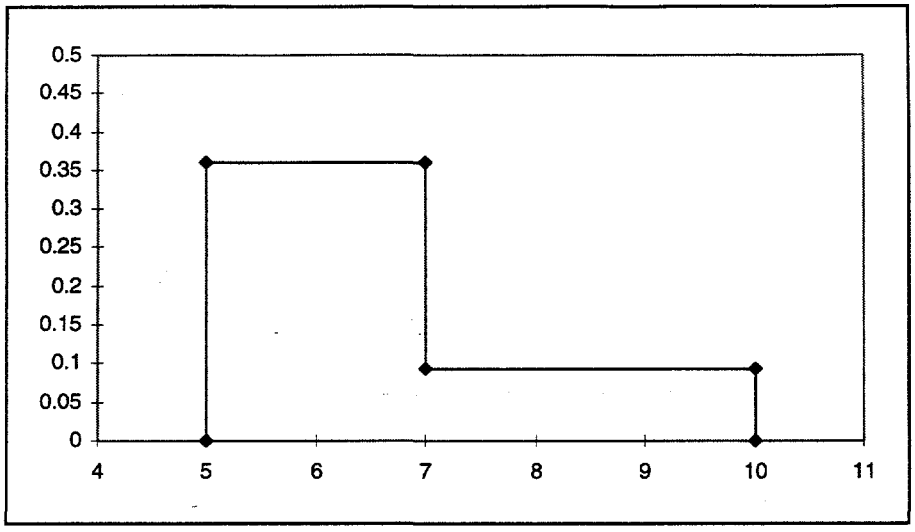


Figure 4-8: Continuous Linear PDF

CONTINUOUS LOGARITHMIC ***n ordered_pair$_1$ ordered_pair$_2$ ... ordered_pair$_n$***

A continuous logarithmic distribution is used to approximate an arbitrary distribution (either an empirical distribution or some other continuous distribution form not implemented within LHS) in terms of a set of piecewise loguniform distributions. Its input format and interpretation of data are identical to that of the continuous linear distribution above. LHS uses the specified ordered pairs to generate a piecewise-loguniform distribution with the quantiles specified by the user. In other words, logarithmic interpolation is used between the user-specified quantiles to define the CDF for this distribution. If $n = 2$, a loguniform distribution is generated between the two points.

```
Example:   Item20    5.4    CONTINUOUS LOGARITHMIC  4    #
                     0.01    0.0    #
                     0.05    0.35   #
                     0.1     0.79   #
                     1.0     1.0
```

This example constructs an empirical CDF named "Item20" with a minimum at $y = 0.1$, a maximum at $y = 1.0$, and its 35th and 79th percentiles at $y = 0.05$ and $y = 0.1$, respectively. This distribution will be sampled as a piecewise loguniform distribution with 35% of the samples based on a loguniform distribution over the interval (0.01, 0.05), 44% drawn similarly over the interval (0.05, 0.1), and the remaining 21% drawn from a loguniform distribution over the interval (0.1, 1.0). Note that both the distribution values (the first column in this example) and the quantiles (the second column in this example) increase monotonically. Figure 4-9 shows the PDF of a continuous logarithmic distribution corresponding to the example with "Item20".

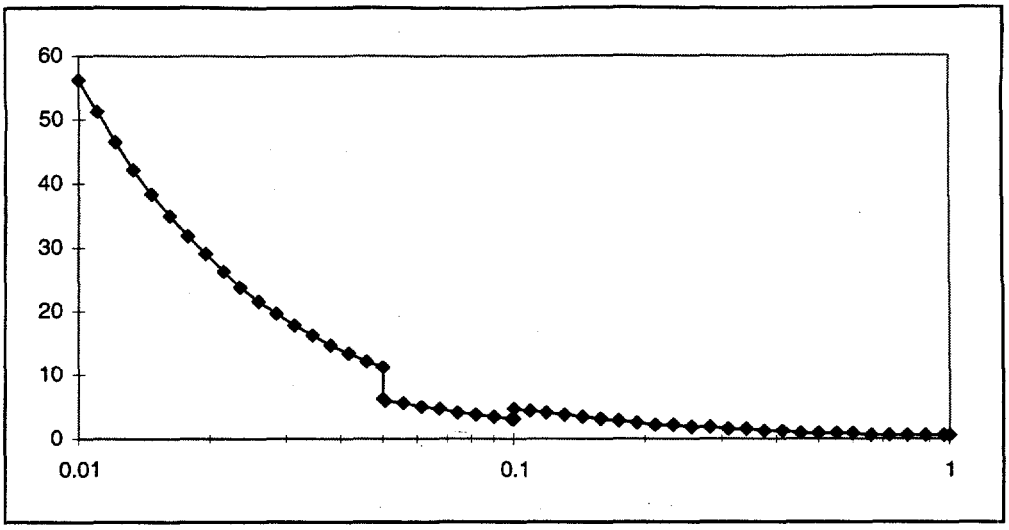


Figure 4-9: Continuous Logarithmic PDF

CONTINUOUS FREQUENCY ***n ordered_pair$_1$ ordered_pair$_2$ ... ordered_pair$_n$***

The keyword CONTINUOUS FREQUENCY provides an alternative method for specifying a piecewise uniform continuous distribution. It is similar in format to the continuous linear distribution described previously in that the user specifies the number of ordered pairs to be read, followed by a series of ordered pairs. Once again, the first value in each ordered pair is the value of the distribution at a particular point. However, instead of specifying the cumulative probability associated with that value as the second item in the ordered pair, as was done for the continuous linear distribution, the user specifies a relative frequency associated with the value. *The frequencies in the ordered pairs are relative only to each other, and must be positive.* The frequencies need not increase monotonically, although values must still increase monotonically.

LHS internally converts the continuous frequency distribution into a cumulative distribution function (continuous linear distribution) by assuming that between any two user-specified points, the relative probability density is the average of the input frequencies at the two points. The first and last points entered by the user are assumed to represent the minimum and maximum values of the distribution, respectively.

Example:
```
Item21   0.5   CONTINUOUS FREQUENCY   4   #
      11.0   1.0   #
      23.0  18.6   #
      30.0   7.2   #
      38.6   2.4
```

This example specifies a continuous distribution function that is piecewise uniform over three intervals. The relative probability density of the three intervals is 9.8, 12.9, and 4.8, so the total *relative* probability density over the entire distribution is 27.5. The relative interval probability densities are then normalized by the total relative probability density and aggregated to form the

cumulative distribution function that is sampled by LHS. Based on these values, the equivalent continuous linear representation for this example is:

```
Item21    0.5    CONTINUOUS LINEAR    4    #
      11.0    0.0       #
      23.0    0.356     #
      30.0    0.825     #
      38.6    1.0
```

If all input frequencies are identical, a uniform distribution is generated. If only two points are input, LHS adds an imaginary third point halfway between the two input points with a frequency of zero before converting the frequencies into a cumulative function. The continuous frequency distribution always uses linear interpolation. Figure 4-10 shows the PDF of the continuous frequency distribution corresponding to the "Item21" example.

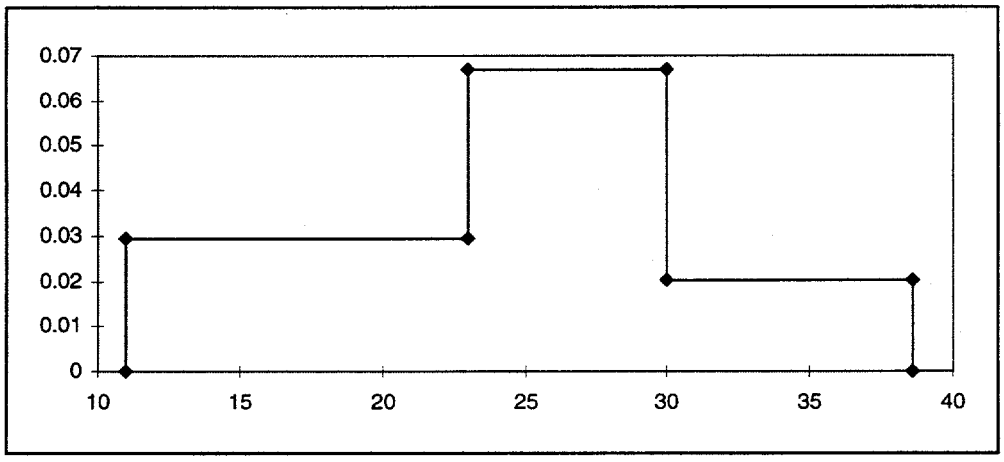


Figure 4-10: Continuous Frequency (Continuous Linear) PDF

UNIFORM* ***n*** ***obs$_1$*** ***obs$_2$ ... obs$_n$*** ***first_point*** ***endpoint$_1$*** ***endpoint$_2$ ... endpoint$_n$***

The user can enter a piecewise uniform distribution using an alternative format by specifying the UNIFORM* keyword. The results of sampling this distribution are similar to those generated by the continuous linear distribution type. This distribution enables the user to specify a piecewise uniform distribution in the form of an *n*-part histogram. The first parameter for this keyword specifies the number of intervals *n* for which piecewise uniform distributions will be generated. Next, the user specifies a series of *n* integer values, *obs*. Each value obs_i represents the number of observations that LHS will draw from the i^{th} interval. Therefore, *the sum of all obs_i must be exactly equal to the number of observations LHS has been requested to produce during this particular execution.* If the user wishes to run a second calculation with a different number of observations, the values of obs_i must be adjusted so that they sum to the new number of observations requested. Furthermore, each value of obs_i must be greater than or equal to zero.

Following the n values of obs_i, the user must specify the endpoints of the n intervals that are to be sampled. These intervals are assumed to be contiguous, so the endpoints of the n intervals are specified by entering a *first_point* (the distribution minimum) followed by n interval *endpoint* values. The value of *first_point* must be less than that of all *endpoint* values, and the *endpoint* values must be specified in increasing order. Note that the subintervals do not have to be of equal width.

```
Example:  Item13 0.29 UNIFORM*  3      #
               302    693     5        #
            -1.0    1.0    7.0    8.3
```

This example produces three piecewise uniform distributions. The first uniform distribution will select exactly 302 samples from the interval (–1.0, 1.0). The second uniform distribution will select exactly 693 samples from the interval (1.0, 7.0). The third uniform distribution will select exactly 5 samples from the interval (7.0, 8.3). This example would only be valid for problems that request LHS to generate 1000 observations from all distributions. For other requested numbers of observations, the second line of the example input would have to be changed so that it sums to the new requested number of observations. Figure 4-11 shows the PDF of a piecewise uniform distribution corresponding to the example "Item13".

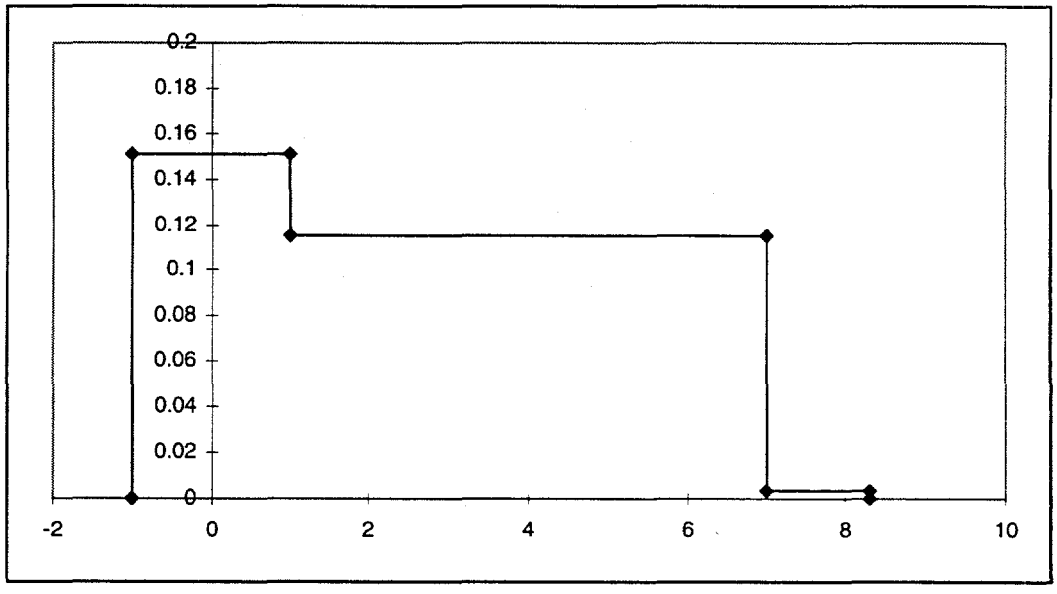


Figure 4-11: Piecewise Uniform PDF

LOGUNIFORM* ***num*** ***obs$_1$*** ***obs$_2$... obs$_n$*** ***first_point*** ***endpoint$_1$*** ***endpoint$_2$... endpoint$_n$***

The keyword LOGUNIFORM* generates a series of piecewise continuous LOGUNIFORM distributions in exactly the same way that UNIFORM* generates a series of piecewise UNIFORM distributions. All of the restrictions from the UNIFORM* distribution apply to this distribution as well. Furthermore, for the LOGUNIFORM* distribution, *first_point* and all *endpoint* values must be positive.

```
Example:  Item28   4.7   LOGUNIFORM*   3#
                182     723      95      #
            0.4     1.0     7.0     14.6
```

This example produces three piecewise loguniform distributions. The first loguniform distribution will select exactly 182 samples from the interval (0.4, 1.0). The second loguniform distribution will select exactly 723 samples from the interval (1.0, 7.0). The third uniform distribution will select exactly 95 samples from the interval (7.0, 14.6). This example would only be valid for problems that request LHS to generate 1000 observations from all distributions. For other requested numbers of observations, the second line of the example input would have to be changed so that it sums to the new requested number of observations. Figure 4-12 shows the PDF of a piecewise loguniform distribution corresponding to "Item28".

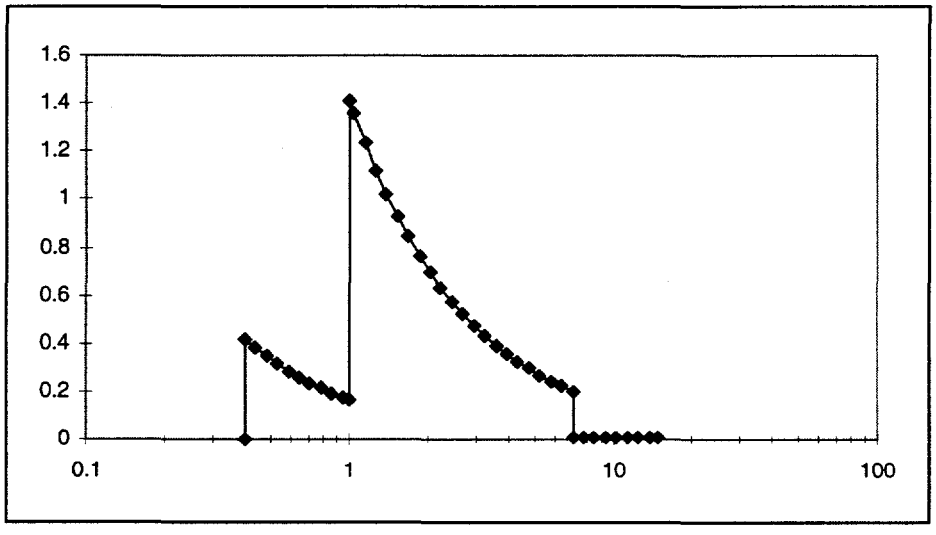


Figure 4-12: Piecewise Loguniform PDF

4.2.5 Miscellaneous Continuous Distributions

LHS recognizes a number of additional distributions that may be needed for specialized applications. This section describes how the exponential distribution, the maximum entropy distribution, the Weibull distribution, the Pareto distribution, the gamma distribution, the beta

distribution, the inverse Gaussian distribution, and the triangular distribution can be sampled using LHS.

EXPONENTIAL λ

An exponential distribution is used to predict the time between events (such as radioactive decay), or a lifetime with a constant probability of failure. An exponential distribution is defined by the following probability density and cumulative distribution functions:

$$f(x) = \lambda e^{-\lambda x}, \quad x \geq 0$$

$$F(x) = 1 - e^{-\lambda x}, \quad x \geq 0 .$$

The user must specify $\lambda > 0$.

Example: `Item28 0.36 EXPONENTIAL 2.0`

This example produces an exponential distribution named "Item28" with a point value of 0.36 and a distribution parameter λ = 2.0. Figure 4-13 shows the corresponding PDF of this exponential distribution.

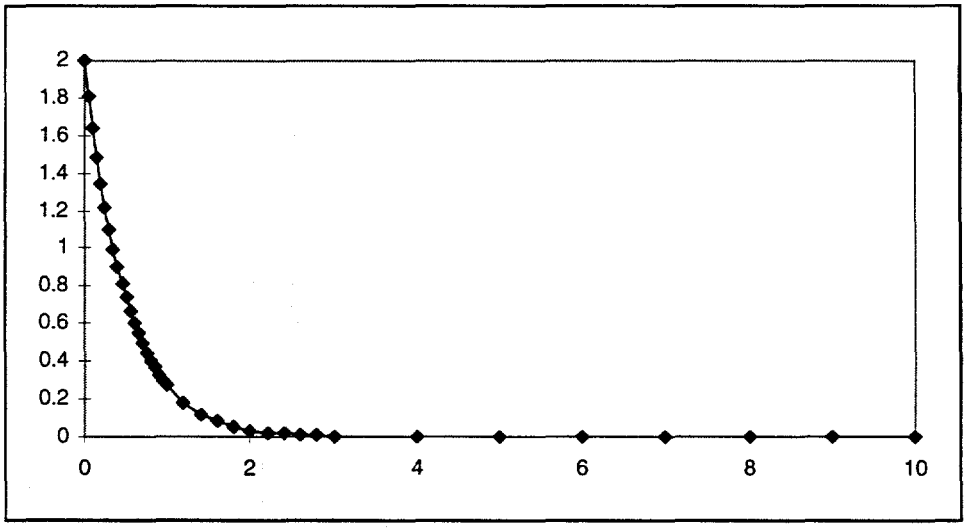


Figure 4-13: Exponential Distribution PDF ($\lambda = 2$)

MAXIMUM ENTROPY *A* μ *B*

The maximum entropy distribution implemented in LHS is a truncated exponential distribution, where *A* and *B* are the values of the distribution at its lower and upper bounds, respectively, and μ is the mean of the distribution. The user must specify $0 \leq A < \mu < B$. LHS then computes a distribution parameter λ for a bounded exponential distribution that provides the appropriate mean value.

Example: `Item33 0.25 MAXIMUM ENTROPY 0.0 1.0 3.0`

This example produces a truncated exponential distribution meeting maximum entropy criteria with a mean of 1.0, sampled between 0.0 and 3.0.

Some other types of distributions that satisfy maximum entropy criteria are implemented in LHS under other names. A maximum entropy distribution in which only the bounds (but no mean) are specified is equivalent to a uniform distribution. A maximum entropy distribution in which distribution bounds and quantiles are specified (but no mean) is equivalent to a cumulative linear distribution. The distribution form implemented with the MAXIMUM ENTROPY keyword is used when the analyst wishes to specify only the bounds and the mean of the distribution. Other maximum entropy distributions in which bounds, a mean, and a variance are specified, or in which bounds, quantiles, and a mean are specified, are not implemented in LHS. Figure 4-14 shows several PDFs for the maximum entropy distribution with A = 0.0, B = 3.0, and varying μ.

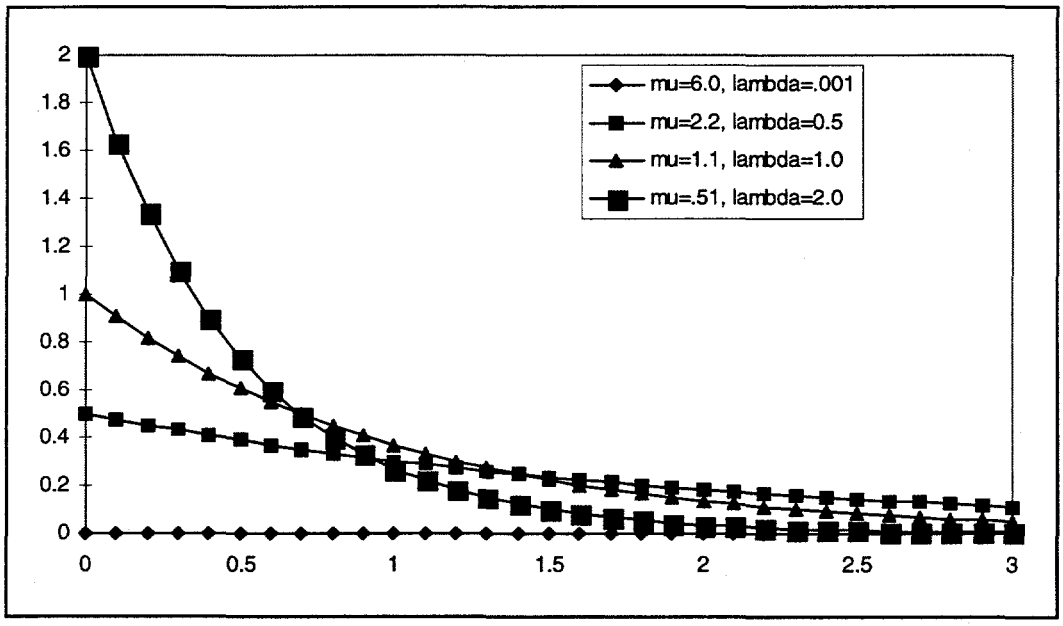


Figure 4-14: Maximum Entropy PDF ($A = 0.0$, $B = 3.0$, varying μ)

WEIBULL α β

A Weibull distribution is most commonly used in reliability studies to help predict the lifetime of a device. It is defined by the following density and cumulative distribution functions:

$$f(\omega) = \left(\frac{\alpha}{\beta}\right)\left(\frac{\omega}{\beta}\right)^{(\alpha-1)} e^{-\left(\frac{\omega}{\beta}\right)^{\alpha}}$$

$$F(\omega) = 1 - e^{-\left(\frac{\omega}{\beta}\right)^{\alpha}}, \quad 0 \leq \omega \leq \infty.$$

The user must specify α and β, both greater than zero.

Example: `Item29  0.35  WEIBULL  0.2  0.4`

This input produces a Weibull distribution named "Item29" with the distribution parameters α = 0.2 and β = 0.4. Figure 4-15 shows the PDF of the Weibull distribution corresponding to "Item29".

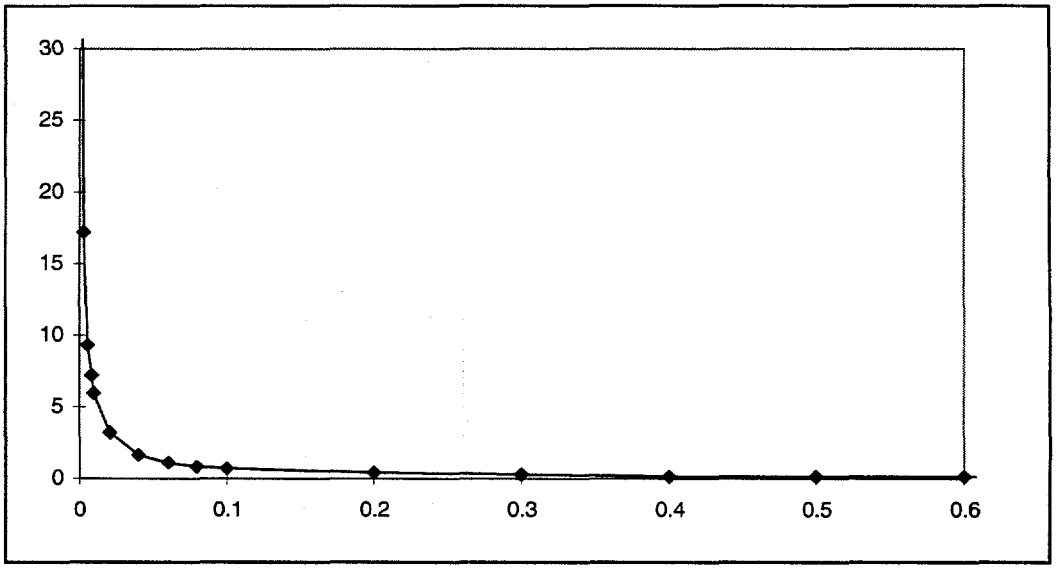


Figure 4-15: Weibull Distribution PDF ($\alpha = 0.2$, $\beta = 0.4$)

PARETO α β

A Pareto distribution is used to find a distribution of high numbers (such as middle/high incomes). It is defined by the following density and cumulative distribution functions:

$$p(x) = \frac{\alpha\, \beta^{\alpha}}{x^{\alpha+1}}$$

$$P(x) = 1 - \left(\frac{\beta}{x}\right)^{\alpha}, \quad \text{where} \quad \beta \leq x \leq \infty.$$

The user must specify $\alpha > 2.0$ and $\beta > 0.0$.

Example: `Item30 0.35 PARETO 2.4 0.5`

This example will produce a Pareto distribution named "Item30" with distribution parameters $\alpha = 2.4$ and $\beta = 0.5$. Figure 4-16 shows the PDF of the Pareto distribution corresponding to the "Item30" example.

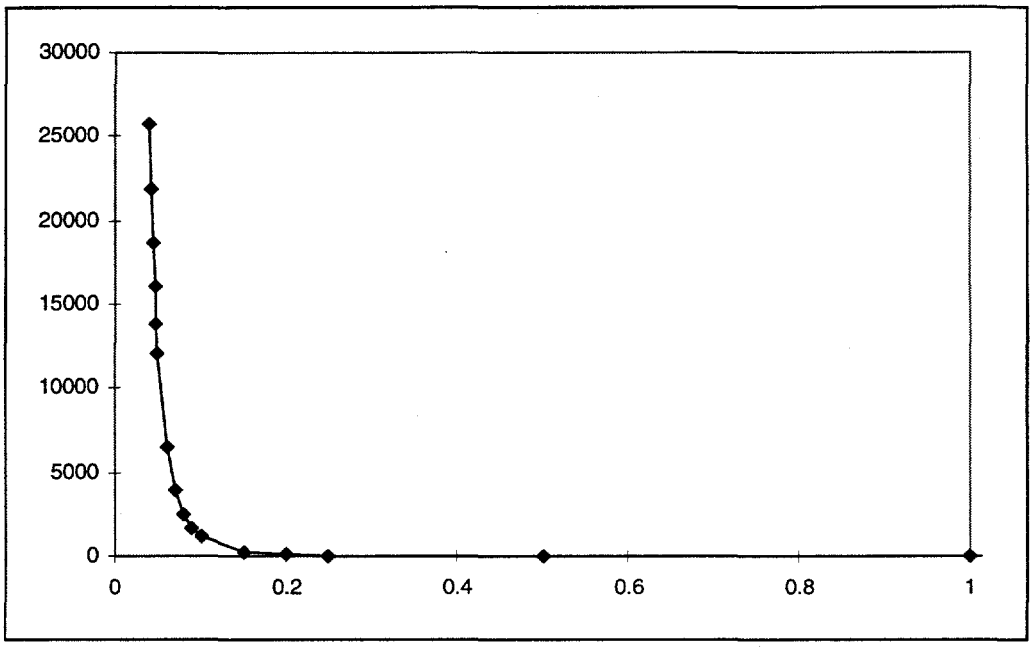


Figure 4-16: Pareto Distribution PDF ($\alpha = 2.4$, $\beta = 0.5$)

GAMMA α β

A gamma distribution is used to predict the time to finish a specific task. It has a density function defined by

$$f(x) = \frac{\beta^{\alpha}\, x^{(\alpha-1)}\, e^{-\beta x}}{\Gamma(\alpha)} \qquad \text{where} \qquad \Gamma(\alpha) = \int_0^{\infty} y^{(\alpha-1)} e^{-y}\, dy.$$

The user must specify α, the parameter of the gamma function, and β, a scaling factor, both of which are real numbers. The mean provided by this distribution is α/β. Note that the gamma

distribution is found in the literature in two different forms. These forms differ only in the definition of β. In the other form, β is defined as the reciprocal of its definition in this form. Thus the other form of the gamma distribution results in a mean value of αβ. The user should be careful in selecting the parameters for a gamma distribution to ensure that the parameters entered are based on the definition of β used in this program.

Example: `Item31 0.35 GAMMA 2.0 3.0`

The preceding example will produce a gamma distribution named "Item31" in which the gamma function parameter is 2.0 and the result is scaled by 3.0. Figure 4-17 shows the PDF of a gamma distribution.

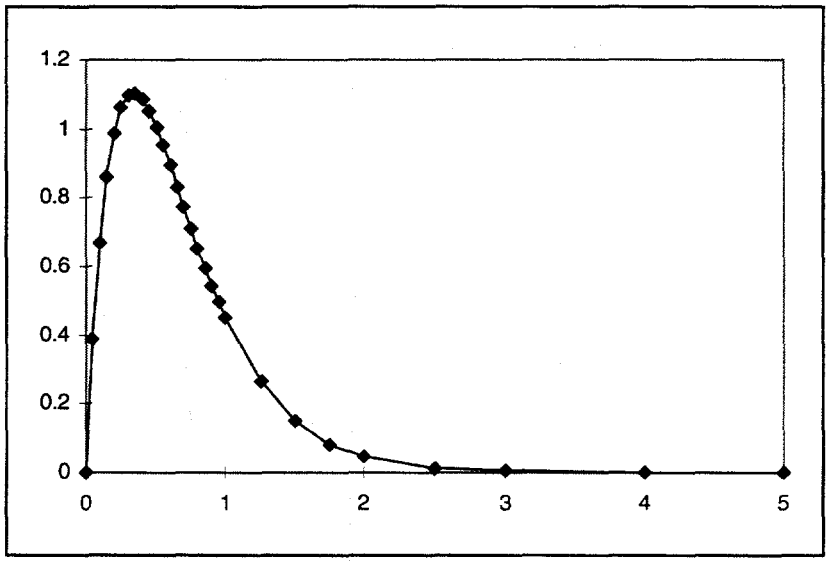


Figure 4-17: Gamma Distribution PDF ($\alpha = 2.0$, $\beta = 3.0$)

BETA ***A B p q***

A beta distribution is used as a rough model in the absence of sufficient data. The distribution is normalized to occur over a range from *A* to *B*, and is based on two shape factors, *p* and *q*. A beta distribution is defined with a density function of

$$f(\beta) = \frac{\beta}{\int_A^B \beta} \qquad \text{where} \qquad \beta = x^{(p-1)}(1-x)^{(q-1)},$$

p and *q* are shape parameters, and *A* and *B* are the endpoints of the distribution. The following conditions must be satisfied: $p, q \geq 0.001$, and $0 \leq A < B$. Figure 4-18 has been provided to help the user see the effect of various choices of *p* and *q*. 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.

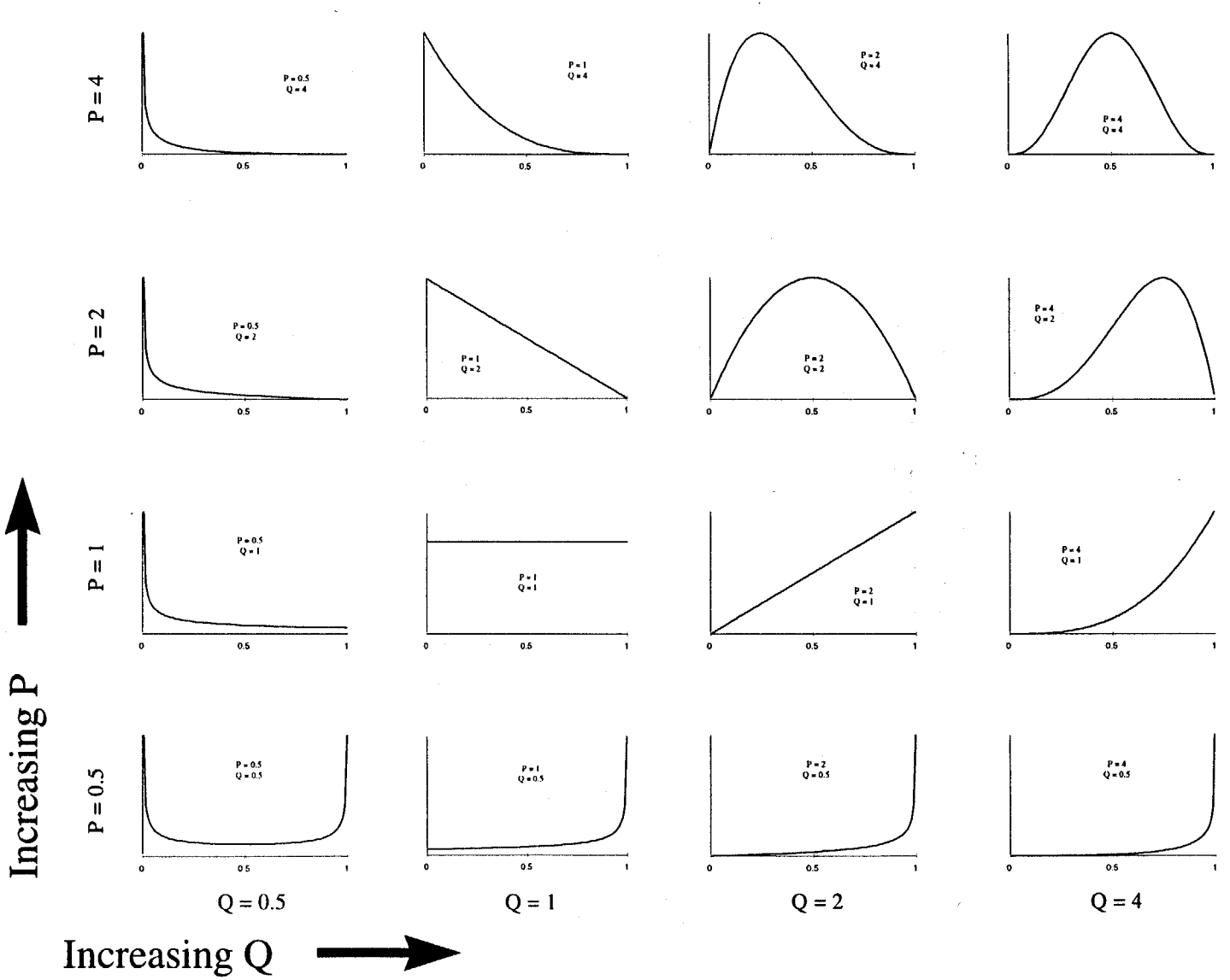


Figure 4-18: Effect of Varying P, Q for Beta Distribution

Example: `Item18  0.2  BETA  2.0  4.0  2.0  2.0`

This example defines a beta distribution named "Item18" that will generate sample values between 2.0 and 4.0 based on the shape factors $p = 2.0$ and $q = 2.0$.

INVERSE GAUSSIAN μ λ

An inverse Gaussian distribution is characterized by the density function

$$f(x) = \sqrt{\frac{\lambda}{2\pi x^3}}\, e^{-\left(\frac{\lambda\cdot(x-\mu)^2}{2\mu^2 x}\right)}.$$

The user must specify the distribution parameters μ and λ, both greater than zero.

Example: `Item32 0.35 INVERSE GAUSSIAN  0.01  0.3`

This example produces an inverse Gaussian distribution named "Item32" with a point value of 0.35 and distribution parameters $\mu = 0.01$ and $\lambda = 0.3$. Figure 4-19 shows the PDF of an inverse Gaussian distribution.

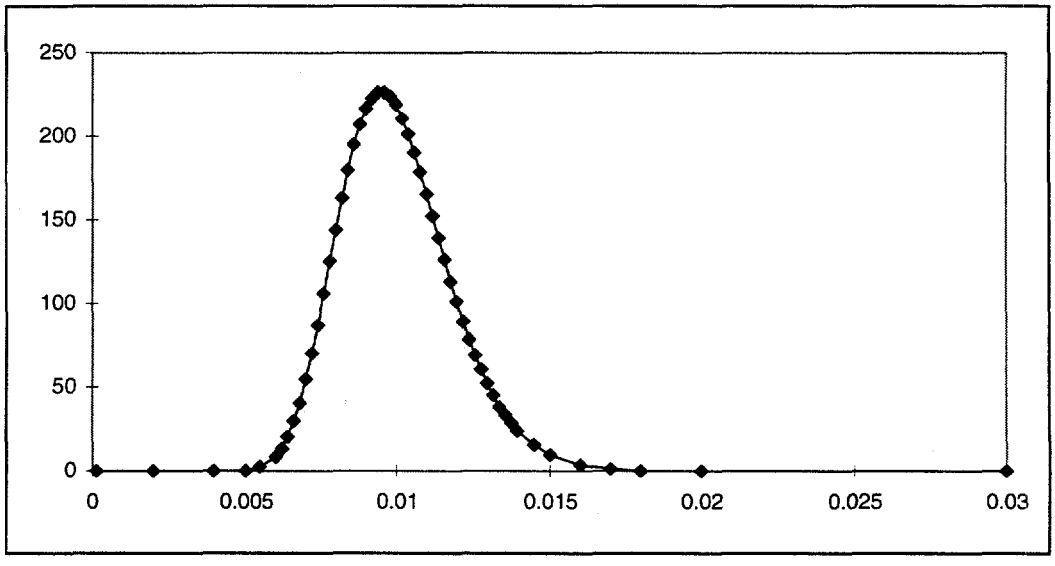


Figure 4-19: Inverse Gaussian PDF ($\mu = 0.01$, $\lambda = 0.3$)

TRIANGULAR *a b c*

A triangular distribution can be used to model certain situations where specific data are lacking. The probability density function for this distribution is a triangle with a distribution minimum at *a*,

a distribution maximum at c, and a most likely value at b (the apex of the triangle). In other words, all samples drawn from this distribution will come from the interval (a, c), and will be most densely populated in the area near b. No samples are drawn from the region where x is less than a or the region where x is greater than c. The values of the parameters must be such that $a < c$, and b must be located within the inclusive region $[a, c]$. In other words, b must satisfy the condition $a \le b \le c$.

For the most general case where $a < b < c$, the triangular distribution has the following probability density and cumulative distribution functions:

$$f(x) = \frac{2(x-a)}{(c-a)(b-a)} \quad a \le x \le b$$

$$= \frac{2(c-x)}{(c-a)(c-b)} \quad b \le x \le c$$

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

$$= \frac{b-a}{c-a} - \frac{(x+b-2c)(x-b)}{(c-a)(c-b)} \quad b \le x \le c .$$

In this general case, the mean, variance, and median of the distribution are

$$E(x) = \frac{a+b+c}{3}$$

$$V(x) = \frac{a(a-b) + b(b-c) + c(c-a)}{18}$$

$$\text{median} = a + \sqrt{\frac{(c-a)(b-a)}{2}} \quad b \ge \frac{a+c}{2}$$

$$= c - \sqrt{\frac{(c-a)(c-b)}{2}} \quad b \le \frac{a+c}{2}$$

The probability density function for this distribution is shown in Figure 4-20.

In the special case where $b = a$, the triangular distribution has the following probability density and cumulative distribution functions:

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

$$F(x) = \frac{(x-a)(2c-x-a)}{(c-a)^2} \quad a \le x \le c$$

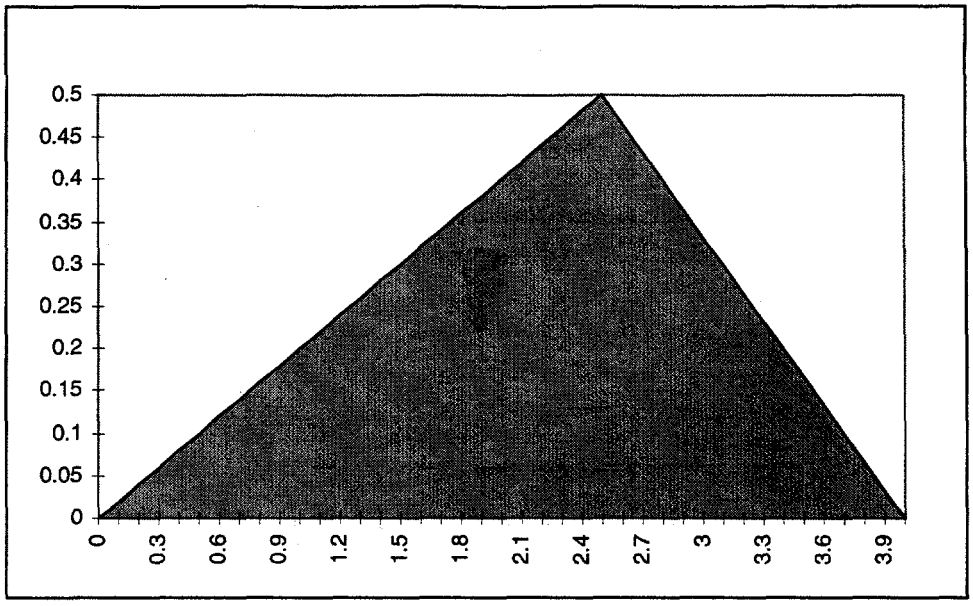


Figure 4-20: Triangular Distribution PDF ($a = 0.0$, $b = 2.5$, $c = 4.0$)

In this special case, the mean, variance, and median of the triangular distribution are

$$E(x) = \frac{2a + c}{3}$$

$$V(x) = \frac{(c - a)^2}{18}$$

$$\text{median} = c - \frac{c - a}{\sqrt{2}}.$$

The probability density function for this distribution is shown in Figure 4-21.

Finally, in the special case where $b = c$, the triangular distribution has the following probability density and cumulative distribution functions:

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

$$F(x) = \frac{(x - a)^2}{(c - a)^2} \quad a \le x \le c$$

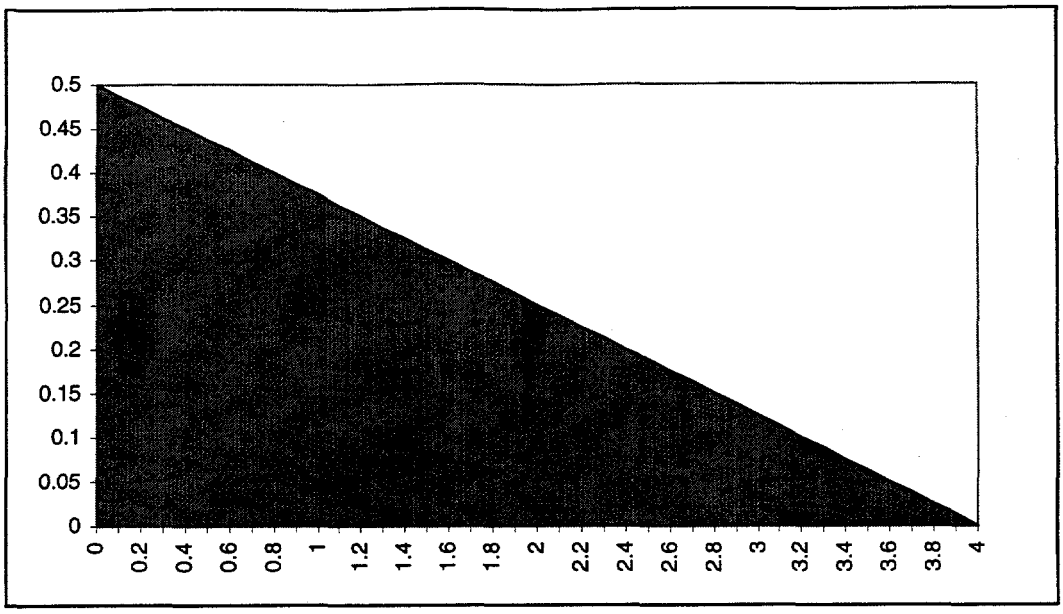


Figure 4-21: Triangular Distribution PDF ($a = b = 0.0$, $c = 4.0$)

In this special case, the mean, variance, and median of the triangular distribution are

$$E(x) = \frac{a + 2c}{3}$$

$$V(x) = \frac{(c - a)^2}{18}$$

$$\text{median} = a - \frac{c - a}{\sqrt{2}}.$$

The probability density function for this distribution is shown in Figure 4-22.

The following three examples illustrate these three cases of triangular distribution.

General case example: `Item15 2.2 TRIANGULAR 0.0 2.5 4.0`

Special case $b = a$ example: `Item16 0.37 TRIANGULAR 0.0 0.0 4.0`

Special case $b = c$ example: `Item17 3.24 TRIANGULAR 0.0 4.0 4.0`

Each of these examples defines a triangular distribution over the interval (0.0, 4.0). The first example has a most likely value of 2.5, which is within the interval, while the second example has a most likely value equal to the distribution minimum, and the third example has a most likely value equal to the distribution maximum.

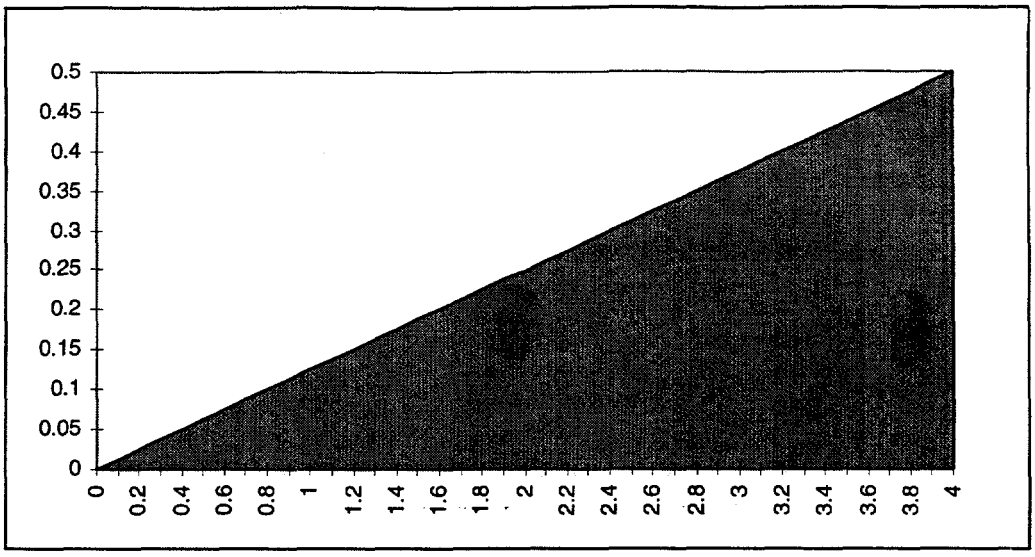


Figure 4-22: Triangular Distribution PDF ($a = 0.0,\ b = c = 4.0$)

4.3 Discrete Distributions Recognized by LHS

The keywords described in this section are used to specify which discrete distributions are to be sampled by the LHS program. Users may select from among five traditional discrete distribution functions and two empirical discrete distribution functions in order to achieve the best possible representation of their data. The traditional and empirical discrete distributions are described in the following two sections. The parameters for the various distributions are summarized in a brief tabular reference in Appendix F.

4.3.1 Common Discrete Distributions

LHS recognizes a number of common discrete distributions that an analyst may find useful for statistical modeling applications. This section describes how the Poisson, binomial, negative binomial, geometric, and hypergeometric distributions can be sampled using LHS.

POISSON λ

A Poisson distribution is used to help predict the number of discrete events that happen in a given time interval. The random variable x has a Poisson distribution if it has a density function of

$$f(x) = \frac{\lambda^x e^{-\lambda}}{x!} \qquad x=0, 1, 2, 3, \ldots$$

The user must specify a positive real value for the frequency λ. This is a discrete distribution that returns real number representations of sampled integer values. The code execution speed decreases significantly as the input parameter λ becomes large.

Example: `Item24  2.0  POISSON  3.0`

In this example, "Item24" specifies a Poisson distribution with a frequency of 3.0 and an input point estimate value of 2.0. Figure 4-23 shows the PDF for two sample Poisson distributions.

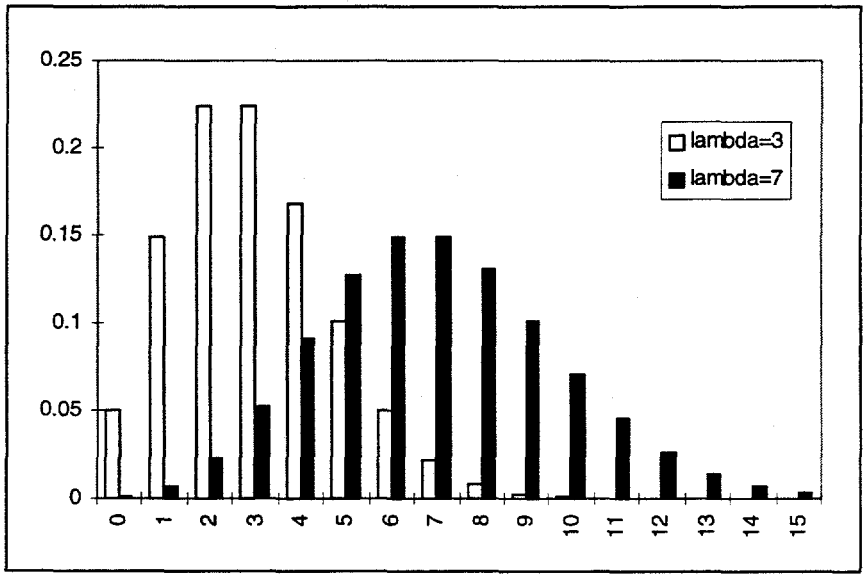


Figure 4-23: Poisson PDF with Varying λ

BINOMIAL *p n*

A binomial distribution is used to predict the number of failures (or defective items) in a total of n tests (or a batch of n products) with a probability p of being a failure (or the probability of being defective). A binomial distribution is defined by the density function

$$f(y) = \binom{n}{y} p^y (1-p)^{(n-y)} \qquad y=0, 1, 2, \ldots$$

The user must specify p, the failure probability ($0 < p < 1$), and n, the number of tests ($n > 1$). This is a discrete distribution that returns real number representations of sampled integer values. The code execution speed decreases as the integer input parameter n becomes large.

Example: `Item26 0.34 BINOMIAL 0.45 50`

This example specifies a binomial distribution in which each observation of "Item26" contains the number of failures in a hypothetical series of 50 tests, where each unit has a failure probability of 45%. Figure 4-24 shows the PDF of a binomial distribution.

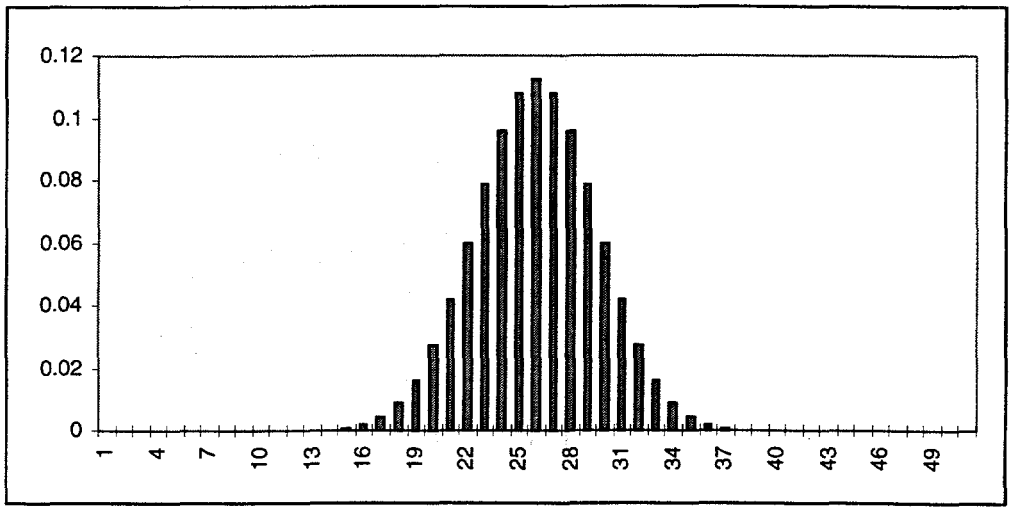


Figure 4-24: Binomial Distribution PDF ($p = 0.45$, $n = 50$)

NEGATIVE BINOMIAL *p n*

A negative binomial distribution is used to find the number of times to perform a test to have n successes, with a probability of success of p. A negative binomial distribution is defined by the density function

$$f(x) = \binom{n+x-1}{x} p^n (1-p)^x$$

The user must specify the probability of success p ($0 < p < 1$) and a number of tests n ($n > 1$). This is a discrete distribution that returns real number representations of sampled integer values.

Example: `Item27 67 NEGATIVE BINOMIAL 0.5 100`

In this example, the variable "Item27" is represented by a negative binomial distribution in which each observation contains the number of trials needed to achieve 100 successful trials with a probability of 50% to achieve success. Figure 4-25 shows the PDF of a negative binomial distribution.

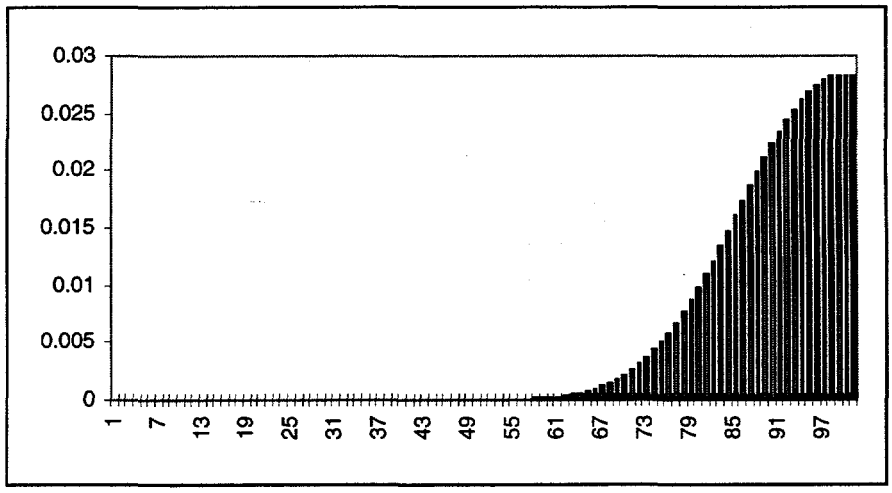


Figure 4-25: Negative Binomial PDF ($p = 0.5, n = 100$)

GEOMETRIC *p*

A geometric distribution represents the number of successful trials that might be observed before a failure occurs. This is a discrete distribution that is defined by the density function

$$f(y) = (1 - p)^y p \qquad y=0, 1, 2, \ldots$$

This is a discrete distribution that returns real number representations of sampled integer values. The user must specify p, the probability ($0 < p < 1$) of success. The code execution speed decreases as the input parameter p becomes small.

Example: `Item25 0.43 GEOMETRIC 0.67`

This example produces a geometric distribution named "Item25" that represents the number of successful trials until a failure with a 67% chance of success. Figure 4-26 shows the PDF of a geometric distribution.

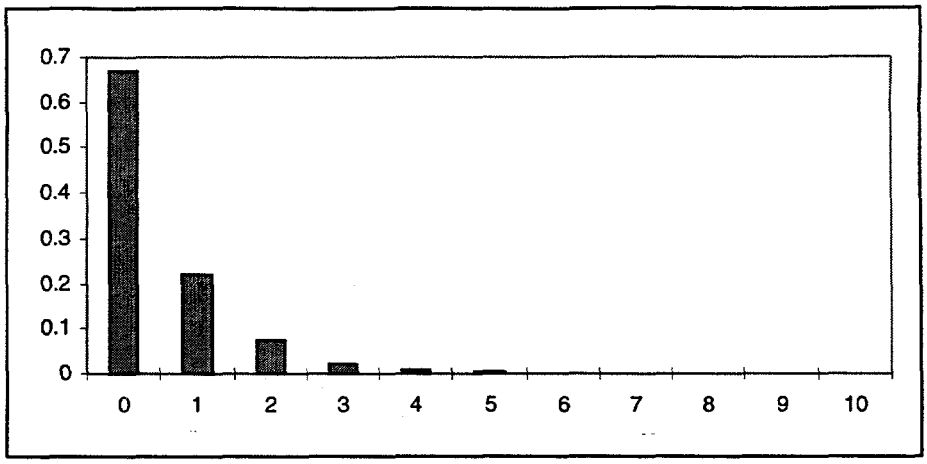


Figure 4-26: Geometric Distribution PDF ($p = 0.67$)

HYPERGEOMETRIC N_N N_1 N_R

A hypergeometric distribution is used to define the number of failures in a set of tests that has a known proportion of failures. A random variable x is said to have a hypergeometric distribution if it satisfies the following density function:

$$f(x) = \frac{\binom{N_1}{x}\binom{N_N - N_1}{N_R - x}}{\binom{N_N}{N_R}}$$

where N_N is the total number of tests, N_1 is the number of items in the subpopulation (number of tests that failed), and N_R is the number of selections. Note that $N_1 < N_R < N_N$. This is a discrete distribution that returns real representations of sampled integer values.

Example: `Item27 0.34 HYPERGEOMETRIC 110 30 45`

In this example, each observation of the variable "Item27" will contain the number of failed tests that were found in a randomly picked batch of 30 out of 110 total tests, where there are 45 total failed tests. Figure 4-27 shows the PDF of a hypergeometric distribution.

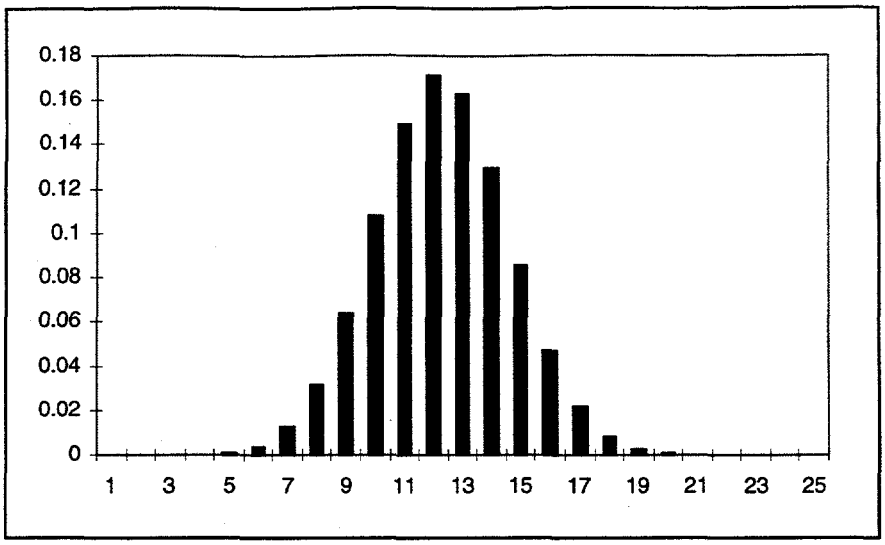


Figure 4-27: Hypergeometric Distribution PDF ($N_N = 110$, $N_I = 30$, $N_R = 45$)

4.3.2 User-Defined Discrete Distributions

The user-defined discrete distributions implemented within LHS provide an opportunity to specify the points of a discrete probability distribution function as either a cumulative distribution function or as a histogram (an unnormalized discrete density function). The discrete histogram function is converted internally by LHS into a discrete cumulative distribution function prior to sampling. The input format for these distributions is similar to their continuous counterparts described in Section 4.2.4.

DISCRETE CUMULATIVE ***n ordered_pair ordered_pair ordered_pair ...***

A discrete cumulative distribution is used to represent an arbitrary empirical distribution for which a limited number of discrete outcomes may occur. The user enters the CDF for this arbitrary distribution as a series of ordered pairs. The user must first specify n, an integer number of ordered pairs that will be used to represent the CDF ($n > 1$). The n-ordered pairs that represent the CDF follow, forming the remainder of the input for this distribution. Within each ordered pair, the first number is the value of the distribution; the second number is the cumulative probability (CDF value) associated with this distribution value. *The probabilities in the ordered pairs (the second entry in the ordered pair) must increase monotonically starting with a value greater than 0.0 and ending with 1.0.* The distribution values (the first entry in the ordered pair) must also increase monotonically. LHS uses these points to generate a discrete distribution in which the probability *density* associated with a particular value is the difference between its cumulative probability and that of its immediate predecessor (or, for the first value, the difference between its cumulative probability and zero).

```
Example:  Item22 7.5 DISCRETE CUMULATIVE          3 #
                      5.0 0.33333  #
                      7.0 0.66667  #
                     10.0 1.0
```

In this example, one-third of the observations for the random variable named "Item22" will be set to exactly 5.0, another third to exactly 7.0, and the remainder to exactly 10.0. The PDF for this distribution is shown in Figure 4-28.

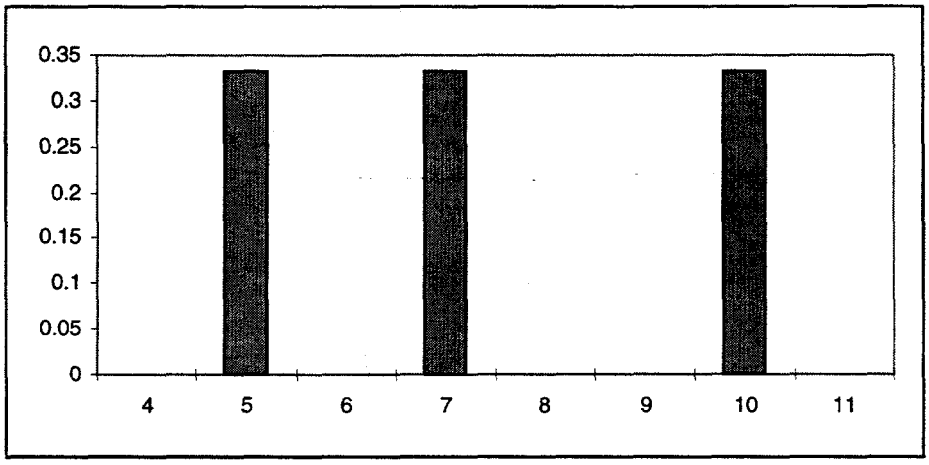


Figure 4-28: Discrete Cumulative PDF

DISCRETE HISTOGRAM ***n ordered_pair$_1$ ordered_pair$_2$ ... ordered_pair$_n$***

The keyword DISCRETE HISTOGRAM provides an alternative method for the user to specify an empirical discrete distribution. It is similar in format to the discrete cumulative distribution described previously, in that the user specifies the number of ordered pairs to be read, followed by a series of ordered pairs. Once again, the first value in each ordered pair is the value of the distribution at a particular point. However, instead of specifying the cumulative probability associated with that value as the second item in the ordered pair, as was done for the discrete cumulative distribution, the user specifies the relative frequency associated with the value (or the length of the "bar" that would be associated with that value on a histogram graph). *The frequencies in the ordered pairs are relative only to each other, and must be positive.* The frequencies need not increase monotonically, although values must still increase monotonically.

```
Example:  Item23   7.5   DISCRETE HISTOGRAM      3   #
                          5.0   17.0   #
                          7.0   17.0   #
                         10.0   17.0
```

This example generates a discrete distribution named "Item23" that is equivalent to the distribution generated in the example for the discrete cumulative distribution in that one-third of the observations for this random variable will be set to exactly 5.0, another third to exactly 7.0, and the

remainder to exactly 10.0. This occurs because all three discrete values are associated with the same relative frequency (the value 17.0). LHS internally converts this histogram into a discrete continuous distribution prior to sampling. This distribution is shown in Figure 4-29.

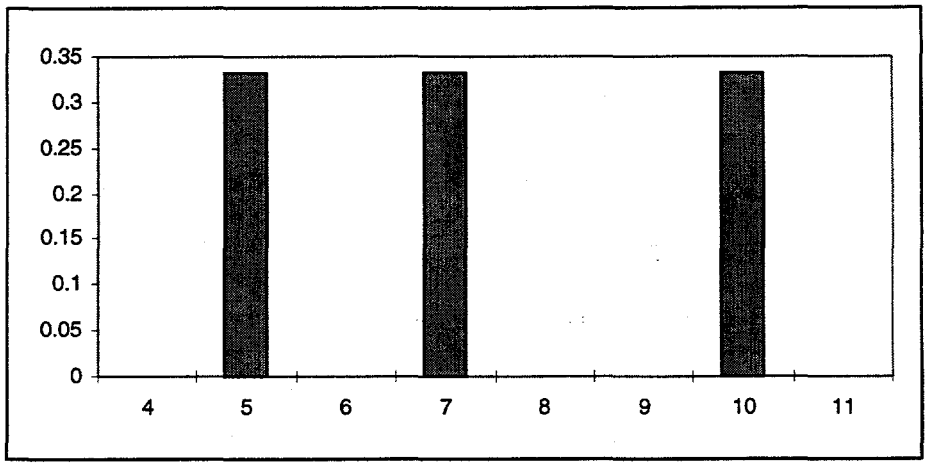


Figure 4-29: Discrete Histogram (Discrete Continuous) PDF

4.4 Keywords Not Related to Distributions

The purpose of the distribution input file is to provide a place for the analyst to specify all data related to the random variables to be sampled by LHS. Thus the vast majority of the keywords in a typical distribution input file are random variable names and their distribution description information. However, there are three additional keywords that can appear in this file. These keywords enable the analyst to insert a nonsampled variable name into the LHS sampled data output file, to specify alias names for random variables, and to initiate a nonzero rank correlation between sampled random variables. These three miscellaneous keywords are described in this section.

4.4.1 Constants

As originally structured, LHS produced a sampled output file that contained only those random variables that were sampled by the LHS program. Thus, if a user wished to use some point estimate values in a model while sampling other model parameters, he or she would be required to place the point estimate data in a separate file and provide a method for both the point estimate data and the sampled data to be read by the model program. Alternatively, the user could build a translation program that reads the LHS output file and the point estimate data file, merges them into a single entity, and writes the data in a format familiar to the model program. Both of these options are unnecessarily cumbersome. For this reason, the CONSTANT distribution type was added to LHS. Recall that LHS always automatically generates a point estimate value for each distribution sampled. Depending on the sampling options selected, this point estimate value will be the mean of

the samples generated, the median of the samples, or the user-specified value that precedes the distribution description. This point estimate value is placed in the LHS output file in order to enable programs that receive LHS results to determine at run time whether to use the sampled results or the point estimate value on a random variable-by-random variable basis. The CONSTANT distribution type simply enables the user to make additional entries in the point estimates (constant values) section of the LHS output file without placing additional values in the sampled data section or requiring LHS to perform additional sampling or pairing operations. It enables the user to maintain all program input data in a single file and minimize data file quality assurance requirements—whether the data are sampled or point value in nature.

CONSTANT ***val***

The keyword CONSTANT places the distribution name and point estimate value *val* in the constant block section of the LHS output file.

```
First Example:  NewOne 150.0 CONSTANT 152.5
Second Example:  Pi 3.14159 CONSTANT  3.14159
```

The first example places the variable named "NewOne" into the constant block section of the LHS output file. If the user has requested that LHS compute the point estimate values for sampled distributions as either the mean or the median of the sampled data, the value associated with NewOne in the LHS output file will be 152.5. Otherwise, if the user has requested that LHS use the specified point estimate value, the value associated with NewOne in the LHS output file will be 150.0. While this example demonstrates the flexibility that can be used in specifying LHS input, it does not represent a typical application of the CONSTANT keyword. One usually specifies the same value for both the user-specified point estimate and the CONSTANT distribution's *val* parameter, as shown in the second example.

4.4.2 Distribution Aliases

It is not unusual in certain risk assessment applications to specify that two variables are "totally correlated." This might occur, for example, when the same valve model from the same manufacturer has been used in two different locations in a plant system. There is no reason to believe that these two seemingly identical valves will have different failure probabilities, but there is uncertainty as to exactly what that failure probability should be. Thus, the probability of valve failure is sampled from one distribution, but given two names—one for each physical valve—so that the risk model can readily identify failure probabilities for both components.

The keyword SAME AS is used to implement this condition. This keyword is used by first defining a random variable that defines the distribution to be sampled for one of the totally correlated variable names. The second totally correlated variable is then linked to the first using the SAME AS keyword so that its name becomes a valid alias for the same distribution samples. The

LHS sampled data output file will contain only one set of values, and the file will contain a notation that indicates that this single set of values is to be used for both totally correlated variables.

SAME AS ***old_variable***

The keyword SAME AS defines a new name (alias) that is to be associated with an existing distribution. The *old_variable* parameter specifies the name of the existing random variable that is to be referred to by an additional new name. Note that *old_variable* must be the name of a random variable that is explicitly defined with a distribution description. The value of *old_variable* cannot point to the name of another variable that is defined using the SAME AS keyword.

Example:
```
First    0.0  NORMAL  0.0  1.0
Second   SAME AS  First
Third    SAME AS  First     $  This input is valid
$ Fourth SAME AS   Second   $  This is not valid
```

In this example, the random variable named "First" is explicitly defined as a normal distribution, and the random variables named "Second" and "Third" are set up to be totally correlated to First. In other words, Second and Third are simply different names (aliases) for the observations generated when the distribution First is sampled. The input for the variable named "Fourth" would be invalid because its *old_variable* parameter points to the variable Second, which was itself defined using the SAME AS keyword, instead of correctly pointing to the variable First as the object of its alias.

4.4.3 Controlling Correlation

Chapter 2 described how the LHS software, through the use of restricted pairing techniques, can intentionally pair samples from random variables in such a manner as to control correlation between those variables. By default, LHS assumes that the user desires the pairwise correlation between all pairs of random variables to approach zero (i.e., all random variables should be independent of one another to the degree possible). If the user wishes to specify other pairwise correlations between random variables, that is done using the CORRELATE keyword.

CORRELATE ***first_variable second_variable corr***

This keyword causes LHS to attempt to create a correlation of *corr* between the ranks of the sampled values for *first_variable* and the ranks of the sampled values for *second_variable*. The parameters *first_variable* and *second_variable* are the names of random variables that are explicitly defined with a distribution description and *not* using the SAME AS keyword. The specified correlation value parameter *corr* must satisfy the condition ($-1 < corr < 1$). This statement will be evaluated by LHS to ensure that the data item names are legitimate prior to LHS processing.

Example:

```
X  Normal 0.0  1.0
Y  Uniform  1.7  3.5
Z  SAME AS  Y
CORRELATE  X  Y  0.5
$ CORRELATE  Y  X  0.5   $ Identical to above
$ CORRELATE  X  Z  0.7   $ This is not valid
```

In this example, the first Correlate statement will cause LHS to attempt to establish a rank correlation of 0.5 between the random variables *X* and *Y*. Note that the order in which the random variable names appear in the Correlate statement does not matter, so the second Correlate statement (seen in comments in this example) would, if used, produce results identical to the first Correlate statement.

The third Correlate statement (also seen in comments in the example) would not be valid because it involves the random variable *Z*, which is defined using a SAME AS statement. It could be corrected by replacing the variable name *Z* with the variable name Y so that it no longer points to a distribution that is defined using the SAME AS keyword.

There are two important points to remember about the use of correlation within LHS. First, LHS only considers CORRELATE statements when it operates in the restricted pairing mode. This is because the random pairing process is just that – random – and cannot control correlation between random variables. There are two ways that LHS can be forced into a random pairing mode: the user can (1) select Random Pairing in the user interface (see Chapter 3); or (2) set up a problem in which LHS is told to generate fewer observations than the number of random variables that are being sampled. In this second condition, LHS cannot accomplish restricted pairing because the mathematical algorithm that performs restricted pairing fails for these conditions. When the user attempts to run such a problem (more random variables than observations) with restricted pairing, LHS generates a warning message in its message output file to tell the user that restricted pairing could not be used, and a modified form of random pairing is used instead (see Section 3.3.5). This warning message, however, can be easily missed. Therefore it is important to bear this situation in mind when running problems with either a large number of random variables or a small number of observations. For best results, the user should ensure that the number of observations requested is at least $^4/_3$ the number of random variables being sampled (not including CONSTANT and SAME AS keywords).

The second point to remember about the CORRELATE keyword is that it truly specifies *pairwise* correlations between random variables. Thus, if more than two random variables are to be correlated with one another, the user must enter a CORRELATE record for *every pairwise combination* of variables to be so correlated. Consider four random variables named A, B, C and D that are all to be correlated with one another, with all pairs having a pairwise correlation coefficient of 0.2. This is accomplished using the following sample input:

Example:

```
A  Normal   0.0   1.0
B  Uniform  0.5   2.5
C  Lognormal  0.1  3.0
D  Uniform   -100.0  100.0
$
Correlate  A  B  0.2
Correlate  A  C  0.2
Correlate  A  D  0.2
Correlate  B  C  0.2
Correlate  B  D  0.2
Correlate  C  D  0.2
```

The user must be extra careful when attempting to induce negative correlation between more than two random variables. There are mathematical limitations to the ways that random variables can be correlated with one another. For example, the following three-way correlation is statistically impossible:

Example:

```
Correlate  A  B  -0.95
Correlate  A  C  -0.95
Correlate  B  C  -0.95
```

The first two statements imply that large values of A tend to be paired with small values of both B and C, while the last statement implies that small values of B tend to be paired with large values of C. This condition cannot be realized by any real sampling scheme. When the LHS software encounters such a condition, it generates a warning and makes the smallest adjustments possible to the requested correlations so that it can generate its results. However, such conditions generally indicate that the user has either made a mistake in specifying input to the software or does not thoroughly understand the correlation conditions that he or she is attempting to model since contradictory correlation information has been entered. It may be that the user actually intended that large values of A should be paired with small values of both B and C, so that B and C should in fact have a strong positive pairwise correlation, such as:

Example:

```
Correlate  A  B  -0.95
Correlate  A  C  -0.95
Correlate  B  C   0.95
```

When the software detects physically unrealizable correlation conditions, it is important for the user to step back and reevaluate the software input to ensure that these contradictions are eliminated.

This page intentionally left blank

5 Viewing Results with CRAVE

CRAVE is an interactive scientific plotting program that runs in the Microsoft Windows graphical environment.[1] The program was written to support the plotting needs of the risk assessment community by plotting data written in the LHS output data format. CRAVE also plots data files from the CONTAIN code written in PVEC data format.

5.1 The Main Screen

When you invoke CRAVE, the Main screen appears, shown in Figure 5-1. The Main screen consists of the menu bar, the speed bar, and the project window, which contains the curves, title, and legend.

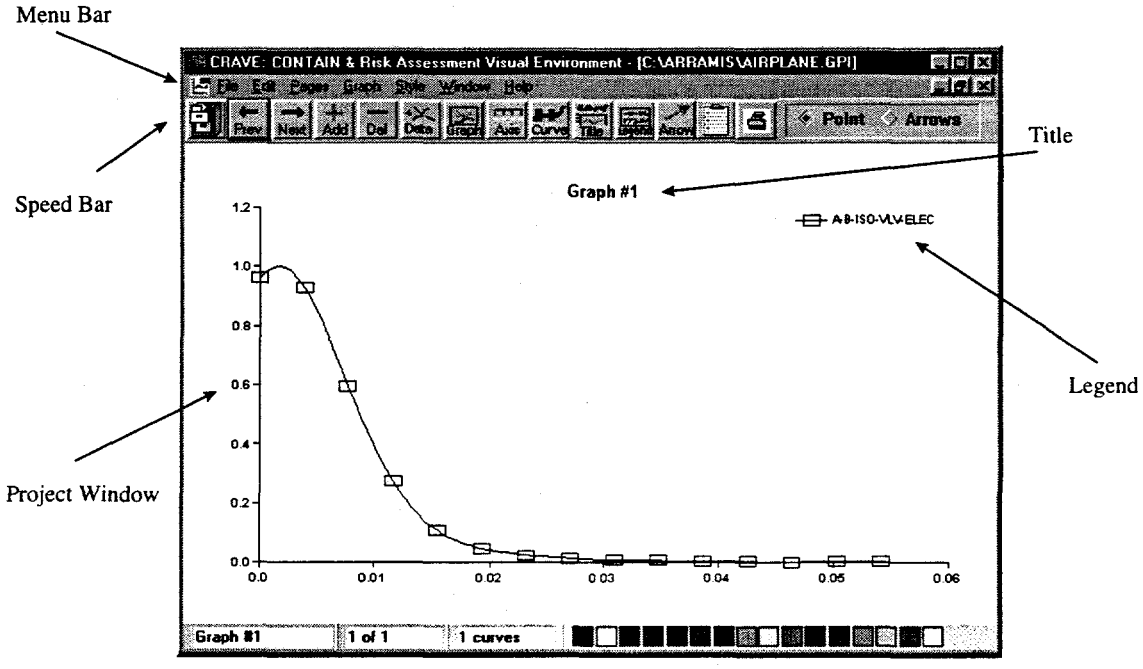


Figure 5-1: The Main Screen of CRAVE

5.1.1 Project Windows

CRAVE is a multiple document interface (MDI) program. Multiple documents may be opened at any given time, where a document is defined as a graphics project window for this application. Documents are referred to as "projects" in this document. The graphs in each project are shown in their own window, and more than one window can be shown on screen at once through standard Windows conventions provided by the MDI interface.

[1] The bulk of the text in this chapter is based on the help files written by Sandia's Ken Washington for the CRAVE graphics package.

Under the MDI architecture, only one window is active at any given time, and this window is said to have the "focus." This means that all mouse actions apply to this window and that any manipulations also apply to the graphs in this window. A window is given the focus when the mouse is used to click anywhere within the desired window. A window can also be made active and given the focus by selecting it from the list on the Window menu option.

Each project window can have several graph pages (see Section 5.1.2). Each graph page will in turn have one or more curves. When a new window is made active, the pages included in that window will be listed on the Pages menu. The desired page can be selected from this menu, or the left and right arrows on the speed bar can be used to scan through the pages one at a time.

Each project window can be minimized into an icon or can be maximized to fill the main window column. This is accomplished by clicking on the minimize button or maximize button in the upper left-hand corner of the window. Graphic project files are related to windows and each window can be saved in a separate file.

5.1.2 Graph Pages

Each project window will normally consist of more than one graph, with each graph kept on a separate page. Note that graphs that are related to each other should be kept in the same project on different pages. This is preferable to creating multiple projects with only one page per project.

When a new graph project is created, the first page with no curves is automatically created and displayed. You can add new pages to the graph project by selecting Page | Add Page on the menu, or by clicking the Add speed bar item. The Ins key can also be used to add a page. When a page is added, a default title based on its location in the project will be created and it will be displayed. Page titles can be changed at any time. Pages can be deleted by selecting the Del page item from the Pages menu, or by clicking the Del button on the speed bar.

The status bar changes automatically to reflect the page being displayed and the number of curves in the active page. You can add curves to the page by selecting the Data Load item from the Graph menu. Note that each page can have more than one curve.

You can select the desired graph page from the Pages menu or by clicking the Prev and Next speed bar arrows.

The Pages menu shows the names of all graph pages in the project. The name of each graph page is taken from the second title line of the graph. The second title line (and, hence, the graph name) can be changed in the Overall Graph Settings dialog that is invoked from the Graph menu or by clicking on the Graph speed bar button.

5.1.3 Speed Bar

The speed bar, shown in Figure 5-2, contains shortcuts for commonly used items within CRAVE.

Figure 5-2: Speed Bar

The first button, shown with a filing cabinet on it, is used for loading a project into a new window. Pressing this button is equivalent to selecting File | Load Window from the menu (see Section 5.1.4).

The next two buttons, Prev and Next, move you to either the previous page or the next page in the current window.

The next two buttons, Ad and Del, either add a page or remove a page from the current project. Pressing one of these buttons is equivalent to selecting Pages | Add Page or Pages | Del Page from the menu, respectively.

The Data button allows you to select data to include in the current graph. Pushing this button is equivalent to selecting File | Data Load from the menu (see Section 5.1.4).

The Graph button allows you to edit the overall features of the graph. Pushing this button is equivalent to selecting Graph | Overall from the menu to invoke the Overall Graph Settings dialog (see Section 5.3.1).

The Axis button will invoke the Axis Specification dialog. Pushing this button is equivalent to selecting Graph | Axes from the menu (see Section 5.3.2).

The Curve button will invoke the Define Curve dialog. Pushing this button is equivalent to selecting Graph | Curves from the menu (see Section 5.3.3).

The Title button will invoke the Graph Title dialog. Pushing this button is equivalent to selecting Graph | Title from the menu (see Section 5.3.6).

The Legend button will invoke the Legend Options dialog. Pushing this button is equivalent to selecting Graph | Legend from the menu (see Section 5.3.5).

The Arrow button will invoke the Arrow Definition dialog. Pressing this button is equivalent to selecting Graph | Arrow from the menu (see Section 5.3.4).

The next button, shown with a clipboard drawn on it, will copy the current selection to the clipboard. Pressing this button is equivalent to selecting Edit | Copy from the menu.

The last button, shown with a printer on it, will invoke the Microsoft Windows standard Print dialog. Pressing this button is equivalent to selecting File | Print from the menu.

5.1.4 File Management

Each graph project can be saved and loaded later. Project files are saved by selecting the Save or Save As commands under the File menu. When Save As is selected, the file can be renamed. When the file is saved for the first time, both menu options are equivalent since the project does not yet have a name. The project will be named once it is saved.

Projects are loaded by selecting File | Load from the menu. Alternatively, you can click on the first button of the speed bar to load a project. *Important: Loading a project is different than loading data into a graph for plotting.* Data are loaded for plotting by selecting File | Data Load from the menu or by pressing the corresponding Data button on the speed bar.

When a project is saved, all graph pages in the project are saved in the file. Other projects (or windows) are not affected by the save. The project file can be given any file name acceptable to DOS; however, the extension need not be provided since the GPI extension will be added automatically. The standard Windows file Save and Load dialog boxes that are invoked can be used to navigate the available file space, including any disk and directory. The graphics project file can be saved to any location as long as sufficient room is available on the disk.

When the file is saved, the names and locations of the data files and the data itself are saved in the file. *If the source data are later changed on the disk, then the data reload capability from the Define Curve dialog must be used to update the curves in a saved project.*

5.1.5 Printing

Graphs can be printed to any printer supported by the Microsoft Windows environment. The printer is selected from the Printer Setup item on the main menu. Alternatively, you can select the Print item on the main menu to invoke the print dialog box. The printer can be selected from this dialog box, or the Printer Setup menu option can be used to select the active printer.

Printing is keyed to a graph project window. When the print dialog box is invoked, all pages in the graph project window will be listed in the list box. The graphs that are desired for printing are simply selected from this list box. You may select multiple graphs by clicking on them with the mouse while pressing the Control key. These names are the same names that appear on the Pages menu item for the graph pages in the project window. The All button can be clicked to select all graphs, or the Current button can be used to select the graph on the currently active page.

The option button on the left side of this dialog box is used to select the desired printing layout. The last layout selected and the active printer is saved to the win.ini file under the [CRAVE] section.

When the OK button is pressed, you will be asked if you want to print the selection. If you say "no," the dialog box will close with your layout selection saved. If you say "yes," the graphs will print and the dialog box will close with your layout selection saved. The Print button will immediately initiate the print job and the Dialog box will not close. This is useful for quickly printing different orientation graphs. The Dialog box will immediately close and your layout selection will be discarded if you select the Cancel button.

Complex graphs can take a long time to print. If you are using the print manager, this can also cause large temporary files to be created on your hard disk.

5.2 Loading Data

Data are loaded into a plot by selecting the File | Data Load menu item or by clicking the Data button on the speed bar. This will most likely be the first thing you need to do to generate a new plot when you start the program. This will open a dialog box in which the files that contain the data to be plotted can be identified. If there is no active file, then the dialog box will automatically invoke the Curves Data Selection dialog (shown in Figure 5-3) in which the file containing the data to be plotted can be identified. The New File button in the center of the dialog box can also be used to select a new file from which additional data can be selected at any time.

When a file has been selected, the names of the available datasets in the file are automatically read and shown in the column on the left. These datasets must be in the LHS format. If the file is not in the LHS output format, the list box showing the available datasets will be empty.

The following paragraphs explain various options on the Curves Data Selection dialog.

Add. This button adds the selected datasets to the graph page. Note that a dataset can be added more than once to a graph. This is allowed so that a dataset can be displayed on a graph in more than one format or type. The data can only be added to one graph at a time. To add the data to another graph, you have to display that graph in the project window and then invoke the data load dialog box.

All. Pressing the All button will automatically add all datasets listed on the left from the file to the graph page.

Del. This button deletes the selected datasets in the list on the right from the graph.

Clear. This button clears all datasets from the graph. Note that the graph will have no curves after this action is taken.

Use Global File. This button replaces the list on the left with the list of the datasets in the last file that was selected. This is useful if a file is to be used that was selected for a different page. Normally, each page maintains its own available Curves list.

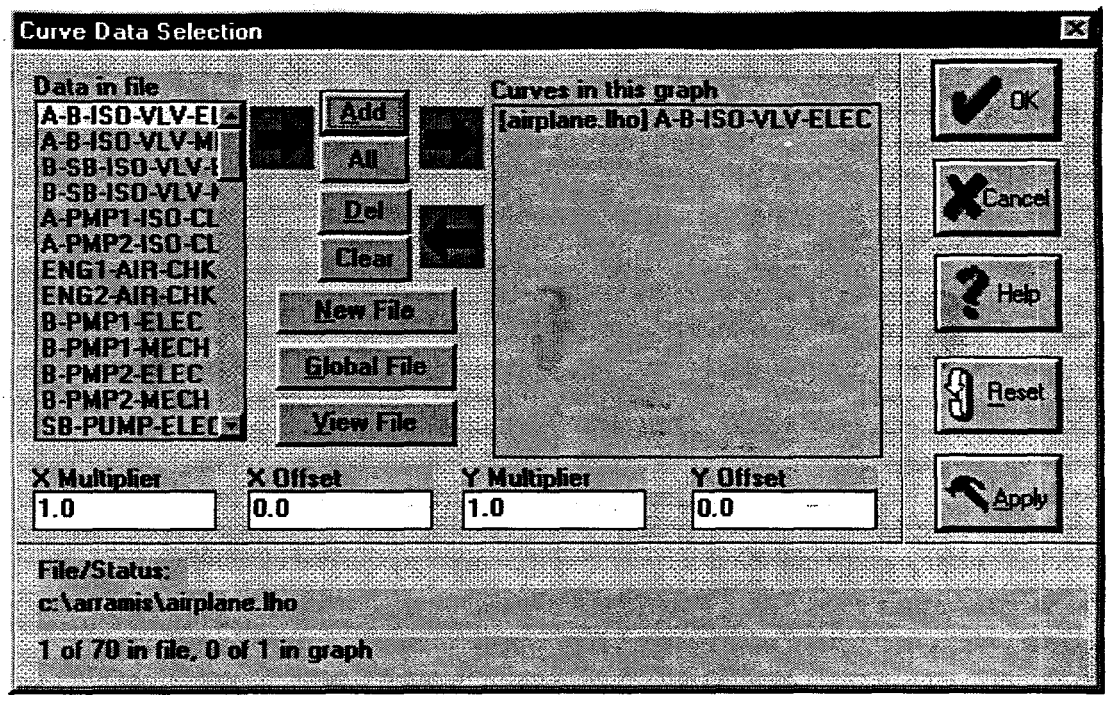


Figure 5-3: Curves Data Selection Dialog

View File. This button will launch the ARRAMIS™ file viewer so you may view the sampled data.

Reset. This button removes all of the selected datasets in the list on the right from the graph.

The results of any of the above actions may be canceled by selecting the Cancel button; however, Cancel only works if the Apply feature has not been used.

5.3 Formatting the Graph

Six options are available under the Graph menu item for formatting your graph: Overall, Axes, Curves, Arrows, Legend, and Titles.

5.3.1 Overall

Graphs are the main visual element of CRAVE. A project window can contain many graph pages, with each graph on a separate page. You can modify the appearance of the graph being displayed in the active page in the active project window by invoking the Overall Graph Settings dialog, shown in Figure 5-4.

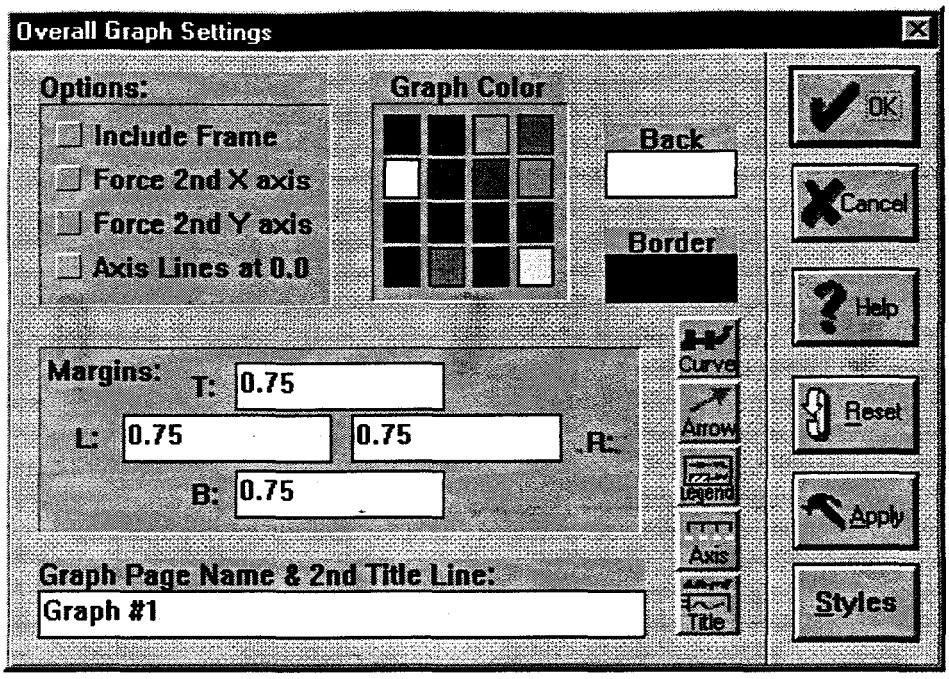


Figure 5-4: Overall Graph Settings Dialog

The following paragraphs explain various options on the Overall Graph Settings dialog.

Include Column. The graph will be framed with a thick border in the foreground color if this option is checked.

Force 2nd X axis. Normally, a secondary X axis is shown only if one or more of the curves in the graph have the Use 2nd X Axis option checked. Checking this option here will force a secondary X axis to be included in the graph even if no curves use it. The scale will match the scale of the primary X axis if no curves use the secondary X axis.

Force 2nd Y axis. Normally, a secondary Y axis is shown only if one or more of the curves in the graph have the Use 2nd Y Axis option checked. Checking this option here will force a secondary Y axis to be included in the graph even if no curves use it. The scale will match the scale of the primary Y axis if no curves use the secondary Y axis.

Axis Lines at 0.0. A copy of the axis will be included at the data coordinates (0.0, 0.0) if this option is selected. The axis at (0.0, 0.0) will include major and minor tick marks just like the primary axis. This option is most useful when the Hide Axis option on the Axis Specification screen is selected.

Color Settings. The graph can be framed in any color, and the background color can be modified using the color controls in the graph dialog box. To select a background color, click on the background square, and then click on the desired color. Apply the same procedure for the foreground color.

Margins. The margins of this graph (in inches) can be set by specifying the desired number in the indicated dialog box position. An easier way to modify the graph margin, however, is by using the mouse on the display. Note that each page has its own margin settings so that the legend can optionally be moved outside of the graph area independently for each page.

Button Selections. The buttons in this dialog match some of the ones shown on the main speed bar. The appropriate dialog box will be invoked if one of these buttons is clicked.

Style. This button will invoke the Styles dialog (shown in Figure 5-5), which will allow several characteristics, such as fonts, colors, and legend position of one graph page to be copied to other pages in the same project.

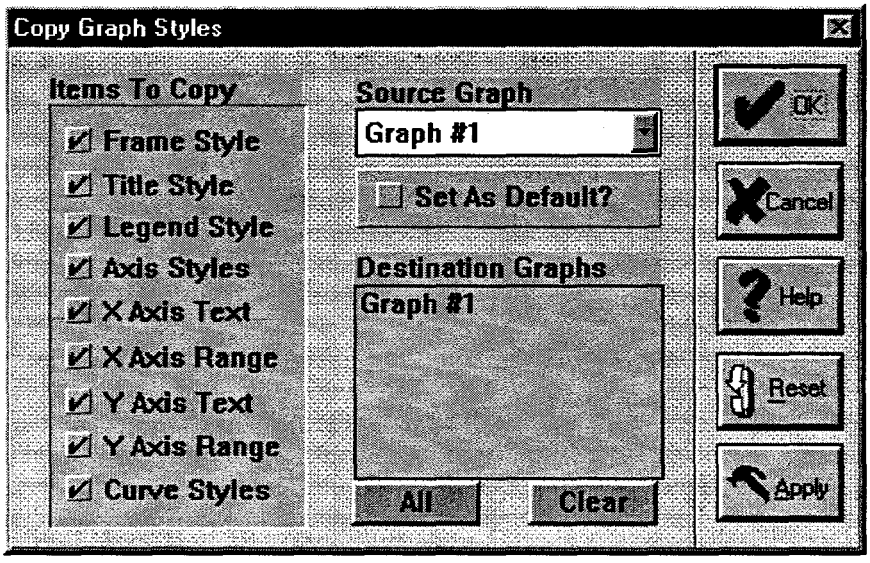


Figure 5-5: Styles Dialog

5.3.2 Axes

Each plot must have at least one X axis and one Y axis. This is normally a primary X axis and a primary Y axis; however, it is possible to have a graph that uses only secondary axes. Primary axes are automatically included if the Use 2nd Axis option is not checked in the Curve dialog (see Section 5.3.3) for any curve in the graph. This is the default. Secondary axes are automatically added if the Use 2nd Axis option is checked in the Curve dialog for any curve in the graph. A secondary axis can be manually added from the Overall Graph Settings dialog. The attributes of any one of the four available axes can be set from the Axis Specification dialog using the self-explanatory controls.

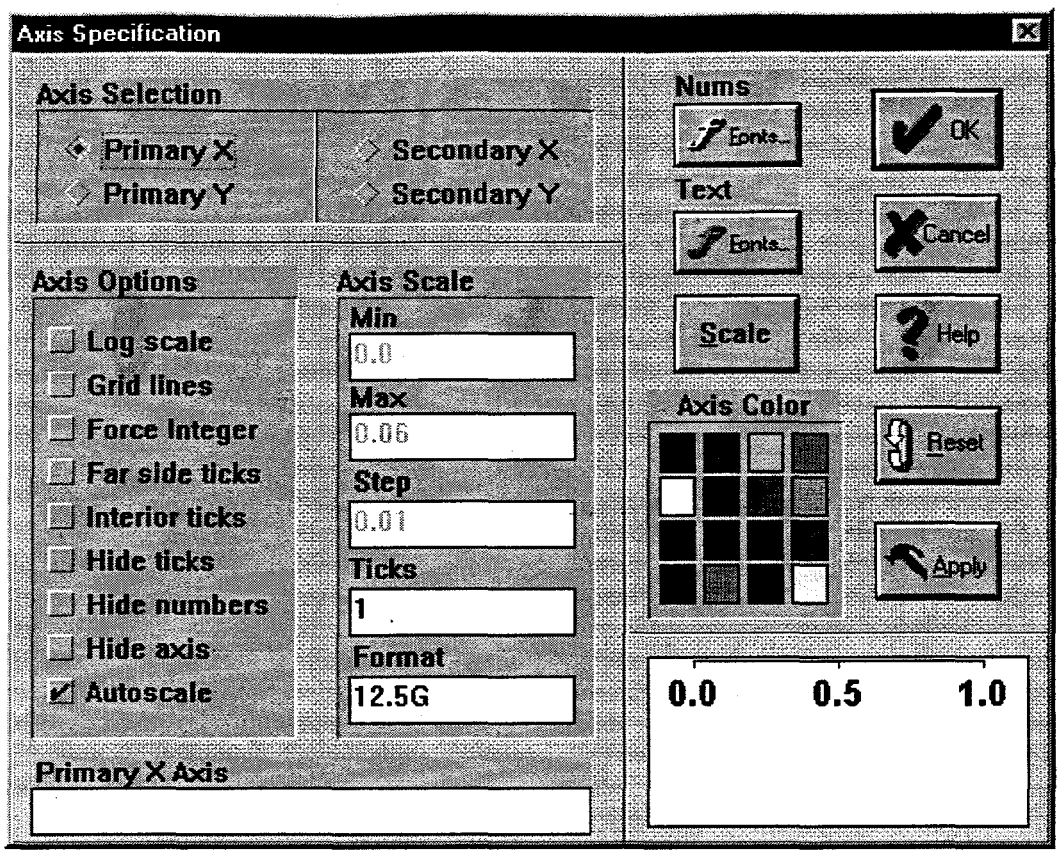


Figure 5-6: Axis Specification Dialog

The following points about this dialog may not be obvious to the new user:

- The sample does not accurately reflect all settings of the dialog box. Only fonts and colors are shown.
- Log axes may be used on data that includes zero and negative values. When this is done, the minimum axis value will be set to 0.001 times the smallest positive data value. Also, the step value is disabled when an axis is set to be log type.
- If autoscaling is set, the axis limits cannot be modified since they will be automatically set by the code. Manual scaling is best set by clicking on the Scale button. This unchecks the autoscale box and sets the stop value to a rounded value based on the minimum and maximum values. This is exactly what autoscaling does, with the difference being that the limits are determined from the data rather than from the minimum and maximum controls in the axis dialog box.
- The Hide . . . boxes can be used to create special effects, such as tick marks with no axis line. Hidden axis elements are not hidden from axes included at (0.0, 0.0) as a result of checking the Include axes at 0.0 option in the Overall Graph Settings dialog.

5.3.3 Curves

Curves are the main component of a graph page. Each graph page can have any number of curves, and each curve can be any of the available types, although there are a few restrictions on the types of curves that can be intermixed. The Define Curve dialog is shown in Figure 5-7.

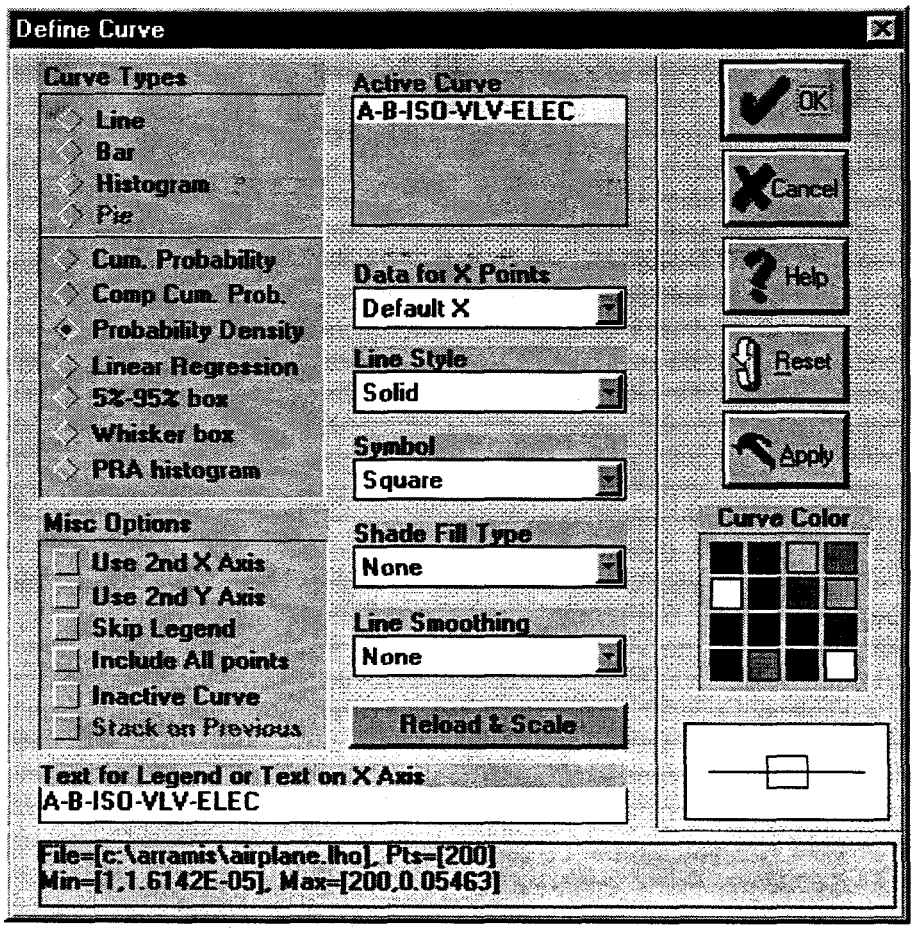


Figure 5-7: Define Curve Dialog

The following curve types are available.

General curve types applicable to any kind of data

- Line
- Bar
- Histogram
- Pie

Special curve types applicable to risk assessment sampled data

- Cumulative probability
- Complementary cumulative probability
- Probability density
- Linear regression
- 5% - 95% PRA box
- Whisker box
- PRA histogram

Any curve type may be selected with any data; however, some types will appear as nonsense when used with incompatible data types. For example, you would not want to plot temperature vs. time data using the cumulative probability type or the PRA box types. New curves from ordinary data files default to the "line" type, while new curves from LHS sampled data files default to the "cumulative probability" type.

You can also modify the curve by selecting one or more of the miscellaneous curve options shown in the lower left-hand corner of the Dialog box. Other attributes of the curve, such as line type, symbol, and color may be set from this Dialog box using the self-explanatory controls.

5.3.4 Arrows

Arrows can be used to point to data on a graph. This is done by selecting the Arrows option button on the right side of the speed bar and using the mouse to draw the arrow. When the left mouse button is lifted, the arrow will be permanently placed on the graph. The appearance of the arrow, including the arrowhead type, line color, and text font can then be set by invoking the Arrow Definition dialog, shown in Figure 5-8, and using the self-explanatory controls in this Dialog box. This Dialog box is invoked from the menu by double-clicking on any arrow or by clicking on the arrow speed bar button. To be able to double-click on an arrow, the Point option button must be selected on the right side of the speed bar.

CRAVE does not allow the user to specify the *first* arrow on a graph from the arrow definition dialog box. The first arrow *must* be drawn with the mouse using the procedure outlined above. Subsequent arrows can be added from the dialog box or drawn with the mouse.

The text controls within the arrow definition dialog box can be used to specify text that will be placed on the graph at the starting end of the arrow. The first text line will be above the line, and the second text line will be below the arrow line. The font of the arrow text will be initially set equal to the font of the legend text, but can be changed from within the arrow dialog box.

The arrow feature can also be used to place text directly on the graph (i.e., without an accompanying arrow). This is done by first creating an arrow and then, using the arrow definition dialog box, entering the desired text, and selecting the Text Only option.

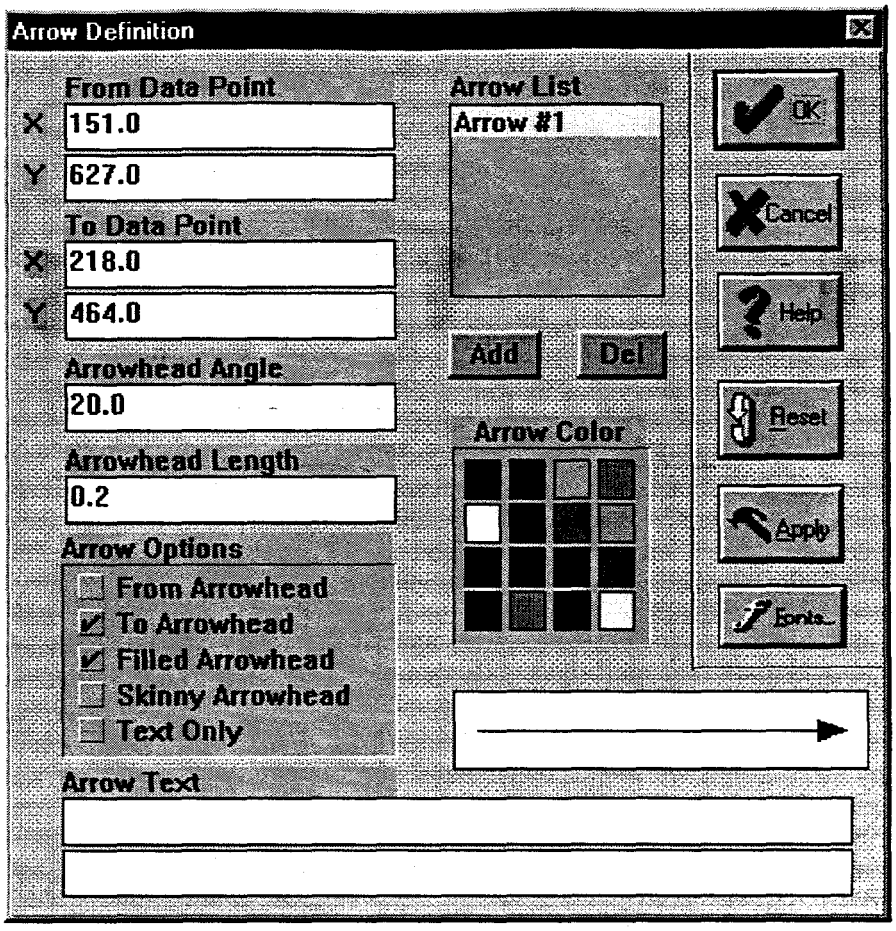


Figure 5-8: Arrow Definition Dialog

To delete an arrow, you must invoke the Arrow Definition dialog, select the arrow to delete, and press the Del button.

5.3.5 Legend

Legends can be repositioned by using the mouse or by typing in any coordinates in the Legend Options dialog, shown in Figure 5-9. You may also reposition the legend to any one of the four corners of the graph using one of the four buttons in the Legend Options dialog. The legend can also be moved outside of the graph boundaries using the mouse or as a result of specifying a given absolute position. Relative location inside the box can be specified by placing "@" in front of a fraction as the X and/or Y coordinates. For example, X Screen Position = @0.3 will place the left edge of the legend 30% from the left edge of the plot. If @ is left off, then the legend position is given in logical coordinates. To change the logical coordinates, select File | Preferences to invoke the Global Options dialog, shown in Figure 5-10. Here the maximum coordinate can be specified.

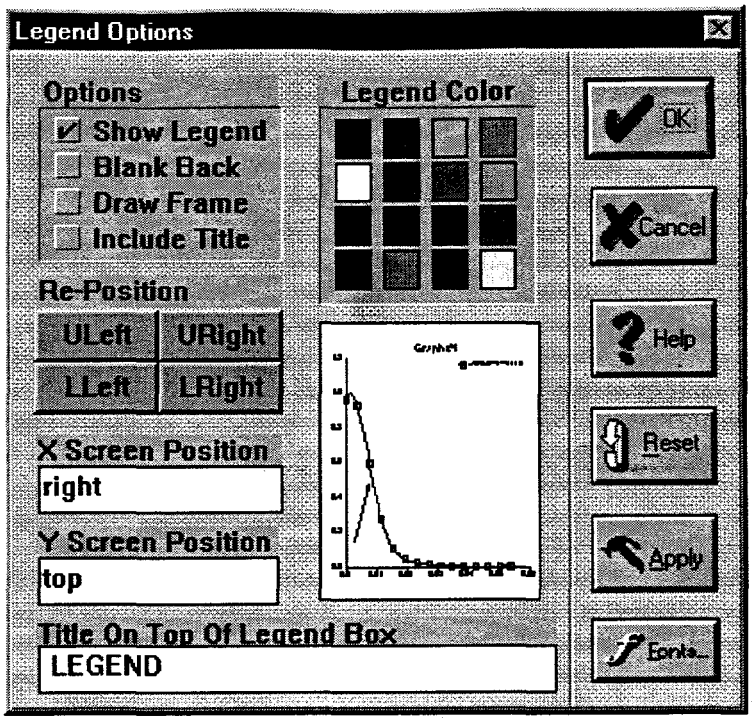


Figure 5-9: Legend Options Dialog

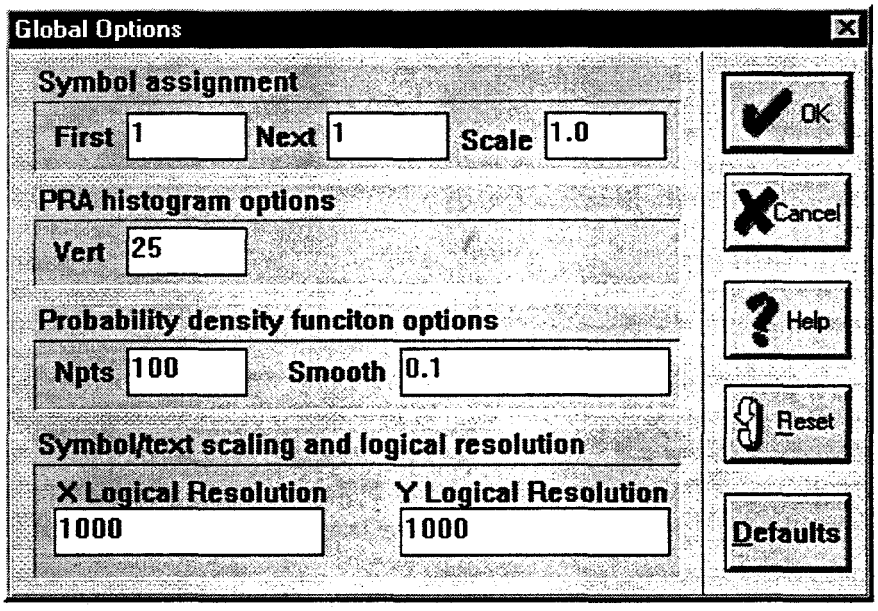


Figure 5-10: Global Options Dialog

By default the logical coordinate grid is 1000 × 1000 in size. The easiest way to position the legend is to use the mouse; when this is done, CRAVE automatically updates the appropriate logical coordinates in the legend dialog box.

An important concept of legends in CRAVE is that the names of the curves are set in the curve dialog box, not in the legend box. Therefore, to modify the actual contents of the legend text, you must change the text in the Define Curve dialog.

Legends for PRA boxes, Whisker boxes, and PRA histograms are automatically included when a curve of that type is present. This can be suppressed by checking the Skip Legend checkbox in the Define Curve dialog for all PRA curves. The legend box can be suppressed altogether by clearing the Show Legend check in the Legend Options dialog.

You can set visual attributes of the legend, such as font and color, by using the self-explanatory controls in the Legend Options dialog.

5.3.6 Titles

Each graph has a title that can consist of one or two lines. The first title line is initially blank. The second title is initially set to the default graph page name. You can change the first title line of existing graphs by typing in a revised text in the Graph Title dialog, shown in Figure 5-11, or by selecting the Style | Main Title menu option. If this is done, then the first title line of all graphs will be set to the entered text.

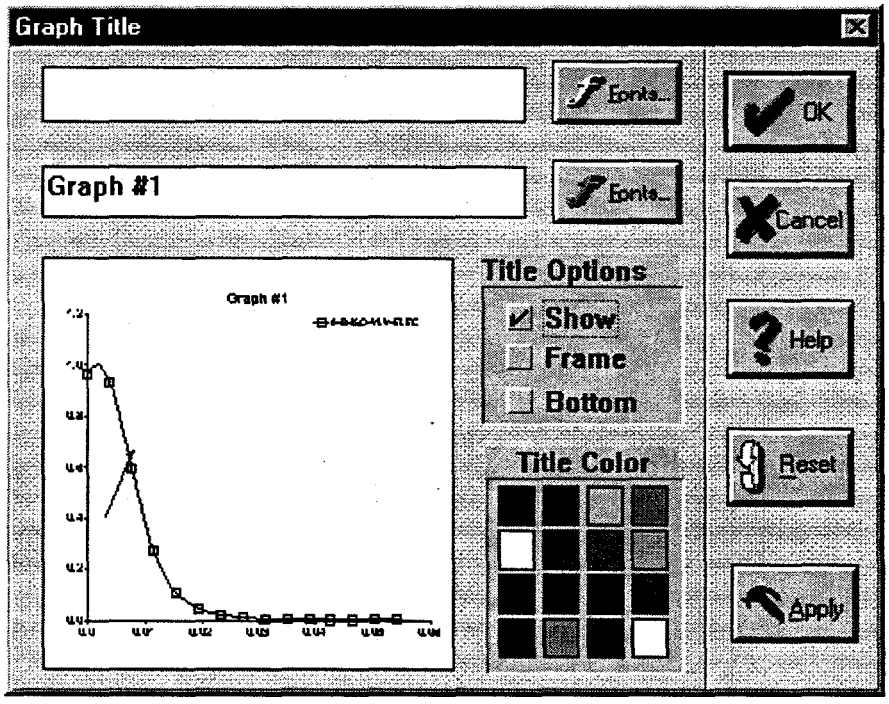


Figure 5-11: Graph Title Dialog

The graph page name is always set equal to the second title line; therefore the second title line can be set in either the Graph Title dialog or in the Overall Graph Settings dialog. This name is also used to identify the graphs for printing purposes.

The following paragraphs explain various options on the Graph Title dialog.

Show. This option is normally checked. Unchecking this option will hide the title.

Columnd. A box will be drawn around the title if this option is selected.

Bottom. The title will be shown on the bottom of the graph aligned with the left edge of the primary Y axis if this option is selected. If this option is not selected, then the title will be centered on the top of the graph.

Fonts and colors of the titles may be selected from the title dialog box using the self-explanatory controls.

5.4 Mouse Actions

As in other Microsoft Windows applications, the mouse is used to select items on the main menu, to activate graph project windows, and to select items on the speed bar.

The mouse is also used for the following interactive tasks.

Graphs. If nothing inside the graph is being pointed to, double clicking will call up the Overall Graph Settings dialog. When the graph is selected, any one of the four handles may be dragged with the mouse to modify the margins. Note that each graph page has its own set of margins so this mouse action only affects the margins of the page being viewed. You can also move the graph around on the page with the mouse by holding the left button down while pointing at the middle.

Curves. The Define Curve dialog will be called up when one of the curves is double-clicked with the mouse. Curves may not be moved on the graph with the mouse.

Axes. The Axis Specification dialog will be called up when any of the four axes are double-clicked. The selected axis will be active within the dialog box, but any of the other axes may also be changed. The axes may not be moved with the mouse; however, they will move along with the graph when the margins are changed as described above.

Titles. The Graph Title dialog will be invoked when the title is double clicked. The title may not be moved with the mouse; however, titles will move with the graph as margins and locations are changed using the method described above.

Legend. The legend can be moved by dragging it to the desired location. The legend may be moved freely inside and outside of the graph boundaries without affecting the graph. Note that unlike axes and titles, legends move independently from the graph margins. Thus the legend can be moved to avoid overlapping important data on the graph. The Legend Options dialog will be called up when the legend box is double clicked.

Arrows. If the arrows option button to the right of the speed bar is selected, the left mouse button will be used to draw arrows on the graph. Arrows on the graph may be double-clicked to invoke the Arrow Definition dialog. The first arrow on the plot must be drawn with the mouse, but subsequent arrows can be manually added in the dialog box. If the point option button is selected, the mouse may be used to move, resize, and redirect an arrow.

Multiple Selections and Colors. In addition to the above mouse tasks, one or more items on the graph may also be selected with the mouse. Multiple items are selected by holding down the Shift key while clicking on the desired items. Selected items will be shown with small graph handles (boxes) on the edges. Once selected, one of the colors may be selected in the status bar and all the selected items will turn that color.

6 Summary

6.1 Summary of LHS Use

The objective of this document has been to describe both the theory and use of the Latin hypercube sampling software in the hope that a prospective user might become comfortable not only with the mechanics of the program's use, but also with the computations that the software performs and the results that it generates.

The LHS software currently operates in a variety of modes. The most commonly used (and the most user friendly) mode of operation involves specifying distributions and sampling parameters through a Windows-based user interface. This interface provides methods for the analyst to specify distributions, correlations, and equations, to control sampling, and to specify file names and reporting options. The interface software performs basic consistency checks on the input data before passing it to the LHS DLL for sampling. The operation of this interface was described in Chapter 3, and the distribution input requirements were described in Chapter 4.

The second mode of operation is based on the software's integration with parts of the ARRAMIS™ risk and reliability analysis software (e.g., the SETAC event tree analysis package). SETAC uses a less advanced Windows user interface and enables the analyst to intermingle LHS input with the event tree definition. In this mode, the user must type actual lines of LHS input to describe distributions, correlations, and equations, but the Windows user interface generates the program control keywords automatically. In this mode, the interface checks only the program control keywords for validity before invoking the LHS DLL to perform sampling. Any errors in the distribution, correlation, or equation input are detected only within the LHS DLL. The requirements for generating input in this mode were described in Chapter 4.

The third mode of operation for LHS is based on a traditional "stand-alone execution" of the LHS software. This mode of operation is provided primarily for analysts who wish to use LHS on computers that do not support the Microsoft Windows operating environment (e.g., workstations and mainframe computers). This mode bypasses all user interfaces and requires the analyst to manually generate not only the distribution, correlation, and equation input, but also the program control keyword input. The requirements for the program keyword control input are described in Appendix A, and the method for program execution is described in Appendix B.

The fourth and final mode of operation for LHS is a backward compatibility mode. It is intended to allow persons who used the original LHS software to continue to use their existing input files with only minimal modification as they transition to the upgraded software. A few incompatibilities were introduced during the upgrade process (specifically to the NORMAL and LOGNORMAL distribution types). The changes that the user will be required to make to his or her input in order to ensure valid results while running old input files using the upgraded LHS software are described in Appendix D.

While LHS has been configured to run in these various modes for the sake of flexibility and backward compatibility, it is expected that most analysts will use the software based on the Windows user interface. An added advantage of this mode is that it provides direct access to the CRAVE graphics package for graphical output of the LHS results. The use of CRAVE is described in Chapter 5.

6.2 Summary of LHS Capabilities

LHS has been a powerful tool for sampling statistical distributions in uncertainty analyses for more than 15 years. Key to its success has been its flexibility: the ability to perform traditional Monte Carlo uncertainty analyses and more advanced Latin hypercube sampling, and to perform both using either random or restricted pairing techniques. The upgraded capabilities that are documented in this report, such as the added distribution types, the computation of point estimate values, and the ability to identify individual random variables by user-defined names, should enable LHS to remain a viable uncertainty analysis tool for many years to come.

7 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, NM.

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, NM.

Iman, R.L., and Shortencarier, M.J. (1984). "A Fortran 77 Program and User's Guide for the Generation of Latin Hypercube and Random Samples for Use with Computer Models," NUREG/CR-3624, Technical Report SAND83-2365, Sandia National Laboratories, Albuquerque, NM.

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

Iman, R.L., and Helton, J.C. (1985). "A Comparison of Uncertainty and Sensitivity Analysis Techniques for Computer Models." Technical Report SAND84-1461, Sandia National Laboratories, Albuquerque, NM.

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, 221, 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, NM.

Appendix A Keyword Input Description and File Structure

The LHS program requires certain parameters to be defined in order to perform its sampling calculations. The program recognizes 55 keywords that dictate the characteristics of the generated sample such as size and type of sample, number of samples desired, correlation structure for input variables, and type of distribution specified on each variable. Other keywords are used to control the program output.

The program input may be placed in a single file, or the global program commands and distribution-specific commands may be placed in separate files. However, all input files require the same basic input format. The format of the distribution-specific input commands was described in Chapter 4. An analyst who is using the Windows-based LHS user interface will find the information in that chapter sufficient for constructing distribution input for LHS because the remaining keywords and specifications are handled automatically by the Windows-based user interface. The purpose of this appendix is to provide a reference for the user of the stand-alone LHS Fortran package regarding those keywords and file structures that must be created manually in order for LHS to function in the absence of the Windows-based user interface. Note that a portion of the text from Chapter 4 is reproduced here in order to give a complete picture of the input file requirements in a single place.

A.1 Input File Characteristics

All files read by LHS are based on a common file format that provides the user with a great deal of latitude in generating easily read, self-documenting input. Each LHS input file is a space-delimited free-format ASCII text file. As the file is read, its contents are treated as case-insensitive. The format supports continuation lines as well as whole-line and trailing comments.

A.1.1 Keyword File Characteristics

LHS assumes that each input line will contain no more than 80 ASCII characters. If longer input lines are used, the code will ignore all input that is not contained within the first 80 columns. Each line is assumed to contain either a whole-line comment or single command (possibly followed by a trailing comment). Each command in the keyword file must be fully contained on a single line. Note that this is different from the distribution input file in that continuation lines are not permitted in the section of the file that is used to specify command keywords to control LHS execution.

The LHS input file format provides for both full-line and trailing comments. Under this format, all blank lines and all text following "$" characters (dollar sign) are considered comments. A full-line comment is any line that contains only blank spaces or contains a dollar sign as its first nonblank character. A trailing comment consists of any text that follows a dollar sign on any line in the file. In a trailing comment, the dollar sign must be preceded by one or more blank spaces. Since LHS

ignores all text that follows a dollar sign, it is not possible to place comments between items on a single line.

A command may begin and end at any point on the line (an arbitrary number of leading and trailing spaces may be used as desired). If a command keyword is composed of more than one word, those words must be separated by *one and only one blank space*. However, the user may include as many blank spaces as desired between a keyword and any required alphabetic or numeric values. For example,

```
LHSOPTS      Random Sampling
```

is a legal input line because the keywords LHSOPTS and Random Sampling may be preceded and followed by an arbitrary number of blank spaces, but only a single blank space is permitted within the keyword Random Sampling. It should be noted that LHS treats commas (",") and tab characters as being fully equivalent to and interchangeable with blank space characters.

The alphanumeric input to LHS has been designed to allow maximum user flexibility. All character input is case-insensitive, so a user can provide any combination of upper- and lower-case characters for LHS input. Thus, the keywords Normal, NORMAL, normal, and noRMaL are all considered equivalent by LHS.

All numeric input is read in Fortran list-directed (free-format) input. Thus a user may specify numeric input in any Fortran-standard format, including integer (146), floating point (15.643), and exponential notation (1.426E-3).

The LHS keyword file contains the keywords that control program execution. These keywords specify, for example, the run title, the number of observations to be generated, the types of point estimate calculations to be performed, the names of the files to be used, and the types of reports to be generated. *Continuation lines cannot be used in the keyword section of this file*. The keywords that can be used in this file are described in the following section.

A.1.2 Input File Structure

LHS was designed to allow two different forms of input file structure: single-file and double-file. Single-file input simply places the distribution information and the command keyword information in a single file. It is possible to freely intersperse command keywords and distribution definitions within this single input file as long as all command keywords occur prior to the Dataset: keyword. However, the preferred method of file development bunches the command keywords near the top of the input file (with appropriate comments), followed by the Dataset: keyword and the complete set of distribution definitions. The single-file input structure is assumed whenever the command keyword PRETRIN is not used in the input file. Note that continuation lines are allowed in the distribution section of this input file.

The double-file input structure groups the command keywords in one file and the distribution definitions in a second file. This file structure was developed to aid the analyst in quality assurance

of the distribution definition information because it allows the analyst to change the parameters of the LHS run (e.g., change the number of observations requested or the output file name) without modifying the file that contains the distribution definitions. The name of the second input file (the distribution definition file) is specified using the PRETRIN keyword (described later in this appendix). If the PRETRIN keyword is present, LHS will read distribution definitions *only* from this second file and will ignore any distribution definition information contained in the command keyword file.

A.2 Keywords to Control Program Execution

The following is a list of keywords that set up the execution of the program. Note that LHS is designed to be run as one element in a large suite of codes—each of which has its own set of keywords. Therefore, LHS ignores keywords that it does not recognize.

A.2.1 Sampling and Processing Control Keywords

The keywords described in this section are used to control the processing that occurs in LHS. They are used to determine the starting point of the random number sequence, the types of sampling and pairing to be performed, the number of observations to be generated, and the method to be used for computing the point estimate values for the sampled distributions.

LHSSEED ***iseed***
(Required)

The positive integer *iseed* specified with this keyword provides the "seed value" or starting point for the LHS random number generator. The value specified by this keyword must be within the range of representable positive integers ($1 \leq n \leq 2^{31} - 1$). There is anecdotal evidence to suggest that the randomness of the random number generator may be somewhat compromised for very small seed values, so it is recommended that random seed values less than approximately 1000 be avoided. The seed value is printed in the LHS message file at the beginning of each sample. If the keyword LHSREPS (described later in this section) specifies a number greater than 1, the current value of the random seed is retrieved and printed at the start of the generation of each new sample. This allows the user to regenerate by rerunning the program with the new seed and with the LHSREPS parameter omitted (or having LHSREPS 1).

Example: `LHSSEED 15964`

A run with this example would start the random number generator based on an integer seed value of 15964.

LHSOBS ***num***
(Required)

This keyword must be followed by a positive integer that specifies the number of observations that LHS will generate to be included in the sample for each distribution. The maximum number of observations is defined by a parameter in the initialization file (see Appendix E).

Example: `LHSOBS 500`

An LHS run containing this keyword would produce 500 observations for each variable.

LHSREPS ***nreps***
(Optional, default: *nreps* = 1)

This keyword can be used to generate multiple samples (replications). When present, it must be followed by a positive integer to specify the number of complete sample sets of the data (each of the size LHSOBS). The samples are then written back to back in the output file. If LHSREPS is omitted, a single sample is generated.

Example: `LHSREPS 3`

This example would produce three repetitions of the specified distribution samples, each with the size LHSOBS.

LHSPVAL ***ipval***
(Optional, default: *ipval* = 1)

This keyword allows the user to select which method LHS will use to calculate the representative point value found in the first block of the LHS sampled data output file. If this keyword is omitted, the point values printed will represent the mean. LHSPVAL, followed by an integer number *ipval*, specifies whether the point value in the LHS output file is to represent the mean (*ipval = 1*) or median (*ipval = 2*) of the distribution. If a zero is specified (*ipval = 0*), then the point value specified for the distribution in the input file is copied directly to the sample output file. Note that the "optional" distribution point values become required input if zero is specified for this keyword.

Example: `LHSPVAL 0`

This example would cause LHS to enter the user-specified point value for each distribution in the point estimates block of the LHS output file. Note that if the analyst needs to import an LHS *input* file for use in TEMAC3, then LHSPVAL *must* be set to 0. TEMAC3 can, however, use LHS output files that were created with LHSPVAL set to values other than 0.

LHSOPTS ***options***
(Optional, default: no options selected)

This keyword allows the user to control the sampling and pairing methods that LHS will use in preparing its sampled data output file. If the keyword is omitted, LHS will use Latin hypercube sampling with restricted pairing. If the keyword is present, it must be followed by one or more of the following keywords.

RANDOM SAMPLE This option causes LHS to suppress Latin hypercube sampling and use purely random (Monte Carlo) sampling techniques in its place. If this option is omitted, LHS uses Latin hypercube sampling.

RANDOM PAIRING This option causes LHS to suppress the restricted pairing method and generate a sample with purely random pairing. If this option is omitted, LHS uses restricted pairing.

```
Example:  LHSOPTS   RANDOM PAIRING
Example:  LHSOPTS   RANDOM SAMPLE
Example:  LHSOPTS   RANDOM SAMPLE   RANDOM PAIRING
```

The first example would cause LHS to produce its sample using Latin hypercube sampling techniques, and to use random pairing of variables. The second example would cause LHS to use random sampling with restricted pairing. The third example would cause LHS to use both random sampling and random pairing.

A.2.2 Output and Reporting Keywords

The keywords described in this section are used to define how LHS generates its output. Using these keywords, the user can control the title to be placed in the output file, the types of reports to be generated, and the format of the sampled data output file.

LHSTITL ***title***
(Optional, default: *title* = blank)

This keyword can be used to specify the title of each LHS run. If specified, it must be followed with alphanumeric data (up to 70 characters long) to help describe the application of the sample. This information will be printed as a one-line header on each page of the output. If it is omitted, a blank header is generated at the top of each page.

```
Example:  LHSTITL  This is a test EVNTRE/LHS run
```

An LHS run using this example LHSTITL record would have "This is a test EVNTRE/LHS run" written as a header on each page of the message output file and as a comment in the sampled data output file (if the default format is used).

LHSRPTS ***options***
(Optional, default: *options* = none)

This keyword is used to specify which reports LHS will print in the message output file. If LHSRPTS is omitted, the message file will contain only the title, run specifications, and descriptions of the distributions sampled. If LHSRPTS is included, it must be followed by one or more of the following three additional keywords. These additional keywords can appear in any order (separated by blanks). They function as follows:

CORR Print both the achieved raw correlation matrix and the achieved rank correlation matrix associated with the actual sample.

HIST Print a text-based histogram for each random variable.

DATA Print the complete set of all data samples and their ranks in the message output file. Use of this option makes the individual sample input vectors available in the message file. It should be used with caution, however, because it can cause the generation of an extremely large message output file.

Example 1: `LHSRPTS CORR HIST`
Example 2: `LHSRPTS DATA`
Example 3: `LHSRPTS HIST DATA CORR`

If Example 1 were used, the message output file would contain both raw and rank correlation matrices for the sample and a histogram for each variable. If Example 2 were specified, the message file would contain all of the sample data and rank data for every observation of every variable sampled. If Example 3 were specified, all three types of reports would be written to the message output file.

LHNONAM
(Optional, default: False)

This keyword specifies that the sampled data file will be written in the old LHS output format. In this format, all header, name and point estimate information is omitted from the sampled data output file. The file contains only the sampled data for each variable on an observation-by-observation basis. This option was provided to ensure backward compatibility of LHS with older analyses that made use of this file format. This format is not compatible with the ARRAMIS™ risk and reliability analysis software suite.

Example: `LHNONAM`

This example causes LHS to write its output in the old format that omits headers, names, and point estimate values.

LHSSCOL
(Optional, default: False)

This keyword specifies that LHS is to write its sampled data output file in a single column – that is, one value per line. This has been found helpful by certain postprocessing programs that have difficulty interpreting data files that contain more than one value per line. This format is not compatible with the ARRAMIS™ risk and reliability analysis software suite.

Example: `LHSSCOL`

This example causes LHS to generate its sampled data file in a single-column format with one value per line.

A.2.3 File Specification Keywords

LHS and its associated codes make use of several files. The keywords described in this section are used to specify the names of those files. Currently, file names should be specified in the MS DOS 8.3 file format (including, at the user's option, the full file path name, but without long file names or spaces), although enhancements to the software that will remove these restrictions are planned for the near future.

LHSOUT *filename*
(Required)

This keyword, followed by a file name, allows the user to specify the name of the LHS sampled data output file. This output file is designed to be read by other programs and is not formatted for easy examination by the analyst. The file designed for the analyst to read is the message file, which is specified by the next keyword description.

Example: `LHSOUT TestRun.LSP`
Example: `LHSOUT C:\ARRAMIS\NewProb\TestRun.LSP`

An LHS run with this example would write the sampled output to a file named "TestRun.LSP."

LHSMSG *filename*
(Required)

This keyword, followed by a file name, allows the user to specify the name of the output file that contains the LHS user output and reports ("messages") from the LHS execution.

```
Example:  LHSMSG TestRun.LMO
Example:  LHSMSG C:\ARRAMIS\NewProb\TestRun.LMO
```

An LHS run with this example would write the message and user reports to a file named "TestRun.LMO."

PRETRIN *filename*
(Optional, default: none)

The keyword PRETRIN, followed by a file name, allows the LHS program to open the specified secondary file that contains distribution description information. If this keyword is omitted, LHS will read the distribution description information from the keyword file (see Section 4.1.2).

```
Example:  PRETRIN TestRun.LDI
Example:  PRETRIN C:\ARRAMIS\NewProb\TestRun.LDI
```

This example would cause LHS to open the file TestRun.LDI to read the distribution and correlation specifications.

When the PRETRIN keyword is used, LHS assumes that all distribution definitions and correlation information will be read from the file designated by this keyword. Any random variables defined in other input files will be ignored. Within that file, each random variable must be defined either on a line that begins with the keyword Data: or within a block that begins with the keyword Dataset: (the definition of correlation between variables must follow these same rules – see Chapter 4).

LHSPOST *filename*
(Optional, default: none)

This keyword is required if the user wishes to define data in terms of a combination of other distributions using the LHS postprocessor. The keyword is followed by a file name that specifies the sample file to be generated as output by the postprocessor. As such, it is ignored by LHS, but since LHS and its postprocessor share common input files, this keyword has been included here for the sake of completeness.

```
Example:  LHSPOST TestRun.MSP
Example:  LHSPOST C:\ARRAMIS\NewProb\TestRun.MSP
```

This example would cause the postprocessor to generate a sampled data output file named TestRun.MSP

There is currently no keyword by which the analyst can specify the name of the message output file from the LHS postprocessor. The program automatically generates a message output file name that is identical to the keyword file name, but with the three character file suffix MMO. Thus, if the LHS keyword file is named "TestRun.Key", then the postprocessor message output file will be named "TestRun.MMO".

This page intentionally left blank.

Appendix B LHS Execution and Output

B.1 Running LHS

LHS is available as both a Windows-based program and as a stand-alone executable for the MS DOS operating system. Execution of the Windows-based program was described in Chapter 3. The MS DOS version of the program requires a minimum of 2 megabytes of available extended memory on an IBM-compatible personal computer with an 80386 (or higher) CPU with an 80387 (or compatible) math coprocessor.

To run LHS, type the following at the MS-DOS prompt:

LHS *filename* <enter>

where *filename* is the name of the keyword file. If the parameter *filename* is omitted, the program will prompt the user for the name of the keyword file.

B.2 The Message Output File

The message file is the output file that the user will read. It consists of three parts: the header page, an input review, and an uncertainty data block.

B.2.1 Header Page

The header page contains a listing of the time and date the program was run, which files were accessed during execution, and which options were specified. Options include random or Latin hypercube sampling, random or restricted pairing, and output options such as correlation matrices and histograms.

B.2.2 Input Review

This section is simply an annotated copy of the input to LHS, including the distributions and correlation specifications that were performed, the starting point of the random number generator, the number of random variables, and the number of observations.

B.2.3 Uncertainty Data Block

This data block contains the information specified by the LHSRPTS keyword. If DATA was specified, then there will be a listing of each sampled distribution as well as the rank information

from the sample. If CORR was specified, then the raw and rank correlation matrices associated with the actual sample generated are printed. If HIST was specified, then one text histogram will be printed for each random variable. Any combination of the above three options may be used.

B.3 The Sampled Data Output File

The file format consists of three blocks: the point estimate data block, the uncertainty header block, and the uncertainty data block. Since LHS is meant to be run in a suite of codes, this output will have characteristics similar to the input files of other programs so that other programs may easily use this output. All three blocks must be present in the file in the stated order. The point estimate data block starts at the beginning of the file and proceeds until the keyword "@UNCERTAINTY" is found as the first item on the line. This marks the beginning of the uncertainty header block. This header block ends with the keyword "@SAMPLEDATA". The uncertainty data block follows this keyword and occupies the rest of the file.

B.3.1 Common Characteristics of the Point Estimate and Uncertainty Header Blocks

1. All data fields are space-delimited. All processors should treat commas and tab characters as if they were blank spaces.

2. A data record may occupy multiple lines, but each line may have no more than 80 characters. Characters that occur in columns greater than 80 are ignored. If a data record is to be continued on the following line, it must contain a continuation character (either a "#" or a "%") as the last character prior to any trailing blanks or comments (see below). The continuation character must be preceded by one or more blank spaces.

3. All text following a "$" (dollar sign) is considered a comment. If a dollar sign is the first item on a data record, the entire record is treated as a comment. A dollar sign and comment may also follow all data items on a record or the continuation character (trailing comment). In these cases, the dollar sign must be preceded by one or more blank spaces. Comments may not come between data items on a single record (except following a continuation character on a continued line).

4. Blank lines are considered comment records and are ignored. Any number of blank lines and/or comment lines may be placed between a line and its continuation or between consecutive continuation lines.

5. All names used in the file must meet the following criteria: (a) each name must contain no more than 16 characters; (b) the "$" (dollar sign), "#" (number sign), "%" (percent sign), "," (comma), " " (blank space), and tab characters are not valid within names; and (c) a name must not form a valid representation of a real number for an ANSI-standard Fortran compiler. All names are to be treated as case-insensitive, so the names FRED, Fred, fred, and freD are all

equivalent. Names that are longer than 16 characters may be truncated to 16 characters at the discretion of the processor.

6. Where numbers are specified, they may be in any format recognized by an ANSI-standard Fortran compiler.

7. Each name must be unique within each section. Duplicate definitions in the same section for a single data item are not valid. However, an item may be defined once in each section. If an item is defined once in the point estimate block and once in the uncertainty header block, both definitions are effective. The reading program would have to decide whether its calculations should use the point estimate value, the distribution, or both.

8. A line and all of its continuation lines (with all trailing and intervening comments eliminated) must together consist of no more than 32,767 characters.

B.3.2 Point Estimate Block

This block starts the beginning of the file and proceeds until the keyword @UNCERTAINTY is found as the first item on a line. It begins with a single comment record (starting with a dollar sign) that documents the version of the LHS file format being used. The record should be exactly as follows:

```
$ LHS File Format Version 1.00
```

The record need not begin in the first column. Any program to read this record (in order to verify input file compatibility) may assume that only one blank character exists between each of the words, but should not make any assumptions regarding which characters will be in upper or lower case.

This initial comment record will be followed by several additional comment records that document the pedigree of the file, such as the name and version of the program, the data and time the program was run, and the list of files accessed by the program.

A data record in the point estimate data block consists of anything that is not discarded as a comment. Only one type of data record will be written out: a list of one or more names followed by one or two real number values. The first value represents the point value for each of the items found on the list of named data items. The second item, which is optional, is the standard deviation associated with the point estimate value. The point estimate value that appears at the end of the line is assigned to all named data items on that line (including all active continuation lines). The list is space delimited (see Section 2 above).

Any processor reading this data format should look for the following common errors: (a) a record consisting of a list of names that is not followed by a legal value is illegal; (b) a record consisting of a value without a corresponding list of names is illegal; (c) names that follow the value(s) on a record are illegal.

Any processor wishing to skip this section can simply read and discard all lines until the keyword "@UNCERTAINTY" is found as the first data item on a noncomment line (the keyword "@UNCERTAINTY" may be preceded by leading blank spaces).

B.3.3 The Uncertainty Header Block

The uncertainty header block contains two input sections that may occur in either possible order. The first section consists of a single line containing the keyword @OBSERVATIONS as its first data item, followed by a positive integer that represents the number of observations that are found in the uncertainty data block. It is critical that the value on this record accurately reflect the true number of observations present in the file. Any processor reading this data format must be sure that there are at least this number of observations present in the file. It may, however, ignore any additional observations that may be present.

The second section describes the number and names of the distributions that are found in the uncertainty data block. The section begins with a single line containing the keyword @VARIABLES as its first data item, followed by a positive integer that represents the number of distributions that are found in the uncertainty data block. It is critical that the value on this record accurately reflect the true number of variables present in the file. Any processor reading this data format must be sure that there are at least this number of distributions present in the file. It may, however, ignore any additional distributions that may be present.

The *@VARIABLES n* line must be followed by exactly *n* lines that provide names for the *n* distributions. Each of these lines begins with the primary name for the distribution, followed by a ":" (colon character). Following the colon character is an optional list of secondary names for the distribution. Thus, each of the *n* distributions is given exactly one primary name and an arbitrary number of secondary names. The list of names is space-delimited and may extend to one or more continuation lines. Comments may be present at any point in this block.

The order in which these records appear is significant. All names on the first record encountered in this section are assumed to correspond to the first distribution in the uncertainty data section of the file. The names on the second record are assumed to correspond to the second distribution, and so forth. For this reason, the number of records found in this section must correspond exactly to the number that follows the @VARIABLES keyword, and that value must exactly match the number of distributions found in the uncertainty data block. Anything else is an error condition.

The uncertainty header block is terminated by the @SAMPLEDATA keyword.

B.3.4 Uncertainty Data Block

The uncertainty data block is completely different from either of the preceding blocks. It contains only numeric data (no names). Comments lines are <u>not</u> supported in this block. Continuation lines exist only as described below. This block contains most of the numeric data generated by LHS.

It contains n_{obs} data records – one record for each observation. The data record for a file containing n_{var} variables consists of two integers followed by n_{var} real values. The first integer value contains the observation number. The second integer value contains the number of the distribution values that follow (n_{var}). These integers are followed by one value from each of the n_{var} distributions. In other words, there will be (n_{var}+2) columns of data – the first tells which observation is on that line, the second contains the number of distributions, and the next n_{var} columns each correspond to one of the distributions performed.

The records in this section are generally written and read using a Fortran list-type format in an implied "Do Loop." Thus, one record may consist of several lines without explicit continuation characters. The Fortran program will read additional lines as needed. The following hypothetical code example would read one entire uncertainty data block (all variables and all observations):

```
Do i=1, NObs
     Read (1,*) i, NVar, (Values(i,j), j=1, NVar)
End Do
```

The block is typically written to the file using similar code. Note that the use of the Fortran "*" format for reading and writing does not allow for continuation characters or comment lines. Thus, these are not allowed in this data block.

Note that this block occupies the remainder of the file. Thus, an attempt to read this file following the completion of the above loop should result in an end-of-file error condition. Figure B-1 contains a sample LHS output file.

```
$ LHS File Format Version 1.00
$
$ This LHS run was executed on  9/ 4/97 at 12:31:46.46
$ with LHS Version: 2.10 Release 2, Compiled 03/10/1997
$ The run title was:
$ LHS INPUT FOR NUCLEAR TECHNOLOGY TEST PROBLEM
$ Message output file for this run: LHS.MSG
$
$ Input file(s) for this run:
$
$ Point Values for the distributions follow:
$
$ All point values represent the optional point
$ values that were found in the input file.
$
    BA                    2.0000000E-01
    BB                    2.0000000E-01
    BC                    1.0000000E-01
    BD                    1.0000000E-01
    I1                    1.0000000E+00
    I2                    2.0000000E+00
$
@UNCERTAINTY
  @OBSERVATIONS     10
  @VARIABLES      6
    BA:
    BB:
    BC:
    BD:
    I1:
    I2:
@SAMPLEDATA
          1           6   0.174378        0.204163        0.203658
   0.712264E-01   0.175162        1.02023
          2           6   0.127138        0.135356        0.616336E-01
   0.994375E-01    5.18535       0.524736
          3           6   0.153722        0.247982        0.108648
   0.400652        1.12832        4.05337
          4           6   0.195077        0.189196        0.934008E-01
   0.280316E-01   0.809043        2.92969
          5           6   0.331527        0.151177        0.185418
   0.633564E-01    1.50779        1.50650
          6           6   0.131032        0.102713        0.117998
   0.845984E-01   0.371710        1.93410
          7           6   0.881318E-01    0.290377        0.511588E-01
   0.417372E-01   0.644589        1.72556
          8           6   0.258039        0.405209        0.679013E-01
   0.134097       0.603218       0.877367
          9           6   0.226539        0.122867        0.408741E-01
   0.145608       0.269958        1.20896
         10           6   0.285970        0.176833        0.293484E-01
   0.529115E-01   0.309393        2.55828
```

Figure B-1: Sample Output File

Appendix C Sample Problem

This appendix will take you through a sample problem by showing all stages of sampling data. A primary event list will be imported from a fault tree file.

C.1 Importing a Primary Event List

The fault tree VINDELIVER was imported by selecting File | Import from the menu. The fault tree is shown in Figure C-1; it contains nine primary events.

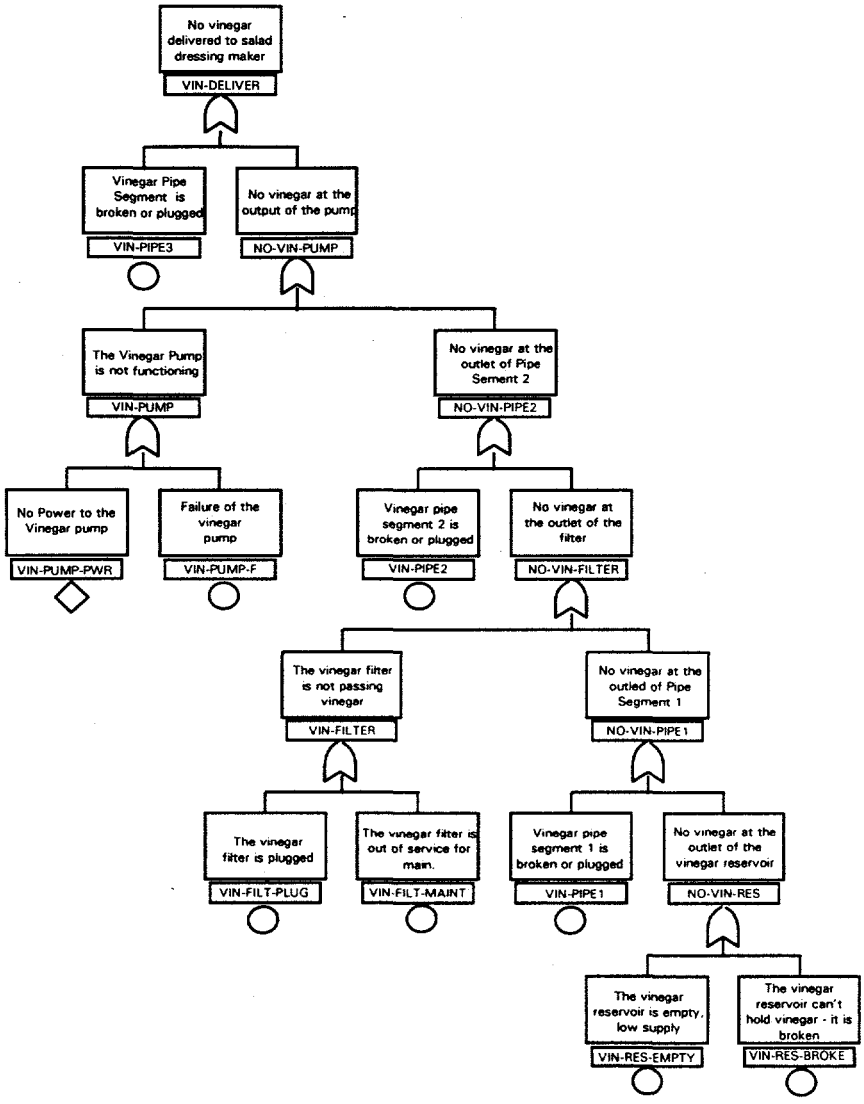


Figure C-1: Fault Tree for Sample Problem

C.2 Specifying Processing Options

The following processing options were selected:

- Run title of Sample Run for LHS Man.
- 100 observations.
- Latin Hypercube Sampling.
- Restricted Pairing.
- Accept the User-Specified Point Estimate Value.
- Calculate and Print Correlation Matrices.

Figure C-2 shows the LHS Processing Options screen after the above selections were made.

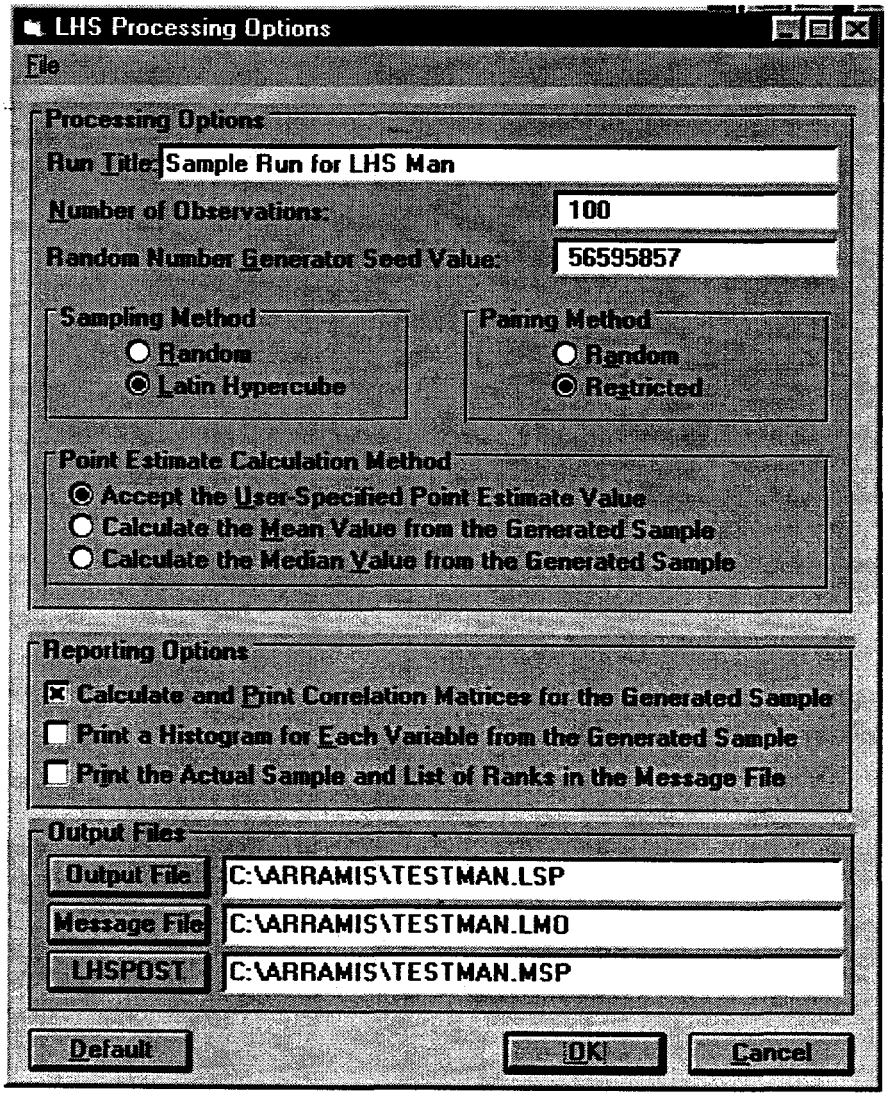


Figure C-2: The LHS Processing Options Screen

C.3 Specifying Distributions

For this analysis, the variables VIN-PIPE1, VIN-PIPE2, and VIN-PIPE3 are to be modeled with a NORMAL distribution, with a mean at 0.003 and a standard deviation of 0.001. The variables VIN-FILT-MAINT, VIN-FILT-PLUG, VIN-PUMP-F, VIN-PUMP-PWR, VIN-RES-BROKE, and VIN-RES-EMPTY are to be modeled with a LOGNORMAL distribution with a mean at 1.1 and an error factor of 2.0.

Once the list was imported, distributions for each variable were assigned. Figure C-3 shows the Distributions screen after the primary events were imported.

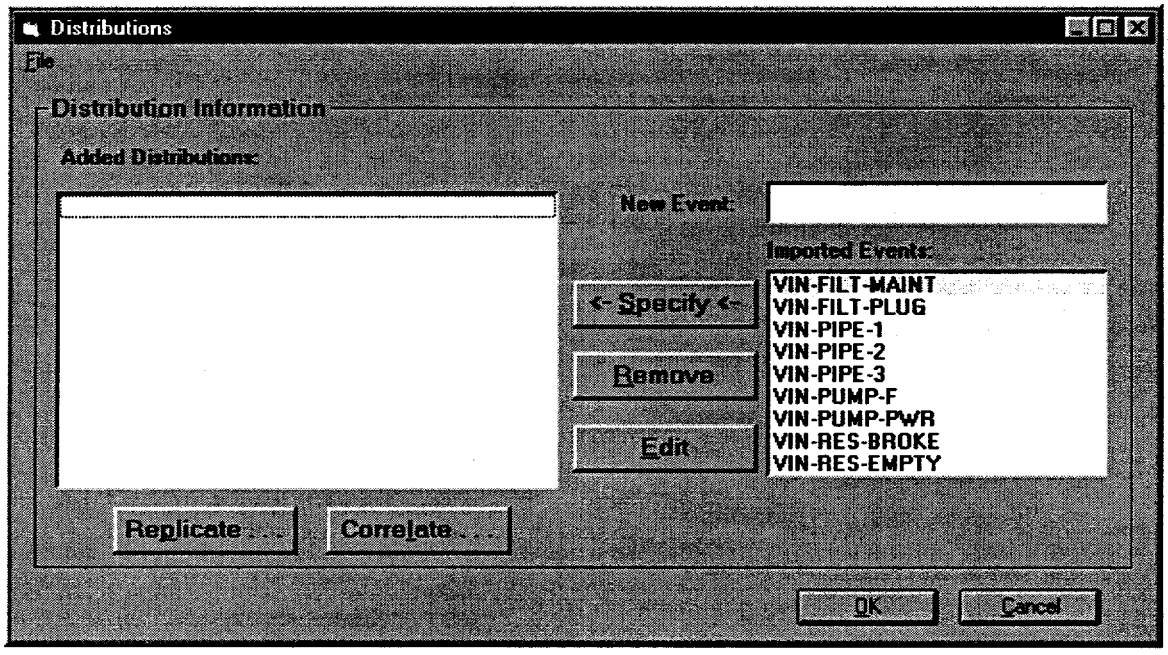


Figure C-3: Distributions Screen after Importing Primary Event List

The variable VIN-PIPE1 was first selected for specification. Figure C-4 shows the Parameters screen for this variable.

Since VIN-PIPE2 and VIN-PIPE3 should contain the same distribution, we will replicate VIN-PIPE1. Select VIN-PIPE1 from the left column on the Distributions and press the Replicate button. Figure C-5 shows the state of the Replicate screen.

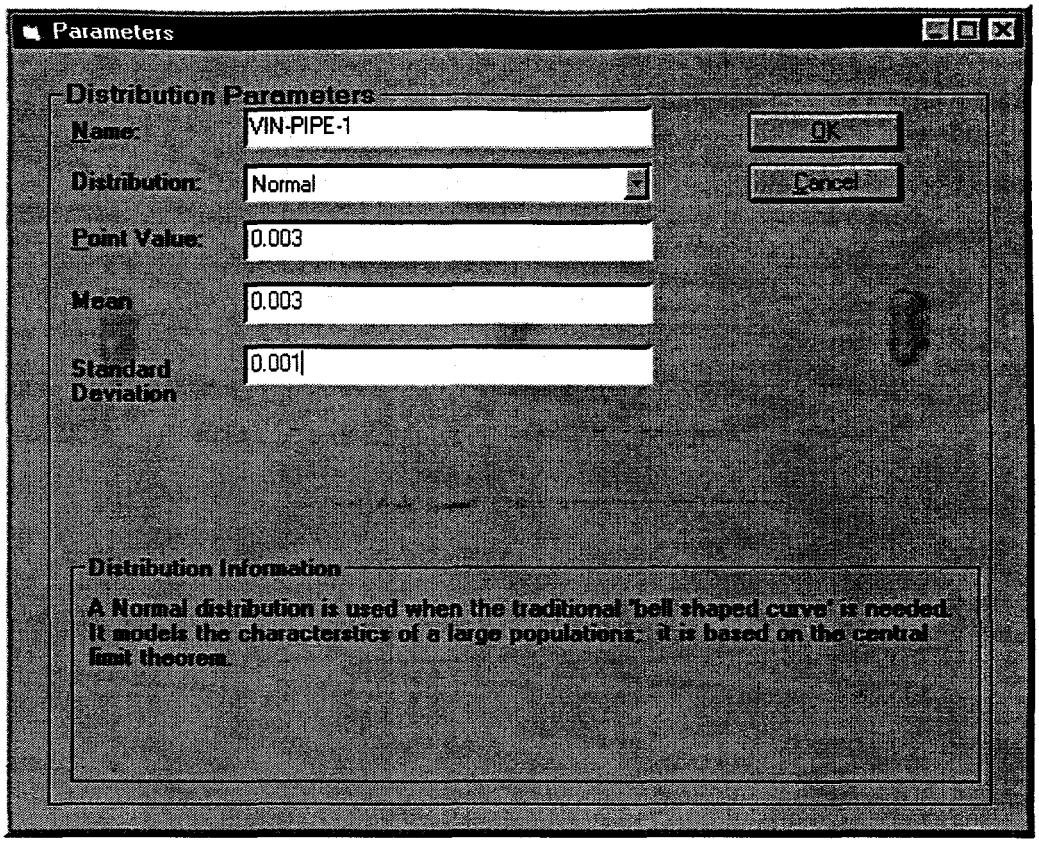


Figure C-4: Parameters Screen

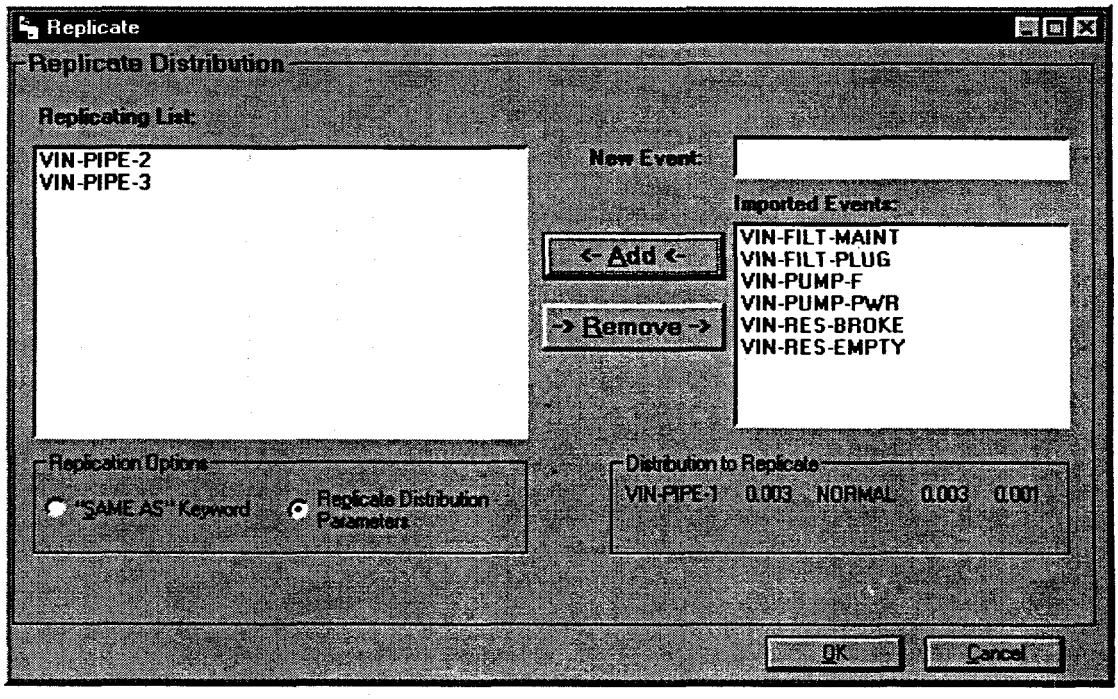


Figure C-5: Replicate Screen

The distribution for VIN-FILT-MAINT was specified and replicated for the remaining variables. Figure C-6 shows the Distributions screen after all variables were specified.

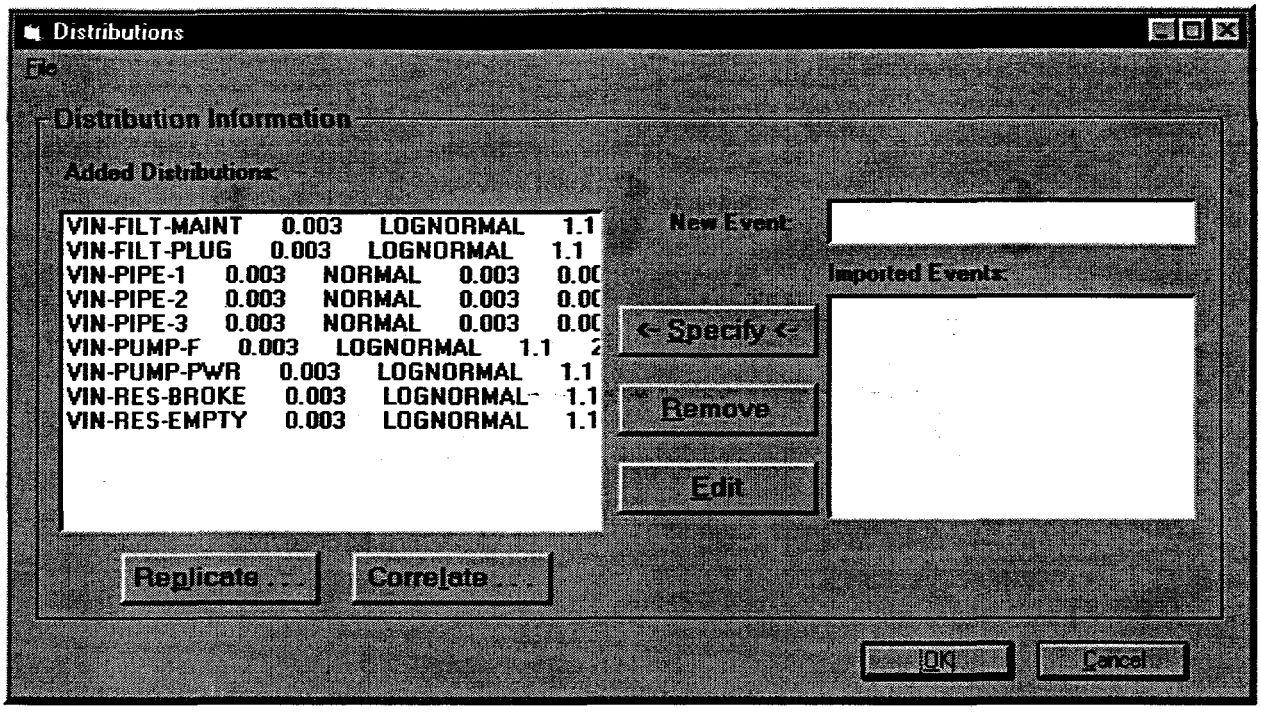


Figure C-6: Distributions Screen after All Distributions Were Specified

C.4 Correlating Variables

The variables VIN-PIPE2 and VIN-PIPE3 were chosen to be correlated to VIN-PIPE1 with a correlation coefficient of 0.5. Figure C-7 shows the Correlation screen after this was selected.

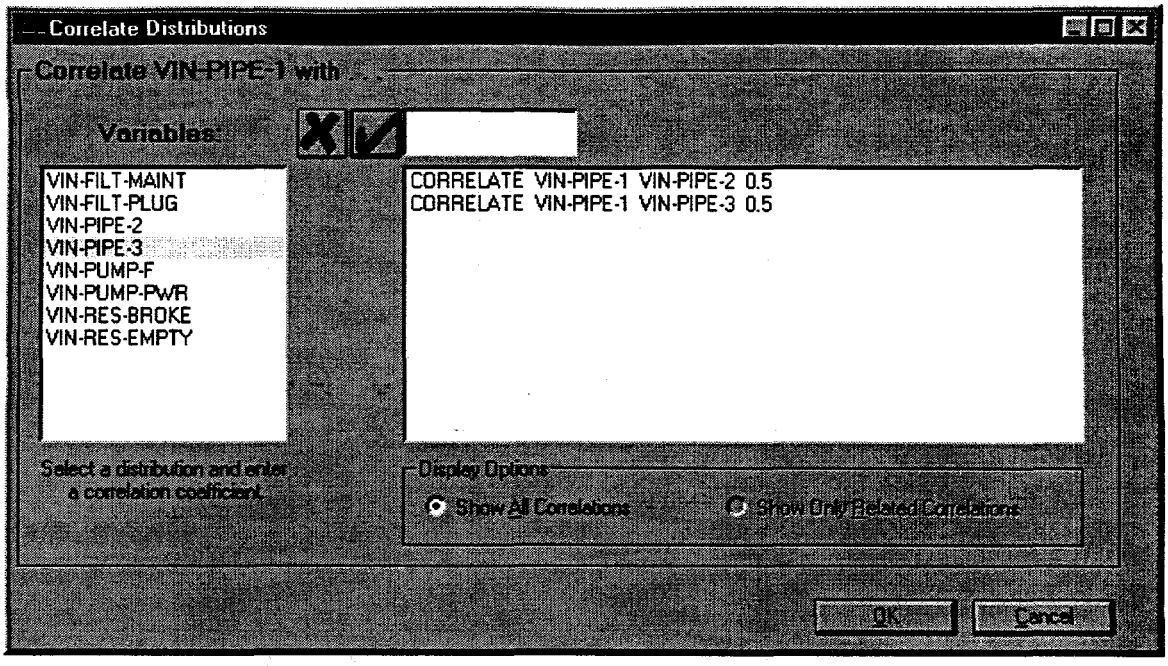


Figure C-7: Correlation Screen

C.5 Input Files

The ASCII text input file is shown in Figure C-8.

```
LHSTITL Sample Run for LHS Man.
LHSOBS  100
LHSSEED  56595857
LHSPVAL  0
LHSRPTS   CORR
LHSOUT   C:\ARRAMIS\TESTMAN.LSP
LHSPOST  C:\ARRAMIS\TESTMAN.MSP
LHSMSG   C:\ARRAMIS\TESTMAN.LMO
DATASET:
VIN-FILT-MAINT    0.003     LOGNORMAL     1.1     2.0
VIN-FILT-PLUG     0.003     LOGNORMAL     1.1     2.0
VIN-PIPE-1     0.003     NORMAL     0.003     0.001
VIN-PIPE-2     0.003     NORMAL     0.003     0.001
VIN-PIPE-3     0.003     NORMAL     0.003     0.001
VIN-PUMP-F     0.003     LOGNORMAL     1.1     2.0
VIN-PUMP-PWR   0.003     LOGNORMAL     1.1     2.0
VIN-RES-BROKE  0.003     LOGNORMAL     1.1     2.0
VIN-RES-EMPTY  0.003     LOGNORMAL     1.1     2.0
CORRELATE  VIN-PIPE-1  VIN-PIPE-2  0.5
CORRELATE  VIN-PIPE-1  VIN-PIPE-3  0.5
```

Figure C-8: LHS Input File Prior to Post Processing

C.6 Results

Figure C-9 shows the probability density function and the cumulative density function for VIN-PIPE1 plotted with CRAVE.

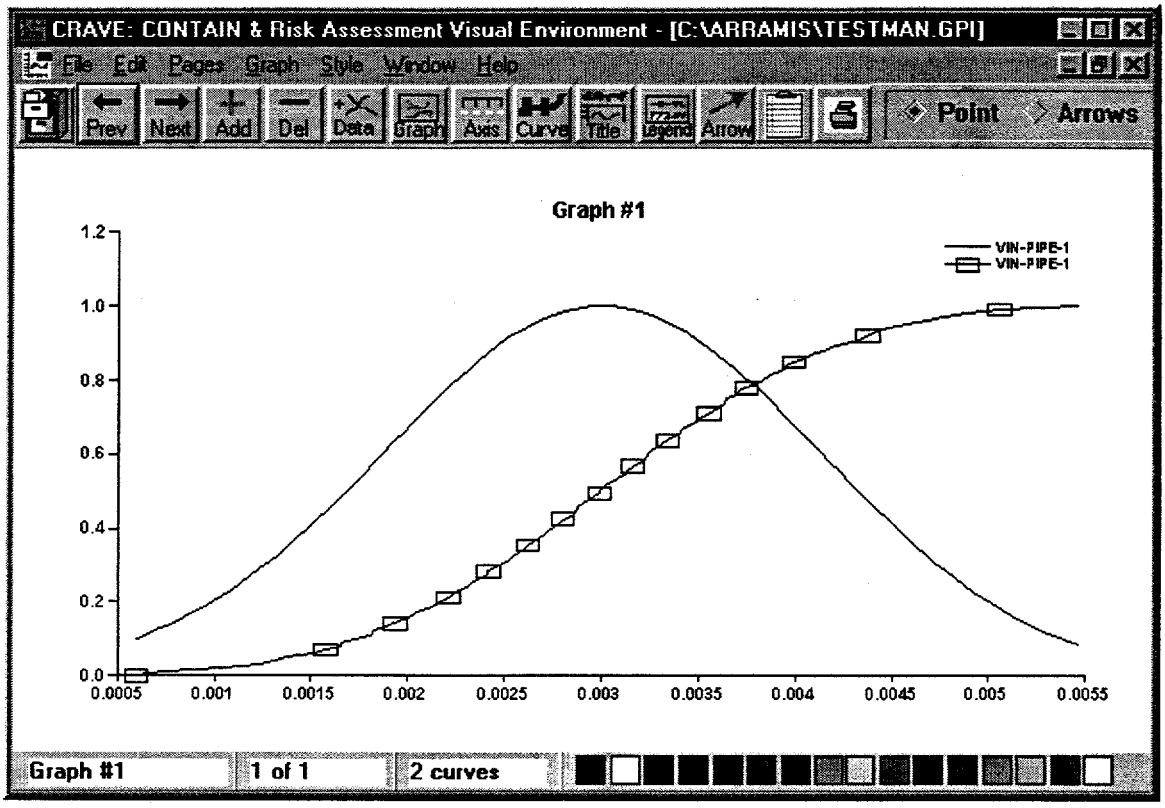


Figure C-9: Results Plotted with CRAVE

The message output file, C:\ARRAMIS\TESTMAN.LMO, is shown in Figure C-10.

```
                    * * * * * * * * * L H S * * * * * * * * *
                    LATIN HYPERCUBE AND RANDOM SAMPLING PROGRAM
                    * * * * * * * * * * * * * * * * * * * * * *

Run on  8/27/97 at  7:38:52.87 with LHS Version: 2.10 Release 2, Compiled 03/10/1997

    SAMPLE RUN FOR LHS MAN.
   Random Seed =     56595857
   Number of Variables =    9
   Number of Observations =  100
   The sample was written to the file: C:\ARRAMIS\TESTMAN.LSP
   Input was read from the file:
   An input correlation matrix has been specified
   The correlation matrices (raw data and rank correlations) will be printed.
```

Figure C-10: Message Output File (1 of 5)

```
SAMPLE RUN FOR LHS MAN.

VARIABLE  DISTRIBUTION          RANGE            LABEL

    1     LOGNORMAL WITH THE PARAMETERS BELOW     VIN-FILT-MAINT       3.0000000E-03
                  MEAN =   1.10                        ERROR FACTOR =   2.00

    2     LOGNORMAL WITH THE PARAMETERS BELOW     VIN-FILT-PLUG        3.0000000E-03
                  MEAN =   1.10                        ERROR FACTOR =   2.00

    3     NORMAL WITH THE PARAMETERS BELOW     VIN-PIPE-1           3.0000000E-03
                  MU =  3.000E-03                    SIGMA =  1.000E-03

    4     NORMAL WITH THE PARAMETERS BELOW     VIN-PIPE-2           3.0000000E-03
                  MU =  3.000E-03                    SIGMA =  1.000E-03

    5     NORMAL WITH THE PARAMETERS BELOW     VIN-PIPE-3           3.0000000E-03
                  MU =  3.000E-03                    SIGMA =  1.000E-03

    6     LOGNORMAL WITH THE PARAMETERS BELOW     VIN-PUMP-F           3.0000000E-03
                  MEAN =   1.10                        ERROR FACTOR =   2.00

    7     LOGNORMAL WITH THE PARAMETERS BELOW     VIN-PUMP-PWR         3.0000000E-03
                  MEAN =   1.10                        ERROR FACTOR =   2.00

    8     LOGNORMAL WITH THE PARAMETERS BELOW     VIN-RES-BROKE        3.0000000E-03
                  MEAN =   1.10                        ERROR FACTOR =   2.00

    9     LOGNORMAL WITH THE PARAMETERS BELOW     VIN-RES-EMPTY        3.0000000E-03
                  MEAN =   1.10                        ERROR FACTOR =   2.00
```

Figure C-10: Message Output File (2 of 5)

```
SAMPLE RUN FOR LHS MAN.

INPUT                          RANK                          CORRELATION                          MATRIX
PAGE   1

    3  1.0000

    4  0.5000  1.0000

    5  0.5000  0.0000  1.0000

           3       4       5

VARIABLES
```

Figure C-10: Message Output File (3 of 5)

```
SAMPLE RUN FOR LHS MAN.

CORRELATIONS  AMONG  INPUT  VARIABLES  CREATED  BY  THE  LATIN  HYPERCUBE  SAMPLE  FOR  RAW  DATA
PAGE   1

    1  1.0000

    2  0.0613  1.0000

    3 -0.0475  0.0264  1.0000

    4 -0.0160  0.0010  0.4926  1.0000

    5 -0.0506 -0.0048  0.4884 -0.0147  1.0000

    6 -0.0541  0.0067 -0.0112 -0.0366 -0.0058  1.0000

    7  0.0304 -0.0248 -0.0243 -0.0093 -0.0048 -0.0948  1.0000

    8 -0.0469 -0.0675 -0.0013 -0.0195  0.0031  0.0577  0.0575  1.0000

    9  0.0099 -0.0172 -0.0147 -0.0282  0.0091  0.0071  0.0057  0.0268  1.0000

            1       2       3       4       5       6       7       8       9

VARIABLES
```

Figure C-10: Message Output File (4 of 5)

```
SAMPLE RUN FOR LHS MAN.

CORRELATIONS  AMONG  INPUT  VARIABLES  CREATED  BY  THE  LATIN  HYPERCUBE  SAMPLE  FOR  RANK  DATA
PAGE   1

    1  1.0000

    2 -0.0120  1.0000

    3  0.0104 -0.0312  1.0000

    4  0.0039 -0.0376  0.4979  1.0000

    5  0.0038  0.0238  0.4777 -0.0017  1.0000

    6 -0.0162 -0.0163  0.0044  0.0096  0.0027  1.0000

    7 -0.0155  0.0574 -0.0261 -0.0012  0.0050  0.0186  1.0000

    8  0.0207 -0.0259 -0.0011 -0.0250 -0.0027 -0.0159 -0.0665  1.0000

    9  0.0181  0.0040 -0.0475  0.0200 -0.0203  0.0242 -0.0163 -0.0052  1.0000

           1       2       3       4       5       6       7       8       9

VARIABLES
```

Figure C-10: Message Output File (5 of 5)

C.7 Postprocessing

To demonstrate the postprocessing abilities of LHS, a new variable, PIPE-FAILURES was introduced. This variable represents the probability of all pipes failing, or VIN-PIPE-1 * VIN-PIPE2 * VIN-PIPE3. This postprocessing screen is shown in Figure C-11. The LHSPOST output file, C:\ARRAMIS\TESTMAN.MSP, is shown in Figure C-12. Figure C-12 has been truncated to shown only observations 1–10 and 91–100. The CRAVE PDF plot of PIPE-FAILURES is shown in Figure C-13.

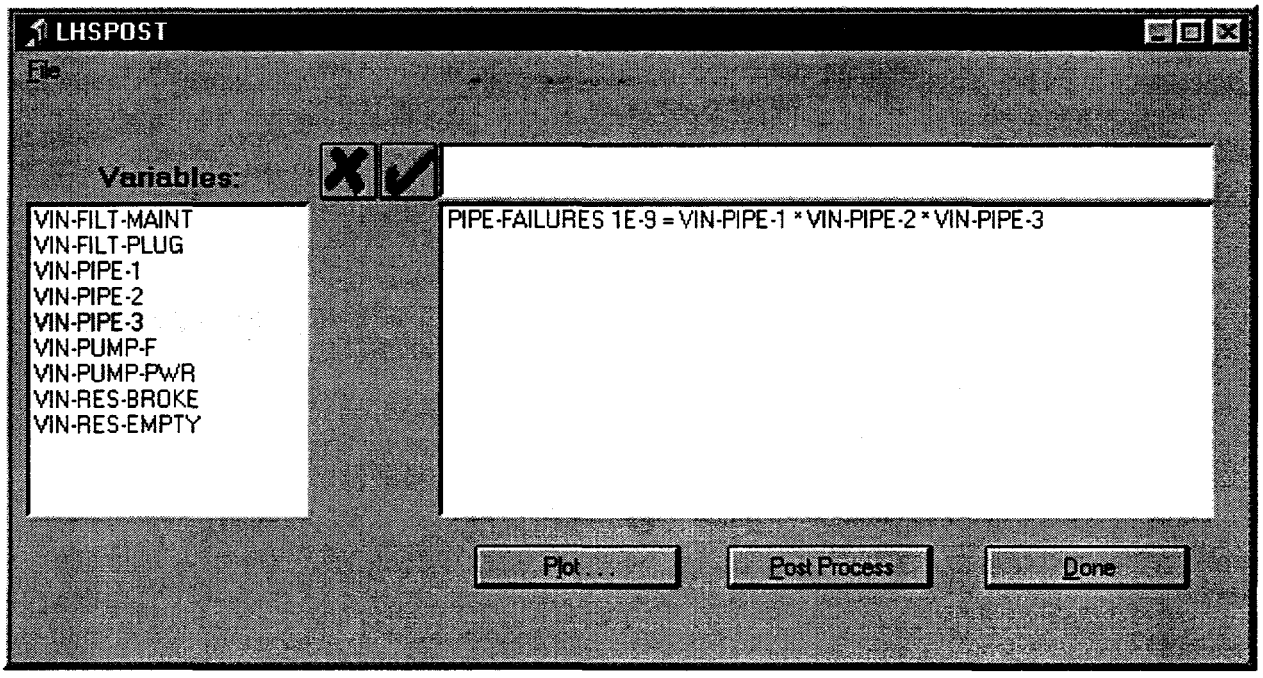


Figure C-11: LHS Postprocessing Screen

```
$ LHS File Format Version 1.00
$
$ This file was created by an LHSPOST run on 09/04/1997 at 12:37:32.171
$ with LHSPOST Version: Version 1.01A DIR, Sept. 24, 1996
$
$ Input file(s) for this run: C:\arramis\testman.lhi
$ and: C:\arramis\testman.lhi
$
$ Sampled data was read from the following file(s):
$ C:\ARRAMIS\TESTMAN.LSP
$
$ The point values in this file were read from the program input.
$
      PIPE-FAILURES          0.100000E-08
$
@UNCERTAINTY
   @OBSERVATIONS           100
   @VARIABLES                1
      PIPE-FAILURES:
@SAMPLEDATA
            1           1    0.130731E-08
            2           1    0.221792E-07
            3           1    0.220179E-08
            4           1    0.683304E-07
            5           1    0.236616E-07
            6           1    0.571191E-08
            7           1    0.445664E-07
            8           1    0.682643E-07
            9           1    0.837648E-08
           10           1    0.140749E-07
              .
              .
              .
           91           1    0.121496E-07
           92           1    0.708263E-08
           93           1    0.854287E-07
           94           1    0.177415E-07
           95           1    0.315649E-07
           96           1    0.174961E-07
           97           1    0.162453E-07
           98           1    0.721146E-08
           99           1    0.197937E-07
          100           1    0.406409E-07
```

Figure C-12: LHSPOST Output File

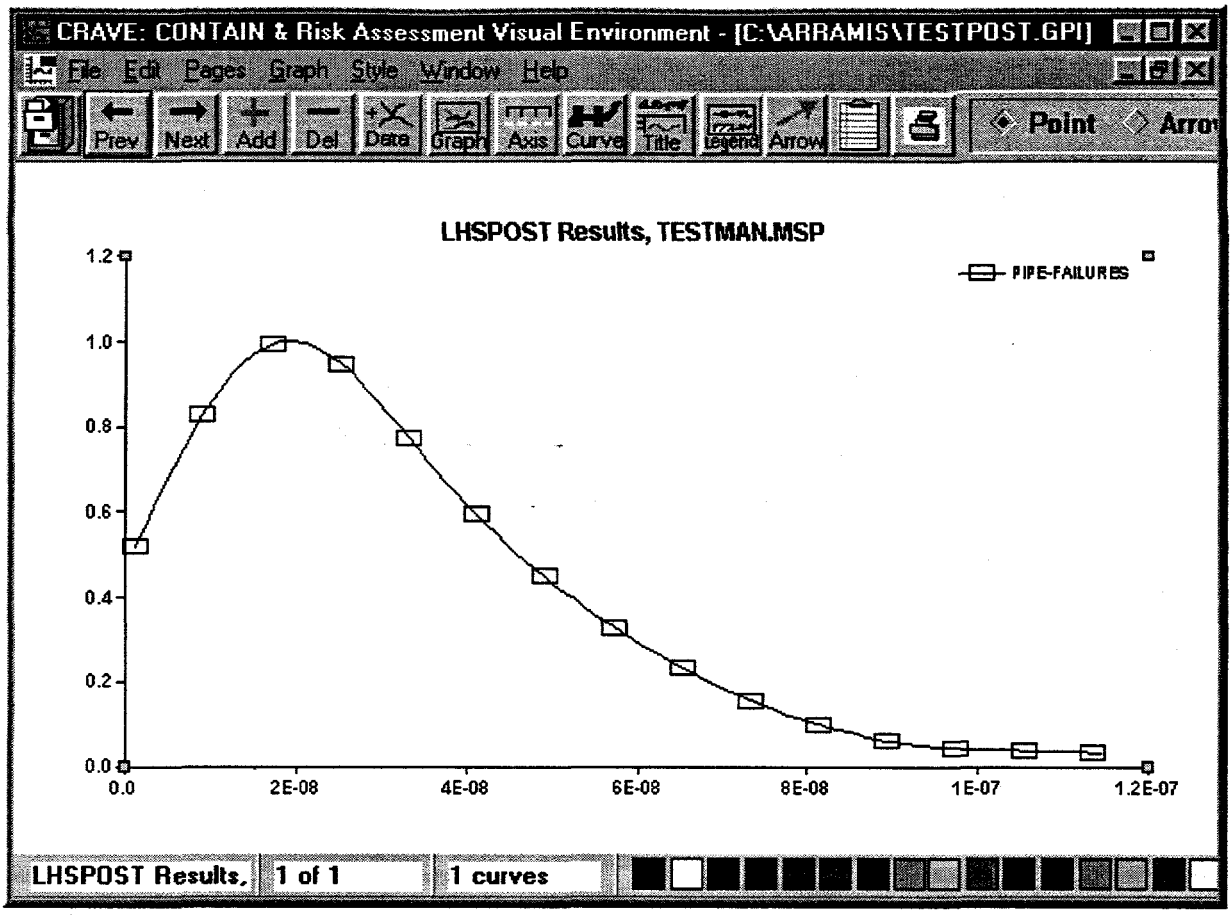


Figure C-13: CRAVE Plot, PIPE-FAILURES

This page intentionally left blank

Appendix D Compatibility with Previous Versions of LHS

The current version of the LHS program still supports most aspects of the original input format as documented in the original LHS User's Guide (SAND83-2365). To use this format, the keyword NOBS must be somewhere in the input file. In this format, there can only be one input file (i.e., there is no use of the keyword PRETRIN).

D.1 Required Conversions

In upgrading the LHS software, we exercised great care to maintain a high degree of compatibility between the upgraded version and the input requirements from the previous version of the software. However, certain changes were introduced in the interpretation of keywords from the previous input format, and these changes will require the user to make minor changes to an input file that was run using the previous version of the software if it is to produce predictable results using the current product.

- The random seed value must now be within the range $[1, 2^{31}-1]$. Negative or zero values for the random seed value are no longer permitted.

- The meaning of the distribution parameters for the NORMAL distribution was changed to be more consistent with traditional use. The new parameters for this distribution are the mean and standard deviation. However, a new distribution type (NORMAL-B) was defined to enable users who wish to use old input files to continue to use the old input parameters and thus avoid introducing errors during the numeric conversion process. Therefore the user must change the distribution type to NORMAL-B for all existing NORMAL distributions within the input file if he or she desires to continue to sample the distribution as it was previously defined.

- The meaning of the distribution parameters for the LOGNORMAL distribution was also changed to be more consistent with traditional use. The new parameters for this distribution are the mean and error factor. However, a new distribution type (LOGNORMAL-B) was defined to enable users who wish to use old input files to continue to use the old input parameters. Therefore, the user must change the distribution type to LOGNORMAL-B for all existing LOGNORMAL distributions within the input file if he or she desires to continue to sample the distribution as it was previously defined.

D.2 Input File Characteristics

The differences between the current input file format and the format used in previous versions of LHS are as follows:

- Everything must be typed in capital letters

- There are no comments (whole line or trailing)
- There are no continuation characters
- No blank lines are allowed
- A return on the last line of input is not permitted
- Each line must be left justified
- Tab characters are not allowed.

D.3 Keywords to Control Program Execution

The following is a list of keywords that set up the execution in the old format.

D.3.1 Reporting and Output Options

TITLE ***title***
(Optional)

This keyword functions in the same way as LHSTITL.

Example: `TITLE  AUTO RELIABILITY RUN - 8/30/96`

This example causes LHS to write the title AUTO RELIABILITY RUN – 8/30/96 as a title in all of its output files.

NAMED OUTPUT
(Optional)

This keyword will generate the LHS sampled data output in the standard format.

Example: `NAMED OUTPUT`

This example causes LHS to write its sampled data in the new output format (including names and point estimate values).

POINT VALUE TYPE ***num***
(Optional)

This keyword functions in the same manner as LHSPVAL.

Example: `POINT VALUE TYPE  2`

This example causes LHS to compute the sample median and include it in the sampled data output file as the point value (if the keyword NAMED OUTPUT has also been used).

D.3.2 Output File Names

SAMPLE FILE ***filename***
(Optional)

This keyword functions in the same way as LHSOUT. If omitted, the sample output file will be written to LHS-S.OUT.

Example: `SAMPLE FILE  SAMPLE.OUT`

This example causes LHS to write its output sample definition file to SAMPLE.OUT

MESSAGE FILE ***filename***
(Optional)

This keyword functions in the same way as LHSMSG. If omitted, the message output file will be written to LHS-P.OUT.

Example: `MESSAGE FILE  MESSAGE.OUT`

This example causes LHS to write its output message file to MESSAGE.OUT.

OUTPUT ***options***
(Optional)

This keyword functions in the same way as LHSRPTS. Any combination of *CORR*, *HIST*, and/or *DATA* may follow.

```
Example:  LHSRPTS  CORR  HIST
Example:  LHSRPTS  DATA  CORR
Example:  LHSRPTS  HIST
```

These examples cause various combinations of reports to be generated by LHS.

D.3.3 Sampling Options

RANDOM SEED ***num***
(Required)

This keyword functions in the same manner as LHSSEED.

Example: `RANDOM SEED 65274`

This example sets the seed number for the random number generator to the value 65274.

NOBS ***num***
(Required)

This keyword functions in the same manner as LHSOBS.

Example: `NOBS 100`

This example will cause LHS to draw a sample consisting of 100 observations.

NREPS ***num***
(Optional)

This keyword functions in the same manner as LHSREPS.

Example: `NREPS 2`

This example will cause LHS to generate two replications of the sample.

RANDOM SAMPLE
(Optional)

This keyword functions in the same way as LHSOPTS RANDOM SAMPLE.

Example: `RANDOM SAMPLE`

This example causes LHS to use random sampling instead of stratified sampling.

RANDOM PAIRING
(Optional)

This keyword functions in the same way as LHSOPTS RANDOM PAIRING.

Example: `RANDOM PAIRING`

This example causes LHS to use random pairing instead of restricted pairing.

D.4 Keywords to Specify Distributions to be Sampled

The old format supports all of the distributions defined in Section 4.3. The same keywords are used for each; however, the input format is different. To specify a distribution, note the following format:

distribution optional_field_for_naming_variable
dist_param dist_param dist_param ...

where the first line contains the distribution the user specified and an optional field to name the distribution. If a name is omitted, then a default name is assigned. The parameters required to perform the distribution are then supplied on the following lines.

```
Example:    NORMAL    THIS IS AN EXAMPLE VARIABLE
            0.0  1.0
```

In this example, a standard normal distribution (mean = 0, variance = 1) is defined using the old LHS distribution format.

D.5 Keyword to Control Correlation

Instead of using the keyword CORRELATE followed by two variables and a value, the keyword CORRELATION MATRIX is supported. 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. 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 (1982) requires $n > k$, where n is NOBS and k is the number of variables. That is, the technique can only be applied directly if $n > k$; otherwise an error message is printed and a modified form of random pairing is used (see Section 3.3.5). 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 $n \times k$ 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. In addition, if 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.

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 and a message to that effect is printed after the correlation matrix.

Example:

```
CORRELATION MATRIX
3     1  2  0.8
      1  5  0.7
      2  5  0.6
```

This example defines three pairwise correlations between random variables. The first correlation specified requests LHS to achieve a rank correlation of 0.8 between the first and second variables that were defined in the LHS input. The second correlation specified requests LHS to achieve a rank correlation of 0.7 between the first and fifth variables that were defined in the LHS input. The last correlation specified requests LHS to achieve a rank correlation of 0.6 between the second and fifth variables that were defined in the LHS input.

REFERENCES

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

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, NM.

Appendix E Adjusting LHS Array Sizes

LHS supports dynamic dimensioning to allow the user greater flexibility in running large problems. Thus instead of calling on the ARRAMIS™ development team for a recompiled, larger version of LHS, the user can modify the array dimensions by editing an initialization file.

The SIPRA.INI file contains the array dimensions for each of the programs within the ARRAMIS™ code suite. Each code, including LHS, first opens this file to determine its array sizes. If the code cannot find this file, or if there is a problem reading this file, an error message is printed to the screen and default dimensions are used. Note that the code ***will not terminate***; default dimensions will be used and execution will continue.

To edit the SIPRA.INI file, select File | Edit SIPRA.INI File from the menu on the Main screen. This will invoke the SIPRA.INI editor, shown in Figure E-1.

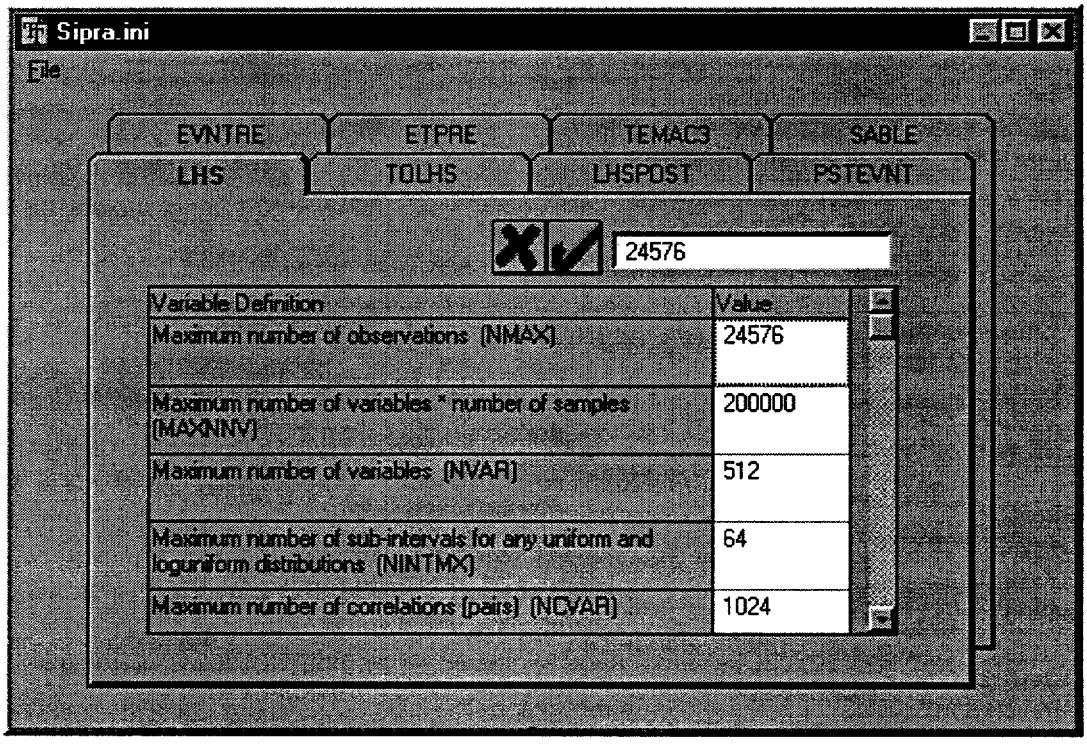


Figure E-1: SIPRA.INI Editor

You may alter the dimensions of any of the ARRAMIS™ codes with this interface. To modify the LHS dimensions, select the LHS tab. This screen contains two columns; the left column contains the names and descriptions of the dimensions, and the right column contains the actual dimension values.

To change an array dimension:

1. Select the row containing the dimension you would like to alter. This will place the dimension in the edit box above the grid.
2. Edit the dimension in the edit box and press the Checkmark button to save your results.
3. If you would like to restore the default dimensions, select File | Default Settings.
4. Select File | Save from the menu, which will save these results.

After the SIPRA.INI file has been saved with the new dimensions, you may rerun your analysis. LHS will read the altered SIPRA.INI file and dimension itself accordingly.

Appendix F Distribution Input File Summary

F.1 Distributions Recognized by LHS

F.1.1 Defined Distributions in LHS

Distribution Name	Parameters			
NORMAL	*mean*	*standard-deviation* *> 0.0*		
TRUNCATED NORMAL	*mean*	*standard_deviation* *> 0.0*	*lower* *0.0 ≤ lower < upper ≤ 1.0*	*upper*
BOUNDED NORMAL	*mean*	*standard_deviation* *> 0.0*	*lower_bound* *lower_bound < upper_bound*	*upper_bound*
NORMAL-B	*value_at_0.001*	*value_at_0.999*		
LOGNORMAL	*mean* *> 0.0*	*error_factor* *> 1.0*		
LOGNORMAL-N	*mean* *> 0.0*	*standard_deviation* *> 0.0*		
TRUNCATED LOGNORMAL	*mean* *> 0.0*	*error_factor* *> 1.0*	*lower* *0.0 ≤ lower < upper ≤ 1.0*	*upper*
TRUNCATED LOGNORMAL-N	*mean* *> 0.0*	*standard_deviation* *> 0.0*	*lower* *0.0 ≤ lower < upper ≤ 1.0*	*upper*
BOUNDED LOGNORMAL	*mean* *> 0.0*	*error_factor* *> 1.0*	*lower_bound* *lower_bound < upper_bound*	*upper_bound*
BOUNDED LOGNORMAL-N	*mean* *> 0.0*	*standard_deviation* *> 0.0*	*lower_bound* *lower_bound < upper_bound*	*upper_bound*
LOGNORMAL-B	*value_at_0.001* *> 0.0*	*value_at_0.999* *> 0.0*		
UNIFORM	*A* *A < B*	*B*		
LOGUNIFORM	*A* *0.0 < A < B*	*B*		
EXPONENTIAL	λ > 0			
MAXIMUM ENTROPY	*A* $0 \le A < \mu < B$	μ	*B*	
WEIBULL	α > 0	β > 0		
PARETO	α > 2.0	β > 0.0		
GAMMA	α	β		
BETA	*A* $0 \le A < B$	*B*	*p* ≥ 0.001	*q* ≥ 0.001

Distribution Name	Parameters		
INVERSE GAUSSIAN	μ > 0	λ > 0	
TRIANGULAR	a $a < c$	b $a \leq b \leq c$	c
POISSON	λ > 0		
BINOMIAL	p $0 < p < 1$	n > 1	
NEGATIVE BINOMIAL	p $0 < p < 1$	n > 1	
GEOMETRIC	p $0 < p < 1$		
HYPERGEOMETRIC	N_N $N_1 < N_R < N_N$	N_1	N_R

F.1.2 User-Defined Continuous Distributions

CONTINUOUS LINEAR ***n ordered_pair$_1$ ordered_pair$_2$... ordered_pair$_n$***
and
CONTINUOUS LOGARITHMIC ***n ordered_pair$_1$ ordered_pair$_2$... ordered_pair$_n$***

- n is an integer number of ordered pairs that will be used to represent the CDF ($n > 1$).
- Within each ordered pair, the first number is the value of the distribution; the second number is the cumulative probability (CDF value) associated with this distribution value.
- The probabilities must increase monotonically starting with 0.0 and ending with 1.0.
- The distribution values (the first entry in the ordered pair) must also increase monotonically.

CONTINUOUS FREQUENCY ***n ordered_pair$_1$ ordered_pair$_2$... ordered_pair$_n$***

- n is an integer number of ordered pairs that will be used to represent the CDF ($n > 1$).
- The first value in each ordered pair is the value of the distribution at a particular point.
- The second value in the ordered pair is a relative frequency associated with the value.
- The frequencies are relative only to each other, and must be strictly positive.
- The values must increase monotonically.

UNIFORM* ***n*** ***obs_1*** ***obs_2...*** ***obs_n*** ***first_point*** ***$endpoint_1$*** ***$endpoint_2$...*** ***$endpoint_n$***
and
LOGUNIFORM* ***num*** ***obs_1*** ***obs_2...*** ***obs_n*** ***first_point*** ***$endpoint_1$*** ***$endpoint_2$...*** ***$endpoint_n$***

- The first parameter specifies the number of intervals n in an n-part histogram.
- n is followed by a series of n integer values *obs*.
- Each value obs_i represents the number of observations that LHS will draw from the i^{th} interval. The sum of all obs_i must equal the number of observations requested from LHS.
- The obs_i are followed by n contiguous intervals that are to be sampled.
- The intervals are specified by entering a *first_point* (the distribution minimum) followed by n interval *endpoint* values.

F.1.3 User-Defined Discrete Distributions

DISCRETE CUMULATIVE ***n*** ***$ordered\_pair_1$*** ***$order\_pair_2$...*** ***$ordered\_pair_n$***

- n is an integer number of ordered pairs that will be used to represent the CDF ($n > 1$).
- Within each ordered pair, the first number is the value of the distribution; the second number is the cumulative probability (CDF value) associated with this distribution value.
- Probabilities increase monotonically starting with a value greater than 0.0 and ending with 1.0.
- The distribution values (the first entry in the ordered pair) must increase monotonically.

DISCRETE HISTOGRAM ***n*** ***$ordered\_pair_1$*** ***$ordered\_pair_2$...*** ***$ordered\_pair_n$***

- n is an integer number of ordered pairs that will be used to represent the CDF ($n > 1$).
- The first value in each ordered pair is the value of the distribution at a particular point.
- The second value in the ordered pair is a relative frequency associated with the value.
- Frequencies are relative only to each other, and must be strictly positive.
- The values must increase monotonically.

F.2 Keywords Not Related to Distributions

CONSTANT ***val***

SAME AS ***old_variable***

- Note *old_variable* must be the name of a random variable that is explicitly defined with a distribution description.

CORRELATE ***first_variable*** ***second_variable*** ***corr***

- The specified correlation value parameter *corr* must satisfy the condition ($-1 < corr < 1$).

F.3 Input File Specifications

F.3.1 File Characteristics

- Space-delimited case-insensitive free-format ASCII text file.
- Each input line must contain no more than 80 ASCII characters.
- Continuation character is either a "#" or a "%", preceded by one or more blanks, at the end of the line.
- All text following "$" characters (dollar sign) are considered comments.
- Each continuation line must end with a continuation character unless it terminates the command.
- A command may begin and end at any point on the line (an arbitrary number of leading and trailing spaces may be used as desired).
- If a command keyword is composed of more than one word, those words must be separated by *one and only one blank space.*
- Multiple blank spaces may be included between a keyword and any required alphabetic or numeric values.
- LHS treats commas (",") and tab characters as being fully equivalent to and interchangeable with blank space characters.
- Numeric input may be specified in any Fortran-standard format.
- In a trailing comment, the dollar sign must be preceded by one or more blank spaces.
- It is not possible to place comments between items on a single line.
- Full-line comments may be placed between a continued line and its continuation or between consecutive continuation lines.

F.3.2 Names of Random Variables

- Names may contain no more than 16 characters
- The "$" (dollar sign), "#" (number sign), "%" (percent sign), "," (comma), " " (blank space), and tab characters are not valid within names
- A name must not form a valid representation of a real number for an ANSII-standard Fortran compiler.

F.3.3 Distribution File Structure

- LHS will take as its input any line from the file on which the first keyword is Data:.
- All lines that follow the keyword Dataset: (on a line by itself) are assumed to belong to LHS input regardless of whether they begin with the keyword Data:.
- Keywords used to describe distribution input follow a common format:

name point_value dist-name dist-parm dist-parm .....

- The *point_value* parameter represents the value that the analyst wants used for this variable by models that cannot accept or effectively use the sampled results provided by LHS.
- The *point_value* field is required only when the analyst specifies that LHS is to use the user-specified value for the point estimate instead of computing the sample mean or median for this purpose. In all other instances, the *point_value* parameter is optional and can be omitted.

DISTRIBUTION:

Nelson Deane
Lockheed Martin Astro Space
Space Power Programs
4041 N. First Street
San Jose, CA 95134

Dr. Frank Kampas
Lockheed Martin Co.
Room 20B41, Building B
720 Vandenberg Rd.
King of Prussia, PA 19406

Dr. Chuong Ha
Lockheed Martin Astro Space
Space Power Programs
4041 N. First Street
San Jose, CA, 95134-1503

Steve Loughlin
Lockheed Martin Co.
Room 20B41, Building B
720 Vandenberg Rd.
King of Prussia, PA 19406

M. E. Hall
Los Alamos National Laboratory
Mail Stop K557
Los Alamos, NM 87545

D. Stack
Los Alamos National Laboratory
Mail Stop K556
Los Alamos, NM 87545

M. A. Cunningham, Chief
Probabilistic Risk Analysis Branch
Division of Safety Technology
Office of Nuclear Regulatory Research
MS 10E 50
US Nuclear Regulatory Commission
Washington, DC 20555-0001

M. Drouin
Probabilistic Risk Analysis Branch
Division of Safety Technology
Office of Nuclear Regulatory Research
MS 10E 50
US Nuclear Regulatory Commission
Washington, DC 20555-0001

Gareth Parry
Division of Systems Safety and Analysis
Office of Nuclear Reactor Regulation
MS 9B9
US Nuclear Regulatory Commission
Washington, DC 20555-0001

Nathan Siu
Probabilistic Risk Analysis Branch
Division of Safety Technology
Office of Nuclear Regulatory Research
MS 10E 50
US Nuclear Regulatory Commission
Washington, DC 20555-0001

H. J. Vandermolen
Probabilistic Risk Analysis Branch
Division of Safety Technology
Office of Nuclear Regulatory Research
MS 10E 50
US Nuclear Regulatory Commission
Washington, DC 20555-0001

MS0405	D. E. Bennett III, 12333
MS0405	M. P. Bohn, 12333
MS0405	R. J. Breeding, 12333
MS0405	D. D. Carlson, 12333
MS0405	N. J. Dhooge, 12333
MS0405	M. A. Dvorack, 12333
MS0405	M. K. Fuentes, 12333
MS0405	T. R. Jones, 12333
MS0405	S. A. Kalemba, 12333
MS0405	Y. T. Lin, 12333

MS0405 K. J. Maloney, 12333
MS0405 W. H. McCulloch, 12333
MS0405 D. T. Pabst, 12333
MS0405 K. B. Sobolik, 12333
MS0405 J. L. Tenney, 12333
MS0405 M. L. Young, 12333
MS0449 M. D. Murphy, 6231
MS0718 G. S. Mills, 6341
MS0718 F. L. Kanipe, 6341
MS0734 R. D. Waters, 6803
MS0739 R. K. Cole Jr., 6421
MS0744 D. L. Berry, 6403
MS0744 D. A. Powers, 6404
MS0746 D. J. Anderson, 6411
MS0746 M. E. Armendariz, 6411
MS0746 C. B. Atcitty, 6411
MS0746 J. R. Bechdel, 6411
MS0746 J. E. Campbell, 6411
MS0746 L. D. Chapman, 6411
MS0746 R. M. Cranwell, 6411
MS0746 C. P. Harlan, 6411
MS0746 J. H. Linebarger, 6411
MS0746 D. P. Miller, 6411
MS0746 D. G. Robinson, 6411
MS0746 S. L. Rountree, 6411
MS0746 L. P. Swiler, 6411
MS0746 H. O. Whitehurst, 6411
MS0747 A. L. Camp, 6412
MS0747 D. B. Bertolino, 6412
MS0747 T. D. Brown, 6412
MS0747 R. S. Campbell, 6412
MS0747 R. G. Cox, 6412
MS0747 V. J. Dandini, 6412
MS0747 S. L. Daniel, 6412
MS0747 J. A. Forester, 6412
MS0747 K. H. Jorgensen, 6412
MS0747 B. E. Meloche, 6412
MS0747 D. W. Whitehead, 6412
MS0747 G. D Wyss, 6412 [50]
MS0748 B. S. Altman, 6413
MS0748 A. S. Benjamin, 6413
MS0748 R. Beraun, 6413
MS0748 N. N. Brown, 6413
MS0748 J. J. Gregory, 6413
MS0748 J. H. Lee, 6413
MS0748 D. B. Mitchell, 6413
MS0748 S. P. Nowlen, 6413
MS0748 R. E. Pepping, 6413
MS0748 L. Shyr, 6413
MS0748 T. J. Tanaka, 6413
MS0783 E. K. Lemen, 6311
MS0835 V. J. Romero, 9113
MS0835 B. L. Bainbridge, 9113
MS1345 D. P. Gallegos, 6416

MS9018 Central Technical Files, 8940-2
MS0899 Technical Library, 4916 [5]
MS0619 Review and Approval Desk, 12690, for DOE/OSTI [2]