 CMES, vol.83, no.1, pp.23-55, 2012

New Optimization Algorithms for Structural Reliability Analysis

S.R. Santos[1], L.C. Matioli[2] and A.T. Beck[3]

Abstract: Solution of structural reliability problems by the First Order method require optimization algorithms to find the smallest distance between a limit state function and the origin of standard Gaussian space. The Hassofer-Lind-Rackwitz-Fiessler (HLRF) algorithm, developed specifically for this purpose, has been shown to be efficient but not robust, as it fails to converge for a significant number of problems. On the other hand, recent developments in general (augmented Lagrangian) optimization techniques have not been tested in aplication to structural reliability problems. In the present article, three new optimization algorithms for structural reliability analysis are presented. One algorithm is based on the HLRF, but uses a new differentiable merit function with Wolfe conditions to select step length in linear search. It is shown in the article that, under certain assumptions, the proposed algorithm generates a sequence that converges to the local minimizer of the problem. Two new augmented Lagrangian methods are also presented, which use quadratic penalties to solve nonlinear problems with equality constraints. Performance and robustness of the new algorithms is compared to the classic augmented Lagrangian method, to HLRF and to the improved HLRF (iHLRF) algorithms, in the solution of 25 benchmark problems from the literature. The new proposed HLRF algorithm is shown to be more robust than HLRF or iHLRF, and as efficient as the iHLRF algorithm. The two augmented Lagrangian methods proposed herein are shown to be more robust and more efficient than the classical augmented Lagrangian method.

Keywords: Structural reliability, HLRF-based algorithms, Nonlinear programming, augmented Lagrangian methods.

[1] State University of Paraná, Department of Mathematics, CP 415, 87303-100, Campo Mourão-PR, Brazil. Ph.D. Program in Numerical Methods in Engineering, Federal University of Paraná, Curitiba-PR, Brazil. Partially supported by Fundação Araucária. E-mail: solaregina@fecilcam.br

[2] Federal University of Paraná, Department of Mathematics, Centro Politécnico, CP 19081, 81531-990, Curitiba-PR, Brazil. E-mail: matioli@ufpr.br

[3] University of São Paulo, São Carlos School of Engineering, Department of Structural Engineering, 13566-590, São Carlos-SP, Brazil. E-mail: atbeck@sc.usp.br

1 Introduction

Let $\mathbf{X}$ be a random variable vector of system parameters, describing the uncertainties in structural loads, material resistances and member geometry:

$$\mathbf{X} = (X_1, X_2, \ldots, X_n)^T . \tag{1}$$

A performance function

$$h(\mathbf{X}) = h(X_1, X_2, \ldots, X_n) \tag{2}$$

is written in such a way as to divide the failure and survival domains:

$$D_f = \{\mathbf{x} | h(\mathbf{x}) \leq 0\} \text{ is the failure domain,} \tag{3}$$

$$D_s = \{\mathbf{x} | h(\mathbf{x}) > 0\} \text{ is the safety domain.} \tag{4}$$

The limit state equation is defined by $h(\mathbf{X}) = 0$. Hence, the probability of failure can be evaluated by:

$$P_f = \int \ldots \int_{h(\mathbf{X})<0} f_{\mathbf{X}}(x_1, x_2, \ldots, x_n)\, dx_1 dx_2 \ldots dx_n, \tag{5}$$

where $f_{\mathbf{X}}(x_1, x_2, \ldots, x_n)$ is the joint probability density function for the vector of random variables and the integration is performed over the failure domain, $h(\mathbf{X}) < 0$. The evaluation of Eq. (5) is not easy because: 1. it involves a multi-dimensional integral; 2. the exact form of the joint probability density function is rarely known; 3. the limit state equation $h(\mathbf{X}) = 0$ is not always given in closed form, but as the solution to some numerical algorithm.

Direct solution of Eq. (5) via Monte Carlo is only feasible when P_f is not too small and/or the limit state function is given in closed form.

Approximate solutions can be obtained efficiently using the so-called First Order (FORM) or Second Ordem (SORM) Reliability Methods. Both methods involve a transformation of the original vector of random variables $\mathbf{X}$ to a space of standard Gaussian variables $\mathbf{Y}$. In this space, the so-called design point is the point over the limit state closest to the origin, and is also the point over the failure domain with the maximum likelyhood. The distance (in $\mathbf{Y}$ space) from the design point to the origin is known as reliability index, and denoted by β. Once the design point has been located, the first order solution accounts for a linearization of the limit state function at the design point, resulting in the linear estimate of the failure probability:

$$P_f = \Phi(-\beta) \tag{6}$$

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

Standard textbooks on structural reliability methods provide the details on the transformation required to map points from $\mathbf{X}$ to $\mathbf{Y}$ space [Haldar and Mahadevan (2000); Melchers (2001)]. This transformation involves the Principle of Normal Tail Approximation, proposed by Ditlevsen (1981), which consists in estimating the parameters of an equivalent normal distribution for each design variable, at each point in the design space.

To evaluate β, it is necessary to find the design point, that is, the point $\mathbf{y}^*$ on the failure surface closest to the origin of the standard Gaussian space. This can be expressed by the following constrained optimization problem:

$$\begin{array}{ll} \text{minimize} & f(\mathbf{y}) \\ \text{subject to} & h(\mathbf{y}) = 0 \end{array} \tag{7}$$

where, $f(\mathbf{y}) = \frac{1}{2}\beta^2 = \frac{1}{2}\mathbf{y}^T\mathbf{y}$ is the objective function, with $f : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}$ is the failure surface and $f, h \in C^1$.

The solution of problem (7) has motivated development of dedicated algorithms, as the Hasofer and Lind (1974) and Rackwitz and Fiessler (1978) algorithm (HLRF). This algorithm has been shown to be very efficient, but does not converge for many problems [Liu and Kiureghian (1986, 1992)]. In fact, there is no mathematical proof of convergence for the HLRF algorithm.

Liu and Kiureghian (1992) proposed a modified version of the HLRF algorithm (denoted M-HLRF), including a merit function to induce convergence. The authors have shown that, although the M-HLRF algorithm performs better than HLRF, it does not always generate a globally convergent sequence.

Santosh, Saraf, Ghosh, and Kushwaha (2006) presented an improvement to HLRF by using the same merit function of Liu and Kiureghian (1992), combined with a linear search using the Armijo rule. Convergence proofs for the algorithm where not presented.

Liu and Kiureghian (1986, 1992) compared the performance of different algorithms in solving the reliability problem (Eq. 7). The Projected Gradient, Augmented Lagrangian and Sequential Quadratic Programming methods were tested by the authors, and compared with the HLRF and M-HLRF algorithms. A lack of robustness of the HLRF and of the Augmented Lagrangian methods were identified by the authors. Moreover, the Sequential Quadratic Programming and M-HLRF methods were shown to be more efficient than the other methods for the problems tested by the authors.

Zhang and Kiureghian (1997) developed an improved HLRF algorithm, called iHLRF, by introducing a non-differentiable merit function and using the Armijo

rule to select the step size in linear search. The authors established the conditions under which the algorithm generates a sequence that converges to the minimizer of problem (7).

In this article, a new HLRF-based algorithm, denoted nHLRF, is proposed, including a differentiable merit function and using the Wolfe conditions to determinate the step length. However, unlike previous studies [Liu and Kiureghian (1986, 1992); Santosh, Saraf, Ghosh, and Kushwaha (2006)], convergence of the proposed nHLRF algorithm is proved.

The augmented Lagrangian method studied by Liu and Kiureghian (1992) used a classical quadratic penalty function introduced by Hestenes (1969) and Powell (1969). However, since the publication of Liu and Kiureghian (1992), many new developments of augmented Lagrangian methods were obtained, for example by Ben-Tal and Zibulevsky (1997); Birgin, Castillo, and Martínez (2005); Censor and Zenios (1992); Chen and Teboulle (1993); Eckstein (1993); Iusem and Teboulle (1993, 1995); Iusem, Teboulle, and Svaiter (1994); Kiwiel (1997); Martinez (2000); Matioli and Gonzaga (2008); Rômulo and Gonzaga (2003); Teboulle (1992); Tseng and Bertsekas (1993). This research lead to new penalty functions and modern augmented Lagrangian methods. Based on developments by Tseng and Bertsekas (1993) and Santos and Matioli (2011), two new augmented Lagrangian methods for problem (7) are proposed herein.

The remaining sections of this article are organized as follows. In Section 2, classical and improved versions of the HLRF algorithms are presented. The new, proposed nHLRF algorithm is presented in Section 3, were the proof of convergence is also presented. In Section 4, augmented Lagrangian methods are presented. Numerical problems are presented in Section 5.

2 HLRF algorithm and improvements

The Hasofer and Lind (1974), Rackwitz and Fiessler (1978) algorithm (HLRF) is obtained with the simple sequence:

$$\mathbf{y}^{k+1} = \frac{1}{\|\nabla h(\mathbf{y}^k)\|^2}\left[\nabla h\left(\mathbf{y}^k\right)^T \mathbf{y}^k - h\left(\mathbf{y}^k\right)\right]\nabla h\left(\mathbf{y}^k\right) \tag{8}$$

where the vector $\mathbf{y} \in \mathbb{R}^n$ is defined in the standard Gaussian space and $h : \mathbb{R}^n \to \mathbb{R}$ is the failure surface. The HLRF is a very popular and efficient algorithm for solution of problem (7). However, there is no garantee of convergence and, in fact, the algorithm is known to fail in a considerable number of problems [Liu and Kiureghian (1986, 1992)].

Liu and Kiureghian (1986, 1992) presented an alternative algorithm, called M-HLRF, to circumvent convergence problems of the HLRF algorithm. The authors combined the search direction yielded by HLRF (d^k):

$$d^k = \frac{1}{\|\nabla h(\mathbf{y}^k)\|^2}\left[\nabla h\left(\mathbf{y}^k\right)^T \mathbf{y}^k - h\left(\mathbf{y}^k\right)\right]\nabla h\left(\mathbf{y}^k\right) - \mathbf{y}^k \tag{9}$$

with a linear search in this direction. The linear search is performed until a sufficient decrease in a merit function, $m(\mathbf{y})$, is reached. The following merit function was considered:

$$m(\mathbf{y}) = \frac{1}{2}\left|\mathbf{y} - \frac{\nabla h(\mathbf{y})^T \mathbf{y}}{\|\nabla h(\mathbf{y})\|^2}\nabla h(\mathbf{y})\right|^2 + \frac{1}{2}c \cdot h(\mathbf{y})^2. \tag{10}$$

Although the robustness was improved, Liu and Kiureghian (1986) acknowledged that convergence of the method cannot be guaranteed, since merit function $m(\mathbf{y})$ can have minima which are not solutions to the original problem (Eq. 7). Furthermore, d^k may not be a descent direction for the merit function (Eq. 10) in some cases.

An improved algorithm (iHLRF) was presented by Zhang and Kiureghian (1997), also based on a linear search in the HLRF direction:

$$\mathbf{y}^{k+1} = \mathbf{y}^k + \alpha^k d^k. \tag{11}$$

Note that the recursive formula of the HLRF algorithm is obtained when a full step is used ($\alpha^k = 1$).

The following non-diferentiable merit function is used in the linear search:

$$m(\mathbf{y}) = \frac{1}{2}\mathbf{y}^T\mathbf{y} + c\,|h(\mathbf{y})|. \tag{12}$$

The apropriate step α^k is found using the popular Armijo rule, as the first value to satisfy a condition of sufficient decrease in the merit function:

$$m(\mathbf{y}^{k+1}) \leq m(\mathbf{y}^k) + m_1 \alpha^k \nabla m(\mathbf{y}^k)^T d^k \tag{13}$$

where $\nabla m(\mathbf{y}^k) = \mathbf{y}^k + c \cdot sign(h(\mathbf{y}^k))\nabla h(\mathbf{y}^k)$ and $m_1 \in (0,1)$.

Zhang and Kiureghian (1997) showed that, for $\forall \mathbf{y}^k \in \mathbb{R}^n$, the direction d^k, given in Eq. (9) is a descent direction to $m(\mathbf{y})$, given in Eq. (12), as long as the following condition is satisfied:

$$c > \frac{\left\|\mathbf{y}^k\right\|}{\|\nabla h(\mathbf{y}^k)\|}. \tag{14}$$

In order to satisfy condition (14), Zhang and Kiureghian (1997) established the following rules for updating c^k:

if $\left|h(\mathbf{y}^k)\right| \geq \Delta$

$$c^k = \eta \max\left\{ \frac{\left\|\mathbf{y}^k\right\|}{\left\|\nabla h(\mathbf{y}^k)\right\|}, \frac{1}{2}\frac{\left\|\mathbf{y}^k + d^k\right\|^2}{|h(\mathbf{y}^k)|} \right\};$$

otherwise,

$$c^k = \eta \frac{\left\|\mathbf{y}^k\right\|}{\left\|\nabla h(\mathbf{y}^k)\right\|}.$$

Values of the parameter Δ and η suggested by Zhang and Kiureghian (1997) are $10^{-3}\left|h(\mathbf{y}^0)\right|$ and 2, respectively.

The iHLRF algorithm has been shown to be more efficient and more reliable than the HLRF and M-HLRF algorithms [Zhang and Kiureghian (1997)]. However, one important drawback of the algorithm is the fact that the merit function used is non-differentiable. In order to check if a sufficient reduction in the merit function was achieved during linear search, it is necessary to evaluate the gradient of the merit function (Eq. 12), which is not defined in all points of the domain.

3 Proposed nHLRF algorithm and proof of convergence

In this section, a new (nHLRF) algorithm is proposed, using a differentiable merit function which decreases at each iteration along the linear search in the HLRF direction (Eq. 9). The following merit function is proposed:

$$m(\mathbf{y}) = \frac{1}{2}\mathbf{y}^T\mathbf{y} + \frac{c}{2}h(\mathbf{y})^2 \tag{15}$$

where $c > 0$.

To establish the proof of convergence of the nHLRF algorithm, it is necessary that the HLRF search direction (given in Eq. 9) be a descent direction for the merit function (Eq. 15), for all $\mathbf{y}^k \in \mathbb{R}^n$. In the following, an important result of this paper is presented.

Theorem 3.1. *The HLRF search direction, given in Eq. (9), is a descent direction at any point $y^k \in \mathbb{R}^n$ for the merit function (Eq. 15), provided that:*

$$c > -\frac{1}{h(\mathbf{y}^k)} \cdot \frac{(\mathbf{y}^k)^T \nabla h(\mathbf{y}^k)}{\left\|\nabla h(\mathbf{y}^k)\right\|^2} \tag{16}$$

where $h(\mathbf{y}^k) \neq 0$.

Proof. It must be proven that $\nabla m(\mathbf{y}^k)^T d^k < 0$, for all $\mathbf{y}^k \in \mathbb{R}^n$. One has,

$$\nabla m(\mathbf{y}^k) = \mathbf{y}^k + c \cdot h(\mathbf{y}^k) \nabla h(\mathbf{y}^k). \tag{17}$$

Replacing Eq. (9) in Eq. (17), one obtains

$$\nabla m(\mathbf{y}^k)^T d^k = -\left(\mathbf{y}^{kT} \mathbf{y}^k - \frac{(\nabla h(\mathbf{y}^k)^T \mathbf{y}^k)^2}{\|\nabla h(\mathbf{y}^k)\|^2} \right) - \left(c \cdot h(\mathbf{y}^k)^2 + \frac{h(\mathbf{y}^k) \mathbf{y}^{kT} \nabla h(\mathbf{y}^k)}{\|\nabla h(\mathbf{y}^k)\|^2} \right). \tag{18}$$

Using the Schwartz inequality: $\left|\nabla h(\mathbf{y}^k)^T \mathbf{y}^k\right| < \left\|\nabla h(\mathbf{y}^k)\right\| \cdot \left\|\mathbf{y}^k\right\|$, one concludes that

$$\mathbf{y}^{kT} \mathbf{y}^k - \frac{(\nabla h(\mathbf{y}^k)^T \mathbf{y}^k)^2}{\|\nabla h(\mathbf{y}^k)\|^2} \geq 0. \tag{19}$$

Considering condition (Eq. 16) in evaluation of Eq. (18):

$$c > -\frac{1}{h(\mathbf{y}^k)} \cdot \frac{\mathbf{y}^{kT} \nabla h(\mathbf{y}^k)}{\|\nabla h(\mathbf{y}^k)\|^2}, \tag{20}$$

one is ensured that

$$c \cdot h(\mathbf{y}^k)^2 + \frac{h(\mathbf{y}^k) \mathbf{y}^{kT} \nabla h(\mathbf{y}^k)}{\|\nabla h(\mathbf{y}^k)\|^2} > 0. \tag{21}$$

Hence, combining Eqs. (19) and (21), it follows that $\nabla m(\mathbf{y}^k)^T d^k < 0$.
Furthermore, it is observed that $\nabla m(\mathbf{y}^k)^T d^k = 0$ only if

$$h(\mathbf{y}^k) = 0 \text{ and } \mathbf{y}^{kT} \mathbf{y}^k - \frac{(\nabla h(\mathbf{y}^k)^T \mathbf{y}^k)^2}{\|\nabla h(\mathbf{y}^k)\|^2} = 0,$$

which is equivalent to saying that $\mathbf{y}^k \in \mathbb{R}^n$ satisfies the KKT conditions, thus being a stationary point for problem (7). □

Theorem 3.1 guarantees that merit function (Eq. 15) decreases in direction d^k from $\mathbf{y}^k$ for some $\alpha > 0$. However, a simple reduction of the merit function, $(m(\mathbf{y}^{k+1}) < m(\mathbf{y}^k))$, does not guarantee convergence of the algorithm.

In order to guarantee convergence, a first condition to be imposed is the existence of α satisfying the condition expressed in Eq. (13). However, this may lead to low

performance, due to very short steps. To avoid small steps, the following Curvature condition is considered:

$$\nabla m(\mathbf{y}^k + \alpha d^k)^T d^k \geq m_2 \nabla m(\mathbf{y}^k)^T d^k, \tag{22}$$

for some $m_2 \in (m_1, 1)$, and m_1 given by Eq. (13).

The sufficient decrease (Eq. 13) and curvature (Eq. 22) conditions are known collectively as Wolfe conditions, where $0 < m_1 < m_2 < 1$.

Theorem (3.2) ensures that for any descent direction, d^k, there are points $\mathbf{y}^k + \alpha^k d^k$ satisfying Eqs. (13) and (22).

Theorem 3.2. *Let $m : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable. Suppose that, for all $\mathbf{y}^k \in \mathbb{R}^n$ and $d^k \in \mathbb{R}^n$, $\nabla m(\mathbf{y}^k)^T d^k < 0$ and assume that $\{m(\mathbf{y}^k + \alpha d^k) | \alpha > 0\}$ is bounded from below. Then, if $0 < m_1 < m_2 < 1$, there is $\alpha_u > \alpha_l > 0$ such that $m(\mathbf{y}^k + \alpha^k d^k)$ satisfies the Wolfe conditions, given in Eqs. (13) and (22).*

Proof. See Dennis and Schnabel (1996). □

Theorem 3.3 establishes the conditions for the algorithm to converge to a stationary point of $m(\mathbf{y}^k)$, that is, $\nabla m(\mathbf{y}^k) \to 0$.

Theorem 3.3. *Let $m : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, let $\mathbf{y}^0 \in \mathbb{R}^n$ be a given initial point for the sequence $\{\mathbf{y}^k\}$, defined by $(\mathbf{y}^k + \alpha^k d^k)$ where $d^k \in \mathbb{R}^n$, and the step length $\alpha^k > 0$ satisfying Eqs. (13) and (22) with $0 < m_1 < m_2 < 1$. Assume that:*
(i) the level set $S = \{\mathbf{y} : m(\mathbf{y}) \leq m(\mathbf{y}^0)\}$ is bounded;
(ii) in some neighborhood of S, the function $m(\mathbf{y})$ is Lipschitz continuously differentiable, that is, there exists a constant $L > 0$ such that:

$$\|\nabla m(\mathbf{y}) - \nabla m(\bar{\mathbf{y}})\| \leq L \|\mathbf{y} - \bar{\mathbf{y}}\|$$

for all $\mathbf{y}, \bar{\mathbf{y}} \in S$.
Then,

$$\lim_{k \to \infty} \left\| \nabla m(\mathbf{y}^k) \right\| = 0.$$

Proof. See Dennis and Schnabel (1996). □

From the above one concludes that, given a initial point $\mathbf{y}^0$ and the direction d^k, if the hypothesis of theorems 3.1, 3.2 and 3.3 are valid, algorithm nHLRF will generate a sequence of points which satisfy the Wolfe conditions and which converges to a stationary point of $m(\mathbf{y})$, which is also solution to the problem in Eq. (7).

Given the theoretical results that establish convergence of the method, the nHLRF algorithm is presented next. For this purpose, it is considered that the design variables are defined in standard normal space and are statistically independent.

Algorithm 3.1.
Data: $\mathbf{y}^0 \in \mathbb{R}^n$, $\eta > 0$, $\alpha \in (0,1]$, $0 < m_1 < m_2 < 1$
$k = 0$
WHILE *the stopping criterion is not satisfied*
  COMPUTE *the search direction* d^k *following Eq. (9)*
  DETERMINE c^k
    IF $h(\mathbf{y}^k) = 0$
      Choose $c^k > 0$
    ELSEIF
    $$c^k = \eta \left| \frac{1}{h(\mathbf{y}^k)} \cdot \frac{\mathbf{y}^{kT} \nabla h(\mathbf{y}^k)}{\|\nabla h(\mathbf{y}^k)\|^2} \right|$$
  WHILE *the Wolfe conditions (Eqs. 13 and 22), are not satisfied*
    IF $m(\mathbf{y}^k + \alpha d^k) - m(\mathbf{y}^k) > m_1 \alpha \nabla m(\mathbf{y}^k)^T d^k$
      DO $\alpha = 0.5\alpha$
    ELSEIF $\nabla m(\mathbf{y}^k + \alpha d^k) d^k < m_2 \nabla m(\mathbf{y}^k)^T d^k$
      DO $\alpha = 2\alpha$
    END
    $\alpha^k = \alpha$
    $\mathbf{y}^{k+1} = \mathbf{y}^k + \alpha^k d^k$
  END
  $\beta = \left\| \mathbf{y}^{k+1} \right\|$
  $k = k + 1$
END

In Algorithm 3.1 the conditions to calculate c^k are established in order to circumvent the numerical problem caused when $h(\mathbf{y}^k) = 0$, ensuring that direction d^k, given in Eq. (9), decreases for any $\mathbf{y}^k \in \mathbb{R}^n$. Details about the choice of parameters are given in Section 6.

4 Augmented Lagrangian Methods

Augmented Lagrangian methods are used to minimize problems with equality and inequality constrains. These methods were developed to reduce the possibility of ill-conditioned subproblems generated in the classical approach of penalty methods. For this purpose, at each iteration, estimates of the Lagrange multiplier for equality constraints are obtained.

To understand augmented Lagrangian methods, consider the problem in Eq. (23) in the format of Eq. (7), but with m equality constraints

$$\begin{array}{ll} \text{minimize} & f(\mathbf{y}) \\ \text{subject to} & h(\mathbf{y}) = 0 \end{array} \tag{23}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $h : \mathbb{R}^n \to \mathbb{R}^m$, $\mathbf{y} \in \mathbb{R}^n$ and $f, h \in C^1$.

The Lagrangian function associated to Eq. (23) is defined as:

$$(\mathbf{y}, \lambda) \in \mathbb{R}^n \times \mathbb{R}^m \to \ell(\mathbf{y}, \lambda) = f(\mathbf{y}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{y}) \tag{24}$$

where λ_i, with $i = 1, \ldots, m$, is the Lagrange multiplier associated with the i^{th} equality constraints.

Now consider the following problem:

$$\begin{array}{l} \text{minimize } l(\mathbf{y}, \lambda) = f(\mathbf{y}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{y}) \\ \text{subject to } \; h(\mathbf{y}) = 0. \end{array} \tag{25}$$

At the feasible set, objective function of problems (23) and (25) coincide, hence (23) and (25) have the same solution set. Therefore, from the penalty of problem (25), Hestenes (1969) and Powell (1969) defined the augmented Lagrangian function as:

$$(\mathbf{y}, \lambda, \rho) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_{++} \longmapsto \mathscr{L}(\mathbf{y}, \lambda, \rho) = f(\mathbf{y}) + \sum_{i=1}^{m} \lambda_i h_i(\mathbf{y}) + \frac{\rho}{2} \sum_{i=1}^{m} h_i(\mathbf{y})^2 \tag{26}$$

where ρ is the penalty parameter and λ_i, $i = 1, \ldots, m$, is the estimate of the i^{th} Lagrange multiplier.

Performance of the augmented Lagrangian method with the classical quadratic penalty of Hestenes (1969) and Powell (1969) has been analyzed by Liu and Kiureghian (1992) with respect to structural reliability problems.

In the present article, two new augmented Lagrangian methods are presented to address structural reliability problems. Both methods can also be applied to general nonlinear programming problems with equality constraints. Both methods presented herein use modern quadratic penalties. These methods are extensions of results presented by Matioli and Gonzaga (2008) and Tseng and Bertsekas (1993) for nonlinear programming problems with inequality constraints without introducing slack variables. For the format of problem (7), the augmented Lagrangian functions are defined as:

$$(\mathbf{y},\lambda,\gamma)\in\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}_{++}\rightarrow\mathscr{L}(\mathbf{y},\lambda,\gamma)=f(\mathbf{y})+\lambda h(\mathbf{y})+\frac{\gamma}{2}h(\mathbf{y})^2 \tag{27}$$

and

$$(\mathbf{y},\lambda,\omega)\in\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}_{++}\rightarrow\mathscr{L}(\mathbf{y},\lambda,\omega)=f(\mathbf{y})+\lambda h(\mathbf{y})+\frac{\omega}{2}h(\mathbf{y})^2 \tag{28}$$

where, λ is the Lagrange multiplier associated with $h(\mathbf{y})=0$ and γ and ω are the penalty parameters. The penalty parameter γ in Eq. (27) is defined as:

$$\gamma=\frac{\lambda^2}{r},\ \text{with}\ \ r>0 \tag{29}$$

and ω in Eq. (28) is defined as:

$$\omega=\frac{|\lambda|}{r},\ \text{with}\ \ r\neq 0. \tag{30}$$

Remember that the penalty parameter must be positive.

Note that parameter γ in Eq. (27) and paramater ω in Eq. (28) are both dependent of the multipliers. This is the diference in relation to the classical penalty (Eq. 26).

In augmented Lagrangian methods, the penalty parameter is one of few that we have freedom of choice for its update. In Lemma 4.1, it is shown that for a particular choice of the penalty parameter, the three augmented Lagrangian functions are equivalent.

Lemma 4.1. *If $\lambda\neq 0$, for all $i=1,\ldots,m$, and the penalty parameters γ in (27) and ω in (28) are updated as $\gamma=\frac{\lambda^2}{r}$ (given by Eq. 29) with $r=\frac{\lambda^2}{\rho}$ and $\omega=\frac{|\lambda|}{r}$ (given by Eq. 30) with $r=\frac{|\lambda|}{\rho}$ and $\rho>0$ (given by Eq. 26), then the functions (26), (27) and (28) are the same.*

Proof. The proof is immediate. It is shown for one case, the other is identical. In fact, replacing $\gamma=\frac{\lambda^2}{r}$ with $r=\frac{\lambda^2}{\rho}$ in (Eq. 27), it follows that (Eq. 26) and (Eq. 27) are the same. □

Santos and Matioli (2011) showed that the functions (27) and (28) generated by augmented Lagrangian methods have a local minimizer, on the same way as using the classical penalty of Hestenes (1969) and Powell (1969).

The algorithm for the augmented Lagrangian method using the penalty functions proposed in this paper is presented in the sequence.

Algorithm 4.1.
Data: $\boldsymbol{y}^0 \in \mathbb{R}^n$, $\lambda^0 \in \mathbb{R}$, $\gamma^0 \in \mathbb{R}_{++}$, $w^0 \in \mathbb{R}_{++}$, $\omega^0 \in \mathbb{R}_{++}$
$k = 0$
WHILE *the stopping criterion is not satisfied*
$\boldsymbol{y}^{k+1} \in$ *argmin* $\left\{\mathscr{L}(\boldsymbol{y}, \lambda^k, \gamma^k) \text{ or } \mathscr{L}(\boldsymbol{y}, \lambda^k, \omega^k) : \boldsymbol{y} \in \mathbb{R}^n\right\}$
UPDATE *the Lagrange multipliers*
$\lambda^{k+1} = \lambda^k + \gamma^k h(\boldsymbol{y}^{k+1})$ *(for function 27)*
OR
$\lambda^{k+1} = \lambda^k + \omega^k h(\boldsymbol{y}^{k+1})$ *(for function 28)*
UPDATE *the penalty parameter*
IF $\left\|h(\boldsymbol{y}^{k+1})\right\| \geq \delta \left\|h(\boldsymbol{y}^k)\right\|$
DO $r^{k+1} < r^k$
ELSE
DO $r^{k+1} = r^k$
END
$\gamma^{k+1} = \dfrac{\lambda^2}{r^{k+1}}$, *(for function 27)*
OR
$\omega^{k+1} = \dfrac{|\lambda|}{r^{k+1}}$, *(for function 28)*
$k = k + 1$
END

In Algorithm 4.1 the function $\mathscr{L}$ refers to augmented Lagrangian functions (27) or (28). Details of the selection and updating of the parameters are given in Section 6. To compare the efficiency of the proposed methods, the classical method of Hestenes (1969) and Powell (1969) is also implemented.

Algorithm 4.2.
Data: $\boldsymbol{y}^0 \in \mathbb{R}^n$, $\delta \in (0,1)$, $\lambda^0 \in \mathbb{R}$, $\rho^0 \in \mathbb{R}_{++}$
$k = 0$
WHILE *the stopping criterion is not satisfied*
$\boldsymbol{y}^{k+1} \in$ *argmin* $\left\{f(\boldsymbol{y}) + \lambda^k h(\boldsymbol{y}) + \dfrac{\rho^k}{2} h(\boldsymbol{y})^2 : \boldsymbol{y} \in \mathbb{R}^n\right\}$
$\lambda^{k+1} = \lambda^k + \rho^k h(\boldsymbol{y}^{k+1})$,
IF $\left\|h(\boldsymbol{y}^{k+1})\right\| \geq \delta \left\|h(\boldsymbol{y}^k)\right\|$
DO $\rho^{k+1} > \rho^k$
ELSE
DO $\rho^{k+1} = \rho^k$
END
$k = k + 1$

END

In the classical Algorithm 4.2, Hestenes (1969) and Powell (1969) updated the Lagrange multiplier λ forcing it to satisfy the KKT conditions. In iteration k, parameters λ^k, ρ^k are known, and the algorithm determines $\mathbf{y}^{k+1}$. Therefore, if $\lambda^{k+1} = \lambda^k + \rho^k h(\mathbf{y}^{k+1})$ then $\nabla_{\mathbf{y}} \mathscr{L}(\mathbf{y}^{k+1}, \lambda^{k+1}, \rho^k) = \nabla_{\mathbf{y}} \mathscr{L}(\mathbf{y}^{k+1}, \lambda^{k+1}) = 0$. This is used herein to justify the choice of λ^{k+1} in Algorithm 4.1. Only the case of function (27) will be presented, because the process is analogous for function (28). In fact, taking the derivative of (27) with respect to $\mathbf{y}$ and keeping λ^k and γ^k fixed, one obtains

$$\nabla_{\mathbf{y}} \mathscr{L}(\mathbf{y}, \lambda^k, \gamma^k) = \nabla f(\mathbf{y}) + \sum_{i=1}^{m} \left(\lambda^k + \gamma^k h(\mathbf{y}) \right) \nabla h(\mathbf{y}). \tag{31}$$

The gradient evaluated for $\mathbf{y}^{k+1}$ results:

$$\nabla_{\mathbf{y}} \mathscr{L}(\mathbf{y}^{k+1}, \lambda^k, \gamma^k) = \nabla f(\mathbf{y}^{k+1}) + \sum_{i=1}^{m} \left(\lambda^k + \gamma^k h(\mathbf{y}^{k+1}) \right) \nabla h(\mathbf{y}^{k+1}). \tag{32}$$

Therefore, if $\lambda^{k+1} = \lambda^k + \gamma h(\mathbf{y}^{k+1})$, then $\nabla_{\mathbf{y}} \mathscr{L}(\mathbf{y}^{k+1}, \lambda^{k+1}, \gamma^k) = \nabla_{\mathbf{y}} \mathscr{L}(\mathbf{y}^{k+1}, \lambda^{k+1}) = 0$.

In the next section, some classical numerical reliability problems are presented. These problems are used in the sequence to compare the robustness and efficiency of the proposed and classical augmented Lagrangian methods, and of the HLRF, iHLRF and nHLRF algorithms.

5 Numerical problems

This section presents 25 selected problems from the literature, which are used in the sequence to compare the robustness and performance of the new proposed algorithms. With the exception of problems 22 and 23, all random variables involved are statistically independent. For each problem, the probability distributions and moments (mean and standard deviation) of the design variables are presented, as well as the performance function $h(\mathbf{X})$. For problems 1 to 6, all variables follow standard normal distributions, hence only the performance functions are presented.

Problem 1 [Borri and Speranzini (1997)]: $h(\mathbf{X}) = 0.1(X_1 - X_2)^2 - \dfrac{(X_1 + X_2)}{\sqrt{2}} + 2.5$.

Problem 2 [Borri and Speranzini (1997)]: $h(\mathbf{X}) = -0.5(X_1 - X_2)^2 - \dfrac{(X_1 + X_2)}{\sqrt{2}} + 3$.

Problem 3 [Grooteman (2008)]: $h(\mathbf{X}) = 2 - X_2 - 0.1X_1^2 + 0.06X_1^3$.

Problem 4 [Grooteman (2008)]: $h(\mathbf{X}) = 3 - X_2 + 256X_1^4$.

Problem 5 [Wang and Grandhi (1999a)]: $h(\mathbf{X}) = 1 + \dfrac{(X_1+X_2)^2}{4} - 4(X_1-X_2)^2$.

Problem 6 [Grooteman (2008)]: $h(\mathbf{X}) = 2 + 0.015\sum_{i=1}^{9} X_i^2 - X_{10}$.

Problem 7 [Santosh, Saraf, Ghosh, and Kushwaha (2006)]: The performance function is $h(\mathbf{X}) = X_1^3 + X_2^3 - 18$, the variables are statistically independent and normally distributed with parameters $\mu_{X_1} = \mu_{X_2} = 10$ and $\sigma_{X_1} = \sigma_{X_2} = 5$.

Problem 8 [Santosh, Saraf, Ghosh, and Kushwaha (2006)]: $h(\mathbf{X}) = X_1^3 + X_2^3 - 18$, the variables are normally distributed with mean values $\mu_{X_1} = 10$ and $\mu_{X_2} = 9.9$ and standard deviations $\sigma_{X_1} = \sigma_{X_2} = 5$.

Problem 9 [Grooteman (2008)]: The relationship between the variables considered in this problem is given by the function $h(\mathbf{X}) = 2.5 - 0.2357(X_1 - X_2) + +0.0046(X_1 + X_2 - 20)^4$, where X_1 and X_2 are normally distributed with mean values 10 and standard deviations 3.

Problem 10 [Santosh, Saraf, Ghosh, and Kushwaha (2006)]: The random variables considered in this problem are normally distributed with mean values $\mu_{X_1} = 10$ and $\mu_{X_2} = 9.9$, and standard deviations $\sigma_{X_1} = \sigma_{X_2} = 5$. The performance function is $h(\mathbf{X}) = X_1^3 + X_2^3 - 67.5$.

Problem 11 [Grooteman (2008)]: In this problem the performance function is given by $h(\mathbf{X}) = X_1X_2 - 146.14$, where X_1 and X_2 are normally distributed with mean values $\mu_{X_1} = 78064.4$ and $\mu_{X_2} = 0.0104$ and standard deviations $\sigma_{X_1} = 11709.7$ and $\sigma_{X_2} = 0.00156$.

Problem 12 [Wang and Grandhi (1996)]: The performance function considered in this problem is $h(\mathbf{X}) = 2.2257 - \dfrac{0.025\sqrt{2}}{27}(X_1+X_2)^3 + 0.2357(X_1-X_2)$, where X_1 and X_2 are normally distributed, with mean values 10 and standard deviations 3.

Problem 13 [Santosh, Saraf, Ghosh, and Kushwaha (2006)]: The performance function is $h(\mathbf{X}) = X_1X_2 - 2000X_3$, where the variables X_1 and X_2 are normally distributed with mean values 0.32 and 1400000, and standard deviations 0.032 and 70000, respectively. X_3 is a lognormal variable with mean 100 and standard deviation 40.

Problem 14 [Haldar and Mahadevan (2000)]: In this problem X_1 and X_2 are lognormal variables with mean values 38 and 54, and standard deviations 3.8 and 2.7, respectively. The performance function is given by $h(\mathbf{X}) = X_1X_2 - 1140$.

Problem 15 [Mahadevan and Pan (2001)]: Linear performance function given by $h(\mathbf{X}) = X_1 + 2X_2 + 3X_3 + X_4 - 5X_5 - 5X_6$, where the variables are lognormal with

mean values $\mu_{X_i} = 120$, for $i = 1,\ldots,4$, $\mu_{X_5} = 50$ and $\mu_{X_6} = 40$, and standard deviations $\sigma_{X_i} = 12$, for $i = 1,\ldots,4$, $\sigma_{X_5} = 15$ and $\sigma_{X_6} = 12$.

Problem 16 [Liu and Kiureghian (1992)]: In this problem the random variables have the same distributions of the variables of problem 15 and the following performance function is considered

$$h(\mathbf{X}) = X_1 + 2X_2 + 2X_3 + X_4 - 5X_5 - 5X_6 + 0.001\sum_{i=1}^{6}\sin(100X_i). \tag{33}$$

Problem 17 [Mahadevan and Pan (2001)]: The performance function is given by

$$\begin{aligned} h(\mathbf{X}) \;=\; & -240758.1777 + 10467.364X_1 + 11410.63X_2 + 3505.3015X_3 - \\ & -246.81X_1^2 - 285.3275X_2^2 - 195.46X_3^2 \end{aligned} \tag{34}$$

where $X_i, i = 1\ldots,4$ are lognormal with mean values $\mu_{X_1} = 21.2$, $\mu_{X_2} = 20$ and $\mu_{X_3} = 9.2$, and standard deviations $\sigma_{X_1} = 0.1$, $\sigma_{X_2} = 0.2$ and $\sigma_{X_3} = 0.1$.

Problem 18 [Santosh, Saraf, Ghosh, and Kushwaha (2006)]: The performance function is $h(\mathbf{X}) = X_1X_2 - 78.12X_3$, where X_1 and X_2 are normal variables and X_3 follows a Type-I extreme value distribution. For this problem, the mean values of variables are $\mu_{X_1} = 2\times10^7$, $\mu_{X_2} = 10^{-4}$ and $\mu_{X_3} = 4$, and the standart deviations are $\sigma_{X_1} = 0.5\times10^7$, $\sigma_{X_2} = 0.2\times10^{-4}$ and $\sigma_{X_3} = 1$.

Problem 19 [Santosh, Saraf, Ghosh, and Kushwaha (2006)]: The performance function considered in this problem is the same of problem 18, but now X_1 and X_2 follow the lognormal distribution and X_3 follows a Type-I extreme value distribution, with the same moments as the variables of problem 18.

Problem 20 [Liu and Kiureghian (1992)]: The performance function considered in this problem is

$$\begin{aligned} h(\mathbf{X}) \;=\; & 1,1 - 0.00115X_1X_2 + 0.00117X_1^2 + 0.00157X_2^2 + \\ & +0.0135X_2X_3 - 0.0705X_2 - 0.00534X_1 - 0.0149X_1X_3 - \\ & -0.0611X_2X_4 + 0.0717X_1X_4 - 0.226X_3 + 0.0333X_3^2 - \\ & -0.558X_3X_4 + 0.998X_4 - 1.339X_4^2 \end{aligned} \tag{35}$$

where X_1 follows Type-II extreme value distribution with mean $\mu_{X_1} = 10$ and standard deviation $\sigma_{X_1} = 5$; X_2 and X_3 are normal random variables with mean values $\mu_{X_2} = 25$ and $\mu_{X_3} = 0.8$, and standard deviations $\sigma_{X_2} = 5$ and $\sigma_{X_3} = 0.2$; and X_4 is lognormal variable with mean $\mu_{X_4} = 0.0625$ and standard deviation $\sigma_{X_4} = 0.0625$.

Problem 21 [Wang and Grandhi (1999b)]: The performance function considered in this problem is $h(\mathbf{X}) = X_1^4 + 2X_2^4 - 20$, where the variables are normally distributed with mean values $\mu_{X_1} = \mu_{X_2} = 10$ and standard deviations $\sigma_{X_2} = \sigma_{X_2} = 5$.

Problem 22 Haldar and Mahadevan (2000): The performance function and the probability distributions are the same considered in problem 14, but now X_1 and X_2 are dependent random variables with correlation equal to 0.3.

Problem 23 [Liu and Kiureghian (1992)]: The reliability of the three-bay, five-story, linear elastic frame structure in Fig. 1 is examined. Taken from Liu and Kiureghian (1986), this problem has 21 basic random variables: 3 applied loads, 2 Young's moduli, 8 moments of inertia, and 8 cross-sectional areas. It is assumed that the structure fails if the horizontal displacement at node 1 exceeds 0.2 ft. Thus, the limit state function is expressed as $h(\mathbf{X}) = 0.2 - u_1(\mathbf{X})$. Informations about the probability distribution and the correlation matrix of variables can be obtained in Liu and Kiureghian (1986).

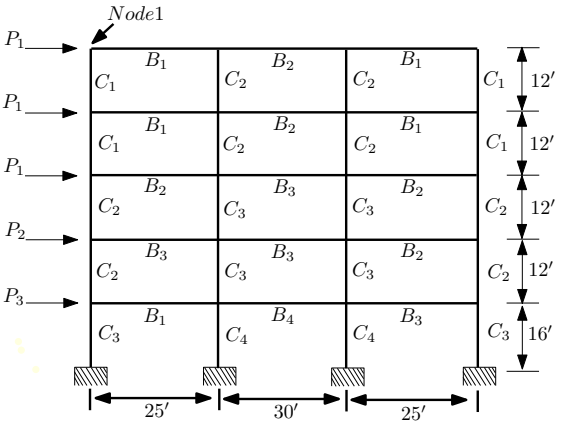

Figure 1: Frame structure of problem 23

Problem 24 [Torii and Lopez (2011)]: This problem refers to the analysis of the pipeline network represented in Fig. 2. All pipes have a length of 50m, except pipe 12 that measures $100m$. Node 1 has a prescribed head of 15m, while all other nodes have a demand of $0.005m^3/s$. Besides, all pipe diameters are equal to $0.11m$, the material relative roughness is $ks = 0.046mm$ and the fluid is water at 15°C. It is assumed that the boundary conditions and the materials relative roughness are Gaussian random variables. For the prescribed head at node 1, a standard deviation equal to 3m is assumed. For the demands at the nodes, standard deviations are equal to $0.001m^3/s$. Standard deviation of material relative roughness is 0.0092mm. This example is described in more details by Torii and Lopez (2011). The limit state function is given by $h(\mathbf{X}) = 5 - h_{17}(\mathbf{X}) \leq 0$, where h_{17} is the nodal head at node

17. In this case, the system is assumed to fail when the nodal head at node 17 is smaller than 5m.

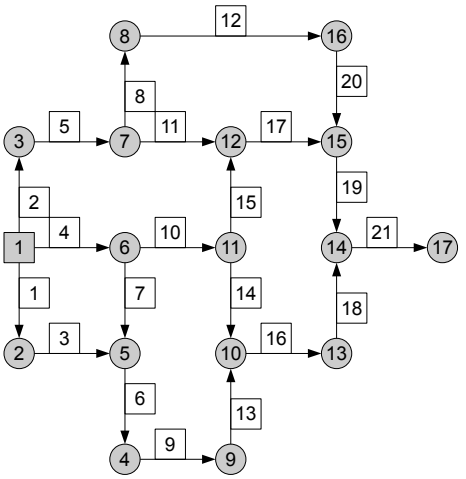

Figure 2: Pipe network of problem 24

Problem 25 [Torii and Lopez (2011)]: This problem refers to the analysis of the water distribution network represented in Fig. 3. In this case, all pipes have a length of 50m. Node 10 has a prescribed head of $20m$, while all other nodes have a demand of $0.005m^3/s$. Besides, all pipe diameters are equal to $0.20m$, the material relative roughness is $ks = 0.002mm$ and the fluid is water at 15°C. This problem was modeled as discussed by Torii and Lopez (2011). It is assumed that the boundary conditions and the material relative roughness are Gaussian random variables. For prescribed head at node 10, a standard deviation equal to $1m$ is assumed. For the demands at the nodes, standard deviations are equal to $0.00125m^3/s$. Standard deviation of material relative roughness is $0.0005mm$. The limit state function is given by $h(\mathbf{X}) = 12 - \min g_i(\mathbf{X}) \leq 0$, where $\min g_i$ is the minimum nodal head observed in the network. In this case, the system is assumed to fail when any nodal head is smaller than 12m.

6 Tested algorithms

For the purpose of comparing robustness and efficiency of the three proposed algorithms to the existing classical augmented Lagrangian and HLRF and iHLRF algorithms, the six algorithms were implemented in Matlab 7.8 version R2009a. All problems were solved using and Intel(R) Core(TM)2 Duo CPU T5870 2GHZ

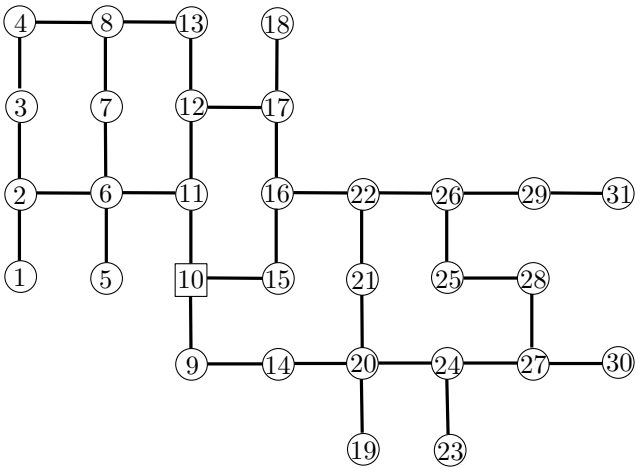

Figure 3: Pipe network of problem 25 with its nodes numbered

processor with 3GB of RAM memory. Some criteria used in the present implementation are presented in this section.

The parameters of the nHLRF algorithm are as follows: $\eta = 10$, $c^0 = 100$, $\alpha^0 = 1$, $m_1 = 0.1$ and $m_2 = 0.9$. The same values were used for η, α^0 and m_1, in the iHLRF algorithm.

For all runs, the maximum number of iterations was set to 100 and the maximum computational time was set to one hour. Moreover, the algorithms are stopped when:

$$\left|h(\mathbf{y}^k)\right| < \varepsilon \;\text{ and }\; 1 - \left(\frac{\nabla h(\mathbf{y}^k)^T \mathbf{y}^k}{\|h(\mathbf{y}^k)\| \, \|\mathbf{y}^k\|} \right) < \varepsilon$$

where $\varepsilon = 10^{-4}$.

The initial point for all algorithms and all problems was set as the mean point of the original variables. However, none of the six algorithms were able to solve problems 5, 24 and 25 when started at the means.

In the case of problem 5, the algorithms HLRF, iHLRF and nHLRF runned into numerical errors to calculate the descent direction d^k, given by Eq. (9), because $\nabla h(\mathbf{y}^0) = 0$. For augmented Lagrangian methods, the errors occured because functions (26), (27) and (28) became constant at $\mathbf{y}^0$, hence the problem could not be solved.

In the case of problems 24 and 25, a division by zero occured in evaluation of the gradient of performance function, which is obtained by finite differences. Hence, the initial points were considered as (in the reduced space): $\mathbf{y} = (0,1)^T$, for problem

5, and $\mathbf{y} = (0.1, \ldots, 0.1)^T$, for problems 24 and 25.

The three variants of the augmented Lagrangian method considered herein are referenced as follows, with regards to the penalty functions considered. The augmented Lagrangian method (Eq. 27) using the penalty function proposed by Matioli and Gonzaga (2008) is called herein LAPM (M for Matioli). The method (Eq. 28) resulting from use of the penalty function of Tseng and Bertsekas (1993) is called LAPB (B for Bertsekas). Finally, the augmented Lagrangian method using the classical (Eq. 4.2) penalty function of Hestenes (1969) and Powell (1969) is called LAPC (C for classic) herein.

In augmented Lagrangian methods, the procedure used to update penalty parameters is based on a measure of infeasibility of the problem, since there is no need to penalize in every iteration. Thus, if the measure of infeasibility defined as $\left\|h(x^{k+1})\right\| \geq \delta \left\|h(x^k)\right\|$ with $\delta \in (0,1)$, is not satisfied, it is because there was no significant gain in viability and hence the penalty parameter is increased. Therefore, in Algorithm 4.2 the penalty parameter is updated by $\rho^{k+1} = 2\rho^k$. For Algorithm 4.1, the penalty parameters γ^{k+1}, given by Eq. (29), and ω^{k+1}, given by Eq. (30), are updated by: $r^{k+1} = 0.01r^k$.

This procedure of penalty parameter updating yielded good results for most problems tested. However, the procedure or alternative options were not thoughtfully tested. Hence, a future specific study on the alternatives for penalty parameter updating could increase efficiency of the proposed methods significantly.

Other parameters used in Algorithms 4.2 and 4.1 are: $\lambda^0 = 1$, $r^0 = 1$, $\rho^0 = 1$ and $\delta = 0.1$.

It is important to note that several methods can be used to solve the unconstrained subproblems resulting from the augmented Lagrangian functions (Eqs. 26, 27 and 28). However, the main goal of the following numerical analysis is to verify applicability, robustness and efficiency of augmented Lagrangian methods in the solution of structural reliability problems, considering the same criteria. Thus, no attention was dedicated to find the most suitable algorithm to solve the unconstrained subproblems. Hence, an internal Matlab routine, called *fminunc* and which finds the minimum of a scalar function of several variables starting at an initial estimate, was used to minimize the unconstrained problems. If the analytical gradient is provided by the user, then the routine uses the BFGS Quasi-Newton method with a mixed quadratic and cubic line search procedure (it was the condition considered in this article), otherwise, it is used the Conjugate Gradient method. In *fminunc* routine, the maximum number of iterations and function evaluations was set to 1000, furthermore, the termination tolerance on the function value and termination tolerance on $\mathbf{y}^k$, were considered as $1e-15$. In augmented Lagragian methods, the

maximum number of outer iterations of the algorithm was set to 100.

7 Numerical results

For each problem, Tab. 1 shows results for the HLRF algorithm in the first line, for the iHLRF in the second line and for the nHLRF algorithm in the third line. Four measures are selected for comparison: *iter* is the number of iterations (direction search), *fun* is the total number of performance function evaluations, *grad* is the number of performance function gradient evaluations and *time* is the computational time spent executing the algorithm. In each case, obtained reliability indexes (β) are also presented.

Table 1: Comparison of optimization algorithms HLRF, iHLRF and nHLRF

Problem	*Method*	*iter*	*fun*	*grad*	*time (seconds)*	β
1	HLRF	1	2	2	0.1074	2.5000
	iHLRF	1	3	2	0.0213	2.5000
	nHLRF	1	3	3	0.0353	2.5000
2	HLRF	1	2	2	0.0003	3.0000
	iHLRF	1	3	2	0.0054	3.0000
	nHLRF	1	3	3	0.0093	3.0000
3	HLRF	1	2	2	0.0002	2.0000
	iHLRF	1	3	2	0.0035	2.0000
	nHLRF	1	3	3	0.0106	2.0000
4	HLRF	1	2	2	0.0001	3.0000
	iHLRF	1	3	2	0.0032	3.0000
	nHLRF	1	3	3	0.0071	3.0000
5	HLRF	4	5	5	0.0007	0.3536
	iHLRF	4	9	5	0.0045	0.3536
	nHLRF	4	9	9	0.0104	0.3536
6	HLRF	1	2	2	0.0001	2.0000
	iHLRF	1	3	2	0.0042	2.0000
	nHLRF	1	3	3	0.0084	2.0000
7	HLRF	7	8	8	0.0008	2.2401
	iHLRF	7	15	8	0.0067	2.2401
	nHLRF	7	15	15	0.0169	2.2401
8	HLRF	100	100	100	0.0119	1.1656
	iHLRF	32	133	33	0.0366	2.2260

Continued on next page

Table 1 – continued from previous page

Problem	*Method*	*iter*	*fun*	*grad*	*time (seconds)*	β
	nHLRF	32	133	133	0.0459	2.2260
9	HLRF	1	2	2	0.0003	2.5000
	iHLRF	1	3	2	0.0028	2.5000
	nHLRF	1	3	3	0.0093	2.5000
10	HLRF	100	100	100	0.0128	1.1466
	iHLRF	47	189	48	0.0554	1.9003
	nHLRF	37	138	138	0.0488	1.9003
11	HLRF	5	6	6	0.0007	5.4280
	iHLRF	5	11	6	0.0051	5.4280
	nHLRF	5	11	11	0.0093	5.4280
12	HLRF	1	2	2	0.0002	2.2257
	iHLRF	1	3	2	0.0043	2.2257
	nHLRF	1	3	3	0.0143	2.2257
13	HLRF	5	6	6	0.0240	2.1911
	iHLRF	5	11	6	0.0064	2.1911
	nHLRF	5	11	11	0.0115	2.1911
14	HLRF	3	4	4	0.0494	5.2127
	iHLRF	3	7	4	0.0057	5.2127
	nHLRF	3	7	7	0.0159	5.2127
15	HLRF	13	14	14	0.0261	3.0483
	iHLRF	15	35	16	0.0286	3.0483
	nHLRF	15	82	82	0.0194	3.0483
16	HLRF	100	100	100	0.0487	2.3481
	iHLRF	100	97175	101	28.560	2.3480
	nHLRF	64	2022	2022	0.2063	2.3482
17	HLRF	11	12	12	0.0228	0.8292
	iHLRF	34	140	35	0.0403	0.8292
	nHLRF	14	34	34	0.0302	0.8292
18	HLRF	9	10	10	0.0102	3.3221
	iHLRF	12	30	13	0.0212	3.3221
	nHLRF	12	31	31	0.0157	3.3221
19	HLRF	6	7	7	0.0159	4.4282
	iHLRF	6	13	7	0.0124	4.4282
	nHLRF	6	13	13	0.0171	4.4282
20	HLRF	100	100	100	0.0388	1.1643
	iHLRF	10	30	11	0.0120	1.3651

Continued on next page

Table 1 – continued from previous page

Problem	*Method*	*iter*	*fun*	*grad*	*time (seconds)*	β
	nHLRF	13	42	42	0.0210	1.3653
21	HLRF	100	100	100	0.0111	0.9863
	iHLRF	39	196	40	0.0366	2.3655
	nHLRF	56	276	276	0.0737	2.3655
22	HLRF	4	5	5	0.0194	4.5297
	iHLRF	4	9	5	0.0595	4.5297
	nHLRF	4	9	9	0.0534	4.5297
23	HLRF	3	4	4	0.7738	2.6666
	iHLRF	3	7	4	0.4114	2.6666
	nHLRF	4	12	12	0.5347	2.6666
24	HLRF	not converged				
	iHLRF	not converged				
	nHLRF	not converged				
25	HLRF	not converged				
	iHLRF	not converged				
	nHLRF	not converged				

Following Tab. 1, it is observed that the algorithms HLRF, iHLRF and nHLRF successfully solved 72%, 88% and 92% of the problems. Thus, the nHLRF algorithm solved a greater number of problems, and may therefore be considered more robust than the HLRF and iHLRF algorithms. Results show that none of the three algorithms (HLRF, iHLRF and nHLRF) were able to solve problems 24 and 25, because during the iterative process $\nabla h(\mathbf{y}^k) = 0$, causing errors in the calculation of search direction d^k. Note also that the HLRF algorithm reached the maximum number of iterations ($k = 100$) when solving problems 8, 10, 16, 20 and 21, and this also occurred with iHLRF for problem 16. Although the algorithms HLRF and iHLRF have not satisfied the stopping criterion for problem 16, reliability indexes obtained are very close to the correct solution.

In order to simplify analysis of the results, smaller tables with specific information were built. Tab. 2, Tab. 3 and Tab. 4 refer to the number of times that each algorithm, HLRF, iHLRF and nHLRF, performed better than the others, taking into account the number of iterations (*iter*), the number of performance function evaluations (*fun*), the number of gradient evaluations (*grad*) and the computational time (*time*). The values related to "*tie*" indicate the amount of problems for which the numbers coincided. For the construction of Tab. 2, results of 22 problems that were solved by at least one of the two algorithms, HLRF or iHLRF, were considered. In this case, problems 16, 24 and 25 were not considered.

Table 2: Comparison between HLRF and iHLRF algorithms

Method	*iter*	*fun*	*grad*	*time*
HLRF	3	18	3	13
iHLRF	4	4	4	9
tie	15	0	15	0

In the construction of Tab. 3 and Tab. 4, problems 24 and 25 were not considered, because it was not solved by any of the three algorithms (HLRF, iHLRF and nHLRF).

Table 3: Comparison between HLRF and nHLRF algorithms

Method	*iter*	*fun*	*grad*	*time*
HLRF	4	18	18	13
nHLRF	5	5	5	10
tie	14	0	0	0

Results presented in Tab. 2 and Tab. 3 show that, when it converges, HLRF is better than algorithms iHLRF and nHLRF in terms of performance (least number of performance function and gradient evaluations, and least computational time). In fact, since iHLRF and nHLRF perform a linear search to compute the step length, α^k, these algorithms naturally require more performance function and gradient evaluations. The lower performance, however, is compensated by the better robustness of iHLRF and nHLRF algorithms.

Algorithms iHLRF and nHLRF are more robust, because they resolved 22 and 23 problems, respectively, whereas the HLRF algorithm only solved 18 of the tested problems. This result reinforces the observation that the HLRF algorithm is efficient when it converges, but there is no guarantee of convergence. The nHLRF algorithm presented herein, on the other hand, is garanteed to convergence.

Table 4: Comparison between iHLRF and nHLRF algorithms

Method	*iter*	*fun*	*grad*	*time*
iHLRF	3	5	21	17
nHLRF	3	3	2	6
tie	17	15	0	0

Results presented in Tab. 4 show that the algorithms iHLRF and nHLRF have equivalent performance related to the number of iterations. However, the iHLRF

algorithm performs better with respect to *fun*, *grad* and *time*. This happens because in iHLRF algorithm, α^k is obtained by the Armijo rule, whereas in the nHLRF algorithm, the Wolfe conditions are used. The Wolfe conditions require the derivative of the function to be minimized, hence the higher number of function h and gradient ∇h computations. The benefit for the lesser efficiency is the guarantee of convergence of the nHLRF algorithm. While the iHLRF and HLRF algorithms reached the maximum number of iterations in solution of problems 16 and 8, 10, 16, 20 and 21, respectively, the nHLRF algorithm was able to solve these problems within the maximum number of iterations.

Results obtained for the augmented Lagrangian methods are presented in Tab. 5. For each problem, results for LAPC are shown in the fisrt line, LAPM in the second line and LAPB in the third line. For the augmented Lagrangian methods, *it.ext* represents the number of outer iterations of the algorithm, *it.int* is the number of iterations of the internal algorithm, *fun* is the total number of performance function evaluations, *grad* is the number of performance function gradient evaluations and β is reliability index obtained in each case. It is important to note that the values of *fun* and *grad* are the same, for this reason are listed in a single column.

Table 5: Comparison of optimization algorithms HLRF, iHLRF and nHLRF

Problem	*Method*	*it.ext*	*it.int*	*fun/grad*	*time (seconds)*	β
1	LAPC	5	9	18	0.6023	2.5000
	LAPM	2	4	6	0.1686	2.5000
	LAPB	2	4	6	0.0698	2.5000
2	LAPC	5	9	18	0.2371	3.0000
	LAPM	2	4	6	0.0506	3.0000
	LAPB	2	4	6	0.0482	3.0000
3	LAPC	5	8	17	0.1386	2.0000
	LAPM	2	3	6	0.0471	2.0000
	LAPB	2	3	6	0.0475	2.0000
4	LAPC	5	9	17	0.1094	3.0000
	LAPM	2	4	6	0.0473	3.0000
	LAPB	2	4	6	0.0504	3.0000
5	LAPC	1	13	15	0.0299	0.3536
	LAPM	1	13	15	0.0439	0.3536
	LAPB	1	13	15	0.0443	0.3536
6	LAPC	5	8	17	0.0708	2.0000
	LAPM	2	3	6	0.0488	2.0000

Continued on next page

Table 5 – continued from previous page

Problem	*Method*	*it.ext*	*it.int*	*fun/grad*	*time (seconds)*	β
	LAPB	2	3	6	0.0499	2.0000
7	LAPC	2	20	25	0.0425	2.2401
	LAPM	4	29	40	0.0924	2.2401
	LAPB	3	22	31	0.0752	2.2401
8	LAPC	2	46	52	0.0535	2.2260
	LAPM	4	52	66	0.1070	2.2260
	LAPB	3	43	55	0.0939	2.2260
9	LAPC	5	9	18	0.0705	2.5000
	LAPM	2	4	6	0.0461	2.5000
	LAPB	2	4	6	0.0527	2.5000
10	LAPC	2	37	46	0.0518	1.9003
	LAPM	4	45	59	0.1099	1.9003
	LAPB	3	44	55	0.0876	1.9003
11	LAPC	1	68	90	0.0626	5.3333
	LAPM	1	68	90	0.0820	5.3333
	LAPB	1	68	90	0.0950	5.3333
12	LAPC	5	9	18	0.0718	2.2257
	LAPM	2	3	6	0.0454	2.2257
	LAPB	2	3	6	0.0502	2.2257
13	LAPC	1	64	75	0.1975	2.2046
	LAPM	6	132	178	0.2423	2.1911
	LAPB	3	137	177	0.1712	2.1911
14	LAPC	4	71	95	0.2296	5.2127
	LAPM	4	64	75	0.1724	5.2127
	LAPB	4	62	87	0.1402	5.2127
15	LAPC	9	78	362	0.7052	3.0486
	LAPM	6	63	274	0.3319	3.0483
	LAPB	8	48	344	0.3170	3.0494
16	LAPC	4	72	160	0.2266	2.3484
	LAPM	5	69	171	0.2346	2.3486
	LAPB	4	88	190	0.2252	2.3483
17	LAPC	2	53	71	0.0787	0.8292
	LAPM	2	38	57	0.1766	0.8292
	LAPB	3	58	80	0.1261	0.8292
18	LAPC	1	88	107	0.1377	3.3225
	LAPM	4	120	159	0.2193	3.3221

Continued on next page

Table 5 – continued from previous page

Problem	*Method*	*it.ext*	*it.int*	*fun/grad*	*time (seconds)*	β
	LAPB	3	123	155	0.1774	3.3221
19	LAPC	5	309	407	0.4854	4.4129
	LAPM	5	194	241	0.3088	4.4169
	LAPB	4	219	272	0.3718	4.4103
20	LAPC	13	146	165	0.4636	1.3656
	LAPM	2	18	21	0.1173	1.7198
	LAPB	3	57	61	0.1737	1.3704
21	LAPC	2	64	75	0.0654	2.3655
	LAPM	4	68	87	0.1833	2.3655
	LAPB	3	70	84	0.1362	2.3655
22	LAPC	3	51	67	0.2757	4.5297
	LAPM	5	60	84	0.2082	4.5297
	LAPB	3	39	55	0.6174	4.5297
23	LAPC	11	88	106	7.8007	2.6670
	LAPM	2	14	16	4.9509	2.6670
	LAPB	3	63	73	6.7781	2.6676
24	LAPC	34	179	344	163.2402	1.8103
	LAPM	3	34	47	24.9473	1.8106
	LAPB	5	34	60	30.5753	1.8106
25	LAPC	25	262	626	535.6507	2.2016
	LAPM	3	35	83	77.4204	2.2004
	LAPB	3	39	85	83.3869	2.2004

Based on the results shown in Tab. 5, one observes that the methods LAPC, LAPM and LAPB successfully solved 100% of the problems. So, the augmented Lagrangian methods may therefore be considered more robust than HLRF, iHLRF and nHLRF algorithms which solved a smaller number os problems, as seen in Tab. 1.

Performance of the methods LAPC, LAPB and LAPM are compared in Tab. 6, Tab. 7 and Tab. 8. The comparison takes into acount the number of outer iterations of the algorithm (*it.ext*), the number of iterations of the inner algorithm (*it.int*), the number of performance function evaluations (*fun*), the number of gradient evaluations of performance function (*grad*) and the computational time (*time*). In this analysis, 25 problems were considered, because they were resolved by each one of the methods.

Following Tab. 6, it can be observed that the LAPM method showed better performance in solving structural reliability problems than the LAPC method, with

Table 6: Comparison between LAPC and LAPM algorithms

Method	*it.ext*	*it.int*	*fun/grad*	*time*
LAPC	8	7	8	10
LAPM	12	16	15	15
tie	5	2	2	0

respect to all measures of comparison considered *it.ext*, *it.int*, *fun*, *grad* and *time*. The same occurs with the LAPB method, as can be seen in Tab. 7.

Table 7: Comparison between LAPC and LAPB algorithms

Method	*it.ext*	*it.int*	*fun/grad*	*time*
LAPC	7	7	8	9
LAPB	13	16	15	16
tie	5	2	2	0

Hence, results presented herein show that the proposed LAPM and LAPB augmented Lagrangian methods are not only more robust, but also more efficient than the "*classical*" LAPC method. It should be noted that the bad performance atributed by Liu and Kiureghian (1992) to augmented Lagrangian methods refers to the classical method: hence, there is significant room for consideration of augmented Lagrangian methods in solution of structural reliability problems.

Table 8: Comparison between LAPM and LAPB algorithms

Method	*it.ext*	*it.int*	*fun/grad*	*time*
LAPM	5	9	9	13
LAPB	9	6	7	12
tie	11	10	9	0

A comparison between LAPM and LAPB methods shows that the LAPB method performs better than LAPM with respect to the number of outer iterations of the algorithm (*it.ext*), see Tab. 8. However, the greatest computational effort of augmented Lagrangian methods is associated with the solution to unconstrained subproblems (minimizing the augmented Lagrangian function), as in the external algorithm, only the updating of penalty parameter and of the Lagrange multiplier is performed. The LAPM method was found to be more efficient than LAPB with respect to *it.int*, *fun* and *grad*, and for computational time the efficiency is similar.

In practical problems of structural reliability, there is a concern about the number of evaluations of performance functions and derivatives, because in most cases, the performance function is implicit (e.g., the solution to an expensive finite element model). Thus, Tab. 9, Tab. 10 and Tab. 11 refer to the number of times that each algorithm performed better, taking into account the total number of calls to limit state equation (*fun*) and the calculation of gradient (*grad*).

Table 9: Comparison between LAPC and HLRF, iHLRF, nHLRF algorithms

Method	*fun*	*grad*	*Method*	*fun*	*grad*	*Method*	*fun*	*grad*
LAPC	6	7	LAPC	7	4	LAPC	7	6
HLRF	19	18	iHLRF	18	21	nHLRF	18	19
tie	0	0	tie	0	0	tie	0	0

Table 10: Comparison between LAPM and HLRF, iHLRF, nHLRF algorithms

Method	*fun*	*grad*	*Method*	*fun*	*grad*	*Method*	*fun*	*grad*
LAPM	6	7	LAPM	7	3	LAPM	7	6
HLRF	19	18	iHLRF	18	22	nHLRF	18	19
tie	0	0	tie	0	0	tie	0	0

Table 11: Comparison between LAPB and HLRF, iHLRF, nHLRF algorithms

Method	*fun*	*grad*	*Method*	*fun*	*grad*	*Method*	*fun*	*grad*
LAPB	6	7	LAPB	7	3	LAPB	6	6
HLRF	19	18	iHLRF	18	22	nHLRF	19	19
tie	0	0	tie	0	0	tie	0	0

Comparing HLRF-based and augmented Lagrangian algorithms, it is noted that HLRF-based algorithms are more efficient, particulary with respect to number total of performance function and derivative evaluations, as seen in Tab. 9, Tab. 10, Tab. 11. This is justified by the generality of augmented Lagrangian methods, which can be applied to optimization problems with equality and even inequality constraints, as discussed by Matioli and Gonzaga (2008) and Bertsekas (1997), while algorithms HLRF, iHLRF and nHLRF were developed specifically for structural reliability problems.

It is also important to note that augmented Lagrangian methods presented better results in solution of problems with larger number of variables, as is the case for

problems 16, 24 and 25. By the way, this fact can be considered an important advantage of the augmented Lagrangian methods, since most practical problems involves a large number of random variables. Regarding the number of problems solved, the algorithms LAPC, LAPM and LAPB were more robust, as they solved 100% of the problems, followed by nHLRF algorithm, which solved 92% of the problems, and by the algorithms iHLRF and HLRF, that were only able to solve only 88% and 75% of the problems, respectively.

8 Concluding remarks

In this article, three new optimization algorithms were presented, for application to structural reliability problems.

The so-called nHLRF (n for new) algorithm uses the HLRF search direction, makes a linear search in this direction based on the Wolfe conditions, in order to minimize a new (proposed) differentiable merit function to obtain the step length. It was shown that, under certain assumptions the algorithm generates a sequence that converges to a local minimizer of the problem. These assumptions are converted in rules for the updating of nHLRF parameters.

Two new augmented Lagrangian methods, applied to resolution of problems with equality constraints, where also introduced. These methods are extensions of results presented by Matioli and Gonzaga (2008) and by Bertsekas (1997) for problems with inequality constraints. The modern quadratic penalty parameters introduced via these methods are more robust and more efficient than "*classical*" penalty parameters considered in the comparative study of Liu and Kiureghian (1992). Hence, this makes results of Ref. Liu and Kiureghian (1992) out-dated with respect to the comparative performance of augmented Lagrangian methods in solution of structural reliability problems. Theoretical studies on the convergence of the proposed LAPM and LAPB methods are presented by Santos and Matioli (2011).

Numerical results obtained herein, in aplication to 25 reference problems from the literature, indicate that the new proposed methods are competitive and promising. The nHLRF algorithm presented a similar performance to the iHLRF algorithm, related to the number of iterations and performance function evaluations. However, the nHLRF solved a larger number of problems, compared do the HLRF and iHLRF algorithms.

The proposed augmented Lagrangian methods (LAPM and LAPB) were shown to be more efficient than the augmented Lagrangian method with "*classical*" penalty (LAPC), as they solved the larger number of problems, with a smaller number of iterations, performance function evaluations and computational time. The two pro-

posed methods (LAPB and LAPM) showed similar performance for the evaluated criteria.

Comparing the HLRF-based and augmented Lagrangian methods, it was noted that HLRF-based algorithms are more efficient than the augmented Lagrangian methods, particularly with respect to the number of performance function evaluations and gradient evaluations. However, augmented Lagrangian methods have the advantage of being more general [Santos and Matioli (2011); Matioli and Gonzaga (2008); Bertsekas (1997)] than HLRF algorithms, which were developed specifically for structural reliability problems. Moreover, augmented Lagrangian methods are probably more suitable for problems involving large number of random variables.

No definitive conclusions with respect to robustness and performance of the proposed methods and algorithms can be made, as the numerical results presented herein are based on a limited set of structural reliability problems.

As future research, a specific study of different techniques for updating penalty parametes in the proposed augmented Lagrangian methods is recomended. Also, a study on more suitable algorithms to solve the unconstrained subproblems generated in the augmented Lagrangian approach is recomended as, in our opinion, the efficiency of the proposed augmented Lagrangian methods could be significantly improved.

Acknowledgement: We thank André Jacomel Torri for his valuable contribution which very much improved this article.

References

Ben-Tal, A.; Zibulevsky, M. (1997): Penalty-barrier multiplier methods for convex programming problems. *SIAM Journal on Optimization*, vol. 7, pp. 347–366.

Bertsekas, D. P. (1997): Multiplier methods: a survey. *Automatica*, vol. 12, pp. 133–145.

Birgin, E. G.; Castillo, R.; Martínez, J. M. (2005): Numerical comparisson of Augmented Lagrangian algorithms for nonconvex problems. *Computational Optimization and Applications*, vol. 31, pp. 31–56.

Borri, A.; Speranzini, E. (1997): Structural reliability analysis using a standard deterministic finite element code. *Structural Safety*, vol. 19, pp. 361–382.

Censor, Y.; Zenios, S. (1992): The proximal minimization algorithm with d-functions. *Journal Optimization Theory and Applications*, vol. 73, pp. 451–464.

Chen, C.; Teboulle, M. (1993): Convergence analysis of a proximal-like minimization algorithm using Bregman function. *SIAM Journal on Optimization*, vol. 3, pp. 538–543.

Dennis, J. E.; Schnabel, R. D. (1996): *Numerical methods for unconstrained optimization and nonlinear equations*. Society for Industrial and Appied Mathematics.

Ditlevsen, O. (1981): Principle of normal tail approximation. *Journal of the Engineering Mechanics Division*, vol. 107(EM6), pp. 1191–1207.

Eckstein, J. (1993): Nonlinear proximal algorithms using Bregman functions with aplications to convex programming. *Mathematics of Operations Research*, vol. 18, pp. 202–226.

Grooteman, F. (2008): Adaptive radial–based importance sampling method for structural reliability. *Structural Safety*, vol. 30, pp. 533–542.

Haldar, A.; Mahadevan, S. (2000): *Probability, reliability and statistical methods in engineering design*. John Wiley & Sons.

Hasofer, A. M.; Lind, N. C. (1974): Exact and invariant second-moment code format. *Journal of the Engineering Mechanics Division*, vol. 100(EM1), pp. 111–121.

Hestenes, M. (1969): Multiplier and gradient methods. *Journal Optimization Theory and Applications*, vol. 4, pp. 303–320.

Iusem, A.; Teboulle, M. (1993): On the convergence rate of entropic proximal optimization methods. *Computational and Applied Mathematics*, vol. 12, pp. 153–168.

Iusem, A.; Teboulle, M. (1995): Convergence rate of nonquadratic proximal point methods for convex and linear programming. *Mathematics of Operations Research*, vol. 20, pp. 657–677.

Iusem, A.; Teboulle, M.; Svaiter, B. (1994): Entropy-like proximal methods in convex programming. *Mathematics of Operations Research*, vol. 19, no. 4, pp. 790–814.

Kiwiel, K. (1997): Proximal minimization methods with generalized Bregman functions. *SIAM Journal on Control and Optimization*, vol. 35, no. 4, pp. 1142–1168.

Liu, P. L.; Kiureghian, A. D. (1986): Optimization algorithms for structural reliability analysis. *Report UCB/SESM 86/09, Department of civil engineering, University of California, Berkeley*.

Liu, P. L.; Kiureghian, A. D. (1992): Optimization algorithms for structural reliability. *Structural Safety*, vol. 9, pp. 161–177.

Mahadevan, S.; Pan, S. (2001): Multiple linearization methods for nonlinear reliability analysis. *Journal of Engineering Mechanics*, vol. 127, no. 11, pp. 1165–1172.

Martinez (2000): Box-quacan and the implementation of augmented Lagrangian algorithms for minimization with inequality constraints. *Computational and Applied Mathematics*, vol. 19, pp. 31–56.

Matioli, L. C.; Gonzaga, C. C. (2008): A new family of penalty for augmented Lagrangian methods. *Numerical linear algebra with applications*, vol. 15, pp. 925–944.

Melchers, R. E. (2001): *Structural reliability analysis and prediction*. John Wiley & Sons.

Powell, M. J. D. (1969): A method for nonlinear constraints in minimizations problems in optimization. *R. Fletcher*, pp. 283–298.

Rackwitz, R.; Fiessler, B. (1978): Structural reliability under combined random load sequences. *Computers & Structure*, vol. 9, pp. 489–494.

Rômulo, A. C.; Gonzaga, C. C. (2003): A nonlinear programming algorithm based on non-coercive penalty functions. *Mathematical Programming*, vol. 96, no. 1, pp. 87–101.

Santos, S.; Matioli, L. C. (2011): Two new augmented Lagrangian algorithms with quadratic penalty for equality problems. *Technical Report, Available from http://people.ufpr.br/ matioli/minhahome/pesquisa.html*.

Santosh, T. V.; Saraf, R. K.; Ghosh, A. K.; Kushwaha, H. S. (2006): Optimum step length selection rule in modified HLRF method for structural reliability. *International Journal of Pressure Vessels and Piping*, vol. 83, pp. 742–748.

Teboulle, M. (1992): Entropic proximal mappings with applications to nonlinear programming. *Mathematics of Operations Research*, vol. 17, pp. 97–116.

Torii, A. J.; Lopez, R. H. (2011): Reliability analysis of water distribution networks using the adaptive response surface approach. *Journal of Hydraulic Engineering, DOI:10.1061/(ASCE)HY.1943-7900.0000504*.

Tseng, P.; Bertsekas, D. (1993): On the convergence of exponential multiplier method for convex programming. *Mathematical Programming*, vol. 60, pp. 1–19.

Wang, L.; Grandhi, R. V. (1996): Safety index calculations using intervening variables for structural reliability. *Computers and Structures*, vol. 59, pp. 1139–1148.

Wang, L.; Grandhi, R. V. (1999): Higher-order failure probability calculation using nonlinear approximations. *Computer methods in applied mechanics and engineering*, vol. 168, pp. 185–206.

Wang, L.; Grandhi, R. V. (1999): Structural reliability analysis and optimization: use of approximations. *NASA/CR, 209154*.

Zhang, Y.; Kiureghian, A. D. (1997): Finite element reliability methods for inelastic structures. *Report UCB/SEMM - 97/05, Department of civil and environmental engineering, University of California, Berkeley*.