

COMPUTATIONAL ISSUES IN FORM WITH UNIFORM RANDOM VARIABLES

André. T. Beck¹

¹Department of Structural and Engineering, São Carlos School of Engineering,
University of São Paulo (atbeck@sc.usp.br)

Abstract. *The First Order Reliability Method is well accepted as an efficient way to solving structural reliability problems with linear or moderately non-linear limit state functions. High non-linearity is introduced in reliability problems by non-linear mechanical responses, but also by correlation between the random variables and by highly non-Gaussian probability distribution functions. Correlation and non-Gaussian distributions introduce non-linearities in the mapping to the standard Gaussian space, hence making search for the design point more challenging. This article discusses computational issues in finding the design point in problems involving uniform and other bounded random variables. The discussion covers the Principle of Normal Tail Approximation, and the mapping to standard Gaussian space. It addresses three different techniques to impose the domain of bounded random variables, when mapping them back from standard Gaussian to original design space. A simple but challenging academic problem is presented, involving a simply-supported beam subject to a concentrated load of random intensity. The concentrated load can occupy a random position over the beam, following an uniform distribution. Although the underlying mechanical problem is very simple, the uniform distribution introduces severe non-linearities, which makes finding the design point a very demanding task. The article also addresses algorithms that can be used to find the design point in highly non-linear problems. It is shown that the Hassofer-Lind-Rackwitz-Fiessler (HLRF) algorithm fails to converge. The performance of the improved HRFL (iHLRF) and of the Sequential Quadratic Programming algorithms are investigated with respect to two alternative mappings to standard Gaussian space and to three techniques to impose the bounds in the inverse probability mapping.*

Keywords: *structural reliability, FORM, design point, probability mapping, normal tail approximation, standard Gaussian space.*

1. INTRODUCTION

The First Order Reliability Method is well accepted as an efficient way to solve structural reliability problems with linear or moderately non-linear limit state functions. High non-linearity is introduced in reliability problems by non-linear mechanical responses [6], but also

by correlation between the random variables and by highly non-Gaussian probability distributions. Correlation, non-Gaussianity and generally bounded distributions introduce non-linearities in the mapping to standard Gaussian space, making the search for the design point more demanding.

This article discusses challenges in finding the design point in problems involving uniform and other bounded random variables. The discussion covers the Principle of Normal Tail Approximation [2,3], and the mapping to the standard Gaussian space. It is observed that the conventional mapping becomes highly non-linear as, during interactive design point search, some variable is pushed against its lower or upper limit. An alternative mapping is proposed, in order to make computations more stable. As standard Gaussian space is unbounded, during design point search any bounded random variable could try to exceed its limits, causing breakdown of the computations. Hence, bounds have to be imposed when mapping random variables back to the original design space. Three different schemes are presented in this article to impose the limits of bounded random variables: bisection, reflection and limit. The three schemes are tested in combination with the traditional and the alternative (proposed) probability mapping. A challenging academic problem is also presented in the article. It involves a simply-supported beam subject to a concentrated load of random intensity. The load can occupy a random position over the beam, following an uniform distribution. Although the underlying mechanical problem is very simple, the uniform distribution introduces strong non-linearities, which makes finding the design point a challenging task.

It is shown that the Hasofer-Lind-Rackwitz-Fiessler (HLRF) algorithm fails to converge for most configurations of the proposed problems. Hence, the improved HRFL (iHLRF) and Sequential Quadratic Programming (SQP) algorithms are also employed in the solutions. The performance of these algorithms is investigated with respect to two mappings to standard Gaussian space and to the three schemes to impose the bounds in the reverse probability mapping. In total, twenty-four configurations of the proposed benchmark problem are investigated. The article is finished with relevant conclusions with respect to the several alternatives proposed herein for the solution of reliability problems involving uniform and other bounded random variables.

2. FORM AND PROBABILITY MAPPING

2.1. The First Order Reliability Method (FORM)

Let \mathbf{X} be a random variable vector describing the uncertainties in geometry, material properties and loading. A limit state function $g_{\mathbf{x}}(\mathbf{X})$ is written in such a way as to divide the random variable domain in safety and failure domains:

$$\Omega_s = \{\mathbf{x} | g_{\mathbf{x}}(\mathbf{x}) > 0\} \text{ is the safety domain}$$

$$\Omega_f = \{\mathbf{x} | g_{\mathbf{x}}(\mathbf{x}) \leq 0\} \text{ is the failure domain.} \quad (1)$$

The failure probability is given by:

$$P_f = P[\mathbf{X} \in \Omega_f] = \int_{\Omega_f} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}. \quad (2)$$

where $f_{\mathbf{X}}(\mathbf{x})$ denotes the joint probability density of random vector \mathbf{X} . The First Order Reliability Method (FORM) is generally accepted as an efficient way to solving Eq. (2), for low dimensions of vector \mathbf{X} and for linear or moderately non-linear limit state functions $g_{\mathbf{x}}(\mathbf{X})$. In the FORM method, Eq. (2) is solved indirectly, by introducing a convenient mapping from the space of the original random vector \mathbf{X} (\mathbf{x} -space) to the so-called standard Gaussian space (\mathbf{u} -space) [3, 9, 10]:

$$\mathbf{u} = \mathbb{T}[\mathbf{x}], \quad g(\mathbf{u}, t) = g_{\mathbf{x}}(\mathbb{T}^{-1}[\mathbf{u}], t) \quad (3)$$

where all components of vector \mathbf{u} are independent and identically distributed standard Gaussian random variables. This mapping can be accomplished by means of the Principle of Normal Tail Approximation (to be described) and by the Nataf transformation.

In standard space, the joint probability density $f_{\mathbf{U}}(\mathbf{u})$ is rotationally symmetric: hence, the point \mathbf{u}^* on the limit state function $g(\mathbf{u})$ which is closest to the origin represents the most probable failure point, also known as **design point**. This feature allows search for the design point to be cast as a constrained optimization problem:

$$\begin{aligned} \mathbf{u}^* &= \arg \min \frac{1}{2} \mathbf{u}^t \cdot \mathbf{u} \\ \text{subject to } g(\mathbf{u}) &= 0 \end{aligned} \quad (4)$$

From Eq. (4), $\beta = \|\mathbf{u}^*\| = (\mathbf{u}^{*t} \cdot \mathbf{u}^*)^{1/2}$ is the so-called Hasofer-Lind reliability index [5, 11] which comes to be the distance between \mathbf{u}^* and the origin of standard Gaussian space. The FORM method, therefore, consists in finding the design point \mathbf{u}^* and approximating the original limit state function $g(\mathbf{u})$ by a tangent hyper-surface at the design point. Hence, the first-order approximation of the failure probability becomes:

$$P_f = P[g(\mathbf{U}) \leq 0] \simeq \Phi(-\beta) \quad (5)$$

where $\Phi(\cdot)$ is the cumulative standard Gaussian probability distribution function.

For the highly non-linear problems to be considered herein, the first order approximation of the failure probability is unlikely to be accurate. However, it is still important to be able to find the design point, for instance, in order to perform Monte Carlo simulation with importance sampling using the design point [4] or to construct precise response surface approximations centered on the design points [7].

It is important to point out that the concept of a design point as the most probable failure point is theoretical, and associated to the transformation to standard Gaussian space. For instance, for problems involving only uniform random variables, all points in the failure domain are equally likely in \mathbf{x} -space... For problems involving a single uniform random variable, multiple equally likely failure points would exist in \mathbf{x} -space. However, the concept and the usual interpretations remain valid in \mathbf{u} -space.

2.2 Traditional normal tail approximation

The Principle of Normal Tail Approximation [3] allows a non-Gaussian variable X to be transformed in an “equivalent” Gaussian variable Y by means of a point-wise probability mapping:

$$F_Y(x^*) = F_X(x^*) \quad (6)$$

Eq. (6) is a probability-preserving equation, which makes the probability content to the “left” and “right” of point x^* to remain unaltered in the mapping. However, two parameters (μ_Y and σ_Y - mean and standard deviation) of the “equivalent” Gaussian variable Y have to be determined. Hence, an additional equation is required. The second equation that has been traditionally used to complete the mapping is:

$$f_Y(x^*) = f_X(x^*) \quad (7)$$

This equation is completely arbitrary, and is not required for probability preservation. Other equations and rules for the probability mapping have also been presented elsewhere (Chen and Lind, 1983).

Introducing the Hasofer-Lind transformation [5]:

$$u^* = \frac{x^* - \mu_Y}{\sigma_Y} \quad (8)$$

the mapping can be accomplished directly to standard Gaussian space. Manipulating Eqs. (6) to (8), well-known equations are obtained for the desired parameters (μ_Y and σ_Y) :

$$\begin{aligned} u^* &= \Phi^{-1}(F_X(x^*)) \\ \sigma_Y &= \frac{\phi(u^*)}{f_X(x^*)} \\ \mu_Y &= x^* - u^* \sigma_Y \end{aligned} \quad (9)$$

Now, for illustration purposes, assume that random variable X follows an uniform distribution with parameters :

$$X \sim U(x_a, x_b) \quad (10)$$

Figure 1 illustrates how the traditional mapping behaves as the point x^* approaches the upper limit ($x^* = x_b - \varepsilon$ with $\varepsilon \rightarrow 0$). It can be observed that $\sigma_Y \rightarrow 0$, which could cause breakdown of the numerical computations. The observed behavior also introduces strong non-linearities in the probability mapping, which can make the design point search unstable, if any of the x^* points comes too close to a random variable limit.

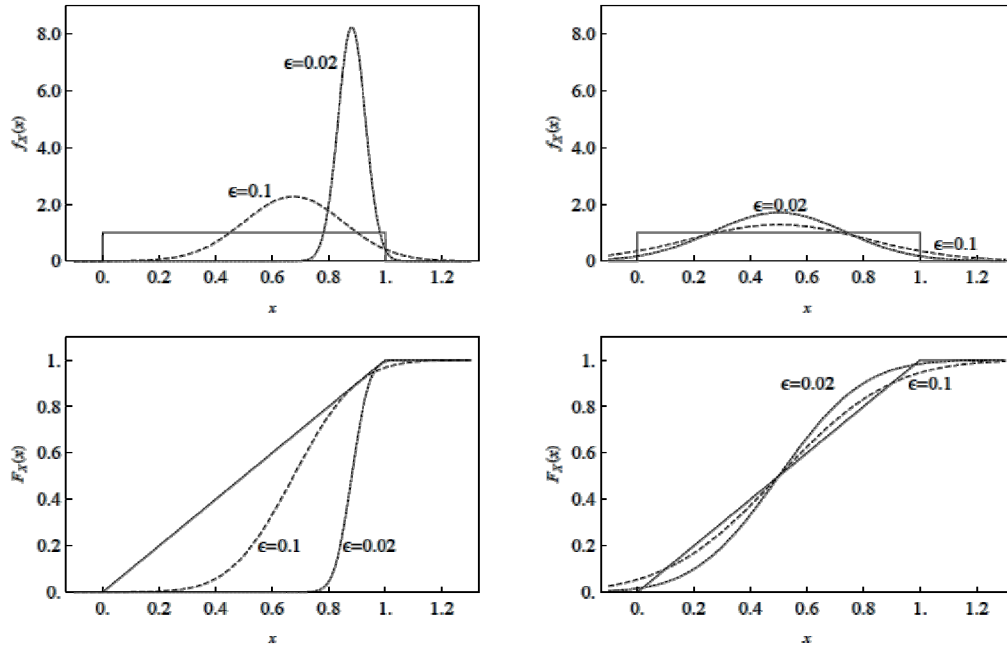


Figure 1. Traditional (left) and alternative (right) standard Gaussian mapping as point x^* approaches the upper limit of a bounded random variable ($x^* = x_b - \varepsilon$ with $\varepsilon \rightarrow 0$).

2.3 Alternative normal tail approximation

In an attempt to avoid potential problems with the traditional probability mapping, as observed in Figure 1, an alternative mapping is proposed herein. As Eq. (7) is arbitrary, it is proposed to use the simple rule instead:

$$\mu_Y = \mu_X \quad \text{or} \quad \mu_Y = \text{mode}_X \quad (11)$$

The standard deviation σ_Y is obtained directly from Eq. (8), using the relation $u^* = \Phi^{-1}(F_X(x^*))$, which originates in the probability-preserving equation (Eq. 6). Figure 1 shows that the resulting mapping is more stable as a point x^* approaches the upper limit ($x^* = x_b - \varepsilon$ with $\varepsilon \rightarrow 0$). When point x^* is equal to the mean or mode, the alternative mapping cannot be used, and the traditional mapping is used instead (only for one iteration). The alternative mapping proposed herein is compared with the traditional mapping in connection to some example problems in the sequence.

2.4 Imposing random variable bounds in the inverse mapping

With the mapping to standard Gaussian space, all variables become unbounded. Hence, when a new, trial point is evaluated during iterative design point search, in \mathbf{u} -space, it could become unfeasible for exceeding a limit of the original random variable in \mathbf{x} -space. Hence, when mapping a new point back to original design space, the random variable limits have to be imposed. Although this appears to be an obvious problem with the mapping to standard Gaussian space, it has not been explicitly addressed in the published literature.

In this section, three alternative schemes are presented for bounding limited random variables: bisection, reflection and limit. Assume that the lower and upper bounds of random variable X are x_a and x_b , respectively. The bisection bounding scheme is:

$$\begin{aligned} &\text{while } \left[x_{k+1} \geq x_b, x_{k+1} = \frac{x_{k+1} + x_k}{2} \right] \\ &\text{while } \left[x_{k+1} \leq x_a, x_{k+1} = \frac{x_{k+1} + x_k}{2} \right] \end{aligned} \quad (12)$$

The reflection scheme is obtained by:

$$\begin{aligned} &\text{if } [x_{k+1} \geq x_b, x_{k+1} = x_b - (x_{k+1} - x_b)] \\ &\text{if } [x_{k+1} \leq x_a, x_{k+1} = x_a + (x_a - x_{k+1})] \end{aligned} \quad (13)$$

As stated in Eq. (13), the reflection scheme is not guaranteed to generate feasible points. Hence, after performing the reflection, the point should be checked again using the limit scheme.

For the limit scheme, one should choose a limiting value ε which is compatible with the accuracy of the numerical representation of the standard Gaussian cumulative probability distribution function, $\Phi(\cdot)$, and its inverse $\Phi^{-1}(\cdot)$ - see first line of Eq. (9). In this article, $\varepsilon = 10^{-8}$ is considered. This corresponds to a reliability index of $\beta = -\Phi^{-1}(10^{-8}) \approx 5.6$. This is still safely below the operational limit of the StRAnD software [1], beyond which the numerics used in the computations crumble ($\beta = 8, \Phi(-8) \approx 10^{-16}$). The limit bounding scheme is simply :

$$\begin{aligned} &\text{if } [x_{k+1} \geq x_b, x_{k+1} = x_b - \varepsilon] \\ &\text{if } [x_{k+1} \leq x_a, x_{k+1} = x_a + \varepsilon] \end{aligned} \quad (14)$$

The three bounding schemes allow the computation to continue safely, and also allow the sequence of points to return naturally to the interior of the feasible domain, if the design point is located there.

3. OPTIMIZATION ALGORITHMS

3.1 The Hasofer, Lind, Rackwitz and Fiessler (HLRF) algorithm

The most popular algorithm used to solve the reliability problem stated in Eq. (4) is the Hasofer and Lind [5], Rackwitz and Fiessler [11] algorithm (HLRF). This algorithm is known to be very efficient when it converges, but is also known to fail to converge very often [8, 13]. The algorithm is based on finding the zero of successive linearizations of the limit state surface, leading to the well-known sequence:

$$\mathbf{u}_{k+1} = \frac{1}{\|\nabla g(\mathbf{u}_k)\|^2} (\nabla g(\mathbf{u}_k) \cdot \mathbf{u}_k - g(\mathbf{u}_k)) \cdot \nabla g(\mathbf{u}_k)^T \quad (15)$$

In Eq. (15), $\nabla g(\mathbf{u}_k)$ is the gradient vector of the limit-state function with respect to \mathbf{u} evaluated at \mathbf{u}_k .

3.2 The improved Hasofer, Lind, Rackwitz and Fiessler (iHLRF) algorithm

An improved Hasofer, Lind, Rackwitz and Fiessler (iHLRF) algorithm was presented by Zhang and Kiureghian [13]. In the iHLRF algorithm, a linear search is performed:

$$\mathbf{u}_{k+1} = \mathbf{u}_k + \alpha_k \cdot \mathbf{d}_k \quad (16)$$

in the direction given by the HLRF algorithm:

$$\mathbf{d}_k = \frac{1}{\|\nabla g(\mathbf{u}_k)\|^2} (\nabla g(\mathbf{u}_k) \cdot \mathbf{u}_k - g(\mathbf{u}_k)) \cdot \nabla g(\mathbf{u}_k)^T - \mathbf{u}_k \quad (17)$$

In Eqs. (16) and (17), \mathbf{d}_k is a search direction and α_k is a step size, selected from minimization of an appropriate merit function. The following merit function is proposed in ref. [13]:

$$m(\mathbf{u}) = \frac{1}{2} \|\mathbf{u}\| + c |g(\mathbf{u})| \quad (18)$$

where c is a penalty parameter. The linear search is defined as:

$$\alpha_k = \arg \min m(\mathbf{u}_k + \alpha_k \cdot \mathbf{d}_k) \quad (19)$$

The linear search can be performed approximately using the popular Armijo rule [13]. An appropriate step α_k is selected, such as to obtain a sufficient decrease in the merit function:

$$\alpha_k = \max_{i \in \mathbb{N}} [b^i |m(\mathbf{u}_k + b^i \cdot \mathbf{d}_k) - m(\mathbf{u}_k)| \leq -a b^i \|\nabla m(\mathbf{u}_k)\|^2], \quad a, b > 0 \quad (20)$$

Following Zhang and Kiureghian [13], direction \mathbf{d}_k given in Eq. (17) is a descent direction of the merit function in Eq. (18), for $\forall \mathbf{u} \in \mathbb{R}^n$, as long as the following condition is satisfied:

$$c > \frac{\|\mathbf{u}\|}{\|\nabla g(\mathbf{u})\|} \quad (21)$$

This condition is used as a rule for updating the penalty parameter:

if $|g(\mathbf{u}_k)| < 10^{-1} |g(\mathbf{u}_0)|$ then

$$c_k = \eta \max \left[\frac{\|\mathbf{u}_k\|}{\|\nabla g(\mathbf{u}_k)\|}, \frac{1}{2} \frac{\|\mathbf{u}_k + \mathbf{d}_k\|^2}{|g(\mathbf{u}_k)|} \right]$$

$$\text{else } c_k = \eta \frac{\|\mathbf{u}_k\|}{\|\nabla g(\mathbf{u}_k)\|} \quad (22)$$

Literature-suggested values [8, 13] for the algorithmic constants are $\eta = 2, a = 0.1$ and $b = 0.5$. The value $\eta = 2$ is used throughout the article in the numerical examples. The recommend values $a = 0.1$ and $b = 0.5$ are also used for the Armijo line search in the numerical examples. However, it was found that the iHLRF does not converge for many of the tested problems using the recommended values. Hence, the iHLRF algorithm is also tested using $a = 0.5$ and $b = 0.5$. For convenience, the two resulting algorithms are referred to as iHLRF15 and iHLRF55, respectively, in the results section.

3.3 Sequential Quadratic Programming (SQP)

The constrained optimization problem stated in Eq. (4) can be converted in an unconstrained problem by introducing the associated Lagrangian problem [8, 12]:

$$\mathcal{L}(\mathbf{u}, \lambda) = \frac{1}{2} \mathbf{u}^t \cdot \mathbf{u} + \lambda g(\mathbf{u}) \quad (23)$$

If \mathbf{u}^* is a solution to Eq. (4), then there exists a Lagrangian multiplier λ^* such that [8, 12]:

$$\nabla \mathcal{L}(\mathbf{u}^*, \lambda^*) = \mathbf{u}^{*t} + \lambda^* \nabla g(\mathbf{u}^*) = 0 \quad (24)$$

This can be used to cast the optimization problem in terms of the following set of simultaneous equations:

$$\begin{aligned} \mathbf{u}^t + \lambda \nabla g(\mathbf{u}) &= 0 \\ g(\mathbf{u}) &= 0 \end{aligned} \quad (25)$$

The Newton method can be used to solve the above system of non-linear equations iteratively:

$$\begin{Bmatrix} \mathbf{u}_{k+1} \\ \lambda_{k+1} \end{Bmatrix} = \begin{Bmatrix} \mathbf{u}_k \\ \lambda_k \end{Bmatrix} + \alpha_k \begin{Bmatrix} \mathbf{d}_k \\ \kappa_k \end{Bmatrix} \quad (26)$$

The solution to this problem is given by :

$$\begin{Bmatrix} \mathbf{d}_k \\ \kappa_k \end{Bmatrix} = - \begin{bmatrix} \nabla^2 \mathcal{L}(\mathbf{u}_k, \lambda_k) & \nabla g(\mathbf{u}_k)^t \\ \nabla g(\mathbf{u}_k) & 0 \end{bmatrix} \cdot \begin{Bmatrix} \mathbf{u}_k \\ g(\mathbf{u}_k) \end{Bmatrix} \quad (27)$$

This result happens to be the solution and the Lagrangian multiplier for the following quadratic optimization problem:

$$\begin{aligned} (\mathbf{u}^*, \lambda^*) &= \arg \min \frac{1}{2} \mathbf{d}^t \nabla^2 \mathcal{L}(\mathbf{u}, \lambda) \mathbf{u} + \mathbf{u}^t \mathbf{d} \\ &\text{subject to } g(\mathbf{u}) + \nabla g(\mathbf{u})^t \mathbf{d} \end{aligned} \quad (28)$$

Therefore, this algorithm is called Sequential Quadratic Programming algorithm (SQP). The present implementation of SQP follows the guidelines set up by Schittkowski [12], as detailed in [8]. The step α_k in Eq. (26) is computed using the augmented Lagrangian function :

$$\mathcal{L}^+(\mathbf{u}, \lambda) = \frac{1}{2} \mathbf{u}^t \cdot \mathbf{u} + \lambda g(\mathbf{u}) + c g(\mathbf{u})^2 \quad (29)$$

where c is a penalty parameter, also updated iteratively.

The solution in Eq. (27) requires the Hessian of the Lagrangian function ($\nabla^2 \mathcal{L}(\mathbf{u}_k, \lambda_k)$), which is generally not known. In this implementation, the Hessian is replaced by an approximate matrix, which is updated iteratively, following the BFGS scheme. However, due to our annoyance with non-convergence of SQP for some problems, the exact Hessian was also used. The computations using the exact Hessian are identified as SQPeH in the numerical results session.

4. A SIMPLE BUT CHALLENGING BENCHMARK PROBLEM

Consider a simply-supported beam subject to a concentrated load, as illustrated in Figure 2. The load of intensity P can occupy any position over the beam span L . Both load position and intensity are assumed to follow uniform distributions:

$$D \sim U(d_a, d_b), \quad P \sim U(p_a, p_b) = U(0.6, 1.0) \quad (30)$$

The load position limits are $d_a = 0$ for all problems, and d_b varies following Table 1. The different values of d_b change the problem considerably, as can be observed in the design points and reliability index results presented in Table 1. Three different cases of correlation are also considered: no correlation, positive ($\rho=0.5$) and negative ($\rho=-0.5$) correlation between variables D and P . Also, variants of the two-dimensional problem are obtained for the load intensity P following Normal and Log-normal distributions, following Table 1. In total, 24 two-dimensional problems (variants) are considered.

Table 1. Variable data and design point results for two-dimensional problems.

2D Problem	D	P	ρ	ε	β	Design Point d^* p^*
1	U(0.0,1.0)				1.15	0.50 0.95
2	U(0.0,0.6)	U(0.6,1.0)	0.0	0.05	1.41	0.46 0.95
3	U(0.0,0.5)				1.78	0.43 0.97
4	U(0.0,1.0)				1.30	0.51 0.95
5	U(0.0,0.6)	U(0.6,1.0)	0.5	0.05	1.19	0.48 0.95
6	U(0.0,0.5)				1.46	0.44 0.96
7	U(0.0,1.0)				1.29	0.48 0.95
8	U(0.0,0.6)	U(0.6,1.0)	-0.5	0.05	1.94	0.45 0.95
9	U(0.0,0.5)				2.46	0.42 0.97
10			0.0	10^{-8} corner probs.	-	0.50 1.00
11	U(0.0,0.5)	U(0.6,1.0)	0.5		-	0.50 1.00
12			-0.5		-	0.50 1.00
13	U(0.0,1.0)				3.75	0.50 0.95
14	U(0.0,0.6)	N(0.8,0.04)	0.0	0.05	3.87	0.49 0.95
15	U(0.0,0.5)				4.14	0.47 0.95
16	U(0.0,1.0)				0.94	0.50 0.95
17	U(0.0,0.6)	N(0.8,0.16)	0.0	0.05	1.20	0.44 0.96
18	U(0.0,0.5)				1.45	0.40 0.99
19	U(0.0,1.0)				3.46	0.50 0.95
20	U(0.0,0.6)	LN(0.8,0.04)	0.0	0.05	3.59	0.49 0.95
21	U(0.0,0.5)				3.87	0.47 0.96
22	U(0.0,1.0)				0.97	0.50 0.95
23	U(0.0,0.6)	LN(0.8,0.16)	0.0	0.05	1.21	0.43 0.97
24	U(0.0,0.5)				1.44	0.39 1.00

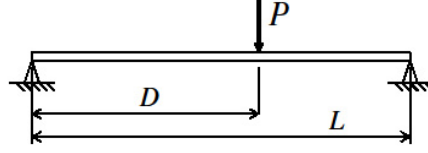


Figure 2. Simply-supported beam subject to a randomly-located concentrated load of random intensity.

The limit state function for the two-dimensional problems is:

$$g_{\mathbf{x}}(\mathbf{X}) = R - P \left(D - \frac{D^2}{L} \right) \quad (31)$$

where the resistance R is given by $R = p_b L / 4 - \varepsilon$. Parameter ε controls the size of the failure domain and hence the magnitude of the failure probability. For most (conventional) problems, $\varepsilon = 0.05$ is considered (Table 1). A problem similar to the one-dimensional problem above (called “corner” problem herein), is obtained for $\varepsilon = 10^{-8}$. For the corner problems, the \perp_{tol} convergence criterion is not used. Variables L, p_a, p_b, d_a, d_b and R are treated as non-dimensional. For all problems, $L = 1$.

5. RESULTS

Results are presented in this section for a combination of four optimization algorithms (HLRF, iHLRF with two sets of parameters, SQP), three schemes for imposing random variable bounds (bisection, reflection and limit) and for two probability mappings (traditional and alternative normal tail approximation). The combination results in 24 algorithms, which are employed in the solution of 24 two-dimensional problems and four 8 and 16-dimensional problems. Due to strong problem non-linearities, several initial points have to be considered for each problem and algorithm. Six different starting points are considered for each problem, totaling 6048 problem solutions. Initial points considered for the two-dimensional problems are:

$$\begin{aligned} \mathbf{x}_1 &= \{\mu_D, \mu_P\}; \mathbf{x}_2 = \{0.03, 0.62\}; \mathbf{x}_3 = \{0.05, 0.62\}; \\ \mathbf{x}_4 &= \{0.05, 0.61\}; \mathbf{x}_5 = \{0.05, 0.80\}; \mathbf{x}_6 = \{0.05, 0.99\}; \end{aligned} \quad (32)$$

In the problem solutions, it is observed that the alternative mapping leads to similar trajectories for close-by initial points, whereas the traditional mapping leads to more irregular trajectories. Difficulties were observed in reaching the prescribed tolerance ($\|\mathbf{x}\|_{\text{tol}} = 10^{-3}$), as towards the end of the convergence history several (successive) points remain clogged.

Figure 3 illustrates the convergence trajectories for one initial point but different optimization algorithms, also for problem 1 (Table 1). In this figure, it can be observed that the HLRF algorithm gets trapped in a circle, bouncing from one side of the mid-span to the other, hence failing to converge. This happens for most of the 24 2D problems, when the HLRF algorithm is used. The iHLRF15, iHLRF55 and SQP algorithms do converge for this problem, as shown.

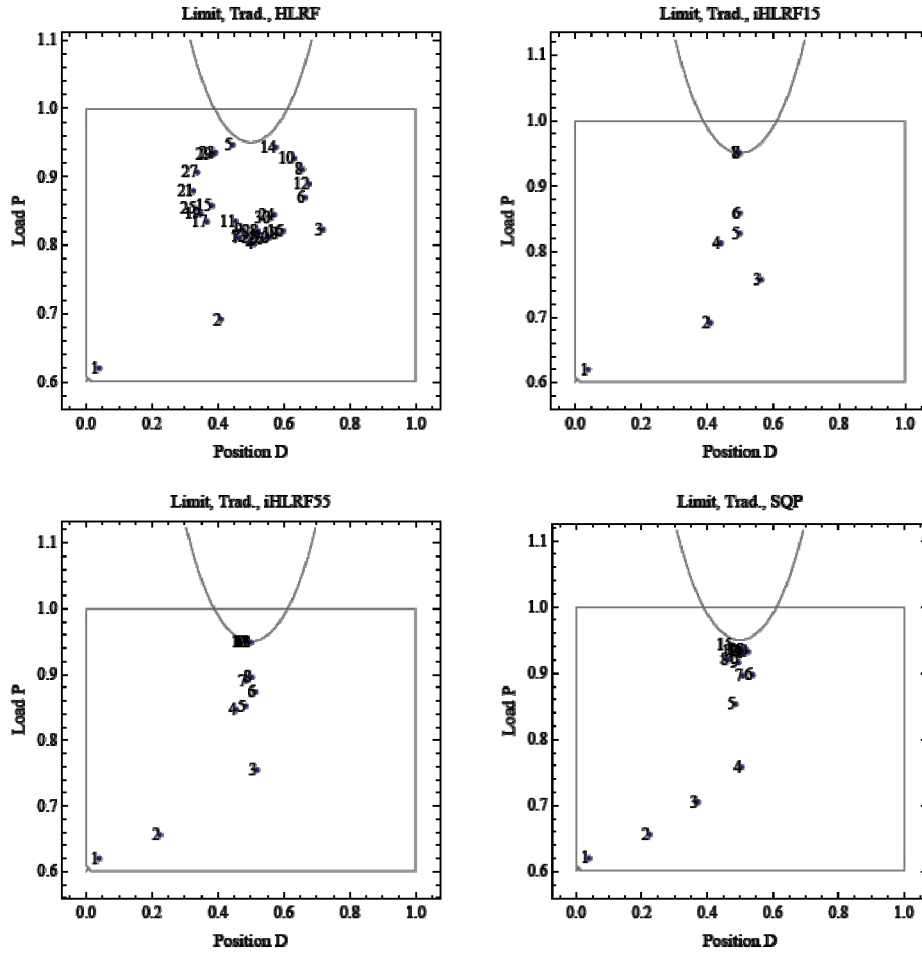


Figure 3. Converge trajectories for different algorithms with traditional mapping; Problem 1.

The performance of different algorithms can be evaluated from Table 2, which summarizes results for the 12 2D problems with uniform probability distributions for load P . As the particular starting points for which convergence was achieved are of no general interest, results are summarized for each algorithm and mapping. Details of starting point and bounding schemes are omitted. In Table 2, a number indicates that convergence was achieved for the six starting points and for the three bounding schemes (18 runs); in this case the number is the average number of limit state function calls required for convergence. Keywords are as indicated in the footnote to Table 2.

For the problems with $d_b = 1.0$, both positive and negative correlations lead to greater convergence difficulties. For the other problems, correlations actually speeded up convergence, leading to smaller number of limit state function calls.

In the solution of the 12 problems summarized in Table 2, the iHLRF55 was the most efficient algorithm. It converged for all problems using the traditional mapping, and for most problems using the alternative mapping. The iHLRF15 algorithm converged for fewer problems, and when it converged, it took on average 13% more limit state function calls than the iHLRF55 algorithm. The SQP algorithm also converged for all problems using the traditional mapping, but in general required from 30 to 42% more limit state function calls than the iHLRF55 algorithm.

In Table 2 it can also be observed that the proposed alternative mapping leads to convergence difficulties for the so-called corner problems. For all other problems, however, the alternative mapping led to an average reduction of around 10% in the required number of limit state function calls.

Table 2. Summary of results for 2D Problems 1 to 12.

$\rho \rightarrow$		$d_b = 1.0$			$d_b = 0.6$			$d_b = 0.5$			<i>corner</i>		
		0.0	+0.5	-0.5	0.0	+0.5	-0.5	0.0	+0.5	-0.5	0.0	+0.5	-0.5
Alter.	HLRF	none	some	few	none	few	few	none	none	one	none	few	none
	iHLRF15	86	107	116	52	56	96	some	46	few	none	few	none
	iHLRF55	85	149	99	46	40	71	58	45	73	some	some	few
	SQP	140	242	203	46	47	81	44	45	97	96	50	some
	SQPcH	47	64	62	44	45	44	44	44	43	most	45	few
Trad.	HLRF	none	none	none	none	some	few	none	31	none	22*	22	16
	iHLRF15	82	143	119	62	44	some	some	68	one	55*	54	51*
	iHLRF55	97	111	111	57	38	86	64	46	75	68*	53	106
	SQP	260	283	223	44	some	most	50	44	most	57	45	77
	SQPcH	67	92	81	45	47	44	45	45	44	47	46	74

Keywords none and some are obvious; few means that convergence was obtained for up to three starting points (for one or more bounding schemes); most means that convergence was obtained for all but up to three starting points; *mean* means that convergence was obtained only for the mean starting point; an * indicates that convergence was only obtained for the limit bounding scheme.

Table 3. Summary of results for 2D Problems 12 to 24.

$d_b \rightarrow$		$P \sim N(0.8, 0.04)$			$P \sim N(0.8, 0.16)$			$P \sim LN(0.8, 0.04)$			$P \sim LN(0.8, 0.16)$		
		1.0	0.6	0.5	1.0	0.6	0.5	1.0	0.6	0.5	1.0	0.6	0.5
Alter.	HLRF	none	none	some	none	none	none	none	none	one	none	none	some
	iHLRF15	492	349	some	167	33	53	700	290	some	99	60	29
	iHLRF55	some	218	145	71	37	55	some	315	147	65	52	31
	SQP	some	some	some	some	27	27	some	some	78	124	28	27
	SQPcH	some	none	none	27	27	27	mean	none	some	27	27	27
Trad.	HLRF	none	none	none	none	none	none	none	none	some	none	none	none
	iHLRF15	449	162	85	153	61	44	712	190	most	110	42	29
	iHLRF55	some	159	93	75	41	60	none	198	82	57	48	42
	SQP	some	some	some	some	30	27	some	some	52	97	27	27
	SQPcH	some	none	28	28	27	27	mean	none	28	28	27	27

Keywords following footnote to Table 2.

For 2D Problems with $d_b = 1.0$ or $d_b = 0.6$, the convergence sequences never tried to leave the random variable limits, for the four tested algorithms. Hence, bounding schemes were not activated. For the conventional problem with $d_b = 0.5$ (problems 3, 6 and 9), bounding schemes were requested occasionally, for some of the starting points.

As expected, for the corner problems the bounding schemes were more active, as the convergence sequence led to more frequent boundary hits. Figure 4 illustrates convergence

difficulties using the alternative mapping and the limit bounding scheme. For the HLRF and iHLRF15 algorithms, the sequence remains trapped between the limits d_b and p_b . For the traditional mapping (Figure 5), the convergence is much smoother: the sequence “sticks” at the upper limit p_b and follows on until the “design point” is found. For the traditional mapping, even the HLRF algorithm converges to the design point of the corner problem. This failure of the alternative mapping in allowing the design point of corner problems to be found is a little surprising. However, the success of the traditional mapping is certainly a consequence of the robust computational scheme implemented herein.

Convergence difficulties were encountered using the bisection and reflection bounding schemes, even when the traditional mapping is used. For the HLRF and iHLRF15 algorithms, the sequences also remain trapped near the design point, but are not able to get there. Only the SQP algorithm was able to find the design point using the bisection and reflection schemes. For all other algorithms, only the limit scheme allowed the design point to be found. Clearly, this is a particularity of the corner problems considered here, as the design point was arbitrarily located at the upper limits of the two random variables. This would not occur very often in practical problems.

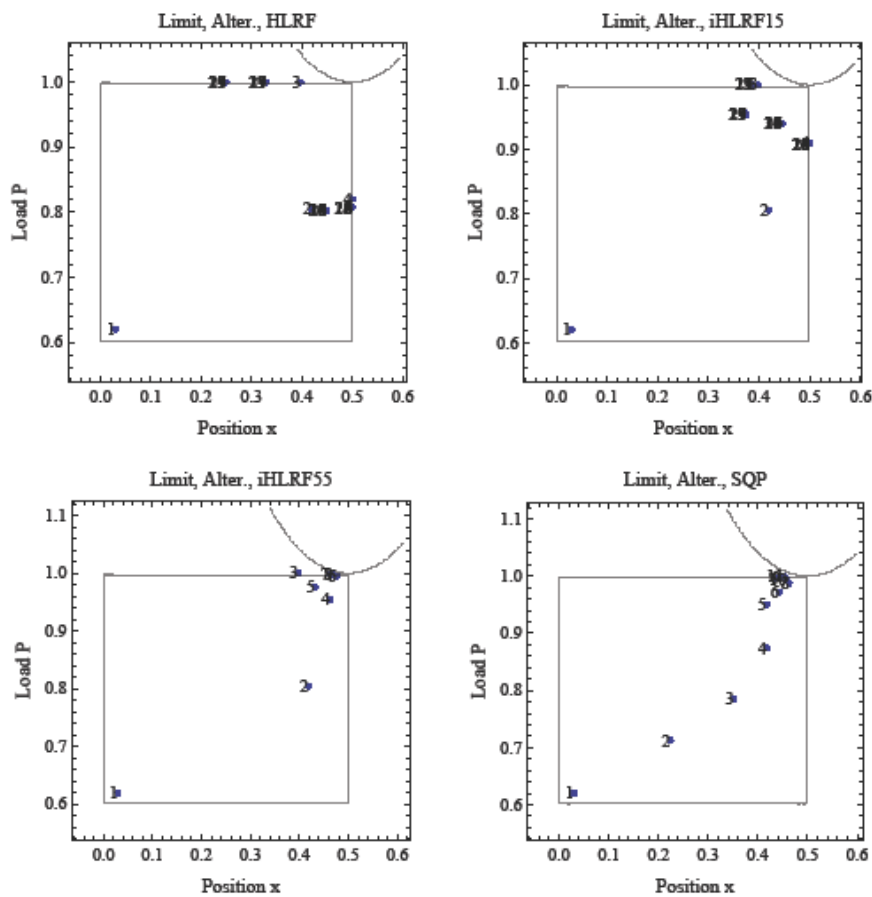


Figure 4. Converge trajectories for different algorithms with alternative mapping and limit bounding; 2D Problem 12.

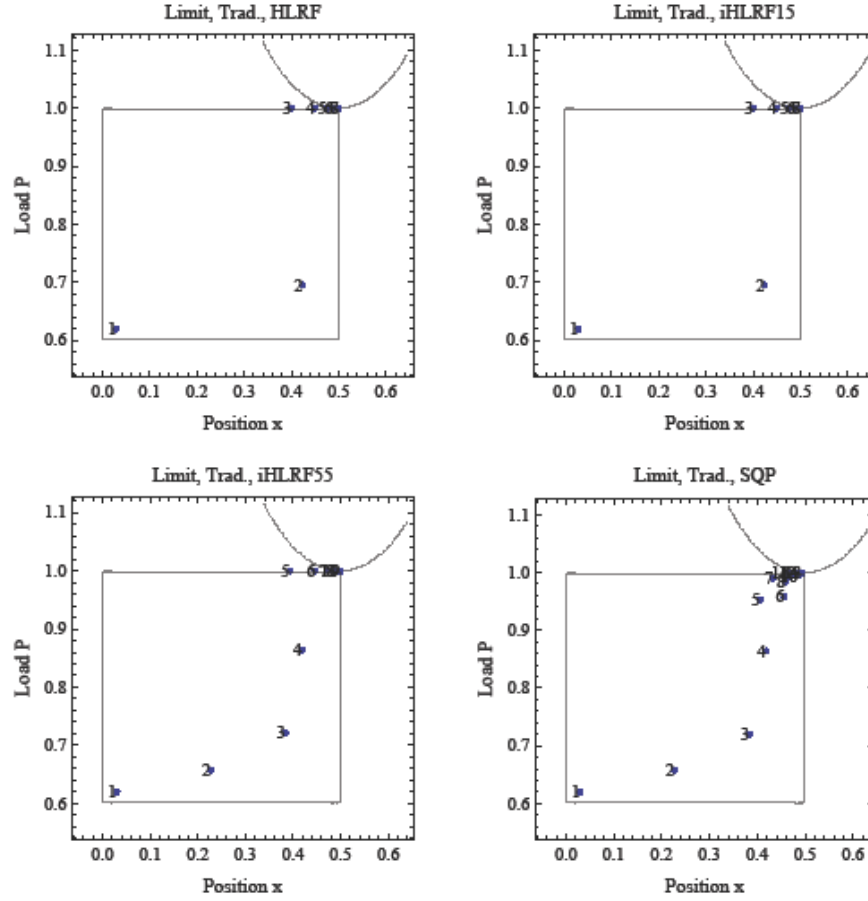


Figure 5. Converge trajectories for different algorithms with traditional mapping and limit bounding; 2D Problem 12.

In general, it was also observed that the alternative mapping leads to more frequent boundary hits, in comparison to the traditional mapping. This justifies the observed convergence difficulties for the corner problem and the alternative mapping. Moreover, both iHLRF algorithms tested herein lead to more frequent boundary hits than the SQP algorithm, which appears to lead to more well-behaved convergence sequences.

Results obtained for problems 12 to 24, with Normal or Log-normal distribution of load intensity P are presented in Table 3. None of the studied algorithms was able to identify the design point for all the starting points of all problems. The SQP algorithm failed to converge for most problems involving small coefficients of variation of the load P (c.o.v. = 0.05 for problems 13 to 15 and 19 to 21). Figure 6 illustrates convergence difficulties for SQP algorithm and problem 13. The SQP algorithm remains trapped in a circle, bouncing from one side of the mid-span to the other. This behavior also happened for the computation using the exact Hessian (SQPeH).

The iHLRF15 and iHLRF55 algorithms solved most problems, but failed to converge for some of them. For those problems where both algorithms converged, the iHLRF55 algorithm required between 20 to 30% less limit state function calls than the iHLRF15 algorithm.

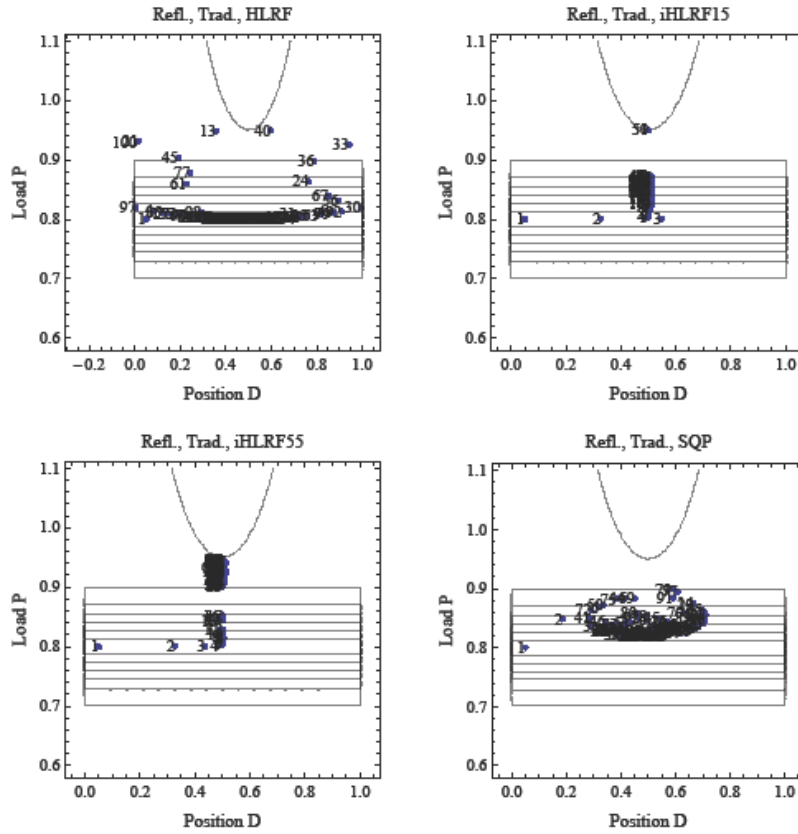


Figure 6: Converge trajectories for different algorithms with traditional mapping; 2D Problem 13.

6. CONCLUSIONS

This article presented and discussed strategies for finding the design point in problems involving uniform and other bounded random variables. The article introduced a challenging benchmark problem, involving one concentrated load acting on a simply supported beam. Although mechanically simple, finding the design point was shown to be quite challenging, due to the uniform distribution of load positions and due to random load intensities. Several topics were discussed and illustrated with respect to results obtained for this benchmark problem.

In particular, the article discussed alternatives to be considered when random variables approach their upper or lower limits during design point search. An alternative transformation was suggested, in order to smoothen the mapping to standard Gaussian space: the mean of the equivalent normal distribution is simply given by the mean, or the mode, of the original distribution. This alternative procedure has indeed smoothened some of the studied problems, leading to convergence for some cases where the traditional normal tail approximation failed, but also reducing in around 10% the required number of limit state function calls. However, the suggested alternative mapping failed when the design point was located very close to a random variable limit.

Three schemes were presented herein to impose the bounds of the random variables in the inverse probability mapping: bisection, reflection and limit. The bisection and reflection

schemes resulted in similar behavior for most problems, with the reflection scheme leading to convergence failure in a few cases. The limit scheme was shown to perform better in the (unlikely) case of a design point lying very close to a random variable limit. Indeed, for some of the presented “corner” problems, the limit scheme was the only scheme leading to convergence.

Four algorithms were studied with respect to their robustness and efficiency in finding the design point, in combination with the alternative and traditional normal tail approximation, and with the three bounding schemes tested herein. The HLRF algorithm converged for only a few cases, remaining trapped in some sort of circle for most of the problems studied. The Sequential Quadratic Programming (SQP) algorithm converged for most of the problems studied, but failed to converge for the 2D problems involving Normal and Log-normal load intensities and small c.o.v.. The improved HLRF (iHLRF) algorithm converged for some of the problems studied, using the Armijo line search with the literature-recommended values of $a=0.1$, $b=0.5$. However, using the parameters $a=0.5$, $b=0.5$, the algorithm converged for a larger number of problems with a significant reduction in the mean number of limit state function calls.

Results were also obtained for multi-dimensional versions of the 2D problems presented herein. These results will be presented at the conference.

Acknowledgements

The author kindly acknowledges financial support of this research project by the agencies CNPq (National Council for Research and Development), CAPES (National Council for Post-Graduate Education) and FAPESP (São Paulo State Foundation for Research).

REFERENCES

- [1] Beck AT, 2007: StRAnD Version 1.03 Users Manual, Department of Structural Engineering, University of São Paulo (in portuguese).
- [2] Chen X, Lind NC, 1983: Fast probability integration by three-parameter normal tail approximation, *Structural Safety* 1, 269-276.
- [3] Ditlevsen O, 1981: Principle of normal tail approximation, *J. Eng. Mech. Div (ASCE)* 107, 1191-1208.
- [4] Englund S, Rackwitz R, 1993: A benchmark study on importance sampling techniques in structural reliability, *Structural Safety* 12, 255-276.
- [5] Hasofer AM, Lind NC, 1974: An exact and invariant first order reliability format. *Journal of the Engineering Mechanics Division*, 10, 111-21.
- [6] Haukaas T, Der Kiureghian A, 2006: Strategies for finding the design point in non-linear

finite element reliability analysis, Probabilistic Engineering Mechanics 21, 133-147.

- [7] Leonel ED, Beck AT, Venturini WS, 2011: On the Performance of Response Surface and Direct Coupling Approaches in Solution of Random Crack Propagation Problems, Structural Safety 33, 261-274.
- [8] Liu PL, Der Kiureghian A, 1991: Optimization algorithms for structural reliability, Structural Safety 9, 161-177.
- [9] Madsen HO, Krenk S, Lind NC, 1986: Methods of structural safety. Prentice Hall, Englewood Cliffs.
- [10] Melchers RE, 1999: Structural Reliability Analysis and Prediction, second edition, John Wiley and Sons, NY.
- [11] Rackwitz R and Fiessler B, 1978: Structural reliability under combined random load sequences, Computers & Structures 9, 489–494.
- [12] Schittkowski K, 1983: On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function, Math. Operationsforschung und Statistik, Ser. Optimization 14, 197-216.
- [13] Zhang Y and Der Kiureghian A, 1997: Finite element reliability methods for inelastic structures, Report UCB/SEMM