NASA/CR—2002-212008

Probabilistic Analysis and Density Parameter Estimation Within Nessus

Cody R. Godines and Randall D. Manteufel
University of Texas at San Antonio, San Antonio, Texas

December 2002

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at 301–621–0134

- Telephone the NASA Access Help Desk at 301–621–0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076

NASA/CR—2002-212008

Probabilistic Analysis and Density Parameter Estimation Within Nessus

Cody R. Godines and Randall D. Manteufel
University of Texas at San Antonio, San Antonio, Texas

Prepared under Grant NAG3–2297

National Aeronautics and
Space Administration

Glenn Research Center

December 2002

Acknowledgments

The authors greatly acknowledge the guidance and encouragement of Drs. Christos C. Chamis and Shantaram Pai and the support of the NASA Glenn Research Center and the Minority University Research and Education Program, NASA Headquarters. Dr. Ben Thacker and Mr. David Riha are recognized for their participation and contributions to the two courses. Drs. Harry Millwater and Ronald Bagley are thanked for being on Mr. Cody Godines' thesis committee. Also acknowledged are Undergraduate Research Assistants, Mr. Henock Perez, Mr. Luis Rangel, and Mr. Jody Cronenberger for their contributions to this project. The UTSA Alliance for Minority Participation (AMP) program is also acknowledged for supporting Mr. Ronald Magharing as a collaborating undergraduate student during the summer of 2000.

The Aerospace Propulsion and Power Program at
NASA Glenn Research Center sponsored this work.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100

Available electronically at http://gltrs.grc.nasa.gov

TABLE OF CONTENTS

TABLE OF CONTENTS iii
NOMENCLATURE v
1.0 INTRODUCTION 1
2.0 EDUCATION 3
ME 5543, Probabilistic Engineering Design 3
ME 4723, Reliability And Quality Control In Engineering Design 4
Enrollment Data 5
CAPS Lab 5
NESSUS Student User's Manual 6
3.0 RESEARCH 7
3.1 INTRODUCTION 9
Engineering Studies, Uncertainties, and Reliability 9
Reliability Methods 12
Reliability Analysis Methodology 17
Areas of Application 19
Purpose and Scope 20
Organization 23
3.2 RELIABILITY AND THE DENSITY FUNCTION 25
Random Variables 27
Reliability 46
Reliability Calculations Using the Mean and Standard Deviation 49
Reliability Calculations Using Probability Calculations 50
3.3 RANDOM SAMPLING AND ESTIMATION 55
Random Sampling 56
Estimators and Estimation 69
Sampling Method Comparison 75
3.4 NESSUS ENHANCEMENT WITH LHS ABILITIES 79
Introduction to NESSUS 79
Current State of NESSUS Monte Carlo Thread 80
Flow of Monte Carlo Subroutine Calls 81
Monte Carlo Output 82
New Latin Hypercube Thread 92
Flow of LHS Subroutine Calls 93
The lhs_xsample.f90 Subroutine 97
The *lhs_calc.f90* Subroutine 101
Latin Hypercube Output 102
3.5 TEST CASES 113
Test Case 1: Stage II Crack Propagation – Paris Law 114
Response Function and Design Variables 114
Convergence of Sampling Methods 123
Test Case 4: Nonlinear Response, Non-Normal Variables 140
Response Function and Design Variables 140
Convergence of Sampling Methods 141
Test Case 6: Maximum Radial Stress of Rotating Disk 152

Response Function and Design Variables 152
Convergence of Sampling Methods 155
Test Case 8: Nonlinear Response, Standard Normal Variables 167
Response Function and Design Variables 167
Convergence of Sampling Methods 167
Results 179
3.6 CONCLUSIONS 187
Summary 187
Conclusions 187
LHS Enhancement 187
MC and LHS Comparison 190
4.0 ACCOMPLISHMENTS 195
4.1 Accomplishments: Education 195
4.2 Accomplishments: Research 196
4.3 Student Achievements 197

5.0 REFERENCES 199

APPENDIX I 203
APPENDIX II 207
APPENDIX III 225
APPENDIX IV 231

NOMENCLATURE

PDF	probability density function
CDF	cumulative distribution function
μ	mean
σ	standard deviation
p	proportion of responses observed to be below a certain value of the response relative to the total number of observations after a long series of observations are made; also, represents probability of future observations of same response being below certain value
100p%	percent associated with above probability, p
$100p^{th}$ percentile	value of response, z, for which 100p% of observations will lie below this value
MC	Monte Carlo Sampling Method
LHS	Latin Hypercube Sampling Method
θ	target parameter
$\hat{\theta}$	estimator of the target parameter
$\bar{X}$, $\mathbf{X}$	random vector input
$Z(\bar{X})$, $Z(\mathbf{X})$	response of system under consideration
NESSUS	Numerical Evaluation of Stochastic Structures Under Stress
SAE	Society of Automotive Engineers
MPP	Most probable point

1.0 INTRODUCTION

The goals of this project were to promote an innovative educational and research experience for students at UTSA in the area of probabilistic structural analysis, probabilistic methods, and reliability. The NASA John H. Glenn Research Center (GRC) is a leader in the reliability of turbomachinery for aircraft propulsion and has developed advanced analysis methods and tools such as the computer code, NESSUS (Numerical Evaluation of Stochastic Structures Under Stress), in collaboration with Southwest Research Institute (SwRI) [Chamis, 1996; Southwest Research Institute, 1995; Pai, 1995; Millwater et al., 1992]. The staff at SwRI are experts in the area of probabilistic analysis. Because of the close proximity of SwRI and the University of Texas at San Antonio (UTSA) collaboration frequently exists to promote education and research. This report describes the collaborative effort between UTSA, SwRI and GRC to improve undergraduate and graduate education in engineering at UTSA. This project includes both education and research objectives.

The education component consisted of the development and offering of two courses in mechanical engineering. These courses exposed students to probabilistic methods, emphasizing the identification and quantification of uncertainties in structures, materials, loads, and failure modes. In these courses, students studied probabilistic methods and learned to apply techniques for assessing reliability and identifying important variables, especially for structural problems using the NESSUS computer program. These engineering courses are intended to expose students to both theoretical and computational methods used in probabilistic analyses. Dr. Ben Thacker and Mr. David Riha, research engineers in the Probabilistic Mechanics and Reliability Section of SwRI, helped to develop the course content and served as instructors for the two

courses. Both individuals have been involved with the NESSUS code development for over ten years and have organized and taught annual SwRI short courses on probabilistic analysis and design [Southwest Research Institute, 1996]. All students attending these classes, or participating as research assistants, had the opportunity to develop unique skills in the growing field of probabilistic design.

The research portion of this report presents the master's thesis completed by Mr. Cody Godines. His thesis had two main objectives. The first goal that was successfully obtained was the enhancement of NESSUS with the ability to perform Latin Hypercube Sampling. The aim of the second task was to compare Latin Hypercube Sampling to that of Monte Carlo. This was done by comparing their error in estimating the mean, standard deviation, and 99th percentile of the probability density function of four test cases. These test cases are a few of the responses put forth by the Society of Automotive Engineers (SAE) for the purpose of testing probabilistic methods.

The grant has provided support for UTSA's Center for Advanced Propulsion Studies (CAPS) laboratory as it continues to establish an educational and research infrastructure to conduct more long-term research projects in this area. In particular, NASA funding from this project has supported two graduate students and four undergraduate students, two course instructors, a part-time Research Engineer, a part-time Systems Engineer and the Principal Investigator.

2.0 EDUCATION

This project provided students at UTSA a unique educational experience in both theoretical and computational probabilistic structural analysis methods by supporting the development and offering of two courses. Syllabi for both courses are provided in Appendix I. Neither course would have been offered if not for this Partnership Award.

In these courses, students had the opportunity to interact with leading researchers in the area. They were introduced to the NESSUS computer program for probabilistic analysis of structural and mechanical systems. Emphasis was placed on the identification and quantification of uncertainties in engineering designs, and the methods used to accommodate these uncertainties to achieve safe, efficient, and reliable designs. The application areas for probabilistic analysis and design continue to grow and include: structural analysis, fracture mechanics, reliability-based design optimization, automotive structures, thermal-fluids, geomechanics, turbine engine structures, biomechanics, and other engineering applications. Hence, students were exposed to probabilistic methods that have a wide range of applications. They gained valuable hands-on experience with analytical and computational probabilistic methods that will distinguish them from other engineering graduates. Each course is briefly described here.

ME 5543, Probabilistic Engineering Design

This graduate level course was taught in the Spring 2000 semester. Although exceptional undergraduate students can petition to take graduate courses at UTSA, none did and only graduate students attended this class. The instructor was Dr. Ben Thacker of SwRI with

assistance from Dr. Randall Manteufel, Callie Bast, and Mark Jurena at UTSA. The course covered topics in probability and statistics, probabilistic design, computational methods, and reliability. A final project required students to write a program in the language of their choice that would perform probabilistic calculations using two competitive computational techniques. The students then chose a response to study that had a significant number of uncertain variables with various non-normal probability distributions. The programs were written in such languages as Fotran, C++, Visual Basic, and some students even used Mathcad to perform their calculations. Mr. Cody Godines was a student in this class and his final project involved the design of a scuba tank, which was presented to NASA-GRC and is given in Appendix II. His Fortran code was named Quest. All of the coding performed by Mr. Cody Godines was done using the SGI O2 (R5000) workstations in the Center for Advanced Propulsion Studies. These workstations were paid for by prior NASA grants and UTSA cost sharing and are mentioned below.

ME 4723, Reliability And Quality Control In Engineering Design

A second course was offered during the summer 2000 semester. This class was a senior-level undergraduate course that was used to satisfy technical electives in the mechanical engineering degree program. The instructor was Mr. David Riha of SwRI with assistant from the Dr. Manteufel, C. Bast, and M. Jurena at UTSA. The course covered topics in probability theory, reliability, testing, probabilistic design, and introduction to the NESSUS computer program. Students learned how to assess component and system reliability, assess uncertainties in a system, describe uncertainties using random variables, identify important random variables in the system, provide information for risk-based decision analyses and reliability-based optimization, and develop designs that are more cost-effective and reliable.

Enrollment Data

Five graduate students successfully completed the Spring 2000 course, ME 5543, entitled "Probabilistic Engineering Design." Fifteen undergraduate students and two graduate students completed the Summer 2000 course, ME 4723, entitled, "Reliability and Quality Control." All undergraduate students were upper-division students within two semesters of graduation. A large percentage of these students are minority students and all are enrolled as degree seeking students in engineering (either MS or BS). The overwhelming majority of these students are in the mechanical engineering program, although enrollment is open to electrical and civil engineering students as well.

CAPS Lab

Both courses consisted of a significant laboratory component. The Center for Propulsion Studies (CAPS) laboratory at UTSA was utilized for this project. This lab currently contains the following equipment:

2 SGI Indigo II (R8000) workstations

13 SGI O2 (R5000) workstations

2 Cd-ROM Disk drives

1 4mm DAT drive

1 Lexmark B/W laser printer (Optra S1250N)

1 Lexmark Color laser printer (Optra SC1275N)

All of the above equipment was purchased with prior NASA grants and UTSA cost sharing. No new additional computer equipment was purchased from the existing project, although, some funds were required for maintenance and supplies.

The SGI computers provide a computational laboratory with advanced graphical capabilities. This project helps ensure a high level of educational and research use of this equipment in the area of probabilistic structural analysis methods. NESSUS is currently installed and running on these computers, hence there was no additional expense for this software.

NESSUS Student User's Manual

A first version of the NESSUS Students User's Manual was written during the first year of a NASA-UTSA 1997 Partnership Award and completed during the second year of the grant. It includes a brief overview of the program, explanations of the minimum number of NESSUS keywords necessary to work laboratory example problems, explanation of output files and a set of example problems or assignments drawn from structural analysis and reliability applications. During the first year of this grant, this manual was enhanced by the inclusion of two additional example problems, as well as, detailed solutions for all of the example problems in the manual.

The first of these two new problems presents a probabilistic analysis of a simple piping system fluid flow problem. The second problem included in the revised manual is a pressure vessel design optimization problem adapted from an SwRI NESSUS Short Course problem [Southwest Research Institute, 1996]. The manual was used in both courses offered during the first year of this grant and will continue to be utilized in subsequent courses. This revised student manual is provided as a separate entity that supplements this report.

3.0 RESEARCH

Undergraduate and graduate students were supported in faculty-supervised research. Graduate student, Mr. Cody Godines, completed a probabilistic design analysis of a scuba tank while enrolled in the graduate theory course conducted last spring. His paper, which is about the redesign of a high-pressure vessel (scuba tank), is provided in Appendix II of this report. Two well-known methods of probabilistic analysis were used: Monte Carlo and First Order Reliability Method. Strength degradation and fatigue effects were taken into account. A total of six design variables were assumed stochastic. Using these two probabilistic methods, design optimization reduced the probability of failure of the system. Cody also began working on an MS Thesis topic during the past year. A number of topics were explored with emphasis on improving the tools or methods in the NESSUS program. He successfully finished the addition of the Latin Hypercube Sampling (LHS) scheme in NESSUS. LHS is a stratified sampling scheme where the statistics of the response are quantified throughout its range, not just in the region(s) of high probability. This algorithm is employed for cases when traditional reliability methods (FORM, SORM, AMV) fail to converge upon an estimate of the response density parameter (probability, mean, standard deviation, etc.). This is usually the case in ill-behaved systems. Systems with disjoint failure regions or those having irregular limit states are examples of ill-behaved systems. LHS represents another method in the suite of methods that are available to the analysts.

Undergraduate students Luis Rangel and Santiago Navarro assisted in the development of a new fluid pipe flow problem. This problem was adapted from a thermal systems design textbook [Hodge and Taylor, 1999]. The textbook describes an uncertainty analysis applied to the problem to estimate the range of anticipated behavior for a specified piping system and selected pump. The uncertainties in the piping system include: pipe lengths, diameters, bending/expansion/contraction loss coefficients, friction factor correlation, wall roughness, and

elevation changes. The pump manufacturer normally provides the nominal characteristic curve for a given pump; hence, the uncertainty was estimated to be ten percent. The probabilistic analysis was completed using NESSUS and shows the range of anticipated behavior, which is in excellent agreement with the uncertainty analysis provided in the textbook. The uncertainty analysis was a benchmark to compare our analysis and give confidence to students. The advantage of the NESSUS software was more clearly demonstrated by predicting the probability that the design system would maintain a specified minimum flow rate given all of the uncertainties in the system. The uncertainty analysis is unable to provide this information. The important parameters were also identified using NESSUS. The probabilistic sensitivity factors were found to be in good agreement with those identified by the uncertainty analysis. Another advantage of the NESSUS software is that the important parameters can be characterized throughout the range of operation, not just at the nominal operating point. For this case, the relative importance of parameters do not change significantly as a function of the systems flow rate. However, this advantage may be more prominent in other systems. Both Luis Rangel and Santiago Navarro were senior level mechanical engineering students at the time of their contribution to the fluid flow problem and graduated in December 2000.

Ronald Magharing is a sophomore undergraduate student who participated in the Alliance for Minority Participation (AMP) research program. Because of this program, Ronald was supported by the AMP program while working with those supported by this NASA grant. Ronald primarily worked with Luis for 10 weeks during the summer in the CAPS laboratory where he was exposed to probabilistic methods and tools. He assisted in completing the piping system analysis.

3.1 INTRODUCTION

Engineering Studies, Uncertainties, and Reliability

Engineers study many different types of systems using experimental, analytical, and/or numerical techniques. There are two types of studies - physical and mathematical. A physical study would occur by observing the real, physical system. A system's inputs are measured, as are responses and this is repeated for various combinations of inputs. This is often done so that a mathematical model between a system's inputs and responses can be formed.

In a mathematical study, inputs to the mathematical model governing a system can be set to certain values and the resulting response values can be calculated. A calculated response value using certain inputs should represent the response that would be observed in the real, physical system with similar inputs. This is shown in Figure 1.

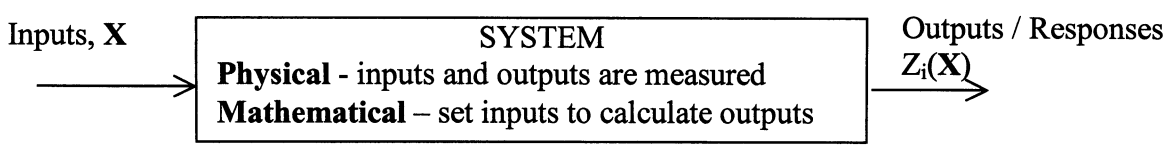


Figure 1 Two types of system studies and their differences

A system under observation generally has various responses that can be studied and that depend on many input variables. Responses can be anything from the stress or displacement of a system at critical locations, to a fatigue life, flow rates, or even measures of how well bone heals around an implant. Depending on the response, inputs can be many variables and some examples are the geometry of the system, material properties, loads, flow rates, and/or surface roughness. There is a vast amount of responses and related inputs that can be measured/recorded

or calculated/predicted. If a critical response is measured, it would be to determine if the system is good, or safe, as far as its expectations are concerned. If repeated response measurements are recorded, it would be found that the response varies. For a number of measurements, the system response will be in the safe region; however, chances are that the system will fail in a long series of measurements. Therefore, the important question cannot be: is my system safe? Rather, it is: what is the *probability* of observing a good, or safe, system response? The probability of observing safe system responses is termed the reliability of the system. Conversely, the probability of failure would be the chances of observing a system response that implies that the system failed as far as its job functions and other anticipated characteristics are concerned.

A system response will be random because the variables which is depends on are also random. In fact, during the course of a physical study, an engineer will detect a natural randomness in the inputs or system parameters, as well as in the response. Measured geometry, loading, and material properties are examples of items that will exhibit inherent variability if physically studied. At this point one might ask – why would the geometry and material properties be considered random? Justifying the question by stating that if studying one system, they would have one value. The answer is evident if we realize that engineering analyses are meant to be as efficient and as general as possible. They are meant to apply to a whole set of systems – the one being studied as well as the ones still manufactured yet not chosen to study. The geometry or material properties would be different if the experimentalist would have chosen the next one on the assembly line, or the one after that. This is an excellent way to account for the periodic replacement of certain system components. The loading on a system can also be considered random due to the fact that it will change from application to application. Therefore, many variables that a system response depends on could be any of a range of values; however,

certain values of each variable are more probable to occur than others.

We have that system responses are random and the probability that the system will be safe, or its reliability, is a desirable quantity. One can calculate the reliability in one of two ways. The first way would be to measure the system responses from the physical system. This can be next to impossible, expensive, and/or time consuming. An alternative approach is to mathematically model the system response, account for the uncertainties of the underlying dependencies or random variables, and use mathematical techniques to answer the same set of questions. The latter technique would be termed a reliability analysis. One weakness of a reliability analysis is that we must have confidence in the mathematical modeling of the system as well as in the modeling of the uncertainties of the design variables to which the concerned response depends on. There is another Achilles' heel to mention, for once committed to performing a mathematical reliability analysis, the answer must be efficiently obtained with confidence and accuracy.

The solution of most engineering responses involve computationally expensive algorithms, and accounting for uncertainties through the use of a statistical or probabilistic method requires additional computations to an already complex problem. A reliability analysis will significantly increase computational time because sometimes a single response evaluation could take hours, even days to obtain. The ideal reliability analysis would then be one that performs the fewest number of response evaluations and gives an answer to within an acceptable error limit. There are a number of different methods that can be used in a reliability analysis, each with their own advantages and disadvantages. It is up to an analyst to decide which one to use. Also, it would be ideal if the method chosen calculates low error and low effort answers.

Reliability Methods

Most Probable Point Methods

Several different methods can be used to estimate the reliability of a system. Some common methods use the most probable point (MPP) as the main step in approximating the probability of failure of the system, from which the reliability can be calculated. The system response exists over a domain of probable variables. This probable domain, characterized by a joint density function, can be approximated with normal distributions. Parameters of these equivalent normal distributions can be used to map the response and the joint density function to a reduced space. The domain point in the reduced space that implies a failed response *and* that has the highest joint probability is called the most probable point. Most probable point methods approximate the probability of failure by approximating the response using the MPP as a base point, and in the standard normal space. Using the MPP location to estimate the probability of failure will depend on the method under consideration. Table 1 shows a summary of the common MPP reliability methods as well as a description of each.

Table 1 MPP reliability methods

Most Probable Point Methods	*Description*	*Necessary Items for Reliability Calculation*	*Mathematical Comments*
First Order Reliability Method (FORM)	Hyper plane approximation of failure surface at the MPP	MPP in standard normal space	Ratio of failure region to sample space same in 1-D as in n-D
Second Order Reliability Method (SORM)	Quadratic hyper-surface approximation of failure surface at the MPP	MPP in standard normal space, and principal curvatures at the MPP	Failure surface approximated by incomplete or complete quadratic
Higher Order Reliability Method (HORM)	High order hyper-surface approximation of failure surface at the MPP	MPP in standard normal space, and necessary curvatures to fit approximate surface	Failure surface approximated using function and 1st derivatives at two points on failure surface
Mean Value (MV), Advanced Mean Value (AMV)	MPP locus technique	MPP in standard normal space	Used by FORM, SORM, or HORM for a more efficient MPP location

The most common MPP method is the first order reliability method (FORM). Obtaining the FORM solution involves approximating the response surface as a first order hyper-plane at the MPP, in a standard normal space. This method will be conservative if the approximate failure region does actually contain non-failure points. Hassofer-Lind and Rakowitz-Fiessler (HL-RF) made two separate contributions to the FORM. First, Hasofer and Lind noted that invariant calculations can be avoided if the first order approximations to the failure regions are performed at a point on the failure surface [Hasofer and Lind, 1974]. Rackowitz-Fiessler then suggested an approach to finding the MPP. This is a constrained optimization problem. The algorithm will involve finding the minimum distance from the origin of an approximate space of standard normal variables to a coordinate constrained to lie on the failure surface [Rackwitz, 1976; Ang and Tang, 1984].

Second Order Reliability Methods (SORM) approximate the response as a quadratic surface at the appropriate MPP and the probability estimate is obtained using the principal curvatures at the MPP in the standard normal-space; however, this requires additional computations to obtain second derivatives of the response [Breitung, 1984; Wu and Wirsching, 1987; Tvedt 1990]. Also, higher order reliability methods (HORM) are possible to perform but do require the appropriate amount of additional computations for gradient calculations on more than one point on the failure surface [Grandhi and Hopkins, 1997]. Also, because the probability density function in the standard normal space exponentially decays as the distance from the origin increases the HORMs would typically only be used for responses that are highly nonlinear in standard normal space. Therefore, not only are additional response evaluations required once the MPP is located, but the optimization technique used to locate the MPP might not even be successful. Other types of FORM, SORM, and HORM could be considered those that locate the

MPP in a different manner. A mean value (MV) analysis linearly approximates the response surface using the mean of the underlying random variables as the base point. The approximate surface will be exact at the mean and, in general, inexact away from the mean. A mean value solution would then require the use of a reliability method (e.g. FORM) to compute approximate MPP locations and probabilities for the various response levels; however, the surface is approximate and therefore so is this MPP locus. The advanced mean value (AMV) solution updates the response along the MPP locus and associates the previously calculated probabilities with the updated response value. If a reliability method, complete with its own optimization algorithm, is repeatedly used to update the MPP locus, a complete CDF of the response is the result. This is the methodology of the AMV+ ("AMV plus") method [Southwest Research Institute, 1995].

As shown in Table 1, there is no mention of the HL-RF transformation/algorithm combination. That is because it is commonly used with the methods shown in the table to locate the MPP and obtain a reasonable, invariant answer that includes distribution information.

Sampling Methods

Random sampling is another way to estimate probability density parameters of any measurable or computable response. Sampling is extremely robust because there are no response function constraints, i.e., differentiable, continuous, etc. that would prohibit its use. Their disadvantage is that many function evaluations are needed to confident in a low error answer. Table 2 shows some common sampling techniques.

Table 2 Sampling reliability techniques

Sampling Techniques	*Description*	*Necessary Items for Probability Calculation*	*Mathematical Comments*
Monte Carlo	Random samples from each underlying random variable	Sufficient number of samples	Computationally expensive
Latin Hyper-Cube	Individual variate space divided into equal probability bins	Sufficient number of samples	Enforce equal probability of variate sample occurrence
Distributed Hyper-Cube	Algorithm used to adjust samples for better distribution	Sufficient number of samples	Different algorithms can be used to adjust samples
Quasi-Monte Carlo	Samples deterministically generated	Sufficient number of samples	Samples more uniformly cover hypercube

Monte Carlo (MC) is the most common type of sampling technique. From each underlying random variable for which the response is dependent on, n random values are taken such that they are distributed according to what is seen in nature for that variable. The samples for all individual underlying variables are then paired to form coordinates in a generally multidimensional space that is the domain of the response. The response is then evaluated n number of times and the density parameters needed to calculate the reliability of the system or the associated probability of failure of the system can then be estimated from those response evaluations. The more response evaluations made, the more accurate the answer and the more computer time will be spent making the additional evaluations [Southwest Research Institute, 1995].

Another technique is Latin Hypercube Sampling (LHS). It is a stratified sampling without replacement in which, for each underlying random variable the response is dependent on, n random values are taken from n equal probability regions of that variable's space such that the n regions completely span the variables probable space. The values from each underlying

random variable are then paired with each other to form coordinates in multidimensional space and, finally, n response evaluations can be calculated. These response values are used to estimate the desired density parameters [Ayyub and Lai, 1991]. The advantage of LHS is that is enforces a random sampling rule that all values must have an equal probability of occurring. Distributed hypercube sampling (DHS) uses a swapping algorithm to more evenly distribute the samples throughout the probability space [Manteufel, 2001]. Quasi-monte carlo is a relatively new technique that samples points based on a deterministic, low-discrepancy sequence of numbers [Robinson and Atcitty, 1999].

As shown in Table 2, all of the sampling techniques require that a sufficient number of samples of the response be computed in order to ensure that the estimates of density parameters are close to the true values.

Hybrid Methods

Hybrid probability methods are those that use the MPP location and response sampling to obtain a reliability estimate. Some common hybrid reliability methods are shown in Table 3.

Table 3 Hybrid reliability methods

Hybrid Method	*Description*	*Necessary Items for Probability Calculation*	*Mathematical Comments*
Spherical Based Importance Sampling	Sampling forced outside of hypershpere	Sufficient number of samples and MPP location	Any hypershpere radius can be used
Adaptive Importance Sampling	Sampling around MPP with adjusted failure surface	Sufficient number of samples and MPP location	First or second order failure surface at the MPP can be used

Spherical based importance sampling uses the MPP to direct samples outside of a hypersphere, closer to the failure region. Harbitz (1986) defines a hypershpere whose surface contains the MPP. Adaptive importance sampling involves approximating the response at the MPP. If the approximate response surface is a hyperplane, then the distance to the plane is changed and an event probability can be calculated. If the response is approximated with a

parabola, then its curvature is changed, and a corresponding probability can then be calculated [Southwest Research Institute, 1995].

General Problems with Reliability Methods

The problem of concern when seeking to estimate the reliability of a system is that obtaining accurate results is a computationally expensive task. Also, each response calculation takes a certain amount of computational time. This computational time is usually the limiting factor in obtaining accurate results. The MPP methods break down if the MPP(s) cannot be found or are found in an inefficient manner, i.e., evaluating the response too many times. This can be the result of studying a response that is highly non-linear and/or contains singularities, or is implicitly defined [Wu et al., 1990]. For the sampling methods, inaccurate results are mostly due to using too few response evaluations to obtain probabilities that are far removed from higher probability areas of the response density of concern. Response values will first be calculated around the probable areas of the response density. The hybrid methods are confronted with both types of problems.

Reliability Analysis Methodology

Reliability analyses are performed in a methodical manner. For the system under study, a mathematical model of the system response is used to represent the physics of the system. The mathematical model is dependent on a number of variables. Some of these design variables are modeled as being uncertain, stochastic, or random, while others are deterministically modeled. A design variable should be stochastically modeled in the analysis when it is an important variable for that response. Important variables are those that will exhibit a high variation *and* significantly affect the response when changed. The reliability analysis process continues after the design variable values, statistics, and/or distributions are known or estimated. Design

variable statistical data may be available, but if it is not, testing should be performed that accurately measures the statistics of the appropriate design variable. This must be performed prior to the beginning of the mathematical reliability analysis. After a mathematical model has been accepted and all design variables can be correctly modeled, a reliability analysis is performed by using a known probabilistic method. The result of such an analysis is usually a complete or partial cumulative distribution function (CDF) of the response, which is used to quantify the reliability of the system and from which a probability density function (PDF) can be calculated. A reliability analysis can also identify the important variables of the response, which allows insight on possible new designs that have a higher reliability. This methodology is outlined in Figure 2.

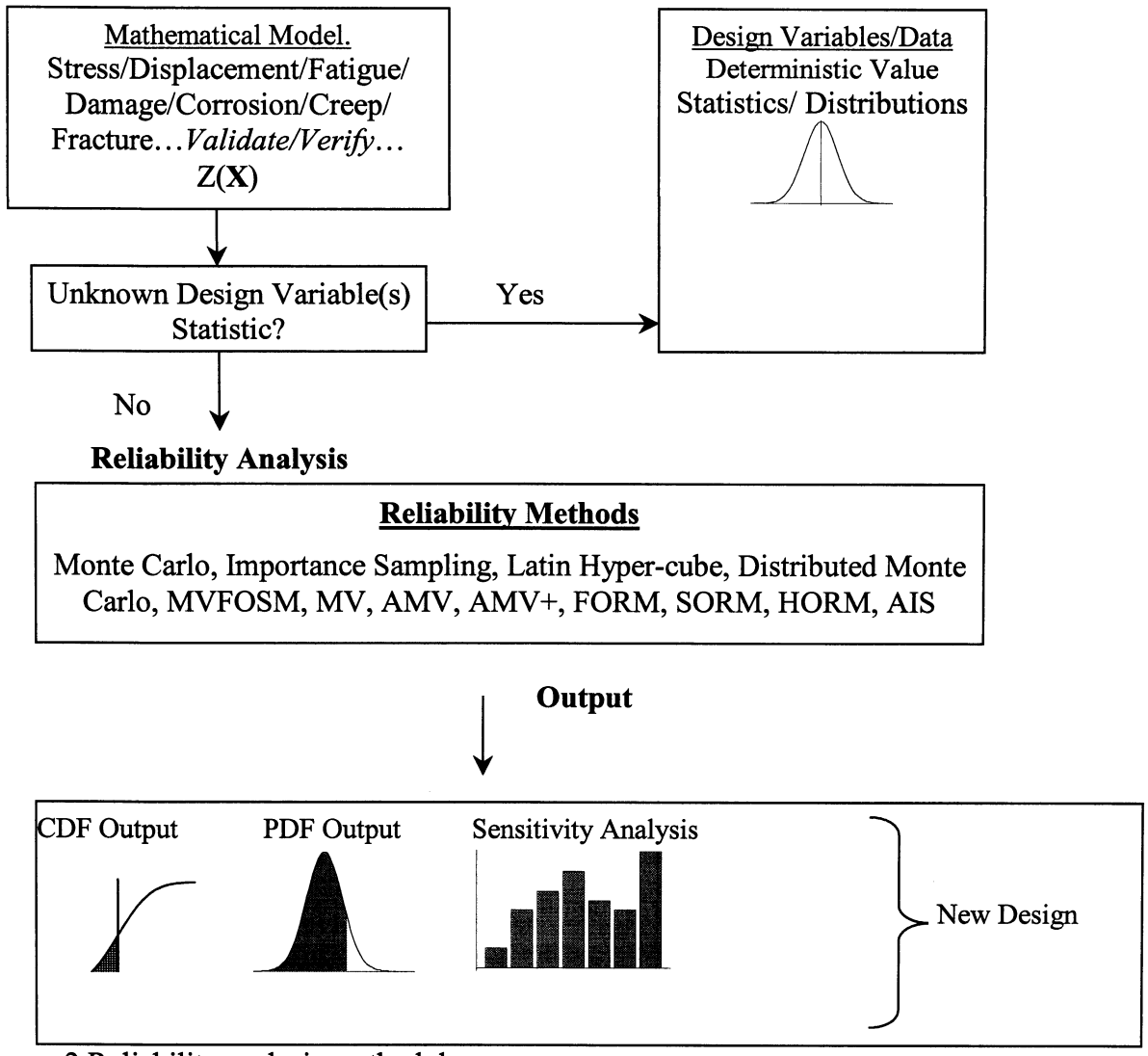


Figure 2 Reliability analysis methodology

Areas of Application

Reliability analyses of many different system and their responses have been performed. It can be shown that a reliability analysis can be a part of many diverse disciplinary backgrounds. Turbine blade responses due to uncertainties in blade frequencies, damping characteristics, and flow variations around the blades have been studied [Shah, et al. 1990]. Simulations of the human factor, i.e. marital status, in making probabilistic structural assessments have been studied [Chamis, 1993]. Probabilistic analyses of the cervical spine and a risk assessment of neck injury to female aviators have also been investigated [Thacker et al. 1997]. Using a probabilistic

method, the distribution of a composite's fatigue life when subjected to mechanical *and* thermal loading has also been studied [Shah et al., 1995]. Fluid mechanics can also benefit from the use of probabilistic analysis methods. Manteufel et al. (1997) has studied the travel time of buoyancy-driven gaseous and gravity-driven aqueous wastewater flows due to uncertainties in hydro-geological parameters. Analyzing a fluid system while accounting for uncertainties in design variables would be important if the probability of a response event (i.e., a specific flow rate) is desired. Harris et al. (2002) shows the use of probabilistic methods in the design of a fluidic system. A fluid dynamics problem containing fluid-structural interactions could also benefit from a probabilistic analysis. For example, Higgins et al. (1999) show that uncertainties present in the design variables of a fluid dynamics problem affect the reliability of the interacting structure. Basically, statistical and probabilistic methods can be used to aid in the design of any system such that a mathematical model can be formed to accurately predict the concerned response.

Purpose and Scope

The purpose of this work is to enhance the Numerical Evaluation of Stochastic Structures Under Stress (NESSUS) program with the capability to perform LHS sampling, and to compare the efficiency of LHS to that of MC, which is an existing method that NESSUS contains. NESSUS is a probabilistic finite element code that has the capability of performing reliability analysis using almost all of the different methods just discussed. The NESSUS code was developed for the National Aeronautics and Space Administration's Glenn Research Center (NASA-GRC) located in Cleveland, Ohio. The NESSUS code was developed by Southwest Research Institute (SwRI), in San Antonio, Texas. After the necessary debugging involved with an enhancement of a program, confidence in the new Latin Hypercube implementation is gained

by studying the distributions of several response density parameters as they vary with the number of samples used to obtain each estimate of the respective parameter. The density parameters estimated were the mean, standard deviation, and the 99th percentile of the response of four different test cases put forth by the Society of Automotive Engineers for the purpose of comparing different probabilistic methods. The results were compared to the same study performed using Monte Carlo sampling.

Latin Hypercube Sampling Enhancement

The scope of the NESSUS LHS enhancement was limited to the addition of seven Fortran 90 files to the existing 907 NESSUS files for the purpose of obtaining an LHS sample set, evaluating the necessary response, and estimating response density parameters. The LHS thread is organized, non-repetitive in any calculations, and is documented in this thesis. Any changes made to the source code were done for the purpose of implementing the LHS method or to improve the current capabilities of the code. Potential changes that could be made to better NESSUS were noted. All of these actions and observations that took place during this half of the work are comments in the first file in the LHS thread – lhs_main.f90. Thus, they are a permanent part of the source. They are also documented in this paper in Sections 3.4 and 3.6.

This LHS enhancement portion of the research was completed taking the following steps:

1. Obtain the source code from Southwest Research Institute.
2. Study the existing Monte Carlo thread. Follow the subroutine path, any reading from files, and any writing to files.
3. Study the Latin Hypercube method. Learn how samples are obtained and how correlations among the variables can be obtained before the sample set is used to evaluate responses.

4. Write the necessary code (add new files and change existing files) needed to perform LHS sampling.

Convergence Studies

The scope of the convergence studies was limited to repeatedly estimating the mean, standard deviation and 99th percentile of the response of 4 test cases using a various number of response evaluations and two statistical methods – Monte Carlo and Latin Hypercube. By repeatedly estimating those density parameters the variation of repeated estimates about the exact or true value is captured. Given an appropriate amount of response evaluations, the distributions of all three parameters are centered about the exact value; therefore, the variation of repeated estimates is an important comparison quantity. Confidence in single estimates of each parameter using each method was used to compare MC to LHS.

There were two types of confidence measures that were used to compare MC and LHS. For the first type, the estimation error from the exact (assumed) parameter was compared for MC and LHS for a specific number of response evaluations and at the fifty percent (50%) confidence level. The second type of confidence statement was the comparison of MC and LHS in terms of the number of calculations necessary for 99.7% of repeated parameter estimates to be within a specific estimation error that varied from test case to test case and from density parameter to density parameter.

The test cases are part of a set problems put forth by the Society of Automotive Engineers (SAE) G-11 Probabilistic Methods Committee. They have been compiled over the years from the probabilistic mechanics community in order to compare probabilistic algorithms and reveal both advantages and disadvantages.

This convergence study portion of the research was completed taking the following steps:

1. Obtain the test cases from Southwest Research Institute.
2. Obtain any literature that performs a probabilistic analysis on any of the test cases.
3. Write the input (*.dat) files for NESSUS to use during the MC and LHS runs.
4. Code up the response functions in Mathematica 4.0 (Wolfram Research, 1996) and compare calculations to those of a NESSUS MC or LHS run.
5. Perform NESSUS runs for all test cases.
6. Plot results and draw conclusions from observations.

Organization

This thesis is organized to present the necessary background and the results of the numerical test cases in a manner that allows the reader to understand all the concepts and results that will be talked about. The section 3.2 is a background on using statistics to obtain density function from data. Basic statistics and response function concepts and are discussed from an engineering reliability point of view. Estimation is the topic of section 3.3, where the similarities and differences of Monte Carlo and Latin Hypercube Sampling are discussed. Next, and still in section 3.3, using MC and LHS to obtain estimations of the mean, standard deviation, and 99th percentile from response data is discussed. After which, the general topic of using an estimator, that is itself a random variable, to obtain estimates of density parameters is discussed. NESSUS is the topic of section 3.4. Its present state is introduced and then its Monte Carlo capabilities, inputs, and outputs are discussed. The new Latin Hypercube module is introduced first by discussing the capabilities, and the input and output files. Section 3.4 continues with a discussion on the method used to obtain the necessary correlation between the variables for which a response is dependent on. The section is finished after a discussion of the necessary

changes made to the original source code. Section 3.5 compares the convergence to a low variation of the distribution of means, standard deviations, and 99th percentiles using MC and LHS with an increasing number of response evaluations for four SAE test cases. An estimator with a low variation will imply that there is a greater probability for a single estimate being within the same error interval when compared to the probability associated with an estimator distribution that has a high variation. The error in estimation along with the effort required to obtain accurate results are the decisive measures used to compare MC and LHS. Results were obtained from the existing capabilities of NESSUS, as well as, the new LHS capabilities of NESSUS. When needed, computational checks and graphics were obtained using Mathematica 4.0 [Wolfram Research, 1996]. Finally, section 3.6 contains a summary of research findings as well as specific conclusions drawn.

3.2 RELIABILITY AND THE DENSITY FUNCTION

The probability at which certain response values will be observed can be a useful tool in engineering analysis, design, or marketing/production of a system. The reliability of a system can be quantified by estimating the probability of observing "safe" system responses. For the most part, systems are designed so that values of various responses a system can have – displacement, stress, temperatures, accelerations, etc., are expected to be in a range of safe values. However, due to randomness in variables like loads, which will vary from application to application, or geometry, which will vary because repeatedly manufactured products are not exactly alike, a system will exhibit a variation in its different responses. Unfortunately, some of the systems will fail; and, so, it is no longer enough to state that a system is expected to be safe. It is necessary for many analysts, designers, and manufactures to state that their product is, for example, 99% reliable. That could mean that 1 out of every 100 like products manufactured could fail at their duties or that 1 out of every 100 applications of a single product will result in the failure of that product. Depending on the product, a statement like that can be a selling point, or a reason to go back to the drawing board. In either case, the reliability of a system can never be stated, or calculated, if the uncertainties in the variables for which the response is dependent on are never accounted for.

In order to calculate the reliability of a system, the uncertainties in the underlying design variables that govern a response need to be mathematically modeled. A response with stochastic dependencies will itself be a random variable, and, because of this, the probability that the response will be safe can be estimated through the use of a variety of probabilistic methods. Typically, a random response will be characterized by its probability density function, or just density function, which itself is defined by its many parameters. At least three of these density

parameters can be used to obtain the reliability of the system. The first type of desired parameter is a measure of central tendency of the response and is called the mean of the response. Another parameter of interest is a measure of the average spread of the response about the mean, or expected value. Yet another response density parameter is a response proportion, or ratio. The proportion parameter is the ratio of the number of responses that would be observed to lie in a certain range(s) of the response, or bin(s), to the total number of response measurements, or calculations, after a large number of response observations have been made. All three of these parameters can be used to evaluate the reliability of a system.

Two major dilemmas are encountered when attempting to calculate the reliability of a system. The first is that for most practical responses that are studied, their density and hence their parameters can never be exactly known. The density and parameters of a response can only be estimated. This section will discuss estimating the density of a response based on a number of measurements. The topic of section 3.3 will be random sampling and estimating parameters of the density of a response based on a number of measurements. It will also discuss how estimators, being a function of random variables are also random and they will have a density associated with them that is, for a good estimator, centered around the true density parameter. The second major problem confronted with when performing a reliability analysis is that only a certain amount of computer time can be spent on the necessary calculations. Fewer response evaluations implies less computer time that many response evaluations. Section 3.5 will be the comparison of the efficiency, in terms of response evaluations, of Monte Carlo and Latin Hypercube Sampling when they are used to estimate the mean, standard deviation, and the 99th percentile of several stochastic responses. The 99th percentile parameter is related to the response proportion parameter and is used in this efficiency study because it is already known

that it will be difficult to efficiently estimate and will therefore bring out the differences in the two methods that are compared.

Very important quantitative statements about response events can be made when the uncertainties of the underlying random variables that the response is dependent on are accounted for and used in a reliability analysis. This type of probabilistic analysis allows an analyst to make probability statements about observing safe system responses when we realize that system inputs, like loads, are random and that the system itself will not be exactly the same if repeatedly manufactured. In order to compute the reliability of a system its density, a characteristic of all random variables, needs to be estimated. This section will discuss random variables and estimating their density function based on a number of measurements.

Random Variables

We begin with a discussion of random variables, their origin, and how probability statements can be made from data or a continuous fit to data. Suppose n measurements, or observations of a variable, are recorded and displayed in a manner similar to Figure 3. This variable can be anything, from a geometric length, to pressure loads, to crack sizes. We will assume for the sake of discussion that the measurements in Figure 3 are the crack sizes of 200 different systems. Therefore, the crack size, a_i, is a variable that is shown to be random. Also, the "i" subscript in a_i is to indicate that it is an "initial" crack size. All of the recordings lie between 0 and 0.03. The data is spread out in the vertical direction of the plot only for clarity. The dark region around 0.008 is an indication that the most values recorded were in that region. This group is the sample space of crack sizes based on 200 measurements of 200 different systems. It is a sample of a whole population of possible crack sizes on systems that by chance were not purchased for the sake of the measurements or that have yet to be manufactured. These

values must be used to estimate the density of the population of all possible crack sizes. It can also be said that any mathematical manipulations of the data are done for the sole purpose of trying to estimate the density of the population. The density calculated using the sample measurements is merely an estimate of the density of the population. Both are densities of the crack size random variable only one is an estimate and the other in never obtainable so we must make sure that the estimated density captures information about the population of values, not just the information of the sample of measurements.

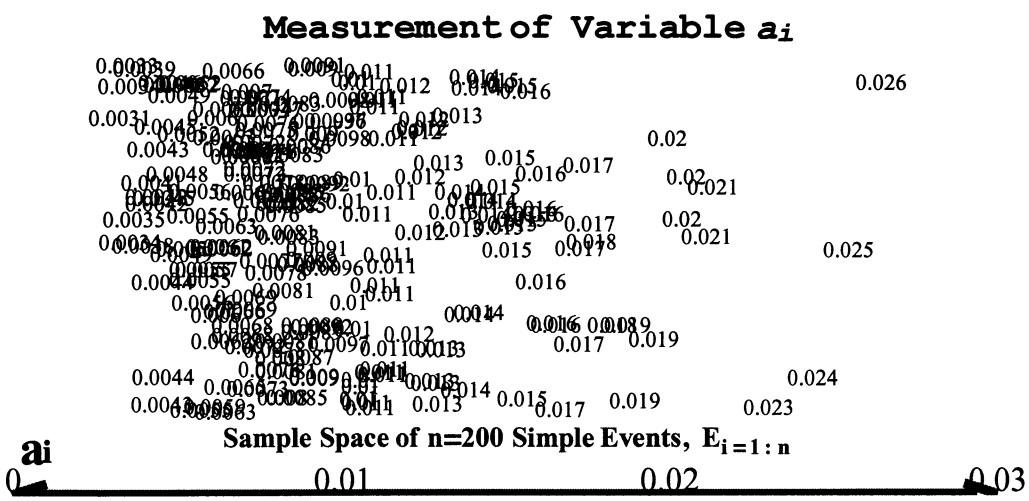


Figure 3 Different numerical values recorded while measuring variable a_i

Each individual observation, or sample point, is a simple event, E_i. It cannot be decomposed into simpler events. The probability of each simple event can be calculated according to the relative frequency concept of probability. This probability must be a measure of one's belief or expectation in the occurrence of that event during one future observation, measurement, or experiment. Now, it is known based on the measurements made that these crack size values do exist. In assigning the probability to each simple event, it is assumed, for

the moment; that these values are actually the population and the probability of each simple event is obtained by conceptualizing that all the n=200 systems were randomly mixed and we want the probability of obtaining each of the crack size values during one experiment. This experiment is a random selection and observation of one crack size value in the sample that is assumed to be the population. In assigning probabilities to each simple event, they must adhere to certain axioms of probability, which are given in Equations 1, 2, and 3 [Wackerly, et al. p.27].

Axiom 1. $P(A) \geq 0.0$ (1)

Axiom 2. $P(S) = 1.0$ (2)

Axiom 3. If A1, A2, A3,…,A_n
are pairwise mutually exclusive events in S, then

$$P(A_1 \cup A_2 \cup A_3 \cup \ldots \cup A_n) = \sum_{i=1}^{n} P(A_i) \qquad (3)$$

For any experiment with S as its associated sample space, and for every event A in the sample space, the probability of A, P(A) is assigned such that those three Axioms are true. The sample space, S, consists of all of the n=200 measurements made, and the event, A, is considered to be a simple event – it consists of one and only one of the measurements already made. The experiment would be to randomly draw one value from the group. Because there are 200 *different combinations* of one-draw experiments from the group of measurements of Figure 3, we can say that the sample space size, n, is 200. Now, we can assign the probability of each simple event in the group of measurements the numerical value of $P(E_i)=1/n$, for i=1,2,…,n. This

probability is a relative frequency. The probability is the frequency of the event, E_i, which is 1, relative to the total number of possible observations, or the size of the sample space, n=200.

Note that all of the three Axioms are adhered to. That is, each event, E_i, has an assigned probability that is greater than or equal to zero. For the second axiom, consider the sample space, S, which consists of all events in the sample space. In other words, it is the union of all simple events - $S = E_1 \cup E_2 \cup E_3 \cup ... \cup E_n$. The operator $\cup$ is the commonly used union operator in set notation and it is also known as the 'OR' operator. Therefore, the sample space, S, is the set of all events given by E_1 *or* E_2 *or* E_3 *or…or* E_n. These simple events are pairwise mutually exclusive. This is determined by considering any two events, E_j and E_k, which represent two different measurements of a variable; hence, they are simple events. The events have nothing in common and observing one event will not imply the observation of the other. If this can be said about all possible event pairs, E_j and E_k, then all of the simple events for which the sample set, S, is composed of are pairwise mutually exclusive. The probability of observing any in the set of values that make up the whole sample space can therefore be determined using Axiom 3. The probability of observing any value that is a part of the sample space, S, is given by $P(S) = P(E_1 \cup E_2 \cup E_3 \cup ... \cup E_n) = \sum_{i=1}^{n} P(E_i) = \sum_{i=1}^{n} \frac{1}{n} = 1.0$. This agrees with Axiom 2. Thus, we have a function, or mapping, from a response value, or event, to its probability, or relative frequency, of occurrence. The probability function for the group of crack size observations is shown in Figure 4.

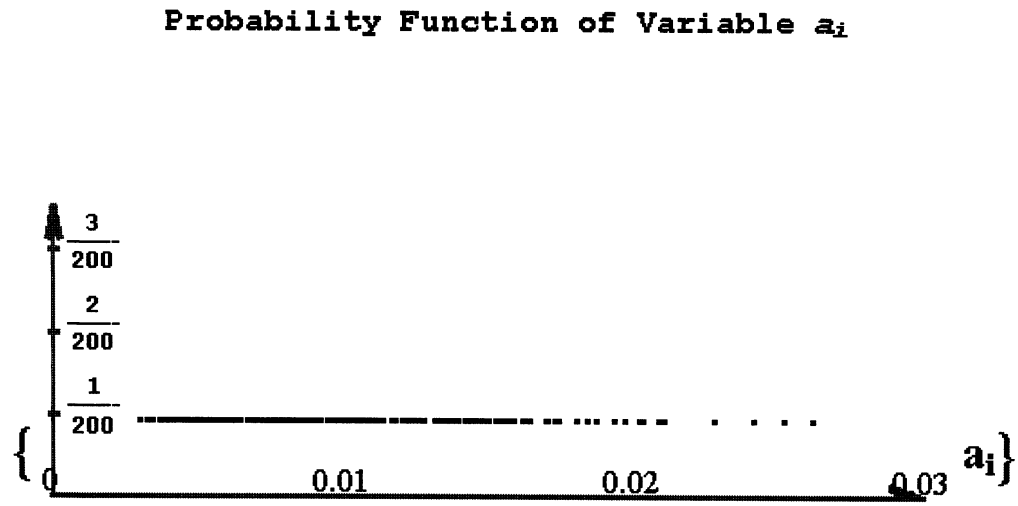


Figure 4 Probability function for simple events of the crack size variable, a_i

All of the simple events in Figure 4 have an equal probability of occurring. Each event has a 1/200 or 0.5% chance of occurring if one crack size measurement were repeated at *random* from the same 200 different parts that the measurements were originally made from. These 200 systems are one sample set that is used make judgments about the all of the systems that have already been manufactured, or have yet to be manufactured. All of the possible systems and their crack size measurements are known as the population of crack size values. The probabilities of all of the events that exist in the population can never be completely known because of the large, sometimes infinite, number of measurements that would have to be made in order to collect the necessary data. Probabilities about a future manufactured part, or a new measurement to be made from a recently purchased part, are estimated from what was observed in making the 200 measurements. However, the representation of the data in Figure 4 is poor in that the probability information cannot easily be extrapolated to find the probability of observing a value that was not originally observed. The frequency of observation of each simple event has already been normalized with respect to the size of the sample space; however, there does exist a better representation of the data with better extrapolation properties. It is obtained when the simple events are gathered to form mutually exclusive compound events.

Compound events can be decomposed into simple events. These types of events are made up of the union of simple events. A compound event can be the observation of measurements that lie within a range of possible values, e.g. from 0.5 to 1.0, or 1.0 to 1.5. Fortunately, the Axioms given by Equations 1, 2, and 3 apply to generic events, A_i (not related to a_i), which can be simple events or compound events. Compound events can be formed in any manner; however, it would help if they are orderly, mutually exclusive and completely span the range of values of a variable. Figure 5 shows bins that are the compound events of the crack size response.

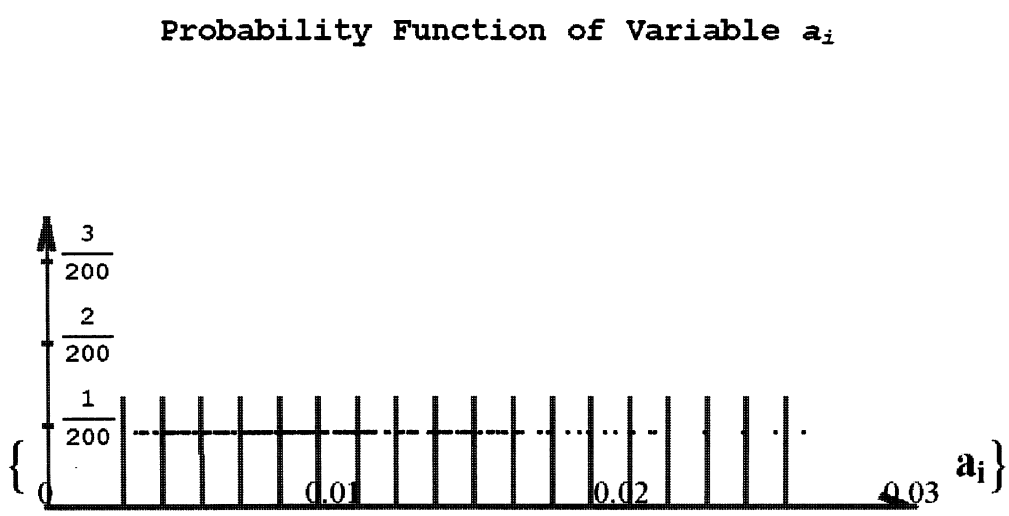


Figure 5 Compound events and the simple events that they are composed of

The bins are separated by a vertical line for clarity. The compound events are composed of simple events. The important next step is to determine the respective probabilities of the compound events. These probabilities are calculated using Axiom 3 shown in Equation 3. The result of the application of the axiom is given in Equation 4.

$$P(A) = P(E_j \cup E_{j+1} \cup E_{j+2} \cup \ldots) = \sum_{j=1}^{n_{E_j \in A}} P(E_j) = \sum_{j=1}^{n_{E_j \in A}} \frac{1}{n} = \frac{n_{E_j \in A}}{n} \tag{4}$$

Thus, the probability assigned to the observation of event A that is composed of the union of mutually exclusive simple events, E_j, is determined by summing the probabilities of each simple event in the event, A. In other words, the probability of observing a range of crack size values, which is a compound event, is the summation of the probabilities of observing each value in that range, which are simple events. The result is the proportion, or ratio, of the number of simple events that are elements of the compound event A, $n_{E_j \in A}$, to the size of the sample space, n. The desired discrete probability function, or rather all of the probability information that can be obtained from the original data, is better represented by the plot of Figure 6, obtained when the range of measured values is divided into non-overlapping, mutually exclusive bins, and the probability of these compound events are calculated according to Equation 4.

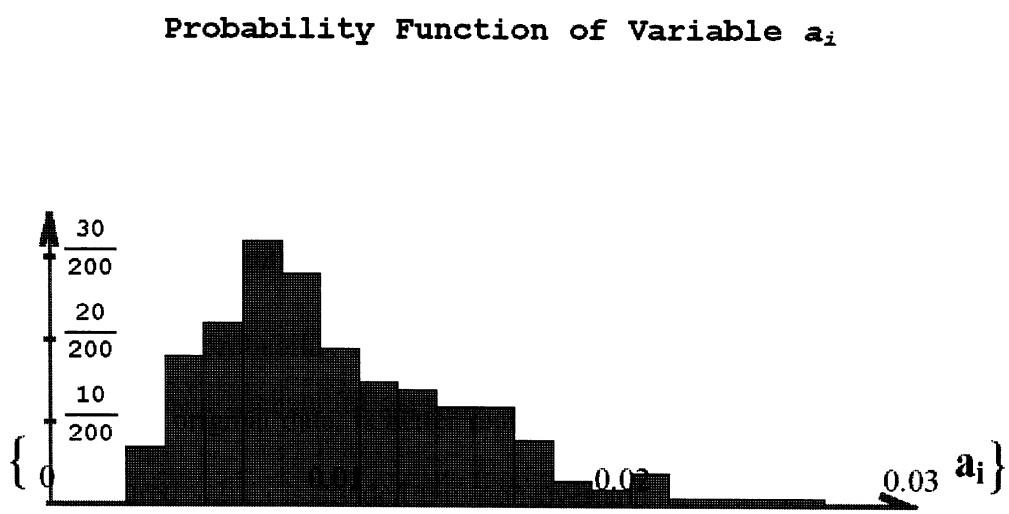


Figure 6 Probability function for compound events for the crack size variable, a_i

The original n=200 measurements that made up a sample of the whole population of crack size values has thus been extrapolated, using compound events, to obtain the probabilities

associated with values that were *not* a part of the original sample of measurements. We can conclude that forming compound events and calculating their probabilities extrapolates the original data for the purpose of making approximate probability statements about the population of values. Consider the implied meaning of the word extrapolate, yes, some of the crack size values that were not a part of the original data and for which we now have a probability associated with them are in between the original data values when visualized against an ordered scale as shown in Figure 6. This would be mathematical interpolation of data because we are estimating the probability function between at least two known values of the probability function. However, the original data is part of one sample *set* for which we know their probabilities and the population of all values in *not a part of this set* and therefore, the known probabilities are extrapolated outside of the original sample set. In this case, we assume that the estimated probabilities of the population of all values follow logically from the known probabilities of the sample set.

In the original 200 observations of a_i, the probability of each simple event was able to be determined and those probabilities followed the three Axioms of probability. The probabilities are used to measure our belief in future events of the original n=200 observations based on those same observations, or measurements already made. Zero probability is assigned to events that were not part of the original set of measurements and that are part of the whole population of possible crack size values. Yet, just because a value was not observed does not imply that its probability of occurrence is zero. In fact, it is likely that the probability of an unobserved crack size value will be close to that of the probability of an observed value so long as the two crack size values are close to each other. Of course, this "closeness" is a relative measure and must be small compared to the range of probable values. If the distance between two crack size values is

on the order of the range of probable values then we cannot assume that their associated probabilities are close to each other because the probability function can change dramatically from the beginning of a range of crack sizes to the end. In any case, gathering simple events to form compound events helps to extrapolate the probabilistic characterization of the original measurements to other measurements that were not part of the observed data, but, do have a probability of occurrence. Figure 6 is thus a better representation of the probability function of the original data because it allows future, yet unobserved events to be associated with respective probability values.

Also, because the probabilities are summed within each bin, the range of the probability function shown in Figure 6 is from 0 to about 30/200 and is greater than the range of the probability function shown in Figure 4 or Figure 5, which is from 0 to 1/200. The right tail of the probability function of Figure 6 contains 4 compound events, bins, or range of crack size values, each containing one simple event. This can be verified by observing those bins in Figure 5. By extrapolating the results of the original data we assign the same 1/200 probability to a future observation of the crack size, a_i, that will be any of the values in each of those bins. Since all four compound events are mutually exclusive, Axiom 3 can be used to determine that the probability that a future observation of a_i will result in a value that lies in *any* of the four bins 4/200 = 2%. We might either be content or a little confused at that last statement. Let us discuss this further while simultaneously obtaining the reason why Figure 6 is still not the best representation of the data so don't readily accept its simple interpretation.

Consider the last bin on the right of the probability function of Figure 6. For the sake of numbers, let us say that the bin range is from 0.0255 to 0.0265. From Figure 6 we can conclude that the probability that a single measurement of a crack size, a_i, will be any of the many values

between 0.0255 to 0.0265 (inches) in a future observation is 1/200 (0.5%) based on the 200 original measurements made and manipulated to apply to the whole population of probable values. The value of a_i that did land in this range in the original measurements is about 0.026. If all of the original set of parts from which the measurements made are randomly shuffled, and one part is randomly picked from that mix, the probability of picking the part with a crack size of $a_i = 0.026$ is also 1/200 (0.5%). It is important to realize that these are not the same experiments. The equivalent experiment to the probability information of Figure 6 is one in which all of the parts that could be purchased along with those that have yet to be made but still have a probability of being made are randomly place in a large room and one is randomly picked from this mix so that the crack size can be measured. This room represents the whole population of parts with associated crack sizes. The probability that the single crack size measurement will be between 0.0255 to 0.0265 inches is 1/200 (0.5%) which should agree with the mathematical manipulation of the original n=200 measurements. Also, this hypothetical random shuffling and selection is mentioned because random sampling, by definition, occurs when each of the values has an equal probability of occurring [Wackerly, et. al. p.67]. It is a fundamental rule of random selection. If they were not randomly shuffled, and were moved such that the high crack size parts were always in front of a blindfolded selector then the probability of observing certain crack sizes would definitely be different than in the case of random mixing and selection. Random sampling via computer simulation of response measurements will also be discussed in section 3.3.

Figure 6 represents probabilities associated with the whole population of crack size values, which can come from parts that were, by chance, not purchased or even those that have

yet to be manufactured, but still have a probability of occurrence. Probabilities can be calculated that apply to the original data, but it is more practical to be able to calculate the probabilities of any possible measurement. In short, the probability of observing the compound event where a_i is in between 0.0255 and 0.0265 is obtained by summing the probabilities of the mutually exclusive simple events which it is composed of, as given by Axiom 3. This probability is calculated to be 1/200, and this interpretation of Figure 6 also agrees with Axiom 1 (non-negative probability for each event) and, more importantly, Axiom 2 (probability of all values is 1.0 or 100%). This is *not* the probability of each event, so one cannot say that the probability that a_i = 0.0259 is 1/200, or that the probability that a_i = 0.0261 is 1/200 (these are two arbitrary values between the example range of 0.0255 and 0.0265). As there are an infinite number of values in the example range, statements like this don't agree with Axiom 3 and will result in probabilities that are over 1.0 or 100%, which is impossible. A probability greater than 100% would be the equivalent of someone measuring crack sizes and saying that out of 200 measurements, 247 of them (124%) were recorded to be within a certain range.

There are several methods for storing all of the probability function information. One way would be to keep all of the original data and use it for computing probabilities. In that case, the major drawbacks are that a lot of storage space would be used to store the data and that computations need to be performed on the data to get the necessary probability information. Another way would be to store the discrete function of Figure 6. This would amount to storing the bin range and its respective probability, or relative frequency of occurrence, for all bins that make up the complete range of the variable. This is not a bad idea; however, for Figure 6, Equation 5 would be the discrete function that would need to be stored in order to predict future

occurrences of the variable a_i. This method of information storage is better than the first method in that the probabilities are already computed and the original data does not need to be kept. Additional computations would be necessary if the probability that the crack will be within a bin not shown in Equation 5, like a bin consisting of two and a half of the bins shown, or a bin that is a subset of the ones shown. For example, it can be *assumed* that the probability 7/200 for the bin $0.00270 \leq x < 0.00405$ is equally distributed throughout that range; that is, all of the values in that range have an equal probability of occurrence. Therefore, if we divide the range into seven equal bins we can state that there is a 1/200 probability that the crack size will be in the first sub-bin of $0.00270 \leq x < 0.00289$. The same can be said about the second sub-bin, the third, and so forth, up to the seventh sub-bin of $0.00385 \leq x < 0.00405$. That is fine, however, these are extra steps that can be performed before the data is stored.

$$f_{a_i}(x) = \begin{cases} 0 & x < 0.00270 \\ 7/200 & 0.00270 \le x < 0.00405 \\ 18/200 & 0.00405 \le x < 0.00540 \\ 22/200 & 0.00540 \le x < 0.00675 \\ 32/200 & 0.00675 \le x < 0.00810 \\ 28/200 & 0.00810 \le x < 0.00945 \\ 19/200 & 0.00945 \le x < 0.01080 \\ 15/200 & 0.01080 \le x < 0.01215 \\ 14/200 & 0.01215 \le x < 0.01350 \\ 12/200 & 0.01350 \le x < 0.01485 \\ 12/200 & 0.01485 \le x < 0.01620 \\ 8/200 & 0.01620 \le x < 0.01755 \\ 3/200 & 0.01755 \le x < 0.01890 \\ 2/200 & 0.01890 \le x < 0.02025 \\ 4/200 & 0.02025 \le x < 0.02160 \\ 1/200 & 0.02160 \le x < 0.02295 \\ 1/200 & 0.02295 \le x < 0.02430 \\ 1/200 & 0.02430 \le x < 0.02565 \\ 1/200 & 0.02565 \le x < 0.02700 \\ 0 & 0.02700 \le x \end{cases} \tag{5}$$

Further manipulation of the data leads to a better way to store all of the probability information. The next step would then be to normalize each probability, or relative frequency, by the bin width associated with that probability. In doing so, the assumption that in all individual values of the crack size have equal probabilities of occurring is enforced; and, the result is known as a probability density function (PDF). The PDF for the crack size variable is shown in Figure 7. The probability density function is just that – it is a measure of how much of the probability is encountered per unit, or volume, of the crack size space.

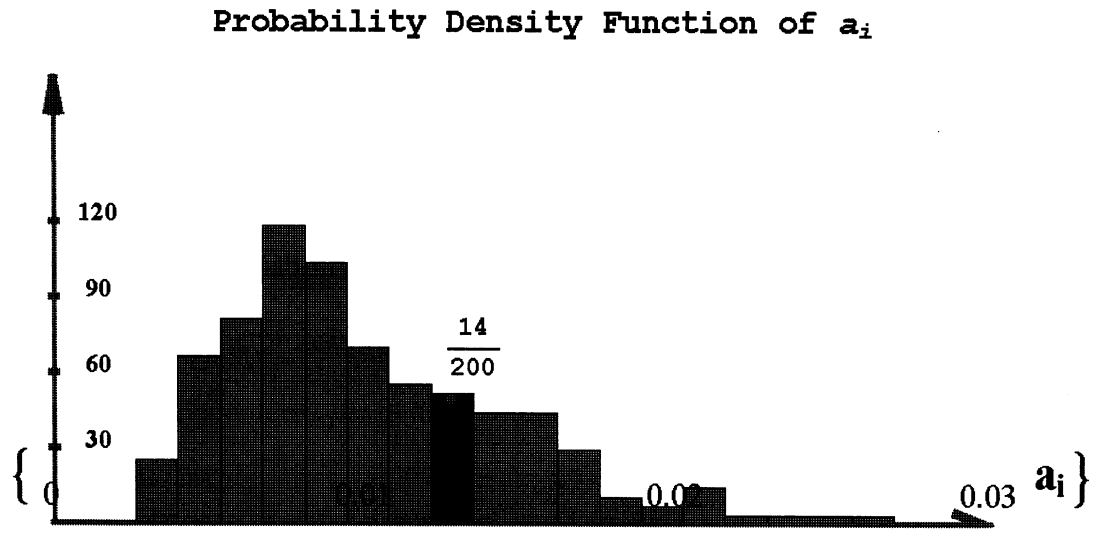


Figure 7 Probability density function of measured variable a_i

Notice that the shape of the probability function of Figure 6 is the same the shape of the PDF in Figure 7. This is because all bins were of equal width: 0.00135. Also, the scale for the probability density function has increased because the relative frequencies, which varied from 0/200 to 32/200, are each being divided by a number less than one; hence, the scale of the probability density function *for this case* is greater than the scale for the probability function. Characterizing the crack size data with a PDF is a better method because a curve can be fit to the discrete function. This allows a single equation to represent the probability characteristics of the response of interest. This will be further discussed immediately after this short summary and statement on recapturing the probability information.

So far, crack size data was collected from measurements of 200 different systems – bars, beams, rods, etc, and it is accepted that systems with these crack size values do exist because we did measure those values. Systems with other crack sizes truly or conceptually exist outside of this small *sample set* of systems obtained from a manufacturer. After all, one more system can be purchased, the crack size can be measured and it is possible that the value is one not yet observed. Or maybe, of all the systems made by a manufacturer, there is one crack size value

that does not yet exist; however, it is possible that the next manufactured system will have a crack size of that value. The set of all possible values are part of a larger body or set of data known as the *population*. The probabilities associated with all values in the population are desired and we estimate this information from the sample. We first obtain the probabilities associated with each simple element in the sample such that each simple event, or crack size value has an equal probability of occurrence. This probability information is extrapolated to the population by forming compound events and computing the probability of observing a value within each compound event, or bin. Now, the only way to properly ensure that the probability, or relative frequency, is equally distributed throughout the compound event, or bin range, is to divide the associated probability by the respective bin range. This is done for all bins and we arrive at the discrete probability density function of Figure 7. Therefore, in each bin, we have a measure of how much of the probability mass is contained per unit volume of the sample space.

Now, the probability of any range of values of a_i occurring in a future measurement is calculated by obtaining the volume between the probability density function and the zero plane of the domain. The PDF of Figure 7 exists over a 1-D domain and the probabilities are calculated by obtaining the area under the PDF over any region(s) of interest. Since, for this case, the probability density function is a measure of the probability per unit of length, obtaining the area is a matter of multiplying the PDF by the length, or range of the variable under consideration. For example, the probability density function value for the 10th bin of Figure 7, which is highlighted, is 51.8519 (the first two bins have zero probability density values). The range of this bin is from 0.01215 to 0.01350 inches, which is a bin width of 0.00135. Calculating the area under the PDF in this region is a matter of multiplying 51.8519 by 0.00135, which is equal to 14/200. This value is shown in Figure 7 and also in Equation 5; hence, calculating the

area under the PDF over a certain range of crack size values is the method used to get the probability of those values occurring in future observations.

As previously mentioned, this form of transforming the original n=200 measurements to characterize the probability of any crack size value occurring is ideal because now a curve can be fit to the discrete data. The curve fit for this data is shown in Figure 8. Fitting a curve to this discrete function makes more sense in terms of a mapping from a domain of response, or crack size values to a range of probability density values. This is because each crack size value, as many as there may be in a continuous interval containing all probable values, can have its own probability density value. The major requirement for this PDF to be a good representation of the probability of observing certain crack size values is that it needs to obey the 3 Axioms of probability. The first axiom of nonnegative probabilities is obviously true since the PDF in Figure 8 shows only positive probability density values. The second axiom naturally states that there should be a 100% probability of observing all events. For the *discrete* PDF, it can be readily shown that this true. Consider the PDF, which is obtained from the probability function by dividing each relative frequency value by its respective bin width; and that all bar, or bin widths are equal to 0.00135. By obtaining the area under each rectangular bar, we are multiplying the PDF bar value by the bin width; therefore, reversing the process and obtaining the same probability function for which the numerical values were shown in Equation 5. Now, all of these bins are mutually exclusive compound events, so the probability of observing an event consisting of the union of all events, or bins shown in Figure 8 is obtained using Axiom 3. This amounts to summing all of the probability, or relative frequency values of Equation 5. Once this is done we find that the probability of observing any of the whole range of crack size values is 100% for the discrete probability density function.

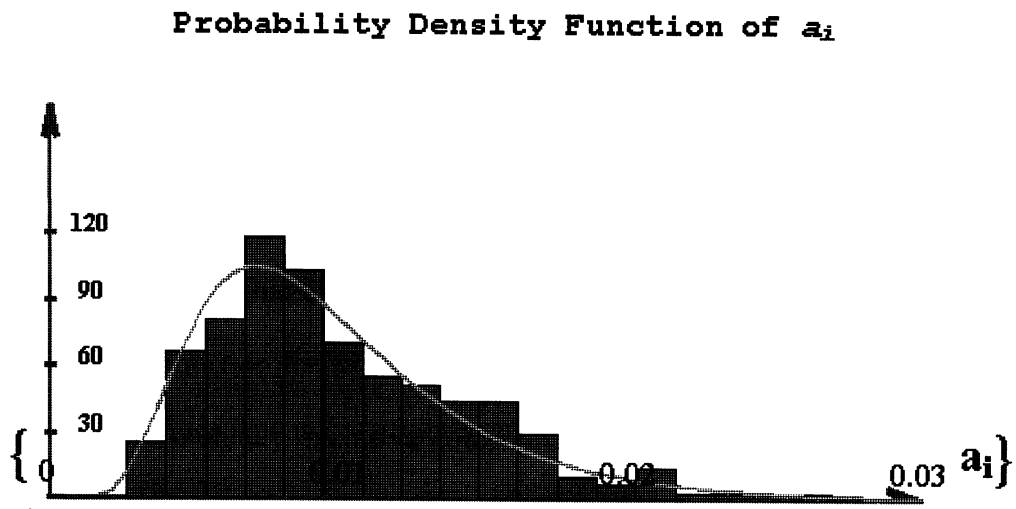


Figure 8 Discrete and continuous probability density function of variable a_i

Obtaining the area under the continuous PDF in Figure 8 is not necessarily a difficult task for this continuous function that exists over a 1-D domain because it amounts to integrating the function over the concerned values of the crack size, a_i. The problem lies in *obtaining* a curve fit that ensures that the probability of observing any of the possible crack size values in a future measurement is indeed 100%. This is Axiom 2 given in Equation 2. The fundamental reason that a curve cannot be fit to the data of Figure 6 is because that would imply that each crack size value has its own probability associated with it when in fact that is not the case. A range of crack size values has one probability of occurrence associated with it. If a curve were fit to the data and interpreted incorrectly, probabilities over 100% could be mathematically calculated based on the curve fit. While this is not going to be a discussion on distribution (PDF) selection, it can be said that there are a number of PDFs, given by mathematical equations, for which it has been ensured that they obey the laws of probability. A continuous PDF, or just a PDF from here on out, can be of any form. They usually have parameters associated with them that place the probable values in a certain region of the space of real numbers, and that refine the shape of the fit to match the form of the discrete curve that we are trying to simplify. The PDF for the curve in Figure 8 is given in Equation 6.

$$f_{a_i}(x) = \frac{e^{-\frac{\left\{\ln x - \left[\ln \mu_{a_i} - 0.5 \ln\left(1 + \left(\frac{\sigma_{a_i}}{\mu_{a_i}}\right)^2\right)\right]\right\}^2}{2 \ln\left(1 + \left(\frac{\sigma_{a_i}}{\mu_{a_i}}\right)^2\right)}}}{x\sqrt{2\pi \ln\left(1 + \left(\frac{\sigma_{a_i}}{\mu_{a_i}}\right)^2\right)}} \tag{6}$$

PDFs are usually given by $f_{a_i}(x)$, where the subscript a_i denotes the variable whom the PDF, given by $f(x)$, belongs to. The variable x is a dummy variable that is merely the input to the function mapping. The PDF of Equation 6 is known as the Log-Normal distribution. The Log Normal distribution shown is a two-parameter distribution. Its two parameters are the *mean*, μ_{a_i}, and the *standard deviation*, σ_{a_i}. For this distribution, the mean is a location parameter and the standard deviation is a shape parameter. One trait of the *two*-parameter Log-Normal distribution of Equation 6 is that negative values have no probability of occurrence [Southwest Research Institute, 1995].

Recalling Figure 8, it can be seen that the probable values of the crack size are found in the region of the domain from 0 to 0.03. If we wanted to obtain the probability that a future measurement would be any of these values in this domain, or a subset of values like any value below 0.015, or any value between 0.01 and 0.02, we'd have to integrate the PDF over these regions as needed. Fortunately, there does exist another representation of the probability characteristics of the crack size random variable that allows information like that to be easily read from a graph. This representation is called a cumulative distribution function (CDF), and can be directly obtained from the PDF using the transformation shown in Equation 7.

$$F_{a_i}(x) = \int_{-\infty}^{x} f_{a_i}(t)\,dt \tag{7}$$

The CDF at any point, x, integrates the PDF from negative infinity up to that point; therefore, the value of the CDF at any point, x, is the probability of the future measurement of a crack size value that will be below x. The probability, p, can be any value between 0 and 1. Multiply this probability by 100 and we arrive at the probability in terms of percent, 100p%. The 100pth *percentile* of the density, f_{a_i}, is the value of x, for which $F_{a_i}(x) = p$. Using Equation 7 to estimate a CDF would result in a continuous function.

The CDF, or rather an estimate of it, can also be obtained directly from the original n=200 measurements, in which case, a discrete CDF would result. There are several methods to obtain a discrete CDF and we will briefly mention one common method that uses the original n=200 simple events. First, the simple events should be sorted in ascending order. The value of the CDF for the lowest crack size value would be 1/200, and its value at the largest crack size value would be 200/200. In general, its value at the jth crack size observation would be j/200. This implies that the CDF ranges from 0 (0%) to 1 (100%). The PDF and CDF for the crack size variable are shown in Figure 9.

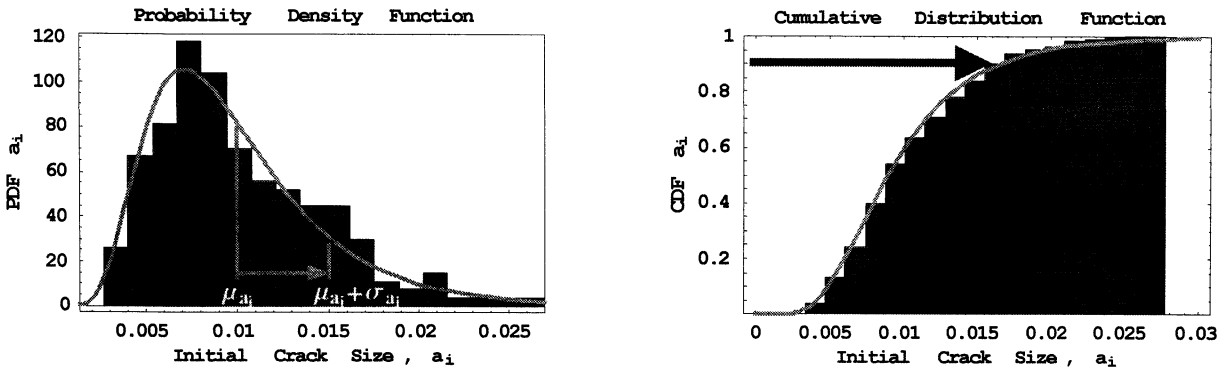


Figure 9 The PDF and CDF of the crack size random variable

The PDF of Figure 9 has its mean, μ_{a_i}, and the location of one standard deviation away from the mean, $\mu_{a_i} + \sigma_{a_i}$, shown by the vertical lines coming down from the PDF. The CDF of a_i uses arrows to indicate that, for this case, the 90th percentile of a_i is estimated to be 0.015 (inches). The mean, standard deviation, and specific percentiles of a response are important density parameters because they can be used to calculate the reliability of the system.

Reliability

The reliability of a system would be the probability of observing future "safe" system responses. This reliability will be anywhere between 0 and 1, and it is usually given by p_s. The only other type of system response would be an "unsafe" system response and, therefore it would be the other half of the complete percentage. The probability of failure is usually given by $p_f = 1 - p_s$. Calculating the reliability of the system is usually one of the main goals of a reliability analysis. The reliability can be estimated using the mean and standard deviation together, which is usually not very accurate. The reliability can also be calculated by estimating the probability associated with specific "safe" response events. In any case, a reliability analysis has a certain structure, or methodology to it.

A reliability analysis typically begins with a response that is to be studied. This response is usually complicated because it models the physics of an actual system. Responses are typically given by Z(X), where Z is a generic response variable and X represents the multidimensional space for which the response exists over. As an example, let us consider a response that is the number of cycles to fracture, N_f, of a system being cyclically loaded. This response is shown in Equation 8.

$$Z(X) = N_f = \frac{\left(a_f^{\,1-m/2} - a_i^{\,1-m/2}\right)}{c\left(1.1215\Delta\sigma\right)^m \pi^{\,m/2}\left(1-m/2\right)} \tag{8}$$

This response is later used in the convergence study discussed in section 3.5. The system will eventually fatigue fracture at a specific number of cycles, N_f. The variable a_i, is the initial crack size within the part in units of inches, c and m are model constants, $\Delta\sigma$ is the cyclic loading on the part in units of kilo-pounds per square inch (ksi), and a_f is the final crack size, given by Equation 9.

$$a_f = \frac{1}{\pi}\left(\frac{K_{IC}}{1.1215\Delta\sigma}\right)^2 \tag{9}$$

The variable K_{IC} in Equation 9 is the fracture toughness of the stressed material and its units are $ksi - in^{1/2}$. The Equations 8 and 9 represent the response of a type of component in a larger structure that must be repeatedly replaced. Therefore, the geometry of the system is considered random, and is modeled as such. The other dependencies are treated as constant values for this 1-D example. The dependencies of Equations 8 and 9 are shown in Table 4. The final crack size, a_f, is not shown because it is a function of the variables that are shown.

Table 4 Design variables for fatigue fracture response

Variable	*Description*	*Value or Distribution*
K_{Ic}	Fracture toughness $(ksi\sqrt{in})$	60
a_i	Initial crack size (in)	LN (0.01,0.005)
c	Paris constant (-)	1.2E-10
$\Delta\sigma$	Cyclic load (ksi)	100
m	Paris exponent (-)	3

In this case, it is unacceptable for the system to fracture (mechanical failure) before 5,000 load cycles. Therefore, the reliability of the component, which is indeed all possible components that could be purchased, would be the probability of observing a component that would have a lifetime longer than 5,000 load cycles. This reliability can be calculated if the mean and standard deviation of the response, μ_Z and σ_Z, respectively, can be computed. The reliability can also be calculated if the probability that the response is part of a set of safe response values can be computed, $\Pr[Z \in Z_{SAFE}]$. Thus, a reliability analysis is entered knowing the response under study and its dependencies, Z(X) and X, respectively. The way the dependencies are modeled, either as random variables, or as a constant, is also known. If the variables for which the response under study are modeled as random variables, then the PDF and its defining parameters should be known. The variables that are modeled as deterministic, that is, as a constant, should have a value associated with each one in order to be able to calculate response values. Furthermore, a probabilistic analysis should be started with a complete understanding of the system response and the set of response values that are considered safe, Z_{SAFE}.

$$\Pr\left[\left|Z-\mu_Z\right|<k\sigma_Z\right]\geq 1-\frac{1}{k^2} \tag{10}$$

or

$$\Pr\left[\left|Z-\mu_Z\right|\geq k\sigma\right]\leq \frac{1}{k^2} \tag{11}$$

Reliability Calculations Using the Mean and Standard Deviation

If the mean and standard of the response, μ_Z and σ_Z, respectively, can be computed, the reliability of the system can be *estimated* using Tchebysheff's theorem. A discussion on Tchebysheff's theorem can be found in section 3.4 of Wackerly, et al. The reliability must be estimated because even if the mean and standard deviation can be exactly calculated, which is usually never the case, reliability calculation using Tchebysheff's theorem is still an estimate. Therefore, in using the mean and standard deviation of a response to calculate the reliability of a system there is a bit of a compounding of errors. This is seen in many analytical situations, which includes reliability analyses, where we assume that the response model and each PDF of the underlying random variables are exact. In any case, at least a value for the reliability is arrived at which does have theoretical roots.

Tchebysheff's theorem provides bounds for probabilities. It is usually needed when the distribution of a random variable, like the response, Z(X), is unknown. Tchebysheff's theorem states that if Z is a random variable with a finite mean and standard deviation, μ_Z and σ_Z, respectively, then for any $k \geq 1$ the following holds true.

Equations 10 and 11 can be used to give estimates of the reliability of the system governed by the response of Equations 8 and 9. While, either equation can be used for a reliability estimate, here, Equation 11 is used for an estimate based on knowledge, or at least estimates of, the mean and standard deviation of the response, μ_Z and σ_Z, respectively. As an

example, suppose n=200 response values are calculated and estimates of the mean and standard deviation are computed to be 17,200 and 8,800 cycles, respectively. The values of the response below 5,000 cycles are located below the mean at $\mu_Z - k\sigma_Z$, where k=1.386. The probability that Z anywhere outside of the region defined by the distance $k\sigma_Z$ away from and on either side of the mean is computed with $\Pr\left[\left|Z-\mu_Z\right| \geq k\sigma_Z\right] \leq 1/k^2 = 0.52$. Divide this number by two to get 26%, an upper bound to the probability that the response will be less than 5,000 load cycles. Therefore, the *estimated* probability of failure is given by $p_f = \Pr[Z \notin Z_{SAFE}] \leq 26\%$. The reliability of the system is given by $p_s = \Pr[Z \in Z_{SAFE}] = 1 - p_f \geq 74\%$. Due to the inequalities of Tchebysheff's theorem it can be said that the probabilities estimated are bounds to the actual probabilities. The actual probability of failure would most likely be less than the 26% calculated and the reliability will most likely be greater than the 74% calculated. Thus, we have successfully used the mean and standard deviation *together* to obtain an estimate of the reliability of the system governed by the response shown in Equations 8 and 9. These probabilities obtained using Tchebysheff's theorem will be compared to a more accurate answer in the following pages.

Reliability Calculations Using Probability Calculations

The reliability of a system can also be calculated by estimating the probability of observing safe system responses. Reconsider the problem previously discussed, where the reliability is given by $p_s = \Pr[Z \in Z_{SAFE}]$. Obtaining this value would be a matter of performing the first integration from the left that is shown in Equation 12.

$$p_s = \Pr[Z(X) \in Z_{SAFE}] = \int_{Z \in Z_{SAFE}} f_Z(t)dt = \iiint_{X \Rightarrow Z \in Z_{SAFE}} f_X(T)dT = \int_{a_i \Rightarrow Z \in Z_{SAFE}} f_{a_i}(t)dt \qquad (12)$$

The equalities in Equation 12 summarizes the method of distribution functions, which is used for finding the probability of observing values of a random variable, when it is dependent on other variables. Proceeding from left to right, we shall explain all of the terms. We have a function, Z, dependent on, in general, many variables, X, and want to know the probability that we will observe a safe function response. This can be calculated by integrating the one-dimensional PDF of Z, f_Z, over the region for which we want to know the associated probability, Z_{SAFE}. This integration, as usual, is done using a dummy variable, t. The same answer would be calculated if we find the region Z_{SAFE} in the M-dimensional X space and integrate the joint probability density function (JPDF) of X, f_X, over that region. Theoretically, finding the Z_{SAFE} region can be done because Z=Z(X) and each event in the X space has one and only one Z value associated with it. The JPDF has the similar property of being able to obtain the probability of events in its domain by integrating the function over that domain, except the only difference is that it is M-D and the events are joint events, while the PDF is 1-D with observations from only one group. In applying the general method of distribution functions to this example under study we consider the last integral of Equation 12. In order to obtain the probability that the response is safe, or over 5,000 load cycles, we integrate the PDF of the crack size variable over the region of crack sizes that imply a safe response. Any crack size below 0.031 inches will imply a safe response. This limiting crack size value was obtained using a root finding technique, something that can almost never be done for practical responses dependent on many variables. This integration over the safe region of the domain of the response under consideration is shown in Figure 10. The probability of observing a system with a lifetime that is over 5,000 load cycles is 99.58%. Stated differently, this system under study is 99.58% reliable.

This uncertainty in the response is due to the constant replacement of a component of the system that contains a crack size that is naturally random, and is modeled as such in our mathematical analysis.

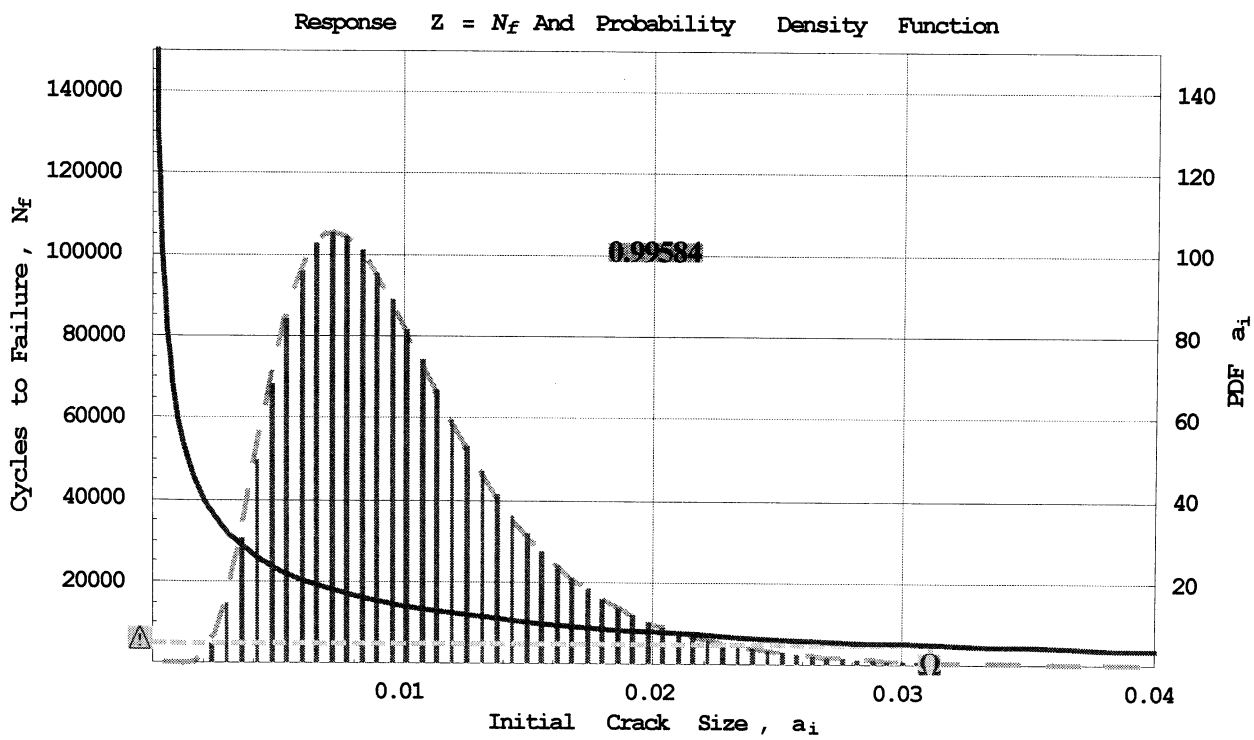


Figure 10 Integration over the density function of the domain of a response

The reliability obtained by using the method of distributions, $p_s = 0.9958$, and the probability of failure, $p_f = 0.0042$, are not the same as the values obtained using Tchebysheff's theorem, which were calculated to be $p_s \geq 0.74$ and $p_f \leq 0.26$. The reliability obtained using Tchebysheff's theorem will be greater than 74% and the probability of failure will be less than 26%. This is empirically proven in this example when we assume that the integration performed is exact, and therefore, so is our reliability calculated using the method of distributions. Although, probabilities obtained using Tchebysheff's theorem are erroneous, at least they

provide bounds for the actual probabilities of a response with an unknown density or distribution function [Wackerly et al, p.245].

The reliability of a system can be computed with knowledge of certain density parameters. Here, the reliability was computed using Tchebysheff's theorem along with the mean and standard deviation of a response. The reliability was also calculated by calculating the probability of observing a set of events of the variables for which the response is dependent on and that imply a safe response.

True, the mean and standard deviation of a concerned system response are desirable, but even more so is the reliability. Also, for the most part, a system will be designed in a highly reliable manner; therefore, a reliability analysis usually entails calculating high probabilities associated with the reliability or low probabilities associated with system failure. The surface separating the safe and failure region is the same for both calculations. The only difference is that to obtain the reliability we integrate the JPDF over the safe region and to obtain the failure probability we integrate the JPDF over the unsafe region. If comparing different reliability methods, it would be good to know how well they estimate the mean, standard deviation, and the probability of observing safe system responses, which will be a high probability for a good design. For ease of such a comparison study, and to be able to test methods with responses that are purely mathematical, we can compare the ability of several methods in estimating a high percentile. For the example just mentioned the 99.58^{th} percentile of the *crack size variable* is 0.031 inches. The 0.42^{th} percentile of the *number of cycles to failure* variable is 5,000 cycles. The mean, standard deviation, and percentiles are all density parameters that can generally never be exactly known. They must almost always be estimated in order to calculate the reliability of a system under study. Section 3.3 will now discuss how Monte Carlo and Latin Hypercube

Sampling can be used to estimate the mean, standard deviation, and the 99th percentile of the density of any response under that is random due to the randomness of the variables for which it depends on.

3.3 RANDOM SAMPLING AND ESTIMATION

A response function will be random if the variables it depends on are also naturally random. The randomness of the response can be characterized if the PDF, or just the density, of the response can be obtained. The density itself is defined by its parameters, some of which can be used to estimate the reliability of a system under study that is governed by a mathematical, physics-based response. Accurate estimation of the parameters used to compute the reliability of a system is important.

One type of desired parameter is a measure of central tendency of the response and is called the mean of the response. Another parameter that an analyst might be interested in is a measure of the average spread of the response about the mean, or expected value. This measure of variation is called the standard deviation of the response. The mean and standard deviation can be used together to conservatively estimate the probability of observing certain ranges of system responses. Yet another response density parameter is a response proportion, or ratio. The proportion parameter is the ratio of the number of responses that would be observed to lie in a certain range of the response, or bin, to the total number of response measurements, or calculations, after a long series of response observations have been made. This proportion, or relative frequency is a measure of the probability that a response will be observed to lie within a specific range, like a safe response range, in some future event. This is the relative frequency concept of probability, and it is a simple application to predict future events compared to the rigorous definition of probability. Calculating response proportion related to safe system events is a direct way to estimate the reliability because it is the probability of observing safe system events. Computing high reliabilities associated with a specific response range is difficult for many reliability methods. Equally difficult would be to compute the response range associated

with a high probability, and the high limit of this response range would be the percentile associated with that high probability. Percentile estimation allows purely mathematical functions to be added to the list of test responses used in a reliability method comparison.

The mean, standard deviation, and a high percentile are desired density parameters that can never be exactly known; therefore, they must be estimated. These parameters can be estimated using a sample of response evaluations obtained using Monte Carlo (MC) and Latin Hypercube Sampling (LHS). Both methods are types of random sampling and can be compared with each other in their ability to efficiently, and accurately estimate desired density parameters.

This section will discuss two random sampling methods – Monte Carlo and Latin Hypercube Sampling, estimators and estimation, and how sampling methods can be compared. Section 3.4 will discuss the enhancement of NESSUS to be able to perform Latin Hypercube Sampling. Section 3.5 will discuss the comparison of MC and LHS in their estimation abilities, and section 3.6 will then conclude this written work.

Random Sampling

Random sampling is a common computer simulation of response events that might be physically observed. Many times the simulation is preferred over actually measuring a response because sometimes the response measurement, be it anything from a stress to a lifetime of a part, is too difficult and/or expensive to obtain. Quickly described, coordinates in the multidimensional space for which a response exists over are obtained and that are distributed such that the probability of joint events in the M-D space is approximately what might be seen in nature. Response values can be calculated from these coordinates, and these values will be distributed according to what will be observed in nature due because the coordinates in its domain had a joint density that was (hopefully) accurately modeled. These response values are

then used to estimate its density parameters. There are several steps used to obtain random samples of a response.

1. Obtain 1-D coordinates for each individual variable that a response depends on.
   - Distribute each set of coordinates according to their known density.
2. Pair the individual variable values with each other to form M-D coordinates.
   - Pair according to correlation that might exist.
3. Evaluate the response at each of those M-D coordinates.
   - Have successfully mimicked observations in nature.

There are quite a few sampling methods to choose from. Monte Carlo and Latin Hypercube Sampling are two sampling methods. Their only difference lies in the first step of random sampling – obtaining 1-D coordinates from each individual underlying random variable for which a response is dependent on.

Monte Carlo Sampling – Its Special Characteristics

Monte Carlo sampling is a popular computer simulation of what might be observed in the physical world. A mathematical model of a response is known and so are the properties of the random variables it depends on. These properties can be the PDF, f_x, the CDF, F_x, or both. While there are several steps in random sampling, Monte Carlo random sampling is performed if the act of obtaining 1-D coordinates for each individual random variable for that a response depends on is done in a certain manner. First, we assume that we know how many response evaluations we can afford to take in order to estimate the necessary density parameters. If we agree that we can calculate n response values, then we can deduce that we need n M-D

coordinates to be made by randomly pairing the n 1-D coordinates of each underlying random variable with each other. In order to perform Monte Carlo Sampling, for each of the M random variables that is part of the domain of the response, we generate n random numbers between 0 and 1. Then, for each random variable, X_j, where, $j = 1,2,\ldots,M$, we use Equation 13 to obtain a vector of random samples.

$$X_j(i) = F_{X_j}^{-1}[Random_i(0,1)] \qquad i = 1,2,\ldots,n \tag{13}$$

Thus, we can then compute a vector of n random samples from each of the M random variables that a response depends on using the inverse CDF, $F_{X_j}^{-1}$, of each random variable. The $Random_i(0,1)$ term is the random number generated, and there are many random number algorithms to perform this task, but that is not the issue here. The point of emphasis however, is that these n dissimilar random numbers should be uniformly distributed. They should each have an equal probability of occurrence. Since the distribution function of every random variable, F_x, will range from 0 to 1 and most random variables that are observed in nature have a distribution function that is a one to one mapping, this function can be inverted to obtain as many random variable values as there are random numbers between 0 and 1. If the n random numbers generated are uniformly distributed then each of the random variable values obtained has an equal probability of occurring in this random sample due to the one to one mapping property of the CDF.

If this process is performed correctly and completely, we should have a vector of length n for each of the random variables in the domain of the response. Figure 11 shows the independent inversion of 2 variables that exist in the domain of a response under study. For the record, these variables are the initial crack size, a_i, and the cyclic loading, $\Delta\sigma$, variables that exist over the

domain of the first test case response that will be studied in section 3.5.

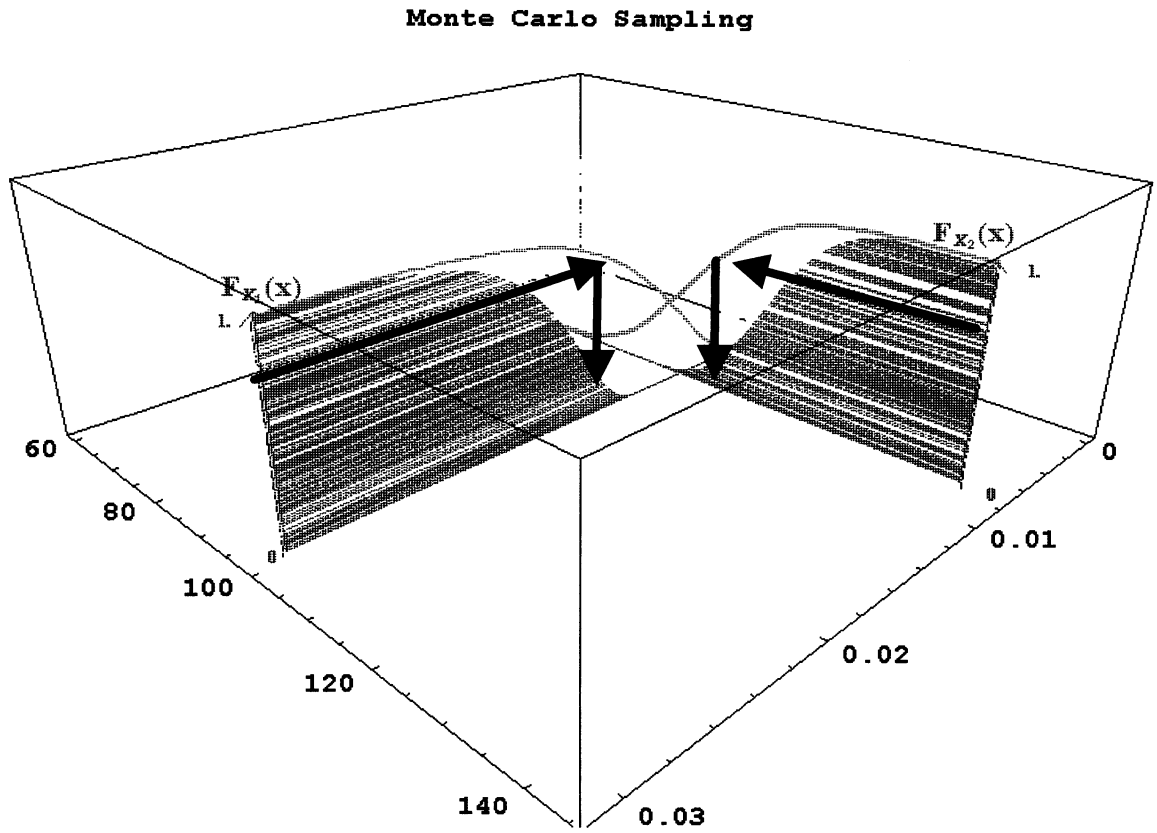


Figure 11 Inverting each CDF for Monte Carlo sampling

In Figure 11, n=200 random numbers between 0 and 1 are used as inversion points from the range of the distribution function of each underlying random variable. The dark arrows show an inverse for each random variable, and the gray shaded area under the CDFs are the rest of the random numbers being inverted. Evident from Figure 11 is that the distribution of the random numbers for Monte Carlo Sampling is not always uniform.

Latin Hypercube Sampling – Its Special Characteristics

Latin Hypercube Sampling is also a computer simulation of what might be observed in the physical world. It too begins with knowledge of a mathematical model of a response and the PDF or CDF of the random variables it depends on. As with Monte Carlo Sampling, the method that LHS uses to obtain the 1-D coordinates for each individual random variable for that the response under study depends on is what makes LHS unique. First, we assume that we know we can afford to calculate n response values. We therefore first need to obtain n 1-D coordinates of each underlying random variable. In order to perform Latin Hypercube Sampling, for each of the M random variables that exist in the domain of the response, we generate n random numbers between 0 and 1 using any good random number generator. They should each have an equal probability of occurrence, but, as seen from Figure 11, for Monte Carlo sampling, they do not. For LHS, the generated random numbers are *not* used in the inversion of the distribution function. Instead, an additional step is taken that defines LHS. For each of the underlying random variables, all of the probable space from 0 to 1 is stratified, or divided into n equal probability bins. One of the dissimilar n random numbers between 0 and 1 is used within each bin as a percent increase from the lower limit of the bin to the upper limit. Using this new value between 0 and 1, along with the inverse CDF, $F_{X_j}^{-1}$, we can calculate our desired random variable value. Repeating this process for as many response evaluations that will be made and for each random variable will result in a set of n M-dimensional coordinates that are used to evaluate the response. Thus, for each random variable, X_j, where, $j = 1,2,\ldots,M$, we use Equation 14 to obtain a vector of random samples.

$$X_j(i) = F_{X_j}^{-1}\left[\frac{Random_i(0,1) + i - 1}{n}\right] \quad i = 1,2,\ldots,n \tag{14}$$

For example, if n=200 response evaluations are to be calculated, then n=200 random numbers are generated. Pretend the first random number is 0.33. Since the first bin is from 0 to 1/200 (=0.005), the first number used as an inversion point in the probability space of the random variable under consideration is 0.33/200 (=0.00165). If the second random number generated is 0.54, the second inversion point for the respective random variable is 0.0077, which lies within the second bin of 1/200 (=0.005) to 2/200 (=0.01). This ensures that all of the probable space of the random variables is completely spanned; and therefore, the fundamental concept of equal probable variable values in a random sample in more closely enforced. Since the distribution function of every random variable, F_x, will range from 0 to 1 and most random variables that are observed in nature have distribution functions that are one to one mappings, the CDF can be inverted to obtain as many random variable values as there are numbers between 0 and 1.

If this process is performed correctly and completely, we should have a vector of length n for each of the random variables in the domain of the response. The elements of each vector should be randomly shuffled in order to obtain random LHS samples for each underlying random variable. Figure 12 shows the independent inversion of 2 variables that exist in the domain of a response under study. Again and for the record, these variables are the initial crack size, a_i, and the cyclic loading, $\Delta\sigma$. Also, in Figure 12, the initial random numbers generated that are used as percent increases from the lower bin value are the *same* raw random numbers that were used to directly obtain MC samples.

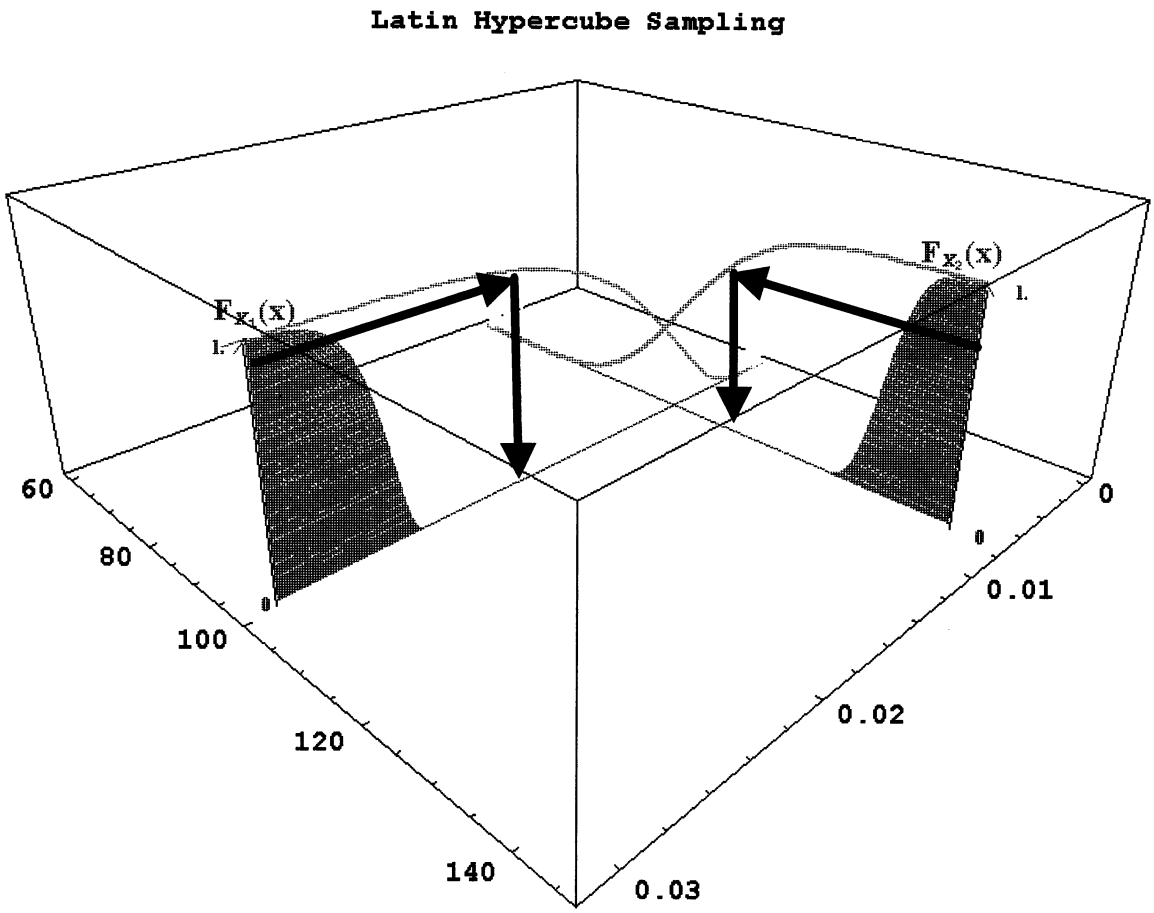


Figure 12 Inverting each CDF for Latin Hypercube Sampling

In Figure 12, n=200 random numbers between 0 and 1 are used as percent increases from the lower values of the n=200 equal probability bins that span the probable space for each underlying random variable. Because the CDF, or F_X, for each random variable is a one to one mapping, a uniformly distributed set of numbers that are used as inversion points implies that each random variable value obtained from the inversion has an equal probability of occurring. It must be reminded that set of inversion points is *not* the original set of random numbers generated, and this is a unique characteristic of LHS. It is obvious from Figure 12 that the LHS distribution of the numbers used as inversion point is more uniformly distributed than the MC distribution of inversion points for each underlying random variable when this Figure is

compared to Figure 11.

The essential differences between MC and LHS sampling is now apparent. Monte Carlo sampling used the original set of n random numbers as inversion points for each underlying random variable's CDF, while Latin Hypercube Samples come from using the original set of n random numbers as percent increases from the lower value of in each of the n equal probability bins. This still does not cover the random sampling process, and we will now step through it slowly but surely.

For either MC or LHS, we now have a set of coordinates for each underlying random variable that exists in the domain of the response under study. These points are shown in Figure 13. The points shown are actually the Monte Carlo points, but what has yet to and will be said from henceforth *applies to both Monte Carlo and Latin Hypercube Sampling*.

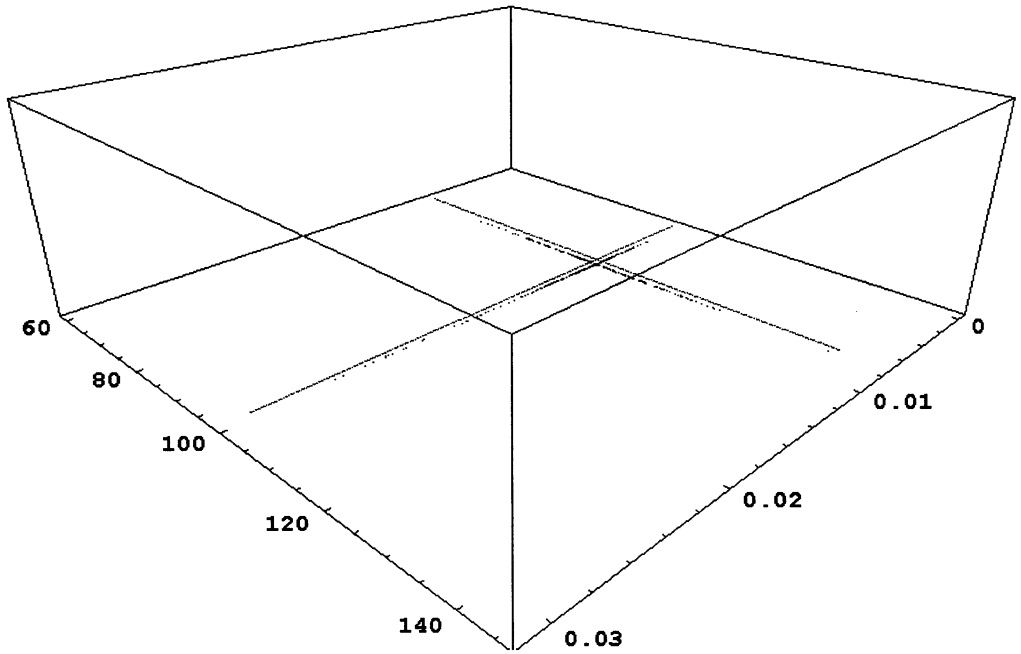


Figure 13 Coordinates for each underlying random variable are obtained

The coordinates of each underlying random variable that is part of the domain of the response, shown in Figure 13, are not just any coordinates. They are distributed according to its

PDF, which was already known before the reliability analysis began and whose related CDF was used as the function to invert. The relation of the 1-D coordinates to their own density can be seen in Figure 14. The density of each of the two random variable used in this example are shown as a continuous curve, while the density that is approximated by taking n random samples of each variable is shown as a bar graph. The larger the amount of samples that are taken from each random variable, the closer the approximate density will be to the true density.

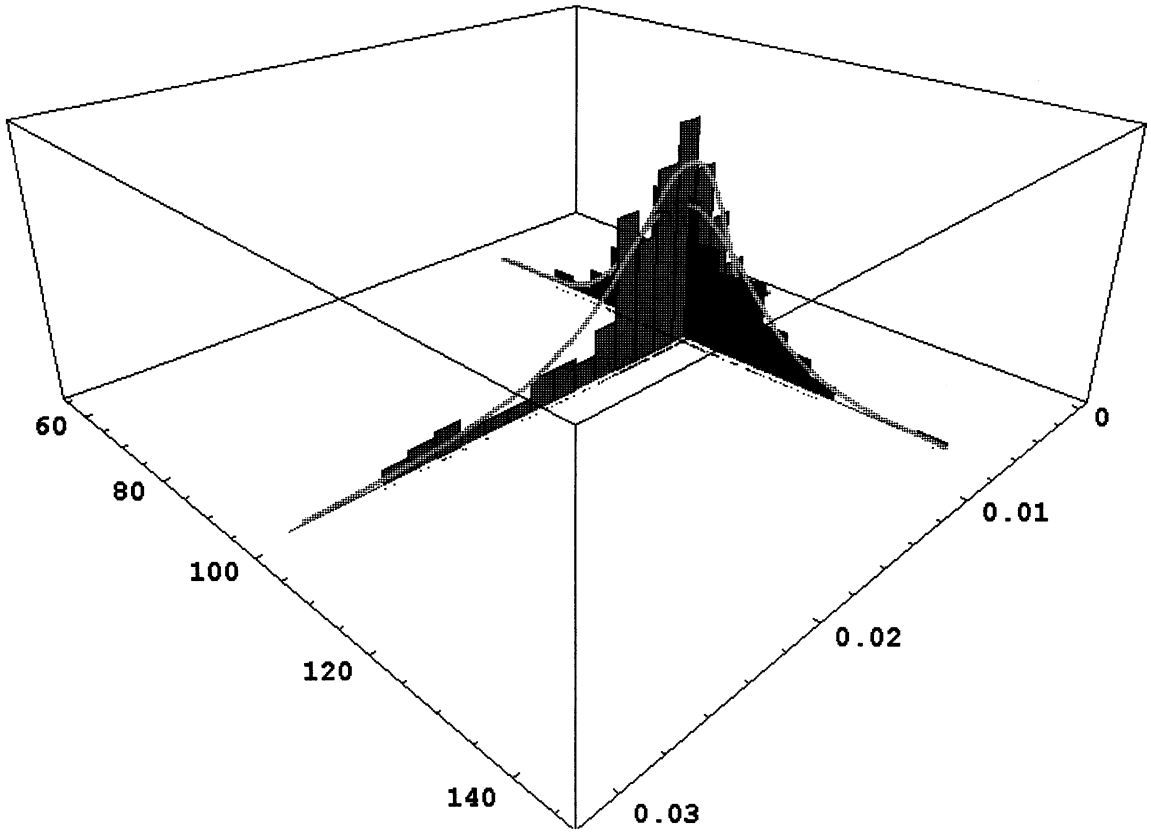


Figure 14 Distribution of individual coordinates and their relation to their PDF

The coordinates of each underlying random variable can be paired with each other in order to obtain M-dimensional coordinates that exist in the domain of the response. They should be paired with each other in a random manner if the variables are independent. Independent variables have nothing in common with each other. That is to say that knowledge of one variable implies nothing about the other. If it is known that a correlation exists between pairs of underlying random variables, actions should be taken to obtain the multidimensional coordinates in such a manner as to capture the correlation that is desired. Inducing correlation amongst the variables in a random sample is not going to be discussed here. Randomly paired 1-D coordinates that form M-dimensional coordinates are shown in Figure 15.

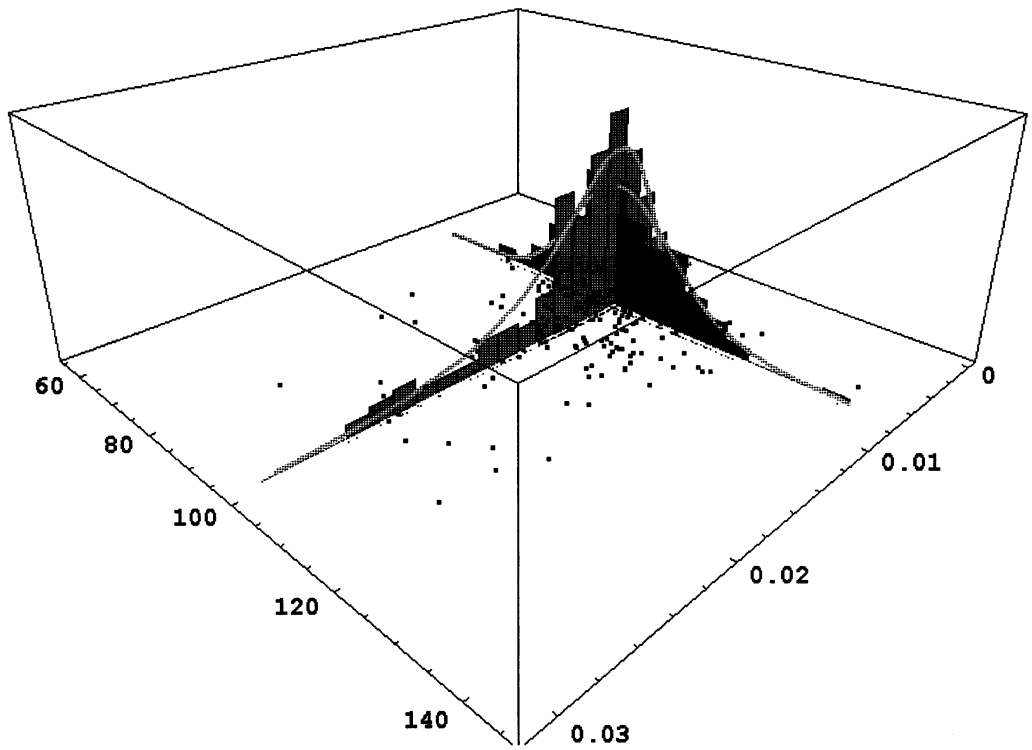


Figure 15 M-dimensional coordinates obtained by pairing 1-D coordinates that are appropriately distributed

The M-D coordinates in Figure 15 are shown along with the estimated and assumed exact density of each individual variable for which a response is dependent on as a reminder that they came from 1-D coordinates that are (hopefully) appropriately distributed. If these M-D coordinates were properly paired with each other according to the correlation between the variables that exists in nature, which we are trying to simulate, then they have an associate joint density function, JPDF, or $f_X(X)$. In this case, X is then an M-D vector. The JPDF estimated from the M-D coordinates along with the assumed exact JPDF calculated from the previously known PDF of each individual variable is shown in Figure 16.

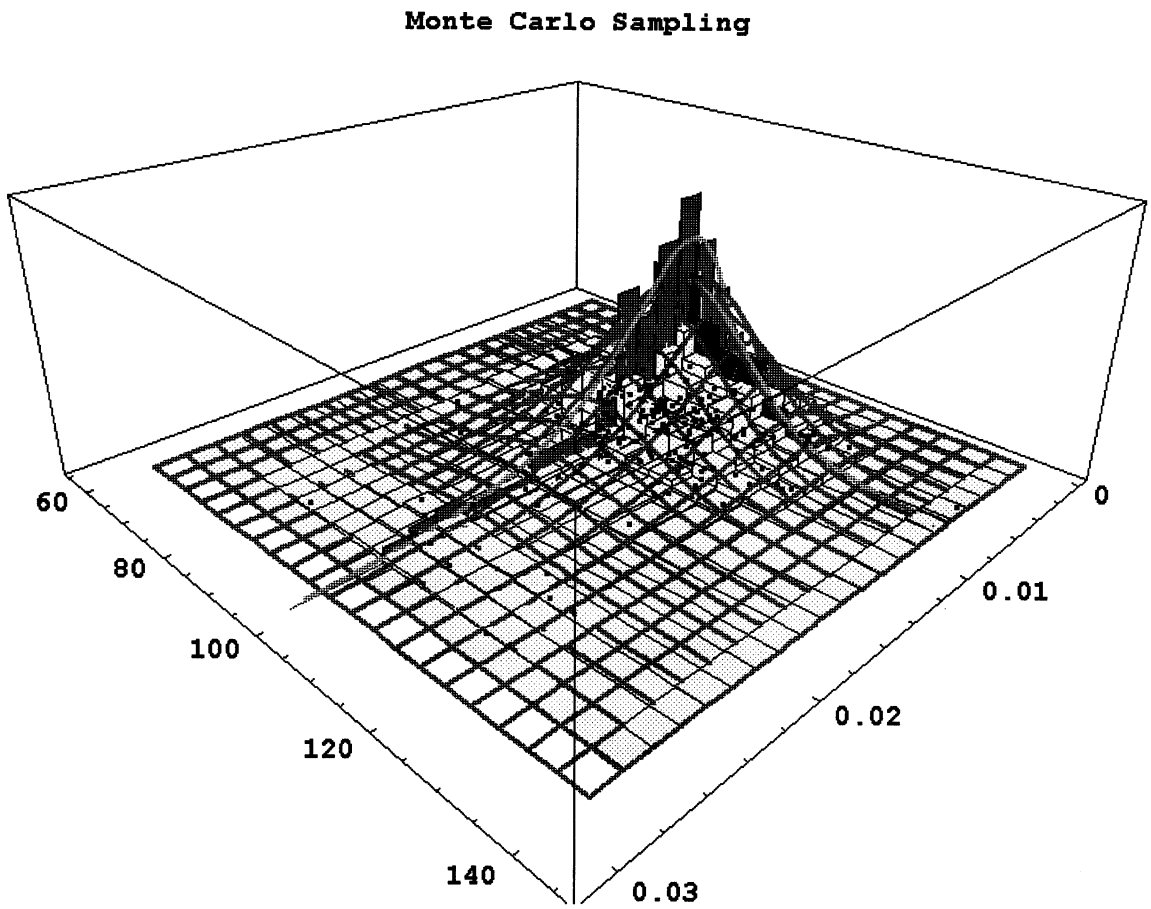


Figure 16 Estimated M-dimensional JPDF and the exact JPDF

The discrete JPDF shown as rectangular prisms in Figure 16 is essentially an estimate of the assumed exact JPDF shown as a "see through" surface. Estimating the JPDF is where the errors lie when random sampling is used to estimate parameters of the density of a response. If enough coordinates are used to obtain the JPDF estimate, then the JPDF is more accurately captured and we are more closely simulating something in nature. If each of these simple joint events is actually measured from a physical system under study, then each would imply a specific response value. This is true whether or not we decide to measure the response. We can simulate the measurement of a response if we have a mathematical model that accurately captures the relationship of the response to the variables it depends on. For a reliability analysis, we do have a mathematical model and so we use it to evaluate n response values at each of the n M-D coordinates. This is portrayed in Figure 17. Contour lines representing different surface levels of the response are shown in Figure 17. The contour plot is placed at the top face of the box that bounds the density functions for clarity. There are eight contours, equally spaced at 15,000 load cycles apart. The response under study is the same one discussed in section 3.2. The lowest contour shown and labeled is 5,000 load cycles. The highest contour shown is 105,000 load cycles and the highest one labeled is 35,000 load cycles. From Figure 17, we can see that when cyclic loading a specimen until fatigue fracture occurs, the number of cycles for this event to happen will decrease as the initial crack size of the specimen increases. Also, the system will fail sooner if the load change that defines the cyclic loading is large than when it is small. Out of the n=200 response evaluations made, 4 of them were under 5,000 load cycles. Recall that in section 3.2, the response levels over 5,000 cycles implied a safe part and response levels under 5,000 cycles implied an unsafe repeatedly replaced component.

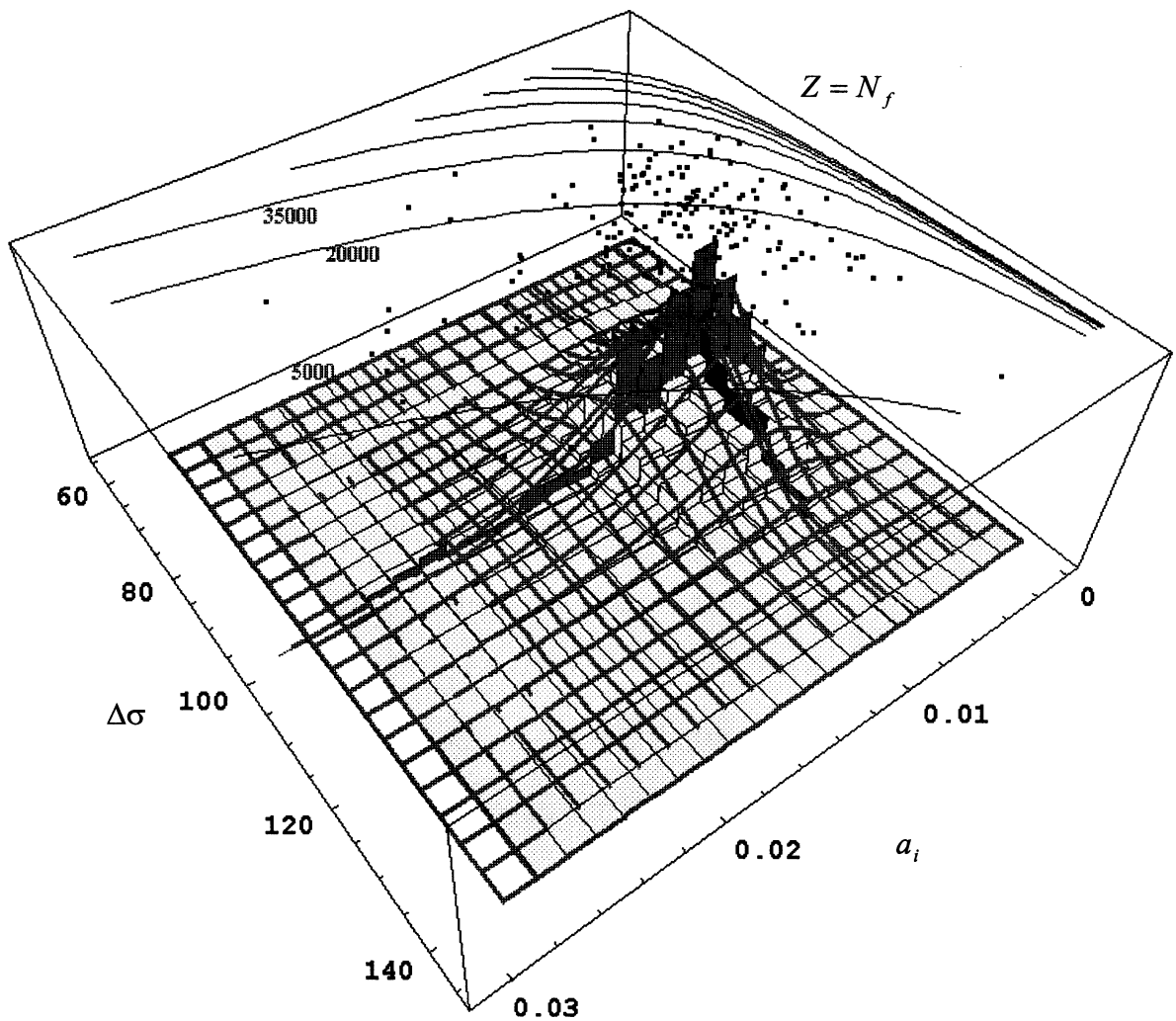


Figure 17 Response is evaluated at each of the n M-dimensional coordinates

These n=200 response evaluations can be used to estimate the parameters of the response density. Some of the parameters that can be estimated from this set of response data are the mean, standard deviation, and the 99th percentile of the response. The mean and standard deviation can be used *together* to estimate the probability of certain response events occurring in the future. With certain assumptions about the distribution, this probability can be used to estimate the reliability of the system. These 200 response calculations can be used to estimate the reliability of the system if the response level that divides the space into failure and safe

regions is known. High probabilities are usually difficult to estimate, as are low probabilities. Therefore, if comparing sampling methods, as is purpose of this work, it would be beneficial to merely estimate a high percentile because it is already known that this estimation will be difficult. We will now discuss using n response calculations to estimate the mean, standard deviation, and the 99th percentile of a response. After which, we will discuss ways to compare sampling methods, and this would conclude this section.

Estimators and Estimation

Estimators are rules, or algebraic expressions, that estimate density parameters using a set of data. Given a set of data and an estimator, a resulting estimation can be made that hopefully is close to the true value of the density parameter of interest. For the same density parameter, there can be several types of estimators, or functions, dependent on a set of data. In estimating a density parameter using several estimators, if would be found that some of them have better estimation characteristics than others. Comparing estimators for the same density parameter is not the subject of this work, so it will not be discussed. What will be discussed in this section is the use of a commonly used estimator in the separate estimation the mean, standard deviation, and 99th percentile of the response.

The mean of a response is a measure of where the central tendency of a set of responses lies. The mean of a response, μ_Z, is estimated by calculating the mean of a sample of responses, $\overline{Z}$. The widely used mean estimator is shown in Equation 15.

$$\hat{\theta}_{\mu_Z} = \overline{Z} = \frac{1}{n}\sum_{i=1}^{n} Z_i \tag{15}$$

The mean estimator, $\hat{\theta}_{\mu_Z}$, uses n values of a response, $Z_{i=1,2,\ldots,n}$, in order to estimate the mean of the response. The observed responses are summed together, and the result is divided by the total number of observations, n. The mean estimator uses a sample of responses to obtain a single mean estimate. Since the estimator is a function of random variables it too will be random. Multiple estimates will produce different values because of the random nature of obtaining samples of a response. Multiple estimates will be centered about the true mean and the variation of multiple estimates about the true mean will decrease as more response evaluations are used to calculate each estimate. For most responses, the shape of the probability distribution of the mean is mound-shaped even for small sample sized (n=5). It will approach normality for sample sizes greater than or equal to 30. Observing the shape of the distribution of the mean estimator can only be performed if repeated experiments are performed [Wackerly et al, 1996].

The standard deviation is a response density parameter that is a measure of the average spread of the response about the mean, or expected value. It can be estimated using a set of n response values. The most commonly used estimator for the standard deviation of a response is shown in Equation 16.

$$\hat{\theta}_{\sigma_Z} = S_Z = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(Z_i - \overline{Z}\right)^2} \tag{16}$$

The standard deviation estimator sums up the square of the error of each response value from the mean, divides by n-1, and then takes the square root of the result. It is an apparent average deviation of all of the response values away from the mean. The standard deviation estimator, $\hat{\theta}_{\sigma_Z}$, also uses a sample set of n response values to estimate the standard deviation of a response. It is a function of random variables and is therefore also random. The estimator will have a distribution associated with it, which, is centered about the true standard deviation. Also,

its variation about the true standard deviation of the response will decrease as the number of response values, n, that are used for each standard deviation estimation increases. The probability distribution of the standard deviation estimator shown in Equation 16 has a longer tail in the positive direction, i.e. is positively skewed, for small sample sizes; however, it is approximately normal when n>25 response values are used to obtain each standard deviation estimate [Wackerly et al].

The 100pth percentile is a response density parameter that is a measure of the location within the range of values of a response for which it can be observed that 100p% of the values fall below it in a long series of response observations. It can also be estimated using a set of n response values. The 100pth percentile estimator is shown in Equation 17.

$$\begin{aligned} &\hat{\theta}_{100p\%} = Z'_j \\ &j = \text{int}[np + 0.5] \\ &Z'_1 < Z'_2 < \ldots < Z'_j \ldots < Z'_n \end{aligned} \tag{17}$$

The 100pth percentile is estimated using Equation 17 by first sorting the set of n responses from least to greatest. The jth element of those responses is chosen as the 100pth percentile so long as j is the integer part of the np+0.5. This is essentially rounding j off to the nearest integer. In a sorted list, kept in such a manner that the 1st element the smallest and the nth element the greatest, the jth element will have the property such that j/n=p, where p is the fraction of the responses that are equal to or below the jth element, and 100p% is the percent of response values below the jth element. To choose the jth element bases on the fraction, p, is a matter of multiplying n by p. However, this may not turn out to be an integer and one method to settle this problem is to select the (j-1)th element, that is a lower percentile than desired, or the (j+1)th element, which is a higher percentile than what is sought. There are a few methods used to settle this dilemma, and the method used *for this estimator* is to round off to the nearest integer. For

completion, the 99th percentile is estimated using the general formula of Equation 17 by letting p=0.99. The percentile estimator of Equation 17 is a function of random variables, and will therefore have an associated density, or distribution. While, during the duration of this study, percentile estimation was not encounter in literature to the extent that anything can be said about the centering of multiple percentile estimates around the true value; nor can anything be said about the variation of the 99th percentile estimator distribution as the number of response values used to obtain each percentile estimate is increased. This is left to the portions of section 3.5 that discuss some empirical distributions obtained by making multiple 99th percentile estimates for the purpose of capturing its distribution.

All of the estimators mentioned are functions of random variables and they will therefore also be random. This can leads to problems when using a set of n response values for the purpose of estimating the appropriate parameter(s). In order to completely understand the problems encountered when using Monte Carlo or Latin Hypercube Samples along with Equations 15,16, and 17 for the purpose of estimating parameters of the density of the response we must first discuss estimation and its characteristics.

Estimation is a method used to estimate parameters of the density of a response. The mean, standard deviation, and 99th percentile can all be considered desired response density parameters. Furthermore, the true, or exact value of each of the parameters can be considered a *population parameter*. This is because they could be exactly calculated if the whole population of response values is known. This whole population is mostly very large. Usually, the population of all response values is infinite in size. Since a population parameter can rarely be obtained, we let the population parameter be the *target parameter* of interest; and, we can only deduce something about the target, which is the exact density parameter. This can be performed

in two ways: hypothesis testing and estimation [Wackerly et al]. Hypothesis testing will not be discussed here or in any part of this work.

Population parameters of the response are targets of interest; unfortunately, due to mathematical complexity of the response and its relation to its dependencies and/or lack of time necessary for computations, we must settle for estimates of target parameters. Estimation involves using data that is a *sample* of the population to deduce something about a target parameter. There are two types of estimation: point and interval. *Point estimation* uses sample data to obtain a single value that estimates the target parameter. *Interval estimation*, which will not be discussed in this work, uses sample data to obtain an interval that encloses target parameter.

Point estimation uses an *estimator*, which is an equation or rule, to calculate a value that is an estimate of a target density parameter using sample data. The target parameter is usually given by θ, and the estimator, $\hat{\theta}$. Estimators are typically random because they are functions of random variables. They will have a distribution associated with it that is captured when multiple estimations are used to produce multiple estimates. This concept is depicted in Figure 18.

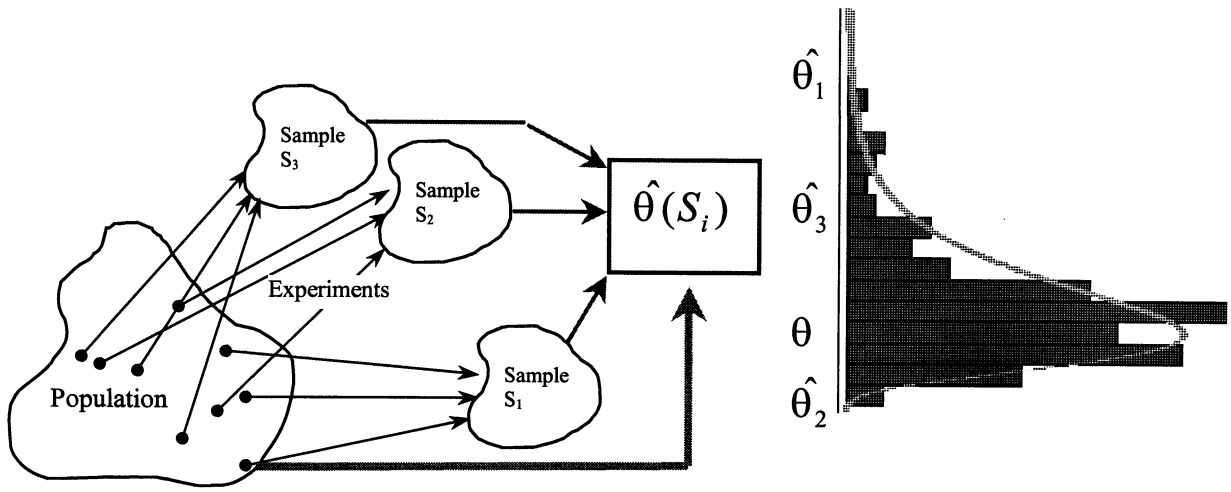


Figure 18 Population and Sample Spaces, Estimator and Estimates

Suppose a set, S_1, of n response values are calculated using Monte Carlo, Latin Hypercube, or any other random sampling method and an estimate of a density parameter is obtained using that sample set. A single estimate can be calculated using sample set 1, and this estimate is termed $\hat{\theta}_1 = \hat{\theta}(S_1)$. A different estimate will be obtained using a second, third, and so fourth, sample set. In general, individual estimates of the target, $\hat{\theta}_i$, are obtained by using the estimator, $\hat{\theta}$, with the sample set, S_i, as in $\theta_i = \hat{\theta}(S_i)$. It is evident from Figure 18 that a single estimate is not enough to conclude anything about the target parameter, $\hat{\theta}$. It might be close to the target like $\hat{\theta}_2$ and the questionable $\hat{\theta}_3$, or the estimate may be far from the target like $\hat{\theta}_1$; and, unfortunately, a single estimate also gives no information about where it lies with respect to the target.

If the process of estimation is repeated for a number of repetitions, more sample sets would be drawn from the population, and more estimates would be made. In fact, the estimator would exhibit statistical characteristics and a PDF of the estimator, $PDF(\hat{\theta})$, would be the result

of the repetitions. This concept is also portrayed in Figure 18, which shows a discrete probability density function of the estimator as horizontal bars and the estimate values range along the vertical. The PDF is rotated like such to show that the origin of the PDF is repeated estimates of a statistic, and a continuous curve is shown to remind the reader that the discrete information may be fitted to a continuous distribution. In fact, an exact distribution for the estimator *does* exist and can be captured as the estimation is repeated over and over again. Furthermore, the estimator changes with the number of response evaluations, n, and the sample set, S_i. Consequently, the distribution of an estimator will be different and for different sample set sizes, n, *and*, for the each method used to obtain response sample sets. Observing the distributions of estimators and how they vary with different sample set sizes is one way to compare sampling methods.

Sampling Method Comparison

Consider a possible density of an estimator, shown in Figure 19. The estimator, $\hat{\theta}$, has been used many times and we now need to note how close the density of the estimate, $PDF\,\hat{\theta}$, clusters around the target parameter, θ. Usually, the target parameter and the density of an estimator are never known. In a typically reliability analysis a single estimate of needed density parameters are calculated and its relation to the target is never known. That is the importance of studies like this one, which capture the distribution of multiple estimates about the target parameter for the purpose of comparing MC and LHS.

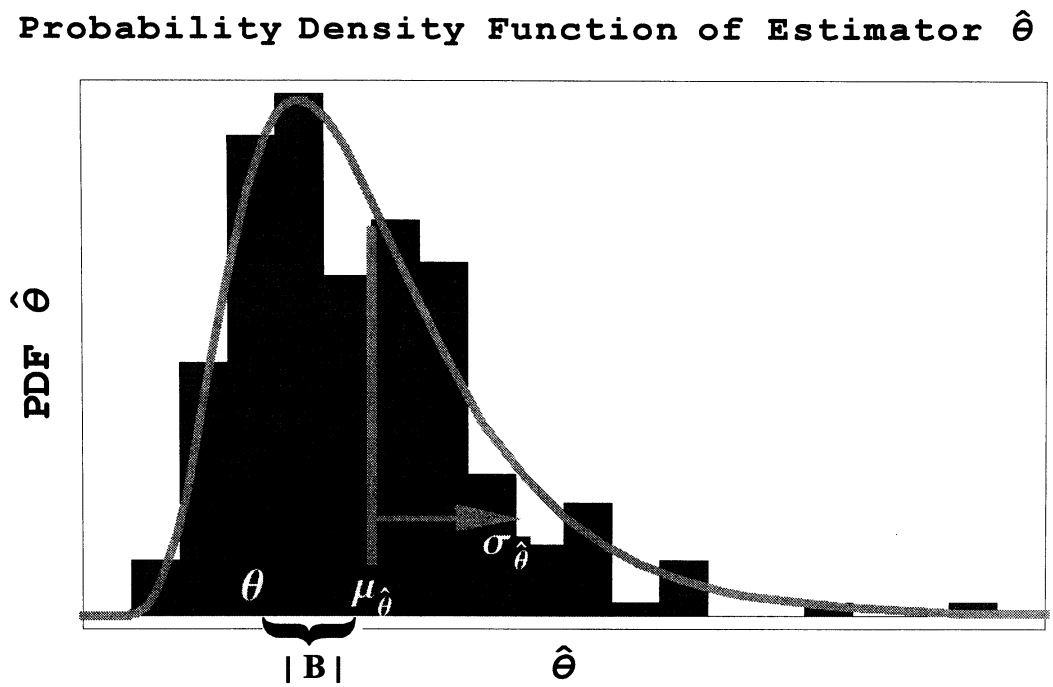


Figure 19 Possible estimator density and position with respect to target

In order to measure the goodness of a point estimator or a sampling method, we can consider the mean of the estimator density, its variation, and probabilities concerning certain regions of possible estimates. First, if the mean of the estimator density, $\mu_{\hat{\theta}}$, is not the same as the target, θ, then the estimator is *biased.* The bias of an estimator or the sampling method that uses an estimator is the difference from the mean of the estimator distribution to the target parameter. The bias is shown in Equation 18.

$$B = \mu_{\hat{\theta}} - \theta \tag{18}$$

A bias can be positive if the distribution of estimates is centered about a point above the target parameter. It can also be negative if the mean of the estimator distribution is lower than the exact value of the density parameter of interest.

Second, the standard deviation of the estimator distribution, $\sigma_{\hat{\theta}}$, can be calculated using

Equation 16, except in this case, the sample set $Z_{i=1,2,\ldots,n}$ is replaced with the set of estimates, $\theta_{i=1,2,\ldots,r}$. The subscript r is used for the later case since the number of estimates, r, has nothing to do with the number of response values used for each estimate, n. As the number of repetitions increases, the distribution of the estimator is more accurately captured. The standard deviation of the estimator distribution is also called the *standard error* of the estimator.

Finally, we discuss probabilities associated with specific ranges of estimates. Knowledge of possible ranges of estimates and their associated probabilities allow confidence statements to me made that, in effect, can be used to compare different sampling methods. Confidence statements are important because they determine the probability that one single estimate of a density parameter will lie within a specific region of possible estimates. If the exact density parameter is known, then confidence statements can be made that deal with the probability of a single estimate lying within a certain error from the target parameter.

It is known that there are three variables to consider when comparing methods by making confidence statements: (1) *effort*, number of samples, or response evaluations, used to make a future estimate, or computational time (2) *confidence*, measure of possibility that the future estimate will lie within a certain error or interval from the true value, and (3) *error* or interval that a certain confidence is placed in. In order to be able to compare methods one must set two of the variables equal to each other across the methods and compare the left over variable. For example, we can set the effort and confidence level to n=1,000 response samples and 50%, respectively, for MC and LHS. Confidence statements will be made as such: It is found that there is a 50% probability that a single mean estimate using n=1000 LHS samples will be within 0.20% from the true mean. At the same effort and confidence level, it was found that MC has an

estimation error of 0.50%. The same effort and confidence was used and it is found that LHS had a lower error than MC. Another way to compare methods would be to set the error and effort equal across the two methods and compare the confidence that a future estimate would lie within that error using a certain amount of effort. This will not be discussed in this work. On the other hand, the third and final way to compare methods like MC and LHS is to set the confidence and error equal for both methods and compare the effort required to obtain the like results. This type of statement will be used in this work. For example, it can be stated that there is a 99.7% confidence (probability) that a single mean estimate for the response of a certain system will be within ±1.5% of the true mean using n=10,000 MC samples. In comparison, there is a 99.7% chance that a single mean estimate will be within ±1.5% of the true mean using LHS-500. The type of confidence statement just made is of the type – equal confidence and error, different effort. In this case, LHS would require much less computational effort than MC when confidently estimating the mean of the response. These types of confidence statements allow random sampling methods to be compared when they are used to estimate the same parameter of the same response. Furthermore, statements like these appear throughout section 3.5, which discusses the estimation of the mean, standard deviation, and 99th percentile of 4 different responses using both MC and LHS.

3.4 NESSUS ENHANCEMENT WITH LHS ABILITIES

Introduction to NESSUS

The Numerical Evaluation of Stochastic Structures Under Stress (NESSUS) software is a probabilistic analysis tool that also has the capability of solving structural mechanics problems using the nonlinear finite element and boundary element methods. Furthermore, these two capabilities can be combined to form a complete probabilistic finite element analysis tool. The program was originally developed for the National Aeronautics and Space Administration's Glenn Research Center (NASA-GRC) by Southwest Research Institute (SwRI). NASA-GRC is located in Cleveland, Ohio, and the Southwest Research Institute location that performs NESSUS development is located in San Antonio, Texas.

The probabilistic methods include Monte Carlo, first and second-order reliability methods, convolution methods, two types of radius-based importance sampling methods, plane and curvature-based adaptive importance sampling methods, a mean value method, and two advanced mean value methods. A user can choose from a variety of design variable distributions and can even evaluate the reliability of systems with more than one failure mode.

An analyst can code up his/her own response and have NESSUS approximate its statistics, or NESSUS can be wrapped around any external code to give it the capability of performing probabilistic analysis of any response – regardless of its scientific origin.

The code was enhanced with the capability of performing Latin Hypercube Sampling. Because LHS and MC sampling perform nearly the same steps, the thread the code needs to perform MC sampling was studied so that if any preexisting subroutines, variables, actions could be performed during the LHS thread, they could be performed in the appropriate manner and at the right time.

At the time of this work, NESSUS was in a state of change from old Fortran programming techniques and abilities to new ones. Global parameters were beginning to be used and common blocks being avoided. Another important thing to mention is that the input file had been recently changed to a completely different format. As a result, the code consisted of old subroutines that used items stored in common blocks and read input from the old input file format, new subroutines that used global variables and read input from the new input file format. Yes, it is good programming practice to leave well enough along; however, sometimes changes had to be made to the original code. All the necessary changes to the original code, given to the author by the researchers in the Structural Integrity and Reliability section at Southwest Research Institute for the purpose of the LHS enhancement, are comments in the source code in the first new file that the LHS thread encounters – lhs_main.f90. The author wrote all the subroutines referenced unless otherwise noted, in which case the author of the subroutine will be given due credit.

Current State of NESSUS Monte Carlo Thread

The NESSUS Monte Carlo method has many capabilities. An analyst can analyze a response that is dependant on stochastic variables that come from 11 commonly used underlying distributions: Normal, Weibull, Lognormal, Maximum Entropy, Uniform, Frechet, Extreme Value – I, Chi-squared, Curve-Fit, Truncated Weibull, and Truncated Normal. The user may input probability levels to estimate appropriate response values, response values can be entered to estimate a probability level, and the software can calculated an entire cumulative distribution function without the user entering any percentiles or response levels. A reliability analysis of multiple failure modes can be performed with Monte Carlo sampling. The new input deck, probabilistic analysis section, is shown in Figure 20. As one can see from the shown section, an

analyst can choose the start seed used by the random number generator subroutine. The number of samples in the sample set and the maximum computational time are both user controlled. Because data acquisition is important, the user is allowed to keep all of the design variable samples used to calculate responses, the responses, in x-space, and in u-space or ordered subsets of all of the data. A histogram of the response can also be computed using the Monte Carlo analysis method.

```
*METHOD MONTE # Monte Carlo method
(MONTE)
    SEED  6974350.
    SAMPLES     100
    MAXTIME 500000
    XSKIP 1
    USKIP 1
    HISTOGRAM 20.0
*END METHOD MONTE
```

Figure 20 Monte Carlo section of NESSUS input deck

Flow of Monte Carlo Subroutine Calls

A flow chart of the subroutine calls in a NESSUS Monte Carlo analysis was useful in implementing the LHS routines. Especially if a certain variable was required to be found – all files in the source directory were searched for that name and the ones that were in the MC path were looked at first. A flow chart of the subroutine calls made during a Monte Carlo analysis using NESSUS is shown in Appendix IV-A. There are over 10 levels (or generations) of

subroutines. The MC thread begins by prompting for an input file location, opening appropriate files, printing headers to the screen and output file, initializing all variables, reading the input file and assigning values to variables, and finally, while in the level 3 subroutine *new_nessus.f90*, the MC path enters *fpi.f* – a level 4 subroutine.

Unfortunately, it is not until the code is in the level 7 subroutine of *monte.f* that the Monte Carlo sample are obtained and the typical calculations are performed. This depth of subroutines in the MC analysis made it difficult at times to correctly implement the LHS scheme because the author wished to follow the MC thread, but also leave it as early as possible, and since, a few needed variables were set down in the level 7 subroutine *inranv.f*, the author made the decision to exit early and change the original source as needed.

Monte Carlo Output

The NESSUS program will make and write to several output files. Some of the files are only made for certain probabilistic methods. For the current Monte Carlo sampling technique, the NESSUS program creates a main output file, and two optional files, as well as a command line output. An example output file as well as the subroutines that produce the written output is shown in Appendix IV-B.

The main output file, *filename.out,* gets its name from the input file, *filename.dat*, but has a different extension (.out). Portions of a MC output file are shown in Figures 21 through 25. The output file consists of the following main sections:

1. Code Title and License Agreement
2. Input Echo
3. Parameter Information
4. Model Information
5. Output Summary

The first section consists of the NESSUS title, the code version, the date that the program was used to produce that output file, license information, and all the input file main sections encountered when the program does a check to see if there are no obvious errors in the input file. This portion of the output file is shown in Figure 21. The next section is an input, or *filename.dat* echo. Because of this echo, the user does not need to keep the original *filename.dat* file that was used for the appropriate run. The current MC input echo in the output file shows the old input file format regardless of the type of input file used to run the program. It was mentioned that NESSUS is in a state of changing programming techniques and input file format. Well, some of the subroutines read data from the format of the old input file; therefore, a temporary *filename.dat* file, written in the old input file format, is needed when executing the program using the new input file format. This section of the output file, shown in Figure 22, was produced when the new input file format was used for program execution. Therefore, along the path of subroutines that the NESSUS MC method takes to perform the necessary calculations, a temporary input file is written in the old format and that is the file used to echo the input to the output file.

```
NN   NN  EEEEEEEE SSSSSSS  SSSSSSS UU     UU  SSSSSSS
NNN  NN  EE       SS       SS      UU     UU  SS
NN N NN  EE       SS       SS      UU     UU  SS
NN N NN  EEEEEE   SSSSSS   SSSSSS  UU     UU  SSSSSS
NN  N NN EE             SS       SS UU     UU       SS
NN   NNN EE             SS       SS UU     UU       SS
NN   NN  EEEEEEEE SSSSSSS  SSSSSSS  UUUUUU  SSSSSSS

DATE: 12-26-2001  12:22  - LEVEL 3.00( 39)  - DATED JUL    1, 2000
Build Date: 08/14/01 12:01:21
```

THIS IS A PROPRIETARY PROGRAM. IT MAY ONLY BE USED UNDER THE TERMS OF THE LICENSE AGREEMENT BETWEEN SOUTHWEST RESEARCH INSTITUTE AND THE CLIENT.

SOUTHWEST RESEARCH INSTITUTE DOES NOT MAKE ANY WARRANTY OR REPRESENTATION WHATSOEVER, EXPRESSED OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS OF ANY PURPOSE WITH RESPECT TO THE PROGRAM; OR ASSUMES ANY LIABILITY WHATSOEVER WITH RESPECT TO ANY USE OF THE PROGRAM OR ANY PORTION THEREOF OR WITH RESPECT TO ANY DAMAGES WHICH MAY RESULT FROM SUCH USE.

```
*TITLE SAE TEST CASE 1
*DESCRIPTION
SAE TEST CASE 1 CYCLES TO FAILURE NON-LINEAR, NON-NORMAL 4 RANDOM VARIABLES NO
CORRELATION
*ZFDEFINE
*RVDEFINE
*PADEFINE
*MODELDEFINE
*END NESSUS

End of file reached: checking data..
```

Figure 21 Header and introduction section of the MC *filename.out* file

```
1==================================================================================
                 *****   INPUT ECHO   *****
==================================================================================

 LINE

    1  *FPI
    2  NESSUS generated FPI deck: Analytical model: ANALYTICAL_1
    3  *RVNUM        4
    4  *GFUNCTION   USER
    5  *METHOD      MONTE
    6  *PRINTOPT
    7  *ANALTYP     PLEVEL
    8  *END
    9  *MONTE    1     1
   10       100          172     0.00000
   11    MAXTIME
   12     500000.
   13  *PLEVELS       20    1
   14    -5.199082    -4.753258    -4.264844    -3.719124    -3.090522
   15    -2.326785    -1.281729    -1.036431    -0.6741892   -0.1010067E-06
   16     0.6741892    1.036431     1.281729     2.326785     3.090522
   17     3.719124     4.264844     4.753258     5.199082     5.611680
   18  *DEFRANVR
   19  KIC
   20    60.00000      6.000000        NORM
   21  AI
   22    0.1000000E-01  0.5000000E-02    LOGN
   23  C
   24    0.1200000E-09  0.1200000E-10    LOGN
   25  DS
   26    100.0000      10.00000        LOGN
   27  *END

==================================================================================
==================================================================================
```

Figure 22 Input echo section of the MC *filename.out* file

The third section on the parameter information repeats the problem title and mostly summarizes what is evident from the input file or also the echo of the file. It does contain more background information about the procedures to be used during the program operation. Figure 23 shows this section as seen in the *filename.out* file. The fourth section interprets the mathematical model analyzed and is shown in Figure 24. The problem title is repeated as are the desired response levels, underlying random variable statistics as inputted by the user, the response type, the method used and appropriate parameters for that method, and options that the user can define in the input file, like writing all monte carlo sample in x or u space to a *filename.smx* or *filename.smu*, respectively.

```
==================================================================
==================================================================
1=================================================================
          *****   PARAMETER INTERPRETATION   *****
==================================================================

Problem Title:  NESSUS generated FPI deck: Analytical model: ANALYTICAL_1

Number of Random Variables:    4

Type of Response (g) Function Approximation:
   6 = User-defined response function
   Response function must be programmed in subroutine RESPON

Number of Datasets:  0

Solution Technique:
   6 = Standard Monte Carlo method (Radius = 0)
      *MONTE keyword is required in model input data

Analysis Type:
   2 = User-defined probability levels (P-levels)
      *PLEVELS keyword is required in model input data
      Time consuming analysis because of iteration procedures

Confidence Interval Calculation on CDF:
   0 = No

Print option:
   0 = Short printout

Debugging Option:
   -1 = No

==================================================================
==================================================================
```

Figure 23 Parameter interpretation section of the MC *filename.out* file

```
1==================================================================================
            *****   MODEL INTERPRETATION   *****
==================================================================================
Problem Title:  NESSUS generated FPI deck: Analytical model: ANALYTICAL_1
User-Defined Probability P-levels:
   Number      P-Level      u-level
+  ------      -------      -------

     1      0.10033E-06    -5.1991
     2      0.10021E-05    -4.7533
     3      0.10009E-04    -4.2648
     4      0.99987E-04    -3.7191
     5      0.99909E-03    -3.0905
     6      0.99883E-02    -2.3268
     7      0.99969E-01    -1.2817
     8      0.15000        -1.0364
     9      0.25010        -0.67419
    10      0.50000         0.0000
    11      0.74990         0.67419
    12      0.85000         1.0364
    13      0.90003         1.2817
    14      0.99001         2.3268
    15      0.99900         3.0905
    16      0.99990         3.7191
    17      0.99999         4.2648
    18      1.0000          4.7533
    19      1.0000          5.1991
    20      1.0000          5.6117
Random Variable Statistics:
   Random Variable  Distribution     Mean        Standard Deviation
+  ---------------  ------------     ----        ------------------

     KIC            NORMAL        60.00          6.000
     AI             LOGNORMAL      0.1000E-01     0.5000E-02
     C              LOGNORMAL      0.1200E-09     0.1200E-10
     DS             LOGNORMAL      100.0          10.00
User-Defined Response Function Equation Parameters (Sub [RESPON]) :
   Equation Number =    1
Standard Monte Carlo Method (Radius = 0):
      Minimum Sample Size =     100
                Seed =  172.000
         Allowable Error = 0.100000
     Allowable Confidence = 0.950000
      Maximum Sample Size =  2000000
   Maximum Wall Time (sec) =  500000.
      Empirical CDF Print = OFF
         Histogram Print = OFF
   X-space samples will be written to jobid.smx file. Skip factor =    1
   u-space samples will be written to jobid.smu file. Skip factor =    1
==================================================================================
==================================================================================
```

Figure 24 Model interpretation section of the MC *filename.out* file

The fifth section summarizes all results and is termed the output summary section. The output summary section of the *filename.out* file is shown in Figure 25. The problem title is repeated along with the type of response analyzed. It is worthwhile to mention that approximate statistics of the response are calculated and shown in the *filename.out* file. These statistics would be the mean and standard deviation of the response, approximated by fitting the actual response to a first order surface with the mean of all random variables as the base point. This requires R+1 additional response evaluations, where R is the number of random variables, in order to calculate the response and gradients at the base point. This is necessary information used to obtain this type of curve fit; however, depending on the computational time needed to obtain each response and the method used to obtain a solution, it could be an expensive and unnecessary step. The output file then repeats some already shown information: the method, the number of samples, and the number of variables.

```
1==========================================================================
              *****   OUTPUT SUMMARY   *****
==========================================================================



PROBLEM TITLE: NESSUS generated FPI deck: Analytical model: ANALYTICAL_1

RESPONSE FUNCTION (LIMIT STATE): USER-DEFINED FUNCTION
                    IN SUBROUTINE [RESPON]

APPROXIMATE STATISTICS FOR Z:
     MEDIAN_Z = Z(MEDIAN_X) =  15747.1

     Z_Approx = 1st Order Taylor Series of Z about MEAN_X
     Normal Format on Z_Approx:

     MEAN_Z = Z(MEAN_X)
     MEAN_Z =  14189.2

     STD_DEV_Z = SQRT [ SUM [((dZ/dXi)*STD_DEV_Xi )^2] ]
     STD DEVfl_Z =  7085.86


NOTE:  Standardized Normal Variates are used in the following analysis.
     This means that the random variable, u, represents a normal
     probability distribution with mean = 0 and standard
     deviation = 1.  For example, u = -3 implies that the chance
     of observing a u value <= -3 is .00135 (cdf).  Also, u = 3
     implies that the chance of observing a u value <= 3 is 0.99875.


NUMBER OF SAMPLES FOR PLEVELS ANALYSIS:      100

MONTE CARLO SOLUTION:
  NUMBER OF VARIABLES =   4
  NUMBER OF SAMPLES =      100
  SAMPLE MEAN =   1.77381E+04
  SAMPLE STD. DEV. =   8.94694E+03

RANDOM VARIABLE STATISTICS:

 Random    Input      Input      Sample     Sample    % error   %error
Variable   Mean      Std. Dev.    Mean     Std. Dev.   Mean    Std. Dev.
--------  -------  ----------  --------  ----------  --------  ----------
KIC        60.00      6.000     60.43      6.013      0.71      0.22
AI       0.1000E-01 0.5000E-02 0.9704E-02 0.5273E-02   2.96      5.46
C        0.1200E-09 0.1200E-10 0.1207E-09 0.1241E-10   0.54      3.38
DS         100.0      10.00      99.84      8.715      0.16     12.85

CDF SUMMARY
  Pr(Z<=Z0)        u            Z0            #Pts<=Z0     Error(*)
-------------  -------------  -------------  ----------  -------------
 0.1500006     -1.036431      10112.86          15     0.4665636
 0.2500954     -0.6741884     12352.94          25     0.3393892
 0.9999999      5.199082      53208.12         100     0.6208121E-04
 1.000000       5.611680      53208.12         100     0.1964426E-04
***************************************************************************
Probabilistic Sensitivity Results printed by level
Level=  6 Z0= 3698.51    CDF=0.998833E-02 No. Failure Samples=      1
                 d(p)       d(p)       d(p) sig      d(p)  sig
 Random         -----      ------     -----*---     ------*---
 Variable       d(mu)      d(sig)     d(mu)  p      d(sig)  p
---------------------------------------------------------------------------
 KIC        0.5471E-03  -0.1485E-02   0.3286       -0.8920
 AI         2.660        1.488        1.332         0.7446
```

Figure 25 Output summary section of the MC *filename.out* file

After all of what's been mentioned, the output file then shows actual results that might be of interest to an analyst. In Figure 25, the mean and standard deviation of the response based on the response evaluations obtained by using a user-defined number of MC samples is shown. This is followed by a brief table of the input mean and standard deviation of the underlying random variables for which the response is dependent on, the same sample statistics obtained from an array of MC samples, and the error between the respective values. This is a quantitative statement about the capturing the individual random variable distributions. The joint probability density function would be estimated from the samples by appropriate pairing of the random variable with each other to obtain coordinates in R-dimensional space. After this error check is shown, a table of the cumulative density function at user-defined points is shown. The table shows a probability level and the appropriate standard normal level, response level, number of response elements under that level, and a sampling error for every level of the CDF that the user specified in the input file. Sampling sensitivities are shown next for every user specified CDF level. The available calculated sensitivities are the change in probability with respect to the mean and standard deviation of each underlying random variable, and the same two sensitivities multiplied by the ratio of the respective underlying random variable to the probability level. The last thing to printed to the output file that is not shown in Figure 25 is the cpu computational time from program execution to finish.

There are two optional files that the user has the capability of creating during a MC analysis through a flag in the input file. They both list the underlying random variable (URV) samples used to obtain samples of the response. One of them lists the samples in the original or x-space of the URVs and the other lists the same samples in u-space or standard normal space.

The latter is obtained replacing each x-sample from every random variable with the u-value calculated by inverting a standard normal CDF at the same probability level as the x-sample.

The file that contains the underlying random variable samples in x-space that were used to obtain the response samples is the *filename.smx* file. A portion of this file is shown in Figure 26. This Figure shows the file header, which describes the format of the file. It also shows the filename of the input file with no extension – here the *filename* is shown to be c1M_1h_1. The individual underlying random variable x-space samples and response are shown row-by-row, for every sample taken. For this analysis, there are four random variables that the response is dependent on. The first column of the first row shows the coordinate of the first random variable, the second column of the first row shows the coordinate of the second random variable, and so on, up to the fourth column which is the last random variable. These first four columns of the first row make up one coordinate in the multivariate space that is the domain of the response; therefore, using this coordinate in 4-D space a response is calculated and shown in the fifth column of the first row. This continues for all of the samples taken in the analysis.

```
# FORMAT: DESCRIPTION,TITLE,JOBID,#LEVELS,#RVS,#GFNS,XPTS(1:N),GFNS(1:M)
# X-SPACE SAMPLES AND G FUNCTION RESULTS
# TITLE: NESSUS generated FPI deck: Analytical model: ANALYTICAL_1
# JOBID: c1M_1h_1
#    1    4    1
 41.99481    0.1039068E-01    0.1428974E-09    112.3249    5649.685
 62.08616    0.1088208E-01    0.1045636E-09    106.9302    12274.04
 65.94169    0.1262470E-01    0.1425862E-09    95.48286    12352.94
 68.68149    0.1914397E-01    0.1096084E-09    87.97140    15973.79
```

Figure 26 The *filename.smx* file that contains x-space samples

The file that contains the coordinates of the underlying random variable in u-space is called the *filename.smu* file. A portion of this file is shown in Figure 27. The file header describes the format of the data, the title of the analysis, and the original *filename* with no extension, which in this case is c1M_1h_1. The u-space samples shown are obtained by

inverting the standard normal CDF at a probability level equal to the aprobability level of the CDF of the respective x-space sample using the actual distribution type and parameters of that underlying random variable as specified in the *filename.dat* input file. The first column of the first row is the u-space sample of the first random variable and the second column of the same row is the u-space sample of the second random variable, and so forth, up to the fourth column. All four columns make up one coordinate in a 4-D space.

```
# FORMAT: DESCRIPTION,TITLE,JOBID,#LEVELS,#RVS,#GFNS,UPTS(1:N)
# U-SPACE SAMPLES
# TITLE: NESSUS generated FPI deck: Analytical model: ANALYTICAL_1
# JOBID: c1M_1h_1
#    1    4    1
 -3.000865    0.3173205    1.800579    1.215031
 0.3476939    0.4151398   -1.330517    0.7216054
 0.9902823    0.7295850    1.778722   -0.4135108
  1.446915     1.610935   -0.8581620   -1.234903
```

Figure 27 The *filename.smu* file that contains u-space samples

Output is also written to the standard output stream of the computer being used, which is either a DOS or UNIX window. The output is written to the screen as the analysis is being performed and is mostly a repeat of what is already written to the *filename.dat* or *filename.out* files. While it is not shown due to its length, it consists of quite a few sections. The output screen shows a program header similar to the one shown in Figure 21, response and random variable information, followed by a CDF summary of the response and the total cpu time elapsed during the analysis.

New Latin Hypercube Thread

The first thing that was dealt with in the addition of a new statistical method via more subroutines, defining new global variables, and all other related tasks was to determine what flag to use in the input file that indicates the Latin Hypercube method and the location of the call to the first new LHS subroutine in the LHS thread. Since the tasks for MC and LHS are almost identical and they would require the same initial input information from any input file, the LHS section in the *filename.dat* file is identical to the MC section, except for two "LHS" keywords. The LHS section of the input file is shown in Figure 28.

```
  *METHOD LHS # Latin HyperCube Method
(LHS)
     SEED     172.
     SAMPLES     100
     MAXTIME 500000
     XSKIP 1
     USKIP 1
     EMPCDF
     HISTOGRAM 20.0
  *END METHOD LHS
```

Figure 28 Latin Hypercube section of NESSUS input deck

The only difference between the type of input needed for the Monte Carlo method and Latin Hypercube method is that for the latter case, the input file contains the key lines *METHOD LHS and *END METHOD LHS instead of *METHOD MONTE and *END METHOD MONTE. The seed that starts the random number generator can be specified by the user along with the total number of samples taken during the analysis. The key lines XSKIP 1 and USKIP 1 indicate that samples should be written to appropriate output files. Unfortunately, the options of specifying a maximum analysis time and printing a histogram output are not yet available when using the newly implemented LHS method.

Flow of LHS Subroutine Calls

The path that NESSUS takes when the LHS method is used an analysis is the same as the Monte Carlo method up to a point in the code. After this, the LHS thread becomes unique to that method.

The first file encountered by the program when LHS samples are taken is the *nessus.f* program file that quickly calls the *nesmain.f* subroutine. In other words, the *nessus.f* subroutine is the parent of its child, *nesmain.f*. The *nesmain.f* subroutine performs quite a few initialization tasks by calling other subroutines. These tasks include prompting the user for input, opening the necessary files for execution, and writing headers to some of the files. A call is then made to the *new_nessus.f90* subroutine. This subroutine and all of its children and so on down the line is only called if the new input deck format is used. The relationship between the files just mentioned is portrayed in Figure 29. This Figure shows the parent subroutines on the left and their respective children to the right of the parent.

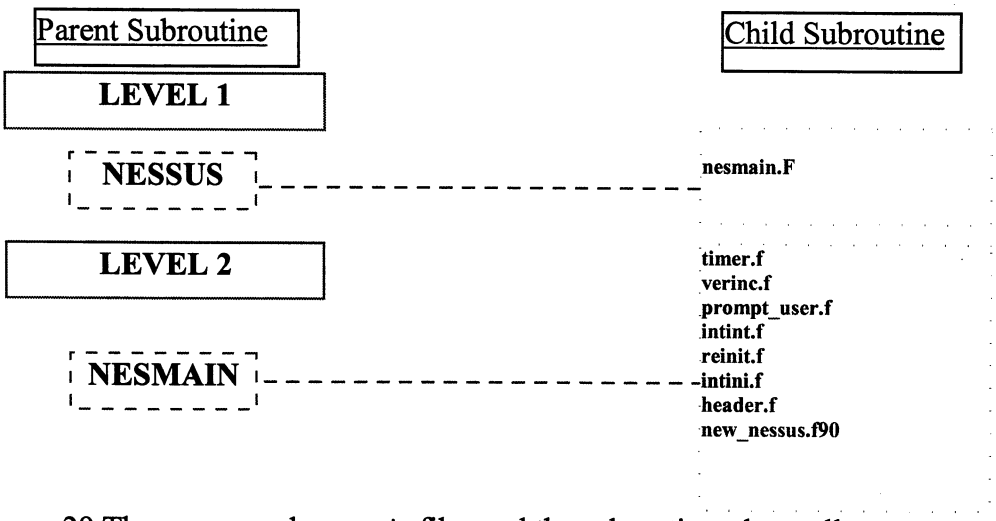


Figure 29 The *nessus* and *nesmain* files and the subroutines they call

From this point on, the word subroutine will be not be used and unless otherwise specified, an italicized name will be that of a subroutine. When *new_nessus* is then entered several subroutines are called that perform specific tasks. The path to the working directory is set, and global variables are cleared or initialized by setting them equal to 0, a NULL string, or the FALSE logical operator, depending on the variable type by calls to *set_working_directory* and *init_input*, respectively. The parameters in the new NESSUS input file are then read when the call to *read_nessus_input* is made. *Model_setup* is entered where a file is opened but is not used for the LHS method. Finally, the LHS breakaway point is encountered. It is in - *new_nessus* and right before the *fpi.f* call, which is never made when the LHS method is used. After the breakaway point, nothing else happens in the program; therefore, the LHS thread can stop anywhere in its unique path without missing any tasks that would have been performed by returning from the LHS thread and continuing on with the program execution.

The breakaway call is made to *lhs_main* with no arguments. The file containing this subroutine, and all other files with subroutines that will be mentioned in this section are shown in Appendix IV-C. This type of clean LHS call was important because the only dependencies that the LHS thread has with what has been previously performed in the program up to this point are through the variables that were set by reading input from the new input deck format. Yes, the LHS method will only work with a new input deck format. The next set of parent-child subroutine relationships are shown in Figure 30.

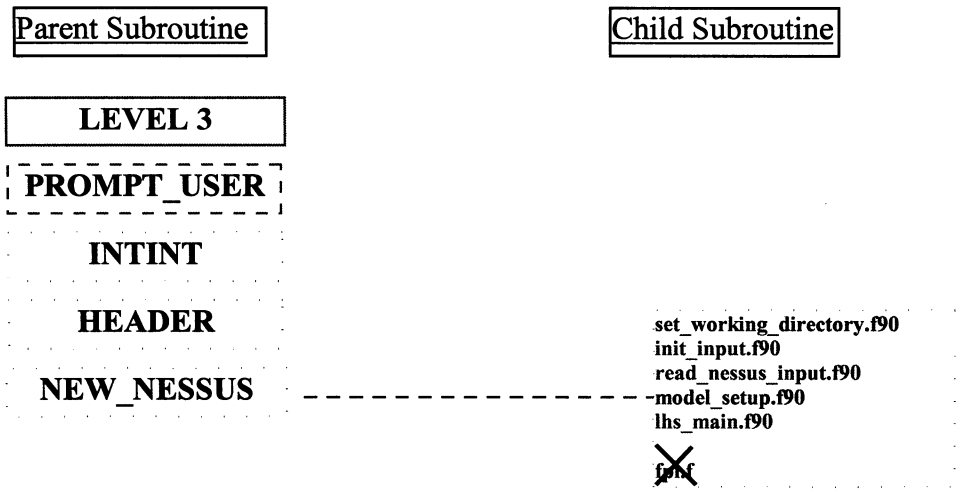


Figure 30 Third level of LHS subroutines. *Fpi.f* is crossed out because it is never entered

The two main subroutines encountered in *lhs_main* are *lhs_xsample*, which obtains LHS samples, and *lhs_calc*, which performs calculations with those samples. Obtaining LHS samples and performing calculations with these samples are the two main steps in the NESSUS LHS algorithm. Another new subroutine encountered within the LHS thread is named *write_files*. It writes output to the appropriate files without any arguments. Global variables are set right before the call to *write_files* that indicate what to write, and which files to write it to. These variables are then reset to their initial state at the end of the *write_files* subroutine. The children of *lhs_main* are shown in Figure 31.

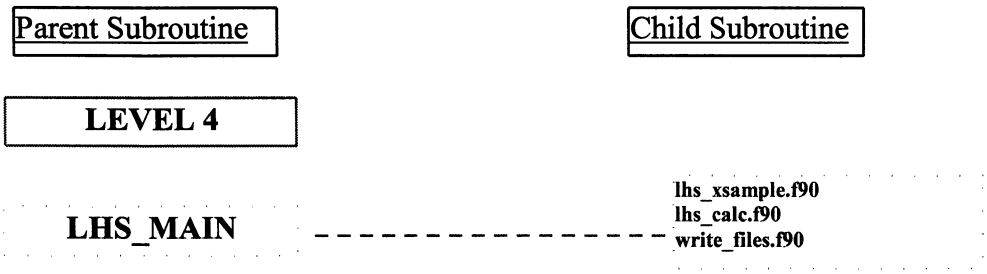


Figure 31 Child subroutines of *lhs_main*

LHS samples are obtained for each random variable by obtaining an array of samples from that random variable. The length of the array is the number of samples to be used in the

analysis, as entered in the *filename.dat* file. The samples from each random variable are then paired with each other in such a manner that the desired correlation between pairs of the random variables, also entered in the *filename.dat* file, is obtained to within a certain degree. Thus, the samples are arranged to achieve the desired correlation. Once the samples are paired with each other they become coordinates in the multivariate space that is the domain of the response to be analyzed. These steps are performed in *lhs_xsample*, which is in level 5 of the LHS thread.

The calculations performed after LHS samples are obtained are simple. The response is evaluated using the samples previously obtained as its inputs. The mean and standard deviation of the response based on the samples taken is then calculated. The response is then sorted. After this, the response value and its corresponding probability level entered in the *filename.dat* file are then written to output streams or files. The code is then stopped and the analysis is complete. These steps are performed in *lhs_calc*, where the command to stop the program is also located.

There will always be a difference between what is desired from a program and how it is implemented. The writing of a program should be neat and simple, and also take necessary actions to minimize the computational time for each analysis. Therefore, in order to have a full understanding of the new subroutines, *lhs_xsample* and *lhs_calc*, they will now be discussed in further detail.

The lhs_xsample.f90 Subroutine

All of the steps necessary to obtain LHS samples are not performed by *lhs_xsample* itself. Some calls are made to a combination of already existing subroutines written by several employees of Southwest Research Institute, and new subroutines written by either the author or Randall Manteufel, both from the University of Texas at San Antonio. A depiction of *lhs_xsample* and its children is shown in Figure 32.

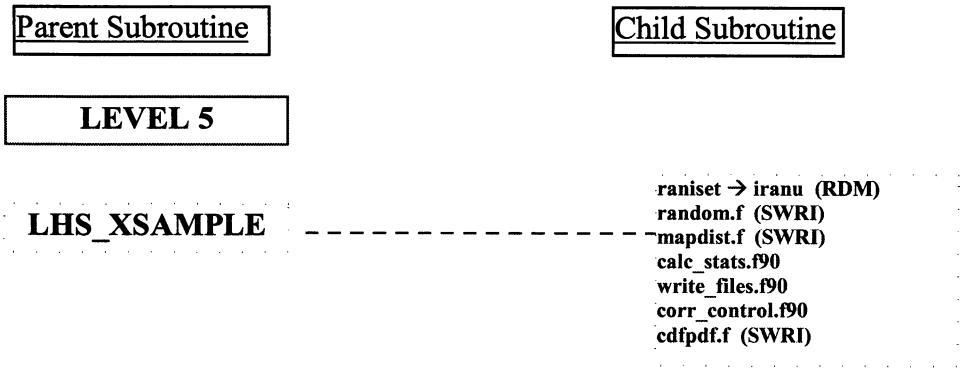


Figure 32 The *lhs_xsample* and its children

The first operation performed in *lhs_xsample* is to obtain LHS samples of each random variable. The number of samples taken from each random variable is the same as the number of response evaluations to be calculated, as entered in the *filename.dat* file. For each random variable, one sample is taken from equal probability bins that are non-overlapping, and span all of the probability range of the variable – from 0 (0%) to 1.0 (100%). This is a stratified sampling without replacement. Therefore, the first step is to divide the probability space (0,1) into bins of size 1/n, where n is the number of response evaluations to be calculated. A sample from the first bin, or strata, will be between 0 and 1/n. A sample from the jth bin will be between (j-1)/n and j/n, for all j between 1 and n. Thus, preserving the equal probability of simple events assumption of the relative frequency probability concept. This is performed in *lhs_xsample* by

first randomly filling an array with integers ranging from 1 to n. The subroutines *raniset* and *iranu* are used to do this and they are located in the same file as *lhs_xsample.* The *raniset* and *iranu* subroutines were written by Randall D. Manteufel, the Chair of the thesis committee for this work. A random number between 0 and 1.0 is then generated by *random* (SwRI) and used as the percent increase from the low and high limits of the bin number that is the first entry of the array of integers. This is done for all values of the integer array and the results are stored in another array. Thus, the cumulative probability values corresponding to yet unknown random variable values of the first variable are known and stored. This is done simultaneously for all random variables in about the first 10 lines of code in *lhs_xsample* (not including comments).

The cumulative probabilities are then used to obtain the associated random variable values by inverting the cumulative probability distribution of the respective random variable. This is done by a call to *mapdist*, an existing NESSUS subroutine written by a programmer at Southwest Research Institute. At this point the program has an array filled with coordinates of a multidimensional space that could be used to obtain response values. However, due to the random nature of obtaining the samples, spurious correlation between variables might exist where none is desired, or where a different correlation is desired.

The sample statistics of each random variable are then calculated and written to a file with calls to *calc_stats* and *write_files*. Also, the statistics of the cumulative probability values are calculated using the same two subroutines. Thus, an analyst will have the opportunity to check that the mean and standard deviation of each random variable, as entered in the *filename.dat* file, is recaptured. The mean and standard deviation of the cumulative probability values of each random variable should be 0.5 and 0.289, respectively. This is merely a check to see if the cumulative probability values are uniformly distributed between 0 and 1. The array of

random variable values and their respective cumulative probability values are written to files.

The random variables are then rearranged with respect to each other in order to obtain the desired correlation, as entered in the *filename.dat* file, or zero correlation if none is entered. This is performed by a call to *corr_control*. The whole array of random variable values that will be used to evaluate the response is passed into this subroutine. The first actions of *corr_control* are rearranging the correlation values that were input by the user into an array in the proper order of the random variables as known by the variable rv_def(j)%name, which stores the names of the random variables. That is, if rv_def(2)%name= 'Kic' and rv_def(3)%name='ai', then the correlation matrix value corr_desired(3,2) should be the desired correlation between the variables Kic and ai, as entered by the analyst. The reason this initial rearrangement must be performed is that a user can enter the desired correlations in any order and the results will be stored in the variables corr_def%rv(j,k) and corr_def%coef(j), where, if there is correlation between at least one pair of random variables, the j subscript goes from 1 to the number of correlated pairs and the k subscript goes from 1 to 2 – for the first and second random variable in the correlated pair. The process of inducing correlation is easier if the array storing the desired correlation between the random variables stores them in the same order as the array storing the samples to be arranged, along the dimension of that sample array whose index implies a certain random variable.

This desired correlation matrix $[C]_{rxr}$ is then accepted as the rank correlation of the random variables, [C*]. Cholesky decomposition in *corr_control* then produces a lower and upper triangular matrices, [P] and [P'], respectively. A matrix $[R]_{nxr}$ is found such that its rank correlation matrix is [I], the identity matrix. The length of each column of [R], or rather its size along the first dimension is the number of response samples to be taken, n. The number of

columns is the number of random variables, r. It has been suggested by Iman, et. al. (1982) to use the Van der Waerden scores for [R]. This publication also serves as the basis for the method the new LHS thread uses to correlate the variables. The first column of matrix [R] are the Van der Waerden scores for the first random variable, the second column of matrix [R] contains the scores for the second random variable, up until the last column of [R], which contains the scores for the last random variable. The Van der Waerden scores are a random placement of the values $\Phi^{-1}[i/(n+1)]$ for i=1,2,…n; and, this would be done for every random variable, or column of [R]. The rank correlation of [R] is then approximately [I], the identity matrix. The matrix [R] is then post multiplied by the upper triangular matrix [P'] and the matrix [R*]=[R][P'] is produced and its rank correlation matrix [M] is close to [C*], the target rank correlation matrix that was accepted to be [C], the user-entered correlation matrix. Therefore, if the array containing the samples of the underlying random variables, [X], is arranged so that its ranked order of each random variable is identical to the ranked order of [R*], the rank correlation of [X] will also be [M], which is close to [C*]=[C]. The rank correlation matrix of [X] is then close to that which the user entered in the *filename.dat* file. This rearrangement of [X] is the last step in *corr_control*, which then passes [X] back to *lhs_xsample.*

The next step the LHS thread takes in *lhs_xsample* is to obtain the cumulative probability of each of the random variable sample points based on their respective distribution type and entered parameters. This is done with a call to cdfpdf, written by SwRI. The mean and standard deviation of each random variable sample set in the array of the random variables and the mean and standard deviation of its cumulative probability array are then calculated and written to files by calls to *calc_stats* and *write_files*, for each array just mentioned. After this the cumulative probability array is written to a file. The random variable sample array is not yet written to a file

because some additional information still needs to be written to that file. The *lhs_xsample* subroutine then passed control to the *lhs_main* subroutine, which prints an output header to the screen and the *filename.out* file and then calls *lhs_calc* to finish the LHS analysis.

The *lhs_calc.f90* Subroutine

The only calling argument of *lhs_calc* is the array of samples of the underlying random variables that have already been sorted to approach the desired correlation as inputted by a NESSUS user. The first step performed by this subroutine is the calculate the response for the system under study based on the values in the array passed in that are the coordinates in the multidimensional space which the response exists over. The response is calculated with a call to *evaluate_models,* written by SwRI. The coordinates and response value are then written to the appropriate output file and this is repeated for the number of user specified response samples to be taken. The result is thus a vector of response values, which can be used to estimate the density of the response. Consequently, these values can also be used to estimate the parameters of the density.

The mean and standard deviation of the response vector is calculated with a call to *vector_stats*. These statistics are then written to the console and main output file. The response vector is then assigned to a temporary variable and sorted from least to greatest in *qsort* (SwRI). The percentile, z_p, of the system associated with the probability level p is calculated using this sorted list. The desired probability levels for which an analyst would seek a percentile for are not entered as probabilities in the *filename.dat* file. They are entered as standard normal u values. These u values have an associated cumulative probability of occurring that is obtained from the cdf of a standard normal variable. Therefore, the first step in calculating the response percentiles is to obtain these probability values associated with the user entered standard normal

u values. This is done with a call to *cdfnof.* Now, each probability value, p, is used to calculate the element of the sorted sample set for which p-percent of the total number of samples taken is equal to or below this value. If this number is not an integer the number is rounded to the nearest integer and used to locate the response percentile, or z-value, in the sorted vector that contains all the response calculations. So, for all probability levels entered in the *filename.dat* file, the respective percentile for the system under study is calculated and written to the console and main output file. The subroutine *lhs_calc* is exited, control is passed back to *lhs_main*, and the program is stopped; thus, concluding the analysis calculations, with all necessary output written to the respective files.

Latin Hypercube Output

A NESSUS LHS analysis will result in output to several files as well as the standard output stream – the console, which contain all the information about the problem being solved, some intermediate information about the samples used to for calculations, as well as the result of the calculations. For the new LHS sampling technique, the NESSUS program creates a main output file, and four files that contain LHS sample information, as well as a command line, or console output.

The main output file for an LHS analysis, *filename.out,* also gets its name from the input file, *filename.dat*, but has a different extension (.out). Portions of a latin hypercube output file are shown in Figures 33 through 36.

The output file consists of the following main sections:

1. Code Title and License Agreement

2. Input Echo

3. Output Summary

The first section of the *filename.out* file consists of the NESSUS title, the code version, the date that the program was used to produce that output file, license information, and all the input file main sections encountered when the program does a check to see if there are no obvious errors in the input file. It is the same as what would be seen in the first part of the *filename.out* file in a monte carlo output. This portion of the output file is shown in Figure 33.

```
NN   NN  EEEEEEEE SSSSSSS  SSSSSSS UU    UU  SSSSSSS
NNN  NN  EE       SS       SS      UU    UU  SS
NN N NN  EE       SS       SS      UU    UU  SS
NN N NN  EEEEEE   SSSSSS   SSSSSS  UU    UU  SSSSSS
NN  N NN EE             SS       SS UU    UU       SS
NN   NNN EE             SS       SS UU    UU       SS
NN   NN  EEEEEEEE SSSSSSS  SSSSSSS  UUUUUU  SSSSSSS


DATE: 12-26-2001  12:22  - LEVEL 3.00( 39)  - DATED JUL    1, 2000
Build Date: 08/14/01 12:01:21



THIS IS A PROPRIETARY PROGRAM. IT MAY ONLY BE USED UNDER THE TERMS
OF THE LICENSE AGREEMENT BETWEEN SOUTHWEST RESEARCH INSTITUTE AND THE
CLIENT.

SOUTHWEST RESEARCH INSTITUTE DOES NOT MAKE ANY WARRANTY OR
REPRESENTATION WHATSOEVER, EXPRESSED OR IMPLIED, INCLUDING ANY WARRANTY
OF MERCHANTABILITY OR FITNESS OF ANY PURPOSE WITH RESPECT TO THE
PROGRAM; OR ASSUMES ANY LIABILITY WHATSOEVER WITH RESPECT TO ANY USE OF
THE PROGRAM OR ANY PORTION THEREOF OR WITH RESPECT TO ANY DAMAGES WHICH
MAY RESULT FROM SUCH USE.


*TITLE SAE TEST CASE 1
*DESCRIPTION
SAE TEST CASE 1 CYCLES TO FAILURE NON-LINEAR, NON-NORMAL 4 RANDOM VARIABLES NO
CORRELATION
*ZFDEFINE
*RVDEFINE
*PADEFINE
*MODELDEFINE
*END NESSUS

End of file reached: checking data..
```

Figure 33 Header and introduction section of the LHS *filename.out* file

The next section is an input, or *filename.dat*, echo. Because of this echo, the user does not need to keep the original *filename.dat* file that was used for the appropriate run. The current monte carlo input echo shows the old input file format; however, the new LHS method will result in an echo of the new input file format. The input echo section of the LHS output file is shown in Figures 34 and 35. This input echo is shown in two Figures only because it could not be shown on a single page.

```
1==================================================================================
                    ********** INPUT ECHO **********
==================================================================================
LINE
   1  *NESSUS
   2  # Generated by NESSUS GUI, version: 2.9.1 (Build 123)
   3  # Date generated: Wed Sep 19 11:38:50 GMT+01:00 2001
   4
   5  *TITLE SAE Test Case 1
   6  *DESCRIPTION
   7  SAE Test Case 1 Cycles to Failure Non-Linear, Non-Normal 4 random variables No
   8  correlation
   9  *END DESCRIPTION
  10
  11  #
  12  # Problem Statement:
  13  #       g=(af**temp-ai**temp)/c/(1.1215*ds)**ParisM/CPi**(ParisM/2.0)
  14  #       /temp
  15  #       af=1.0/CPi*(Kic/1.1215/ds)**2.0
  16  #       temp=1.0-ParisM/2.0
  17  #       CPi=3.1415926535
  18  #       ParisM=3.0
  19
  20  #
  21  # Z-function definitions
  22  #
  23  *ZFDEFINE
  24    *MODEL analytical_1
  25  #       g=(af**temp-ai**temp)/c/(1.1215*ds)**ParisM/CPi**(ParisM/2.0)
  26  #       /temp
  27      *TYPE ANALYTICAL
  28        CPi ParisM ai c ds temp af
  29      *END TYPE
  30      *CVARIABLE g
  31      *END CVARIABLE g
  32    *END MODEL analytical_1
  33    *MODEL analytical_2
  34  #       af=1.0/CPi*(Kic/1.1215/ds)**2.0
  35      *TYPE ANALYTICAL
  36        Kic ds CPi
  37      *END TYPE
  38      *CVARIABLE af
  39      *END CVARIABLE af
  40    *END MODEL analytical_2
  41    *MODEL analytical_3
  42  #       temp=1.0-ParisM/2.0
  43      *TYPE ANALYTICAL
  44        ParisM
  45      *END TYPE
  46      *CVARIABLE temp
  47      *END CVARIABLE temp
  48    *END MODEL analytical_3
  49  *END ZFDEFINE
  50
```

Figure 34 Input echo section of the LHS *filename.out* file

```
51  #
52  # Variable definitions and mappings
53  #
54  *RVDEFINE
**
58    *DEFINE Kic
59  #     Mean   Stdev  Type
60        60.0   6.0    Normal
61    *END DEFINE Kic
**
84  # Random variable correlations
85  #
86    *CORRELATIONS
87          Kic,ai,0.0
**
93    *END CORRELATIONS
94  *END RVDEFINE
95
96  #
97  # Probabilistic analysis settings
98  #
99  *PADEFINE
100     *METHOD LHS # Latin HyperCube Method (LHS)
101       SEED     172.
102       SAMPLES      100
103       MAXTIME 500000
104       XSKIP 1
105       USKIP 1
106       EMPCDF
107       HISTOGRAM 20.0
108     *END METHOD LHS
109     *ANALYSIS_TYPE ULEVEL
110 #       Values are standard normal
***
117       -1.281728756502709
***
131     *END ANALYSIS_TYPE
132 *END PADEFINE
133
134 #
135 # Model definitions
136 #
137 *MODELDEFINE
138 *MODEL analytical_1
139  (af**temp-ai**temp)/c/(1.1215*ds)**ParisM/CPi**(ParisM/2.0)/temp
140 *END MODEL analytical_1
141 *MODEL analytical_2
142  1.0/CPi*(Kic/1.1215/ds)**2.0
143 *END MODEL analytical_2
144 *MODEL analytical_3
145  1.0-ParisM/2.0
146 *END MODEL analytical_3
147 *END MODELDEFINE
148 *END NESSUS
```

Figure 35 Input echo section of the LHS *filename.out* file continued

The lines of the input file are numbered in the echo and, as one can see from Figures 34 and 35, not all of the input echo is shown. This is only to conserve space. Any line that shows a # after the line number is a comment in the *filename.dat* file. This input echo shows the non-comment portions of the input file to be an initial title and problem definition section. The next two sections are the response (z-function) and random variable definition sections. The final two sections are the probabilistic analysis and model definition sections.

The final part of the output file is shown Figure 36. This is the part containing an output summary of the calculations performed during an LHS analysis. The method is written to this part of the file, along with the number of underlying random variables and number of response samples calculated. The mean and standard deviation of the values of each random variable used to evaluate the response is calculated in an LHS analysis along with their errors with respect to what was inputted by the user. These values are written to the *filename.out* file. The next part of the output summary section are what an analyst would be mainly concerned with. This part shows the mean and standard deviation of all of the response values calculated. A cdf summary is also shown. This cdf summary shows the cumulative probability and its respective standard normal u-value for this cumulative probability and response value at this cumulative probability. The number of response samples less than or equal to this response value is also written to this file. There is also a column showing the error at this probability level. This error calculation is not available for the LHS method of analysis. A row containing all of this cdf information would be shown in the output file for all probability levels entered by the user. Some lines of this section are not shown, but they are only repeats of the same calculations for other underlying random variables or probability levels.

```
1==========================================================================
              ********** OUTPUT SUMMARY **********
==========================================================================

 LATIN HYPERCUBE SOLUTION
  NUMBER OF VARIABLES   4
  NUMBER OF SAMPLES     100

 RANDOM VARIABLE STATISTICS
 Random   Input          Input          Sample      Sample          % error    % error
 Variable Mean           Std. Dev.      Mean        Std. Dev.       Mean       Std. Dev.
 ---------------------------------------------------------------------------------------
 KIC      .600000E+002   .600000E+001   .599831E+002  .604878E+001  .282526E-001  .806490E+000
** Skipped some lines

 RESPONSE STATISTICS
  Response                Response
  Mean                    Std. Dev.
  ---------------------------------------
  0.1728057625E+005       0.8975426443E+004

  CDF SUMMARY
   Pr(Z<Z0)        U              Z0          #Pts<=Z0       Error(*)
 ---------------------------------------------------------------------------------
 ** Skipped some lines.
  0.25010E+000   -.67419E+000   0.10352E+005      25       NA
  0.50000E+000   -.10101E-006   0.15884E+005      50       NA
 ** Skipped some lines
```

Figure 36 Output summary of the LHS *filename.out* file

Another output file that an LHS analysis produces is the *filename.lpr* file. The extension gives an analyst a clue as to the contents of this file. The file contains information about the latin hypercube cumulative probability values that are random, and for each random variable initially obtained. A portion of the *filename.lpr* file is shown in Figure 37. For each random variable in an LHS analysis, n number of cumulative probability values between 0 and 1 are obtained in a manner such that one value is at a random location within n non-overlapping bins that completely span the 0 to 1 probability space. These cumulative probability values for all the random variables are then randomly paired up with one another to form coordinates in probability space.

```
# Latin Hypercube Sampling Matrix File
# JOBID: *TITLE
# For each row(1:# Samples =      100) : Input_Vector(1:#RVs =  4) GNFS(1:#GFNS=  3)
# These are RANDOM SAMPLES with SPURIOUS CORRELATION between variables.
# LHS_PROB_SAMPLES :: Randomly sample from each probability bin and randomly pair up coordinates


MEAN of SAMPLE (by columns = random variable)
  0.5005148035E+000    0.5001036996E+000    0.4999802384E+000    0.4998676664E+000

STANDARD DEVIATION of SAMPLE (by columns = random variable)
  0.2897709240E+000    0.2905316145E+000    0.2897500653E+000    0.2902432264E+000

CORRELATION COEFFICIENT MATRIX (Linear)
  0.9900000000E+000
 -0.2027871831E+000    0.9900000000E+000
 -0.1198529373E+000    0.5176967792E-001    0.9900000000E+000
  0.1144922854E+000    0.4847054425E-001   -0.1639291176E+000    0.9900000000E+000

SPEARMAN RANK CORRELATION COEFFICIENT MATRIX
  0.1000000000E+001
 -0.2038163816E+000    0.1000000000E+001
 -0.1201560156E+000    0.5376537654E-001    0.1000000000E+001
  0.1162436244E+000    0.5111311131E-001   -0.1661806181E+000    0.1000000000E+001

***** SAMPLES *****
  0.4216206467E+000    0.8582091978E+000    0.6819894556E+000    0.4667808370E+000
  0.8552980312E-001    0.1994025926E+000    0.7593756445E+000    0.6464600554E+000
  0.7941528279E+000    0.1465789530E+000    0.3724654824E+000    0.1073635109E+000
** The rest of the file is not shown.
```

Figure 37 LHS *filename.lpr* file that contains p-space LHS random samples

The *filename.lpr* file contains a brief header and short problem description. A row containing the mean of the LHS cumulative probability values is then shown, followed by one showing the standard deviation of the values. It is a requirement from the relative frequency standpoint of probability that all values in a sample set have an equal probability of occurrence; thus, it should be seen in drawing samples that the cumulative probability range from 0 to 1 is uniformly distributed (equal probability) with a mean of 0.5 and a standard deviation of 0.289. Next, two different correlation matrices are written to *filename.lpr*. The first matrix shown is the correlation coefficient calculated from the random variable sample data. While, this matrix is not supposed to be equal to any correlation entered by a NESSUS user, it is written to this file only for the sake of completion. The correlation value, r_{XY}, between two variables, X and Y, can be calculated from data using Equation 19.

$$r_{XY} = \sum_{i=1}^{n} \frac{(X_i - \bar{X})(Y_i - \bar{Y})}{n s_X s_Y} \tag{19}$$

In Equation 19, the mean and standard deviation of X is $\bar{X}$ and s_X, respectively. The same notation is used for the second random variable, Y. All of the cumulative probability samples are shown next in *filename.lpr*. The second matrix to be shown is a ranked correlation matrix, also known as a Spearman rank correlation matrix. The Spearman correlation coefficient is calculated using Equation 20.

$$rs_{XY} = 1 - \frac{6}{n(n^2 - 1)}(d_1^2 + d_2^2 + \ldots + d_n^2) \tag{20}$$

The value d_j is the difference in ranks of the jth sample of X and Y. That is $d_j = rank(X_j) - rank(Y_j)$, for j=1,2,…n. The $rank(X_j)$ would be equal to 1 if X_j is the smallest in the set of all X's. It would be equal to 2 is only one value in the set of all X's is smaller than X_j. The logic continues until the $rank(X_j)$ would be equal to n if X_j is the largest value in the set of all X's. The same nomenclature and logic is true for Y. Finally, all of the cumulative probability samples are shown in the *filename.lpr* file.

The next file written by NESSUS that is new to the LHS method is the *filename.lxr* file. It has the same format as the *filename.lpr* file except the cumulative probability values of each random variable have been transformed to the x-space of that random variable by inverting its cumulative density function. The pairing of the random variables to form coordinates in multidimensional space is random; thus, the file contains latin hypercube samples in x-space that are randomly paired. A portion of the *filename.lxr* file is shown in Figure 38.

```
# Latin Hypercube Sampling Matrix File
# JOBID: *TITLE
# For each row(1:# Samples =      100) : Input_Vector(1:#RVs = 4) GNFS(1:#GFNS= 3)
# These are RANDOM SAMPLES with SPURIOUS CORRELATION between variables.
# LHS_X_SAMPLES :: LHS_PROB_SAMPLES(0,1) then INVERT RESPECTIVE PDF


MEAN of SAMPLE (by columns = random variable)
  0.5998305321E+002    0.9957446613E-002    0.1199856350E-009    0.9996446536E+002

STANDARD DEVIATION of SAMPLE (by columns = random variable)
  0.6048782811E+001    0.4808179970E-002    0.1202384214E-010    0.9953423045E+001

CORRELATION COEFFICIENT MATRIX (Linear)
  0.9900000000E+000
 -0.1973590140E+000    0.9900000000E+000
 -0.1753209067E+000    0.1927820834E-001    0.9900000000E+000
  0.2053832378E+000    0.4079494495E-001   -0.2126879984E+000    0.9900000000E+000

SPEARMAN RANK CORRELATION COEFFICIENT MATRIX
  0.1000000000E+001
 -0.2038163816E+000    0.1000000000E+001
 -0.1201560156E+000    0.5376537654E-001    0.1000000000E+001
  0.1162436244E+000    0.5111311131E-001   -0.1661806181E+000    0.1000000000E+001

***** SAMPLES *****
  0.5881350875E+002    0.1484333982E-001    0.1251766346E-009    0.9867971015E+002
  0.5178715646E+002    0.6004077650E-002    0.1280948548E-009    0.1033043708E+003
  0.6492549598E+002    0.5443585435E-002    0.1155917271E-009    0.8792079324E+002
** The rest of the samples are not shown.
```

Figure 38 LHS *filename.lxr* file that contains x-space LHS random samples

As seen in Figure 38, a header and short problem description is located at the top of *filename.lxr*. Rows showing the mean and standard deviations of the sample set of each random variable are then shown. After this, the lower halves of a correlation matrix and rank correlation matrix are shown. The last section of this file shows the x-space samples of the underlying random variables, which are randomly paired up to form coordinates in a multidimensional space. It is these samples that need to be arranged with respect to each other in order to be correlated as desired by the NESSUS user. In the NESSUS LHS thread, that was the next step to be performed and the results were an arranged set of the values in the *filename.lxr* file.

The file that contains the latin hypercube x-space samples that are arranged to exhibit correlation between pairs of the random variables is the *filename.lxc* file. This file is shown in Figure 39. This file contains a brief header and problem description. After the header, the mean and standard deviation of each random variable set is printed. The lower half of the correlation

matrix and rank correlation matrix are then shown. The correlation values that are entered by a user are shown above the Spearman rank correlation matrix. These values should be compared with the Spearman rank correlations because the entered values were assumed to be the desired rank correlation among the variables. The next and final part of the *filename.lxc* file shows all of the random variable samples that have already been arranged exhibit a correlation closer to the desired correlation. These values form coordinates in a multidimensional space in which the response under consideration exists; therefore, they are used to calculate response values. The response values calculated are shown in the final column of the samples section of the *filename.lxc* file. So, in the samples section, each row first shows the coordinates of a multidimensional space, and the response evaluated at that coordinate of the space.

```
# Latin Hypercube Sampling Matrix File
# JOBID: *TITLE
# For each row(1:# Samples =      100) : Input_Vector(1:#RVs = 4) GNFS(1:#GFNS= 3)
# These are RANDOM SAMPLES with ADJUSTED CORRELATION between variables.
# LHS_X_SAMPLES :: DECOMPOSE random LHS_X_SAMPLES to yield samples with desired correlation


MEAN of SAMPLE (by columns = random variable)
  0.5998305321E+002    0.9957446613E-002    0.1199856350E-009    0.9996446536E+002

STANDARD DEVIATION of SAMPLE (by columns = random variable)
  0.6048782811E+001    0.4808179970E-002    0.1202384214E-010    0.9953423045E+001

CORRELATION COEFFICIENT MATRIX (Linear)
  0.9900000000E+000
 -0.2460925957E+000    0.9900000000E+000
  0.1271867747E+000   -0.5553565269E-001    0.9900000000E+000
 -0.1020203598E+000   -0.3154875326E-001   -0.6401985701E-001    0.9900000000E+000

SPEARMAN RANK CORRELATION COEFFICIENT MATRIX \ DESIRED
  0.1000000000E+001    0.0000000000E+000    0.0000000000E+000    0.0000000000E+000
 -0.2839843984E+000    0.1000000000E+001    0.0000000000E+000    0.0000000000E+000
  0.1240204020E+000   -0.3576357636E-001    0.1000000000E+001    0.0000000000E+000
 -0.1123912391E+000   -0.6192619262E-001   -0.5056105611E-001    0.1000000000E+001

***** SAMPLES *****
  0.6184054255E+002    0.4622058805E-002    0.1350493465E-009    0.9007977956E+002    0.3047257227E+005
  0.6474329889E+002    0.5551887945E-002    0.1370210211E-009    0.8034276478E+002    0.3925144922E+005
  0.7370048225E+002    0.1075698962E-001    0.1115591843E-009    0.9961692948E+002    0.1605809668E+005
```

Figure 39 LHS *filename.lxc* file that contains x-space LHS correlated samples

All of the x-space samples given in the *filename.lxc* file can be transformed to their respective cumulative probability value based on their distribution. These cumulative probability

values for the correlated x-space sample set are given in the *filename.lpc* file. It gets its name from the fact that it contains latin hypercube cumulative probability values whose underlying random variables have already been correlated with one another. The partial contents of the file are shown in Figure 40. There is a brief header and problem description followed by the mean and standard deviation of the cumulative probability values of each random variable. The mean and standard deviation should be 0.5 and 0.289, respectively. The correlation coefficient and rank correlation is shown next. The spearman correlation matrix values should be exactly like the spearman correlation matrix shown in the *filename.lxc* file because the cumulative distribution of each random variable is monotonically increasing. The cumulative probability values corresponding to the x-space samples in the *filename.lxc* file are shown next and this is the last part of the *filename.lpc* file. There is one other output that the NESSUS LHS method will produce. It is output to the screen and it is merely a repeat of information shown in the other five output files.

```
# Latin Hypercube Sampling Matrix File
# JOBID: *TITLE
# For each row(1:# Samples =      100) : Input_Vector(1:#RVs =  4) GNFS(1:#GFNS=  3)
# These are RANDOM SAMPLES with ADJUSTED CORRELATION between variables.
# LHS_PROB_SAMPLES :: LHS_X_SAMPLE adjusted for correlation and calculate cumulative probability


MEAN of SAMPLE (by columns = random variable)
  0.5005149309E+000    0.5001038167E+000    0.4999803626E+000    0.4998677827E+000

STANDARD DEVIATION of SAMPLE (by columns = random variable)
  0.2897709245E+000    0.2905316077E+000    0.2897500814E+000    0.2902432202E+000

CORRELATION COEFFICIENT MATRIX (Linear)
  0.9900000000E+000
 -0.2818361634E+000    0.9900000000E+000
  0.1222783427E+000   -0.3623551157E-001    0.9900000000E+000
 -0.1108243235E+000   -0.6093321833E-001   -0.5046986477E-001    0.9900000000E+000

SPEARMAN RANK CORRELATION COEFFICIENT MATRIX
  0.1000000000E+001
 -0.2839843984E+000    0.1000000000E+001
  0.1240204020E+000   -0.3576357636E-001    0.1000000000E+001
 -0.1123912391E+000   -0.6192619262E-001   -0.5056105611E-001    0.1000000000E+001

***** SAMPLES *****
  0.6204858005E+000    0.8112500821E-001    0.8914555030E+000    0.1592674185E+000
  0.7853966958E+000    0.1563639538E+000    0.9161463367E+000    0.1600590842E-001
  0.9887970272E+000    0.6519773811E+000    0.2478385493E+000    0.5045476255E+000
```

Figure 40 LHS *filename.lpc* file contains the cumulative probability of the correlated LHS samples

3.5 TEST CASES

Density parameters can be used to obtain estimates of the reliability of a system. They must be accurately estimated using as little computational effort (computer time) as possible. Usually, they density parameters are estimated only once. An often avoided and important question is: Where will a single estimate of a response density parameter lie with respect to the exact value of the parameter? This can be answered for specific responses through studies like this one that attempts to capture the distribution of several estimators as a function of the number of samples, or response evaluations, and the method used to obtain the coordinate sets of the domain of the response – Monte Carlo and Latin Hypercube Sampling.

The Society of Automotive Engineers (SAE) has put forth a number of test cases that can be used to compare different probabilistic or statistical methods. This discussion is limited to 4 test cases with varying number of random variables, distributions, and nonlinearity. They are labeled test case 1, 4, 6, and 8 only to be consistent with the file names originally given to each case. For each test case and each method, 900 different files were needed and over 144,000,000 response evaluations were obtained, so organization was top priority. For these test cases the mean, standard deviation, and 99th percentile of the response is estimated using Monte Carlo and Latin Hypercube sampling schemes. The distribution for each estimator and each method was attempted to be completely captured by repeatedly calculating the respective estimate 100 times. This, in turn, was performed when the following number of response evaluations were used to calculate the mean, standard deviation, or 99th percentile: 100, 300, 1,000, 3,000, 10,000, 30,000, 100,000, 300,000, and 1,000,000. The exact value of the appropriate parameter was assumed to be the average value of 100 estimations of the respective parameter when 1 million MC samples were used to calculate the response values used to compute the each parameter estimate.

Test Case 1: Stage II Crack Propagation – Paris Law

Response Function and Design Variables

Fracture by fatigue is a common failure mode in metallic structures. A structure will fatigue when it is subjected to cyclic stresses below the material's yield or ultimate tensile stress. Fatigue is a time-delayed material fracture due to time varying stresses. It will only occur in the regions of the material for which at least one of the principal stresses reach a state of tension during the varying system loading. Fatigue is the result of stochastic loading, i.e. load variations, on a structure and it is because of this that fatigue fracture can occur in systems that are not ordinarily considered to be cyclically loaded. This type of failure can be seen in metals and their alloys, polymers, and ceramics. Observations of metals and polymers has shown that there is a correlation between the number of cycles that cause failure and the applied cyclic loading, initial crack sizes, and material properties, among other factors. Between the two material classes, however, the mechanism of deformation is different due to their microstructural differences. Ceramics to fracture by fatigue, but, depending on the environment, once a crack is nucleated, their fatigue life is relatively short when compared to the other two classed of materials. Fatigue fracture occurs in three stages – crack nucleation, crack propagation, and either overload or final fast fracture.

Crack nucleation is the first (I) stage of fatigue fracture and occurs due to plastic flow in flawed areas. These are areas of high stress concentrations and local plastic flow can occur even under global elastic loading conditions. At some point a crack is considered nucleated and initially propagates along a crack plane whose normal is not parallel to the loading axis. This stage is dictated by plasticity not fracture mechanics considerations. The crack continues to grow and eventually reaches a critical size – a stage II crack forms and the next stage of fatigue

fracture begins.

The second stage (II) of fatigue fracture is generally dictated by slow crack growth rates. The relation between crack growth rates and the stress and its range can be predicted with less error and more confidence than the first stage of crack growth. Also, the direction of the crack growth is normal to the principal tensile axis during this stage. As the crack propagates, the area that is not cracked decreases and eventually becomes unable to sustain the same load types. This is the beginning of the last stage of fatigue fracture.

During the final stage (III) of fatigue fracture the material can be considered to overload since the load is now distributed over a smaller local effective area, or quickly fractured because the material's fracture toughness reduces along with the effective area depending on if you choose to approach the problem from an elasticity or fracture mechanics point of view and which type of failure actually occurs.

The first case response measures the number of load cycles to failure for the *second stage* of crack growth. The general model is a power law commonly known in the fracture mechanics discipline as Paris' Law, given by Equation 21.

$$\frac{da}{dN_{II}} = c(\Delta K)^m \tag{21}$$

The number of load cycles, N_{II}, is over the second stage only; hence the subscript. The Paris Law relates the change in crack size, a, with respect to the change in load cycles during the second stage, N_{II}, to a constant, c, that depends on the material and the load stress ratio ($\sigma_{min}/\sigma_{max}$), and a empirical constant, m, which is usually between 2 and 7 [Courtney 2000]. This crack growth rate is also a function of a changing material-load state, $\Delta K \propto \Delta\sigma\sqrt{a}$, which

is a function proportional to the change in stress during a load cycle and the square root of the current crack size. This material-load function is a measure of the fracture crack-stress state of the material. If it is initially less than a threshold value for the system, ΔK_{th}, at the beginning of the stage I cycles, fatigue fracture will not occur since the crack will not propagate. However, placing a material in this state usually an over design of the system. Therefore, it is usually the case where a crack will form in stage I and grow to be a stage II crack where the Paris Law applies. The crack will continue to propagate and so ΔK will also increase and eventually approach a materials critical fracture toughness, K_{Ic}. When this happens, the crack growth rate will increase and stage III of fatigue fracture will begin. This stage soon ends because fast failure occurs by tensile failure, fatigue crack-advancement, or, for the most part, both modes of failure.

In order for Equation 21 to be an accurate fit to what is observed experimentally, it is *assumed* that there is an initial, or existing crack that is larger than some important microstructural scale, e.g. a grain size. It is also *assumed* that the critical crack length at which stage III fast failure will occur is known. The numerical analysis that leads to the determination of the number of cycles spent in stage II crack growth begins with the identity $da/dN = da/dN$, which is integrated to determine the number of stage II cycles given by

$$N_{II} = \int_{a_i}^{a_f} \frac{da}{da/dN_{II}} \tag{22}$$

The denominator *for this stage* is given by the Paris Law, $da/dN_{II} = c(\Delta K)^m$, where $\Delta K = \mathcal{A}\Delta\sigma\sqrt{a}$. The parameter $\mathcal{A}$ is usually crack size, geometry, and load-type dependent, but an average value of $\mathcal{A}$, can be used for the purpose of simplifying and completing the analysis.

Its value is typically close to unity and its value comes from experimental data on test specimens. It is important to use an expression for ΔK that comes from experimental tests that match the situation for the system under consideration. It is assumed that the system under study is similar to the one shown in Figure 41.

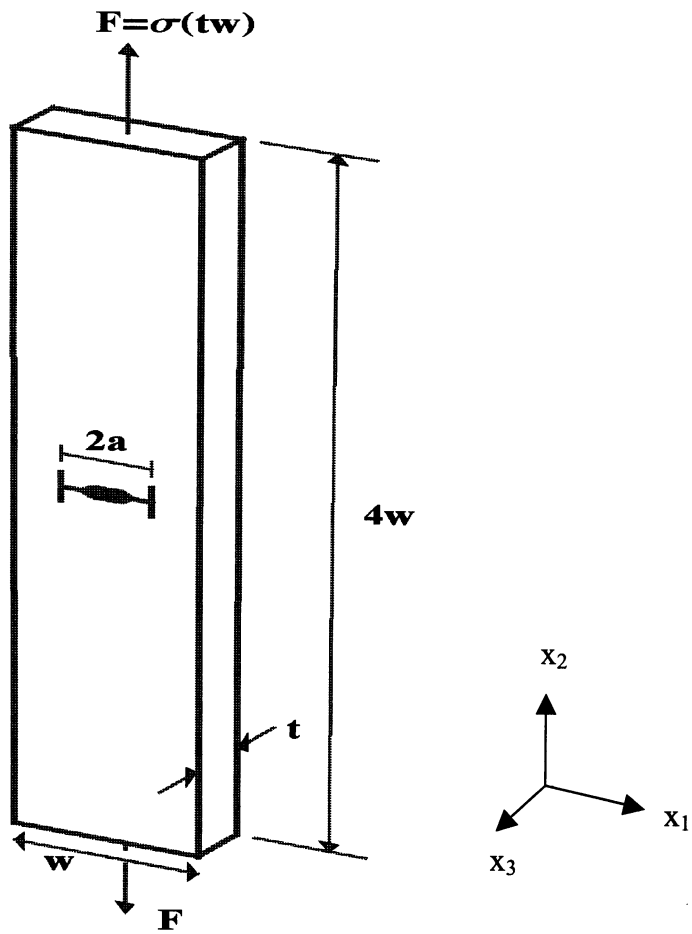


Figure 41 Center notched specimen placed in tension (not to scale)

Figure 41 shows the geometry, crack size, crack type, and loading for a specimen that is similar to situation of a system that might be studied and has a response similar to the one that is in the process of being derived. The loading and the length of the specimen are both in the x_2

direction, the specimen width (w) is along the x_1 direction, and specimen's thickness is in the x_3 direction.

The load direction and crack orientation deems the specimen to be in Mode I facture. The tensile load is normal to the crack surfaces and the crack grows in a direction normal to the applied load. The crack is considered to be a notch with whose tip is sharper than the notch. This type of crack tip can be obtained by subjecting the specimen to prestresses or thermal shocks.

When crack tip region undergoes only elastic deformation, i.e., there is no plastic deformation, the normal stress in the load direction, σ_{x_2}, is high near the crack tip and gradually approaches the nominal value of $\sigma_{nom} = F(tw)$ in moving in the x_1 direction. The normal stress in the x_1 direction, σ_{x_1}, is zero at the crack surface because free surfaces cannot support normal stresses. It rises to a peak value due to a constraint effect between the crack surface and the material in the x_1 direction away from the crack. The tensile stress in the x_3 direction, σ_{x_3}, is zero at the surface, and if the material is thin in this direction (small t), then it might be assumed that the stress is zero throughout the thickness. Plane stress conditions would be the prevailing stress state in the region of the crack tip. If the thickness of the specimen is increased, the tensile stress σ_{x_3} is still zero at the surface, but will increase progressively into the thickness due to deformation constraints with the rest of the material. The σ_{x_3} stress will reach a maximum value of $\nu\left(\sigma_{x_1} + \sigma_{x_2}\right)$ at a critical point, x_c, away from the surface and in the x_3 direction. The state of stress transitions from a condition of plane stress to plane strain from the surface up to x_c. Thus, in a thick specimen, a triaxial stress state exists away from the surface and into the thickness.

In many applications for materials that are considered to be more ductile than brittle,

fracture occurs because the stress at the crack tip is a region of high stresses. The high stresses in the region of the crack cause plastic deformation in that area. There is therefore a plastic zone near the crack tip that is bounded by the remaining region of elastically deforming material. Also, plastic deformation blunts the crack tip, and because this type of deformation is irreversible work expended - there is no recovered energy, it will slow the crack propagation process. The larger the region of plastic deformation, the slower the crack will grow, and vice-versa. The plane strain or triaxial stress state near a crack tip undergoing local plastic deformation is extremely complex; however, it can be said that this stress state has a smaller plastic zone size compared to plane stress and this state is more evident near the plastic-elastic boundary. Both of these factors reduce the toughness of the material. A minimum value of the fracture toughness will be reached as conditions of plane strain prevail over plane stress. This value is termed the plain strain fracture toughness, K_{Ic}. The plane strain fracture toughness is usually used in design because it allows a conservative approach to be taken. Therefore, we *assume* that the system under study resembles that shown in Figure 41 *and* that a plane strain, or triaxial, stress state is dominant throughout the system [Courtney 2000, Ch.9].

The previously discussed $\measuredangle$ parameter in $\Delta K = \measuredangle \Delta\sigma \sqrt{a}$ can be found in literature - see Courtney 2000, page 429. This book shows that $\measuredangle = \sqrt{\pi}\sqrt{\frac{w}{\pi a}\tan\left(\frac{\pi a}{w}\right)}$. These experimental results, or data fits, used in obtaining fracture toughness values have low error for test specimens whose lengths are 4 times the width and whose total crack lengths are one third of the width, $2a = w/3$. Also, plane strain conditions are met when the thickness, t is between 10% and 20% of the width. For the system whose response is being studied in this section, the total crack length is roughly half of the width, $2a = w/2.0445$. This might not be within the valid region for

which the parameter $\measuredangle$ has the closed form solution mentioned and given in Courtney 2000. However, take into consideration that if $2a = w/2.0445$, then the Stage II cycles predicted from any equation that uses the $\measuredangle = \sqrt{\pi}\sqrt{\frac{w}{\pi a}\tan\left(\frac{\pi a}{w}\right)}$ relationship can either be lower than what would be observed experimentally (conservative analysis), or higher than an experimental value obtained from actual tests (non-conservative analysis). An experienced analyst would not be left in the dark at this point because the total crack length, 2a, is over 45% larger when $2a = w/2.0445$ than when $2a = w/3$. Since the term ΔK scales with the square root of the crack size in its fundamental form $\Delta K = \measuredangle \Delta\sigma\sqrt{a}$ as well as in the parameter $\measuredangle$, where increasing the crack length increases ΔK, it can be said that using an increased value for the total crack length is a conservative approach to the analysis. Increasing ΔK will increase the rate of crack growth given by the Paris Law, $da/dN_{II} = c(\Delta K)^m$.

Furthermore, if the actual system under consideration is only assumed to be in states of plane strain, when, in actuality, a significant fraction of the system might be in a state of plane stress, this would be a conservative assumption on top of the large crack size argument just mentioned. This would be assuming that the fracture toughness would be at its minimum value – the plane strain fracture toughness, K_{Ic}. In the numerical analysis, we therefore use a higher than actual ΔK value that increases as the crack grows and approaches a lower than actual value of the material fracture toughness K_{Ic}. When this happens the unstable and quick Stage III, fatigue fracture begins. The numerical analysis will therefore produce lower lifetimes, or load cycles to failure, than if the actual system under consideration were fatigued to failure.

The parameter $\measuredangle$ $(= 1.1215\sqrt{\pi})$ is then used to continue our conservative analysis. If we substitute the now known expression for ΔK into Equations 21 and 22, we arrive at the

following expression for the number of Stage II cycles that occurred resulting in the crack growing from its initial size, a_i, to its final size, a_f. Note that by using a constant value of $\measuredangle$ we assume that it will not vary considerably over the crack sizes encountered in stage II crack growth.

$$N_{II} = \int_{a_i}^{a_f} \frac{da}{c\left(1.1215\Delta\sigma\sqrt{\pi\, a}\right)^m} \tag{23}$$

Once Equation 23 is integrated and N_{II} is solved for, we arrive at the following response function under study

$$Z = N_f = N_{II} = \frac{\left(a_f^{\,1-m/2} - a_i^{\,1-m/2}\right)}{c\left(1.1215\Delta\sigma\right)^m \pi^{\,m/2}\left(1-m/2\right)} \tag{24}$$

The term Z is a generic response variable commonly used in reliability analyses and the number of cycles to *failure*, N_f, is set equal to the load cycles the withstood during Stage II crack growth. Usually, the number of cycles to failure would be the sum of all three stages of fatigue fracture – that is, $N_f = N_I + N_{II} + N_{III}$. Therefore, using the equality $N_f = N_{II}$ to determine the number of cycles to failure implies that we assume that the number of cycles encountered during Stages I and III of fatigue fracture are negligible when compared to the amount of cycles spent in Stage II crack growth.

The final crack size is determined by setting $\Delta K = K_{Ic}$ and solving for the crack size, a, which is then considered to be the final crack size, a_f. This is shown in Equation 25.

$$a_f = \frac{1}{\pi}\left(\frac{K_{IC}}{1.1215\Delta\sigma}\right)^2 \tag{25}$$

The response given by Equations 24 and 5 is a function of several design variables. Table 5 describes each variable and lists their statistics that are assumed true and used in estimating the statistics of the response $Z = N_f$.

Table 5 Design variables for test case 1

Variable	*Description*	*Value or Distribution*
K_{Ic}	Fracture toughness $(ksi\sqrt{in})$	N (60,6)
a_i	Initial crack size (in)	LN (0.01,0.005)
c	Paris constant (-)	LN (1.2E-10, 1.2E-11)
$\Delta\sigma$	Cyclic load (ksi)	LN (100,10)
m	Paris exponent (-)	3

The fracture toughness, ΔK, is normally distributed with a mean of 60 and standard deviation of 6, or 10% of the mean. The initial crack size, a_i, which is the lower limit in the integration of the Paris Law fit, is log-normally distributed with a mean and standard deviation of 0.01 and 0.005 – a COV of 50%. The mean and standard deviations for all lognormal variables in this paper are that of the lognormal distribution *not* of the underlying normal distribution. The Paris 'c' constant, is also log-normally distributed whose mean is 1.2E-10 and a COV of 10%.

This value would be obtained from da/dN_{II} vs. ΔK data, as would the Paris exponent, m, which has a deterministic value of 3 for this analysis. The cyclic load, $\Delta\sigma$, is log-normally distributed with a mean of 100 and 10% COV. These are the variables used to calculated values for the response of Equations 24 and 25.

Convergence of Sampling Methods

Before we discuss the convergence of the statistics of the response given by Equations 24 and 25 using MC and LHS, let us first introduce a common graphical representation of statistics of data known as a box and whiskers plot. A box and whiskers plot is shown in Figure 42.

The box and whiskers plot shown in Figure 42 shows the location of all of the data points as small stars (⋆), the 25th, 50th, and 75th percentiles, and consequently shows a region where 50% of the data lies. The length of the box, H, is known as the step and is used to determine other locations of interest. Another region known as the inner fence has a lower value at 1.5H less than the 25th percentile and an upper value at 1.5H greater than the 75th percentile. Any data points that lie outside the inner fence are known as outliers. There is an outer fence that is not shown in Figure 42 that has its limits at 3H away from the same percentiles as the inner fence. This plot shows the location of the mean of the data with a large filled-in star (★).

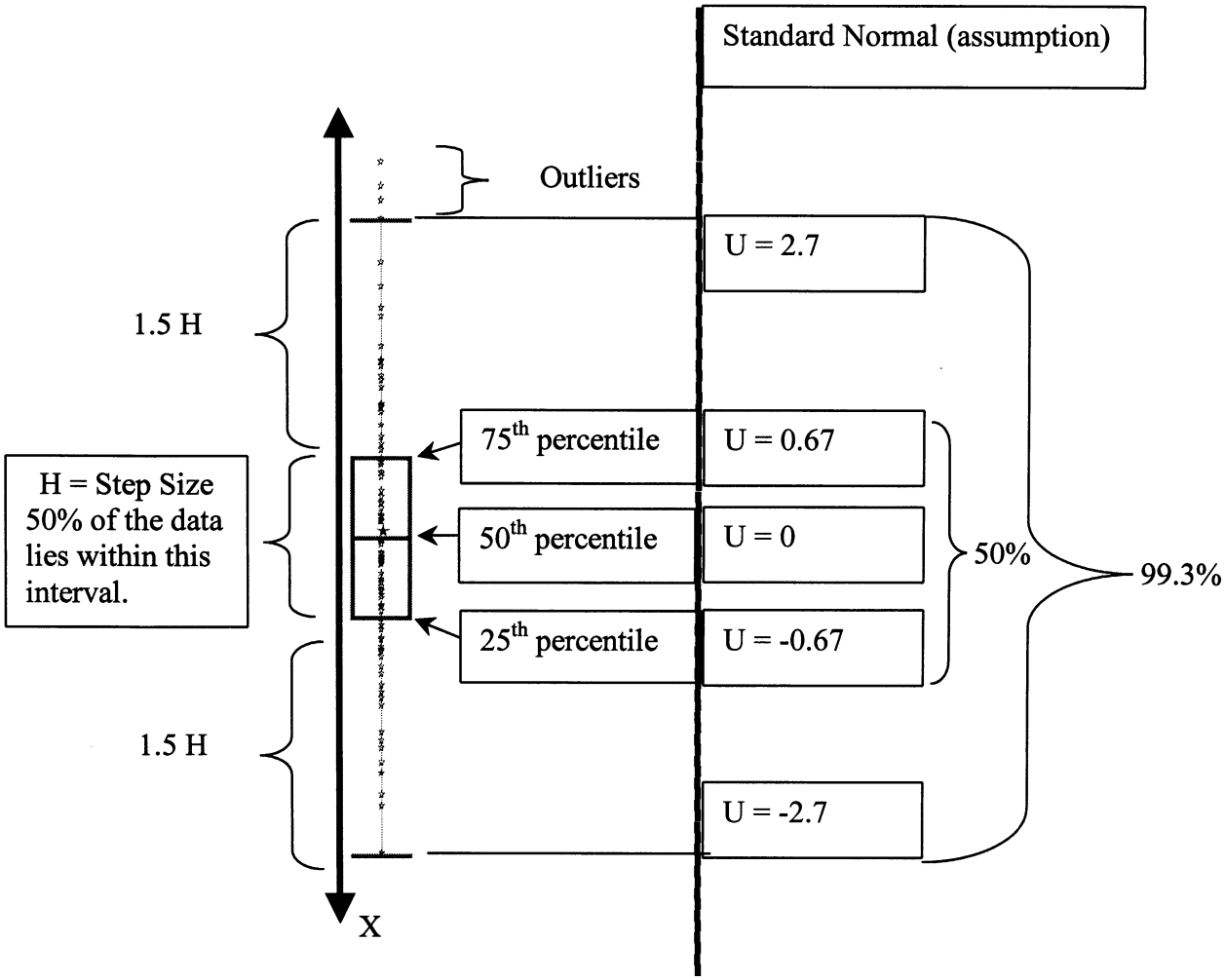


Figure 42 A box and whiskers plot.
The 25th, 50th, and 75th percentiles are indicated by the lower, middle, and upper lines, respectively. These lines define a box that is the middle 50% of the data. Here, the inner fence (considered as whiskers) is shown. Anything outside of this fence is considered an outliers.

The mean of the test case 1 response was estimated many times using both Monte Carlo and Latin Hypercube Sampling. The distribution of the mean estimator for each method was captured by repeatedly calculating the response mean 100 times using a completely different set of responses for a varying number of samples that ranged from 100 to 1 million. The distributions for the different mean estimators are shown in Figure 43. A box and whiskers plot is shown for each method and every sample level that the repeated estimated were performed. Fifty percent of the data for each distribution lies within the inner box of the box and whiskers plot. The LHS distributions are shown with an offset in the positive number of samples direction only for clarity. The mean of each distribution is represented with a filled in star (★). The horizontal line is the average value of 100 estimations of the mean when 1 million MC samples were used to calculate the response values used to compute each mean estimate.

The distributions shown in Figure 43 are apparently normally distributed. Even if the distributions are shown to have slight skew, it must be reminded that these distributions are not the exact distributions of the respective mean estimator, they are only estimates of the distributions obtained by calculating the mean of a number of response evaluations 100 different times. Observing slight skew in an estimated distribution of means can be neglected and is not worthy of mentioning because it could be reduced as the distribution of means is more accurately captured. It is therefore safe to say that both methods have an associated mean distribution that is apparently normally distributed for all the number of samples used to calculate each mean and this agrees with Wackerly et. al. (1996) which states that the distribution of the mean estimator is normal for sample sizes greater than or equal to 30.

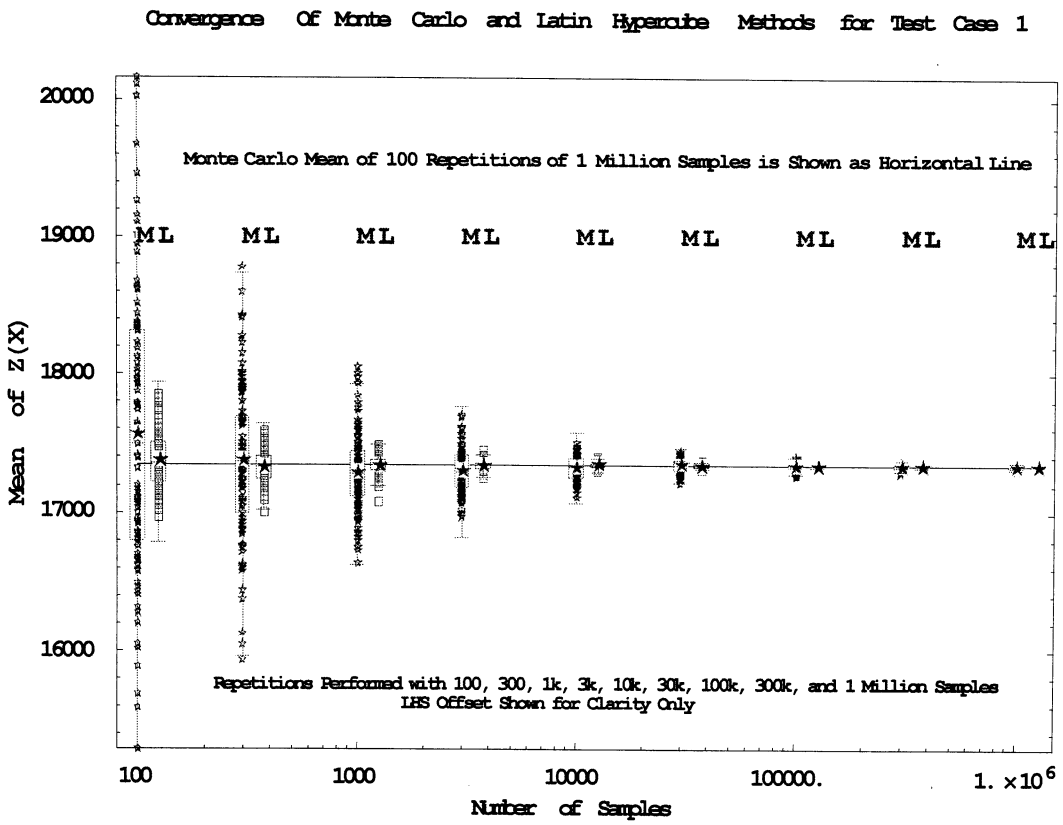


Figure 43 Distributions of means for test case 1 response using MC and LHS

Another measure of the goodness of an estimator is the standard error, or standard deviation, of the distribution under consideration. Using either MC or LHS samples, the standard error of the mean estimator distribution decreases as the number of samples, or response evaluations, used to calculate each mean of the responses increases, as shown in Figure 43. This statement implies that calculating the mean of the response a repeated number of times will result in values that are closer to each other when the amount of response values used to calculate the mean is large. It is important to know this because it implies that the as the effort, measured in computational time or number of samples used to obtain a single estimate of the mean increased then all of the possible mean values that can be calculated will be more centered around each other, or the mean of the respective distribution under consideration. Also, since the

mean of the respective distributions are approximately equal to the exact value of the mean of the response, then more confidence can be place in a single estimate lying within a certain range of the exact value as the number of samples used to calculate this mean is increased. The standard error of the distribution of means using LHS samples to evaluate the response and calculate each mean is smaller than the same distribution captured using MC samples when the two distributions are compared at the same sample level. This is true for all of the sample levels shown in Figure 43.

Since an analyst typically performs only one set of calculations leading up to a single estimate of a target value, it would be important for he or she to have confidence that the estimate, even if it is not close to the true value, be close enough to be within an acceptable error limit. A statement like this can be made *from the information given in Figure 43* using the middle 50% of the box plots for each distribution. These are the types of statements that allow one to compare the efficiency of different methods in obtaining estimates of density parameters, like the mean of a response. When 1,000 Monte Carlo samples are used form the coordinates utilized to calculate 1,000 response values after which, a mean of those responses can be calculated, it is 50% likely that the single mean estimate calculated will be within 0.50% of the target parameter or the true mean – roughly about 95 cycles on either side of the true mean. It must be noted that the box plot for the Monte Carlo method at the 1000 sample level is not symmetrical about the true mean, so the actual statement should be that it is 50% likely that a single estimate will be between 230 cycles below the true mean and 95 cycles above the true mean. Unfortunately, for the sake of comparison of methods, especially over multiple test cases, this is too much information and only confuses the main emphasis of each comparison. Also, the 25th and 75th percentile values, which form the limit for the middle 50% of the data, will almost

never be the same distance from the target parameter. Therefore, it will be easier for the sake of comparing the MC and LHS sampling methods over the test cases in this work if the range of the middle 50% box plot will be taken to be ± the smaller of the difference between the 25th percentile and the true value and the difference between the 75th percentile and the true value. Such a simplification falsely gives the method under consideration better confidence interval properties; however, this will be done for both methods and all test cases, and surely any unjust statement will be apparent from Figures similar to Figure 43. Keep in mind that, yes, the goal here is to quantify confidence in single mean estimates using MC and LHS; in spite of this, we gladly sacrifice the accuracy of our statements for a more organized effort at making general statements about the two methods. Using 1000 Latin Hypercube samples to form the coordinates necessary to calculate 1000 response values and after that the mean of those responses, it is 50% likely that the single mean estimate calculated will be within 0.20% of the target parameter or the true mean which is about 33 cycles on either side of the true mean. Therefore, for this test case, LHS gives an analyst the same confidence that a single mean estimate will have a lower error than MC. Also, it is important to note that there are three variables to consider when comparing methods by making confidence statements: (1) *effort*, number of samples, or response evaluations, used to make a future estimate, or computational time (2) *confidence*, measure of possibility that the future estimate will lie within a certain error or interval from the true value, and (3) *error* or interval that a certain confidence is placed in. In order to be able to compare methods one must set two of the variables equal to each other across the methods and compare the left over variable. For what was just mentioned, the effort and confidence level were set to 1000 samples and 50%, respectively for both methods. It was then found that LHS had the lower error of 0.20% from the true mean than the MC error of 0.50%. The same effort and confidence

was used and it is found that LHS had a lower error than MC.

Another way to compare methods would be to set the error and effort equal across the two methods and compare the confidence that a future estimate would lie within that error using a certain amount of effort. This will not be discussed in this paper. On the other hand, the third and final way to compare methods like MC and LHS is to set the confidence and error equal for both methods and compare the effort required to obtain the like results. From another standpoint, this is comparing the effort required to obtain the same distribution of the respective density parameter estimator – in this case the mean estimator.

The coefficient of variation (COV) of the distribution of the mean estimator will converge to various values as the effort, or number of samples used to calculate each mean value, is increased for both of the methods used to obtain response values. The COV [$\equiv \sigma/\mu$] is the ratio of a distributions standard deviation to its mean. Since all of the test cases to be discussed in this work have essentially unbiased mean estimators for all of the sample levels considered and both methods, the COV is a measure of the variation of repeated mean estimates about the true mean for a specific number of samples. Figure 44 shows the COV of the mean estimator distribution for MC and LHS as it varies for all of the sample levels that the repeated mean estimations were performed for the purpose of capturing the distribution of the mean estimator. This figure shows the actual calculated COVs as points and a Log-Log linear curve fit line that approximates the COV of the distribution of the estimator continuously for all number of samples between 100 and 1 million. The curve fit derivation and equations are shown in Appendix IV-D. The horizontal line at the COV value of 0.005 (0.5%) is shown to emphasize the difference in effort required to obtain the same variation about the mean of the mean estimator distribution, also the true mean, for both methods.

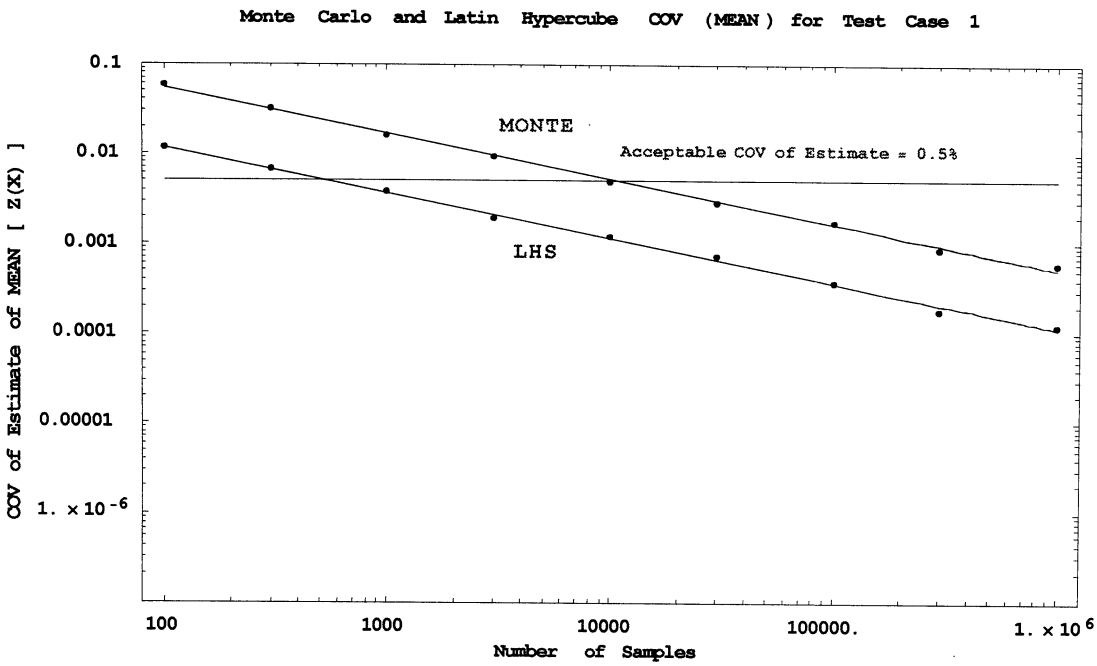


Figure 44 COV of mean distributions for test case 1 response using MC and LHS

From Figure 44 one can see that the rate of convergence to various COV levels is the same for MC and LHS. This rate of convergence is the slope in Log-Log space, and is the 'm' exponent in the model of the curve fit, $COV = cn^m$. Where 'c' and 'm' are the two constants that define the curve fit, n is the number of samples, and COV is the coefficient of variation. These two constants are not the same constants as the ones in Table 5 that are the constants of the Paris Law of Equation 21. These two curve fits are of the same form, but fit different data and have different constants associated with each fit. Both methods show a rate of convergence on the order of –1/2. Furthermore, the mean estimator would have been observed to reach a COV level of 0.5% at n=10,000 using the MC method while the LHS method needed about 500 samples to converge to the same level. Both distributions are unbiased at those respective sample levels; therefore, they have the same mean that is approximately equal to the true mean. Since they have the same COV, they are the same distribution, centered about the true mean of

the response for test case 1. A COV level of 0.5% implies that the standard error (standard deviation) of the distribution of the mean estimator is 0.5% of the mean, and $3\sigma_{\hat{\theta}}$ is 1.5% of the mean. Therefore, if we assume that the mean estimator distribution for MC and LHS is normally distributed and its mean is the true mean at n=10,000 and n=500, respectively, then we are in the game for making the desirable confidence statement previously discussed because for a normal distribution, 99.73% of the data lies within $3\sigma_{\hat{\theta}}$ of the mean. Both assumptions are approximately true for both MC and LHS at n=10,000 and 500, respectively. From the information given in Figure 44, it can be stated with 99.7% confidence that a single mean estimate for the response of test case 1 will be within ±1.5% of the true mean using MC-10,000. In comparison, there is a 99.7% chance that the same estimate will be within ±1.5% of the true mean using LHS-500. The estimation error of 1.5% is 260 cycles from the mean. The type of confidence statement just made is of the type – equal confidence and error, different effort. LHS requires much less computational effort than MC when confidently estimating the mean of the test case 1 response.

The standard deviation of the test case 1 response, another density parameter, was also estimated many times using MC and LHS. Like the mean estimator, the standard deviation estimator, being a function of random variables, is also random, and has a certain distribution the will vary with the number of response evaluations used to estimate the standard deviation, n, and the method used to obtain the coordinate sets, MC or LHS. The distribution of the standard deviation estimator was approximated by repeatedly calculating the standard deviation 100 different times for each method and a varying number of response evaluations. The resulting distributions are shown in Figure 45. A box and whiskers plot is shown for each distribution, the LHS is offset only for clarity, and the horizontal line, treated to be the exact standard deviation

of the test case 1 response, is the mean of 100 estimations of the standard deviation when 1 million MC samples were used to calculate each estimation.

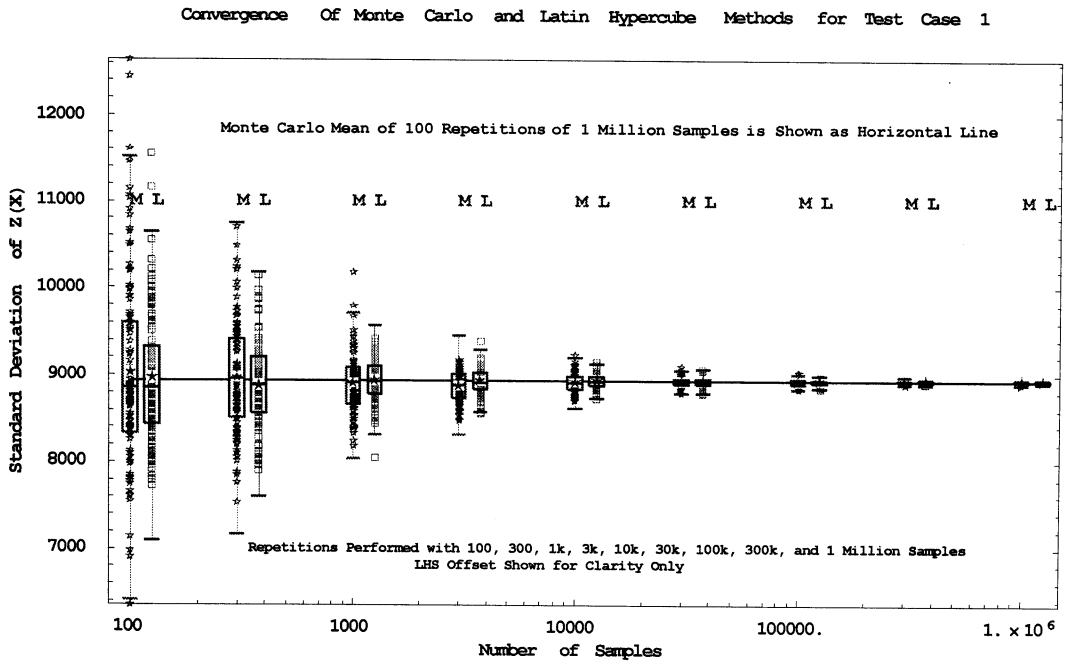


Figure 45 Distributions of standard deviations for test case 1 response using MC and LHS

Both methods have associated standard deviation distributions that appear normally distributed for all the number of samples that defines the standard deviation estimator. True, some of these distributions might show slight skew; however, recall that these distributions are not the exact distributions of the appropriate standard deviation estimator, they are only estimates of the standard deviation distribution. In any case, observing slight skew in an estimated distribution of standard deviations can be neglected and is not worthy of mentioning because the skew could be reduced as the distribution is more accurately captured, or in a different set of random circumstances. It is therefore safe to say that both methods have an associated standard deviation distribution that is apparently normally distributed for all the number of samples used to calculate each standard deviation and this agrees with the Wackerly et al. (1996) statement

that the probability distribution of the standard deviation estimator is positively skewed for small sample sizes, but approximately normal for large sizes (n>25). Also, the distribution of the standard deviation estimator using both MC and LHS is unbiased and, therefore, centered about the exact standard deviation value for all the number of samples, or effort levels, shown. This is in agreement with Wackerly et al (1996).

The standard error of the standard deviation estimator distributions shown in Figure 45 decrease as the number of samples, or response evaluations, used to calculate each standard deviation of the responses increases. The LHS standard deviation distribution has a lower standard error than the MC standard deviation distribution for all number of samples shown in Figure 45.

Confidence statements can be made from the information given in Figure 45 using the middle 50% of the box plots for each distribution. When 300 MC samples are used form the coordinates utilized to calculate 300 response values and after that the standard deviation of those responses is computed, it is 50% likely that the single standard deviation estimate calculated will be within 4.85% of the target parameter or the true standard deviation. This is about 435 cycles on either side of the true standard deviation. Using 300 LHS samples to form the coordinates necessary to calculate 300 response values and then a single standard deviation estimate, it is 50% likely that the single standard deviation estimate calculated will be within 3% of the target parameter or the true standard deviation. This is about 265 cycles on either side of the target. Therefore, for this test case, at the n=300 and 50% effort and confidence levels, respectively, it was found that LHS had the lower error of 3% from the true standard deviation than the MC error of 4.85%. The same effort and confidence were used and LHS had a lower error than MC.

The COV $[\equiv \sigma/\mu]$ of the standard deviation distribution will be different as the effort, or number of samples used to calculate each standard deviation value, is increased and for both of the methods used to obtain response values. The COV is a measure of the variation of repeated standard deviation estimates about the true standard deviation for a specific number of samples and method, or rather a specific standard deviation distribution, only because these distributions are essentially unbiased as discussed when Figure 45 was considered. The COV of the standard deviation estimator distribution for the Monte Carlo and Latin Hypercube methods as it varies for all of the sample levels is shown in Figure 46.

From Figure 46 one can see that MC and LHS have the same rate of convergence to various COV levels. It is on the order of $-1/2$. The MC standard deviation estimator is shown to have a COV level of 0.5% using n=50,000 samples for each standard deviation estimate, while the LHS method needed about 30,000 samples to converge to the same level. Both distributions are approximately unbiased and normal those respective sample levels; therefore, they have the same mean that is approximately equal to the true standard deviation of the test case 1 response. Therefore, they are the same distribution at the levels just mentioned. Therefore, it can be stated with 99.7% confidence that a single standard deviation estimate for the response of test case 1 will be within ±1.5% of the true standard deviation using MC-50,000. In comparison, there is a 99.7% chance that the same estimate will be within ±1.5% of the true standard deviation using LHS-30,000. The estimation error of 1.5% is 134 cycles from the true standard deviation. The LHS method will estimate the standard deviation of the test case 1 response with equal confidence and error, but with less effort, or numerical calculations than the MC method. These statements are based on the best-fit line for the data shown in Figure 46.

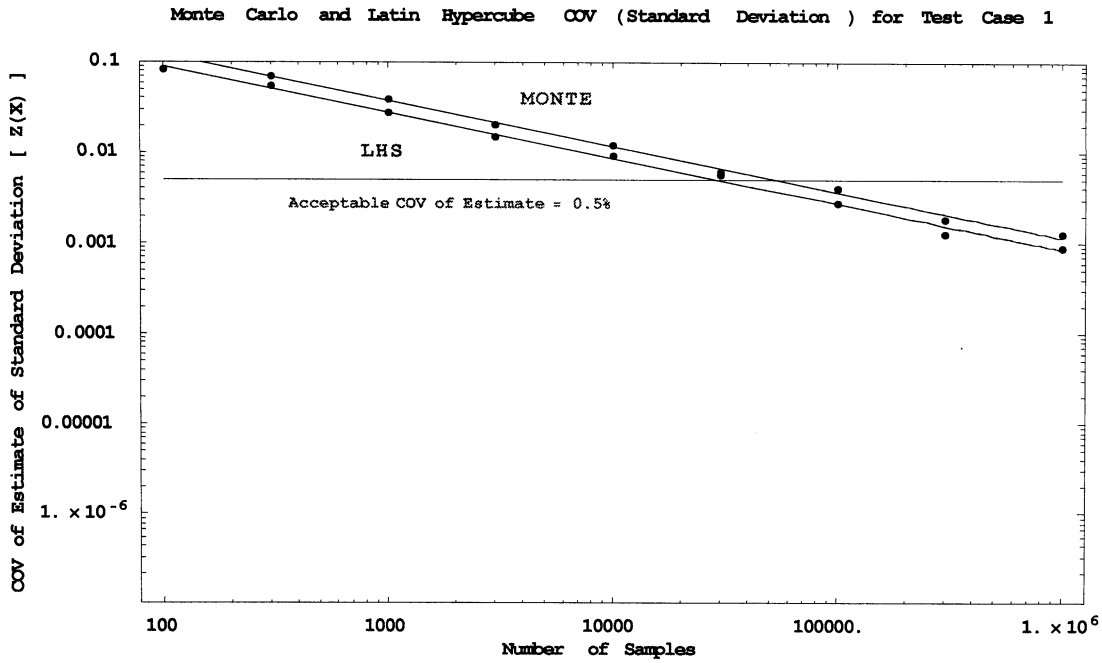


Figure 46 COV of standard deviation distributions for test case 1 response using MC and LHS

The 99th percentile of the test case 1 response was also estimated many times using MC and LHS. The percentile estimator is a function of random variables and is therefore random, has a certain distribution, and will vary with the number of response evaluations used to estimate the percentile, n, and the method used to obtain coordinate sets, MC or LHS. The corresponding MC and LHS 99th percentile distributions are shown below in Figure 47.

The distribution of the 99th percentile estimator will be different for each method and for each amount of samples used to obtain each value of the 99th percentile. This variation is shown in Figure 47. Both methods have a 99th percentile distribution that is positively skewed when 100 samples were used to capture the respective distribution. Above this sample level, the distributions are approximately normally distributed.

The distribution of the 99th percentile estimator using both MC and LHS is apparently biased with respect to the exact percentile value for the n=100 and 300 sample, or effort levels,

shown. The bias is the difference between the mean of a distribution and the target parameter, or true 99th percentile. Starting at n=1,000 the magnitude of the bias for both methods was calculated to be under 1,000 response units (cycles) and lowers thereafter. A bias of 1,000 cycles implies that the mean of the respective distribution is off by about a 2% error with respect to the true 99th percentile. So, by quantifying the bias error we can conclude that the distributions are negligibly biased throughout. However, the question remains as to why the 99th percentile distribution for both methods exhibits such a bias at the two lowest sample levels. This can be answered by thinking about the response of test case 1. It has a density that can be estimated by calculating n samples, or values of the response. This density has certain parameters, like a mean, standard deviation, and 99th percentile, associated with it. The density, and its parameters, can be approximated by taking, for example, n=100 response evaluations using MC and calculating their statistics, which are estimations of the density parameters. The problem is that like any random variable, its values will be centered about the mean (average value) and the mode (most likely value), so when estimating a density with a small amount of samples, the distribution estimation will first begin to be acceptable around the true mean and mode values of the response. A single percentile estimation using a few amount of samples will therefore be closer to the mean and mode, which are lower values than large percentiles like a 99th percentile and a higher values than small percentiles like a 1.0 percentile. It is therefore likely that a single estimation of a 99th percentile will be less than the true value of this percentile and multiple values of this estimate will be centered around lower values when a small number of response samples are used to estimate this percentile. This explains why the 99th percentile distributions shown in Figure 47 are negatively biased when few samples were used to estimate this parameter of the distribution of the response. Therefore, a 99th percentile distribution, like

the ones in Figure 47, can be expected to be unbiased so long as a sufficient number of response evaluations are performed.

The standard error of the 99th percentile estimator distributions shown in Figure 47 decrease as the number of samples used to calculate each percentile increases, so calculating the percentile a number of times will result in values that are closer to each other when the amount of response values used to calculate each percentile is large. The LHS-99th percentile distribution has a visibly lower standard error than the MC 99th percentile distribution for all number of samples shown in Figure 47.

Approximate confidence statements can be made from Figure 47 using the middle 50% of the box plots for each distribution. The 50% confidence statements for the 99th percentile distribution of this test case will be made at the n=10,000 sample level for both methods. When MC-10,000 was used to calculate the 99th percentile of the test case 1 response, it is 50% likely that this single estimate calculated will be within 1% of the target parameter or the true 99th percentile. This is about 470 cycles on either side of the target. Using LHS-10,000 is used to calculate the 99th percentile, it is 50% likely that the single estimate calculated will be within 0.7% of the target. This is about 340 cycles on either side of the target. Therefore, for this test case, at the n=10,000 and 50% effort and confidence levels, respectively, it was found that LHS had the lower error of 0.7% from the true 99th percentile than the MC error of 1%. The same effort and confidence were used and LHS had a lower error than MC.

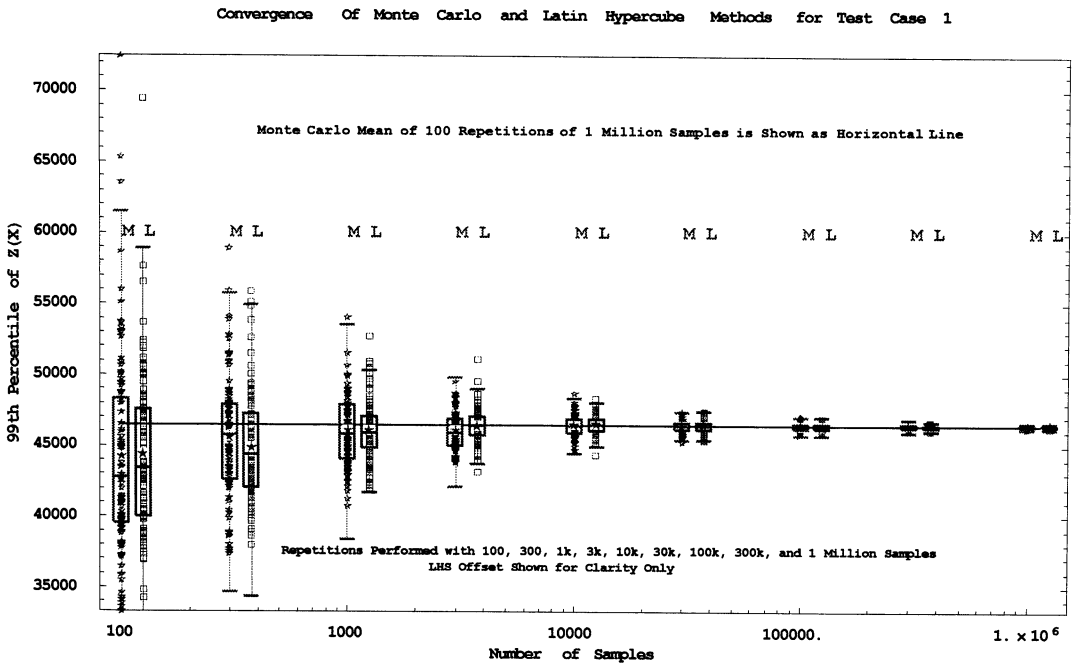


Figure 47 Distributions of 99th percentile for test case 1 response using MC and LHS

The COV $[\equiv \sigma/\mu]$ of the percentile distributions shown in Figure 47 are measures of the variation of repeated percentile estimates about the mean of the specific 99th percentile distribution, or about the true 99th percentile value once the respective distribution centers around this true value. The COV of the 99th percentile distributions using MC and LHS as it varies over the number of samples used per 99th percentile calculation is shown in Figure 48. Both methods show the same –1/2 rate of convergence to lower COV levels. Also, from Figure 48, one can see that the MC 99th percentile distribution would reach a COV level of 0.5% using over 100,000 samples used for each percentile estimate, while the LHS method needed about 80,000 samples to converge to the same level. Both distributions can be considered to be normal and unbiased at those respective sample levels; therefore, they are the same distribution. A COV level of 0.5% implies that $3\sigma_{\hat{\theta}}$ is 1.5% of the mean, which in this case is the true 99th percentile. Therefore, it can be stated with 99.7% confidence that a single 99th percentile estimate for the response of test

case 1 will be within ±1.5% of the target using MC-100,000. In comparison, there is a 99.7% chance that the same estimate will be within ±1.5% of the true 99th percentile using LHS-80,000. The range of 1.5% from the true 99th percentile is any value within 700 cycles from the target – the true 99th percentile. The LHS method required 20,000 less samples than the MC method. Therefore, it is better to estimate the 99th percentile for the test case 1 response using the LHS method, because it estimates this percentile with equal confidence and associated error, but with less effort, or numerical calculations than the MC method. These statements are based on the best-fit line for the data shown in Figure 48.

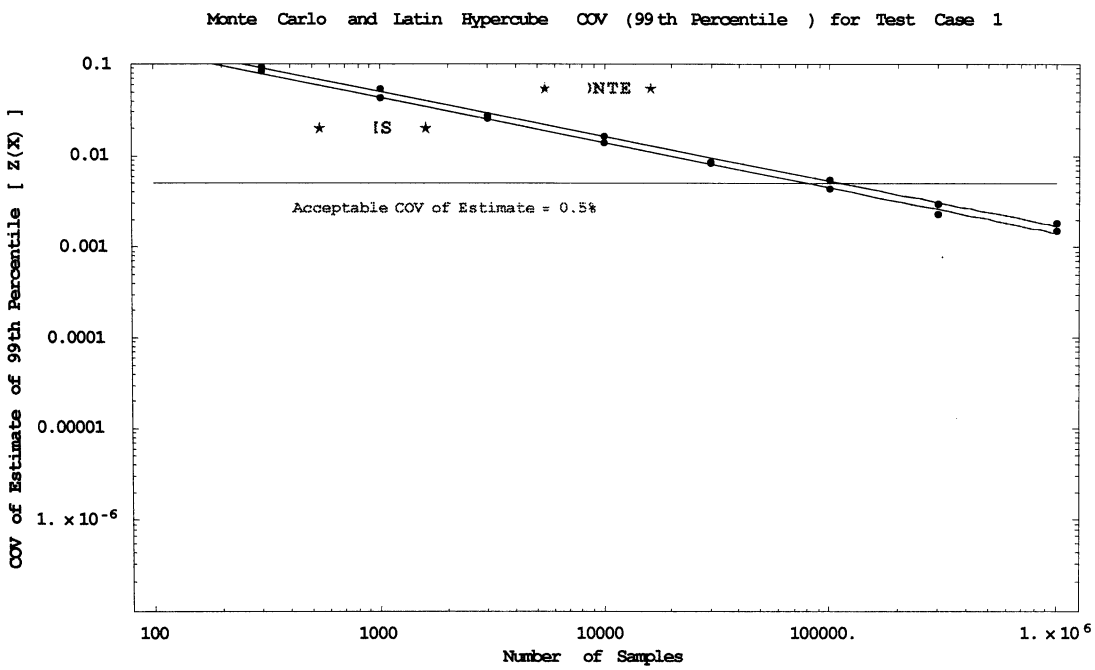


Figure 48 COV of 99th percentile distributions for test case 1 response using MC and LHS

Test Case 4: Nonlinear Response, Non-Normal Variables

Response Function and Design Variables

The response for the fourth case is nonlinear in one variable and linear in another. It is shown in mathematical form in Equation 26.

$$Z = X_1^{3.5} - 100X_2 + 50 \tag{26}$$

The design variable statistics are shown in Table 6. One of the design variables is a uniform distribution whose lower and upper bounds are zero and 100, respectively. The mean of this distribution is 50 and its standard deviation is about 29. The other random design variable is exponentially distributed, whose only parameter, β , is 0.05. This random variable has a mean of 0.05, which is also the standard deviation. The response given by 26 is purely mathematical and therefore any measurement on its scale will be discussed in terms of "units of the response".

Table 6 Design variables for test case 4

Variable	*Description*	*Distribution*
X_1	N/A	U (min=0, max=100)
X_2	N/A	E ($\beta = 0.05$)

Convergence of Sampling Methods

For the response of test case 4, given by Equation 26, the mean density parameter was estimated many times using MC and LHS. This estimator is random, and its distribution the will vary with the number of response evaluations used to estimate the mean, n, and the method used to obtain the coordinate sets, MC or LHS. This variation in the distribution of means was captured and is shown in Figure 49. Both methods appear to posses a mean distribution that is normally distributed for all the number of samples used to calculate each mean shown, that is, they possess symmetry about the distribution median value and they are therefore unskewed. The distribution of the mean estimator using both MC and LHS is centered about the exact, or target, value for all the number of samples used to calculate each mean estimate shown. In other words, both MC and LHS produce an unbiased mean estimator when they are used to capture its distribution. This is eye to eye with Wackerly et. al. (1996) which writes that the mean estimator distribution is unbiased and normal for sample sizes greater than or equal to 30.

Using either MC or LHS samples, the standard error of the distribution of means decreases as the number of samples, or response evaluations, used to calculate each mean increases, as shown in Figure 49. Hence, repeated estimates of the mean of the test case 4 response will be closer the exact value of the mean of the response (because the estimators are unbiased) when a large number of samples are used to calculate each mean estimate. The standard error of the distribution of means using LHS is much smaller than the same distribution captured using MC when the two distributions are compared at the same sample level, and for all of the sample levels shown in Figure 49.

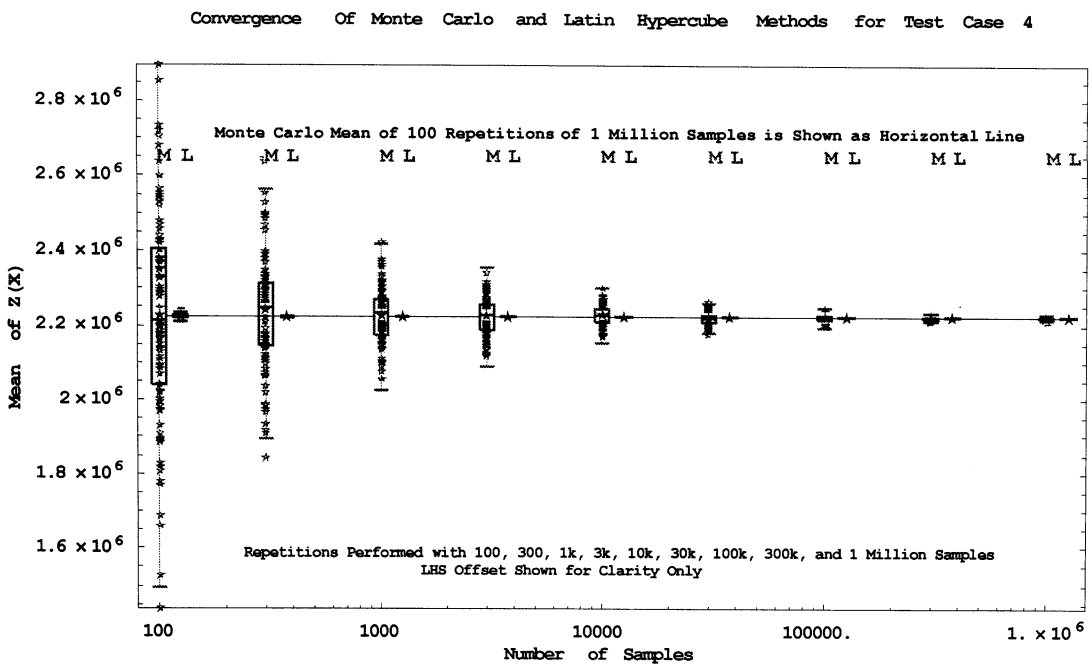


Figure 49 Distributions of means for test case 4 response using MC and LHS

Confidence statements allow one to compare the efficiency of different methods in obtaining estimates of density parameters, like the mean of a response, and can be made from the information given in Figure 49 using the middle 50% of the box plots shown. Using 1,000 Monte Carlo samples to calculate a single mean, it is 50% likely that this single estimate will be within 2.2% of the target parameter. This error implies a range of about 48,000 response units on either side of the true mean. Using 1,000 Latin Hypercube samples to calculate a single mean of the test case 4 response, it is 50% likely that it will be within 0.003% of the true mean, which is about 65 response units to either side. Therefore, for this test case, LHS gives an analyst the same confidence that a single mean estimate will have a much lower error than MC.

The COV [$\equiv \sigma/\mu$] of the mean estimator distribution using MC and LHS as it varies over a range of sample levels that the repeated mean estimations were performed to capture the respective distribution of the means is shown in Figure 50. The horizontal line at a COV of 0.002 (0.2%) is shown to emphasize the difference in effort required to obtain the same variation

about the mean of the distribution, also the true mean, for both methods.

From Figure 50, it is clear that MC and LHS show the different slopes, or rates of convergence to lower COV levels, in Log-Log space; and, the LHS method is shown to have a vastly lower COV than the MC method for any given sample level. The slopes are –1/2 and –1 using MC and LHS, respectively. Also, one can see that the mean estimator would have been observed to reach a COV level of 0.2% at n=400,000 using the MC method while the LHS method only needed about 100 samples to converge to the same level. Both distributions are unbiased and normal those respective sample levels. A COV level of 0.2% implies that $3\sigma_{\hat{\theta}}$ is 0.6% of the mean. From the information given in Figure 50, it can be stated with 99.7% confidence that a single mean estimate for the response of test case 4 will be within ±0.6% of the true mean using MC-400,000. In comparison, there is a 99.7% chance that the same type of estimate will be within ±0.6% of the true mean using LHS-100. An estimation error of 0.6% is 13,300 units from the true mean of test case 4. The type of confidence statement just made is of the type – equal confidence and error, and much, much less effort with LHS samples.

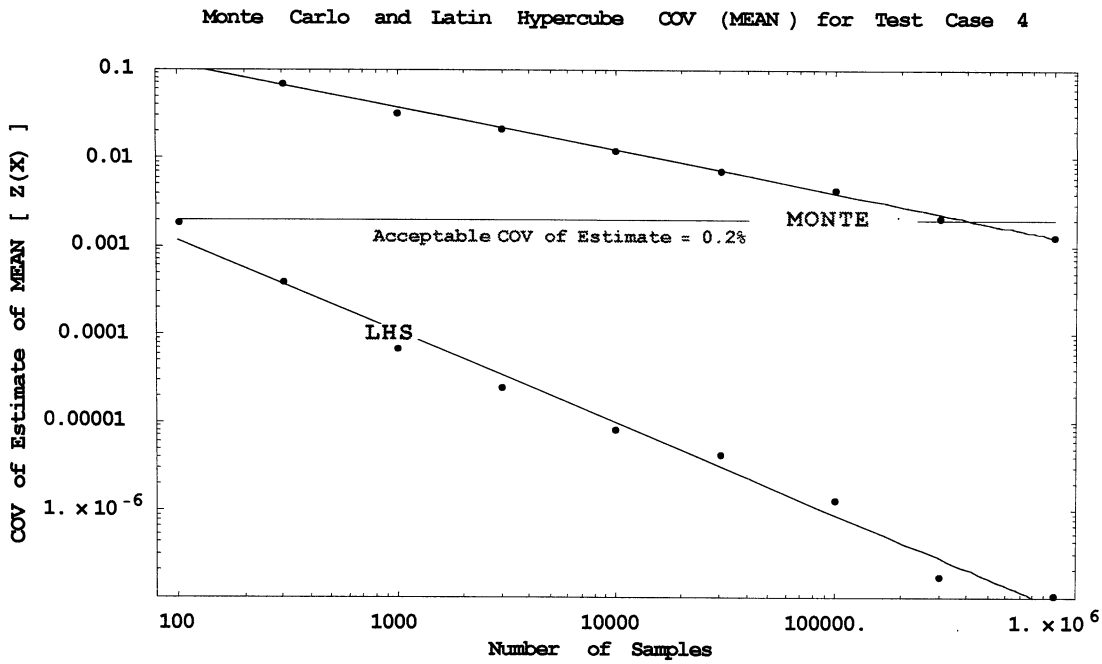


Figure 50 COV of mean distributions for test case 4 response using MC and LHS

The test case 4 response, given by Equation 26, is random and the standard deviation was estimated many times using MC and LHS. Even the estimator is random and will vary with the number of response evaluations used to estimate the standard deviation, n, and the method used to obtain the coordinate sets, MC or LHS. This variation is shown in Figure 51. Both methods have associated standard deviation distributions are normally distributed (symmetrical) for all the number of samples shown. Although the MC standard deviation distribution appears negatively skewed at the sample level of n=100, it is important to note that this distribution is not the exact. In a different set of likely circumstances, it could have been approximated differently, changing and possible lowering the skew of the estimate of the standard deviation. So, observing slight skew in the estimated distribution of standard deviations can be neglected and is not worthy of mentioning because the it could be reduced as the distribution of standard deviations is more accurately captured, or in a different set of random circumstances. Also, the distributions of the

standard deviation estimator using both MC and LHS are unbiased and, therefore, centered about the exact standard deviation value for all the number of samples, or effort levels, shown in Figure 51. The standard deviation estimator can be mathematically proven to be unbiased, and Figure 51 supplements those proofs by giving them another experimental verification [Wackerly et al, 1996].

The standard error of the standard deviation estimator distributions shown in Figure 51 tends to decrease as the number of samples used in a single standard deviation estimate increases. The LHS standard deviation distribution has a much lower standard error than the MC standard deviation distribution for all number of samples shown in Figure 51.

Using the middle 50% of the box plots shown, some important confidence statements can be made. When 300 Monte Carlo samples from each underlying random variable are paired with each other and used to form the coordinates needed to calculate 300 response values and the standard deviation of those responses is computed, it is 50% likely that a single standard deviation calculation will be within 2.25% of the target parameter or the true standard deviation. This error is about 62,000 response units on either side of the true standard deviation. Using 300 Latin Hypercube samples to calculate a single standard deviation of the test case 4 response, it is 50% likely that the single standard deviation estimate calculated will be within 0.13% of the true standard deviation, which is 3,500 units on either side of the target. Therefore, for this test case, at the n=300 and 50% effort and confidence levels, respectively, it was found that LHS had the lower error of 0.13% from the true standard deviation than the MC error of 2.25%. The same effort and confidence were used and LHS had a lower error than MC.

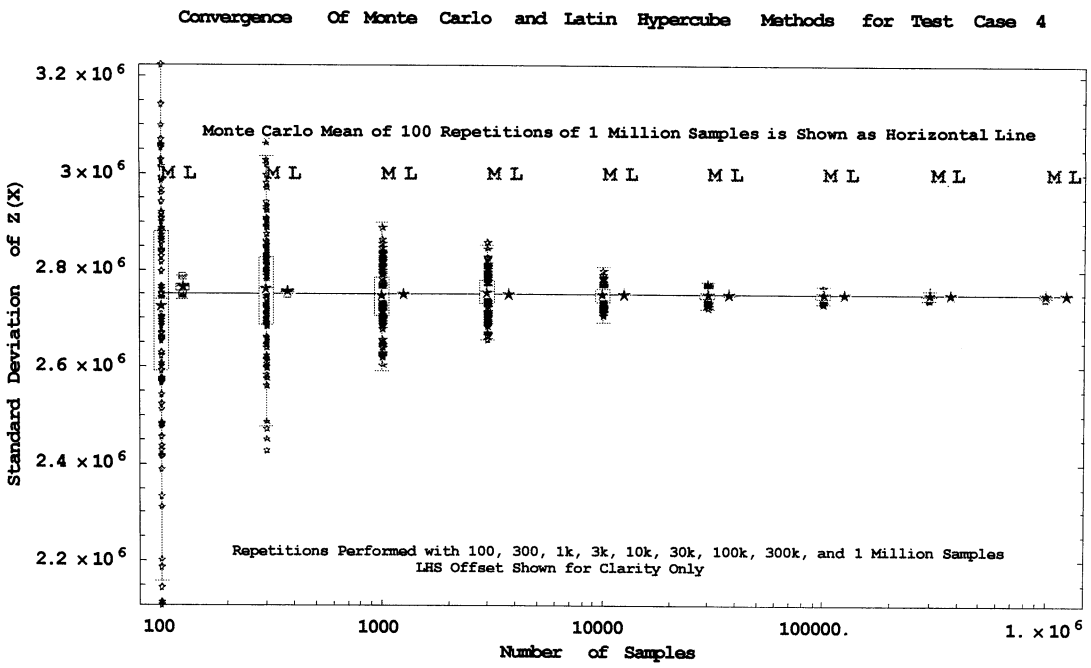


Figure 51 Distributions of standard deviations for test case 4 response using MC and LHS

The COVs of the standard deviation estimator distributions shown in Figure 51 are shown in Figure 52, along with a curve fit that approximates the COV all values between n-=100 and 1 million. The horizontal line at the COV value of 0.002 (0.2%) is shown to highlight the difference in computations required to obtain the same variation about the mean of the standard deviation distribution, considered to be the true standard deviation, for both methods. Essentially, if the respective distributions are unbiased, normal, and have the same COV, then they are the same distribution as far as the variation about the target parameter is concerned. An important thing to note from Figure 52 is that the rate of convergence to specific COV levels is greater for LHS than MC. The LHS method converges on the order of –1 while the MC method converges with a rate of –0.5.

From Figure 52, the standard deviation estimator distribution would have been observed to reach a COV level of 0.2% using MC with n=150,000 samples for each standard deviation estimate, while the LHS method needed only 100 samples to converge to the same level. Both distributions are normal and unbiased at those respective sample levels. A COV level of 0.2% implies that $3\sigma_{\hat{\theta}}$ is 0.6% of the mean, which in this case is the true standard deviation. Therefore, it can be stated with 99.7% confidence that a single standard deviation estimate for the response of test case 4 will be within ±0.6% of the true standard deviation using MC-150,000. In comparison, there is a 99.7% chance that the same estimate will be within ±0.6% of the true standard deviation using LHS-100. The range of ±0.6% from the true standard deviation implies any value within 16,500 units from this target. The LHS method will estimate the standard deviation of the test case 4 response with equal confidence and error, but with far less numerical calculations than the MC method. Hence, if that desired confidence and error was being sought after, and each response calculation took 10 minutes, LHS would produce a good result in a little over 16 hours, while MC would take almost three years to compute the same type of answer.

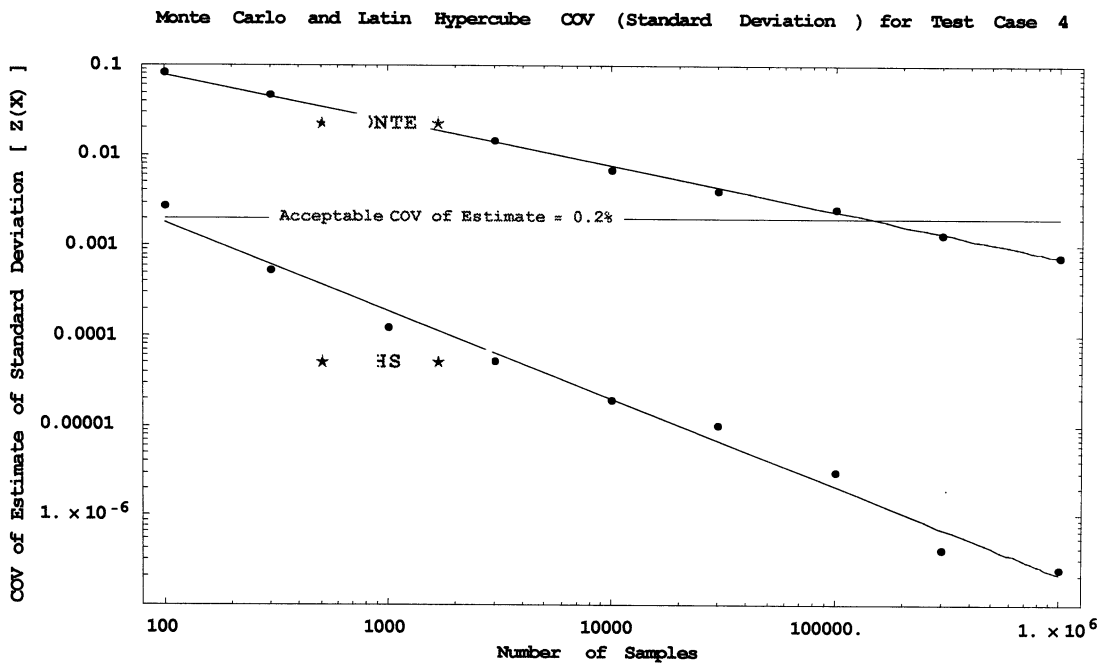


Figure 52 COV of standard deviation distributions for test case 4 response using MC and LHS

The 99th percentile of the response given by Equation 26 was estimated many times for the purpose of studying its distribution with respect to the number of response values calculated, n, and the sampling method used – MC or LHS. This variation is shown in Figure 53. The MC method has a 99th percentile distribution that is negatively skewed when 100 samples were used to calculate each percentile in that distribution. Above this sample level, the MC distributions are approximately normally distributed based on using 100 repetitions to capture the distributions shown. The LHS 99th percentile distributions are normally distributed for all of the response evaluation levels shown.

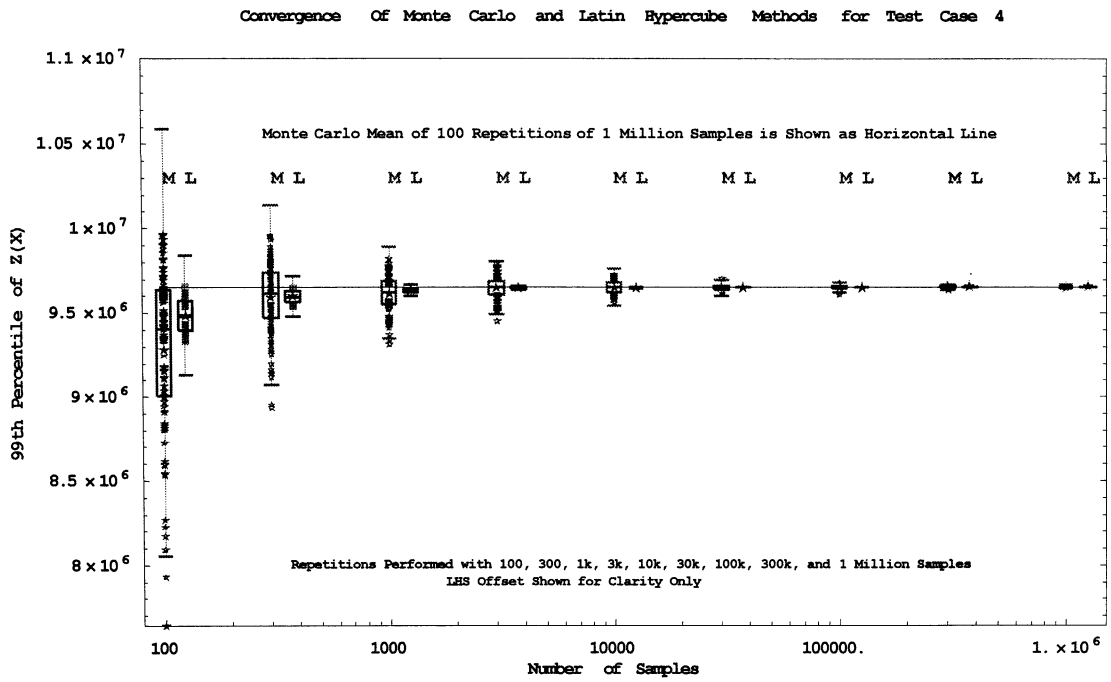


Figure 53 Distributions of 99th percentile for test case 4 response using MC and LHS

The distributions of the 99th percentile estimator using both MC and LHS are definitely biased with respect to the exact percentile value for the first few sample levels. These distributions are biased at the low sample levels because, in short, the test case 4 response density was estimated by calculating n samples, or values of the response. Estimating a parameter of the density, like a single percentile estimation, using few samples could lead to erroneous results because the response data that forms the response density will be closer to the mean and mode, which is are lower values than large percentiles like a 99th percentile and a higher values for small percentiles like a 1.0 percentile. It is therefore likely that a single estimation of a 99th percentile will be less than the true value of this percentile and multiple values of this estimate will be centered around a lower mean value when few response evaluations are used to estimate the density which this percentile comes from. That is the reason why the 99th percentile distributions shown in Figure 53 are negatively biased when few samples

were used to estimate this parameter of the distribution of the response. The biases reduce to nothing as more response evaluations are used to estimate each percentile. Therefore, the 99th percentile distribution for the response of test case 4, some of which are shown in Figure 53, can be considered unbiased so long as a sufficient number of response evaluations are performed. This allows the tails of the response to be properly estimated. Also, consider the percentile trying to be estimated – the 99th percentile, which is the value of the response that 99/100 values are equal to or below that value. The denominator is 100 and at least 1,000 response evaluations were necessary to reduce the MC 99th percentile distribution bias to about 40,000 response units, which, in this case is about a 0.4% error from the true 99th percentile. The LHS bias is negligible (less than 0.4%) at the n=1,000 sample level.

The standard error of the 99th percentile estimator distributions shown in Figure 53 tends to decrease as the number of samples used to calculate each percentile increases. The LHS 99th percentile distribution has a visibly lower standard error than the MC distribution for all number of samples shown in Figure 53.

Single estimate confidence when using a specific number of response evaluations to calculate each estimate is important because it is the probability that this estimate will lie within a specific error from the target. Using MC-3,000 to calculate 3,000 response values and the 99th percentile of those responses, it is 50% likely that the single estimate calculated will be within 0.34% of the target parameter or the true 99th percentile. This error, or interval, is about 33,000 response units on either side of the target. Using 3,000 LHS samples to form the coordinates necessary to calculate 3,000 response values that have a certain 99th percentile, it is 50% likely that the single estimate calculated will be within 0.04% of the target. This interval is about 3,800 response units away from the target. Therefore, for this test case, at the n=3,000 and 50% effort

and confidence levels, respectively, it was found that LHS had the lower error of 0.04% from the true 99th percentile than the MC error of 0.34%. The same effort and confidence were used and, yet, LHS had a lower error than MC.

The COVs $[\equiv \sigma/\mu]$ of the 99th percentile distributions of Figure 53 are shown in Figure 54. It must be noted that Figure 54 shows a faster rate of convergence to specific COV levels using LHS than MC. The LHS method converges on the order of –1 while the MC method converges with a rate of –0.5. Furthermore, for both methods, the COV decreases as the number of samples used to calculate each percentile value is increased. The horizontal line at the COV value of 0.001 (0.1%) is shown to emphasize the difference in effort required to obtain the same variation about the mean of the 99th percentile distribution, considered to be the true 99th percentile, for both methods on or after the sample level of 1,000.

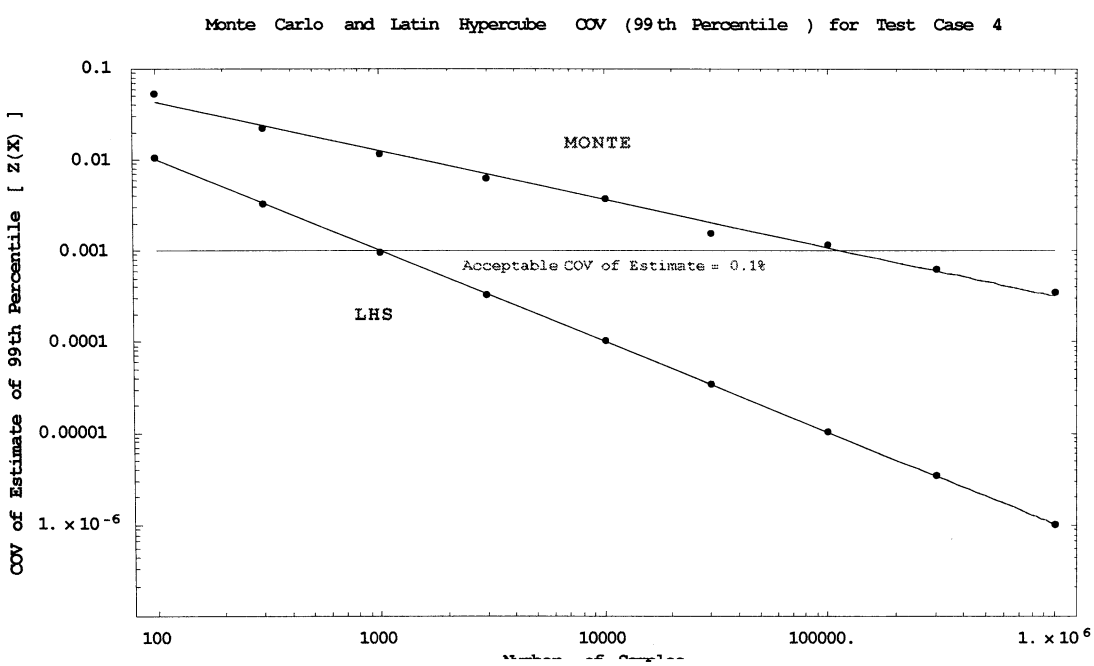


Figure 54 COV of 99th percentile distributions for test case 4 response using MC and LHS

From Figure 54 it is evident that the MC 99th percentile distribution reaches a COV level of 0.1% using over 100,000 samples for each percentile estimate. The LHS method needed only 1,000 samples to converge to the same level of variation. Both distributions can be considered to be normal and unbiased at those respective sample levels. A COV level of 0.1% implies that $3\sigma_{\hat{\theta}}$ is 0.3% of the distribution's mean, which in this case is the true 99th percentile. So, it can be stated with 99.7% confidence that a single 99th percentile estimate for the response of test case 4 will have an estimation error of ±0.3% from the target using MC-100,000. In contrast, there is a 99.7% chance that the same estimate will be within ±0.3% of the true 99th percentile using LHS-1,000. The range of 0.3% from the true 99th percentile is any value within 29,000 response units from the target. It is therefore better to estimate the 99th percentile for the test case 4 response using the LHS method, because it estimates this percentile with equal confidence and associated error, but with less effort, or numerical calculations than the MC method. These statements are based on the best-fit line for the data shown in Figure 54.

Test Case 6: Maximum Radial Stress of Rotating Disk

Response Function and Design Variables

Test case 6 studies a response that is the maximum radial stress of a rotating ring. The stress is solely due to the inertial forces acting on elements of the ring. An initial model of the system under mathematical study is shown in Equation 27.

$$\sigma_r^{\max} = \left(\frac{3+\nu}{8}\right)\left(\frac{\rho}{9.81 \cdot 39.37}\right)\left[\omega \frac{2\pi}{60}\right]^2 (r_o - r_i)^2 \tag{27}$$

The design variables include Poison's ratio, ν, the mass density, ρ, the rotational speed, ω, and the inner and outer radius, r_i and r_o, respectively. The Equation 27 can model a system so long as certain assumptions are true. One of them is that the outside radius more than 10 times the thickness of the ring. Another is that the thickness is constant and the stresses are constant over the thickness [Shigley and Mischke, 1989]. If any of these assumptions are not true for the physical part that is being mathematically modeled, then the actual stresses might be greater or less than that predicted with Equation 27. If the actual stresses are less than a prediction using Equation 27, then the mathematical model can be assumed to be conservative, and the result would be an over design of the ring. The system would still function properly. If the actual stresses in the ring are greater than what is expected using Equation 27 because the actual system does not follow one or more of the restrictions of that equation, the result could be a mechanical failure. If that isn't bad enough, it would surely be accompanied by the consequences of the failure: loss of money, time, reputation, and even injury. That is, unless Equation 27 is multiplied by a factor greater than one in order to reduce the modeling error associated with using Equation 27 with a system that is outside of the boundaries of the restrictions of that equation. Such is the issue for this test case. Thus, we have that the mathematical model of the system being studied here is shown in Equations 28 and 29.

$$Z = \sigma_r^{\max} = \left(\frac{3+\nu}{8}\right)\left(\frac{\rho}{9.81 \cdot 39.37}\right)\left[\omega \frac{2\pi}{60}\right]^2 (r_o - r_i)^2 M \tag{28}$$

$$M = \left(1 - \frac{2r_i}{r_o + r_i}\right)^{-1} \tag{29}$$

There might be a little doubt that M is indeed greater than 1. It will, so long as $0 < r_i < r_o$. Once we take a look at what is probable in the design variables we shall see that this will almost always be the case. The design variables, their description, distribution, and statistics for the 6th test case are shown in Table 7.

There are five design variables associated with this response. The design variable statistics are shown in Table 7. The density, ρ, is normally distributed with a mean of 0.284 lb/in^3 and a 0.7% COV. The inner radius is modeled as a normal random variable with a mean of 2 inches and a 0.5% COV. The outer radius has a mean of 8 inches, a 0.25% COV, and is normally distributed. Poisson's ratio was considered to be normally distributed with a mean of 0.30 a 1.67% COV. The only non-normal random variable used in this analysis is the rotor speed, which was modeled as a uniformly distributed random variable with a range from 10,000 rpm to 11,000 rpm. This type of variable has a mean of 10,500 rpm and a standard deviation of 288 rpm.

Table 7 Design variables for test case 6

Variable	*Description*	*Distribution*
ρ	Density (lb/in^3)	N (0.284, 0.002)
r_i	Inner radius (in)	N (2, 0.01)
r_o	Outer radius (in)	N (8, 0.02)
ν	Poisson's ratio	N (0.30, 0.005)
ω	Rotor speed (rpm)	U (min=10,000, max=11,000)

Recall the discussion about the multiplication factor and it was questionable if it was greater than one. Because of the distribution of r_i, we can be assured that it over 99.9999% probable that it will be between 1.95 and 2.05. This range is five standard deviations to either side of the mean. In the same light, it is also over 99.9999% likely that r_o will be between 7.9 and 8.1. Therefore, it is extremely unlikely that the multiplication factor will be less than one.

Convergence of Sampling Methods

For the response of test case 6, given by Equations 28 and 29 the mean was estimated many times using both MC and LHS. This estimator is random, and its distribution, as repeated mean estimates are made, will vary with the number of response evaluations used to estimate the mean, n, and the method used to obtain the coordinate sets used for response evaluations, MC or LHS. This variation in the distribution of mean estimates is shown in Figure 55.

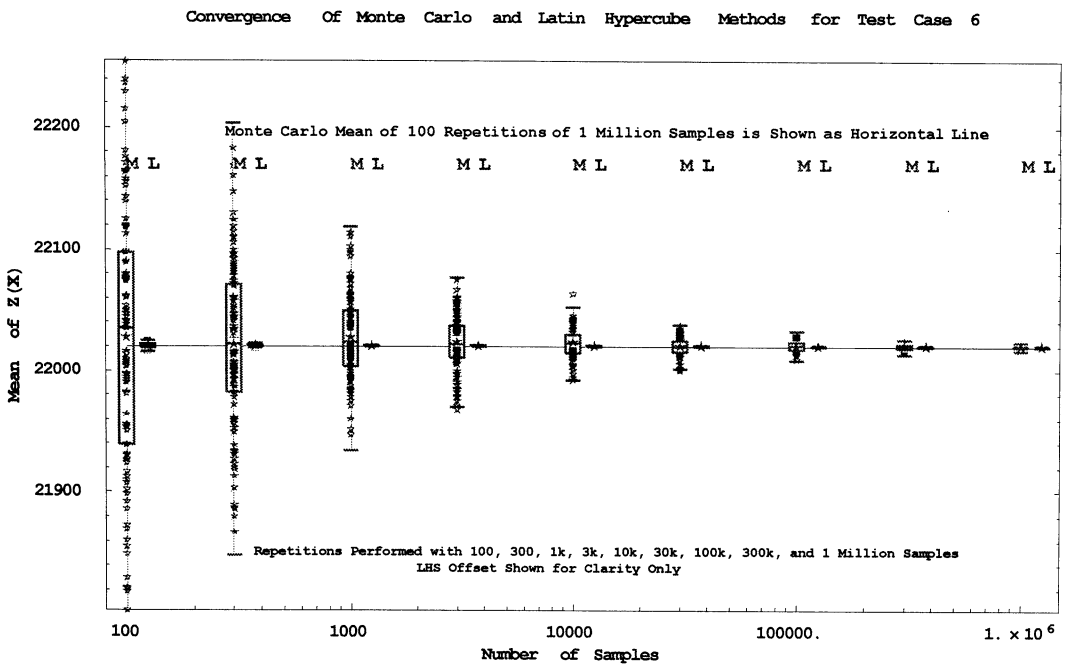


Figure 55 Distributions of means for test case 6 response using MC and LHS

Both methods appear to posses a mean distribution that is normally distributed at all of the sample levels shown. This fact is a supplement to the Wackerly et al. (1996) statement that mean estimator distribution is normal for sample sizes greater than or equal to 30. Both MC and LHS produce an unbiased mean distribution when they are used to acquire it. True, they bias of the distributions shown might not be numerically equal to zero; however, the magnitude of the bias of all the distributions shown in Figure 55 is slight and they can be considered unbiased. This is in agreement with Wackerly et al (1996). Also, the slight bias shown might disappear as the mean distributions are more accurately captured with more than 100 repetitions for each method and at each level.

The standard errors of the distributions in Figure 55 are shown to decrease as the number of samples used to calculate each mean increases. For that reason, repeated mean estimates will be centered about each other, or the mean of the respective distribution, more as the amount of

response values used to calculate the each mean gets larger. The distributions in are unbiased, so their mean is approximately equal to the exact value of the mean of the response. The standard error of the LHS mean distribution is much smaller than the MC mean distribution when the two distributions are compared at the same sample level. This is true for all of the sample levels shown in Figure 55.

Confidence statements are important because they are the numerical values that help evaluate the efficiency of different methods when they are used to obtain density parameter estimates, like the mean of a response. Using the middle 50% of the box plots shown in Figure 55, it is safe to say that when 1,000 MC samples are used to calculate a single mean estimate, it is 50% likely that this single estimate will be within 0.08% of the target parameter. This error implies a range of about 17 response units (psi) on either side of the true mean. Using LHS-1,000 for a single mean estimate of the test case 6 response, it is 50% likely that it will be within 0.0009% of the target parameter or the true mean which is about 0.2 psi on either side of the true mean. Therefore, for this test case, LHS gives an analyst the same confidence, or probability, that a single mean estimate will have a much lower error than MC.

The COV of the mean distributions shown in Figure 55 are shown in Figure 56. The COV is a measure of the variation of repeated mean estimates about the true mean for a specific number of samples only when the distribution under consideration is essentially unbiased. In Figure 56, the horizontal line at a COV of 0.00007 (0.007%) is shown to emphasize the difference in effort required to obtain the same variation about the mean of the distribution, also the true mean, for both methods. MC and LHS have the same rate of convergence, in Log-Log space. The rate of both methods is about –1/2.

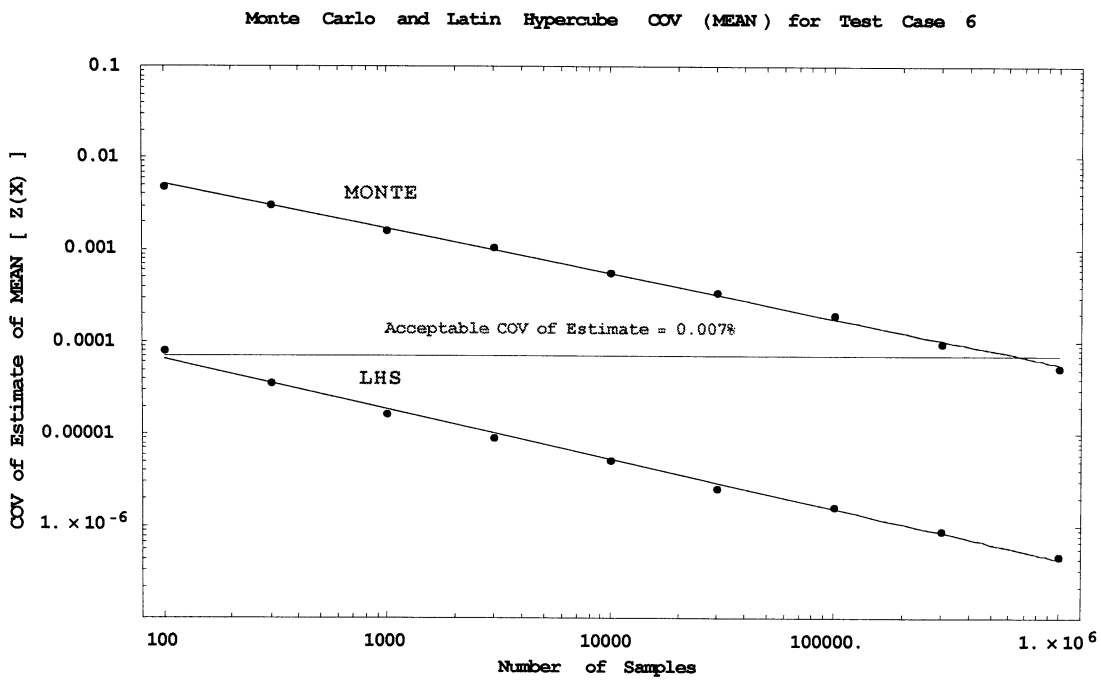


Figure 56 COV of mean distributions for test case 6 response using MC and LHS

One can see from Figure 56 that the mean estimator would have been observed to reach a COV level of 0.007% at n=700,000 using the MC method while the LHS method only needed about 100 samples to converge to the same level. Both distributions are unbiased and normally distributed at those respective sample levels; therefore, they have the same mean that is approximately equal to the true mean of the response. Since they have the same COV, they are the same distribution, centered about the true mean of the response for test case 6. A COV level of 0.007% implies that the standard error (standard deviation) of the distribution of the estimator is 0.007% of the mean, or $3\sigma_{\hat{\theta}}$ is 0.021% of the mean. From the information given in Figure 56, it can be stated with 99.7% confidence that a single mean estimate for the response of test case 6 will be within ±0.021% of the true mean using MC-700,000. In comparison, there is a 99.7% chance that the same type of estimate will be within ±0.021% of the true mean using LHS-100. An estimation error of 0.021% implies any value within 4.6 psi from the true mean of the test

case 6 response. The type of confidence statement just made is of the type – equal confidence and error, and much less effort with LHS samples.

The test case 6 response, given by Equations 28 and 29 has a standard deviation associated with it that was estimated a number of times. The distribution of the standard deviation estimator will vary with the number of response evaluations used to estimate the standard deviation, n, and the method used to obtain the coordinate sets, MC or LHS. This variation is shown in Figure 57. Both methods have associated standard deviation distributions are essentially normally distributed for all the number of samples shown. If that is questionable, recall that these distributions are not exact. In fact, given a different set of random and likely circumstances, each distribution could have been approximated differently, changing and possible lowering any slight skew (or non-normality) that is presented in Figure 57.

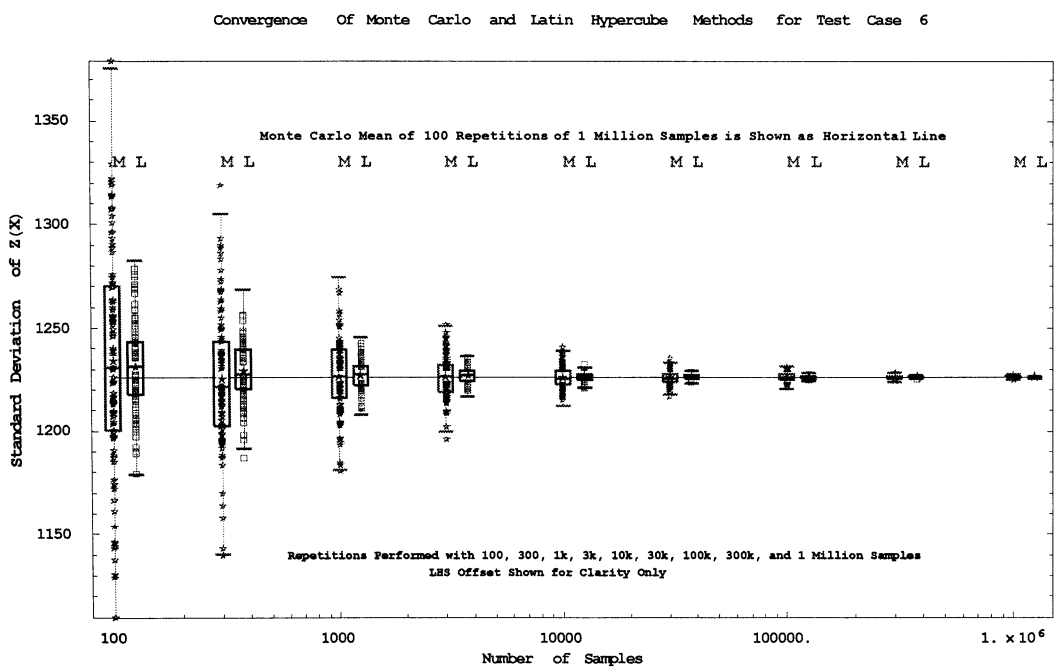


Figure 57 Distributions of standard deviations for test case 6 response using MC and LHS

The distribution of the standard deviation estimator using both MC and LHS is unbiased and, therefore, centered about the exact standard deviation value for all the number of samples, or effort levels, shown in Figure 57. Also, the standard error of the standard deviation estimator distributions for both methods tends to decrease as the number of response evaluations, used in a single standard deviation estimate increases. From this one can expect that when repeatedly calculating the standard deviation of the response of test case 6, the values will be closer to each other, or the mean of the distribution being formed by these repetitions, when the number of response values that form a set of data with an associated standard deviation is large – for both methods. Fortunately, the distributions are unbiased; therefore, their mean is equal to the target value – the standard deviation of the response. What we then have for both MC and LHS is that as more samples are used to calculate each standard deviation, the calculated values are more centered about the mean of the distribution being formed with every standard deviation estimate, which is also the target of interest. Figure 57 shows that the LHS standard deviation estimator distribution has a lower standard error than the MC distribution for all number of samples shown.

Knowledge of the confidence that can be place in a single estimate to be within an acceptable error limit when the estimate is made using a specific number of response evaluations is extremely important. Actual confidence statements can be made from Figure 57 using the middle 50% of the box plots shown. When 300 MC samples from each underlying random variable of test case 6 are paired with each other and used to form the coordinates needed to calculate 300 response values, and then, after that, the standard deviation of those responses is computed, it is 50% likely that a single standard deviation calculation will be within 1.4% of the target parameter or the true standard deviation. This error is about 17 psi on either side of the true standard deviation. Using 300 Latin Hypercube samples to calculate a single standard

deviation of test case 6, it is 50% likely that the single estimate calculated will be within 0.5% of the target parameter or the true standard deviation, which is 6 units on either side of the target. Therefore, for this test case, at the n=300 and 50% effort and confidence levels, respectively, it was found that LHS had the lower error of 0.5% from the true standard deviation than the MC error of 1.4 %. The same effort and confidence were used and LHS had a lower error than MC.

The COV $[\equiv \sigma/\mu]$ of the standard deviation distributions shown in Figure 57 are plotted and fit to a curve in Figure 58. For this estimator, the COV is a measure of the variation of repeated standard deviation estimates about the true standard deviation for a specific standard deviation distribution, only because these distributions are essentially unbiased, as discussed when Figure 57 was considered. The horizontal line at the COV value of 0.005 (0.5%) is shown to draw attention to the difference in computations required to obtain the same variation about the mean of the respective standard deviation distribution, essentially the true standard deviation, for both methods. Basically, two distributions would be identical as far as the variation about the target parameter is concerned if the respective distributions are unbiased, normal, and have the same COV. One thing also apparent from Figure 58 is that the rate of convergence to specific COV levels is the same for MC and LHS. This rate is also the slope of the Log-Log curve fit. Both methods converge on the order of –0.5.

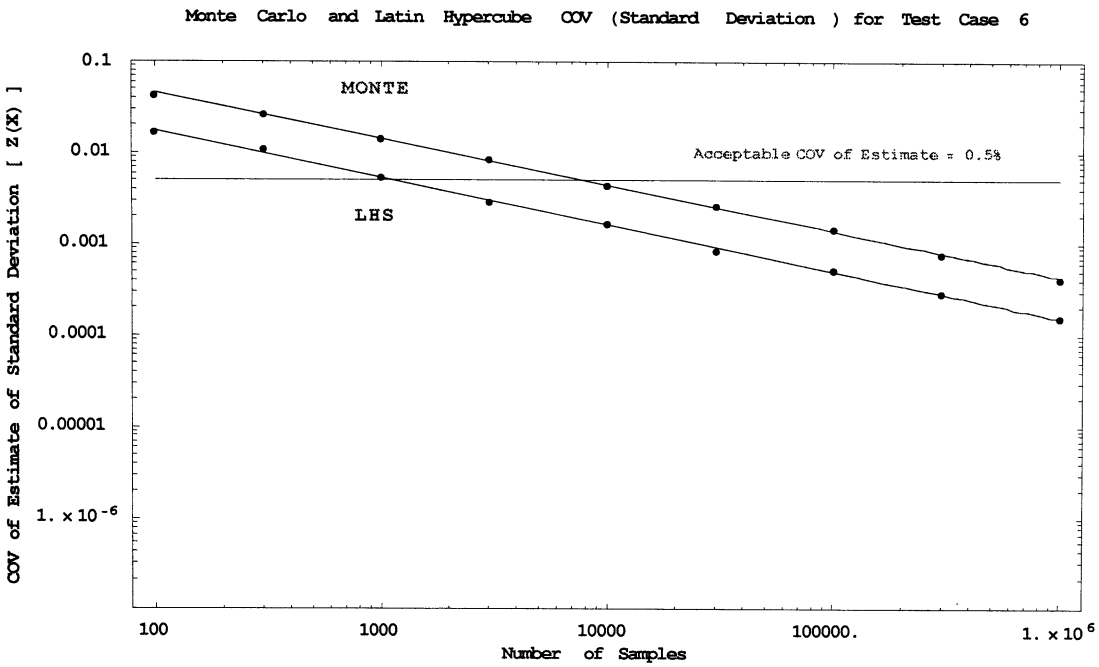


Figure 58 COV of standard deviation distributions for test case 6 response using MC and LHS

Furthermore, from Figure 58, the MC standard deviation distribution would have been observed to reach a COV level of 0.5% using 8,000 samples, while the LHS method needed 1,000 samples to converge to the same level. Both distributions are normal and unbiased at those respective sample levels. Therefore, it can be stated with 99.7% confidence that a single standard deviation estimate for the response of test case 6 will be within ±1.5% of the true standard deviation using MC-8,000. In comparison, there is a 99.7% chance that the same estimate will be within ±1.5% of the true standard deviation using LHS-1,000. An error of 1.5% implies a range of 18.4 psi to either side of the true standard deviation. The LHS method will estimate the standard deviation of the test case 6 response with equal confidence and error, but with far less numerical calculations than the MC method. The number of calculations is indicative of the time it would take to obtain a confident answer. The LHS method will produce a good result in 1/8 the time it would take MC to compute the same type of answer. These

statements are based on the best-fit line for the data shown in Figure 58.

The 99th percentile of the test case 6 response was estimated many times for the purpose of studying its distribution with respect to the number of response values calculated for each estimate, n, and the sampling method used – MC or LHS. These variations are shown in Figure 59. The MC method has a positively skewed, and therefore non-normal, 99th percentile distribution when n=100 samples were used to calculate each percentile in that distribution. Linking a skew statement to a normality statement is easy because as the skew of a distribution is removed it becomes symmetrical about the median value of the distribution, and normally distributed random variables will exhibit symmetry about the median value of the distribution, which in that case is equal to the mean and the mode (most probable). Above this sample level, the MC-99th percentile distributions are approximately normally distributed based on using 100 repetitions to capture the distributions shown. The LHS 99th percentile distributions are considered normally distributed for all of the response evaluation levels shown in Figure 59.

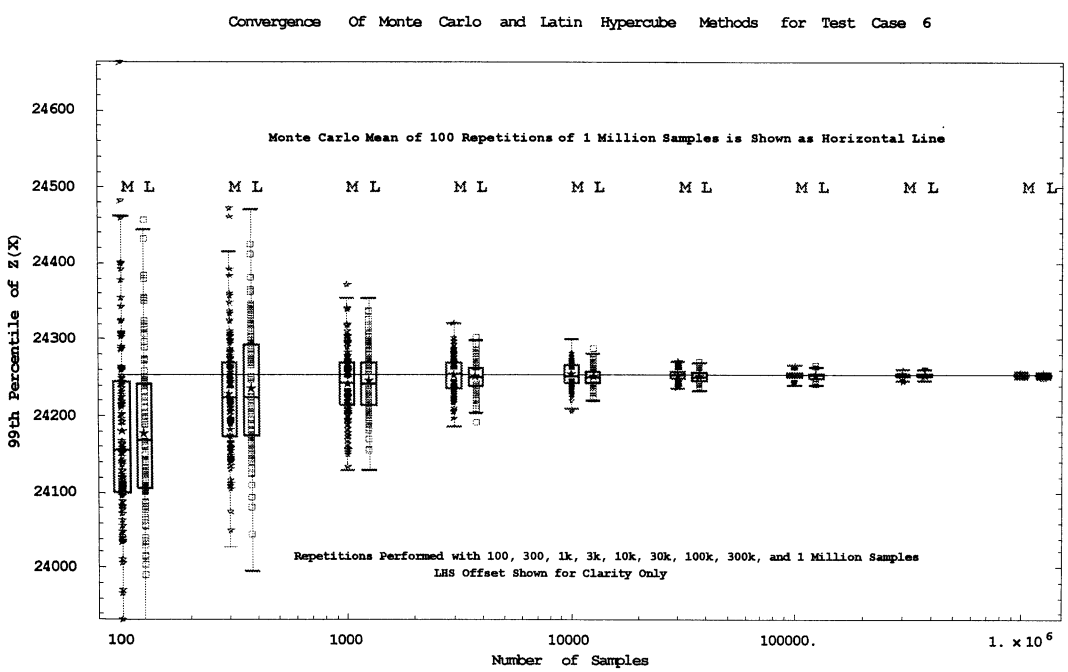


Figure 59 Distributions of 99th percentile for test case 6 response using MC and LHS

The distribution of the 99th percentile estimator using both MC and LHS appear biased with respect to the target value for the first few sample levels. The bias is the difference between the mean of the respective distribution of percentiles and the exact 99th percentile. Starting at n=100 the magnitude of the MC and LHS 99th percentile distributions bias is about 70 psi maximum and approaches zero after this level. This negative bias can be significant or negligible depending on the reponse under consideration and what type of error is acceptable. In any case it represents only a fraction of a percent of error (0.3%) from the true value of the 99th percentile. Furthermore, the bias of the MC and LHS 99th percentile distributions are lower than this at n=300 and above, which results in an even lower error. For all intensive purposes, the distributions are considered unbiased for all sample levels shown due to the small error with respect to the true value.

The standard error of the 99th percentile estimator distributions shown in Figure 59 tends to decrease as the number of response evaluations used to calculate each percentile increases. The LHS distributions have visibly lower standard errors than the MC distributions for all number of samples shown, except for n=300 and 1,000. While it is not visibly lower, for the sake of leaving nothing to question, the standard errors of the MC 99th percentile distribution at n=300 and n=1,000 were calculated to be 76 and 45 psi, respectively. Furthermore, standard errors of the LHS 99th percentile distributions at the n=300 and 1,000 sample levels were calculated to be 75 and 40 psi, respectively.

Single estimate confidence using a specific number of response evaluations to calculate each estimate is important. Confidence is a measure of the probability that this estimate will lie within a specific error from the target. Confidence statements can be made from Figure 59 using the middle 50% data of the box plots for each distribution. Using MC-3,000 to calculate a single

99th percentile estimate, it is 50% likely that the single estimate calculated will be within 0.07% of the target parameter or the true 99th percentile. This error, or interval, is about 16 psi on either side of the target. Using 3,000 LHS samples to calculate a single 99th percentile estimate, it is 50% likely that the single estimate calculated will be within 0.04% of the target. This interval is about 9 psi away from the true parameter. Therefore, for this test case, at the n=3,000 and 50% effort and confidence levels, respectively, it was found that LHS had the lower error of 0.04% from the true 99th percentile than the MC error of 0.07%. The same effort and confidence were used and, yet, LHS had a lower error than MC.

The COV $[\equiv \sigma/\mu]$ of the 99th percentile distribution will vary across the effort levels and method used to obtain the response values and this change is shown in Figure 60. For both methods, the COV decreases as the number of samples used to calculate each percentile value is increased. The horizontal line at the COV value of 0.002 (0.2%) is shown to emphasize the difference in effort required to obtain the same variation about the mean of the 99th percentile distribution. Because they are unbiased throughout the sample levels, the mean is considered the true 99th percentile. It also must be noted that Figure 60 shows both MC and LHS have rates of convergence to specific COV levels that are equal and on the order of –0.5.

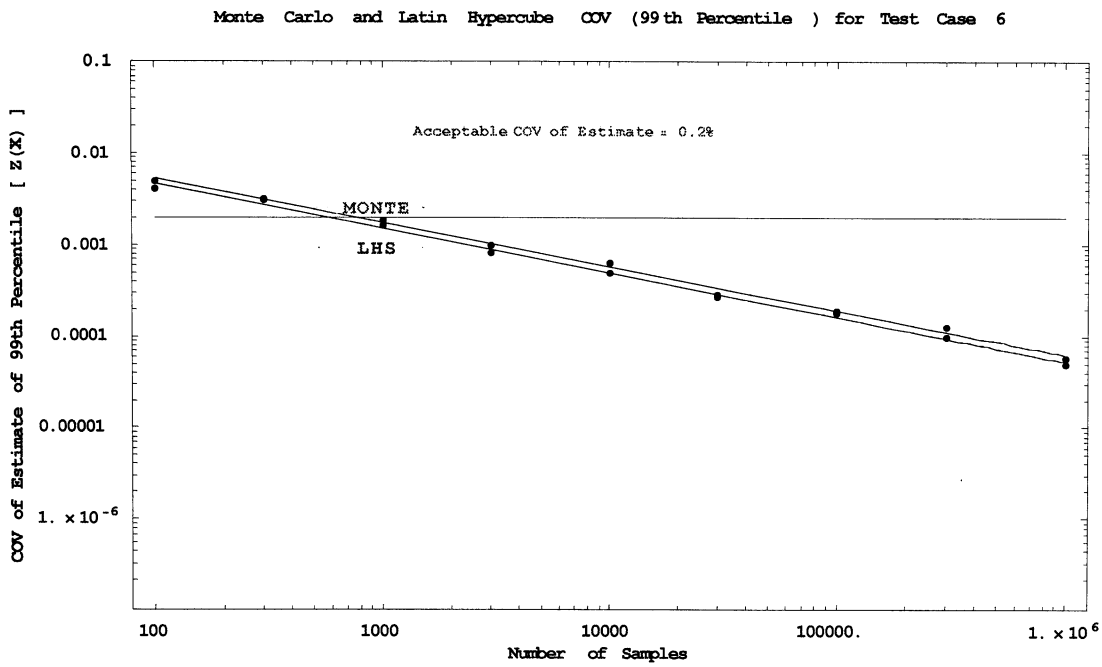


Figure 60 COV of 99th percentile distributions for test case 6 response using MC and LHS

From Figure 60, it is evident that the MC 99th percentile distribution would have been observed to reach a COV level of 0.2% using about 800 samples for each percentile estimate. The LHS method would about 600 samples to converge to the same level of variation. Both distributions can be considered normal and unbiased at those respective sample levels; therefore, they have the same mean that is approximately equal to the true 99th percentile of the test case 6 response. For that reason, it can be stated with 99.7% confidence that a single 99th percentile estimate for the response of test case 6 will have an estimation error of ±0.6% from the target using MC-800. In contrast, there is a 99.7% chance that the same estimate will be within ±0.6% of the true 99th percentile using LHS-600. Any value within 145.5 psi from the true 99th percentile will be within this estimation error. Indeed, an efficiency difference of 200 samples is slight. However, this does show that it is therefore better to estimate the 99th percentile for the test case 6 response using the LHS method, because it estimates this percentile with equal

confidence and associated error, but with less effort, or numerical calculations than the MC method. These statements are based on the best-fit line for the data shown in Figure 60.

Test Case 8: Nonlinear Response, Standard Normal Variables

Response Function and Design Variables

The response for test case 8 is purely mathematical and is given by Equation 30. Several statistics of this response will be repeatedly estimate using Monte Carlo and Latin Hypercube methods. It is dependent on two variables that are considered to be random.

$$Z = 3 - X_1^{\ 2} + 2X_1^{\ 4} - X_2 \tag{30}$$

The two design variables' statistics are shown in Table 8. Both of the underlying random variables are normally distributed with a mean of zero and a standard deviation of one.

Table 8 Design variables for test case 8

Variable	*Description*	*Distribution*
X_1, X_2	NA	N (0, 1)

Convergence of Sampling Methods

For the response of test case 8, given by Equation 30, one of the density parameters, the mean, was estimated many times. The mean estimator, being a function of random variables, is also random, and has a certain distribution that will vary with the number of response evaluations used to estimate the mean, n, and the method used to obtain the coordinate sets – MC or LHS. This variation of the mean estimator distribution is shown in Figure 61. Generally speaking, both methods have an associated mean distribution that is at least mound shaped for all the

number of samples used to calculate each mean shown. While it would be nice to say that they are normally distributed throughout all of the levels shown, for n=100 and n=300, and both methods, there appears to be slight positive skew in those respective distributions. On the other hand, the magnitude of the mean to median offset, which generally implies a skewed distribution, is small for these distributions compared to the range of the each distribution. Also, these distributions are not the exact and only estimates based on 100 repetitions. So, observing slight skew with respect to the estimated distribution of means, while it is a true observation, it can be discarded because the skew could be reduced as the distribution of means is more accurately captured, or even in a different set of random circumstances. It can therefore be said that both methods have an associated mean distribution that is approximately normally distributed for all the number of samples used to calculate each mean and this agrees with Wackerly et al. (1996) which states that the distribution of the mean estimator is normal for sample sizes greater than or equal to 30.

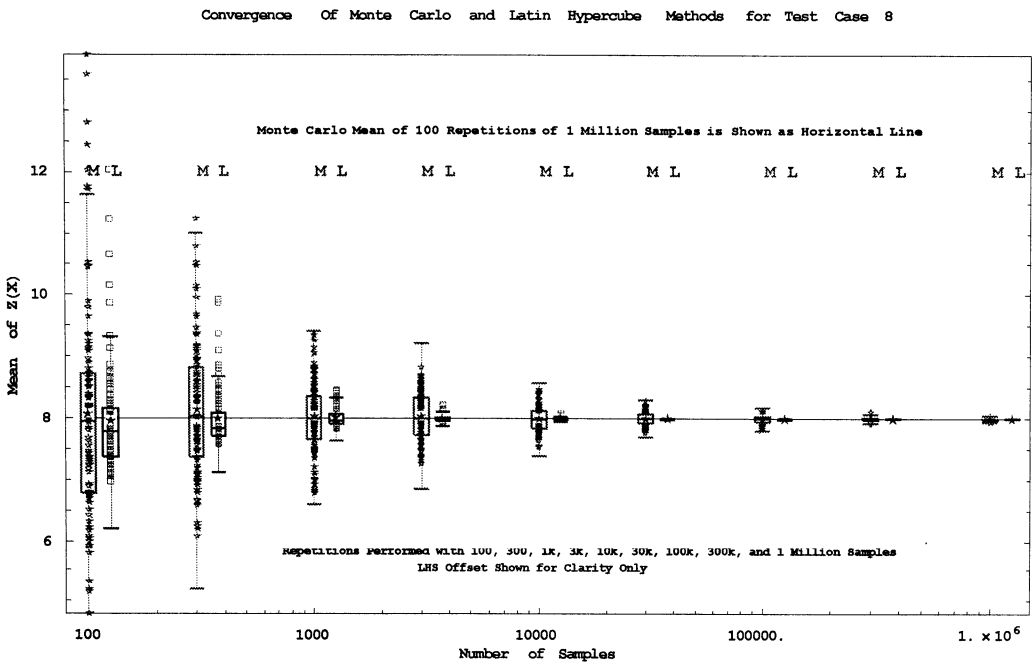


Figure 61 Distributions of means for test case 8 response using MC and LHS

The distribution of the mean estimator using both MC and LHS is centered about the exact value for all the number of samples used to calculate each mean estimate shown in Figure 61. That is, both MC and LHS produce unbiased mean estimators when they are used to capture its distribution.

Examining the standard error, or standard deviation, of the distributions shown in Figure 61 can be another way to measure the goodness of the estimators. For both MC and LHS samples, the standard error of the standard deviation distributions decrease as the number of samples, or response evaluations, used to calculate each mean of the responses increases. For the sake of comparing MC and LHS it must be noted that the standard error of the distribution of means using LHS samples to evaluate the response and calculated each mean is smaller than the distribution captured using MC samples when the two distributions are compared at the same sample level. This is true for all of the sample levels shown in Figure 61.

Usually, only a single estimate of a parameter can be afforded, so it is vital for confidence to exist for that the estimate, while even if it is not close to the true value, will be within an acceptable error limit. A quantitative statement like this can be made from the information given in Figure 61 using the middle 50% of the box plots shown. Using MC-1,000 to form the coordinates utilized to calculate 1,000 response values and, after that, calculate the mean of those responses, it is 50% likely that the single mean estimate calculated will be within 4.2% of the target parameter or the true mean – roughly about 0.34 units on either side of the true mean. In comparison, using LHS-1,000 to calculate a single mean estimate, it is 50% likely that the estimate calculated will be within 1.0% of the target parameter or the true mean which is about 0.08 units on either side of the true mean. In summary, the effort and confidence level were set to 1,000 samples and 50%, respectively for both methods. It was then found that LHS had the

lower error of 1.0% from the true mean than the MC error of 4.2%. Therefore, for this test case, LHS gives an analyst the same confidence (probability) that a single mean estimate will have a lower error than MC, with the same amount of effort.

The COV $[\equiv \sigma/\mu]$ of each distribution of the means will change as the effort, or number of samples used to calculate each mean value, is increased for both of the methods used to obtain response values. Figure 62 shows the COV of the mean estimator distributions acquired using MC and LHS. The horizontal line at the COV value of 0.005 (0.5%) is shown to highlight the difference in effort required to obtain the same variation about the mean of the distribution of means, also the true mean (they are unbiased at all levels), for both methods. It also must be noted that the rates of convergence to smaller COV levels for MC and LHS are different. This rate is the slope of the curve fit line in Log-Log space. The MC method converges on the order of $-1/2$, while the LHS method converges on the order of $-1.75/2$.

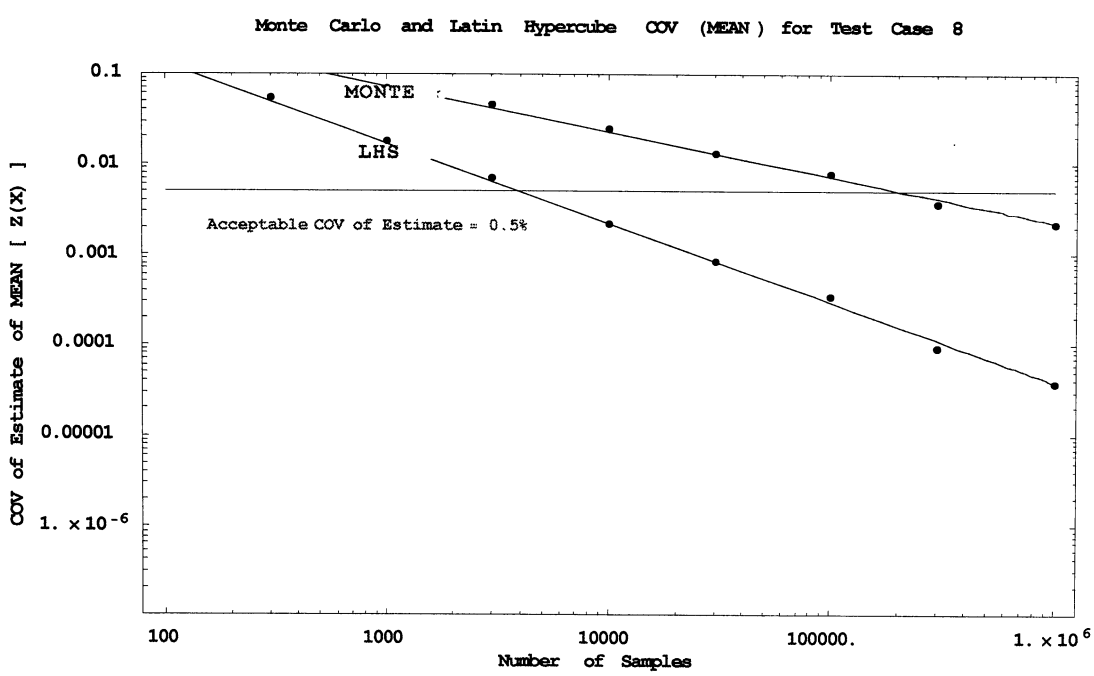


Figure 62 COV of mean distributions for test case 8 response using MC and LHS

From Figure 62 one can see that the mean estimator would have been observed to reach a COV level of 0.5% using MC-200,000 while the LHS method needed about 4,000 samples to converge to the same level. Both distributions can be considered to be normal and unbiased at those respective sample levels; therefore, they have the same mean that is approximately equal to the true mean. From the information given in Figure 62, it can be stated with 99.7% confidence that a single mean estimate for the response of test case 8 will be within ±1.5% of the true mean using MC-200,000. In comparison, there is a 99.7% chance that the same estimate will be within ±1.5% of the true mean using LHS-4,000. The interval of ±1.5% is any value within 0.12 units from the true mean. The type of confidence statement just made is of the type – equal confidence and error, different effort.

For the response of test case 8, the standard deviation was estimated many times. The standard deviation estimator will vary with the number of response evaluations used to estimate the standard deviation, n, and the method used to obtain the coordinate sets, MC or LHS. These various MC and LHS distributions are shown below in Figure 63. Both methods have an associated standard deviation distribution that is mound shaped for all the number of samples shown. Unfortunately, the MC and LHS $\hat{\theta}_\sigma$ distributions are not normally distributed for their first sample levels of n=100 and 300. This is deduced from Figure 63 because a normal distribution will have no skew or asymmetry to it and these respective distributions are definitely positively skewed. Due to all of the outliers shown outside of the upper inner fence for both methods even after the 10,000 sample level it would be safer to accept the skew shown rather than expect it to disappear as these distributions would be more accurately captured. In a sense, this partially agrees with the Wackerly et al. (1996) statement that the probability distribution of

the standard deviation estimator is positively skewed for small sample sizes, but approximately normal for large sizes (n>25). For the sake of completion, the MC standard deviation distribution skew is not visible at n=30,000 and above, and the LHS standard deviation distribution skew is taken to be gone at n=100,000 and above. These are the levels where they are accepted as normal. This final skew/normality statement is made based on observing both a mean-median match up and no outliers for the respective distribution.

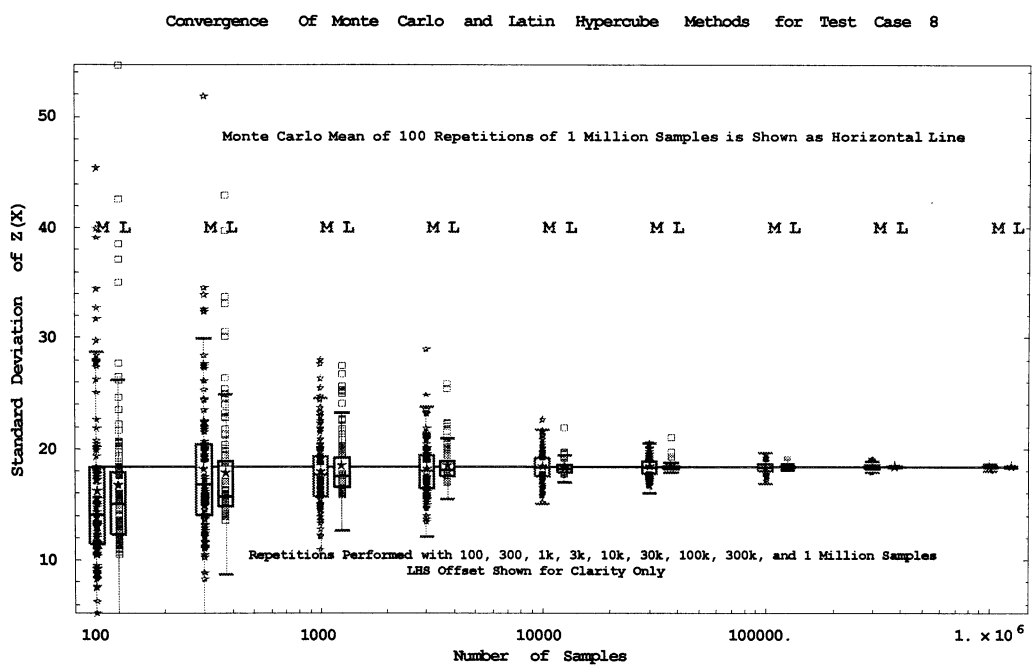


Figure 63 Distributions of standard deviations for test case 8 response using MC and LHS

The distributions of the standard deviation estimator using both MC and LHS are initially biased and eventually reach an unbiased state as number of samples used for each standard deviation estimate increases, as seen from Figure 63. The standard deviation distribution for either method has a non-zero bias of about 2 response units when n=100 samples are used to calculate each standard deviation. Numerically, this is a small bias; however, this is a 10% error with respect to the actual standard deviation. Recall that these distributions are estimates of true

standard deviation distributions and they could be better approximated if more repetitions were performed. The biases are considered nil when $n \geq 300$ samples are used.

The standard errors of the standard deviation estimator distributions shown in Figure 63 decrease as the number of samples, or response evaluations, used to calculate each standard deviation of the responses increases. The LHS standard deviation distribution has a lower standard error than the MC distribution for all number of samples shown in Figure 63.

Confidence that a single estimate, even if it is not close to the true value, will be close enough to be within an acceptable error limit when the estimate is made using a specific number of response evaluations is important. Confidence statements can be extracted from Figure 63 using the middle 50% of the box plots for each distribution. When MC-3,000 will be used to form the coordinates needed to calculate 3,000 response values and, after that, the standard deviation of those responses, it is 50% likely that the single standard deviation estimate calculated will be within 5.5% of the target parameter or the true standard deviation. This is about one response unit on either side of the true standard deviation. Using 3,000 LHS samples to calculate a single standard deviation of the test case 8 response, it is 50% likely that the single standard deviation estimate calculated will be within 3% of the target parameter or the true standard deviation. This is about 0.55 units on either side of the target. Therefore, for this test case, at the n=3,000 and 50% effort and confidence levels, respectively, it was found that LHS had the lower error of 3% from the true standard deviation than the MC error of 5.5%. The same effort and confidence were used and LHS had a lower error than MC.

The COV of the distributions of Figure 63 are shown in Figure 64. The COV is a measure of the variation of repeated standard deviation estimates about the true standard deviation for a specific number of samples and method, or rather a specific standard deviation

distribution, only because these distributions are essentially unbiased when n=300 and above. The horizontal line at the COV value of 0.01 (1%) is shown to emphasize the difference in effort required to obtain the same variation about the mean of the standard deviation distribution, considered the true standard deviation, for both methods.

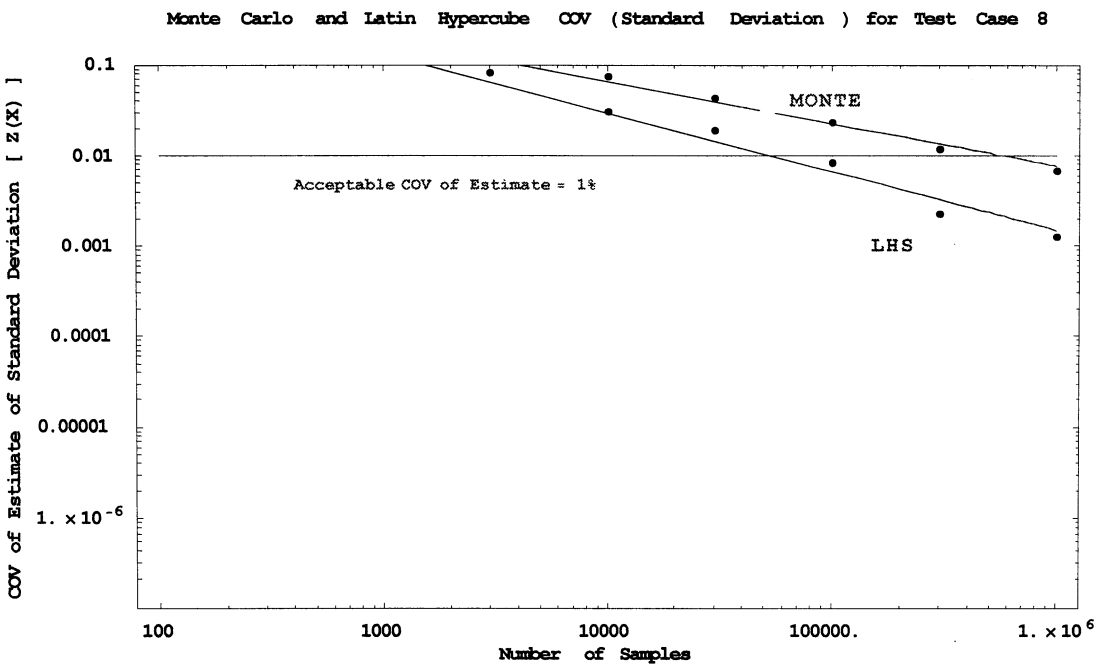


Figure 64 COV of standard deviation distributions for test case 8 response using MC and LHS

From Figure 64 one can see that the standard deviation estimator could have been shown to reach a COV level of 1% using MC with n=600,000 samples for each standard deviation estimate, while the LHS method needed about 60,000 samples to converge to the same level. Both distributions are normal and unbiased at those respective sample levels; therefore, confidence statements are then straightforward because for a normal distribution, 99.73% of the data lies within $3\sigma_{\theta}$ of the mean. It can then be stated with 99.7% confidence that a single standard deviation estimate for the response of test case 8 will be within ±3% of the true standard deviation using MC-600,000. In comparison, there is a 99.7% chance that the same estimate will

be within ±3% of the true standard deviation using LHS-60,000. Any value within 0.55 units from the true standard deviation of the test case 8 response will be within the estimation error of ±3%. The LHS method will estimate the standard deviation of the test case 8 response with equal confidence and error, but with less effort, or numerical calculations than the MC method. These statements are based on the best-fit line for the data shown in Figure 64.

The 99th percentile of the test case 8 response was estimated many times. The percentile estimator is a function of random variables and is therefore random, has a certain distribution, and will vary with the number of response evaluations used to estimate the percentile, n, and the method used to obtain the coordinate sets, MC or LHS. This variation is portrayed in Figure 65.

Both methods have a 99th percentile distribution that is positively skewed, and hence, non-normal, when 100 samples were used to capture the respective distribution. At and above the n=300 sample level, the MC 99th percentile distributions are approximately unskewed and normal based on using 100 repetitions to capture the distributions. Normally distributed random variables will exhibit symmetry about the median value of the distribution, which is equal to the mean and the mode (most probable). Also, a skewed distribution will tend to be non-normal and vice-versa. The LHS 99th percentile distributions are show slight skew when n=300, but the skew is essentially zero at and above the n=1,000 sample levels, where the distributions can be considered normal.

The distribution of the 99th percentile estimator for both MC and LHS are biased with respect to the exact percentile value for the n=100 sample level shown in Figure 65. The bias for both methods can be considered negligible at and after the n=300 sample levels. Recall that a single percentile estimation using a few amount of samples will therefore be closer to the mean and mode, which is are lower values than large percentiles like a 99th percentile and a higher

values for small percentiles like a 1.0 percentile. It is therefore likely that a single estimation of a 99th percentile will be less than the true value of this percentile and multiple values of this estimate will be centered around a lower mean value. This explains why the 99th percentile distributions shown in Figure 65 are negatively biased at the first sample level. So long as a sufficient number of response evaluations are performed, the 99th percentile distribution can be expected to be unbiased.

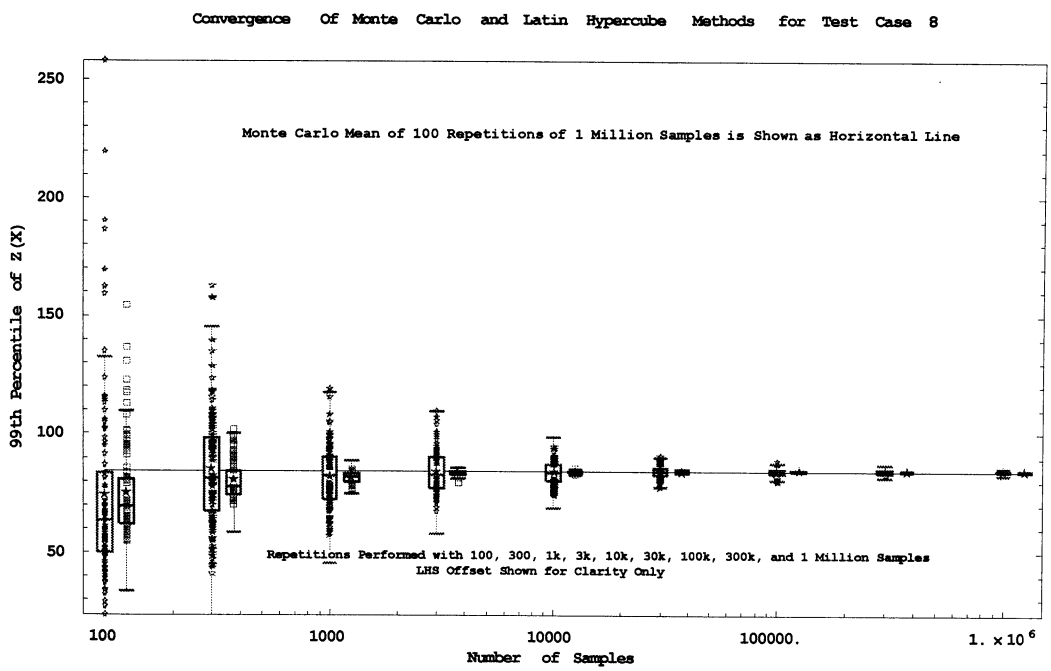


Figure 65 Distributions of 99th percentile for test case 8 response using MC and LHS

The standard error of the 99th percentile estimator distributions shown in Figure 65 decrease as the number of samples, or response evaluations, used to calculate each percentile increases, so calculating the percentile a number of times will result in values that are closer to each other when the amount of response values used to calculate each percentile is large. The LHS distributions have visibly lower standard errors than the MC distributions for all number of samples shown in Figure 65.

Confidence that a single estimate made using a specific number of response evaluations will be close enough to the true value is important. Approximate confidence statements can be made from Figure 65 using the middle 50% of the box plots for each distribution. If MC-3,000 were used to estimate the 99th percentile of the test case 8 response, it will be 50% likely that the single estimate calculated will be within 6.8% of the target parameter or the true 99th percentile. This error is about 6 units on either side of the target. Using LHS-3,000 to estimate the 99th percentile of the test case 8 response, it is 50% likely that the single estimate calculated will be within 0.5% of the target. This is about 0.4 units on either side of the target. Therefore, for this test case, at the n=3,000 and 50% effort and confidence levels, respectively, it was found that LHS had the lower error of 0.5% from the true 99th percentile than the MC error of 6.8%. The same effort and confidence were used and LHS had a lower error than MC.

The COV of the acquired 99th percentile distributions for test case 8 are shown in Figure 66. The COV can be considered to be a measure of the variation of repeated percentile estimates about the target parameter once the respective 99th percentile distribution can be considered unbiased. These distributions under consideration are unbiased at or above the n=300 sample, or effort levels for both methods as discussed when Figure 65 was discussed. The horizontal line at the COV value of 0.005 (0.5%) is shown to emphasize the difference in effort required to obtain the same variation about the mean of the 99th percentile distribution, considered to be the true 99th percentile, for both methods on or after the sample level of 300.

From Figure 66 it is evident that the 99th percentile distribution would have been observed to reach a COV level of 0.5% using MC-1,000,000 for each percentile estimate, while the LHS method needed about 20,000 samples to converge to the same level. Both distributions can be considered to be normal and unbiased at those respective sample levels. Therefore, it can

be stated with 99.7% confidence that a single 99th percentile estimate for the response of test case 8 will be within ±1.5% of the target using MC-1,000,000. In comparison, there is a 99.7% chance that the same estimate will be within ±1.5% of the true 99th percentile using LHS-20,000. The range of 1.5% from the true 99th percentile is any value within 1.26 units from the target. The LHS method required 980,000 less samples than the MC method to reach the same level of confidence and error. Therefore, it is better to estimate the 99th percentile for the test case 8 response using the LHS method, because it estimates this percentile with equal confidence and associated error, but with less effort, or numerical calculations than the MC method. These statements are based on the best-fit line for the data shown in Figure 66.

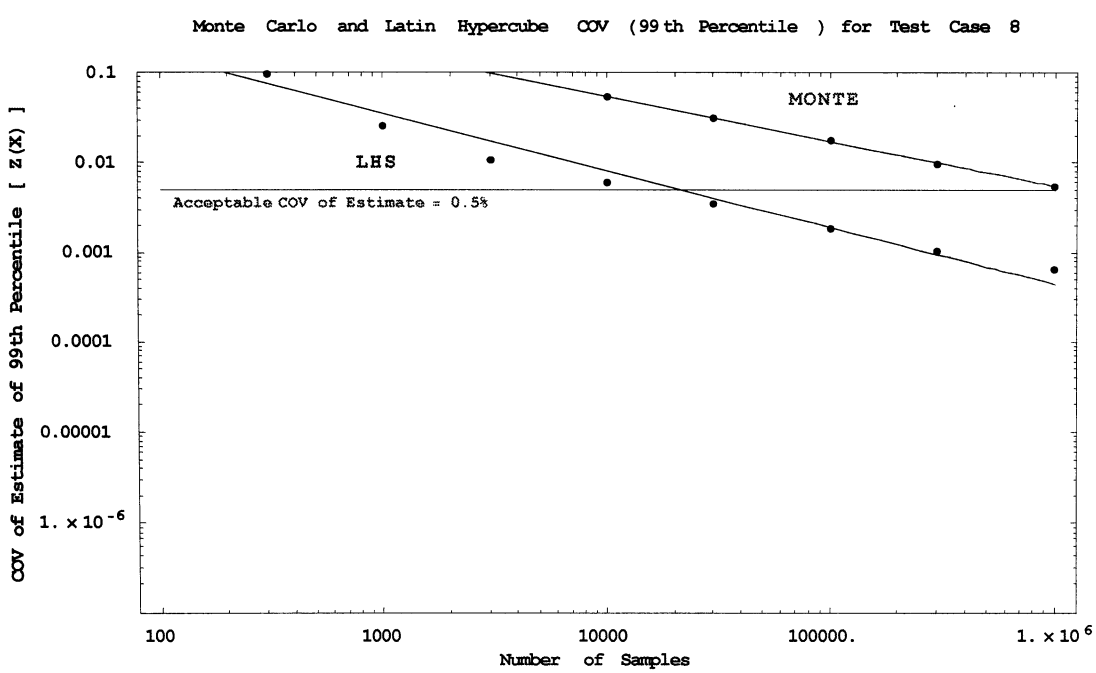


Figure 66 COV of 99th percentile distributions for test case 8 response using MC and LHS

Results

Three parameters of the density of four responses were repeatedly estimated using MC and LHS. This was done to study the properties of the distributions of the respective density parameters. The most important property of such distributions would be the sample level and method that defines the estimator, a confidence level, and the resulting error in estimation. For each parameter estimated, two standpoints were taken in making confidence statements. One was to set the confidence to 50%, and the sample level equal for both methods, and observe the error in estimation for both methods. The other way was to set the confidence to 99.7% and the error equal for both methods and to compare the effort required to obtain this characteristic of the estimator.

MC and LHS were used to estimate the mean of several responses. A summary of these results is shown in Table 9. For test case 1, it was found that when 1,000 MC samples were used to estimate the mean, 50% of the estimates were within 0.50% (95 units) from the true mean. Using LHS-1,000, 50% of the estimates were within 0.20% (33 units) from the true mean. For test case 4, MC-1,000 had a 50% confidence interval of 2.2% (48,000 units) from the true mean, while LHS-1,000 had a 50% confidence interval of 0.003% (65 units) from the mean. It was found that for test case 6, when 1,000 MC samples were used to estimate the mean, 50% of the estimates were within 0.08% (17 units) from the true mean. Using LHS-1,000, 50% of the estimates were within 0.0009% (0.2 units) from the true mean. For test case 8, MC-1,000 had a 50% confidence interval of 4.2% (0.34 units) from the true mean, while LHS-1,000 had a 50% confidence interval of 1.0% (0.08 units) from the mean. Therefore, LHS had a lower error in mean estimation than MC at the 1,000th sample level and 50% confidence interval for all of the four test cases studied.

Table 9 Estimation errors of the mean at the 50% confidence level using MC and LHS

Test Case	*Number of Samples*	*Confidence Level*	*Estimation Error*		
			Method	*%*	*units*
1	1,000	50%	MC	0.50	95
			LHS	0.20	33
4	1,000	50%	MC	2.2	48,000
			LHS	0.003	65
6	1,000	50%	MC	0.08	17
			LHS	0.0009	0.2
8	1,000	50%	MC	4.2	0.34
			LHS	1.0	0.08

On the other hand, since higher confidence is sometimes desired for each estimate of the mean, the confidence level and estimation error were set to specific values for each test case, and the number of response calculations necessary to obtain that confidence and error using MC and LHS was compared. These results are summarized in Table 10. For the test case 1 response, MC-10,000 was required to be 99.7% confident that a single mean estimate will be within 1.5% (260 units) from the true mean. LHS 500 could be used to obtain the same desired single mean estimate confidence. Over 399,000 more response calculations were necessary to be 99.7% confident that a single mean estimate of the test case 4 response will be within 0.6% (13,300 units) from the true mean when using MC instead of LHS. For test case 6, it was observed that MC-700,000 was necessary to be 99.7% confident that a single mean estimate will be within 0.021% (4.6 units) from the true mean. The LHS method required only 100 samples to be equally confident in the same type of estimate. In addition, LHS required only 4,000 response evaluations to be 99.7% confident that a single test case 8 mean estimate will be within 1.5% (0.12 units) from the true mean. MC required about 196,000 more response evaluations in order

to be equally confident in the same type of estimation. It can be stated that LHS estimates the mean of the studied responses with fewer calculations necessary to obtain the same confidence and error than MC.

Table 10 Computations for 99.7% confidence in mean estimate using MC and LHS

Test Case	*Number of Samples*		*Confidence Level*	*Estimation Error*	
				%	*units*
1	MC	10,000	99.7%	1.5	260
	LHS	500			
4	MC	400,000	99.7%	0.6	13,300
	LHS	100			
6	MC	700,000	99.7%	0.021	4.6
	LHS	100			
8	MC	200,000	99.7%	1.5	0.12
	LHS	4,000			

MC and LHS were used to estimate the standard deviation of several responses and the convergence properties of confident estimation studied. A partial summary of the results is shown in Table 11. For test case 1, at the n=300 and 50% sample and confidence levels, respectively, it was found that LHS had the lower error of 3% (265 units) from the true standard deviation than the MC error of 4.85% (435 units). Furthermore, for test case 4, at the n=300 and 50% sample and confidence levels, respectively, it was found that LHS had the lower error of 0.13% (3,500 units) from the true standard deviation than the MC error of 2.25% (62,000 units). When estimating the standard deviation of the test case 6 response, it was found that at the n=300 and 50% effort and confidence levels, respectively, LHS had the lower error of 0.5% (6 units) from the true standard deviation than the MC error of 1.4 % (17 units). LHS had the lower error of 3% (0.55 units) from the true standard deviation than the MC error of 5.5% (1 unit) when they were used to estimate the standard deviation of the test case 8 response at the 50% confidence level, and sample level of 3,000. The LHS method therefore had a lower standard deviation

estimation error than MC when 50% confidence was to be placed in the estimations and at the levels shown in Table 11.

Table 11 Estimation errors of the standard deviation at the 50% confidence level using MCand LHS

Test Case	*Number of Samples*	*Confidence Level*	*Estimation Error*		
			Method	*%*	*units*
1	300	50%	MC	4.85	435
			LHS	3	265
4	300	50%	MC	2.25	62,000
			LHS	0.13	3,500
6	300	50%	MC	1.4	17
			LHS	0.5	6
8	3,000	50%	MC	5.5	1
			LHS	3	0.55

High confidence standard deviation estimation properties of MC and LHS are equally important, and some of these characteristics are shown in Table 12. As shown in Table 12, the confidence level and estimation error were set to specific values for each test case, and the number of response calculations necessary to obtain that confidence and error using MC and LHS was compared. For the test case 1 response, MC-50,000 was required to be 99.7% confident that a single standard deviation estimate will be within 1.5% (134 units) from the true standard deviation. Using LHS-30,000 the same desired single standard deviation estimate confidence can be obtained. Over 149,000 more response calculations were necessary to be 99.7% confident that a single standard deviation estimate of the test case 4 response will be within 0.6% (16,500 units) from the true standard deviation when using MC instead of LHS. For test case 6, it was observed that MC-8,000 was necessary to be 99.7% confident that a single standard deviation estimate will be within 1.5% (18.4 units) from the true standard deviation. The LHS method required only 1,000 samples to be equally confident in the same type of estimate. Furthermore, LHS required 60,000 response evaluations to be 99.7% confident that a

single test case 8 standard deviation estimate will be within 3.0% (0.55 units) from the true standard deviation. It was necessary for 600,000 MC calculations to be made in order to obtain the same type of confidence and error. It can therefore be stated that LHS will estimate the standard deviation of the studied responses with fewer calculations necessary to obtain the same confidence and error than MC.

Table 12 Computations for 99.7% confidence in standard deviation estimate using MC and LHS

Test Case	*Number of Samples*		*Confidence Level*	*Estimation Error*	
				%	*units*
1	MC	50,000	99.7%	1.5	134
	LHS	30,000			
4	MC	150,000	99.7%	0.6	16,500
	LHS	100			
6	MC	8,000	99.7%	1.5	18.4
	LHS	1,000			
8	MC	600,000	99.7%	3.0	0.55
	LHS	60,000			

The 99th percentile of a response is also a density parameter. It represents an important cutoff point in the range of all possible response values. The response will be observed to be under this value 99% of the time, and over this value 1% of the time. It was also estimated many times using MC and LHS over the same four test cases. The estimation errors associated with the 99th percentile of the test cases studied is shown in Table 13. For test case 1, at the n=10,000 and 50% effort and confidence levels, respectively, it was found that LHS had the lower error of 0.7% (340 units) from the true 99th percentile than the MC error of 1% (470 units). The 99th percentile of the second test case was estimated and it was found that LHS had the lower error of 0.04% (3,800 units) from the true 99th percentile than the MC error of 0.34% (33,000 units) at

the 50% confidence level and when 3,000 samples were used to estimate each percentile necessary for this type of study. For the third test case, at the n=3,000 and 50% effort and confidence levels, respectively, it was found that LHS had the lower 99th percentile error of 0.04% (9 units) than the MC error of 0.07% (16 units). The fourth test case response 99th percentile was estimated and it was found that at the n=3,000 and 50% effort and confidence levels, respectively, LHS had the lower estimation error of 0.5% (0.4 units) than the MC error of 6.8% (6 units). The LHS method therefore had a lower 99th percentile estimation error than MC when 50% confidence was to be placed in the estimations at the levels shown in Table 13.

Table 13 Estimation errors of the 99th percentile at the 50% confidence level using MC and LHS

Test Case	*Number of Samples*	*Confidence Level*	*Estimation Error*		
			Method	*%*	*units*
1	10,000	50%	MC	1.0	470
			LHS	0.7	340
4	3,000	50%	MC	0.34	33,000
			LHS	0.04	3,800
6	3,000	50%	MC	0.07	16
			LHS	0.04	9
8	3,000	50%	MC	6.8	6
			LHS	0.5	0.4

High confidence placed in 99th percentile estimates is important. Some confidence properties of using MC and LHS in estimating the 99th percentiles of the responses studies are shown in Table 14. The confidence levels and estimation errors were set to specific values for each test case, and the number of response calculations necessary to obtain that confidence and error using MC and LHS was compared. For the test case 1 response, MC-100,000 was required to be 99.7% confident that a single 99th percentile estimate will be within 1.5% (700 units) from the true 99th percentile. Using LHS-80,000, the same desired single 99th percentile estimate

confidence can be obtained. For the test case 4 response, it is necessary to make 90,000 more response calculations to be 99.7% confident that a single 99th percentile estimate will be within 0.3% (29,000 units) from the true 99th percentile when using MC instead of LHS. For test case 6, it was observed that MC-800 was necessary to be 99.7% confident that a single 99th percentile estimate will be within 0.6% (145.5 units) from the target. The LHS method required only 600 samples to be equally confident in the same type of estimate. Furthermore, LHS required 20,000 response evaluations to be 99.7% confident that a single test case 8 99th percentile estimate will be within 1.5% (1.26 units) from the true 99th percentile. In comparison, MC-1,000,000 was necessary to obtain the same confidence and estimation error. It can therefore be stated that LHS will estimate the 99th percentile of the studied responses with fewer calculations necessary to obtain the same confidence and error than MC.

Table 14 Computations for 99.7% confidence in 99th percentile estimate using MC and LHS

Test Case	*Number of Samples*		*Confidence Level*	*Estimation Error*	
				%	*units*
1	MC	100,000	99.7%	1.5	700
	LHS	80,000			
4	MC	100,000	99.7%	0.3	29,000
	LHS	1,000			
6	MC	800	99.7%	0.6	145.5
	LHS	600			
8	MC	10^6	99.7%	1.5	1.26
	LHS	20,000			

From these results, it can therefore be summarized that the LHS method had a lower estimation error than MC when they were used to estimate the mean, standard deviation, and the 99th percentile of the four different stochastic responses studied. In addition, the LHS method required fewer response calculations than MC in order to be highly confident in estimating the same density parameter.

3.6 CONCLUSIONS

Summary

The purpose of this work is to enhance the NESSUS program with the capability to perform LHS sampling, and to compare the efficiency of LHS to that of MC, which is an existing method within NESSUS. The density parameters estimated were the mean, standard deviation, and the 99th percentile of the response of four different test cases put forth by the Society of Automotive Engineers for the purpose of comparing different probabilistic methods.

Conclusions

LHS Enhancement

The NESSUS LHS enhancement involved the addition of seven Fortran 90 files to the existing NESSUS files for the purpose of performing LHS sampling. These files are named lhs_main.f90, lhs_xsample.f90, calc_statistics.f90, corr_control.f90, lhs_calc.f90, write_files.f90, error_files.f90. These files contain the following subroutines and functions: lhs_main, lhs_xsample, raniset, iranu, calc_stats, vector_rank, vector_stats, corr_control, lhs_calc, write_files, and error_files. The files have been successfully integrated with the NESSUS program so that it now has the added capability of performing LHS sampling.

When the Monte Carlo program actions were studied it was discovered that some things might be improved. That is why some of the features of the NESSUS LHS actions and output are unique to that method. These features are highlighted in the following list.

Characteristics of the LHS enhancement

- ✓ Uses new input file format
- ✓ Early LHS callout from main NESSUS program
- ✓ Uses existing NESSUS subroutines
    - o opening files
    - o random number generator
    - o sorting vectors
    - o pdf, cdf, and their inverses
    - o evaluating response
- ✓ Uses derived types in existing module
- ✓ Echo of new input file format
- ✓ Does not obtain coefficients for linear expansion of response
- ✓ Uses one file for writing output and error messages to output files
    - o globally define desired message and file units
- ✓ All output in scientific notation – neater output
- ✓ Adjusts correlation between dependent variables
- ✓ Two types of correlation printed in output files
    - o Lower triangle of Pearson's and Spearman rank
    - o Desired correlation printed as upper triangle above Spearman

The LHS subroutines are dependent on the new input file format used by NESSUS. Also, it was important for the main LHS file to be called early in the main part of the NESSUS program because it is a different method, and each method had its own unique actions to take. It is especially important for the LHS callout to come early so that no files are unnecessarily opened or written to and so that no variables are changed or declared.

The LHS enhancement it not a stand-alone module, it uses the NESSUS subroutines that open files, generate random numbers, calculate the pdf, cdf, and their inverses, sort arrays, and evaluate the response. This is good because testing of the individual subroutines within

NESSUS, like one that calculates the CDF values, will not need to be performed twice. The LHS subroutines make use of the derived types already defined in a NESSUS module, created for the purpose of removing programming techniques such as common blocks. The current LHS routines will echo the new input file format. A MC analysis will produce an echo of the old input file format, even when a new input file type is used to run the program.

One thing that the LHS module will not perform, while the MC analysis will, is the calculation of the coefficients of linear expansion of the response about the mean of all n random variables. This is an unnecessary set of n+1 calculations that is, most importantly, time consuming.

If there is one thing that separates the actions of the LHS subroutines *that do calculations* from the original NESSUS code is that they *do not ever* write output of error messages to any files. Two things were discovered about write statements while studying the NESSUS code. One is that similar write statements are found in different files. This code repetition is somewhat acceptable, but, again, if changes are made to the output, then all write statements need to be changed. The second thing is that the write statements get in the way of the actual calculations. Sorting through another's program is difficult enough and can be made a little easier if we only see the calculations necessary in the analysis being performed. The LHS subroutines *that do calculations* accomplish this no write feature by assigning a short character string to a global variable along with a vector of integer variables that are file unit numbers. Then it simply calls the *write_files()* or *error_files()* subroutines *with no arguments*, which write output and error messages, respectively, to the appropriate files. The lines of code that are seen in the LHS subroutines that do this are shown below.

```
mssg%description="lhs_header"
mssg%files(1:4)=(/file_lhs_x_rdm,file_lhs_p_rdm,file_lhs_x_corr,file_lhs_p_corr/)
call write_files()
```

The files of code just shown are consistent in form and are easily seen when sorting through the LHS subroutines that do calculations. Also, all of the output written to the appropriate files is in scientific notation. This produces a neater output because even for different number magnitudes and signs, all of the decimal numbers will be aligned.

Fortunately, the LHS subroutines will also adjust the correlation to what is desired and entered by an analyst. Also, the lower half of the Pearson's and Spearman rank correlation matrices are written to two of the LHS output files. The upper half of the desired correlation matrix is written above the Spearman rank correlation matrices in the same file because in the process of rearranging the samples it was assumed that the desired correlation is the Spearman's rank correlation.

MC and LHS Comparison

Three parameters of the probability density function of four responses were repeatedly estimated using MC and LHS to study distributions of the respective density parameters. The most important property of such distributions would be the sample level and method that defines specific distribution of multiple density parameter estimates, a confidence level (or probability level), and the resulting error in estimation.

For each parameter estimated, two standpoints were taken in making confidence statements. One was to set the confidence to 50%, and the sample level equal for both methods, and observe the error in estimation for both methods. The other way was to set the confidence to 99.7% and the error equal for both methods and to compare the effort required to obtain this characteristic of the estimator.

Mean Estimation

Using 1,000 samples, it is 50% likely that a single estimate of the mean of any of the test case responses using LHS will have a lower estimation error associated with it compared to MC. The best LHS performance for this type of estimate was for test case 4, the nonlinear response with non-normal variables, where the LHS estimation error was 0.003% from the assumed true mean and the MC error was 2.2%. The most comparable performance of MC was for test case 8, the nonlinear response with standard normal variables, where the LHS and MC error was 1.0% and 4.2%, respectively.

Furthermore, it was found that in order to obtain 99.7% confidence for a mean estimation to be within a specific error, LHS sampling required fewer calculations than MC. The greatest LHS performance was for the test case 6 response, the maximum radial stress of a rotating disk. For the mean of that response and for the 99.7% confidence level, LHS required 100 samples to be within 0.021% of the true mean, while MC required 700,000 samples for the same estimation error. The most comparable performance of MC was found for the test case 1 response, where MC required 10,000 samples to be within 1.5% of the true mean, while LHS required only 500.

Standard Deviation Estimation

In estimating the standard deviation of the various responses studied, it is 50% likely that a single estimate using LHS will have a lower estimation error associated with it compared to MC, when they are compared using the same amount of samples. The finest LHS performance was for test case 4, the nonlinear response with non-normal variables. For this test case, using 300 LHS samples, the estimation error was found to be 0.13% from the assumed true standard deviation and the 300-sample MC error was 2.25%. MC did best for test case 1, the Paris Law stage to crack propagation response, where the LHS and MC error were 3% and 4.85%,

respectively, and both used 3,000 samples to achieve this 50% confidence level.

In addition, it was found that in order to be 99.7% confidence that a single standard deviation estimation to be within a specific error, LHS sampling required fewer calculations than MC. LHS did best for the test case 8 response, the nonlinear response with standard normal variables. To be 99.7% confident in a single standard deviation estimate of that response, LHS required 60,000 samples to be within 3.0% of the true standard deviation, while MC required 600,000 samples for the same estimation error. MC did best for the test case 6 response, the maximum radial stress of a rotating disk, where MC required 8,000 samples to be within 1.5% of the true standard deviation, while LHS required only 1,000.

99th Percentile Estimation

In estimating the 99th percentile of the various responses studied, it is 50% likely that a single estimate using LHS will have a lower estimation error associated with it compared to MC, when they are compared using the same amount of samples. The most excellent LHS performance was for test case 8, the nonlinear response with standard normal variables. For this test case, using 3,000 LHS samples, the estimation error was found to be 0.5% from the assumed true 99th percentile and the 3,000-sample MC error was 6.8%. MC did best for test case 1, the Paris Law stage to crack propagation response, where the LHS and MC error were 0.7% and 1.0%, respectively, and both used 10,000 samples to achieve this 50% confidence level.

It was also found that in order to obtain 99.7% confidence for a 99th percentile estimation to be within a specific error, LHS sampling required fewer calculations than MC. The finest LHS performance was for the test case 8 response, the nonlinear response with standard normal variables. The 99th percentile of that response can be estimated at the 99.7% confidence level using 20,000 LHS samples, with an associated estimation error of 1.5% from the true parameter,

while MC required 1,000,000 samples for the same estimation error. The most comparable performance of MC was found for the test case 6 response, the maximum radial stress of a rotating disk, where MC required 800 samples to be within 0.6% of the true mean, while LHS required only 600.

Generally speaking, a sample should be selected so that a specific quantity of information is obtained at a minimal cost. For the density parameters estimated and for the test cases studied LHS sampling has proven to be an efficient sampling method. Furthermore, it has been successfully added to the existing methods in the NESSUS probabilistic finite element program.

4.0 ACCOMPLISHMENTS

The goal of this NASA Partnership Award was to advance innovative research and education objectives in theoretical and computational probabilistic structural analysis, reliability, and life prediction methods for improved aerospace and aircraft propulsion system components. This grant resulted in significant accomplishments in research and education, and the enhancement of UTSA's engineering research environment. It allowed six UTSA Mechanical Engineering students; Mr. Cody Godines, Mr. Henock Perez, Mr. Edgar Herrera, Mr. Luis Rangel, Mr. Santiago Navarro and Mr. Ronald Magharing to work directly with the principal investigator, Dr. Randall Manteufel, providing them with a unique research experience that, without this grant, would not have been possible.

4.1 Accomplishments: Education

Graduate students and upper-division undergraduate students were introduced to probabilistic structural analysis methods through two UTSA courses. Two minority graduate student and four minority undergraduate students were supported by this Partnership Award and had the opportunity to work directly with the Principal Investigator. The NESSUS Student User's Manual was revised to include two additional example problems. Solutions for all example problems were added as well. This manual will provide guidance in using NESSUS for future courses and help insure the continuation of probabilistic structural analysis courses at UTSA.

4.2 Accomplishments: Research

Probabilistic structural analysis, reliability, and life prediction methods are supported or facilitated by NESSUS, a stochastic finite element program developed by NASA Lewis Research Center (LeRC) with Southwest Research Institute (SwRI). Mr. Cody Godines, was supported with this Partnership Award throughout his graduate studies. Mr. Godines has studied different probabilistic methods for the purpose of improving the capabilities of NESSUS. This May 2000, he finished his thesis. As part of his thesis work, he enhanced NESSUS with the capability of performing Latin Hypercube Sampling. Once this objective was finished he compared LHS with Monte Carlo in their ability to efficiently estimate parameters of the probability density function of several responses. It was found that LHS performed better for all of the density parameters estimated and for all test cases studied. Dr. Manteufel has worked on probabilistic sampling schemes and published a paper entitled "Evaluating the Convergence of Latin Hypercube Sampling" AIAA-2000-1636 which was presented in the Non-Deterministic Approaches Forum at the 41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, in April 3-6, 2000 [Manteufel, 2000]. Former graduate student, Mark Jurena, supported for his Master's degree on a 1998 Partnership Award, had a paper on his thesis work accepted for presentation at the Probabilistic Mechanics Conference in July, 2000 [Jurena, Manteufel, and Bast, 2000].

4.3 Student Achievements

Mr. Cody Godines, who's thesis topic directly relates to research objectives of this Partnership Award, has presented his thesis results at the 2001 ASME Region X Graduate Student Technical Conference in Kingsville, Texas. Mr. Godines and a fellow colleague, Mr. Rodney Harris, attended the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference in April, 2002, where they presented a paper entitled, "Use Of Probabilistic Methods In Design Of Fluidic Systems". This paper is shown in Appendix III. Mr. Godines also made a trip to Cleveland, Ohio to present his work at the Ohio Aerospace Institute Conference in April 2002. This conference was sponsored by NASA Glenn Research Center and gave students an opportunity to practice their presentation skills. Mr. Godines will also present his thesis work at the 44d AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference in April, 2003. He worked from the summer of 2000 until early spring of 2001 at SwRI as an intern in the Probabilistic Mechanics and Reliability Section of SwRI performing probabilistic fracture mechanics and NESSUS verification. Under the guidance of Dr. Manteufel, he has successfully completed his Master's Degree in Mechanical Engineering. The second graduate student supported on this grant is now working with a local engineering company and is making plans to come back as a full time student to obtain his M.S.M.E (Edgar Herrera). Two students supported on this grant have graduated on December 2000 with their Bachelor's Degrees in Mechanical Engineering (Luis Rangel, Santiago Navarro). Another is on schedule to graduate by December 2003 (Henock Perez).

5.0 REFERENCES

Ang, A. H.-S., and Tang, W. H., "Probability Concepts in Engineering Planning and Design. Volume II: Decision, Risk, and Reliability," John Wiley & Sons, Inc., New York, 1984.

Ayyub, B. M. and Lai, K-L, "Selective Sampling in Simulation-Based Reliability Assessment," *International Journal of Pressure Vessels and Piping,* v46, 1991.

Breitung, K., "Asymptotic Approximation for Multinormal Integrals," *Journal of Engineering Mechanics,* Vol. 110, pp.357-366, 1984.

Chamis, C.C., *Probabilistic Structural Analysis Methods for Space Propulsion System Components*, NASA TM-88861, June, 1986.

Chamis, C. C., "Probabilistic Simulation of the Human Factor in Structural Reliability," *34th AIAA/ ASME/ ASCE/ AHS/ ASC Structures, Structural Dynamics and Materials Conference,* 19-22 April 1993.

Grandhi, R. V. and Hopkins, D. A., "Efficient Evaluation and Optimization of Structural Reliability," NASA Conference Publication, *Conference on Physics & Process Modeling (PPM) and other Propulsion R&T.* Part 2 (of 2), May 1 1997, Cleveland, OH, USA, p 25 Publisher: NASA, Cleveland, OH, USA, 1997.

Harbitz, A., "An Efficient Sampling Method for Probability of Failure Calculation," *Structural Safety,* Vol. 3, pp. 109-115, 1986.

Harris, R., Godines, C. R., Rangel, L., Herrera, E., and Manteufel, R. D., "Use of Probabilistic Methods in Designs of Fluidic Systems," *AIAA-2002-1270, 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Denver, CO. 2002.

Hasofer, A. M. and Lind, N. C., "Exact and Invariant Second Moment Code Format," *Journal of Engineering Mechanics Division,* ASCE, Vol. 100, pp.111-121, 1974.

Higgins, S. A., Hinwood, J. B., "Environmental Design Criteria for a TLP in Cyclonic Conditions," Proc. 1999 *9th International Offshore and Polar Engineering Conference,* 30 May – 4 June, 1999.

Hodge, B.K. and Taylor, Robert P., Analysis and Design of Energy Systems, 3rd Edition, Prentice Hall, New Jersey, 1999, pp. 401-405.

Iman, R. L. and Conover, W. J. "A Distribution-Free Approach to Inducing Rank Correlation Among Input Variables", *Joint Statistical Meetings,* Detroit, Michigan, 1981.

Jurena, M.T., Manteufel, R.D., Bast, C.C., "Structural Reliability Using NESSUS and a Material Strength Degradation Model," *Proceedings, 8th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability*, July, 2000.

Manteufel, R. D., "Distributed hypercube sampling algorithm", *AIAA-2001-1673, 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Seattle, WA. 2001.

Manteufel, R.D., "Evaluating the Convergence of Latin Hypercube Sampling," *Proceedings, 41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, April, 2000.

Manteufel, R. D., Ahola, M. P., "Probabilistic Performance Assessment of Borehole or Shaft with a Degraded Seal at an Unsaturated High-Level Waste Repository," *Radioactive Waste Management and Environmental Restoration*, Vol. 20, pp.213-235, 1997.

Millwater, H., Wu, Y.T., Tomg, T., Thacker, B., Riha, D., and Leung, C., "Recent Developments of the NESSUS Probabilistic Structural Analysis Computer Program," *Proceedings, 33rd AIAA/ASME/AHS/ASC Structures, Structural Dynamics, and Materials (SDM) Conference*, 1992.

Pai, S.S., *Preferred Reliability Practices-Design Considerations for Space Trusses*, PP-ED-1242, NASA Lewis Research Center, 1995.

Rackwitz, R., "Practical Probabilistic Approach to Design," *Bulletin 112*, Comite Euopean du *Beton*, Paris, France, 1976.

Ratkowsky, D. A., "Handbook of Nonlinear Regression Models", Marcel Dekker, Inc., New York, 1990.

Robinson, D. and Atcitty, C., "Comparison of Quasi- and Pseudo- Monte Carlo Sampling for Reliability and Uncertainty Analysis," *Sandia National Laboratories, NM and American Institute of Aeronautics and Astronautics*, 1999.

Shah, A., Nagpal, V. K., Chamis, C. C., "Probabilistic Analysis of Bladed Turbine Disks and the Effect of Mistuning," *31st AIAA/ ASME/ ASCE/ AHS/ ASC Structures, Structural Dynamics and Materials Conference,* 2-4 April 1990.

Shah, A. R., Murthy, P. L. N., Chamis, C. C., "Effect of Cyclic Thermo-Mechanical Load on Fatigue Reliability in Polymer Matrix Composites," 36th *AIAA/ ASME/ ASCE/ AHS/ ASC Structures, Structural Dynamics and Materials Conference* and *AIAA/ ASME Adaptive Structures Forum,* 10-13 April 1995.

Shigley, J. E., and Mischke, C. R., "Mechanical Engineering Design," 5th ed., McGraw-Hill, NY, 1989.

Southwest Research Institute, *NESSUS User's Manual*, 1996.

Southwest Research Institute and Rocketdyne, *Probabilistic Structural Analysis Methods (PSAM) for Selection of Space Propulsion System Components*, Final Report, NASA Contract NAS3-24389, NASA Lewis Research Center, Cleveland, Ohio, 1995.

Southwest Research Institute, "NESSUS Documentation", vol. I, section FPI/THEORETICAL, 1995.

Thacker, B. H., Riha, D. S., Millwater, H. R., and Enright, M. P., "Errors and Uncertainties in Probabilistic Engineering Analysis," *AIAA Journal 2001-1239*, 2001.

Thacker, B. H., Wu, Y. T., Nicolella, D. P. and Anderson, R. C., "Probabilistic Injury Analysis of the Cervical Spine," 38th *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference,* Kissimmee, Florida, 7-10 April 1997.

Tvedt, L., "Distribution of Quadratic Forms in Normal Space-Application to Structural Reliability," *Journal of Engineering Mechanics,* Vol. 116, pp.1183-1197, 1990.

Wackerly, D. D.; Mendenhall, W. III; Scheaffer, R. L., "Mathematical Statistics with Applications," 5th Ed. , Duxbury Press, N.Y., 1996, p.147-8, 169, 322-3.

Wolfram, S., "The Mathematica Book," Wolfram Media, Inc. and Cambridge Press, 1996.

Wu, Y.-T. and Wirshing, P. H., "A New Algorithm for Structural Reliability Estimation," *Journal of Engineering Mechanics,* Vol. 113, pp. 1319-1336, 1987.

Wu, Y.-T., Torng, T. Y., Khalessi, M. R., "A New Iteration Procedure for Efficient Structural Reliability Analysis," Proceedings of *1st International Symposium on Uncertainty Modeling and Analysis (ISUMA),* Washington D.C., December 1990.

APPENDIX I

ME 5543 Probabilistic Engineering Design
Spring 2000

Time and Place:	TTH 5:30p-6:45p, EB1.04.06
Office and hours:	EB1.04.06, TH 6:45p-8:00p
Instructor:	Ben H. Thacker, PhD, PE, bthacker@swri.org, 522-3896
Asst. Instructor:	David S. Riha, driha@swri.org, 522-5221
	Callie Bast, cbast@voyager1.eng.utsa.edu, 458-5588

Course Objective

The objective of the course is to understand the effect of uncertainties in modeling, analysis, and design of physical systems. Fundamentals in probability and statistics will be covered followed by an introduction to probabilistic analysis and design methods. A final project is required where you will apply probabilistic analysis methods to the design of a component of your choice. The final project will involve an analytical and computer solution, presentation, and final report.

Course Outline

- Probability and Statistics
  - Descriptive Statistics
  - Probability Theory
  - Random Variables
  - Statistical Models
- Probabilistic Design
  - Limit State Function
  - Probability of Failure
  - Normal and Lognormal Format
- Probabilistic Analysis Methods
  - Monte Carlo Simulation
  - Response Surface Method
  - First-order Second Moment Theory
  - First-order Reliability Method
  - Advanced Methods
- Advanced Topics
  - Systems Reliability Formulation
  - Series and Parallel Systems

Grading

- 30% Homework
- 30% Mid-term exam
- 40% Final project

Important Dates

Jan 18:	First day of class
March 9:	Mid-term Exam
March 13-17:	Spring break, no classes
April 3-6:	No classes
May 4-5:	Study day, no classes

References

1. Hines, W.H. and Montgomery, D.C., Probability and Statistics in Engineering and Management Science, Wiley, 1990.
2. Benjamin, J. and Cornell, C.A., Probability, Statistics, and Decision for Civil Engineers, McGraw-Hill, 1970.
3. Ang, A-H.S. and Tang, W., Probability Concepts in Engineering Planning and Design, Vol. I: Basic Principals, Wiley, 1975.
4. Ang, A-H.S. and Tang, W., Probability Concepts in Engineering Planning and Design, Vol. II: Decision, Risk, and Reliability, Wiley, 1975.
5. Madsen, H.O., Krenk, S., and Lind, N.C., Methods of Structural Safety, Prentice-Hall, Inc., 1986.
6. Kapur, K.C. and Lamberson, L.R., Reliability in Engineering Design, Wiley, 1977.

Final Project

The goal of the final project is to apply probabilistic analysis methods to a practical problem. You may select the problem from your area of interest. The project needs to meet some basic requirements:

1. At least 6 random variables.
2. Combination of normal and non-normal random variables.
3. Analysis using Monte Carlo simulation and at least two other advanced methods learned in class.

Because of these requirements, computer solution will be required. A spreadsheet solution is acceptable; however, a computer program will most likely be required. You may use the language of your choice (Fortran, C, etc.)

Your final project will include a written report and a presentation to the class. The presentation will take the place of the final exam. The written report must include an introduction, problem formulation, problem statement, solution approach, discussion of results, summary, and computer listing. More detailed instructions will be handed out following the spring break.

ME 5543 Probabilistic Engineering Design, Spring 2000

Monday	Tuesday	Wednesday	Thursday	Friday
JANUARY 17 HOLIDAY	18 **First Day Class** Class Overview, Intro. to Descriptive Statistics	19	20 Descriptive Statistics	21
24	25 Probability Theory: Sets and Events	26	27 Probability Theory: Conditional & Total Prob.	28
31	**FEBRUARY 1** Random Variables: PDF, CDF, Joint PDF	2	3 Random Variables: Properties, Moments	4
7	8 Statistical Models: Discrete & Continuous Distributions	9	10 Statistical Models: Properties and Use	11
14	15 Distribution Fitting: Selecting a Model	16	17 Types of Uncertainties and Modeling Approaches	18
21	22 Probabilistic Design: Limit State Function	23	24 Probabilistic Design: Probability of Failure	25
28	29 Probabilistic Design: Normal Format	**MARCH 1**	2 Probabilistic Design: Lognormal Format	3
6	7 Review	8	9 Mid Term Exam	10
13	14 SPRING BREAK	15	16 SPRING BREAK	17
20	21 Intro to Probabilistic Analysis Methods: Monte Carlo Simulation	22	23 Response Surface Method. Begin Final Project	24
27	28 First-Order Second Moment Methods	29	30 Transformations, Most Probable Point Methods	31
APRIL 3	4 First-order Reliability Method	5	6 First-order Reliability Method, Advanced Mean Value Method	7
10	11 Advanced Mean Value Method	12	13 Probabilistic Sensitivity Factors, Robust Design Methodology	14
17	18 Importance Sampling	19	20 Multiple Failure Modes: Systems Reliability Formulation	21
24	25 Systems Reliability: Series and Parallel Systems	26	27 Computational Issues for Large Scale Structures	28

MAY 1	2 Wrap-up and Review	3	4 Study Day No Class	5
8	9 **FINAL** 5:00-7:45 Presentation of Final Project	10	11	12

APPENDIX II

Syllabus
ME 4723:
Reliability and Quality Control in Engineering Design
Summer 2000
SB 2.02.02

Instructor: David Riha (Southwest Research Institute)
Email: driha@swri.org phone: 522-5221

Teaching Assistants: Callie Bast and Mark Jurena

Office Hours: Tuesday and Thursday 8:00-9:00 PM and by appointment

Textbook: E.E. Lewis, Introduction to Reliability Engineering, 2nd Edition
Handouts will be provided for topics not covered in the book.

Course Grade:		
	Homework	25%
	Biweekly Quizzes	35%
	Final Design Project	40%

Work is due at the beginning (first ten minutes) of the class period one week after it is assigned. No late assignments will be accepted unless prior arrangements are made. Homework should be neat and written on one side of the paper. Assignments must be stapled and folded with the student's name and assignment number on the outside.

Course Description:

- Introduction to statistical methods in reliability and probabilistic engineering design
- Statistical quality control and inspection
- Life prediction and testing
- Design optimization

Course Organization:

Probability Theory (Chapters 1-3)
Reliability (Chapters 6-7, 9)
Reliability Testing and Data Analysis (Chapters 5, 8)
Probabilistic Design (Chapter 4 + other sources)

Prerequisites: Senior level standing in Engineering

Software: The NESSUS probabilistic analysis software will be used in this class

ABET Notebook:
Each student is required to maintain a notebook of all graded material. This notebook must be turned in with the final design report. Two of three notebooks will be retained for ABET accreditation review. All other notebooks will be available after final grades are posted.

PROBABILISTIC REDESIGN OF A HIGH PRESSURE VESSEL BY WAY OF REDUCING THE PROBABILITY OF YIELDING WHILE INCLUDING STRENGTH DEGREDATION EFFECTS

Cody Godines

University of Texas at San Antonio

6900 N. Loop 1604 West

San Antonio, Tx. 78249

Key Words: Probabilistic Design, Probabilistic Sensitivity, Pressure Vessel, Reliability, and Failure

ABSTRACT

It is becoming increasingly important to be able to quantify the reliability of engineering structures that have randomness in loading, material properties, and geometric properties. Probabilistic analysis provides a means to do this. Two well-known methods of probabilistic analysis are simple Monte Carlo and First Order Reliability Method. In order for probabilistic methods to be more widely accepted, they need to be proven to be more useful than conventional deterministic designs. This paper deals with the redesign of a high-pressure vessel. Strength degradation and fatigue effects were taken into account. A total of six design variables were stochastic. Using two probabilistic methods, the probability of failure of the system was reduced.

INTRODUCTION

Pressure vessels are closed structures that contain a fluid at under pressure and are used in a wide variety of situations in today's society. Self-Contained Underwater Breathing Apparatus (SCUBA) tanks, fire extinguishers, propane storage tanks are examples of the many uses of pressure vessels. This paper will discuss the analysis and redesign of a 100 cubic-foot, high-pressure steel SCUBA tank.

In analyzing a SCUBA tank it is ideal to ensure that failure will not occur in such a manner that hinders the performance of the system or endangers the safety of people. The Texas Department of Transportation (TDOT) has set the standards of the American Society of Mechanical Engineers (ASME) pressure vessel code to be met by all SCUBA tanks. Every five years, SCUBA tanks must be hydro-statically tested to 5/3 their working pressure. They fail if they permanently deform more than 10 percent of their original volume. The probability of failing this test for an existing SCUBA tank system will be determined by using a probabilistic method to analyze the tank and account for the uncertainty in the system and test procedure.

Probabilistic analysis is an important tool in the design and analysis of today's structures. It gives the engineer the ability to compute the reliability of the system without the expensive cost of laboratory simulation, whose data is of no use for new designs. Probabilistic methods also allow for the quantification of the uncertainties inherent in the structure as well as those involved in a measurement technique.

The probabilistic analysis of the SCUBA tank will be performed using QUEST, a Fortran 90 code that was developed as during a class taken during his quest for his Masters of Science in Mechanical Engineering degree. The code has two main methods

of probabilistic analysis. The First Order Reliability Method (FORM) was developed by many people but known to the engineering community as the Hasofer-Lind and Rackowitz-Feissler algorithm/method. This will be the main tool used in the analysis of the SCUBA tank. The results of the FORM solution will be checked using the simple Monte Carlo technique, which is the second method of the code.

The probabilistic analysis in this report will determine which input parameters most influence the response of the SCUBA tank and how to change the system uncertainty in order to reduce the probability of failure. A new design will be realized and a new probabilistic analysis will be performed in order to make sure that the probability of failure was reduced. Conclusions will then be drawn.

The System

The system analyzed was a 100 cubic foot, high-pressure steel SCUBA tank. Pressed Steel Tank Company in Milwaukee, WI manufactures it. Its design pressure is 3500 psi. The geometry of the tank will be simplified to that shown in Figure 1.

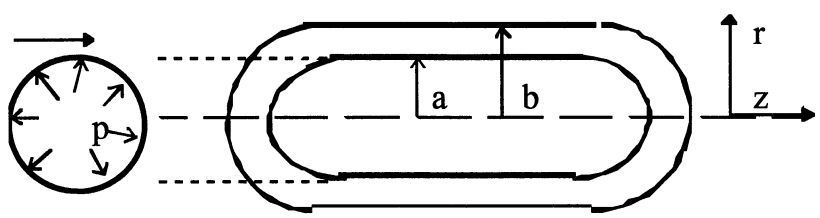


Figure 1. Simplifies SCUBA system to be analyzed and redesigned.

The coordinate system used is a radial ®, axial (z), and tangential (θ) system. Where a and b are the inner and outer radii, respectively. The internal pressure is designated by p. The ends of the tank are assumed to be hemispherical. The true system has a somewhat

hemispherical end where the nozzles of the tank are supposed to be. The other end of a real tank is ellipsoidal "to the one", which designates the ratio of the major to the minor axis. The material used in the analysis was AISI 4130 steel. This is a medium carbon and low alloy chromium-molybendum steel. It is tempered at 1100 °F and water quenched to give if more desirable properties [7]. From the coil, Pressed Steel uses a punch-die combination to produce a seamless pressure vessel so that joint efficiency (of welding) is of no concern.

System Failure

The ASME has set standards that certain mechanical components must meet in order to perform its function(s) safely. These standards can be found in the codes of the AMSE, which are volumes long. One set of these standards regards pressure vessel failure testing. Pressure vessels shall be hydro-statically tested every 5 years to determine if they are fit to resume operation. Here are some of the guidelines of the failure test:

1. Pressure vessels that are hydro-statically tested shall be filled with water to a test pressure.
2. The test pressure shall by be determined by the following formula:

   $$tp = 1.5\ (Stp / Sdt)\ dp \qquad \text{Eq.1}$$

   where, tp is the test pressure, Stp is the allowable stress at the test temperature, Sdt is the allowable stress at the design temperature, and dp is the design pressure.
3. The test temperature must be between 60°F and room temperature.

4. The vessel shall be blocked to permit examination during testing. Examination should occur at a pressure greater than 2/3 but less than 9/10 of the test pressure.
5. There is no upper limit to the test pressure. However, if the vessel is visibly distorted, the inspector may reject the vessel.

The TDOT clarified some of the vague rules of the ASME code when the vessels are SCUBA tanks. They enforce that SCUBA tanks are considered to fail the hydrostatic test if they permanently deform more than 10% of the original volume [3]. In testing SCUBA tanks, it is usual to place the tank in a shallow, sealed pool of water. The original volume is recorded by noting the volume of water displaced. The tank is then filled with water, a moment is waited, the water is released and the new volume is then recorded. Tank rejection is then only a matter of division.

System Failure, Probability of Failure, and the Limit State

When analyzing a structure, one must be aware of the ways that the system can fail. A system fails when it can no longer perform its function properly and/or safely. When will the SCUBA tank fail? ASME and TDOT have already answered that. The SCUBA tank in consideration will fail if, upon testing and examination, it permanently deforms more than 10% of its original volume. Due to the complexity of the literature available on the plastic volumetric expansion, it will be the scope of this work to consider failure to happen when the tank reaches the point of yielding during testing.

The probability of failure is the probability that the system failure will occur. For this system it is the probability that the test pressure reached will exceed that which causes yielding. In mathematical form, the probability of system failure is given by

$$P_f = P(p \geq p_e) = P(p_e - p = 0) = P(g(\bar{x}) = 0) \qquad \text{Eq.2}$$

where, P_f is the probability of tank failure, P() is the probability that the event in () will occur, p is the test pressure, p_e is the pressure that will cause the tank to begin yielding, and $g(\bar{x})$ is defined as the limit state of the problem. The limit-state is a function of design variables that breaks the probability space into safe and fail regions.

Deterministic System Analysis

For the SCUBA tank under consideration and the limit-state previously mentioned, the internal pressure that will cause the tank to yield according to the Von Misses criterion is given by

$$p_e = k\left(1 - \frac{a^2}{b^2}\right) \qquad \text{Eq.3}$$

The new term, k, is the material shear yield stress. The same result would have been obtained if Tresca's yield criterion had been used. The assumptions of Eq.3 are than plane cross sections remain plane, stresses and strains far from ends do not vary along the length of the vessel, and the material is linear elastic. Yielding of the tank begins at the inner wall (r=a) [6].

The yield stress in shear is related to the tensile yield strength by

$$k = 0.577 S_y \qquad \text{Eq.4}$$

The tensile yield strength is designated by S_y. Equation 4 comes directly from the Von Misses failure criteria prediction [1].

Now, let us reflect. Why would a tank that is designed to initially pass the hydrostatic test ever have the possibility of failing the test 5+ years later? What has

happened to the tank during its life? One source that causes the strength of the tank to degrade is fatigue. A 100 cu.ft. tank (3500psi, 238atm) has enough air to last 90 minutes. The pressure in the tank is then totally relieved; it then must be refilled for the diver to use it again. If a diver uses/refills the tank twice every week for 5 years, that results in a total of 520 cycles. After another 5 years, that would be 1040 cycles. That is already in the high cycle zone (above 10^3) on the S-N curve for that material. This fatigue reduces the ultimate tensile strength of the material. However, we are searching for the reason that the tensile yield strength, S_y, degrades due to cyclic loading. It has been proven that "the elastic limits of iron and steel can be changed...by cyclic variation of stress" [1]. Let us proceed to relate the lowering of the ultimate tensile strength, S_{ut}, to the lowering of the tensile yield strength, S_y.

There exists a minimum value of S_{ut} found by ASTM, it is given by the equation

$$S_{ut\,\min} = 0.45H_B \qquad \text{Eq.5}$$

The Brinnell hardness number is designated by H_B [1]. To relate the Brinnell hardness number to the tensile yield strength, a linear curve fit was done on data for tempered and water quenched AISI 4130 steels, for tempering temperatures between 400-1200°F [7]. The linear relation was calculated to be

$$H_B = 2.25S_y \qquad \text{Eq.6}$$

Combining equations 5 and 6 results in the following equation

$$S_{ut\,\min} = 0.45(2.25)S_y \qquad \text{Eq.7}$$

Substituting this into equation 4, the equation for the shear yield stress then becomes

$$k = S_{ut\,\min}\frac{0.577}{0.45(2.25)} \qquad \text{Eq.8}$$

If the average value of the yield strength in tension obtained from Pressed Steel (150 ksi), the minimum ultimate tensile strength then becomes 151.875 ksi (Eq.7). From Pressed Steel, the range of values of the yield strength between plus and minus 3 standard deviations from the mean was 25 ksi. In order to continue the analysis, it was *assumed* that the range of values for the ultimate tensile strength, S_{ut}, between plus and minus 3 standard deviations from the mean was 50 ksi. To be *conservative*, let the minimum value in equation 8 be the design variable S_{ut}, whose mean value obtained from equation 4 is 86.55 ksi. Since S_{ut} is most likely normally distributed [1], this turns the minimum value in equation 8 into a random variable in which almost all of the values fall below 151.875 ksi. The standard deviation is 8.33 ksi. Equation 8 then becomes

$$k = S_f \frac{0.577}{0.45(2.25)} \qquad \text{Eq.9}$$

Recall that the ultimate tensile strength is being lowered by fatigue. In equation 9, S_f is the ultimate tensile strength at a certain number of cycles. At a half cycle, it will equal S_{ut} (the beginning of the S-N curve). It is calculated from

$$S_f = (0.9S_{ut})^2 \frac{N^{-\frac{1}{3}Log_{10}\left(\frac{0.9S_{ut}}{S_e}\right)}}{S_e} \qquad \text{Eq.10}$$

The number of cycles the tank has undergone is designated by N; and S_e is the limit of the ultimate tensile strength of the material when subjected to a large amount of cycles (lower limit of S_f) [1]. The endurance limit is given by the following equations

$$S_e = k_a k_b k_c k_d k_e S'_e \qquad \text{Eq.11}$$

In equation 11, S'_e is the endurance limit of the test specimen (not machine part), k_a is the surface factor, k_b is the size factor, k_c is the load factor, k_d is the temperature factor, and k_e is the miscellaneous factor ($k_b = k_d = k_e = 1$)[1]. Once equations 2,3,9,10, and 11 are combined the limit-state for this problem can be expressed as

$$g(\bar{x}) = \frac{(0.9S_{ut})^2 N^{\left[-\frac{1}{3}Log_{10}\left(\frac{0.9S_{ut}}{k_a k_c S'_e}\right)\right]}}{k_a k_c S'_e} 0.56987\left(1 - \frac{a^2}{b^2}\right) - p \qquad \text{Eq.12}$$

Design Variables

Of the parameters given by equation 12, the deterministic design variables are the number of cycles during a 5 year period of the SCUBA tanks use, N, and the endurance limit of a test specimen of the material, S'_e. The stochastic design variables are the ultimate tensile strength of the material after 1/2 cycle, S_{ut}, the surface factor, k_a, the load factor, k_c, the inner radius, a, the outer radius, b, and the internal testing pressure, p. The design variable definitions are given in Table 1.

Table 1. Design Variable Definitions.

Design Variable	Description	Mean	Standard Deviation	Distribution
N	Cycles	1040	0	—
S'_e	Endurance Limit	63.945	0	—
S_{ut}	Ultimate T. Strength	86.55	8.3333	Normal
k_a	Surface Factor	0.74812	4.49E-02	Lognormal
k_c	Load Factor	0.5770	0.06347	Normal

A	Inner Radius	6.875	2.33E-03	Normal
B	Outer Radius	7.25	2.33E-03	Normal
P	Internal Pressure	5.833	4.86E-03	Normal

Most values in the table come from Shigley [1] or Pressed Steel. Units are in inches and kilo-pounds. The only assumptions made are that a, b, and, p are normally distributed.

Results

Table 2 compares the first run failure results from the FORM technique as well as the well-known Monte Carlo method. Percent differences between the two methods were calculated. The number of samples was increased from 100,000 to 200,000. The run times for the Monte Carlo methods were approximately 10 and 20 seconds. The FORM method took 4 iterations to converge upon 0% error in beta, the dot product, and the g-function. The run time for the FORM method was on the order of 4 seconds.

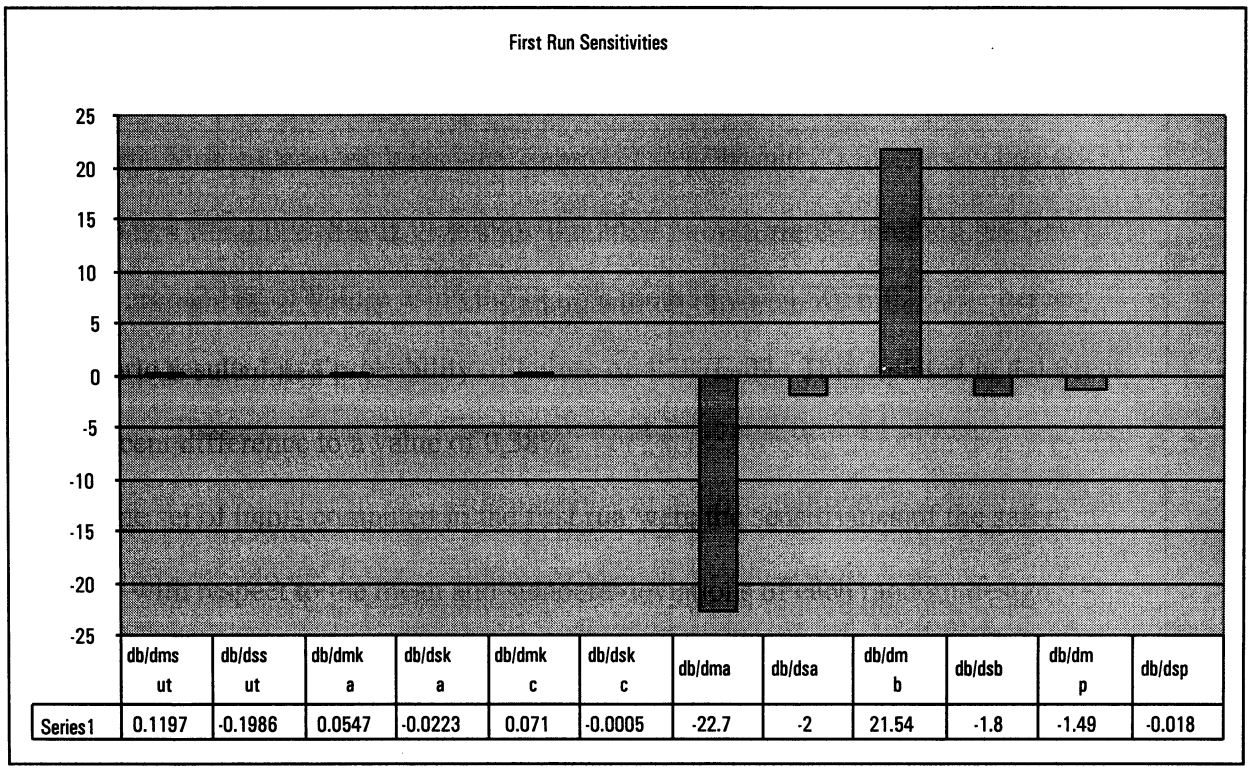
Table 2. First Run Comparisons of Probabilities of Failure

Case Number	Number of Monte Carlo Samples	Probability of Failure MC	Probability of Failure FORM	% Difference
1	100,000	4.783E-02	4.805E-02	0.46
2	200,000	4.787E-02	4.805E-02	0.38

Case 1 had a probability of failure of 4.783E-02 calculated using Monte Carlo while the probability of failure using the FORM method was 4.805E-02. The percent difference was calculated to be 0.46%. Increasing the number of samples used by the

Monte Carlo resulted in a probability of failure of 4.787E-02. This resulted in a decrease in the percent difference to a value of 0.38%.

One set of items computed in the first run were the sensitivities of the safety factor, β, with respect to the mean and standard deviations of each random design variable. Figure 2 shows these sensitivity factors.



	db/dmsut	db/dssut	db/dmka	db/dska	db/dmkc	db/dskc	db/dma	db/dsa	db/dmb	db/dsb	db/dmp	db/dsp
Series1	0.1197	-0.1986	0.0547	-0.0223	0.071	-0.0005	-22.7	-2	21.54	-1.8	-1.49	-0.018

Figure 2. First-Run Sensitivities.

The chart shows that 4 main factors effect the safety factor, β. As the safety factor is increased, the probability of failure will decrease. From Figure 2 it is deduced that the most important design variable parameters are the means of the inner and outer radii which have sensitivity values of -22.7 and 21.54, respectively; and the standard deviations of the inner and outer radii, which have values of -2.00 and -1.8, respectively. Several things can be concluded. If the mean of the inner radius is increased, β will

decrease, and the probability of failure will increase (or vise-versa). The same goes for increasing the standard deviations of the inner and outer radii, but to a lesser effect. If the mean of the outer radius is increased, β will increase, and the probability of failure will decrease (or visa-versa).

From these observations, one can deduce that the probability system failure can be reduced decreasing the mean of the inner and outer radii by the same amount. This will result in a smaller material cost for the manufacturer. Let the amount of decrease be 0.10 of an inch. Therefore, the new means of the inner and outer radii would then be 6.775 and 7.15 inches, respectively. All of the other values for the 2nd run are the same as that shown in Table 1.

Table 3 compares the second run failure results of the FORM and Monte Carlo technique. The number of samples used in the Monte Carlo technique was increased from 100,000 to 200,000. The run times for the Monte Carlo methods were approximately 10 and 20 seconds. The FORM method took 4 iterations to converge upon 0% error in beta, the dot product (just about), and the g-function. The run time for the FORM method was on the order of 4 seconds. . Percent differences between the two methods were calculated.

Table 3. Second-Run Failure Results.

Case Number	Number of Monte Carlo Samples	Probability of Failure MC	Probability of Failure FORM	% Difference
1	100,000	3.76E-02	3.74E-02	0.53
2	200,000	3.74E-02	3.74E-02	0

A probability of failure of 3.76E-02 was calculated using 100,000 samples and the Monte Carlo method while the probability of failure using the FORM method was 3.74E-02. The percent difference was calculated to be 0.53%. Increasing the number of samples used by the Monte Carlo resulted in a probability of failure of 3.74E-02. This resulted in a decrease in the percent difference to a value of 0%. The probability of failure as indicated by the FORM calculations decreased by 22.16% from run 1 to run 2.

The third run performed uses the values in Table 1, and again two items were varied. One was the mean of the inner radius, which was decreased by 0.1 inches. The other varied parameter was the mean of the outer radius. It was changed according to the following formula

$$\Delta_{mb} = 0.1(22.72/21.54)/10 \qquad \text{Eq. 13}$$

The term in ()'s is the ratio of the original sensitivities obtained in the first run. Therefore, the new inner radius mean was 6.775 inches and the new outer radius mean was 7.239452 inches. Table 4 shows the results of all three runs as well as the change in parameters.

Table 4. Results of 1st and 3rd runs compared. Lengths are in inches.

Run Number	Change of Inner Radius Mean	Change of Outer Radius Mean	Monte Carlo Samples	MC Failure Probability	FORM Failure Probability	%Error
1	NA	NA	200,000	4.787E-02	4.805E-02	0.38
2	0.1	0.1	200,000	3.74E-02	3.74E-02	0
3	0.1	0.0105479	200,000	3.85E-04	4.57E-04	18.7

The probabilities of failure of the first and second runs were of the same order of magnitude, and the inner and outer radii were both decreased by 0.1 inch. The probability of failure of the third run was two orders of magnitude lower than that of the first. This occurred when the inner radius was the only one to be significantly decreased. It is therefore recommended that the SCUBA tank be redesigned by decreasing the means of the inner and outer radii by the amounts shown in Table 4. This will decrease the probability of failure of the SCUBA tank.

Summary and Conclusions

In this paper, the probabilistic redesign of a SCUBA tank was performed by decreasing the probability of it yielding during a hydrostatic test. Effects that lowered the material strength and the fatigue of the tank due to filling/refilling were accounted for. Three different runs were performed on the system. Each run was done using the FORM technique and the results were verified using simple Monte Carlo. The random variables for all three runs included the ultimate tensile strength of the material, a surface factor, a load factor, the inner and outer radii, and the internal pressure during testing.

The first run FORM solution yielded a probability of failure of 4.805E-02. A sensitivity analysis showed that the two most important inputs in the design were the mean values of the inner and outer radii, whose sensitivities were -22.72 and 21.54,

respectively. The next highest sensitivity was -2.00 and belonged to the standard deviation of the inner radius.

The second FORM solution was performed after decreasing the mean values of the inner and outer radii by 0.1 inch. The probability of failure calculated was 3.74E-02. Even through a decrease in the inner radius mean produces a decrease in the probability of failure and vice-versa for the outer radius, the inner radius change dominated because of the higher sensitivity.

The third FORM solution was performed after decreasing the mean value of the inner radius by 0.1 inch and decreasing the outer radius by a value equal to the product of one-tenth the decrease of the inner radius with the ratio of the original inner radius sensitivity to the original outer radius sensitivity. The probability of failure calculated was 4.57E-04. The probability of failure was decreased by two orders of magnitude. The third run was the design chosen to be the new one. Using those values should result in a much safer system.

REFERENCES

1. Shigley, Joseph E; Mischke, Charles R.: Mechanical Engineering Design 5th. McGraw Hill, MI 1989.
2. Ugural, Ansel C. Stresses in Plates and Shells 2nd McGraw Hill MI. 1999.
3. Chuse, Robert; Eber, Stephen M.,PE : Pressure Vessels: The ASME Code Simplified 6th McGraw Hill NY 1984.
4. *Reliability and Safety of Pressure Components* Presented at: The Pressure Vessels and Piping Conference and Exhibit in Orlando, Florida. June 27-July 2, 1982. Sponsored by: The Operations, Applications and Components Committee The Pressure Vessels and Piping Division, ASME.
5. *Inelastic Behavior of Pressure Vessel and Piping Components* Presented at: The 1978 ASME/CSME Pressure Vessel and Piping Conference in Montreal, Quebec, Canada from June 25-29, 1978.
6. Chakrabarty, J. Theory of Plasticity McGraw Hill NY. 1987.
7. ASM Metals Handbook v1 10th ed.
8. ASM Metals Handbook v8 9th ed.

Acknowledgments

Fred Treu

Pressed Steel Tank Company

Milwaukee, WI.

Adrian Ellenwood

Oceans Window

San Antonio, TX.

Steve

Tropical Divers

San Antonio, TX.

Tom Clark

A1 Fires and Safety

San Antonio, TX.

APPENDIX III

USE OF PROBABILISTIC METHODS IN DESIGNS OF FLUIDIC SYSTEMS

Rodney Harris**, Cody Godines*, Luis Rangel*,. Edgar C. Herrera*, Randall D. Manteufel[1]
University of Texas at San Antonio, San Antonio, TX 78249-0665

Abstract

A common problem in fluidic systems is the proper selection of pump, pipes and fittings that will produce the desired flow in a system. Systems are often designed with excessive pump capacity as a result of conservatively overestimating the head loses and under estimating pump capacities. Once in operation, excessive throttling may be required which lowers efficiency and often introduces unwanted vibrations or noise. Probabilistic methods are used to aid in the design of fluidic systems. The piping network is examined for the probability that the flow rate through a component is above a minimum acceptable value. This is analogous to the probability of failure for a structural problem. The important variables are identified to lead the engineer in identifying potential design changes. A series piping system is evaluated as an example.

Introduction

Engineering analysis and designs involve computer software that can determine how the system performs under certain conditions of the variables the response is dependent on. This includes simulation of thermal-fluid systems such as pumps, series or parallel piping, valves, and heat exchangers. Due to the many failure modes of complex systems, many software packages are not practical for optimizing designs.

Through the use of the NASA developed program, called Numerical Evaluation for Stochastic Structures Under Stress (NESSUS), reliability-based analysis can be performed as a first step in the design of fluidic systems.

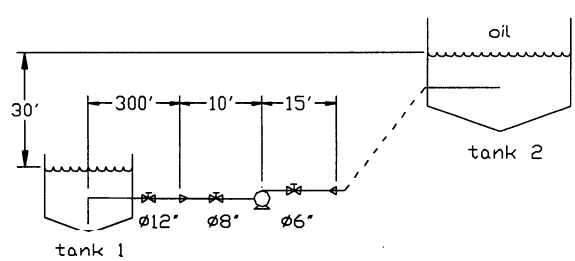
NESSUS is a FORTRAN based code, which includes a Fast Probability Integration (FPI) module for handling probabilistic analysis. The FPI module was utilized in this study of a series-piping system design. This portion of the code requires a user to input the system performance, or response, equation(s) as a function of the variables it depends on. This can be done using a graphical interface, manipulating the input file, or writing a FORTRAN subroutine called RESPON and linking it with the rest of the program. This subroutine provides the user a way to keep track of the conversions used, set up multiple response equations, and write an organized algorithm to aid in the response calculations. The underlying random variables are identified as random by entering their appropriate statistics and distribution type; thus, the response is also random and by estimating its statistics the reliability of the system can be quantified.

The response of a series piping system was written in the RESPON subroutine and the Advanced Mean Value plus iteration (AMV+) method was used in the analysis. The NESSUS *User's Manual* suggests this method because of its efficiency. The Standard Monte Carlo simulation is accurate, yet time consuming, and it was mainly used to verify the results of a few runs.

Methodology

As an example problem, an oil transfer system is designed to pump light oil through a series connection of different sizes and lengths of pipes, as shown in Figure 1. The direction of flow is from tank 1 to tank 2 (through a 30-ft increase in elevation of fluid level). The piping system is designed to operate at a nominal flow rate, Q, of 1500gpm. Valves, reducers, and elbows are also present within the system and contribute to the head or frictional losses, which the pump must overcome in order to maintain system functionality. The dashed line leading to the second tank represents eight-inch diameter pipe with five 90-degree angle elbows, whose contribution to the system head was accounted for in the analysis.

Figure 1. Light oil transport system.



*Graduate Research Assistant
**Undergraduate Research Assistant
[1] Assistant Professor of Mechanical Engineering

Copyright©2002 by R.D. Manteufel, Published by the American Institute of Aeronautics and Astronautics, Inc. with permission

Based on the uncertainty that exists in the system design variables, the flow rate will also be random. One may want to ensure that the system will deliver at least a minimum flow rate by calculating the probability of observing flow rates above a minimum acceptable value, which, for this specific problem, is 1400gpm. This probability would be a measure of the reliability of the system. Conversely, the probability that the flow rate would fall below 1400gpm would be the probability of failure. The reliability of the system shown in Figure 1 and the variables that contribute most to its uncertainty will be discussed.

Although many responses of this system could be studied, the response we are concerned with is the difference in the head the pump can deliver, H_p, and the system head that needs to be overcome, H_s. This response is shown in Equation 1.

$$Z = H_P - H_S \tag{1}$$

The response is a function the flow rate, Q, along with many other variables. It is important to keep in mind that once all dependent variables are realized, except for Q, the two terms become solely dependent on Q. The system will operate, in steady state, at the point where the pump head matches the system head. During the NESSUS execution, the dependent variables, except for Q, are realized, and the RESPON routine, which was coded with the Newton-Raphson routine is called upon and solves for the Q that satisfies the equation $H_p = H_s$. Also, considering the two terms in Equation 1 to be functions, when all dependent variables are set with a value, along with Q, and if the response is negative, then the actual steady state flow rate will be lower than the Q used to calculate the two head terms. This is because the system requires more head than the pump can deliver at that flow rate. If the response is positive then the system will operate at a higher flow rate than that entered.

The pump head is given in Equation 2, where Q is the flow rate in gallons per minute (gpm). HP is a factor that introduces uncertainty into the pump head and its distribution and statistics are shown in Table 1. Using a random variable to reduce modeling error or capture uncertainty is common in a probabilistic analysis. This equation is only a rough approximation of a centrifugal pump near the anticipated operating point for this system, and is a curve fit to manufacturer specified pump characteristic data.

$$H_p = HP\left(572 + 0.0384Q - 0.00006Q^2\right) \tag{2}$$

A form of the energy equation is used to model the system head H_s, as expressed in Equation 3.

$$H_s = \frac{\Delta P}{\rho} + \Delta Z \frac{g}{g_c} + \frac{8Q^2}{\pi^2 g_c} \sum_{i=1}^{J} \frac{\left(F\left(f_i \frac{L_i}{D_i}\right) + K_i\right)}{D_i^4} \tag{3}$$

The system head is consists of three terms representing the pressure difference and elevation difference between ends of the piping system, and the fluid frictional losses through the piping and fitting. For this example, the first term is zero because the fluid in both tanks is exposed to atmospheric pressure. The second term is the head required to overcome an elevation increase, ΔZ. This term is not dependent on the flow rate. The total head loss due to pipe friction and the numerous valves and fittings in the pipe system is accounted for in the last term where f, L, D and K represent the friction factor, pipe length, pipe diameter, and minor loss coefficient of the respective pipe or fitting location. The variable F, like HP in Equation 2, is used to capture more of the uncertainty present in the friction factor equation. The Churchill curve fit is used for the friction factor, (Hodge and Taylor, 1999), and is shown below in Equation 4.

$$f = 8\left[\left(\frac{8}{\mathrm{Re}_D}\right)^{12} + \frac{1}{(A+B)^{1.5}}\right]^{1/12} \tag{4}$$

The parameters A and B, given in Equations 5 and 6, are functions of the local Reynolds number, Re_D, and the relative roughness of the section under consideration, ε/D.

$$A = \left\{2.457 \ln\left[\frac{1}{(7/\mathrm{Re}_D)^{0.9} + (0.27\varepsilon/D)}\right]\right\}^{16} \tag{5}$$

$$B = \left(\frac{37{,}530}{\mathrm{Re}_D}\right)^{16} \tag{6}$$

Although the Churchill equation is complex, it can be used in the transition region between laminar and turbulent flow as well in the turbulent region for non-smooth pipes. The Churchill equation does represent one of the major sources of nonlinearity in the system.

The statistical data for the 16 variables of the system is shown in Table 1, and the other system parameters are shown in Table 2. All of the variables are normally distributed. The uncertainty factors, HP and F, have a mean of 1 and coefficient of variations of 5% and 2.5%,

respectively.

Table 1. Random Variables

Variable	Description	Distribution
HP	Pump-head correction factor	N(1, 0.05)
L6	Length of 6-in. pipe (ft)	N(15, 2.5)
L8	Length of 8-in. pipe (ft)	N(6010, 100)
L12	Length of 12-in. pipe (ft)	N(300, 10)
D6	6-in. pipe diameter (in)	N(6.065, 0.025)
D8	8-in. pipe diameter (in)	N(7.981, 0.025)
D12	12-in. pipe diameter (in)	N(11.938, 0.025)
ΔZ	Elevation increase (ft)	N(30, 0.5)
K6	Minor loss coefficient for fittings in 6-in pipe	N(2.2, 0.205)
K8	Minor loss coefficient for fittings in 8-in pipe	N(3.06, 0.425)
K12	Minor loss coefficient for fittings in 12-in pipe	N(2.2, 0.53)
ρ	Density of light oil (lbm/ft^3)	N(56.9, 1E-7)
μ	Viscosity of light oil (lbm/ft-s)	N(4.3E-2, 7.6E-3)
ε	Pipe roughness (ft)	N(1.5E-4, 3.75E-5)
F	Friction correction factor	N(1, 0.025)

Table 2. System Parameters

Variable	Description	Value
f	Friction factor	See Equation 4
g	Gravitational constant (ft/s^2)	32.174
g_c	Conversion factor (ft-lbm/lbf-s^2)	32.174
Q	Flow rate (ft^3/s)	Specified

Characteristic curves of the pump and system were obtained and show the possible range of interaction between the pump and the piping system. For the pump, this was done by entering Equation 2 as the response in NESSUS, setting a Q value, and executing the code to obtain the pump head value (for that Q) at which there is a 2.5% chance that the pump head will be below that value. This was also done to obtain the 50% and 97.5% quantiles of the pump head, at that flow rate. This was repeated for Q values ranging from 500 to 2000 gpm. The same type of analysis was performed by using Equation 3 and its dependencies, Equations 4, 5, and 6, for flow rates ranging from 1000 to 1900. Some of the results of analyzing the system head, H_s, are shown in Table 3. Interpreting the data in Table 3 must be done cautiously. At the flow rate of 1000 gpm, the system head will be below 225.6 ft-lbf/lbm 2.5% of the time. This value is random only because the underlying random variables are also random. It will be below 252.7 and 278.2 ft-lbf/lbm 50% and 97.5% of the time, respectively.

Table 3. Piping system head required for different values of flow rate at the respective lower limit, middle, and upper limit probability levels P (.025, .5, .975).

	Hs (ft-lbf/lbm)		
(gpm)	P(0.025	P(0.5)	P(0.975
1000	225.6	252.7	278.2
1100	260.9	292.3	322.1
1200	298.6	334.8	369.0
1300	338.8	379.9	418.9
1400	381.5	427.8	471.7
1500	426.7	478.4	527.4
1600	474.2	531.6	586.0
1700	524.1	587.3	647.5
1800	576.3	645.7	711.7
1900	630.9	706.6	778.7

The system head data of Table 3 and similar data for the pump head, H_p, is plotted in Figure 2. As expected, the head that the pump can deliver will decrease and the system head will increase as the flow rate increases. At a specified flow rate, the pump and system head each have three values that were calculated using NESSUS, for which a portion of these values is shown in Table 3. At a given flow rate, the lower value of the respective curve is the value at which there is a 2.5% probability of the appropriate head falling below this value. This line is the lower dashed line of the pump and system head curves. The solid middle line at a certain flow rate is the value at which there is a 50% chance that the pump or system head will be below this value. For the upper, dashed line at a specific flow rate, that is the value at which there is a 97.5% chance of observing a head below that value. Therefore, for either the pump or system curve and at a specific flow rate, 95% of the

time the head value will be between the upper and lower values of the dashed curve.

Consider a low flow rate of 1200gpm and the head curves shown in Figure 2 and let the following values be non-exact read offs from the same figure. The pump head will be below 475ft-lbf/lbm 2.5% of the time and the system head will be above 370 ft-lbf/lbm 2.5% of the time. There is therefore little chance that the pump and system head will ever be equal; hence, there is a very low probability that the pipe system will operate at 1200gpm.

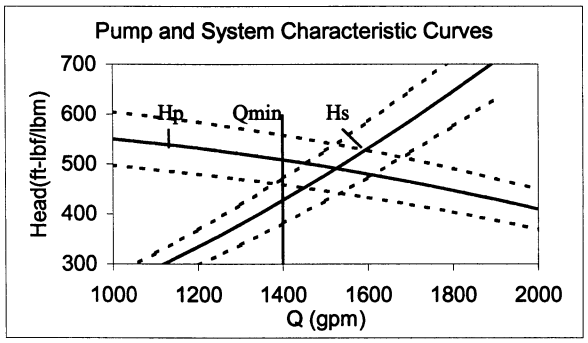


Figure 2. Head curves as functions of flow rate for the pump Hp, and the piping system Hs.

For a flow rate of 1400gpm, the pump will deliver a head between 455 and 555 ft-lbf/lbm 95% of the time and the system head will be between 380 and 480 ft-lbf/lbm 95% of the time so there is a decent chance that the pump and system head will be equal. Therefore, there is a higher probability that the pipe system will operate at a flow rate of 1400gpm than at 1200gpm.
This overlap of probable values of the pump and system head can be considered to begin when the two curves begin to intersect at 1380gpm. The overlap increases up to the intersection of the solid lines of the two curves, which occurs at 1524gpm, close to the nominal value 1500gpm that the system is designed for. Then it decreases until the curves finish their final intersection at 1680gpm. These values were estimated using linear interpolation among the head data, and then pinpointed with additional runs.

Monte Carlo sampling was used to visualize how often the system performed within the diamond region shown in Figure 2. Basically, 500 sets of values of the random variables in Table 1 were obtained using Monte Carlo sampling and formed coordinates in the multidimensional space, which is the domain of the pump and system equations. For each coordinate, a Newton-Raphson routine solved for the Q that satisfied the equation $H_p = H_s = H$, and the (Q,H) pair was recorded. The results are illustrated in Figure 3. As a rule of thumb, it is estimated that 2.5% are to the outlying sided of each of the four lines, hence (2.5%)x(2.5%) are in the outlying corners of each.

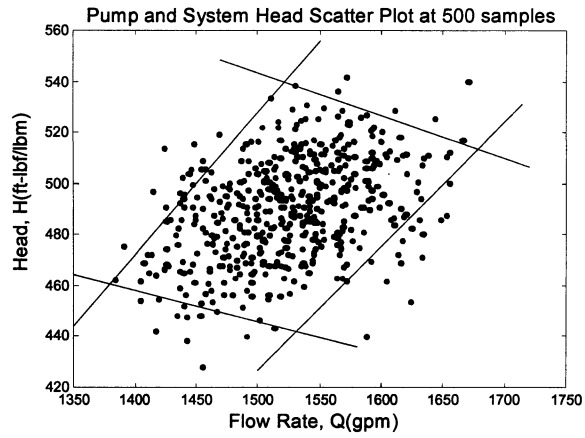


Figure 3. Scatter-plot for Head and Flow.

Figure 3 confirms that the vast majority of observations are within the uncertainty box and that only a few observations are outside (about 95%X95%=90.25% within, and 9.75% outside). As a quick estimate, the probability of operating with a flow rate less that the minimum corner flow rate is the sum of three regions: 2.5%x2.5%+2.5%x95%+2.5%x95% = 4.8% or simply 5%.

For this problem, the cumulative distribution function (CDF) was calculated by NESSUS via multiple runs. The CDF of the flow rate is shown in Figure 4. Entering flow rates ranging from 1380 to 1700gpm at intervals of 20gpm, NESSUS calculated the probability that the response of Equation 1 is less than zero. This implies that the system head is greater than the pump head and therefore this is the probability that the flow rate will be less than that entered by the user.

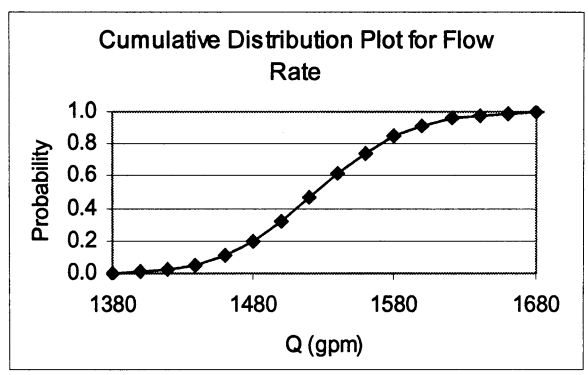


Figure 4. Probability of achieving a flow rate that is less than Q.

Recall that the minimum acceptable flow rate for this problem was 1400gpm. It was calculated that there is a 0.9% probability that the system flow rate will be less

than 1400gpm. A Monte Carlo simulation was used to verify this probability, and there was only a small margin of error (2.69E–4). Figure 5 shows the probability density function (PDF) for the system flow rate, Q. This was derived directly from the data provided by the CDF because the PDF is the derivative of the CDF.

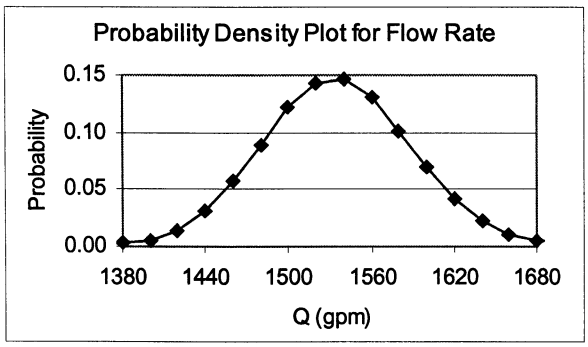


Figure 5. The probability density for the flow rate Q.

The AMV+ method provided by NESSUS also calculates the probabilistic sensitivity factors, or "alphas's" for each of the design variables. The alpha of a variable represents the overall importance of that variable because it is the sensitivity of the response to that variable multiplied by its range or standard deviation; thus, the range is used to weight the sensitivity of the response to a variable. Figure 6 shows the variables as they contributed to the uncertainty of only the system head at a flow rate of 1524gpm and at the 50% head value level. Viscosity contributed to about 67% of the uncertainty of the system head. The friction factor was next contributing 18% to the system head uncertainty, followed by the length and the diameter size of the nominal 8-inch pipe at 7.8% and 6.3% respectively. The total contribution of the remaining components was about 0.5%. The pump head is not a part of this variable importance discussion.

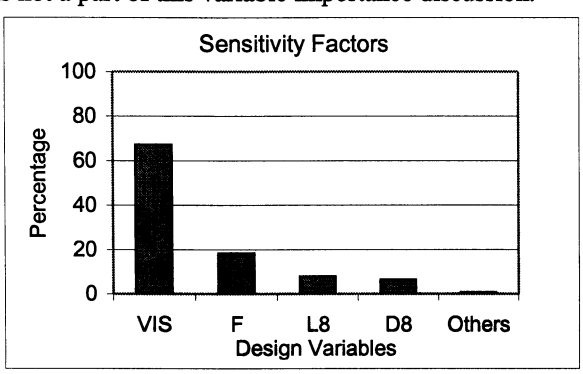


Figure 6. Sensitivity Factors.

CONCLUSIONS

This work demonstrated the use of NESSUS in evaluating the design of a series piping system. The probability that the system flow rate will be less than 1400gpm was calculated with the AMV+ method to be only 0.9%, and this was verified with a Monte Carlo run. The system is thus 99.1% reliable in operating at a flow rates over 1400gpm. However, according to the distribution shown in Figure 5, it is more probable that the oil transfer system will operate closer to 1524gpm, which is the apparent mean of that PDF, than the stated nominal design of 1500 gpm. The viscosity was identified as the design parameter that is most significant in dominating the piping system performance, owing to its 67% significance in system sensitivity. These results demonstrate the versatility of the NESSUS program to a wide variety of applications.

ACKNOWLEGEMENTS

This report was supported by NASA Grant NAG3-229. A special thanks is owed for the support of the technical monitor, Dr. Christos C. Chamis of NASA Glenn Research Center.

REFERENCES

Hodge, B. K. and R.P. Taylor. *Analysis and Design of Energy Systems.* 3rd edition. Prentice-Hall, Inc., New Jersey, 1999.

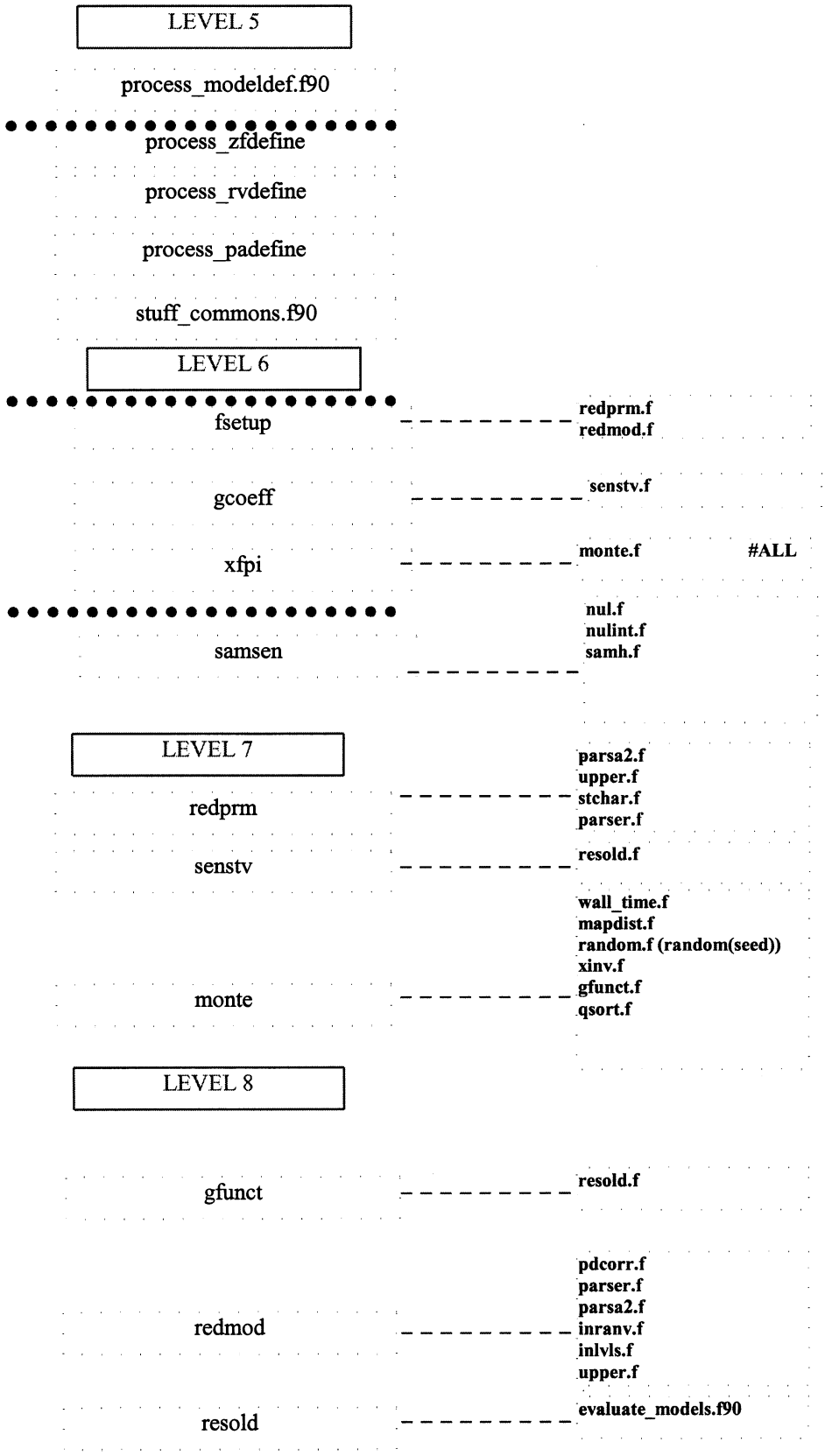
NESSUS User's Manual. Vol. 1, prepared by the Southwest Research Institute for NASA-Glenn Research Center (1996).

Ang, A. H. S. and Tang, Wilson H. *Probability Concepts in Engineering Planning and Design.* John Wiley & Sons, New York, 1984.

APPENDIX IV

APPENDIX IV-A: NESSUS PROGRAM LEVELS & SUBROUTINE PATH FOR MC P-LEVEL ANALYSIS

Actions	Parent Subroutine	Child Subroutine
	LEVEL 1	
	NESSUS	
blank DOS window up. no files made. Read *NESSUS goto new_nessus.	LEVEL 2	timer.f verinc.f prompt_user.f intint.f
STOP	NESMAIN	reinit.f intini.f init_input.f90 header.f new_nessus.f90 #ALL
	LEVEL 3	
"Enter input file.." in DOS window	PROMPT_USER	
Makes all files (0kb) except *.zal, , scodefpi.*, analytical* and *.verify	INTINT	
OUTPUT header SCREEN header	HEADER	set_working_directory.f90 init_input.f90
STOP	NEW_NESSUS	read_nessus_input.f90 model_setup.f90 fpi.f #ALL
	LEVEL 4	
probid=only filename w/o ext or dir	set_working_directory	
initialize derived data types	init_input	
analytical and scodefpi files added	read_nessus_input	read_card (in string_functions.f90) parsa256.f (in string_functions.f90) process_zfdefine.f90 process_rvdefine.f90 process_padefine.f90 process_modeldef.f90 echo_input.f90 stuff_commons.f90 *look #ALL
Nothing happens.	model_setup	
	fpi	fsetu1.f cmpfpi.f dumpsf.f quit.f #ALL


LEVEL 5
process_modeldef.f90
process_zfdefine
process_rvdefine
process_padefine
stuff_commons.f90
LEVEL 6
fsetup
redprm.f
redmod.f
gcoeff
senstv.f
xfpi
monte.f
#ALL
samsen
nul.f
nulint.f
samh.f
LEVEL 7
redprm
parsa2.f
upper.f
stchar.f
parser.f
senstv
resold.f
monte
wall_time.f
mapdist.f
random.f (random(seed))
xinv.f
gfunct.f
qsort.f
LEVEL 8
gfunct
resold.f
redmod
pdcorr.f
parser.f
parsa2.f
inranv.f
inlvls.f
upper.f
resold
evaluate_models.f90

APPENDIX IV-B: OUTPUT FILE AND THE SUBROUTINES THAT PRODUCE THE SHOWN ITEMS

OUTPUT FILE: SAE1.OUT
Unit 1(ILPRINT, munit)

```
NN      NN  EEEEEEEE   SSSSSSS   SSSSSSS  UU    UU   SSSSSSS
NNN     NN  EE        SS        SS        UU    UU  SS
NN N    NN  EE        SS        SS        UU    UU  SS
NN  N   NN  EEEEEE     SSSSSS    SSSSSS   UU    UU   SSSSSS
NN   N  NN  EE              SS        SS  UU    UU        SS
NN    NNN   EE              SS        SS  UU    UU        SS
NN      NN  EEEEEEEE  SSSSSSS   SSSSSSS    UUUUUU   SSSSSSS
```

HEADER

```
          9-06-2001   3:43  - LEVEL 3.00(  39)  - DATED JUL     1, 2000
Build Date: 08/14/01 12:01:21
```

```
THIS IS A PROPRIETARY PROGRAM. IT MAY ONLY BE USED UNDER THE TERMS
OF THE LICENSE AGREEMENT BETWEEN SOUTHWEST RESEARCH INSTITUTE AND THE
CLIENT.

SOUTHWEST RESEARCH INSTITUTE DOES NOT MAKE ANY WARRANTY OR
REPRESENTATION WHATSOEVER, EXPRESSED OR IMPLIED, INCLUDING ANY
WARRANTY
OF MERCHANTABILITY OR FITNESS OF ANY PURPOSE WITH RESPECT TO THE
PROGRAM; OR ASSUMES ANY LIABILITY WHATSOEVER WITH RESPECT TO ANY USE
OF
THE PROGRAM OR ANY PORTION THEREOF OR WITH RESPECT TO ANY DAMAGES
WHICH
MAY RESULT FROM SUCH USE.
• • • • • • • • • • • • • • • •
```

READ NESSUS INPUT

```
*TITLE SAE TEST CASE 1
*DESCRIPTION
SAE TEST CASE 1 CYCLES TO FAILURE NON-LINEAR, NON-NORMAL 4 RANDOM
VARIABLES
*RVDEFINE
*PADEFINE
*MODELDEFINE
*END NESSUS
```

```
End of file reached: checking data..
. . . . . . . . . . . . . . . .
FPI
=====================================================================
===
                         *****     INPUT ECHO     *****
=====================================================================
========


 LINE

      1    *FPI
      2    NESSUS generated FPI deck: Analytical model: ANALYTICAL_1
      3    *RVNUM              4
      4    *GFUNCTION     USER
      5    *METHOD        MONTE
      6    *PRINTOPT
      7    *ANALTYP       PLEVEL
      8    *END
      9    *MONTE      1       1
     10          1000          7654321      1.00000
     11       MAXTIME
     12         500000.
FSETU1 [illegible]       20      1
                         -4.753258      -4.264844      -3.719124
-3.09[illegible]
     15      -2.326785      -1.281729      -1.036431     -0.6741892
-0.1010067E-06
     16      0.6741892       1.036431       1.281729       2.326785
3.090522
     17       3.719124       4.264844       4.753258       5.199082
5.611680
     18    *DEFRANVR
     19    KIC
     20      60.00000       6.000000        NORM
     21    AI
     22     0.1000000E-01   0.5000000E-02    LOGN
     23    C
     24     0.1200000E-09   0.1200000E-10    LOGN
     25    DS
     26      100.0000       10.00000         LOGN
     27    *END


=====================================================================
========
=====================================================================
========
 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
=====================================================================
========
=====================================================================
========
1====================================================================
=========
                *****     PARAMETER INTERPRETATION     *****
```

```
==================================================================
========

 Problem Title:  NESSUS generated FPI deck: Analytical model:
ANALYTICAL_1

 Number of Random Variables:      4

 Type of Response (g) Function Approximation:
      6 = User-defined response function
      Response function must be programmed in subroutine RESPON

 Number of Datasets:   0

 Solution Technique:
      6 = Standard Monte Carlo method (Radius = 0)
          *MONTE keyword is required in model input data

 Analysis Type:
      2 = User-defined probability levels (P-levels)
          *PLEVELS keyword is required in model input data
          Time consuming analysis because of iteration procedures

 Confidence Interval Calculation on CDF:
      0 = No

 Print option:
      0 = Short printout

 Debugging Option:
     -1 = No
```

FSETUP

CMPFPI

```
=====================================================================
=====================================================================
=====================================================================
=====
1====================================================================
=========
                    *****     MODEL INTERPRETATION     *****
==================================================================
========


 Problem Title:  NESSUS generated FPI deck: Analytical model:
ANALYTICAL_1

 User-Defined Probability P-levels:

     Number        P-Level        u-level
+    ______        _______        ________

        1        0.10033E-06      -5.1991
        2        0.10021E-05      -4.7533
        3        0.10009E-04      -4.2648
        4        0.99987E-04      -3.7191
```

```
        5        0.99909E-03      -3.0905
        6        0.99883E-02      -2.3268
        7        0.99969E-01      -1.2817
        8        0.15000          -1.0364
        9        0.25010         -0.67419
       10        0.50000           0.0000
       11        0.74990          0.67419
       12        0.85000           1.0364
       13        0.90003           1.2817
       14        0.99001           2.3268
       15        0.99900           3.0905
       16        0.99990           3.7191
       17        0.99999           4.2648
       18         1.0000           4.7533
       19         1.0000           5.1991
       20         1.0000           5.6117


 Random Variable Statistics:

     Random Variable   Distribution      Mean         Standard
Deviation
+    ________________  _____________     ____
_________________

        KIC                NORMAL         60.00            6.000
        AI                               0.1000E-01      0.5000E-02
        C                                0.1200E-09      0.1200E-10
        DS                                100.0            10.00

 User-Defined Response Function Equation Parameters (Sub [RESPON]) :

     Equation Number =     1


 Standard Monte Carlo Method (Radius = 0):
         Minimum Sample Size =     1000
                        Seed = 0.765432E+07
             Allowable Error = 0.100000
        Allowable Confidence = 0.950000
         Maximum Sample Size =  2000000
     Maximum Wall Time (sec) =  500000.
         Empirical CDF Print = OFF
             Histogram Print = OFF
     X-space samples will be written to jobid.smx file. Skip factor =
1
     u-space samples will be written to jobid.smu file. Skip factor =
1


 ==========================================================================
 ========
 ==========================================================================
 ========
 1=========================================================================
 =========
                        *****     OUTPUT SUMMARY     *****
```

FSETUP.F

CMPFPI.F

gcoeff

```
=================================================================
========

 PROBLEM TITLE: NESSUS generated FPI deck: Analytical model:
ANALYTICAL_1

 RESPONSE FUNCTION (LIMIT STATE): USER-DEFINED FUNCTION
                                  IN SUBROUTINE [RESPON]

 APPROXIMATE STATISTICS FOR Z:
          MEDIAN =    0.1575E+05
          MEAN =    0.1419E+05
          STANDARD DEVIATION =     7086.
```

Unnecessary evaluations for sampling method.

```
 NOTE:  Standardized Normal Variates are used in the following
analysis.
        This means that the random variable, u, represents a normal
        probability distribution with mean = 0 and standard
        deviation = 1.  For example, u = -3 implies that the chance
        of observing a u value <= -3 is .00135 (cdf).  Also, u = 3
        implies that the chance of observing a u value <= 3 is 0.99875.

 NUMBER OF SAMPLES FOR PLEVELS ANALYSIS:      1000

 MONTE CARLO SOLUTION:
   NUMBER OF VARIABLES =   4
   NUMBER OF SAMPLES =      1000
   SAMPLE MEAN =   1.73339E+04
   SAMPLE STD. DEV. =   9.44439E+03
```

XFPI.F

```
 RANDOM VARIABLE STATISTICS:

  Random      Input       Input       Sample      Sample      % error
%error
 Variable     Mean       Std. Dev.     Mean      Std. Dev.    Mean
Std. Dev.
 --------   -------    ----------   --------   ----------  --------  --
--------
 KIC        60.00       6.000       60.18       5.968         0.30
0.53
 AI        0.1000E-01  0.5000E-02  0.1006E-01  0.5368E-02     0.57
7.37
 C         0.1200E-09  0.1200E-10  0.1203E-09  0.1200E-10     0.27
0.02
 DS         100.0       10.00       100.3       9.849         0.30
1.51


 CDF SUMMARY
    Pr(Z<=Z0)          u                Z0        #Pts<=Z0      Error(*)
```

```
-------------- -------------- -------------- ---------- --------------
 0.1003285E-06  -5.199081       0.000000              0   195.6752
 0.1002053E-05  -4.753258       0.000000              0   61.91593
 0.1000893E-04  -4.264844       0.000000              0   19.59079
 0.9998665E-04  -3.719123       0.000000              0   6.198052
 0.9990933E-03  -3.090522      -66.25305              1   1.959873
 0.9988332E-02  -2.326784       3281.138             10  0.6170518
 0.9996893E-01  -1.281728       7861.708            100  0.1859706
 0.1500006      -1.036431       9101.901            150  0.1475404
 0.2500954     -0.6741884       10884.73            250  0.1073243
 0.5000000       0.000000       15224.53            500  0.6197949E-01
 0.7499046      0.6741893       21223.51            750  0.3579298E-01
 0.8499994       1.036431       26439.09            850  0.2603665E-01
 0.9000311       1.281729       29525.22            900  0.2065626E-01
 0.9900117       2.326785       46911.88            990  0.6225501E-02
 0.9990009       3.090523       70770.09            999  0.1960054E-02
 0.9999000       3.719124       74538.91           1000  0.6197845E-03
 0.9999900       4.264845       74538.91           1000  0.1960848E-03
 0.9999990       4.753259       74538.91           1000  0.6204310E-04
 0.9999999       5.199082       74538.91           1000  0.1963180E-04
  1.000000       5.611680       74538.91           1000  0.6212061E-05
(*) Sampling error at 0.95 confidence

**********************************************************************
*
      Probabilistic Sensitivity Results printed by level
**********************************************************************
*


=====================================================================
=
 Level=  1 Z0= 0.00000     CDF=0.100329E-06 No. Failure Samples=
1

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
---------------------------------------------------------------------
-
 KIC         -0.3953E-07    0.7671E-07    -2.364         4.588
 AI           0.1012E-04    0.9823E-04    0.5046         4.895
 C            2642.         -8103.        0.3160       -0.9692
 DS           0.2013E-07    0.4975E-07     2.007         4.959


=====================================================================
=
 Level=  2 Z0= 0.00000     CDF=0.100205E-05 No. Failure Samples=
1

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
```

DUMPSF

```
-----------------------------------------------------------------------
-
 KIC        -0.3948E-06    0.7662E-06    -2.364         4.588
 AI          0.1011E-03    0.9811E-03    0.5046         4.895
 C           0.2639E+05   -0.8093E+05    0.3160       -0.9692
 DS          0.2011E-06    0.4969E-06     2.007         4.959


=======================================================================
=
 Level=  3 Z0= 0.00000     CDF=0.100089E-04 No. Failure Samples=
1

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
-----------------------------------------------------------------------
-
 KIC        -0.3943E-05    0.7653E-05    -2.364         4.588
 AI          0.1010E-02    0.9799E-02    0.5046         4.895
 C           0.2635E+06   -0.8084E+06    0.3160       -0.9692
 DS          0.2009E-05    0.4963E-05     2.007         4.959


=======================================================================
=
 Level=  4 Z0= 0.00000     CDF=0.999867E-04 No. Failure Samples=
1

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
-----------------------------------------------------------------------
-
 KIC        -0.3939E-04    0.7645E-04    -2.364         4.588
 AI          0.1009E-01    0.9789E-01    0.5046         4.895
 C           0.2633E+07   -0.8076E+07    0.3160       -0.9692
 DS          0.2007E-04    0.4958E-04     2.007         4.959

No failure samples found for level  5


=======================================================================
=
 Level=  6 Z0= 3281.14     CDF=0.998833E-02 No. Failure Samples=
10

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
-----------------------------------------------------------------------
-
 KIC        -0.1251E-02    0.2468E-03   -0.7515         0.1483
 AI          1.563          6.122        0.7822         3.065
 C           0.4871E+09   -0.1851E+09    0.5852       -0.2224
 DS          0.1555E-02    0.3261E-02     1.557         3.265
```

```
=====================================================================
=
 Level=  7 Z0= 7861.71     CDF=0.999689E-01 No. Failure Samples=
100

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
---------------------------------------------------------------------
-
 KIC        -0.5014E-02   -0.1293E-02   -0.3010       -0.7759E-01
 AI           20.97         21.00         1.049         1.050
 C          0.2742E+10   -0.5152E+09    0.3292       -0.6184E-01
 DS         0.9617E-02    0.8187E-02    0.9620        0.8190


=====================================================================
=
 Level=  8 Z0= 9101.90     CDF=0.150001     No. Failure Samples=
150

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
---------------------------------------------------------------------
-
 KIC        -0.5393E-02   -0.7575E-03   -0.2157       -0.3030E-01
 AI           30.33         16.45         1.011        0.5482
 C          0.4867E+10    0.5094E+09    0.3894        0.4075E-01
 DS         0.1379E-01    0.8382E-02    0.9190        0.5588


=====================================================================
=
 Level=  9 Z0= 10884.7     CDF=0.250095     No. Failure Samples=
249

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
---------------------------------------------------------------------
-
 KIC        -0.3668E-02   -0.1554E-02   -0.8801E-01   -0.3728E-01
 AI           47.01         8.522        0.9399        0.1704
 C          0.6584E+10   -0.2846E+09    0.3159       -0.1366E-01
 DS         0.2018E-01    0.7090E-02    0.8070        0.2835


=====================================================================
=
 Level= 10 Z0= 15224.5     CDF=0.500000     No. Failure Samples=
500

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
```

```
-----------------------------------------------------------------------
-
 KIC        -0.5228E-02    0.1163E-02   -0.6274E-01    0.1395E-01
 AI           68.36        -15.14        0.6836       -0.1514
 C           0.7685E+10   -0.1757E+10    0.1844       -0.4217E-01
 DS          0.2946E-01   -0.2469E-02    0.5893       -0.4938E-01


=======================================================================
=
 Level= 11 Z0= 21223.5     CDF=0.749905     No. Failure Samples=
251

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
-----------------------------------------------------------------------
-
 KIC         0.9230E-02    0.9673E-03    0.2214        0.2321E-01
 AI          -74.93         47.46        -1.498        0.9487
 C          -0.5949E+10    0.3434E+10   -0.2855        0.1648
 DS         -0.2211E-01    0.1131E-01   -0.8840        0.4523


=======================================================================
=
 Level= 12 Z0= 26439.1     CDF=0.849999     No. Failure Samples=
150

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
-----------------------------------------------------------------------
-
 KIC         0.6017E-02    0.1478E-02    0.2407        0.5914E-01
 AI          -57.59         41.81        -1.920         1.394
 C          -0.4344E+10    0.2924E+10   -0.3475        0.2340
 DS         -0.1742E-01    0.1295E-01    -1.161        0.8632


=======================================================================
=
 Level= 13 Z0= 29525.2     CDF=0.900031     No. Failure Samples=
101

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
-----------------------------------------------------------------------
-
 KIC         0.2334E-02   -0.7683E-03    0.1401       -0.4611E-01
 AI          -44.14         34.47        -2.208         1.724
 C          -0.3231E+10    0.2294E+10   -0.3878        0.2753
 DS         -0.1391E-01    0.1305E-01    -1.391         1.305
```

```
=================================================================
=
 Level= 14 Z0= 46911.9     CDF=0.990012     No. Failure Samples=
11

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
-----------------------------------------------------------------
-
 KIC          0.5415E-03    0.8043E-04    0.3253        0.4831E-01
 AI           -8.268         7.957        -4.139         3.983
 C           -0.2563E+09   -0.3186E+09   -0.3079       -0.3827
 DS          -0.2248E-02    0.3125E-02    -2.251         3.129


=================================================================
=
 Level= 15 Z0= 70770.1     CDF=0.999001     No. Failure Samples=
2

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
-----------------------------------------------------------------
-
 KIC         -0.5568E-04   -0.3061E-05   -0.3344       -0.1838E-01
 AI          -0.9587        0.9510        -4.798         4.759
 C                          -0.6268E+08   -0.7629E-01   -0.7528
 DS                         0.5709E-03    -3.117         5.714


=================================================================
=
 Level= 16 Z0= 74538.9     CDF=0.999900     No. Failure Samples=
1

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
-----------------------------------------------------------------
-
 KIC          0.9970E-05   -0.1070E-04    0.5983       -0.6421
 AI          -0.1032        0.1050        -5.159         5.251
 C           -0.4574E+07   -0.4761E+07   -0.5489       -0.5714
 DS          -0.2799E-04    0.4637E-04    -2.800         4.638


=================================================================
=
 Level= 17 Z0= 74538.9     CDF=0.999990     No. Failure Samples=
1

                 d(p)         d(p)         d(p) sig     d(p)  sig
  Random        -----        ------       -----*---    ------*---
 Variable       d(mu)        d(sig)       d(mu)  p     d(sig)  p
```

DUMPSF

```
-----------------------------------------------------------------------
-
 KIC          0.9980E-06   -0.1071E-05    0.5983        -0.6421
 AI          -0.1033E-01    0.1051E-01    -5.159          5.251
 C           -0.4578E+06   -0.4766E+06   -0.5489        -0.5714
 DS          -0.2802E-05    0.4642E-05    -2.800          4.638


=======================================================================
=
 Level= 18 Z0= 74538.9     CDF=0.999999     No. Failure Samples=
1

                 d(p)          d(p)          d(p) sig     d(p)  sig
  Random        -----         ------        -----*---    ------*---
 Variable       d(mu)         d(sig)        d(mu)  p     d(sig)  p
-----------------------------------------------------------------------
-
 KIC          0.9992E-07   -0.1072E-06    0.5983        -0.6421
 AI          -0.1034E-02    0.1052E-02    -5.159          5.251
 C           -0.4584E+05   -0.4771E+05   -0.5489        -0.5714
 DS          -0.2805E-06    0.4648E-06    -2.800          4.638


=======================================================================
=
 Level= 19 Z0= 74538.9     CDF= 1.00000     No. Failure Samples=
1

                 d(p)          d(p)          d(p) sig     d(p)  sig
  Random        -----         ------        -----*---    ------*---
 Variable       d(mu)         d(sig)        d(mu)  p     d(sig)  p
-----------------------------------------------------------------------
-
 KIC          0.1000E-07   -0.1074E-07    0.5983        -0.6421
 AI          -0.1035E-03    0.1054E-03    -5.159          5.251
 C            -4589.        -4777.       -0.5489        -0.5714
 DS          -0.2809E-07    0.4653E-07    -2.800          4.638


=======================================================================
=
 Level= 20 Z0= 74538.9     CDF= 1.00000     No. Failure Samples=
1

                 d(p)          d(p)          d(p) sig     d(p)  sig
  Random        -----         ------        -----*---    ------*---
 Variable       d(mu)         d(sig)        d(mu)  p     d(sig)  p
-----------------------------------------------------------------------
-
 KIC          0.1002E-08   -0.1075E-08    0.5983        -0.6421
 AI          -0.1037E-04    0.1055E-04    -5.159          5.251
 C            -459.5        -478.3       -0.5489        -0.5714
 DS          -0.2812E-08    0.4659E-08    -2.800          4.638

• • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
   STOP DUE TO FPI ANALYSIS COMPLETE
```

DUMPSF

```
ELAPSED CPU TIME:          0.76 seconds
```

▲

FPI

APPENDIX IV-C: NEW FILES AND SUBROUTINES FOR NESSUS LHS ENHANCEMENT

FILE (1): lhs_main.f90

```
    SUBROUTINE LHS_MAIN()
!
!==================================================================
! Latin Hypercube Main Entrance
!==================================================================
!
! Cody Godines, September 2001
!==================================================================
!
!
!***********     TEMP NOTES ********************************
!
!padef%method=LHS, this is what I chose!
!NESSUS Files Changed:
!========                          ==============
!File                              Reason Changed
!========                          ==============
!
!new_nessus.f90                    Call to LHS_MAIN instead of FPI
!process_padefine.f90          To read in sampling info if LHS
!stuff_commons.f90                CASE ('LHS'), at begining of routine
!master_param.f90                 One place where unit numbers are defined globally
!                                 Long/Short string length
!nessus_derived_types.f90          Type messages and calculation
!init_input.f90                   To initialize new variables in nessus_derived_types.f90
!                                 Use master_param.f90
!intint.f                         To open lhs sample files
!                                 Use master_param.f90, so removed:
!                                  PARAMETER (MRANV=100,MGFUN=20) .and. PARAMETER (MPERT=201)
!                                 due to name conflicts with those accessible by Use statement.f
!monte.f                          BUG_"skip"
!mapdist.f                        Use nessus_derived_types.90, so that its dummy arguments can point to
!                                 the variables in nessus_derived_types.f90
!                                 Use master_param, so removed: parameter (MRANV=100,MGFUN=20)
!stuff_commons.f90                2nd select case for method is inside a case(mv,amv,amv+)
!                                 and contains cases for monte, user, lhs, ...etc
!                                 MOVED it outside of 1st case.  It was obviously written for a reason.
!                                 THEN had to REMOVE RETURN statement that was inside the 1st select case
!                                 because some of the calculations in the 2nd select case would never have been
!                                 performed.
!                                 BUG_"idist"
!intint.f                         Open file_means, file_sd, file_q99 for STUDY
!master_param.f90                 file_means, file_sd, file_q99 for STUDY
!                                 max_corrDiag
!nessus_derived_types.f90         sample_corr(max_corrDiag)
!init_input.f90                   initialize sample_corr(max_corrDiag)
!gcoeff.f                         Output explanation of origin of approximate statistics for Z.
!stuff_commons.f90                Assign xmean, xdev, and iname.
!stuff_commons.f90                Around line 777, it was beta_factor=1.0, changed to 0.0, see BUG_"radius".
!                                 In write_fpi_deck subroutine, include .or.trim(padef%method)=='LHS' around line 772.
!                                 SET common /trunc/ tlower(RV#) and tupper(RV#) to rv_def(RV#)%lower and upper
!                                 around line 573, in the case of 'MAXIMUM ENTROPY'
!                                 Mapdist needs it (also the lhs thread).  It is assigned in the Monte thread in inranv.f.
!                                 CHANGED xupper(RV#) = dble(rv_def(RV#)%upper)  was %lower and not double
!                                 CHANGED xlower(RV#) = dble(rv_def(RV#)%lower)  was not double
!                                 Does not assign ADJ(J) correctly!!@#
!
!NESSUS Files Added:
!========                          ==============
!File                              Reason Added
!========                          ==============
!lhs_main.f90                     To break out of the *NESSUS monte thread right before the fpi subroutine.
```

```
!lhs_xsample.f90                To obtain samples used for calculations.
!lhs_calc.f90                   To do lhs calculations.
!write_files.f90                To have a single subroutine that writes to files
!error_files.f90                Single subroutine to write error messages
!calc_statistics.f90            Has subroutines that calculate needed statistics.
!corr_control.f90               To transform sample to one with desired correlation.
!
!
!==========                     ==============
!Item                           Comment
!==========                     ==============
!GUI method                     Writes padef%method=HYPER to dat file, but error
!                               calls in stuff_commons.f90 are for LHS
!                               calls in other routines...
!                               I wrote LHS in .dat file.
!INPUT ECHO                     Now echoes the new dat file for LHS.
!BUG_"skip"                     MC1000, no sensitivities if ixskip or iuskip != 1.
!BUG_"skip"                     Nothing in .smx or .smu  if ixskip or iuskip != 1.
!BUG_"skip".fixed               monte.f :: (mod(I,IUSKIP), changed I to icount
!DIST(i)                        DIST(i) is of type real*8 (double) , rv_def(i)%idist is of type INTEGER, only a comment
!                               but info was used inside stuff_commons for BUG_"idist"
!BUG_"idist"                    rv_def(i)%idist never assigned in a *NESSUS, *MONTE thread
!BUG_"idist"                    Added SELECT CASE (trim(padef%method))
!                               CASE('MONTE','LHS','MV','AMV','AMV+') to stuff_commons.f90
!                               within that I moved SELECT CASE (trim(rv_def(i)%dist)) from the
!                               CASE('MV','AMV','AMV+')
!                               which was inside of the DO i=1,global%numrv loop.
!                               That is why 'MV','AMV',and,'AMV+' is also there.
!                               'LHS' is there because it also needs it.
!                               Then added rv_def(i)%dist and dist(i) assignments.
!                               This also seemed the appropriate place to assign dist(i)
!                               I need it because mapdist.f is called in a *NESSUS *LHS thread
!                               It is assigned in the inranv.f subroutine, which *LHS never reaches,
!                               and the *NESSUS, *MONTE thread would get to it by:
!                               new_nessus->fpi->fsetu1->fsetup->redprm->redmod->inranv
!                               dist(i) assigned inside the IF(DINAD(1:4).EQ.DINAM(JJ) around line 86.
!                               Be careful...go to the CASE('MV','AMV','AMV+'), there might be a possible incorrect
!                               reassign of dist(i) inside the same do loop, but it gets reassigned later in the previously
!                               mentioned way.
!BUG_"idist".fixed
!BUG_"seed"                     padef%seed is REAL*4,
!                               But the argument in random(padef%seed) must be double precision.
!                               Transfering data form *4 to *8 to *4 , then when transfering to *8, truncation happens
!                               and the seeds are no longer the same.  The random number generator is extremely
!                               sensitive to changes in the seed.
!                               padef%seed is defined: in process_padefine->parser.f, where it must be REAL*4
!BUG_"seed".fixed               Added padef%dseed to nessus_derived_types.f90, initialized it in init_input.f90
!                               and later in process_padefine.f90.
!                               comp%temp_dble(mranv,2) added to nessus_derived_types.f90, need a double
(temporary fix)
!SMX Correlation                It seems, from monte.f aroung line 551, if there is correlation
!                               that the .smx file's input vector is changed and not the zlevels
!BUG_"corr1"                    If correlation, then random variable statistics printed in monte.f are incorrect.
!                               ..modified name "T_" and mean , std are not good.  Source is probably cmpfpi.f.
!BUG_"rv_def"                   If you use gui to enter mean / std.d. and have too many digits, they get printed
!                               together in the *NESSUS dat file.  Error when trying to run.  See
!                               SAE7NC1BUGinRVDEFINE.
!BUG_"problem_statement"        If you enter g-function equations that go past the size of the problem statement window
!                               and save the work, the parts that went past the window will be lost.
!BUG_"open_error"               Parsing error, see and try to open CASE10/BUG_OPEN_ERROR_SAE1NC1.dat
!                               Work around, move *MODEL analytical_2.. underz10+underz11 to separate line.
!                               Original line is only 80 characters but how does parser et. al. work..
!BUG_"corr2"                    LHS Correlation, in stuff commons, the dependent variable loop goes from 1 to zero,
!                               therefore, error messages are automatic.  Random variable not found in
ANALYTICAL_1....
!                               >>global%numrv and global%numdv does not get if certain number of correlations are
!                               present.
!                               See CASE1\SAE1NC1LGOOD.txt and CASE1\SAE1NC1LERROR.txt and notice that
the
!                               only difference
```

```
!                                   is that one correlation is added.  Goto process_rvdefine.
!                                   WorkAround=>      GOOD                        ERROR (the space after ai)
!                                                     ai, c, 0.0                  ai , c, 0.0
!BUG_"radius"                       MC scodefpi.dat file has radius=1.
!BUG_"radius".fixed          stuff_commons.f90, around line 777, it was beta_factor=1.0, changed to 0.0
!BUG_"approx_stat"           Approximate statistics in gcoeff.f for igform=6
!                                   MIGHT be incorrect, if the form of the approximation is:
!                                   exactly like it was written in the senstv subroutine
!                                   g=c0+Sum{c(k,1)(xk-xk0)}+   FIRST ORDER PART
!                                   Sum_ij{cmix(i,j)(xi-xi0)(xj-xj0)}NEGLECT 2nd order terms
!                                   Then
!                                   mean_g = c0-Sum{c(k,1)xk0}+Sum{c(k,1)xk0}
!                                   mean_g = c0 = g(mean)
!WHOOPS!!!
!BUG_"cody_bug1"                    rv_def(i)%mapping(j)%blocks in echo_input  is zero at the time of subroutine use.
!                                   infinite do loop
!                                   WRITE(*,*) 'LINE_START     LINE_END'  + other stuff
!BUG_"cody_bug1".tempfix            ??  get rid of rv_def%median in nessus_derived_types and init_input
!BUG_"ADJ(J)"                       Stuff_commons does not assign it correctly.  Monte thread assigns it way down the
!                                   line in the inranv.f subroutine...
!BUG_"ADJ(J)".fixed          add ADJ(J) = 1.0 in stuff_commons.f90 around line 574
!
!
!***********  END TEMP NOTES ********************************
!
!0  Revision log:
!0          Initial programming LANL NESSUS 2.4 dsr
!0

!1  Purpose:
!1          To perform latin hypercube sampling, finish the
!1          analysis and stop the program.

!2  Calling Argument Input:
!2          None
!2

!3  Calling Argument Output:
!3          None
!3

!4
!4  Internal Variables and Arrays
!4

!5  Used by:
!5  new_nessus.f90

!6  Routines called:
!6  None

!7  Modules Used
!7  master_param.f90
!7  nessus_derived_types.f90

!8   Assumptions and limitations
!8   Only for plevels analysis.
!8   No confidence checks.
!8   Only component analysis.

!9
!9  COPYRIGHT 1998 BY SOUTHWEST RESEARCH INSTITUTE, SAN ANTONIO, TEXAS
!9
!
!  Declare Modules (Global)
!
            USE nessus_derived_types
            USE master_param

!           Declare calling arguments
```

```
!           Declare local variables
double precision :: x_variable_array(padef%isamp,global%numrv)

! This should be done in read_nessus_input.f90, up to *should
  mssg%description="input_echo"
            mssg%files(1:1)=(/file_out/)
            call write_files()
            mssg%description="lhs_header"
            mssg%files(1:4)=(/file_lhs_x_rdm,file_lhs_p_rdm,file_lhs_x_corr,file_lhs_p_corr/)
            call write_files()
! *should

            call lhs_xsample(x_variable_array)


! Send off for calculations.
            mssg%description = "output_header"
            mssg%files(1:2) = (/file_out, file_consl/)
            call write_files()

  call lhs_calc(x_variable_array)



            STOP
            RETURN
            END
```

FILE (2): lhs_xsample.f90

```
        SUBROUTINE lhs_xsample(x_variable_array)
!
!==================================================================
! Latin Hypercube X-Sample Generation
!==================================================================
!
! Cody Godines, September 2001
!==================================================================
!
!
!9
!9  COPYRIGHT 1998 BY SOUTHWEST RESEARCH INSTITUTE, SAN ANTONIO, TEXAS
!9
!
!
!  Declare Modules (Global)
!
            USE nessus_derived_types
            USE master_param


!           Declare calling arguments
double precision :: x_variable_array(padef%isamp,global%numrv)
!           Declare local variables
integer rdm_int_array(padef%isamp,global%numrv)
double precision prob_variable_array(padef%isamp,global%numrv)



!           Fill ranom_int_array with random, uniformly, equal probability
!           non-repeating integers along the first dimension (padef%isamp)
do j=1, global%numrv
            call raniset(padef%isamp, rdm_int_array(1:padef%isamp,j))
end do
```

```
!           Generate Probability(0,1) Samples
!           Fill prob_variable_array
!           Fill x_variable_array
do i=1,padef%isamp

!           Probability(0,1) is percentage increase in probability from the random bin number begining.
!           This locates the cumulative probability for that RV, for the current bin.
!           Get u space coordinate by inverting the standard normal distribution.

            do j=1, global%numrv
                        prob_variable_array(i,j)=(dble(rdm_int_array(i,j)-1)+&
                                                &random(padef%dseed))/dble(padef%isamp)
                        calc%prob(j) = prob_variable_array(i,j)
            end do
!           Get x space coordinate by inverting respective distribution.
            calc%temp_dble(1:mranv,1)=(/(dble(rv_def(j)%mean),j=1,mranv)/)
            calc%temp_dble(1:mranv,2)=(/(dble(rv_def(j)%std) ,j=1,mranv)/)

            call mapdist(global%numrv,calc%prob,calc%temp_dble(1:mranv,2),&
                        &calc%sample_stat,calc%temp_dble(1:mranv,1),x_variable_array(i,1:global%numrv),rv_def%idist,ierr)

end do

!           Calculate Statistics of Samples
if(padef%isamp.le.3000) then                                                    ! TIME CONTROL L@@K
   call calc_stats(prob_variable_array)
            mssg%description="lhs_statistics"
            mssg%files(1:1)=(/file_lhs_p_rdm/)
            call write_files()


            call calc_stats(x_variable_array)
            mssg%description="lhs_statistics"
            mssg%files(1:1)=(/file_lhs_x_rdm/)
            call write_files()


!           Write random sample to files.
do  i=1,padef%isamp
                        calc%prob(1:global%numrv)=(/(prob_variable_array(i,j),j=1,global%numrv)/)
                        calc%x(1:global%numrv)=(/(x_variable_array(i,j),j=1,global%numrv)/)

                        if(i==1) mssg%big_string="section_header"
                        mssg%description="lhs_samples"
                        mssg%files(1:2)=(/file_lhs_x_rdm,file_lhs_p_rdm/)
                        call write_files()
end do


!           CORRELATION CONTROL
!           Transform to new sample set with desired correlation
            call corr_control(x_variable_array)
!           Get probability array that is the CDF of the individual variables of the x_variable_array.
            ! alpha...beta   *****<<<<<< WATCH IT FOR THE NEEDED PARAMETERS
do  i=1,padef%isamp
            do j=1,global%numrv
                        call CDFPDF(1.0d0,1.0d0,dble(rv_def(j)%idist),x_variable_array(i,j),&
                        &calc%temp_dble(j,1),calc%temp_dble(j,2),1,prob_variable_array(i,j),pdf_junk,j)
            end do
end do

!           Calculate Statistics of New Samples
   call calc_stats(prob_variable_array)
            mssg%description="lhs_statistics"
            mssg%files(1:1)=(/file_lhs_p_corr/)
            call write_files()



            call calc_stats(x_variable_array)
```

```
            mssg%description="lhs_statistics"
            mssg%big_string="desired"
            mssg%files(1:1)=(/file_lhs_x_corr/)
            call write_files()


!           Write correlated sample to files.
do  i=1,padef%isamp
                        calc%prob(1:global%numrv)=(/(prob_variable_array(i,j),j=1,global%numrv)/)

                        if(i==1) mssg%big_string="section_header"
                        mssg%description="lhs_samples"
                        mssg%files(1:1)=(/file_lhs_p_corr/)
                        call write_files()
end do

end if                                                                          !   END    TIME    CONTROL
L@@K

! Because output contains sample statistics and if padef%isamp > 3000 then those statistics
!           are never calculated, but there is a residual in calc%sample_stat left over from map_dist()
!           above.
if(padef%isamp.gt.3000) then                                                    ! TIME CONTROL L@@K
            do j=1,global%numrv
                                    calc%sample_stat(j,1)= calc%sample_stat(j,1)/dble(padef%isamp)
                                    calc%sample_stat(j,2)= SQRT((1.0d0/(dble(padef%isamp)-1.0d0))*&
                                    &(calc%sample_stat(j,2)-dble(padef%isamp)*calc%sample_stat(j,1)**2))
            end do
end if                                                                          !   END    TIME    CONTROL
L@@K


RETURN
END SUBROUTINE lhs_xsample
!==================================================================
!==================================================================
    subroutine raniset( n, iset )
!==================================================================
!Randall Manteufel, 2001, UTSA
!randomnly fills the integer array with values from 1 to n
!the array is of length n, each entry has a unique value
!and all values from 1 to n are in the array/set one and only
!one time
!
!n     = input, integer, length of array values
!iset(n)   = output, integer, array of values
!
    implicit double precision (a-h,o-z)
    dimension iset(n)

      do i2 = 1, n
        iset(i2) = i2
      enddo
! extra loop to ensure randomness, ijunk =2
      do ijunk = 1,2
        do i2 = 1, n
          k  = iranu(1,n)
          ik = iset(k)
          iset(k) = iset(i2)
          iset(i2)  = ik
        enddo
      enddo


    return
    end subroutine raniset
```

```
!==========================================================
!==========================================================
        function iranu(ilow, ihigh)
!==========================================================
use nessus_derived_types
!Randall Manteufel, 2001, UTSA
! integer random sample from uniform pdf
!
! ilow = input, integer, low value of pdf
! ihigh = input, integer, high value of pdf
! iranu = output, integer, sampled value
!           Note:   ilow <= iranu <= ihigh

    nbin = ihigh - ilow +1
    iranu = ilow + int( random(padef%dseed) * dble(nbin) )

    return
    end function iranu
```

FILE (3): calc_statistics.f90

```
!This file calculates statistics of variables or sets of variables
SUBROUTINE calc_stats(variable_array)

USE nessus_derived_types
USE master_param
!  Declare Calling Variables
double precision :: variable_array(padef%isamp,global%numrv)

!  Declare Local Variables
double precision :: variable_vector1(padef%isamp)
double precision :: variable_vector2(padef%isamp)

!integer :: last_header_line, skip_line
!last_header_line=7
!  READ IN VALUES
!igtotal = global%numrv
!do k=1, count(mssg%files.ge.0)
!           do i=1, padef%isamp
!                       rewind(mssg%files(k))
!                       skip_line=last_header_line+(i-1)
!                       read(mssg%files(k),'( <skip_line>/,<igtotal>(e20.10 e3,4x) )') (variable_array(i,j),j=1,global%numrv)
!           end do


!  ACTION with VALUES
!  Mean and Standard Deviation of Samples
           calc%sample_stat(1:mranv,1:2)  = 0.0d0
           do j=1,global%numrv
              do i=1,padef%isamp
                       calc%sample_stat(j,1) = calc%sample_stat(j,1)+variable_array(i,j)
```

```
        calc%sample_stat(j,2) = calc%sample_stat(j,2)+variable_array(i,j)**2
                end do
                        calc%sample_stat(j,1)= calc%sample_stat(j,1)/dble(padef%isamp)
                        calc%sample_stat(j,2)= SQRT((1.0d0/(dble(padef%isamp)-1.0d0))*&
                        &(calc%sample_stat(j,2)-dble(padef%isamp)*calc%sample_stat(j,1)**2))
        end do

! Correlation
!
!       Normal data (at least one variable normal) use correlation coefficient, r.
!       Non-normal data, use Spearman rank correlation.
! Check for monotonic relation between two variables.
!               X2>X1 => Y2>=Y1 for monotonic increase.
!               X2>X1 => Y2<=Y1 for monotonic decrease.

! Correlation Coefficient (Linear, Pearsons), will only work if there is a linear relationship between the variables.
!       A measure of how close data resembles a straight line.  Could have perfect prediction without a straight line.
        calc%sample_corr(1:max_corrDiag)= 0.0d0
        i_sample_corr_list=0
        ! For all combinations
        do i=1,global%numrv
                do j=1,i
                        i_sample_corr_list=i_sample_corr_list+1
                        !Calculations
                        do in=1,padef%isamp
                                calc%sample_corr(i_sample_corr_list)= calc%sample_corr(i_sample_corr_list)&
                                &+(variable_array(in,i)-calc%sample_stat(i,1))*(variable_array(in,j)-
calc%sample_stat(j,1))&
                                &/(dble(padef%isamp)*calc%sample_stat(i,2)*calc%sample_stat(j,2))
                        end do
                end do
        end do

!       Correlation Coefficient ( Spearman rank )
!       Spearman rank correlation coefficient.
!               Distribution free correlation analysis.  Can work on discrete or continuous data (like regression), but
!               works on ranked (relative) data (use rank-order numbers).       A Spearman's rs coefficient close to one
indicates
!               good agreement and
!               close to zero, poor agreement.  It is similiar to the R^2 value of regression.  No assumptions are made
!               about the distribution of the underlying data.
!
!       Spearman's method works by assigning a rank to each observation in each group separately
!       (contrast this to rank-sum methods in which the ranks are pooled). Then calculate the sums of the squares
!       of the differences in paired ranks (di^2) according to the formula:
!         rs = 1 - 6*(d1^2 + d2^2 + ... + dn^2)/(n(n^2-1)),
!       in which n is the number of observations.
!       d1=rank(X1)-rank(Y1)
!       rank of X1 is an integer between 1 and the number of observations.  Rank(X1)=1 if X1 is the smallest
!        value in the set of all X's.  Rank(X1)=#observations if X1 is the largest in the set of all X's.
!
!       Yes, the value indicates the "strength" of the relation, but quantifying the strength is complex.
!       Therefore, it is considered to be a non-parametric test.
!
!       The scale is ordindal.
!
! Significance?
        calc%spearman_corr(1:max_corrDiag)= 0.0d0
        i_spearman_corr_list=0
        ! For all combinations
        do i=1,global%numrv
                do j=1,i
                        i_spearman_corr_list=i_spearman_corr_list+1
                        !Calculations
                        call vector_rank(variable_array(1:padef%isamp,i),variable_vector1,padef%isamp)
                        call vector_rank(variable_array(1:padef%isamp,j),variable_vector2,padef%isamp)

                        do in=1,padef%isamp
                                calc%spearman_corr(i_spearman_corr_list)=
calc%spearman_corr(i_spearman_corr_list)&
```

```
                                &+(variable_vector1(in)-variable_vector2(in))**2
                        end do
                        calc%spearman_corr(i_spearman_corr_list)=1.0d0-
6.0d0*calc%spearman_corr(i_spearman_corr_list)&
                        &/(padef%isamp*(padef%isamp**2-1.0d0))
                end do
        end do


!  ACTION END
!end do
RETURN
END SUBROUTINE calc_stats
```

SUBROUTINE vector_rank(var_vector, rank_vector, length)

```
! Declare calling arguments
integer length
double precision :: var_vector(length)
double precision :: rank_vector(length)
! Declare local variables
double precision :: temp_vector(length), larger_than

temp_vector=var_vector
larger_than=2*dsign(maxval(temp_vector,DIM=1),1.0d0)
do i=1,length
        min_loc=minloc(temp_vector,DIM=1)
        rank_vector(min_loc)=dble(i)
        temp_vector(min_loc)=larger_than
end do

RETURN
END SUBROUTINE vector_rank
```

SUBROUTINE VECTOR_STATS(VECTOR, LENGTH)

```
Use nessus_derived_types

! declare calling variables
double precision vector(length)
integer length
! declare local variables

calc%z_sample_stat(1:2)=0.0d0
  do i=1,length
          calc%z_sample_stat(1) = calc%z_sample_stat(1)+vector(i)
   calc%z_sample_stat(2) = calc%z_sample_stat(2)+vector(i)**2
        end do
calc%z_sample_stat(1)= calc%z_sample_stat(1)/dble(length)
calc%z_sample_stat(2)= DSQRT((1.0d0/(dble(length)-1.0d0))*&
&(calc%z_sample_stat(2)-dble(length)*calc%z_sample_stat(1)**2))

write(104,'(e25.10 e3)') calc%z_sample_stat(1)          !cody_ADD begin/end
write(105,'(e25.10 e3)') calc%z_sample_stat(2)          !cody_ADD begin/end

RETURN
END SUBROUTINE vector_stats
```

FILE (4): corr_control.f90

```
!       Obtain desired correlation from a sample set.
!       ________ REFERENCE ____________________
!       Iman, R.L. and Conover, W.J. (1982)
!       A Distribution-Free Approach to Inducing Rank Correlation Among Input Variables
!       Kraus, Allan D. (1987)
!       Marices For Engineers (Cholesky Decomposition)
!  ____________________________________
```

SUBROUTINE corr_control(variable_array)

```
USE nessus_derived_types
USE master_param
!  Declare Calling Variables
double precision :: variable_array(padef%isamp,global%numrv)


!  Declare Local Variables
double precision :: temp_corr_coef(corr_def%ncor)
double precision :: corr_desired(global%numrv,global%numrv)
double precision :: Plow(global%numrv,global%numrv)
!double precision :: Pupp(global%numrv,global%numrv)  !cody Temp Add debug turn off
double precision :: rstar(padef%isamp,global%numrv)
double precision :: r_scores(padef%isamp,global%numrv)
double precision :: variable_vector1(padef%isamp)
double precision :: variable_vector2(padef%isamp)
integer :: random_vector(padef%isamp)


corr_desired(1:global%numrv,1:global%numrv)=0.0d0

!   REARRANGE DESIRED CORRELATIONS to be in the proper order.
!   Lower half of a correlation matrix, not including the diagonal,
!            from top to bottom, left to right.

if(corr_def%ncor==0) then
          do i=1,global%numrv
                    corr_desired(i,i) =1.0d0
          end do
else
          temp_corr_coef(1:corr_def%ncor)=corr_def%coef(1:corr_def%ncor)
          corr_def%coef(1:max_corr) = 0.0d0
end if
do i_corr_list=1, corr_def%ncor
          do i_rv_num=1,global%numrv
           if(trim(corr_def%rv(i_corr_list,1))==trim(rv_def(i_rv_num)%name) )  then
                    i_row_num=i_rv_num
           end if
           if(trim(corr_def%rv(i_corr_list,2))==trim(rv_def(i_rv_num)%name) )  then
             i_col_num=i_rv_num
           end if
          corr_desired(i_rv_num,i_rv_num)=1.0d0
          end do
          if(i_col_num.gt.i_row_num) then
                    i_temp_num=i_col_num
                    i_col_num=i_row_num
                    i_row_num=i_temp_num
          end if
          i_corr_list_mapped=1
          do i_to_row=2,i_row_num
                    do i_to_col=1,i_to_row-1
                              if(i_to_row.ne.i_row_num) i_corr_list_mapped=i_corr_list_mapped+1
                    end do
          end do
          i_corr_list_mapped=i_corr_list_mapped+(i_col_num-1)
          corr_def%coef(i_corr_list_mapped)=temp_corr_coef(i_corr_list)
          corr_desired(i_row_num,i_col_num)=corr_def%coef(i_corr_list_mapped)
          corr_desired(i_col_num,i_row_num)=corr_def%coef(i_corr_list_mapped)
end do
!
!
!
!
!  THEORETICAL.
!                    X  uncorrelated with correlation matrix, I.
!                    C  is desired correlation matrix.  Positive definite and symmetric.
!                       Therefore, it may be written C=PP'.
!                    P  is a lower triangular matrix.
```

```
!                       XP' has desired correlation matrix, C.

! APPLICATION
!           Check to see if desired correlation matrix is positive definite. Not yet.
!

! Accept the desired correlation matrix [C] sample set, given by the user
! to be the target rank correlation matrix [C*] of the sample set.
!           C* = C
!
! Cholesky factorization scheme.  Obtain lower triangular matrix, P, such that
! C = PP'
!
Plow(1:global%numrv,1:global%numrv)=0.0d0
do i=1,global%numrv
            do j=1,i

                        if((i==1).and.(j==i)) then
                                    Plow(i,j)=dsqrt(corr_desired(i,j))
                        else if (j.ne.i) then
                                    Plow(i,j)=corr_desired(i,j)
                                    do m=1,j-1
                                                Plow(i,j)=Plow(i,j)-Plow(j,m)*Plow(i,m)
                                    end do
                                    Plow(i,j)=Plow(i,j)/Plow(j,j)
                        else if((i.ne.1).and.(j==i)) then
                                    Plow(i,j)=corr_desired(i,j)
                                    do m=1,j-1
                                                Plow(i,j)=Plow(i,j)-Plow(j,m)*Plow(i,m)
                                    end do
                                    Plow(i,j)=dsqrt(Plow(i,j))
                        end if

            end do
end do

!
!
!
!           Begin with R, with rank correlation matrix, I.
!             :: Use Van der Waerden scores
!
!           Transform RP' = R*.
!           The rank correlation matrix M of R* would be close to C* = C.
            r_scores(1:padef%isamp,1:global%numrv)=0.0d0
do j=1, global%numrv
            call raniset(padef%isamp,random_vector)
            do i=1, padef%isamp
                        r_scores(random_vector(i),j)=xinv(dble(i)/(dble(padef%isamp+1)))
            end do
end do


!           mulltt multiplies the second array by the transpose of the third array and returns the first
            rstar(1:padef%isamp,1:global%numrv)=0.0d0
            call multt(rstar,r_scores,Plow,padef%isamp,global%numrv,global%numrv)



!
! Reorder columns of X (individual variables) to have same rank as R*.
!           Thus, X will have the same rank matrix as R*.
!           X will also have rank correlation matrix, M, close to C* = C.
do j=1,global%numrv

            call vector_rank(variable_array(1:padef%isamp,j),variable_vector1,padef%isamp)
            call vector_rank(rstar(1:padef%isamp,j),variable_vector2,padef%isamp)

            do i=1,padef%isamp
                        do i_rstar=1,padef%isamp
```

```
                    if(int(variable_vector1(i)+0.1d0)==int(variable_vector2(i_rstar)+0.1d0)) then
                        rstar(i_rstar,j)=variable_array(i,j)
                    end if
                end do
            end do

end do
variable_array=rstar

RETURN
END SUBROUTINE corr_control
```

FILE (5): lhs_calc.f90

```
        SUBROUTINE lhs_calc(x_variable_array)
!
!==========================================================================
! Latin Hypercube X-Sample Calculations
!==========================================================================
!
! Cody Godines, September 2001
!==========================================================================
!
!
!***********     TEMP NOTES ********************************
!
!
!
!===========                                    ===============
!Item                                           Comment
!===========                                    ===============
!
!
!***********     END TEMP NOTES ****************************
!
!0  Revision log:
!0          Initial programming LANL NESSUS 2.4 dsr
!0

!1  Purpose:
!1          To perform latin hypercube sampling, finish the
!1  analysis and stop the program.

!2  Calling Argument Input:
!2          None
!2

!3  Calling Argument Output:
!3          None
!3

!4
!4  Internal Variables and Arrays
!4

!5  Used by:
!5  new_nessus.f90

!6  Routines called:
!6          None

!7  Modules Used
!7           master_param.f90
!7   nessus_derived_types.f90
```

```
!8   Assumptions and limitations
!8    Only for plevels analysis.
!8    No confidence checks.

!9
!9  COPYRIGHT 1998 BY SOUTHWEST RESEARCH INSTITUTE, SAN ANTONIO, TEXAS
!9
!
!   Declare Modules (Global)
!
            USE nessus_derived_types
            USE master_param


!           Declare calling arguments
double precision :: x_variable_array(padef%isamp,global%numrv)
!           Declare local variables
double precision :: z_vector(padef%isamp)
double precision :: z_sorted(padef%isamp)


do i=1,padef%isamp
            call evaluate_models(file_out,0,x_variable_array(i,1:global%numrv),z_vector(i),ierr)
            calc%x(1:global%numrv)=(/(x_variable_array(i,j),j=1,global%numrv)/)
            calc%z = z_vector(i)

            if(i.le.3000) then                          !TIME CONTROL L@@K
                        if(i==1) mssg%big_string="section_header"
                        mssg%description="lhs_samples"
                        mssg%files(1:1)=(/file_lhs_x_corr/)
                        call write_files()
            end if                                      !TIME CONTROL L@@K
end do

call vector_stats(z_vector(1:padef%isamp), padef%isamp)

mssg%description="output_statistics"
mssg%files(1:2)=(/file_out, file_consl/)
call write_files()

z_sorted=z_vector
call qsort(padef%isamp, z_sorted)

do i=1,padef%nlevels
            calc%prob(1) = cdfnof(dble(padef%levels(i)))
            nfind = int(dble(padef%isamp)*calc%prob(1)+0.50000000d0)
            calc%temp_dble(1,1) = dble(nfind+0.01d0)
            calc%temp_dble(1,2) = dble(padef%levels(i))
   calc%z=z_sorted(nfind)

            if ((calc%prob(1).ge. 0.990).and.(calc%prob(1).le.0.991)) then                    !cody_ADD
begin/end
                        write(106,'(e25.10 e3)') calc%z
                        !cody_ADD begin/end
            end if
                                                                        !cody_ADD begin/end
            if(i==1) mssg%big_string="section_header"
            if(i==padef%nlevels) mssg%big_string="section_end"
            mssg%description="output_cdf"
            mssg%files(1:2)=(/file_out, file_consl/)
            call write_files()
end do


RETURN
END SUBROUTINE lhs_calc
```

FILE (6): write_files.f90

SUBROUTINE WRITE_FILES()

```
!=================================================================
! Write_Files
!=================================================================
!
! Cody Godines, September 2001
!=================================================================
!
!
!
!0  Revision log:
!0          Initial programming LANL NESSUS 2.4 dsr
!0

!1  Purpose:
!1          To write any file or screen output
!1

!2  Calling Argument Input:
!2          None
!2

!3  Calling Argument Output:
!3          None
!3

!4
!4  Internal Variables and Arrays
!4

!5  Used by:
!5   Any subroutine requiring amessage written to file(s)

!6  Routines called:
!6          None

!7  Modules Used
!7
!7

!8   Assumptions and limitations
!8
!8

!9
!9  COPYRIGHT 1998 BY SOUTHWEST RESEARCH INSTITUTE, SAN ANTONIO, TEXAS
!
USE master_param
USE nessus_derived_types
!           Declare calling arguments
!
!           Declare local variables
!
double precision :: corr_desired(global%numrv,global%numrv)
!  Begin
!  Write for each file in mssg%files that has been assigned a value >=0.
do j=1, count(mssg%files.ge.0)
  rewind file_dat
select case (trim(mssg%description))

            case("input_echo")
            write(mssg%files(j),'("1",78("="),/,27X,10("*"),"  ","INPUT ECHO","  ",10("*"),/,79("="),//)')
            write(mssg%files(j),'(" LINE",/)')
            do 130 i=1,200000
                        read(file_dat,'(A)',end=130) mssg%big_string
```

```
            write(mssg%files(j),'(1X,I6,3X,A)') i, mssg%big_string
        130 continue
        write(mssg%files(j),'(//,79("="),/,79("="))')



        case("lhs_header" )
         write(mssg%files(j),200) global%title, padef%isamp, global%numrv, global%num_zfmodel
         200        format("# Latin Hypercube Sampling Matrix File",/,&
         &          "# JOBID: ",A<len_trim(global%title)>,/,&
         &          "# For each row(1:# Samples =",I10,") : Input_Vector(1:#RVs =",I3,") GNFS(1:#GFNS=",I3,")")

         if   (mssg%files(j)==100 .or. mssg%files(j)==101)then
          write(mssg%files(j) ,'("# These are RANDOM SAMPLES with SPURIOUS CORRELATION between variables.")')
         elseif (mssg%files(j)==102 .or. mssg%files(j)==103)then
          write(mssg%files(j) ,'("# These are RANDOM SAMPLES with ADJUSTED CORRELATION between variables.")')
         endif

         if   (mssg%files(j)==100) then
          write(mssg%files(j) ,'("# LHS_X_SAMPLES :: LHS_PROB_SAMPLES(0,1) then INVERT RESPECTIVE PDF",//)')
         elseif (mssg%files(j)==101) then
          write(mssg%files(j) ,'("# LHS_PROB_SAMPLES :: Randomly sample from each probability bin and randomly pair up
              coordinates",//)')
         elseif (mssg%files(j)==102) then
          write(mssg%files(j) ,'("# LHS_X_SAMPLES :: DECOMPOSE random LHS_X_SAMPLES to yield samples with
desired
              correlation",//)')
         elseif (mssg%files(j)==103) then
          write(mssg%files(j) ,'("# LHS_PROB_SAMPLES :: LHS_X_SAMPLE adjusted for correlation and calculate cumulative
              probability",//)')
         endif


    case("lhs_statistics")
          igtotal = global%numrv

          write(mssg%files(j), '("MEAN of SAMPLE (by columns = random variable)")')
                write (mssg%files(j),'(<igtotal>(e20.10 e3,4x),/)' ) (calc%sample_stat(irv,1),irv=1,global%numrv)

          write(mssg%files(j), '("STANDARD DEVIATION of SAMPLE (by columns = random variable)")')
                write (mssg%files(j),'(<igtotal>(e20.10 e3,4x),/)' ) (calc%sample_stat(irv,2),irv=1,global%numrv)

          write(mssg%files(j), '("CORRELATION COEFFICIENT MATRIX (Linear) ")')
                i_sample_corr_list=1
                do i=1,global%numrv
                        write (mssg%files(j),'(<i>(e20.10 e3,4x))' ) &
                        &(calc%sample_corr(isc),isc=i_sample_corr_list,i_sample_corr_list+i-1)
                        i_sample_corr_list=i_sample_corr_list+i
                end do

                if(trim(mssg%big_string)=="desired") then
                        ! Map corr_def%coef to full matrix, to be used in write statement
                        ! Already been formatted to proper lower form.
                        i_corr_list=1
                        do i_corr=1,global%numrv
                                if (i_corr==1) then
                                        corr_desired(i_corr,j_corr)=1.0d0
                                else
                                        do j_corr=1,i_corr
                                                if(i_corr==j_corr) then
                                                        corr_desired(i_corr,j_corr)=1.0d0
                                                else
                                                        corr_desired(i_corr,j_corr)=corr_def%coef(i_corr_list)
                                                        corr_desired(j_corr,i_corr)=corr_desired(i_corr,j_corr)
                                                        i_corr_list=i_corr_list+1
                                                end if
                                        end do
                                end if
                        end do
```

```
                    write(mssg%files(j), '(/,"SPEARMAN RANK CORRELATION COEFFICIENT MATRIX \
DESIRED ")')
                    i_sample_corr_list=1
                    do i=1,global%numrv
                              write (mssg%files(j),'(<igtotal>(e20.10 e3,4x))' ) &
                              &(calc%spearman_corr(isc),isc=i_sample_corr_list,i_sample_corr_list+i-
1),(corr_desired(i,jdes),jdes=i+1,global%numrv)
                              i_sample_corr_list=i_sample_corr_list+i
                    end do
          else
                    write(mssg%files(j), '(/,"SPEARMAN RANK CORRELATION COEFFICIENT MATRIX ")')
                    i_sample_corr_list=1
                    do i=1,global%numrv
                              write (mssg%files(j),'(<i>(e20.10 e3,4x))' ) &
                              &(calc%spearman_corr(isc),isc=i_sample_corr_list,i_sample_corr_list+i-1)
                              i_sample_corr_list=i_sample_corr_list+i
                    end do
          end if

     case("lhs_samples")
      if(trim(mssg%big_string)=="section_header") write(mssg%files(j),'(/,"***** SAMPLES *****")')
      igtotal = global%numrv
      if   (mssg%files(j)==100) then          !lhs x random
                    write (mssg%files(j) , '(<igtotal>(e20.10 e3,4x))') &
                    &                 (calc%x(i),i=1,global%numrv)
      elseif (mssg%files(j)==101) then                !lhs p random
                    write (mssg%files(j) , '(<igtotal>(e20.10 e3,4x))') &
                    &                 (calc%prob(i),i=1,global%numrv)
      elseif (mssg%files(j)==102) then                !lhs x correlated
                    write (mssg%files(j) , '(<igtotal>(e20.10 e3,4x), e20.10 e3)') &
                    &                 (calc%x(i),i=1,global%numrv), calc%z
      elseif (mssg%files(j)==103) then                !lhs p correlated
                    write (mssg%files(j) , '(<igtotal>(e20.10 e3,4x))') &
                    &                 (calc%prob(i),i=1,global%numrv)
      endif

     case("output_header")
      write(mssg%files(j),'("1",78("="),/,27X,10("*")," "," OUTPUT SUMMARY"," ",10("*"),/,79("="),//)')
      write(mssg%files(j),'(/,A)')  "  LATIN HYPERCUBE SOLUTION  "
  write(mssg%files(j),'(A,I3)') "   NUMBER OF VARIABLES  ",global%numrv
  write(mssg%files(j),'(A,I9)') "   NUMBER OF SAMPLES  ",padef%isamp

     case("output_statistics")
      write(mssg%files(j),'(/,A)')  "  RANDOM VARIABLE STATISTICS  "
      write(mssg%files(j),'(2x,A6,5x,A5,11x,A5,9x,A6,11x,A6,8x,A7,7x,A7)') 'Random','Input','Input','Sample','Sample','%
error','% error'
      write(mssg%files(j),'(1x,A8,4x,A4,11x,A9,7x,A4,11x,A9,8x,A4,7x,A9)') 'Variable','Mean','Std. Dev.','Mean','Std.
Dev.','Mean','Std. Dev.'
      write(mssg%files(j),"(1x,<98>('-'))")
      do i=1,global%numrv
              write(mssg%files(j),'(1x,A8,2x,<6>(e12.6 e3, 3x))' )
rv_def(i)%name,rv_def(i)%mean,rv_def(i)%std,calc%sample_stat(i,1)&
              &,calc%sample_stat(i,2),dabs((1.0d0-dble(rv_def(i)%mean)/calc%sample_stat(i,1))*100.d0)&
              &,dabs((1.0d0-dble(rv_def(i)%std)/calc%sample_stat(i,2))*100.d0)
      end do
      write(mssg%files(j),'(/,A)')  "  RESPONSE STATISTICS  "
      write(mssg%files(j),'(3x,A8,17x,A8)') 'Response','Response'
      write(mssg%files(j),"(3x,A4,21x,A9,/,2x,<44>('-'))") 'Mean','Std. Dev.'
      write(mssg%files(j), '(<2>(e20.10 e3, 5x))' ) calc%z_sample_stat(1), calc%z_sample_stat(2)

  case("output_cdf")
      if(trim(mssg%big_string)=="section_header") then
              write(mssg%files(j),'(/,A)')  "  CDF SUMMARY  "
              write(mssg%files(j),'(5x,A8,12x,A,20x,A2,11x,A8,9x,A8)') 'Pr(Z<Z0)','U','Z0','#Pts<=Z0','Error(*)'
              write(mssg%files(j),"(<90>('-'))" )
      end if

      write(mssg%files(j) ,'(2x,<3>(e12.5 e3, 5x),I10,11x,e12.5 e3)') calc%prob(1), calc%temp_dble(1,2), calc%z, &
      & int(calc%temp_dble(1,1)), 9999.9
```

```
            if(trim(mssg%big_string)=="section_end") then
                    write(mssg%files(j),'(4x,A)') "(*) Sampling Error at 95% Confidence "
            end if



           case default

end select

end do

!           Rewind units
            rewind file_dat
!  Reinitialize global variables
            mssg%description     = REPEAT(" ",string_short)
            mssg%what_file                  = REPEAT(" ",string_short)
            mssg%after                              = REPEAT(" ",string_short)
            mssg%before                             = REPEAT(" ",string_short)
            mssg%big_string                 = REPEAT(" ",string_long )
            mssg%files(1:file_tot_num)      = -1
RETURN
END SUBROUTINE WRITE_FILES
```

FILE (7): error_files.f90

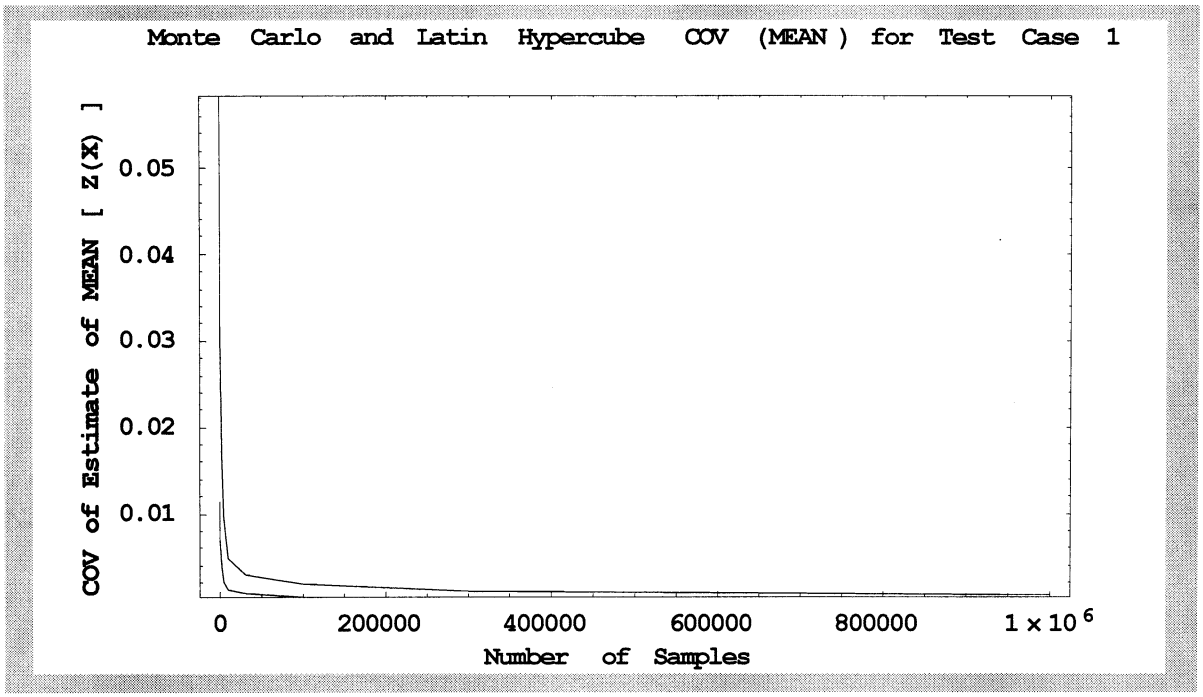
```
        SUBROUTINE ERROR_FILES(description,file,after,before)
!
!==================================================================
! Error Files
!==================================================================
!
! Cody Godines, September 2001
!==================================================================
!
!
!
!0  Revision log:
!0          Initial programming LANL NESSUS 2.4 dsr
!0

!1  Purpose:
!1          To write error messages to the respective files.
!1

!2  Calling Argument Input:
!2          description
!2   file
!2   after
!2   before
!2

!3  Calling Argument Output:
!3          None
!3

!4
!4  Internal Variables and Arrays
!4

!5  Used by:
!5   Any subroutine requiring an error message written to a file(s)

!6  Routines called:
!6          None
```

```
!7  Modules Used
!7
!7

!8   Assumptions and limitations
!8
!8

!9
!9  COPYRIGHT 1998 BY SOUTHWEST RESEARCH INSTITUTE, SAN ANTONIO, TEXAS
!9

USE master_param
! Declare calling arguments
character*16, intent(in) :: description, file, after, before


select case (trim(description))
        case("generic_stop")
                write(*,*) "STOP DUE TO ERROR :",trim(description)
                write(*,*) "File :",file
                write(*,*) "After :",after
                write(*,*) "Before :",before
                stop
        case default
                write(*,*) "STOP DUE TO ERROR :unknown"
                write(*,*) "File :",file
                write(*,*) "After :",after
                write(*,*) "Before :",before
                stop
end select


RETURN
END SUBROUTINE ERROR_FILES
```

APPENDIX IV-D: NONLINEAR REGRESSION BY LEAST SQUARES FOR LOG-LOG COV PLOTS

Non-Linear Models

$$\underline{y}(x) = cx^m$$



[Ratkowsky,p87] m<0 for that shape.

$$Log[\underline{y}(x)] = Log[cx^m] = mLog[x] + Log[c]$$

$$E = \sum_{j=1}^{r} \left(Log[\underline{y}(x_j)] - Log[\hat{y}_j] \right)^2$$

$$\frac{\partial E}{\partial c} = \sum_{j=1}^{r} 2\left(mLog\ [x_j] + Log\ [c] - Log\ [\hat{y}_j]\right)\frac{Log\ [e]}{c} = 0$$

where,

$$\frac{\mathrm{d}\{\mathrm{Log}_{\mathrm{a}}[\mathrm{c}]\}}{\mathrm{dc}} = \frac{\mathrm{Log}_{\mathrm{a}}[e]}{c}$$

$$m\sum_{j=1}^{r} Log[x_j] + rLog\ [c] - \sum_{j=1}^{r} Log[\hat{y}_j] = 0$$

$$m = \frac{\sum_{j=1}^{r} Log[\hat{y}_j] - rLog\ [c]}{\sum_{j=1}^{r} Log[x_j]}$$

$$\frac{\partial E}{\partial m} = \sum_{j=1}^{r} 2\left(mLog[x_j] + Log[c] - Log[\hat{y}_j]\right)Log[x_j] = 0$$

$$m\sum_{j=1}^{r}\left(Log[x_j]\right)^2 + Log[c]\sum_{j=1}^{r} Log[x_j] - \sum_{j=1}^{r} Log[\hat{y}_j]Log[x_j] = 0$$

The result of the m substitution

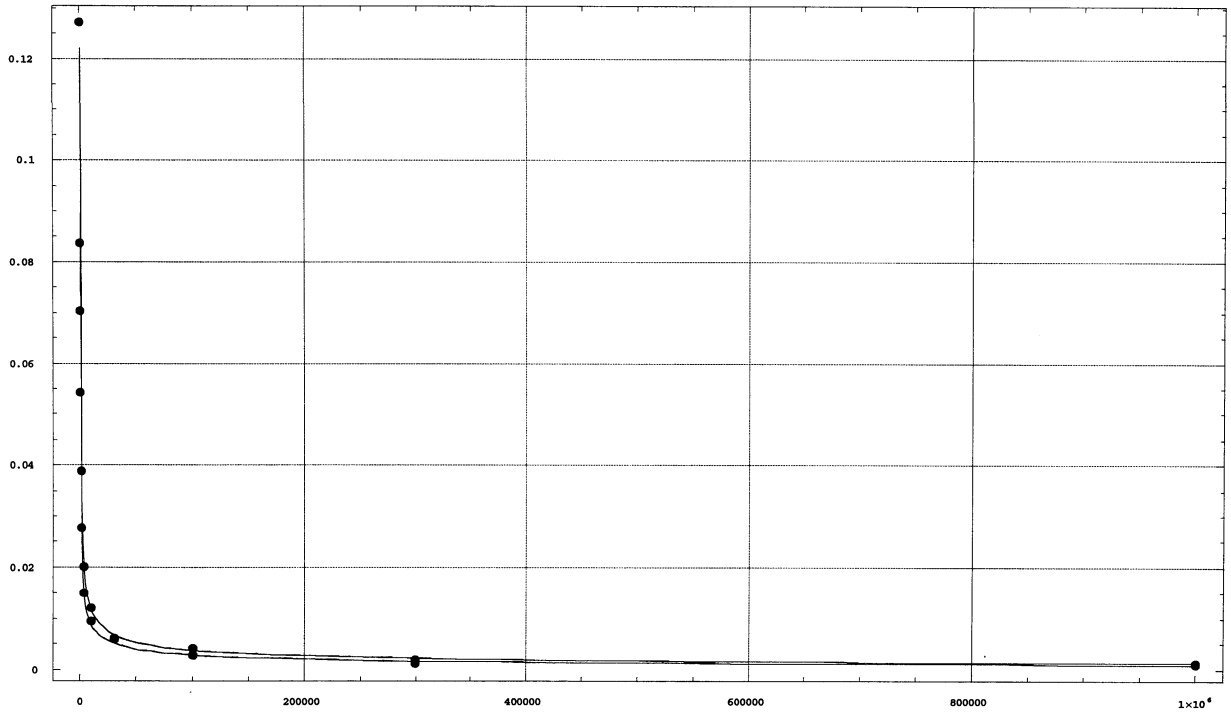
$$\left(\frac{\sum_{j=1}^{r} Log[\hat{y}_j] - rLog[c]}{\sum_{j=1}^{r} Log[x_j]}\right)\sum_{j=1}^{r}\left(Log[x_j]\right)^2 + Log[c]\sum_{j=1}^{r} Log[x_j] - \sum_{j=1}^{r} Log[\hat{y}_j]Log[x_j] = 0$$

$$Log[c] = \frac{\sum_{j=1}^{r} Log[x_j]\sum_{j=1}^{r} Log[\hat{y}_j]Log[x_j] - \sum_{j=1}^{r} Log[\hat{y}_j]\sum_{j=1}^{r}\left(Log[x_j]\right)^2}{\left(\sum_{j=1}^{r} Log[x_j]\sum_{j=1}^{r} Log[x_j] - r\sum_{j=1}^{r}\left(Log[x_j]\right)^2\right)} = value$$

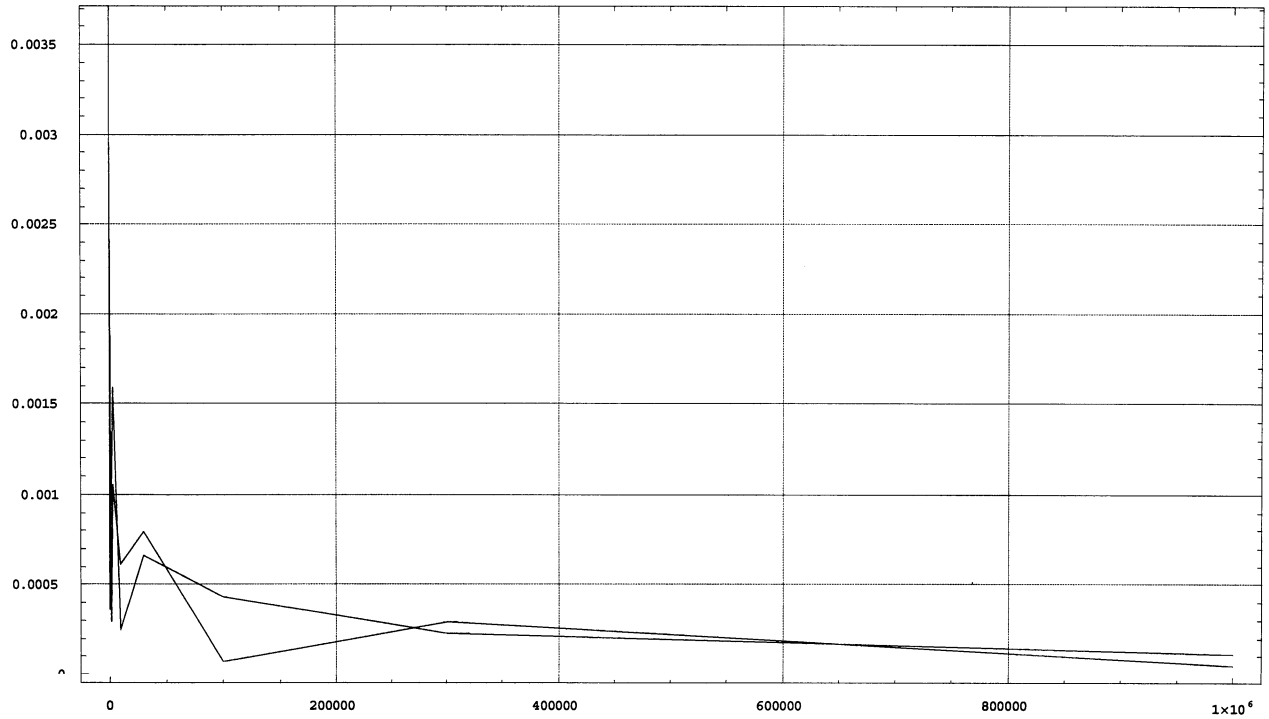
$$c = 10^{value}$$

Fit and Error in COV-n space

MC and LHS COV curve fit in COV-n space for test case 1 (COV of standard deviation)



This shows the MC and LHS absolute value of the error between curve fit and data



REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
December 2002

3. REPORT TYPE AND DATES COVERED
Final Contractor Report

4. TITLE AND SUBTITLE
Probabilistic Analysis and Density Parameter Estimation Within Nessus

5. FUNDING NUMBERS
WBS–22–708–90–53
NAG3–2297

6. AUTHOR(S)
Cody R. Godines and Randall D. Manteufel

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Texas at San Antonio
6900 N Loop 1604 W
San Antonio, Texas 78249–0670

8. PERFORMING ORGANIZATION REPORT NUMBER
E–13696

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR—2002-212008

11. SUPPLEMENTARY NOTES
Project Manager, Christos C. Chamis, Research and Technology Directorate, NASA Glenn Research Center, organization code 5000, 216–433–3252.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category: 39 Distribution: Nonstandard
Available electronically at http://gltrs.grc.nasa.gov
This publication is available from the NASA Center for AeroSpace Information, 301–621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***
This NASA educational grant has the goal of promoting probabilistic analysis methods to undergraduate and graduate UTSA engineering students. Two undergraduate-level and one graduate-level course were offered at UTSA providing a large number of students exposure to and experience in probabilistic techniques. The grant provided two research engineers from Southwest Research Institute the opportunity to teach these courses at UTSA, thereby exposing a large number of students to practical applications of probabilistic methods and state-of-the-art computational methods. In classroom activities, students were introduced to the NESSUS computer program, which embodies many algorithms in probabilistic simulation and reliability analysis. Because the NESSUS program is used at UTSA in both student research projects and selected courses, a student version of a NESSUS manual has been revised and improved, with additional example problems being added to expand the scope of the example application problems. This report documents two research accomplishments in the integration of a new sampling algorithm into NESSUS and in the testing of the new algorithm. The new Latin Hypercube Sampling (LHS) subroutines use the latest NESSUS input file format and specific files for writing output. The LHS subroutines are called out early in the program so that no unnecessary calculations are performed. Proper correlation between sets of multidimensional coordinates can be obtained by using NESSUS' LHS capabilities. Finally, two types of correlation are written to the appropriate output file. The program enhancement was tested by repeatedly estimating the mean, standard deviation, and 99th percentile of four different responses using Monte Carlo (MC) and LHS. These test cases, put forth by the Society of Automotive Engineers, are used to compare probabilistic methods. For all test cases, it is shown that LHS has a lower estimation error than MC when used to estimate the mean, standard deviation, and 99th percentile of the four responses at the 50 percent confidence level and using the same number of response evaluations for each method. In addition, LHS requires fewer calculations than MC in order to be 99.7 percent confident that a single mean, standard deviation, or 99th percentile estimate will be within at most 3 percent of the true value of the each parameter. Again, this is shown for all of the test cases studied. For that reason it can be said that NESSUS is an important reliability tool that has a variety of sound probabilistic methods a user can employ; furthermore, the newest LHS module is a valuable new enhancement of the program.

14. SUBJECT TERMS
Probability; Structures; Stress; Analysis; Sensitivities

15. NUMBER OF PAGES
272

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	