NESSUS Theoretical Manual

February 17, 2012

Contents

PROBABILISTIC METHODS NOMENCLATURE

AIS	Adaptive Importance Sampling
AMV	Advanced Mean Value
CDF	Cumulative Distribution Function
EGRA	Efficient Global Reliability Analysis
FORM	First-Order Reliability Method
$F_X(x)$	CDF of X
$f_X(x)$	PDF of X
$g(\cdot)$	Limit State Function (g-function)
MPP	Most Probable Point
MPPL	Most Probable Point Locus
MV	Mean Value
n	Number of Random Variables
p_f	Probability of Failure
PDF	Probability Density Function
SORM	Second-Order Reliability Method
u	Standardized Normal (Gaussian) Variable
X	Basic (Original) Random Variable
$Z(\cdot)$	Performance or Response Function (Z-function)
α	Probabilistic Sensitivity Factor
β	Minimum Distance
$\Phi(u)$	CDF of u
$\phi(u)$	PDF of u
μ	Mean value
σ	Standard deviation
$*$	Superscript denoting MPP (u^*, X^*)

Chapter 1

Introduction

This manual summarizes the available solution methods in NESSUS, listed below.

- First-order reliability method (FORM)

- Second-order reliability method (SORM)

- Standard Monte Carlo method (MONTE)

- Radius-based importance sampling method (ISAMF, ISAMR)

- Adaptive importance sampling methods (AIS1, AIS2)

- Advance mean-based method (MV, AMV, AMV+)

- Efficient Global Reliability Analysis (EGRA)

- Gaussian Process Response Surface Method (RSM_GP)

- System reliability analysis method (FTREE)

Most of the methods are approximate in nature and strive to provide very efficient analysis relative to the standard Monte Carlo method, which provides an asymptotically exact solution as the number of random sampling approaches infinity.

The selection of an appropriate method should be guided by several considerations, including required computational time, which strongly depends on the complexities of the performance functions (e.g., finite element models versus closed form equations), and solution accuracy, which generally depends on the non-linearity and smoothness of the performance functions.

In the following, some background information is given first, followed by the descriptions of the methods. References are cited throughout and listed at the end of the document.

Chapter 2

Basic Definitions and Probability Integration Concepts

2.1 Response, Performance, and Limit-state Function

Many of the reliability analysis concepts originated from structural reliability analysis, where failure conditions are defined using limit states [1–4]. To expand the concept to CDF (cumulative distribution function) analysis, this document adopts the following definitions and notations that distinguish response functions from limit state functions.

A Z-function is a response function or a performance function such as stress, displacement, natural frequency, fatigue life, etc.

$$Z\left(X\right) = Z\left(X_1, X_2, X_3, \ldots, X_n\right) \tag{2.1}$$

where Xi (i = 1,n) are the random variables.

A g-function is a limit state (also called performance function in the literature) defined as:

$$g = Z\left(X\right) - z_0 = 0 \tag{2.2}$$

where z_0 is a particular value of Z. The g-function is defined such that $g(\boldsymbol{X}) = 0$ is a boundary that divides the random variable space into two regions: failure $[g \leq 0]$ and safe $[g > 0]$. Because the CDF of Z at z_0 equals the probability that $[g \leq 0]$, the CDF can be computed by varying z_0 and computing the point probability.

A component reliability problem has only one g-function whereas a system reliability problem involves multiple g-functions.

2.2 Probability Integration

Given a g-function and given the joint probability density function, PDF, $f_X(\boldsymbol{x})$, the probability of failure is the probability in the failure domain Ω and is given by:

$$p_f = \int_{\Omega} \ldots \int f_x\left(x\right) dx \tag{2.3}$$

This integral can be computed using a straightforward standard Monte Carlo procedure. However, when the g-function is complicated, requiring an intensive numerical calculation for each sample of $\boldsymbol{X}$, and p_f is small, this random sampling procedure becomes impractical for engineering analysis and design. For practical purposes, efficient and approximate analysis tools are needed.

2.3 Most Probable Point Concept

Several methods in NESSUS are based on the Most Probable Point (MPP) concept. In the structural reliability literature, the MPP is also known as the Design Point, which is defined in a coordinate system of an independent, standardized normal vector $\boldsymbol{u}$. In the u-space, the joint probability density function (PDF) is rotationally symmetric around the origin and decays exponentially with the square of the distance from the origin. For a two-variable case, the joint PDF has a bell-shaped surface.

By transforming $g(\boldsymbol{X})$ to $g(\boldsymbol{u})$ using a distribution transformation, the MPP ($\boldsymbol{x}^*$ is a point that defines the minimum distance, β, from the origin $\boldsymbol{u} = 0$) to the limit-state surface $g(\boldsymbol{u}) = 0$. This minimum-distance point is a most-probable-point on $g(\boldsymbol{u}) = 0$ (in the u-space) because the joint PDF at a point $(u_1, u_2, \ldots, u_n)$ in the u-space is proportional to $\exp[-0.5(u_1^2 + u_2^2 + \ldots + u_n^2)]$ where the sum of squares defines the distance. Therefore the density is a maximum when the distance is a minimum. The MPP concept is illustrated in Figure 2.1.
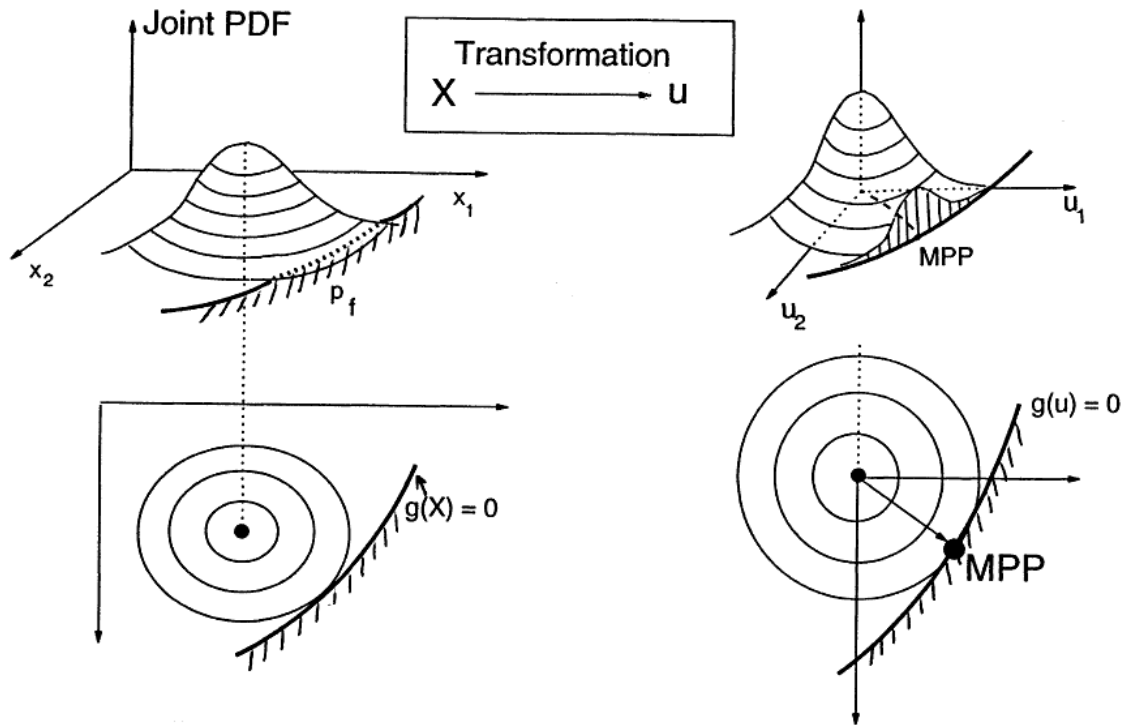
Figure 2.1: Most Probable Point (MPP)

Because of its unique probability properties, the MPP is a key point for fast probability analysis. Once the MPP is identified, it can be used as a basis to develop approximate polynomial g-functions. The approximate g-functions can be analyzed more easily using several methods including the first and second-order reliability methods and the convolution methods. The MPP can also be used as a basis for the adaptive sampling methods described in Section 8.

2.4 Distribution Transformation

Non-normal dependent variables $\boldsymbol{X}$ can be transformed to standardized normal variables $\boldsymbol{u}$ using the Rosenblatt transformation [3–5]:

$$u_1 = \Phi^{-1}\left[F_1\left(x_1\right)\right]$$
$$u_2 = \Phi^{-1}\left[F_2\left(x_2 \mid x_1\right)\right]$$
$$\vdots \tag{2.4}$$
$$u_n = \Phi^{-1}\left[F_n\left(x_n \mid x_1, x_2, \ldots, x_{n-1}\right)\right]$$

where $F_i(x_i)$ is the CDF of X_i, $F_n(X_n \mid \ldots)$ is the conditional CDF, and $\Phi^{-1}(\cdot)$ is the inverse CDF of a standardized normal random variable.

When the variables are mutually independent, this <u>transformation</u> reduces to:

$$u = \Phi^{-1}\left[F_x\left(x\right)\right] \tag{2.5}$$

The <u>inverse transformation</u> is:

$$x = F_x^{-1}\left[\Phi\left(u\right)\right] \tag{2.6}$$

Using the above transformation, the entire g($\mathbf{X}$)-function can be transformed to g($\mathbf{u}$) and allow the probabilistic analysis to be performed in the u-space. Numerically, however, the $\mathbf{X}$-to-$\mathbf{u}$ or $\mathbf{u}$-to-$\mathbf{X}$ transformations are needed only at points required to find the MPP, construct polynomials, and perform importance sampling.

The advantage for transforming to the u-space is that probabilistic analysis becomes mathematically more tractable. The drawback is that the involved transformation may significantly distort the g-function such that an originally flat surface becomes highly curved.

For engineering applications, the Rosenblatt transformation for dependent random variables may be impractical because the available data is often insufficient to establish the joint and the conditional probability distributions. A more realistic model transforms each correlated, non-normal random variable into a normal variable and generates a new set of correlation coefficients for the transformed normals. The generated normal variables are then assumed to have a joint normal density function (which is generally not true) and the correlation coefficients are used to generate a set of independent normal random variables. The inputs required for the second option include only the marginal distributions and the correlation coefficients. This option gives exact solutions for correlated normal random variables.

Consider two random variables X_i and X_j with correlation coefficient R. The correlation coefficient of the transformed normal variables U_i and U_j, denoted as r, can be found by solving the following equation:

$$R = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left(\frac{x_i - \mu_i}{\sigma_i}\right) \left(\frac{x_j - \mu_j}{\sigma_j}\right) \phi_{ij} \, du_i \, du_j \tag{2.7}$$

where

$$\phi_{ij} = \frac{1}{2\pi\sqrt{1 - r^2}} \exp\left(-\frac{u_i^2 - 2ru_iu_j + u_j^2}{2\left(1 - r^2\right)}\right) \tag{2.8}$$

In general, the calculation of r requires iteratively solving the above equation. A convenient approach relating R and r was proposed in Ref. 6 and implemented in NESSUS.

2.5 Most Probable Point (MPP) Search Procedure

The MPP is the key approximation point for the MPP-based methods. Therefore, the identification of the MPP is an important task. In general, the identification of the MPP can be formulated as a standard optimization problem (i.e., find the minimum distance point subject to g=0) and can be solved by many optimization methods. In NESSUS, there are two procedures for finding the MPP. One is implemented specifically for the AMV methods (described in Section 9) and the other is summarized in the following steps:

(a) Assume an MPP $\boldsymbol{x}^*$.

(b) Compute equivalent normal distributions for the non-normal random variables at $\boldsymbol{x}^*$.

(c) Construct a linear g-function using $\boldsymbol{x}^*$ as the expansion point.

(d) Based on the equivalent normal distributions, compute the minimum distance and the updated MPP in the u-space.

(e) Repeat steps (b) to (d) until the MPP converges.

In (c), the equivalent normal distribution is computed by the Rackwitz-Fiessler algorithm [1,3,6] which matches the CDF's and the PDF's for the original and the equivalent normal distribution at the MPP point,

$$F_x \left(x^*\right) = \Phi \left(u^*\right) \tag{2.9}$$

and

$$f_x \left(x^*\right) = \frac{\phi \left(u^*\right)}{\sigma_N} \tag{2.10}$$

where

$$u^* = \frac{x^* - \mu_N}{\sigma_N} \tag{2.11}$$

The parameters μ_N and σ_N are the mean and the standard deviation of the approximate normal and $\phi(\cdot)$ is the standard normal PDF.

From the above three equations, the approximate normal parameters can be derived as:

$$\sigma_N = \frac{\phi \left[\Phi^{-1} \left(F_x \left(x^*\right)\right)\right]}{f_x \left(x^*\right)} \tag{2.12}$$

$$\mu_N = x^* - \Phi^{-1} \left[F_x \left(X^*\right)\right] \sigma_N \tag{2.13}$$

According to the principle of normal tail approximation [7], the MPP obtained by the approximate normal is identical to the one obtained using the inverse transformation. In the equivalent normal approach, which is the approach used in NESSUS, the inverse transformation is not needed.

The above iteration algorithm has been found to be quite efficient and robust. Many other algorithms have been proposed in the literature [8–10]. However, just like using any optimization methods, there is no guarantee that the procedure will always converge. Furthermore, the possibility exists that there are multiple (local and global) MPP's. Some multiple MPP problems may be identified and solved by using the AMV procedure described in Section 9. In general, experience is required in selecting methods. For non-convergent problems, the AMV methods should be tried followed by the importance sampling method (ISAMR, described in Section 7), and, as the last resort, the standard Monte Carlo method.

2.6   Probability Sensitivity Analysis

In deterministic analysis, sensitivity is defined as $\partial Z / \partial X_i$, which measures the change in the performance due to the change in a design parameter. In probabilistic analysis the sensitivity measure is $\partial p / \partial \theta_i$, which measures the change in the probability relative to the change in a distribution parameter (e.g., mean and standard deviation).

Another useful probability sensitivity analysis is the determination of the relative importance of the random variables. This can be done by performing several probabilistic analyses in which one of the random variables is treated as a deterministic variable (i.e., by reducing the standard deviation to zero) for each analysis. Based on the resulting probability changes, the relative importance of the random variables can be determined. Repeated analyses, however, may be very time consuming for large numbers of random variables.

A more efficient way of evaluating the relative importance of the random variables is based on the location of the MPP. At the MPP, $\boldsymbol{u}^* = (u_1^*, u_2^*, \ldots, u_n^*)$, the first-order probability estimate is $\Phi(-\beta)$ where

$$\beta^2 = u_1^{*2} + u_2^{*2} + \dots u_n^{*2} \qquad (2.14)$$

The unit normal vector at the MPP of the g=0 surface is defined as:

$$\alpha = -\frac{\nabla g}{|\nabla g|} \qquad (2.15)$$

The α vector is positive towards the direction of decreasing g (i.e., to the failure region). The sensitivity factors are the projections of the α vector to the u-axes. Thus, they are the directional cosines of the α vector as shown in Figure 2.2, and can be written as:

$$\alpha_i = \frac{u_i^*}{\beta} \qquad (2.16)$$

The directional cosines satisfy the following rule:

$$\alpha_1^2 + \alpha_2^2 + \dots \alpha_n^2 = 1 \qquad (2.17)$$

which implies that each α_i^2 is a measure of the contribution to the probability (since the probability is related to β); higher α (in magnitude) indicates higher contribution. Thus, the sensitivity factors provide a first-order information on the importance of the individual random variable.
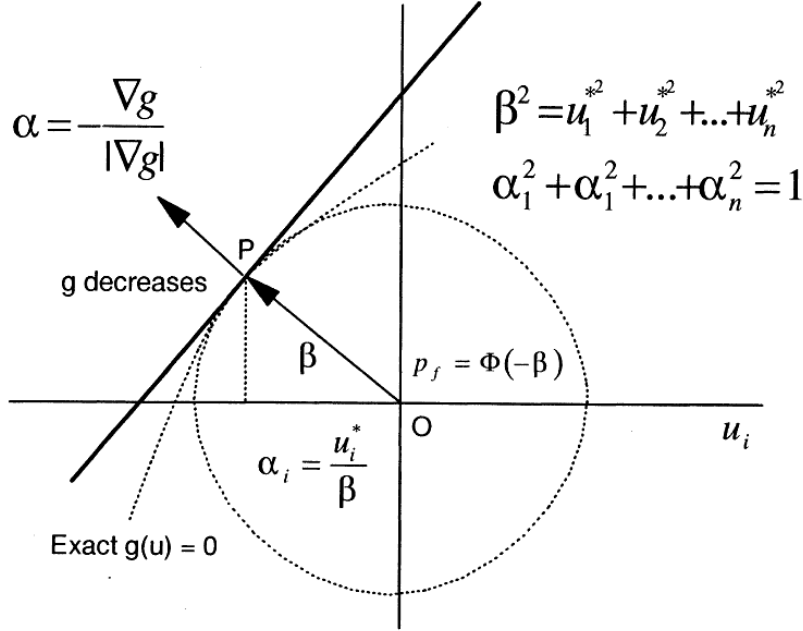
Figure 2.2: Definition of Sensitivity Factors

It can be shown that in the u-space,

$$|\alpha_i| \propto \left(\frac{\partial g}{\partial u_i}\right)_{u^*} \qquad (2.18)$$

and in the X-space,

$$|\alpha_i| \propto \left(\frac{\partial g}{\partial X_i}\right)_{x^*} \sigma_i \qquad (2.19)$$

where σ_i is the normal (or approximate normal for non-normal distribution) standard deviation. It can be concluded that the sensitivity factors are functions of *both* the deterministic sensitivity and the uncertainty (characterized by the standard deviation).

9

In general, the sensitivity factors depend on the g-function as well as the input probability distributions. In a CDF analysis, the sensitivity factors will usually be different for different response or probability levels. This is because the performance sensitivity or the approximate standard deviation may be different for different response or probability levels.

Because the above probabilistic sensitivity analysis is based on the first-order reliability method, α is a good probability sensitivity measure only if $\Phi(-\beta)$ is a good approximation to the true probability (see Section 3 for more information).

Based on MPP, other sensitivity measures with respect to a distribution parameter (mean or standard deviation) or a limit-state function parameter can be computed based on the sensitivity factors and the distribution transformation [4]. However, the computation has not yet been implemented. Section 8 describes a sampling-based method for probabilistic sensitivity analysis.

Chapter 3

First and Second-Order Reliability Methods (FORM and SORM)

The FORM probability solution is based on the linearization of the g-function at the most probable point (MPP) in the u-space. The first-order polynomial, $g_1(\boldsymbol{u})$, is:

$$g_1(u) = a_0 + \sum_{i=1}^{n} a_i (u_i - u_i^*) \tag{3.1}$$

Given $g_1(\boldsymbol{u})$, the probability of failure is a function of the minimum distance to the plane defined by g1 in the u-space [3, 4, 6]:

$$p_f = \Phi(-\beta) \tag{3.2}$$

where β is computed from

$$\beta = \frac{\mu_g}{\sigma_g} = \frac{a_0 + \sum_{i=1}^{n} a_i \mu_i}{\sqrt{\sum_{i=1}^{n} a_i^2 \sigma_i^2}} \tag{3.3}$$

which is allowed to take negative values. A negative β means the origin is in the failure region (i.e., for the $p_f > 0.5$ case) and the α vector is positive from the MPP to the origin. Figure 3.1 summarizes the FORM results.

The SORM analysis requires a second-order polynomial, $g_2(\boldsymbol{u})$,

$$g_2(u) = a_0 + \sum_{i=1}^{n} a_i (u_i - u_i^*) + \sum_{i=1}^{n} b_i (u_i - u_i^*)^2 + \sum_{i=1}^{n} \sum_{j=1}^{i-1} c_{ij} (u_i - u_i^*)(u_j - u_j^*) \tag{3.4}$$

The definition of $g_2(\boldsymbol{u})$ requires $n(n-1)/2$ second order derivatives. NESSUS computes the second-order coefficients by numerical differentiation. Given $g_2(\boldsymbol{u})$, approximate or exact solutions are available [11–13]. A special case of $g_2(\boldsymbol{u})$ is a parabola which has an exact integral solution for p_f [12]. Breitung derived the following asymptotic formula for large β [13]:

$$p_f \approx \Phi(-\beta) \prod_{i=1}^{n-1} (1 - \beta \kappa_i)^{-1/2} \tag{3.5}$$

where κ_i are the principal curvatures at the most probable point.

The first and second-order methods, as well as other MPP-based methods (including the advanced first-order, the convolution method, and the AMV methods), attempt to approximate the original g-functions by simple functions. It has been reported that these MPP-based methods perform well for a wide range of applications. However, it should be cautioned that some nonlinear functions (e.g., a sine wave function) cannot be approximated well by low-order polynomials. In practice, when a performance function is implicitly
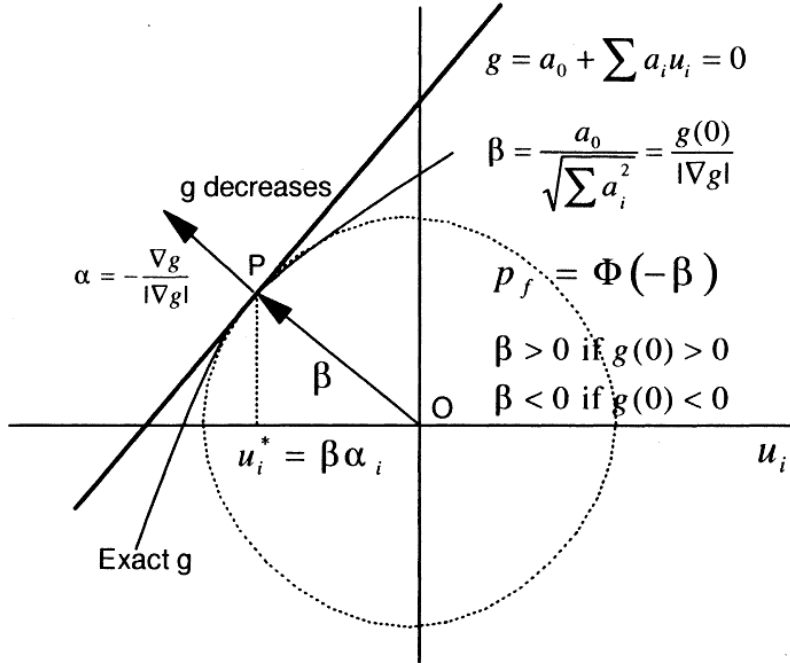
Figure 3.1: First Order Reliability Method

defined and is highly nonlinear, it is difficult to assess the error introduced by the polynomial approximation. Therefore, for any new type of g-functions, the approximate solutions should be checked using other more accurate methods.

Chapter 4

Standard Monte Carlo Method (MONTE)

This method repeatedly generates a set of random values from the joint PDF, $f_X(x)$, and computes corresponding Z-function values. The Z-values are used to construct a CDF for Z or compute the p_f estimate for $Z < z_0$:

$$p_f = \frac{N_f}{N} \tag{4.1}$$

where N is the total number of samples and N_f is the number of samples with negative g values. Figure 4.1 illustrates a Monte Carlo procedure for a reliability analysis.

Figure 4.1: Reliability Analysis by Monte Carlo Simulation

By specifying an error bound and a confidence interval, and assuming that the estimated probability follows a binomial distribution, the number of samples can be computed as follows:

$$\% \, \text{error} = 100 \cdot \Phi^{-1}\left(1 - \frac{\alpha}{2}\right)\sqrt{\frac{1-p}{Np}} \tag{4.2}$$
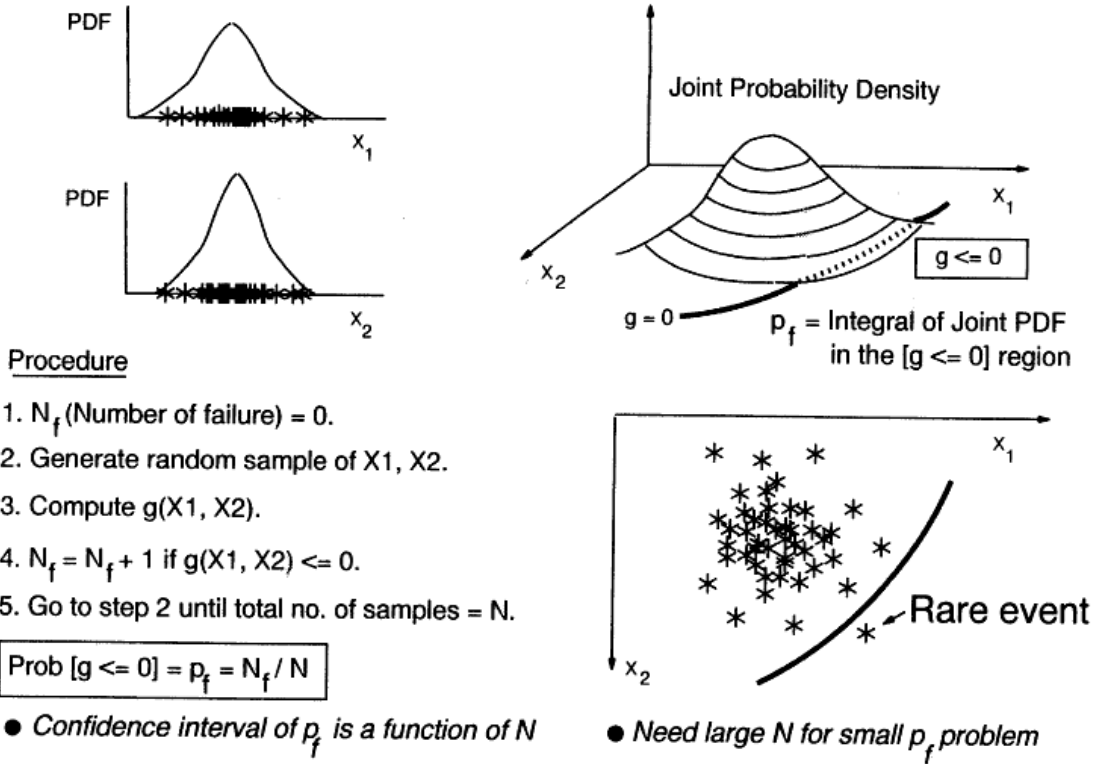
For example, for 10% error with 95% confidence, the required number of samples is

$$N = 396\frac{1-p}{p} \tag{4.3}$$

The probability p is unknown, but can be estimated based on a preliminary analysis with a relatively small sample size and updated with more samples.

In NESSUS, for user-specified multiple Z-levels, the same random samples will be used for all the Z-levels. For a CDF analysis, twenty one Z-levels will be determined automatically. The range of Z is mean± 5 standard deviations where the mean and the standard deviation are estimated using 1000 samples. The increment is 0.5 standard deviation. In addition to the CDF values, the sample mean and standard deviation will be computed.

Whenever practical, this method should be used to supplement other methods to gain experience and confidence in using other more efficient methods. A sample size of 10,000 may not be enough for very small p_f, but may be sufficient for a CDF analysis.

Chapter 5

Radius-based Importance Sampling (ISAMF and ISAMR)

In the radius-based importance sampling methods, samplings are performed in the standardized normal (u) space outside of a sphere (or hyper-sphere for n > 3) as illustrated in Figure 5.1 using a sampling scheme developed by Harbitz [14]. For each sample generated in the u-space, the inverse transformation is used to generate a sample in the X-space for g-function evaluations. The basic assumption is that if the MPP is correct, then according to the MPP definition [g > 0] within the sphere (even though we do not know their values). Therefore, sampling is not needed within the sphere and the sampling efficiency is always better than the standard Monte Carlo method except for the special case of zero radius, which is equivalent to the standard Monte Carlo method.
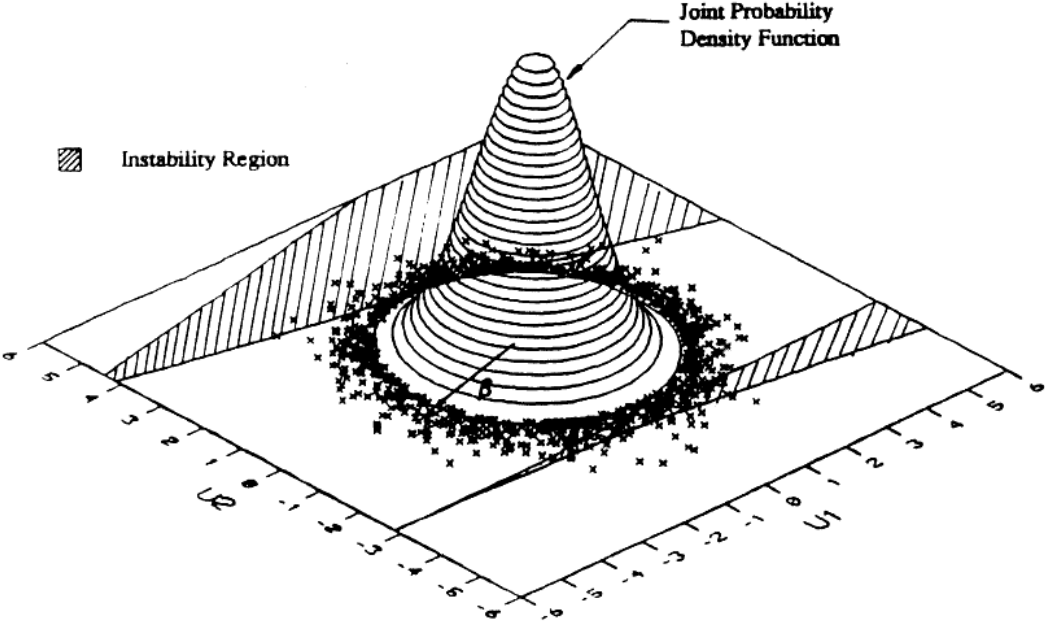

Figure 5.1: Radius-Based Importance Sampling Method

The p_f estimate is:

$$p_f = \frac{N_f}{N} \cdot P_{\text{sampling region}} \tag{5.1}$$

15

where N is the total number of samples (outside the sphere), N_f is the number of samples with negative g values, and psampling region is the probability outside the sphere.

As discussed in Section 6, by specifying an error bound and a confidence interval, the number of samples can be computed as follows:

$$\% \,\text{error} = 100 \cdot \Phi^{-1} \left(1 - \frac{a}{2}\right) \sqrt{\frac{1-p}{Np}} \qquad (5.2)$$

where the probability p is the (conditional) probability of failure in the sampling region, i.e.,

$$p = \frac{N_f}{N} \qquad (5.3)$$

which can be estimated using a preliminary small sample and updated with more samples.

When the ISAMF (the letter F denotes Factor) method is used, NESSUS will first search for the MPP to define the reference radius (i.e., the minimum distance), then use the reduction factor (F) to reduce the sampling radius and increase the sampling region. For example, if $\beta = 3$, and F = 0.9, the sampling radius will be 2.7. This reduction factor can be used to check the accuracy of β. If the calculated β is correct, reducing the sampling radius will not generate more $[g < 0]$ samples.

When the ISAMR (the letter R denotes Radius) method is used, NESSUS will not locate the MPP. Instead, a user-specified radius will be used. This method is useful for problems where NESSUS has difficulty finding the MPP or the user suspects that the calculated MPP is not a global minimum distance point. The user can run several cases with different radius to estimate β. This is particularly useful for problems with very small p_f (e.g., $< 1.e\text{-}7$) that would require a very large number of samples if the standard Monte Carlo method were used.

Both methods must be used in combination with ANALTYPE 1 (i.e., ZLEVELS). For multiple z_0 values, NESSUS will treat each z_0 value as a separate problem, i.e., the code will regenerate random samples for the second and subsequent z_0 values. (Note: The current procedure, which was designed for a single z_0 analysis, might be enhanced by avoiding redundant sampling for multiple z_0 analysis).

Chapter 6

Adaptive Importance Sampling (AIS1 and AIS2)

6.1 Summary

The Adaptive Importance Sampling (AIS) methods minimize sampling in the safe region by adaptively and automatically adjusting the sampling space from an initial approximation of the failure region [15, 16]. The sampling space is defined using a limit-state surface (boundary). The performance of AIS depends on the quality of the initial failure region approximation. Even though the AIS method cannot totally replace the standard Monte Carlo method, it provides efficient accuracy improvement or checking to the MPP-based approximation methods (FORM, SORM, AMV+) for component reliability analysis. AIS also provides an efficient alternative to standard Monte Carlo for system reliability analysis.

There are two methods (AIS1, AIS2) available in NESSUS for selecting the sampling limit-state boundaries. AIS1 uses planes and AIS2 uses parabolic surfaces (see Figure 6.1). Both surfaces are constructed in the u-space and use the MPP to define the beginning sampling limit state. AIS1 changes the distance to the plane and AIS2 changes the MPP curvature. For most problems tested to date, both methods performed equally well with AIS2 having a slight edge. AIS1 is less sensitive to the accuracy of the initial MPP but may require more samples. AIS2 is more sensitive to the accuracy of the initial MPP but is generally more efficient.
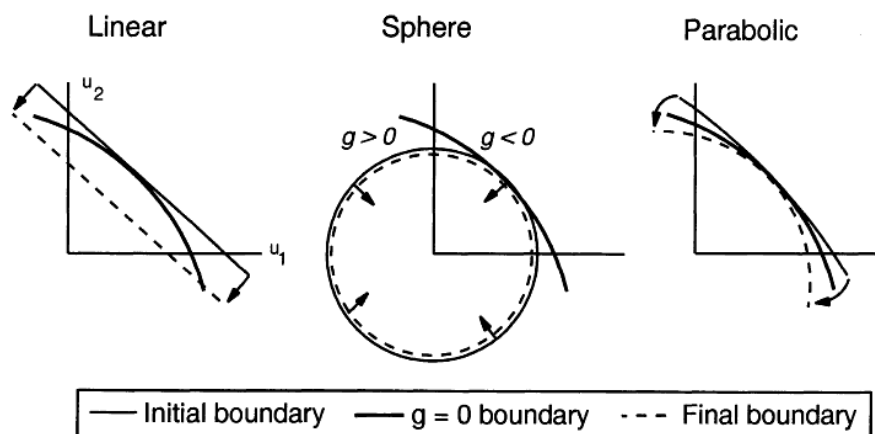

Figure 6.1: Illustration of the AIS Concept

For user-defined g-functions, AIS is highly recommended because of its overall effectiveness in accuracy and efficiency. It should be cautioned, however, that the performance of AIS depends on the accuracy of the MPP. If the starting MPP is far away from the true MPP, AIS may either require a larger number of

samples than usually required or may miss a failure region. Therefore, if NESSUS fails to find the MPP, AIS may not be efficient or may converge incorrectly.

6.2 AIS Methodology

Consider, first, a single limit state. The AIS approach has the following features: (1) the sampling density is proportional to the joint PDF of the random variables, (2) the initial sampling domain is the failure side of the limit state surface constructed from an approximate (e.g., parabolic) surface, (3) additional samples are added after incrementally increasing the sampling region (e.g., by changing the curvatures of the parabolic surface at the most probable point), and (4) the final sampling domain contains, but is only slightly greater than, the failure domain. The objective is to minimize over-sampling in the safe region. Sampling in the safe region is needed only to identify the limit state surface.

To minimize sampling, a good adaptive surface should be able to represent the exact limit-state surface closely. Compatible with the MPP-based methods, there are several possibilities in choosing and varying the sampling limit state surface as illustrated in Figure 6.1, where three kinds of sampling boundary surfaces are shown.

The radius or sphere-based method is the simplest to implement and is the most robust because the initial MPP is only used to define an initial sampling radius. The disadvantage is that the sampling efficiency (related to the probability inside the sphere) decreases as the number of random variables, n, increases.

The plane-based method is also easy to implement and the sampling efficiency depends on the minimum distance and is not affected by n. However, the initial MPP is important in this method and the method is, therefore, not as robust as the sphere-based method and not as efficient as the curvature-based method.

The curvature-based method adaptively changes the curvature of the parabolic surface at the MPP. The method's efficiency and robustness relies on the accuracy of the initial MPP but is generally the most efficient among the three methods because of it's ability to closely envelop the failure region. The curvature-based AIS concept is further illustrated in Figure 6.2, in which the parabolic surface is rotationally symmetric about the vector OP that passes through the origin and the MPP. The sampling domain can be adjusted by increasing or decreasing the curvatures of the parabolic surface. By decreasing the curvatures to the extreme ($-\infty$), the sampling region will cover the entire space. However, if β is correctly calculated, sampling inside the sphere is not needed. Therefore, the AIS2 bound is a sphere, as illustrated in Figure 6.2. For additional checking, the distance to the MPP can also be reduced in case the calculated is in error.
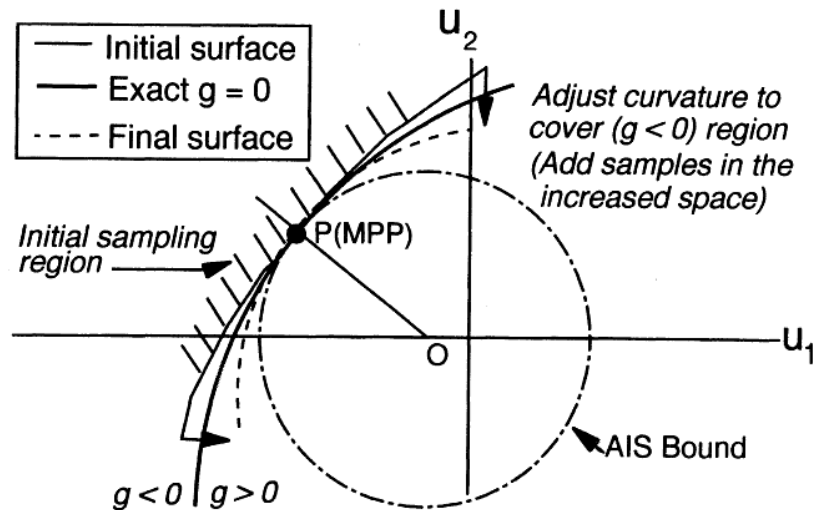

Figure 6.2: Curvature-based AIS Method

To describe how the curvature-based AIS method is related to other methods in NESSUS (AMV+, FORM, and SORM), assume that an initial second-order limit state is available in the form

18

$$g(u) = \nabla g(u^*)^T (u - u^*) + \frac{1}{2}(u - u^*)^T H(u^*)(u - u^*) \tag{6.1}$$

where $\nabla g(\boldsymbol{u}*)$ is the gradient at the MPP, and $\boldsymbol{H}$ is the Hessian matrix containing second-order derivatives. Because Equation 6.1 will be used to define an initial sampling domain for the AIS, only a rough estimate is needed. A linear $g(\boldsymbol{X})$-function generated by the AMV+ method (See Section 9) can be transformed to the u-space and used to generate a second-order $g(\boldsymbol{u})$-function. The transformation procedure to develop a parabolic surface is well documented in the literature [12] and is given by the equation

$$g = \beta - \nu_n + \sum_{i=1}^{n-1} \lambda_i \nu_i^2 \tag{6.2}$$

in which ν_i are independent standardized normal variables, and λ_i are related to the main curvatures by the relation $\kappa_i = 2\lambda_i$. The curvature-based AIS procedure uses Equation 6.2 to develop the initial adaptive sampling boundary and generate the samples in the $[g < 0]$ domain. The corresponding probability in the sampling region, p_s, can be numerically computed [17].

To change the sampling space, the λ_i are changed to λ_i'. The changes in λ_i can be made individually or simultaneously based on a selected probability increment Δp_s. Define the changed or perturbed limit state surface as

$$g' = \beta + \sum_{i=1}^{n-1} \lambda_i' \nu_i^2 \tag{6.3}$$

where the superscript (prime) indicates a perturbed condition.

Let N_1 be the initial number of samples determined based on a confidence interval and an error bound. During the sampling process, this number will be adjusted based on the updated probability of failure in the sampling region. The number of failure points in p_s is denoted as N_1^+. Given a set of λ_1', the sampling is applied to an increased sampling ΔS region. A predetermined number of samples, ΔN, in the ΔS region, is calculated using

$$\Delta N = (p_s' - p_s) N_1 = \Delta p_s N_1 \tag{6.4}$$

By computing the original g-function, the number of failure points, ΔN^+, within ΔX, can be determined and the updated probability estimate becomes

$$\hat{p} = p_s' \cdot \frac{N_1^+ + \Delta N^+}{N_1 + \Delta N} \tag{6.5}$$

The perturbation procedure can be repeated until no more failure points are observed after at least one perturbation. A more detailed computational procedure is given in Ref. 18.

Assuming that the sampled p estimate has a normal distribution, for a $(1 - \alpha)$ confidence interval, the error bound, γ (in percentage), is given by

$$\gamma = \Phi^{-1}\left(1 - \frac{\alpha}{2}\right)\sqrt{\frac{1 - \hat{p}}{\hat{p}N}} \tag{6.6}$$

where $\Phi(\cdot)$ is the standard normal distribution function and $\hat{p}$ is the probability of failure estimate in the importance sampling region. Equation 6.6 can be used to calculate the required samples, N, given an estimated $\hat{p}$. Ideally, we wish to apply AIS to a region very close to the exact failure region resulting in $\hat{p} = 1$, and $N = 0$. This, however, can never be achieved because the sampling region must be greater than the failure region to indicate that the failure region has been covered sufficiently. In addition, the parabolic surface may not be a sufficiently flexible surface to approximate the true surface.

The convergence procedure implemented in the NESSUS code is as follows:

1. Generate samples in the initial sampling domain S. Begin with a small sample (10 points) to estimate $\hat{p}$. Then gradually increase the samples until the error bound γ is acceptable for a selected confidence level.

2. Reduce the curvatures (average curvatures are used in NESSUS) such that the increased probability Δp is a fraction (10%) of the initial probability p. Generate samples only in the increased domain S.

3. Update the probability estimate. Stop the analysis if the probability estimate converges. Otherwise, decrease the curvature again.

4. In case β is in error, slightly reduce the minimum distance (i.e., shift the parabolic surface towards the origin) and generate additional samples. Stop the analysis if no more failure points are observed.

6.3 AIS for System Reliability

In NESSUS a system failure is defined using a fault tree, which provides a way to manage multiple failure modes. The limit state functions are defined in the bottom events. Sequential failures can be modeled using the PRIORITY AND gate. A sequence of g-functions, corresponding to a sequence of updated structural configurations with load redistribution can be explicitly or implicitly defined in the bottom events.

Through a fault tree, all the failure modes can be defined. A failure mode can involve one or more limit states. By adding all the failure modes, and therefore all the limit states, the system limit state surface can be constructed piece by piece. The AIS procedure for system reliability analysis requires the construction of multiple parabolic surfaces. In principle, it is a straightforward extension of the concept for one limit state. The difficult part is to develop a procedure to add failure regions.

The approach implemented in NESSUS adds samples progressively starting from the most important limit state, based on the approximate functions. This approach is ineffective for cases where system failure is governed by the joint effects from several limit states. In such cases, no limit state can be considered as dominate because the MPP of the individual limit state is not a likely event for a system failure. A more effective computational procedure adds samples progressively based on failure modes. The initial sampling regions are based on the approximate limit states generated by the AMV+ method (Section 9) and the increased sampling regions are based on the perturbed limit states.

Figure 6.3 illustrates the above curvature-based AIS procedure for a system with two failure modes that involve three limit states. The probability of failure statement is

$$p_f = P\left\{(g_1 < 0) \cup (g_3 < 0)\right\} \tag{6.7}$$
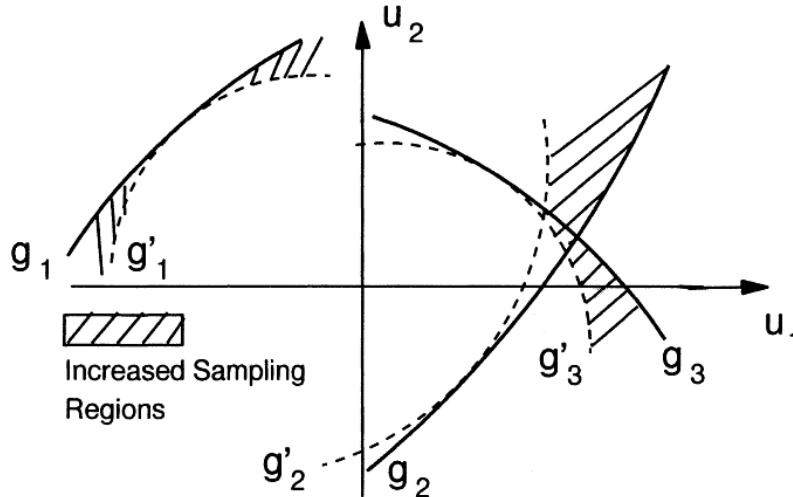

Figure 6.3: AIS for System Reliability

The above adaptive approach for system reliability can be easily modified for use with the radius-based and plane-based methods. However, the curvature-based AIS is recommended for its overall performance in efficiency and robustness.

6.4 Reliability Sensitivity with Respect to a Distribution Parameter

The AIS-based sensitivity analysis, not automated in the current version of NESSUS, can be performed using the AIS-generated samples.

When a distribution parameter is changed, the sensitivity of the probability of failure (denoted as p for simplicity) with respect to a distribution parameter, θ, can be evaluated using

$$\frac{dp}{d\theta} = \int_\Omega \cdots \int \frac{df_X}{d\theta} dx \tag{6.8}$$

Therefore,

$$\frac{dp/p}{d\theta/\theta} = \int_\Omega \cdots \int \frac{\theta}{p} \frac{df_X}{f_X d\theta} f_x dx = \int_\Omega \cdots \int \frac{\theta df_X}{f_X d\theta} \frac{f_X}{p} dx \tag{6.9}$$

or

$$\frac{d \ln p}{d \ln \theta} = E \left[\frac{d \ln f_X}{d \ln \theta} \right]_\Omega \tag{6.10}$$

where the expected value is evaluated using the PDF in the failure region. Because the AIS approach uses the same PDF, the probability sensitivity can be computed using those AIS points in the failure region. No additional g-function calculations are required.

Two useful dimensionless probability sensitivity coefficients, S, are

$$S_\sigma = \frac{d \ln p}{d \ln \sigma} \tag{6.11}$$

$$S_\mu = \frac{d \ln p}{d\mu/\sigma} \tag{6.12}$$

For some random variables, (e.g., independent normal and lognormal variables), these coefficients can be analytically derived. For example, let μ_i and σ_i be the mean and the standard deviation, respectively, of the normal random variable i. It can be shown that

$$\frac{d \ln p}{dk \ln \sigma_i} = E \left[u_i^2 - 1 \right]_\Omega \tag{6.13}$$

and

$$\frac{d \ln p}{d\mu/\sigma_i} = \frac{1}{\sigma_i} E \left[u_i \right]_\Omega \tag{6.14}$$

Assuming that a small change in θ, denoted as $\Delta\theta$, results in a small change in p, Δp, the changes in p can be approximated by

$$\Delta p \dot{=} p \bullet S \bullet \frac{\Delta\theta}{\sigma} \tag{6.15}$$

A random variable approaches deterministic if σ approaches 0, i.e., $\Delta\sigma = -\sigma$. In this case,

$$\Delta p \dot{=} - pS_\sigma \tag{6.16}$$

which suggests that S_σ can be used to rank the importance of the random variable uncertainty (σ_i). Similarly, if we reduce μ by one σ for a random variable,

$$\Delta p \dot{=} - pS_\mu \tag{6.17}$$

Therefore, S_μ can be used to rank the importance of the uncertainty in the mean values.

The above AIS-based sensitivity analysis approach is particularly useful for system reliability problems in which there are multiple MPP's. With the AIS method, the sensitivity calculations are straightforward.

21

Chapter 7

Latin Hypercube Sampling (LHS)

7.1 Introduction

Latin Hypercube Sampling is a sampling method that strives for a more uniform coverage of the sample space than standard Monte Carlo. LHS was first proposed by Ref. 19, and several improvements have since been developed. The concept is an extension of the so-called *Latin square*. In the context of sampling, a square grid that contains sample points such that there is exactly one sample in each row and each column is known as a Latin square. A Latin hypercube is simply a generalization of this concept to arbitrary dimensions. An example of a Latin hypercube in two dimensions is given in Figure 7.1.
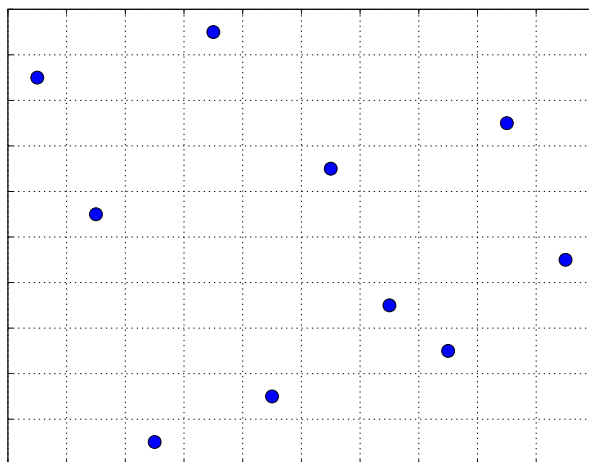
Figure 7.1: An example of a Latin hypercube sample in two dimensions of size $N = 10$

The procedure for generating a Latin hypercube begins with the generation of an $N \times K$ (where K is the number of variables) permutation matrix $\boldsymbol{P}$. Each of the K columns in $\boldsymbol{P}$ is a random permutation of $1, \ldots, N$. Then an $N \times K$ matrix $\boldsymbol{R}$ is constructed containing independent random numbers from the uniform (0,1) distribution. These matrices are used to form the basic sampling plan, represented by the matrix $\boldsymbol{S}$ as

$$S = \frac{1}{N}(\boldsymbol{P} - \boldsymbol{R}) \tag{7.1}$$

Now $\boldsymbol{S}$ is a Latin hypercube sample with marginal distributions that are uniform on (0,1). The elements of $\boldsymbol{S}$ can then be mapped to have the target marginal distributions by using the usual inverse CDF transformation:

$$x_{ij} = F_{x_j}^{-1}(x_{ij}) \tag{7.2}$$

7.2 Correlations

In Ref. 20, Iman and Conover show how the permutation matrix can be modified to target a particular correlation structure in the resulting sample. Their approach is commonly used both to generate correlated Latin hypercube samples and to reduce spurious correlations in samples for independent variables.

In this approach, the elements of $\boldsymbol{P}$ are first mapped to standard normals as

$$y_{ij} = \Phi^{-1}\left(\frac{p_{ij}}{N+1}\right) \tag{7.3}$$

Then the sample covariance of the resulting matrix $\boldsymbol{Y}$ is computed and its Cholesky factor $\bar{\boldsymbol{L}}$ is found such that

$$\bar{\boldsymbol{L}}\bar{\boldsymbol{L}}^T = \text{Cov}(\boldsymbol{Y}) \tag{7.4}$$

Then a new matrix $\boldsymbol{Y}^*$ is computed such that its covariance matrix is equal to the target correlation matrix:

$$\boldsymbol{Y}^* = \boldsymbol{Y}\left(\bar{\boldsymbol{L}}^{-1}\right)^T \boldsymbol{L}^T \tag{7.5}$$

where $\boldsymbol{L}$ is the Cholesky factor of the target correlation matrix.

The final step is to impart the correlations present in $\boldsymbol{Y}^*$ to the permutation matrix $\boldsymbol{P}$. Since each column of $\boldsymbol{P}$ is a permutation of $1,\ldots,N$, the natural approach is to use the ranks of the elements of the columns of $\boldsymbol{Y}^*$ to form a new permutation matrix $\boldsymbol{P}^*$. Thus, the matrix $\boldsymbol{P}^*$ has the same rank correlation as the matrix $\boldsymbol{Y}^*$.

When the method is being used to reduce spurious correlation, the target correlation matrix is the identity matrix, in which case it is easy to see that Eq. (7.5) becomes

$$\boldsymbol{Y}^* = \boldsymbol{Y}\left(\bar{\boldsymbol{L}}^{-1}\right)^T \tag{7.6}$$

Other approaches for reducing spurious correlations in Latin hypercube samples have been proposed as well. Refs. 21, 22 present approaches based on the Gram-Schmidt orthogonalization. These approaches each make use of an iterative procedure whereby sequential columns of the permutation matrix are updated based on the ranks of the residuals from a regression of that column on a neighboring column. The original approach is based on the residuals from a linear regression, and it has been demonstrated to show a considerable improvement over Iman and Conover's method for reducing correlation [21]. The approach was later extended to use quadratic regression, in which case it has been shown to remove not only linear correlations but also "quadratic" correlations from the sample [22].

7.3 Extending an existing sample

One of the drawbacks of Latin hypercube sampling is that its structure makes it difficult to increase the size of an existing sample while preserving the property that the resulting sample is still a Latin hypercube sample. Ref. 23 presents an approach that allows an existing sample of size N to be augmented by N additional points, while retaining the Latin hypercube structure. This approach can be used in conjunction with either of the above methods for modifying the permuatation matrix to achieve desired correlations.

The basic idea of this approach is to generate a new random permutation matrix $\boldsymbol{P}_2$ that defines bin numbers for the new samples. The challenge, however, is that the new points must be located so that the augmented sample is also a Latin hypercube. In short, this is done by creating a new grid that contains 2^K times as many bins as the original grid. Each row of $\boldsymbol{P}_2$ defines a bin on the original grid that corresponds to 2^K bins on the new grid. However, exactly one of these 2^K bins will be "available" for a new point in the sense of forming a new Latin hypercube. The new point is simply placed randomly within the available bin. The procedure is then repeated for each of the rows of $\boldsymbol{P}_2$.

Chapter 8

Advanced Mean Value Methods (MV, AMV, AMV+)

The methods described here have been developed primarily for complicated g-functions that require time-consuming calculations. For simple g-functions, these methods may not offer time savings.

8.1 Mean Value Method

Assume that the Z-function is smooth and Taylor's series expansion of Z exists at the mean values. The Z-function can be expressed as:

$$Z\left(X\right) = Z\left(\mu\right) + \sum_{i=1}^{n}\left(\frac{\partial Z}{\partial X_i}\right)\cdot\left(X_i - \mu_i\right) + H\left(X\right)$$
$$= a_0 + \sum_{i=1}^{n} a_i X_i + H\left(X\right) \tag{8.1}$$
$$= Z_{MV}\left(X\right) + H\left(X\right)$$

where the derivatives are evaluated at the mean values; Z_{MV} is a random variable representing the sum of the first-order terms and $H(\boldsymbol{X})$ represents the higher-order terms.

There are several ways of obtaining a_i. In NESSUS, the coefficients a_i are computed by numerical differentiation method (for user-defined g-functions) or by the least-squares method (for user-defined data sets). The minimum required number of Z-function evaluations is $(n+1)$ using a numerical differentiation method.

By retaining only the first-order terms, the mean and the standard deviation of Z, assuming independent random variables, are approximately:

$$\mu_z \approx a_0 + \sum_{i=1}^{n} a_i \mu_{x_i} \tag{8.2}$$

$$\sigma_z^2 \approx \sum_{i=1}^{n} a_i^2 \sigma_{x_i}^2 \tag{8.3}$$

These two statistical moments can be computed easily after the coefficients a_i are computed. Equations 8.2 and 8.3 constitute the mean-based first-order, second-moment method. However, when the probability distributions, not just the first two moments, of X_i are fully defined, the CDF of the first-order terms, Z_{MV}, is also fully defined. Since the Z_{MV}-function is linear and explicit, its CDF can be computed effectively using many methods. Thus, the mean-value first-order (MVFO, abbreviated as MV) solution defines the CDF of Z_{MV}, not just the two moments.

For nonlinear Z-functions, the MV solution is, in general, not sufficiently accurate. For simple problems, it is possible to use higher-order expansions to improve the accuracy. For example, a mean-value second-order solution can be obtained by retaining second-order terms in the series expansion. However, for problems involving implicit Z-functions and large n, the higher-order approach becomes difficult and inefficient. The AMV method described below provides an alternative to improve the MV solution with minimum additional Z-function evaluations.

8.2 Advanced Mean Value Method

The advanced mean-value first-order (AMVFO, abbreviated as AMV) method improves the mean value method by using a simple correction procedure to compensate for the errors introduced from the truncation of a Taylor's series. In addition, the AMV-based CDF provides information on the nonlinearity of the g-function to detect potential numerical problems [24].

The AMV model is defined as:

$$Z_{AMV} = Z_{MV} + H(Z_{MV}) \tag{8.4}$$

where $H(Z_{MV})$ is defined as the difference between the values of Z_{MV} and Z calculated at the Most Probable Point Locus (MPPL) of Z_{MV}. The MPPL is defined by connecting all the MPP's for different z values [25]. The MPPL of Z_{MV} is generally a nonlinear curve in the u-space. For a complicated structural analysis, the construction of Z_{MV} may be time-consuming, but its MPPL can be computed easily.

The key to the AMV method is the reduction of the truncation error by replacing the higher-order terms $H(\boldsymbol{X})$ by a simplified function $H(Z_{MV})$ dependent on Z_{MV}. Ideally, the $H(Z_{MV})$ function should be based on the exact most-probable-point locus (MPPL) of the Z-function to optimize the truncation error [24, 25]. The AMV procedure simplifies this procedure by using the MPPL of Z_{MV}. As a result of this approximation, the truncation error is not optimum; however, because the Z-function correction points are generally "close" to the exact most probable points, the AMV solution provides a reasonably good CDF estimation for many engineering application problems.

The computational steps for a point CDF analysis are: (1) Based on Z_{MV}, compute the MPP, $\boldsymbol{x}^*$, for a selected CDF value. (2) Compute $Z(\boldsymbol{x}^*)$ to update z for the selected CDF value. Given the MV model, the required number of Z-function calculations equals the number of selected CDF values. The above steps require the construction of the Z_{MV}-function only once for all the CDF levels. Assuming that a numerical differentiation scheme is used to define the Z_{MV}-function, the required number of the Z-function evaluations is $(n + 1 + m)$, where n is the number of random variables and m is the number of CDF levels. Figure 8.1 illustrates the above AMV procedure for a selected CDF. The accuracy of the CDF solution depends on the quality of the MPPL from Z_{MV}, i.e., the solution is good if the approximate locus is close to the exact one.

8.3 Advanced Mean Value Plus (AMV+) Method

In general, the AMV solution can be improved by using an improved expansion point, which can be done typically by an optimization procedure or an iteration procedure. Based initially on Z_{MV} and by keeping track of the MPPL [10], the exact MPP for a limit state $Z = z$ can be computed to establish the AMV+ model defined as:

$$Z_{AMV+} = Z(x^*) + \sum_{i=1}^{n} \left(\frac{\partial Z}{\partial X_i}\right)\bigg|_{x^*} (X_i - x_i^*) = b_0 + \sum_{i=1}^{n} b_i X_i \tag{8.5}$$

where $\boldsymbol{x}^*$ is the converged MPP for $Z = z$.

It is important to note that the variables X_i are generally non-normal and dependent, therefore the above AMV+ model, which is linear in the X-space, is different from the first-order reliability method (FORM) model, which is linear in the u-space. An advantage of Equation 8.5 is that it provides an initial estimate of the curvature (about the MPP) in the u-space that can be used both for approximate probability of failure analysis and for starting the AIS2 analysis described in Section 8. It is particularly useful when the curvature is primarily caused by the non-normal to normal transformation.
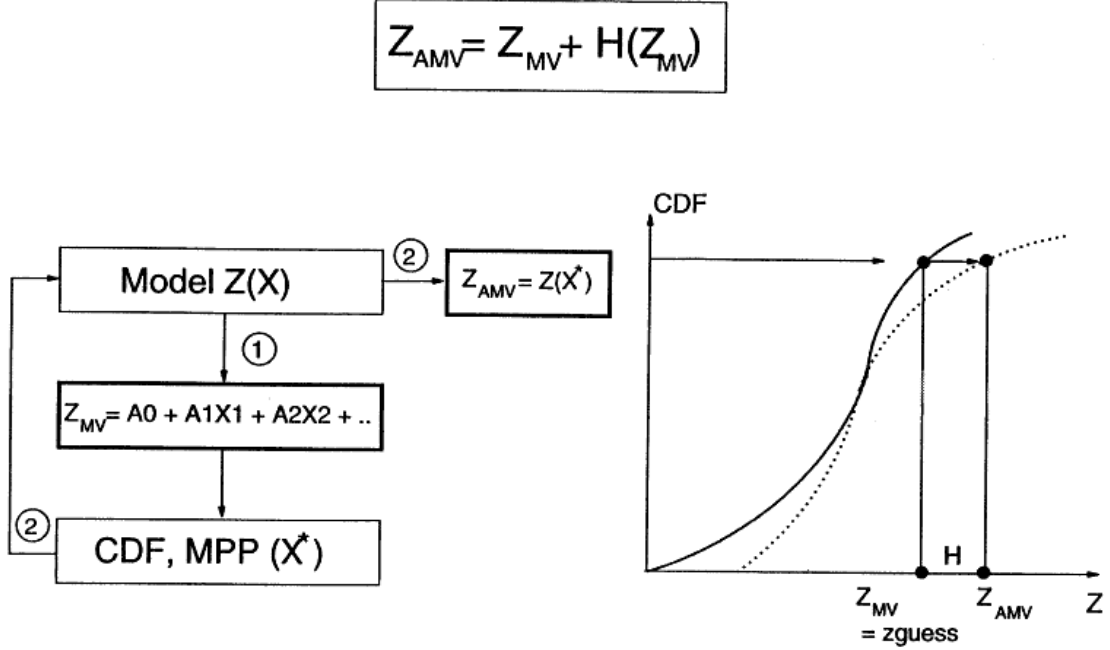
$$Z_{AMV} = Z_{MV} + H(Z_{MV})$$

Figure 8.1: Illustration of the AMV Method

Based on the AMV results, two iteration algorithms, one for specified probability level and the other for specified Z level have been proposed [10,24] to improve the CDF estimates. The two algorithms are illustrated in Figures 8.2 and 8.3. The algorithm for specified probability level is summarized in the following steps:

- Construct a linear Z-function, Z_1, initially mean-based, and search for z_0 such that $P[Z_1 < z_0] =$ probability goal.

- Use most probable point of $Z_1 = z_0$ and re-compute Z.

- Obtain the new Z1-function around the most probable point of $Z_1 = z_0$.

- Repeat the above steps until z_0 converges.

In general, the above steps require the construction of the Z1-function several times. Therefore, for complicated Z-functions which require extensive computations, efficient sensitivity computation schemes are preferred in updating the Z1-function.

8.4   Example

To illustrate the AMV and AMV+ method, consider an example where the Z-function is the first bending natural frequency of a cantilever beam. This frequency (Hz) can be approximated as

$$f_n = 0.5602\sqrt{\frac{Et^2}{12\rho L^4}} \tag{8.6}$$

where E is the modulus of elasticity, t is the thickness, ρ is the material density, and L is the length. The parameters E, t, ρ, and L are lognormal random variables. Their statistical parameters are defined in Figure 8.4.

By numerical differentiation at the mean values of the random variables, the following first-order approximation can be obtained using the results of five Z-function (i.e., Equation 8.6) evaluations.
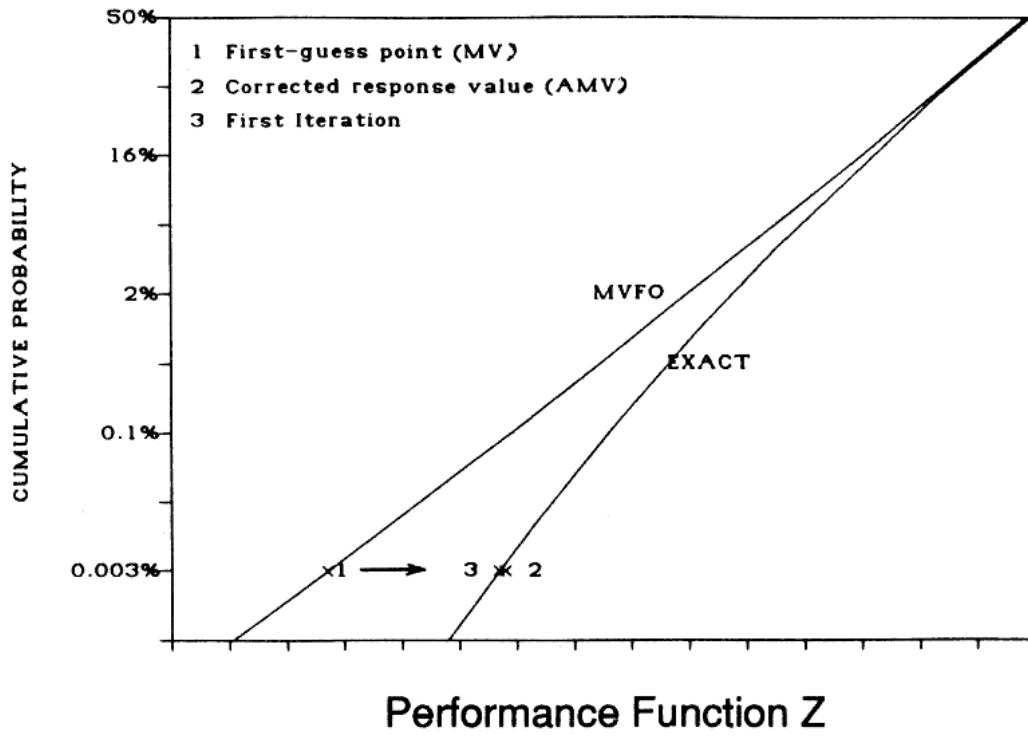
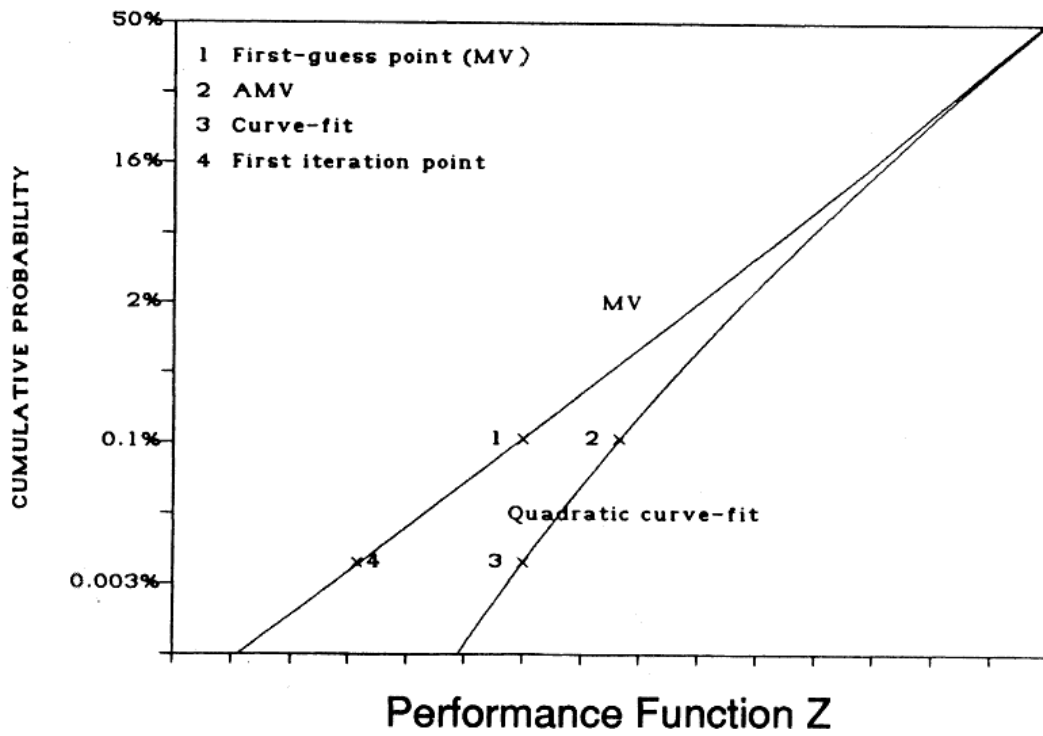Figure 8.2: AMV+ Iteration Algorithm for a Specified P-level

Figure 8.3: AMV+ Iteration Algorithm for a Specified Z-level

27

$$f_{MV} = a_0 + a_1E + a_2t + a_3\rho + a_4L \tag{8.7}$$

Based on Equation 8.7 an MV CDF solution can obtained as shown in Figure 8.4. Note that the CDF solution is plotted on a normal probability paper.

The above MV solution is exact if f_n is a linear function of the four random variables. However, since Equation 8.6 is a non-linear function, Equation 8.7 is subjected to error in the regions away from the mean values. For each selected probability, the AMV solution is obtained by calculating f_n (Equation 8.6) at the most probable point (E^*, t^*, ρ^*, L^*) obtained using Equation 8.7. This requires one function evaluation of f_n for each selected probability.

In Figure 8.4, three points are shown at a selected probability level (CDF = 0.003%). Point 1 is the MV solution, point 2 is the AMV solution, and point 3 is the AMV+ solution.
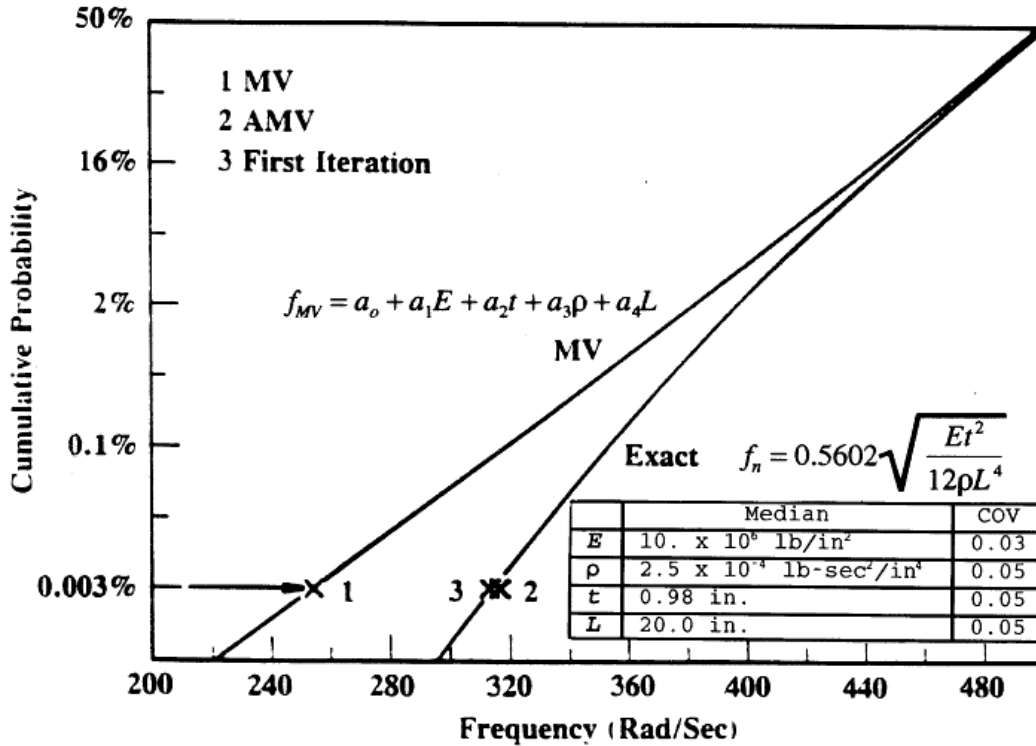
Figure 8.4: The AMV+ Analysis for the Cantilever Beam Example [24]

8.5 AMV-based Methods (AMV and AMV+) for Non-monotonic g-functions

The previous AMV-based methods have been applied to numerous problems to validate the procedure and code [24, 26, 27]. In those tested examples, the AMV-based method provided satisfactory solutions. When the Z-function is a strongly non-monotonic function of $\boldsymbol{X}$, the AMV-based CDF solution may suggest that the Z-function is a strongly non-monotonic function of Z_{MV}. Based on this information, modified CDF solutions has been developed [24].

Chapter 9

Surrogate-Based Reliability Analysis Methods

9.1 Efficient Global Reliability Analysis (EGRA)

Efficient Global Reliability Analysis was introduced in Ref. 28. EGRA locates multiple points on the limit state throughout the random variable space and uses these points to construct a Gaussian process model that provides a global approximation for the entire limit state. By focusing the samples in the only region of the space where accuracy is important (near the limit state) an effective model is created with a minimal number of samples. The resulting model is then used as an inexpensive surrogate to provide function evaluations in a sampling method to perform the probability integration, which leads to highly accurate probability estimates. The method's name is derived from its connection with Efficient Global Optimization (EGO), which was originally proposed by Jones et al. [29] for application to unconstrained minimization problems.

The feature that sets EGRA apart from other surrogate-based reliability methods is its iterative process based on the expected feasibility function (EFF). The EFF is used to select the location at which a new training point should be added to the Gaussian process model by maximizing the expectation that that the point lies on the limit state contour. A point could be expected to lie on the limit state contour if its predicted value is near the limit state value, or if the uncertainty in its prediction is such that there is a significant probability of its true value being near the limit state value. Because the uncertainty is higher in regions of the space with few observations, this provides a balance between exploiting areas that predict good solutions, and exploring areas where more information is needed. The general procedure of EGRA is:

1. Build an initial Gaussian process model of the response function.

2. Find the point that maximizes the EFF. If the EFF value at this point is sufficiently small, stop.

3. Evaluate the response function at the point where the EFF is maximized. Update the Gaussian process model using this new point. Go to Step 2.

9.1.1 Gaussian Process Models

Gaussian process (GP) models are different from other surrogate models because they provide not just a predicted value at an unsampled point, but a full Gaussian distribution with an expected value and a predicted variance. This variance gives an indication of the uncertainty in the model, which results from the construction of the covariance function. This function is based on the idea that when input points are near one another, the correlation between their corresponding outputs will be high. As a result, the uncertainty associated with the model's predictions will be small for input points that are near the points used to train the model, and will increase as one moves further from the training points.

It is assumed that the true response function being modeled, $g(\mathbf{x})$, can be described by: [30]

$$g(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \boldsymbol{\beta} + Z(\mathbf{x}) \tag{9.1}$$

where $\mathbf{h}()$ is the trend of the model, $\boldsymbol{\beta}$ is the vector of trend coefficients, and $Z()$ is a stationary Gaussian process with zero mean (and covariance defined below) that describes the departure of the model from its underlying trend. The trend of the model can be assumed to be any function, but taking it to be a constant value has been reported to be generally sufficient [31]. NESSUS employs a constant trend function and $\boldsymbol{\beta}$ is determined through a generalized least squares estimate (see Eq. 9.8). The covariance between outputs of the Gaussian process $Z()$ at points $\mathbf{a}$ and $\mathbf{b}$ is defined as:

$$Cov\left[Z(\mathbf{a}), Z(\mathbf{b})\right] = \sigma_Z^2 R(\mathbf{a}, \mathbf{b}) \tag{9.2}$$

where σ_Z^2 is the process variance and $R()$ is the correlation function. There are several options for the correlation function, but the squared-exponential function is common, and is used here for $R()$:

$$R(\mathbf{a}, \mathbf{b}) = \exp\left[-\sum_{i=1}^{d} \theta_i (a_i - b_i)^2\right] \tag{9.3}$$

where d represents the dimensionality of the problem, and θ_i is a scale parameter that governs the degree of correlation between the points in terms of dimension i. A large θ_i is representative of a short correlation length.

The expected value $\mu_g()$ and variance $\sigma_g^2()$ of the GP model prediction at point $\mathbf{x}$ are:

$$\mu_g(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \boldsymbol{\beta} + \mathbf{r}(\mathbf{x})^T \mathbf{R}^{-1}(\mathbf{g} - \mathbf{F}\boldsymbol{\beta}) \tag{9.4}$$

$$\sigma_g^2(\mathbf{x}) = \sigma_Z^2 - \begin{bmatrix} \mathbf{h}(\mathbf{x})^T & \mathbf{r}(\mathbf{x})^T \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{F}^T \\ \mathbf{F} & \mathbf{R} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{h}(\mathbf{x}) \\ \mathbf{r}(\mathbf{x}) \end{bmatrix} \tag{9.5}$$

where $\mathbf{r}(\mathbf{x})$ is a vector containing the correlations between $\mathbf{x}$ and each of the n training points (defined by Eq. 9.3), $\mathbf{R}$ is an $n \times n$ matrix containing the correlation between each pair of training points, $\mathbf{g}$ is the vector of response outputs at each of the training points, and $\mathbf{F}$ is an $n \times q$ matrix with rows $\mathbf{h}(\mathbf{x}_i)^T$ (the trend basis function at training point i containing q terms; for a constant trend $q = 1$). This form of the variance accounts for the uncertainty in the trend coefficients $\boldsymbol{\beta}$, but assumes that the parameters governing the covariance function (σ_Z^2 and $\boldsymbol{\theta}$) have known values.

The parameters σ_Z^2 and $\boldsymbol{\theta}$ are determined through maximum likelihood estimation. This involves taking the log of the probability of observing the response values $\mathbf{g}$ given the covariance parameters, which can be written as: [31]

$$\log\left[p(\mathbf{g}|\mathbf{R})\right] = -\frac{1}{n}\log|\mathbf{R}| - \log(\hat{\sigma}_Z^2) \tag{9.6}$$

where $|\mathbf{R}|$ indicates the determinant of $\mathbf{R}$, and $\hat{\sigma}_Z^2$ is the optimal value of the variance given an estimate of $\boldsymbol{\theta}$ and is defined by:

$$\hat{\sigma}_Z^2 = \frac{1}{n}(\mathbf{g} - \mathbf{F}\hat{\boldsymbol{\beta}})^T \mathbf{R}^{-1}(\mathbf{g} - \mathbf{F}\hat{\boldsymbol{\beta}}) \tag{9.7}$$

where $\hat{\boldsymbol{\beta}}$ is the generalized least squares estimate of $\boldsymbol{\beta}$ from:

$$\hat{\boldsymbol{\beta}} = \left[\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F}\right]^{-1} \mathbf{F}^T \mathbf{R}^{-1} \mathbf{g} \tag{9.8}$$

Maximizing Eq. 9.6 gives the maximum likelihood estimate of $\boldsymbol{\theta}$, which in turn defines σ_Z^2.

9.1.2 Expected Feasibility Function

The expected feasibility function is used to select the location at which a new training point should be added to the Gaussian process model. It does this by calculating the expectation that the true value of the response is expected to satisfy the equality constraint $g(\mathbf{x}) = \bar{z}$ (i.e., that it lies on the limit state) based on the expected values and variances predicted by the current GP model. An important feature of the EFF is that it provides a balance between exploiting areas of the search space where the limit state has previously been found, and exploring areas of the search space where the uncertainty is high. First, recognize that at any point in the design space, the GP prediction $\hat{g}()$ is a Gaussian distribution:

$$\hat{g}(\mathbf{x}) \sim N\left[\mu_g(\mathbf{x}), \sigma_g(\mathbf{x})\right] \tag{9.9}$$

where the mean $\mu_g()$ and the variance $\sigma_g^2()$ were defined in Eqs. 9.4 and 9.5, respectively.

Inspired by the contour estimation work in Ref. 32, the expected feasibility function is defined as:

$$EF(\mathbf{x}) \equiv E\left[\epsilon(\mathbf{x}) - \min\left(|\hat{g}(\mathbf{x}) - \bar{z}|, \epsilon(\mathbf{x})\right)\right] \tag{9.10}$$

where the ϵ term is used to focus the search in the immediate vicinity of the response threshold. This expectation can be calculated by integrating over $\bar{z} \pm \epsilon$:

$$EF(\mathbf{x}) = \int_{z-\epsilon}^{z+\epsilon} \left[\epsilon - |\bar{z} - g|\right] f_{\hat{g}} \, dg \tag{9.11}$$

where g denotes a realization of the distribution $f_{\hat{g}}$. Allowing z^+ and z^- to denote $\bar{z} \pm \epsilon$, respectively, this integral can be expressed analytically as:

$$
\begin{aligned}
EF(\mathbf{x}) = (\mu_g - \bar{z}) &\left[2\,\Phi\left(\frac{\bar{z} - \mu_g}{\sigma_g}\right) - \Phi\left(\frac{z^- - \mu_g}{\sigma_g}\right) - \Phi\left(\frac{z^+ - \mu_g}{\sigma_g}\right) \right] \\
- \sigma_g &\left[2\,\phi\left(\frac{\bar{z} - \mu_g}{\sigma_g}\right) - \phi\left(\frac{z^- - \mu_g}{\sigma_g}\right) - \phi\left(\frac{z^+ - \mu_g}{\sigma_g}\right) \right] \\
+ \ \epsilon &\left[\Phi\left(\frac{z^+ - \mu_g}{\sigma_g}\right) - \Phi\left(\frac{z^- - \mu_g}{\sigma_g}\right) \right]
\end{aligned}
\tag{9.12}
$$

where ϵ is proportional to the standard deviation of the GP predictor ($\epsilon \propto \sigma_g$). In this case, z^-, z^+, μ_g, σ_g, and ϵ are all functions of the location $\mathbf{x}$, while $\bar{z}$ is a constant.

Note that the EFF provides a balance between exploration and exploitation. Points where the expected value is close to the threshold ($\mu_g \approx \bar{z}$) and points with a large uncertainty in the prediction will have large expected feasibility values.

9.1.3 Efficient Global Reliability Analysis Algorithm

The following process makes up the EGRA algorithm:

1. Generate a small number of samples from the true response function.

   (a) Only $\frac{(n+1)(n+2)}{2}$ samples are used (where n is the number of random variables). This initial selection is arbitrary, but the number of samples required to define a quadratic polynomial is used as a convenient rule of thumb.

   (b) The samples uniformly span the random variable space over the bounds $x_i = F_i^{-1}[\Phi(\pm 5)]$, though the bounds of this search space can be adjusted if needed.

   (c) Latin hypercube sampling is used to generate the samples.

2. Construct an initial Gaussian process model from these samples.

3. Find the point with maximum expected feasibility.

   (a) The expected feasibility function is built with $\epsilon = 2\sigma_g$.

   (b) To ensure the global optimum of this multimodal function is found, the DIRECT [33] global optimizer is used.

   (c) If the maximum expected feasibility is less than 1E-5, the model has converged. Go to step 6.

4. Evaluate the true response function at this point.

5. Add this new sample to the previous set and build a new GP model. Go to step 3.

6. The resulting GP model is now used to calculate the probability of failure using any sampling method.

9.2 Gaussian Process Response Surface Method (RSM_GP)

The Gaussian Process Response Surface Method is similar to the EGRA method in that it makes use of a Gaussian Process response surface model (see Section 9.1.1) to approximate the limit state function, enabling an efficient estimation of the failure probability. However, while EGRA uses an optimization algorithm to select the locations of the training points, RSM_GP selects the training points randomly.

Recall that surrogate-based methods aim to approximate the functional relationship between the random variables and the performance model response. The first step of the RSM_GP method is to randomly generate the locations of the response surface training points. These locations are generated as a uniform Latin Hypercube design (see Chapter 7) over user-specified bounds for the random variables.

The limit state function is then evaluated at the training point locations, and the resulting data are used to construct a Gaussian process surrogate model (Section 9.1.1). The surrogate model is then sampled using basic Monte Carlo sampling (Chapter 4).

One of the advantages of this method is that it provides the user complete control over the number of times the performance function will be evaluated (this is the number of training points). Using an insufficient number of training points, however, may result in an inaccurate response surface model. In fact, the similar EGRA method (Section 9.1) may be more suitable for cases in which an appropriate number of training points is not known beforehand

Chapter 10

System Reliability Analysis (Fault Tree Method)

A system reliability problem is one that involves multiple limit states due to multiple components and/or multiple failure modes. In NESSUS, a system failure is defined using a fault tree, which provides a systematic way to manage multiple failure modes. A fault tree has three major characteristics; bottom events, combination gates, and the connectivity between the bottom events and gates. NESSUS is presently limited to AND and OR gates. The limit state functions are defined in the bottom events. Sequential failures can be modeled using the PRIORITY AND gate. A sequence of g-functions, corresponding to a sequence of updated structural configurations with load redistribution can be explicitly or implicitly defined in the bottom events as illustrated in Figure 10.1.

Through a fault tree, all the failure modes can be defined. A failure mode can involve one or more limit states. By adding all the failure modes, and therefore all the limit states, the system limit state surface can be constructed piece by piece.

There are several methods for system reliability calculations. A common approach is based on the bounding theory [3,4,34], which provides reliability bounds for serial, parallel, and combined systems. These methods provide approximate solutions.

In NESSUS, system reliability is computed using the adaptive importance sampling method. See Section 8 for details.
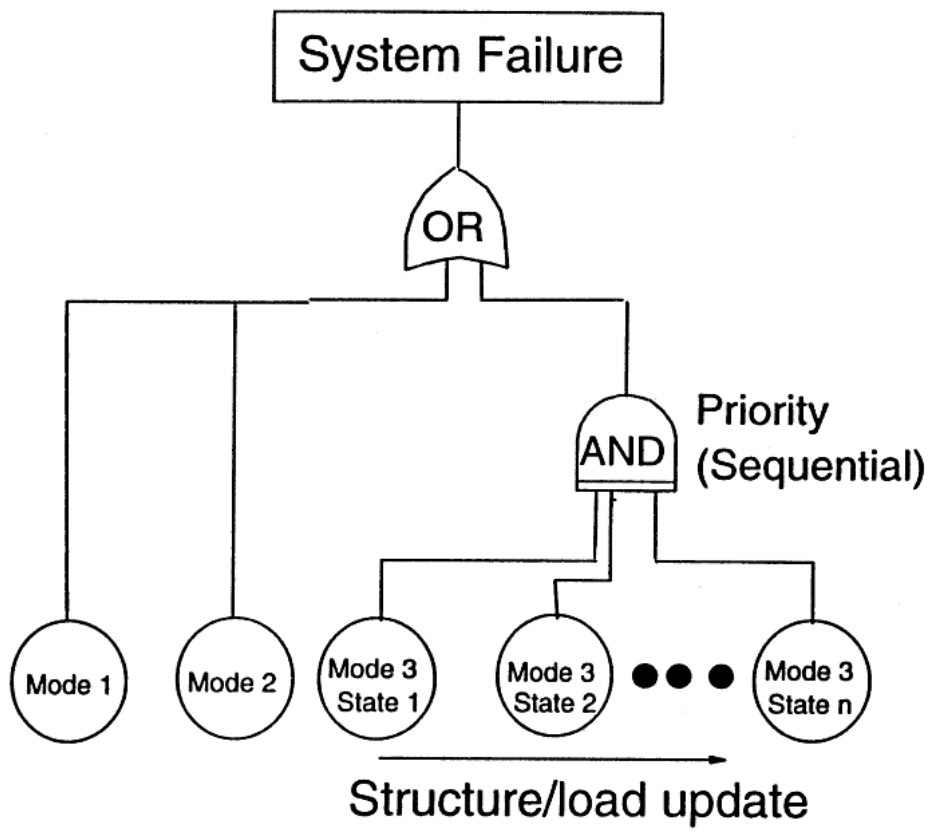
Figure 10.1: Probabilistic Fault Tree Approach for System Reliability Analysis

Chapter 11

Confidence Interval Estimation

11.1 Introduction

The basic goal of the confidence interval estimation is to quantify the confidence on the uncertainty of the CDF of Z. There are two major sources of uncertainties: Z-function modeling error and statistical uncertainty. The Z-function modeling error can be reduced by using a calibration procedure (e.g., using the experimental data). Another strategy is to introduce an additional random variable in the Z-function to characterize the uncertainty in the model [3].

The NESSUS code has an option to address the second source of uncertainty – statistical uncertainty. The approach implemented in NESSUS treats the distribution parameters of the input random variables as random variables, and then evaluates the conditional CDF. This strategy is the essence of the Bayesian approach to parameter estimation.

The approach was developed for NESSUS probabilistic finite element analysis and has been implemented only for linear (GFUNC = 1) and quadratic (GFUNC = 2) Z-functions defined by data sets.

11.2 General Approach

In general, for each random variable, the Bayesian approach can be used to derive the posterior distribution of the parameter, θ, given the data and the prior distribution. The CDF of X can be calculated as

$$P\left[X < x\right] = F_X\left(x \mid \text{data}\right) = \int_{-\infty}^{x} \int_{-\infty}^{\infty} f_X\left(x \mid \theta\right) f^+\left(\theta \mid \text{data}\right) d\theta \, dx \tag{11.1}$$

where θ is a set of parameters and f^+ is the joint posterior distribution of θ. The resulting CDF can then be used for probability analysis. In general, such a distribution would not be in the distribution library.

The above equation can be extended for the CDF of Z. The equation is

$$f_Z\left(Z_0 \mid \text{data}\right) = \int_{Z < Z_0} \int_{-\infty}^{\infty} f_X\left(x \mid \theta\right) f^+\left(\theta \mid \text{data}\right) d\theta \, dx \tag{11.2}$$

where the under bar denotes a vector. This expression can be simplified as

$$F_Z\left(Z_0 \mid \text{data}\right) = E\left[F_Z\left(Z_0 \mid \theta\right)\right] \tag{11.3}$$

which means that the CDF of Z is the expected value of the conditional CDF (conditioned on the statistical parameters). Assuming that the joint posterior distribution can be obtained, the equation can be evaluated, at least, by the Monte Carlo method. However, in general, the joint posterior distribution is difficult to evaluate.

11.3 Definition of Confidence Interval

The analysis, based on Equation 11.2, would result in a CDF. A "confidence interval," as defined herein, is based on the percentiles of the conditional CDF. A wider confidence (probability that the conditional CDF is within the interval) interval indicates more uncertainty in the input statistical parameters. Thus, the confidence interval is a measure of the uncertainty in the input statistical assumptions.

11.4 Simplified Approach

For practical purposes, a simplified approach is used in NESSUS. Assume that X is a normally distributed random variable. Given a sample, N, the sample mean, $\bar{X}$, is a normal variable with mean and standard deviation of

$$\mu_{\bar{X}} = \mu \tag{11.4}$$

$$\sigma_{\bar{X}} = \frac{\sigma}{\sqrt{N}} \tag{11.5}$$

Let S be the sample standard deviation. The variable $[(n-1)S^2]/\sigma^2$ has a chi-square distribution with $(N-1)$ degree of freedom. If this chi-square distribution is approximated by a lognormal, then the distribution of S will be a lognormal. The statistics of S can be approximated as [35]

$$\mu_S = \sigma \tag{11.6}$$

$$\sigma_S = \frac{\sigma}{\sqrt{2\,(N-1)}} \tag{11.7}$$

The simplified approach assumes that the distribution of each X_i is characterized by its mean and standard deviation which are assumed to be independent random variables. Furthermore, it is assumed that $\bar{X}$ is normal and S is lognormal.

The required input data for the confidence interval estimation are the statistics (mean and COV) for both $\mu_{\bar{X}\bar{i}}$ and S_i $(i = 1, \ldots, n)$. These input statistics may be estimated by using Equations 11.4–11.7 where the actual (unknown) mean and standard deviation may be replaced by sample mean and the sample standard deviation. The resulting coefficient of variation are:

$$C_{\text{mean}} = \frac{C_{\text{sample}}}{\sqrt{N}} \tag{11.8}$$

$$C_{\text{std.}} = \frac{1}{\sqrt{2\,(N-1)}} \tag{11.9}$$

The above statistics are based on samples. In practice, the input statistics may be based on other estimation methods including engineering judgment.

11.5 Computational Method

The adopted approach combines the Monte Carlo method and the first-order reliability method. The Monte Carlo method is used to sample the distribution parameters and the first-order reliability method is used to compute the conditional CDF. The 90% and 95% two-sided bounds are then computed.

11.6 Application to the NESSUS Analysis

For the NESSUS analysis, the g-function calculations may be time-consuming, an approach based on the advanced mean-value formulation (see Section 9) is described herein.

In this approach, the Z-function is approximated as

$$Z(X) = Z_{MV}(X) + H(Z_1) \tag{11.10}$$

where Z_{MV} is a linear approximation of $Z(\boldsymbol{X})$ based on the mean-based perturbation and $H(Z_1)$ is a correction function depending on Z_{MV}.

Given that the above equation is based on the perturbations around the mean values, denoted as $\underline{m}$, of the random parameter $\underline{\mu}$, we will assume that the equation remain reasonably accurate when the $\underline{\mu}$ realizations are away from $\underline{m}$. This assumption is reasonable if the standard deviations of $\underline{\mu}$ are small such that the Z_{MV}-function remains reasonably accurate. A rigorous approach would demand a Z_{MV} function and a $H(Z_{MV})$ evaluations for each different $\underline{\mu}$ which would be extremely time-demanding.

The proposed analysis is a two-step procedure. First, the confidence interval analysis is performed for the Z_{MV}-function through the linear Z-function (GFUNC = 1) analysis. Second, the confidence interval is adjusted by the H-function. The second step involves the shifting of the solution points. The amount of shifting for a Z-value is the difference between the Z_{MV} (Mean Value) solution and the AMV solution.

Bibliography

[1] CIRIA. Rationalization of safety and serviceability factors in structural codes. Technical Report 63, Construction Industry Research and Information Association, Storey's Gate, London, July 1977.

[2] E. Leporati. *The Assessment of Structural Safety.* Research Studies Press, 1979.

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

[4] H.O. Madsen, S. Krenk, and N.C. Lind. *Methods of Structural Safety.* Prentice Hall Inc., Englewood Cliffs, New Jersey, 1986.

[5] M. Rosenblatt. Remarks on a multivarate transformation. *The Annals of Mathematical Statistics*, 23(3):470–472, 1952.

[6] R. Rackwitz and B. Fiessler. Structural reliability under combined load sequences. *Journal of Computers and Structures*, 9:484–494, 1978.

[7] O. Ditlevsen. Principle of normal tail approximation. *Journal of Engineering Mechanics Division*, 107:1191–1208, December 1981.

[8] P.-L. Liu and A. Der Kiureghian. Optimization algorithms for structural reliability analysis. Technical Report UCB/SESM-86/09, Dept. of Civil Engineering, Univ. of California, Berkeley, July 1986.

[9] Y.-T. Wu, T.Y. Torng, and M.R. Khalessi. A new iteration procedure for efficient structural reliability analysis. In *Proceedings of ISUMA '90, the First International Symposium on Uncertainty Modeling and Analysis*, pages 72–77. IEEE Computer Society Press, December 1990.

[10] Y.-T. Wu, B. Gureghian, B. Sagar, and R.B. Codell. Sensitivity and uncertainty analysis applied to one-dimensional transport in a layered fractured rock. Part II: Probabilistic methods based on the limit-state approach. *Nuclear Technology Journal*, 104(2):297–308, November 1993.

[11] B. Fiessler, H.J. Neumann, and R. Rackwitz. Quadratic limit states in structural reliability. *Journal of Engineering Mechanics*, 105:661–676, August 1979.

[12] L. Tvedt. Distribution of quadratic forms in normal space application to structural reliability. *Journal of Engineering Mechanics*, 116(6):1183–1997, 1990.

[13] K. Breitung. Asymptotic approximation for multi-normal domain and surface integrals. In *Fourth International Conference on Application of Statistic and Probability in Soil and Structural Engineering*, University di Firenze, Pitagora Editrice, 1983.

[14] A. Harbitz. An efficient sampling method for probability of failure calculation. *Structural Safety*, 3:109–115, 1986.

[15] Y.-T. Wu and O.H. Burnside. Computation of probability of instability using the fast probability integration method. In *Proceedings of the ASCE Speciality Conference on Probabilisitc Methods*, May 1987.

[16] Y.-T. Wu, T.Y. Tong, O.H. Burnside, and M.H. Rheinfurth. Probabilistic rotordynamics analysis using an adaptive importance sampling method. In *Proceedings of the ASCE Speciality Conference on Probabilistic Mechanics and Structural and Geotechnical Reliabiliyt*, 1992.

[17] P. Bjerager. Probability integration by directional simulation. *Journal of Engineering Mechanics*, 114(8):1285–1302, 1987.

[18] Y.-T. Wu. Computational methods for efficient structural reliability and reliabiltiy sensitivity analysis. *AIAA Journal*, 32(8), August 1994.

[19] M.D. McKay, W.J. Conover, and R.J. Beckman. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

[20] R.L. Iman and W.J. Conover. A distribution-free approach to inducing rank correlation among input variables. *Communications in Statistics Part B. Simulation and Computation*, 11(3):311–334, 1982.

[21] A. Owen. Controlling correlations in Latin hypercube samples. *Journal of the American Statistical Association*, 89:1517–1522, 1994.

[22] B. Tang. Selecting Latin hypercubes using correlation criteria. *Statistica Sinica*, 8:965–977, 1998.

[23] C.J. Sallaberry, J.C. Helton, and S.C. Hora. Extension of Latin hypercube samples with correlated variables. Technical Report SAND2006-6135, Sandia National Laboratories, 2006.

[24] Y.-T. Wu, H.R. Millwater, and T.A. Cruse. An advanced probabilistic structural analysis method for implicit performance functions. *AIAA Journal*, 28(9):1663–1669, 1990.

[25] Y.-T. Wu, O.H. Burnsied, and J. Dominguez. Efficient probabilistic fracture mechanics analysis. In *Proceedings of the 4th International Conference on Numerical Methods in Fracture Mechanics*, pages 85–100, Swansea, U. K., 1987. Pineridge Press.

[26] Y.-T. Wu, O.H. Burnside, and T.A. Cruse. Probabilistic methods for structural response analysis. In W.K. Liu and T. Belytschko, editors, *Computational Mechanics of Probabilistic and Reliability Analysis*, chapter 7. Elmepress International, 1989.

[27] Y.-T. Wu and O.H. Burnside. Validation of the nessus probabilistic analysis computer program. In *Proceedings of the 29th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference*, pages 1267–1274, 1988.

[28] B.J. Bichon, M.S. Eldred, L.P. Swiler, S. Mahadevan, and J.M. McFarland. Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA Journal*, 46(10):2459–2468, October 2008.

[29] D. Jones, M. Shonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *INFORMS Journal on Computing*, 12:272–283, 1998.

[30] N.A.C. Cressie. *Statistics for Spatial Data*. Wiley: New York, revised edition, 1993.

[31] J. Sacks, S.B. Schiller, and W. Welch. Design for computer experiments. *Technometrics*, 31:41–47, 1989.

[32] P. Ranjan, D. Bingham, and G. Michailidis. Sequential experiment design for contour estimation from complex computer codes. *Technometrics*, 50(4):527–541, November 2008.

[33] J.M. Gablonsky. An implementation of the direct algorithm. Technical Report CRSC-TR98-29, Center for Research in Scientific Computation, North Carolina State University, August 1998.

[34] O. Ditlevsen. Narrow reliability bounds for structural systems. *Journal of Structural Mechanics*, 112(1):85–104, 1979.

[35] P.H. Wirsching. Error (on confidence) bounds for distribution functions resulting from statistical sampling error. Technical report, PSAM 2nd Annual Report, NASA Contract NAS3-24389, November 1986.