

A Comparison of Failure Probability Estimates by Monte Carlo Sampling and Latin Hypercube Sampling

by

Charles N. Zeeb and Patrick J. Burns

Colorado State University

Department of Mechanical Engineering

Fort Collins, CO 80523

czeeb@lamar.colostate.edu and pburns@colostate.edu

(970) 491-7479 and (970) 491-5778

1 Introduction

This report covers two methods to determine failure probability and other related quantities. The basic problem is as follows. Assume we are given a complex function which depends on several variables. Each variable has its own distribution; all of which are known. Given a certain threshold value for the function above which failure occurs, what is the probability of failure? One way to accomplish this is to calculate the function many times. By binning the function evaluations or trials, an estimate of the failure probability can be obtained. Enough trials must be done to achieve statistical convergence to some tolerance or threshold; more on this below.

One issue is how the values used in each trial are determined. In this report, two methods are explored, Monte Carlo sampling (MCS) and Latin hypercube sampling (LHS).

In Monte Carlo sampling (MCS), a random number generator (RNG), is used to generate (pseudo-)random numbers as needed for the trials. Generally, for MCS, there are three steps for each trial. First the random numbers must be generated. These random numbers are usually uniformly distributed between 0 and 1, so they must be transformed into the proper distribution for each variable. Finally, the values must be plugged into the function to obtain the result. Very fast RNGs exist, so the first step can be very fast. The computation times for the second and third steps depend on the complexity of the distributions and the function.

MCS is a subset of the more general field of Monte Carlo simulation. In MCS, each trial is a specific function and the same number of random variables are required for each trial. In general Monte Carlo simulation, each trial can simulate a process too complicated to model deterministically. The number of random numbers required for each trial can also vary. An example of this is the simulation of emission and absorption of photons of heat or light taking into account reflection and scattering.

Due to the large number of samples typically required, Monte Carlo sampling is often time consuming. There have been several efforts to reduce the number of samples

required. One popular method is Latin Hypercube sampling (LHS). It is defined as [Iman and Shortencarier, 1984]:

“LHS selects n different values from each of k variables X_1, \dots, X_k in the following manner. The range of each variable is divided into n nonoverlapping intervals on the basis of equal probability. One value from each interval is selected at random with respect to the probability density in the interval. The n values thus obtained for X_1 are paired in a random manner (equally likely combinations) with the n values of X_2 . These n pairs are combined in a random manner with the n values of X_3 to form n triplets, and so on, until n k -tuples are formed. This is the Latin hypercube sample. It is convenient to think of the LHS, or a random sample of size n , as forming an $n \times k$ matrix of input where the i^{th} row contains specific values of each of the k input variables to be used on the i^{th} run of the computer model.”

The purpose of this report is two-fold. The work of Maltby [Maltby, 1990] has been used to make estimates in the uncertainty of Monte Carlo simulation results. This report will apply Maltby’s work to obtain a formulation for an error estimate for MCS results and checks the validity of applying these error estimates to MCS and LHS results. The second objective of this report is to determine the relative benefits of MCS and LHS for obtaining failure probability estimates.

2 Error Estimates for MCS

This section applies statistical theory to obtain error estimates for MCS results. The first section covers Bernoulli random variables [Ross, 1988]. For the type of failure probability calculation covered in this report, all MCS results are Bernoulli random variables. The next section covers the central limit theorem and uses it to derive the MCS error estimate.

2.1 Bernoulli Random Variables

A Bernoulli random variable is defined as [Ross, 1988]:

“Suppose that a trial, or experiment, whose outcome can be classified as either a “success” or a “failure” is performed. If we let $X = 1$ when the outcome is a success and $X = 0$ when it is a failure, then the probability mass function is given by

$$p(0) = P\{X = 0\} = 1 - p \tag{1}$$

$$p(1) = P\{X = 1\} = p, \tag{2}$$

where p , $0 \leq p \leq 1$, is the probability that the trial is a “success.”

A random variable X is said to be a Bernoulli random variable (after the Swiss mathematician James Bernoulli) if its probability mass function is given by eqns. (1) and (2) for some $p \in (0,1)$.”

As can be seen from the definition above, each MCS trial can be considered a Bernoulli random trial, but $X = 1$ if the experiment fails, not if it succeeds. Looking at the MCS trials this way is useful because it allows failure statistics to be obtained without requiring knowledge of the distributions of the variables on which the function depends. Therefore, Bernoulli statistics are applicable no matter how many variables the function depends on or what their distributions are. In fact, Bernoulli statistics have been applied to the simulation of the absorption of heat and light where reflections and scattering make the number of random numbers required per trial variable [Maltby, 1990; Zeeb et al., 1999; Zeeb and Romero, 1999].

For Bernoulli statistics to apply to a Monte Carlo simulation, the end result of the trial must be total success or total failure. Partial success such as partial absorption of a photon is not a Bernoulli process.

For a Bernoulli random variable, the mean, μ , and the variance, σ^2 , are defined as:

$$\mu = E[X] = p \quad (3)$$

$$\sigma^2 = E[(X - \mu)^2] = p(1 - \mu)^2 + (1 - p)(0 - \mu)^2 = p(1 - p) , \quad (4)$$

where E is the expected value.

The failure probability estimate obtained by MCS is equal to μ which is in turn equal to p . Since it is a failure probability, it will also be referred to as F .

2.2 The Central Limit Theorem and MCS Error Estimates

The central limit theorem [Ross, 1988] states that if there are N independent and identically distributed random variables, X_1, \dots, X_N , each having mean μ and variance σ^2 , then:

$$P\{H \leq z\} \rightarrow \Phi(z) \quad \text{as } n \rightarrow \infty , \quad (5)$$

where H is defined as:

$$H = \frac{X_1 + \dots + X_N - N\mu}{\sigma\sqrt{N}} , \quad (6)$$

and $\Phi(z)$ is the cumulative distribution function for the standard normal distribution (that is, a normal distribution with $\mu = 0$ and $\sigma^2 = 1$):

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z \exp\left(-\frac{x^2}{2}\right) dx . \quad (7)$$

In other words, H tends toward the standard normal distribution as N goes to ∞ . The central limit theorem was first applied to Bernoulli random variables [Ross, 1988], but as mentioned above, it can be applied to random variables from any distribution.

If we define μ^* as the estimate of the actual mean from the N random variables:

$$\mu^* = \frac{X_1 + \dots + X_N}{N} , \quad (8)$$

then for N large enough, using eqns. (6) and (8), eqn. (5) can be rewritten as:

$$P\left\{\mu - \frac{\sigma}{\sqrt{N}}z \leq \mu^* \leq \mu + \frac{\sigma}{\sqrt{N}}z\right\} = P\{\mu - \eta \leq \mu^* \leq \mu + \eta\} \equiv \Phi(z) - \Phi(-z) = D(z) . \quad (9)$$

η is the confidence half-envelope value and $D(z)$ is the probability that the predicted value of the mean is within plus or minus η of the actual mean. Various sources tabulate z as a function of Φ and/or D [Kreyszig, 1993; Ross, 1988]. Using these tabulated values, it is possible to determine an envelope around the mean for which the probability, or confidence, of μ^* being inside the envelope is $D(z)$.

Examples of such confidence envelopes are given in Figs. 1 to 4 where sample problems are done using the two benchmark problems described in the next section for a given threshold. In Figs. 1 and 3, the μ^* values calculated in five different MCS runs are plotted along with the 95% and 99% confidence envelopes. The confidence envelope is calculated using eqn. (9) with N equal to the number of samples. From the tables, z is 1.960 for 95% confidence and 2.576 for 99% confidence. μ , called the “exact” answer here (and σ which is calculated from it) are assumed to be the average value of the final results of the five runs. μ^* is expected to be outside the 95% confidence interval 5% of the time. Of course the actual number of times μ^* is outside the 95% envelope varies from series of trials to series of trials. μ^* can be seen outside the 95% envelope several times in Fig. 3, but it does not go outside that envelope in any part of Fig. 1 that is visible to the eye.

Figs. 2 and 4 display the “exact” answer, the results of one run, and the 95% and 99% confidence envelopes. These confidence envelopes are determined from the current value of μ^* not from the “exact” answer. While they waver quite a bit for small numbers of samples, they converge to the “exact” confidence envelope as N increases.

Using eqns. (3), (4), and (9), it is possible to define η in terms of the F , the failure probability. It should be remembered F is just the μ value for the MCS trials.

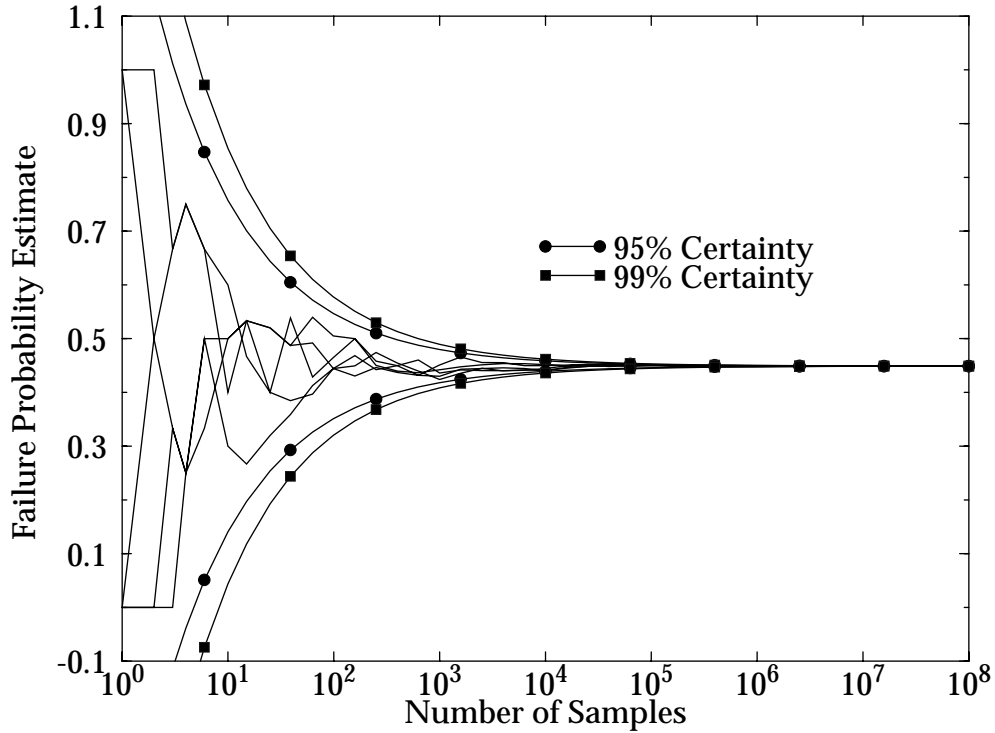


Figure 1 Five MCS Failure Probability Estimate Curves for Problem 1, Threshold = 0.5
Confidence Envelope around the “Exact” Answer, $\mu = 0.449$ (5×10^8 samples)

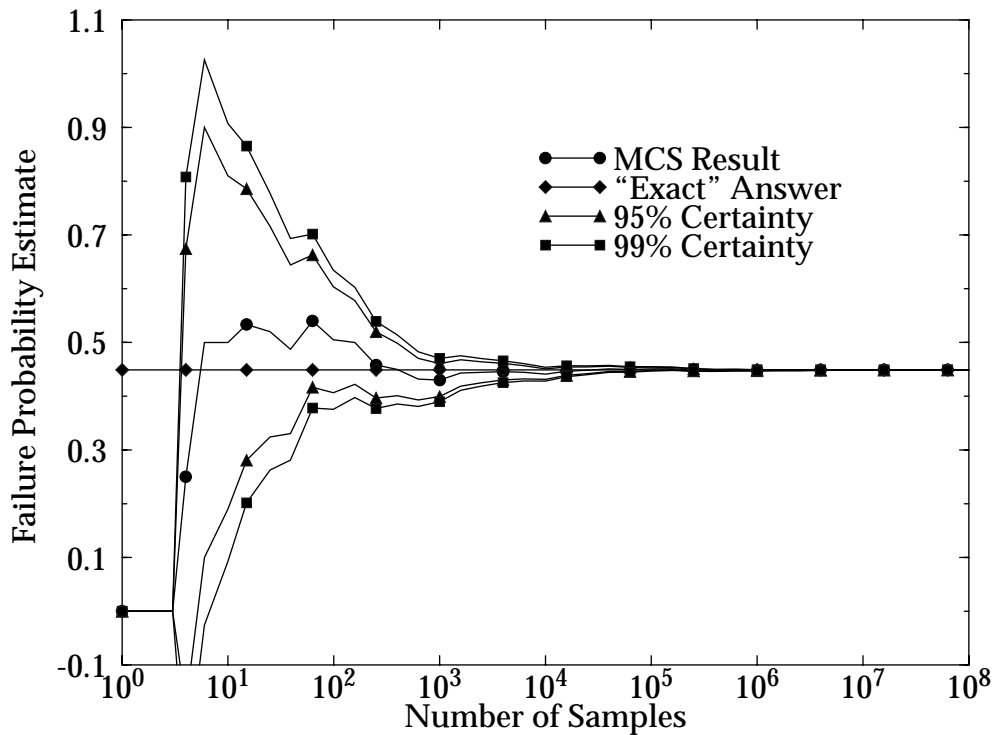


Figure 2 Single MCS Failure Probability Estimate Curve for Problem 1, Threshold = 0.5
Confidence Envelope around the MCS Result, μ^* ($\mu = 0.449$)

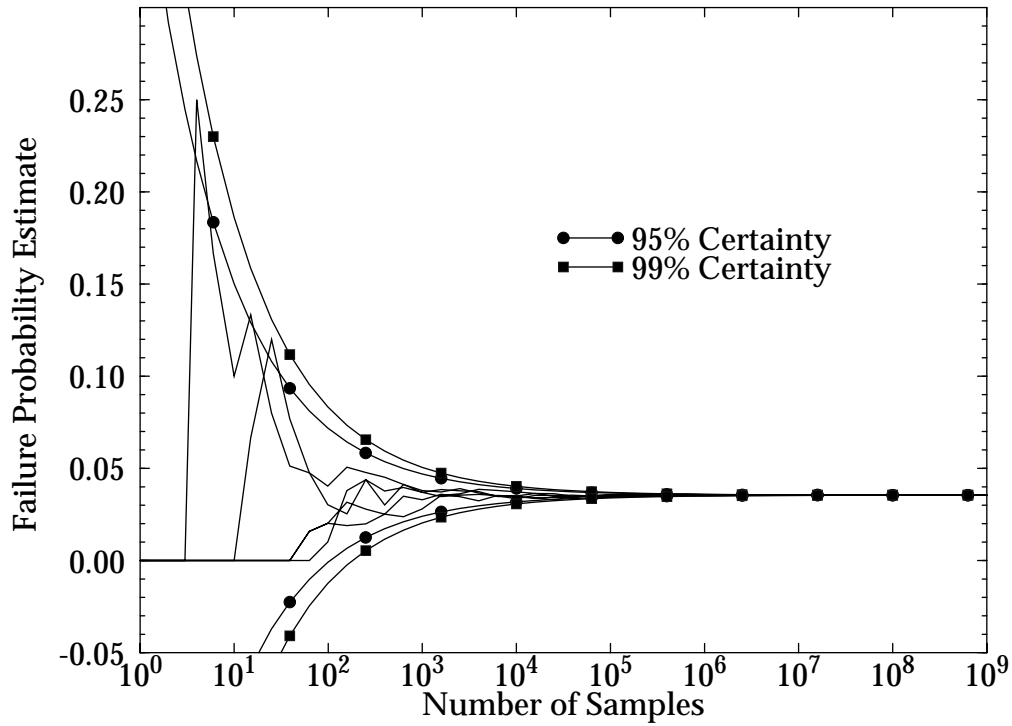


Figure 3 Five MCS Failure Probability Estimate Curves for Problem 2, Threshold = 0.4
Confidence Envelope around the “Exact” Answer, $\mu = 0.0357$ (5×10^9 samples)

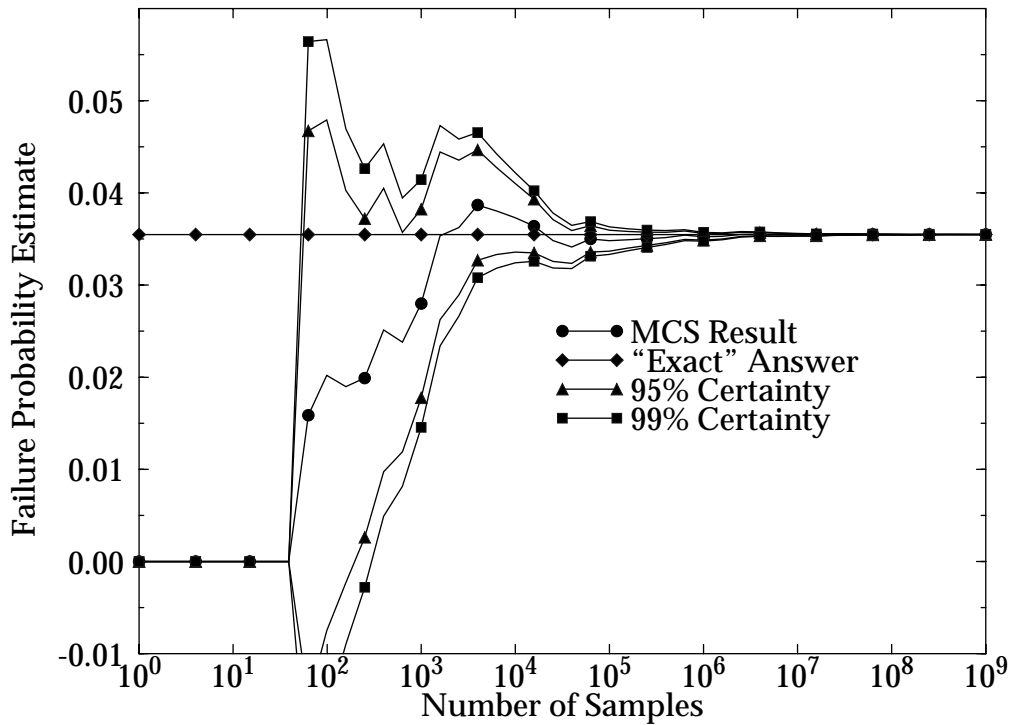


Figure 4 Single MCS Failure Probability Estimate for Problem 2, Threshold = 0.4
Confidence Envelope around the MCS Result, μ^* ($\mu = 0.0357$)

$$\eta = \frac{\sigma}{\sqrt{N}}z = z\sqrt{\frac{F(1-F)}{N}}. \quad (10)$$

For Bernoulli random variables such as F , besides the confidence half-envelope value, η , it is possible to specify δ , the percent half-envelope value:

$$P\{\mu(1-\delta) \leq \mu^* \leq \mu(1+\delta)\} = D(z) \quad (11)$$

$$\delta = \frac{\eta}{\mu} = z\sqrt{\frac{1-F}{NF}}. \quad (12)$$

δ is often referred to as the confidence interval [Zeeb et al., 1999].

η and δ can be used as convergence criteria for MCS problems. Before the run, the user specifies a value for D (the percent confidence) and a maximum value for either η or δ . Then at various points in the run, η or δ is calculated using F^* , the estimated value of F . Once the user defined value of η or δ is obtained, the results are considered converged and computation stops. The second section below gives guidelines in determining how many trials are required. It should be noted that once enough trials have been computed to obtain a rough estimate of the answer, the difference between η or δ calculated by F^* and by F is negligible; this is easily seen in the graphs in the second section below this one.

3 Description of Benchmark Problems

In the tests performed below, two benchmark problems are used. Both are taken from the earlier LHS testing done by Romero and Bankston [1998a, 1998b]. The first benchmark problem (problem 1) evaluates the function:

$$\text{response}(p1, p2) = 1.5 \sin(1.3\theta) \left[0.8r + 0.35 \sin\left(2.4\pi \frac{r}{\sqrt{2}}\right) \right] \quad (13)$$

$$r = \sqrt{p1^2 + p2^2} \quad \theta = \text{atan}\left(\frac{p2}{p1}\right) \quad (14)$$

over the domain $0 \leq p1, p2 \leq 1$. The second benchmark problem (problem 2) evaluates the function:

$$\text{response}(p1, p2) = 1.5 - \sqrt{2.25 - p1^2 - p2^2} \quad (15)$$

over the domain $0 \leq p1, p2 \leq 1$. For both functions, the variables $p1$ and $p2$ are both normally distributed with $\mu = 0.5$, $\sigma = 0.167$. As mentioned in the problem descriptions above, the truncation limits are 0 and 1. Since these truncation limits are equivalent to $\pm 3\sigma$, the effects of truncation are relatively small.

4 Verification of Error Estimate for MCS Results

To test the validity of the error estimates, MCS results were obtained for four threshold values for each of two benchmark problems described in the previous section. The thresholds were chosen by Romero and Bankston to test failure probability estimates at several orders of magnitude. The results ranged from 0.98 to 2.6×10^{-4} . The random numbers were generated using the random number generator RAN2 [Press et al., 1992]. The random numbers were transformed to the proper normal distribution using the Box-Muller method [Press et al., 1992]. If the value was outside of the range 0 to 1, it was rejected and another value was generated to replace it.

The results are shown below in Figs. 5 to 20. For each case, there are two different graphs. As can be seen in Figs. 1 to 4, displaying the MCS results and the confidence envelopes on a linear scale shows clearly that convergence is happening but fine detail is absent. To better display the convergence, the pair of graphs shown below for each test problem use log-log scales and show differences from the “exact” answer instead of the actual values. To show the convergence of multiple runs, the first graph of each pair displays the absolute value of the difference of the “exact” answer and the results for five different runs and the 95% and 99% percent confidence half-envelope values. The “exact” answer is the average of the final results of the five runs and it is used to calculate the confidence envelopes for this graph. The second graph in each pair displays the difference between the results for one MCS run and the “exact” answer and the 95% and 99% confidence half-envelope values. In these graphs, the confidence half-envelope values are calculated from the running μ^* and σ^* values of the MCS simulation, rather from the “exact” answer used in the first graph. In this second series of graphs, the curves representing the confidence half-envelopes are always clipped at the beginning of the graph. The reason for this is that if F or F^* is 0 or 1, the confidence half-envelope value is 0.

While basing the confidence envelope off the “exact” answer in the first series of graphs allows the convergence for multiple runs to be seen, the second type of graph shows the “real time” confidence envelope one would get from the results of an actual run. Note that while the envelope does vary a bit at the beginning of the run, it becomes insensitive to the changes as the run continues and converges close to the “exact” answer envelope. The point where the envelope “stabilizes” varies. In the runs done for this report, three types of behavior were found. In the cases where F was equal to 3.5×10^{-2} or less, the half-envelope value is close to its proper value by the time the number of samples equals to the inverse of F . By 100 times the inverse of F , the envelope has stabilized in that an asymptotic value has been reached. To obtain meaningful results, at least 100 times the inverse of the value being obtained must be used any way. For larger values of F , the results differed a bit. For F around 0.45, the envelope stabilizes quickly after about 10 samples. On the other hand, stabilization is a bit slow for F around 0.98, up to 1,000 samples are required in these cases.

For 10^8 samples, 41 points were computed per curve, while 45 points were computed per curve for 10^9 samples. With this many points, it is expected that the results rise

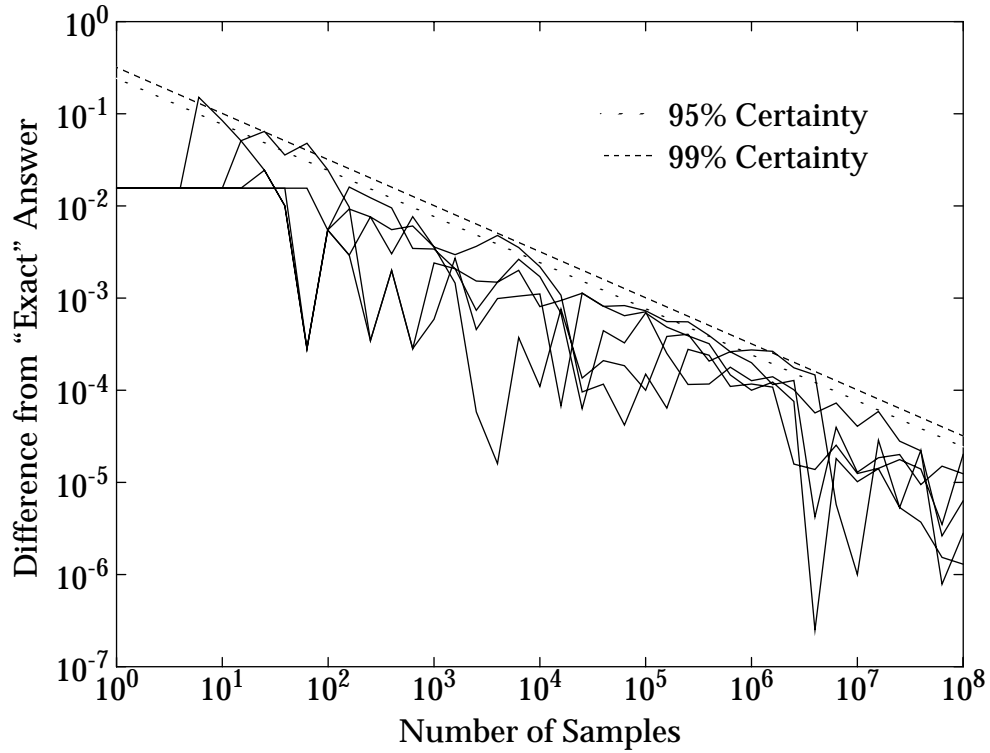


Figure 5 MCS Difference from “Exact” Answer for Problem 1, Threshold = 0.2
Confidence Half-Envelope Calculated from the “Exact” Answer ($\cong 0.9844$)

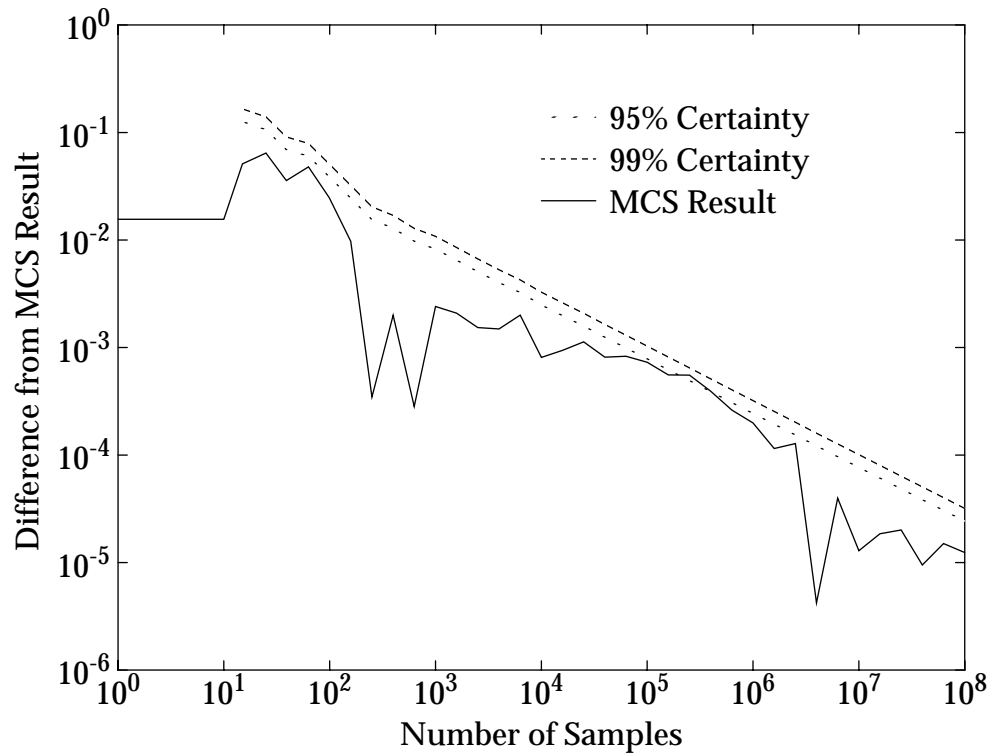


Figure 6 Error in a MCS Result for Problem 1, Threshold = 0.2
Half-Envelope Calculated from the MCS Result (“Exact” Answer $\cong 0.9844$)

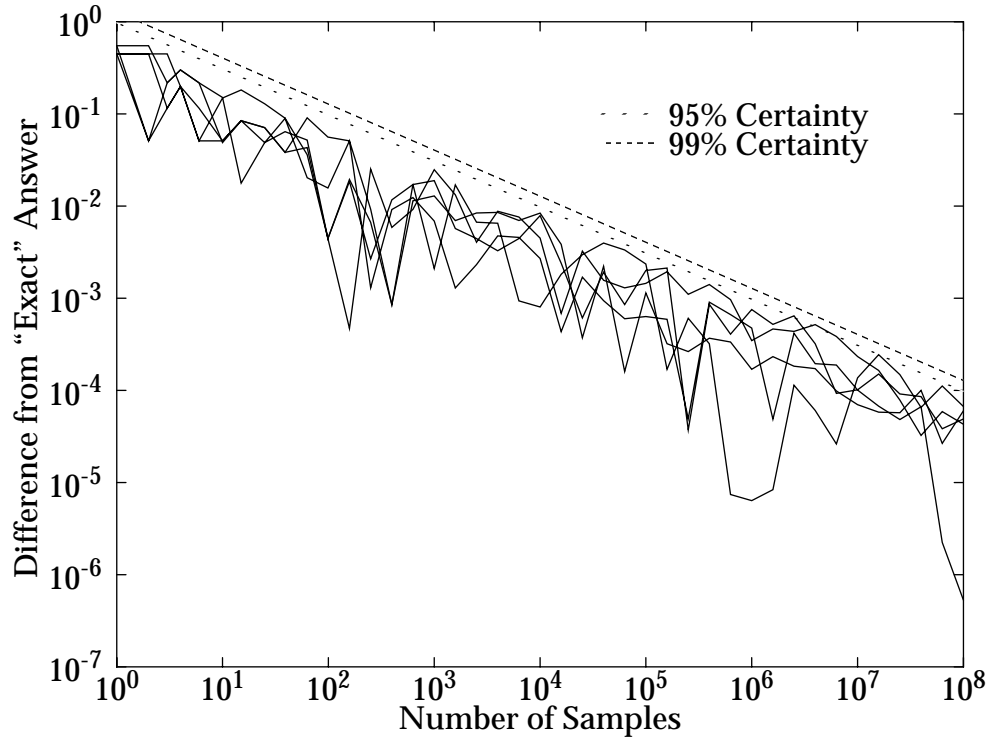


Figure 7 MCS Difference from “Exact” Answer for Problem 1, Threshold = 0.5
Confidence Half-Envelope Calculated from the “Exact” Answer ($\cong 0.4489$)

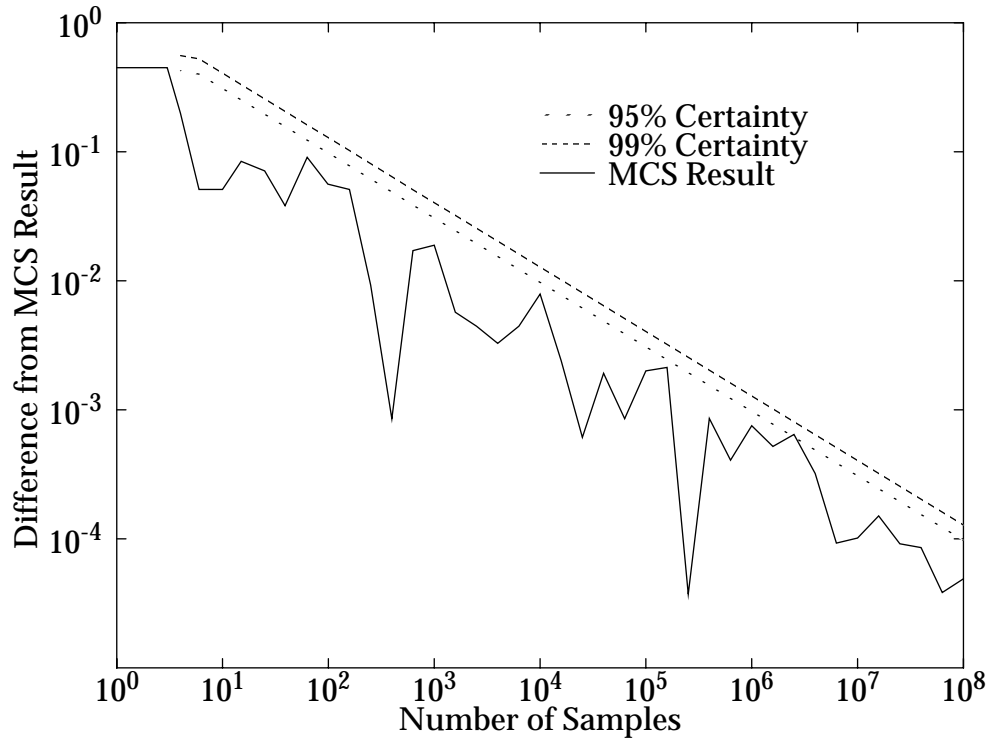


Figure 8 Error in a MCS Result for Problem 1, Threshold = 0.5
Half-Envelope Calculated from the MCS Result (“Exact” Answer $\cong 0.4489$)

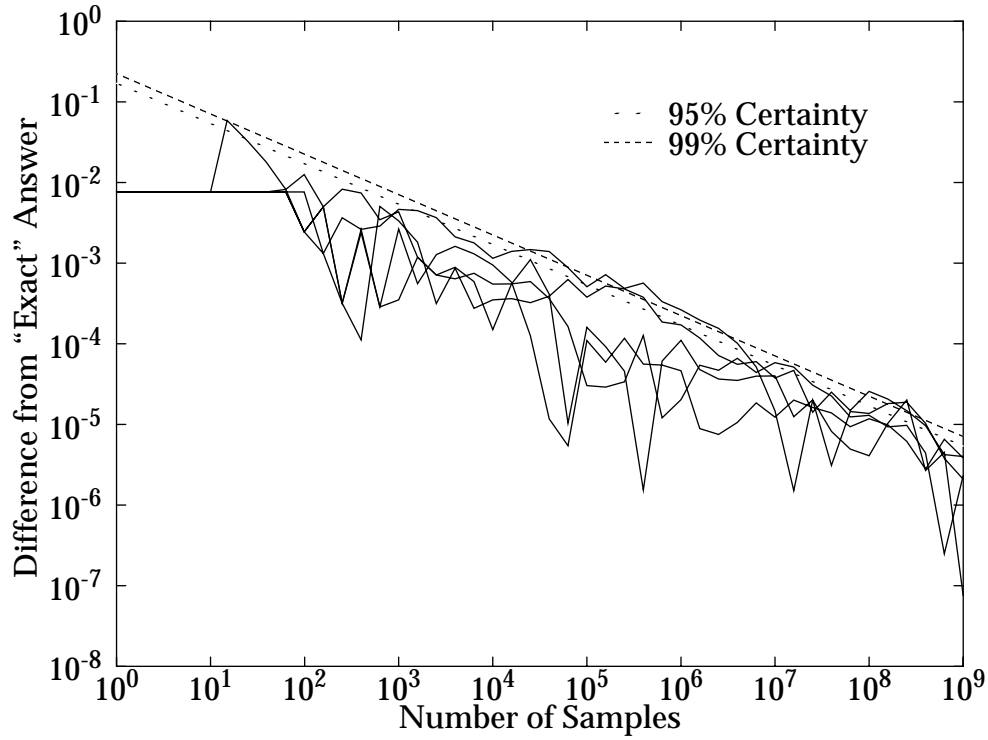


Figure 9 MCS Difference from “Exact” Answer for Problem 1, Threshold = 1.0
Confidence Half-Envelope Calculated from the “Exact” Answer ($\cong 7.650 \times 10^{-3}$)

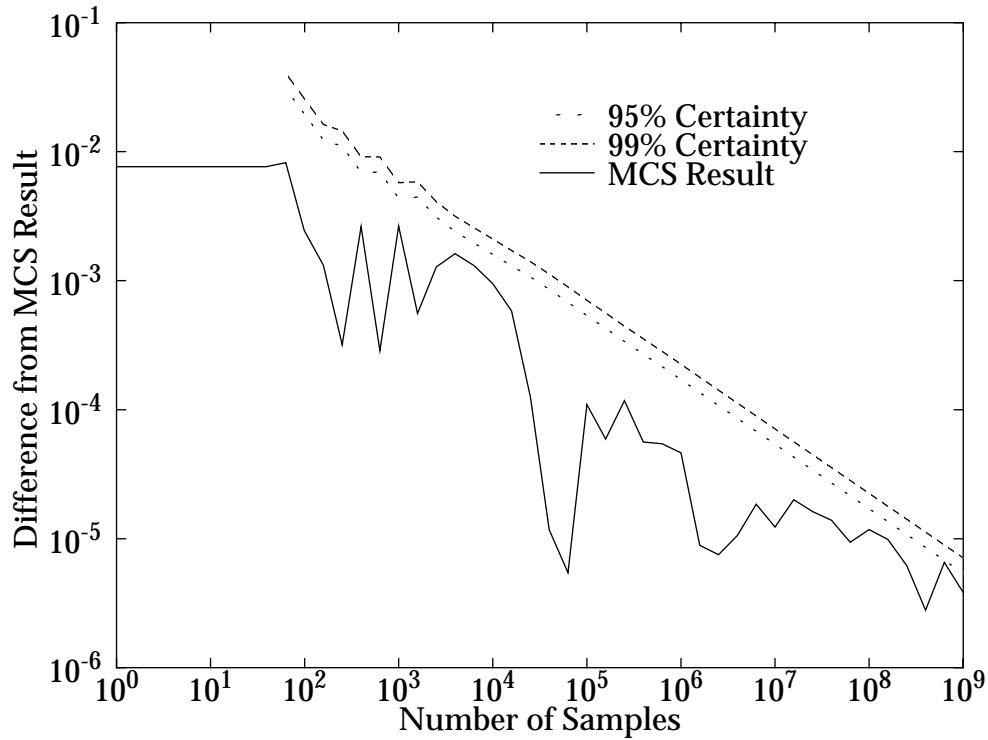


Figure 10 Error in a MCS Result for Problem 1, Threshold = 1.0
Half-Envelope Calculated from the MCS Result (“Exact” Answer $\cong 7.650 \times 10^{-3}$)

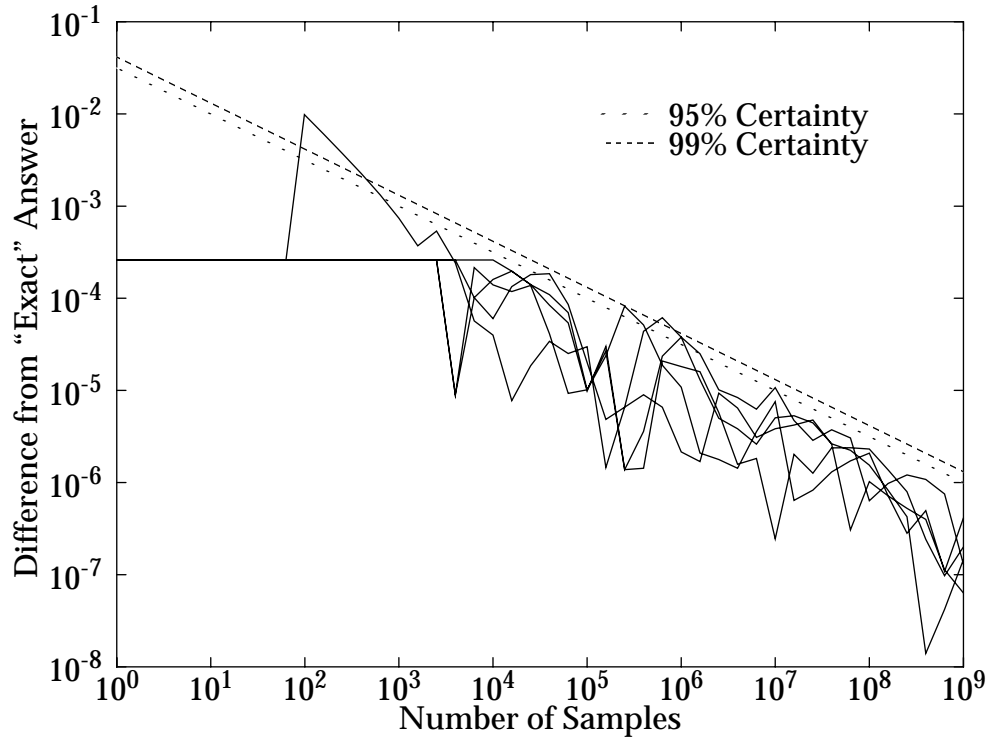


Figure 11 MCS Difference from “Exact” Answer for Problem 1, Threshold = 1.5
Confidence Half-Envelope Calculated from the “Exact” Answer ($\cong 2.602 \times 10^{-4}$)

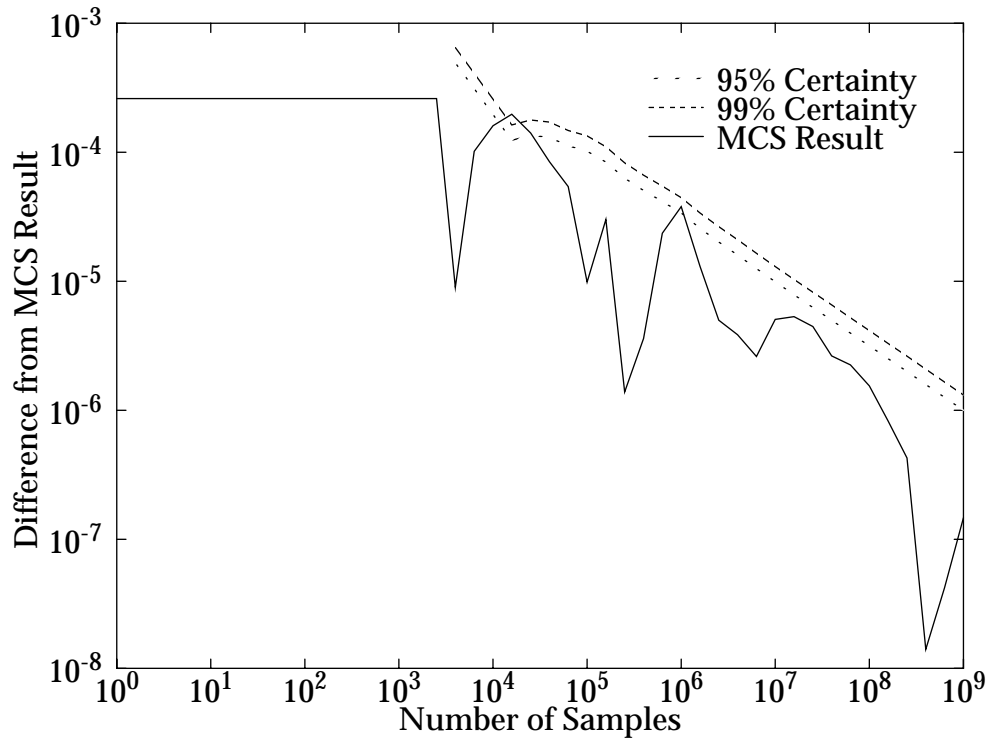


Figure 12 Error for a MCS Result for Problem 1, Threshold = 1.5
Half-Envelope Calculated from the MCS Result (“Exact” Answer $\cong 2.602 \times 10^{-4}$)

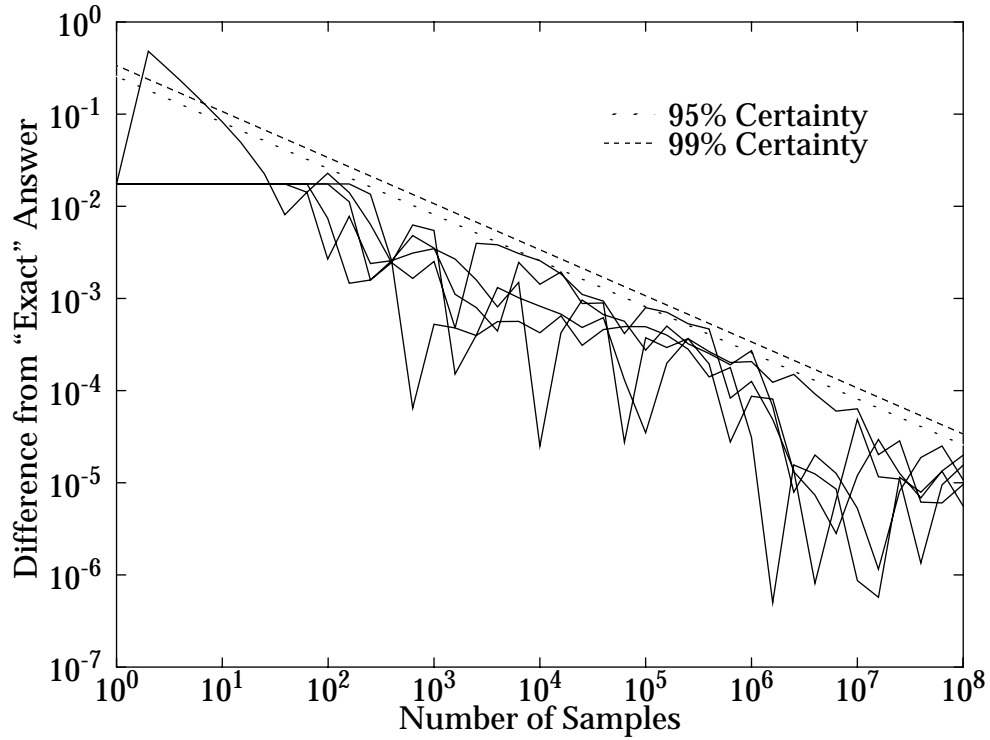


Figure 13 MCS Difference from “Exact” Answer for Problem 2, Threshold = 0.05
Confidence Half-Envelope Calculated from the “Exact” Answer ($\cong 0.9825$)

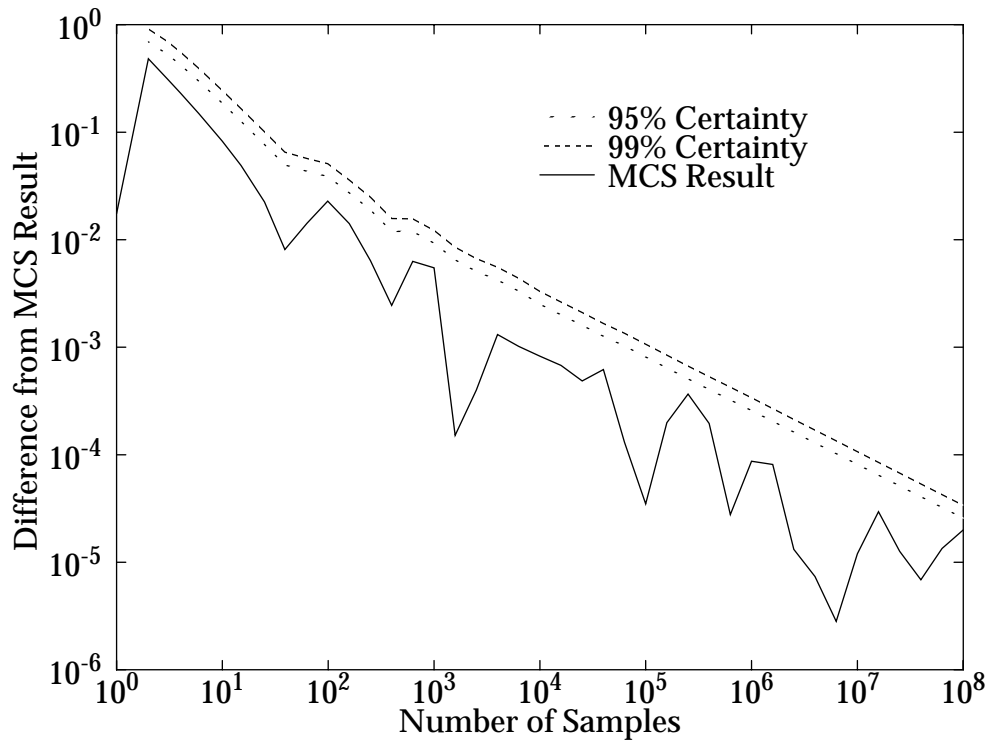


Figure 14 Error in a MCS Result for Problem 2, Threshold = 0.05
Half-Envelope Calculated from the MCS Result (“Exact” Answer $\cong 0.9825$)

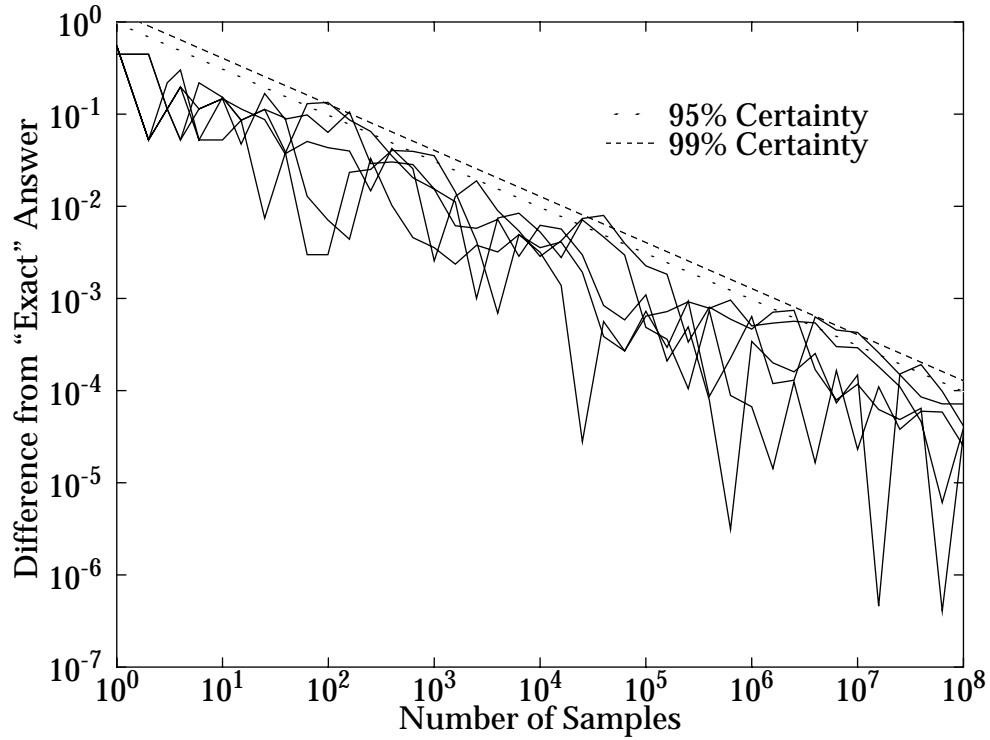


Figure 15 MCS Difference from “Exact” Answer for Problem 2, Threshold = 0.2
Confidence Half-Envelope Calculated from the “Exact” Answer ($\cong 0.4474$)

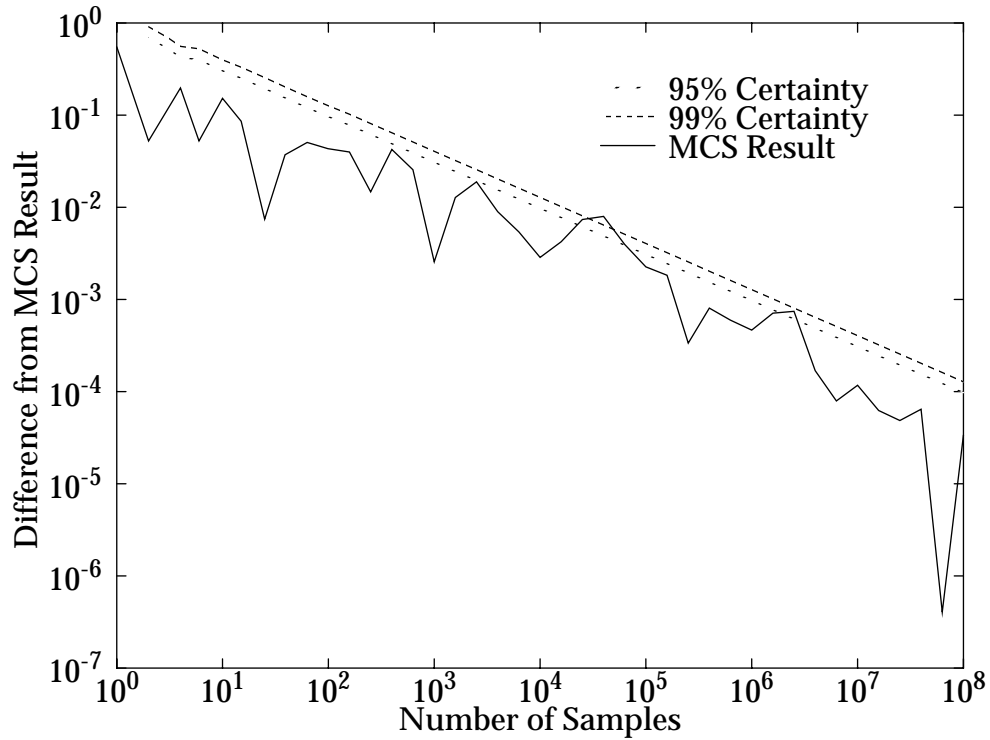


Figure 16 Error in a MCS Result for Problem 2, Threshold = 0.2
Half-Envelope Calculated from the MCS Result (“Exact” Answer $\cong 0.4474$)

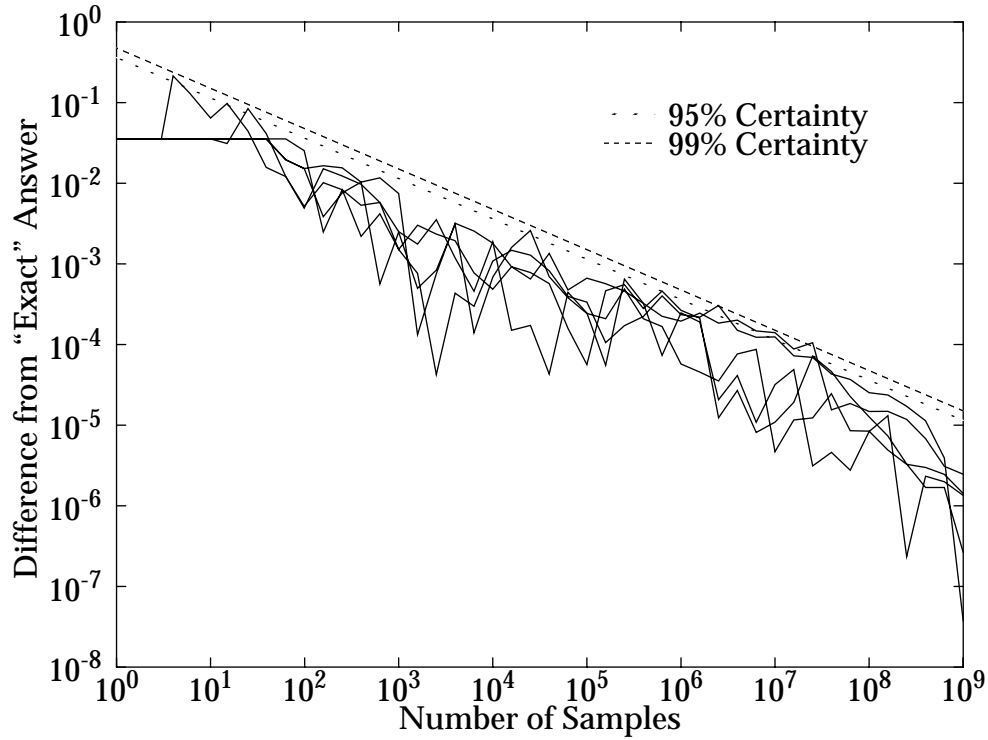


Figure 17 MCS Difference from “Exact” Answer for Problem 2, Threshold = 0.4
Confidence Half-Envelope Calculated from the “Exact” Answer ($\cong 3.549 \times 10^{-2}$)

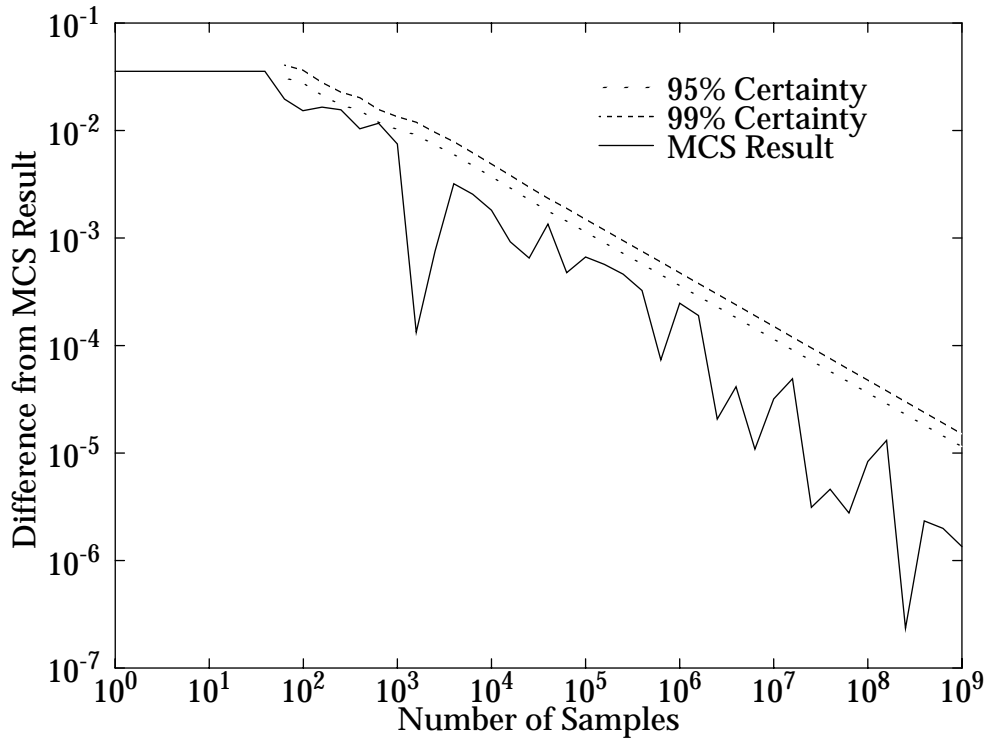


Figure 18 Error for a MCS Result for Problem 2, Threshold = 0.4
Half-Envelope Calculated from the MCS Result (“Exact” Answer $\cong 3.549 \times 10^{-2}$)

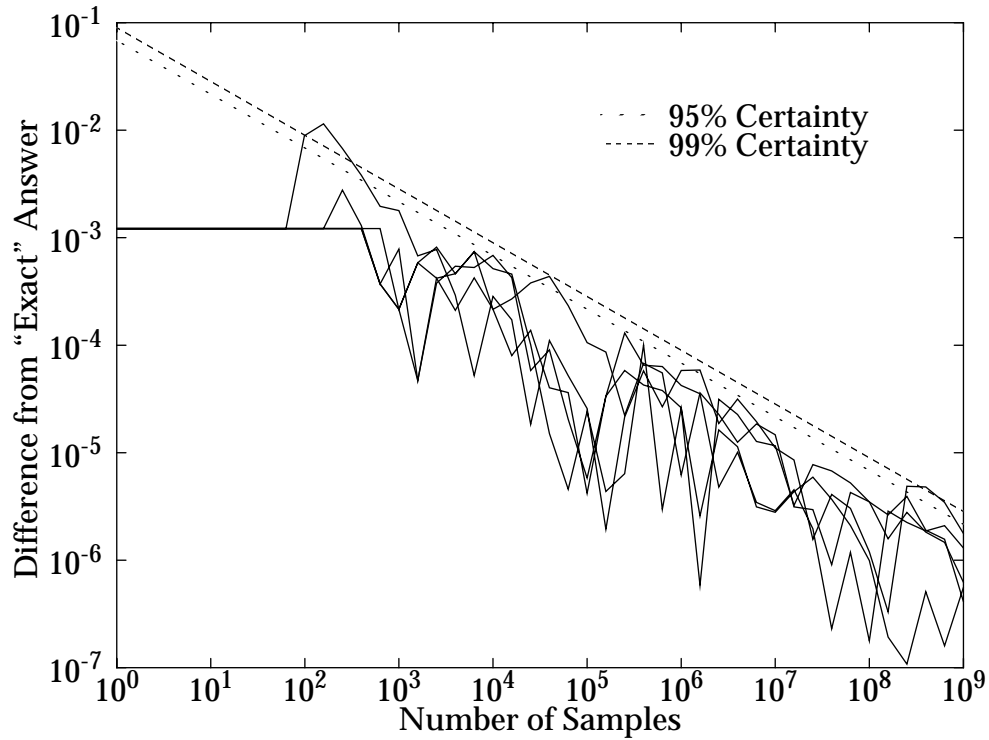


Figure 19 MCS Difference from “Exact” Answer for Problem 2, Threshold = 0.6
 Confidence Half-Envelope Calculated from the “Exact” Answer ($\cong 1.216 \times 10^{-3}$)

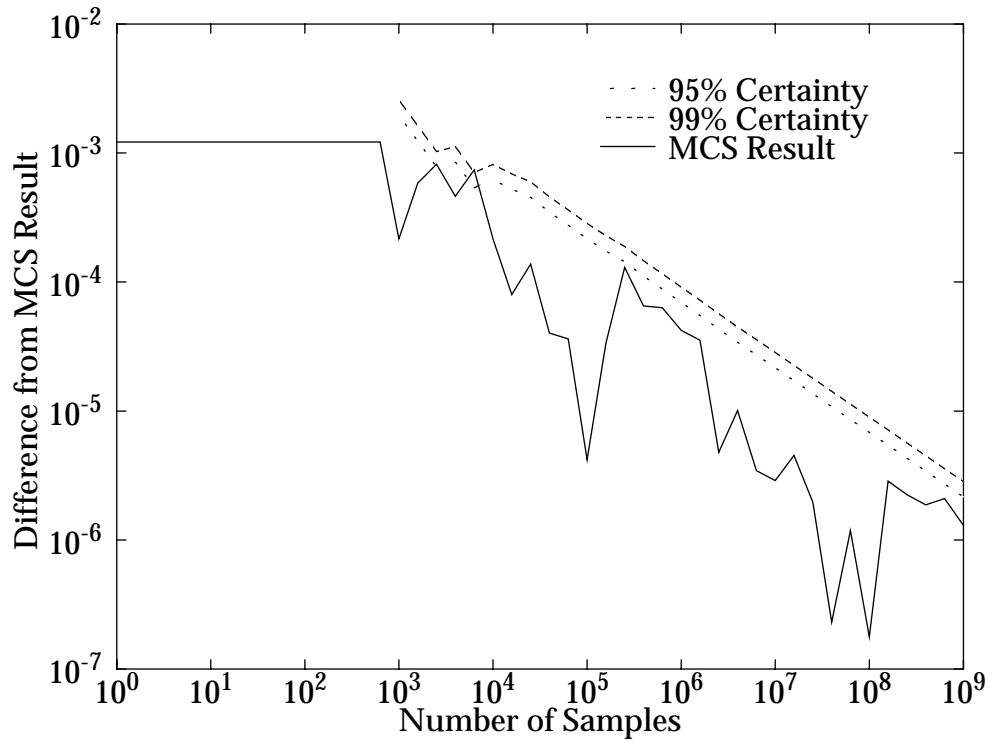


Figure 20 Error in a MCS Result for Problem 2, Threshold = 0.6
 Half-Envelope Calculated from the MCS Result (“Exact” Answer $\cong 1.216 \times 10^{-3}$)

above the 95% certainty point about twice per curve for all the graphs and that about two points rise about the 99% certainty in the graphs showing the results for five runs. This is indeed what generally happens. In 1,720 points shown in all eight graphs, it is expected that a few extreme outlying points would be found such as the result around 100 samples in Fig. 11. On the other hand, some graphs have no points over 99% certainty and few over 95% certainty such as Fig. 7. All in all, the results seem to be quite consistent with the confidence envelopes.

5 Application of MCS Error Estimates to LHS Results

In this section, graphs are presented to display how useful MCS error estimates are in predicting the error in LHS results. To create these graphs, several runs were performed with LHS sample sizes of 10^3 , 10^4 , 10^5 , 10^6 , and 10^7 samples. The LHS samples were obtained using a copy of Iman and Shortencarier's [1984] LHS program that was obtained from Sandia National Laboratories [Romero, 1997]. A program was written to process the LHS samples and obtain the failure probability estimates. The maximum sample size was limited to 10^7 samples due to memory and disk space constraints that will be described below. Due to the amount of CPU time involved, while 10 runs were done for all the other sample sizes, only 5 runs were done with 10^7 samples.

The results are shown below in Figs. 21 to 36. In the graphs, each LHS result represents the result of an entire LHS sample of the size specified on the x axis. Again, there are two different graphs for each case. In the first type of graph, the "exact" answer is obtained from the average of the five 10^7 sample LHS runs and is used to calculate the 95% and 99% confidence half-envelope values plotted on the graph. The differences between the "exact" answer and the results of the individual LHS runs are also plotted. The second graph is actually an overlay of the LHS results onto the five run MCS graphs shown previously. In this case, the convergence envelope is calculated from the average of the MCS results. While the second graph allows one to compare the convergence behavior of LHS and MCS results, the first graph gives a more detailed, less cluttered presentation of the LHS results over a smaller range.

In the graphs, Figs. 21 to 36, the confidence half-envelope values are obtained using the MCS formula, eqn. (10). Since, for a given number of samples, the LHS method is purported to give more accurate results than the MCS method, it is clear that these half-envelopes should be too large. The actual purpose of these graphs is twofold. First it verifies that the MCS and LHS methods converge to the same results. Second, it allows the convergence of the MCS and LHS results to be compared.

The LHS method does seem to converge slightly better than the MCS method for the same number of samples. For the eight cases combined, there are 450 separate results. If the LHS method converges like the MCS method, it is expected that about 4 or 5 results will be above 99% confidence and 22 or 23 points will be above 95% confidence. Compared to the confidence envelope values obtained from the LHS results, it is found that no points are above 99% confidence, although one is close. Furthermore, only 5 points are above 95% confidence. Still when comparing the LHS and MCS results, there does

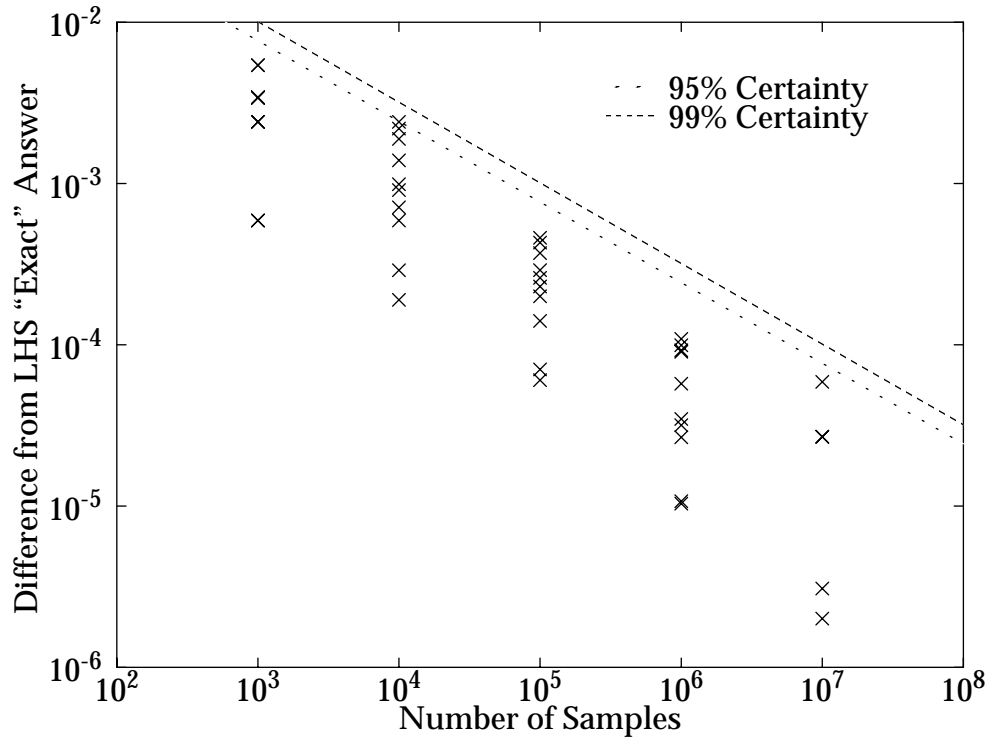


Figure 21 LHS Errors for Problem 1, Threshold = 0.2
Bernoulli Half-Envelope Calculated from the LHS "Exact" Answer ($\cong 0.9844$)

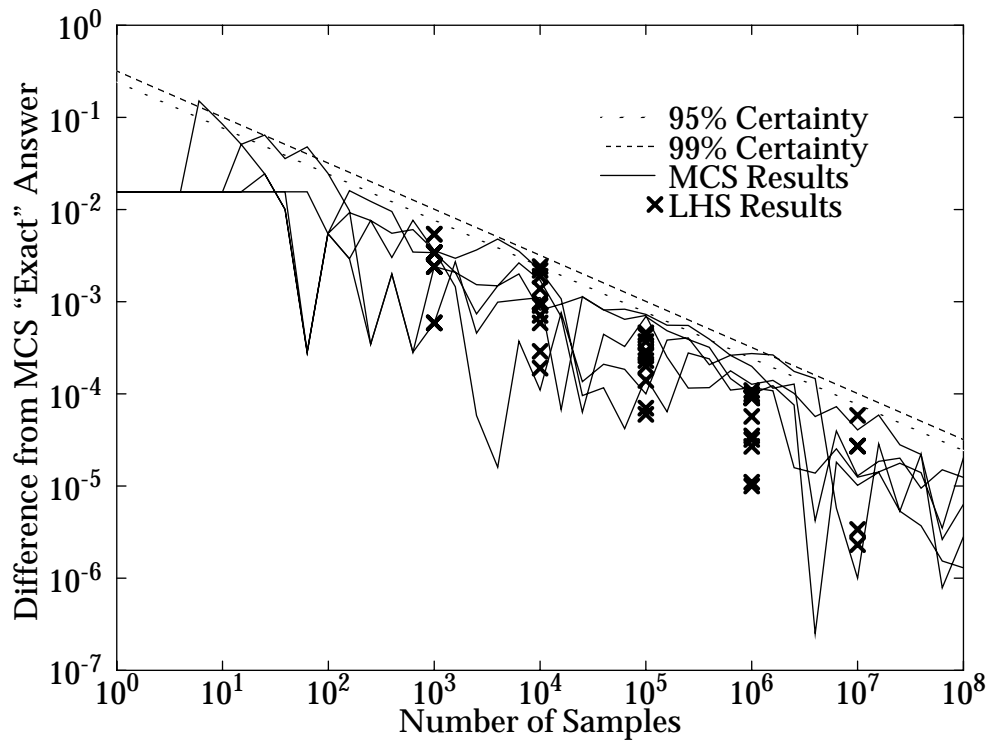


Figure 22 Comparison of MCS Errors to LHS Errors for Problem 1, Threshold = 0.2
Confidence Half-Envelope Calculated from the MCS "Exact" Answer ($\cong 0.9844$)

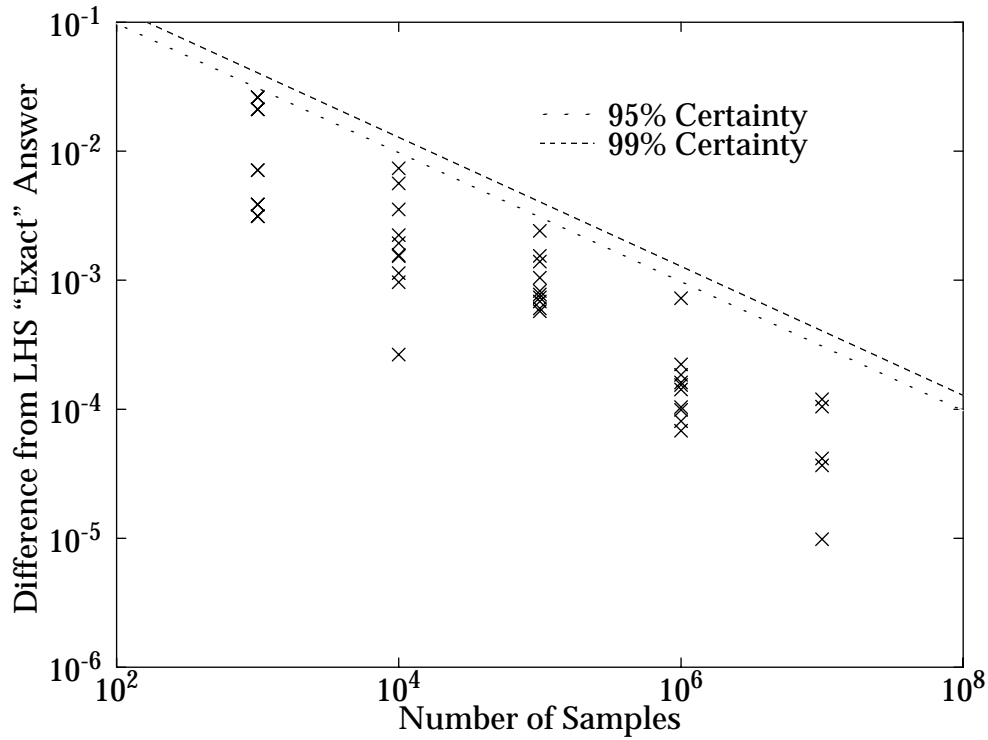


Figure 23 LHS Errors for Problem 1, Threshold = 0.5
Bernoulli Half-Envelope Calculated from the LHS "Exact" Answer ($\cong 0.4489$)

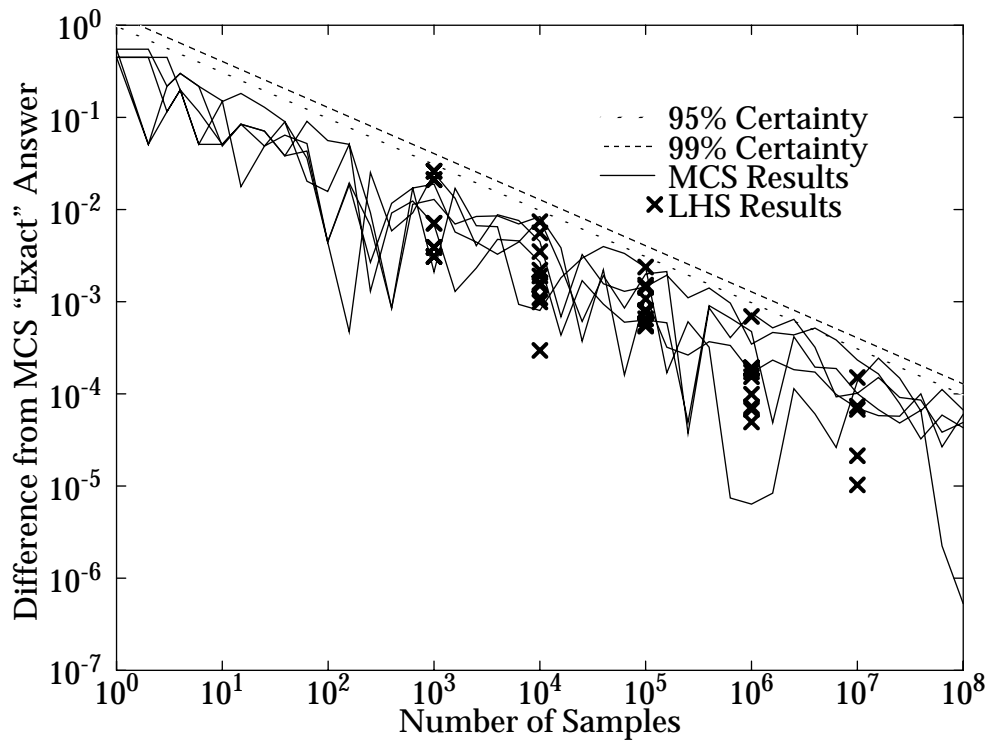


Figure 24 Comparison of MCS Errors to LHS Errors for Problem 1, Threshold = 0.5
Confidence Half-Envelope Calculated from the MCS "Exact" Answer ($\cong 0.4489$)

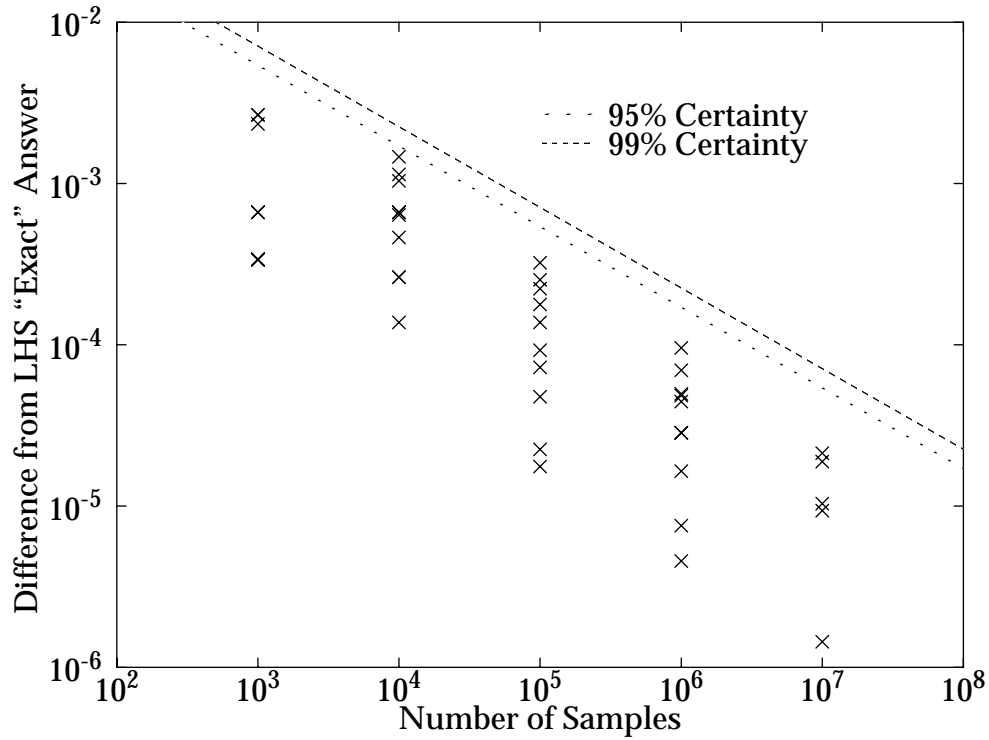


Figure 25 LHS Errors for Problem 1, Threshold = 1.0

Bernoulli Half-Envelope Calculated from the LHS "Exact" Answer ($\cong 7.662 \times 10^{-3}$)

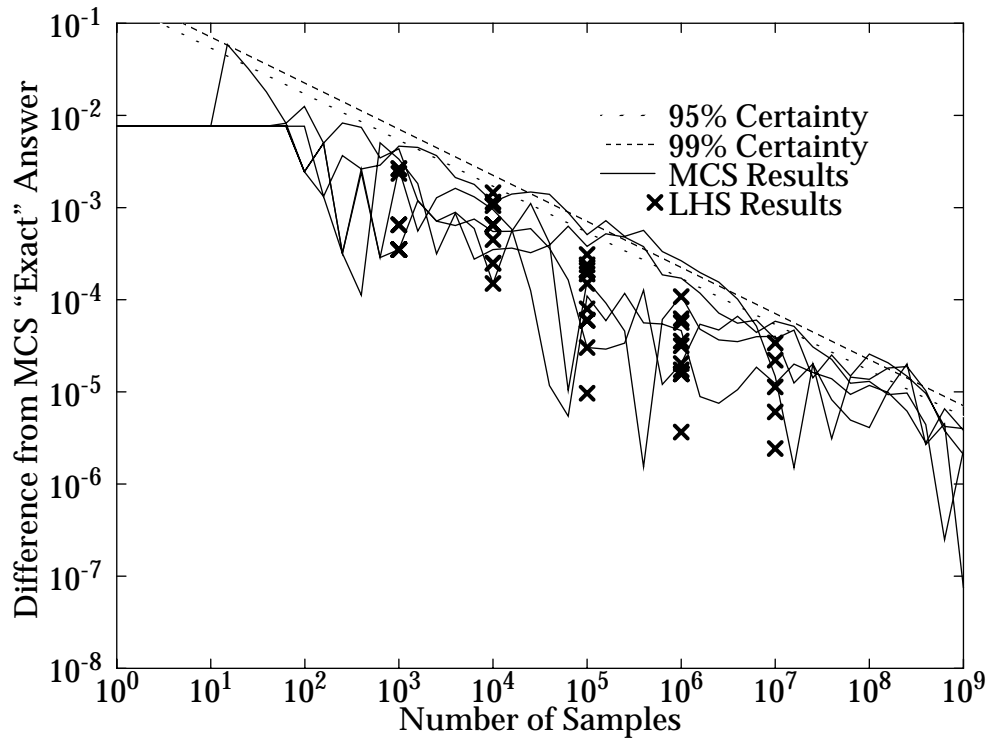


Figure 26 Comparison of MCS Errors to LHS Errors for Problem 1, Threshold = 1.0
Confidence Half-Envelope Calculated from the MCS "Exact" Answer ($\cong 7.650 \times 10^{-3}$)

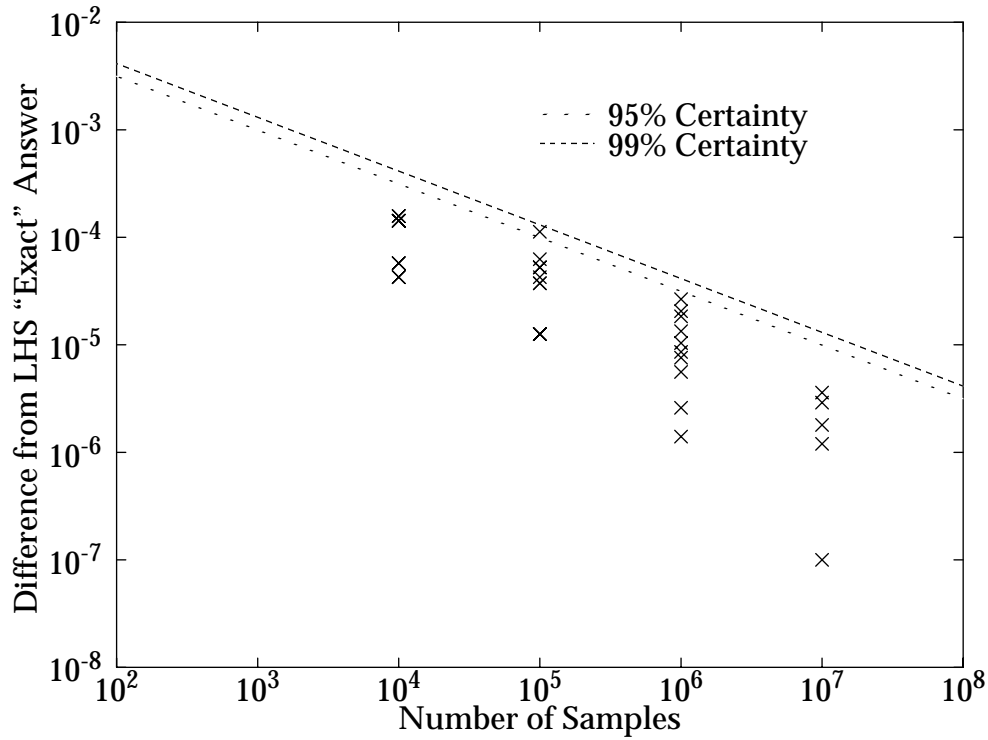


Figure 27 LHS Errors for Problem 1, Threshold = 1.5
Bernoulli Half-Envelope Calculated from the LHS "Exact" Answer ($\cong 2.574 \times 10^{-4}$)

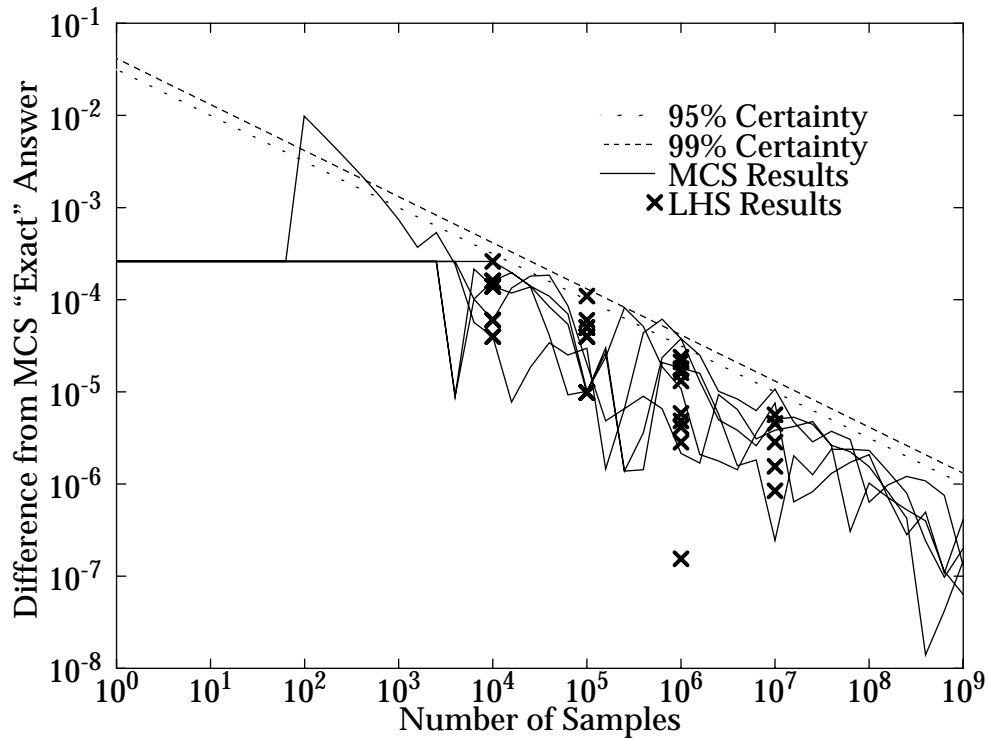


Figure 28 Comparison of MCS Errors to LHS Errors for Problem 1, Threshold = 1.5
Confidence Half-Envelope Calculated from the MCS "Exact" Answer ($\cong 2.602 \times 10^{-4}$)

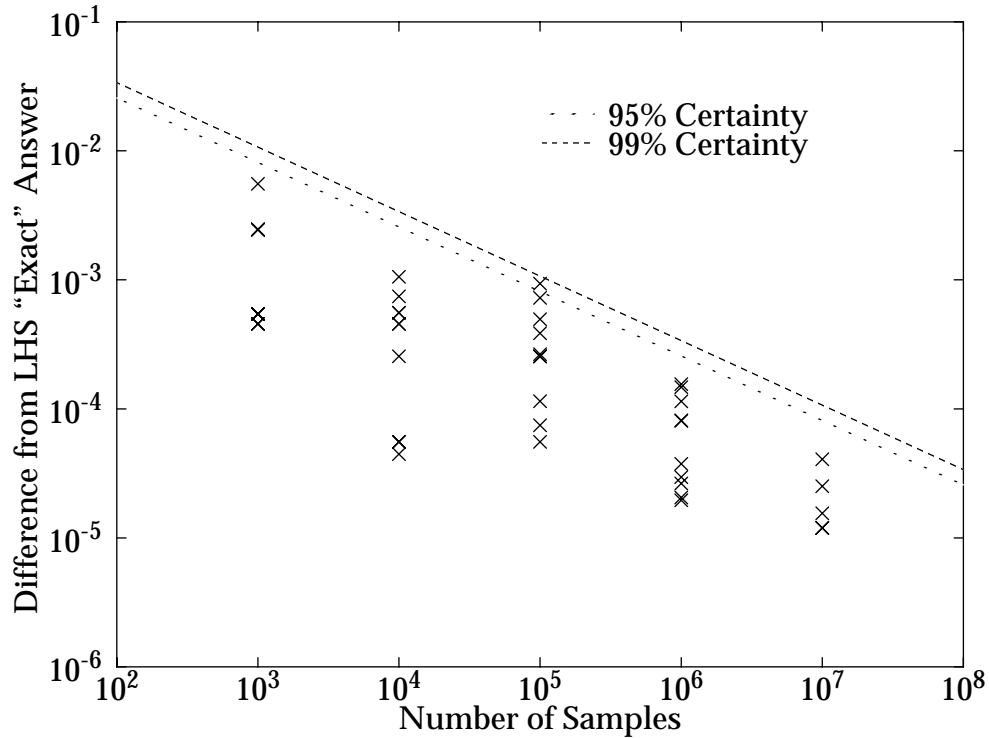


Figure 29 LHS Errors for Problem 2, Threshold = 0.05
Bernoulli Half-Envelope Calculated from the LHS "Exact" Answer ($\cong 0.9825$)

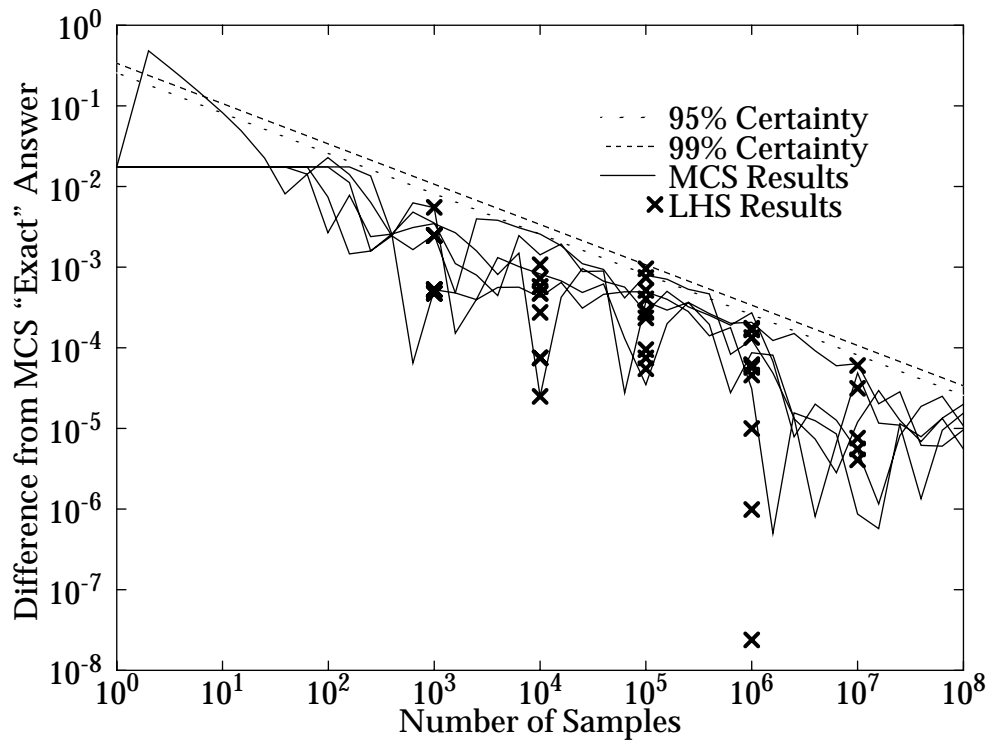


Figure 30 Comparison of MCS Errors to LHS Errors for Problem 2, Threshold = 0.05
Confidence Half-Envelope Calculated from the MCS "Exact" Answer ($\cong 0.9825$)

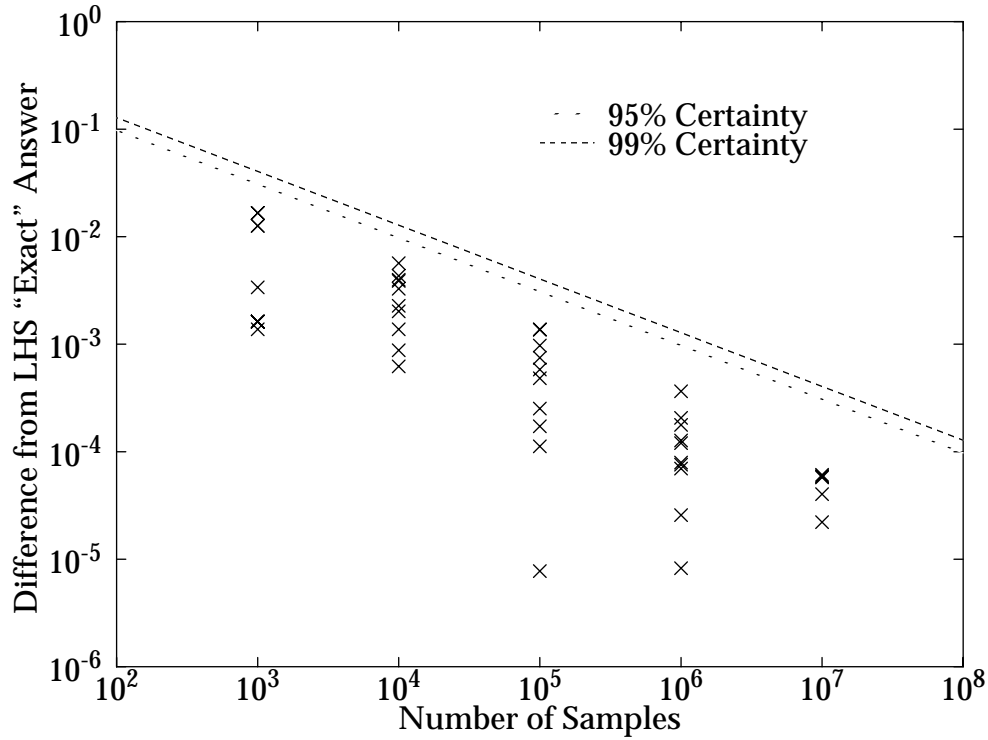


Figure 31 LHC Errors for Problem 2, Threshold = 0.2
Bernoulli Half-Envelope Calculated from the LHS "Exact" Answer ($\cong 0.4474$)

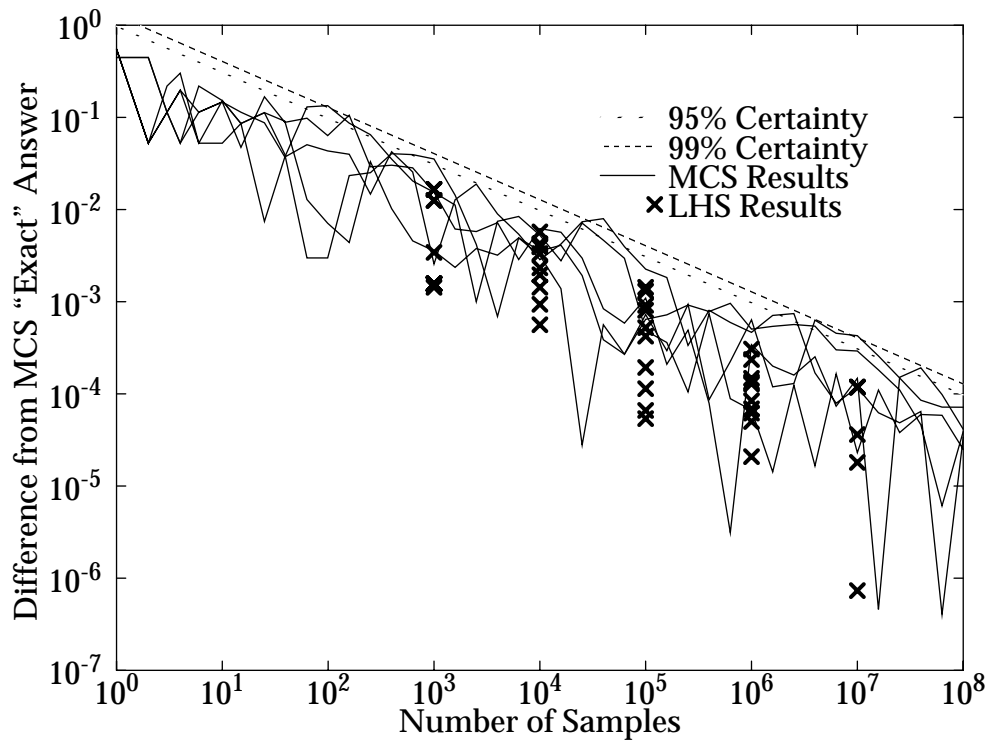


Figure 32 Comparison of MCS Errors to LHS Errors for Problem 2, Threshold = 0.2
Confidence Half-Envelope Calculated from the MCS "Exact" Answer ($\cong 0.4474$)

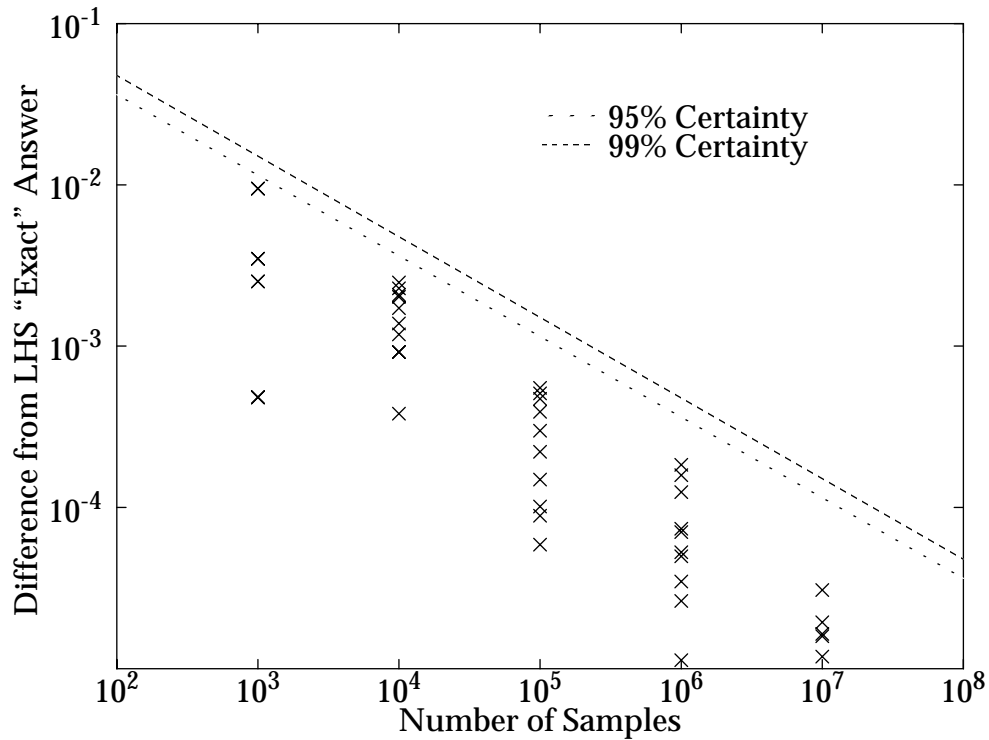


Figure 33 LHS Errors for Problem 2, Threshold = 0.4
Bernoulli Half-Envelope Calculated from the LHS "Exact" Answer ($\cong 3.548 \times 10^{-2}$)

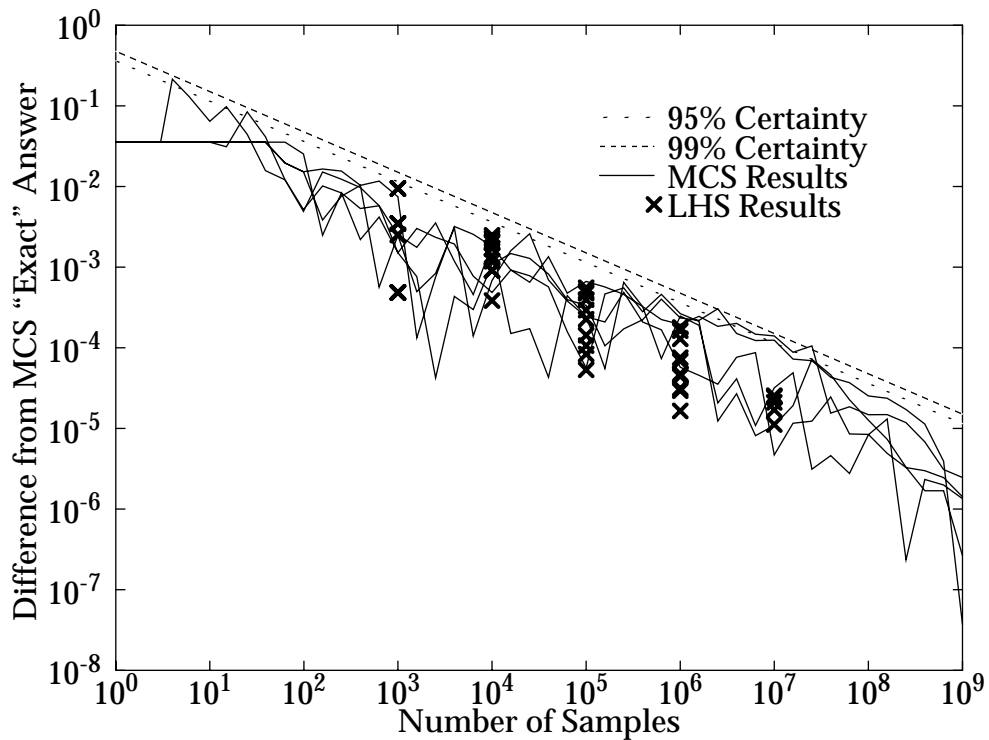


Figure 34 Comparison of MCS to LHS Errors for Problem 2, Threshold = 0.4
Confidence Half-Envelope Calculated from the MCS "Exact" Answer ($\cong 3.549 \times 10^{-2}$)

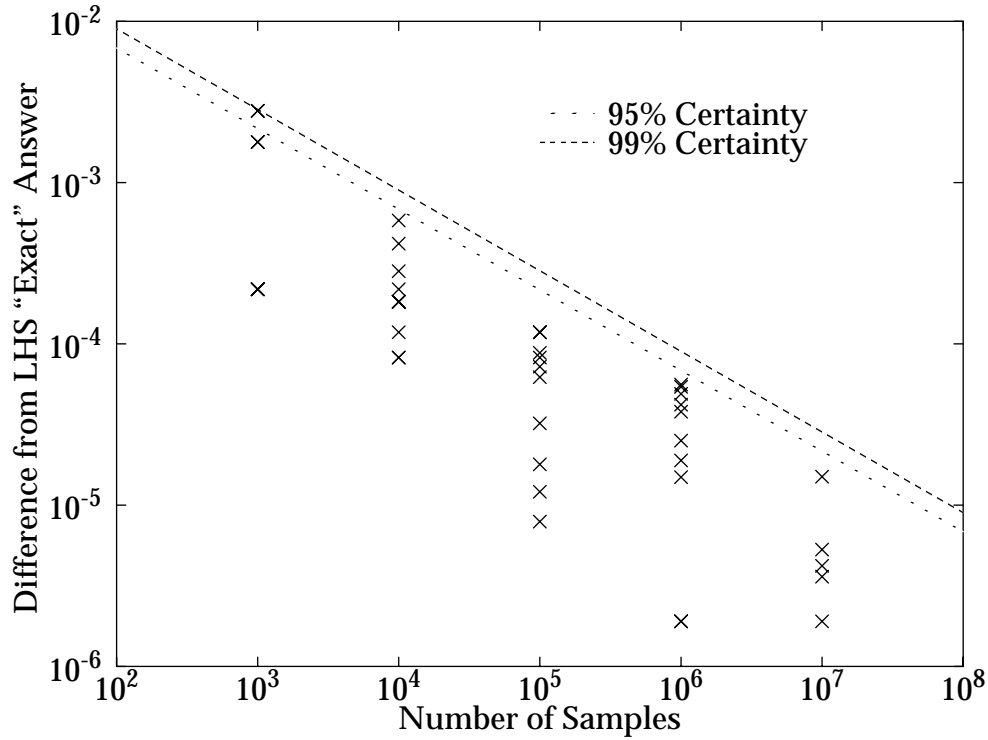


Figure 35 LHS Errors for Problem 2, Threshold = 0.6
Bernoulli Half-Envelope Calculated from the LHS "Exact" Answer ($\cong 1.218 \times 10^{-3}$)

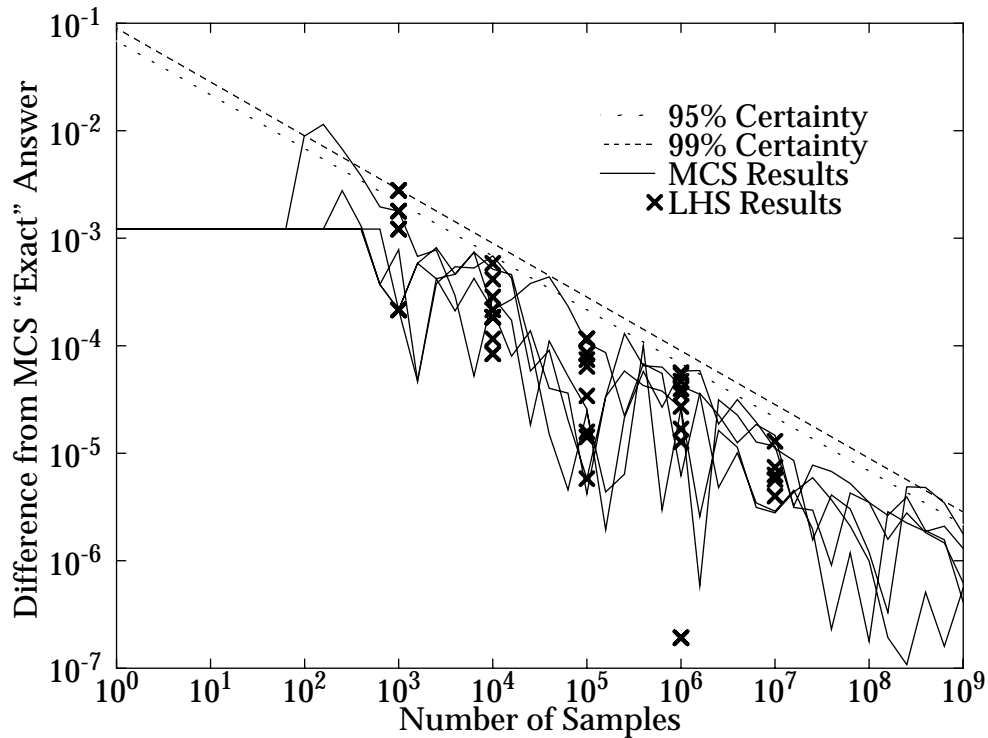


Figure 36 Comparison of Errors MCS to LHS Errors for Problem 2, Threshold = 0.6
Confidence Half-Envelope Calculated from the MCS "Exact" Answer ($\cong 1.216 \times 10^{-3}$)

not seem to be much of a difference. As will be shown later, considering the extra computational time required for LHS sampling, it can be concluded that LHS usually takes longer to achieve the same convergence.

Close inspection of the graphs does bring out a couple of points that need clarifying. First, although there are 10 runs for each LHS sample size except the 10^7 size sample, for the smallest LHS sample size used in each graph, there are less than 10 visible data points, generally around five. The reason for this is that the smaller sample sizes, the greater the granularity of possible results. For example when using 1,000 LHS samples to determine a probability around 1×10^{-3} , one would expect that only 0 to 4 or so of the samples will lead to a failure. Second, when comparing the two graphs in a pair of graphs, one will often find that results do not seem to match up between the graphs. The reason for this is that while the first graph in a pair calculates the “exact” answer from the LHS results, the second graph calculates the “exact” answer from the MCS results. At most, the two “exact” answers differ in the third significant figure. This can cause great differences for results that are near either of the “exact” answers. The larger the error compared to the “exact” answers, the smaller the effect caused by the difference of the “exact” answers.

6 Comparison of Computational Resources Requirements

The most significant difference between the MCS and LHS methods is the computational resources they require. The first part of this section discusses the memory and disk space requirements for each method. The next section discusses timing results. All runs for this report were done on one processor of a Sun workstation with two 296 MHz UltraSparc processors.

6.1 Memory and Disk Space

Disk space requirements depend on how the MCS and LHS methods are implemented. If the samples are generated and used in a single program, the only disk space required is that for output. On the other hand, if the samples are written to disk to be used by another program, the disk space required can be significant. For example, using Iman and Shortencarier’s LHS program to create 10 million samples of two variables each with 14 decimal places of accuracy created an ASCII file 0.48GB in size. File size will increase linearly with the number of samples. If the samples are written to the disk with less precision or as a binary file, the disk space will decrease.

One of the biggest differences between the MCS and the LHS methods is their memory requirements. For MCS, memory requirements are negligible. Random numbers are used as soon as they are generated. Therefore, all that is required is the memory needed to hold the variables used by the random number generator (RNG), and the variables used by the subroutine that converts uniform random numbers to another distribution. If the calculations for the MCS trials are done in the same program that generates the samples, then storage is also required for the variables used in the trials.

The LHS method requires all memory described in the paragraph above and much more. It has to have memory enough to store all the random numbers and then combine them into samples. In the version of the LHSTEST code provided [Romero, 1997], if double precision (8 byte) reals are used, the memory requirement in bytes is given by $8 \cdot \text{NSAMP} \cdot (2 \cdot \text{NVAR} + 2.5)$, where NSAMP is the number of samples and NVAR is the number of variables per sample. The 0.5 of the 2.5 represents a 4 byte integer array used by the code. In the code provided, all reals are declared single precision and promoted to double precision using the -r8 compiler flag. Using this flag, 4 byte integers require 8 bytes of storage. Under these conditions, the 2.5 in the memory requirement equation listed above is replaced by a 3. Either formula can lead to impressive memory requirements. For ten million samples of 2 variables each, 496 to 534 MB of memory are required. The upper range for memory is actually larger than the disk space required to write the results to disk. With the current amount of memory available on machines, most users will be limited to on the order of 100,000 to 1,000,000 samples when using the LHS method.

6.2 Comparisons of Run Time

Timing results for various cases are shown in Table 1. The “Case” column describes the various cases. “P1” stands for problem 1 and “P2” stands for problem 2 (both problems are defined in Section 3). The numbers listed under the “Case” column are the number of samples generated.

Table 1 Timing Information: Total Time (User and System) (s)

Case	MCS No Write	LHS No Write	No Write Ratio LHS/MCS	MCS Write	LHS Write	MCS Single Program	Solve from Sample
P1, 10^6	3.3	40.7	12.3	37	74.9	5.5	36.1
P1, 10^7	33	495.9	16.30	377	849	55.8	368
P1, 10^8						560	
P2, 10^6	3.3			37	74.9	3.5	32.6
P2, 10^7	33			377	849	35.2	333
P2, 10^8	330.8					352	

Several different codes were tested. All LHS samples are generated by code based on the version of the LHSTEST code given to us by Romero [1997]. The results of the previous sections came from three of the codes represented in the table. The program labeled “MCS Single Program” is the program used to obtain the MCS results. This single pro-

gram generates the random numbers, converts them to the proper distributions using the Box-Muller method [Press et al., 1992], and calculates the results for each trial. The only significant output from this program are the final results. The LHS results were obtained using a two step process. First the LHSTEST program was used to generate the normal samples which were then written to disk. The time for this activity is listed in the "LHS Write" column. Next, the samples were read in by a program of our design and the results of the trials were calculated. The time to run this program is listed under "Solve from Sample." Comparing the "MCS Single Program" results to the "Solve from Sample" results, it was noted that the time required to write to the disk is very large. To obtain an idea of how much time the writes to disk require, three more codes were tested. "MCS Write" generates the normal samples and writes them to disk. The programs "MCS No Write" and "LHS No Write" are the "MCS Write" and "LHS Write" modified still to calculate normal samples but not to write them to disk. As can be seen from the table, programs that only generate random samples take the same time for problem 1 and problem 2.

As mentioned before, each code ran on a 296MHz ultraSPARC processor. The workstation has 512 MB of RAM. All codes were compiled with the -O option. The -r8 compile option was also set for the LHS. All codes used RAN2, which is the slowest of RNGs tested by Zeeb and Burns [1999].

It should be noted that it takes 534 MB of RAM to generate ten million LHS samples. Therefore the workstation was forced to use virtual memory to calculate these samples. From the results it appears that the time spent swapping the virtual memory may not have been included in the total time reported. When generating ten million samples, there is a big discrepancy between the total time and the clock time and the penalty for virtual memory in the total time seems small. Therefore, the timing results for ten million LHS samples should not be taken as necessarily correct. It should also be noted that the LHSTEST code we obtained is a large, not necessarily optimized code. It is compared against much smaller, simpler MCS codes that are definitely optimized. Work is currently under way to rewrite the LHC code totally. This rewritten code may be much faster.

By comparing the "no write" and the "write" results, it is shown that MCS is much faster than LHS. MCS is over 12 times faster generating a million samples than LHS. The ratio between the two times increases when ten million samples are generated but it is unclear how much of this change is due to virtual memory penalties. The ratio between the MCS and LCS times depends on which RNG and which distribution are used in the calculations. Faster RNGs and simpler distributions will increase the LHS to MCS time ratio. The lagged Fibonacci generator implemented by Zeeb and Burns [1999] is seven times faster than the generator that was used in this report, RAN2.

Another important fact obtained from the timing results is that the time spent writing to disk is considerable. By comparing the "write" and "no write" results, it is shown that writing a million random number pairs to disk takes about 34 seconds and the time increases linearly as a function of the number of pairs. This time is more than ten times

the amount of time it takes to generate the samples by MCS. It is almost the same amount of time it takes to generate the samples by LHS. What makes this even worse, is that once the samples are written to disk by one program, they must be read by another program. The results above suggest that reading takes just as long as the writing.

7 Conclusions and Recommendations

7.1 Comparing MCS and LHS

In Numerical Recipes, Press et al. [1992] describe LHS as “useful when you must sample an N-dimensional space *exceedingly* sparsely, at M points.” They further note that, “If the response of the system under test is dominated by *one* of the design parameters, that parameter will be found with this sampling technique. On the other hand, if there is an important interaction among different design parameters, then the Latin hypercube gives no particular advantage.”

The results of this report seem to confirm the above statements. Our sample problems were functions of two normally distributed variables. Neither variable seemed to be more important than the other. The LHS method has been found to converge only slightly more quickly than MCS on the benchmark problems. On the other hand, it is much slower than the MCS method. The exact ratio of the two methods depends on the RNG and distribution chosen. In our benchmark problems, compared to the MCS method, the LHS method takes 12 times longer to generate a million samples of two variables each. This is probably a good indication of the general speed differences between the two methods. In the same amount of time, the MCS method can almost certainly generate more accurate results in most cases. The LHS method would be most beneficial in cases where the CPU time to convert the random number to another distribution and/or to calculate the failure probability is very large.

Furthermore, the maximum number of samples generated by the LHS code is limited by memory. Generally only on the order of a million to ten million samples can be generated. The only limit on MCS samples is execution time.

Due to the potential disadvantages LHS method mentioned above, we suggest that further research into replacing the LHS method with the MCS method be done.

7.2 Convergence Criteria

By applying the central limit theorem, two convergence criteria were obtained, the confidence half-envelope, η , and the percent confidence half-envelope, δ . By graphing the progress of 40 MCS runs, it is shown that the MCS results generally fall into a confidence envelope with a probability equal to the probability of that envelope.

Furthermore it is shown that once the current MCS result, F^* , is partially converged, the error in using it to calculate the absolute confidence interval instead of F is negli-

ble. To obtain reasonable MCS probability estimates, the number of trials used must be at least equal to 100 times the inverse of the actual probability. By that point the η calculated by F and F^* are essentially the same. Therefore, using F^* to calculate η as a convergence criteria works well.

η was also tested as a convergence criteria for LHS sampling. η was found to overestimate the uncertainty of the LHS results. None of the 450 results were found to have an error greater than the 99% confidence envelope and only about 1.1% of the results had an error greater than the 95% confidence envelope.

Using δ as a convergence criteria was not tested. Still, simple reasoning suggests that it will be more sensitive to changes in F^* than η , but will still be a very useful convergence criteria.

It is suggested that work be done to start implementing η and δ as convergence criteria. Past experience with Monte Carlo simulations have found good results are obtained using a 95% δ . In the beginning, it may be helpful to run a problem twice using the same confidence half-envelope and seeing how closely F^* agrees between the two results.

7.3 Writing Samples to Disk

Reading and writing the samples to disk was found to be a very time intensive procedure. Writing a million pair of variables to disk by the sample generating program and then having another program read the samples off disk was found to take over 20 times longer than it takes to generate the samples by MCS.

Due to the considerable time savings, it is suggested that work be done to try to combine the sample generating program and the program that processes the samples into one program. The savings in run time and disk space will certainly make this worth the time invested.

8 References

Iman R. L., and Shortencarier, M. J., 1984. "A FORTRAN 77 Program and User's Guide for the Generation of Latin Hypercube and Random Samples to Use with Computer Models," Sandia National Laboratories report SAND83-2365 (RG).

Kreyszig, E., 1993. *Advanced Engineering Mathematics, 7th Ed.*, John Wiley & Sons, New York.

Maltby, J. D., 1990. *Analysis of Electron Heat Transfer via Monte Carlo Simulation*, Ph. D. Dissertation, Department of Mechanical Engineering, Colorado State University, Fort Collins, CO.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., 1992. *Numerical Recipes in FORTRAN: The Art of Scientific Computing, 2nd Ed.*, Cambridge University Press, New York, NY.

Romero, V. J., 1997. Personal Communication, Sandia National Laboratories, Albuquerque, NM.

Romero, V. J. and Bankston, S. D., 1998a. "Monte Carlo Probability Estimation with Finite Element Response Surfaces Built from Progressive Lattice Sampling" presented at the joint AIAA/ASME/ASCE/AHS/ASC 39th Structures, Structural Dynamics, and Materials (SDM) Conference, April 20-23, 1998, Long Beach, CA.

Romero, V. J. and Bankston, S. D., 1998b. "Finite-Element/Progressive Lattice Sampling Response Surface Methodology and Application to Benchmark Probability Quantification Problems," Sandia Laboratories report SAND98-0567, unlimited release.

Ross, S., 1988. *A First Course in Probability, 3rd Ed.*, Macmillan Publishing, New York.

Zeeb, C. N., and Burns, P. J., 1999. "Random Number Generator Recommendation" Report prepared for Sandia National Laboratories, Albuquerque, NM. Available as a WWW document, URL =<http://www.colostate.edu/~pburns/monte/documents.html>.

Zeeb, C. N., Burns, P. J., Branner, K., and Dolaghan, J., 1999. *User's Manual for MONT3D - Version 2.4*, Department of Mechanical Engineering, Colorado State University, Fort Collins, CO. Available as a WWW document, URL =<http://www.colostate.edu/~pburns/monte/manual.html>.

Zeeb, C. N., and Romero, V. J., 1999. "LAYMC: A Monte Carlo Code for Modeling Photon Transport in Layered Participating Media — Theory and User Manual version 2A", to be published as a Sandia National Laboratories report (unlimited distribution), (Expected 99).