Introduction to Uncertainty Quantification
Module 4: Reliability Analysis

1 Introduction

Reliability analysis relates to the study of events that are usually rare, i.e. have low probability of occurrence, but cause the system being studied to fail. System failure is sometimes, but not always, catastrophic. Here, we will define failure simply as the inability of a system to meet a certain performance criterion. This performance criterion may simply involve undesirable behavior of the system, e.g. excessive vibrations causing human discomfort, or it may involve the total failure of a system that potentially causes damage and loss of life, e.g. collapse of a bridge or leakage of nuclear fuel.

Regardless of the consequences, the aim of reliability analysis is to estimate the probability of system failure given uncertainty in the model, its parameters, and its inputs. In this module, we will formalize this problem statement, provide some context for reliability in the context of risk analysis, and describe the basic Monte Carlo approach for estimating reliability.

2 Problem Statement

Consider a system defined through a so-called performance function given by $Y = g(\boldsymbol{X})$ where $\boldsymbol{X}$ is a random vector representing uncertainty in the system and its inputs having joint probability density function $f_{\boldsymbol{X}}(\boldsymbol{x})$ and joint cumulative distribution function $F_{\boldsymbol{X}}(\boldsymbol{x})$. By convention, failure of the system is defined by

$$Y = g(\boldsymbol{X}) < 0 \tag{1}$$

and we define the *failure domain* as the set $\Omega_f = \{\boldsymbol{X} | g(\boldsymbol{X}) < 0\}$. To compute the probability of failure, P_f we therefore must integrate the joint probability density $f_{\boldsymbol{X}}(\boldsymbol{x})$ over the failure domain Ω_f. That is:

$$P_f = \int_{\Omega_f} f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x} = \int_{g(\boldsymbol{X})<0} f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x}, \tag{2}$$

which is visualized in Figure 2 Let's show the failure domain and the probability of failure for a simple problem with standard normal RVs. The complementary region, referred to as the safe region Ω_s is defined by $g(\boldsymbol{X}) > 0$ and corresponds to satisfactory performance of the system. The failure and safe regions are separated by the so-called *limit state* given by $g(\boldsymbol{X}) = 0$. We now define the *reliability* of the system as the probability of satisfactory performance given by

$$R_e = \int_{\Omega_s} f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x} = \int_{g(\boldsymbol{X})>0} f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x} = 1 - P_f \tag{3}$$

Although these equations seem straightforward, their evaluation is practically very challenging because the probability of failure, P_f, is usually very small ($O(10^{-4})$ or smaller) for many engineering systems – especially those whose failure results is catastrophic consequences such as nuclear fuel leak or reactor meltdown where we clearly desire probabilities of failure that are near zero. This module and the one that follows are dedicated to practical methods for estimating the probability of failure. We begin by formulating the simple Monte Carlo estimate and explaining why it become impractical.

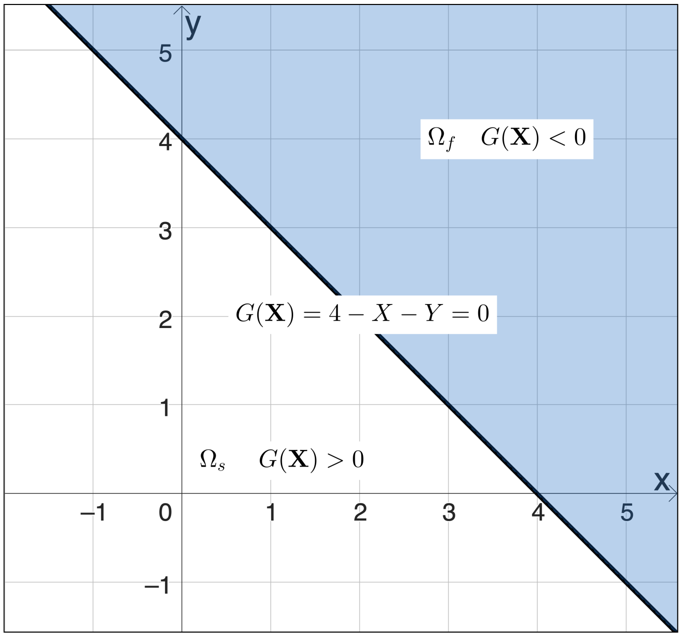

Figure 1: The failure and safety domains for the function $G(\boldsymbol{X}) = 4 - x - y$Uploaded Geogebra figure. Do you want the integral to compute P_f shown as well?

3 Reliability & Risk

Prior to getting into the details of the reliability methods, it is perhaps helpful to think of reliability in the context of *risk*. Risk is a composite number that accounts for not only the the probability of an undesirable event (i.e. failure), but also the consequences of that event. The consequences are very often measured in monetary units (e.g. dollars) although they may also be measured in other units of undesirable outcomes such as casualties resulting from an event. We therefore define the risk, R, as the product of the probability of the event and the consequences, C. If the event is termed as a failure event, then

$$R = P_f C \tag{4}$$

If we consider multiple events over which failure may occur, the the risk from n events is given by the sum of the risks of each individual event as

$$R = \sum_{i=1}^{n} P_{fi} C_i \tag{5}$$

where P_{fi} is the failure probability for event i and C_i are the consequences of event i.

As defined, risk is an objective measure that provides a single number encapsulating both the probability of failure and the impact of that failure. In decision making processes, this becomes subjective because it is necessary to define a *risk tolerance*. Risk tolerance is the amount of risk that the person or group making the decision is willing to assume. In decision making scenarios, different groups will have different risk tolerance. A risk averse strategy will set the risk tolerance very low, whereas a risk taking strategy will allow the risk to be larger.

It is common to measure risk using a plot showing risk contours as illustrated in Figure 3. Connor - Let's discuss how we can generate such a plot.

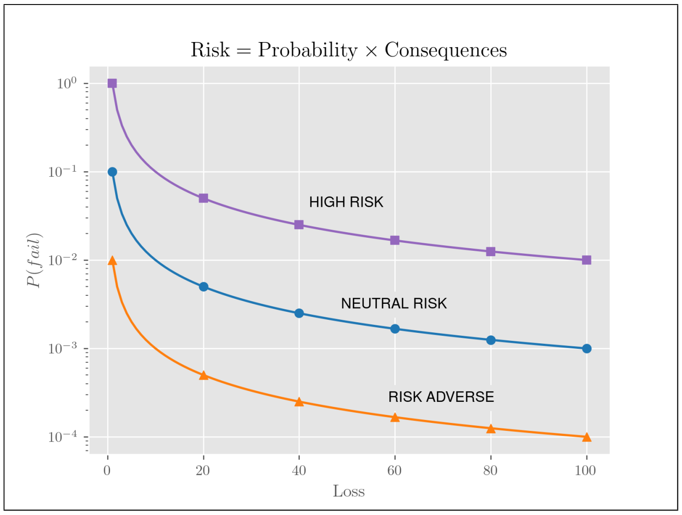

Figure 2: Uploaded Python figure.

For the remainder of this module, we will consider only the estimation of failure probabilities and will not discuss the consequences further.

4 Monte Carlo Estimates of Reliability

The failure probability integral in Eq. (2) can be re-posed as an expectation of the indicator function $I_f = I[g(\boldsymbol{x}) < 0]$ as follows

$$P_f = \mathbb{E}[I_f] = \int_{-\infty}^{\infty} I[g(\boldsymbol{x}) < 0] f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x} \tag{6}$$

where $I_f = 1$ if $g(\boldsymbol{x}) < 0$ and $I_f = 0$ if $g(\boldsymbol{x}) > 0$. In this way, the expectation can be estimated using the Monte Carlo method by

$$P_f \approx P_f^N = \frac{1}{N} \sum_{i=1}^{N} I[g(\boldsymbol{x}_i) < 0] \tag{7}$$

The estimator above is unbiased and consistent. That is, $\mathbb{E}[P_f^N] = P_f$ regardless of N and by the strong law of large numbers the estimator converges to the true probability of failure $P_f^N \to P_f$ as $N \to \infty$.

By the Central Limit Theorem, the estimate of the probability of failure follows a normal distribution with mean P_f and variance $P_f(1 - P_f)/N$. The variance arises because the random variable defined through the identity function $I[g(\boldsymbol{x}_i) < 0]$ follows a Berоulli distribution with success probability equal to P_f. Consequently, the convergence of the probability of failure estimate scales with both P_f and $1/\sqrt{N}$. This is illustrated in Figure 4, which shows the coefficient of variation of P_f for different levels of P_f as the sample size grows up to 10^6 samples. [Connor - Can we show a plot with CoV(Pf) on the y-axis and N on the x-axis (log scale) for different levels of Pf ($10^{-1}, 10^{-2}, 10^{-3}$, up to 10^{-6})]. Due to this slow convergence, numerous methods have been devised to estimate P_f. These include the use of Taylor series approximations, the construction of surrogate models, and the use of advanced Monte Carlo methods with variance reduction.

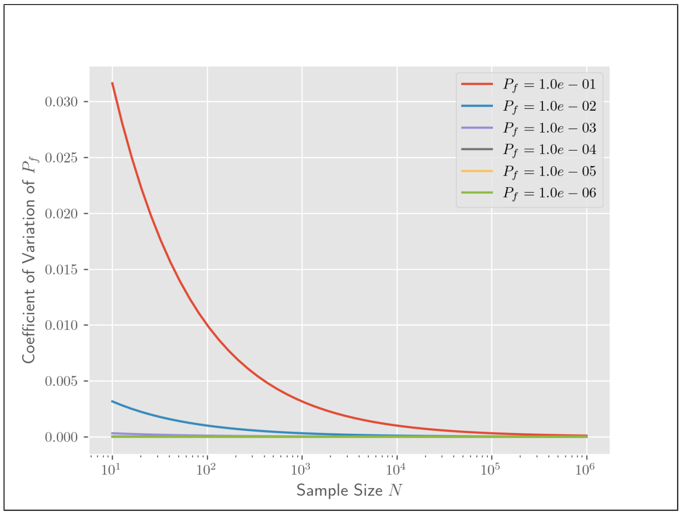

Figure 3: Uploaded Python figure. The lines are hard to distinguish here, and removing the top line for $10^{-}1$ doesn't make it any more clear. Not sure how to make this more clear?

5 Taylor Series Expansion Methods

The first method we will discuss employs a Taylor series approximation of the limit state to estimate P_f. Prior to applying these Taylor series approximation based methods, we must make a suitable isoprobabilistic transformation from the arbitrarily distributed random vector $\boldsymbol{X}$ to a vector $\boldsymbol{Z}$ of uncorrelated standard normal random variables. This can be performed using either the Nataf transformation or the Rosenblatt transformation as discussed in Module 2.3.

To recap, consider that $\boldsymbol{X} = [X_1, X_2, \ldots, X_n]^T$ is a random vector whose components are independent having joint distribution $f_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{i=1}^{n} f_{X_i}(x_i)$. We transform the random vector to a standard normal using the Nataf transformation by

$$Z_i = \Phi^{-1}[F_{X_i}(x_i)] \tag{8}$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function. After the (nonlinear) transformation, we express the performance function by $Y = G(\boldsymbol{Z})$ where again we have failure if $G(\boldsymbol{Z}) < 0$ and

$$P_f = P(G(\boldsymbol{Z} < 0)) = \int_{G(\boldsymbol{Z}<0)} \phi(z)dx \tag{9}$$

where $\phi(\cdot)$ is the standard normal probability density function. We note that the expressions in Eq. (39)I don't think you meant to cite equation 39, but the labels are reused and causing some reference inaccuracies and (9) are equivalent as long as the random variables X_i are independent. They can also be equivalent under more general isoprobabilistic transformations, but we will confine oursleves at this point to cases where the variables are independent.

5.1 First Order Reliability Method (FORM)

Consider that we express the function $G(\boldsymbol{Z})$ using the *first-order Taylor series expansion* as

$$G(\boldsymbol{Z}) \approx G(\boldsymbol{z}^*) + \nabla G(\boldsymbol{z}^*)[\boldsymbol{Z} - \boldsymbol{z}^*]^T \tag{10}$$

where $\boldsymbol{z}*$ is the point around which we linearize and $\nabla G(\boldsymbol{z}^*)$ is the gradient vector composes of partial derivatives $\partial G/\partial z_i$ evaluated at $\boldsymbol{z}^*$. Notice this is the operation we performed in Module 3.2, with different notation and a more general expansion point $\boldsymbol{z}^*$ (in Module 3.2 we use the mean vector as the expansion point). In the *First Order Reliability Method* (FORM), we seek to identify the point $\boldsymbol{z}^*$ that maximizes the probability along the surface $G(\boldsymbol{Z}) = 0$. We refer to this point as the *Most Probable Point* (MPP) and define it formally as

$$\begin{aligned} \boldsymbol{z}^* &= \arg\max_{\boldsymbol{z}} \boldsymbol{\phi}(\boldsymbol{z}) \\ \text{s.t. } & G(\boldsymbol{Z}) = 0 \end{aligned} \tag{11}$$

where $\boldsymbol{\phi}(\boldsymbol{z})$ is the joint standard normal probability density function given by

$$\boldsymbol{\phi}(\boldsymbol{z}) = \prod_{i=1}^{n} \phi_i(z_i) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} e^{-z_i^2/2} = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\sum_{i=1}^{n} z_i^2} = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\|\boldsymbol{z}\|^2} \tag{12}$$

Therefore, the MPP can equivalently be identified by

$$\begin{aligned} \boldsymbol{z}^* &= \arg\min_{\boldsymbol{z}} \|\boldsymbol{z}\| \\ \text{s.t. } & G(\boldsymbol{Z}) = 0 \end{aligned} \tag{13}$$

since minimizing $\|\boldsymbol{z}\|$ will maximize $\boldsymbol{\phi}(\boldsymbol{z})$.

As shown in Figure 5.1 Connor - Add figure like Fig 7.5 from Du notes, the MPP is the point along the limit surface $G(\boldsymbol{Z}) = 0$ that is nearest to the origin. The corresponding distance, given by $\beta = \|\boldsymbol{z}^*\|$ is known as the *reliability index* – which we'll see provides a measure of failure probability measured in terms of standard deviations.

We recognize that $G(\boldsymbol{z}^*) = 0$ and therefore the Taylor series expansion in Eq. (14) can be expressed as

$$G(\boldsymbol{Z}) \approx L(\boldsymbol{Z}) = \sum_{i=1}^{n} \left. \frac{\partial G(\boldsymbol{Z})}{\partial Z_i} \right|_{\boldsymbol{z}^*} (Z_i - z_i^*) = a_0 + \sum_{i=1}^{n} a_i Z_i \tag{14}$$

where

$$a_0 = -\sum_{i=1}^{n} \left. \frac{\partial G(\boldsymbol{Z})}{\partial Z_i} \right|_{\boldsymbol{z}^*} z_i^* \tag{15}$$

and

$$a_i = \left. \frac{\partial G(\boldsymbol{Z})}{\partial Z_i} \right|_{\boldsymbol{z}^*} \tag{16}$$

We can see that $L(\boldsymbol{Z})$ is a linear function of standard normal random variable and is therefore itself a normal random variable having mean

$$\mu_L = a_0 \tag{17}$$

and standard deviation

$$\sigma_L = \sqrt{\sum_{i=1}^{n} a_i^2} \tag{18}$$

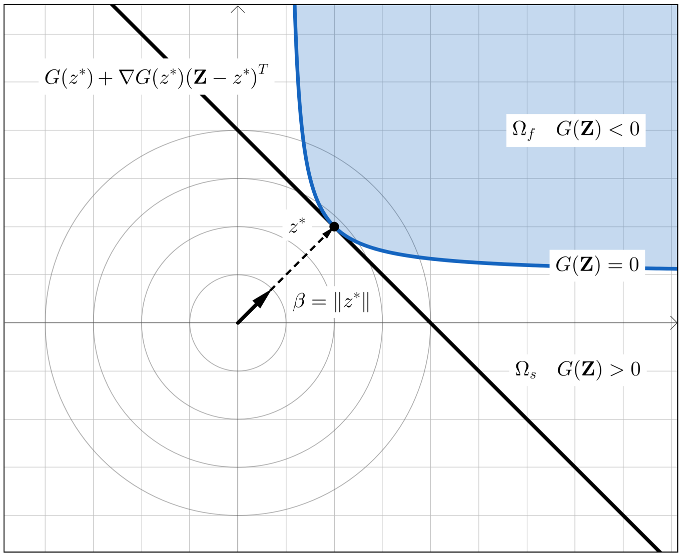
$G(z^*) + \nabla G(z^*)(\mathbf{Z} - z^*)^T$
$\Omega_f \quad G(\mathbf{Z}) < 0$
z^*
$G(\mathbf{Z}) = 0$
$\beta = \|z^*\|$
$\Omega_s \quad G(\mathbf{Z}) > 0$

Figure 4: A diagram showing the geometric interpretation of the First Order Reliability Method.Uploaded Geogebra figure

We can then estimate the probability of failure as

$$P_f = P(L(\boldsymbol{Z}) < 0) = \Phi\left(-\frac{\mu_L}{\sigma_L}\right) = \Phi\left(\frac{\sum_{i=1}^{n} \left.\frac{\partial G(\boldsymbol{Z})}{\partial Z_i}\right|_{\boldsymbol{z}^*} z_i^*}{\sqrt{\sum_{i=1}^{n}\left(\left.\frac{\partial G(\boldsymbol{Z})}{\partial Z_i}\right|_{\boldsymbol{z}^*}\right)^2}}\right) = \Phi\left(\sum_{i=1}^{n} \alpha_i z_i^*\right) \tag{19}$$

where

$$\alpha_i = \frac{\left.\frac{\partial G(\boldsymbol{Z})}{\partial Z_i}\right|_{\boldsymbol{z}^*}}{\sqrt{\sum_{i=1}^{n}\left(\left.\frac{\partial G(\boldsymbol{Z})}{\partial Z_i}\right|_{\boldsymbol{z}^*}\right)^2}}. \tag{20}$$

We can then express the vector

$$\boldsymbol{a} = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T = \frac{\nabla G(\boldsymbol{z}^*)}{\|\nabla G(\boldsymbol{z}^*)\|} \tag{21}$$

such that the probability of failure can be expressed

$$P_f \approx \Phi\left(-\sum_{i=1}^{n} \alpha_i z_i^*\right) = \Phi(-\boldsymbol{a}^T \boldsymbol{z}^*) \tag{22}$$

Next, let us provide a geometric interpretation to our estimate. The MPP is given by a vector from the origin to the nearest point on the limit surface $G(\boldsymbol{Z}) = 0$ and is a point of tangency between the limit surface and the circle (hypersphere in n dimensions) of radius β centered at the origin as illustrated in Figure 5.1 Connor - Add figure like Fig 7.7 from Du notes. Therefore, the MPP is perpendicular to the limit surface. We can then define the unit vector perpendicular to the limit surface at $\boldsymbol{z}^*$ by

$$\frac{\boldsymbol{z}^*}{\|\boldsymbol{z}^*\|} = \frac{\boldsymbol{z}^*}{\beta} \tag{23}$$

We further recognize that the unit vector $\boldsymbol{a}$, which represents the normalized gradient of the G at the MPP, is also perpendicular to the limit surface at $\boldsymbol{z}^*$, but points in the opposite direction. Therefore, we have

$$\frac{\boldsymbol{z}^*}{\beta} = -\boldsymbol{a} \tag{24}$$

, which allows us to express the MPP as

$$\boldsymbol{z}^* = -\beta\boldsymbol{a} \tag{25}$$

Plugging this new expression into our probability of failure estimate in Eq. (22) yields

$$P_f \approx \Phi(-\boldsymbol{a}\boldsymbol{a}^T\beta) = \Phi(-\beta) \tag{26}$$

since $\boldsymbol{a}\boldsymbol{a}^T = \sum_{i=1}^n \alpha_i^2 = 1$.

The First Order Reliability Method can then be simplified to the problem of identifying the MPP, also referred to as the *design point*, by solving the optimization problem in Eq. (13). Once the design is identified, we simply compute the reliability index as the distance from the origin to the MPP, i.e. $\|\boldsymbol{z}^*\|$.

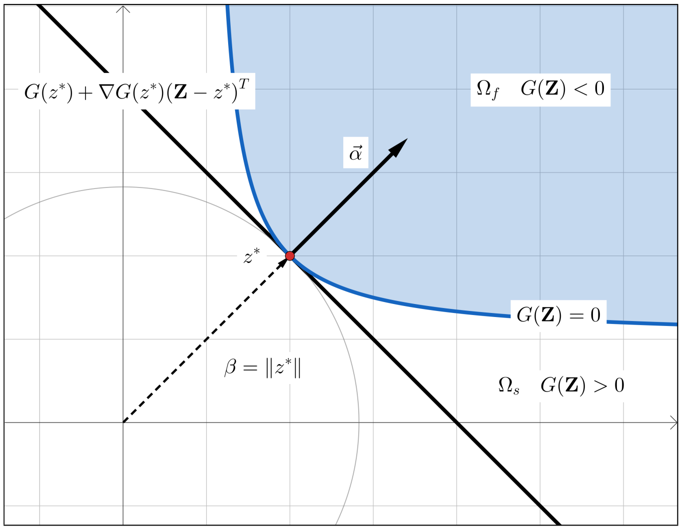
$G(z^*) + \nabla G(z^*)(\mathbf{Z} - z^*)^T$
$\Omega_f \quad G(\mathbf{Z}) < 0$
$\vec{\alpha}$
z^*
$G(\mathbf{Z}) = 0$
$\beta = \|z^*\|$
$\Omega_s \quad G(\mathbf{Z}) > 0$

Figure 5: The MPP point z^* is tangent to the distance function $\|z^*\|$ and the performance function $G(\mathbf{Z}) = 0$.Uploaded Geogebra figure

Design Point Search

The search for the design point typically must be performed numerically. Several numerical optimization routines have been developed to find the design point. Next, we outline a commonly used gradient-based algorithm often referred to as the Hasofer-Lind-Rackwitz-Fiessler (HLRF) algorithm. In this approach, we begin by defining an initial guess for the design point, denoted by $\boldsymbol{z}^{(0)}$ and at each iteration we will denote the current design point by $\boldsymbol{z}^{(k)}$. At each iteration, we then linearize the performance function using the Taylor series expansion around this point as

$$G(\boldsymbol{z}) \approx G(\boldsymbol{z}^{(k)}) + \nabla G(\boldsymbol{z}^{(k)})[\boldsymbol{z} - \boldsymbol{z}^{(k)}]^T \tag{27}$$

From this linearized function, let us assume that function at the next iteration, i.e. $G(\boldsymbol{z}^{(k+1)})$, will be equal to zero as follows

$$G(\boldsymbol{z}^{(k+1)}) = G(\boldsymbol{z}^{(k)}) + \nabla G(\boldsymbol{z}^{(k)})[\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^{(k)}]^T = 0 \tag{28}$$

Recall from Eq. (25) that

$$\boldsymbol{z}^{(k)} = \beta^{(k)}\boldsymbol{a}^{(k)}. \tag{29}$$

Thanks to the linear assumption, the unit vector $\boldsymbol{a}$ is constant and always pointing in the same direction and therefore, we have

$$\boldsymbol{z}^{(k+1)} = \beta^{(k+1)}\boldsymbol{a}^{(k)}. \tag{30}$$

Substituting Eqs. (29) and (30) into Eq. (28) yields

$$\begin{aligned} 0 &= G(\boldsymbol{z}^{(k)}) + \nabla G(\boldsymbol{z}^{(k)})[\beta^{(k+1)}\boldsymbol{a}^{(k)} - \beta^{(k)}\boldsymbol{a}^{(k)}]^T \\ &= G(\boldsymbol{z}^{(k)}) + \nabla G(\boldsymbol{z}^{(k)})(\boldsymbol{a}^{(k)})^T[\beta^{(k+1)} - \beta^{(k)}] \\ &= G(\boldsymbol{z}^{(k)}) + \|\nabla G(\boldsymbol{z}^{(k)})\|[\beta^{(k+1)} - \beta^{(k)}], \end{aligned} \tag{31}$$

which we can solve for the reliability index at the next iteration as

$$\beta^{(k+1)} = \beta^{(k)} + \frac{G(\boldsymbol{z}^{(k)})}{\|\nabla G(\boldsymbol{z}^{(k)})\|}. \tag{32}$$

We can therefore determine the next guess of the design point as

$$\begin{aligned} \boldsymbol{z}^{(k+1)} &= -\beta^{(k+1)}\boldsymbol{a}^{(k)} \\ &= -\left\{\beta^{(k)} + \frac{G(\boldsymbol{z}^{(k)})}{\|\nabla G(\boldsymbol{z}^{(k)})\|}\right\}\boldsymbol{a}^{(k)} \end{aligned} \tag{33}$$

This process is applied iteratively starting at an initial point $\boldsymbol{z}^{(0)}$, where typically $\boldsymbol{z}^{(0)} = \boldsymbol{0}$ is specified as the origin. The iterations are then repeated until a convergence criterion is satisfied. Three commonly applied convergence criteria are the following

$$\begin{aligned} &i. && \|\boldsymbol{z}^{(k+1)} - \boldsymbol{z}^{(k)}\| \leq \epsilon_1 \\ &ii. && \|\nabla G(\boldsymbol{z}^{(k+1)}) - G(\boldsymbol{z}^{(k)})\| \leq \epsilon_2 \\ &iii. && |\beta^{(k+1)} - \beta^{(k)}| \leq \epsilon_3 \end{aligned} \tag{34}$$

where $\epsilon_1, \epsilon_2, \epsilon_3$ are sufficiently small numbers.

For many problems, this simple algorithm will converge rapidly. For other problems, it may fail to converge and may require more advanced optimization algorithms.

5.2 Second Order Reliability Method (SORM)

The main difference between the First Order Reliability Method (FORM) and Second Order Reliability Method (SORM) is that, as the name implies, we apply a second-order Taylor series expansion in SORM. That is, we expand the performance function as

$$G(\boldsymbol{Z}) \approx G(\boldsymbol{z}^*) + \nabla G(\boldsymbol{z}^*)[\boldsymbol{Z} - \boldsymbol{z}^*]^T + \frac{1}{2}[\boldsymbol{Z} - \boldsymbol{z}^*]\boldsymbol{H}(\boldsymbol{z}^*)[\boldsymbol{Z} - \boldsymbol{z}^*]^T \tag{35}$$

where $\boldsymbol{H}(\boldsymbol{z}^*)$ is the Hessian matrix of second derivatives having components

$$H_{ij}(\boldsymbol{Z}) = \left.\frac{\partial^2 G(\boldsymbol{Z})}{\partial Z_i \partial Z_j}\right|_{\boldsymbol{z}^*} \tag{36}$$

In SORM, the design point is identified in the same way as in FORM and then the probability of failure is estimated by applying a correction to Eq. (26) as follows

$$P_f \approx= \Phi(-\beta)\prod_{i=1}^{n-1}(1+\beta\kappa_i)^{-\frac{1}{2}} \tag{37}$$

where κ_i are the principle curvatures of the performance function evaluated at the design point. Additional details will not be provided here because the SORM often provides only a marginal improvement over FORM is rarely considered practical because of the large computational expense necessary to compute the principal curvatures, which require Hessian matrix computations.

6 Advanced Monte Carlo Methods with Variance Reduction

FORM and SORM provide convenient probability of failure estimates when the performance function (and hence limit state) are linear, or close to linear, in the space of standard normal random variables. For many systems, however, the performance function is a nonlinear, sometimes strongly nonlinear, function of non-Gaussian random variables. In these cases, we must resort to the more general and robust Monte Carlo methods. But, as we saw earlier, Monte Carlo methods are inefficient and require far too many model evaluations to be practical for reliability problems with small failure probabilities and expensive computational models. Therefore, we must resort to variance reduction methods (Module 3.3) to reduce the number of model evaluations and improve convergence of the failure probability estimates. In this section, we will review variance methods that have been developed for specifically the purposes of reliability analysis. Some of these methods, such as importance sampling, are direct applications of the methods presented in Module 3.3. while others, such as subset simulation are new.

6.1 Importance Sampling

Recall from Module 3.3, Section 7.3 that an importance sampling estimator can be constructed by reformulating the expectation integral with respect to a new probability density, the *importance sampling density*, as

$$\begin{aligned}\theta = \mathbb{E}[g(\boldsymbol{X})] &= \int_{-\infty}^{\infty} g(\boldsymbol{x})\frac{f_{\boldsymbol{X}}(\boldsymbol{x})}{q_{\boldsymbol{X}}(\boldsymbol{x})}q_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x}\\ &= \mathbb{E}_q\left[g(\boldsymbol{x})\frac{f_{\boldsymbol{X}}(\boldsymbol{x})}{q_{\boldsymbol{X}}(\boldsymbol{x})}\right]\end{aligned} \tag{38}$$

For reliability analysis, we specifically pose this expectation as given in Eq. (39) such that the importance sampling estimator is given by

$$P_f = \mathbb{E}_q\left[I_f\frac{f_{\boldsymbol{X}}(\boldsymbol{x})}{q_{\boldsymbol{X}}(\boldsymbol{x})}\right] = \int_{-\infty}^{\infty} I[g(\boldsymbol{x})<0]\frac{f_{\boldsymbol{X}}(\boldsymbol{x})}{q_{\boldsymbol{X}}(\boldsymbol{x})}q_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x}, \tag{39}$$

which can be estimated by Monte Carlo sampling from $q_{\boldsymbol{X}}(\boldsymbol{x})$ and applying the following estimator

$$P_f \approx \frac{1}{N}\sum_{i=1}^{N} I[g(\boldsymbol{x}_i)<0]\frac{f_{\boldsymbol{X}}(\boldsymbol{x}_i)}{q_{\boldsymbol{X}}(\boldsymbol{x}_i)} \tag{40}$$

The reliability problem then becomes one of identifying the optimal importance sampling density estimator $q_{\boldsymbol{X}}(\boldsymbol{x})$ to minimize the variance in the estimator and thereby minimize the number of samples necessary for an accurate estimate of reliability. Some discussion has already been provided on identifying the optimal importance sampling density for general problems in Module 3.3, Section 7.3 where we introduce the theoretical optimal density (which is infeasible in practice) and the maximum principle. Here, we will briefly introduce four concepts that have been employed to estimate the optimal importance sampling density, as summarized by Tabandeh et al. [1] and [2]:

1. Optimal parametric estimators
2. Optimal nonparametric estimators
3. Surrogate model-based estimators
4. Sequential and adaptive importance sampling

Optimal parametric estimators

The first class of methods aims to find a parametric model, $q_{\boldsymbol{X}}(\boldsymbol{x};\boldsymbol{\theta})$, or a convex combination (mixture) of parametric models given by

$$q_{\boldsymbol{X}}(\boldsymbol{x},\boldsymbol{\Theta})=\sum_{k=1}^{K} w_k q_k(\boldsymbol{x},\boldsymbol{\theta}_k) \tag{41}$$

where $\sum_{k=1}^{K}=1$ and $\boldsymbol{\theta}_k$ are the parameters of the k^{th} probability density. The objective in identifying the optimal parametric model is to minimize a distance measure between the parametric model and the optimal importance sampling density. A common distance measure is the relative entropy as determined by the Kullback-Leibler (KL) divergence between two densities $q_1(\boldsymbol{x})$ and $q_2(\boldsymbol{x})$ given by

$$D_{KL}[q_1(\boldsymbol{x})\|q_2(\boldsymbol{x})]=\int_{\mathbb{R}} q_1(\boldsymbol{x})[\ln q_1(\boldsymbol{x})-\ln q_2(\boldsymbol{x})]d\boldsymbol{x} \tag{42}$$

If we then consider that the true optimal importance sampling density is given by $q^*_{\boldsymbol{X}}(\boldsymbol{x})$, then the optimal parametric model is determined by solving the following optimization problem

$$\begin{aligned}\boldsymbol{\Theta}^* &= \arg\min_{\boldsymbol{\Theta}} D_{KL}[q^*_{\boldsymbol{X}}(\boldsymbol{x})\|q_{\boldsymbol{X}}(\boldsymbol{x},\boldsymbol{\Theta})]\\ &= \arg\max_{\boldsymbol{\Theta}} \int_{\mathbb{R}} q^*_{\boldsymbol{X}}(\boldsymbol{x})\ln q_{\boldsymbol{X}}(\boldsymbol{x},\boldsymbol{\Theta})d\boldsymbol{x}\end{aligned} \tag{43}$$

which follows from applying the definition of KL divergence and retaining only the terms that depend on $\boldsymbol{\Theta}$. Applying the optimal importance sampling density from Module 3.3, i.e. $q^*_{\boldsymbol{X}}(\boldsymbol{x})\propto I_F(\boldsymbol{x})f_{\boldsymbol{X}}(\boldsymbol{x})$, the optimization can then be posed as follows

$$\begin{aligned}\boldsymbol{\Theta}^* &= \arg\max_{\boldsymbol{\Theta}} \int_{\mathbb{R}} I_F(\boldsymbol{x})\ln q_{\boldsymbol{X}}(\boldsymbol{x},\boldsymbol{\Theta})f_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x}\\ &= \arg\max_{\boldsymbol{\Theta}} \mathbb{E}_f\left[I_F(\boldsymbol{x})\ln q_{\boldsymbol{X}}(\boldsymbol{x},\boldsymbol{\Theta})\right]\end{aligned} \tag{44}$$

To solve this optimization problem, we draw an initial training set of M samples from $f_{\boldsymbol{X}}(\boldsymbol{x})$ to estimate the expectation in Eq. (44) using Monte Carlo simulation. Once the estimated optimal importance sampling density is identified, an additional N samples are drawn from $q_{\boldsymbol{X}}(\boldsymbol{x})$ and importance sampling is applied to estimate P_f. In total, the method requires $M+N$ simulations of the computational model.

Optimal nonparametric estimators

A closely related method seeks to estimate the optimal importance sampling density using nonparametric probability density functions, typically using *kernel density estimation.* Here, we define the probability density $q_{\boldsymbol{X}}(\boldsymbol{x})$ as

$$q_{\boldsymbol{X}}(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{1}{M\theta^d} \sum_{m=1}^{M} K_q \left(\frac{\boldsymbol{x} - \boldsymbol{x}^{(m)}}{\theta^{(m)}} \right) \tag{45}$$

where $K_q(\cdot)$ is a kernel function (kernel density) with $K_q(\boldsymbol{x}) \geq 0, \forall \boldsymbol{x}$ and $\int_{\mathbb{R}} K_q(\boldsymbol{x}) d\boldsymbol{x} = 1$. Notice that $x^{(m)}$ provides a location parameter and $\theta^{(m)}$ a scale parameter for each kernel in the summation. A commonly used kernel density is the Gaussian kernel having density

$$K_G(\boldsymbol{x}) \propto \exp\left\{ -\frac{1}{2} \boldsymbol{x} \Sigma \boldsymbol{x}^T \right\} \tag{46}$$

where Σ is the covariance matrix. Once again, the density $q_{\boldsymbol{X}}(\boldsymbol{x}, \boldsymbol{\theta})$ is determined by minimizing a discrepancy measure between it and the optimal importance sampling density $q^*_{\boldsymbol{X}}(\boldsymbol{x})$, which may be the KL divergence (as above), the L_1 error, or commonly the L_2 error such that the optimization is posed as

$$\begin{aligned} \boldsymbol{\theta}^* &= \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^+} \| q_{\boldsymbol{X}}(\boldsymbol{x}, \boldsymbol{\theta}) - q^*_{\boldsymbol{X}}(\boldsymbol{x}) \|_2 \\ &= \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^+} \int_{\mathbb{R}} \left| q_{\boldsymbol{X}}(\boldsymbol{x}, \boldsymbol{\theta}) - q^*_{\boldsymbol{X}}(\boldsymbol{x}) \right|^2 d\boldsymbol{x} \end{aligned} \tag{47}$$

where $\mathbb{R}^+$ are the positive real numbers. This particular measure assigns equal weights to small and large differences, which favors estimates in the body of the distribution where differences have larger magnitude. As a result, it provides a poor estimate in the tails, which are especially important for reliability analysis. To avoid this, a weighting function is often applied such that the optimization is solved as

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^+} \int_{\mathbb{R}} \frac{1}{q_{\boldsymbol{x}}(\boldsymbol{x})} \left| q_{\boldsymbol{X}}(\boldsymbol{x}, \boldsymbol{\theta}) - q^*_{\boldsymbol{X}}(\boldsymbol{x}) \right|^2 d\boldsymbol{x} \tag{48}$$

It has been shown that minimizing the weighted L_2 error above is equivalent to minimizing the variance in the importance sampling failure probability estimate [3].

As before, we solve the optimization problem above using an initial set of M training samples. Once the estimate of the optimal importance sampling density is established, we continue with N samples using the importance sampling density.

Surrogate model-based estimators

As discussed in Module 3.4, surrogate models can provide a cheap and convenient means of approximating the performance function. Numerous methods have been developed that seek to employ surrogate models in an effort to identify the optimal importance sampling density. Recall, once again, that the optimal sampling density is given by $q^*_{\boldsymbol{X}}(\boldsymbol{x}) \propto I_F(\boldsymbol{x}) f_{\boldsymbol{X}}(\boldsymbol{x})$ where $I_f(\boldsymbol{x})$ is the indicator function for failure $g(\boldsymbol{X}) < 0$. Given a computationally efficient surrogate model $\hat{g}(\boldsymbol{x})$, we can then approximate the importance sampling density by approximating the indicator function such that

$$I_F(\boldsymbol{x}) \approx I(\hat{g}(\boldsymbol{x}) < 0) = \begin{cases} 1 & \text{if } \hat{g}(\boldsymbol{x}) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{49}$$

Using, $I(\hat{g}(\boldsymbol{x}) < 0)$ we can directly estimate the optimal importance sampling density.

As with the previously discussed methods, the surrogate-based approach requires an initial M training points from which to fit the surrogate model. Then, an additional N samples are drawn from the importance sampling density to estimate P_f.

Sequential and adaptive importance sampling

Numerous methods have been developed to adaptively or sequentially estimate the optimal importance sampling density. In principle, these operate similarly to the approaches above except that they aim to solve the optimization problem using iterative or adaptive methods that update the optimal importance sampling density as new samples are collected, while somehow leveraging the previously drawn samples. We will not discuss these methods in detail, but two important developments along these lines are the methods proposed by Au and Beck [4] and Papaioannou et al. [2].

6.2 Subset Simulation

Another widely-used variance reduction method for reliability analysis is *subset simulation*, originally proposed by Au and Beck [5]. Consider that we are interested in estimating the probability of the failure event F, again corresponding to $G(\boldsymbol{Z}) < 0$ (equivalently $g(\boldsymbol{X}) < 0$). Subset simulation defines a nested sequence of failure events $F_1 \supset F_2 \supset \cdots \supset F_m = F$ such that $F_k = \cap_{i=1}^{k} F_i, k = 1, \ldots, m$. In our classical reliability setting, where $G(\boldsymbol{Z}) < 0$ corresponds to F we can define $F_i, i = 1, \ldots, m$ as the event $\{G(\boldsymbol{Z}) < C_i\}$ where $C_1 < C_2 < \cdots < C_m = 0$. By the definition of conditional probability, we have

$$\begin{aligned} P_f = P(F_m) &= P\left(\bigcap_{i=1}^{m} F_i\right) \\ &= P\left(F_m | \bigcap_{i=1}^{m-1} F_i\right) P\left(\bigcap_{i=1}^{m-1} F_i\right) \\ &= P\left(F_m | F_{m-1}\right) P\left(\bigcap_{i=1}^{m-1} F_i\right) \\ &= P(F_1) \prod_{i=1}^{m-1} P(F_{i+1}|F_i) \end{aligned} \tag{50}$$

In this representation, the P_f is expressed as a product of conditional probabilities that are, in principle, easier to estimate by Monte Carlo because they are larger than the original failure probability. For example, consider that $P_f = 10^{-4}$. This would require a very large number of samples to evaluate with Monte Carlo simulation. However, this failure probability can be expressed as $P_f = 10^{-4} = P(F_0)P(F_1|F_0)P(F_2|F_1)P(F_3|F_2)$ where $P(F_0) = P(F_1|F_0) = P(F_2|F_1) = P(F_3|F_2) = 0.1$. Meanwhile, estimating intermediate failure probability of 0.1 can be performed with relatively few Monte Carlo samples.

For simplicity, and without loss of generality consider that we will work with uncorrelated standard normal random variables $\boldsymbol{Z}$. The first step in subset simulation is to estimate $P(F_0)$. This is done in a straightforward manner using Monte Carlo simulation. However, traditional Monte Carlo simulation would require specifying C_1 in order to evaluate $P(G(\boldsymbol{Z}) < C_1)$. A priori, we will not know how to set C_1 such that $P(F_1)$ is of appropriate magnitude. If it's too large, i.e. $P(F_1) > 0.3$, then subset simulation will require a large number of subsets. If it's too small, i.e. $P(F_1) < 0.05$ then it will require a large number of simulations in each subset to estimate the conditional failure probabilities. To avoid this issue altogether, we generate N samples according to the standard normal distribution and choose C_1 such that it defines the upper quantile from the samples having probability p_0. That is, we define C_1 such that $p_0 \times N$ samples exceed C_1. In other words, we select C_1 such that $P(F_1) = p_0$.

The main challenge in subset simulation is to generate samples from the conditional probability distributions defined for subset i by $\phi^i(\boldsymbol{x}|G(\boldsymbol{Z}) < C_i)$. This is done using Markov chain Monte Carlo (MCMC) methods as introduced in Module 3.3. A review of MCMC methods for subset simulation can be found in Papaioannou et al. [6]. The particular MCMC implementation for subset simulation occurs in two steps:

1. Given a sample $\boldsymbol{z}_k$ satisfying $G(\boldsymbol{z}_k) < C_i$, generate a candidate sample $\tilde{\boldsymbol{z}}$ from the proposal distribution $q^*(\boldsymbol{z}|\boldsymbol{z}_k)$ according to a standard MCMC algorithm. This candidate is accepted or rejected according to the standard MCMC acceptance criterion. See Module 3.3 for details.

2. If the new sample is accepted according to the MCMC acceptance criterion, evaluate $G(\tilde{\boldsymbol{z}})$. If $G(\tilde{\boldsymbol{z}}) < C_i$, accept the sample as the new state of the Markov chain $\boldsymbol{z}_{k+1} = \tilde{\boldsymbol{z}}$. Otherwise, set $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k$.

Recall that, after completion of the first step of subset simulation, we have a set of $N_c = p_0 \times N$ samples that satisfy $G(\boldsymbol{z}_k) < C_1$. We therefore initialize the two-step algorithm above for each point, resulting in N_c Markov chains, each having stationary probability distribution $\phi^i(\boldsymbol{z}|G(\boldsymbol{Z}) < C_i)$. Each chain is run for a total of N/N_c steps, resulting in a total of N samples drawn from $\phi^i(\boldsymbol{z}|G(\boldsymbol{Z}) < C_i)$. We then define C_{i+1} as the upper quantile of the conditional distribution having probability p_0, i.e. such that $p_0 \times N$ samples drawn from $\phi^i(\boldsymbol{z}|G(\boldsymbol{Z}) < C_i)$ satisfy $G(\boldsymbol{Z}) < C_{i+1}$.

This process is repeated until the newly defined threshold $C_{i+1} < 0$. In this final subset, we estimate $P_{N_s} = P(G(\boldsymbol{Z}) < 0|G(\boldsymbol{Z} < C_i))$ and evaluate the probability of failure as the product of the conditional probabilities given by

$$P_f = (p_0)^{N_s} P_{N_s} \tag{51}$$

Proof of Markov chain stationary distribution

Consider MCMC sampling using the component-wise Metropolis-Hasting (MH) algorithm described in Section 3.3. We begin with a sample $\boldsymbol{z}_k$ in failure domain F_i. According to the component-wise MH algorithm, the proposal density for the transition probability between two states in F_i is given by

$$q(\boldsymbol{z}_{k+1}|\boldsymbol{z}_k) = \prod_{j=1}^{d} q_j(\boldsymbol{z}_{k+1}^{(j)}|\boldsymbol{z}_k^{(j)}) \tag{52}$$

where $q_i(\cdot)$ is the transition probability of component j, i.e. the transitions in each dimension are *independent*. Considering the case where $\boldsymbol{z}_{k+1}^{(j)} \neq \boldsymbol{z}_k^{(j)}$, the transition probability in component j can be expressed using the MH acceptance criterion as

$$q_j(\boldsymbol{z}_{k+1}^{(j)}|\boldsymbol{z}_k^{(j)}) = q_j^*(\boldsymbol{z}_{k+1}^{(j)}|\boldsymbol{z}_k^{(j)}) \min\left\{1, \frac{\phi_j(\boldsymbol{z}_{k+1}^{(j)})}{\phi_j(\boldsymbol{z}_k^{(j)})}\right\} \tag{53}$$

Assuming a symmetric proposal density $q^*(\boldsymbol{z}_{k+1}|\boldsymbol{z}_k) = q^*(\boldsymbol{z}_k|\boldsymbol{z}_{k+1})$ and the identity $\min\{1, a/b\}b = \min\{1, b/a\}a$ for all positive numbers a, b, we see that the Markov chain reversibility condition is satisfied:

$$q_j(\boldsymbol{z}_{k+1}^{(j)}|\boldsymbol{z}_k^{(j)})\phi_j(\boldsymbol{z}_k^{(j)}) = q_j(\boldsymbol{z}_k^{(j)}|\boldsymbol{z}_{k+1}^{(j)})\phi_j(\boldsymbol{z}_{k+1}^{(j)}) \tag{54}$$

Of course, this equality is trivial when $\boldsymbol{z}_{k+1}^{(j)} = \boldsymbol{z}_k^{(j)}$. Considering step two of the algorithm ensures that both states $\boldsymbol{z}_{k+1}$ and $\boldsymbol{z}_k$ lie in F_i and the joint transition probability is the product of the marginals (Eq. (52)), then the complete reversibility criterion follows

$$q(\boldsymbol{z}_{k+1}|\boldsymbol{z}_k)\phi(\boldsymbol{z}_k|F_i) = q(\boldsymbol{z}_k|\boldsymbol{z}_{k+1})\phi(\boldsymbol{z}_{k+1}|F_i) \tag{55}$$

As a result, if the current sample $\boldsymbol{z}_k$ is distributed according to $\phi(\cdot|F_i)$ then the marginal distribution of $\boldsymbol{z}_{k+1}$ is given by

$$\begin{aligned} q(\boldsymbol{z}_{k+1}) &= \int q(\boldsymbol{z}_{k+1}|\boldsymbol{z}_k)\phi(\boldsymbol{z}_k|F_i)d\boldsymbol{z}_k \\ &= \int q(\boldsymbol{z}_k|\boldsymbol{z}_{k+1})\phi(\boldsymbol{z}_{k+1}|F_i)d\boldsymbol{z}_k \\ &= \phi(\boldsymbol{z}_{k+1}|F_i)\int q(\boldsymbol{z}_k|\boldsymbol{z}_{k+1})d\boldsymbol{z}_k \\ &= \phi(\boldsymbol{z}_{k+1}|F_i) \end{aligned} \tag{56}$$

As a result, the next step of the Markov chain is also distributed according to $\phi(\cdot|F_i)$. □

Statistical properties of the subset simulation estimator

Each intermediate failure probability, $P_i = P(F_i)$, is estimated by Monte Carlo simulation. In the first subset, i.e. P_1 is estimated by standard Monte Carlo simulation from random samples following the *strong law of large numbers* and is therefore unbiased, consistent, and normally distributed as $N \to \infty$ by the *Central Limit Theorem.* The coefficient of variation (CoV) of the estimate $\tilde{P}_1$ is given by

$$\delta_1 = \sqrt{\frac{1-P_1}{P_1 N}} \tag{57}$$

All subsequent conditional probabilities, $P_2, \dots, P_m$, are estimated from samples generated using MCMC. More specifically, they are generated using a total of N points from sequential states of N_c Markov chains such that each change has length N/N_c. As proven above, these points follow the appropriate conditional probability distribution but are not *independent.* Nonetheless, the estimators are still unbiased, consistent, and normally distributed – i.e. the Strong Law of Large Numbers and Central limit theorem are still valid [5].

Following Au and Beck, an estimator of the CoV for $\tilde{P}_i$ is now derived. We first assume that the N_c chains are uncorrelated in the following sense. In the $(i-1)$ conditional level, define $I_{F_i}(\boldsymbol{z})$ to be the indicator that $\boldsymbol{z}$ is in F_i. We then assume that $E[I_{F_i}(\boldsymbol{z})I_{F_i}(\boldsymbol{z}')] - P(F_i|F_{i-1})^2 = 0$ if $\boldsymbol{z}$ and $\boldsymbol{z}'$ are from different chains. Note that this assumption is removed in the derivation provided by Shields et al. [7]. Next, let $\boldsymbol{z}_{jk}^{(i)}$ be the k^{th} sample from the j^{th} Markov chain in conditional level i, $I_{jk}^{(i)} = I_{F_i}(\boldsymbol{z}_{jk}^{(i-1)})$, and $P_i = P(F_i|F_{i-1})$ for $i = 1, \dots, m$. We have

$$\mathbb{E}[(\tilde{P}_i - P_i)^2] = E\left[\left(\frac{1}{N}\sum_{j=1}^{N_c}\sum_{k=1}^{N/N_c}(I_{jk}^{(i)} - P_i)\right)^2\right] \tag{58}$$

Because the cross-correlation between chains is assumed to be equal to zero, i.e.

$$\mathbb{E}\left[(I_{jk}^{(i)} - P_i)(I_{ln}^{(i)} - P_i)\right] - P_i^2 = 0 \tag{59}$$

this simplifies as

$$\mathbb{E}[(\tilde{P}_i - P_i)^2] = \frac{1}{N^2}\sum_{j=1}^{N_c}\mathbb{E}\left[\left(\sum_{k=1}^{N/N_c}(I_{jk}^{(i)} - P_i)\right)^2\right] \tag{60}$$

For the j^{th} chain,

$$\begin{aligned} \mathbb{E}\left[\left(\sum_{k=1}^{N/N_c}(I_{jk}^{(i)} - P_i)\right)^2\right] &= \sum_{k=1}^{N/N_c}\sum_{l=1}^{N/N_c} \mathbb{E}\left[(I_{jk}^{(i)} - P_i)(I_{jl}^{(i)} - P_i)\right] \\ &= \sum_{k=1}^{N/N_c}\sum_{l=1}^{N/N_c} R_i(k-l) \end{aligned} \tag{61}$$

where

$$R_i(k) = \mathbb{E}\left[(I_{jl}^{(i)} - P_i)(I_{j,l+k}^{(i)} - P_i)\right] = \mathbb{E}[I_{jl}^{(i)} I_{j,l+k}^{(i)}] - P_i^2 \tag{62}$$

is the covariance between $I_{jl}^{(i)}$ and $I_{j,l+k}^{(i)}$ for any $l = 1, \dots, N/N_c$ and it is independent of l due to stationarity. The covariance is also independent of the chain index j because all chains are probabilistically equivalent. Evaluating the sum in Eq. (61) with respect to $k - l$ yields,

$$\begin{aligned} \mathbb{E}\left[\left(\sum_{k=1}^{N/N_c}(I_{jk}^{(i)} - P_i)\right)^2\right] &= \frac{N}{N_c}R_i(0) + 2\sum_{k=1}^{N/N_c-1}\left(\frac{N}{N_c} - k\right)R_i(k) \\ &= \frac{N}{N_c}\left[R_i(0) + 2\sum_{k=1}^{N/N_c-1}\left(1 - \frac{kN_c}{N}\right)R_i(k)\right] \end{aligned} \tag{63}$$

Applying Eq. (63) to Eq. (60) yields

$$\begin{aligned} \mathbb{E}[(\tilde{P}_i - P_i)^2] &= \frac{1}{N}\left[R_i(0) + 2\sum_{k=1}^{N/N_c-1}\left(1 - \frac{kN_c}{N}\right)R_i(k)\right] \\ &= \frac{R_i(0)}{N}\left[1 + 2\sum_{k=1}^{N/N_c-1}\left(1 - \frac{kN_c}{N}\right)\rho_i(k)\right] \end{aligned} \tag{64}$$

where $\rho_i(k) = R_i(k)/R_0(k)$ is correlation coefficient at lag k of the stationary sequence $\{I_{jk}^{(i)} : k = 1, \dots, N/N_c\}$.

Finally, we recognize that $I_{jk}^{(i)}$ is a Bernoulli random variable such that $R_i(0) = \mathrm{Var}[I_{jk}^{(i)}] = P_i(1 - P - i)$ and therefore the variance of $\tilde{P}_i$ is given by

$$\sigma_i^2 = \mathbb{E}[(\tilde{P}_i - P_i)^2] = \frac{P_i(1 - P_i)}{N}[1 + \gamma_i] \tag{65}$$

where

$$\gamma_i = 2\sum_{k=1}^{N/N_c-1}\left(1 - \frac{kN_c}{N}\right)\rho_i(k). \tag{66}$$

The CoV, δ_i, of $\tilde{P}_i$ is therefore given by

$$\delta_i = \sqrt{\frac{1 - P_i}{P_i N}(1 - \gamma_i)} \tag{67}$$

The covariance sequence $R_i(k), i = 0, \dots, N/N_c - 1$ can be estimated using the Markov chain samples in the $(i - 1)$ conditional level by

$$R_i(k) \approx \left(\frac{1}{N - kN_c}\sum_{j=1}^{N}{}_c\sum_{l=1}^{N/N_c-k} I_{jl}^{(i)} I_{j,l+k}^{(i)}\right) - \tilde{P}_i^2 \tag{68}$$

It follows then that the correlations $\rho_i(k)$, and hence γ_i can be estimated.

Here we notice that the CoV estimator δ_i in Eq. (67) is nearly the same as the Monte Carlo estimator for subset one, δ_1 in Eq. (57), with the only difference being the factor $(1-\gamma_i)$ that corrects for the correlation in the Markov chains. In this way, we can think of the effective Monte Carlo estimator from a sample of size $N/(1+\gamma_i)$. That is, the correlation in the chains serves to reduce the effective sample set size by a factor $1/(1+\gamma_i)$ such that larger correlations will cause lower effective sample size and hence higher CoV.

If we further assume that all conditional estimators $\tilde{P}_i$ are uncorrelated, then their CoVs are additive, and we have

$$\delta^2 = \sum_{i=1}^{m} \delta_i^2. \tag{69}$$

Drawbacks of subset simulation

Subset simulation is very effective for many problems. However, as noted by several authors, it can have trouble estimating failure probabilities under certain conditions. As noted by Breitung [8], who provided a number of counter examples, subset simulation may break down for complex limit states such as those with multiple failure regions and those with strong non-linearities.

Furthermore, it is often assumed that subset simulation is performed in standard normal space, as in the derivations above. In general this is not a requirement, but under conditions of strongly non-Gaussian distributions and nonlinear failure domains, the conditional levels may become very difficult to sample using standard random walk Metropolis Hastings methods. To correct for this, Shields et al. [7] proposed to use an affine invariant ensemble sample with stretch moves that is capable of sampling from degenerate distributions while Wang et al. [9] proposed to use Hamiltonian Monte Carlo. Numerous other MCMC methods have been proposed for use in subset simulation, some of which are summarized by Papaioannou et al. [6].

6.3 Other Variance Reduction Methods

Although importance sampling and subset simulation are perhaps the most widely-used Monte Carlo methods for reliablity analysis, it is worth mentioning that several other variance reduction methods have been proposed. Xiao et al. [10] use a strategy called *parallel tempering*, which is motivated by thermodynamic integration and closely related to *transitional Monte Carlo* [11]. In fact, Catanach and Beck [12] show that the tempering approach serves to generalize subset simulation.

Other works have specifically looked at the use of stratified sampling methods, including Latin hypercube sampling for reliability analysis. Olsson et al. [13], for example, show how to use Latin hypercube sampling within importance sampling to improve the efficiency of reliability analysis.

Finally, recent methods have explored the use of control variates, and related methods, for probability of failure estimation. This is most common in the setting of *multi-fidelity* methods in which multiple models of varying fidelity are employed and information from all models is fused to estimate failure probabilities. For example, Kramer et al. [14] presented an approach to fuse probability of failure estimates from models of different fidelities, while Peherstorger et al. [15] proposed the multi-fidelity importance sampling.

7 Surrogate Models for Reliability

A common approach to reliability analysis, as in more general uncertainty quantification problems, is to construct a surrogate model for the input-output relationship defined through the model. This surrogate model can then be deployed with Monte Carlo simulation to rapidly assess the probability of failure. The main challenge of this approach is that it requires the surrogate model to be accurate in the failure regime, which is often located in the tails of the input distribution where little training data is available for the surrogate model.

To encourage the generation of surrogate model training data in the vicinity of the failure domain, specifically near the limit surface, active learning strategies can be employed in which the training data are generated sequentially using a so-called *learning function* (or acquisition function). These methods are motivated by approaches in Bayesian optimization, reviews of which can be found in [16,17]. Here, we will introduce the general framework for active learning using Gaussian process regression (Kriging) surrogates for reliability analysis, with some discussion of specific learning functions.

The general active learning procedure for reliability, as first introduced by Bichon et al. [18] and outlined by Echard et al. [19], involves the following general steps:

1. *Generate initial training data.* In this step, a small number of Monte Carlo samples are drawn and the model is evaluated.
2. *Fit a Kriging surrogate model.* Using the initial training data, a Kriging model is fit, as discussed in detail in Module 3.4.
3. *Identify the "best" point to add to the training set.* This involves the computational of a *learning function* evaluated at each point in a set, $\mathcal{S}$, of a large number of candidate samples. These samples are generated in the domain using Monte Carlo sampling. Various learning functions will be discussed below.
4. *Repeat steps 2 and 3 until a convergence criterion is satisfied.* Here, the Kriging model is actively retrained at each iteration to improve its accuracy for P_f estimation. Some discussion of stopping criteria will be provided below.
5. *Estimate probability of failure.* This is done using Monte Carlo simulation employing the final, converged Kriging surrogate model.

Learning Functions

The most important component of the active learning process described above is the definition of a *learning function.* The learning function is used to select the next point in the sample space to add to the training set. Here we will specifically discuss two widely used learning functions.

Expected Feasibility Function

The first active learning approach for reliability analysis was the *Expected Global Reliability Analysis* (EGRA) method proposed by Bichon et al. [18]. In the EGRA, a new point is selected at each iteration according to the so-called *Expected Feasibility Function* (EFF). Consider the Kriging surrogate model $\hat{g}(\boldsymbol{x})$ approximation of the true performance function $g(\boldsymbol{x})$. The EFF uses the Kriging prediction to assess how well the true value of the performance function at a point $\boldsymbol{x}$ is expected to satisfy the equality constraint $g(\boldsymbol{x}) = a$. Knowing that the Kriging prediction is Gaussian with mean $\mu_{\hat{g}(\boldsymbol{x})}$ and standard deviation $\sigma_{\hat{g}}(\boldsymbol{x})$, we can compute the expectation $\mathbb{E}[g(\boldsymbol{z}) = a]$ by integrating over a region in the immediate vicinity of a, i.e.

$$\text{EFF}(\boldsymbol{x}) = \int_{a-\epsilon}^{a+\epsilon} [\epsilon - |a - g|] f_{\hat{g}}(g) dg \tag{70}$$

where g denotes a realization of the Gaussian random variable $\hat{g}(\boldsymbol{x})$. This integral can be expressed analytically as

$$\begin{aligned}\mathrm{EFF}(\boldsymbol{x}) =&(\hat{g}(\boldsymbol{x})-a)\left[2\Phi\left(\frac{a-\hat{g}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right)-\Phi\left(\frac{(a-\epsilon)-\hat{g}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right)-\Phi\left(\frac{(a+\epsilon)-\hat{g}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right)\right]\\ &-\sigma_{\hat{g}}(\boldsymbol{x})\left[2\phi\left(\frac{a-\hat{g}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right)-\phi\left(\frac{(a-\epsilon)-\hat{g}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right)-\phi\left(\frac{(a+\epsilon)-\hat{g}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right)\right]\\ &+\left[\Phi\left(\frac{(a-\epsilon)-\hat{g}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right)-\Phi\left(\frac{(a+\epsilon)-\hat{g}(\boldsymbol{x})}{\sigma_{\hat{g}}(\boldsymbol{x})}\right)\right]\end{aligned} \tag{71}$$

where Φ and ϕ are the standard normal CDF and PDF respectively. For reliability analysis, we set $a = 0$ to identify the limit state and use $\epsilon = 2\sigma_{\hat{g}}(\boldsymbol{x})$. The candidate point in step 3 that maximizes the EFF, i.e.

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\mathcal{S}} \mathrm{EFF}(\boldsymbol{x}) \tag{72}$$

The EFF specifically seeks points that are both in the vicinity of the limit surface and have high uncertainty. The iterations cease when the $\max \mathrm{EFF}(\boldsymbol{x}) < \tau$ where τ is typically set equal to 0.001.

U-function

In reliability analysis, the sign of the performance function is critical. When $g(\boldsymbol{x})$ is negative, failure occurs. When $g(\boldsymbol{x})$ is positive, the system is safe. The $U-function$ specifically seeks points whose Kriging prediction has a high probability of incorrectly predicting the sign of the performance function. The U-function is defined through the relation

$$|\hat{g}(\boldsymbol{x})| - U(\boldsymbol{x})\sigma_{\hat{g}}(\boldsymbol{x}) = 0 \tag{73}$$

The quantity

$$U(\boldsymbol{x}) = \frac{|\hat{g}(\boldsymbol{x})|}{\sigma_{\hat{g}}(\boldsymbol{x})} \tag{74}$$

specifically measures the distance, in units of Kriging standard deviations, between the prediction $\hat{g}(\boldsymbol{x})$ and the limit surface $g(\boldsymbol{x}) = 0$. It can be interpreted as the reliability index for the probability of incorrectly predicting the sign of $g(\boldsymbol{x})$. That is, the quantity $\Phi(-U(\boldsymbol{x}))$ represents the probability of incorrectly predicting the sign.

To apply the U-function in the active learning framework above, we seek to identify the point that maximizes the probability of incorrect sign prediction. Hence, we seek the point that minimizes $U(\boldsymbol{x})$ as

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathcal{S}} U(\boldsymbol{x}) \tag{75}$$

The iterations stop when $\min U(\boldsymbol{x}) > \tau_u$ where τ_u is typically set equal to 2 such that the probability of incorrect sign prediction, $\Phi(-2) = 0.023$.

References

[1] A. Tabandeh, G. Jia, P. Gardoni, A review and assessment of importance sampling methods for reliability analysis, Structural Safety 97 (2022) 102216.

[2] I. Papaioannou, C. Papadimitriou, D. Straub, Sequential importance sampling for structural reliability analysis, Structural safety 62 (2016) 66–75.

[3] G. L. Ang, Kernel method in Monte Carlo importance sampling, University of Illinois at Urbana-Champaign, 1991.

[4] S.-K. Au, J. L. Beck, A new adaptive importance sampling scheme for reliability calculations, Structural safety 21 (2) (1999) 135–158.

[5] S.-K. Au, J. L. Beck, Estimation of small failure probabilities in high dimensions by subset simulation, Probabilistic engineering mechanics 16 (4) (2001) 263–277.

[6] I. Papaioannou, W. Betz, K. Zwirglmaier, D. Straub, Mcmc algorithms for subset simulation, Probabilistic Engineering Mechanics 41 (2015) 89–103.

[7] M. D. Shields, D. G. Giovanis, V. Sundar, Subset simulation for problems with strongly non-gaussian, highly anisotropic, and degenerate distributions, Computers & Structures 245 (2021) 106431.

[8] K. Breitung, The geometry of limit state function graphs and subset simulation: Counterexamples, Reliability Engineering & System Safety 182 (2019) 98–106.

[9] Z. Wang, M. Broccardo, J. Song, Hamiltonian monte carlo methods for subset simulation in reliability analysis, Structural Safety 76 (2019) 51–67.

[10] S. Xiao, S. Reuschen, G. Köse, S. Oladyshkin, W. Nowak, Estimation of small failure probabilities based on thermodynamic integration and parallel tempering, Mechanical Systems and Signal Processing 133 (2019) 106248.

[11] J. Ching, Y.-C. Chen, Transitional markov chain monte carlo method for bayesian model updating, model class selection, and model averaging, Journal of engineering mechanics 133 (7) (2007) 816–832.

[12] T. A. Catanach, J. L. Beck, Bayesian updating and uncertainty quantification using sequential tempered mcmc with the rank-one modified metropolis algorithm, arXiv preprint arXiv:1804.08738 (2018).

[13] A. Olsson, G. Sandberg, O. Dahlblom, On latin hypercube sampling for structural reliability analysis, Structural safety 25 (1) (2003) 47–68.

[14] B. Kramer, A. N. Marques, B. Peherstorfer, U. Villa, K. Willcox, Multifidelity probability estimation via fusion of estimators, Journal of Computational Physics 392 (2019) 385–402.

[15] B. Peherstorfer, T. Cui, Y. Marzouk, K. Willcox, Multifidelity importance sampling, Computer Methods in Applied Mechanics and Engineering 300 (2016) 490–509.

[16] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, N. De Freitas, Taking the human out of the loop: A review of bayesian optimization, Proceedings of the IEEE 104 (1) (2015) 148–175.

[17] P. I. Frazier, A tutorial on bayesian optimization, arXiv preprint arXiv:1807.02811 (2018).

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

[19] B. Echard, N. Gayton, M. Lemaire, Ak-mcs: an active learning reliability method combining kriging and monte carlo simulation, Structural Safety 33 (2) (2011) 145–154.

Nomenclature

Functions

$\mathbb{E}[\cdot]$ Expected value of a random variable. Also denoted $\mu_X \triangleq E[X]$