Computers and Geotechnics 76 (2016) 33–42

Contents lists available at ScienceDirect

Computers and Geotechnics

journal homepage: www.elsevier.com/locate/compgeo

Research Paper

An efficient quasi-Newton approximation-based SORM to estimate the reliability of geotechnical problems

Peng Zeng [a,1], Rafael Jimenez [b,*], Tianbin Li [a]

[a] State Key Laboratory of Geohazard Prevention and Geoenvironment Protection (Chengdu University of Technology), Chengdu, China
[b] Department of Ground Engineering, Technical University of Madrid, Madrid, Spain

ARTICLE INFO

Article history:
Received 5 November 2015
Received in revised form 5 January 2016
Accepted 6 February 2016
Available online 3 March 2016

Keywords:
Second order reliability method
BFGS
SR1
Rectangular foundation
Shallow footing
Layered soil slope

ABSTRACT

The first order reliability method (FORM) is efficient, but it has limited accuracy; the second order reliability method (SORM) provides greater accuracy, but with additional computational effort. In this study, a new method which integrates two quasi-Newton approximation algorithms is proposed to efficiently estimate the second order reliability of geotechnical problems with reasonable accuracy. In particular, the Hasofer–Lind–Rackwitz–Fiessler–Broyden–Fletcher–Goldfarb–Shanno (HLRF–BFGS) algorithm is applied to identify the design point on the limit state function (LSF), and consequently to compute the first order reliability index; whereas the Symmetric Rank-one (SR1) algorithm is nested within the HLRF–BFGS algorithm to compute good approximations, yet with a reduced computational effort, of the Hessian matrix required to compute second order reliabilities. Three typical geotechnical problems are employed to demonstrate the ability of the suggested procedure, and advantages of the proposed approach with respect to conventional alternatives are discussed. Results show that the proposed method is able to achieve the accuracy of conventional SORM, but with a reduced computational cost that is equal to the computational cost of HLRF–BFGS-based FORM.

1. Introduction

Reliability analyses have been proposed as a rational complement to deterministic geotechnical design [1], as they can more directly quantify the influence of the uncertainty about input parameters and their correlation relationships (see e.g., [2,3]).

Due to its simplicity and efficiency, the first order reliability method (FORM)—that linearly approximates the limit state function (LSF) to estimate the probability of failure—has been widely used in geotechnical reliability analyses (see e.g., [4–7]). However, the linearization that is inherent to FORM introduces errors in many cases (see e.g., [7,8]), and the second order reliability method (SORM)—which extends FORM to consider the curvatures of the LSF, hence providing a better approximation—has also been employed as an alternative.

Brzakała and Puła [9] and Bauer and Puła [10] analyzed the probability of foundation settlements exceeding an allowable threshold using SORM and a polynomial response surface method (RSM)-based SORM, respectively; Cho [11] combined an artificial neural network (ANN)-based RSM and SORM to compute the reliability of slopes; Lü and co-authors [2,12,13] employed various RSMs with SORM to analyze tunnel supports; Chan and Low [14] introduced a practical SORM for foundation reliability analysis using a point-fitted paraboloid method; and Zeng and co-authors [8,15] applied SORM to evaluate the system reliability of tunnels and slopes, respectively. However, these methods are often more computationally expensive than FORM, due to the need to evaluate the curvatures of the LSF or to construct the response surface function. Therefore, an approach that aims to combine the higher accuracy of SORM-based reliability solutions with the lower computational cost of traditional FORM-based approaches is considered as a useful contribution to the geotechnical field.

This paper proposes an attempt in that direction. In particular, our proposed approach uses the recently proposed Hasofer–Lind–Rackwitz–Fiessler–Broyden–Fletcher–Goldfarb–Shanno (HLRF–BGFS) algorithm [16] to locate the FORM design point efficiently; and it integrates such algorithm with the Symmetric Rank-one (SR1) algorithm [17] to approximate the Hessian matrix (i.e., the second order derivative matrix). The goal is that the identified design point can be used, together with the approximated Hessian matrix, to efficiently estimate the second order probability of

* Corresponding author at: ETSI Caminos, C. y P., C/Profesor Aranguren s/n, 28040 Madrid, Spain.
E-mail address: rjimenez@caminos.upm.es (R. Jimenez).
[1] Department of Ground Engineering, Technical University of Madrid, Madrid, Spain

http://dx.doi.org/10.1016/j.compgeo.2016.02.003
0266-352X/

failure. Details of the algorithms are discussed first, and then the performance of the proposed method is tested using three typical geotechnical example cases taken from the literature.

2. Conventional SORM

Conventional SORM estimates the second order probability of failure using (i) the design point and gradient information computed by FORM and (ii) the Hessian matrix computed using finite differences. For completeness, traditional methods to compute FORM solutions and the Hessian matrix, as well as to compute the SORM reliability based on them, are discussed below. Additional details can be found in traditional reliability references such as [18] and [19].

2.1. The first order reliability method (FORM)

The FORM reliability analysis of a LSF defined by $G(\mathbf{x}) = 0$, where $\mathbf{x}$ is a vector of random variables in physical space can be tackled in different ways. One possibility is to solve it in the original space using standard mathematical software [20], although it is probably more common to transform the vector $\mathbf{x}$ into a space of uncorrelated standard normal random variables, $\mathbf{u}$, so that the limit state surface can be rewritten as $g(\mathbf{u}) = 0$. Then, FORM aims to find the point on $g(\mathbf{u}) = 0$ with shortest distance to the origin, as the neighborhood of such "design point" has the greatest contribution to the probability of failure. In other words, the problem is equivalent to solve the following constrained optimization problem [6]:

$$\mathbf{u}^* = \min_{\mathbf{u}} \|\mathbf{u}\| \qquad \text{subject to} \quad g(\mathbf{u}) = 0 \tag{1}$$

where $\|\cdot\|$ is the norm of a vector. Then, the reliability index, β, can be computed as

$$\beta = \|\mathbf{u}^*\| \tag{2}$$

and the probability of failure can be approximated as

$$P_f \approx \Phi(-\beta) \tag{3}$$

Although direct optimization using standard mathematical software is possible (see e.g., the MATLAB *fmincon* function employed in [19] and the spreadsheet method employed in [21]), specific algorithms such as Hasofer–Lind–Rackwitz–Flessler (HLRF)-based algorithms have often been proposed to solve Eq. (1) (see e.g., [22–26]). Among them, the improved HLRF (iHLRF) algorithm [25]—a tradeoff between efficiency and accuracy—is probably the more commonly used in engineering practice (see e.g., [4,6]), and it will also be employed in this study for comparison. Here, we discuss some aspects related to its computational efficiency; additional details of the algorithm can be found in [25]. In the iHLRF algorithm, the gradient vector, $\nabla g(\mathbf{u})$, needed to search the design point, can be approximated using a forward difference scheme, given as

$$\nabla g(\mathbf{u})_i \approx \frac{g(\mathbf{u}_i + \Delta h) - g(\mathbf{u}_i)}{\Delta h} \tag{4}$$

where Δh is the step size, subscript i indicates the ith element of a vector, and $g(\mathbf{u}_i + \Delta h)$ is a notation convention to indicate the evaluation of $g(\cdot)$ at a vector equal to $\mathbf{u}$, except for its ith component, which is equal to $u_i + \Delta h$.

2.2. Computing the Hessian matrix

In addition to the design point and gradient information computed by FORM, an additional effort is required in SORM to compute the Hessian matrix at the design point, $\mathbf{u}^*$. In engineering practice, when analytical solutions are commonly not available, the Hessian matrix, $\mathbf{H}$, may be computed using a forward finite difference scheme given by (see e.g., [27,28])

$$\mathbf{H}(i,i) \approx \frac{g(\mathbf{u}_i^* + 2\Delta h) - 2g(\mathbf{u}_i^* + \Delta h) + g(\mathbf{u}_i^*)}{\Delta h^2} \tag{5}$$

$$\mathbf{H}(i,j) \approx \frac{g(\mathbf{u}_i^* + \Delta h, \mathbf{u}_j^* + \Delta h) - g(\mathbf{u}_i^* + \Delta h, \mathbf{u}_j^*) - g(\mathbf{u}_i^*, \mathbf{u}_j^* + \Delta h) + g(\mathbf{u}_i^*, \mathbf{u}_j^*)}{\Delta h^2} \tag{6}$$

where $g(\mathbf{u}_i^* + \Delta h, \mathbf{u}_j^* + \Delta h)$ is a generalization of the previously explained notation that indicates that both the ith and jth components of $\mathbf{u}^*$ are increased by Δh. Note that $g(\mathbf{u}_i^* + \Delta h)$, $g(\mathbf{u}_i^*)$, $g(\mathbf{u}_i^* + \Delta h, \mathbf{u}_j^*)$, $g(\mathbf{u}_i^*, \mathbf{u}_j^* + \Delta h)$ and $g(\mathbf{u}_i^*, \mathbf{u}_j^*)$ in Eqs. (5) and (6), are already available from the last iteration of the iHLRF algorithm. Therefore, only $g(\mathbf{u}_i^* + 2\Delta h)$ and $g(\mathbf{u}_i^* + \Delta h, \mathbf{u}_j^* + \Delta h)$ are needed to compute the Hessian matrix. Additionally, $\mathbf{H}$ is symmetric, so that $\mathbf{H}(i,j) = \mathbf{H}(j,i)$. Thus, $n(n+1)/2$ new LSF evaluations are theoretically required to compute the Hessian matrix in conventional SORM. This could significantly increase the computational cost, particularly for a large number of random variables and an 'expensive' LSF.

2.3. Estimating the second order probability of failure

Prior to computing the second order probability of failure, random variables in the U-space should be further transformed, to a rotated standard normal V-space, using an orthogonal transformation $\mathbf{V} = \mathbf{PU}$, where $\mathbf{P}$ is an $n \times n$ orthogonal rotation matrix whose nth column is $\boldsymbol{\alpha}$ ($\boldsymbol{\alpha} = \mathbf{u}^*/\|\mathbf{u}^*\|$) and that can be obtained using Gram-Schmidt orthogonalization [29]. After rotating the coordinates, a rotated diagonal Hessian matrix, $\mathbf{H}_{\text{rot}}$, can be obtained as:

$$\mathbf{H}_{\text{rot}} = \mathbf{P} \cdot \frac{\mathbf{H}}{\|\nabla g(\mathbf{u}^*)\|} \cdot \mathbf{P}^T \tag{7}$$

where $\nabla g(\mathbf{u}^*)$ is the gradient vector at the design point (available from the last iteration of the iHLRF algorithm). The principal curvatures, κ_i, of the LSF at the design point are the first $n-1$ diagonal elements of $\mathbf{H}_{\text{rot}}$; i.e.,

$$\kappa_i = [\mathbf{H}_{\text{rot}}]_{ii} \qquad (i = 1, 2, \ldots, n-1) \tag{8}$$

Various formulas have been proposed to evaluate the second order probability of failure with such principal curvatures, κ_i (see e.g., [19,30–37]). (For brevity, they are not reviewed herein, although they will be employed in the computations presented later.)

3. A proposed quasi-Newton approximation-based SORM

In our approach, two types of quasi-Newton approximation—the BFGS and SR1 algorithms—are combined for a more efficient estimation of the second order probability of failure. (Quasi-Newton methods are alternatives to "full" Newton methods, which *approximate* the Hessian matrices needed at every iteration of gradient-based optimization approaches [38].) In particular, the BFGS algorithm is used, together with the original HLRF algorithm proposed by Hasofer and Lind [22] and Rackwitz and Flessler [23], to search the design point and to compute the first order reliability index, whereas the SR1 algorithm is nested within the BFGS algorithm to approximate the Hessian matrix. Details are illustrated below.

3.1. The HLRF–BFGS algorithm to solve FORM

To compute the second order probability of failure, an efficient FORM with good convergence behavior is of interest. Recently, Periçaro et al. [16] proposed using the BFGS algorithm to search the design point in FORM. This algorithm, referred to as the HLRF–BFGS algorithm, has the advantage of incorporating the information about curvatures of the LSF, thus being more robust and efficient than other HLRF-based algorithms, particularly when finite element analyses are involved (see e.g., [16,26]). In addition, the HLRF–BFGS algorithm is as efficient as the original HLRF algorithm [22,23], since it requires just one function and gradient evaluation at each iteration; therefore requiring only $k(n+1)$ LSF evaluations (k is the number of iterations). The HLRF–BFGS algorithm employs a search direction given by

$$\mathbf{d}_k = \frac{\left[\nabla g(\mathbf{u}_{k-1})^T \mathbf{B}_{k-1}^{\mathrm{BFGS}} \mathbf{u}_{k-1} - g(\mathbf{u}_{k-1})\right] \mathbf{B}_{k-1}^{\mathrm{BFGS}} \nabla g(\mathbf{u}_{k-1})}{\nabla g(\mathbf{u}_{k-1})^T \mathbf{B}_{k-1}^{\mathrm{BFGS}} \nabla g(\mathbf{u}_{k-1})} - \mathbf{B}_{k-1}^{\mathrm{BFGS}} \mathbf{u}_{k-1} \tag{9}$$

where, for convenience, subscripts related to k are employed to indicate values of vectors or matrices at different iterations, and where $\mathbf{B}^{\mathrm{BFGS}}$ is the inverse of the Hessian matrix (i.e., $\mathbf{B}^{\mathrm{BFGS}} = (\mathbf{H}^{\mathrm{BFGS}})^{-1}$), which is approximately computed using a recursive BFGS updating formula:

$$\mathbf{B}_k^{\mathrm{BFGS}} = \mathbf{B}_{k-1}^{\mathrm{BFGS}} + \left(1 + \frac{\mathbf{q}_k^T \mathbf{B}_{k-1}^{\mathrm{BFGS}} \mathbf{q}_k}{\mathbf{p}_k^T \mathbf{q}_k}\right) \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k} - \frac{\mathbf{p}_k \mathbf{q}_k^T \mathbf{B}_{k-1}^{\mathrm{BFGS}} + \mathbf{B}_{k-1}^{\mathrm{BFGS}} \mathbf{q}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k} \tag{10}$$

where

$$\mathbf{p}_k = \mathbf{d}_k \tag{11}$$

$$\mathbf{q}_k = \mathbf{d}_k + [\nabla g(\mathbf{u}_k) - \nabla g(\mathbf{u}_{k-1})]\xi_k \tag{12}$$

and ξ_k is given by

$$\xi_k = \frac{g(\mathbf{u}_{k-1}) - \nabla g(\mathbf{u}_{k-1})^T \mathbf{B}_{k-1}^{\mathrm{BFGS}} \mathbf{u}_{k-1}}{\nabla g(\mathbf{u}_{k-1})^T \mathbf{B}_{k-1}^{\mathrm{BFGS}} \nabla g(\mathbf{u}_{k-1})} \tag{13}$$

Thus, the design point for a new iteration can be computed as

$$\mathbf{u}_k = \mathbf{u}_{k-1} + \mathbf{d}_k \tag{14}$$

so that the algorithm repeats the sequence until the following stopping criteria are satisfied:

$$1 - \frac{|\nabla g(\mathbf{u}_k)^T \mathbf{u}_k|}{\|\nabla g(\mathbf{u}_k)\| \cdot \|\mathbf{u}_k\|} < \varepsilon \quad \text{and} \quad |g(\mathbf{u}_k)| < \varepsilon \tag{15}$$

3.2. The SR1 quasi-Newton method to approximate the Hessian matrix

As shown in the previous section, the HLRF–BFGS algorithm uses the design point and the gradient vector between two successive iterations to compute the Hessian matrix approximately; then, this approximated Hessian matrix, $\mathbf{H}^{\mathrm{BFGS}}$, might be used instead of the true Hessian matrix to estimate the second order probability of failure. However, as indicated by some researchers (see e.g., [17,27,38]), the SR1 algorithm can often outperform the BFGS algorithm in providing good approximations to the true Hessian matrix. Therefore, the SR1 algorithm is used in this study to evaluate the Hessian matrix employed in second order reliability analyses. Such Hessian matrix is obtained as [17,38]:

$$\mathbf{H}_k^{\mathrm{SR1}} = \mathbf{H}_{k-1}^{\mathrm{SR1}} + \frac{(\mathbf{y}_k - \mathbf{H}_{k-1}^{\mathrm{SR1}} \mathbf{s}_k)(\mathbf{y}_k - \mathbf{H}_{k-1}^{\mathrm{SR1}} \mathbf{s}_k)^T}{(\mathbf{y}_k - \mathbf{H}_{k-1}^{\mathrm{SR1}} \mathbf{s}_k)^T \mathbf{s}_k} \tag{16}$$

where

$$\mathbf{s}_k = \mathbf{d}_k \tag{17}$$

$$\mathbf{y}_k = \nabla g(\mathbf{u}_k) - \nabla g(\mathbf{u}_{k-1}) \tag{18}$$

To preserve the numerical stability of the SR1 algorithm, a safeguard is required; it is given by [17,38]

$$|\mathbf{s}_k^T(\mathbf{y}_k - \mathbf{H}_{k-1}^{\mathrm{SR1}} \mathbf{s}_k)| > \eta \|\mathbf{s}_k\| \, \|\mathbf{y}_k - \mathbf{H}_{k-1}^{\mathrm{SR1}} \mathbf{s}_k\| \tag{19}$$

where η is a very small positive number (e.g., $\eta = 10^{-6} \sim 10^{-8}$). The SR1 update is performed only if Eq. (19) holds; otherwise the update is skipped (i.e., $\mathbf{H}_k^{\mathrm{SR1}} = \mathbf{H}_{k-1}^{\mathrm{SR1}}$).

3.3. Implementation procedure

To facilitate the understanding of the quasi-Newton approximation-based SORM, and to promote its wider future use, its implementation is detailed below (see also the flowchart in Fig. 1):

(1) $k = 0$.
(2) Apply initial guesses to $\mathbf{u}$ and $\mathbf{B}^{\mathrm{BFGS}}$ (e.g., using $\mathbf{u}_0 = \mathbf{0}$ and $\mathbf{B}_0^{\mathrm{BFGS}} = \mathbf{I}$). Compute the gradient vector, $\nabla g(\mathbf{u}_0)$, using a forward difference scheme.
(3) Make $k = k + 1$. Compute the search direction vector, $\mathbf{d}_k$, and the coefficient, ξ_k, with Eqs. (9) and (13), respectively.
(4) Evaluate the new design point, $\mathbf{u}_k$, using Eq. (14); and the new gradient vector, $\nabla g(\mathbf{u}_k)$, using Eq. (4).
(5) Compute $\mathbf{p}_k$, $\mathbf{q}_k$, $\mathbf{s}_k$ and $\mathbf{y}_k$ using Eqs. (11), (12), (17) and (18).
(6) If $k = 1$, apply an initial scaling of $\mathbf{S}_1^T\mathbf{Y}_1/\mathbf{S}_1^T\mathbf{S}_1\mathbf{I}$ to $\mathbf{H}_0^{\mathrm{SR1}}$; otherwise, skip. (A detailed discussion of the scaling needed to make the SR1 algorithm more robust is outside the scope of this paper; for details and further discussion, see e.g., [39].)
(7) Compute the new $\mathbf{B}_k^{\mathrm{BFGS}}$ with Eq. (10).
(8) Evaluate the safe-guard in Eq. (19): if it holds, update the Hessian matrix, $\mathbf{H}_k^{\mathrm{SR1}}$, using Eq. (16); otherwise, make $\mathbf{H}_k^{\mathrm{SR1}} = \mathbf{H}_{k-1}^{\mathrm{SR1}}$.

Repeat Steps 2–7 until the stopping criteria in Eq. (15) are both achieved. Then, the first order reliability index, β_{FORM}, can be easily computed as $\beta_{\mathrm{FORM}} = \|\mathbf{u}_k\|$. And, consequently, the second order probability of failure can be estimated using the SORM methods proposed in the references cited in Section 2.3, which are all based on the information obtained above ($\mathbf{u}_k$, β_{FORM}, $\nabla g(\mathbf{u}_k)$ and $\mathbf{H}_k^{\mathrm{SR1}}$).

The reader should note that, except for evaluations required by the HLRF–BFGS algorithm, no additional LSF evaluation is needed in the quasi-Newton approximation-based SORM. Therefore, only $k(n+1)$ LSF evaluations are needed in our proposed method, instead of the $k(n+1) + n_{\mathrm{add}} + n(n+1)/2$ evaluations needed in traditional SORM (n_{add} is the total number of additional LSF evaluations needed to compute the merit function and to select the step size in the iHLRF algorithm).

4. Case studies

The reliability of three typical geotechnical problems—settlement of a rectangular foundation, bearing capacity of a shallow footing, and stability of a layered soil slope—are considered in this study as benchmark tests of the proposed method, in which the iHLRF algorithm, the HLRF–BFGS algorithm, and conventional SORM are employed for comparison with our proposed method. The same convergence criteria given in Eq. (15), with $\varepsilon = 0.001$,

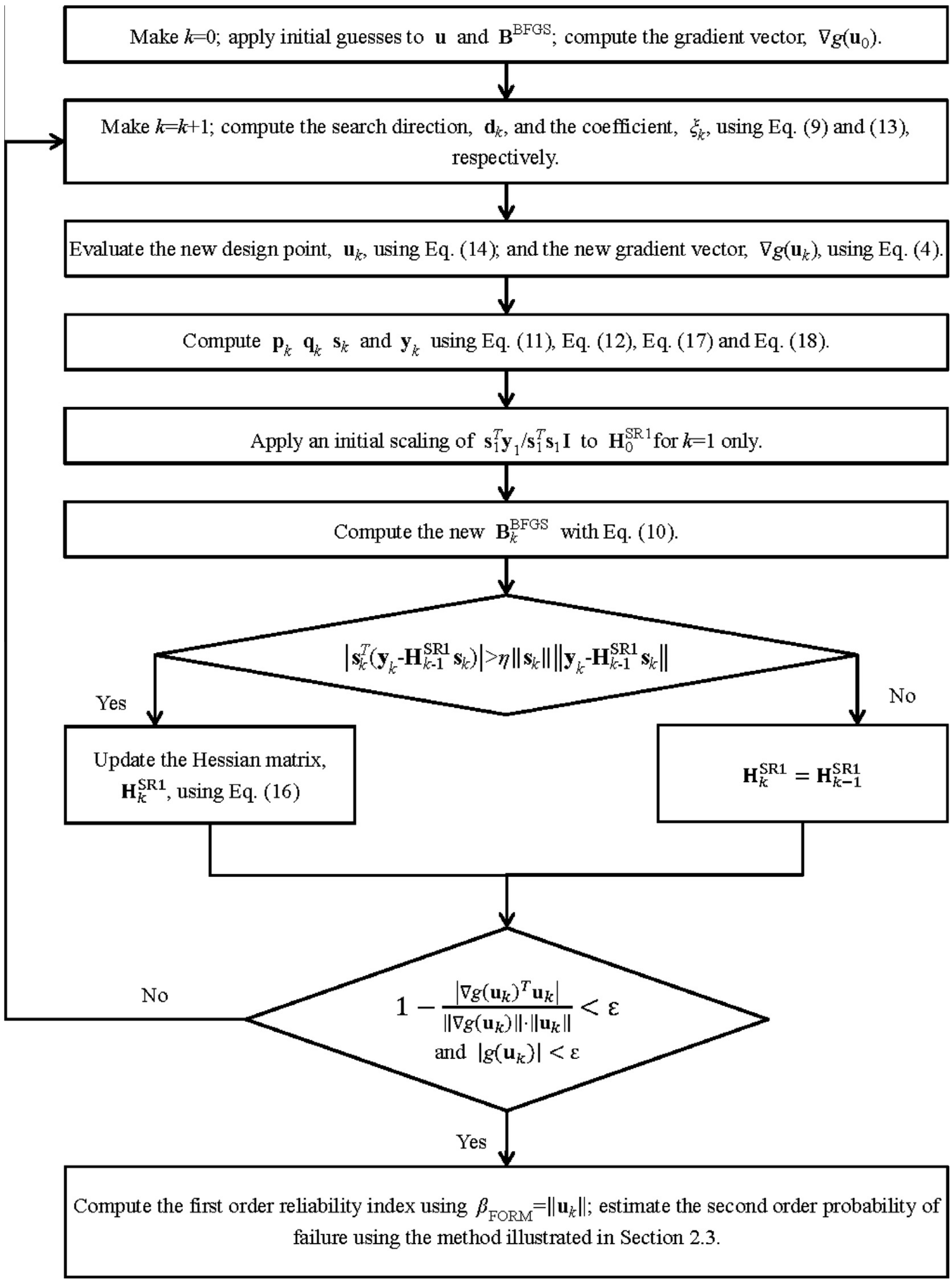


Fig. 1. A flowchart to illustrate the implementation procedure of quasi-Newton approximation-based SORM.

are used for these four methods to make results comparable. Note that lack of information about convergence criteria prevents us from being able to compare results with other results published in the literature, as efficiency and accuracy depend to a large extent on such criteria; instead, we use our own implementations of the algorithms discussed (with results computed for the same convergence criteria), as well as the results of Monte Carlo simulations (MCS) as the 'reference' or 'exact' solutions.

To estimate the second order probability of failure, the average of seven formulas based on the same information (i.e., β_{FORM} and κ_i) is used in this study. The formulas are those proposed by Tvedt [30], Breitung [31], Hohenbichler and Rackwitz [32], Hong [35] (who provides two formulas), Zhao and Ono [36] and Phoon [19].

To measure computational efficiency, we use the number of deterministic LSF evaluations (FEs) required in each analysis. This is because the computational effort demanded by other parts of the algorithm is often negligible when compared to that needed by LSF evaluations, particularly when expensive numerical methods (such as finite elements or finite differences) are involved. Thus, the number of FEs can be used as a general indicator of the computational efficiency in real problems: a larger number of FEs indicates less efficiency, and vice versa.

4.1. Immediate settlement of a rectangular foundation

We start with a case considering the immediate settlement of a flexible rectangular foundation; see Fig. 2. This case was previously studied by Chan and Low [14], using a point-fitted paraboloid-based SORM. The settlement ΔH of a flexible rectangular foundation can be computed as [14]:

$$\Delta H = 0.5Bq_{\circ}\frac{1-\nu^2}{E_s}m\left(I_1 + \frac{1-2\nu}{1-\nu}I_2\right)I_F \tag{20}$$

Magnitudes used in Eq. (20) are illustrated in Fig. 2 together with their prescribed values. Then, the LSF for a limiting settlement of $(\Delta H)_{\text{limit}} = 50$ mm can be written as

$$G(\mathbf{x}) = (\Delta H)_{\text{limit}} - \Delta H \tag{21}$$

Three random variables (i.e., contact stress q_o, Poisson's ratio ν and elastic modulus E_s) are considered in this study. They are assumed to be independent and normally distributed; their means and standard deviations are listed in Fig. 2. Further details are available in Chan and Low [14].

Table 1 lists the number of LSF evaluations and the reliability results computed by different methods, including MCS with 500,000 simulations. The two FORM algorithms (iHLRF and HLRF–BFGS) provide identical probabilities of failure, with a slight relative error of 13.1% with respect to the MCS result, hence implying that the LSF is slightly non-linear. P_f estimates improve after considering the curvatures of the LSF at the vicinity of the design point, and both conventional SORM and the method proposed in this study approximate well to the MCS result, with differences of only 1.2% and 1.5%, respectively.

Regarding the computational cost, the proposed method needs only 16 FEs, which is the least among all the methods considered. Such good efficiency is due to two aspects: one is that the HLRF–BFGS algorithm outperforms the iHLRF algorithm in finding the design point; the other is that the FEs needed to compute the Hessian matrix in conventional SORM are not needed with our proposed method. In summary, the proposed method provides a very similar P_f result to conventional SORM, but with approximately 40% fewer FEs.

Table 2 compares the Hessian matrices evaluated using the traditional forward difference scheme and the SR1 algorithm, respectively, showing that their elements agree well with each other. This explains the good performance of our proposed method.

Table 1
Computed reliability results for the rectangular foundation example.

$n = 3$ random variables		FE[a]	β	P_f	Δ (%)[b]
FORM	iHLRF	20[c]	1.237	0.1081	13.1
	HLRF–BFGS	16[c]	1.237	0.1081	13.1
SORM	Conventional	20[c] + 6[d]	1.301	0.0967	1.2
	This study	16[c] + 0[d]	1.299	0.0970	1.5
MCS		500,000	1.307	0.0956	–

[a] FE = Number of deterministic function evaluations.
[b] Δ = Relative error in relation to MCS, computed based on the results of P_f.
[c] Number of FE required by FORM.
[d] Number of FE required for computing Hessian matrix.

Table 2
Hessian matrices computed by different methods for the rectangular foundation example.

Forward difference scheme			SR1 algorithm		
0.000	0.689	0.321	−0.025	0.703	0.342
0.689	1.178	−0.276	0.703	1.168	−0.289
0.321	−0.276	−0.255	0.342	−0.289	−0.307

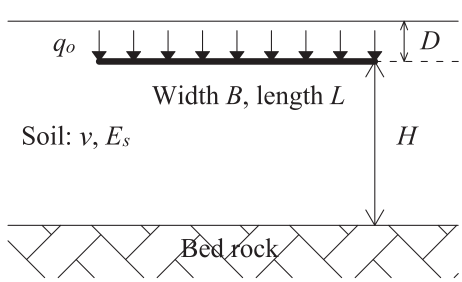
4.2. Bearing capacity of a shallow footing

The second example case, also proposed by Chan and Low [14], considers the bearing capacity of a shallow footing resting on an homogeneous silty sand; see Fig. 3. The LSF due to exceedance of its bearing capacity is given as

$$G(\mathbf{x}) = q_{\text{ult}} - q \tag{22}$$

where q_{ult} is the vertical bearing resistance computed with the well-known polynomial bearing capacity equation; and q is the vertical applied pressure (corrected to account for the excentricity of the loads). Equations to compute q_{ult} and q are summarized in Appendix A, and more details about them can be found in [14]. Five random variables—cohesion, c'; friction angle, φ'; unit weight, γ; horizontal load, P_H; and vertical load, P_V—are considered, and they are all assumed to be normally distributed. Their moments and correlation structure are shown in Fig. 3, together with the deterministic parameters involved.

Table 3 presents the number of LSF evaluations and the reliability results computed using the five different reliability methods



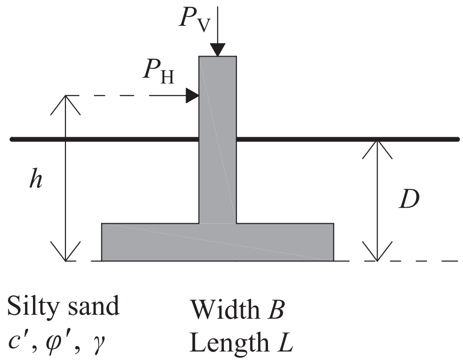
Random variables

	Mean	Standard deviation
q_o (kPa)	280	40
ν	0.25	0.08
E_s (MPa)	50	2.5

Deterministic parameters

Width	B (m)	30
Length	L (m)	40
Embedment depth	D (m)	3
Stratum thickness	H (m)	10
Number of corners	m	4
Influence factor	I_1	0.073
Influence factor	I_2	0.089
Influence factor	I_F	0.95

Fig. 2. Description of the rectangular foundation and of the parameters and variables involved.



Deterministic parameters

Width	B (m)	5
Length	L (m)	25
Depth	D (m)	1.8
Load position	h (m)	2.5

Random variables

	Mean	Standard deviation
c' (kPa)	15	4.5
φ' (°)	25	5
γ (kN/m^3)	20	2
P_H (kN/m)	400	40
P_V (kN/m)	800	80

Correlation matrix

c'	φ'	γ	P_H	P_V
1	-0.5	0	0	0
-0.5	1	0.5	0	0
0	0.5	1	0	0
0	0	0	1	0.5
0	0	0	0.5	1

Fig. 3. Description of the shallow footing example and of the parameters and variables involved.

Table 3
Computed reliability results for the shallow footing example.

$n = 3$ random variables		FE[a]	β	$P_f(\times 10^{-2})$	Δ (%)[b]
FORM	iHLRF	42[c]	1.641	5.04	−18.3
	HLRF–BFGS	42[c]	1.641	5.04	−18.3
SORM	Conventional	42[c] + 15[d]	1.569	5.83	−5.5
	This study	42[c] + 0[d]	1.571	5.81	−5.8
MCS		500,000	1.541	6.17	–

[a] FE = Number of deterministic function evaluations.
[b] Δ = Relative error in relation to MCS, computed based on the results of P_f.
[c] Number of FE required by FORM.
[d] Number of FE required for computing Hessian matrix.

Table 4
Hessian matrices computed using different algorithms for the shallow footing example.

Forward difference formula				
−10.913	−10.612	−3.250	4.372	−4.618
−10.612	66.687	7.804	−8.902	9.211
−3.250	7.804	0.000	−0.817	0.854
4.372	−8.902	−0.817	−1.246	4.049
−4.618	9.211	0.854	4.049	−9.083
SR1 algorithm				
−11.446	−10.085	−3.590	4.827	−5.128
−10.085	64.722	7.733	−8.948	9.075
−3.590	7.733	0.581	−0.703	1.003
4.827	−8.948	−0.703	−1.168	3.487
−5.128	9.075	1.003	3.487	−8.381

considered. In this case, both FORM methods give consistent reliability results with the same computational effort, with an error of −18.3% with respect to MCS; whereas conventional SORM and the proposed method improve such estimates, with relative errors of −5.5% and −5.8%, respectively. Moreover, the proposed method only requires the same number of LSF evaluations as FORM, which is 15 (or approx. 30%) less than conventional SORM.

We compared the Hessian matrix obtained with the two SORM methods (see Table 4). Again, it is found that they are similar to each other, although larger differences are observed than in Case 1; note, however, that such larger differences have still a very limited influence on the computed probabilities of failure.

4.3. Stability of a layered soil slope

Next, a 4-layer soil slope example originally proposed by Zolfaghari et al. [40], that was recently analyzed by Zeng et al. [15] to identify its probabilistic representative slip surfaces (RSSs), is employed to extend our tests of the methods under discussion. Fig. 4 shows the geometry of the slope, characterized by an inclined planar and weak seam, and the unit weights of the soil layers. Eight strength parameters are considered as random variables with lognormal distributions, as shown in Table 5. The cohesion, c, and friction angle, φ, of each layer are assumed to be negatively correlated, and the strength parameters of one layer are assumed to be independent of those of the other layers (i.e., $\rho_{c_i,\varphi_i} = -0.5$ and $\rho_{c_i,c_j} = \rho_{\varphi_i,\varphi_j} = \rho_{c_i,\varphi_j} = 0$ for $i,j = 1,\ldots,4$ and $i \neq j$).

Fig. 4 also shows the 6 'best' RSSs identified by Zeng et al. [15] for this slope. (Exact coordinates of their nodes can be found in Table A.3 of [15].) The LSF corresponding to each RSS is given by $G(\mathbf{x}) = FS(\mathbf{x}) - 1$, where FS is the factor of safety for that specific slip surface. Spencer's method, as implemented in SLOPE8R [41] with minor modifications to improve convergence, is used to compute such FS values. Reliability results computed by the different methods considered herein for the first 4 LSFs (i.e., those associated to RSSs 1–4) are listed in Table 6. (RSSs 5 and 6 in [15] are not considered because they have very small probabilities of failure that have an almost negligible contribution to the overall probability of failure; moreover, the number of Monte Carlo simulations required to achieve accurate probabilities of failure for them are too large.)

Results show that FORM produces significant errors for LSFs 1–4 (from about 80% to 490%), suggesting that LSFs 1–4 are all highly

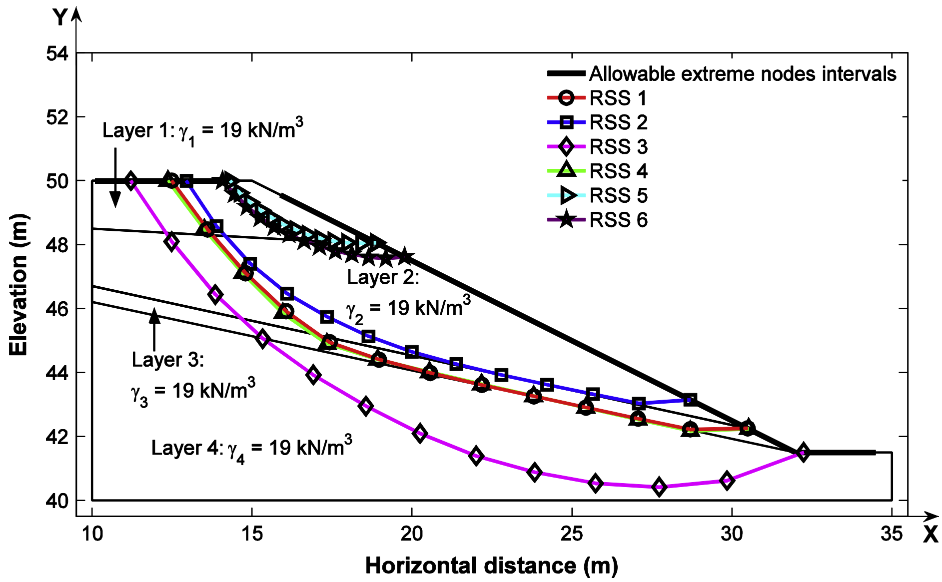


Fig. 4. Geometry of layered soil slope and RSSs identified in [15].

Table 5
Statistical parameters of random variables considered for the slope stability example.

Layer	Random variable	Mean value	Standard deviation	Distribution type
1	c_1 (kPa)	18	9.0	Lognormal
	φ_1 (°)	16	4.8	Lognormal
2	c_2 (kPa)	20	10.0	Lognormal
	φ_2 (°)	14	4.2	Lognormal
3	c_3 (kPa)	12	3.6	Lognormal
	φ_3 (°)	10	2.0	Lognormal
4	c_4 (kPa)	20	10.0	Lognormal
	φ_4 (°)	18	5.4	Lognormal

non-linear. SORM solutions outperform FORM in this case, providing probabilities of failure that compare better to MCS results. Note that the error of the proposed method for LSF 4 is relatively large (84.1%), although it is still significantly better than the FORM results (478.7% for the iHLRF algorithm and 489.6% for the HLRF–BGFS algorithm). Note also that relative errors computed using reliability indices instead of probabilities of failure would be much smaller.

Regarding the computational cost, the HLRF–BGFS algorithm can sometimes significantly reduce the number of function evaluations required by the iHLRF algorithm to compute FORM solutions. Similarly, the proposed SORM method, which always requires the same number of FEs as the HLRF–BGFS algorithm, requires significantly fewer FEs than traditional SORM (between approx. 30% and 60%, for the RSSs considered), and that the computational savings tend to increase as the number of random variables involved, n, increases. This suggests that the proposed method could represent a significant advantage when the LSFs are computationally expensive or when the number of random variables are large.

Table 6
Computed reliability results for the slope stability example.

		FE[a]	β	P_f	Δ (%)[b]
LSF 1 (RSS 1)					
$n = 6$ random variables		FE[a]	β	$P_f(\times 10^{-3})$	Δ (%)[b]
FORM	iHLRF	35[c]	2.510	6.03	131.0
	HLRF–BFGS	35[c]	2.511	6.02	130.7
SORM	Conventional	35[c] + 21[d]	2.786	2.67	2.29
	This study	35[c] + 0[d]	2.772	2.78	6.51
MCS		100,000	2.793	2.61	–
LSF 2 (RSS 2)					
$n = 4$ random variables		FE[a]	β	$P_f(\times 10^{-3})$	Δ (%)[b]
FORM	iHLRF	25[c]	2.852	2.17	77.9
	HLRF-BFGS	25[c]	2.848	2.20	80.3
SORM	Conventional	25[c] + 10[d]	3.041	1.18	−3.3
	This study	25[c] + 0[d]	3.066	1.09	−10.7
MCS		100,000	3.031	1.22	–
LSF 3 (RSS 3)					
$n = 8$ random variables		FE[a]	β	$P_f(\times 10^{-4})$	Δ (%)[b]
FORM	iHLRF	107[c]	3.129	8.76	199.0
	HLRF–BFGS	54[c]	3.131	8.72	197.6
SORM	Conventional	107[c] + 36[d]	3.387	3.53	20.5
	This study	54[c] + 0[d]	3.409	3.26	11.3
MCS		1,000,000	3.438	2.93	–
LSF 4 (RSS 4)					
$n = 8$ random variables		FE[a]	β	$P_f(\times 10^{-5})$	Δ (%)[b]
FORM	iHLRF	55[c]	3.732	9.49	478.7
	HLRF-BFGS	45[c]	3.728	9.67	489.6
SORM	Conventional	55[c] + 36[d]	4.109	1.99	21.3
	This study	45[c] + 0[d]	4.011	3.02	84.1
MCS		2,500,000	4.153	1.64	–

[a] FE = Number of deterministic function evaluations.
[b] Δ = Relative error in relation to MCS, computed based on the results of P_f.
[c] Number of FE required by FORM.
[d] Number of FE required for computing Hessian matrix.

5. Summary and conclusions

A new method has been proposed in this study for an efficient estimation of the second order probability of failure of geotechnical problems. To reduce the computational cost, the method builds on both the HLRF–BGFS and SR1 algorithms; in particular, instead of using the Hessian matrix computed with a forward difference scheme that is traditionally employed in SORM, it uses good approximations to the Hessian matrix provided by the SR1 algorithm, and it incorporates them to the HLRF–BGFS algorithm. (Such good approximations to the Hessian matrix are responsible, to a large extent, of the improved performance observed.) In addition, as SORM builds on results (design point and gradient information) provided by FORM, an efficient and robust FORM algorithm is of interest to successfully implement the second order reliability analysis, and that is why we use the HLRF–BGFS algorithm—an algorithm that incorporates information about curvatures of the LSFs, hence being more robust and efficient than other HLRF algorithms—to compute FORM solutions.

Three example cases of common geotechnical engineering problems—settlement of a rectangular foundation, bearing capacity of a shallow footing, and stability of a layered soil slope—have been employed to test the proposed method, and Monte Carlo simulation results have been employed as the 'reference' or 'exact' solution.

Results show that, for the cases considered in this study, the proposed method can often provide probability of failure estimates which are as good as those of conventional SORM, but with a reduced computational effort that is indeed equal to that required by HLRF–BGFS-based FORM. In particular, for most of the cases studied in this paper, probabilities of failure computed by the proposed method are found to be close to those of conventional SORM; and the absolute values of their relative errors with respect to MCS results are normally less than 12%, which is considered acceptable in engineering practice. (Although the error with respect to MCS is relatively large (84.1%) for LSF 4 in the layered soil slope example, the proposed method still improves the FORM results significantly; and note that relative errors computed using the reliability indices would be much smaller.) More importantly, the proposed method requires a reduced computational effort, as the SR1 algorithm approximates the Hessian matrix with only $k(n+1)$ LSF evaluations, a value which, for large n, can be significantly smaller than the $k(n+1)+n_{\text{add}}+n(n+1)/2$ LSF evaluations required by conventional SORM. Results also suggest that the HLRF–BGFS algorithm can often identify the design point with less computational effort than the iHLRF algorithm, particularly when complex LSFs are involved. This contributes further to the efficiency of the proposed approach.

Finally, from a practical viewpoint, the reliability results computed by the proposed method can also be used as an indicator of the non-linearity of the LSF involved. As pointed out by Rackwitz [42], the first order approximation can be adequate for 90% of all practical applications; however, when MCS are not available or cannot be obtained due to their computational cost, it is difficult to predict when the non-linearity of the LSF corresponds to one case belonging to the remaining 10%. In this circumstance, the proposed method provides an interesting tool, as the first- and second-order probabilities of failure can be obtained simultaneously and without additional LSF evaluations. Then, both results can be compared: if they coincide, the LSF can be considered as linear in practice; otherwise, the LSF would be non-linear (with its non-linearity increasing with an increasing difference between both probabilities of failure) and adequate tools to deal with such non-linearity—such as a SORM method that compute better Hessian matrix, or simulation methods—can be employed.

Conflict of interest

This work does not have any conflict of interest that needs to be reported.

Acknowledgments

This research was partially supported by the Spanish Ministry of Economy and Competitiveness under Grant BIA2015-69152R and the State Key Laboratory of Geohazard Prevention and Geoenvironment Protection (Grant Nos. SKLGP2011Z002 and SKLGP2015Z016). The first author was supported by China Scholarship Council (CSC) and, for insurance coverage, by Fundación José Entrecanales Ibarra. Dr. Solange Regina and Dr. Gislaine Periçaro kindly provided their MATLAB codes for the HLRF–BGFS algorithm. Their support is greatly appreciated.

Appendix A

An analytical method to compute the vertical bearing resistance, q_{ult}, of a shallow footing in a uniform sand deposit subjected to a vertical load, P_{V}, and also to a horizontal load that produces moment (with P_{H} being the horizontal load and h being the height with which it acts in relation to the foundation plane) was summarized by Chan and Low [14] based on the Annex D of Eurocode 7.

Following [14], q_{ult}, can be computed as

$$q_{\text{ult}} = c'N_c s_c i_c + \gamma D N_q s_q i_q + 0.5\gamma B' N_\gamma s_\gamma i_\gamma \quad \text{(A.1)}$$

where N_q, N_c and N_γ are the traditional dimensionless bearing resistance factors that depend (strongly non-linearly) on the soil's friction angle, as

$$N_q = e^{\pi \tan\varphi'} \tan^2(45^\circ + \varphi'/2) \quad \text{(A.2)}$$

$$N_c = (N_q - 1)\cot\varphi' \quad \text{(A.3)}$$

$$N_\gamma = 2(N_q - 1)\tan\varphi' \quad \text{(A.4)}$$

and where s_q, s_c, s_γ are dimensionless factors that introduce corrections to account for the shape of the footing (B and L are the width and length of the footing) and the excentricity of the loads. They can be computed as

$$s_q = 1 + (B'/L')\sin\varphi' \quad \text{(A.5)}$$

$$s_c = (s_q N_q - 1)/(N_q - 1) \quad \text{(A.6)}$$

$$s_\gamma = 1 - 0.3(B'/L') \quad \text{(A.7)}$$

where $B' = B - 2e_B$, $L' = L$, and $e_B = P_{\text{H}} \times h/P_{\text{V}}$. Similarly, i_q, i_c and i_γ are dimensionless correction factors to account for the inclination of the resultant load which can be computed as

$$i_q = \left(1 - \frac{P_{\text{H}}}{P_{\text{V}} + B'L'c'\cot\varphi'}\right)^m \quad \text{(A.8)}$$

$$i_c = i_q - \frac{1 - i_q}{N_c \tan\varphi'} \quad \text{(A.9)}$$

$$i_\gamma = \left(1 - \frac{P_{\text{H}}}{P_{\text{V}} + B'L'c'\cot\varphi'}\right)^{m+1} \quad \text{(A.10)}$$

where c' is the cohesion of the soil and m can be written as

$$m = \frac{2 + B'/L'}{1 + B'/L'} \quad \text{(A.11)}$$

Using this approach, the bearing capacity failure would be theoretically exceeded when the vertical applied pressure, q, which can be computed as

$$q = P_{\text{V}}/B' \quad \text{(A.12)}$$

becomes greater than the value of q_{ult} computed using Eq. (A.1).

Appendix B

List of symbols and acronyms

Symbol	Description
General	
FORM	First order reliability method
SORM	Second order reliability method
BFGS	Broyden–Fletcher–Goldfarb–Shanno
SR1	Symmetric rank-one
HLRF	Hasofer–Lind–Rackwitz–Fiessler
LSF	Limit state function
RSM	Response surface method
ANN	Artificial neural network
$\mathbf{X}$	A vector of random variables in physical space
$\mathbf{U}$	A vector of uncorrelated standard normal random variables
β	Reliability index
P_f	Probability of failure
iHLRF	Improved Hasofer–Lind–Rackwitz–Fiessler
$\nabla g(\cdot)$	Gradient vector
Δh	Step size
$\mathbf{H}$	Hessian matrix
$\mathbf{u}^*$	Design point
n	Number of random variables
$\mathbf{V}$	Random variables in V-space
$\mathbf{P}$	Orthogonal rotation matrix
$\boldsymbol{\alpha}$	Unit design point vector
$\mathbf{H}_{\mathrm{rot}}$	Rotated diagonal matrix
κ_i	Principal curvatures
k	Number of iterations
$\mathbf{d}_k$	Search direction
$\mathbf{B}$	Inverse of the Hessian matrix
$\mathbf{p}_k$, $\mathbf{q}_k$, ξ_k	Variables used for BFGS updating
ε	Stopping criterion
$\mathbf{s}_k$, $\mathbf{y}_k$	Variables used for SR1 updating
T	(as superscript) transpose operator
η	A very small positive number
$\mathbf{I}$	Identity matrix
n_{add}	Total number of additional LSF evaluations needed to compute the merit function and to select the step size in the iHLRF algorithm
MCS	Monte Carlo simulation
FEs	Number of deterministic LSF evaluations
Rectangular foundation example	
ΔH	Settlement of a flexible rectangular foundation
$(\Delta H)_{\mathrm{limit}}$	Limiting settlement
q_{o}	Contact stress
ν	Poisson's ratio
E_{s}	Elastic modulus
B	Width of footing
L	Length of footing
D	Embedment depth
H	Stratum thickness
m	Number of corners
I_1, I_2, I_F	Influence factors
Shallow footing example	
q_{ult}	Vertical ultimate bearing resistance
q	Vertical (equivalent) applied pressure
N_q, N_c, N_γ	Dimensionless factors for the bearing resistance
φ'	Friction angle of soil
s_q, s_c, s_γ	Dimensionless shape correction factors
B	Width of the footing
L	Length of the footing
P_{H}, P_{V}	Horizontal load and vertical load
h	Position of horizontal load
i_q, i_c, i_γ	Dimensionless correction factors to account for load inclination
c'	Cohesion of soil
D	Depth of foundation
γ	Unit weight of soil
Layered soil slope example	
c	Cohesion
φ	Friction angle
ρ	Correlation coefficient
RSS	Representative slip surface
FS	Factor of safety
γ	Unit weight

References

[1] Duncan J. Factors of safety and reliability in geotechnical engineering. J Geotech Geoenviron Eng 2000;126(4):307–16.
[2] Lü Q, Chan CL, Low BK. Probabilistic evaluation of ground-support interaction for deep rock excavation using artificial neural network and uniform design. Tunn Undergr Sp Tech 2012;32:1–18.
[3] Zeng P, Senent S, Jimenez R. Reliability analysis of circular tunnel face stability obeying Hoek–Brown failure criterion considering different distribution types and correlation structures. J Comput Civ Eng 2014. http://dx.doi.org/10.1061/(ASCE)CP.943-5487.0000464.
[4] Jimenez-Rodriguez R, Sitar N, Chacón J. System reliability approach to rock slope stability. Int J Rock Mech Min 2006;43(6):847–59.
[5] Low BK, Zhang J, Tang WH. Efficient system reliability analysis illustrated for a retaining wall and a soil slope. Comput Geotech 2011;38(2):196–204.
[6] Cho SE. First-order reliability analysis of slope considering multiple failure modes. Eng Geol 2013;154:98–105.
[7] Low BK. FORM, SORM, and spatial modeling in geotechnical engineering. Struct Saf 2014;49:56–64.
[8] Zeng P, Jimenez R. An approximation to the reliability of series geotechnical systems using a linearization approach. Comput Geotech 2014;62:304–9.
[9] Brzakała W, Puła W. A probabilistic analysis of foundation settlements. Comput Geotech 1996;18(4):291–309.
[10] Bauer J, Puła W. Reliability with respect to settlement limit-states of shallow foundations on linearly-deformable subsoil. Comput Geotech 2000;26(3–4):281–308.
[11] Cho SE. Probabilistic stability analyses of slopes using the ANN-based response surface. Comput Geotech 2009;36(5):787–97.
[12] Lü Q, Low BK. Probabilistic analysis of underground rock excavations using response surface method and SORM. Comput Geotech 2011;38(8):1008–21.
[13] Lü Q, Sun H-Y, Low BK. Reliability analysis of ground–support interaction in circular tunnels using the response surface method. Int J Rock Mech Min 2011;48(8):1329–43.
[14] Chan CL, Low BK. Practical second-order reliability analysis applied to foundation engineering. Int J Numer Anal Meth Geomech 2012;36(11):1387–409.
[15] Zeng P, Jimenez R, Jurado-Piña R. System reliability analysis of layered soil slopes using fully specified slip surfaces and genetic algorithms. Eng Geol 2015;193(2):106–17.
[16] Periçaro GA, Santos SR, Ribeiro AA, Matioli LC. HLRF–BFGS optimization algorithm for structural reliability. Appl Math Model 2015;39(7):2025–35.
[17] Lim J, Lee B, Lee I. Second-order reliability method-based inverse reliability analysis using Hessian update for accurate and efficient reliability-based design optimization. Int J Numer Meth Eng 2014;100(10):773–92.
[18] Ditlevsen O, Madsen HO. Structural Reliability Methods: Coastal, Maritime and Structural Engineering Department of Mechanical Engineering Technical University of Denmark, 2007.
[19] Phoon KK. In: Phoon KK, editor. Reliability-based design in geotechnical engineering: computations and applications. New York: Taylor & Francis; 2008.

[20] Low B, Tang W. Efficient reliability evaluation using spreadsheet. J Eng Mech 1997;123(7):749–52.
[21] Low B, Tang W. Efficient spreadsheet algorithm for first-order reliability method. J Eng Mech 2007;133(12):1378–87.
[22] Hasofer AM, Lind NC. Exact and invariant second moment code format. J Eng Mech Div 1974;100(1):111–21.
[23] Rackwitz R, Flessler B. Structural reliability under combined random load sequences. Comput Struct 1978;9(5):489–94.
[24] Liu PL, Der Kiureghian A. Optimization algorithms for structural reliability. Struct Saf 1991;9(3):161–77.
[25] Zhang Y, Der Kiureghian A. Two improved algorithms for reliability analysis. In: Rackwitz R, Augusti G, Borri A, editors. Reliability and optimization of structural systems. Springer; 1995. p. 297–304.
[26] Santos SR, Matioli LC, Beck AT. New optimization algorithms for structural reliability analysis. Comput Model Eng Sci 2012;83(1):23–56.
[27] Michael E, Barron B. Second-order reliability formulations in DAKOTA/UQ. In: 47th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference. American Institute of Aeronautics and Astronautics; 2006.
[28] Barbato M, Gu Q, Conte J. New multidimensional visualization technique for limit-state surfaces in nonlinear finite-element reliability analysis. J Eng Mech 2010;136(11):1390–400.
[29] Haldar A, Mahadevan S. Probability, reliability, and statistical methods in engineering design. John Wiley; 2000.
[30] Tvedt L. Two second-order approximations to the failure probability. Veritas report RDIV/20-004083. Oslo; 1983.
[31] Breitung K. Asymptotic approximations for multinormal integrals. J Eng Mech 1984;110(3):357–66.
[32] Hohenbichler M, Rackwitz R. Improvement of second order reliability estimates by importance sampling. J Eng Mech 1988;114(12):2195–9.
[33] Cai GQ, Elishakoff I. Refined second-order reliability analysis. Struct Saf 1994;14(4):267–76.
[34] Köylüoğlu HU, Nielsen SR. New approximations for SORM integrals. Struct Saf 1994;13(4):235–46.
[35] Hong H. Simple approximations for improving second-order reliability estimates. J Eng Mech 1999;125(5):592–5.
[36] Zhao Y, Ono T. New approximations for SORM: Part 1. J Eng Mech 1999;125(1):79–85.
[37] Zhao Y, Ono T. New approximations for SORM: Part 2. J Eng Mech 1999;125(1):86–93.
[38] Nocedal J, Wright S. Numerical optimization. New York: Springer; 2006.
[39] Sun L. Updating the self-scaling symmetric rank one algorithm with limited memory for large-scale unconstrained optimization. Comput Optim Appl 2004;27(1):23–9.
[40] Zolfaghari AR, Heath AC, McCombie PF. Simple genetic algorithm search for critical non-circular failure surface in slope stability analysis. Comput Geotech 2005;32(3):139–52.
[41] Duncan JM, Song KS. SLOPE8R: a computer program for slope stability analysis with non-circular slip surfaces. Technical report, Berkeley (CA): Department of Civil Engineering, University of California; 1984.
[42] Rackwitz R. Reliability analysis—a review and some perspectives. Struct Saf 2001;23(4):365–95.